Optimizing ANN Hyperparameters with Metaheuristic Algorithms for Inverse Kinematics of a 3-DoF Rehabilitation Exoskeleton Robot

Rania Bouzid[1,2], Hassène Gritli[2,3]*, and Jyotindra Narayan[4,5]

[1] Polytechnic School of Tunisia, University of Carthage,
B.P. 743, 2078 La Marsa, Tunisia,
rania.bouzid@istic.ucar.tn,

[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.

[3] 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.
grhass@yahoo.fr, hassene.gritli@istic.ucar.tn,

[4] Department of Mechanical Engineering, Indian Institute of Technology Patna,
Patna 801106, India.

[5] Department of Computing, Imperial College London, SWS 2AZ, UK.
n.jyotindra@gmail.com, jnarayan@ic.ac.uk

Abstract. This paper presents a novel approach for optimizing the hyperparameters of Artificial Neural Networks (ANNs) using metaheuristic algorithms to solve the inverse kinematics problem for a 3-DoF rehabilitation exoskeleton robot. The exoskeleton aims to assist in rehabilitation by enabling accurate control of upper limb movement, which is critical for patients recovering motor functions. Two distinct datasets are employed: a random step size dataset and a sinusoidal signal dataset, reflecting different movement patterns in rehabilitation scenarios. The focus is on finding optimal values of three critical hyperparameters of the ANN: activation function, hidden layer size, and learning rate. Metaheuristic algorithms, specifically Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), are utilized to determine the optimal configuration of these hyperparameters. The performance of the proposed method is evaluated by comparing the Mean Squared Error (MSE) across both datasets. Our results demonstrate the effectiveness of the proposed approach in enhancing the precision and reliability for the exoskeleton control, with the lowest MSE values achieved being 0.083801 for the GA-ANN algorithm and 0.040729 for the PSO-ANN algorithm on the sinusoidal signal-based dataset.

Keywords: Artificial Neural Network (ANN), Inverse kinematics, Rehabilitation exoskeleton robot, Random dataset, Sinusoidal dataset, Hyperparameters, Metaheuristic algorithms

S. Krichen et al. (eds.), *Proceedings of International Conference on Decision Aid and Artificial Intelligence (ICODAI 2024)*, Atlantis Highlights in Intelligent Systems 12,
https://doi.org/10.2991/978-94-6463-654-3_9

1 Introduction

Inverse Kinematics is a fundamental problem in robotics, pivotal for controlling robotic arms and ensuring accurate movement in various applications [1]. As robots become increasingly integrated into complex environments, the need for precise inverse kinematics solutions has grown, driven by advancements in automation and robotic technology [2]. Efficient inverse kinematics solutions enable robots to perform intricate tasks, such as assembly and manipulation [3]. Despite its importance, traditional methods often struggle with nonlinearities and singularities, making them less adaptable to varying robot configurations and tasks [4–6]. This has led to intelligent methods that can provide more robust and flexible solutions to inverse kinematics problems [7].

Recent research has explored various intelligent techniques to address these challenges [8–16]. For instance, Almusawi et al. [11] employed neural networks to enhance inverse kinematics solutions for six DoF manipulators, showing promising results in accuracy and computation time. The research [13] presented a multilayer feedforward neural network (MLFFNN) designed to solve forward and inverse kinematics for a 3-DoF articulated robot, achieving near-zero MSE using the Levenberg-Marquardt method for training. To address the complexity of analyzing serial manipulators, researchers in [15] employed an adaptive neuro-fuzzy inference system (ANFIS) for both inverse kinematics and forward dynamics. The simulated model of the manipulator is presented within MATLAB, with the ANFIS networks trained on data from the mathematical model and results validated against the model. In the study [17], the authors presented a novel method for solving the inverse kinematics of an upper-limb exoskeleton, ensuring joint-space constraints in closed-chain mechanisms. Using the Harmony exoskeleton, two improved Projected-Gradient algorithms show effective performance with a maximum error of 0.05 degrees.

Other researchers have proposed using metaheuristic algorithms to resolve inverse kinematics difficulties [18–21]. In their work, Rokbani et al. [22] applied PSO for 2-DoF robots, achieving improved performance in dynamic environments. Researchers presented a solution to the inverse kinematics problem for robot manipulators using GA algorithms, comparing conventional and continuous GAs. Simulation results show that continuous GA outperforms conventional GA by providing smoother and faster solutions [23]. Variant metaheuristic algorithms, such as Ant Colony Optimization (ACO) and Artificial Bee Colony (ABC), have proven effective in solving inverse kinematics problems. The paper [24] introduced a 7-DoF upper limb rehabilitation exoskeleton, FREE, designed with human-like mechanical structures for safe and effective motion. A MIMO shoulder motion prediction model and a projected gradient-based inverse kinematics algorithm were developed, achieving accurate joint motion with a maximum error of 3.04×10^{-3} rad during daily activities.

More advanced research employed metaheuristic algorithms combined with neural networks to address these challenges. The research [25] proposed a hybrid method combining neural networks with GA to enhance inverse kinematics solutions for a six-joint Stanford robotic manipulator. In another study [26], PSO was combined with ANNs to optimize inverse kinematics for a 3-R robot, demonstrating enhanced adaptability and efficiency. These studies highlight the potential of integrating intelligent methods to overcome the limitations of traditional approaches in solving inverse kinematics problems [27].

Despite progress, research using metaheuristic algorithms to optimize ANN hyperparameters for inverse kinematics problems is limited [28]. Current methods often face high computational costs, struggle with dynamic environments, and are constrained by specific datasets. Existing methods for solving inverse kinematics problems still face several limitations, such as high computational costs, susceptibility to local minima, and the need for extensive training data for ANNs [29–31]. Our approach in the present work optimizes ANN hyperparameters like hidden layer size and activation function using metaheuristics, tested on random and sinusoidal datasets, aiming to minimize MSE and offer a more adaptable solution for a 3-DoF Exoskeleton manipulator. Our novelty lies in the use of a random step-size dataset and a sinusoidal signal dataset for training the ANNs. The random step-size dataset introduces variability in the input data, allowing the models to generalize better across a broad range of inverse kinematics configurations. This randomness helps the models handle unpredictable, real-world scenarios by learning from diverse data. The sinusoidal signal dataset, on the other hand, mimics smooth, continuous movements typical of human-like motion, enhancing the model's ability to accurately predict joint angles for more natural and repetitive motions.

The main contributions of the present study are listed as follows:

- **PSO-ANN and GA-ANN**: This study employs two metaheuristic algorithms, namely PSO and GA, in conjunction with ANN to solve the inverse kinematics problem for a 3-DoF exoskeleton manipulator. These metaheuristic algorithms are used to determine the optimal hyperparameters for the ANN models.
- **Comparative analysis**: A comparative study is conducted between the results obtained from the PSO-ANN and GA-ANN approaches, evaluating their effectiveness in solving the inverse kinematics problem for the 3-DoF exoskeleton.
- **Simulation results**: Simulation results are presented to demonstrate the performance of the proposed methods, particularly in minimizing the MSE in inverse kinematics solutions.

The remaining parts of this paper are structured as follows: Section 2 covers the problem formulation for inverse kinematics and the ANN model determination. Section 3 details the proposed techniques, datasets, and metaheuristic algorithms. Section 4 presents the results, and Section 5 concludes with findings and some possible future work directions.

2 Problem Statement

2.1 Inverse Kinematics Problem

In the analysis and design of exoskeleton manipulators, Denavit-Hartenberg (DH) parameters are a standard method used to systematically represent the robot's kinematic chain [32], making it easier to determine the position and orientation of the end-effector relative to the base of the manipulator. By defining four parameters —link length, link twist, link offset, and joint angle— DH parameters simplify the process of transforming coordinates from one frame to another within the robot's structure as represented in Table 1.

Table 1. DH parameters for an n-DoF robotic manipulator.

Joint Number i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	α_1
2	θ_2	d_2	a_2	α_2
3	θ_3	d_3	a_3	α_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	θ_n	d_n	a_n	α_n

The transformation matrix for each joint i is given by the following matrix [8,33]:

$$\mathcal{A}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

For an n-DoF exoskeleton robot, the overall transformation from the base frame to the end-effector frame is the product of the individual joint transformations:

$$\mathcal{T} = \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_i \times \ldots \times \mathcal{A}_n \tag{2}$$

However, inverse kinematics presents a set of challenges, particularly when dealing with robots that have multiple DoFs. One of the primary difficulties is the existence of multiple solutions for a given end-effector position. Selecting the most appropriate solution that avoids issues such as joint limits, singularities, and potential collisions is a significant challenge. The problem is further complicated by the nonlinearity of the equations involved, which often necessitates the use of optimization techniques. These techniques, however, can struggle with the complexity and non-convexity of the solution space, making it difficult to find global optima. Moreover, the computational demands of solving inverse kinematics problems can be high, especially in real-time scenarios where solutions must be computed rapidly and reliably.

The forward kinematics of the robot is given from Eqs. (1) and (2), and is defined like so:

$$\mathcal{T}(\mathbf{q}) = \begin{bmatrix} \mathcal{R}(\mathbf{q}) & \mathcal{P}(\mathbf{q}) \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathcal{R}(\mathbf{q})$ is the rotation matrix and $\mathcal{P}(\mathbf{q})$ is the position vector, the inverse kinematics problem is to solve for $\mathbf{q}$ as given a desired $\mathcal{R}_d$ and $\mathcal{P}_d$:

$$\mathcal{T}_d = \begin{bmatrix} \mathcal{R}_d & \mathcal{P}_d \\ 0 & 1 \end{bmatrix} \tag{4}$$

To solve for q in the inverse kinematics, the equation can be expressed as:

$$\mathbf{q} = \mathbf{f}^{-1}\left(\mathcal{T}_\mathrm{d}\right) = \mathbf{f}^{-1}\left(\begin{bmatrix} \mathcal{R}_\mathrm{d} & \mathcal{P}_\mathbf{d} \\ 0 & 1 \end{bmatrix}\right) \tag{5}$$

where $\mathbf{f}^{-1}$ denotes the inverse kinematic function, which computes the joint angles or positions q corresponding to the desired pose $\mathcal{T}_d$.

2.2 On 3-DoF Exoskeleton Robots

In our work, we will consider a 3-DoF exoskeleton robot with joint angles θ_1, θ_2 and θ_3 and link lengths l_1, l_2 and l_3, as depicted in Fig. 1.

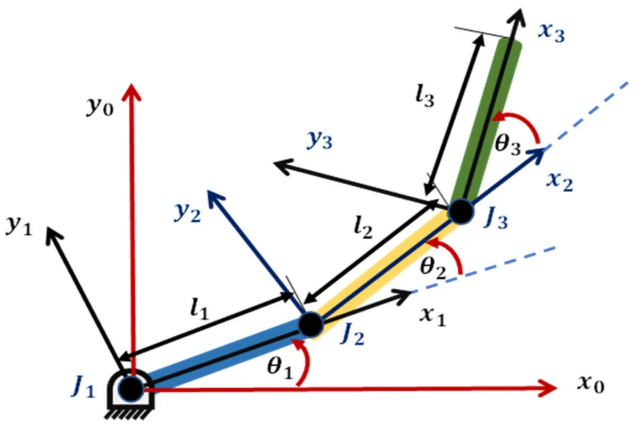


Fig. 1. Three-link planar exoskeleton robotic arm [34].

The FK equations describe the position of the end-effector (x, y) in terms of the joint angles θ_1, θ_2, and θ_3. These equations are derived based on the lengths of the manipulator links a_1, a_2, and a_3. Let's consider the links $a_1 = l_1$, $a_2 = l_2$, and $a_3 = l_3$, the FK can be expressed as follows [35]:

$$x = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) + l_3\cos(\theta_1+\theta_2+\theta_3) \tag{6}$$

$$y = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1+\theta_2+\theta_3) \tag{7}$$

where (x, y) represents the position of the end-effector, and θ_1, θ_2, and θ_3 are the joint angles.

2.3 Determination of the ANN Architecture

The objective function of our approach is to address inverse kinematics challenges by optimizing the hyperparameters of an ANN using a metaheuristic algorithm. Specifically, we focus on optimizing parameters such as hidden layer size, activation function, and minimizing the MSE.

Given:
- $\mathbf{h}$: Vector of hyperparameters
- $\text{ANN}_{\mathbf{h}}$: ANN model configured with hyperparameters $\mathbf{h}$
- $\mathbf{X}_{\text{train}}$: Training input data
- $\mathbf{Y}_{\text{train}}$: Training target data
- $\mathbf{X}_{\text{val}}$: Validation input data
- $\mathbf{Y}_{\text{val}}$: Validation target data
- $\hat{\mathbf{Y}}_{\text{val}}$: Predicted output matrix for validation data
- $f(\mathbf{h})$: Objective function to be minimized (MSE)

Objective Function

The objective function to minimize is the MSE on the validation set:

$$f(\mathbf{h}) = \frac{1}{N} \sum_{i=1}^{N} \left(\text{ANN}_{\mathbf{h}}(\mathbf{X}_{\text{val}})_i - \mathbf{Y}_{\text{val},i} \right)^2 \tag{8}$$

Metaheuristic Algorithm Process

Initialization: Initialize a population of candidate solutions (hyperparameters), $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_P$, where P is the population size.

Evaluation: For each candidate solution $\mathbf{h}_j$, evaluate its fitness using the objective function ((8)).

Selection: Select a subset of the best-performing candidate solutions based on their fitness.

3 Proposed Methodology

Combining metaheuristic algorithms like PSO or GA with ANN is an effective approach to optimizing the hyperparameters of the ANN. These metaheuristics are used to explore the hyperparameter space systematically, searching for the combination that minimizes MSE, as shown in Fig. 2.

3.1 Parameter Specifications

The exoskeleton manipulator has three arm segments with lengths of $l_1 = 0.5$ meters, $l_2 = 1.0$ meters, and $l_3 = 1.5$ meters, respectively. For the dataset generation and testing, a total of 10000 samples were employed, providing a

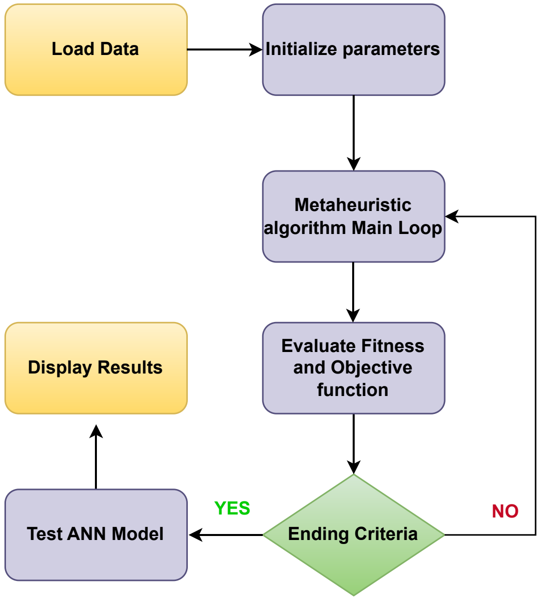


Fig. 2. Flowchart of metaheuristic algorithms with ANNs.

comprehensive set of data points for accurate modeling and evaluation of the manipulator's kinematic behavior. These parameters are essential in defining the robot's workspace and in facilitating the inverse kinematics calculations. In this work, two datasets, namely, random and sinusoidal datasets, are generated based on the FK of the exoskeleton manipulator. For the random dataset, the joint angles θ_i are generated using a uniform distribution across the full range of possible joint angles. Specifically, for each joint i (where $i = 1, 2, 3$), the angle θ_i is computed as:

$$\theta_i = 2\pi \times \text{rand\_values}(:, i) - \pi \tag{9}$$

Here, the function generates random values between 0 and 1 for each joint, and these values are then scaled to span the range $-\pi$ to π, ensuring that all possible configurations of the manipulator are sampled. Fig. 3 provides a visual representation of the angles generated for the random dataset, illustrating the broad distribution of joint angles achieved through this method.

For the sinusoidal signal dataset shown in Fig. 4, the joint angles θ_i are generated using sinusoidal functions with specific amplitudes, frequencies, and phases for each joint. The parameters used for generating the sinusoidal signals are demonstrated in Table 2. The joint angle θ_i for each joint i (where $i = 1, 2, 3$) is computed as:

$$\theta_i = \text{Amp}(i) \cdot \sin\left(2\pi \cdot \text{Freq}(i) \cdot t + \text{Phase}(i)\right) \tag{10}$$

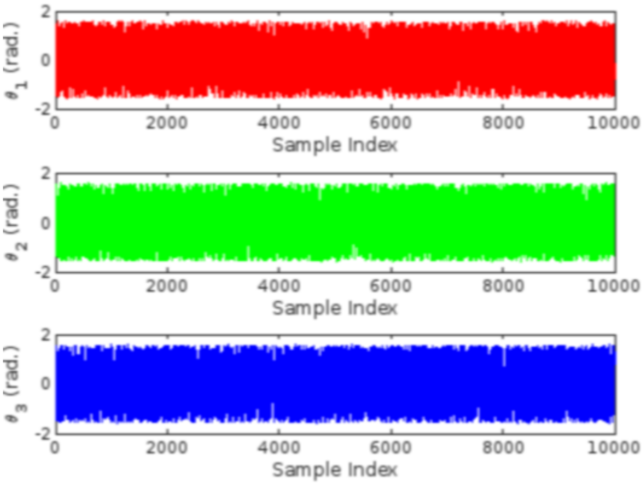


Fig. 3. Representation of angles for the Random dataset.

where Amp(i) is the amplitude for the i-th joint, Freq(i) is the frequency for the i-th joint, Phase(i) is the phase for the i-th joint, t is the time vector.

PSO, inspired by the social behavior of birds [36], and GA, modeled on the process of natural selection [37], both iteratively adjust the hyperparameters, such as activation function, number of neurons to reduce the MSE. The optimization process is conducted with 100 particles, iterating over 50 generations to

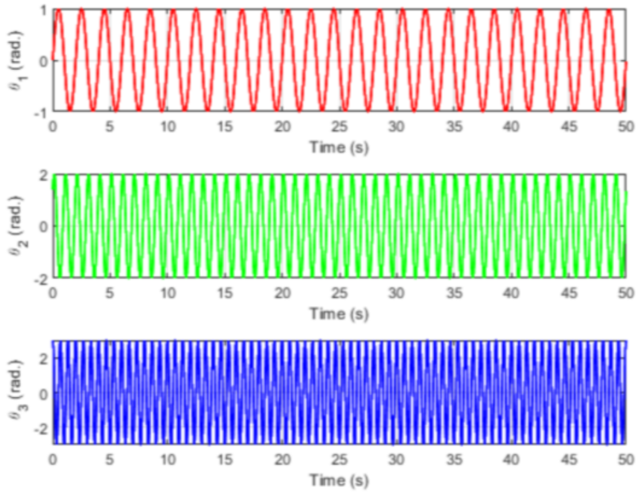


Fig. 4. Presentation of the generated sinusoidal signals for the three articular variables of the robot.

explore the solution space effectively. The inertia weight (w) is set to 0.5, balancing exploration and exploitation during the search process. Both the cognitive weight (c_1) and the social weight (c_2) are set to 2.0, emphasizing the influence of personal best and global best positions in guiding the particles towards optimal solutions.

The GA algorithm is initialized with a population size of 100 and evolves across 50 generations to identify optimal solutions. A crossover fraction of 0.8 is used, meaning that 80% of the individuals in the population participate in crossover during each generation. Additionally, a mutation rate of 0.02 is employed to introduce genetic diversity. In the GA implementation, chromosomes represent the ANN's hyperparameters, specifically the hidden layer size and the activation function. The crossover mechanism used is a one-point crossover, where a random point is selected to exchange genetic material between two parent chromosomes, producing offspring. Discussing this single-point crossover strategy in our methodology would improve our understanding of its effects on genetic diversity and convergence speed.

After testing various parameters, we found that the best results were achieved using those selected arbitrarily. Notably, these parameters included:

- **Hidden Layer Size**: Selected within the range of 1 to 100, allowing for sufficient complexity to capture the underlying patterns in the data.
- **Activation Function**: Chosen among options like `logsig`, `tansig`, and `purelin`, providing flexibility to adapt to different characteristics of the data.

It is worth mentioning that The input layer of the ANN consists of the end-effector positions (x, y), serving as the target data for the inverse kinematics problem. The output layer provides the predicted joint angles $(\theta_1, \theta_2, \theta_3)$ corresponding to the input end-effector positions.

4 Results and Discussions

4.1 Results of Training PSO-ANN

The results demonstrate the use of PSO to optimize hyperparameters for an ANN applied to two distinct datasets: Random step-size and sinusoidal signal (see Table 3).

The following key parameters and outcomes are compared:

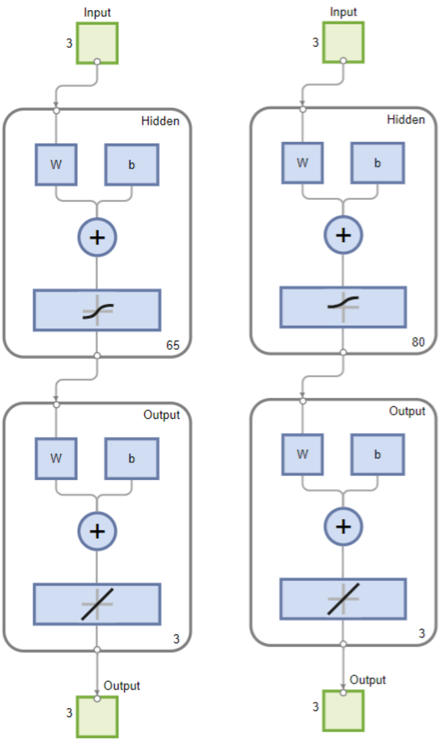
Table 2. Sinusoidal signal's parameters for generating the dataset.

Parameter	Values	Unit
Amplitudes	[1, 2, 3]	degree (°)
Frequencies	[0.5, 1.0, 1.5]	Hz
Phases	$[0, \frac{\pi}{4}, \frac{\pi}{2}]$	radian (rad)

Table 3. PSO for ANN Hyperparameter Selection for the Random and Sinusoidal Datasets.

Parameter	Random Dataset	Sinusoidal Dataset
Epoch	81	458
Hidden Layer Size	65	80
Activation Function	`tansig`	`tansig`
Best Fitness (MSE)	0.17856	0.040729

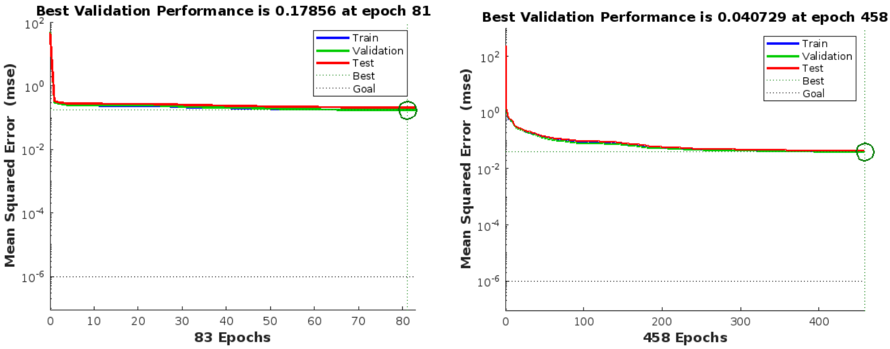
- **Epochs:** The number of training iterations needed to achieve optimal performance. For the random dataset, 81 epochs were sufficient, whereas the sinusoidal dataset required 458 epochs.
- **Hidden Layer Size:** The optimal number of neurons in the hidden layer. The random dataset performed best with 65 neurons, while the sinusoidal dataset required a slightly larger hidden layer with 80 neurons (see Fig. 5).



(a) For the random dataset (b) For the sinusoidal dataset

Fig. 5. ANNs architecture for PSO algorithm.

– **Activation Function:** The `tansig` (hyperbolic tangent sigmoid) function was used in the hidden layers of both datasets to introduce non-linearity.
– **Best Fitness (MSE):** The minimum MSE achieved by the ANN, indicating the model's accuracy as shown in Fig. 6. The sinusoidal dataset achieved a lower MSE of 0.040729, reflecting better performance compared to the random dataset, which had an MSE of 0.17856.



(a) MSE for the random dataset (b) MSE for the sinusoidal dataset

Fig. 6. Best ANN's performances with PSO algorithm.

The PSO model shows varied performance across datasets (see Fig. 7 and Fig. 8). For a random dataset, the model exhibits moderately strong correlations, with R-values of 0.88514 for training, 0.8838 for validation, and 0.86336 for test data, resulting in an overall R-value of 0.88166. In contrast, when applied to a dataset with a sinusoidal signal, the PSO model performs exceptionally well, achieving near-perfect R-values of 0.99104 for training, 0.99105 for validation, and 0.99027 for test data, with an overall R-value of 0.99093, indicating high predictive accuracy and robustness.

4.2 Results of Training GA-ANNs

Following the training process, the results underscore the effectiveness of GA in optimizing the hyperparameters of ANN (see Table. 4).

The key parameters and outcomes are summarized as follows:

– **Epochs:** The number of training iterations needed to achieve optimal performance. For the random dataset, 117 epochs were sufficient, while the sinusoidal dataset required 274 epochs.
– **Hidden Layer Size:** The optimal number of neurons in the hidden layer. The random dataset performed best with 92 neurons, while the sinusoidal dataset required a hidden layer with marginally fewer neurons, i.e., 90 (see Fig. 9).

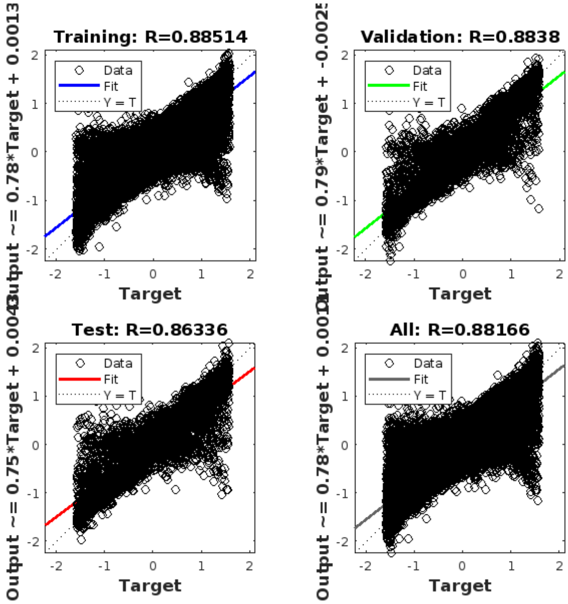


Fig. 7. Regression analysis for PSO algorithm using the proposed random dataset.

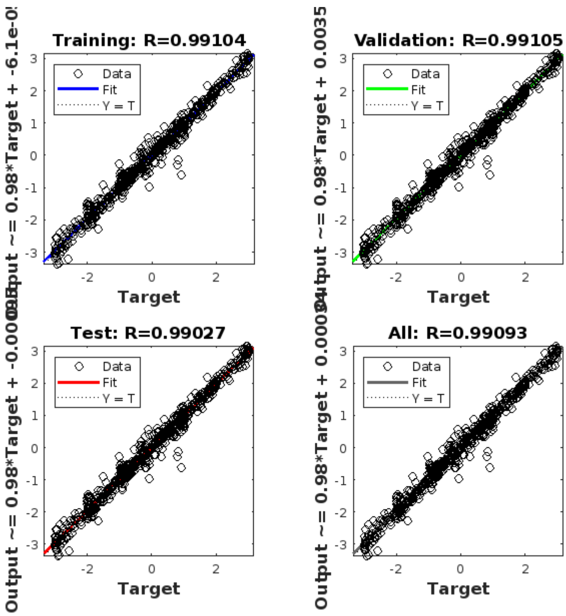
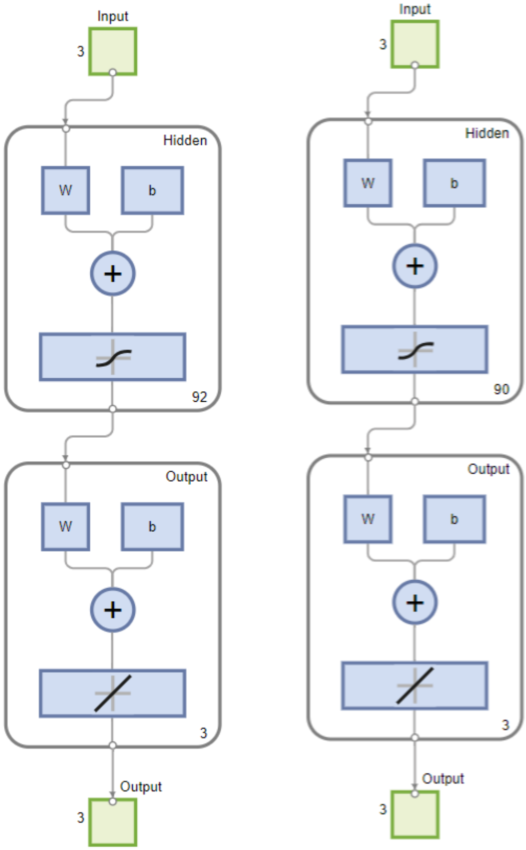


Fig. 8. Regression analysis for PSO algorithm using the sinusoidal dataset.

Table 4. GA for ANN Hyperparameter Selection for Random and Sinusoidal Datasets

Parameter	**Rand**	**Sin**
Epoch	117	274
Hidden Layer Size	92	90
Activation Function	`tansig`	`tansig`
Best Fitness (MSE)	0.18779	0.083801

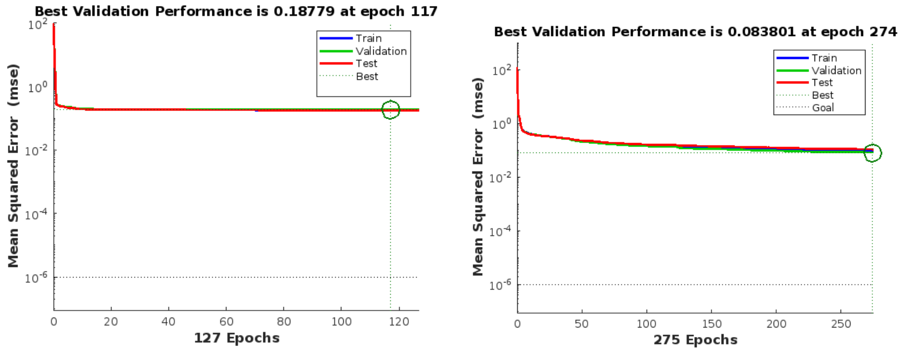


(a) For the random dataset (b) For the sinusoidal dataset

Fig. 9. ANNs architecture for GA algorithm by considering the two datasets.

– **Activation Function:** The `tansig` (hyperbolic tangent sigmoid) function was used in the hidden layers of both datasets to introduce non-linearity.
– **Best Fitness (MSE):** The minimum MSE achieved by the ANN, indicating the model's accuracy. The sinusoidal dataset achieved a lower MSE of

0.083801, reflecting better performance compared to the random dataset, which had an MSE of 0.18779 as presented in Fig. 10.



(a) MSE for random dataset (b) MSE for sinusoidal dataset

Fig. 10. Best ANN's performances for GA algorithm for the two datasets.

After training, we obtain the performance of a GA in regression tasks across different datasets: training, validation, test, and all combined, as displayed in the provided Fig. 11and Fig. 12. The first figure shows a moderate correlation with R values around 0.88, indicating that the model has a decent but not perfect fit when applied to random data, possibly due to the complexity or noise in the dataset. In contrast, the second figure reflects a regression task on sinusoidal data, where the model achieves much higher R values close to 0.98 across the training, validation, test, and combined datasets. This indicates that the GA model effectively captures the underlying patterns in the sinusoidal data, which is more regular and predictable.

5 Conclusion and Future Works

This paper demonstrates a novel hybrid method combining PSO-ANN and GA-ANN to improve the accuracy of inverse kinematics solutions, specifically for a 3-DoF robotic exoskeleton that can be employed for the rehabilitation purpose of disabled persons who need physical reeducation. By using PSO/GA to optimize the ANN hyperparameters, the method addressed limitations in existing inverse kinematics approaches. The method was tested on two datasets—random step size and sinusoidal signal—with results showing a significant reduction in MSE. The PSO-ANN achieved the lowest MSE of 0.040729 on the sinusoidal dataset. The study highlights the method's potential for complex robotic systems, with future work planned to extend the research to manipulators and exoskeletons with higher degrees of freedom and explore more recent metaheuristic algorithms to further improve inverse kinematics solutions.

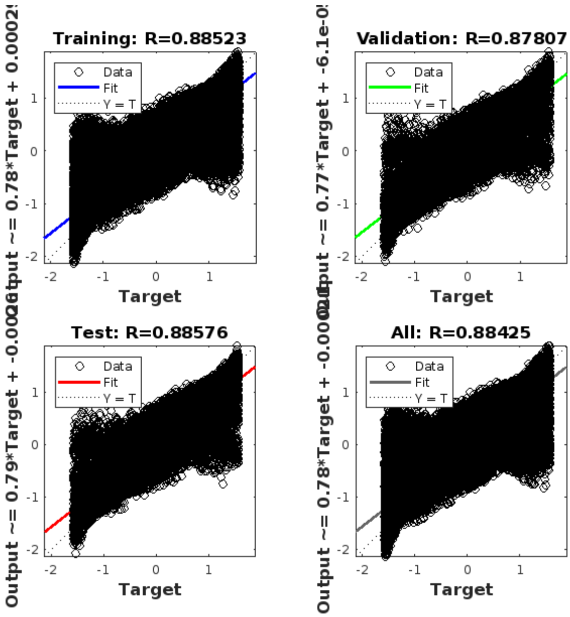


Fig. 11. Regression analysis for GA algorithm using the proposed random dataset.

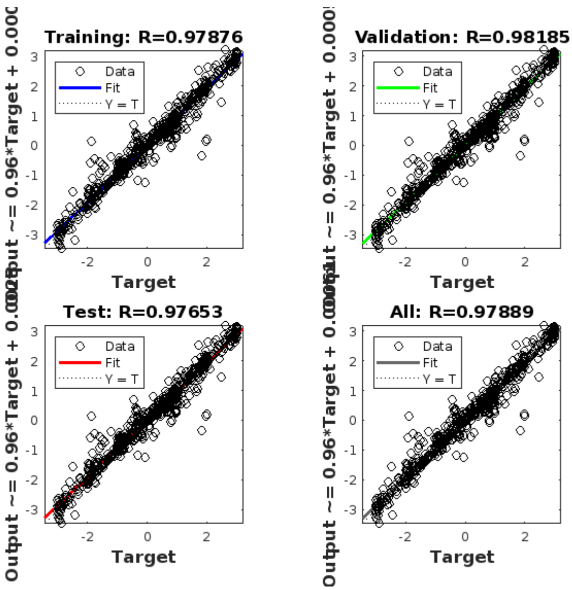


Fig. 12. Regression analysis for GA algorithm using the adopted sinusoidal dataset.

References

1. Kucuk, S., Bingul, Z.: Robot kinematics: Forward and inverse kinematics. INTECH Open Access Publisher London, UK (2006)

2. Liu, Y., Yang, J., Zheng, Y., Wu, Z., Yao, M.: Multi-robot coordination in complex environment with task and communication constraints. International Journal of Advanced Robotic Systems **10**(5), 229 (2013)
3. Alatartsev, S., Stellmacher, S., Ortmeier, F.: Robotic task sequencing problem: A survey. Journal of intelligent & robotic systems **80**, 279–298 (2015)
4. Yahya, S., Moghavvemi, M., Yang, S., Mohamed, H.A.: Motion planning of hyper redundant manipulators based on a new geometrical method. In: 2009 IEEE International Conference on Industrial Technology, pp. 1–5. IEEE (2009)
5. Selig, J.M.: Geometrical methods in robotics. Springer Science & Business Media (2013)
6. Sheng, L., Yiqing, W., Qingwei, C., Weili, H.: A new geometrical method for the inverse kinematics of the hyper-redundant manipulators. In: 2006 IEEE International Conference on Robotics and Biomimetics, pp. 1356–1359. IEEE (2006)
7. Gao, R.: Inverse kinematics solution of robotics based on neural network algorithms. Journal of Ambient Intelligence and Humanized Computing **11**(12), 6199–6209 (2020)
8. Bouzid, R., Gritli, H., Narayan, J.: 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 **6**(2), 90–110 (2024)
9. Bouzid, R., Narayan, J., Gritli, H.: Investigating neural network hyperparameter variations in robotic arm inverse kinematics for different arm lengths. In: 2024 Third International Conference on Power, Control and Computing Technologies (ICPC2T), pp. 351–356 (2024)
10. Bouzid, R., Gritli, H., Narayan, J.: ANN approach for SCARA robot inverse kinematics solutions with diverse datasets and optimisers. Applied Computer Systems **29**(1), 24 – 34 (2024)
11. Almusawi, A.R., Dülger, L.C., Kapucu, S.: A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242). Computational intelligence and neuroscience **2016**(1), 5720,163 (2016)
12. Bouzid, R., Narayan, J., Gritli, H.: Solving inverse kinematics problem for manipulator robots using artificial neural network with varied dataset formats. In: E. Campos-Cantón, G. Huerta-Cuellar, E. Zambrano-Serrano, E. Tlelo-Cuautle (eds.) Complex Systems and Their Applications, pp. 55–78. Springer Nature Switzerland, Cham (2024)
13. Sharkawy, A.N., Khairullah, S.S.: Forward and inverse kinematics solution of a 3-dof articulated robotic manipulator using artificial neural network. International Journal of Robotics & Control Systems **3**(2) (2023)
14. Bouzid, R., Narayan, J., Gritli, H.: Exploring artificial neural networks for the forward kinematics of a SCARA robotic manipulator using varied datasets and training optimizers. Engineering Research Express **6**(4), 045,209 (2024)
15. Deshmukh, D., Pratihar, D.K., Deb, A.K., Ray, H., Ghosh, A.: Anfis-based inverse kinematics and forward dynamics of 3 dof serial manipulator. In: Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020, pp. 144–156. Springer (2021)
16. Bouzid, R., Narayan, J., Gritli, H.: Feedforward backpropagation artificial neural network for modeling the forward kinematics of a robotic manipulator. In: 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), pp. 302–307. Sakheer, Bahrain (2023)

17. Dalla Gasperina, S., Ghonasgi, K., de Oliveira, A.C., Gandolla, M., Pedrocchi, A., Deshpande, A.: A novel inverse kinematics method for upper-limb exoskeleton under joint coordination constraints. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3404–3409. IEEE (2020)
18. Rokbani, N., Neji, B., Slim, M., Mirjalili, S., Ghandour, R.: A multi-objective modified pso for inverse kinematics of a 5-dof robotic arm. Applied Sciences **12**(14) (2022)
19. Abdor-Sierra, J.A., Merchán-Cruz, E.A., Rodríguez-Cañizo, R.G.: A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators. Results in Engineering **16**, 100,597 (2022)
20. Rokbani, N., Mirjalili, S., Slim, M., Alimi, A.M.: A beta salp swarm algorithm meta-heuristic for inverse kinematics and optimization. Applied Intelligence **52**(9), 10,493–10,518 (2022)
21. Rokbani, N., Mirjalili, S., Slim, M., Alimi, A.M.: A beta salp swarm algorithm meta-heuristic for inverse kinematics and optimization. Applied Intelligence **52**(9), 10,493–10,518 (2022)
22. Rokbani, N., Alimi, A.M.: Inverse kinematics using particle swarm optimization, a statistical analysis. Procedia Engineering **64**, 1602–1611 (2013)
23. Momani, S., Abo-Hammour, Z.S., Alsmadi, O.M.: Solution of inverse kinematics problem using genetic algorithms. Applied Mathematics & Information Sciences **10**(1), 225 (2016)
24. Pei, S., Wang, J., Guo, J., Yin, H., Yao, Y.: A human-like inverse kinematics algorithm of an upper limb rehabilitation exoskeleton. Symmetry **15**(9), 1657 (2023)
25. KöKer, R.: A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. Information Sciences **222**, 528–543 (2013)
26. Shastri, S., Parvez, Y., Chauhan, N.R.: Inverse kinematics for a 3-r robot using artificial neural network and modified particle swarm optimization. Journal of The Institution of Engineers (India): Series C **101**(2), 355–363 (2020)
27. Zafar, M.H., Moosavi, S.K.R., Sanfilippo, F.: Inverse kinematic modelling of a 3-dof robotic manipulator using hybrid deep learning models. Procedia CIRP **120**, 213–218 (2023)
28. Abd Elaziz, M., Dahou, A., Abualigah, L., Yu, L., Alshinwan, M., Khasawneh, A.M., Lu, S.: Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. Neural Computing and Applications pp. 1–21 (2021)
29. Chong, H.Y., Yap, H.J., Tan, S.C., Yap, K.S., Wong, S.Y.: Advances of metaheuristic algorithms in training neural networks for industrial applications. Soft Computing **25**(16), 11,209–11,233 (2021)
30. Bouzid, R., Gritli, H., Narayan, J.: Optimized inverse kinematics of a 2-DoF robotic manipulator using a hybrid approach combining an ANN with a metaheuristic algorithm. In: 2024 IEEE International Conference on Artificial Intelligence & Green Energy (ICAIGE24) (2024)
31. Bouzid, R., Narayan, J., Gritli, H.: Hybrid metaheuristic and artificial neural network approach for solving inverse kinematics of a SCARA manipulator robot. In: 2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). University of Bahrain, Bahrain (2024)
32. Hayat, A.A., Chittawadigi, R.G., Udai, A.D., Saha, S.K.: Identification of denavit-hartenberg parameters of an industrial robot. In: Proceedings of conference on advances in robotics, pp. 1–6 (2013)

33. Ganapathy, S.: Decomposition of transformation matrices for robot vision. Pattern Recognition Letters **2**(6), 401–412 (1984)
34. Azeez, M.I., Abdelhaleem, A., Elnaggar, S., Moustafa, K.A., Atia, K.R.: Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced artificial bee colony algorithm. Scientific reports **13**(1), 11,164 (2023)
35. Alavandar, S., Nigam, M.J.: Inverse kinematics solution of 3dof planar robot using anfis. Int. J. of Computers, Communications & Control **3**, 150–155 (2008)
36. Jain, M., Saihjpal, V., Singh, N., Singh, S.B.: An overview of variants and advancements of pso algorithm. Applied Sciences **12**(17), 8392 (2022)
37. Phuoc, L.M., Martinet, P., Lee, S., Kim, H.: Damped least square based genetic algorithm with ggaussian distribution of damping factor for singularity-robust inverse kinematics. Journal of mechanical science and technology **22**, 1330–1338 (2008)