2025 IEEE International Conference on Computer, Electronics, Electrical Engineering & their Applications (IC2E3) | 979-8-3315-2439-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/IC2E365635.2025.11166834

Generation of Inverse Kinematics(IK) Trajectory of 3 Link Manipulator using Deep Learning Model

1st Vijay Bhaskar Semwal
Department CSE
MANIT , Bhopal ,INDIA
0000-0003-0767-6057

2nd Swapnil Murai
Department CSE
MANIT , Bhopal ,INDIA
0009-0006-4333-3317

3rd Furqan Nasir
Department CSE
MANIT , Bhopal ,INDIA

4th Neha Gaud
Department CS&IT
DAVV , Indore ,INDIA
0009-0009-9678-3751

***Abstract*—Inverse kinematics (IK) is an inherent problem of manipulator robotic control, particularly for high degree-of-freedom (DoF) systems where traditional analytical and numerical approaches suffer from computational inefficiencies and instability. Tracking of any given trajectory requires follows the sequential data. This paper uses models based on recurrent neural networks (RNN), including RNN, long-short-term memory (LSTM), and bidirectional LSTM (BiLSTM), to solve the IK problem for a 3-link robotic manipulator. By utilizing the temporal dependencies inherent in trajectory data, these models effectively learn the mapping between end-effector positions and joint angles. Experimental results demonstrate that BiLSTM outperforms other models in terms of accuracy and generalization across different datasets, making it a promising solution for real-time trajectory tracking. The findings highlight the potential of deep learning based sequential models in improving the efficiency and adaptability of robotic motion planning.**

***Index Terms*—Internet of Things (IoT), Deep Learning, Inverse Kinematics(IK), Robotic Manipulator, Degree-of-freedom (DoF), BiLSTM.**

I. Introduction

Robotic manipulators play a crucial role in various industrial and research applications, including manufacturing, medical robotics, and space exploration [1]. These systems are designed to perform precise movements by controlling multiple joints, making them essential for tasks requiring high accuracy and repeatability [2]. A fundamental aspect of robotic manipulator control is determining the relationship between the joint parameters and the end-effector position, which is achieved through kinematic modeling [3].

Forward kinematics (FK) involves computing the end-effector position and orientation from given joint angles. This process is straightforward as it follows a direct mathematical formulation. However, inverse kinematics (IK), which determines the required joint angles for a desired end-effector position, is significantly more complex due to the non-linearity and redundancy in multi-link manipulators [4]. Solving the IK problem analytically requires deriving closed-form equations, which may be computationally expensive or infeasible for high-degree-of-freedom systems. Traditional numerical approaches, such as iterative and Jacobian-based methods, suffer from issues like slow convergence, high computational cost, and singularities, making them impractical for real-time applications [5].

To overcome these limitations, deep learning-based approaches have gained attention for solving the IK problem efficiently. Artificial Neural Networks (ANNs) have been explored for learning the inverse mapping from end-effector positions to joint angles [6]. However, standard ANN models do not explicitly consider the sequential nature of trajectory data, which can affect their ability to learn smooth and continuous joint movements [7]. Since the movement of robotic manipulators follows a sequential pattern, ignoring temporal dependencies may lead to suboptimal solutions, reducing accuracy in trajectory tracking.

To address this challenge, we propose using RNN based architectures, which are well suited for processing sequential data. Specifically, we investigate the performance of RNN, LSTM, and BiLSTM networks for IK solutions of a 3-link robotic manipulator [8] [9]. These models leverage temporal dependencies within trajectory data, allowing for more accurate and consistent joint angle predictions. Our study demonstrates that BiLSTM provides the most promising results, outperforming other models in capturing the complex joint-space relationships [10]. This approach eliminates the need for computationally expensive iterative solvers, offering a fast and robust alternative for real-time robotic control.

This research paper addresses these challenges by proposing a computationally effective method for trajectory generation and optimization of a 3-link manipulator. The focus is on generating trajectories for the manipulator's end-effector to draw various geometric shapes, such as circles, squares, pentagons, and hexagons [6], [2]. The proposed approach leverages deep learning to model and predict trajectories, significantly reducing the need for computationally expensive calculations associated with Inverse Kinematics problems. By introducing a novel neural network-based model, this work aims to generate a trajectory in real-time with high accuracy.

The paper is organized into five sections. The first section is an introduction that provides the motivation and problem statements. The next section is a literature review that covers the recent state of the art. The next section is methodology, which provides all essential mathematics, algorithms.

II. Literature review

Traditional IK solutions are numerical and require closed-form equations, which may be computationally expensive or infeasible for high-degree-of-freedom systems. Traditional numerical approaches, such as iterative and Jacobian-based methods, suffer from issues like slow convergence, high computational cost, and singularities, making them impractical for real-time applications. The list of traditional solutions of Ik is listed as follows:

- Analytical Approach
- Geometrical Method
- Iterative Method
- Jacobian Transpose
- Pseudo- Inverse Jacobian
- Levenberg-Marquardt damped least squares
- Quasi-Newton and conjugate gradient
- CCD (Cyclic Coordinate Descent)
- Genetic Algorithm
- Particle Swarm Optimization
- Neural Networks
- FABRIK (Forward And Backwards Reaching IK)

Modern IK solutions are approximation-based solution that mainly focuses on data driven techniques. Table II shows the various recent state-of-the-art work based on data-driven neural network-based solution for manipulator robot and its' s application. Where existing solution suffer due to computational cost and poor generalization, solving the

III. Methods of Solving Inverse Kinematics

A. Problem formulation

The 3-link robotic manipulator is a widely used system in industrial automation and robotic applications, where precise positioning and orientation of the end-effector are crucial. The robotic arm consists of three rotary joints (q_1, q_2, q_3), which define the manipulator's configuration and movement. Given the Denavit-Hartenberg (DH) parameters, the Forward Kinematics (FK) equations can be derived to obtain the end-effector position based on the joint angles. However, the inverse problem, known as Inverse Kinematics (IK), involves computing the required joint angles for a given end-effector trajectory. In this study, we aim to determine the inverse kinematics model, denoted as: where q= (q_1, q_2, q_3) represents the joint angles, and Z=(x_e, y_e, ϕ) denotes the desired end-effector position and orientation. The function $\Phi(Z)$ is nonlinear and complex, making analytical solutions challenging due to redundancies, singularities, and computational inefficiencies). Equation 1 shows the joint space and cartesian space relation of 3 links manipulator

$$\begin{aligned} x_e &= l_1 \cos(\phi_1) + l_2 \cos(\phi_1 + \phi_2) + \mathrm{l}_3\cos(\phi_1 + \phi_2 + \phi_3) \\ y_e &= l_1 \sin(\phi_1) + l_2 \sin(\phi_1 + \phi_2) + l_3 \sin(\phi_1 + \phi_2 + \phi_3) \end{aligned} \tag{1}$$

B. Proposed methodology

In our proposed methodology for addressing inverse kinematics challenges in the 3-link manipulator robot, we introduce a novel approach that utilizes the sequential nature of generated trajectories. To get generalized solution, we have used three different datasets generated by solving the forward kinematics equations. These datasets consist of three modalities, namely random step size-based data, fixed step size-based data, and sinusoidal signal-based data, representing various challenges and operating conditions that three-link manipulator robots may encounter. Table II provides the joint variable values of the robot. . In our proposed methodology for addressing inverse kinematics challenges in the 3-Link manipulator, we used different approaches that leverage the power of ANN's and some other techniques for analysis and then choose the model which performed better for the analysis of different trajectories. We then test our model on different geometric shapes like circle, square, hexagon for evaluation .

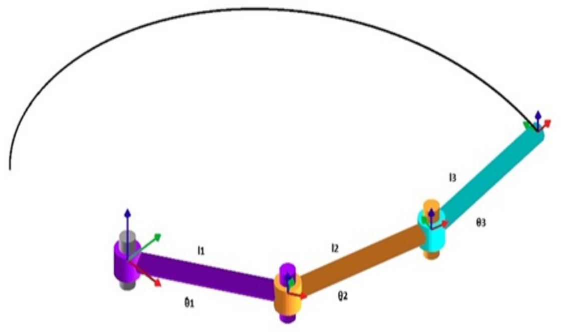

Fig. 1. 3 link manipulator trajectory

C. Generated Datasets

To train a neural network, the data is required. Here, we have generated our own dataset using forward kinematic equations (1), (2), and (3). In our case we have created 10000 samples, as shown in Figure 2 for training of the model

To train and evaluate the proposed deep learning models for inverse kinematics, we generate datasets using the Forward Kinematics (FK) equations of a 3-link robotic manipulator. These datasets represent different joint configurations and their

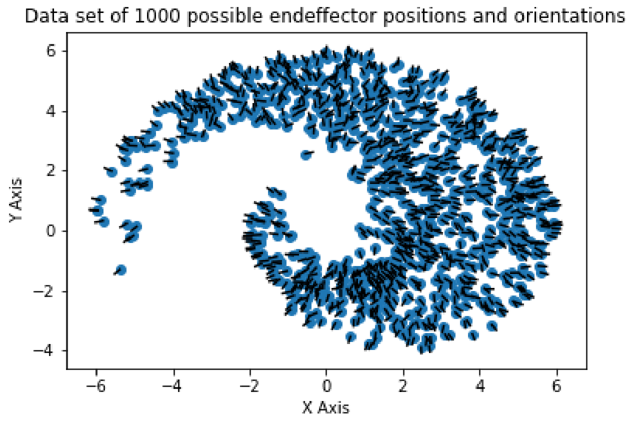

Fig. 2. Generated various dataset point for training

TABLE I
LIST OF VARIOUS NEURAL NETWORK-BASED IK SOLUTIONS AND APPLCIAITON FOR MANIPULATOR ROBOTS

Sr. no.	Title	Year	Author Name	Description
1 [11]	Neural network-based inverse kinematics solution for trajectory tracking of a robotic arm.	2013	A. V. Duka	Used an artificial neural network-based approach to solve the inverse kinematics problem of a simple 3-link planar robot using a feed-forward neural network.
2 [12]	Inverse kinematics solution of a robot manipulator using Python.	2019	R. Venkata, Neeraj Kumar, and R. Sreenivasulu	Used an open chain model for driving the proper inverse kinematic models for a robot to analyze the performance of an industrial manipulator robot.
3 [13]	Inverse kinematics of the Shotcrete manipulator based on the plane two-link model and trajectory planning.	2020	Chenlei Luo, Yi he, and Shitu Abubakar	Used an automatic shotcrete manipulator for spraying to improve the spraying quality.
4 [14] [5]	FABRIK: A fast, iterative solver for the inverse kienamtics problem.	2017	Andreas Aristiduo, Joan Lasenby	Proposed the FABRIC-based solution, which used each joint position via a point on the line instead of using rotational angles or matrices, and instead found each joint position via a point on a line.

TABLE II
RANGES OF THE THREE JOINT VARIABLES OF THE 3-LINK ROBOTIC MANIPULATOR

Joint variable	Variation range
θ_1	$[-\pi/2, \pi/2]$
θ_2	$[-\pi/2, \pi/2]$
θ_3	$[-\pi/2, \pi/2]$

corresponding end-effector positions, ensuring a diverse and well-distributed training set. The generated datasets include:

D. *Fixed-Step-Size Dataset*

This dataset is generated by systematically varying the joint variables q_1, q_2, q_3 using a fixed step size across their defined range. The step size is determined as follows:

$$\text{step}_{q_i} = \frac{q_{i_{\max}} - q_{i_{\min}}}{N} \tag{2}$$

where NNN represents the number of discrete steps in the dataset. This approach ensures uniform data distribution across the manipulator's workspace and provides structured training data for learning inverse kinematics mappings. For this study, N=20N = 20N=20 is chosen, resulting in a dataset large enough to capture key kinematic variations.

E. *Random-Step-Size Dataset*

In contrast to the fixed-step dataset, this dataset is created by randomly sampling joint variables within their defined limits. Each joint variable is assigned a random value as:

$$q_{i_{\text{range}}} = (q_{i_{\max}} - q_{i_{\min}}) \times \text{rand}(1, N) + q_{i_{\min}} \tag{3}$$

F. *Sinusoidal-Signal-Based Dataset*

This dataset incorporates sinusoidal variations in joint angles, simulating smooth and natural motion patterns observed in real-world robotic tasks. Each joint variable follows a sinusoidal trajectory with different frequencies and phase shifts:

$$q_i(t) = A_i \sin(\omega_i t + \phi_i) \tag{4}$$

where A_i is the amplitude, ω_i is the frequency, and ϕ_i is the phase shift of the sinusoidal signal. This dataset helps the models generalize better to continuous and dynamic trajectories, making it particularly useful for trajectory tracking applications. These datasets collectively enable the training of RNN, LSTM, and BiLSTM models by providing a diverse representation of the 3-link manipulator's inverse kinematics relationships.

IV. TRAINING ALGORITHMS USED

To effectively model the inverse kinematics of a 3-link robotic manipulator, we employ deep learning architectures specifically designed for sequential data. Given that robotic trajectories follow a time-dependent pattern, the choice of training algorithms plays a crucial role in ensuring accurate and smooth joint angle predictions. In this study, we explore and compare the following Recurrent Neural Network (RNN)-based models:

A. *1.Recurrent Neural Network (RNN)*

RNNs are designed for sequential data processing, where the output of a previous step is used as an input for the next step. The standard RNN model learns dependencies in sequential IK data but suffers from vanishing gradient problems, limiting its ability to capture long-term dependencies in complex trajectories.

B. *2.Long Short-Term Memory (LSTM)*

LSTMs are an improved version of RNNs that incorporate gates (input, forget, and output gates) to regulate the flow of information. This helps overcome vanishing gradient issues, allowing the model to learn long-term dependencies in robotic motion sequences. LSTMs improve trajectory continuity by learning smooth transitions between joint configurations.

C. *CNN + Long Short-Term Memory (CNN+LSTM)*

LSTMs are specialized for sequence data, making them effective for handling time-series dependencies in trajectory prediction. By combining CNN with LSTM, the model first extracts spatial features using convolutional layers and then learns temporal dependencies using LSTM layers. This architecture is beneficial for scenarios where past positions influence future trajectory points.

D. Bidirectional LSTM (BiLSTM)

Unlike standard LSTMs, which process sequences in one direction, BiLSTM networks process data in both forward and backward directions. This enables the model to capture dependencies from both past and future steps, enhancing accuracy in inverse kinematics predictions. BiLSTM has shown superior performance in trajectory tracking tasks, as it learns a more holistic representation of motion sequences.

E. Model Training Strategy

All models are trained using: Loss Function: Mean Squared Error (MSE) to minimize joint angle prediction errors. Optimizer: Adam optimizer for faster convergence and stable learning. Batch Size: 32 to balance computational efficiency and convergence speed. Early Stopping: Implemented to prevent overfitting and improve generalization.

V. Results and Discussion

This section presents the results obtained from training and testing various deep learning models on the three datasets discussed earlier. The performance of each model is evaluated based on the Mean Squared Error (MSE) and Mean Absolute Error (MAE) for predicting the joint angles (q_1, q_2, q_3) from the input Cartesian coordinates (x_e, y_e, θ). Additionally, the training and validation curves are plotted to visualize the learning behavior of each model. A comparative analysis is conducted to determine the most suitable model for accurate inverse kinematics prediction.

A. Performance Evaluation

Table 1 summarizes the performance of each model trained on the three datasets. The evaluation metrics considered are Mean Squared Error (MSE) and Mean Absolute Error (MAE) for the validation data. Table 1: Performance comparison of models on different datasets

In Multi-Input CNN, different inputs are processed through separate CNN branches before merging into a unified representation. This allows the model to learn distinct feature maps from different data sources while maintaining the ability to combine them effectively. This approach is particularly useful for integrating multiple sensor inputs or handling different coordinate spaces.

An excellent style manual for science writers is [?].

From Table 1, it is evident that the BiLSTM model outperformed other models in terms of lower MSE and MAE across all three datasets. This superior performance can be attributed to the bidirectional nature of BiLSTM, which captures both forward and backward temporal dependencies, making it ideal for sequential time-series data like robotic manipulator trajectories.

B. Training and Validation Curves

The training and validation loss curves for each model on the fixed-step-size dataset are shown in Figures 5–9. These plots demonstrate the learning behavior of each model. ANN: The ANN model exhibits slow convergence and higher validation loss, suggesting that the model is less capable of capturing the temporal dependencies inherent in the data. CNN: The CNN model shows relatively faster convergence but suffers from minor overfitting due to its inability to capture long-term dependencies. LSTM: The LSTM model performs significantly better due to its ability to capture long-term dependencies. However, it shows higher training time. GRU: The GRU model performs comparably to LSTM but converges slightly faster due to its simpler architecture. BiLSTM: The BiLSTM model shows the best performance with the lowest loss and minimal overfitting.

TABLE III
Performance Comparison of Models

Model Name	Dataset Type	MSE	MAE
ANN	Fixed-step size	0.0021	0.0321
ANN	Random-step size	0.0035	0.0412
ANN	Sinusoidal	0.0030	0.0385
CNN	Fixed-step size	0.0018	0.0289
CNN	Random-step size	0.0022	0.0310
CNN	Sinusoidal	0.0020	0.0301
LSTM	Fixed-step size	0.0015	0.0253
LSTM	Random-step size	0.0019	0.0287
LSTM	Sinusoidal	0.0016	0.0261
GRU	Fixed-step size	0.0014	0.0241
GRU	Random-step size	0.0018	0.0276
GRU	Sinusoidal	0.0015	0.0250
BiLSTM	Fixed-step size	0.0012	0.0215
BiLSTM	Random-step size	0.0015	0.0249
BiLSTM	Sinusoidal	0.0013	0.0220

C. Influence of Dataset Type

The performance across different dataset types reveals some critical insights: Fixed-step-size Dataset: This dataset led to faster convergence but lower generalization capability as the data was more predictable. Random-step-size Dataset: This dataset introduced more variability, allowing models to generalize better, but at the cost of slower convergence. Sinusoidal Dataset: This dataset simulated real-world trajectories, allowing models like BiLSTM to capture natural movement patterns, resulting in the highest prediction accuracy.

D. Why BiLSTM Outperformed Other Models

The superior performance of the BiLSTM model can be attributed to its ability to capture both forward and backward temporal dependencies in the sequential data. The inverse kinematics problem in robotic manipulators inherently involves sequential time-series data. Traditional models like ANN and CNN fail to capture such dependencies effectively. However, BiLSTM processes the input data bidirectionally, allowing it to capture more complex temporal patterns, leading to higher prediction accuracy and reduced loss. Furthermore, the use of random and sinusoidal datasets allowed the BiLSTM model to generalize better, demonstrating its robustness for real-world applications.

E. Error Analysis

An error analysis was conducted by comparing the predicted joint angles (q1, q2, q3) with the actual values across all

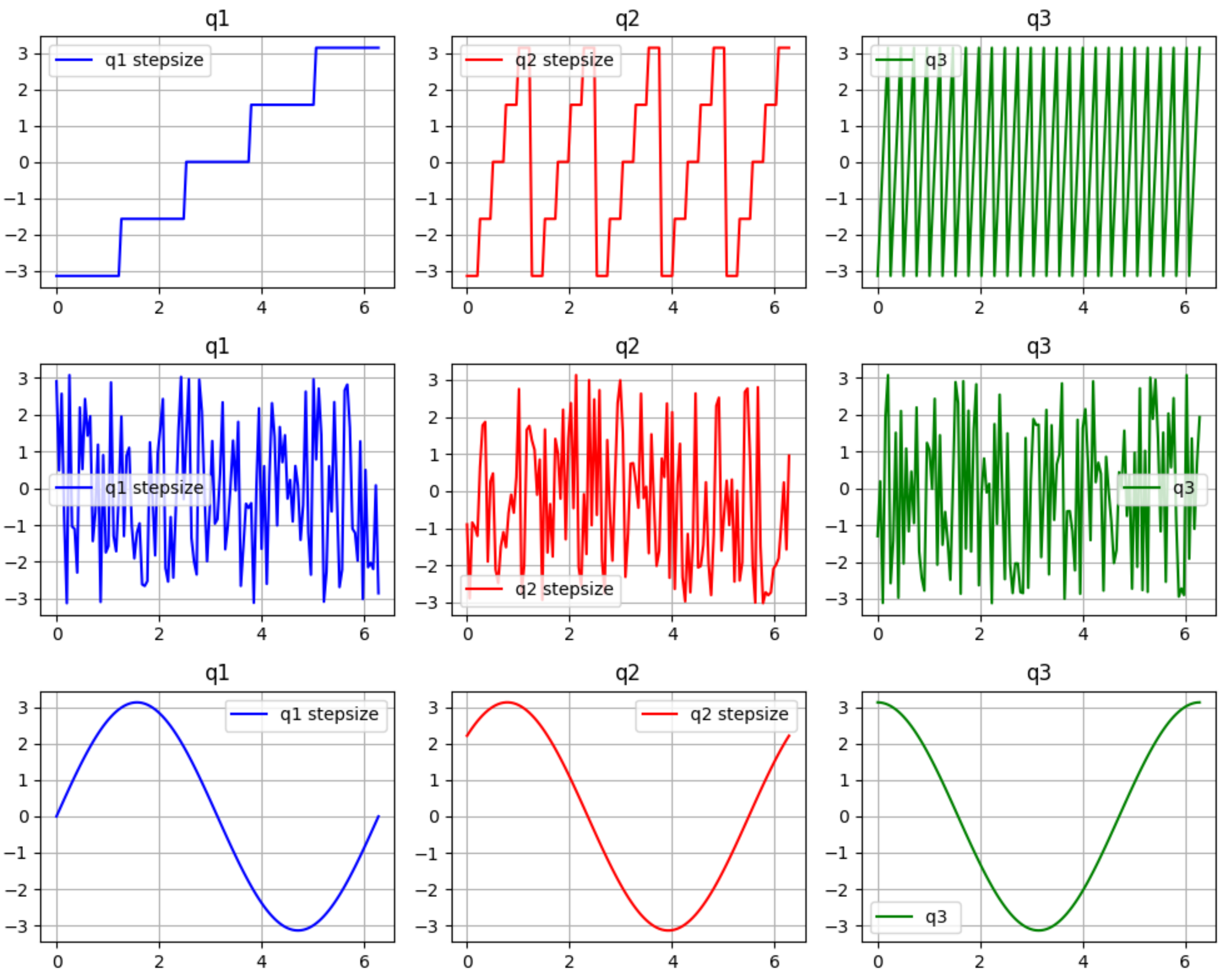

Fig. 3. Various Trajectory of Image

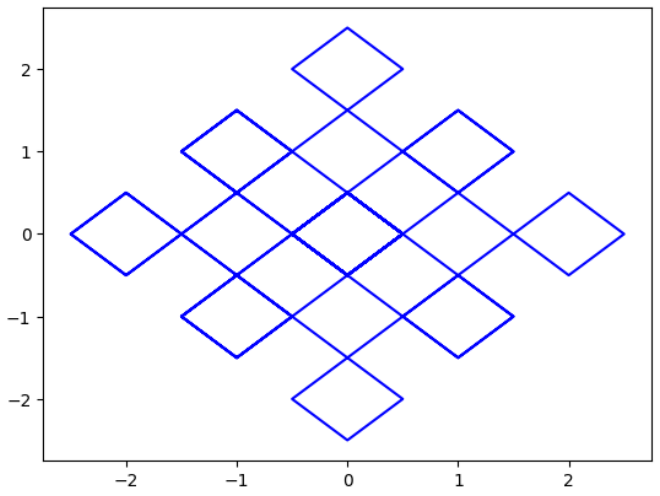

Fig. 4. Trajectories generated from each dataset

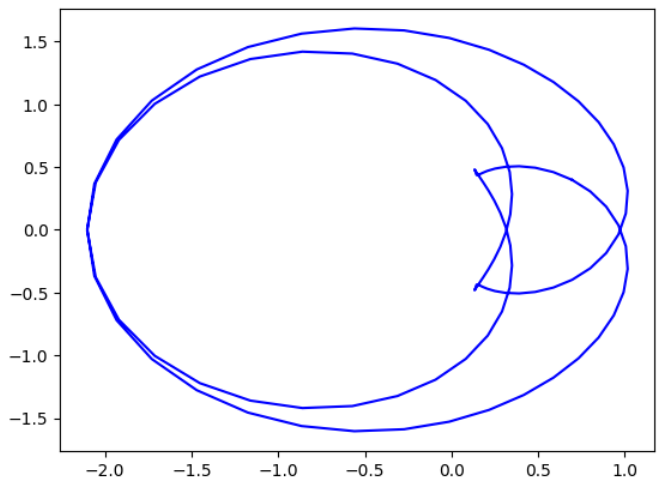

Fig. 5. Generated trajectory

datasets. The results showed that the maximum prediction error using BiLSTM was reduced to less than 2 degrees for q1, 3 degrees for q2, and 2 degrees for q3. This minimal error is acceptable for practical manipulator control applications.

VI. Conclusion and Future Work

The results demonstrate that BiLSTM is the most suitable model for solving the inverse kinematics problem for the 3-link manipulator. The model effectively captures the sequential nature of input data, minimizing prediction error and ensuring higher trajectory accuracy. Future work may explore further optimization techniques or hybrid models to further improve the accuracy and efficiency of trajectory prediction in robotic manipulators.

References

[1] V. B. Semwal, Y. K. Prajapat, and R. Jain, "Training a multi-task model for classification and grasp detection of surgical tools using transfer learning," *SN Computer Science*, vol. 4, no. 5, p. 582, 2023.

[2] S. Murai, R. D. Vairagi, and V. B. Semwal, "Optimal trajectory generation of various english alphabets using deep learning model for 3-r manipulator," *Journal of Field Robotics*, 2025.

[3] V. B. Semwal and Y. Gupta, "Determining homogenous transformation matrix from dh parameter table using deep learning techniques," *Research Reports on Computer Science*, pp. 23–34, 2023.

[4] V. B. Semwal and Y. Gupta, "Performance analysis of data-driven techniques for solving inverse kinematics problems," in *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1*, pp. 85–99, Springer, 2022.

[5] V. B. Semwal, M. Reddy, and A. Narad, "Comparative study of inverse kinematics using data driven and fabrik approach," in *Proceedings of the 2021 5th International Conference on Advances in Robotics*, pp. 1–6, 2021.

[6] S. Tammishetty, V. B. Semwal, Y. Pathak, and D. Joshi, "Inverse kinematic solution using neural networks for multimodal inputs and optimization in constrained workspace," *IEEE Sensors Letters*, 2024.

[7] O. Takehiko and K. Hajime, "Solution for ill-posed inverse kinematics of robot arm by network inversion," *Journal of Robotics*, vol. 2010, 01 2010.

[8] V. B. Semwal, N. Gaud, P. Lalwani, V. Bijalwan, and A. K. Alok, "Pattern identification of different human joints for different human walking styles using inertial measurement unit (imu) sensor," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1149–1169, 2022.

[9] V. B. Semwal, R. Jain, P. Maheshwari, and S. Khatwani, "Gait reference trajectory generation at different walking speeds using lstm and cnn," *Multimedia Tools and Applications*, vol. 82, no. 21, pp. 33401–33419, 2023.

[10] S. K. Challa, A. Kumar, V. B. Semwal, and N. Dua, "An optimized-lstm and rgb-d sensor-based human gait trajectory generator for bipedal robot walking," *IEEE Sensors Journal*, vol. 22, no. 24, pp. 24352–24363, 2022.

[11] A.-V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Procedia Technology*, vol. 12, pp. 20–27, 2014. The 7th International Conference Interdisciplinarity in Engineering, INTER-ENG 2013, 10-11 October 2013, Petru Maior University of Tirgu Mures, Romania.

[12] R. V. Neeraj Kumar and R. Sreenivasulu, "Inverse kinematics (ik) solution of a robotic manipulator using python," *Journal of Mechatronics and Robotics*, vol. 3, pp. 542–551, Sep 2019.

[13] C. Luo, Y. He, and S. Abubakar, "Solving inverse kinematics of the shotcrete manipulator based on the plane two-link model and trajectory planning," *J. Robotics*, vol. 2020, pp. 8844979:1–8844979:12, 2020.

[14] A. Aristidou and J. Lasenby, "Fabrik: A fast, iterative solver for the inverse kinematics problem," *Graphical Models*, vol. 73, no. 5, pp. 243–260, 2011.