Kinematic Estimation with Neural Networks for Robotic Manipulators

Michail Theofanidis(✉), Saif Iftekar Sayed(✉), Joe Cloud, James Brady(✉), and Fillia Makedon(✉)

HERACLEIA Human-Centered Computing Laboratory,
Department of Computer Science and Engineering,
University of Texas at Arlington, Arlington, USA
{michail.theofanidis,saififtekar.sayed,joe.cloud,
james.brady2}@mavs.uta.edu, makedon@uta.edu

Abstract. In this paper, we focus on estimating the forward kinematic equation of robots with multilayer feed-forward neural networks. The effectiveness of this approach is tested on a simulated kinematic model of the 7-DOF Sawyer Robotic Arm. In the initial sections of the paper, we discuss related work that associates with the creation of model agnostic control schemes on a kinematic level. Moreover, we formalize the kinematic problem as a supervised problem and we propose an MLP architecture to solve the problem. Lastly, we present experimental results and discuss the potential and importance to create model agnostic control schemes with machine learning.

Keywords: Robot kinematics · Forward kinematics
Neural networks for engineering

1 Introduction

Kinematics is the branch of classical mechanics, which studies the motion of bodies, without consideration of acting forces or moments. Robot kinematics provide mathematical tools to model and analyze the motion and structure of robotic manipulators, which is a fundamental component of robot control. In general, robotic manipulators are composed by a series of links and joints, followed by a gripper (the end effector). The joints of a robot can be either rotational or prismatic and they can be controlled by a certain actuator, such as an electric motor. To move the robot's end effector along a particular trajectory, actuation must be caused by the motors of the joints. The equations that describe the relationship between the position of the end effector and the position of the joints are addressed as the kinematic equations of the robotic arm.

Specifically, the mapping from the joint space of the robot to the Cartesian space of the robot's end effector is known as forward kinematics, while the inverse of this relationship is addressed as the inverse kinematics. Traditionally, the kinematic equations of a robot are derived from the kinematic model of the

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11141, pp. 795–802, 2018.
https://doi.org/10.1007/978-3-030-01424-7_77

robot, which describes the spacial relationship of each link and joint of the robot. Spacial relationships can be decomposed into rotational and translational and they can be represented mathematically by homogeneous transformations matrices [3]. In this paper, we focus on estimating the forward kinematic equations of robots with neural networks.

2 Related Work

A considerable amount of research has been conducted in the fields of both machine learning and control theory to try and create reliable control algorithms, that enable robotic arms to perform tasks autonomously and adapt to new environments [1]. Since robotic systems can be abstracted as continuous time systems that moves along a trajectory given a particular control input in the joint domain, it is worthwhile to investigate control frameworks based on neural networks that have the capability to solve nonlinear problems. According to the relevant literature, two different network architectures have been employed successfully to solve control problems in robotics [5]. Feed forward neural networks and recurrent neural networks.

The architecture of the neural network is based on whereas the system has full knowledge, partial knowledge or no knowledge of the robot's plant dynamics [9]. When the system has full or partial knowledge of the dynamics, feed forward neural networks have been used to compensate uncertainties due to modeling or sensor error [6]. In the case of model-free control of robotic systems, neural networks are used as function approximators that estimate the kinematic and dynamic equations of the robot. Note though, that both the forward and inverse kinematic and dynamic equations of robotic arms can not be fully learned by a single feed forward neural network, but they can be partially learned with recurrent neural networks [7].

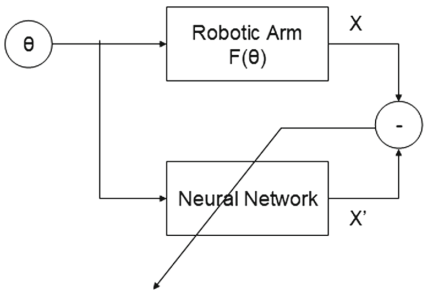

Fig. 1. Learning the forward kinematics with supervised learning.

3 Problem Formulation

As previously explained, the forward kinematics is a function F that connects the vector of joint positions θ with the Cartesian coordinates of the robot's end effector X:

$$X = F(\theta) \tag{1}$$

A very important property of Eq. 1 is that it is a one-to-one function, regardless of the geometrical properties of the robotic arm [7]. This statement holds true for every possible open loop kinematic chain and thus, every possible joint configuration can be uniquely mapped to one and only one end effector Cartesian coordinate [10]. Practically, this means that F can be learned in a supervised manner by a neural network as Fig. 1 suggests.

In addition, Fig. 1 indirectly suggests that the forward kinematics problem is independent of the robot geometry. That is not the case with the inverse kinematics problem, whose goal is to find a set of joint configurations given a particular end-effector position and orientation [3]. The difficulty of the inverse kinematics problem arises from its dependence on the physical configuration of the robot and that is has multiple solutions. Thus, any machine learning algorithm that tries to learn the inverse kinematics problem, will only be able to find one solution per kinematic configuration [4,8,11]. Also, the leaner might learn different inverse kinematics solutions for different kinematic configurations within the same workspace of a particular robot [2].

4 Experimental Testbed

The fact that the forward kinematics problem can be solved with classical supervised learning algorithms, means that the training process can occur off-line with training samples that are collected from measurements. These training samples will constitute a dataset whose input is measured from the robot joint encoders, and the output is the equivalent Cartesian coordinates of the robot end effector. A problem with this approach is that the Cartesian coordinates must be obtained from an external sensor and most mechanical manipulators possess only internal sensors. However, if the geometric characteristics of the robot are known, then the training dataset can be also generated from a simulated kinematic model of the robot. In this section, we present how we derived the kinematics of the 7-DOF Sawyer Robotic arm, and how well a multilayer perceptron neural network can learn to estimate the equation.

4.1 Kinematics of the Sawyer Robot

Figure 2 illustrates the kinematic model of the Sawyer Robotic Arm. The model was constructed by reverse engineering the geometrical properties of the physical robot. According to the homogeneous transformation of the joint frames from Fig. 2, the DH Table 1 of the model was composed. Note though, that in the table we do include the elevation of the robot above the world frame, which

is estimated to be 0.3160 m. Based on the DH table, the homogeneous coordinate matrix of the frames can be derived according to matrix (2). Finally, we computed the forward kinematic equations of the robot according to Eq. (2).

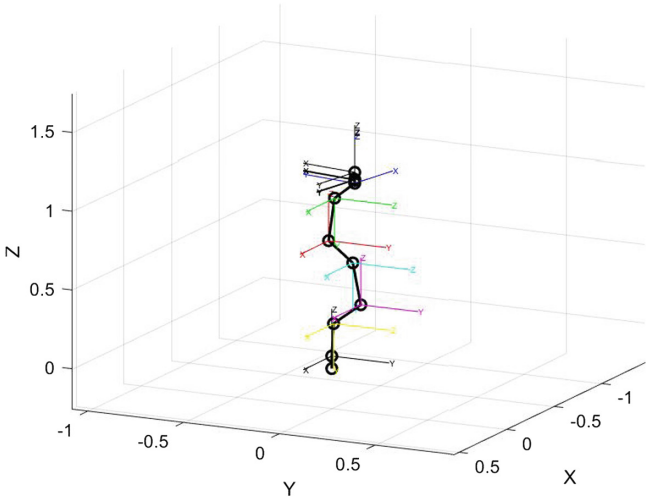

Fig. 2. Kinematic model of the sawyer robot.

Table 1. DH Table for the 7DOF sawyer robotic arm

i	α_i	a_i	d_i	θ_i
1	$-90°$	0.0810	0	θ_1
2	$90°$	0	0.1910	θ_2
3	$-90°$	0	0.3990	θ_3
4	$90°$	0	-0.1683	θ_4
5	$-90°$	0	0.3965	θ_5
6	$90°$	0	0.1360	θ_6
7	0	0	0.1785	θ_7

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

$$ {}^{1}_{7}T = {}^{1}_{2}T * {}^{2}_{3}T * {}^{3}_{4}T * {}^{4}_{5}T * {}^{5}_{6}T * {}^{6}_{7}T \tag{3} $$

$$\begin{cases} -175^\circ \leq \theta_1 \leq 175^\circ \\ -175^\circ \leq \theta_2 \leq 175^\circ \\ -175^\circ \leq \theta_3 \leq 175^\circ \\ -170^\circ \leq \theta_4 \leq 170^\circ \\ -170^\circ \leq \theta_5 \leq 170^\circ \\ -170^\circ \leq \theta_6 \leq 170^\circ \\ -180^\circ \leq \theta_7 \leq 180^\circ \end{cases} \tag{4}$$

4.2 Network Architecture

To solve the forward kinematics problem of Eq. 3 the multi-layered feed-forward neural network of Fig. 3 is proposed. The input layer of the network represents a vector of joint angle values (θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7), while the output of the network stands for the cartesian coordinates of the robot's end effector. Both the input and output units contain linear units for normalization purposes.

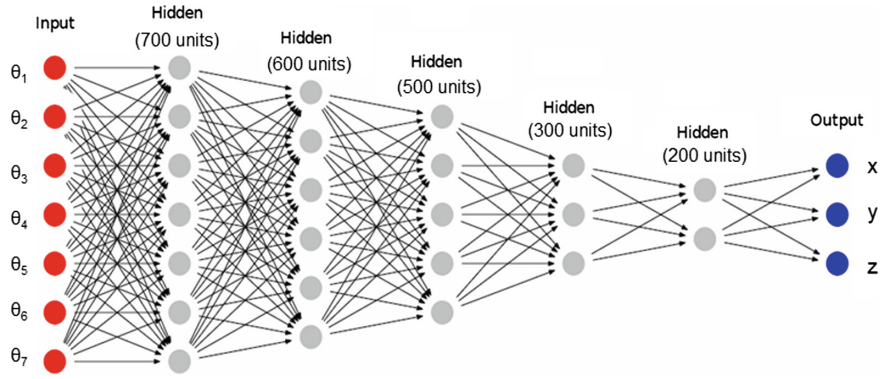

Fig. 3. Network architecture.

The network was trained using the backpropagation algorithm with the mean squared error of the output units as a metric. During the backpropagation process, we used adam optimizer. To produce the training dataset of the network, 4 million random kinematic configurations of joint angles with their equivalent Cartesian positions were utilized. During the creation of the dataset, we made sure that the joint angle values uniformly cover the ranges of Eq. 4. Because of the size of the dataset, the network was trained with a batch size of 100 units and 30 epochs. Also, 10% of the dataset was used for cross validation and 10% was used for testing purposes.

4.3 Experimental Results

After the training was complete, the networks achieved 99.997% validation accuracy. To demonstrate the effectiveness of the network, in this section we will compare the network estimations with the output of the forward equation as computed by Eq. 3 for the same input joint trajectory samples. Figure 4 shows the sample trajectory in joint space.

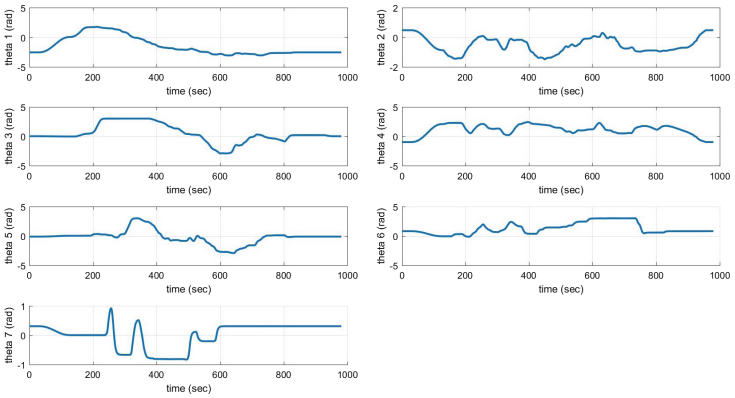

Fig. 4. Experimental joint space trajectories.

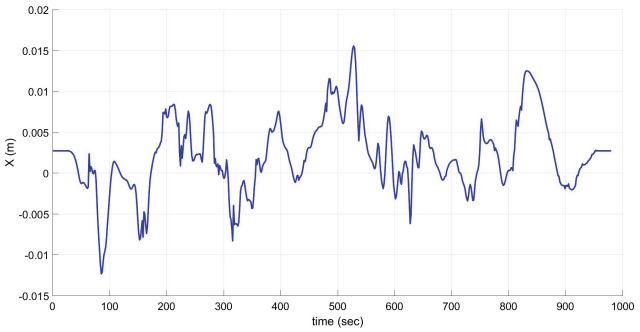

Fig. 5. Error between the forward kinematic equation and the network in the x dimension.

The difference between the estimations of the forward kinematic equations and the proposed network is shown in Figs. 5, 6 and 7, where every figure represents one of the cartesian dimensions of the robot's end effector. Note that the scale in the vertical axis is in meters.

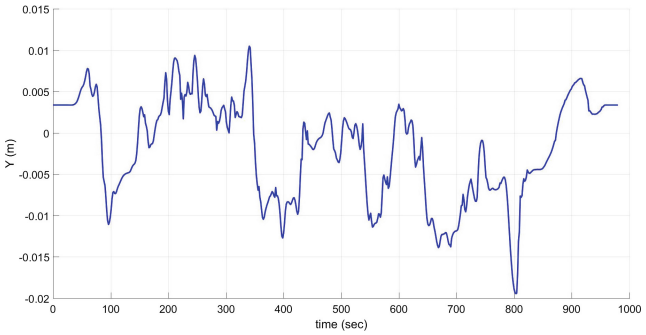

Fig. 6. Error between the forward kinematic equation and the network in the y dimension.

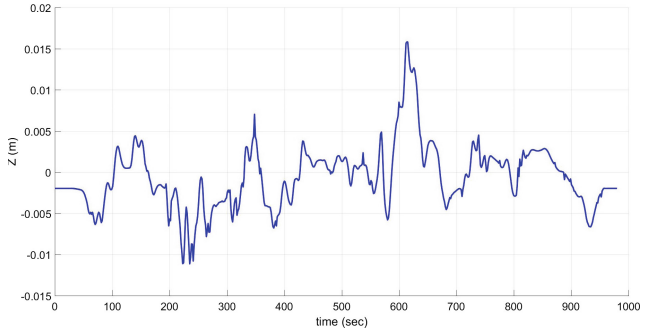

Fig. 7. Error between the forward kinematic equation and the network in the z dimension.

5 Conclusions

In this work, we presented how to estimate the forward kinematic equations of a kinematically redundant robotic arm with a neural network. The proposed network architecture showed promising results between different kinematic configurations. However, it is worthy to mention that although the forward kinematics equations can be estimated algebraically in a simple manner, learning the same equations is an arduous process for a neural network. The proposed architecture was found after training multiple models with different parameters, such as the number of units per level and the number of levels, on the same dataset with different resolution. That was possible to achieve, because the workspace of the robot can not possibly change.

Acknowledgments. This work is supported in part by the National Science Foundation under award numbers 1338118 and 1719031. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)
2. Bingul, Z., Ertunc, H., Oysu, C.: Comparison of inverse kinematics solutions using neural network for 6R robot manipulator with offset. In: 2005 ICSC Congress on Computational Intelligence Methods and Applications, p. 5. IEEE (2005)
3. Craig, J.J.: Introduction to Robotics: Mechanics and Control, vol. 3. Pearson/Prentice Hall, Upper Saddle River (2005)
4. Duka, A.V.: Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. Procedia Technol. **12**, 20–27 (2014)
5. Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. Neurocomputing **285**, 23–34 (2018)
6. Jin, L., Zhang, Y., Li, S.: Integration-enhanced zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. IEEE Trans. Neural Netw. Learn. Syst. **27**(12), 2615–2627 (2016)
7. Jordan, M.I., Rumelhart, D.E.: Forward models: supervised learning with a distal teacher. Cogn. Sci. **16**(3), 307–354 (1992)
8. Karlik, B., Aydin, S.: An improved approach to the solution of inverse kinematics problems for robot manipulators. Eng. Appl. Artif. Intell. **13**(2), 159–164 (2000)
9. Lin, D., Wang, X., Nian, F., Zhang, Y.: Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems. Neurocomputing **73**(16–18), 2873–2881 (2010)
10. Nguyen, L., Patel, R., Khorasani, K.: Neural network architectures for the forward kinematics problem in robotics. In: 1990 IJCNN International Joint Conference on Neural Networks, pp. 393–399. IEEE (1990)
11. Tejomurtula, S., Kak, S.: Inverse kinematics in robotics using neural networks. Inf. Sci. **116**(2–4), 147–164 (1999)