A neural network-based methodology for inverse kinematics of a multi-finger robotic hand for gripping

Abhijit Das

Surface Robotics Lab,
Central Mechanical Engineering Research Institute,
Durgapur, 721309, India
Email: abhijit.random739@gmail.com

Sankha Deb*

FMS and Computer Integrated Manufacturing Lab,
Department of Mechanical Engineering,
IIT Kharagpur, Kharagpur, 721302, India
Email: sankha.deb@mech.iitkgp.ernet.in
*Corresponding author

Abstract: Robotic grasping and manipulation require controlling the gripper movement through different points in its work volume, necessitating inverse kinematics computations to determine joint angles. In the present work, a novel methodology, based on a radial basis function neural network, has been proposed for the inverse kinematics solution and a genetic algorithm-based approach for optimising the neural network parameters. Instead of taking the entire work volume of the hand for neural network training, a subspace of points is created in close vicinity of the given destination point. The joint variables corresponding to a destination point are obtained using a random walk algorithm that uses the forward kinematics model of the hand. Then, the subspace of points and the corresponding joint variables obtained above are used to train the neural network. This approach can provide an approximate yet fairly quick and effective solution to the inverse kinematics problem of multi-finger robot hands.

Keywords: multi-finger robot gripper; multi-finger grasping; inverse kinematics; artificial neural network; genetic algorithm.

Reference to this paper should be made as follows: Das, A. and Deb, S. (2016) 'A neural network-based methodology for inverse kinematics of a multi-finger robotic hand for gripping', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 15, No. 4, pp.281–294.

Biographical notes: Abhijit Das is currently a Junior Scientist in the Surface Robotics Lab of Central Mechanical Engineering Research Institute, Durgapur, India. He obtained his Bachelor's degree and Master's degree in Mechanical Engineering from National Institute of Technology (NIT) Durgapur, India in 2000 and 2002, respectively. He is pursuing his PhD in Indian Institute of Technology Kharagpur in the area of robotics.

Sankha Deb is currently an Assistant Professor of Mechanical Engineering at Indian Institute of Technology Kharagpur. He obtained his PhD in Industrial Engineering from Ecole Polytechnique Montreal, Canada and Master's degree in Manufacturing Process Engineering from Indian Institute of Technology Kharagpur. He earlier served as an Assistant Professor at Indian Statistical Institute Calcutta and at Indian Institute of Technology Guwahati. He was invited as a Visiting Professor in University of Montreal. He published many research papers and co-authored one book on robotics technology and flexible automation. His research interests are Computer Integrated Manufacturing, automation and robotics, and soft computing.

1 Introduction and background

Several multi-finger robot grippers have been developed for different grasping and manipulation tasks. They include the Hanafusa Asada Hand, the Robonaut hand, the Honda Humanoid Robot Hand, the Salisbury hand, the Karlsrhue Hand, the Barret hand, the DLR hand, the GIFU hand, Ca.U.M.Ha hand, the Shadow hand and others (Hanafusa and Asada, 1977; Hirai et al., 1998; Lovchik and Diftler, 1999; Bicchi, 2000; Townsend, 2000; Figliolini and Rea, 2007; Shadow Robot Company, 2008; Deb and Deb, 2010) having degrees of freedom varying from 9 to 24. They have employed various actuation mechanisms, such as kinematic linkages, tendons, gears, etc. for opening and closing the fingers to accomplish grasping of objects of various shapes. The grippers have been designed and analysed for grasp stability, as well as dexterity, based on various criteria (Mishra and Silver, 1989; Al-Gallaf et al., 1993; Martell and Gini, 2007).

Another related important area of research had been the closure of an object, which dates back to the work of Realeaux (1876). Form closure is a geometric concept that expresses an idea of immobilising an object by putting constraints against its motion, in the form of point contacts, at specific locations of the object. Realeaux's idea was limited to 2D objects, and its conclusion was that a minimum of four point contacts is required for immobilising an object. Realeaux's method did not consider the amount of force required. Dizioglu and Lakshiminarayana (1984) took this concept one step further, and proposed at least seven points of contact for 3D objects. Mishra et al. (1987) devised a proper mathematical background for force closure, considering point contacts. Point contacts may be with or without friction. An altogether different approach has been developed by Mirtich and Canny (1994), for grasping objects in 2D or 3D. Using realistic models, they have shown that two fingers can grasp any 2D object, and three fingers can grasp any 3D object. It can be concluded, from the works done by Markenscoff and Yapadimitriou (1987) and Nguyen (1986), that a robot hand having three fingers, with friction at the tips, can stably grip an object.

Robotic grasping and manipulation requires controlling the movement of the gripper, through different points, in its work volume. Inverse kinematics computations are necessary in order to determine the joint angles in the gripper. The complexity involved in finding the inverse kinematics solution increases, as the number of joints in case of multi-finger hands increases. In order to deal with the complexity in solving the inverse kinematics equations, various approaches based on application of artificial intelligence (AI) techniques have been developed by researchers.

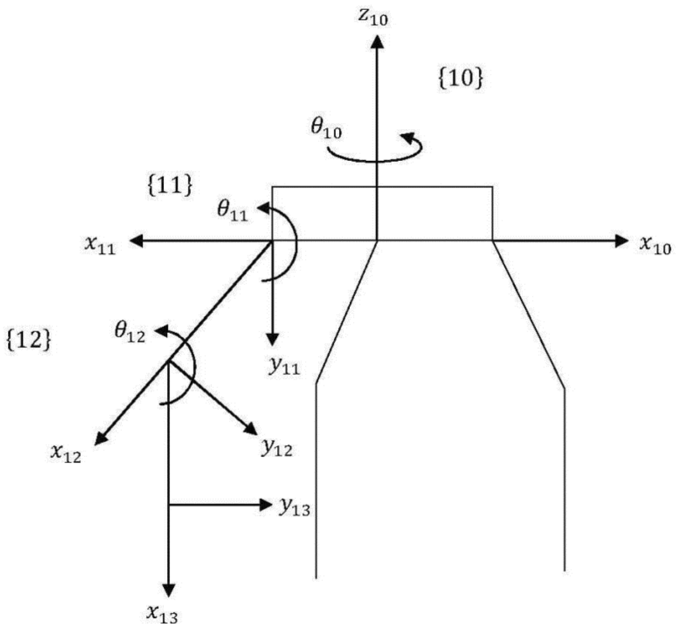
Khwaja et al. (1998) reported the use of genetic algorithm for finding out the inverse kinematics of any arbitrary robotic manipulator. They formulated the inverse kinematics problem as a multi-objective optimisation problem, and proposed a solution using a predator function, for fast convergence. As per their claim, premature convergence cannot be avoided in this method, and also the method cannot be applied in real time. Parker et al. (1989) have used genetic algorithm with a multi-objective fitness function to find out the inverse kinematics of any redundant robotic system. Ramirez and Rubiano (2011) have reported on compensating the error in the analytical equation of the inverse kinematics, using genetic algorithm. According to them, the error shows up at the boundary of the workspace. Tabandeh et al. (2006) have reported the use of adaptive niching and filtering techniques along with conventional genetic algorithm, to find out all the multiple solutions, of an inverse kinematics problem. They reported the success of the algorithm, when applied to a PUMA560 robot. Chao (2013) has reported the use of real-coded genetic algorithm for finding out the inverse kinematics of 3 degrees of freedom (DOF) planer robot.

Kim and Lee (1993) have reported the use of fuzzy logic in the inverse kinematics of a redundant manipulator. For a redundant manipulator the Jacobian matrix is not square. So a pseudo-inverse can be found for mapping the Cartesian space to the joint space. But the calculations related to the pseudo-inverse are time consuming. The researchers reported the use of a fuzzy logic-based rule table to bypass the computational complexity, so that the mapping from Cartesian space to joint space can be done in real time. Xu and Nechyba (1993) also reported the same thing in their paper. Fuzzy logic has also been used to develop the controller, used to optimise the finger force distribution around the object (Glossas and Aspragathos, 2001; Rovetta and Wen, 1991; Boughdiri et al., 2013).

Dash et al. (2011) have reported the comparative study of the use of multi-layer perceptron neural network (MLPNN) and polynomial processor neural network (PPNN), and concluded that use of neural network is better than the closed form solution of the inverse kinematics problems. Feng et al. (2012) worked on a concept of gathering additional points around the destination point using the idea of electromagnetic attraction, and then training a neural network for inverse kinematics. Rezzoug and Gorce (2006) used neural network-based reinforcement learning to find out the hand position, orientation and finger posture for a multi-finger robotic hand. In their case, the finger contact locations are defined in the object reference frame. The problem was to learn the hand position and orientation in the hand reference frame, and finger postures. Al-Gallaf (2006) used a Neural Inverse Kinematics approach, for optimal task force distribution in multi-finger robotic hand. Al-Gallaf (2004) further used neuro-fuzzy algorithm for mapping the object displacement during grasping, to hand joint positions and displacements. These works clearly demonstrated the usefulness of the AI techniques, in solving the robot inverse kinematics problem.

In the present work, a random walk algorithm has been developed for the inverse kinematics problem. The random walk algorithm uses the forward kinematics model, which is easier to formulate. Although the algorithm converges very quickly, it gives suboptimal solution. The algorithm does not generate a general mapping function between the input and the output. The algorithm has to be run every time, for obtaining the joint variables, for every destination point within the work volume. To overcome these limitations, a neural network-based approach has been developed for finding the inverse kinematics solution of a robotic hand. The neural network can generate the mapping function between the input and the output. The mapping function remains valid for a large portion of the work volume. The proposed approach has been demonstrated with the help of a modified, three finger and Hanafusa Asada robotic hand. The original Hanafusa Asada hand has three fingers, each having two revolute joints at the interphalangeal regions, and one revolute joint at the wrist, as shown in Figure 1. Thus, this hand has 7 DOFs. But the proposed hand has four revolute joints per finger, as shown in Figure 2. Thus, the modified hand has 12 DOFs. The inverse kinematics equations for a finger, form a set of six linear equations, with four transcendental variables. From this set of equations, the joint variables corresponding to a point in space can be found out. But the procedure of computing the solution is complex. To avoid the computational complexity, a neural network-based approach has been used. The parameters for setting up the network are not known a priori. Genetic algorithm has been used to set optimised values for the parameters. In the present problem, the input data corresponds to predefined Cartesian coordinate points in space, and the output data corresponds to the joint space variables of the robot hand.

Figure 1 A schematic diagram showing the coordinate systems and angular rotation at each joint for a single finger of Hanafusa Asada hand

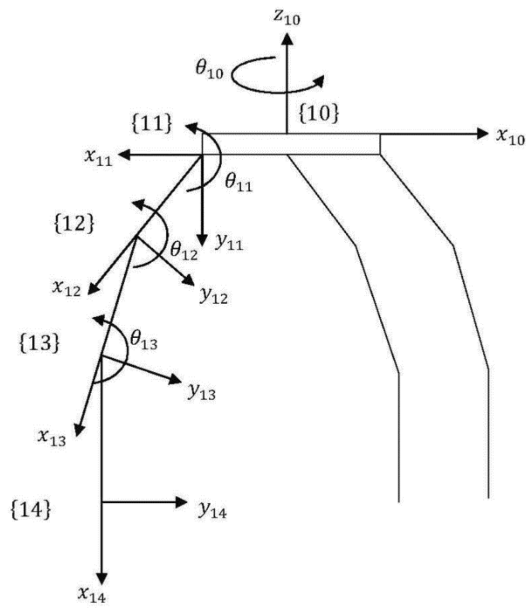


2 Proposed methodology of finding inverse kinematics solution

During grasping and manipulation, a robot gripper must be controlled to move continuously through different points in its work volume, before reaching the required destination point. This necessitates inverse kinematics computations, in order to determine the joint angles, i.e., continuous mapping of the end-effector tip coordinates in the Cartesian space (that forms the input space) into the joint variable space (that forms the output space). Further, the complexity involved in finding the inverse kinematics solution of a multi-finger robot hand increases with increasing number of joints.

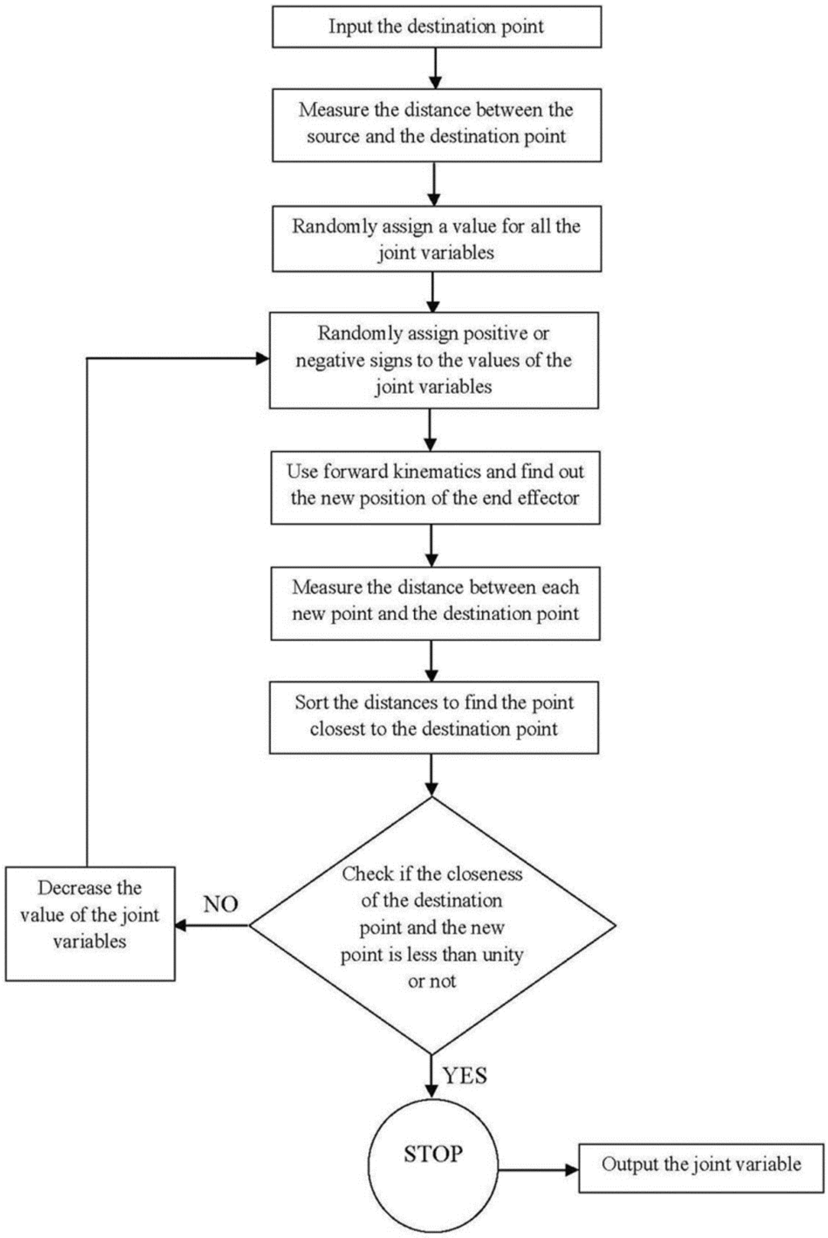
Keeping the above in mind, we have proposed random walk algorithm for solving the inverse kinematics of the robot hand. The algorithm starts with a known initial position of the end-effector, and iterates with the motor rotation directions, until it reaches a point close enough to the destination point (depending on the specified goal value). Initially, the algorithm is given the same value of rotation for all the joint variables. The magnitude of rotation for each joint variable goes on decreasing as the number of iterations increases. The working of the algorithm is illustrated in Figure 3. The algorithm quickly reaches the vicinity of the destination point, but it is very difficult for this algorithm to reach to the actual destination point as it is a random walk. The forward kinematics model in the random walk algorithm then generates a subspace of points in close vicinity of the destination point within the work volume.

Figure 2 A schematic diagram showing the coordinate systems and angular rotation at each joint for the first finger of the modified hand



The proposed algorithm has some advantages as well as limitations, the limitation being that it has to be run individually for every destination point, and it cannot generate a mapping function between the input and the output. So we have developed a neural network-based methodology of solving the inverse kinematics in conjunction with the random walk algorithm. The advantage of this proposal is that the neural network generates a mapping function, which is valid for a portion of the work volume. This saves computational time, when finding out joint variables corresponding to a new destination point. When the mapping function fails, we can remap the input and output, using the random walk algorithm and the neural network. For creating the training dataset, although the entire work volume of the robot can be considered as the input space, it may, however, bring in lot of noise in the input data because of its sheer size, and thus it may lead to increasing the complexity of the neural network architecture. Creating a subspace of points in close vicinity of the destination point, and then using this subspace to train the neural network, can provide a better approach. This is likely to reduce the noise in input data, and make the process of building and optimisation of the neural network architecture faster and simpler.

Figure 3 Flow chart to illustrate the working of the random walk algorithm



A radial basis function neural network ($F(x, \varphi, w)$) has been proposed by us in this paper, to solve the inverse kinematics problem that can generate a mapping function between the input and the output. A Gaussian function has been used as the radial basis function (φ). The neural network has been posed as a least square problem (*NRMSE*) and singular value decomposition (SVD) method has been used to solve it (de Lacerda, 2003;

Li et al., 2007; Baker, 2005; Haykin, 2010). A *K*-means cluster algorithm has been used to randomly initialise the training parameters (centres c_i, and spreads r_i of each hidden neuron). The tuning parameter (number of hidden neurons) is generally randomly assigned. The pseudo-code of the algorithm is as follows:

Begin
Divide input dataset into training data, validation data and test data
Initialize Population $\{ c_i, r_i\ (by\ k - clustering\ algorithm)\}$
Initialize Weights $(pseudo\ inverse,\ using\ training\ data)$
Validate $(cross\ validation, using\ validation\ data)$
Evaluate population $(using\ test\ data)$
While *not* $\{F(x, \varphi, w) < NRMSE\ or, generation > max. generation\}$
Do
Population
Crossover $(template)$
Mutation $(template)$
New Population
Update Weights $(pseudo\ inverse,\ using\ training\ data)$
Validate $(cross\ validation, using\ validation\ data)$
Evaluate population $(using\ test\ data)$
If $\{F(x, \varphi, w) < NRMSE\ or, generation = max. generation\}$
Stop
Else
Population= New Population
End

A genetic algorithm (GA)-based approach has been proposed in this paper to optimise the neural network (both the training and tuning parameters). The fitness function has been chosen inversely proportional to the closeness of the new points to the destination point obtained from the random walk algorithm. The parameters of the Genetic Algorithm were set as follows: maximum number of iterations at 300, population size at 100, crossover probability at 0.8 and mutation probability at 0.02. These values were arrived at after analysing the response of the fitness function upon varying the different parameters of GA. Figure 4 shows the convergence of the GA with the generations.

The template crossover and mutation have been used. Each chromosome in GA contains information regarding the centre, the spread and the number of hidden units. Based on this information, a neural network model is built. The neural network is trained in batch mode using a training set of input and output data and then it is validated using cross-validation technique and finally tested by feeding a set of unknown data. Competition among the contesting chromosomes ultimately gives the best neural network model which can perform a good mapping between the input and the output.

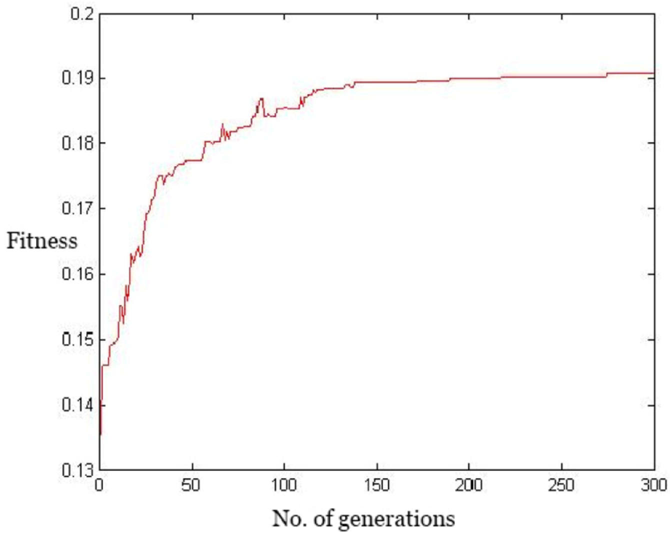
The training dataset for the neural network (i.e., the set of the end-effector tip coordinates representing the destination points and the corresponding joint variables) was created, using a forward kinematics model in the random walk algorithm, explained in the next paragraph.

The forward kinematics model that has been developed is shown in Figure 5.

The parameters for the model are the lengths of the links of each finger, range of angular rotations possible for each finger and angular rotation at each joint. The random walk algorithm takes in destination coordinates as the input, and it generates joint variables as output. It may be mentioned that the links of each finger are assumed to be rigid, and the nature of contact between the finger and the object is point contact.

The D–H parameters table for each finger is shown in Table 1.

Figure 4 Convergence of GA with the generations (see online version for colours)



For the first finger, the homogeneous transformation matrices are given below.

$$\left({}^{0}T_{1}\right)_{1} = \begin{bmatrix} \cos\theta_0 & 0 & -\sin\theta_0 & l_{10}\cos\theta_0 \\ \sin\theta_0 & 0 & -\cos\theta_0 & l_{10}\sin\theta_0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left({}^{1}T_{2}\right)_{1} = \begin{bmatrix} \cos\theta_{11} & -\sin\theta_{11} & 0 & l_{11}\cos\theta_{11} \\ \sin\theta_{11} & \cos\theta_{11} & 0 & l_{11}\sin\theta_{11} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left({}^{2}T_{3}\right)_{1} = \begin{bmatrix} \cos\theta_{12} & -\sin\theta_{12} & 0 & l_{12}\cos\theta_{12} \\ \sin\theta_{12} & \cos\theta_{12} & 0 & l_{12}\sin\theta_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left({}^{3}T_{4}\right)_{1} = \begin{bmatrix} \cos\theta_{13} & -\sin\theta_{13} & 0 & l_{13}\cos\theta_{13} \\ \sin\theta_{13} & \cos\theta_{13} & 0 & l_{13}\sin\theta_{13} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The final transformation matrix for the first finger is as follows:

$$({}^{0}T_3)_1 = ({}^{0}T_1)_1({}^{1}T_2)_1({}^{2}T_3)_1({}^{3}T_4)_1.$$

The same form of transformation matrix has been developed for the other two fingers. The above forward kinematics model has been implemented using MATLAB.

Figure 5 Forward kinematics model of the hand

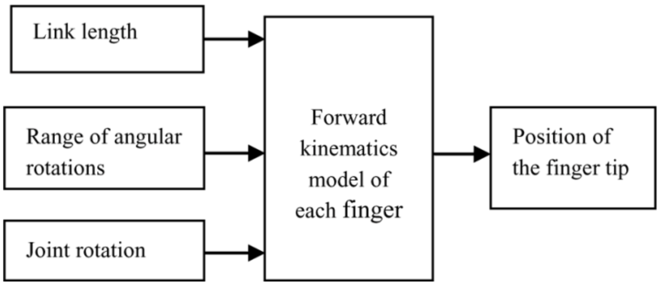


Table 1 D-H parameters table

First finger				*Second finger*				*Third finger*			
θ_i	d_i	a_i	α_i	θ_i	d_i	a_i	α_i	θ_i	d_i	a_i	α_i
θ_{10}	0	l_{10}	–90	θ_{20} + 120	0	l_{20}	–90	θ_{30} + 240	0	l_{30}	–90
θ_{11}	0	l_{11}	0	θ_{21}	0	l_{21}	0	θ_{31}	0	l_{31}	0
θ_{12}	0	l_{12}	0	θ_{22}	0	l_{22}	0	θ_{32}	0	l_{32}	0
θ_{13}	0	l_{13}	0	θ_{23}	0	l_{23}	0	θ_{33}	0	l_{33}	0

3 Illustrative example: results and discussions

A 3D destination or target point *T* (4.1844, 10.15138, –0.64204) has been set for the first finger of the three finger hand, to illustrate our proposed methodology for finding the inverse kinematics solution. The finger moves near the target point from the home position, using the random walk algorithm, stated earlier. In total, 400 more points are generated around the target point, by randomly changing the joint variables, within a scope of 10^0, using the forward kinematics model. A sample of the normalised values of joint variables of the finger, and the corresponding normalised Cartesian coordinate points around the destination point, obtained from forward kinematics model in the random walk algorithm, are shown in Tables 2 and 3, respectively.

Next, these data have been used to train the neural network. After optimisation by Genetic Algorithm, The optimised ANN has 13 hidden neurons. When the target point data are given to the neural network, it gives the following joint variable values

$$\theta_1 = 1.1807, \theta_2 = 0.2090, \theta_3 = -1.3718, \theta_4 = -2.000$$

The joint variables, when fed to the forward kinematics model, gives an output of (4.1825, 10.1744, –0.63949) which is, for all practical purposes, fairly close to the given target point. Thus the above method can provide a fairly effective and approximate inverse kinematics solution. It can be used for any number of joint variables. It effectively merges the power of a random search with neural network-based learning, and fairly simple to implement.

Table 2 Normalised joint variables

θ_{10}	θ_{11}	θ_{12}	θ_{13}
0.31569	0.040732	0.712776	0.801059
0.237315	0.059531	0.007256	0.939358
0.040131	0.39826	0.018422	0.602491
0.374312	0.560898	0.663551	0.081973
0.887031	0.425691	0.07314	0.723282
0.542922	0.929007	0.084161	0.25851
0.467181	0.370633	0.261989	0.466503
0.050006	0.648552	0.417366	0.27794
0.590599	0.914165	0.734675	0.993862
0.88475	0.558835	0.440806	0.690822
0.786834	0.791077	0.32235	0.544791
0.71328	0.887114	0.99325	0.346526
0.884204	0.035385	0.776114	0.588656
0.404449	0.278858	0.007058	0.092891
0.927947	0.021947	0.440297	0.34544
0.448094	0.707858	0.397923	0.052575
0.108454	0.320988	0.697032	0.470968
0.399773	0.60297	0.620282	0.672765
0.982333	0.806758	0.326713	0.763486
0.831066	0.071728	0.362478	0.987359

Table 3 Normalised coordinate variables

x	y	z
0.5936	0.3873	0.0492
0.4484	0.0512	0.0099
0.6597	0.1089	0.2738
0.6409	0.5611	0.5534
0.1132	0.4207	0.2762
0.4363	0.4845	0.7059
0.4423	0.3664	0.2923
0.8315	0.348	0.5463
0.5493	0.8044	0.734
0.1968	0.6456	0.4181
0.2812	0.6341	0.5986

Table 3 Normalised coordinate variables (continued)

x	*y*	*z*
0.4697	0.9107	0.8783
0.1789	0.595	0.0854
0.406	0.2052	0.2747
0.0869	0.4394	0.0836
0.5461	0.5231	0.6236
0.8018	0.3904	0.3097
0.6257	0.5792	0.4887
0.1275	0.6994	0.5788
0.1518	0.4031	0.0137

A number of simulation experiments have been conducted to evaluate the performance of our proposed neural network-based methodology. Tables 4 and 5 show some results. The 'actual output' in Table 4 corresponds to the outputs from the random walk algorithm, which generates random coordinate points around the destination points. The 'calculated points' in Table 5, corresponds to the outputs generated by the ANN algorithm. When compared, Tables 4 and 5 show that the outputs of the ANN-based algorithm is very close to the outputs from the random walk algorithm.

Table 4 Actual output

x	*y*	*z*
5.103313	11.31081	0.168574
4.1844	10.15138	–0.64204
5.983426	11.23161	0.608865
4.922277	9.493917	–1.44087
5.732002	9.618959	–1.12848
5.955701	10.67446	–0.90664
3.945857	11.19174	–0.31745
5.033183	9.23399	–1.69282
5.817164	11.46712	0.109072
6.495608	10.40415	–1.39247
4.181162	10.2758	–0.7236
5.857731	9.949393	0.227375
5.772003	9.833922	–2.21219
4.710892	10.39367	–0.70736
4.934877	11.87044	–0.04307
7.165218	10.363	–1.08536
5.855523	8.353339	–1.78274
4.787082	12.4109	0.516813
4.131956	11.29092	–1.2415
3.49835	11.05703	–1.21491

Table 5 Calculated output

x	y	z
5.14216	11.14618	0.221044
4.182504	10.1744	–0.63949
6.051557	11.10133	0.698538
4.91003	9.259056	–1.54784
5.727614	9.62339	–1.1824
5.938768	10.69789	–0.89011
4.034977	10.95054	–0.45161
5.038202	8.936702	–1.77081
5.854115	11.40822	0.223036
6.521662	10.52045	–1.37729
4.196406	10.23457	–0.7007
5.880504	9.883186	0.24028
5.881925	9.653266	–2.24256
4.691767	10.29223	–0.73219
5.033441	11.63904	–0.00847
7.185749	10.50519	–1.0887
5.799326	8.716356	–1.72555
4.92126	12.10593	0.56387
4.088231	11.26472	–1.16309
3.385128	11.25674	–1.04352

4 Conclusions

The complexity involved in finding the inverse kinematics solution of a multi-finger robot hand increases with increasing number of joints. In the present work, a methodology based on random search and application of a radial basis function neural network has been proposed for finding the inverse kinematics solution. Further, a genetic algorithm-based approach has been developed for optimising the parameters of the neural network. They have been implemented using MATLAB. Instead of taking the entire work volume of the robot hand for training the neural network, a subspace of points is created in close vicinity of the given destination point. In the next step, the joint variables corresponding to the above points are obtained using a forward kinematics model of the hand. Then, the subspace of points and the corresponding joint variables, as obtained above, are used to train the neural network. An illustrative example has been presented to demonstrate the application of the above methodology. It is anticipated that the proposed random search and neural network-based approach can provide an approximate yet fairly quick and effective solution for the inverse kinematics problem of multi-finger robot hands.

References

Al-Gallaf, E.A. (2004) 'Neurofuzzy inverse jacobian mapping for multi-finger robot hand control', *Journal of Intelligent and Robotic Systems*, Vol. 39, pp.17–42.

Al-Gallaf, E.A. (2006) 'Multi-fingered robot hand optimal task force distribution neural inverse kinematics approach', *Robotics and Autonomous Systems*, Vol. 54, pp.34–51.

Al-Gallaf, E.A., Allen, A. and Warwick, K. (1993) 'A survey of multi-fingered robot hands: issues and grasping achievements', *Mechatronics*, Vol. 3, No. 4, pp.465–491.

Baker, K. (2005) *Singular Value Decomposition Tutorial*, http://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf (Accessed 15 March, 2014).

Bicchi, A. (2000) 'Hands for dextrous manipulation and robust grasping: a difficult road towards simplicity', *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 6, pp.652–662.

Boughdiri, R., Bezine, H. and Alimi, A.M. (2013) 'Fuzzy logic control for grasping 3d objects with sliding contacts', *Proceedings of International Conference on Control, Engineering and Information Technology*, Vol. 2, pp.116–119.

Chao, S. (2013) 'The study of robot movement inverse solution based on genetic algorithm', *Modern Applied Science*, Vol. 7, pp.53–58.

Dash, K.K., Choudhury, B.B., Khuntia, A.K. and Biswal, B.B. (2011) 'A neural network based inverse kinematics problem', *Proceedings of 2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Trivandrum, India, pp.471–476.

de Lacerda, E.G.M. (2003) *Model Selection of RBF Networks via Genetic Algorithms*, PhD Thesis, Informatics Center of Pernambuco Federal University.

Deb, S.R. and Deb, S. (2010) *Robotics Technology and Flexible Automation*, 2nd ed, Tata McGraw Hill Education Private Limited, New Delhi.

Dizioglu, B. and Lakshiminarayana, K. (1984) 'Mechanics of form closure', *Acta Mechanica*, Vol. 52, pp.107–118.

Feng, Y., Yao-nan, W. and Yi-min, Y. (2012) 'Inverse kinematics solution for robot manipulator based on neural network under joint subspace', *Int. Journal of Comput Commun.*, Vol. 7, No. 3, pp.459–472.

Figliolini, G. and Rea, P. (2007) 'Ca.U.M.Ha robotic hand (Cassino-underactuated-multifinger-hand)', *Proceedings of 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Zurich, pp.1–6.

Glossas, N.I. and Aspragathos, N.A. (2001) 'Fuzzy logic grasp control using tactile sensors'. *Mechatronics*, Vol. 11, pp.899–920.

Hanafusa, H. and Asada, H. (1977) 'Stable prehension by a robot band with elastic fingers', *Proceedings of 7th International Symposium on Industrial Robots*, pp.361–368.

Haykin, S. (2010) *Neural Networks and Learning Machines*, PHI Learning Private Ltd., New Delhi.

Hirai, K., Hirose, M., Haikawa, Y. and Takenaka, T. (1998) 'The development of Honda humanoid robot', *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, Vol. 2, pp.1321–1326.

Khwaja, A.A., Rahman, M.O. and Wagner, M.G. (1998) 'Inverse kinematics of arbitrary robotic manipulators using genetic algorithms', in Lenarcic, J. and Husty, M.L. (Eds.): *Advances in Robotic Kinematics: Analysis and Control*, Kluwer Academic Publishers, MA, USA, pp.375–382.

Li, C.H. and Park, S.C. (2007) 'Neural network for text classification based on singular value decomposition', *Proceedings of Seventh International Conference on Computer and Information Technology*, Fukushima, Japan, pp.47–52.

Lovchik, C. and Diftler, M. (1999) 'The robonaut hand: a dexterous robot hand for space', *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, Vol. 2, pp.907–912.

Markenscoff, X. and Yapadimitriou, C.H. (1987) *Optimum Grip of a Polygon, Report No. STAN-CS-87-1153*, Department of Computer Science, Stanford University, April.

Martell, J.W.S. and Gini, G. (2007) 'Robotic hands: design review and proposal of new design process', *World Academy of Science, Engineering and Technology*, Vol. 26, pp.85–90.

Mirtich, B. and Canny, J. (1994) 'Easily computable optimum grasp in 2D and 3D', *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, pp.739–747.

Mishra, B. and Silver, N. (1989) 'Some discussion of static gripping and its stability', *Proceedings of IEEE Transactions on Systems. Man. and Cybernetics*, Vol. 19, No. 4. pp.783–796.

Mishra, B., Schwartz, J.T. and Sharir, M. (1987) *On the Existence and Synthesis of Multifinger Positive Grips*, Algorithmica, Springer-Verlag Inc., New York, USA, pp.542–558.

Nguyen, V-D. (1986) *The Synthesis of Stable Force-Closure Grasp. Technical Report 905*, MIT Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Parker, J.K., Khoogar, A.R. and Goldberg, D.E. (1989) 'Inverse kinematics of redundant robots using genetic algorithm', *Proceedings of IEEE International Conference on Robotics and Cybernetics*, Scottsdale, USA, Vol. 1, pp.271–276.

Ramirez, A.J. and Rubiano, F.A. (2011) 'Optimization of inverse kinematics of a 3R robotic manipulator using genetic algorithm', *World Academy of Science, Engineering and Technology*, Vol. 59, pp.1425–1430.

Realeaux, F. (1876) *Kinematics of Machinery*, Macmillan and Company, London.

Rezzoug, N. and Gorce, P. (2006) 'Robotics grasping: a generic neural network architecture', in Lazinica, A. (Ed.): *Mobile Robots Towards New Applications, Advanced Robotic Systems International and pro literatur Verlag*, Mammendorf, Germany.

Rovetta, A. and Wen, X. (1991) 'Fuzzy logic in robot grasping control', *Proceedings IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS'91*, Vols. 3–5, pp.1632–1637.

Shadow Robot Company (2008) *Shadow Dexterous Hand C5 Technical Specification*, http://www.shadowrobot.com/downloads/shadow_dextrous_hand_technical_specification_C5.pdf (Accessed 2 November, 2015).

Tabandeh, S., Clark, C. and Melek, W. (2006) 'A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering', *Proceedings of 2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, pp.1815–1822.

Townsend, W.T. (2000) 'The barret hand grasper-programmably flexible part handling and assembly', *Industrial Robot: An International Journal*, Vol. 27, No. 3, pp.181–188.

Xu, Y. and Nechyba, M.C. (1993) 'Fuzzy inverse kinematic mapping: rule generation, efficiency and implementation', *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems' 93*, Vol. 2, pp.911–918.