The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Inverse kinematics in Hyper-redundant robot using Adaptive Neural Network

Chatklaw Jareanpon

Polar Lab, Department of Computer Science, Faculty of Informatics
Mahasarakham University , Thailand
(Tel: 66-86-225-5574, Fax: 66-43-754-359)

chatklaw.j@msu.ac.th

Abstract: The hyper-redundant robot has more degrees-of-freedom. The most difficulty of the hyper-redundant is to finding the inverse kinematic problem. Most of usually used method is Neural Network. However, it is difficult to find the suitable structure and number of node. This paper shows the novel algorithm that can find the suitable structure and number of node depends on the problem. The performance of this algorithm will demonstrated in the computer simulation and compare with the Back-propagation with same structure. The algorithm shows the good performance to adapt the number of node with less error to solve the 8-20 serial link chain hyper-redundant robots.

Keywords: Inverse kinematics, Hyper-redundant robot, Neural Network.

1 INTRODUCTION

The hyper-redundant robot is the robot that has more than the minimum numbers of degrees-of-freedom are termed "many kinematically redundant". The hyper-redundant are used in operation to snakes, elephant trunks, and tentacles. There are a number of very important applications such as obstacle avoidance, manipulated task. However, the most difficulty in Hyper-redundant robot is controlling the inverse kinematics of its.

The different techniques used for solving inverse kinematics can be classified as algebraic that do not guarantee closed form solutions, geometric that usually used the curve and constraint, and iterative. The iterative methods converge to only a single solution. The most learning method that are usually using Neural networks. The Neural Netwoks usually used to solve the problem of inverse kinematics are Back-propagation and Kohonen network. However, the most question of using neural network is "How many are the best number of the nodes? "

In this paper, I proposed a new sequential learning algorithm, which is able to adapt the structure of the network. Using this algorithm, it is possible to find the suitable number of hidden node.

2 HYPER-REDUNDANT ROBOT

Highly redundant manipulators or *hyper degree of freedom (HDOF)* has more degrees of freedom (DOF). A HDOF manipulator can perform manipulation tasks, such as moving in non-convenient environments, and pushing and caging a various sizes and shapes of objects. Due to all-in-one arms, a HDOF manipulator significant enhances the caging method as it, allows caging to perform in a variety of configuration. However, this arm must always maintain a certain shape around an object.

HDOF has been used by several researchers for solving control problems such as kinematic modeling [1], [2] path planning [3], [4], inverse kinematics [5], [6], [7], [8], locomotive gait design [9], [10], obstacle avoidance [11], [12], serpentine locomotion control [13], [14] and sidewinding locomotion control [15] problems.

In our work, we study the shape control of a highly kinematic structure, called a HDOF arm manipulator. The HDOF is composed of serial chain links $l_i, i = 1,....,N$, connected to other with revolute joints $j_i, i = 0,...,N-1$. Each link is a straight rigid part of length L . The link l_1 and link l_N are called the base and the tail, respectively. The angle θ_1 is defined as the angle between link l_1 and x-axis. The set of angle defines the manipulator configuration as shown in Figure 1.

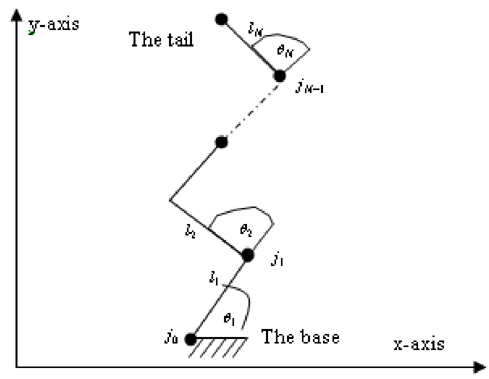

Fig. 1. A hyper degree of freedom (HDOF) structure.

3 INVERSE-KINEMATICS PROBLEM

A HDOF manipulator has ability to move in highly constrained environment or grasp various sized and shaped objects. HDOF manipulators are categorized into 3 types of mechanisms which are serial rigid, parallel rigid and tentacle-like. A serial rigid robot arm is consisting of links and joints in chain structure. In forward kinematics or direct kinematics, the joint displacement and link parameters are given in order to find the end-effector position. Conversely, the inverse kinematics is to solve for the joint displacement when the end-effector position is given.

The transformation matrix relating i^{th} coordinate system (coordinate of end of link *i*) to the *(i-1)*th (coordinate of end of link *i-1*) coordinate system is ${}^{i-1}_{i}T$ given by

$$
{}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \sin\theta_i & -\sin\alpha_i \sin\theta_i & \alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)
$$

where parameters are

a_i = the distance from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$

α_i = the angle from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$

d_i = the distance from $\hat{X}_i$ to $\hat{X}_{i-1}$ measured along $\hat{Z}_i$

θ_i = the angle from $\hat{X}_i$ to $\hat{X}_{i-1}$ measured along $\hat{Z}_i$

$\hat{Z}_i = \hat{Z}_i axis$ of frame *{i}*

$\hat{X}_i = \hat{X}_i axis$ of frame *{i}*

The transformation matrix is divided into two parts which are rotational part and translation part.

$$
{}^{i-1}_{i}T = \left[\begin{array}{ccc|c} & ({}^{i-1}_{i}R) & & ({}^{i-1}_{i}P) \\ \hline 0 & 0 & 1 & 1 \end{array}\right] \quad (2)
$$

Transformation matrix form the base frame {0} to link *n* is described by

$$
{}^{0}_{n}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \ldots {}^{n-1}_{n}T \quad (3)
$$

Since a HDOF manipulator has large number of de grees of freedom (DOF), the inverse kinematics solutio n is not unique. Moreover, the solution of inverse kine matics of the robot arm is difficult to find. The algebr aic and numerical methods are usually employed to sol ve the inverse kinematics problem. The concept of alg ebraic method is to transform the kinematics equations to a high degree polynomial in the tangent of the hal f-angle of joint variable. However, it is complicate in t he nonlinear system.

The numerical methods that are widely used in solv ing for inverse kinematics is the Newton-Raphson itera tion method. Other optimization techniques can also be used. The concept of inverse kinematics problem is si milar to minimization problem where the error between the current position and desired position is minimized. Therefore, nonlinear optimization techniques such as n eural network [16] and genetic algorithm [17] can be applied to this problem.

4 RADIAL BASIS FUNCTION (RBF) NETWORK

The Radial Basis Function (RBF) Networks is a single hidden layer feed forward neural network as shown in Figure 2. Each node of the hidden layer has a parameter vector called the center. This center is used to compare with the network input vector to produce a radial symmetrical response. The response of the hidden layer are scaled by the connection weights of the output layer and then combined to produce the network output. The response of the j^{th} hidden node to input data vector x_i, dimensionality M, is given by (4).

$$
\phi_{ij} = \exp(-\alpha \|x_i - x_j\|^2) \quad (4)
$$

where c_j is an M-dimensional center and $\propto$ is a constant which determines the spread factor of the symmetric response of the hidden node. The network output is defined as

$$
\hat{\gamma}_i = \sum_{j-1}^{k} \varphi_{ij} h_j \quad (5)
$$

where h_j are the network's second layer connection weights and k is the number of hidden nodes.

The widely used RBF network, may use other functions e.g. piecewise linear, cubic approximation, the thin plate spline, the multiquadratic, and the inverse multiquadratic function in place of the Gaussian.

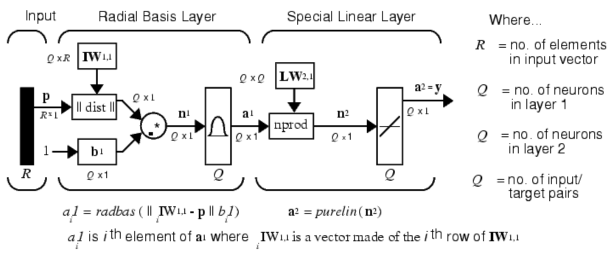

Fig. 2. Architecture of an RBF neural network

The performance of the networks is measured by a Mean-Squared-Error (MSE). The main objective of the training procedure is to approximate the underlying function of the system.

5 THE NEW RBF NETWORK WITH PROPOSED ADAPTIVE STRCUTURE

To determine an appropriate structure for the RBF network, a modified RBF network with adaptive structure is proposed. Initially, in the proposed structure, the radial basis layer has one hidden neuron. Afterward, the network iteratively appends one RBF node to the hidden layer at each training epoch until the error falls beneath an error goal or the maximum number of neurons has been reached. Unlike a traditional network with a fixed structure, the proposed network gradually searches for a minimum number of hidden nodes needed to meet the performance goal. The overall algorithm is given as the following:

- Step 1. Initialize the network using the Structure having a single neuron in the hidden layer.
- Step 2. For each training epoch, feed all input vectors to the network and train the network according to the RBF training algorithm.
- Step 3. Find the input vector in which the network output yields the greatest error.
- Step 4. Add one neuron to the hidden layer with its weight vector equals to the vector obtained from Step 3.
- Step 5. Repeat Step 2 until the performance goal is met or the maximum number of hidden nodes has been reached.

5 EXPERIMENTAL AND RESULT

In this section, experimental results are presented in Table 1 and Figure 3. I try to solve the inverse-kinematics in vary the number of links from the 8 and 20 serial link chains. I will measurement the algorithm in term of the error, time and number of nodes when compares with back-propagation neural network with same structure. The error is calculated from the distance between end-effector and target position. Our experiment is tested on the Core-I7 3.4GHz and 16 GB of RAM.

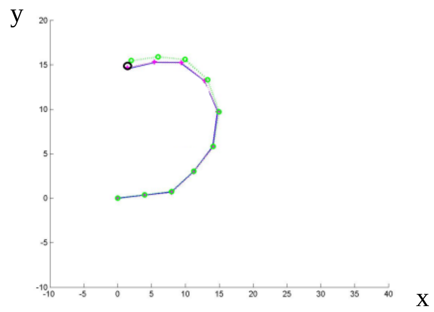

Fig. 3. Comparison between Back-Propagation Neural (Dot line) and our proposed method (solid line) of 9 links serial link chain

Table 1. The comparison between Back-Propagation and Our proposed method

Link	Back-Propagation			Our proposed method		
	Node	err	time (s)	Node	err	time (s)
8	18	1.250	258	18	1.020	260
9	24	3.041	301	24	2.570	284
12	41	3.225	440	41	3.000	365
14	52	4.655	665	52	4.080	556
16	51	4.787	896	51	4.220	803
18	52	5.200	994	52	5.050	925
20	80	6.250	1226	80	6.220	1198

From Table 1, when the number of the link increased, the error and time will increase. Especially, the time is very much consuming.

6 DISCUSSION

Our proposed method is show the performance of finding the suitable number of the node. The proposed method can solve the problem of inverse kinematic. Moreover, we can able to find the suitable number of node for other neural network such as Back-propagation.

In the future, I will try to reduce the time consuming, because the time is abundantly grown when add the number of link.

REFERENCES

[1] S. Chirikjain and J. Burdisk, "The kinematics of Hyper-Redundant robot locomotion," IEEE Transactions on robotics and automation., vol. 11, no. 6, Dec. 1995.

[2] F. Matsuno and K. Mogi, "Redundancy controllable system and control of snake robots based on kinematic model," Proc. of the 39th IEEE conference on decision and control., pp. 4791-4796, 2000.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[5] H. Chang, "A closed-form solution for inverse-kinematic of robot manipulators with redundant," IEEE journal of robotics and automation., vol. ra-3, no. 5, Oct. 1987.

[6] Y. Li and H. Leong, "Kinematics control of redundant manipulators using a CMAC neural network combined with genetic algorithm," Robotica., vol. 22, pp. 611-621, 2004.

[7] K. Mathia and R. Saeks, "Inverse kinematics via linear dynamic network," World congress on neural networks, 1994.

[8] Y. Zhau, T. Huang, and Z. Yang, "A new numerical algorithm for the inverse position analysis of all serial manipulators," Robotica., vol. 00, pp. 1-4, 2005.

[9] G. Kulali and et al, "Intelligent gait synthesizer for serpentine robots," Proc. of the 2002 IEEE international conference on robotics and automation., pp. 1513-1518, 2002.

[10] A. Crespi and et al, "An amphibious robot capable of snake and lamprey-like locomotion," Proc. of the 2005 IEEE international conference on robotics and automation., pp. 3035-3039, 2005.

[11] H. Xie, "Real-time collision avoidance for a redundant manipulator in an unstructured environment," Proc. of the 1998 IEEE international conference on intelligent robotics and system., pp. 1925-1930, 1998.

[12] S. Ma and M. konno, "An obstacle avoidance scheme for hyper-redundant manipulators – Global motion planning in posture space –," Proc. of the 1997 IEEE international conference on robotics and automation., pp. 161-166, 1997.

[13] J. Ostrowski and J. Burdick, "Gait kinematics for a serpentine robot," Proc. of the 1996 IEEE international conference on robotics and automation., pp. 1294-1299, 1996.

[14] J. Conradt and P. Varshavskaya, "Distributed central pattern generator control for a serpentine robot," Proc. of the joint international conference on artificial neural network., 2003.

[15] J. Burdick, J. Radford, and S. Chirikjian, "Sidewinding" locomotion gait for hyper-redundant robots,", Proc. of the 1993 IEEE international conference on robotics and automation., pp. 101-106, 1993

[16] K. Mathaia and R. Saeks, "Inverse Kinematics via Linear Dynamic Networks", Proceeding on World Congress on Neural Networks, pp. 47-52, 1994.

[17] Y. Li and A. Hong Lenog, "Kinematics control of redundant manipulators using CMAC neural network combined with a genetic algorithm", Robotica, pp. 611-621, 2004.

