International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
3rd National Conference on Recent Trends & Innovations In Mechanical Engineering
15th & 16th March 2019
Available online at www.ijrat.org

Inverse Kinematics 2d Robot

P.Srinivas[1], G.Akhil[2], G.Naveen[3], Dileep[4], Nikhil[5], Pavan[6]
[1]*Associate Professor, Mechanical Engineering, NNRG, Hyderabad*
[2, 3,4,5,6] *Student, Mechanical Engineering, NNRG, Hyderabad*

Abstract: Robots have penetrated today in almost all industrial fields, being much more precise than humans in the execution of operations, but also faster, more dynamic, more stable and more resilient, working 24 h of the 24 h possible. One of the most important problems in robot kinematics and control is, finding the solution of Inverse Kinematics. Inverse kinematics computation has been one of the main problems in robotics research. As the Complexity of robot increases, obtaining the inverse kinematics is difficult and computationally expensive. Traditional methods such as geometric, iterative and algebraic are inadequate if the joint structure of the manipulator is more complex. As alternative approaches, neural networks and optimal search methods have been widely used for inverse kinematics modeling and control in robotics This paper proposes neural network architecture that consists of 6 sub-neural networks to solve the inverse kinematics problem for robotics manipulators with 2 or higher degrees of freedom. The neural networks utilized are multi-layered perceptron (MLP) with a back-propagation training algorithm. This approach will reduce the complexity of the algorithm and calculation (matrix inversion) faced when using the Inverse Geometric Models implementation (IGM) in robotics. The obtained results are presented and analyzed in order to prove the efficiency of the proposed approach.

1. INTRODUCTION

In inverse kinematics learning, the complexity is in the geometric and non linear equations (trigonometric equa-tions) and in the matrix inversion, this in addition to some other difficulties faced in inverse kinematics like having multiple solutions. The traditional mathematical solutions for inverse kinematics problem, such as geometric, itera-tive and algebraic, may not lead always to physical solu-tions. When the number of manipulator degrees of free-dom increases, and structural flexibility is included, ana-lytical modeling becomes almost impossible. A modular neural network architecture was proposed by Jacobs *et al*. and has been used by many researches [2,3,5,6].

However, the input-output relation of their networks is continuous and the learning method of them is not suffi-cient for the non-linearity of the kinematics system of the robot arm.

This paper proposes neural network architecture for inverse kinematics learning. The proposed approach con-sists of 6 sub-neural networks. The neural networks util-ized are multi-layered perceptron (MLP) with a back- propagation training algorithm. They are trained with end-effector position and joint angles.

In the sections that follow, we explain the inverse kinematics problem, and then we propose our neural network approach; we present and analyze the results in order to prove that neural networks provide a simple and effective way to both model the manipulator inverse kinematics and circumvent the problems associated with algorithmic solution methods.

The proposed approach is presented as a strategy that could be reused and implemented to solve the inverse kinematics problems faced in robotics with highest de-grees of freedom. The basics of this strategy are explained in details in the sections that follow.

Inverse Kinematics

Inverse kinematics computation has been one of the main problems in robotics research. This problem is generally more complex for robotics manipulators that are redun-dant or with high degrees of freedom. Robot kinematics is the study of the motion (kinematics) of robots. They are mainly of the following two types: *forward kinematics* and *inverse kinematics*. Forward kinematics is also known as direct kinematics. In forward kinematics, the length of each link and the angle of each joint are given and we have to calculate the position of any point in the work volume of the robot. In inverse kinematics, the length of each link and position of the point in work volume is given and we have to calculate the angle of each join. As in fig 1.1, fig 1.2

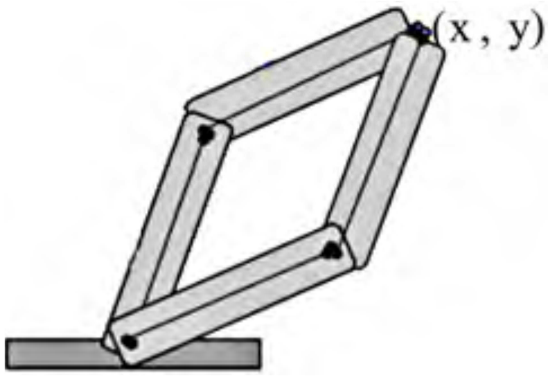

Fig 2.2

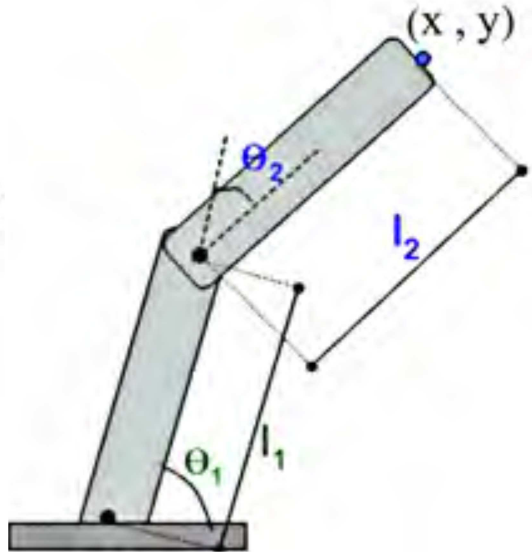

Fig 1.1

$$x = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2)$$
$$y = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2)$$
$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 cos(\theta_2)$$

$$cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

$$x = l_1 cos(\theta_1) + l_2(cos(\theta_1)cos(\theta_2) - sin(\theta_1)sin(\theta_2))$$
$$x = cos(\theta_1)(l_1 + l_2 cos(\theta_2)) - sin(\theta_1)(l_2 sin(\theta_2))$$
$$y = cos(\theta_1)(l_2 sin(\theta_2)) + sin(\theta_1)(l_1 + l_2 cos(\theta_2))$$
$$cos(\theta_1) = \frac{x + sin(\theta_1) l_2 sin(\theta_2)}{l_1 + l_2 cos(\theta_2)}$$

$$sin(\theta_1) = \frac{(l_1 + l_2 cos(\theta_2))y - l_2 sin(\theta_2)x}{l_1^2 + l_2^2 + 2 l_1 l_2 cos(\theta_2)}$$

Neural Network Approach

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear rela-tionships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. The most common neural network model is the multilayer percep-tron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to pro-duce the output when the desired output is unknown. A graphical representation of an MLP is shown in Fig 2.1

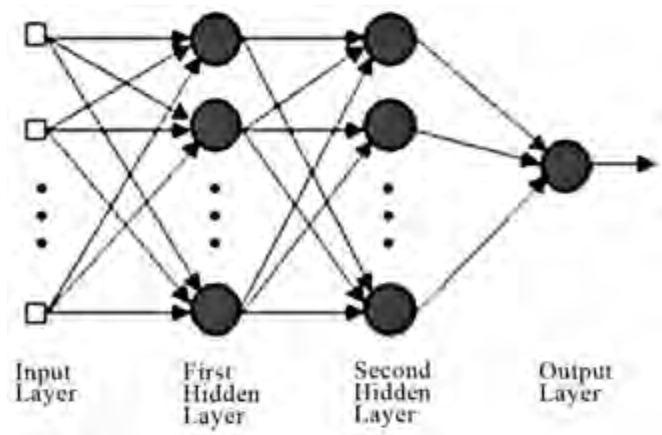

Fig 2.1

2. PYTHON PROGRAM FOR INVERSE KINEMATICS

```
a1 = 6.2  # length of link a1 in cm
a2 = 5.2  # length of link a2 in cm
a3 = 0  # length of link a3 in cm
a4 = 7  # length of link a4 in cm

# Desired Position of End effector
x = -7
y = 9

# Equations for Inverse kinematics
r1 = sqrt(x**2+y**2)  # eqn 1
phi_1 = arccos((a4**2-a2**2-r1**2)/(-2*a2*r1))  # eqn 2
phi_2 = arctan2(y, x)  # eqn 3
theta_1 = rad2deg(phi_2-phi_1)  # eqn 4 converted to degrees

phi_3 = arccos((r1**2-a2**2-a4**2)/(-2*a2*a4))
theta_2 = 180-rad2deg(phi_3)

print('theta one: ', theta_1)
print('theta two: ', theta_2)
```

3. CONCLUSION

The most commonly used anthropomorphic robotic mechatronic systems, which are currently being used,have been studied by eliminating the heavy, matrix 3D
spatial system, the study being simplified in a plan by considering the main work plan of the system and the plan, the rotation required to restore the spatial parameters of the anthropomorphic 3D system. In other words, we can greatly ease the work of the anthropomorphic robot engineer by moving from 3D systems to a 2D system. In this study we will study the inverse kinematics of the plan system,

REFERENCES:

[1] Al-Alaoui M.A., et al. A Cloning Approach to Classsifier Training, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 2002; 32(6).

[2] Alba E, Chicano JF. Training Neural Networks with GA Hybrid Algorithms, Lecture Notes in Computer Science Vol. 3102 / 2004, Springer Berlin / Heidelberg, 2004: 852-863.

[3] Aristidou A, Lasenby J. Inverse Kinematics: a review of existing techniques and introduction of a new iterative fast solver. Cambridge University Engineering Department, Technical Report, 2009.

[4] Asada HH. Introduction to Robotics. Department of Mechanical Engineering, Massachusetts Institute of Technology.

[5] Bassam Daya, Shadi Khawandi, Mohamed Akoum. Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics. J. Software Engineering & Applications, 2010; (3): 230-239

[6] Bingul Z, Ertunc HM, Oysu C. Applying Neural Network to Inverse Kinematic Problem for 6R Robot Manipulator with Offset Wrist. Adaptive and Natural Computing Algorithms, 2005: 112-115.

[7] Courty N, Arnaud E. Inverse Kinematics Using Sequential Monte Carlo Methods. Proceedings of the 5th international conference on Articulated Motion and Deformable Objects, 2008: 1-10

[8] Demuth H, Beale M, Hagan M. Neural Network Toolbox 6. User's Guide", The Mathworks, Inc., 2008.

[9] Finschi L. An Implementation of the Levenberg-Marquardt Algorithm", Eidgenossische Technische Hochschule Zurich, 1996.