01 Jan 1991

Intelligent Control of a Robotic Arm Using Hierarchical Neural Network Systems

Xavier J. R. Avula

Missouri University of Science and Technology, avula@mst.edu

Luis C. Rabelo

Recommended Citation

X. J. Avula and L. C. Rabelo, "Intelligent Control of a Robotic Arm Using Hierarchical Neural Network Systems," *Proceedings of the Seattle International Joint Conference on Neural Networks, 1991*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1991.

The definitive version is available at https://doi.org/10.1109/IJCNN.1991.155428

INTELLIGENT CONTROL OF A ROBOTIC ARM USING HIERARCHICAL NEURAL NETWORK SYSTEMS

Luis C. Rabelo+ and Xavier J. R. Avula*

+ Center for Technology Transfer
*** Dept. of Mechanical and Aerospace Eng. and Engineering Mechanics**
University of Missouri-Rolla
Rolla, Missouri 65401

ABSTRACT

In this paper, two artificial neural network systems are considered in a hierarchical fashion to plan the trajectory and control of a robotic arm. At the higher level of the hierarchy the neural system consists of four networks: a Restricted Coulomb Energy (RCE) network to delineate the robot arm workspace, two standard Back Propagation (BP) networks for coordinates transformation, and a fourth network which also uses BP and participates in the trajectory planning by cooperating with other knowledge sources. The control emulation process which is developed using a second neural system at a lower hierarchical level provides the correct sequence of control actions. An example is presented to illustrate the capabilities of the developed architectures.

1. INTRODUCTION

In recent years, there have been several attempts to apply artificial neural networks (ANN's) to the engineering problem of robotic control. Guez and Ahmad [5] presented a hybrid approach using a multi-layered feedforward network to the iterative solution of robotic manipulators which resulted in accelerated convergence in the inverse kinematics. Guo and Cherkassky [6] presented a solution algorithm, using an analog neural computational scheme to implement the Jacobian control technique in real time which is desirable in practical control problems. Sobajic et al. [18] investigated the control of a constrained robot manipulator using backpropagation, and showed that the manipulator could be moved toward a target in the presence of different disturbances. Bassi and Bekey [3] provided a simulated control strategy which indicated that it is practical to control a manipulator to an arbitrary degree of precision by using a neural network whose transformation has a relatively low precision. Liu et al. [8] employed an adaptive neural network in building a generic architecture for robot hand control. This architecture allows device-independent control and separates the low level control problems from high level functionality. In our study we have developed an architecture which can yield a strategy for dynamic decision-making that allows the robot end-effector to reach it's goal using apriori and on-line contextual information.

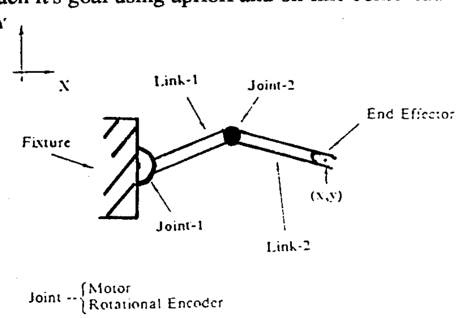


FIGURE 1. Robot Arm Structure

In this paper, two different artificial neural systems associated with the prototype of a scheme which uses the integration of artificial neural networks and knowledge-based systems for the motion control of a 2D arm robot with joint links of equal length is reported (see Fig. 1). These artificial neural systems participate in the tasks of the trajectory planning at the higher hierarchical level (artificial neural system I) and the process of control-emulation (artificial neural system II) at the lower level.

The control-emulation is based on previous developments by Werbos [20,21], Barto et al. [2], Nguyen and Widrow [13], Jordan [7], and Kong and Kosko [9]. The problem to be solved is to provide the correct sequence of control actions to incite the robot arm to go from an initial position to a target position. This "correct sequence" of control actions is decided by a plan generated by the higher order element which includes the kind of desired trajectory to follow. In this study we have utilized forward modelling because of the availability of data. The forward model is taught by the high level planner using simulated sensor feedback and implemented in a standard backpropagation (BP) network [16]. Then, the emulator network of the robot arm dynamics is developed and the controller implementation is started. The controller is implemented using a BP network which is driven by the high level planner which takes the decision on the kind of trajectory to be followed (i.e., linear, circular, linear/circular). The frequency of these trajectory changes is totally handled by the high level planner. In addition, the controller receives

0-7803-0164-1/91/0000-0747$01.00©1991 IEEE

input from the emulator at a higher frequency. The methodology which has been successfully used to train BP networks in the motion analysis is utilized to train the emulator-controller system. This methodology is proven to be very efficient in the development of the required networks which especially involve large training data sets.

2. ARTIFICIAL NEURAL SYSTEM I

The contributions of the artificial neural system I to the motion analysis process is divided into three distinct tasks, each with its own associated neural network arrangement (See Fig. 2). The first part, which consists of the preliminary motion feasibility analysis, uses a Restricted Coulomb Energy (RCE) [14,15] network to delineate the robot-arm workspace and evaluate whether the end-effector could reach the proposed location. If the goal is feasible, a second system of artificial neural networks is employed using BP to map the angular coordinates of the two possible robot configurations that are consistent with the end-effector. Finally, the third network using BP analyzes both alternatives and select the most adequate one in accordance with the initial position. This third network makes an initial proposition to the motion decision making mechanism which uses the cooperation of other knowledge sources (e.g. knowledge bases, algorithms, procedures) with more contextual information leading to the final decision. This final decision will create a plan for the control actions which are going to be implemented by the lower elements of the hierarchy.

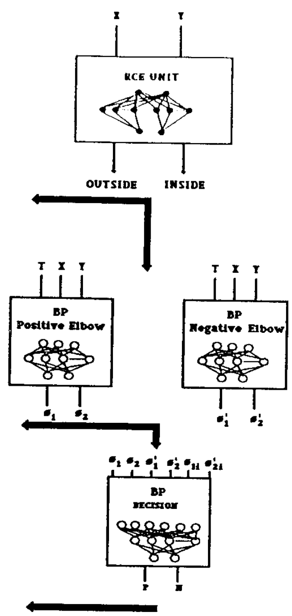
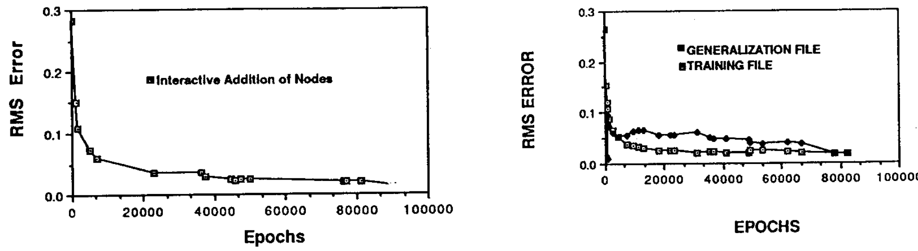


FIGURE 2. Artificial Neural System I

The NDS 500 network package from Nestor nc. (which uses the RCE paradigm) was used to obtain the workspace mapping [11,12,17]. The training data is clustered according to the class boundaries to help the learning process. In addition, to reduce the level of overlapping of the cells in the internal layer, very small minimum influence fields were selected. The training data file consisted of 4500 points. The categorization success rate after the network achieved equilibrium (five passes over the data set) was perfect for the training data set and for the 499 point generalization testing data set. This final network had 426 units. Of these, 115 had overlapping influence fields, while the rest possessed influence fields that were exclusive. The learning process showed a healthy input subspace growth. This growth, represented by the number of internal layer cells versus the number of training samples presented, is a good indication of the quality of the categorization.

The artificial neural networks for coordinate transformation were developed using the standard propagation paradigm, one for each possible configuration. Applying the techniques to achieve an appropriate architecture (e.g., increasing the hidden units based on the training RMS error) and monitoring the generalization values, two architectures of 15 units each in the hidden layer were developed [1,4,19]. After an appropriate architecture was found, increments to the training size were applied to raise the total number of training input vectors from 50 to 400 [19]. The history of the training, as seen by the RMS error as a function of the number of epochs (i.e., one pass over the entire training set) and training data size show the effects of the data increase (see Fig. 3). Both trainings had a higher number of epochs (since the original networks for each case had one hidden unit). Other trainings starting with a higher number of initial hidden units reduced the number of epochs needed significantly. Temperature was indeed a destabilizing factor which added complexity to the concept to be learned. A mapping, in which temperature effects were neglected, and an architecture with less dimensions was found (i.e., 8 hidden units). On the other hand, if temperature is used, a neural network with more dimensions will be needed (i.e. hidden units).

The last stage of the robotics motion analysis system is responsible for choosing one of the two possible solution configurations in accordance with the initial position of the arm. In this stage, an ANN based on BP is used. It has six inputs representing both possible configurations and the initial position of the arm, two outputs specifying the rank given to each configuration.



(a) **Interactive Addition of Nodes** (b) **Utilization of Training and Validation Files**

FIGURE 3. Training Techniques Utilized

3. ARTIFICIAL NEURAL SYSTEM II

The artificial neural system II has the mission to provide the correct sequence of control actions and to emulate the robot arm. This artificial neural system consists of two BP ANNs (see Fig. 4). One is the neurocontroller which, based on plans developed by the higher order element, will provide the control sequence and the other one is the emulator of the robotic arm.

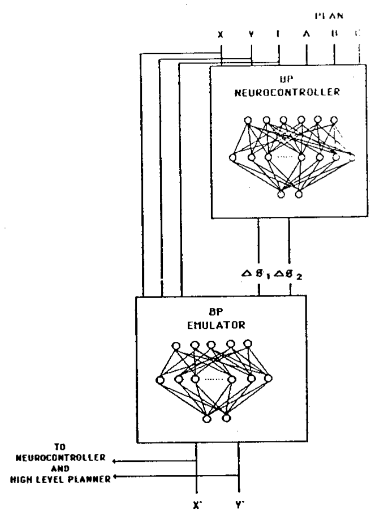


FIGURE 4. Artificial Neural System II

The arm emulator is needed in order to identify the arm dynamic behavior. The knowledge, in order to develop the emulator, should be encoded in the higher order element. The higher order element could examine the arm responses in the cartesian plane with different motor actions. This process will be implemented repetitively until an efficient emulator is developed.

An ANN is successfully developed using the techniques previously mentioned. This ANN has 5 inputs identifying the current X, Y cartesian positions, the elbow and shoulder angle increments (which could define the width of the motors drive pulses in our discrete-time model), and the temperature (see Fig. 4). The ANN has two outputs which correspond to the X, Y position (i.e., next state) after the movement has been performed. The ANN developed has 42 hidden units. The training sessions of this ANNs were very difficult due to abundant oscillations resulting in multiple reinitialization of various parameters. It was apparent that the relationship was harder to learn, and the number of training data samples utilized (6000) in order to achieve an appropriate accuracy were the major reasons for this behavior.

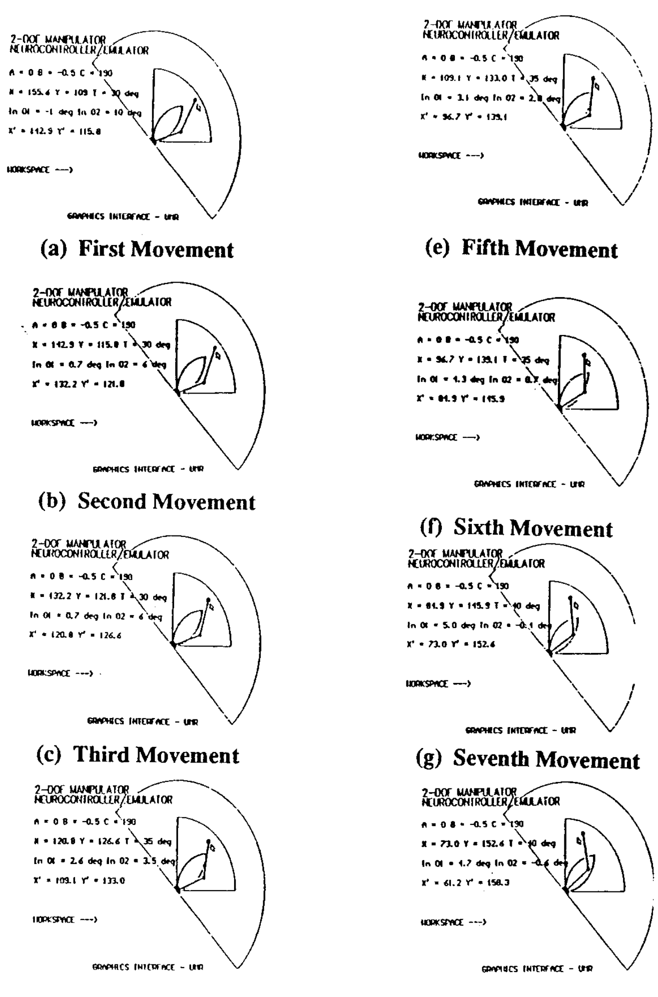
A neurocontroller is implemented to provide a sequence of orders in order to drive the arm from an initial position to a target position. The emulator is used to teach the controller by the higher order element which provides the plans. The plans in this application define the type of trajectory desired. The trajectory could be defined as linear or circular. The higher element decides according to sensory feedback or emulator responses to modify the plans given to the neurocontroller. To define the type of trajectory the coefficients of a line (B * x + C) or circle (A * x + B * x + C) are provided to the neurocontroller. The neurocontroller taking into consideration the present state vector (X, Y, T) and the plan (A, B, C) generates the discrete increments for the elbow and shoulder angles (see Fig. 4). Then the "new" present state vector is used repeatedly till the targeted position is achieved. The necessary comparison is done by the higher element which will decide to send the STOP order A=B=C=0).

4. AN EXAMPLE

Here we consider an example to illustrate the integration of the different artificial neural networks toward the development and accomplishment of a motion task. The goal of this task is to drive the manipulator from the initial position X = 155.6 cm, Y = 109.0 cm, Ø1 = 16.8, Ø2 = 36.4, and a temperature T = 30 C to the final position X = 58.1 cm, Y = 159.7 cm at temperature T = 40 C using a linear trajectory.

After the RCE Unit declares that the targeted position is indeed inside of the workspace, the two ANNs developed for coordinate transformation mappings are executed.

A selection from the two possible configurations is made by using a BP network which takes into consideration the initial position. The high level planner develops the plan for the neurocontroller which in this example are the coefficients of the trajectory between the initial and the goal position (A = 0, B = -0.5, and C = 190). The artificial neural system II will be monitored by the high level planner in order to decide when it is going to be reasonable to stop (current work is developing an architecture which is going to be able to changes plans dynamically). Finally, the Neurocontroller drives the manipulator arm from the initial position to the final position using a quasi-linear trajectory. In spite of the temperature changes, the neurocontroller is able to adapt and the manipulator reaches a point near the final position. See Fig. 5 for details of the eight movements required by the systems in order to reach the targeted position.



(a) First Movement

(e) Fifth Movement

(b) Second Movement

(f) Sixth Movement

(c) Third Movement

(g) Seventh Movement

(d) Fourth Movement

(h) Eighth Movement

FIGURE 5. Neurocontroller/Emulator Execution

5. CONCLUSIONS

Artificial neural networks (ANN's) have capabilities to learn, perform massively parallel processing, and to adapt to complex environmental changes. These capabilities are especially significant in robotics for providing enhanced systems with abilities of learning and self-organization, and efficient real-time operation. In this paper, we have demonstrated the utilization of several ANN's to support the robot motion decision analysis and drive a manipulator arm. The supervised learning models which are introduced here have the following capabilities:

* Perform automated modeling of workspaces in spite of their arbitrary shape,
* Support the decision-making using contextual information in order to provide an initial solution,
* Provide systems which are adaptable to environmental changes, and even constructional imperfections and other faults,
* Develop controllers which can create their own knowledge bases about the system dynamics.

REFERENCES

1. ASH, T., "Dynamic Node Creation in Backpropagation Networks", Institute for Cognitive Science Report 8901, University of California, San Diego, 1989.

2. BARTO, A., SUTTON, R. and ANDERSON, C., "Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems," IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, pp 834-846.

3. BASSI, D. F. and BEKEY, G. A., "High Precision Position Control by Cartesian Trajectory Feedback and Connectionist Inverse Dynamics Feedforward" Proceedings, International Joint Conference on Neural Networks (IJCNN), Washington, DC, 1989, pp. II-325-332.

4. CHAUVIN, Y., "Dynamic Behavior of Constrained Back-Propagation Networks," Advances in Neural Information Processing Systems 2, edited by David Touretzky, Morgan Kaufmann Publishers, 1990, pp 642.

5. GUEZ, A. and AHMAD, Z., "Accelerated Convergence in the Inverse Kinematics via Multi-Layered Feedforward Networks", Proceedings, IJCNN, Washington, 1989, pp. II-341-347.

6. GUO, J. and CHERKASSKY, V., "A Solution to the Inverse Kinematic Problem in Robotics Using Neural Network Processing", Proceedings, IJCNN, Washington, DC, pp. II-299-304.

7. JORDAN, M., "Generic Constraints on Underspecified Target Trajectories," IJCNN Conference Proceedings, Vol.1, June 1989, pp I217-I225.

8. LIU, H., IBERALL, T, and BEKEY, G. A., "Neural Network Architecture for Robot Hand Control", IEEE Control Systems Magazine, April, 1989, pp 38-42.

9. KONG, S., KOSKO, B., "Comparison of Fuzzy and Neural Truck Backer-Upper Control Systems," IJCNN Proceedings, San Diego 1990, pp III349-III358.

10. McCLELLAND, J., RUMELHART, D., "Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises," MIT Press, Cambridge, 1988.

11. NESTOR, Inc., USER'S GUIDE, Nestor, Inc., 1989.

12. NESTOR, Inc., NDS REFERENCE MANUAL, Nestor, Inc., 1989.

13. NGUYEN, D., and WIDROW, B., "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks," Proceedings of the International Conference Joint Conference on Neural Networks, Washington D.C. 1989, pp 357-363.

14. REILLY, D., COOPER, L, ELBAUM, C. "A Neural Model for Category Learning," Biological Cybernetics, Vol. 45, 1982, pp 35-41.

15. REILLY, D., SCOFIELD, C., ELBAUM, C., COOPER, L., "Learning System Architectures Composed of Multiple Learning Modules," Proceedings of the First International Conference on Neural Networks, 1987.

16. RUMELHART, D., McCLELLAND, J., and the PDP Research Group, PARALLEL DISTRIBUTED PROCESSING: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, Cambridge, MA: MIT Press, 1988.

17. SCOFIELD, C., REILLY, D., ELBAUM, C., COOPER, L., "Pattern Class Degeneracy in an Unrestricted Storage Density Memory," Nestor Inc., 1988.

18. SOBAJIC, D., LU, J. J. and PAO, Y. H., " Intelligent Control of the INTELLEDEX 6057 Robot Manipulator", Proceedings, IJCNN, San Diego, 1988, pp II-633-640.

19. TSUNG, FU-SHENG, COTTRELL, Garrison W., "Sequential Adder Using Recurrent Networks," IJCNN Conference Proceedings, Vol.2, June 1989, pp II133-II139.

20. WERBOS, P., "Building and Understanding Adaptive Systems: A Statistical/Numerical Approach to Factory Automation and Brain Research," IEEE Transactions on Systems, Man, and Cybernetics, 17, pp 7-20.

21. WERBOS, P., "Backpropagation and Neurocontrol: A Review and Prospectus," IJCNN Conference Proceedings, Vol. 1, June 1989, pp I209-I216.