To be presented at the 15th Annual International Conference IEEE Engineering in Medicine & Biology Society, San Diego, CA, October 28-31, 1993.

FROM CEREBELLAR MODEL ARTICULATION CONTROLLER TO A REFERENCE MODEL FOR INTELLIGENT CONTROL

James S. Albus
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract **-- The Cerebellar Model Articulation Controller (CMAC) is a neural net model for computation in the brain. The Real-time Control System (RCS) is a reference model architecture for intelligent control that evolved out of the CMAC paradigm. RCS has been used in a number of intelligent control applications at NIST and elsewhere.**

I. INTRODUCTION

The Cerebellar Model Articulation Controller (CMAC) is a neural net model that can compute a variety of vector functions of the form

$$\mathbf{P}(t+dt) = \mathbf{H}(\mathbf{S}(t))$$

RCS is a reference model for intelligent control that can (in principal) be constructed entirely from modules that compute such functions.

In CMAC the function **H** is the product of two functions **F** and **G** such that

$$\mathbf{A}(t) = \mathbf{F}(\mathbf{S}(t))$$
$$\mathbf{P}(t+dt) = \mathbf{G}(\mathbf{A}(t))$$

and

S(t) represents a vector of firing rates s(i,t) on a set of input fibers at time t

A(t) represents a vector of firing rates a(j,t) of a set of association cells at time t

P(t+dt) represents a vector of firing rates p(k,t+dt) on a set of output fibers at time t+dt

The function **F** is fixed, serving as an address decoder (or recoder) that transforms the input vector **S** into an association cell vector **A**. The number of association cells in vector **A** is typically two orders of magnitude larger than the number of input fibers. This increases the dimensionality of the **A** space. However, only a few percent of the association cells are non-zero for any particular input vector. Therefore, the **A** vector is sparse. As a result, CMAC can store, or recognize, a large number of non-linear functions [1].

The function **G** depends on the values of a matrix of synaptic weights w(j,k) that connect the association cells to the output cells. The weights w(j,k) may be modified during the learning process so as to modify the function **G**, and hence the overall function **H**.

II. DISCUSSION

CMAC neural nets can be designed to compute many kinds of functions of the form **P** = **H**(**S**). For example, **H** may be an arithmetic or logical function, where the input **S** consists of real, integer, or Boolean variables, and the output **P** is the value of the function expressed in real, integer, or Boolean variables. On the other hand, **H** may be a memory recall function, where the input **S** is an address, and the output **P** is the contents of the address. The process of training the CMAC is the process of storing a value in a memory address. **H** may also be a list processing function, where **S** is an address, and **P** is a pointer to the next address in the list. **H** may be a pattern recognition function, where **S** represents the attributes of an object, and **P** is the name of the object. **H** may be a database function, where **S** is the name of an object, and **P** its attributes.

In cases where one or more of the output variables in **P** loops back to become a part of the input **S**, the input is a function of the previous output, i.e., it contains state variables. In this case, the **S** vector consists of input and state information, and the CMAC neural net can function as a finite state automaton (fsa). If the **H** function in a CMAC fsa performs a summation, the output **P** may contain the integral of the input. If the **H** function of a CMAC fsa performs a difference, the output **P** may contain the temporal differential of the input.

If the input part of **S** to the fsa consists of both command and sensory feedback variables, the CMAC fsa can perform task decomposition. If the input part of S contains both a planned action and a world state vector, the output **P** may contain the predicted result of the planned action. If the input **S** is a predicted result, the output **P** can be an evaluation of **S**. If the input **S** consists of both sensory observations and world model predictions, the output **P** can be a correlation or a difference function, or a recognition function.

A single neuron can function as a delay element, and a series of delay elements can be configured as a tapped delay line. Pairs of neurons can compute either a temporal or spatial difference function, and hence temporal or spatial gradients.

Combinations of CMAC neural nets can compute the vector product of two input vectors, and can sum over any number of products so as to compute the correlation between two signals, or the convolution of a signal with a filter. Correlations or convolutions can be performed in either the temporal or spatial domains, or both.

CMAC neural nets can produce image shifting and scaling operations, and can transform coordinates for either iconic maps or symbolic entities. Iconic map coordinate transforms can be performed by computing for each pixel, the address of its new position under the transform, and moving its present value to the new address. A symbolic entity is transformed by adding an off-set vector to its current position and orientation attributes. A transformation of coordinates for an iconic map is equivalent to an image warping or scrolling operation.

Given the above capabilities, a network of CMACs can perform regressions, and recursive estimations such as Kalman filtering, Fourier transforms, and compute coefficients for any number of series approximations such a Taylor series, Bessel functions, Hermite polynomials, Hankel functions, etc.

Even with these capabilities, however, the question remains of how to organize this processing capacity so as to produce the phenomena of intelligent behavior. This question is addressed by the NIST Real-time Control System (RCS) Reference Model Architecture for intelligent control [2].

RCS evolved out of the attempt to build intelligent control systems out of modules that could (at least in principal) be constructed from CMAC neural nets. RCS defines the elements of intelligence as task decomposition, world modeling, sensory processing, and value judgment. It defines a hierarchy of nodes containing these elements, and a set of data pathways interconnecting the nodes. The RCS architecture demonstrates how task decomposition, world modeling, sensory processing, and value judgment functional modules can be interconnected so as to produce intelligent behavior. Each layer of the RCS hierarchy has a characteristic loop bandwidth, planning horizon, and short term memory interval. Each layer also has a characteristic range and resolution on cognitive maps.

RCS suggests how CMAC neural net modules could be used to generate planning and control functions for task decomposition, how models of the world could be stored, retrieved, and correlated with observations, as well as used to predict the results of planned actions, and how visual, acoustic, and tactile patterns could be recognized, and how value judgments could be computed.

It should be noted that learning is not the most important property of the CMAC neural net or of the RCS architecture. Computation is more important. RCS predicts that intelligent behavior is the product of a complex set of computational processes operating on a set of state variables representing both internal goals and priorities, and sensory observations of the external world. Learning is merely one means by which world model data is updated, motor and perceptual skills are acquired, and values are developed.

RCS has been used in a number of applications of intelligent machine systems, in factories, mines, telerobotics, and unmanned vehicles. An outline for a theory of intelligence based on CMAC and RCS was recently published [3].

REFERENCES

[1] J.S. Albus, "Data Storage in the Cerebellar Model Articulation Controller (CMAC)," *Transactions ASME,* September 1975.

[2] J.S. Albus, "RCS: A Reference Model Architecture for Intelligent Control," IEEE Computer, May 1992.

[3] J.S. Albus, "Outline for a Theory of Intelligence," IEEE Trans. on Systems, Man, and Cybernetics, Vol.21, No.3, May/June 1991.