BIOLOGICAL VISUO-MOTOR CONTROL OF A PNEUMATIC ROBOT ARM

M. ZELLER, K. R. WALLACE AND K. SCHULTEN
Theoretical Biophysics Group, Beckman Institute
University of Illinois at Urbana-Champaign
Urbana-Champaign, IL 61801, USA

ABSTRACT:
We are investigating the application of biologically plausible neural architectures to the problem of the controlling movement of a limb in response to visual stimuli. Our approach is inspired by the ability of biological systems to develop highly accurate control strategies for movement by means of associations between random motor actions and the sensory consequences of these acts. This is achieved using information processing algorithms based upon Kohonen's self-organizing feature map algorithm (SOFM). As a means of enhancing the abilities of this system we have incorporated additional processing stages, the operation of which are based upon strategies that are consistent with neurobiological approaches to the problem of motor control.

INTRODUCTION

Both classical control strategies and artificial neural networks, have been employed in a number of different manners for the control of industrial robots. In contrast to biological systems, however, artificial neural networks are generally confined to one dedicated task or simple sensory to motor transformations. It would appear that a combination of the engineering approach of robotics and biologically motivated models of motor control offers potentially promising synergies: more flexible robot control applications on the one hand, and greater insight into the biological basis of motor learning on the other. As a means of pursuing this objective and as a practical demonstration of the capabilities of neural network algorithms within an engineering framework, one research theme within our group has been to implement neural control algorithms on a pneumatically driven robot arm (*SoftArm*). This robot provides an excellent testbed for biologically plausible models of motor control as has been demonstrated in previous studies (Hesselroth et al., 1994).

Our approach involves the application of *self-organizing feature maps*, originally proposed by Kohonen (Kohonen, 1982), as the basic information processing element. Such networks have been successfully applied to the problem of controlling movement in several technical applications (Ritter et al., 1989). In

In: Dagil, Akay, Chen, Fernandez and Gosh (editors). Artificial Neural Networks in Engineering. ASME Press (1995).

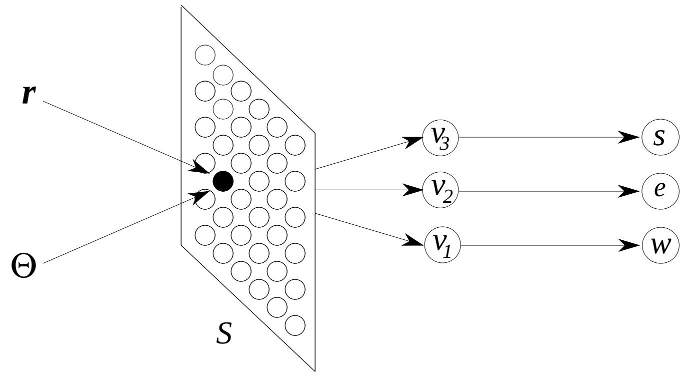

Figure 1: The basic network used to control movement. w, e and s, refer to the joint wrist, elbow and shoulder joints of the arm respectively. Definitions of all other quantities may be found in the text.

common with a number of previous studies of motor control, for example that of Kupperstein (Kupperstein, 1989), our present approach involves the development of connections between an input (sensory) and output (motor) map, the connections between these maps being achieved by means of a learning process.

OPERATION OF THE NETWORK

To simplify the description of the algorithm, we will first discuss application of the algorithm for use in the control of a simple movement system consisting of a simulation of the human arm moving in the plane. It is assumed that one end of this chain, in this case corresponding to the shoulder, is fixed in the workspace.

Figure 1 illustrates the elements of the basic network. Neurons in layer S project *via independent excitatory synapses* to a set of motor cells v_i responsible for setting the joint angles of a simulation of the human arm moving in the plane. We assume an input space defined by M independent sensory input sources. In a biological system these sources might, for example, correspond to neurons providing tactile input from receptors distributed over the body surface. In the present work, however, we will be concerned with proprioceptors, which indicate the respective joint angles of each of the segments of the limb and visual receptors which specify the location of a target point of interest.

Two different types of sensory information converge upon neurons within the network S. Exteroceptive input

$$\mathbf{r} = [x_1, x_2] \tag{1}$$

is derived from a Euclidean coordinate system defined by the visual field presented to the network. Proprioceptive input, denoted $\boldsymbol{\Theta}$, is derived from the intrinsic coordinate system of the joints of the human arm where

$$\boldsymbol{\Theta} = [\theta_1, \theta_2, \theta_3] \tag{2}$$

The indices $1, 2, 3$ specify the wrist, elbow and shoulder joints of the limb, respectively. Control of limb movements in the workspace is achieved by modifying the synaptic weights of the projections from neurons in the sensory layer S to the motor cells. Each neuron in the sensory layer has a vector

$$\mathbf{V}_j = [v_j^1, v_j^2, v_j^3] \tag{3}$$

associated with it which corresponds to the output of the motor neuron when activated by a neuron in the sensory layer. This output alters the joint angles of each segment of the limb simulation in the workspace.

During learning, adjustment of v_j^i, the i^{th} component of $\mathbf{V}$, is calculated as

$$v_j^i(t) = v_j^i(t-1) + \epsilon(t) h_{js} (u^i(t) - v_j^i(t-1)) \tag{4}$$

where, in this instance, $v_j^i(t-1)$ represents a *random* value generated during the previous iteration of the algorithm leading to movement of a particular limb segment (Coiton et al., 1991).

Prior to learning, all components of the vector $\mathbf{V}$ are assigned random values and the total number of learning steps is specified. For each learning step a sensory input vector $\mathbf{U} = [\mathbf{r}, \mathbf{\Theta}]$ is then formed from exteroceptive input $\mathbf{r}$ given by the values of the endpoint of the limb and proprioceptive input $\boldsymbol{\Theta}$ specified by the joint angles of the limb. The Kohonen algorithm is applied to the sensory layer and the vector of motor signals $\mathbf{V}_s$ associated with the neuron s, chosen according to the Euclidean distance criteria, in the sensory layer, initiates movement of the arm to a new randomly chosen position in the workspace. The components of $\mathbf{V}_s$ are then adjusted according to (4) and this sequence of operations is repeated for the total number of learning steps. Following a suitable number of learning steps, typically 3000, goal-directed movements to visual targets can be executed by the network as described in (Wallace and Schulten, 1994).

APPLICATION OF THE ALGORITHM TO CONTROL THE *SoftArm* PNEUMATIC ROBOT SYSTEM

The *SoftArm* is modeled upon the human arm and has four joints resulting in five degrees of freedom. It exhibits the essential mechanical characteristics of skeletal muscle system by means of agonist-antagonist pairs of *rubbertuators* which are mounted on opposite sides of rotating joints. When air pressure in a *rubbertuator* is increased, the diameter of the tube increases thereby causing the length of the tube to decrease and the joint to rotate. Stiffness of the joint is determined by the total pressure in both the agonist and antagonist tubes such that the compliance of individual joints may be varied. A more detailed description of the mechanical characteristics is given in (Hesselroth et al., 1994). The complete robot system consists of the *SoftArm*, air supply, control electronics (servo drive units) and a workstation which includes a serial interface, connected to the robot's servo drive units, and a video input card. The servo drive units provide the internal control circuitry of the robot, operate the servo valve units and send joint angle data, available from optical encoders mounted on each joint, to the computer.

Use of the basic algorithm for control of the *SoftArm* results in average accuracies in the region of 12% of the dimensions of the workspace of this system. Clearly such performance is unacceptable. A number of factors contribute to this poor performance, including the mechanical characteristics of the SoftArm and the increased dimensionality of the problem (nine dimensions as opposed to five for the simulations of planar arm movement). The principal problems that arise, however, are the need for a single network to provide a 'good' representation of two distinct sensory input spaces, namely the visual and proprioceptive spaces, and the discretizing effect that results from the use of small numbers of neurons to map these input spaces. In general, while the use of larger number of neurons in the network can lead to some improvements in performance there is no simple linear relationship between greater numbers of neurons and accuracy (Ritter, 1989).

AN IMPROVED ALGORITHM FOR MOTOR CONTROL

There is now considerable evidence to suggest that biological systems adopt a distributed approach to the problems inherent in motor control. This is evidenced by the processing which occurs in parallel at many sites of motor activity, connected in a semi-hierarchical fashion (Johnson, 1992). These observations have led us to propose a model of cerebro-cerebellar interactions that results in significant improvements in the performance of the basic algorithm outlined (Wallace and Schulten, 1994). For this model we adopt a task-related strategy in which the motor cells specify the intended absolute endpoint of a movement while an additional component provides information regarding *a relative movement* necessary to achieve a desired target point.

Extension of the algorithm is achieved by defining a set C consisting of D neurons. Each neuron $j \in C$ has a randomly assigned value, denoted by φ_j, associated with it which lie in the closed interval $[0, 1]$. In addition each neuron $j \in C$ has a set of connections denoted by $e^i_j, i = 1, 2, \ldots, m$, corresponding to 'projections' to the motor cells controlling movement of the individual joints of the limb arising from the cerebellar component.

We assume that the projections e^i_j *can assume multiple values* in the range $[0, 1]$, the actual value being *dependent upon the movement being learnt.* This task-related approach derives from the observation that, anatomically, cerebellar organization is based upon task-related 'zones' which constitute the basic operational unit of the cerebellar cortex (Ito, 1984). We assume that distinct tasks are reflected in distinct patterns of the projections from C to S, thereby, resulting in different patterns of activation to the motor cells.

Following the initial phase of the basic algorithm during which the sensory layer and motor cell connections are developed, an additional phase is introduced to the simulations during which initially random cerebellar input, modulating the output from the motor cells, is introduced. This phase involves repeated practicing until movement to a particular target approximates the target location to some predetermined accuracy. The cerebellar component is obtained by calculating

$$\gamma = \arccos\left(\frac{\hat{\mathbf{r}}_p \cdot \mathbf{r}_{\theta_1}}{|\hat{\mathbf{r}}_p||\mathbf{r}_{\theta_1}|}\right) \tag{5}$$

where $\hat{\mathbf{r}}_p$ denotes the location of the target point *relative to the position of the wrist joint* and $\mathbf{r}_{\theta_1}$ denotes the vector running from the wrist joint to the tip of the hand. For target points that involved negative rotation of the wrist joint, the sign of γ is defined to be negative. This value is normalized

$$\hat{\gamma} = (\gamma + \pi)/2\pi \tag{6}$$

and the cerebellar neuron c given by

$$\parallel \varphi_c - \hat{\gamma} \parallel = \min \parallel \varphi_j - \hat{\gamma} \parallel \tag{7}$$

for $j = 1, 2, \ldots, D$ calculated. The resulting correction factor v_c^i, is calculated as

$$v_c^i = \kappa^i e_c^i \varphi_c \tag{8}$$

where $i = 1, 2, 3$, κ^i is a scaling constant which determines the 'stiffness' of the relevant joint and

$$e_c^i = \begin{cases} \hat{e}_c^i \text{ if } \psi \text{ had previously equalled 1 during practice of the task} \\ \text{uniform deviate in the range } [0, 1] \text{ otherwise} \end{cases} \tag{9}$$

where $\hat{e}_c^i$ denotes the value associated with e_c^i when the quantity $\psi = 1$ indicating that the movement has attained the required accuracy. This condition leads to storage of the set of values output by the cerebellar component leading to the required target point being attained. For a given task, therefore, the nature of the input arising from the cerebellar component is chosen on the basis of (1) the particular task being performed, by virtue of e_c^i assuming multiple, task dependent values and (2) whether a movement of the required accuracy had previously occurred for the task. The final motor signal sent to each joint is

$$v_s^i(S) + v_c^i(C) \tag{10}$$

where S and C explicitly denote the input to the motor cells arising from the sensory and cerebellar components respectively. At each point q during movement to a target point, the quantity δ is calculated

$$\delta = ||\mathbf{r}_p - \mathbf{r}_q|| \tag{11}$$

and compared with a predetermined value $\hat{\delta}$. At each point q the quantity ψ is calculated as

$$\psi = \begin{cases} 0 & \text{if } \delta > \hat{\delta} \\ 1 & \text{if } \delta \leq \hat{\delta} \end{cases} \tag{12}$$

If during movement to a target point, $\psi = 1$ then the movement is terminated and *all values of* e_c^i stored.

DISCUSSION

With the additional processing stage introduced in the algorithm, the absolute accuracy of movement of the planar arm simulation that can be obtained is significantly better than can be obtained through use of the simple algorithm alone. For an arbitrary value of $\hat{\delta} = 1\%$ of the workspace and a set of target points in the workspace, convergence to the target points will typically occur within 100 iterations of the introduction of the cerebellar component, the final error typically being 0.3% of the workspace. More precise movements are easily obtained with only moderately larger number of iterations of the cerebellar processing component.

Simulations in which the improved algorithm is required to control movement of a 3 segment limb moving in 3 dimensions, with 4 degrees of freedom, indicate similar improvements in accuracy over the basic algorithm. Here, however, the problem lies in choosing a set of values for the κ^i that can allow the whole workspace to be arbitrarily approximated equally well. Thus, we currently considering strategies by which additional contextual information regarding the required task can be introduced to the algorithm such that an optimal set of initial parameters can be specified to the algorithm.

REFERENCES

Y. Coiton, J. C. Gilhodes, J. L. Velay, and J. P. Roll. A neural network model for the intersensory coordination involved in goal-directed movements. *Biol. Cybern.*, 66:167–176, 1991.

T. Hesselroth, K. Sarkar, P. P. van der Smagt, and K. Schulten. Neural network control of a pneumatic robot arm. *IEEE Transactions of System, Man and Cybernetics*, 24(1):28–37, 1994.

M. Ito. *The Cerebellum and Neural Control.* Raven Press: New York, 1984.

P. B. Johnson. Towards an understanding if the cerebral cortex and reaching movements: a review of recent approaches. In R. Caminiti, P. Johnson, and Y. Burnod, editors, *Control of arm movement in space. Neurophysiological and Computational Approaches*, Exp. Brain Res. Series 22, pages 199–261, 1992.

T. Kohonen. Analysis of a simple self-organizing process. *Biol. Cybern.*, 44:135–140, 1982.

M. Kuperstein. Neural model of adaptive hand-eye coordination for single postures. *Science Wash. DC*, 239:1308–1311, 1989.

H. Ritter. Combining self-organizing maps. In *International Joint Conference on Neural Networks*, 1989.

H. J. Ritter, T. M. Martinetz, and K. J. Schulten. Topology-conserving maps for learning visuo-motor-coordination. *Neural Networks*, 2:159–168, 1989.

K. R. Wallace and K. Schulten. A model of cortical processing during motor learning. *Submitted to Biological Cybernetics*, 1995.