machines

MDPI

Article

Application of Different Supervised Learning Algorithms to the Kinematic Synthesis of Four-Bar Linkage Mechanisms

Onur Denizhan [1], Berrin Denizhan [2,*] and Meng-Sang Chew [3]

[1] Department of Mechanical Engineering, Batman University, 72060 Batman, Turkey; onur.denizhan@batman.edu.tr
[2] Department of Industrial Engineering, Sakarya University, 54187 Serdivan, Turkey
[3] Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, USA; mc0p@lehigh.edu
* Correspondence: denizhan@sakarya.edu.tr; Tel.: +90-264-295-56-78

Abstract

The kinematic synthesis of four-bar linkages has traditionally relied on graphical and analytical precision-point methods, which can become cumbersome or prone to branch and circuit defects. This study examines artificial neural networks (ANNs) trained with Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) for two-position synthesis, under controlled conditions and with kinematic verification of every prediction. Identical networks were trained repeatedly with paired data divisions and initializations. In a configuration-level study, BR reached $R = 1.0000$ and outperformed SCG in all 30 runs ($p = 1.9 \times 10^{-9}$); linkages reconstructed from the BR predictions deviated from the exact motion by less than 3 μm, versus up to 0.72 m for SCG, although SCG still achieved $R > 0.96$. In a task-level study, networks mapping the prescribed task to the link dimensions were trained on defect-free solutions and tested on 200 new synthesis problems: all BR-derived mechanisms were defect-free and reached the prescribed position within 0.61 mm on average, versus 16 mm for SCG. BR accuracy saturated at about 50 configurations and, with 0.5–2 mm input noise, decreased to the noise level. Correlation coefficients alone therefore do not indicate synthesis quality, and kinematic verification is essential.

Keywords: supervised learning; kinematic synthesis; two-position synthesis; Bayesian regularization; kinematic verification; defect-free synthesis

1. Introduction

The four-bar linkage is one of the most fundamental and extensively used mechanisms in machine design, forming the basis of countless applications in vehicle components, rehabilitation robotics, and rotary/reciprocating machinery [1]. Classical treatments of mechanism synthesis, dating back to foundational texts on the kinematic synthesis of linkages [2] and continuing through modern kinematics and machine-design references [3], establish graphical and analytical precision-point methods as the traditional approach for determining the link dimensions required to satisfy prescribed motion-, path-, or function-generation tasks. However, finding practical and useful mechanisms for the motion synthesis problem has proven elusive, as a large majority of synthesized mechanisms turn out to be defective with respect to their assembly modes [4]. To address the limitations of classical closed-form synthesis, optimization-based approaches—including genetic algorithms [5] and hybrid GA–differential-evolution methods for path synthesis [6]—have been proposed to search the design space for defect-free, high-precision four-bar solutions. This body

Check for updates

Academic Editor: Dan Zhang

Received: 5 October 2026
Revised: 17 November 2026
Accepted: 20 November 2026
Published: 25 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

of classical and optimization-based literature has, in turn, motivated growing interest in automotive and mechanical design applications of four-bar linkages beyond textbook synthesis, including engine-hood lift mechanisms and spring-assisted static balancing systems, where dimensional synthesis is coupled with force- and energy-based design criteria [7–11].

In parallel, artificial neural networks (ANNs) have emerged as a powerful data-driven alternative for addressing nonlinear inverse kinematic problems without requiring explicit closed-form derivations. This approach has a long history in the broader field of mechanism and manipulator kinematics: early work established that neural networks could be trained to solve the inverse kinematics problem for robotic manipulators without explicit analytical inversion [12], and subsequent studies extended ANN-based inverse kinematics to parallel mechanisms, showing that both multilayer perceptron and radial basis function networks can achieve satisfactory accuracy and speed for complex parallel kinematic chains [13]. A recent comprehensive review has further consolidated the state of the art in machine-learning-based kinematic analysis and synthesis across serial and parallel mechanisms, highlighting single, modular, and hybrid neural network architectures and identifying future integration challenges [14]. Large-scale, data-driven approaches have also emerged, such as the construction of datasets comprising on the order of a hundred million planar linkage mechanisms to support data-driven kinematic design [15].

Within this broader context, several studies have applied ANN-based approaches specifically to the analysis and synthesis of four-bar and slider-crank mechanisms. The Levenberg–Marquardt backpropagation algorithm—a second-order method that combines the speed of the Gauss–Newton method with the stability of steepest descent [16]—has been applied to the two-position kinematic synthesis problem of the four-bar planar linkage mechanism [1], while related work has extended ANN-based kinematic analysis to slider-crank and quick-return mechanisms, including position, velocity, and acceleration analysis using Levenberg–Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient algorithms [17,18]. Notably, a systematic comparison of supervised learning algorithms for the position analysis of the slider-crank (R-RRT) mechanism found that the Bayesian Regularization algorithm yields the best accuracy, while the Levenberg–Marquardt backpropagation algorithm exhibits the best training-speed performance [17], underscoring that algorithm choice materially affects both accuracy and efficiency even within the same mechanism family. Beyond gradient-based feedforward approaches, machine learning techniques combining pattern recognition and autoencoder neural networks have been used to build compact databases of defect-free invariant motions for four-bar motion generation [4], and more recently, deep generative frameworks employing conditional generative adversarial networks have been proposed to synthesize four-bar mechanisms satisfying specified kinematic and quasi-static conditions [19], as well as direct-parameterization architectures for diverse four-bar function-generation mechanisms [20]. Further data-driven approaches to four-bar path synthesis include case-based neural networks trained on kinematic simulations [21], ANNs combined with Fourier descriptors of the coupler curve [22], image-based variational path synthesis [23], variational-autoencoder representations of coupler curves [24], conditional β-variational autoencoders that generate several mechanism types for one coupler curve [25], and large datasets of 4-, 6- and 8-bar mechanisms for machine-learning-driven path synthesis [26].

Despite this growing body of work, the choice of training algorithm used to optimize an ANN remains a critical factor governing prediction accuracy, convergence speed, and generalization capability. Bayesian Regularization (BR) and the Scaled Conjugate Gradient (SCG) algorithm represent two conceptually distinct approaches. Bayesian Regularization, originally formulated by MacKay [27] and later adapted for efficient implementation within

the Levenberg–Marquardt framework by Foresee and Hagan [28], treats network training within a Bayesian framework that expands the cost function to search not only for minimal error but for minimal error using minimal weights, allowing estimation of the effective number of parameters actually required by the model. This regularization mechanism makes Bayesian-regularized networks highly resilient against over-training and overfitting, and it reduces or eliminates the need for a separate validation set or extensive trial-and-error selection of the number of hidden neurons, as demonstrated in web-spam classification applications [29]. In contrast, the Scaled Conjugate Gradient algorithm, introduced by Møller [30], is fully automated, includes no critical user-dependent parameters, and avoids the time-consuming line search required by other conjugate-gradient and quasi-Newton methods, relying instead on a validation-based early-stopping criterion to control overfitting.

Bayesian-regularized and SCG-based ANNs have each demonstrated strong performance across diverse engineering regression problems, including fluid-flow process optimization [31] and austenite-formation-temperature prediction in steels [32], with such comparative studies generally finding that both algorithms produce acceptable predictions, though their relative accuracy and generalization performance vary depending on the application and validation strategy employed. Foundational treatments of neural network design, universal approximation, and pattern-recognition theory underlying these algorithms are well established [33–36].

Although successful applications of both BR and SCG have been reported across various regression and mechanism-analysis contexts—including recent modeling and experimental studies of four-bar mechanisms with prescribed coupler-point velocity characteristics [37]—a direct, systematic comparison of these two algorithms specifically for the kinematic synthesis of the four-bar linkage mechanism has not been reported in the literature. This study addresses this gap by developing and comparing ANN models trained with Bayesian Regularization and Scaled Conjugate Gradient algorithms for the four-bar linkage kinematic synthesis problem. The study has two parts. In the first part (Sections 4.1–4.3), for a given analytical two-position synthesis solution, the ANN learns the configuration-level relation between the position of joint C and the link angles (θ, φ, ψ) of the synthesized mechanism along its motion between the two prescribed positions. In the second part (Section 4.4), the ANN maps the synthesis task itself (the prescribed displacement and rotation together with the free choices) to the link vectors of the dyads, and is tested on new synthesis problems that were not used for training. Network performance is evaluated in terms of regression correlation, error distribution and training convergence behavior, complemented by repeated training from random initializations and by mechanism-level measures (coupler-point deviation, joint position deviation, coupler-length consistency, loop closure and branch consistency) computed after reconstructing the linkage from the predicted angles.

Artificial neural networks have already been applied to four-bar synthesis [1,4,19,20], and BR and SCG are standard training algorithms; the present work does not propose a new network architecture. Its contributions are the following. (i) A controlled comparison of BR and SCG for four-bar synthesis, with identical architectures, paired data divisions and initial weights, repeated training (30 and 10 runs) and a statistical test, which to the authors' knowledge has not been reported for linkage synthesis. (ii) A kinematic verification procedure in which every network prediction is converted back into a linkage and simulated, and is judged by the error in achieving the prescribed motion, branch consistency, assembly feasibility and transmission angle, rather than by regression metrics only. (iii) A task-level formulation of two-position synthesis in which defective solutions are excluded from the training data by a branch, toggle and transmission-angle filter, so that the learned mapping is single-valued and defect-free, tested on 200 synthesis tasks not used for training.

(iv) Quantitative evidence that regression metrics hide large differences at the mechanism level: networks with $R = 1.000000$ and $R = 0.9998$ differ by a factor of about 27 in the error of the synthesized motion.

These contributions distinguish the present study from previous ANN-based synthesis studies in three concrete respects, which are summarized in Table 10 (Section 4.7). First, most recent studies [19,20,23–26] concentrate on network architectures, curve representations or datasets for path and function synthesis, whereas the present study isolates the effect of the training algorithm under otherwise identical conditions. Second, the present study evaluates every prediction at the mechanism level by kinematic simulation, rather than by the regression error of the predicted parameters alone. Third, because two-position synthesis has a closed-form solution, the exact answer is known for every sample, so the accuracy of the networks can be measured without the ambiguity of approximate synthesis. The validity limits of the results are stated explicitly in Section 4.8.

2. Two-Position Synthesis of a Four-Bar Linkage

Two-position synthesis is a kinematic method for designing a mechanism to guide a rigid body or link through two specified positions. In this approach, the initial and final positions of the linkage are prescribed, and geometric construction is used to determine the permissible locations of the fixed and moving pivots that enable the target motion.

Figure 1 shows the detailed schematic of the four-bar linkage mechanism featuring right- and left-side dyads. The links are assumed to be completely rigid, and joints A, B, C, and D are modeled as ideal revolute joints. Based on the geometric relationships illustrated in Figure 1, the kinematic synthesis equations for the two prescribed positions are formulated as follows:

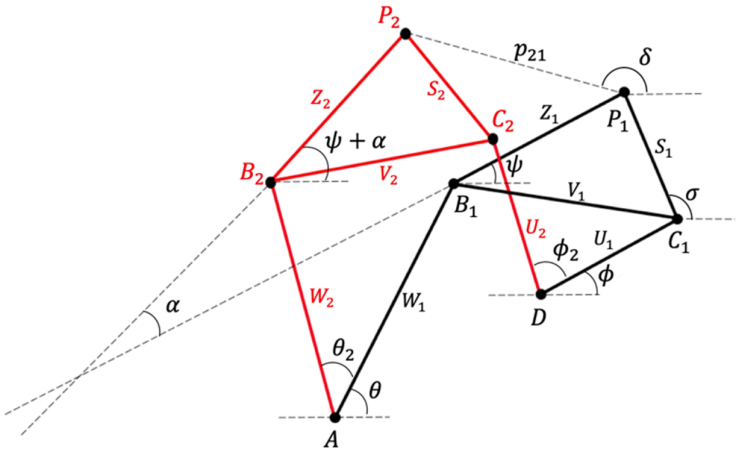


Figure 1. Right- and left-side dyads of the four-bar linkage in the two prescribed positions [1].

Left-side dyad:

$$we^{i\theta} = \frac{ze^{i(\psi+\alpha)} - p_{21}e^{i\delta} - ze^{i\psi}}{1 - e^{i\theta_2}} \tag{1}$$

Right-side dyad:

$$ue^{i\varphi} = \frac{se^{i(\sigma+\alpha)} - p_{21}e^{i\delta} - se^{i\sigma}}{1 - e^{i\varphi_2}} \tag{2}$$

where w is the length of the rigid link $[AB]$ ($\left|\vec{W}_1\right| = \left|\vec{W}_2\right|$), z is the length of the vectors $\vec{Z}_1$ and $\vec{Z}_2$ ($\left|\vec{Z}_1\right| = \left|\vec{Z}_2\right|$), u is the length of the rigidlink $[DC]$ ($\left|\vec{U}_1\right| = \left|\vec{U}_2\right|$), s is the length of the vectors $\vec{S}_1$ and $\vec{S}_2$ ($\left|\vec{S}_1\right| = \left|\vec{S}_2\right|$), θ is the angle of link $[AB]$ at the first position $(\vec{W}_1)$, θ_2 is the rotation of link $[AB]$ between the first and last positions ($\vec{W}_1$ and $\vec{W}_2$), φ is the angle of

link $[DC]$ at the first position ($\vec{U}_1$), φ_2 is the rotation of link $[DC]$ between the first and last positions ($\vec{U}_1$ and $\vec{U}_2$), ψ is the angle of link $[BP]$ at the first position ($\vec{Z}_1$), α is the rotation of link $[BP]$ between the first and last positions ($\vec{Z}_1$ and $\vec{Z}_2$), δ is the direction angle of the displacement of the coupler point P, σ is the angle of link $[CP]$ at the first position ($\vec{S}_1$) and p_{21} is the distance between the first and last positions of the coupler point P.

Equations (1) and (2) govern the two-position kinematic synthesis of the planar four-bar linkage. Following the analytical solution procedure outlined in traditional kinematic texts, three design parameters are initially prescribed: $p_{21} = 2.416$ m, $\alpha = 43.3°$ and $\delta = 165.2°$ while the remaining six parameters are selected as free choices, with the values $s = 1.035$ m, $z = 1.298$ m, $\sigma = 104.1°$, $\psi = 26.5°$, $\theta_2 = 38.4°$ and $\varphi_2 = 85.6°$ [1]. All angles in this paragraph are given in degrees and were converted to radians (multiplication by $\pi/180$) before the complex-number computations of Equations (1) and (2). All other angles reported in this paper, including $\theta = 1.2492$ rad and $\varphi = 0.2691$ rad below and the network outputs, are given in radians.

Using the above parameters in conjunction with Equations (1) and (2), the problem of two-position synthesis for the four-bar planar linkage mechanism can then be solved.

Coordinate system and sign conventions. A fixed Cartesian frame is placed at the first position of the coupler point, $P_1 = (0, 0)$, with the x-axis horizontal (real axis) and the y-axis vertical (imaginary axis). Every vector is written in complex polar form $re^{i\gamma}$, where γ is measured counter-clockwise from the positive x-axis. The left dyad consists of the vector W (from the fixed pivot A to the moving pivot B) and the coupler vector Z (from B to P); the right dyad consists of U (from the fixed pivot D to the moving pivot C) and S (from C to P). The displacement vector $P_{21} = p_{21}e^{i\delta}$ points from P_1 to P_2; α is the rotation of the coupler between the two positions, and θ_2 and φ_2 are the rotations of links AB and DC, respectively. Equations (1) and (2) are the standard loop-closure (dyad) equations $W(e^{i\theta_2} - 1) + Z(e^{i\alpha} - 1) = P_{21}$ and $U(e^{i\varphi_2} - 1) + S(e^{i\alpha} - 1) = P_{21}$ [2,3], solved for W and U. The first prescribed position is used as the reference configuration.

Resulting mechanism. With the prescribed values, Equations (1) and (2) give $w = 2.4669$ m, $\theta = 1.2492$ rad (71.57°), $u = 1.4856$ m and $\varphi = 0.2691$ rad (15.42°). The fixed pivots are therefore located at $A = (-1.9415, -2.9196)$ m and $D = (-1.1800, -1.3987)$ m, and the moving pivots in the reference configuration at $B_1 = (-1.1616, -0.5792)$ m and $C_1 = (0.2521, -1.0038)$ m. The linkage has a ground link $|AD| = 1.7008$ m, input link $|AB| = 2.4669$ m, coupler $|BC| = 1.4762$ m and output link $|DC| = 1.4856$ m. Since the sum of the shortest and longest links (3.9431 m) exceeds the sum of the other two (3.1864 m), the linkage is of the non-Grashof (triple-rocker) type; link AB can rotate from $-96.8°$ to $+80.4°$ relative to its reference position before a toggle configuration is reached, so that the required rotation $\theta_2 = 38.4°$ lies well inside the feasible range.

Assembly mode and branch. For a given angle of link AB, the position of joint C is obtained as the intersection of two circles (centered at B with radius $|BC|$ and centered at D with radius $|DC|$), which gives two assembly modes (open and crossed). The assembly mode is fixed by the reference configuration obtained from Equations (1) and (2) and is maintained by continuity as AB rotates from 0 to θ_2. Along this motion the discriminant of the circle intersection never vanishes (no toggle position is crossed), the transmission angle remains between 32.1° and 74.4°, and the mechanism reaches the second prescribed position exactly (closure error at machine precision). The synthesized mechanism is therefore free of branch (circuit) defects between the two prescribed positions. All data samples used in Section 3 belong to this single assembly branch; no samples from the alternative assembly mode are included, and no defective mechanisms had to be filtered out.

3. Neural Network Application to the Two-Position Synthesis of a Four-Bar Linkage

The developed artificial neural network model for the kinematic synthesis of the four-bar linkage mechanism is constructed as a feedforward, fully connected network with two neurons in the input layer, one hidden layer with 100 neurons, and three neurons in the output layer. The Cartesian coordinates of joint C (x_c, y_c) serve as the network inputs, and the mechanism angles (θ, φ, and ψ) are taken as the outputs. The network was trained and evaluated on 75 samples obtained from the analytical kinematic synthesis solution and randomly divided into 70% training data (53 samples), 15% validation data (11 samples) and 15% test data (11 samples), as is the standard practice for supervised ANN training. The same 2–100–3 architecture, the same transfer functions, the same input/output normalization (mapminmax) and the same data-division ratios were used for both training algorithms; only the training function differs (trainbr or trainscg). In all cases the weights and biases were initialized with the default Nguyen–Widrow method (initnw) of MATLAB 2025b. Note that MATLAB's trainbr disables validation-based stopping and merges the validation samples into the training set; therefore, the BR network is trained on 64 samples (Table 2) and the SCG network on 53 samples (Table 4), while both are tested on 11 samples.

The hidden layer used a sigmoid (tan-sigmoid) transfer function to capture the non-linear relationships intrinsic to the synthesis problem. The output layer used a linear transfer function to enable the network outputs to cover the entire continuous range of the target angle values without saturation. This architecture—a single hidden layer with a sigmoid activation, followed by a linear output layer—is a common architecture for function-approximation problems of this type, since it allows the network to approximate any continuous nonlinear mapping between the input and output spaces, given a sufficient number of hidden neurons, as suggested by the universal approximation theorem.

The structure of the feedforward neural network is shown in Figure 2. The network comprises two inputs (x_c, y_c), a single hidden layer containing 100 neurons, and an output layer of three neurons corresponding to the three outputs (θ, φ, and ψ). In Figure 2, w and b denote the network weights and biases, respectively (the weight w is not to be confused with the link length w). As noted above, a sigmoid activation function is used in the hidden layer and a linear activation function in the output layer.

The four unknown parameters w, u, θ, and $]\varphi$ are obtained by applying the classical analytical procedure for two-position linkage synthesis, which yields $w = 2.4669$ m, $u = 1.4856$ m, $\theta = 1.2492$ rad, and $]\varphi = 0.2691$ rad. With these parameters known, a dataset is generated for the motion of the planar four-bar linkage between the two prescribed positions. To this end, 75 intermediate configurations of the mechanism are evaluated, and the complete set of mechanism parameters is computed for each. As stated above, the coordinates of joint C (x_c, y_c) are taken as the network inputs, while the mechanism angles θ, φ, and ψ serve as the outputs of the supervised learning algorithms.

Data generation. The 75 samples were generated deterministically, not by random sampling: link AB was rotated from its reference angle $\theta = 1.2492$ rad to $\theta + \theta_2$ in 74 equal increments (0.519° each), so that the 75 configurations include both prescribed positions. For each configuration the positions of B, C and P were computed on the assembly branch defined in Section 2, and the coordinates (x_c, y_c) of joint C together with the angles θ (link AB), φ (link DC) and ψ (vector BP) were recorded. The inputs lie in the ranges $x_c \in [-1.464, 0.252]$ m and $y_c \in [-1.004, 0.087]$ m, and the outputs in $\theta \in [1.249, 1.919]$ rad, $\varphi \in [0.269, 1.763]$ rad and $\psi \in [0.463, 1.218]$ rad. Consequently, the 75 samples are not independent mechanisms; they are 75 configurations of one and the same synthesized mechanism. Only the division of these samples into training, validation and test subsets (MATLAB dividerand) is random. The number of 75 configurations was chosen as a

compromise between a dense coverage of the motion and a small dataset; its influence on the results is examined in Section 4.5, which shows that the accuracy of the BR network no longer improves beyond about 50 configurations.

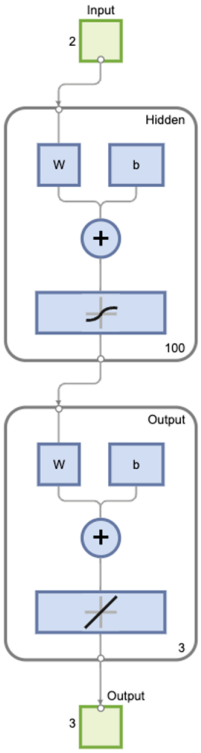


Figure 2. Structure of the artificial neural network.

Well-posedness of the mapping. Because joint C is connected to the fixed pivot D by the rigid link DC, every sample (x_c, y_c) lies on a circular arc of radius u about D, and $\varphi = \text{atan2}(y_c - y_D, x_c - x_D)$ is uniquely defined by the input. On the selected assembly branch, $]\varphi$ increases monotonically from 0.2691 to 1.7631 rad as AB rotates through θ_2, and θ and ψ are single-valued, monotonic functions of φ. Hence, within the sampled range and on the fixed branch, the mapping $(x_c, y_c) \rightarrow (\theta, \varphi, \psi)$ is a well-defined, single-valued and injective function. The non-uniqueness that is generally present in four-bar synthesis (alternative assembly modes, or different link dimensions producing similar motions) is excluded here by construction, because the link dimensions are fixed by the analytical solution of Equations (1) and (2) and only one branch is sampled. This also defines the scope of the trained networks: they are surrogates of the configuration-level kinematics of one synthesized mechanism, not general inverse maps from design requirements to mechanism dimensions. Such a task-level map is studied separately in Section 4.4.

Model capacity. The 2–100–3 network has 603 adjustable weights and biases, which exceeds the number of training samples. For BR this is mitigated by the Bayesian weight penalty, which reduces the effective number of parameters to 55.9 (Table 1). Nevertheless, because the samples lie on a one-dimensional smooth curve, the very low errors reported below should be interpreted as accurate interpolation within the sampled range of motion, and not as evidence of generalization to other mechanisms or design tasks. To verify that the conclusions are not an artefact of the network size, the BR network was also trained with 5, 10, 20 and 50 hidden neurons (10 runs each); the mean test MSE was 3.5×10^{-12}, 5.3×10^{-14}, 9.1×10^{-15} and 3.6×10^{-14} rad^2, respectively, compared with 5.7×10^{-14} rad^2 for 100 neurons (same seeds). Already 10 hidden neurons reach the accuracy of the 100-neuron network, which confirms that the results are not an artefact of the network size.

Table 1. Bayesian-regularization algorithm training process.

Parameter	Initial Value	Stopped Value	Target Value
Epoch	0	76	1000
Elapsed time (s)	-	2	-
Performance (MSE, rad^2)	21.4	1.63×10^{-13}	0
Gradient	20.7	1.12×10^{-8}	1×10^{-7}
μ (Marquardt parameter)	0.005	5×10^{4}	1×10^{10}
Effective number of parameters	603	55.9	0
Sum of squared weights	2.64×10^{4}	179	0

We shall begin this approach using the Bayesian Regularization algorithm.

3.1. Bayesian Regularization Algorithm

The training algorithm, Bayesian Regularization (BR), is based on the Levenberg–Marquardt framework and improves network generalization by minimizing a regularized objective function:

$$F = \beta E_D + \alpha E_W \tag{3}$$

where E_D is the sum-squared error, E_W is the sum of squared network weights, and α and β are hyperparameters (unrelated to the coupler rotation α of Section 2) that control the relative importance of error minimization versus weight minimization [27,28]. In this Bayesian framework, the weights and hyperparameters are treated as random variables, and their optimal values are determined by maximizing the posterior probability of the weights using Bayes' theorem. Foresee and Hagan [28] developed an efficient Gauss–Newton approximation that includes this step directly in the Levenberg–Marquardt algorithm without the computational cost of a full Bayesian treatment.

During training, this optimization automatically determines the effective number of parameters used by the network and pushes redundant weights toward zero, thus reducing the effective complexity of the network. The Bayesian Regularization training includes an inherent weight penalty and is therefore resistant to overfitting; it usually does not require a separate validation set, unlike early-stopping-based training strategies. In this study, the algorithm was implemented using the *trainbr* function of the MATLAB Neural Network Toolbox.

3.2. Scaled Conjugate Gradient Algorithm

The Scaled Conjugate Gradient (SCG) algorithm was introduced by Møller [30]. It is a second-order training algorithm that belongs to the conjugate gradient family. Instead of updating the network weights along the negative gradient, as is done by the standard backpropagation algorithm, it updates the weights along a series of mutually conjugate search directions [30].

Because the SCG algorithm does not contain an intrinsic weight-penalty term, network generalization is controlled using a validation-based early-stopping criterion [30]. Under this strategy, training is halted when a specified number of consecutive validation error checks fail to improve, thereby preventing overfitting [30]. For the implementation of the algorithm in this study, the *trainscg* function of the MATLAB Neural Network Toolbox was utilized. To ensure a fair comparison, the SCG network has exactly the same 2–100–3 architecture, transfer functions, normalization, initialization method and data-division ratios as the BR network.

4. Results and Discussion

This section presents the results of the BR and SCG networks (Sections 4.1 and 4.2), the evaluation on independent data with repeated training and mechanism-level measures (Section 4.3), the task-level synthesis of new problems with kinematic verification (Section 4.4), the effect of the dataset size and of measurement noise (Section 4.5), and a discussion of the training algorithms, a comparison with previous studies and the validity limits of the results (Sections 4.6–4.8).

4.1. Results of the Bayesian Regularization Network

Table 1 lists the training parameters of the Bayesian Regularization algorithm. Training converged in about 2 s after 76 epochs (out of a maximum of 1000). The performance (MSE) decreased quickly from 21.4 to 1.63×10^{-13} and the gradient decreased from 20.7 to 1.12×10^{-8}, approaching the target threshold of 1×10^{-7}. The effective number of parameters decreased from 603 to 55.9, demonstrating the algorithm's ability to automatically prune unnecessary network complexity. The sum of squared weights decreased from 2.64×10^{4} to 179. These results confirm that the Bayesian Regularization algorithm succeeded in converging to a highly optimized low-complexity network configuration that is consistent with its underlying principle of penalizing large weight magnitudes to avoid overfitting.

Table 2 shows the sizes, MSE values and correlation coefficients obtained for each subset with the BR-trained network. The training set (64 observations) obtained an MSE of 1.6304×10^{-13} with $R = 1$. Importantly, no separate validation set was used (0 observations), following the standard implementation of the Bayesian Regularization algorithm, which does not need a validation-based early-stopping criterion. The test set (11 observations) and the complete dataset of 75 observations (training + test) gave MSE values of 2.1631×10^{-13} and 1.7085×10^{-13}, respectively, both with $R = 1$. The "All data" row is the performance of the final network on all 75 samples; its MSE is the sample-weighted average of the training and test values, so these samples are not independent of the training data and do not constitute an independent test. The results indicate an extremely high interpolation accuracy within the sampled range of motion. A genuinely independent test set is evaluated in Section 4.3.

Table 2. Bayesian-regularization algorithm training results.

	Observations	**MSE (Rad²)**	***R***
Training (incl. validation samples)	64	1.6304×10^{-13}	1.000
Validation	0	–	–
Test	11	2.1631×10^{-13}	1.000
All data (training + test)	75	1.7085×10^{-13}	1.000

Figure 3 shows the regression plots of the Bayesian-regularized ANN for the four-bar linkage kinematic synthesis problem, with network outputs versus target values for the training, testing, and combined data sets. In all cases the correlation coefficient reached $R = 1$, with the fitted line coinciding essentially with the ideal Y = T line (intercepts on the order of 10^{-7}–10^{-9}). This near-unity correlation is consistent for both the training and test subsets, which indicates accurate interpolation within the sampled range of motion without signs of overfitting. This confirms the effectiveness of the Bayesian Regularization algorithm in optimizing the trade-off between data-fitting accuracy and model complexity for this kinematic synthesis application.

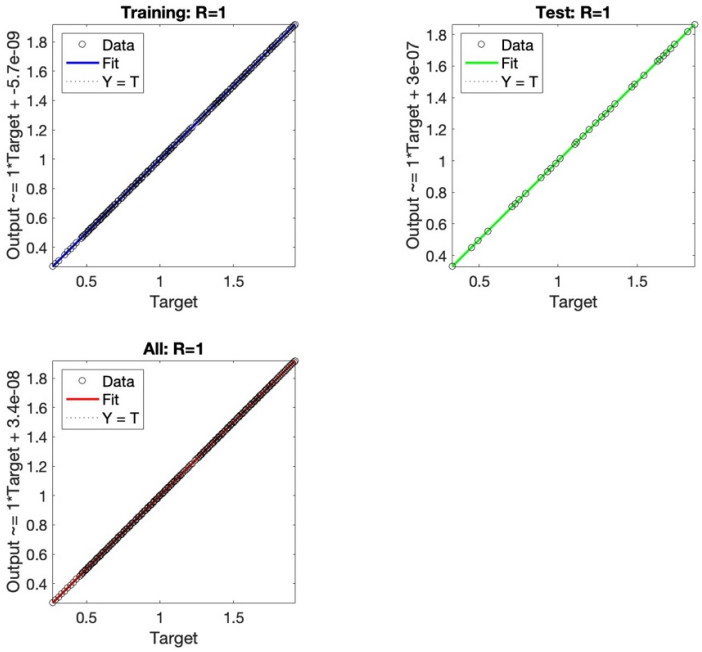


Figure 3. Bayesian-regularization algorithm regression plots.

Figure 4 shows the error histogram (20 bins) of the trained Bayesian-regularized ANN. The error is the difference between the target and predicted output values. The errors are closely clustered around zero and range approximately from -1.1×10^{-6} to 1.1×10^{-6} rad, with most of the samples (in both the training and test sets) lying between -4×10^{-7} and 4×10^{-7} rad, near the zero-error line. The narrow, nearly symmetric error distribution, together with the well-balanced distribution of test-set errors with respect to the training set (without a clear bias to one side of zero), further confirms the high interpolation accuracy of the network for the four-bar linkage kinematic synthesis problem.

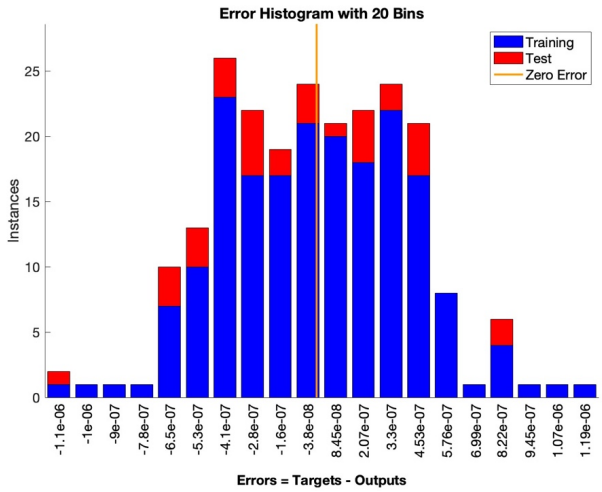


Figure 4. Bayesian-regularization algorithm: error histogram.

Figure 5 shows the mean squared error (MSE) of the Bayesian-regularized ANN through the training process. The best network was at epoch 76 with a performance of 1.6304×10^{-13}. Both training and test MSE curves show a steep decline in the first 10 or so epochs, and then a smooth convergence to a plateau at an imperceptibly small error level. The high similarity between the training curve and test curve in the entire process of training, with no divergence in the later epochs, shows that overfitting does not occur

and the Bayesian-regularization framework is effective in obtaining both high accuracy and good interpolation capability for the four-bar linkage kinematic synthesis model.

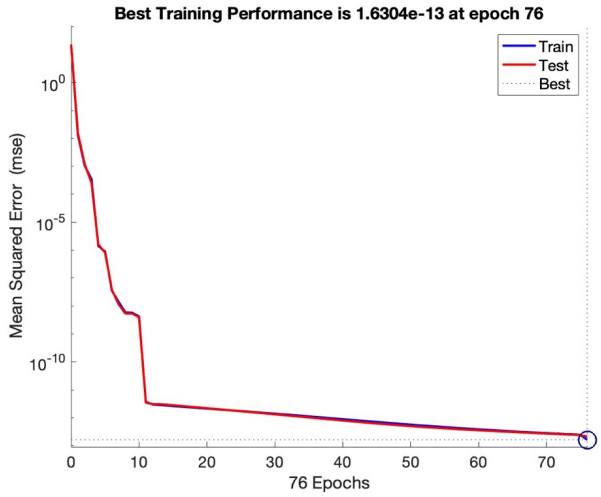


Figure 5. Bayesian-regularization algorithm training performance plot (no validation set is used by trainbr).

Figure 6 shows the regression plot of the final BR network evaluated on all 75 samples of the dataset (training and test samples together). The correlation coefficient is $R = 1$, with the fitted regression line (Output $\approx 1 \cdot$Target $+ 3.4 \times 10^{-8}$) practically coincident with the ideal Y = T line over the entire target range (approximately 0.27–1.92 rad). Because 64 of these 75 samples were used for training, this figure summarizes the overall fitting accuracy and is not an independent test. The performance of the networks on configurations that were not used for training is reported in Section 4.3.

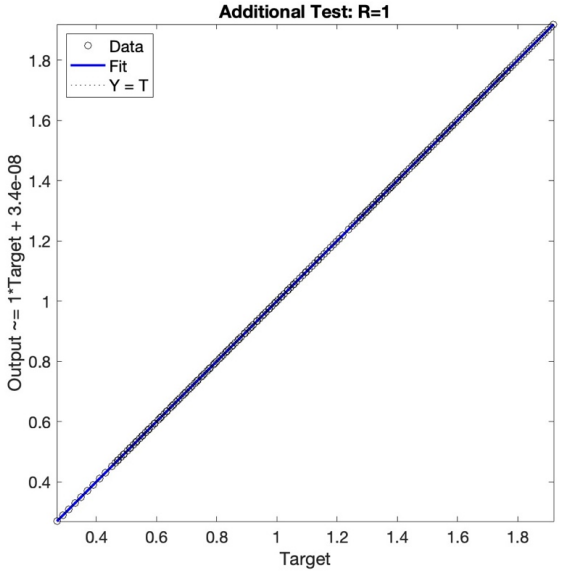


Figure 6. Bayesian-regularization algorithm regression plot for all 75 samples (training + test).

The error histogram (20 bins) for all 75 samples is shown in Figure 7. The errors are clustered around zero, ranging from about -1.1×10^{-6} to 1.19×10^{-6} rad, and most of the points are within $\pm 4 \times 10^{-7}$ rad of the zero-error line. This narrow, near-symmetric distribution is consistent with the training and test results in Figure 4.

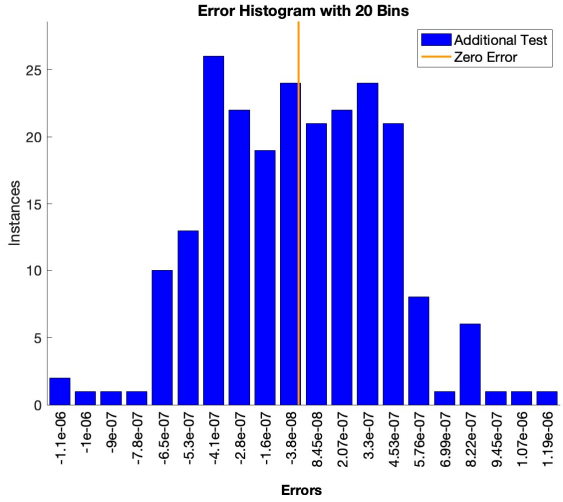


Figure 7. Bayesian-regularization algorithm error histogram for all 75 samples (training + test).

4.2. Results of the Scaled Conjugate Gradient Network

Training process parameters of the SCG algorithm are summarized in Table 3. Training was stopped after 66 epochs (out of a maximum of 1000), triggered by six consecutive validation checks without further improvement in validation performance, as determined by the early-stopping criterion inherent to the SCG algorithm. The performance (MSE) decreased from 3.63 to 0.00115, and the gradient decreased from 5.06 to 0.025, which is still higher than the target threshold of 1×10^{-6}, so the training was stopped by the validation-check criterion instead of meeting the gradient convergence target.

Table 3. Scaled Conjugate Gradient algorithm training process.

Parameter	Initial Value	Stopped Value	Target Value
Epoch	0	66	1000
Elapsed time (s)	-	<1	-
Performance (MSE, rad^2)	3.63	1.15×10^{-3}	0
Gradient	5.06	0.025	1×10^{-6}
Validation checks	0	6	6

The dataset sizes, MSE and correlation coefficients obtained for each subset of the SCG-trained network are shown in Table 4. The training set (53 observations) produced an MSE of 0.0015 and $R = 0.9961$. The validation and test sets (11 observations each) had MSE values of 0.0100 and 0.0110, with $R = 0.9785$ and $R = 0.9657$, respectively. Evaluated on all 75 samples (training, validation and test together), the network gives an MSE of 0.0041 and $R = 0.9894$; as for the BR network, this is the sample-weighted average of the subset values and not an independent test.

Table 4. Scaled Conjugate Gradient algorithm training results.

	Observations	MSE (Rad^2)	*R*
Training	53	0.0015	0.9961
Validation	11	0.0100	0.9785
Test	11	0.0110	0.9657
All data (training + validation + test)	75	0.0041	0.9894

Figure 8 shows the regression plots of the ANN trained by the Scaled Conjugate Gradient (SCG) algorithm for the training, validation, test and combined (all) datasets, with correlation coefficients of $R = 0.996$, 0.978, 0.966 and 0.989, respectively. The high *R*-

values indicate that the network output and the target values are strongly linearly correlated for all subsets. The fitted regression lines are very close to the ideal Y = T line. The relatively uniform correlation for the training, validation and test subsets indicates a reasonable fit, which is, however, clearly less accurate than that of the BR network.

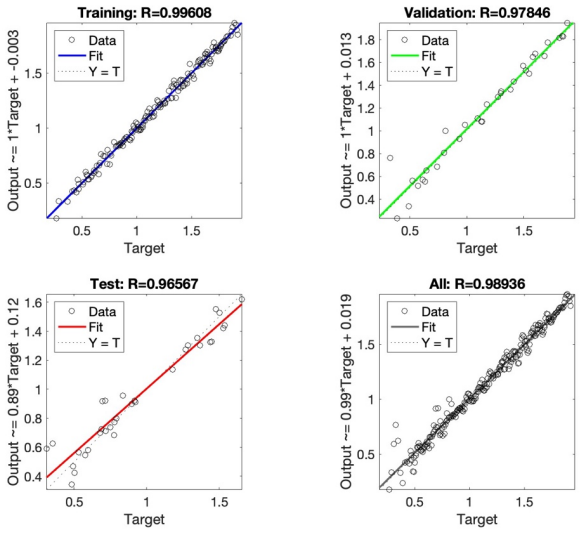


Figure 8. Scaled Conjugate Gradient algorithm regression plots.

Figure 9 is a histogram (20 bins) of errors (difference between target and output values) for the ANN trained using the Scaled Conjugate Gradient algorithm. The distribution is roughly uni-modal and centered around zero, with most samples (training, validation and test) lying within the error band from −0.02 to 0.02 rad. However, the errors are far from negligible, ranging from about −0.42 to 0.14 rad, with a few cases (especially in the training set) having larger deviations on the tails of the distribution. This indicates that the network trained by SCG can provide reasonably good predictions for most of the samples, but there exist some data points with relatively larger prediction errors for the four-bar linkage kinematic synthesis problem.

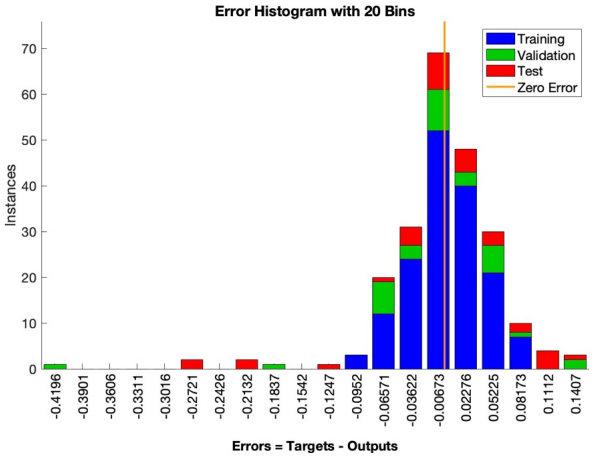


Figure 9. Scaled Conjugate Gradient algorithm: error histogram.

The mean squared error (MSE) of the SCG-trained ANN for the training, validation and test subsets is shown in Figure 10. The best validation performance (0.010004) was achieved at epoch 60, after which training was terminated by early stopping. The training MSE continued to decrease steadily over the 66 epochs, reaching values below 10^{-3}. Meanwhile, the validation and test curves plateaued around the 20–30 epoch mark and remained

relatively flat afterwards, with the validation error deviating slightly from the training error in the later epochs. This difference between the training and the validation/test curves is typical of the early-stopping mechanism embedded in the SCG algorithm that stops the training once no improvement can be observed on the validation performance, preventing overfitting.

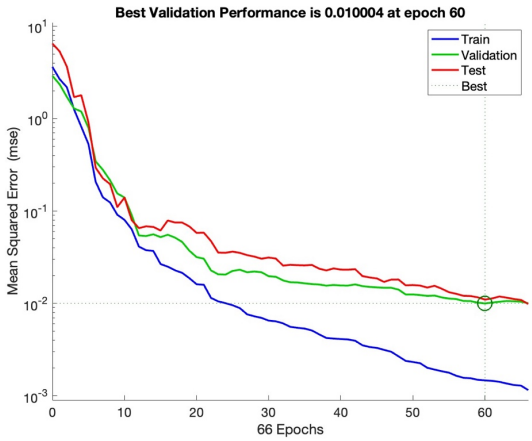


Figure 10. Scaled Conjugate Gradient algorithm: training performance plot.

Figure 11 shows the regression plot of the SCG network for all 75 samples (training, validation and test together; $R = 0.98936$, fitted line Output $\approx 0.99 \cdot$Target $+ 0.019$). This figure corresponds to the "All" panel of Figure 8 and is not an independent test. The visible scatter, particularly at higher target values, reflects a moderate level of prediction error of the SCG network.

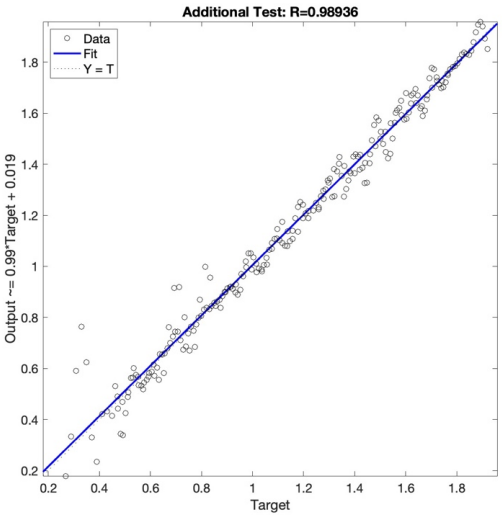


Figure 11. Scaled Conjugate Gradient algorithm regression plot for all 75 samples (training + validation + test).

Figure 12 presents the error histogram (20 bins) of the SCG network for all 75 samples. Most errors lie in a narrow band around zero (roughly −0.031 to 0.011 rad), with a small number of outliers reaching about 0.28 rad. Thus, the SCG network predicts most configurations with acceptable accuracy, but a limited number of configurations show comparatively large deviations, whose effect at the mechanism level is quantified in Section 4.3.

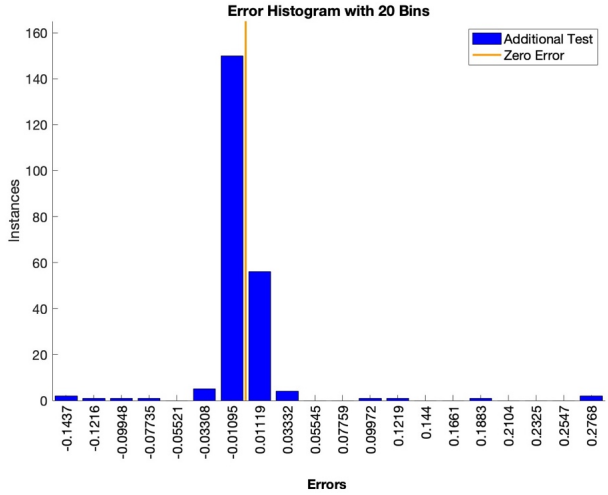


Figure 12. Scaled Conjugate Gradient algorithm error histogram for all 75 samples (training + validation + test).

4.3. Independent Test Set, Repeated Training and Mechanism-Level Evaluation

Independent test set. To evaluate the networks on configurations that were not used in any phase of training, an additional set of 74 configurations was generated with the procedure of Section 3, at the mid-points between consecutive configurations of the 75-sample dataset (link AB rotated by $(k-0.5)\theta_2/74$ from its reference position, $k = 1, \ldots, 74$). None of these configurations is contained in the 75-sample dataset. Because the networks are mechanism-specific surrogates, this test assesses the interpolation accuracy between training configurations; it does not, and is not intended to, demonstrate generalization to other mechanisms or design problems. It should also be noted that the samples of the 75-sample dataset are 0.519° of crank rotation apart, so that every test configuration of Sections 4.1 and 4.2 lies between two training configurations; the test subsets therefore measure interpolation between closely spaced samples, which explains why $R = 1.0000$ is attainable. Generalization to genuinely new synthesis problems is examined in Section 4.4.

Repeated training. ANN results depend on the random initial weights and on the random data division, so a single training run cannot establish the superiority of one algorithm over another. Both networks were therefore trained 30 times. Because this study (100 training runs in total, including the network-size study) was carried out in a fully scripted environment, both algorithms were re-implemented in Python 3.15/NumPy 2.5.5 following exactly the formulations used by MATLAB: the Gauss–Newton approximation to Bayesian regularization within the Levenberg–Marquardt algorithm [28] (initial $\mu = 0.005$, decrease/increase factors 0.1/10, $\mu_{\max} = 10^{10}$, at most 1000 epochs) and Møller's scaled conjugate gradient [30] ($\sigma = 5 \times 10^{-5}$, $\lambda = 5 \times 10^{-7}$, early stopping after 6 consecutive validation failures, best-validation weights retained), with the same 2–100–3 tan-sigmoid/linear architecture, mapminmax normalization and Nguyen–Widrow initialization. For run k ($k = 1, \ldots, 30$), seed k controls the random 70/15/15 data division and seed $1000 + k$ the initial weights, so that BR and SCG start from the same data division and the same initial weights. The results are reported as mean ± standard deviation over the 30 runs, and the paired differences in test MSE are assessed with the Wilcoxon signed-rank test.

Mechanism-level measures. To assess whether the predicted angles correspond to a valid four-bar mechanism, the linkage is reconstructed for every sample from the predicted angles $\hat{\theta}$, $\hat{\varphi}$ and $\hat{\psi}$, using the fixed pivots A and D and the link lengths obtained in Section 2: $\hat{B} = A + we^{i\hat{\theta}}$, $\hat{C} = D + ue^{i\hat{\varphi}}$, $\hat{P}_L = \hat{B} + ze^{i\hat{\psi}}$ and $\hat{P}_R = \hat{C} + se^{i(\sigma+\hat{\psi}-\psi_1)}$, where $\psi_1 = 26.5°$ is the reference value of ψ. The following measures are computed: (i) the coupler-point (trajectory) deviation $e_P = \left|\hat{P}_L - P\right|$; (ii) the joint-C deviation $e_C = \left|\hat{C} - C\right|$;

(iii) the coupler-length (link-dimension) consistency $e_{BC} = \left\|\hat{C} - \hat{B}\right| - \left|BC\right\|$; (iv) the loop-closure error $e_{\text{loop}} = \left|\hat{P}_L - \hat{P}_R\right|$; (v) the branch consistency, i.e., the percentage of samples whose reconstructed configuration lies in the same assembly mode as the reference configuration (same sign of the cross product of the vectors BC and BD); and (vi) the assembly feasibility, i.e., the percentage of samples for which a closed configuration exists for the predicted input angle $\hat{\theta}$ ($\left\|BC\right| - \left|DC\right\| \le \left|\hat{B} - D\right| \le \left|BC\right| + \left|DC\right|$).

A priori error bounds. The order of magnitude of these measures can be bounded directly from the angular errors reported in Sections 4.1 and 4.2. Since $\hat{P}_L - P = w\left(e^{i\hat{\theta}} - e^{i\theta}\right) + z\left(e^{i\hat{\psi}} - e^{i\psi}\right)$ and $\left|e^{ia} - e^{ib}\right| \le |a - b|$, it follows that $e_P \le (w + z) \cdot \max|\Delta| = 3.765\text{m} \times \max|\Delta|$ and $e_C \le u|\Delta\varphi|$, where Δ denotes the angular prediction error. For the BR network ($\max|\Delta| \approx 1.2 \times 10^{-6}$rad , Figures 4 and 7), the coupler-point deviation is therefore below 4.5×10^{-6} m and the joint-C deviation below 1.8×10^{-6} m, which is negligible compared with $p_{21} = 2.416$ m. For the SCG network most errors lie within ± 0.02 rad (Figure 9), corresponding to coupler-point deviations of up to about 0.075 m (3.1% of p_{21}); however, the largest errors (about 0.28–0.42 rad, Figures 9 and 12) may produce deviations of the order of 1 m, which would be unacceptable in a precision design. These bounds show that correlation coefficients alone ($R > 0.96$ for SCG) overstate the practical adequacy of a network, and that angle-level regression metrics must be complemented by mechanism-level measures such as those in Table 5.

Table 5. Results of 30 repeated training runs and mechanism-level measures on the independent test set (mean ± standard deviation over the runs; for e_P, e_C, e_{BC} and e_{loop} the maximum over the samples of each run is used).

Metric	BR (trainbr)	SCG (trainscg)
Test-subset MSE (rad^2)	$(4.09 \pm 7.6) \times 10^{-14}$	$(1.64 \pm 3.4) \times 10^{-3}$
Test-subset R	$1.0000 \pm < 10^{-12}$	0.996 ± 0.006
Independent-set MSE (rad^2)	$(1.34 \pm 2.1) \times 10^{-14}$	$(1.09 \pm 2.4) \times 10^{-3}$
Independent-set R	$1.0000 \pm < 10^{-12}$	0.997 ± 0.006
max e_P (m)	$(8.4 \pm 5.8) \times 10^{-7}$	0.213 ± 0.16
max e_C (m)	$(5.8 \pm 4.7) \times 10^{-7}$	0.128 ± 0.14
max e_{BC} (m)	$(3.9 \pm 3.7) \times 10^{-7}$	0.200 ± 0.14
max e_{loop} (m)	$(4.5 \pm 4.0) \times 10^{-7}$	0.230 ± 0.16
Branch consistency (%)	100	100
Assembly feasibility (%)	100	100
Runs with lower test MSE (of 30)	30	0
Wilcoxon signed-rank p-value	1.9×10^{-9}	–

Table 5 summarizes the results. BR gave a lower test MSE than SCG in all 30 runs (Wilcoxon signed-rank $p = 1.9 \times 10^{-9}$); the superiority of BR therefore does not depend on a particular initialization or data division. On the independent test set, the mean maximum coupler-point deviation was 8.4×10^{-7} m for BR and 0.213 m for SCG (3.5×10^{-5}% and 8.8% of p_{21}, respectively); the worst runs reached 2.5×10^{-6} m and 0.72 m. All BR and all SCG reconstructions remained on the reference assembly branch, and a closed configuration existed for 100% of the predicted input angles of both networks. Hence, in this problem the SCG errors degrade the accuracy of the motion but not the assembly feasibility. These mechanism-level results show that the BR predictions yield linkage configurations that satisfy the closure constraints of the synthesized mechanism to within 2.5×10^{-6} m, whereas the SCG predictions violate them by up to 0.72 m (coupler-point deviation) and 0.62 m (loop closure).

4.4. Task-Level Synthesis and Kinematic Verification on New Synthesis Problems

Formulation. To examine whether the networks can solve synthesis problems that they have not seen, a second, task-level formulation was considered. The network inputs are the prescribed displacement magnitude p_{21} and direction δ, the prescribed coupler rotation α, and the rotations θ_2 and $]\varphi_2$ of links AB and DC (free choices); the outputs are the Cartesian components of the dyad vectors W and U (5 inputs, 4 outputs). The remaining free choices are fixed at $z = 1.298$ m, $\psi = 26.5°$, $s = 1.035$ m and $\sigma = 104.1°$. Since Equations (1) and (2) are linear in W and U, each task has exactly one solution, so this mapping is single-valued. Cartesian components are used instead of lengths and angles to avoid the 2π discontinuity of angles.

Data and defect filter. Synthesis tasks were sampled by Latin hypercube sampling in the ranges $p_{21} \in [2.0, 2.8]$ m, $\delta \in [150°, 180°]$, $\alpha \in [35°, 52°]$, $\theta_2 \in [30°, 48°]$ and $\varphi_2 \in [70°, 100°]$, which contain the task of Section 2. For every sampled task, the exact solution of Equations (1) and (2) was simulated by rotating link AB from 0 to θ_2 on the assembly mode of the first position. A task was rejected as defective when a toggle position was met, when the mechanism did not reach the second prescribed position on the same branch (branch defect), or when the transmission angle fell below 20°; about 17% of the sampled tasks were rejected. The training data therefore contain only defect-free solutions. The dataset consists of 300 defect-free tasks, randomly divided 70/15/15 as in Section 3, and an independent set of 200 further defect-free tasks was drawn with a different random seed.

Independence of the new tasks. No task of the independent set occurs in the dataset. In the normalized input space, the median distance from a new task to its nearest training task is 0.228 (minimum 0.063), which is equal to the median nearest-neighbor distance between the training tasks themselves (0.224). The new tasks are therefore not near-duplicates of training tasks, in contrast with the closely spaced configurations of Sections 4.1–4.3.

Network and kinematic verification. A 5–20–4 network was used for both algorithms, with all other settings as in Section 4.3, and each algorithm was trained 10 times with paired seeds. Every predicted solution $(\hat{W}, \hat{U})$ was verified kinematically: the linkage with fixed pivots $\hat{A} = B_1 - \hat{W}$ and $\hat{D} = C_1 - \hat{U}$ was assembled in the first position, and link AB was rotated through the prescribed angle θ_2 in 100 steps, solving the position problem at each step on the assembly mode of the first position. Because the moving pivots in the first position are fixed by the free choices, the first position is reached exactly and all errors appear along the motion. The motion obtained was compared with the prescribed motion through the error in reaching the second prescribed position, $\left|\hat{P}_2 - P_2\right|$, the error in the coupler rotation, $|\hat{\alpha} - \alpha|$, and the maximum deviation from the coupler path of the exact solution. It was also checked whether each predicted mechanism is assemblable over the whole motion, free of branch and toggle defects, and has a transmission angle of at least 20°.

Table 6 shows that BR gave a lower new-task MSE than SCG in all 10 runs (Wilcoxon $p = 0.002$). All mechanisms built from the predictions of both algorithms were assemblable and free of branch and toggle defects, so the defect-free character of the training data was carried over to the new tasks. The mechanisms obtained from the BR predictions reached the prescribed second position within 0.61 mm on average; 84.5% of the new tasks were reached within 1 mm and all within 10 mm, and the worst case over all 2000 verified BR solutions was 6.4 mm (0.3% of p_{21}). The SCG-predicted mechanisms missed the prescribed position by 16.4 mm on average and by up to 138 mm, and only 41% of the tasks were reached within 10 mm. The largest coupler-rotation errors were 0.14° for BR and 2.9° for SCG on average over the runs (0.2° and 4.9° in the worst runs). The maximum deviation from the exact coupler path was equal to the error at the second position in every run, because the deviation grows along the motion. Notably, the SCG network still has $R = 0.9998$ on the

new tasks, which again shows that the correlation coefficient alone does not indicate the quality of the synthesized motion.

Table 6. Task-level synthesis: results on the 200 new synthesis tasks (mean ± standard deviation over 10 training runs; "max" is the maximum over the 200 tasks of each run).

Metric	BR (Trainbr)	SCG (Trainscg)
Test-subset MSE (m^2)	$(2.89 \pm 0.81) \times 10^{-7}$	$(2.50 \pm 1.9) \times 10^{-4}$
New-task MSE (m^2)	$(2.61 \pm 0.49) \times 10^{-7}$	$(2.39 \pm 1.6) \times 10^{-4}$
New-task R	$1.000000 \pm 3 \times 10^{-8}$	$0.99983 \pm 1.1 \times 10^{-4}$
Mean error in reaching P_2 (mm)	0.61 ± 0.06	16.4 ± 7.1
Max error in reaching P_2 (mm)	3.8 ± 1.1	85 ± 36
Mean coupler-rotation error (°)	0.020 ± 0.002	0.62 ± 0.30
Max coupler-rotation error (°)	0.14 ± 0.03	2.9 ± 1.2
Max link-length error (mm)	3.0 ± 0.8	86 ± 37
Tasks reaching P_2 within 1 mm (%)	84.5 ± 3.6	0.9 ± 1.3
Tasks reaching P_2 within 10 mm (%)	100	41 ± 21
Assemblable, branch- and toggle-free (%)	100	100
Transmission angle $\geq 20°$ (%)	100	99.7 ± 0.4
Runs with lower new-task MSE (of 10)	10 (Wilcoxon $p = 0.002$)	0

Table 7 lists eight of the new tasks, spread over the range of p_{21}, together with the kinematic errors of the run-1 networks. Figure 13a shows task T8: the coupler paths of the exact and BR-predicted mechanisms are indistinguishable, whereas the SCG-predicted mechanism misses the prescribed position P_2 by 47 mm. Figure 13b shows the distribution of the error in reaching P_2 over all 200 new tasks.

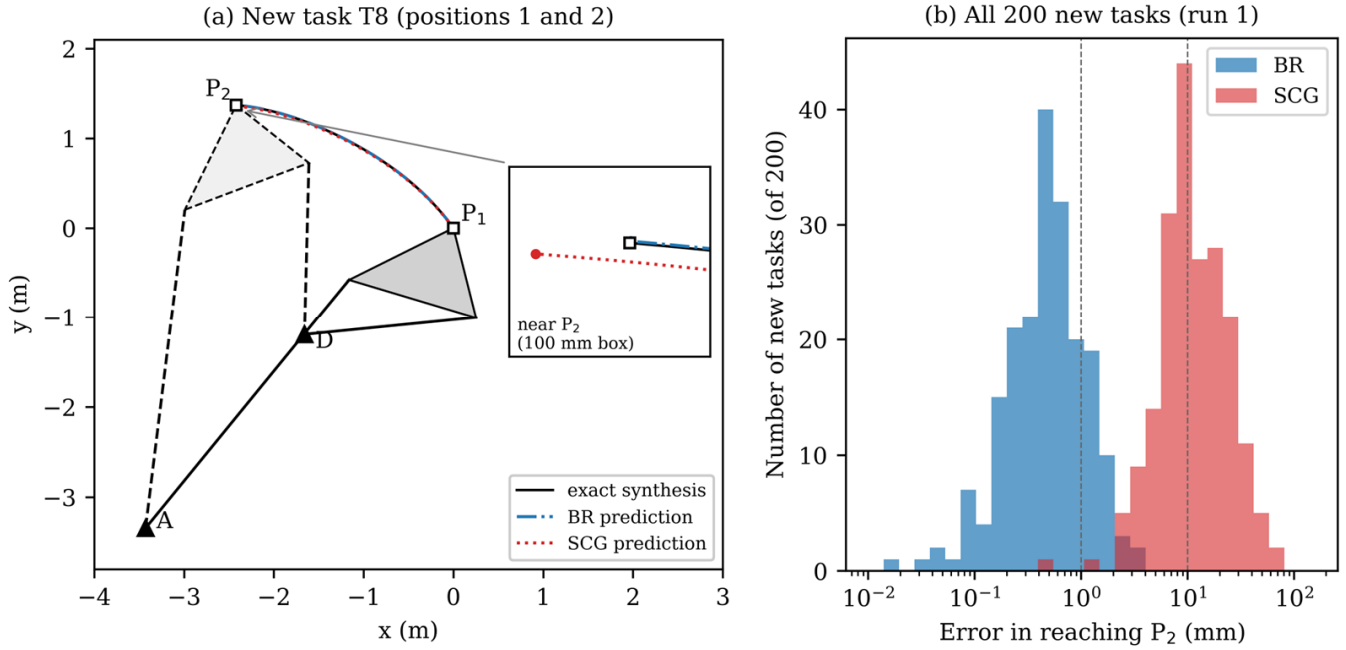


Figure 13. Kinematic verification on new synthesis tasks: (**a**) linkage of new task T8 in the two prescribed positions, with the coupler paths of the exact, BR-predicted and SCG-predicted mechanisms (inset: region around P_2); (**b**) distribution of the error in reaching the prescribed position P_2 over the 200 new tasks (run 1).

Table 7. Examples of new synthesis tasks not used for training, with the kinematic errors of the mechanisms obtained from the network predictions (run 1). $e_{P2} = \left|\hat{P}_2 - P_2\right|$, $e_\alpha = |\hat{\alpha} - \alpha|$.

Task	p_{21} (m)	δ (°)	α (°)	θ_2 (°)	φ_2 (°)	BR e_{P2} (mm)	BR e_α (°)	SCG e_{P2} (mm)	SCG e_α (°)
T1	2.017	161.8	36.3	30.4	92.0	0.34	0.021	23.3	0.43
T2	2.144	174.8	43.7	33.9	70.7	0.24	0.002	7.1	0.35
T3	2.258	163.5	48.5	46.3	92.4	1.46	0.042	3.1	0.37
T4	2.368	164.9	35.2	36.2	84.9	0.99	0.034	5.4	0.32
T5	2.475	160.4	45.0	40.3	88.2	0.15	0.006	2.5	0.07
T6	2.586	158.0	47.4	43.2	79.6	0.67	0.024	11.0	0.11
T7	2.681	162.2	35.8	43.4	75.7	0.96	0.030	8.0	0.09
T8	2.780	150.5	37.4	32.3	82.9	0.98	0.030	47.2	0.73

4.5. Effect of the Number of Configurations and of Measurement Noise

Number of configurations. To justify the choice of $N = 75$ configurations, the configuration-level networks of Sections 4.1–4.3 were retrained with datasets of 15, 25, 50, 75, 150 and 300 equally spaced configurations (crank increments from 2.74° to 0.13°), with 10 paired runs per case (5 runs for BR with 300 configurations because of its computational cost). All networks were evaluated on the same dense independent set of 500 configurations at random crank angles, and the maximum coupler-point deviation e_P was computed as in Section 4.3. The results are given in Table 8.

Table 8. Effect of the number of configurations in the dataset (mean ± standard deviation over the runs; evaluation on 500 independent configurations).

N	Crank Increment (°)	BR MSE (Rad²)	BR Max e_P(mm)	SCG MSE (Rad²)	SCG Max e_P(mm)	SCG R
15	2.74	$(3.7 \pm 12) \times 10^{-8}$	1.05 ± 2.9	0.24 ± 0.62	1450 ± 1300	0.827
25	1.60	$(2.3 \pm 7.0) \times 10^{-10}$	0.11 ± 0.27	$(1.0 \pm 1.4) \times 10^{-2}$	543 ± 370	0.974
50	0.78	$(2.7 \pm 2.7) \times 10^{-14}$	0.0025 ± 0.0024	$(1.6 \pm 1.9) \times 10^{-3}$	251 ± 190	0.995
75	0.52	$(2.4 \pm 3.7) \times 10^{-14}$	0.0010 ± 0.0008	$(1.0 \pm 1.2) \times 10^{-3}$	217 ± 93	0.997
150	0.26	$(2.9 \pm 7.6) \times 10^{-14}$	0.0007 ± 0.0008	$(1.6 \pm 1.3) \times 10^{-4}$	97 ± 67	0.9995
300	0.13	$(4.7 \pm 5.0) \times 10^{-15}$	0.0004 ± 0.0002	$(8.4 \pm 8.8) \times 10^{-5}$	88 ± 85	0.9997

Table 8 shows that the BR network needs about 50 configurations (a crank increment below about 0.8°) to reach its full accuracy: with 15 or 25 configurations the coupler-point deviation increases to about 1 mm and 0.1 mm, respectively, with a large scatter between runs, whereas from 50 configurations on it remains at the micrometre level. The choice of 75 configurations therefore provides a safety margin above this threshold at a small computational cost; using significantly more configurations does not improve the BR results appreciably. The SCG network improves steadily with the number of configurations, but its coupler-point deviation is still about 88 mm with 300 configurations. Its limitation is therefore caused by the training algorithm (early stopping before full convergence) rather than by the size of the dataset.

Influence of the analytical (noise-free) data. Because the data are generated from exact analytical equations, they contain no measurement error, and the mapping to be learned is smooth and one-dimensional. This explains why both algorithms reach high correlation coefficients and why the BR errors are so small. To examine how the results change for imperfect data, Gaussian noise with standard deviations of 0.5 mm and 2 mm was added to the input coordinates (x_c, y_c) of the 75 training configurations, which represents measured rather than computed joint positions, and the networks were evaluated on the noise-free independent set (Table 9).

Table 9. Effect of measurement noise on the input coordinates (75 configurations; mean ± standard deviation over 10 runs; evaluation on 500 noise-free independent configurations).

Noise σ(mm)	BR MSE (Rad^2)	BR Max Angle Error (°)	BR Max e_P(mm)	SCG MSE (Rad^2)	SCG Max e_P(mm)	BR Better (of 10)
0	$(2.4 \pm 3.7) \times 10^{-14}$	3.6×10^{-5}	0.0010 ± 0.0008	$(1.0 \pm 1.2) \times 10^{-3}$	217 ± 93	10
0.5	$(8.7 \pm 2.5) \times 10^{-9}$	0.020	0.55 ± 0.16	$(1.2 \pm 1.2) \times 10^{-3}$	256 ± 140	10
2.0	$(1.2 \pm 0.4) \times 10^{-7}$	0.068	1.94 ± 0.65	$(5.0 \pm 6.3) \times 10^{-4}$	209 ± 160	10

With noisy inputs, the coupler-point deviation of the BR network increases to 0.55 mm and 1.94 mm, i.e., to approximately the level of the noise added to the data. The extremely small errors obtained with noise-free data are therefore a consequence of the exact analytical data, and with measured data the accuracy of the BR network is limited by the quality of the data rather than by the network. The SCG errors are hardly affected by the noise because they are dominated by the incomplete convergence of the training. BR remained more accurate than SCG in all runs and at all noise levels. Together with the kinematic verification in Sections 4.3 and 4.4, this shows that the high correlation of the SCG network does not translate into an accurate mechanism: although its R values exceed 0.99, the mechanisms reconstructed from its predictions deviate from the exact motion by centimetres to decimetres.

4.6. Comparison of the Training Algorithms

Two training algorithms, Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG), were evaluated for the kinematic synthesis of the four-bar linkage mechanism based on the regression correlation, error distribution, and convergence behavior of the trained artificial neural networks (ANNs). The BR-trained network yielded a correlation coefficient of $R = 1.0000$ for the training, test, and combined datasets, and the best training performance was a mean squared error (MSE) of 1.6304×10^{-13} at epoch 76. The associated error histograms revealed prediction errors restricted to a very narrow band of approximately $\pm 10^{-6}$ rad, which were symmetrically distributed around zero for both the training and test subsets, as well as for the complete 75-sample dataset ($R = 1.0000$). The close overlap of the training and test MSE curves throughout the training process, with no divergence in later epochs, demonstrates that the Bayesian Regularization framework successfully controlled the model complexity via its inherent weight-penalty mechanism. This resulted in a network with high data-fitting accuracy; its accuracy on configurations that were not used for training is reported in Section 4.3.

In contrast, the SCG-trained network yielded correlation coefficients of $R = 0.996$, $R = 0.978$, and $R = 0.967$ for the training, validation, and test subsets, respectively ($R = 0.989$ for the combined dataset), indicating a strong but less accurate fit. The training performance curve showed that the validation and test MSE values plateaued around epochs 20–30, with the best validation performance (MSE = 0.010004) recorded at epoch 60. After this point, early stopping halted the training process because the validation error stopped improving while the training error continued to decrease. This behavior is typical of the early-stopping strategy used by SCG to avoid overfitting; however, it also resulted in a less precise overall fit, as evidenced by the wider spread of the error histogram (approximately −0.42 to 0.14 rad) compared with the BR model. Likewise, evaluation on the complete 75-sample dataset yielded $R = 0.98936$, with visibly greater scatter about the ideal fit line and a small number of larger-magnitude outlier errors (up to approximately 0.28 rad), particularly at higher target values.

Taken together, these results demonstrate that both algorithms successfully captured the underlying nonlinear relationships governing the kinematic synthesis of the four-bar

linkage mechanism, but they differed substantially in the quality of the fit and the magnitude of the residual errors. The difference of several orders of magnitude in the achievable MSE and error range between the two approaches can be attributed to their respective regularization strategies. Bayesian Regularization optimizes a combined objective function composed of data-fitting error and weight magnitude within a probabilistic framework. This enables training to continue until a very fine error minimum is reached without triggering overfitting. Conversely, SCG relies on monitoring a separate validation set and halting training once validation performance stagnates. This strategy inherently limits how tightly the network can fit the data, since training is stopped based on validation error trends rather than the continued optimization of a regularized objective function.

These conclusions are not based on a single training run: over 30 random initializations BR gave a lower test MSE than SCG in all 30 runs ($p = 1.9 \times 10^{-9}$, Table 5), and both networks used identical architectures, normalization, initialization methods and data divisions. Moreover, the comparison is not restricted to angle-level regression metrics. When the linkage is reconstructed from the predicted angles, the BR errors correspond to coupler-point deviations, coupler-length inconsistencies and loop-closure errors of the order of micrometres, whereas the isolated large SCG errors can correspond to coupler-point deviations that are significant relative to the prescribed displacement p_{21} (Section 4.3).

The task-level study (Section 4.4) confirms these findings on genuinely new synthesis problems and in terms of the prescribed motion itself. The mechanisms obtained from the BR predictions reached the prescribed position about 27 times more accurately than those obtained from SCG (0.61 mm versus 16.4 mm on average), although the correlation coefficients of the two networks differ only in the fourth decimal place. The value of the present work therefore does not lie in a new network architecture, but in the controlled comparison of the training algorithms and in the kinematic verification, which reveals the practical consequence of the algorithm choice for the synthesized mechanism. Filtering defective solutions from the training data proved effective: none of the 4000 verified predictions produced an unassemblable mechanism or a branch defect.

The results have practical implications when choosing a training algorithm for ANN-based applications in kinematic synthesis. For applications in which high-precision dimensional synthesis is required (e.g., when an ANN surrogate is used to evaluate the configurations of a synthesized linkage inside a design or optimization loop), the significantly lower error levels obtained with Bayesian Regularization are highly beneficial, assuming the computational cost of BR training is acceptable. This cost is typically higher, making BR better suited for small-to-moderate datasets. However, since SCG also led to consistently high correlation coefficients ($R > 0.96$) for all data subsets, it may remain an option for larger datasets or applications where computational efficiency is more critical than the final order-of-magnitude accuracy, provided that its mechanism-level errors (Table 5) are acceptable for the application.

4.7. Comparison with Previous Studies

Table 10 compares the present study with previous ANN-based studies on four-bar synthesis. The comparison is qualitative, because the studies address different synthesis tasks (path, function and motion generation), use different datasets and report different error measures, so their numerical errors cannot be compared directly.

Table 10. Comparison of the present study with previous ANN-based studies on four-bar linkage synthesis.

Study	Synthesis Task	Approach	Main Focus
Vasiliu and Yannou [21]	Path generation	Neural network trained on kinematic simulations (case-based)	Fast approximate solutions, used as initial guesses for optimization
Khan et al. [22]	Path generation (closed curves)	ANN with Fourier descriptors, followed by local optimization	Representation of curve shape
Deshpande and Purwar [4]	Motion generation	Pattern recognition with autoencoder networks	Defect-free invariant motions
Denizhan [1]	Two-position motion generation	Feedforward ANN, Levenberg–Marquardt	Feasibility of ANN-based two-position synthesis
Nurizada and Purwar [24]	Path generation	Variational autoencoder with a fully connected network	Invariant representation of coupler curves
Lee et al. [19]	Kinematic and quasi-static conditions	Conditional generative adversarial network	Generative synthesis with target conditions
Nurizada et al. [25]	Path generation, several mechanism types	Conditional β-variational autoencoder	Multiple solutions and mechanism types
Kim et al. [20]	Function generation	Direct parameterization, mixture of expert networks	Diverse four-bar function generators
Present study	Two-position motion generation	Feedforward ANN, BR vs. SCG (controlled, repeated)	Effect of training algorithm; kinematic verification; defect filtering

The previous studies concentrate mainly on network architectures, curve representations and datasets for path or function synthesis. The present study addresses a complementary question that these studies do not isolate: how strongly the training algorithm alone affects the quality of the synthesized mechanism, assessed by kinematic simulation. Its result is consistent with the earlier comparison for the slider-crank mechanism [17], in which BR also gave the highest accuracy. It also adds that the difference between the algorithms, which is barely visible in the correlation coefficient, amounts to more than one order of magnitude in the error of the synthesized motion.

Compared with analytical precision-point synthesis [2,3], the trained network provides a solution in a single forward pass without solving the synthesis equations, which is useful when many candidate designs must be evaluated, for example inside an optimization or design-exploration loop, and it can be trained on defect-free solutions only. Compared with optimization-based synthesis [5,6], it requires no initial guess and no iterative search once trained. However, for two-position synthesis the closed-form solution is itself inexpensive, so the ANN is not proposed as a replacement for it; the two-position problem is used here because its exact solution allows a rigorous evaluation of the training algorithms.

4.8. Validity Limits

The conclusions of this study are valid under the following conditions: (i) planar four-bar linkages with revolute joints; (ii) two-position motion generation; (iii) task parameters within the sampled ranges $p_{21} \in [2.0, 2.8]$ m, $\delta \in [150°, 180°]$, $\alpha \in [35°, 52°]$, $\theta_2 \in [30°, 48°]$ and $\varphi_2 \in [70°, 100°]$, with the free choices z, ψ, s and σ fixed; (iv) feedforward networks with one hidden layer; and (v) computed data, or measured data with a noise level of

the order of 1 mm (Section 4.5). Outside these conditions, in particular for extrapolation beyond the sampled ranges, the accuracy of the networks has not been verified.

In addition, the scope of the present results should be stated clearly. Both networks were trained and tested on configurations of a single mechanism, obtained from one analytical two-position synthesis solution and lying on a single assembly branch, and the independent test set of Section 4.3 assesses interpolation between training configurations within the same range of motion. Section 4.4 extends the evaluation to new synthesis tasks, but only within the sampled ranges of p_{21}, δ, α, θ_2 and φ_2 and with the free choices z, ψ, s and σ held fixed. The results therefore do not validate the networks for arbitrary four-bar mechanisms, for three- or multi-position synthesis, for path or function generation, for the automatic avoidance of branch, order or circuit defects, or for reliability in engineering design. Future work will address these aspects by training on datasets that span different prescribed positions and link dimensions (with the design requirements as inputs and the link dimensions as outputs), by extending the defect filtering of the training data to multi-position tasks, by extending the comparison to further training algorithms (e.g., Levenberg–Marquardt without regularization or resilient backpropagation), and by experimental validation of synthesized linkages.

5. Conclusions

This study compared Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training of artificial neural networks for the two-position kinematic synthesis of four-bar linkages, under controlled conditions and with kinematic verification of every prediction. The main conclusions are as follows.

(1) At the configuration level, BR reached near-perfect accuracy ($R = 1.0000$, test MSE of the order of 10^{-14} rad^2) and was more accurate than SCG in all 30 paired runs (Wilcoxon $p = 1.9 \times 10^{-9}$). The linkages reconstructed from the BR predictions deviated from the exact motion by less than 3 µm, whereas those reconstructed from the SCG predictions deviated by up to 0.72 m, although the SCG network still reached $R > 0.96$.

(2) In task-level synthesis, networks mapping the two-position task and the free choices to the link vectors were trained on 300 defect-free tasks and tested on 200 new tasks. All mechanisms obtained from the BR predictions were assemblable and free of branch and toggle defects, and reached the prescribed second position within 0.61 mm on average (6.4 mm in the worst case) and the prescribed coupler rotation within 0.2°; the corresponding values for SCG were 16 mm (138 mm) and 4.9°.

(3) The BR accuracy no longer improves beyond about 50 configurations, which justifies the dataset of 75 configurations, while SCG remains inaccurate even with 300 configurations. With measurement noise of 0.5–2 mm on the inputs, the BR errors increase to approximately the noise level, so that with measured data the accuracy is limited by the data quality.

(4) Correlation coefficients and MSE values of the network outputs are not sufficient indicators of synthesis quality: networks with almost identical R values (1.000000 and 0.9998) differed by a factor of about 27 in the error of the synthesized motion. Kinematic verification of the predicted mechanisms should therefore be part of the evaluation of ANN-based synthesis methods.

Compared with previous ANN-based studies, which concentrate on network architectures, curve representations and datasets for path and function synthesis [19–26], the present work isolates the effect of the training algorithm and quantifies it at the mechanism level. Its findings agree with those obtained for the slider-crank mechanism [17], where BR also gave the highest accuracy. The results are valid for planar four-bar linkages with revolute joints in two-position motion generation, within the sampled ranges of the task parameters and with fixed free choices (Section 4.8). Future work will extend the approach

to three- and multi-position synthesis, path and function generation, variable free choices, more complex mechanisms and experimental validation of synthesized linkages.

Author Contributions: Conceptualization, O.D.; methodology, O.D.; software, O.D. and B.D.; validation, O.D.; formal analysis, O.D.; investigation, O.D.; resources, O.D. and B.D.; data curation, O.D.; writing—original draft preparation, O.D.; writing—review and editing, M.-S.C. and B.D.; visualization, O.D.; supervision, M.-S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study (the 75-configuration training/test dataset, the 74-configuration independent test set and the task-level datasets) and the MATLAB.

Acknowledgments: The authors wish to express their gratitude to the Ministry of National Education of the Republic of Türkiye, which indirectly made this work possible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Denizhan, O. Two-position synthesis of the four-bar planar linkage mechanisms using artificial neural networks. In Proceedings of the 5th International Conference on Applied Engineering and Natural Sciences (ICAENS 2023), Konya, Türkiye, 10–12 July 2023; pp. 964–970. [CrossRef]
2. Hartenberg, R.S.; Denavit, J. *Kinematic Synthesis of Linkages*; McGraw-Hill: New York, NY, USA, 1964.
3. Norton, R.L. *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*, 5th ed.; McGraw-Hill: New York, NY, USA, 2012.
4. Deshpande, S.; Purwar, A. A machine learning approach to kinematic synthesis of defect-free planar four-bar linkages. *J. Comput. Inf. Sci. Eng.* **2019**, *19*, 021004. [CrossRef]
5. Cabrera, J.A.; Simon, A.; Prado, M. Optimal synthesis of mechanisms with genetic algorithms. *Mech. Mach. Theory* **2002**, *37*, 1165–1177. [CrossRef]
6. Lin, W.Y. A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. *Mech. Mach. Theory* **2010**, *45*, 1096–1107. [CrossRef]
7. Denizhan, O.; Chew, M.S. Linkage mechanism optimization and sensitivity analysis of an automotive engine hood. *Int. J. Automot. Sci. Technol.* **2018**, *2*, 7–16. [CrossRef]
8. Denizhan, O. Three-Position Four-Bar Linkage Mechanism Synthesis, Static Balancing and Optimization of Automotive Engine Hood. Master's Thesis, Lehigh University, Bethlehem, PA, USA, 2015.
9. Denizhan, O. Incorporation of Kinematic Analysis, Synthesis and Optimization into Static Balancing. Ph.D. Thesis, Lehigh University, Bethlehem, PA, USA, 2021.
10. Denizhan, O.; Chew, M.S. Optimum synthesis and design of a hood linkage for static balancing in one-step. *Teh. Vjesn.* **2023**, *30*, 855–862. [CrossRef]
11. Denizhan, O.; Chew, M.S. Optimum design of a spring-loaded linkage mechanism in the presence of friction for static balancing. *J. Eng. Des. Anal.* **2019**, *2*, 5–11. [CrossRef]
12. Jack, H.; Lee, D.M.A.; Buchal, R.O.; Elmaraghy, W.H. Neural networks and the inverse kinematics problem. *J. Intell. Manuf.* **1993**, *4*, 43–66. [CrossRef]
13. Ghasemi, J.; Moradinezhad, R.; Hosseini, M.A. Kinematic synthesis of parallel manipulator via neural network approach. *arXiv* **2019**, arXiv:1904.04668. [CrossRef]
14. Han, X.; Zhao, P.; Zhao, X.; Zi, B. Review on machine learning-based approaches for the kinematic analysis and synthesis of mechanisms. *Front. Mech. Eng.* **2025**, *20*, 11. [CrossRef]
15. Heyrani Nobari, A.; Srivastava, A.; Gutfreund, D.; Ahmed, F. LINKS: A dataset of a hundred million planar linkage mechanisms for data-driven kinematic design. In Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE 2022), St. Louis, MO, USA, 14–17 August 2022; Volume 3A, p. V03AT03A013. [CrossRef]
16. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [CrossRef]
17. Denizhan, O. Comparison of different supervised learning algorithms for position analysis of the slider-crank mechanism. *Alex. Eng. J.* **2024**, *92*, 39–49. [CrossRef]

18. Denizhan, O. Position, velocity, and acceleration analysis of the quick-return mechanism. In Proceedings of the 5th International Conference on Scientific and Academic Research (ICSAR 2024), Konya, Türkiye, 23–24 December 2024; pp. 1065–1071.
19. Lee, S.; Kim, J.; Kang, N. Deep generative model-based synthesis framework of four-bar linkage mechanisms with target conditions. *J. Comput. Des. Eng.* **2024**, *11*, 318–332. [CrossRef]
20. Kim, W.R.; Jung, J.; Ha, J.U.; Lee, D.; Shim, J.K. Data-driven dimensional synthesis of diverse planar four-bar function generation mechanisms via direct parameterization. *arXiv* **2025**, arXiv:2507.08269. [CrossRef]
21. Vasiliu, A.; Yannou, B. Dimensional synthesis of planar mechanisms using neural networks: Application to path generator linkages. *Mech. Mach. Theory* **2001**, *36*, 299–310. [CrossRef]
22. Khan, N.; Ullah, I.; Al-Grafi, M. Dimensional synthesis of mechanical linkages using artificial neural networks and Fourier descriptors. *Mech. Sci.* **2015**, *6*, 29–34. [CrossRef]
23. Deshpande, S.; Purwar, A. An image-based approach to variational path synthesis of linkages. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 021005. [CrossRef]
24. Nurizada, A.; Purwar, A. An invariant representation of coupler curves using a variational autoencoder: Application to path synthesis of four-bar mechanisms. *J. Comput. Inf. Sci. Eng.* **2024**, *24*, 011008. [CrossRef]
25. Nurizada, A.; Lyu, Z.; Purwar, A. Path generative model based on conditional β-variational auto encoder for four-bar mechanism design. *J. Mech. Robot.* **2025**, *17*, 061004. [CrossRef]
26. Nurizada, A.; Dhaipule, R.; Lyu, Z.; Purwar, A. A dataset of 3M single-DOF planar 4-, 6-, and 8-bar linkage mechanisms with open and closed coupler curves for machine learning-driven path synthesis. *J. Mech. Des.* **2025**, *147*, 041702. [CrossRef]
27. MacKay, D.J.C. Bayesian interpolation. *Neural Comput.* **1992**, *4*, 415–447. [CrossRef]
28. Foresee, F.D.; Hagan, M.T. Gauss-Newton approximation to Bayesian learning. In Proceedings of the International Conference on Neural Networks (ICNN'97), Houston, TX, USA, 9–12 June 1997; Volume 3, pp. 1930–1935. [CrossRef]
29. Chandra, A.; Suaib, M.; Beg, D.R. Web spam classification using supervised artificial neural network algorithms. *arXiv* **2015**, arXiv:1502.03581. [CrossRef]
30. Møller, M.F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **1993**, *6*, 525–533. [CrossRef]
31. Hirschen, K.; Schäfer, M. Bayesian regularization neural networks for optimizing fluid flow processes. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 481–500. [CrossRef]
32. Rakhshkhorshid, M.; Teimouri Sendesi, S.A. Bayesian regularization neural networks for prediction of austenite formation temperatures (Ac1 and Ac3). *J. Iron Steel Res. Int.* **2014**, *21*, 246–251. [CrossRef]
33. Demuth, H.B.; Beale, M.H.; De Jesús, O.; Hagan, M.T. *Neural Network Design*, 2nd ed.; Martin Hagan: Stillwater, OK, USA, 2014.
34. Bishop, C.M. *Neural Networks for Pattern Recognition*; Clarendon Press: Oxford, UK, 1995.
35. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1994.
36. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
37. Hülako, H.; Çakar, O. A study on four-bar mechanism with constant coupler point velocity: Modelling, numerical simulation and experimental application. *Gazi J. Eng. Sci.* **2025**, *11*, 1–15. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.