Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
June 2024, vol. 29, no. 1, pp. 24–34
https://doi.org/10.2478/acss-2024-0004
https://content.sciendo.com

ANN Approach for SCARA Robot Inverse Kinematics Solutions with Diverse Datasets and Optimisers

Rania Bouzid[1*], Hassène Gritli[2], Jyotindra Narayan[3]
[1,2]*Laboratory of Robotics, Informatics and Complex Systems (RISC Lab, LR16ES07), National Engineering School of Tunis, University of Tunis El Manar, BP 37, Le Belvédère, 1002, Tunis, Tunisia*
[1]*Polytechnic School of Tunisia, University of Carthage, B.P. 743, 2078 La Marsa, Tunisia*
[1,2]*Higher Institute of Information and Communication Technologies, University of Carthage, Technopole of Borj Cédria, Route de Soliman, BP 123, Hammam Chatt 1164, Ben Arous, Tunisia*
[3]*Mechatronics and Robotics Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Guwahati (IITG), Guwahati 781039, Assam, India*

***Abstract* – In the pursuit of enhancing the efficiency of the inverse kinematics of SCARA robots with four degrees of freedom (4-DoF), this research delves into an approach centered on the application of Artificial Neural Networks (ANNs) to optimise and, hence, solve the inverse kinematics problem. While analytical methods hold considerable importance, tackling the inverse kinematics for manipulator robots, like the SCARA robots, can pose challenges due to their inherent complexity and computational intensity. The main goal of the present paper is to develop efficient ANN-based solutions of the inverse kinematics that minimise the Mean Squared Error (MSE) in the 4-DoF SCARA robot inverse kinematics. Employing three distinct training algorithms – Levenberg-Marquardt (LM), Bayesian Regularization (BR), and Scaled Conjugate Gradient (SCG) – and three generated datasets, we fine-tune the ANN performance. Utilising diverse datasets featuring fixed step size, random step size, and sinusoidal trajectories allows for a comprehensive evaluation of the ANN adaptability to various operational scenarios during the training process. The utilisation of ANNs to optimise inverse kinematics offers notable advantages, such as heightened computational efficiency and precision, rendering them a compelling choice for real-time control and planning tasks. Through a comparative analysis of different training algorithms and datasets, our study yields valuable insights into the selection of the most effective training configurations for the optimisation of the inverse kinematics of the SCARA robot. Our research outcomes underscore the potential of ANNs as a viable means to enhance the efficiency of SCARA robot control systems, particularly when conventional analytical methods encounter limitations.**

***Keywords* – Artificial neural network, generated datasets, inverse kinematics, optimisers, SCARA robot.**

I. Introduction

The study of robot manipulators involves two fundamental aspects: forward kinematics and inverse kinematics [1]. Forward kinematics pertains to the determination of the end-effector position and orientation based on the joint angles or displacements of the robot manipulator [1]. It establishes a mapping from the joint space to the Cartesian space, allowing us to predict the spatial configuration of the robot's end-effector given specific joint configurations. This is crucial for understanding the robot's overall workspace and its ability to reach desired positions. Conversely, inverse kinematics involves solving for the joint angles or displacements required to achieve a desired end-effector position and orientation. It establishes the relationship between the Cartesian space and the joint space, enabling control and planning of the robot's movements. Inverse kinematics is particularly important for tasks such as trajectory planning, where precise control of the end-effector position is necessary [1], [2]. Both forward and inverse kinematics play integral roles in the design, control, and optimisation of robot manipulators, contributing to their efficiency and versatility in various applications, including manufacturing, automation, and research.

Inverse kinematics is generally considered more challenging than forward kinematics due to its inherent complexity and non-uniqueness. While forward kinematics involves determining the end-effector's position and orientation based on known joint angles, inverse kinematics requires solving mathematical equations to find the joint angles or displacements needed to achieve a desired end-effector configuration. Inverse kinematics problems often have multiple solutions, and finding a unique and optimal solution can be intricate, especially for robots with redundant degrees of freedom or complex geometries [2], [3]. The nonlinearity, coupled equations, and potential singularities in the solution space make inverse kinematics problems computationally demanding. Moreover, the sensitivity to inaccuracies in sensor measurements or

* Corresponding author's e-mail: rania.bouzid@istic.ucar.tn
Article received 2024-01-08; accepted 2024-07-12

modelling errors adds another layer of complexity. As a result, addressing inverse kinematics challenges is crucial for achieving precise control and accurate planning of robotic manipulator movements in various applications. The conventional techniques such as geometrical and analytical suffer from non-closed formulations [4], [5], and numerical ones involve multiple iterations leading to computationally expensive processes [6].

In recent years, significant works using intelligent techniques have been carried out by researchers in the inverse kinematics modelling of robot manipulators [7]–[12]. The study [7] focused on determining the angles between links when the position of the end effector is known. The proposed solution employs the Adaptive Neuro Fuzzy Interface System (ANFIS), a hybrid intelligent system blending neural network learning capabilities with fuzzy logic. The research [8] aimed to enhance the efficiency and accuracy of the inverse kinematics algorithm in robotics by leveraging an improved backpropagation ANN and introducing additional techniques to address specific challenges associated with traditional approaches. The simulation results suggest that the proposed algorithm performs better in solving the inverse kinematics problem for robotics with six degrees of freedom (6-DoF). The authors in [9] proposed formulating the inverse kinematics problem for redundant manipulators as a quadratic programming optimisation problem. Quadratic programming is a mathematical optimisation technique used to solve constrained optimisation problems with a quadratic objective function. In their research, Habibkhah et al. [10] presented a novel approach using ANNs for solving the inverse kinematics of a 3-DoF redundant manipulator. The inclusion of a virtual auxiliary function and the testing in different scenarios suggested a comprehensive evaluation of the effectiveness of the proposed method in achieving accurate and flexible manipulator control. Kamlesh et al. [11] addressed the intricate challenge of solving inverse kinematics in the design of robotic manipulators. The focus is on a 5-axis serial-link robotic manipulator with rotary joints, and they propose the use of a Deep Artificial Neural Network (DANN) model to tackle the inverse kinematics problem. Wagga et al. [12] addressed challenges in analytical inverse kinematics for high-DoF robotic manipulators. It introduces Deep Learning networks, with Bidirectional Long–Short Term Memory outperforming. The study validates results with noise-injected data and compares analytical and soft computing solutions, emphasising neural network singularity avoidance. Real-world simulations use the RoboDK simulator, evaluating network performance based on manipulator complexity. The proposed strategies apply to manipulators with different degrees of freedom. Recently, Bouzid et al. [13] addressed the problem of inverse kinematics for a 2-DoF manipulator robot using different ANN architectures. They used also three different optimisers and three different datasets to investigate the efficiency of these ANN models by varying different hyper-parameters to find the optimal ANN architecture.

A SCARA robot, standing for Selective Compliance Articulated Robot Arm, is an industrial robotic manipulator engineered for applications requiring precise motion within a single horizontal plane. Its distinctive design comprises two rotary joints for positioning the end-effector and an additional vertical prismatic joint, which adds flexibility to vertical movements [14]. This unique attribute allows SCARA robots to maintain rigidity and precision in the horizontal plane, making them particularly suitable for tasks demanding high-speed, accurate operations. Their operational scope primarily lies in planar motion tasks such as pick-and-place actions, assembly, dispensing, and other activities carried out on flat surfaces. Despite their limited reach due to their planar nature, SCARA robots excel in industries like electronics manufacturing, automotive assembly, and pharmaceuticals. They offer a cost-effective solution for a variety of applications, benefiting from simplified kinematics, ease of programming, and compatibility with various end-effectors, such as grippers and soldering tools. The article [15] delves into the world of SCARA PRR-type robot manipulators, exploring their design, control, and simulation. It details the creation of kinematic equations, the development of software for handling Cartesian and joint velocities, and the implementation of trajectory planning. The research culminates in practical experiments where the robot adeptly picks and places objects within its workspace, showcasing its real-world performance. In essence, the article combines mechanical engineering, kinematics, software development, and unique control strategies to enhance our understanding and application of SCARA robots.

The article [16] proposed a structured ANN model, specifically a multi-layered perceptron neural network (MLPNN) trained using gradient descent-type learning rules, to address the inverse kinematics problem for a 4-DoF SCARA manipulator. The study involves an exploration of various ANN configurations, ultimately finding that the MLPNN exhibits minimal Mean Squared Error (MSE), showing promise for accurately determining the joint angles required to control the SCARA manipulator's end effector. Narayan et al. [17] explored the inverse kinematics for a 4-DoF SCARA robot using an adaptive neuro-fuzzy inference system (ANFIS) and Gaussian membership function. ANFIS, combining fuzzy inference and neural network approaches, addresses the challenging nature of inverse kinematics, offering satisfactory results validated in MATLAB simulations. Demby et al. [18] addressed the challenge of computing the inverse kinematics for robotic arms, crucial for determining joint movements to achieve a desired end-effector pose. Recognising the complexity of this task, especially in higher-DoF systems, the study explored the application of ANNs and ANFIS for solving inverse kinematics problems in 4- to 7-DoF serial robots. Unlike task-specific solvers, the focus was on learning entire robot workspaces, making any end-effector pose reachable from any current pose. The experiments revealed that both ANN and ANFIS could converge to the underlying inverse kinematics function but with some approximation errors and considerations about time and effort compared to alternative methods in the literature. Jimenez et al. [19] introduced a novel approach for modelling a 3-DoF open SCARA using quaternions and neural networks to solve the inverse kinematics

problem. The ANN structure comprised three hidden layers with 25 neurons each and one output layer, trained using Bayesian Regularization (BR) backpropagation. The inverse kinematic problem is formulated as a system of six nonlinear equations and unknowns. Comparative testing using a straight path demonstrates the results against the Newton–Raphson method.

Drawing from prior research, it is apparent that there is a lack of studies comparing different methodologies in configuring datasets and hyper-parameters within neural networks to address the challenge of inverse kinematics in robot manipulators. Our research makes a significant contribution by introducing an intelligent method based on ANN tailored specifically for solving the inverse kinematics problem in a 4-DoF SCARA manipulator. The primary objective of the ANN is to infer the joint variables based on the spatial coordinates and orientations of the manipulator's end-effector. Our proposed network employs deep learning methodologies to facilitate rapid convergence during training, with a key emphasis on achieving minimal MSE and training error, ideally driving them towards zero. Utilising the MATLAB toolbox, the ANN is trained on three generated distinct datasets: a random-step-size dataset, a fixed-step-size dataset, and a sinusoidal-signal-based dataset with varying frequencies. The subsequent testing and verification stages aim to assess the reliability of the trained ANN in minimising approximation errors and accurately estimating inverse kinematics. The research demonstrates the effectiveness of the trained ANN in successfully addressing the inverse kinematics problem for the SCARA manipulator. The network capability to minimise approximation errors and provide accurate estimations across various hyper-parameters underscores its potential as a reliable tool in robotic kinematics.

The subsequent sections of our paper are structured as follows: Section II delineates the inverse kinematics problem for a robot. In Section III, we provide an overview of the proposed evolutionary approach and encapsulate the simulation experiments conducted on a SCARA robot. In Section IV, the training results are presented. The concluding Section V offers insights into the conclusions drawn from the study and outlines potential avenues for future research.

II. Problem Formulation

The 4-DoF SCARA manipulator robot is illustrated in Fig. 1. The robotic arm is configured in an RRPR arrangement, and a detailed description of the DH Parameters of the SCARA Robot is provided in Table I. The values of the constant parameters (length of different segments of the robot), namely d_1, a_1, a_2, and $d4$, are also provided in Table I. The SCARA robot is composed of three rotary articular joints and a unique translation/prismatic articular joint. The joint variables of the four articulations are, in the order, θ_1, θ_2, d_3 and θ_4.

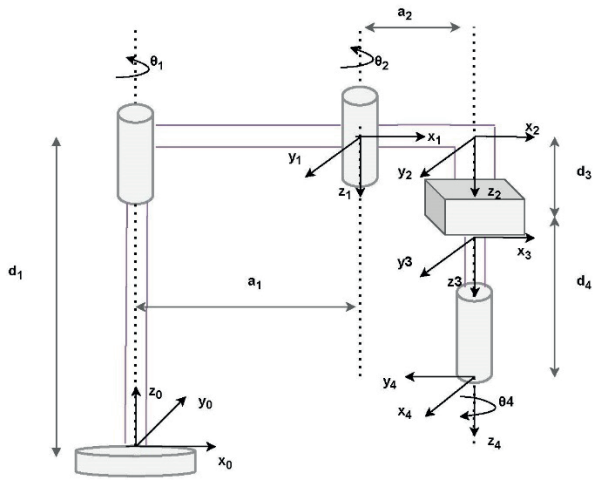


Fig. 1. Representation of the 4-DoF SCARA robotic manipulator.

TABLE I
DH Parameters of the 4-DoF SCARA Manipulator Robot Illustrated in Fig. 1

Joint number	Joint angle (rad)	Joint offset (mm)	Link length (mm)	Twist angle (rad)
1	θ_1	$d_1 = 400$	$a_1 = 250$	0
2	θ_2	0	$a_2 = 150$	π
3	0	d_3	0	0
4	θ_4	$d_4 = 150$	0	0

Relying on the DH method and the corresponding parameters given in Table I, the forward kinematics model of the Cartesian position (defined by the coordinates x, y, z) and the orientation angle (φ) of the end-effector of the 4-DoF SCARA robot, is described by the following equations [17]:

$$x = a_1 \times \cos(\theta_1) + a_2 \times \cos(\theta_1 + \theta_2); \quad (1)$$
$$y = a_1 \times \sin(\theta_1) + a_2 \times \sin(\theta_1 + \theta_2); \quad (2)$$
$$z = d_1 - d_4 - d_3; \quad (3)$$
$$\varphi = \theta_1 - \theta_2 - \theta_4. \quad (4)$$

The inverse kinematics of the SCARA manipulator robot presented in Fig. 2 can be represented with a set of equations that describes the joint angles needed to position the end-effector (typically the robot's hand or tool) at a desired Cartesian position and orientation (typically represented as roll, pitch, and yaw angles or a rotation matrix).

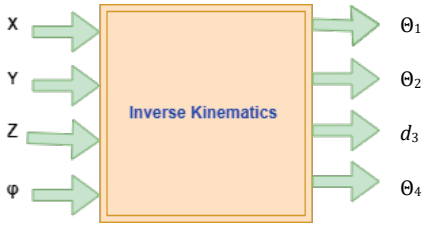


Fig. 2. Representation of the inverse kinematics model of the 4-DoF SCARA robot.

In our current study focusing on the inverse kinematics of SCARA robots, we are working with the 4-DoF model illustrated in Fig. 2. Our objective is to determine the inverse kinematic model, denoted as $q = \Phi(Z)$, where q represents the joint variables (θ_1, θ_2, d_3 and θ_4) of the robot, and $\Phi(Z)$ is a nonlinear function that relates these joint angles to the robot's end-effector position (x, y, and z) and orientation (φ) in the 3D space. The specific form of this inverse kinematic model is currently unknown and is a subject of investigation in our research. This model plays a crucial role in enabling the robot to accurately position and orient its end-effector in a given workspace, and our study seeks to define this relationship for our 4-DoF SCARA robot model.

III. Proposed Methodology

In our proposed methodology for addressing inverse kinematics challenges in the 4-DoF SCARA robot, we introduce a novel approach that leverages the power of ANNs to tackle these problems. We diversify our approach by utilizing three different datasets generated through the forward kinematics model, which is defined by Figs. (1)–(4), of the SCARA robot for training and evaluation. These datasets consist of fixed step size-based data, random step size-based data, and sinusoidal signal-based data, representing various operating conditions and challenges that SCARA robots may encounter. Each joint variable of the robot is defined within some specific interval as illustrated in Table II.

TABLE II
Ranges of the Four Joint Variables of the 4-DoF SCARA Robotic Manipulator

Joint variable	Variation range
θ_1	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
θ_2	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
d_3	$[-0.2\,, 0.2]$
θ_4	$[-\frac{\pi}{2}, \frac{\pi}{2}]$

A. Generated Datasets

In the present work, three datasets will be generated and then considered for the design and optimisation of the ANN models. These datasets are generated through the forward kinematics described by Figs. (1)–(4).

1) Fixed-step-size Dataset

This set of data includes the joint variables of the SCARA manipulator robot and other constant parameters as shown in Tables I and II. This first type of dataset is generated via expressions (1)–(4) by performing iterations using a specific step size for the joint variables according to the following expressions:

$$\text{step}_{\theta_i} = \frac{\theta_i^{\max} - \theta_i^{\min}}{N} = \frac{\pi}{N}, \quad \forall\ i = 1,2,4; \tag{5}$$

$$\text{step}_{d_3} = \frac{d_3^{\max} - d_3^{\min}}{N} = \frac{0.4}{N}. \tag{6}$$

Note that in (5) and (6), N serves as a parameter that defines both the fixed step size and the overall length of the generated fixed-step-size dataset. For this work, we chose $N = 20$ to build the dataset in question. This choice resulted in a dataset size of N to the power of 4, which is big enough. The rotation joints have a fixed step of $\pi/20$ and the translation joint has a fixed step of 0.02.

2) Random-step-size Dataset

We use random datasets in the present work. These datasets have data points that are chosen or made without any pattern. These sets of data are not organised in any specific way or pattern; instead, their individual items are chosen randomly inside the intervals that were specified in Table II. We use random step sizes as an important part. These step sizes are used to set up ranges for important measurements like θ_1, θ_2, d_3 and θ_4. The ranges are calculated using these equations:

$$\theta_1^{\text{range}} = (\theta_1^{\max} - \theta_1^{\min}) \times \text{rand}(1, N) + \theta_1^{\min}; \tag{7}$$

$$\theta_2^{\text{range}} = (\theta_2^{\max} - \theta_2^{\min}) \times \text{rand}(1, N) + \theta_2^{\min}; \tag{8}$$

$$d_3^{\text{range}} = (d_3^{\max} - d_3^{\min}) \times \text{rand}(1, N) + d_3^{\min}; \tag{9}$$

$$\theta_4^{\text{range}} = (\theta_4^{\max} - \theta_4^{\min}) \times \text{rand}(1, N) + \theta_4^{\min}. \tag{10}$$

Note that in (7)–(10), rand is a function that makes random numbers chosen between 0 to 1. Furthermore, the parameter N shows how big the dataset is. We will use $N = 1000$. When working with the 4-DoF SCARA robot, how we show variables for a set of data can be very important for how well the robot can move. This is shown in Fig. 3. The robot's position and how it is facing are defined by the four joint variables θ_1, θ_2, d_3 and θ_4. When we pick these variables randomly between $-\pi/2$ and $\pi/2$ it shows all the different ways the robot can be positioned. We can learn a lot about how the robot moves by making a graph of each angle. This plot in Fig. 3 shows how flexible the robot is and how it can handle different real-life problems.

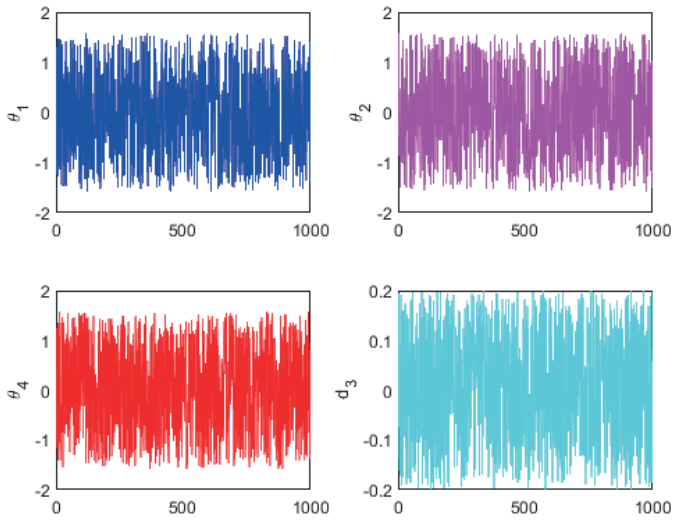


Fig. 3. Representation of variations of the four joint variables of the 4-DoF SCARA robotic manipulator generated using the random-step-size dataset.

3) Sinusoidal-signal-based Dataset

In this work, we employ the third type of dataset that captures variations in sinusoidal signals, a crucial component for multiple facets of our investigation. This dataset contains vital elements that characterise these signals, such as their frequency and phase components. When delving into this sinusoidal-signal-based dataset, our primary focus revolves around understanding the frequency – a foundational and ubiquitous feature. Table III illustrates the different parameters and their values used for the generation of the four sinusoidal signals for the four joint variables θ_1, θ_2, d_3 and θ_4 of the SCARA robotic manipulator, and, therefore, for the generation of the sinusoidal-signal-based dataset using the forward kinematics of the robot defined by (1)–(4).

TABLE III
DIFFERENT PARAMETERS EMPLOYED TO SIMULATE THE SINUSOIDAL SIGNALS AND THEN TO GENERATE THE SINUSOIDAL-SIGNAL-BASED DATASET

Parameter	Value
Frequency ω_1 for θ_1	1.5
Frequency ω_2 for θ_2	5
Frequency ω_3 for d_3	10
Frequency ω_4 for θ_4	10
Phase ϕ_1 for θ_1	0
Phase ϕ_2 for θ_2	$\frac{\pi}{4}$
Phase ϕ_3 for d_3	$\frac{\pi}{3}$
Phase ϕ_4 for θ_4	$-\frac{\pi}{4}$
Amplitude Γ_1 for θ_1	$\frac{\pi}{2}$
Amplitude Γ_2 for θ_2	$\frac{\pi}{2}$
Amplitude Γ_3 for d_3	0.2
Amplitude Γ_4 for θ_4	$\frac{\pi}{2}$

The sinusoidal variations for the four joint variables θ_1, θ_2, d_3 and θ_4, are computed using the following equations:

$$\theta_1^{\text{range}} = \Gamma_1 \times \sin(\omega_1 \times \xi_1 + \phi_1); \quad (11)$$
$$\theta_2^{\text{range}} = \Gamma_2 \times \sin(\omega_2 \times \xi_2 + \phi_2); \quad (12)$$
$$d_3^{\text{range}} = \Gamma_3 \times \sin(\omega_3 \times \xi_3 + \phi_3); \quad (13)$$
$$\theta_4^{\text{range}} = \Gamma_4 \times \sin(\omega_4 \times \xi_4 + \phi_4). \quad (14)$$

In (11)–(14), the parameters ξ_1, ξ_2, ξ_3, and ξ_4, are variable scalars that vary between two limits associated to the corresponding intervals, as illustrated in Table II.

The sinusoidal-signal-based dataset intricately connects to the trio of frequency, phase, and amplitude, all of which are vital in defining the behavior of these signals, symbolized by the parameters of the four joint variables, θ_1, θ_2, d_3 and θ_4, of the SCARA robot. By utilizing specific values of these parameters as indicated in Table III, we can then create a series of detailed subplots. Each of these subplots offers an in-depth view of how the signal reacts to changes in frequency, phase, and amplitude.

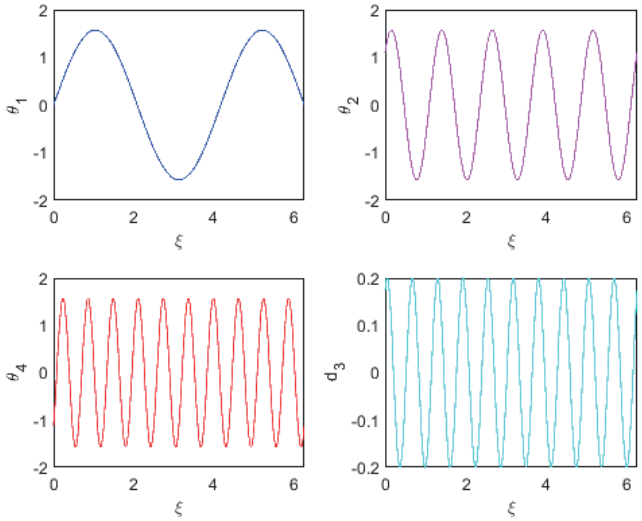


Fig. 4. Representation of the generated sinusoidal signals using the parameters in Table III.

Figure 4 visually represents these subplots, serving as a vital tool to elucidate the complex interplay between the signal inherent traits and the input variables. This aids in obtaining a holistic view of how the dataset transforms when parameters vary. Such a graphical representation, highlighting its importance, becomes essential for professionals and scholars striving to grasp the subtle intricacies of systems deeply impacted by these foundational elements.

B. Training Algorithms Used

To ensure the effectiveness and robustness of our solution, we employ three distinct optimisation algorithms.

1) Levenberg–Marquardt (LM):

The LM algorithm is an optimisation technique designed for solving nonlinear least squares problems [20], [21], widely employed in diverse fields ranging from computer vision to machine learning. This algorithm amalgamates principles from the steepest descent and Gauss-Newton methods. A distinctive feature of LM is the introduction of a damping parameter, denoted as λ, which dynamically adjusts the step size during optimisation. This adaptive mechanism allows the algorithm to navigate through various objective functions, enhancing stability and convergence. Operating iteratively, LM algorithm continually refines parameter estimates by considering both gradient information and the curvature of the objective function. Its favourable convergence properties, especially in the presence of ill-conditioned problems or noisy data, make LM particularly effective for tasks such as curve fitting, parameter estimation, and optimisation where the objective function is expressed as the sum of squared differences between observed and predicted values.

2) Bayesian Regularization (BR):

The BR algorithm is a regularisation approach widely used in machine learning and statistical modelling to enhance model generalisation and mitigate overfitting [22]. It incorporates Bayesian principles by introducing a prior distribution over the model parameters, reflecting initial beliefs about those

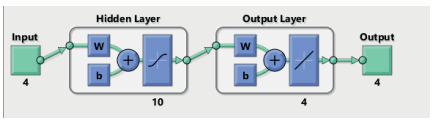
parameters before observing the data. As the model is trained, this prior is combined with the likelihood function, representing the probability of observing the data given the parameters, to yield a posterior distribution. The regularisation term, derived from this posterior distribution, is then added to the objective function. This term acts as a penalty, discouraging overly complex models and helping to strike a balance between fitting the training data and preventing overfitting. Hyper-parameters associated with the prior distribution allow for fine-tuning the strength of regularisation. The BR provides a principled and effective means of incorporating prior knowledge and improving model performance, particularly in scenarios with limited data or when there are reasonable expectations about the parameter values.

3) Scaled Conjugate Gradient (SCG):

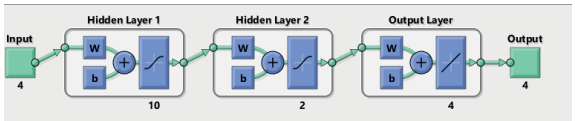
The SCG algorithm is a powerful optimisation method commonly employed for training ANNs [23]. Introduced by Møller in 1993, the SCG belongs to the class of conjugate gradient methods, making it particularly adept at navigating large-scale parameter spaces. Notably, SCG incorporates an adaptive learning rate mechanism that dynamically adjusts the learning rates for each parameter during training. This adaptive approach accelerates convergence without the risk of overshooting the minimum. One of SCG strengths lies in its efficiency in handling scenarios with a high number of parameters, making it well-suited for training complex neural networks with numerous weights and biases. Furthermore, the algorithm is designed to mitigate the issue of getting stuck in saddle points during optimissation, enhancing its applicability in the realm of deep learning.

IV. Results of Training

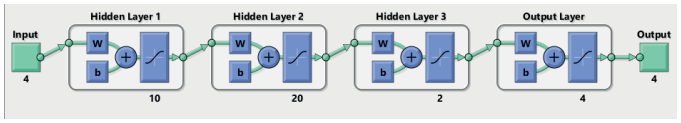
The comprehensive approach that will be achieved in the sequel is specifically tailored for the 4-DoF SCARA robotic manipulator, addressing the intricacies of their kinematic configurations and expanding the applicability of our solution to a wide array of real-world scenarios. By integrating these elements, we aim to enhance the accuracy and adaptability of inverse kinematics solutions for the SCARA robot, further advancing their capabilities in a variety of tasks and applications. Throughout our testing phase, which spanned 1000 epochs, we delved into a multitude of configurations for hidden layers, spanning from 1 to 5 layers for each specified layer count (as visualized in Fig. 5), and encompassed diverse dataset sizes.



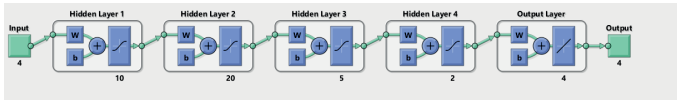
(a) ANN architecture with one hidden layer: ANN 1 (10).



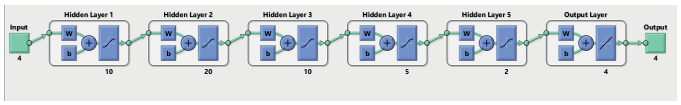
(b) ANN architecture with two hidden layers: ANN 2 (10,2).



(c) ANN architecture with three hidden layers ANN 3 (10,20,2).



(d) ANN architecture with four hidden layers: ANN 4 (10,20,5,2)\.



(e) ANN architecture with five hidden layers: ANN 5 (10,20,10,5,2).

Fig. 5. Different ANN architectures with a different number of hidden layers and their corresponding number of neurons.

A. Performances Analysis Using the LM Algorithm

For the SCARA robot and the solution of its inverse kinematic problem, the training process using the LM algorithm produced varying results with different step sizes as shown in Fig. 6. When a fixed-step-size dataset was employed, the MSE reached an impressively low value of 0.00013318 at the 132nd epoch for an ANN with five hidden layers. In contrast, using a random-step-size dataset, a considerably higher MSE of 0.16364 was observed at the 32nd epoch. Lastly, when dealing with a dataset based on a sinusoidal signal, the MSE plummeted to 9.1083e-05 at the 387th epoch. Detailed results can be found in Table IV. Notably, the highest MSE was associated with the random-step-size dataset for the ANN with five hidden layers, reaching a value of 0.20024 at the 13th epoch.

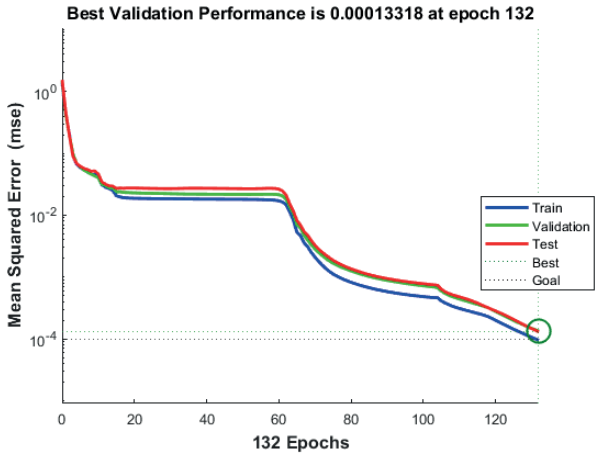
TABLE IV
Results of Training with ANNs for Different Datasets Using the LM Algorithm

Model	Fixed dataset		Random dataset		Sinusoidal dataset	
	MSE	Epoch	MSE	Epoch	MSE	Epoch
ANN 1	0.00044232	199	0.17035	107	9.1083e-05	387
ANN 2	0.0012819	275	0.17702	92	0.00020547	1000
ANN 3	0.020101	72	0.16364	32	0.00011501	237
ANN 4	0.022521	66	0.17123	45	9.2739e-05	236
ANN 5	0.00013318	132	0.20024	13	0.00010669	187

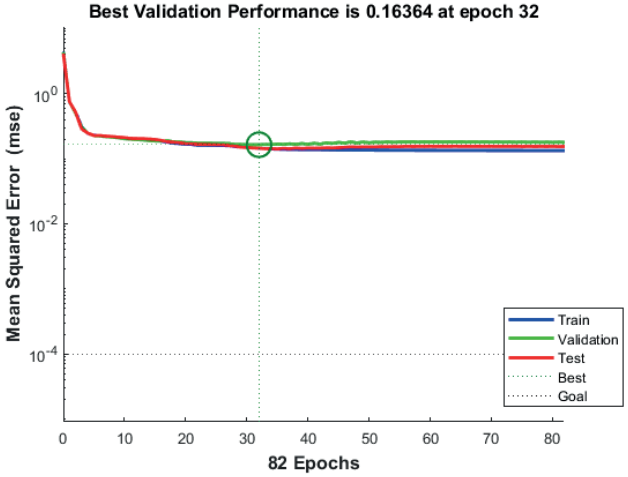
The elevated MSE of 0.20024 at the 13th epoch, specifically in the case of the ANN with five hidden layers utilising a random-step-size dataset, had a profound and adverse impact on the overall performance of the SCARA robot's inverse kinematics solution. This high error value led to a skewed and heavily distorted error histogram, as seen in Fig. 7. As a result, the representation of the robot's kinematic solution became significantly compressed, making it challenging to accurately predict and control the robot's end-effector position and orientation. Furthermore, the increased noise in the regression task's output caused by this substantial error hampered the precision and reliability of the robot's movements, ultimately affecting its operational efficiency and accuracy in executing tasks (Fig. 7).

B. Performances Analysis Using the BR Algorithm

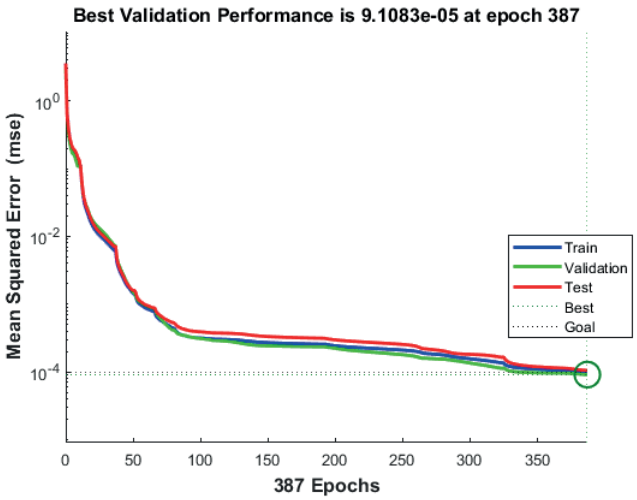
In the BR algorithm during training, we observed distinct performance outcomes based on various configurations as presented in Table IV. When employing a fixed-step-size dataset, the MSE reached an impressively low value of 0.00012062 at the 208th epoch for a network with five hidden layers. However, with a random-step-size dataset, the MSE peaked at 0.14885 by the 24th epoch for a network comprising four hidden layers. On a sinusoidal-signal-based dataset, the MSE plummeted to 8.9403e-05 at the 95th epoch. The results shown in Fig. 8 reveal that the highest MSE was associated with the random-step-size configuration, attaining 0.18105 at the 57th epoch for a network with two hidden layers. This elevated error impacted the error histogram, leading to an increase in its magnitude, albeit still less than the error histogram of the LM algorithm. Furthermore, the regression task in the BR algorithm displayed some noise but less than the corresponding task in the LM algorithm. During training, the regression coefficient (R) stands at a commendable 0.87428, demonstrating a relatively strong relationship between the predicted and actual values. Additionally, during the validation phase, the R-value remains robust at 0.83888, underscoring the model's generalisation capability. Even in the testing phase, where the model faces unseen data, the R-value stands at a respectable 0.86058, as demonstrated in Fig. 9.



(a) Fixed-step-size dataset.

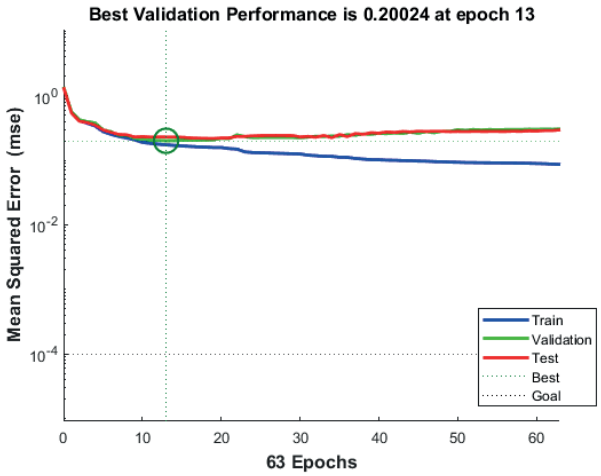


(b) Random-step-size dataset.

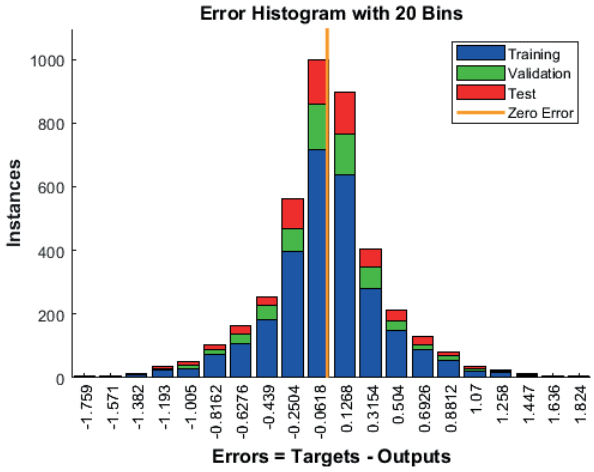


(c) Sinusoidal-signal-based dataset.

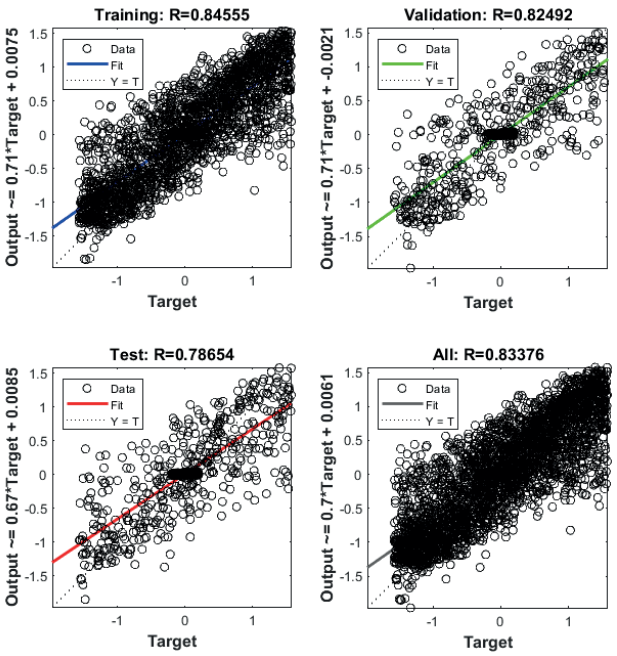
Fig. 6. Best ANN performances obtained by training the different ANN architectures using the LM algorithm.



(a) MSE using random-step-size dataset.

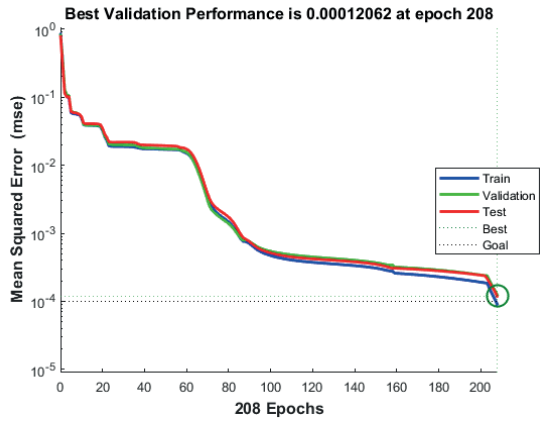


(b) Error histogram using random-step-size dataset.

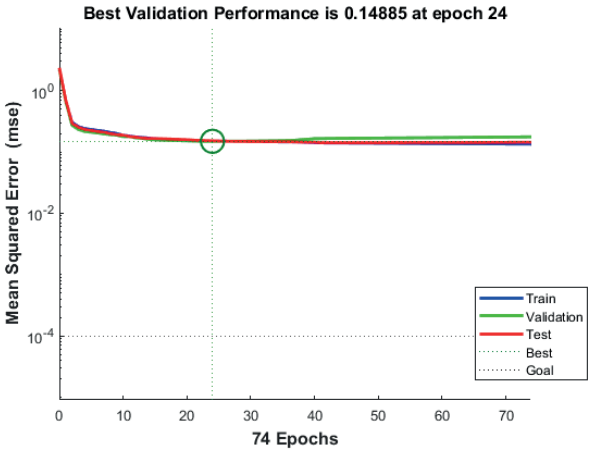


(c) Regression using random-step-size dataset.

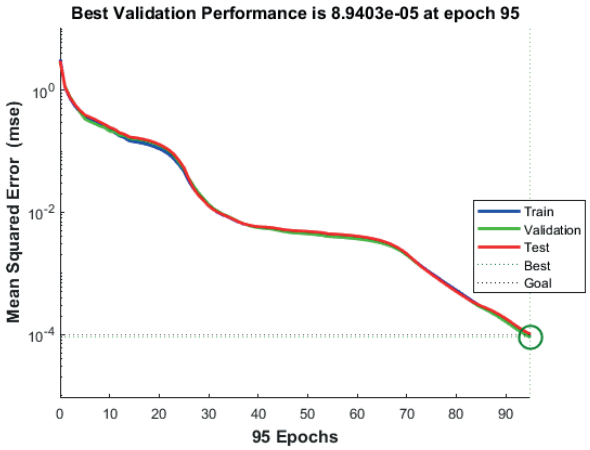
Fig. 7. ANN performances of highest MSE obtained using the random-step-size dataset.



(a) Fixed-step-size dataset.

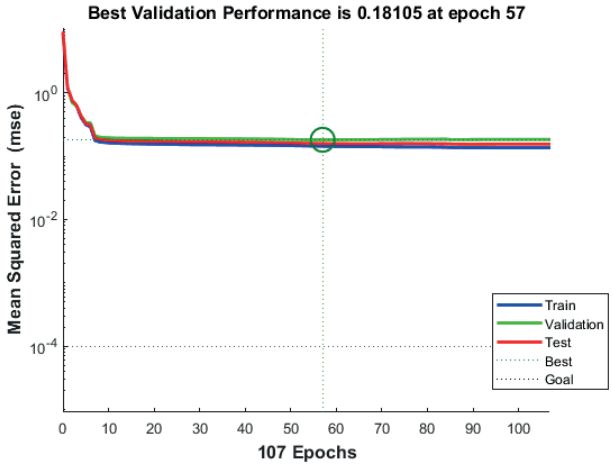


(b) Random-step-size dataset.

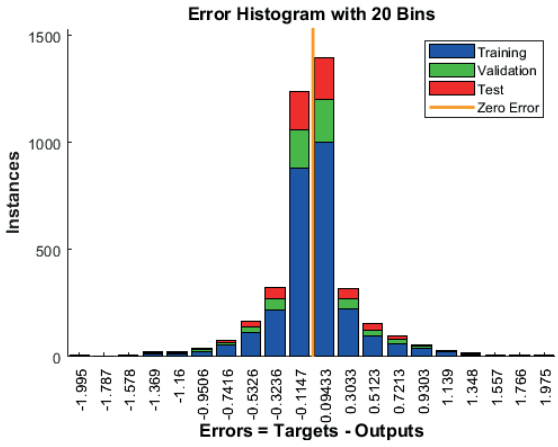


(c) Sinusoidal-signal-based dataset.

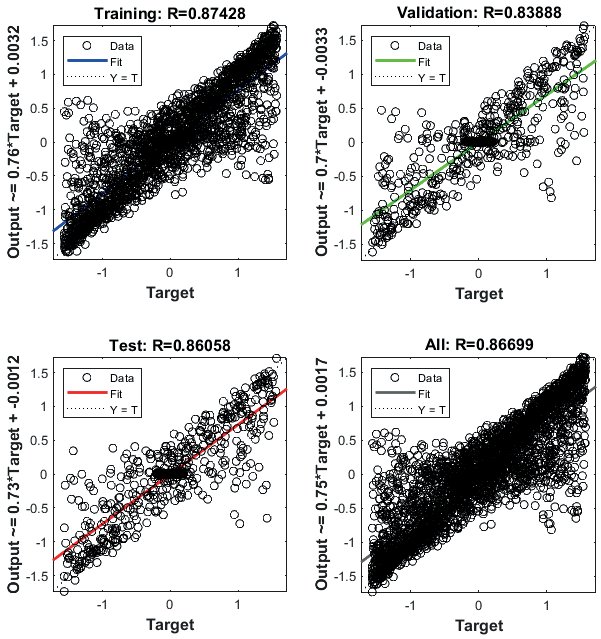
Fig. 8. Best performances of different ANNs obtained by training them with the BR algorithm.



(a) MSE with random-step-size dataset.



(b) Error histogram with random-step-size dataset.



(c) Regression with random-step-size dataset.

Fig. 9. Performances of the ANN with the highest MSE obtained using the random-step-size dataset.

TABLE V

RESULTS OF TRAINING WITH ANN FOR DIFFERENT DATASETS USING THE BR ALGORITHM

Model	Fixed dataset		Random dataset		Sinusoidal dataset	
	MSE	Epoch	MSE	Epoch	MSE	Epoch
ANN 1	0.00016086	501	0.15088	39	0.00011219	386
ANN 2	0.00018703	1000	0.18105	57	0.00010052	705
ANN 3	0.00035146	468	0.17814	24	0.00010583	365
ANN 4	0.039348	13	0.14885	24	9.7481e-05	127
ANN 5	0.00012062	208	0.16759	177	8.9403e-05	95

C. Performance Analysis Using the SCG Algorithm

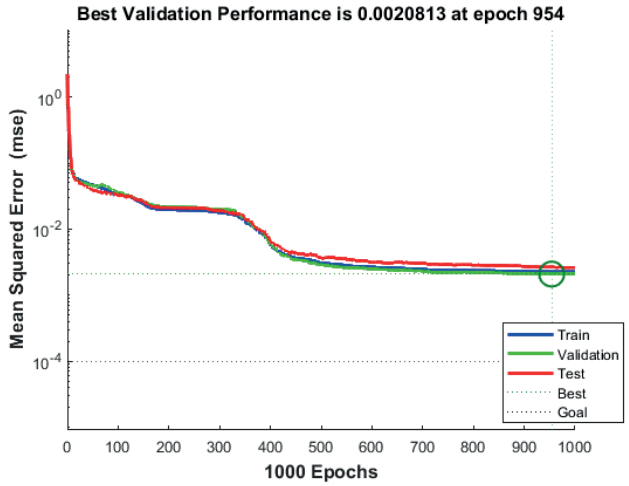
Moving on to the SCG algorithm, we can observe varying results shown in Table VI for different data configurations. For a fixed-step-size dataset, the best performance was achieved with an ANN having two hidden layers with 10 and 2 units, where the MSE reached a remarkable low of 0.0020813. This level of accuracy was attained after an extensive training process that spanned 954 epochs, indicating the algorithm's ability to finely tune the model's weights. In contrast, for the random-step-size dataset, the SCG algorithm yielded an MSE of 0.16942, with convergence occurring in a notably faster 335 epochs, albeit still significantly higher than the fixed-step-size dataset. Furthermore, when dealing with a sinusoidal-signal-based dataset, the SCG algorithm achieved an MSE of 0.025053, following a lengthy training period of 1000 epochs, utilising an ANN with two hidden layers containing 10 and 2 units (as seen in Fig. 10).

In this specific context, the most elevated MSE emerges when employing the random-step-size dataset, peaking at 0.23678 after 277 epochs. This high MSE signifies a suboptimal representation of regression values, impacting the accuracy of the model's predictions for training, testing, and validation. Consequently, this discrepancy explains the increased noise observed in the regression graph, underscoring the algorithm's challenges in accurately capturing and representing the underlying patterns in the data across different phases of evaluation (Fig. 11).

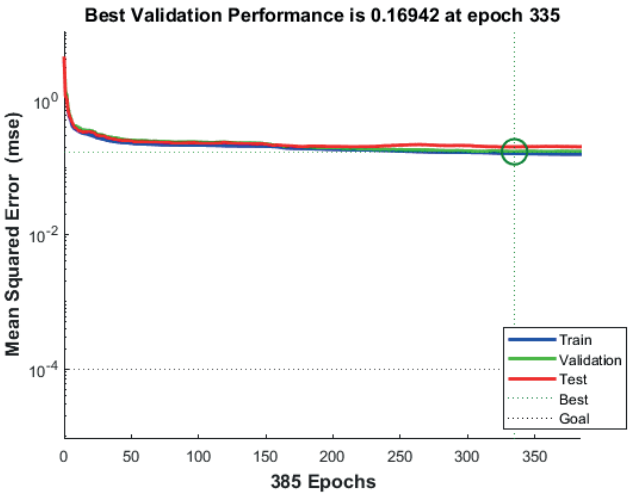
TABLE VI

RESULTS OF TRAINING WITH ANN FOR DIFFERENT DATASETS USING THE SCG ALGORITHM

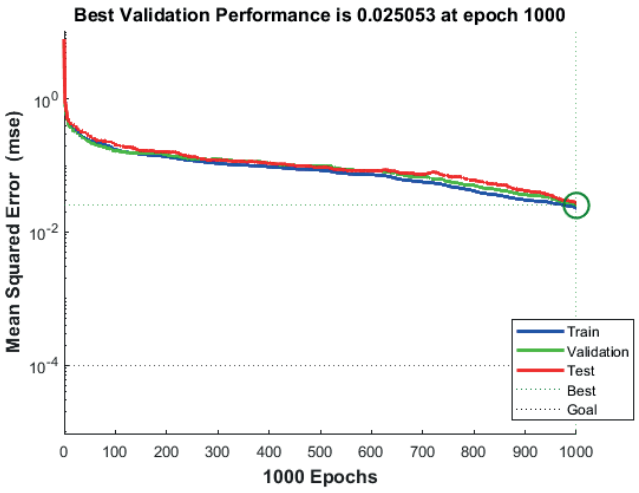
Model	Fixed dataset		Random dataset		Sinusoidal dataset	
	MSE	Epoch	MSE	Epoch	MSE	Epoch
ANN 1	0.0029239	985	0.16942	335	0.078042	882
ANN 2	0.0020813	954	0.19131	419	0.025053	1000
ANN 3	0.019836	953	0.22003	462	0.17232	364
ANN 4	0.023319	974	0.23678	277	0.051377	999
ANN 5	0.038353	866	0.21713	248	0.030405	1000



(a) Fixed-step-size dataset.

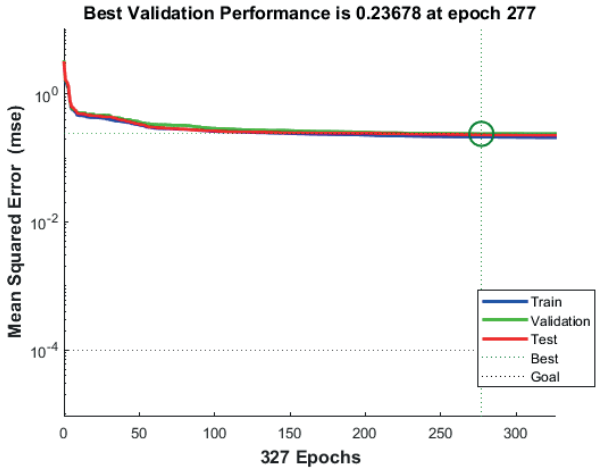


(b) Random-step-size dataset.

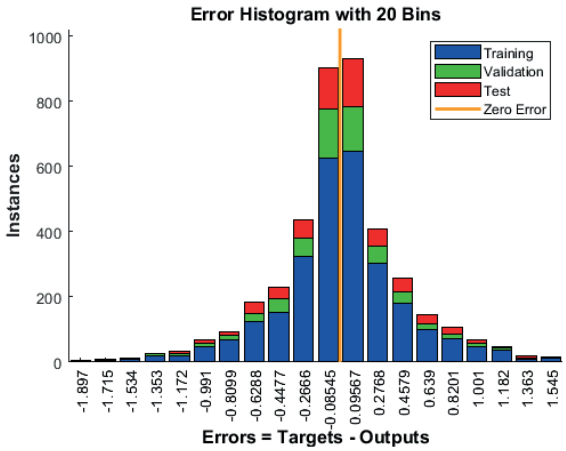


(c) Sinusoidal-signal-based dataset.

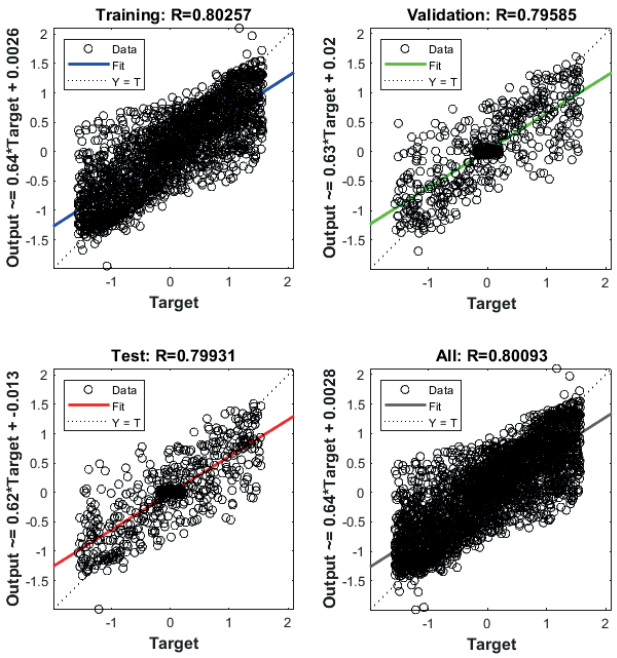
Fig. 10. Performances of the best ANN models obtained by generating the three datasets and via the SCG algorithm.



(a) MSE with random-step-size dataset.



(b) Error histogram with random-step-size dataset.



(c) Regression with random-step-size dataset.

Fig. 11. ANN's performances of highest MSE obtained using the random-step-size dataset.

V. Conclusion and Future Research

Our research revolves around a comprehensive examination of a 4-DoF SCARA robot, with a special emphasis on enhancing its ability to calculate inverse kinematics using ANNs. We introduced and applied intelligent techniques based on ANNs, carefully assessing their effectiveness across various algorithmic approaches. Concurrently, our study sheds light on the intricate field of inverse kinematics, highlighting the pivotal role played by ANNs in ensuring the smooth operation of the SCARA robot.

As a future work, our goal is to explore advanced neural network architectures, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), with the expectation that this extensive exploration will significantly boost the capabilities of SCARA robots, particularly in navigating complex and dynamically changing operational environments. Our inquiry also extends to the fusion of sensor data and the implementation of adaptive learning strategies, designed to bolster the resilience of these solutions in practical, real-world applications. It is of utmost importance to rigorously evaluate the scalability and adaptability of these innovations, with specific attention to their relevance in the domain of 6-DoF industrial robots.

References

[1] S. Kucuk and Z. Bingul, *Robot Kinematics: Forward and Inverse Kinematics*. INTECH Open Access Publisher London, UK, 2006. https://doi.org/10.5772/5015

[2] R. Singh, V. Kukshal, and V. S. Yadav, "A review on forward and inverse kinematics of classical serial manipulators," in *Advances in Engineering Design. Lecture Notes in Mechanical Engineering*, P.K. Rakesh, A.K. Sharma, I. Singh, Eds. Springer, Singapore, 2021, pp. 417–428. https://doi.org/10.1007/978-981-33-4018-3_39

[3] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer Graphics Forum*, vol. 37, no. 6, Sep. 2018, pp. 35–58. https://doi.org/10.1111/cgf.13310
[4] A. El-Sherbiny, M. A. Elhosseini, and A. Y. Haikal, "A comparative study of soft computing methods to solve inverse kinematics problem," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2535–2548, Dec. 2018. https://doi.org/10.1016/j.asej.2017.08.001
[5] J. Narayan, E. Singla, S. Soni, and A. Singla, "Adaptive neuro-fuzzy inference system–based path planning of 5-degrees-of-freedom spatial manipulator for medical applications," Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 232, no. 7, pp. 726–732, June 2018. https://doi.org/10.1177/0954411918781418
[6] A. Aristidou and J. Lasenby, "Fabrik: A fast, iterative solver for the inverse kinematics problem," *Graphical Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011. https://doi.org/10.1016/j.gmod.2011.05.003
[7] S. Koceski and G. Vladimirov, "Inverse kinematics solution of a robot arm based on adaptive neuro fuzzy interface system," *International Journal of Computer Applications*, vol. 178, no. 39, pp. 10–14, Aug. 2019. https://doi.org/10.5120/ijca2019919268
[8] R. Gao, "Inverse kinematics solution of robotics based on neural network algorithms," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 12, pp. 6199–6209, Mar. 2020. https://doi.org/10.1007/s12652-020-01815-4
[9] A. A. Hassan, M. El-Habrouk, and S. Deghedie, "Inverse kinematics of redundant manipulators formulated as quadratic programming optimization problem solved using recurrent neural networks: A review," *Robotica*, vol. 38, no. 8, pp. 1495–1512, Aug. 2020. https://doi.org/10.1017/S0263574719001590
[10] S. Habibkhah and R. V. Mayorga, "The computation of the inverse kinematics of a 3 DOF redundant manipulator via an ANN approach and a virtual function," in *ICINCO*, vol. 1, 2020, pp. 471–477. https://doi.org/10.5220/0009834904710477
[11] S. K. Shah, R. Mishra, and L. S. Ray, "Solution and validation of inverse kinematics using deep artificial neural network," *Materials Today: Proceedings*, vol. 26, no. 2, pp. 1250–1254, 2020. https://doi.org/10.1016/j.matpr.2020.02.250
[12] N. Wagaa, H. Kallel, and N. Mellouli, "Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm," Engineering Applications of Artificial Intelligence, vol. 123, Part B, Aug. 2023, Art. no. 106301. https://doi.org/10.1016/j.engappai.2023.106301
[13] R. Bouzid, H. Gritli, and J. Narayan, "Investigating feed-forward back-propagation neural network with different hyperparameters for inverse kinematics of a 2-DoF robotic manipulator: A comparative study," *Chaos Theory and Applications*, vol. 6, no. 2, pp. 90–110, Jun. 2024. https://doi.org/10.51537/chaos.1375866
[14] J. Fang and W. Li, "Four degrees of freedom SCARA robot kinematics modeling and simulation analysis," *International Journal of Computer, Consumer and Control*, vol. 2, no. 4, pp. 20–27, 2013.
[15] M. Uk, F. Sajjad Ali Shah, M. Soyaslan, and O. Eldogan, "Modeling, control, and simulation of a SCARA PRR-type robot manipulator," *Scientia Iranica*, vol. 27, no. 1, pp. 330–340, 2020.
[16] P. Jha and B. Biswal, "A neural network approach for inverse kinematic of a SCARA manipulator," *IAES International Journal of Robotics and Automation*, vol. 3, no. 1, 2014, Art. no. 52. https://doi.org/10.11591/ijra.v3i1.3201
[17] J. Narayan and A. Singla, "ANFIS based kinematic analysis of a 4-DOFs SCARA robot," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, Solan, India, Sep. 2017, pp. 205–211. https://doi.org/10.1109/ISPCC.2017.8269676
[18] J. Demby's, Y. Gao, and G. N. DeSouza, "A study on solving the inverse kinematics of serial robots using artificial neural network and fuzzy neural network," in *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, New Orleans, LA, USA, Jun. 2019, pp. 1–6. https://doi.org/10.1109/FUZZ-IEEE.2019.8858872
[19] E. Jiménez-López, D. S. De La Mora-Pulido, L. A. Reyes-Ávila, R. S. De La Mora-Pulido, J. Melendez-Campos, and A. A. López-Martínez, "Modeling of inverse kinematic of 3-DOF robot, using unit quaternions and artificial neural network," Robotica, vol. 39, no. 7, pp. 1230–1250, Jan. 2021. https://doi.org/10.1017/S0263574720001071
[20] A. Ranganathan, "The Levenberg-Marquardt algorithm," *Tutorial on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.
[21] M. Kayri, "Predictive abilities of Bayesian regularization and Levenberg-Marquardt algorithms in artificial neural networks: a comparative empirical study on social data," *Mathematical and Computational Applications*, vol. 21, no. 2, May 2016, Art. no. 20. https://doi.org/10.3390/mca21020020
[22] J. Narayan and S. K. Dwivedy, "Biomechanical study and prediction of lower extremity joint movements using Bayesian regularization-based backpropagation neural network," *Journal of Computing and Information Science in Engineering*, vol. 22, no. 1, Jul. 2022, Art. no. 014503. https://doi.org/10.1115/1.4051599
[23] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993. https://doi.org/10.1016/S0893-6080(05)80056-5

Rania Bouzid received her License degree in Informatics Science in June 2020, awarded by the Higher Institute of Information and Communication Technologies (ISTIC), University of Carthage, Tunisia. Her academic journey continued with the completion of her Master's degree in Data Science and Smart Services (D3S) in November 2023, also from ISTIC. Currently, and since December 2023, she is pursuing her PhD at the Polytechnic School of Tunisia (EPT), University of Carthage. Her research interests encompass manipulator robots and artificial intelligence, with a particular focus on the kinematic modeling of manipulator robots, rehabilitation exoskeletons, machine learning, deep learning, metaheuristic algorithms, and Artificial Neural Networks.
E-mail: rania.bouzid@istic.ucar.tn
ORCID iD: https://orcid.org/0009-0003-9641-1380

Hassène Gritli received in 2014 his PhD degree and later in 2020 the HDR degree both in Electrical Engineering from the National Engineering School of Tunis (ENIT), Tunisia. From 2014 until 2020, he was an Associate Professor at the Higher Institute of Information and Communication Technologies (ISTIC), Tunisia. Since 2020 and until today, he is a Professor-Lecturer in Automatics and Industrial Informatics at ISTIC. He is presently a tenured research scientist in Robotics and Automatic Control at the Laboratory of Robotics, Informatics and Complex Systems (RISC Lab) at ENIT. His main research interests include: Biped robots; exoskeleton robots; healthcare robotics; chaos theory; control theory; data science; machine and deep learning; and application to real-world models. He organized some Special Sessions in International conferences, and he was a (Lead) Guest Editor of some special issues in international journals. He published more than 150 technical papers in international journals and conferences, and as book chapters. He serves as a regular reviewer for many international journals and conferences in the field of Robotics, Control Theory and Data Science.
E-mail: grhass@yahoo.fr
ORCID iD: https://orcid.org/0000-0002-5643-134X

Jyotindra Narayan is a Postdoctoral Research Associate in healthcare robotics and AI at the Brain and Behaviour Lab, Imperial College London (UK), and the University of Bayreuth (Germany). He earned his PhD in January 2023 from IIT Guwahati, India, focusing on control schemes for pediatric lower-limb exoskeletons. He holds a Master's degree in Engineering from Thapar University, India, specializing in CAD/CAM and robotics. Dr Narayan has published over 60 articles, including over 25 in Q1 and Q2 journals. He is an academic editor for PLOS ONE Journal and guest editor for the International Journal of Advanced Robotic Systems. His research interests include rehabilitation robotics, robust control theory, bioengineering, and applied artificial intelligence.
E-mail: n.jyotindra@gmail.com
ORCID iD: https://orcid.org/0000-0002-2499-6039