FA6 - 9:30

Proceedings of the 31st Conference
on Decision and Control
Tucson, Arizona • December 1992

Robust Control of Dynamic Systems using Neuromorphic Controllers: A CMAC Approach

Mario Sznaier

Electrical Engineering Dept.,
University of Central Florida,
Orlando, Fl 32816–2450
msznaier@frodo.engr.ucf.edu

Abstract

During the last few years there has been considerable interest in the use of trainable controllers based upon the use of neuron-like elements, with the expectation being that these controllers can be trained, with relatively little effort, to achieve good performance, even when only minimal knowledge of the plant dynamics is available. However, good performance hinges on the ability of the neural net to generate a "good" control law even when the input does not belong to the training set, and it has been shown that neural-nets do not necessarily generalize well. In this paper we address this problem by proposing a feedback controller based upon the use of a CMAC neural net. We show that the proposed controller has good generalization properties. Moreover, by proper choice of the training set the resulting closed-loop system is guaranteed to be robustly stable with respect to model uncertainty.

1. Introduction

A substantial number of control problems can be summarized as the problem of designing a controller capable of achieving acceptable performance under design constraints and model uncertainty. However, the problem is far from solved, even for linear systems. In most cases engineering goals take the form of time–domain constraints reflecting both performance and physical considerations, such as the presence of "hard" actuator limits or the need to maintain the states of the plant confined to a "safe" region of operation. However, although there currently exist several computationally-efficient design methods capable of handling a wide variety of frequency–domain specifications ([1] and references therein), there presently exist very few methods that allow for systematically dealing with time–domain constraints. Moreover, most of these methods assume exact knowledge of the dynamics involved (i.e. exact knowledge of the model). Such an assumption can be too restrictive, severely limiting the applicability of the resulting controllers.

On the other hand, during the last decade a large research effort led to procedures for designing robust controllers, capable of achieving stability under various classes of plant perturbations while, at the same time, satisfying frequency–domain constraints. However, most of these design procedures cannot accommodate directly time domain constraints [1].

As an alternative to analytical controller design methods, during the last few years considerable attention has been focused on the use of neural–net based controllers, with the expectation being that these controllers can be trained to achieve good performance, even when only minimal knowledge of the plant is available. As an example, we can mention the neuromorphic controller used by Barto et. al. [2] to control an inverted pendulum when the control force is restricted to have bounded magnitude.

Although there is a growing body of different control configurations using neural–net based controllers (see for instance [3] and references therein), the issues involved in using neuromorphic controllers can be illustrated with the simple topology shown in figure 1. Here the goal is to follow, ideally without error, a reference trajectory. The controller consist of a feed–forward net trained, using back–propagation methods, to learn the (approximate) inverse of the plant. Therefore, it is expected that when presented with the reference trajectory, the controller will produce the (previously learned) control action that minimizes the tracking error. It is important to note that this configuration resembles an adaptive controller, where the feed–forward net provides the parametric structure and where the back–propagation training functions as a gradient adaptation mechanism [4].

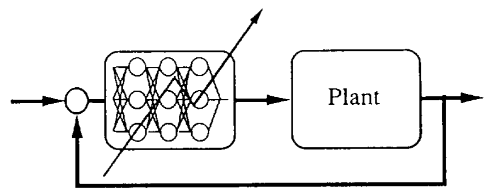

Figure 1. A Simple Neural–Net Based Controller

The success of the controller depicted in figure 1 hinges on the following issues: 1)plant invertibility 2)effective learning and 3)ability of the neural net to generalize, i.e. to provide an appropriate output even when the input is not a member of the training set. Plant invertibility has been exhaustively studied in control theory and therefore poses no new problems. However, the other two issues remain largely ignored in neural net applications to controls. Moreover, the combination of feed–forward nets and back–propagation training exhibits some undesirable properties which may adversely affect the performance of the controller, in particular the stability of the closed–loop system. It is well known (see for instance [5]) that the error surfaces can have local minima and multitude of areas with shallow slopes. Hence the back–propagation algorithm is not guaranteed to converge to a global minimum of the error. Furthermore, even when it does, convergence may take a prohibitively large amount of time, due to the shallow regions. Concerning generalization, it has been shown [6] that feed–forward nets *do not* necessarily generalize well. Therefore, it follows that the *stability* properties of the resulting closed–loop system are generally unknown. Since most critical control applications require "hard", rather than factual proof of closed-loop stability, these difficulties are a major stumbling block preventing the use of neuromorphic controllers, in spite of their potential to outperform classical controllers.

In this paper we propose to solve these problems by using a neural net (CMAC) with inherent good generalization properties and by incorporating a–priori knowledge of the plant dynamics into the design and training processes. We show that by using this knowledge, the resulting neuromorphic controller is capable of robustly stabilizing a family of plants. Furthermore, we give bounds on the mismatch (in the sense of a norm) between the nominal plant (used for the initial training of the network) and the actual plant such that stability of the closed loop system is guaranteed.

The paper is organized as follows: In section II we introduce some required concepts and we present a formal definition to our problem. In section III we briefly describe the CMAC network. Section IV contains

Supported in part by a grant from Florida Space Grant Consortium

the bulk of the theoretical results. The main result of the section shows that by incorporating information about the plant dynamics into the design and training processes, the resulting controller is guaranteed to stabilize a family of systems. In section V we present an example of application. Finally, in section VI we summarize our results and indicate directions for future research.

2. Problem Formulation and Preliminary Results

2.1 Statement of the Problem

Consider the family of discrete-time systems represented by the following state-space realization:

$$\underline{x}_{k+1} = A(q)\underline{x}_k + B(q)\underline{u}_k \qquad (P)$$

where $\underline{x}$ represents the state, $\underline{u}$ represents the control input, _ indicates a vector quantity, $q \in \mathcal{Q}$ compact represents uncertainty, and where the dynamics satisfy the following conditions:

$$A(q) = A_o + \Delta_A(q),\ B(q) = B_o + \Delta_B(q) \qquad (U)$$

Finally, assume that the uncertainties $\Delta_A(q)$ and $\Delta_B(q)$ are norm-bounded by:[‡]

$$\max_{q\in\mathcal{Q}} \|\Delta_A(q)\| \leq \Delta_1, \quad \max_{q\in\mathcal{Q}} \|\Delta_B(q)\underline{u}\| \leq \Delta_2(\underline{u}), \qquad (1)$$

Then, the basic control problem that we address in this paper is the following:

• **Robust Constrained Control Synthesis Problem:** Given the family (P) find a *feedback* controller such that, for all $q \in \mathcal{Q}$, the resulting closed-loop system satisfies the following specifications:

i) The states remain confined to a region $\mathcal{G} \subset R^n$, where $\mathcal{G}$ is a compact, convex balanced set (i.e such that $\underline{x} \in \mathcal{G} \Rightarrow \lambda\underline{x} \in \mathcal{G}$ for $|\lambda| \leq 1$) containing the origin in its interior.

ii) The control effort is constrained by $\underline{u}_k \in \Omega \subset R^m$, where Ω is a compact, convex set containing the origin in its interior.

iii) Given an open, convex, target set O containing the origin in its interior, the system is driven to O, for any initial condition $\underline{x}_o \in \mathcal{G}$. This performance specification is closely related to the concept of practical stability [8].

Remark 1: Note that due to the existence of state and control constraints, this control problem does not admit, in general, a closed-form solution. Hence it is specially well suited for a training-based approach, requiring only knowledge of the appropriate control action for a finite given sets of inputs.

2.2 Definitions and Preliminary Results

In this subsection we introduce the definitions required to analyze the properties of the closed-loop system obtained when using a CMAC-based controller. These ideas, illustrated in figure 2, formalize the concept of "quantization" of state-space.

• **Def. 1:** Consider a closed set $\mathcal{G} \subseteq R^n$. A family $\mathcal{C}$ of closed sets C_i is called a *closed cover* of $\mathcal{G}$ if $\mathcal{G} \subseteq \bigcup_i C_i$

[‡] Since all finite dimensional norms are equivalent [7], it is unnecessary to specify the actual norm. We will use this freedom in selecting the norm in section IV.

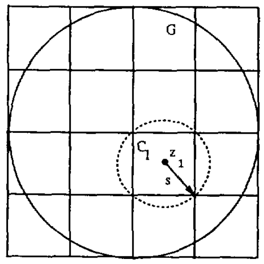

Figure 2: A Closed Cover Formed by Square Boxes of Size s

• **Def. 2:** Consider a closed set $\mathcal{G} \subseteq R^n$ and a closed cover $\mathcal{S} = \{S_i\}$. A quantization χ of $\mathcal{G}$ is a set $\chi = \{z_i\}$ containing exactly one element from each set S_i.

• **Def. 3:** Given a quantization χ of a set $\mathcal{G}$, the *size* of the quantization with respect to some norm $\mathcal{N}$ defined in $\mathcal{G}$ is defined as:

$$s = \min_i \{r: C_i \subseteq B(\underline{z}_i, r)\ \forall i\}$$

where $B(\underline{z}_i, r)$ indicates the $\mathcal{N}$-norm ball centered at $\underline{z}_i$ and with radius r.

Consider now the case where the sets of the family $\mathcal{C}$ that defines a quantization χ have pairwise disjoint interiors (i.e. $\text{int}(C_i) \cap \text{int}(C_j) = \emptyset, i \neq j$). In this case, $\mathcal{C}$ induces an equivalence relation in $\mathcal{G}$ as follows:

• **Def. 4:** Consider a closed cover $\mathcal{C}$ of $\mathcal{G}$ with pairwise disjoint interiors, and two points $\underline{x}_1, \underline{x}_2 \in \mathcal{G}$. $\underline{x}_1$ and $\underline{x}_2$ are *equivalent modulo* $\mathcal{C}$ if $\exists\ i$ such that $\underline{x}_1$ and $\underline{x}_2 \in \text{int}(C_i)$. To complete the partition of $\mathcal{G}$ into equivalence classes, we assign the points that are in $C_i \cap C_j$ (i.e. in the common boundary) *arbitrarily* to either one of the classes. Two points equivalent modulo $\mathcal{C}$ will be denoted as $\underline{x}_1 \equiv \underline{x}_2$.

• **Def. 5:** Consider a quantization $\chi = \{\underline{z}_i\}$ of a given set $\mathcal{G}$. It follows from Definitions 2 and 4 that for *any* point $\underline{x} \in \mathcal{G}$ there exists an element $\underline{z} \in \chi$ such that $\underline{z} \equiv \underline{x}$. We will define the operator that assigns $\underline{x} \to \underline{z}$ as the *quantization* operator and we will denote it as: $\underline{z} = \chi(\underline{x})$.

Finally, we show that the set $\mathcal{G}$ induces a norm in R^n. This norm will be used to design a CMAC-based controller, guaranteed to stabilize the family (P).

• **Def. 6:** [9] The *Minkowsky Functional* (or gauge) p of a convex set $\mathcal{G}$ containing the origin in its interior is defined by

$$p(\underline{x}) = \inf_{r>0} \left\{ r : \frac{\underline{x}}{r} \in \mathcal{G} \right\} \qquad (2)$$

A well known result in functional analysis (see for instance [9]) establishes that p defines a seminorm in R^n. Furthermore, when $\mathcal{G}$ is compact, this seminorm becomes a norm. In the sequel, we will denote this norm as $\|\underline{x}\|_{\mathcal{G}} \stackrel{\Delta}{=} p(\underline{x})$

3. The CMAC Neural Net

3.1 Description of CMAC

In this section we provide a brief description of the Cerebellar Model Articulation Controller (CMAC) neural net. The reader is referred to [10–11] for more details. Originally introduced by Albus [10] for learning to control a robotic arm, the CMAC network has often been overlooked by the Neural Net community, mainly because it was considered impractical. However, in the last few years it has become the focus of growing interest, prompted by the disadvantages of back-propagation mentioned earlier. In particular CMAC has been successfully used to learn state-dependent control actions. Among the recent applications we can mention as examples the work of Miller and coworkers [12-13], Ersu and coworkers [14], and Moody [15].

A diagram of CMAC is shown in figure 3. The input space X is discretized and mapped into a "conceptual" memory M, in such a way that each input $\underline{x}$ excites exactly A^* association cells in M. The mapping $S : X \rightarrow M$ is such that inputs that are close (in the sense of some metric) in input space will have their corresponding sets of association cells overlap, with more overlap for closer inputs. The output corresponding to a given input $\underline{x}$ is obtained by adding the contents of the A^* association cells excited. The number A^* can be thought as the ratio of generalization width to quantization width. A larger A^* provides for better generalization at the price of larger memory requirements or reduced resolution for individual input patterns. Finally, in real implementations, to reduce the memory requirement the "conceptual memory" M is mapped, (using hashing) into a physical memory M'.

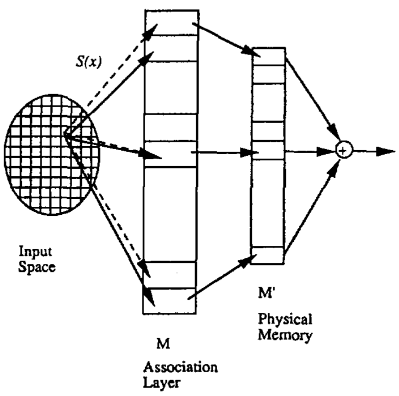

Figure 3. Diagram of a CMAC Neural Net

In order to complete the description of CMAC, a training rule must be provided. Consider, for simplicity the case where the output is a scalar u. Let $\underline{x}$ be the input, u_d the desired output and u^k the output of the net after the k^{th} iteration. Then, the simplest (one-shot error correction) CMAC rule evenly distributes the error among the contents of the A^* association cells excited by the input pattern, i.e.:

$$w_i^{k+1} = w_i^k + \frac{u_d - u^k}{A^*},\ i \in S(\underline{x}) \tag{3}$$

where $S(\underline{x})$ indicates the set of association cells excited by the input $\underline{x}$ and w_i is the content of the i^{th} cell. It has recently been shown [16] that, in the absence of collisions in the hashing map, this training scheme amounts to solving a system of simultaneous equations using the Gauss-Seidel iterative procedure. Therefore, the training is guaranteed to converge, provided that some mild structural restrictions are observed.

It has been argued that the CMAC structure has the potential to provide for good generalization properties through the overlapping of the sets of association cells. For a new input $\underline{x}$ which is close to the learned inputs $\underline{x}_1, \underline{x}_2, ... \underline{x}_k$ the association cells $S(\underline{x})$ will have some ovelap with the sets $S(\underline{x}_1), S(\underline{x}_2)...S(\underline{x}_k)$ and therefore a natural interpolation will occur. Let $u_1, ... u_k$ denote the corresponding learned outputs. The overlapping of the sets guarantees that the output $u = \text{CMAC}(\underline{x}) = \sum_{i \in S(\underline{x})} w_i$ will be close in some sense to the learned outputs u_i. However, it *does not necessarily* imply that u will belong to the convex hull of the points u_i as illustrated by the simple example shown in figure 4. There we have a situation where the set $S(\underline{x}) \subset S(\underline{x}_1) \cup S(\underline{x}_2)$. However, $u(\underline{x}) = 2$ which *does not* belong to the segment $\overline{u(\underline{x}_1)u(\underline{x}_2)}$. One can easily envision a situation where the generated control action (which has opposite direction to those corresponding to the closest training points ($\underline{x}_1$ and $\underline{x}_2$)) can move the system in the wrong direction. It follows that, unless provisions are made during training to ensure that a situation similar to this cannot arise, the generalization properties of CMAC are not enough to guarantee stability of the closed-loop system.

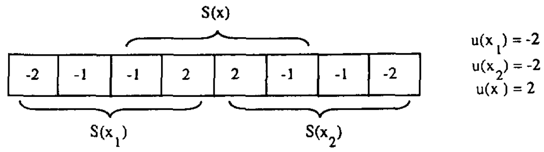

Figure 4. Example Illustrating Potential Problems when Using a CMAC

In this paper, to avoid this problem we will make the following assumption: There exist a set $\mathcal{B}$ of inputs such that i) every association cell in M is excited by at least one input $\underline{b} \in \mathcal{B}$ ii) for $\underline{b}_i, \underline{b}_j \in \mathcal{B}$, $S(\underline{b}_i) \bigcap S(\underline{b}_j) = \emptyset$, i.e, the set of association cells corresponding to the elements of $\mathcal{B}$ are mutually disjoint. The set $\mathcal{B}$ will be called a *basis* for CMAC.

When the training set is limited to the set $\mathcal{B}$, then the learning process (3) becomes trivial, converging in one iteration. It may be argued then that in this case the whole CMAC approach becomes trivial, since the storage capacity of the network appears to be under-utilized. However, note that two of the key features that make CMAC attractive, namely robustness to unit failure (obtained by distributing the information among several cells) and speed of computation remain intact. Furthermore, as we show in next section, by limiting the training set to $\mathcal{B}$, CMAC is guaranteed to produce an appropriate output for each possible point of the input space, thus making unnecessary the use of training patterns outside $\mathcal{B}$.

3.2 Designing a CMAC for Control Systems Applications

From the definitions of section 2.2 and the description above, it follows that a CMAC architecture is conceptually equivalent to considering a quantization χ of the input space and mapping all the elements of an equivalence class to the same set of association cells. Hence, a CMAC design can proceed as follows:

1) Select a closed cover $\mathcal{C}$ for $\mathcal{G}$. For simplicity, in the sequel we consider the case where $\mathcal{C}$ is formed by n-dimensional hypercubes, with sides of size δx parallel to the coordinate axes. Thus, each equivalence class $C(\underline{x}^-, \underline{x}^+)$ is formed by all the vectors $\underline{x}$ such that their components satisfy: $x_i \in [x_i^-, x_i^+)$, $i = 1, \ldots n$, where $x_i^+ - x_i^- = \delta x$.

2) Form a quantization by selecting one element from each equivalence class. We will select as representative of the class $C(\underline{x}^-, \underline{x}^+)$ the point $\underline{z} \triangleq \frac{1}{2}(x_1^- + x_1^+, x_2^- + x_2^+, \ldots, \underline{x}_n^- + \underline{x}_n^+)$. Thus, given an input vector $\underline{x}$, the corresponding $\underline{z} = \chi(\underline{x})$ can be easily found by discretizing each coordinate x_i of $\underline{x}$ with resolution δx.

The operation of the resulting CMAC can be described by the composition of the quantization and association maps, i.e:

$$\begin{aligned} A^* &= S \circ \chi(\underline{x}) \\ \underline{u} &\triangleq \text{CMAC}(\underline{x}) = \sum_{A^*} w_i \end{aligned} \tag{4}$$

To complete the description, we need to determine the number of association cells excited by each input pattern and the mapping S from input-space to association-cells space. We will choose A^* according to the following formula: $A^* \triangleq c_q^n$, were the integer c_q is a design parameter. Each input pattern will be mapped to the c_q^n cells starting at its equivalence class and extending c_q units in each direction, as shown in figure 5. Finally, we will choose as basis B a subset of $\{\underline{z}_i\}$ with coordinates spaced $c_q\delta x$. Hence, each group of A^* association cells corresponds to intervals of dimension δx along each coordinate axis, and contains c_q^n equivalence classes. It is easily seen that this choice of sets, along with rule (4) guarantees that for each point in input space (after discretizing) the output is given by the linear combination of the outputs of the corresponding basis points, i.e. if

$$\underline{z} = \chi(\underline{x}) = \sum_{i=1}^{2^n} \lambda_i \underline{b}_i; \lambda_i \geq 0; \sum \lambda_i = 1$$

then:

$$\text{CMAC}(\underline{z}) = \sum \lambda_i \text{CMAC}(\underline{b}_i) \tag{5}$$

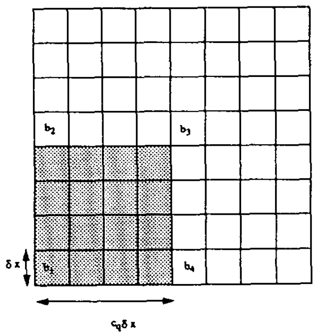

Figure 5. Determining the State Space Regions Corresponding to Each Set of Association Cells

4. Theoretical Results

In this section we present the basic theoretical results on solving the Robust Constrained Control Problem using CMAC. The main result of this section shows that by a proper choice of the parameters δx and c_q, the design procedure of section 3.2 leads to a controller guaranteed to satisfy the requirements specified in section 2.1. The proof proceeds along the following steps: i) obtain a lower bound on the amount that the norm of the present state of the system can be decreased in one step ii) use this bound to show that for each element of the basis there exist a control law that decreases the norm of the present state. iii) show that by proper choice of the basis this is true for any state. It follows then that the control law generated by CMAC is guaranteed to stabilize the system. These ideas are formalized in Lemmas 1, 2 and 3 (proved in the Appendix) and in Theorem 1.

We will address first the case were no collisions occur during hashing and then we will indicate how to modify our results to take the effects of hashing into account.

• **Lemma 1:** Consider a target set $O \subset \mathcal{G}$ and let O_1 be the subset formed by the equivalence classes entirely contained in O, i.e $O_1 \triangleq \cup \mathcal{C}_i,\ \mathcal{C}_i \subseteq O$ (see figure 6). Let:

$$\Lambda = \min_{\underline{x} \in \mathcal{G}-O_1} \left\{ \lambda > 0 : (\frac{1}{\lambda}\underline{x}) \in \partial\mathcal{G} \right\} \tag{6}$$

where $\partial\mathcal{G}$ denotes the boundary of the set $\mathcal{G}$. Then:

$$\min_{\underline{x}\in\mathcal{G}-O_1} \left\{ \|\underline{x}\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega} \{\|A_o\underline{x} + B\underline{u}\|_{\mathcal{G}} + \Delta_1\|\underline{x}\|_{\mathcal{G}}\} \right\} > \Lambda \min_{\underline{y}\in\partial\mathcal{G}} \left\{ 1 - \min_{\underline{u}\in\Omega} \{\|A_o\underline{x} + B_o\underline{u}\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\} \right\} \tag{7}$$

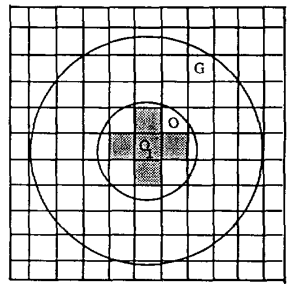

Figure 6: The Sets $\mathcal{G}$, O and O_1 (shadowed) of Lemma 1

• **Lemma 2:** Assume that:

$$1 - \max_{\|\underline{x}\|_{\mathcal{G}}=1} \left\{ \min_{\underline{u}\in\Omega} \{\|A_o\underline{x} + B_o\underline{u}\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\} \right\} = \delta > 0 \tag{8}$$

Then, for each element $\underline{b}_i \in B$, $\underline{b}_i \in \mathcal{G} - O_1$ there exist an admissible control law $\underline{u}_i$ such that $\|A(q)\underline{b}_i + B(q)\underline{u}\|_{\mathcal{G}} < \|\underline{b}_i\|_{\mathcal{G}}$ for all $q \in Q$.

• **Lemma 3:** Assume that condition (8) holds. Then, δx and c_q can be selected such that, for any input $\underline{x}_o \in \mathcal{G} - O$ the control action $\underline{u}_o$ generated by CMAC is admissible and such that $\|A(q)\underline{x}_o + B(q)\underline{u}_o\|_{\mathcal{G}} < \|\underline{x}_o\|_{\mathcal{G}}$

• **Theorem 1:** Assume that (8) holds and define the vector $\underline{\delta x} \triangleq (\delta x, \delta x, \dots \delta x)$. If the CMAC design parameters $\underline{\delta x}$ and c_q are selected such that:

$$\|\underline{\delta x}\|_{\mathcal{G}} < \frac{\Lambda\delta}{c_q - \frac{1}{2} + \frac{\|A\|_{\mathcal{G}}}{2}} \tag{9}$$

and the resulting CMAC is trained according to (3), then the resulting controller solves the robust constrained control problem.

Proof: Let $\underline{x}_o$ be an arbitrary initial condition in $\mathcal{G}$. If $\underline{x}_o \in O$ the theorem is trivial, so consider the case where $\underline{x}_o \notin O$. Then, from Lemma 3 it follows that, as long as $\underline{x}_k \notin O$, the sequence $\mathcal{U} = \{\underline{u}_o, \underline{u}_1 \dots\}$ of control actions generated by CMAC is admissible and such that:

$$\begin{aligned} \|\underline{x}_1\|_{\mathcal{G}} &< \|\underline{x}_0\|_{\mathcal{G}} - \mu \\ \|\underline{x}_2\|_{\mathcal{G}} &< \|\underline{x}_1\|_{\mathcal{G}} - \mu \\ &\vdots \\ \|\underline{x}_m\|_{\mathcal{G}} &< \|\underline{x}_{m-1}\|_{\mathcal{G}} - \mu \end{aligned} \tag{10}$$

where $\mu = \Lambda\delta - [c_q - 1 + \frac{1}{2}(1 + \|A\|_{\mathcal{G}})] \|\underline{\delta x}\|_{\mathcal{G}} > 0$. It follows then that there exists n_o such that $\underline{x}_{n_o} \in O$. Furthermore, since $\underline{x}_o \in \mathcal{G}$ then $\|\underline{x}_o\|_{\mathcal{G}} \leq 1$. Hence $\|\underline{x}_i\|_{\mathcal{G}} < 1$ which implies that $\underline{x}_i \in \mathcal{G}$ for all i. Therefore all the requirements specified in section 2.1 are satisfied ◇.

•**Corollary 1:** The size δx of the quantization introduced in Theorem 1 is inversely proportional to Λ. Hence, as the size of the target set gets smaller, the number of cells increases, while the size of the state-space region that they cover decreases. However, note that the target set O is achieved through a sequence of *intermediate* sets $O_i, i = 1, 2 \dots, n$ with $O_1 \equiv int(\mathcal{G})$ and $O_n \equiv O$. Since Λ in (6) can be thought of as a lower bound of the ratio of the norm of the next state of the system to the norm of the present state, it follows that to guarantee the practical stability of the closed-loop system, it suffices to choose:

$$\Lambda = \max_i \Lambda_i;\ \Lambda_i = \min_{\underline{x}\in\overline{O}_i - O_{i+1}} \left\{ \lambda > 0 : (\frac{1}{\lambda}\underline{x}) \in \partial\overline{O}_i \right\} \tag{11}$$

where $\overline{O}$ denotes the closure of O.

•**Corollary 2:** Consider now the effects of hashing. Assume that there is at most one collision per equivalence class. Then, the control action $\underline{u}_c$ generated in this case satisfies:

$$\|\underline{u}_c - \underline{u}_o\|_{\mathcal{G}} \le \frac{2(Bu)_{\max}}{c_q^n} \triangleq \delta u$$

where $\underline{u}_o$ is the control action generated in the absence of collisions and where:

$$(Bu)_{\max} \triangleq \max_{\underline{u}\in\Omega} \|B\underline{u}\|_{\mathcal{G}}$$

Hence collisions fit naturally in our formalism as another source of uncertainty. It follows that in this case stability can be guaranteed by selecting:

$$\|\delta\underline{x}\|_{\mathcal{G}} \le \frac{\Lambda\delta - \delta u}{c_q - \frac{1}{2} + \frac{\|A\|_{\mathcal{G}}}{2}}$$

and c_q large enough so $\delta u < \Lambda\delta$.

5. A Simple Example

Consider the problem of bringing the angular velocity of a spinning space station with a single axis of symmetry from an initial condition $\underline{x}_o$, $\|\underline{x}_o\|_2 = R_x$ to a final state such that $\|\underline{x}_f\|_2 \le R_f$. This situation can model the case where a CMAC-based controller is used to bring a system to some region (for instance a region where the constraints are not binding) where some relatively easy to design controller can take over. The nominal system can be represented by [17]:

$$A_o = \begin{pmatrix} \cos T & \sin T \\ -\sin T & \cos T \end{pmatrix} \; B_o = \begin{pmatrix} \sin T & (1-\cos T) \\ (\cos T - 1) & \sin T \end{pmatrix}$$

$$\mathcal{G} = \{\underline{x} \in R^2 : \|\underline{x}\|_2 \le R_x\},\ \Omega = \{\underline{u} \in R^2 : \|\underline{u}\|_2 \le 1\} \qquad (12)$$

where T is the sampling interval. Note that in this case $\|\underline{x}\|_{\mathcal{G}} = \frac{\|\underline{x}\|_2}{R_x}$. Let $\|\Delta A\|_2$ and $\|\Delta B\|_2\|\underline{u}\|_2$ be the uncertainty bounds introduced in (1). Then:

$$\begin{aligned}\|A_o\underline{x}_k + B_o\underline{u}\|_2 + \|\Delta B\|_2\|\underline{u}\|_2 &= (\|\underline{x}_k\|_2^2 + 2\underline{x}^T A_o^T B_o\underline{u} + \underline{u}^T B_o^T B_o\underline{u})^{\frac{1}{2}} \\ + \|\Delta B\|_2\|\underline{u}\|_2 &= (\|\underline{x}_k\|_2^2 + 2\underline{x}^T B_o^T\underline{u} + \alpha^2\|\underline{u}\|_2^2)^{\frac{1}{2}} + \|\Delta B\|_2\|\underline{u}\|_2\end{aligned} \qquad (13)$$

where $\alpha^2 \triangleq 2(1-\cos T)$. It can be easily shown that, as long as $\|\underline{x}\|_2 > \alpha$ and $\|\Delta B\|_2 < \alpha$, the minimum over $\underline{u} \in \Omega$ of (13) is achieved by selecting: $\underline{u}_o = \frac{-B_o\underline{x}_k}{\|B_o\underline{x}_k\|_2}$. For this value $\underline{u}_o$ we have:

$$\|A_o\underline{x}_k + B_o\underline{u}_o\|_2 + \|\Delta B\|_2\|\underline{u}_o\|_2 = \|\underline{x}_k\|_2 - \alpha + \|\Delta B\|_2 \qquad (14)$$

From (14) it follows that:

$$\begin{aligned}\delta &= 1 - \max_{\|\underline{x}\|_{\mathcal{G}}=1} \left\{ \min_{\underline{u}\in\Omega} \{\|A_o\underline{x} + B_o\underline{u}\|_{\mathcal{G}} + \|\Delta B\|_2\|\underline{u}_o\|_2 + \|\Delta A\|_2\} \right\} \\ &= \frac{\alpha - \|\Delta B\|_2 - \|\Delta A\|_2}{R_x}\end{aligned} \qquad (15)$$

Since in this case $\|.\|_{\mathcal{G}}$ is simply the euclidian norm scaled by R_x it follows that $\|A\|_{\mathcal{G}} = 1$. Hence, from (9) we have that:

$$\|\delta\underline{x}\|_{\mathcal{G}} = \frac{\|\delta\underline{x}\|_2}{R_x} < \frac{\Lambda\delta}{c_q} = \frac{\Lambda(\alpha - \Delta)}{c_q R_x} \qquad (16)$$

where $\Delta \triangleq \|\Delta A\|_2 + \|\Delta B\|_2$. Hence:

$$\delta x \le \frac{\Lambda(\alpha - \Delta)}{\sqrt{2}c_q} \qquad (17)$$

Since the norm of the present state of the system can be decreased at each stage by Λ it follows that δx should be selected such that:

$$\delta x \le R_x - \Lambda R_x = R_x(1-\Lambda) \qquad (18)$$

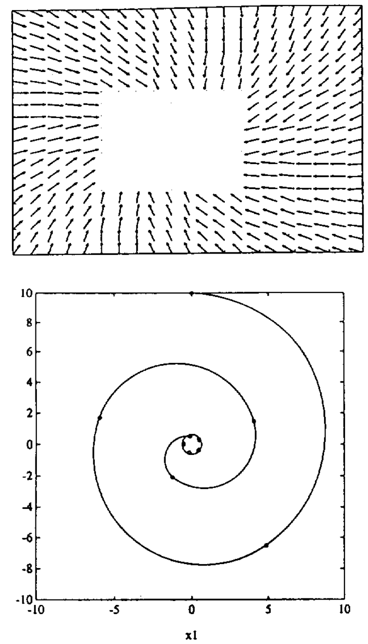

Figure 7: a) The Trained CMAC
b) Trajectory for the Simple Example with $\underline{x}_o = (0,\ 10)$

to guarantee that the present and next state of the system are in different equivalence classes. Hence, the region $\|\underline{x}\|_2 \le \alpha$ can be reached by designing a CMAC such that:

$$\frac{\Lambda(\alpha - \Delta)}{\sqrt{2}c_q} \le R_x(1-\Lambda) \iff \Lambda \le \frac{1}{1 + \frac{\alpha - \Delta}{R_x\sqrt{2}c_q}} \qquad (19)$$

In our case selecting $T = 2.5$ sec., $c_q = 3$ and $R_x = 10$ yields $\alpha = 1.898$ $\Lambda = 0.9615$ and $\delta x = 0.3848$

Figure 7 a) shows the contents of the association cells for the trained CMAC (i.e. the two dimensional control vector), the untrained square in the center corresponding to the region O_1. Figure 7 b) shows a sample trajectory driving the system from the boundary of the admissible region to the target set.

6. Conclusions

During the last few years, there has been considerable interest in the use of trainable controllers based upon the use of neuron like elements. These controllers can be trained, for instance by presenting several instances of *"desirable"* input–output pairs, to achieve good performance, even in the face of poor or minimal modeling. However, the use of neuromorphic controllers has been hampered by the facts that good performance hinges on the ability of the neural–net to generalize the input–output mapping to inputs that are not part of the training set. Through examples [6], it has been shown that neural–nets *do not* necessarily generalize well. Therefore, it follows that the *stability* properties of the closed–loop system are unknown. Moreover, it is conceivable that poor generalization capabilities may result in limit cycles or even in destabilizing control laws. In this paper we address these problems by proposing a controller based upon a neural–net (CMAC) with good generalization properties. Using the similarity of CMAC with quantization of state–space, we develop an analytical framework to investigate the properties of the resulting closed–loop system. These theoretical results are used to show that by incorporating partial a–priori information about the plant in the design process, the resulting controller is guaranteed to stabilize a family of plants. Perhaps the most valuable contribution of this paper results from the qualitative aspects of equation (9), that identify the factors that affect any controller based upon the quantization of state-space (independently of the specific implementation). Most notably, through the norm of the operator that appears in (9), it is possible to formalize the idea of *"poor"* modeling and to design a *"robust"* controller capable of accommodating modeling errors and disturbances.

There are several questions that remain open. The proposed design process is guaranteed to yield a stabilizing controller for all possible members of the family (P). However, since it is based upon a "worst-case" approach, it may achieve so at the expense of performance. Since one of the main reasons for using neural-net based controllers is their ability to yield good performance with imperfect models, the proposed off-line training may be combined with an on-line training (such as the one proposed in [18]) with the goal of improving performance. This research direction is currently being pursued.

Finally, the results of Theorem 1 that guarantee stability can be overly restrictive in some cases, since they result from a "worst-case" type analysis. A relaxed version of these conditions will be highly desirable.

7. References

[1]. M. Sznaier, "Set Induced Norm Based Robust Control Techniques," to appear in *Advances in Control and Dynamic Systems,* C. T. Leondes Editor, Academic Press, 1992.

[2]. A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst. Man. Cybern.,* Vol SMC-13, pp. 834-846, 1983.

[3]. T. Miller III, R. S. Sutton and P. J. Werbos editors, *Neural Networks for Control,* MIT Press, Cambridge, Mass., 1990.

[4]. D. Psaltis, A. Sideris and A. Yamamura, "A Multilayered Neural Network Controller," *IEEE Control Systems Magazine,* Vol 8, 3, pp 17–21, April 1988.

[5]. R. Hecht-Nielsen, *Neurocomputing,* Addison-Wesley, 1991.

[6]. Sideris, A., "Controlling Dynamical Systems by Neural Networks," *28th IEEE Conf. on Decision and Control,* Tampa, Florida, 1989.

[7]. R. A. Horn and C. R. Johnson, *Matrix Analysis,* Cambridge University Press, 1985.

[8]. B. R. Barmish, M. Corless and G. Leitmann, "A New Class of Stabilizing Controllers for Uncertain Dynamical Systems," *SIAM Journal Contr. Opt.,* Vol 21, 2, pp 246–255, 1983.

[9]. J. B. Conway, *A Course in Functional Analysis,* Vol 96 in Graduate Texts in Mathematics, Springer-Verlag, New-York, 1990.

[10]. J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Trans. ASME J. Dynamic Syst. Meas. Contr.,* Vol 97, pp. 220-227, 1975.

[11]. J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller (CMAC)," *Trans. ASME J. Dynamic Syst. Meas. Contr.,* Vol 97, pp. 228-233, 1975.

[12]. W. T. Miller III, "Real-Time Application of Neural Networks for Sensor Based Control of Robots with Vision," *IEEE Trans. Syst. Man. Cybern.,* Vol SMC-19, pp. 825-831, 1989.

[13]. W. T. Miller III, F. H. Glantz and L. G. Kraft, "CMAC: An Associative Neural Network Alternative to Backpropagation," *Proc IEEE,* Vol 78, pp. 1561-1567, 1990.

[14]. E. Ersu and J. Militzer, "Real-Time Implementation of an Associative Memory-Based Learning Control Scheme for Non-Linear Multivariable Processes," *Proc. 1st Measurements and Control Symposium on Applications of Multivariable Systems Techniques,* pp. 109-119, 1984.

[15]. J. Moody, "Fast Learning in Multi-Resolution Hierarchies," *Advances in Neural Information Processing Systems 1,* D. S. Touretzky Editor, Morgan Kauffman Publishers, 1989.

[16]. Y. F. Wong and A. Sideris, "Learning Convergence in the Cerebellar Model Articulation Controller," to appear in *IEEE Trans. on Neural Networks,* 1992.

[17]. M. Sznaier and M. Damborg, "Heuristically Enhanced Feedback Control of Constrained Discrete Time Linear Systems," *Automatica,* Vol 26, No 3, pp. 521-532, 1990.

[18]. C. S. Lin and H. Kim, "CMAC-Based Adaptive Critic Self-Learning Control," *IEEE Trans. on Neural Networks,* Vol 2, 5, pp. 530-533, 1991.

Appendix. Proofs of Lemmas 1, 2 and 3

Proof of Lemma 1: Given any $\underline{x} \in \mathcal{G} - O_1$ it can be expressed as $\lambda_o \underline{y}_o$ with $\underline{y}_o \in \partial\mathcal{G}$ and $0 < \lambda_o \leq 1$. Then:

$$
\begin{aligned}
\|\underline{x}\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega} & \left\{\|A_o\underline{x} + B\underline{u}\|_{\mathcal{G}} + \Delta_1\|\underline{x}\|_{\mathcal{G}}\right\} \\
&= \|\lambda_o\underline{y}_o\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\left\{\|A_o\lambda_o\underline{y}_o + B\underline{u}\|_{\mathcal{G}} + \Delta_1\|\lambda_o\underline{y}_o\|_{\mathcal{G}}\right\} \\
&\geq \|\lambda_o\underline{y}_o\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\left\{\|\lambda_o A_o\underline{y}_o + B(\lambda_o\underline{u})\|_{\mathcal{G}} + \lambda_o\Delta_1\|\underline{y}_o\|_{\mathcal{G}}\right\} \\
&= \lambda_o\left\{\|\underline{y}_o\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\left\{\|A_o\underline{y}_o + B\underline{u}\|_{\mathcal{G}}\right\} + \Delta_1\|\underline{y}_o\|_{\mathcal{G}}\right\} \\
&\geq \min_{\substack{\underline{y}\in\partial\mathcal{G} \\ \lambda\underline{y}\in\mathcal{G}-O_1}} \lambda\left\{\|\underline{y}\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\left\{\|A_o\underline{y} + B_o\underline{u}\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\|\underline{y}\|_{\mathcal{G}}\right\}\right\} \\
&\geq \Lambda \min_{\underline{y}\in\partial\mathcal{G}}\left\{1 - \min_{\underline{u}\in\Omega}\left\{\|A_o\underline{y} + B_o\underline{u}\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\right\}\right\}
\end{aligned}
\tag{A1}
$$

since $\|\underline{y}\|_{\mathcal{G}} = 1$, $\underline{y} \in \partial\mathcal{G}$ and $0 \leq \lambda_o \leq 1 \diamond$

Proof of Lemma 2: From the hypothesis and Lemma 1 it follows that:

$$
\begin{aligned}
&\max_{\underline{u}\in\Omega}\{\|\underline{b}_i\|_{\mathcal{G}} - \|A\underline{b}_i + B\underline{u}\|_{\mathcal{G}}\} = \|\underline{b}_i\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\{\|A\underline{b}_i + B\underline{u}\|_{\mathcal{G}}\} \\
&\geq \|\underline{b}_i\|_{\mathcal{G}} - \min_{\underline{u}\in\Omega}\{\|A_o\underline{b}_i + B_o\underline{u}_i\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\|\underline{b}_i\|_{\mathcal{G}}\} \\
&\geq \Lambda\left\{1 - \max_{\underline{x}\in\partial\mathcal{G}}\left\{\min_{\underline{u}\in\Omega}\|A_o\underline{x} + B_o\underline{u}\|_{\mathcal{G}} + \Delta_2(\underline{u}) + \Delta_1\right\}\right\} = \Lambda\delta > 0
\end{aligned}
\tag{A2}
$$

Proof of Lemma 3: From the definition of quantization it follows that there exists $\underline{z}_o \in \chi$ such that $\underline{x}_o \equiv \underline{z}_o$. Write $\underline{z}_o$ as $\sum_{i=1}^{2^n} \lambda_i \underline{b}_i$ where $0 \leq \lambda_i \leq 1$, $\sum \lambda_i = 1$ and where $\underline{b}_i$ are the vertices of the smallest hypercube containing $\underline{z}_o$. It can be easily shown, for instance by induction on n (the dimension of the space) that if $\lambda_i \neq 0$ then $\|\underline{z}_o - \underline{b}_i\|_\infty \leq (c_q - 1)\delta x$. Hence, for these λ_i $\|\underline{z}_o - \underline{b}_i\|_{\mathcal{G}} \leq (c_q - 1)\|\underline{\delta x}\|_{\mathcal{G}}$, where $\underline{\delta x} \stackrel{\Delta}{=} (\delta x, \delta x, \ldots \delta x)$. Denote by $\underline{u}(\underline{x})$ the control action generated by CMAC Then:

$$
\underline{u}(\underline{x}_o) = \underline{u}(\underline{z}_o) = \sum_{i=1}^{2^n} \lambda_i \underline{u}(\underline{b}_i) \in \Omega
$$

since Ω is convex. Consider now:

$$
\begin{aligned}
\|A\underline{z}_o + B\underline{u}\|_{\mathcal{G}} &= \|A\sum\lambda_i\underline{b}_i + B\sum\lambda_i\underline{u}(\underline{b}_i)\|_{\mathcal{G}} \leq \sum\lambda_i\|A\underline{b}_i + B\underline{u}(\underline{b}_i)\|_{\mathcal{G}} \\
&\leq \sum\lambda_i\left(\|\underline{b}_i\|_{\mathcal{G}} - \Lambda\delta\right) \\
&\leq \sum\lambda_i\left(\|\underline{z}_o\|_{\mathcal{G}} + (c_q - 1)\|\underline{\delta x}\|_{\mathcal{G}} - \Lambda\delta\right) \\
&= \|\underline{z}_o\|_{\mathcal{G}} + (c_q - 1)\|\underline{\delta x}\|_{\mathcal{G}} - \Lambda\delta
\end{aligned}
\tag{A3}
$$

Hence:

$$
\begin{aligned}
\|\underline{x}_o\|_{\mathcal{G}} - \|A\underline{x}_o + B\underline{u}_o\|_{\mathcal{G}} &\geq \|\underline{z}_o\|_{\mathcal{G}} - \frac{1}{2}\|\underline{\delta x}\|_{\mathcal{G}} - \|A\underline{z}_o + B\underline{u}_o\|_{\mathcal{G}} \\
&\quad - \frac{1}{2}\|A\|_{\mathcal{G}}\|\underline{\delta x}\|_{\mathcal{G}} \geq \Lambda\delta - \left[c_q - 1 + \frac{1}{2}(1 + \|A\|_{\mathcal{G}})\right]\|\underline{\delta x}\|_{\mathcal{G}}
\end{aligned}
\tag{A4}
$$

where $\|A\|_{\mathcal{G}} = \sup_{\|\underline{x}\|_{\mathcal{G}}=1} \|A\underline{x}\|_{\mathcal{G}}$. It follows that, if:

$$
\|\underline{\delta x}\|_{\mathcal{G}} < \frac{\Lambda\delta}{c_q - \frac{1}{2} + \frac{\|A\|_{\mathcal{G}}}{2}} \tag{A5}
$$

then $\|A\underline{x}_o + B\underline{u}\|_{\mathcal{G}} < \|\underline{x}_o\|_{\mathcal{G}} \diamond$.