An ANN and Virtual Vector Function Approach for the Computation of the Inverse Kinematics of Redundant Manipulators

SHAHNAZ HABIBKHAH, RENE V. MAYORGA
Faculty of Engineering and Applied Science
University of Regina
3737 Wascana Parkway, Regina, Saskatchewan, S4S 0A2
CANADA

Abstract: - This Paper presents an Artificial Neural Network (ANN) method for the solution of the Inverse Kinematics (IK) of redundant manipulators. This problem normally has an infinite number of solutions. Several conventional approaches based on numerical methods have been proposed over the years. However, it is well known that ANN implementations for the solution of the problem for redundant manipulators are inefficient since they lead to inaccurate solutions. The main issue on the implementation of ANN solutions is that an attempt is made on mapping a relation from a given number of variables in the manipulator task space to a larger number of variables in the joint space. This Paper proposes the inclusion of a virtual vector function in the task space to "complete" it; so that the number of variables in the task space is at least equal to the number of variables in the joint space. Here, the proposed approach is successfully tested on a 3 DOF planar redundant manipulator performing four diverse target trajectories inside the manipulator's workspace. Additionally, in order to define the target trajectories, some manipulator's links limitations are considered and some conditions are set for the target trajectories.

Keywords: - Artificial Neural Network (ANN), Virtual Vector Function, 3 DOF Redundant planar Manipulator.
Received: February 16, 2021. Revised: November 22, 2021. Accepted: December 14, 2021. Published: December 30, 2021.

1 Introduction

The inverse kinematics (IK) problem of robot manipulators is concerned with transforming the position of a robotic arm from Cartesian coordinates to joint coordinates [1]. The IK computation has been an important problem in robotics and has received considerable attentions over the years. Solving the IK is difficult and time-consuming; in particular for Redundant Manipulators (RM) [3]. A manipulator for which the number of DOF is higher than what is required to perform a task; is considered kinematically redundant [7, 8]. As the robot complexity increases, obtaining an explicit solution for the IK becomes difficult [4-6].

In order to solve the IK problem, numerical solutions are presented by [8-16]. In [9], recursive least squared are used and a dynamic parameter identification algorithm according to particle swarm optimization to solve the IK of a 3 DOF manipulator with 3 joint angles, two links in 3D. In [10], a combination of geometric and numerical approaches is presented. In [11], a Newton-Raphson numerical method is used based on a composite Jacobian to solve IK. In [11], a 7 DOF RM is considered and the position and orientation of the end effector, as well as position coordinates of other links are taken in to account as constrains. In [8], a numerical solution to IK of RM is proposed in which the constraints are set for obstacle avoidance and joints limitations. Aiming to solve IK problem of a 7 DOF RM; [12], has combined the Newton Raphson iteration method with the damped least square method. The presented solution by [13], is a swarm optimization technique and the algorithm type is firefly. In [14], a solution to IK of RM and Hyper RM (HRM) is presented and joint angles limits, singularity avoidance and obstacle avoidance constraints are used. In order to propose a numerical solution to IK of HRMs a Jacobian pseudo inverse is used in [15]. A new geometrical method to solve IK of HRMs is found in [16]. In this method, same angles between each two consecutive links are found which has led to easy links control and stable movement of the manipulator.

On the other hand, [5, 17-29] have presented non-conventional solutions to the IK of manipulators. In [17], a fuzzy logic tool is used for nonlinear mapping between Cartesian and joint coordinates and for each joint, a decentralized fuzzy controller with a fuzzy associated memory has been used to perform the IK.

In [18] a fuzzy based solution is presented to solve the IK of RMs, and a 4 DOF manipulator is examined by this method. The linguistic fuzzy rules are defined based on internal and external restrictions which are dealt with end position of links and obstacle respectively. In [19] this tool is used to present a solution for both non redundant and redundant manipulators.

In [20], conventional and continuous Genetic Algorithms (GA) are used to solve IK. In [20], it is proved that continuous GA performs better than the conventional GA since it has provided smoother and faster solutions.

Another non-conventional solution is by an Adaptive Neuro-Fuzzy Inference System (ANFIS) which is used by [5, 21-25]. In [5] this approach is used for both 2 DOF and 3DOF robot manipulators and position coordinates are used to train the ANFIS related to the 2DOF, and in order to train 3 DOF forward kinematics equations, as well as a summation of 3 joint angles, are used. In [21] this tool is used to solve the IK of a dual-arm six degrees of freedom robot. In [22], an ANFIS is used to solve the IK of a 3 DOF planar and the end effector's position as well as its orientation (RM) are inputs of the ANFIS. In [23], four ANFIS are used to solve IK of 4 DOF SCARA manipulator. The end effector positions (x, y, z) are considered as inputs, and joint angles ($\theta_1, \theta_2, \theta_3, \theta_4$) are the outputs of the system. In this [23], a Sugeno fuzzy model [24] is chosen as the Fuzzy Inference System (FIS) model and the membership functions of each ANFIS's inputs are Gaussian membership functions which can be found in [25].

In [26], an ANN is used for a solution to the IK and singularity avoidance of RMs. In [26], a null space vector to avoid singularities is computed by an ANN and used to in the scheme computation of the IK which is realized by another ANN. A comparative study between Artificial Neural Network, Adaptive Neuro-Fuzzy Inference System & Genetic Algorithms methods in solving IK of 5 DOF is found in [27], and according to its claim, both ANN and ANFIS had better results than the other methods.

For a planar 3 DOF RM, there are three joint angles and two end effector coordinates; so in general, there is no explicit solution for the IK. However, an explicit solution can be obtained by including joint constraints functions in the set of equations, and be implemented numerically.

In [28], an ANN is used to solve the IK of a 3 DOF planar manipulator. The training inputs of the NN are position (x,y) of the end effector and summation of joint angles, as orientation. In [29], this tool is used to solve the IK for a six degree of freedom (DOF) robotic arm and the neural network's inputs are the end effector's desired position and orientation and feedback of the current joint angles.

In this Paper, a general ANN approach is presented to solve the IK problem of redundant manipulators. In particular, a 3 DOF planar redundant manipulator is considered. The pre-set joints are utilized to find the position coordinates of the end effector using Forward Kinematics equations. Additionally, the 3 joints variables are considered to compute a virtual (auxiliary) vector function that "completes" the task space; so that the number of variables in the task space is at least equal to the number of variables in the joint space, and facilitates the ANN implementation. The 2D coordinates of the end effector and the virtual vector function are used to prepare training data for the ANN. The ANN type here is feedforward and the optimization method for the training function is Levenberg- Marqardt (LM) [30-32]. It is important to mention that the considered manipulator is redundant; so by using only the position coordinates of the end effector there will be no explicit solution for the IK; or there might be some solutions for a limited constraints range of the joint angles. In order to test the proposed approach; some target trajectories are designed inside the workspace and considering some manipulators limitations. Here it is shown the trained ANN ability to successfully track them.

The remainder of this Paper is organized as follows: Section 2 describes the robot manipulator kinematics equations, the end effector workspace, the virtual function, and the target trajectories conditions. Section 3 explains the ANN structure and design. In section 4, the desired target trajectories are tested regarding the conditions and simulation results of implementing ANN are presented. Finally, the section 5 contains some conclusions.

2 Problem Description

2.1 Robot Manipulator Kinematics

The configuration of the 3 DOF planar RM is depicted in Fig. 1. The Forward kinematic equations ((1) and (2)) set the end-effector positions for specified values of robot arm joint parameters. This procedure is depicted in Fig. 2. In the following equations, θ_1, θ_2, and θ_3 are the first, second and third

joint angles of the manipulator, and l_1, l_2 , and l_3 are the lengths of the robot arms.

$$X = l1 * cos(\theta1) + l2 * cos(\theta1 + \theta2) + l3 * cos(\theta1 + \theta2 + \theta3) \quad (1)$$

$$Y = l1 * sin(\theta1) + l2\, sin(\theta1 + \theta2) + l3 * sin(\theta1 + \theta2 + \theta3) \quad (2)$$

On the other hand, the setting of the end-effector's position coordinates will yield infinite number of solutions for the joint angles. To overcome this issue, a virtual dummy vector function *Fv(θ)* can added as a constraint to solve the IK problem, Fig. 3.

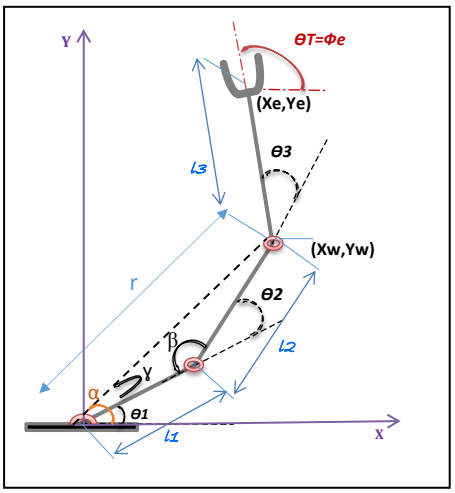

Fig.1 3 DOF redundant manipulator

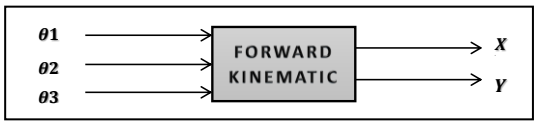

Fig.2 3DOF RM forward kinematic

If the virtual vector function contains only one component of a particular form as in [28]; in fact, it may be possible to obtain an explicit closed form solution. In this Paper, aiming to present a general approach, the virtual vector includes two components [33]. Here, the *cos* and *sin* of the summation of three joint angles are used as components of the virtual vector:

$$Fv = \begin{pmatrix} Fv1 \\ Fv2 \end{pmatrix} = \begin{pmatrix} cos(\theta1 + \theta2 + \theta3) \\ sin(\theta1 + \theta2 + \theta3) \end{pmatrix} = \begin{pmatrix} cos(\theta T) \\ sin(\theta T) \end{pmatrix} \quad (3)$$

In which θ_1, θ_2 and θ_3 are the first, second and third joint angles of the manipulator and θ_T is the summation of them that can be called orientation of the end effector. In the rest of the paper, these data will be used to train ANN to solve IK problem and simulation results will be discussed.

In [34], orientation of the end effector is defined *cos(.)* of the summation of the joint angles. Considering this orientation besides the position coordinates of the end effector as inputs of ANN, also leads to have solutions for the IK problem. However, due to alternative feature of the *cos(.)* function, solutions exist only for a limited range of the joint angles.

2.2 The End Effector Position Workspace

According to the kinematics equations, the end effector's coordinates can be obtained from the joint angles amount and changing the angles in a permitted range leads to different position coordinates. In this Paper, the following ranges are defined for angles in (4). Using the possible end positions coordinates, the workspace of the robot arm is obtained and illustrated in Fig. 4 while the length of the three links of the robot are supposed *l1=10, l2=7, l3=5*:

$$0 \le \theta1 \le \pi,\ -\pi \le \theta2 \le 0 \ \ and \ \ -\pi/2 \le \theta3 \le \pi/2 \quad (4)$$

The target trajectories are needed to be defined inside this area. Moreover, based on trigonometry rules, the following conditions should be met to define a target trajectory [33, 35]:

$$(l_1 - l_2)^2 \le (x - l3\cos\theta_T)^2 + (y - l3\sin\theta_T)^2 \le (l_1 + l_2)^2 \quad (5)$$

$$(\sqrt{(x - l3\cos\theta_T)^2 + (y - l3\sin\theta_T)^2} - l_1)^2 \le \quad (6)$$

$$l_2^2 \le (\sqrt{(x - l3\cos\theta_T)^2 + (y - l3\sin\theta_T)^2} + l_1)^2 \quad (7)$$

According to the above equations, placing the end effector of the manipulator in a target position, depends on the length of the manipulator as well as the desired orientation. Therefore, there might be some positions of the end effector inside the blue workspace area of the Fig.4 which are infeasible to get with specific orientations.

Fig.3 Inverse Kinematic solution for 3 DOF RM

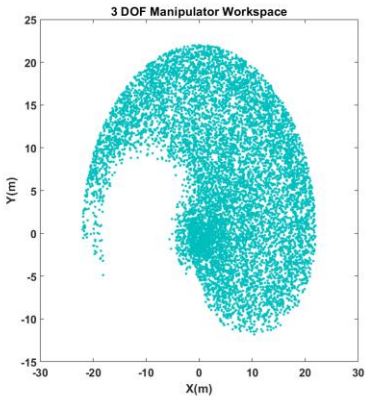

Fig.4 3 DOF RM Workspace

3 Artificial Neural Networks approach

3.1 ANN Structure

An adaptive network is a network structure with a set of nodes whose functionality is performing a node function on the incoming signals and generating a single output. The node function's parameters are modifiable, so the overall behavior of the network is changeable. The adaptive network shown in Fig. 5 is feedforward since there is no feedback link in the network [24]. For an ANN to be effective, a training function is needed. One of the most popular training function is *trainbr*, which updates the bias and weights using an LM optimization method. Aiming to produce an efficient network, here a linear combination of squared errors and weights is minimized using the Bayesian regularization (BR) backpropagation method to find an optimal combination of errors & weights, [25, 36, 37].

3.2 The ANN Design (Arranging its structure)

In order to generate input data sets to train ANN, the pre-set random joint angles are feed in (1), (2), and (3) and the corresponding end positions coordinates and virtual vector functions values are obtained. Afterwards, these data are used as the inputs of the ANN while the outputs of the ANN are set as the joint angles. So, providing enough input and output data sets, will lead to a well training the ANN. Moreover, a good design of ANN, needs testing data. For this purpose, for each angle 8500 random samples are created; from which 7000 are used for training aim, and 1500 are used for testing aim. The number of hidden neurons (the size of hidden layer) is 140. Fig. 6 shows the designed ANN in MATLAB.

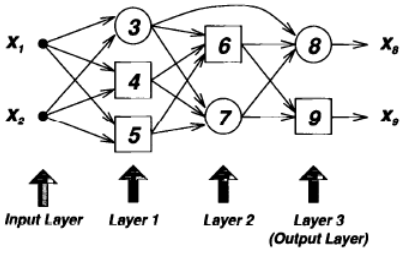

Fig.5 Feedforward adaptive network [24]

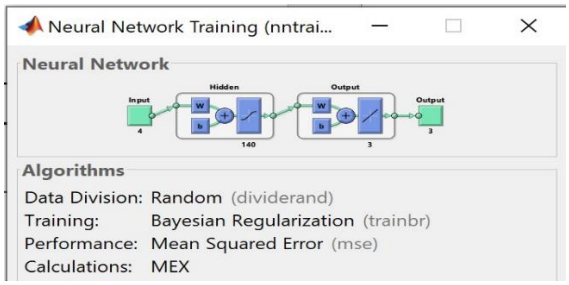

Fig.6 Considered ANN Block and algorithm in Matlab

4 Simulation

4.1 Designing the Target Trajectory

Here four end-effector target trajectories are designed inside the manipulator workspace for which the conditions in (5) and (6) are considered, while in [28] some trajectories are tracked without considering the manipulator's limitations. The 1st, 2nd, 3rd, and 4th target trajectories are mentioned in (7), to (10) respectively in which x and y are
The coordinates of the trajectories.

$$t = 0{:}\,0.01{:}\,2\pi;\ y = 3.087 + 8\sin(t); x = 12.9 + 8\cos(t) \quad (8)$$

$$t = 1.4\pi{:}\,.01{:}\,2.75\pi; y = 3.087 + 8\sin(t); x = 12.9 + 8\cos(t) \quad (9)$$

$$\text{a rectangle with: } x = 5{:}\,15; y = -1{:}\,10 \quad (10)$$

$$\text{a semi rectangle with: } x = 5{:}\,15; y = -1{:}\,10 \quad (11)$$

The coordinates of the above trajectories are used as x and y in (5) and (6) while the orientation is supposed $\tan^{-1}(y/x)$. The results of conditions analysis related to the first and second trajectories are depicted in Fig. 7 and Fig. 8, and in Fig. 9 and Fig. 10 respectively. The results of conditions analysis related to the third and fourth trajectories are

depicted in Fig. 11 and Fig. 12, and in Fig 13 and Fig. 14 respectively.

In Fig. 7, Fig.9, Fig. 11, and Fig.13 the condition 1 mentioned in (5) is considered for the trajectories. In these figures the upper and lower bounds of (5) are named "*upper bound*" and "*lower bound*" respectively, and the function between two inequalities is named "*function*" that must be located between two bounds to satisfy (5).

In Fig. 8, Fig.10, Fig.12, and Fig.14 the condition 2 mentioned in (6) is considered. The upper and lower bounds of (6) are named "upper bound function" and "lower bound function" respectively, and the term l_2^2 is named "constraint" that must be placed between upper bound function and lower bound function to meet the condition (6).

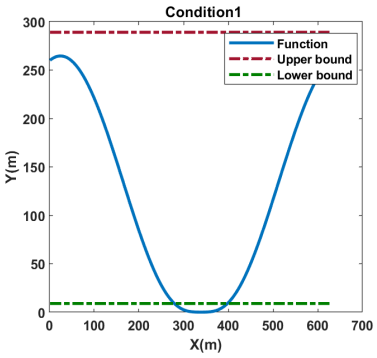

Fig.7 Condition1 analysis for the first trajectory

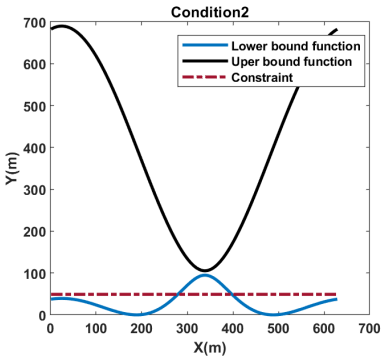

Fig.8 Condition2 analysis for the first trajectory

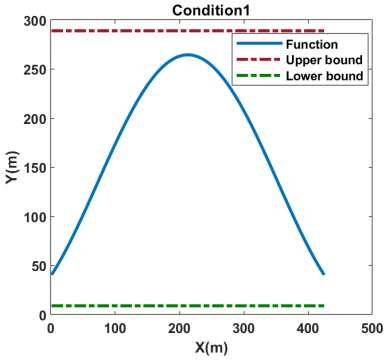

Fig.9 Condition1 analysis for the second trajectory

Considering Fig.7 and Fig.8, and also Fig.11 and Fig.12, obviously the first and the third trajectories do not meet the conditions, so it is impossible for the end effector of this manipulator to locate at some positions of the 1st and 3rd trajectories with defined orientations. On the other hand, all parts of the 2nd and 4th trajectories meet the conditions, so all positions of these trajectories are feasible for the manipulator's end effector with considered orientations. Next, the ability of the ANN to track the trajectories is verified.

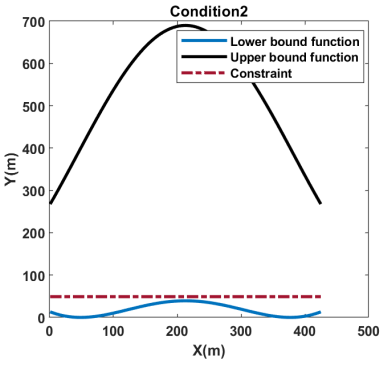

Fig.10 Condition2 analysis for the second trajectory

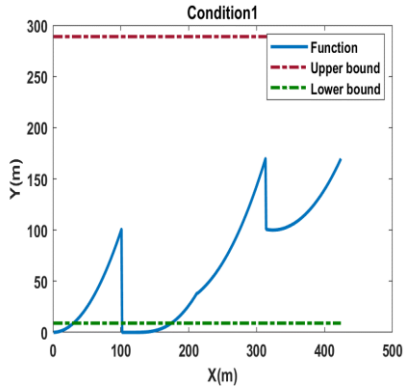

Fig.11 Condition1 analysis for the third trajectory

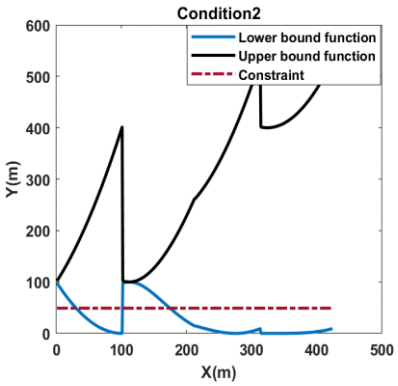

Fig.12 Condition2 analysis for the third trajectory

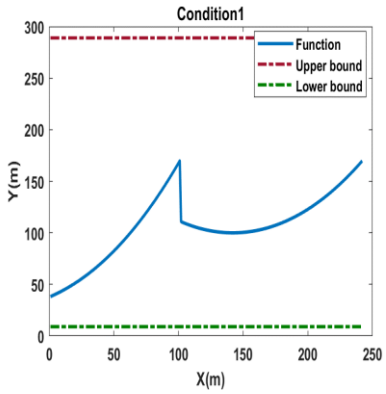

Fig.13 Condition1 analysis for the fourth trajectory

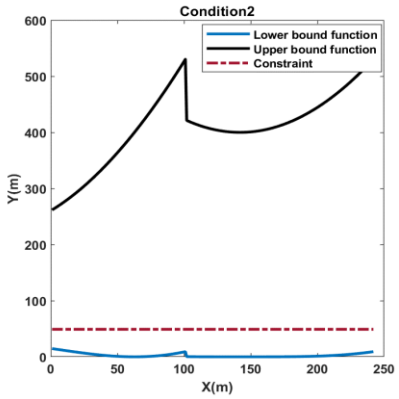

Fig.14 Condition2 analysis for the fourth trajectory

4.2 Results

In this section, the ANN training and its training performance, regression, and error histogram are shown, respectively. According to these figures, the ANN performance for both training and testing data have acceptable mean squared error, and the error histogram is in a good range.

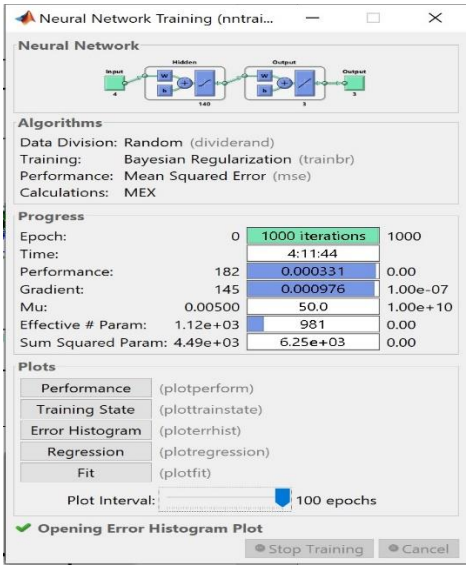

Fig.15 ANN training

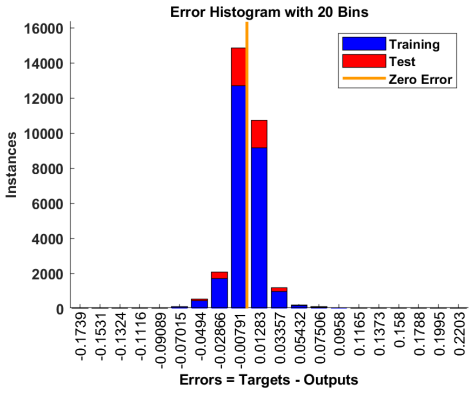

Fig.16 Error histogram of the ANN

Next, the position coordinates and virtual dummy functions data are prepared using the target trajectories coordinates. To prepare data for *Fv*, θ_T is assumed to be the orientation of the end effector which is equal to *atan2(y, x)*. In the following figures, tracking result of the ANN related to the first, second, third, and fourth target trajectories are depicted respectively. Clearly, some parts of the trajectory in Fig. 17 and Fig. 19 are not tracked by the ANN. These areas are related to those coordinates of the trajectories which do not meet the conditions.

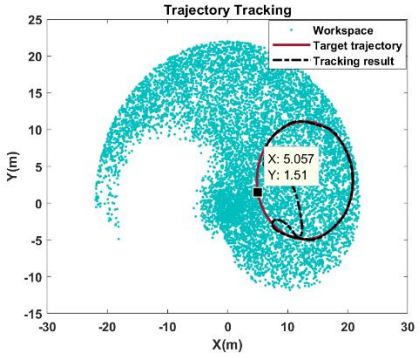

Fig.17 Tracking result of the first trajectory

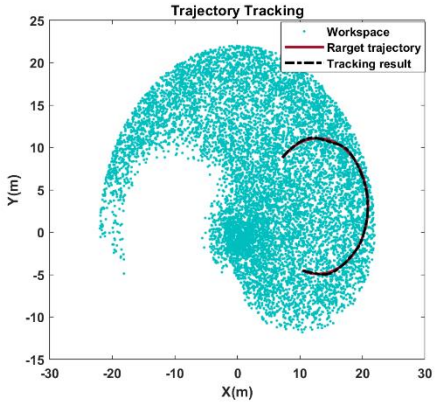

Fig.18 Tracking result of the second trajectory

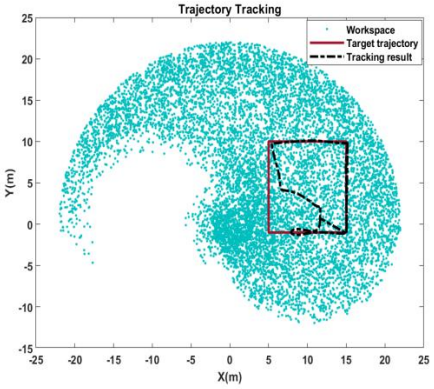

Fig.19 Tracking result of the third trajectory

Indeed, the second trajectory is a part of the first trajectory for which the position coordinates and orientations meet the conditions. Similarly, the fourth trajectory is a part of the third trajectory for which the position coordinates and orientations meet the conditions. The Fig. 18 and Fig. 20 are related to the second and fourth trajectories which have met the conditions. These trajectories are tracked with a good accuracy. The manipulator's links during tracking the second and fourth trajectories are shown in Fig. 21 and Fig. 22, respectively.

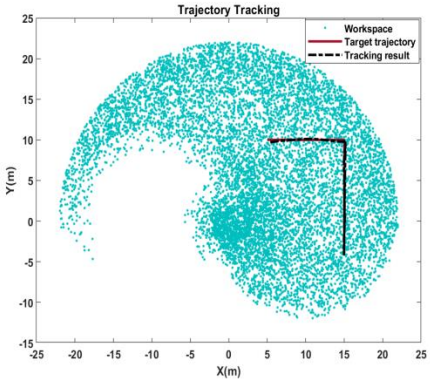

Fig.20 Tracking result of the fourth trajectory

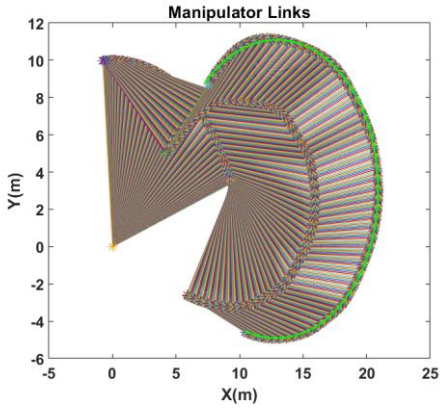

Fig.21 Manipulator's links during tracking trajectory2

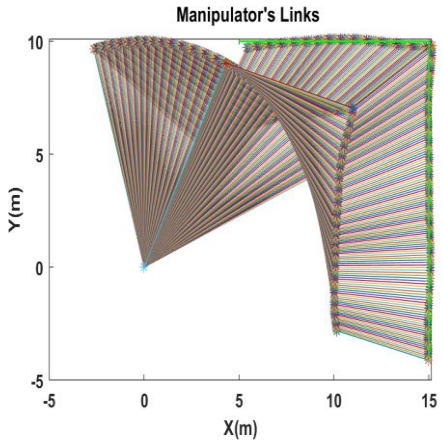

Fig.22 Manipulator's links during tracking trajectory4

As observed at a portion of the concave curve in the Figure 21, the proposed approach can deal with joint configurations close to singularities (the manipulator fully stretched) without losing accuracy in the trajectory tracking. This is due to the auxiliary virtual vector function acting also as a constraint [26], for the singularities prevention.

5 Conclusions

In this Paper it is proposed a non-conventional ANN and a virtual vector function approach to the IK problem of redundant manipulators; in particular, a 3 DOF redundant planar manipulator is studied. In order to define the virtual vector function with two components; a trigonometric function of the summation of joint angles for each component is considered. Additionally, some target trajectories are defined inside the workspace of the manipulator, and the ANN ability to track them is verified. The target trajectories are constrained by the manipulator's geometric limitations and the considered ANN complies with these limitations. The considered ANN is able to follow the accepted target trajectories with a high accuracy.

The proposed approach can be used to solve the IK of other redundant manipulators with the similar configurations. For this purpose, steps as indicated in [33] can be taken. Also, in order to verify if a manipulator target trajectory is feasible; the conditions developed in this paper can be considered with regards to the manipulator's links length, and the target trajectory's positions and orientations for the manipulator.

In general cases that the virtual vector function is complicated (even if it consists of only one component) as required by some tasks; it is very difficult to have an explicit solution. Therefore, the proposed approach is also suitable for general cases.

References

[1] Wang, L-CT, and Chih-Cheng Chen. "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators." IEEE Transactions on Robotics and Automation 7.4 (1991): 489-499.

[2] Murray, Richard M. A mathematical introduction to robotic manipulation. CRC press, 2017.

[3] Fu, Zhongtao, Wenyu Yang, and Zhen Yang. "Solution of inverse kinematics for 6R robot manipulators with offset wrist based on geometric algebra." Journal of mechanisms and robotics 5.3 (2013): 031010

[4] Alavandar, Srinivasan, and M. J. Nigam. "Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators." International Journal of Computers Communications & Control 3.3 (2008): 224-234

[5] Alavandar, Srinivasan, and M. J. Nigam. "Inverse kinematics solution of 3DOF planar robot using ANFIS." Int. J. of Computers, Communications & Control 3 (2008): 150-155

[6] Daya, Bassam, Shadi Khawandi, and Mohamed Akoum. "Applying neural network architecture for inverse kinematics problem in robotics." Journal of Software Engineering and Applications 3.03 (2010): 230

[7] Chirikjian, Gregory Scott. Theory and applications of hyper-redundant robotic manipulators. Diss. California Institute of Technology, 1992

[8] Sciavicco, Lorenzo, and Bruno Siciliano. "A solution algorithm to the inverse kinematic problem for redundant manipulators." IEEE Journal on Robotics and Automation 4.4 (1988): 403-410

[9] Batista, Josias, et al. "Dynamic Model and Inverse Kinematic Identification of a 3-DOF Manipulator Using RLSPSO." Sensors 20.2 (2020): 416

[10] Gómez, S., et al. "Design of a 4-DOF robot manipulator with optimized algorithm for inverse kinematics." International Journal of Mechanical and Mechatronics Engineering 9.6 (2015): 929-934

[11] Oh, Se- Young, David Orin, and Michael Bach. "An inverse kinematic solution for kinematically redundant robot manipulators." Journal of Robotic Systems 1.3 (1984): 235-249

[12] Zhao, Jie, et al. "A Synthetic Inverse Kinematic Algorithm for 7-DOF Redundant Manipulator." 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE, 2018

[13] Dereli, Serkan, and Raşit Köker. "Calculation of the inverse kinematics solution of the 7-DOF redundant robot manipulator by the firefly algorithm and statistical analysis of the results in terms of

speed and accuracy." Inverse Problems in Science and Engineering (2019): 1-13

[14] Kelemen, Michal, et al. "A novel approach for a inverse kinematics solution of a redundant manipulator." Applied Sciences 8.11 (2018): 2229

[15] Chirikjian, Gregory S., and Joel W. Burdick. "A hyper-redundant manipulator." IEEE Robotics & Automation Magazine 1.4 (1994): 22-29

[16] Zhao, Jingdong, Liangliang Zhao, and Hong Liu. "Motion planning of hyper-redundant manipulators based on ant colony optimization." 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2016

[17] Howard, David W., and Ali Zilouchian. "Application of fuzzy logic for the solution of inverse kinematics and hierarchical controls of robotic manipulators." Journal of Intelligent and Robotic Systems 23.2-4 (1998): 217-247

[18] Plam, R. "Control of a redundant manipulator using fuzzy rules." Fuzzy Sets and Systems 45.3 (1992): 279-298

[19] Xu, Yangsheng, and M. I. C. H. A. E. L. C. Nechyba. "Fuzzy inverse kinematic mapping: Rule generation, efficiency, and implementation." Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93). Vol. 2. IEEE, 1993

[20] Momani, Shaher, Zaer S. Abo-Hammour, and Othman MK Alsmadi. "Solution of inverse kinematics problem using genetic algorithms." Applied Mathematics & Information Sciences 10.1 (2016): 225

[21] Ankarali, Arif. "ANFIS inverse kinematics and precise trajectory tracking of a dual arm robot." Proceedings of the 2012 International Conference on Modelling, Simulation and Visualization Methods. 2012.

[22] Duka, Adrian-Vasile. "ANFIS based Solution to the Inverse Kinematics of a 3DOF planar Manipulator." Procedia Technology 19 (2015): 526-533

[23] Jasim, Wesam Mohammed. "Solution of inverse kinematics for SCARA manipulator using adaptive neuro-fuzzy network." International Journal on Soft Computing 2.4 (2011): 59

[24] Jang, Jyh-Shing Roger, Chuen-Tsai Sun, and Eiji Mizutani. "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]." IEEE Transactions on automatic control 42.10 (1997): 1482-1484

[25] MATLAB and Statistics Toolbox, The Math Works, Inc. Natick, Massachusetts, United States, (1994-2020)

[26] Mayorga, René V., and Pronnapa Sanongboon. "An artificial neural network approach for inverse kinematics computation and singularities prevention of redundant manipulators." Journal of Intelligent and Robotic Systems 44.1 (2005): 1-23.

[27] El-Sherbiny, Ahmed, Mostafa A. Elhosseini, and Amira Y. Haikal. "A comparative study of soft computing methods to solve inverse kinematics problem." Ain Shams Engineering Journal 9.4 (2018): 2535-2548.

[28] Duka, Adrian-Vasile. "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm." Procedia Technology 12.1 (2014): 20-27.

[29] Almusawi, Ahmed RJ, L. Canan Dülger, and Sadettin Kapucu. "A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242)." Computational intelligence and neuroscience 2016 (2016)

[30] Yu, Hao, and Bogdan M. Wilamowski. "Levenberg-marquardt training." Industrial electronics handbook 5.12 (2011): 1

[31] Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No.06, 1999, pp. 989–993, 1994

[32] Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996

[33] S. Habibkhah, "An Artificial Neural Networks Approach for Inverse Kinematics of Redundant Manipulators", M.A.Sc.

Thesis, Industrial Systems Engineering, University of Regina,, Canada, April 2020.

[34] Nakamura, Yoshihiko. Advanced robotics: redundancy and optimization. Addison-Wesley Longman Publishing Co., Inc., 1990

[35] Géradin, Michel, and Pierre Duysinx. "An Introduction to Robotics: Mechanical Aspects." University of Liege, Belgium, November 2004.

[36] MacKay, David J. C. "Bayesian interpolation." Neural computation. Vol. 4, No. 3, 1992, pp. 415–447.

[37] Foresee, F. Dan, and Martin T. Hagan. "Gauss-Newton approximation to Bayesian learning." Proceedings of the International Joint Conference on Neural Networks, June, 1997

Sources of funding

This Paper research was supported by a grant (RGPIN-2019-07269) from the Natural Sciences & Engineering Research Council (NSERC), Canada.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)