Book of Abstracts of MMT Symposium
Mechanism and Machine Theory Symposium
Flores et al. (Eds)

MMT Symposium
Celebrating 60 years since the journal's foundation
June 26-28, 2024 | Guimarães, Portugal

USING NEURAL NETWORKS FOR THE KINEMATICS OF CLOSED LOOP MECHANISMS AND SERIAL ROBOTS

Soumya Kanti Mahapatra[1], Naveen Kumar Maddu[2], Ashitava Ghosal[1]

[1] Dept. of Mechanical Engg., Indian Institute of Science, Bangalore, India, {soumyam@iisc.ac.in; asitava@iisc.ac.in}
[2] Dept. of Design and Manufacturing, Indian Institute of Science, Bangalore, India, {naveenmaddu@iisc.ac.in}

1 INTRODUCTION

In recent years, neural networks and data-driven AI/ML techniques have been extensively used in several areas of science and engineering. In this work, we attempt to use a neural network for solving the kinematics of serial robots and mechanisms. In serial robots, the forward map between the joint and Cartesian coordinates is known to be one-to-one while for the inverse kinematics, there can be more than one joint configuration for a given end-effector position and orientation. For a parallel manipulator or a closed-loop mechanism, both the forward and inverse kinematics are not one-to-one. While analytical or numerical methods are the foundation of traditional ways to solving kinematics, data-driven approaches present an alternative way of approaching this problem. In this work, we explore the use of feed-forward neural networks (FNN) for the inverse kinematics problems for serial robots and for the forward and inverse problems of parallel manipulators and closed-loop mechanisms, offering an examination of their potential uses and drawbacks. To illustrate the use of FNN's, we use the basic 2R planar robot, a 6R spatial PUMA 560 manipulator, and a planar four-bar mechanism. Some of the key findings reported in this paper are that a careful use of the loss function will lead to a single solution and decreasing errors with epochs can be better explained if the analytical kinematics equations or the geometry is used.

2 METHODOLOGY

2.1 Kinematics

As mentioned, the inverse problem of serial robots and the forward and inverse problems of parallel manipulators and closed-loop mechanisms give rise to multiple solutions. For example, in the case of the well-known planar 2R robot, the spatial PUMA 560 robot, and the planar four-bar mechanism, the kinematics equations are given by [1]

$$\text{2R} - \theta_2 \text{ eqn.: } x^2 + y^2 = l_1^2 + l_2^2 + 2\,l_1\,l_2\cos\theta_2 \tag{1}$$

$$\text{PUMA 560} - \theta_1 \text{ eqn.: } -x\sin\theta_1 + y\cos\theta_1 = d_3 \quad \Rightarrow \sqrt{x^2+y^2}\sin\left(\theta_1 - \text{Atan2}\,(y,x)\right) = -d_3 \tag{2}$$

$$\text{Four-bar} - \text{Freudenstein's eqn.: } l_0^2 + l_1^2 + l_3^2 - l_2^2 - 2\,l_0\,l_1\cos\theta_1 = 2\,l_3\cos\phi_1\,(l_1\cos\theta_1 - l_0) + 2\,l_1\,l_3\sin\theta_1\sin\phi_1 \tag{3}$$

where for the 2R robot, l_1 and l_2 are the links lengths, d_3 is a link-offset for PUMA 560, and $l_i,\ i = 0,1,2,3$ are the link lengths of the four-bar mechanism. It can be clearly seen that two values θ_2 $(= \pm\cos^{-1}((x^2+y^2-{l_1}^2-{l_2}^2)/(2l_1l_2)))$ can be obtained for a given (x,y) coordinates of the end-effector in the case of the planar 2R robot. Likewise in the case of the PUMA 560, for a given (x,y) of the wrist, two values of the first joint variable θ_1 can be obtained and in the case of a four-bar, the input-output equation gives two values of ϕ_1 for a given θ_1 and vice-versa.

2.2 Neural Network

As shown in [2], FNNs are used extensively in many problems in science and engineering. Due to the one-to-many solution natures of the kinematic solutions, simple loss functions like Mean Square Error (MSE) [3] for the outputs do not provide sufficiently accurate results for basic FNNs with no joint limits. Hence, we use the reconstruction loss [4] given by:

$$\mathcal{L} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\left\|f^{-1}\left(\mathcal{G}\left(X_i\right)\right) - X_i\right\|^2 + \lambda\left\|\mathcal{G}\left(X_i\right) - Y_i\right\|^2\right) \qquad \text{with } \lambda \geq 0 \tag{4}$$

where X is the input to the neural network, Y is the output, $\mathcal{G}$ is the function representing the neural network, f^{-1} is the inverse to the function we want to approximate using the neural network. For serial manipulator f^{-1} is the forward kinematics equation. For the first few epochs, λ is assigned some value and then set to zero afterwards. In equation (4), the first term is the *reconstruction* part and the second term is the *prediction* part. The input X and the output Y are the data used for training the neural network. They can be obtained from experiments and measurements – in this work, we generate data from the kinematics equations presented in [1]. The FNN uses softplus activation functions with hidden layer dimensions of (100, 100, 100) for 2R and PUMA 560.

3 RESULTS AND DISCUSSION

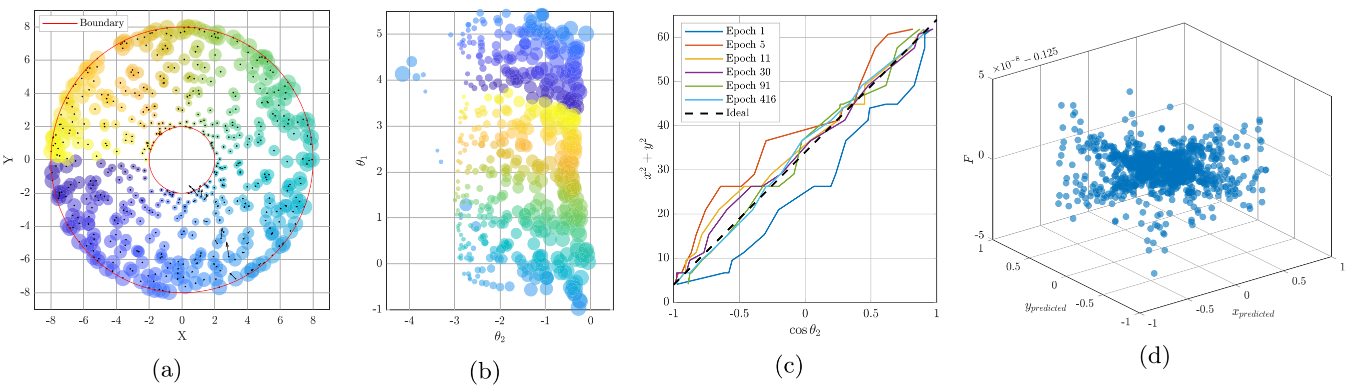

Figure 1: 2R manipulator results (a) showing generate plots (black arrows show the position where the point moved after prediction by the FNN), (b) scatter plot of the joint space mapping of the input points (same colour and size), (c) plot for the equation (1) as the training progresses. (d) Scatter plot for the equation (2) with $F =$ LHS of (2).

2R: The training data is generated using the forward kinematics equations with $\theta_1, \theta_2 \in [-\pi, \pi]$, $l_1 = 5$ and $l_2 = 3$. For f^{-1}, we use the forward kinematics equations. Some of the main observations:

- The predicted θ_2 gets restricted to either $[0, \pi]$ or $[-\pi, 0]$ (see Fig. 1b) – single solution output.
- The coefficient of determination (R^2) for x is 0.9942 and for y is 0.9811 with RMS error of 0.6495.
- With the trained model, using the test data points, if we plot $\cos\theta_2$ vs $(x^2 + y^2)$, the points converge to a straight line with a slope of $2l_1l_2$ and intercept intercept of $l_1^2 + l_2^2$. This can be used to obtain the link lengths of the robot without knowing it a *priori.*
- With increasing epochs, the FNN model approximates eqn (2) as expected and seen in Fig. 1c.

PUMA 560: For the data generation, we use $\theta_1, \theta_2, \theta_3 \in [-\pi, \pi]$ and the DH parameters parameter values given by, $a_2 = 0.4318$, $a_3 = 0.019$, $d_3 = 0.125$, and $d_4 = 0.432$. Some of the observations:

- Similar to 2R, the solution (joint) space for θ_1 is restricted to one solution, unlike two in the case of the analytical method.
- The R^2 value for x is 0.9757, for y is 0.9898 and for z is 0.9990 with RMS error of 0.0614.
- A plot of x, y and LHS of eqn (2), shows all the points will be concentrated on the plane $Z = -d_3$ (see Fig. 1d).

Four-bar: In this case we choose $l_0 = 1$, $l_2 = 3$, $l_3 = 4$, and $l_0 = 5$. For generating point, we choose $\theta_1 \in [-\pi, \pi]$ and solve eqn (3) to obtain ϕ_1. Here, f^{-1} can be found by training a separate FNN with MSE loss function. In this case, the solution converges to Freudenstein's equation. More simulations are being done to get a better understanding of what the neural network is doing.

4 CONCLUSION

In this study, we explore the use of neural networks and data-driven techniques for the kinematics of serial robots and closed-loop mechanisms. It was observed that a simplistic and traditional FNN with a MSE loss function does not yield suitable results due to the presence of multiple solutions. Using reconstruction loss solves the issue of possible multivalued inputs/outputs. A better understanding of the decreasing error can be reached by considering the analytical kinematics equations or the geometry – it is shown that with increasing epochs, the output of the neural network converges to the analytical kinematics equations thereby confirming the validity and accuracy of the data-driven approach. The results in this work indicate that the data driven approach perhaps can work for cases where the analytical solutions are very difficult to obtain or not known. In the future, we plan to explore different neural networks and use the neighborhood information with feedback to more quickly and accurately solve the kinematics problems and extend these approaches for path planning.

REFERENCES

[1] A. Ghosal, *Robotics: Fundamental Concepts and Analysis.* Oxford University Press, 2009.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[3] M. Esmat, M. Abdel-Nasser, and A.-W. A. Saif, "Solving inverse kinematics for 3r manipulator using artificial neural networks," in *20th International Multi-Conference on Systems, Signals & Devices*, 2023, pp. 116–120.

[4] S. Kumar, S. Tan, L. Zheng, and D. M. Kochmann, "Inverse-designed spinodoid metamaterials," *npj Computational Materials*, vol. 6, no. 1, p. 73, jun 2020.