Presented at the Research Congress 2013
De La Salle University Manila
March 7-9, 2013

ROBOT CONTROL SYSTEM USING ELMAN NEURAL NETWORK INVERSE KINEMATICS SOLUTION

Tumlos, Leo Niño Butch O. & Musngi, Marlon M.
MEM Department, Gokongwei College of Engineering, De La Salle University Manila

Abstract: Model-based control is now a significant technology for the control of robots. Models and control schemes are continuously refined to meet the requirements of higher performance and lower cost. The control strategies used in most robots involve position coordination in the Cartesian space through the inverse kinematics method. Inverse kinematics comprises the computations to determine the joint angles needed to achieve the position and orientation for the robot end-effector. The inverse kinematics problem is usually complex for robotic manipulators. There are three traditional methods used for solving inverse kinematics problems: geometric, algebraic and iterative. Computing for the inverse kinematics solution using these traditional methods is a time -consuming study, especially when the joint structure of the manipulator is more complex. The computation of inverse kinematics using artificial neural networks is particularly useful where less computation time is needed, such as in real time adaptive robot control. Traditional methods will become prohibitive due to the high complexity of the mathematical structure of the formulation, wherein robots have to work in the real world that cannot be modeled concisely using mathematical expressions. A neural network-based inverse kinematics solution methods yield multiple and precise solutions with an acceptable error and it is suitable for real-time adaptive control of robotic manipulators. This research focuses on the design and development a control system for a robot arm, using the Elman neural network based inverse kinematics solution approach. From the recurrent networks family, Elman Network is selected because of its feedback loops that have a weighty impact on the learning capability and performance of the network. This research integrates the mechanical and electronic systems design, with the model-based control algorithm, to establish an integral robot control system. Experiments were done using a 5 degree of freedom revolute Teachmover robot. Key points within the robot's workspace were identified as testing points and both simulations and actual movements were done to verify results. These results show an average of 5.31% average distance error.

Key Words: Robotics, Inverse Kinematics, Elman Network

1. INTRODUCTION

Even though many academic researches have been made on the aspects of model-based robot control, a lot of applied research is still needed to assess the effectiveness of the algorithm in real robot applications. Applied research integrates the mechanical and electronic systems design, with the developed model-based control algorithm, to establish an integral robot control system. It also incorporates the testing and simulation on an actual robot. Through applied research, it will allow the robot manufacturers to continuously improve robot performance in terms of accuracy, reliability, and productivity, and lower the cost of industrial robot automation.

The study involves designing and developing a control system for a 5 DOF revolute robot arm, using a neural network based inverse kinematics solution approach. The Elman Network was chosen because of its feedback loops, which have a substantial impact on the learning capability and performance of a network.

2. THEORETICAL & CONCEPTUAL FRAMEWORK

The notional structure for the study comprises of two key areas: Model-Based Robot Control, and the 5 DOF revolute robot arm features and specification. Model-based robot control includes the inverse kinematics solution methods, the neural network strategy and approaches to inverse kinematics, the Backpropagation algorithm, and the Elman neural network strategy. The features and specification for the 5 DOF robot arm includes the physical structure, the actuators, and the microcontroller used. The features and specification of the TeachMover robot arm was used for data collection.

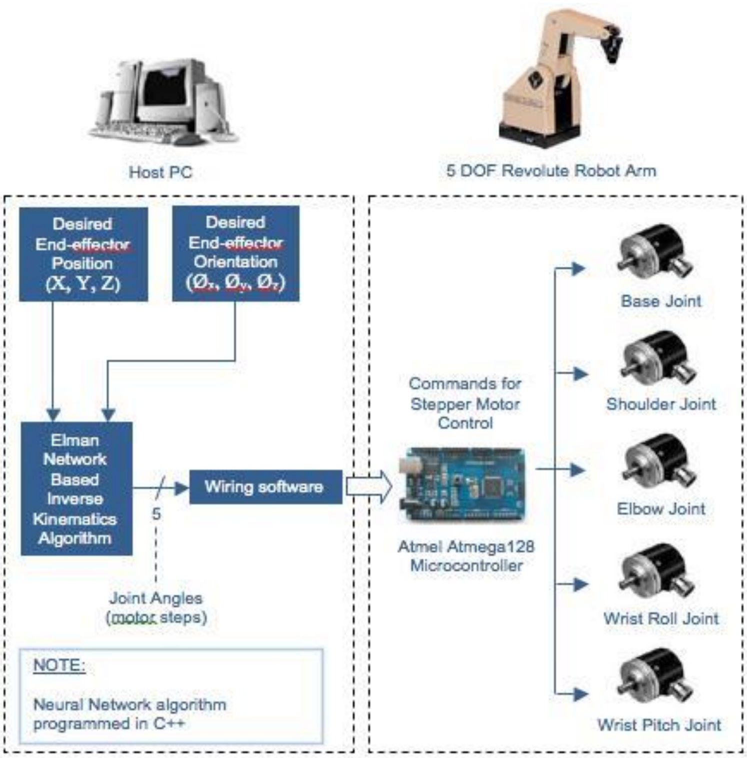

Figure 1. Conceptual Framework – Development of a Control System for a 5 DOF Robot Arm

Using Elman Neural Network Inverse Kinematics Solution Approach.

3. METHODOLOGY

The structure of the feedback Elman network used in inverse kinematics problem solution is given in Fig.2. This network has three layers which are input, hidden and output (Elman, 1991, and Rodriguez et al., 1999). In the hidden layer and output layer nodes, the sigmoidal activation function, which is a nonlinear function, has been used as denoted in Fig.2.

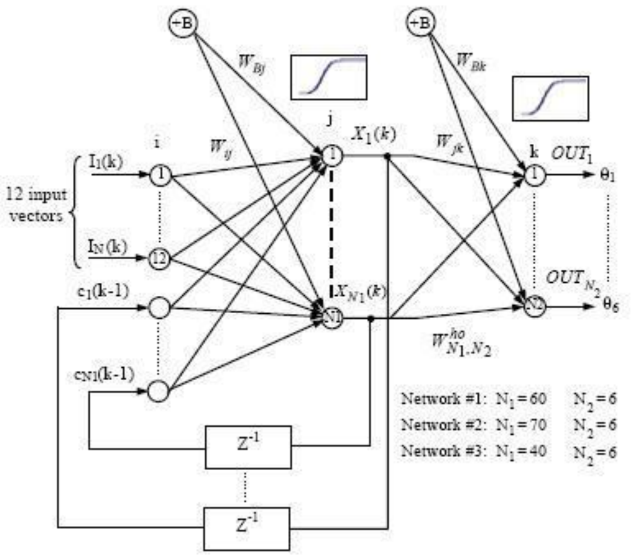

Figure 2. Structure of the Elman network used in inverse kinematics problem solution (Adopted from Köker, R., 2005).

The weights of the recurrent connections from hidden layer to context layer are fixed as ‘1’, namely, they do not have any effect as multiplier. The forward weights get trained using the back propagation algorithm. The context units act like input units in the forward phase. The computations of the output of hidden units and output units are done in the same way as is calculated for feed forward networks. After the computation of the outputs of the hidden units, these current values are given into the corresponding context units via the recurrent connections through a unit delay. These values are used in the next time step. At the first time step they have to be set to some initial values. During the backward phase of training, initial values for the outputs are used and the forward weights are adjusted by back propagation (Cheng et al., 2002).

The expressions for the outputs of each hidden layer neurons are given in the formula below:

( ) = (! (( )! Σ ( )! ( ) Σ ! ! ( ) (!)) **(Eq. 1)**

Elman network output can be given as:

( ) = (!! (( ) Σ ! ( ) ( ) ( ))) **(Eq. 2)**

where $I_i(k)$ is the ith input to the network (external inputs), $X_i(k-1)$ is recurrent input to the network, $W^{ih}_{1,0}$ is the weight value from the first neuron of input layer to the first neuron of hidden layer, $X_i(k)$ is the ith recurrent input, $W^{ho}_{1,0}$ is the weight value from the first neuron of input layer to the first neuron of hidden layer, $OUT_k(n)$ is the nth output (Temurtas et al., 2004).

Three feed forward Elman neural networks with sigmoid activation function are used to solve inverse kinematics problem based on reliability. The block diagram of inverse kinematics solution with unique Elman network is given in Fig.2-7. Each network has been obtained with the possibly least error.

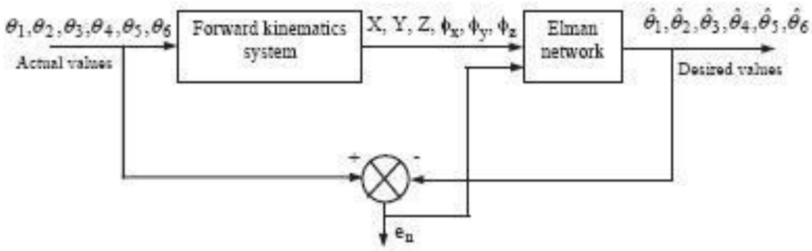

Figure 3. Neural network-based inverse kinematics solution with unique Elman network (Adopted from Köker, R., 2005).

Firstly, three different sets of ϑ1,ϑ2,ϑ3,4,ϑϑ5,000ϑ6) data, which consist of the (joint angles according to the different (X, Y, Z, ox, oy, oz, nx, ny, nz, ax, ay, az) cartesian coordinate parameters, were generated separately using in the work volume of a robotic manipulator. Attempts have been made to obtain well structured learning sets to make the learning process successful and easy. These values were recorded in files to form the learning sets of the networks. Each 3000 of these data was used in the training of Elman networks, and the remaining 1000 were used in the testing of each network one by one. During the training, the unlearned data of the first two networks have been selected and included in the training set of the third network. The training process is not completed until the error is acceptable for each network. To understand whether the error is acceptable or not, information about distance that refers to difference between the end effector and the target point is used. This distance, also known as an error, can be obtained easily in metric form by using three-dimensional (3-D) distance equation between two points in 3-D space. Euclidian distance equation given below can be used in the distance computation. As a result, it can be decided whether the error is acceptable or not depending on the obtained distance. Here, the sensitivity of the application is also important in the decision of an acceptable error.Figures and tables should be referred to in the text. They should be centered as shown below and must be of good resolution. Where equations are used, adequate definition of variables and parameters must be given, as shown in the example below.

$$E = [(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2]^{1/2} \qquad \textbf{(Eq. 3)}$$

All of the obtained results from Elman networks are tested using Euclidian distance equation given above. As stated above, this is an important point for the selection of the best result of neural networks. Using direct kinematics equations, each Elman network's results can be tested to select the network with minimum distance 'd' from target.

4. RESULTS AND DISCUSSION

To simulate the effectiveness of the control system developed for the TeachMover robot arm, including the weights generated from the previous program, 40 points from its workspace are tested for position accuracy in both graphical and actual simulations. The actual position values are manually measured from the tip of the gripper of the TeachMover (when closed) referenced to a scaled grid line.

Table 1. TeachMover Graphical and Actual Simulation Results.

Sample	Target Values			Graphical Simulation Values			% Error	Actual Simulation Values (Measured)			% Error
	X	Y	Z	X	Y	Z		X	Y	Z	
1	-4	14.5	12	-3.75	14.46	11.98	**0.31**	-4.25	14.25	12.38	**33.67**
2	-4	14.5	10	-3.95	14.36	9.78	**2.78**	-4.50	14.38	9.88	**13.47**
3	-4	14.5	8	-3.98	14.32	8.01	**0.08**	-4.50	14.25	7.88	**33.67**
4	-4	14.5	6	-3.84	14.34	5.97	**2.34**	-4.13	14.25	5.88	**11.22**
5	-2	14.5	12	-1.93	14.81	12.00	**0.00**	-2.25	14.38	12.38	**33.67**
6	-2	14.5	10	-2.05	14.75	9.77	**9.73**	-2.50	14.50	10.13	**0.00**
7	-2	14.5	8	-2.05	14.72	8.01	**0.46**	-2.50	14.63	8.00	**0.00**
8	-2	14.5	6	-1.97	14.71	5.96	**1.46**	-2.25	14.63	6.00	**0.00**
9	0	14.5	12	0.02	14.94	12.01	--	-0.13	14.75	12.63	--
10	0	14.5	10	0.01	14.89	9.76	--	-0.25	14.75	10.00	--
11	0	14.5	8	0.00	14.86	7.99	--	0.00	14.75	7.88	--
12	0	14.5	6	-0.02	14.83	5.94	--	0.00	14.75	5.50	--
13	2	14.5	12	2.04	14.80	12.00	**0.00**	2.13	14.63	12.50	**22.45**
14	2	14.5	10	2.07	14.75	9.75	**14.70**	2.38	14.63	10.00	**0.00**
15	2	14.5	8	2.04	14.72	7.99	**0.37**	2.38	14.75	8.00	**0.00**
16	2	14.5	6	1.90	14.71	5.94	**7.59**	2.13	14.63	5.75	**22.45**
17	4	14.5	12	4.01	14.39	11.99	**0.02**	4.25	14.50	12.63	**0.00**
18	4	14.5	10	4.06	14.33	9.74	**4.68**	4.25	14.63	10.13	**6.73**
19	4	14.5	8	3.97	14.32	7.99	**0.12**	4.50	14.50	8.00	**0.00**
20	4	14.5	6	3.71	14.37	5.95	**5.94**	4.38	14.50	5.75	**0.00**
21	-4	15	12	-3.7	14.48	11.99	**2.43**	-4.25	14.50	12.13	**21.70**
22	-4	15	10	-3.92	14.37	9.78	**20.13**	-4.13	14.50	9.88	**13.02**
23	-4	15	8	-3.96	14.33	8.02	**1.18**	-4.13	14.38	8.13	**20.35**
24	-4	15	6	-3.83	14.35	5.96	**13.49**	-3.88	14.50	5.88	**21.70**
25	-2	15	12	-1.9	14.82	12.01	**0.53**	-2.13	14.88	12.25	**10.85**
26	-2	15	10	-2.03	14.76	9.77	**5.66**	-2.13	14.88	9.88	**6.51**
27	-2	15	8	-2.03	14.73	8.01	**0.34**	-2.13	14.88	8.13	**8.14**
28	-2	15	6	-1.95	14.71	5.95	**4.25**	-2.13	14.75	5.88	**21.70**
29	0	15	12	0.05	14.94	12.02	--	-0.25	14.88	12.25	--
30	0	15	10	0.01	14.89	9.76	--	-0.13	14.88	9.75	--
31	0	15	8	0.02	14.86	8	--	0.00	14.88	7.88	--
32	0	15	6	0.02	14.84	5.93	--	-0.13	14.88	14.88	--
33	2	15	12	2.06	14.8	12.02	**0.65**	2.25	15.00	12.25	**0.00**
34	2	15	10	2.08	14.75	9.75	**16.72**	2.13	14.63	9.88	**20.31**
35	2	15	8	2.07	14.72	7.99	**0.81**	2.13	14.88	7.75	**16.93**
36	2	15	6	1.96	14.71	5.93	**4.75**	2.13	14.75	5.88	**22.57**
37	4	15	12	4.03	14.39	12.01	**0.26**	4.13	14.50	12.25	**21.70**
38	4	15	10	4.07	14.33	9.74	**21.47**	4.13	14.50	9.75	**26.04**
39	4	15	8	4.01	14.31	7.99	**0.15**	3.88	14.38	7.88	**20.35**
40	4	15	6	3.78	14.35	5.94	**26.63**	3.88	14.50	5.88	**21.70**

Results have shown that the average % distance error for the graphical simulation is approx 5.31%, while for the actual simulation is approx 14 %. Main contributors for the increase in % error for the actual simulation are the inaccuracies in the step / degree specification of the robot mechanical gear ratios, and the parallax error when doing the position measurements manually.

5. CONCLUSIONS

Results have shown that the average % distance error for the graphical simulation is approx 5.31%, while for the actual simulation is approx 14 %. Main contributors for the increase in % error for the actual simulation are the inaccuracies in the step / degree specification of the robot mechanical gear ratios, and the parallax error when doing the position measurements manually.

6. REFERENCES

Cheng, Y.C., Qi, W.M., Cai, W.Y., “Dynamic Properties of Elman and Modified Elman Neural Network. Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4–5, pp. 637–640, 2002.

Elman, J.L., “Distributed Representations, Simple Recurrent Networks, and Grammatical Structure”, Machine Learning 7, 195–225, 1991.

Hasan, A.T., Hamouda, H.M.S., Ismail, N., Al-Assadi, H.M.A.A., “An Adaptive-Learning Algorithm to Solve the Inverse Kinematics Problem of a 6 D.O.F Serial Robot Manipulator”, Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University Putra Malaysia, 43400UPM Serdang, Selangor, Malaysia, 2006.

Köker, R., Öz, C., Çacar, T., Ekiz, H., “A Study of Neural Network Based Inverse Kinematics Solution for a Three-Joint Robot”, Department of Computer Engineering, Sakarya University, 54187 Sakarya, Turkey, 2003.

Köker, R., “Reliability-Based Approach to the Inverse Kinematics Solution of Robots using Elman’s Networks”, Department of Computer Engineering, Sakarya University, 54187 Sakarya, Turkey, 2005.

Martin, J.A., De Lope, J., Santos, M., “A Method to Learn the Inverse Kinematics of Multi-Link Robots by Evolving Neuro-Controllers”. Dep. De Sistemas Informáticos y Computación, Universidad Complutensede Madrid, Spain, 2009.