Kinematic Analysis Using Multilayer Perceptrons for 3-DOF Manipulator Control System

* Won-Moook Jeong
한국기술교육대학교 메카트로닉스공학부
e-mail : *jwonmook@gmail.com*

Abstract

In this paper, we tested a kinematic model of a 3-DOF robotic arm using a multilayer perceptron neural network. In the introduction of the thesis, the reason for the need for manipulator (robot arm) control is explained, and in the main text, the technology used in the study is explained. Finally, along with the experimental results, the direction and importance of follow-up studies are presented.

I. Introduction

Recently, with the development of artificial intelligence field, research on the analysis of classical mechanical field using it is increasing. As a result, industries such as remote work and factory automation are developing. It is a situation in which kinematic analysis is essential.

In this study, the kinematic analysis of a robot arm composed of three joints and having three degrees of freedom was studied to predict the position using a multilayer perceptron.

II. Main subject

2.1 MLP(Multi-Layer Perceptron)

We used a multilayer perceptron (MLP) for kinematic analysis. Kinematic Modeling was constructed using three hidden layers using only data without applying the classical analysis technique, Kinematic System. All three hidden layers used Relu (Rectified Linear Unit) as an activation function, and the input value of the entire learning model was composed of the angle of each joint, and the output value was composed of the position coordinates (x,y) of the joint end.

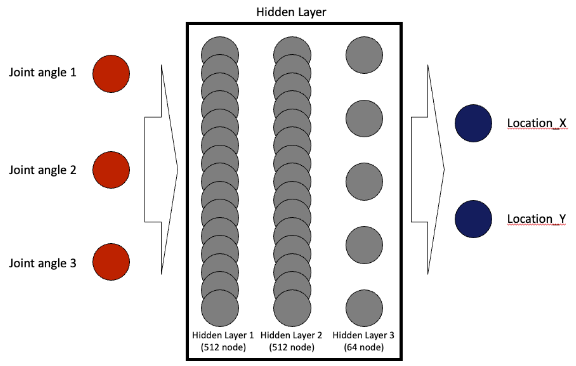

Figure 1. Machine Learning Model Structure

2.2 Kinematics

In order to obtain the ideal data, the training data was generated using the existing kinematic analysis model.

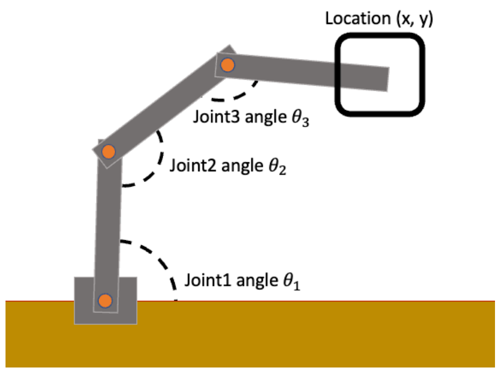

Figure 2. Machine Learning Model Structure

$$x = l_1 cos\,\theta_1 + l_2 cos(\theta_1 + \theta_2) + l_2 cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = l_1 sin\,\theta_1 + l_2 sin(\theta_1 + \theta_2) + l_3\, sin(\theta_1 + \theta_2 + \theta_3)$$

Figure 3. Location Coordinate Calculation Equation

Since the joint consists of 3 links and 3 joints, the coordinates of the position of the end point are calculated by the formula in Figure 3. In this way, about 5000 training data / control group data were generated and the study was conducted.

Ⅲ. Implement

The library used for implementation is Pytorch 1.9.0, written in Python3.8. The length of all links is the same as 1m, the range of joint angle is 0 ~ 360°, and the learning epoch was conducted 50 times. MSE was used as the Loss function.

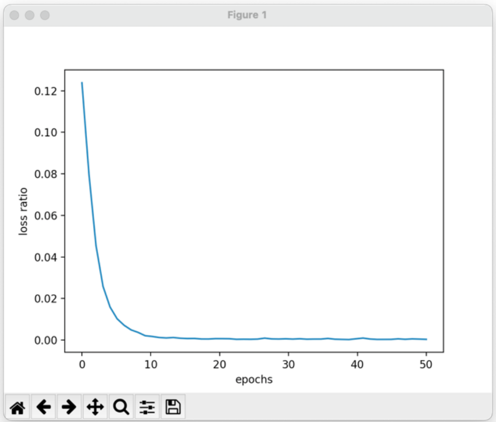

Figure 4. Loss Descent in the Learning Process

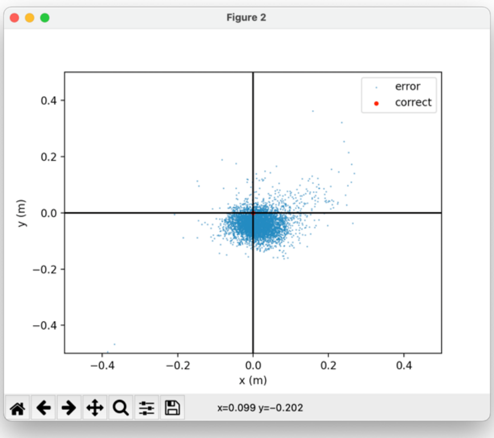

Figure 5. Error Distribution

As a result of the experiment after actual learning, the error was distributed as shown in Figure 5, showing an average error of about 1.17%. In order to check whether the model is suitable, a comparative experiment was conducted with data not used during data training.

Ⅳ. Conclusion

As the number of 3D models and links increases, the difficulty of kinematic analysis increases exponentially. If these difficulties are solved using artificial intelligence, even a complex model can be analyzed relatively simply, and if artificial intelligence is used in various kinematics fields such as inverse kinematics analysis in the future, it is expected that efficient research will proceed.

References

[1] T. Ogawa and H. Kanada, "Solution for Ill-posed inverse kinematics of robot arm by network inversion," Journal of Robotics, vol. 2010, Article ID 870923, 9 pages, 2010.