Discrete-Time CMAC NN Control of Feedback Linearizable Nonlinear Systems Under a Persistence of Excitation

S. Jagannathan

Abstract—The local structure of CMAC neural networks (NN) result in better and faster controllers for nonlinear dynamical systems. A CMAC neural network-based discrete-time controller which linearizes the unknown multiinput and multioutput (MIMO) nonlinear system through feedback is presented. Control action is defined in order to achieve tracking performance for this unknown nonlinear system. An efficient and localized weight addressing scheme for the CMAC NN’s is described using an appropriate choice of the B-spline receptive field functions that form a basis. A uniform ultimate boundedness of the closed-loop system is given in the sense of Lyapunov using the persistency of excitation (PE) condition. Simulation results are shown to demonstrate the theoretical conclusions.

Index Terms—CMAC NN, discrete-time control, nonlinear control, persistence of excitation.

I. INTRODUCTION

In recent years, learning-based neural-network controllers have emerged as an alternative to adaptive control. Given a sufficient number of hidden layer neurons, it is well known that multilayer feedforward NN’s are theoretically capable of representing arbitrary mappings [3]. Therefore research in NN for control applications is being pursued by several researchers [1], [4]–[5], [7]–[9], [11]. For feedforward NN’s, since all the weights are updated during each learning cycle, the learning is essentially global in nature and slow. On the other hand, the CMAC NN’s utilize the information on local NN structure, and thus the learning is faster with a good function approximation. In addition, this function approximation generated by these CMAC NN’s is insensitive to the order of presentation of the training data [7].

The CMAC is a perceptron-like associative memory network with overlapping receptive fields. This network computes a nonlinear function over a domain of interest. The contents of these memory locations are referred as weights. The output of the CMAC NN is a linear combination of these weights [7]. While the advantages of using CMAC NN over conventional NN’s in many closed-loop applications are well known in literature [7]–[8], to our knowledge, there has been no study of using CMAC NN’s for feedback linearization of nonlinear systems in discrete-time which guarantee stability and bounded weight estimates. Note in [1] a CMAC-based NN controller is presented for a restricted class of nonlinear systems of the form

\[ x(k+1) = f(x(k)) + u(k). \]

This paper presents the design of a CMAC NN controller in discrete-time for feedback linearization of nonlinear systems which are expressed in affine form

\[ x(k+1) = f(x(k)) + g(x(k))u(k). \]

The controller is composed of CMAC NN’s incorporated into a dynamical system. It is shown through Lyapunov stability analysis that the tracking error is guaranteed to be bounded in the presence of bounded disturbances. However, the CMAC NN weights are guaranteed to be bounded only under a PE condition. This concludes that all the signals in the closed-loop system are guaranteed to be bounded.

A continuous-time nonlinear system is considered and a discrete-time controller is designed using CMAC NN’s and demonstrated. Finally, by appropriately choosing receptive field functions (eg. B-spline functions), these CMAC NN’s are shown to effectively provide universal controllers.

II. BACKGROUND ON CMAC NEURAL NETWORKS

Let \( R \) denote the real numbers, denote the real \( n \)-vectors, \( R^{n \times n} \) the real \( m \times n \) matrices. Let \( S \) be a compact simply connected set of \( R^n \). With maps \( f: S \rightarrow R^k \), define \( C^k(S) \) as the space such that \( f \) is continuous. We denote \( \| \cdot \| \) any suitable vector norm. Given a matrix \( A = [a_{ij}], A \in R^{n \times m} \), the Frobenius norm is denoted by \( \| \cdot \|_F \). Throughout this paper unless specified explicitly, is nothing but the vector 2-norm over the space defined by stacking the matrix columns into a vector, so that it is compatible with the vector 2-norm, that is \( \| A x \| \leq \| A \| \cdot \| x \| \).

Given \( x_k \in R^k, \) define \( x = [x_0, x_1, \cdots, x_N]^T, y = [y_0, y_1, \cdots, y_N]^T, \) and weight matrices \( W^T = [u_0]^T. \) Then the CMAC NN ideal output can be expressed as

\[ y = W^T \sigma(x), \]

with the vector of receptive field functions \( \sigma(z) = [\sigma(z_1), \cdots, \sigma(z_0)]^T \) defined for a vector \( z = [z_1, \cdots, z_1]^T. \)

Define the CMAC net actual output as

\[ g(k) = \hat{W}(k) \sigma(x(k)). \]

Then, for suitable CMAC NN approximation properties, \( \sigma(x(k)) \) must form a basis. For instance it is well-known in the literature that basis splines (B-splines), radial basis functions (RBF’s) form a basis. However, it is common in the literature to use B-splines as receptive field functions in
CMAC NN’s to gain computational advantage [7]. A spline is a function with an order \( n \) usually constructed using low order polynomial pieces, connected at break points with certain smoothness condition. The order \( n \) corresponds to the number of coefficients in the polynomial pieces.

Given \( N \) partitions of the real interval \( x \in [a, b] \), an \( n \)th order spline function \( \eta(x) \) of the form [7]

\[
f(x) = \eta(x) = \sum_{j=1}^{N+n+1} w_j B_{n,j}(x)
\]

(3)

can be constructed to approximate \( f(x) \) using linear combination of \( B \)-splines \( B_{n,j}(x) \) weighted by coefficients \( w_j \) called as weights. \( B \)-spline representation is preferred over other spline formulations due to their superior numerical and computational properties. The significant properties include positivity, compact support and normalization [7]. This sequence of normalized \( B \)-splines, \( B_{n,1}(x), B_{n,2}(x), \ldots, B_{n,N+n+1}(x) \) constructed on the break points or knot sets forms a basis set for all polynomial splines of order \( n \) on \( N \) partitions of the interval \( x \in [a, b] \). The \( B_{n,j}(x) \) can be obtained from the recurrence relation as [7]

\[
B_{n,j}(x) = \begin{cases} 
1 & \text{for } x \in [\theta_{j-1}, \theta_j) \\
0 & \text{otherwise}
\end{cases}
\]

(4b)

where

\[
B_{n,j}(x) = \left[ \frac{x - \theta_{j-1}}{\theta_j - \theta_{j-1}} B_{n-1,j-1}(x) + \frac{\theta_j - x}{\theta_j - \theta_{j-1}} B_{n-1,j}(x) \right]
\]

(4a)

The \( B \)-spline index \( j \) is associated with the region of local support, and once the partition number of an active interval has been determined for \( B \)-spline of order \( n \), higher order receptive functions can be employed to approximate not only functions but also function derivatives and this property is very useful for closed-loop control applications. Note the choice of the \( N_2 \) for a specified \( S \subset R^n \), and the CMAC neural net reconstruction error bound \( \varepsilon_N \) are current topics of research if the function to be approximated is rapidly varying. However, in [1] using linear approximation the choice of \( N_2 \) for a specified \( S \subset R^n \), maximum partition size \( \delta \) and Lipschitz constant \( L \) is given by

\[
N_2 = \frac{\varepsilon_N}{\delta L}.
\]

(4c)

Note multidimensional receptive fields are needed to approximate functions in multidimension. It is also shown that multidimensional receptive field functions generated through tensor product of one dimensional receptive field functions also satisfy the three important properties of positivity, compact support, and normalization. The result of using CMAC NN’s with \( B \)-spline function as receptive field functions is that only \( 2^n \) receptive fields are active at any time and therefore only \( 2^n \) weights needs to be adjusted to get exact function approximation. In the remainder of this paper, we use CMAC NN’s (4) for controls purposes. Note the stability results given in this paper is independent of the order of the \( B \)-spline function employed.

B. Stability of Dynamical Systems

In order to formulate the discrete-time controller, the following stability notions are needed. Consider the nonlinear system given by

\[
x(k+1) = f(x(k), u(k)), \quad y(k) = h(x(k))
\]

(5)

where \( x(k) \) is a state vector, \( u(k) \) is the input vector and \( y(k) \) is the output vector. The solution is uniformly ultimately bounded (UUB) if for all \( x(t_0) = x_0 \) there exists \( \varepsilon > 0 \) and a number \( N(\varepsilon, x_0) \) such that \( \| x(k) \| \leq \varepsilon \) for all \( k \geq k_0 + N \).

Consider now the linear discrete-time varying system given by

\[
x(k+1) = A(k)x(k) + B(k)u(k), \quad y(k) = C(k)x(k)
\]

(6)

with \( A(k), B(k), C(k) \) are appropriately dimensioned matrices.

**Lemma 2.1**: Define \( \psi(k_1, k_0) \) as the state-transition matrix corresponding to \( A(k) \) for the system (6), i.e., \( \psi(k_1, k_0) = \prod_{k_0}^{k_1} A(k) \). Then if \( \| \psi(k_1, k_0) \| < 1, \forall k_1, k_0 \geq 0 \), the system (6) is stable.

**Proof**: See [9].

C. Dynamical Systems of the \( n \)th Order MIMO

Consider an \( n \)th-order multiinput and multioutput discrete-time nonlinear system in the following form:

\[
x_1(k+1) = x_2(k) \\
x_{n-1}(k+1) = x_n(k) \\
x_n(k+1) = f(x(k)) + g(x(k))u(k) + d(k)
\]

(7)

where \( x(k) = [x_1(k), \ldots, x_n(k)]^T \) with \( x_i(k) \in R^{m_i}; i = 1, \ldots, n \) \( u(k) \in R^m \) \( d(k) \in R^m \) denotes a disturbance vector acting on the system at the instant \( k \) with \( \| d(k) \| \leq d_M \) a known constant. In addition, \( f: R^m \rightarrow R^m \) and \( g: R^{m \times m} \rightarrow R^{m \times m} \) are unknown smooth functions. Further

\[
\| g(x(k)) \| \geq g > 0, \quad \forall x
\]

(8)

with \( g \) a known lower bound and the inverse of \( g(x(k)) \) holds is assumed.

D. Tracking Problem

Feedback linearization will be used to perform output tracking, whose objective can be described as: given a desired trajectory in terms of output \( x_{out}(k) \) and its delayed values, find a control input \( u(k) \) so that the system tracks the desired trajectory with an acceptable bounded error in the presence of disturbances while all the states and controls remain bounded. In order to continue, the following mild assumptions are employed.

**Assumptions:**

1) The sign of \( g(x) \) is known.

2) The desired trajectory vector with its delayed values is assumed to be available for measurement and bounded by an upper bound.
E. Error Dynamics

Given the desired trajectory $x_{nd}(k)$ and its delayed values, define the tracking error as

$$e_n(k) = x_n(k) - x_{nd}(k).$$

(9)

It is typical in robotics to define a so-called the filtered tracking error, as $r(k) \in \mathbb{R}^n$ and given by

$$r(k) = e_n(k) + \lambda_1 e_{n-1}(k) + \cdots + \lambda_{n-1} e_1(k),$$

(10)

where $e_{n-1}(k), \ldots, e_1(k)$ are the delayed values of the error $e_n(k)$, and $\lambda_1, \ldots, \lambda_{n-1}$ are appropriate dimensioned constant matrices selected so that $|Z^{n-1} + \lambda_1 Z^{n-2} + \cdots + \lambda_{n-1}|$ is stable. Equation (10) can be further expressed as

$$r(k+1) = e_n(k+1) + \lambda_1 e_{n-1}(k+1) + \cdots + \lambda_{n-1} e_1(k+1).$$

(11)

Using (7) in (11), the dynamics of the $m$th-order MIMO system can be expressed in terms of the tracking error as

$$r(k+1) = f(x(k)) + g(x(k))u(k) + d(k) + Y_d$$

(12)

where

$$Y_d \equiv -x_{nd}(k+1) + \sum_{i=0}^{n-2} \lambda_{i+1} e_{n-i}.$$  

(13)

If we knew the exactly the nonlinear functions $f(x(k))$ and $g(x(k))$), and when no disturbances are present, the control input $u(k)$ can be selected as

$$u(k) = g(x(k))^{-1}(f(x(k)) + v(k))$$

where $v(k)$ another auxiliary input given by

$$v(k) = k_v r(k) - Y_d.$$

(15)

Then the filtered tracking error $r(k)$ goes to zero exponentially by properly selecting the gain matrix $k_v$. Since in our problem these functions are not known a priori, the control input $u(k)$ can be given by

$$u(k) = g(x(k))^{-1}(\hat{f}(x(k)) + v(k))$$

where the functional estimation errors are given by

$$\hat{f}(x(k)) = f(x(k)) - \hat{f}(x(k))$$

(19)

and

$$\hat{g}(x(k)) = g(x(k)) - \hat{g}(x(k))$$

(20)

This is an error system wherein the filtered tracking error is driven by the functional estimation error.

In the remainder of this paper, (18) is used to focus on selecting NN tuning algorithms that guarantee stability of the filtered tracking error $r(k)$. Then since (10), with the input considered as $r(k)$ and the output $e_n(k)$ and its delayed values, describes a stable system and standard techniques [10] guarantee that $e_n(k)$ and their delayed values exhibit stable behavior.

III. CMAC Neural-Network Design for Control

In the remainder of this paper, CMAC NN’s, which are linear in the tunable weights, are considered. Assume that there exists constant ideal weights $W_f$ and $W_g$ for both CMAC NN’s so that the nonlinear functions in (7) can be written as

$$f(x(k)) = W_f \phi_f(k) + \varepsilon_f(k)$$

(21)

and

$$g(x(k)) = W_g \phi_g(k) + \varepsilon_g(k)$$

(22)

where $\phi_f(k)$ and $\phi_g(k)$ provide a suitable basis (as B-Spline functions are employed) and $\|\varepsilon_f(k)\| \leq \varepsilon_{N_f}, \|\varepsilon_g(k)\| \leq \varepsilon_{N_g}$, with the bounding constants $\varepsilon_{N_f}, \varepsilon_{N_g}$ known. This allows one to select a simple CMAC NN structure based on a reduced basis and thereafter compensating for the increased magnitude of $\varepsilon_{N_f}, \varepsilon_{N_g}$ by using the gain matrix $k_v$. Note that $\varepsilon_{N_f}, \varepsilon_{N_g}$ become smaller as the number of partition interval becomes greater with more number of B-spline functions to cover the interval. Here, two CMAC NN’s one for estimating $f(x(k))$ and the other for $g(x(k))$ are used.

Let $x \in U$ a compact subset of $\mathbb{R}^n$, Assume that $h(x(k)) \in C^\infty[U]$ i.e., a smooth function $U \rightarrow \mathbb{R}$, so that the Taylor series expansion $h(x(k))$ exists. One can derive using (10) that $\|x(k)\| \leq d_{01} + d_{11} \|r(k)\|$. Then using the bound on $x(k)$ and expressing $h(x(k))$ as (22), yields an upper bound on $h(x(k))$ as

$$\|h(x(k))\| = \|W_h \phi(k) + \varepsilon_h(k)\| \leq C_{01} + C_{11} \|r(k)\|$$

(23)

with $C_{01}$ and $C_{11}$ computable constants. In addition, the hidden layer activation functions, such as radial basis functions, sigmoid, etc. are bounded by an known upper bound

$$\|\phi(k)\| \leq \phi_{\text{max}}.$$  

(24)
A. Structure of the CMAC NN Controller

Define the functional estimates given by

\[ \hat{f}(x(k)) = \hat{W}_f^T(k)\phi_f(k), \]

and

\[ \hat{g}(x(k)) = \hat{W}_g^T(k)\phi_g(k), \]

with \( \hat{W}_f(k) \) and \( \hat{W}_g(k) \) represent the actual weights. Then the closed-loop filtered dynamics (18) become

\[ r(k+1) = k_v r(k) + \hat{W}_f^T(k)\phi_f(k) + \hat{W}_g^T(k)\phi_g(k)u(k) \\
+ (e_r(k) + e_g(k)u(k)) + d(k) \]

where

\[ \hat{W}_f(k) = W_f - \hat{W}_f(k), \]

and

\[ \hat{W}_g(k) = W_g - \hat{W}_g(k). \]

The proposed CMAC NN controller structure is shown in Fig. 1. The output of the plant is processed through a series of delays in order to obtain the past values of the output, and fed as inputs to both the CMAC NN’s for constructing \( \hat{f}(x(k)) \) and \( \hat{g}(x) \) so that the nonlinear functions in (7) can be suitably approximated. Thus, the CMAC NN controller derived in a straightforward manner using filtered error notion naturally provides a dynamical structure.

B. Proposed Controller

In order to guarantee the boundedness of \( \hat{g}(x(k)) \) away from zero for all well-defined values of \( x(k) \), \( \hat{W}_f(k) \), and \( \hat{W}_g(k) \), the control input in (17) is selected in terms of another control input \( u_c(k) \) and a robustifying term \( u_r(k) \) as

\[ u(k) = \begin{cases} 
  u_c(k) + \frac{u_r(k) - u_c(k)}{2} e^{-\|u_c(k)\|-s} & \text{if } I = 1 \\
  u_r(k) - \frac{u_r(k) - u_c(k)}{2} e^{-\|u_c(k)\|-s} & \text{if } I = 0 
\end{cases} \]

where

\[ u_c(k) = \hat{g}(x(k))^{-1}(-\hat{f}(x(k)) + v(k)) \]

\[ u_r(k) = -\mu \frac{\|u_c(k)\|}{\gamma} \text{sgn}(r(k)). \]

the Indicator \( I \) in (30) is

\[ I = \begin{cases} 
  1, & \text{if } \hat{g}(x(k)) \geq \gamma \text{ and } ||u_c(k)|| \leq s \\
  0, & \text{otherwise} 
\end{cases} \]

with \( \gamma < \ln(2/s), \mu > 0, \) and \( s > 0 \) are design parameters. These modifications in the control input is necessary in order to ensure that the functional estimate \( \hat{g}(x(k)) \) is bounded away from zero.

C. Weight Updates for Guaranteed Performance

It is required to demonstrate that the tracking error \( r(k) \) is suitably small and that the CMAC NN weights \( \hat{W}_f(k) \) and \( \hat{W}_g(k) \) remain bounded. In order to proceed, the following are needed.

**Lemma 3.1:** If \( A(k) = I - \alpha (\phi(k))\phi^T(k) \) in (6), where \( 0 < \alpha < 1 \) and \( \phi(k) \) is a vector of basis functions, then \( ||f(k, k_0)|| < 1 \) is guaranteed if there is an \( L > 0 \) such that \( \sum_{k=k_0}^{k+L-1} \phi(k)\phi^T(k) > 0 \) for all \( k \). Then Lemma 2.2 guarantees the stability of the system (6).

**Proof:** See [9].

**Definition 3.2:** An input sequence \( x(k) \) is said to be persistently exciting if there are \( \lambda > 0 \) and an integer \( k_1 \) such that

\[ \lambda_{\min} \left[ \sum_{k=k_0}^{k+L-1} \phi(k)\phi^T(k) \right] > \lambda, \quad \forall k_0 \geq 0 \]

where \( \lambda_{\min}(P) \) represents the smallest eigen value of \( P \).

**Note:** PE is exactly the stability condition needed in Lemma 3.1.

In the following, it is taken that the NN reconstruction error bound \( \varepsilon_N f, \varepsilon_N g \) and the disturbance bound \( d_M \) are non zero. Let the NN weight updates for \( f(x(k)) \) be given by

\[ \hat{W}_f(k+1) = \hat{W}_f(k) + \alpha \phi_f(k)rT(k+1) \]
and NN weight updates for \( g(x(k)) \) are provided by
\[
\dot{W}_g(k+1) = \begin{cases}
\dot{W}_g(k) + \beta g_y(k)u_y(k)u_T(k+1), & \text{if } I = 1 \\
\dot{W}_g(k), & \text{otherwise}
\end{cases}
\]
where \( \alpha > 0, \beta > 0 \) are adaptation gains or learning rate parameters.

Theorem 3.3: Assume that the feedback linearizable system has a representation in the form as in (7) and the control input is given by (30). Assume that the vectors \( \phi(k), g_y(k), u_y(k) \) of the NN for \( f(x(k)) \) and \( g(x(k)) \), respectively, be persistently exciting. Then the filtered tracking error \( r(k) \) and the weight estimates \( \dot{W}_f(k) \) and \( \dot{W}_g(k) \) are UUB provided the following conditions hold:
\[
\frac{\beta}{\mu}||g_y(k)u_y(k)||^2 = \beta||\phi_y(k)||^2 = \beta l^2_{\max} < 1 
\]
\[
\eta < 1 
\]
where
\[
\eta = \begin{cases}
\alpha||\phi_f(k)||^2 + \beta||g_y(k)u_y(k)||^2 = \alpha l^2_{\max} + \beta l^2_{\max}, & \text{for } I = 1, \\
\alpha||\phi_f(k)||^2 = \alpha l^2_{\max}, & \text{for } I = 0.
\end{cases}
\]
\[
\max(a_4, b_0) < 1
\]
with \( a_4, b_0 \) are design parameters chosen using the gain matrix \( k_{\text{max}} \) and the relationship is given during the proof.

Proof: See the Appendix.

Remarks 1: For practical purposes, (A.45) can be considered as bounds for \( r(k), \dot{W}_f(k), \) and \( \dot{W}_g(k) \).

Remarks 2: The CMAC NN reconstruction errors and the bounded disturbances are all embodied in the constants given by \( \delta_f, \delta_g \) and \( \delta_y \). Note that the bound on the tracking error may be kept small if the closed-loop poles are placed closer to the origin.

Remarks 3: If the switching parameter \( s \) is chosen small, it will limit the control input and results in a large tracking error which gives undesirable closed-loop performance. Larger value of \( s \) results in the saturation of the control input \( u(k) \).

Remarks 4: Uniform ultimate boundedness of the closed-loop system is shown without making assumptions on the initial CMAC NN weight values. The CMAC NN’s can be easily initialized as \( \dot{W}_f(k) = 0 \) and \( \dot{W}_g(k) > g^{-1}(g) \). In addition, the CMAC NN’s presented here do no need initial off-line learning phase. No assumptions such as the existence of an invariant set, region of attraction, or a feasible region is needed.

IV. SIMULATION RESULTS

Consider a planar two link robot arm whose state equations in continuous-time can be written as
\[
\begin{align*}
\dot{X}_1 &= X_2 \\
\dot{X}_2 &= F(X_1, X_2) + G(X_1, X_2)U
\end{align*}
\]
where \( X_1 = [q_1, q_2]^T \) being the joint angles, \( U \) be the input vector and the nonlinear functions are described by
\[
F(X_1, X_2) = -[M(X_1)]^{-1}[V(X_1, X_2)X_2 - G_1(X_1)]
\]
\[
G(X_1, X_2) = [M(X_1)]^{-1}
\]
with \( M(X_1) \) being the inertia matrix, \( V(X_1, X_2) \) being the coriolis and \( G_1(X_1) \) centripetal vector, and being the gravitational vector. The actual elements of the inertia matrix, coriolis, centripetal, and gravitational vector are given in [10]. The system when discretized using Euler’s approximation can be rewritten as
\[
\begin{align*}
Z_1(k+1) &= Z_2(k) \\
Z_2(k+1) &= F_2(Z_1(k), Z_2(k)) + G_2(Z_1(k), Z_2(k))U(k)
\end{align*}
\]
where
\[
F_2(Z_1(k), Z_2(k)) = -Z_4(k) + 2Z_2(k) + T^2F(Z_1(k), Z_2(k))
\]
and
\[
G_2(Z_1(k), Z_2(k)) = T^2G(Z_1(k), Z_2(k))
\]
with the transformation matrix given by
\[
\begin{bmatrix}
Z_1(k) \\
Z_2(k)
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
1 & T
\end{bmatrix}
\begin{bmatrix}
X_1(k) \\
X_2(k)
\end{bmatrix}.
\]

Note that (38) can be expressed as (7) by using the transformation matrix (44).

A sampling interval of 10 ms was chosen. The arm parameters are taken as \( l_1 = l_2 = 1 \), \( m_1, m_2 = 1 \) kg, \( m_2 = 2.3 \) kg. The desired trajectory was selected as \( q_1(t) = 0.3 \sin 2\pi kT \), for the case of joint 1 and \( q_2(t) = 0.3 \cos 2\pi kT \) for the joint 2. The input space was partitioned in a grid of size 0.25 and receptive fields are selected to cover the input space \([0.5, 0.5] \times [0.5, 0.5] \) along each input dimension. The initial conditions for both the states and the CMAC NN weight estimates \( f(x) \) are taken to be zero. It is seen that although 625 weights are needed to define the outputs, only \((2 \times 2^3)\) weights are updated at any given instant.

The initial conditions for the parameter estimates for \( g(x) \) are initialized at 0.5. Here \( g \) is chosen to be 1.0. Design parameters are set to \( s = 10, \gamma = 0.05, k\alpha = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \mu = 4, \alpha = \beta = \rho = \delta = 0.1 \). These design parameters are selected by satisfying the conditions (37).

Fig. 2 shows the desired and actual outputs for the MIMO system using the controller derived in this paper and given by (39). Fig. 3 shows the performance of the PD controller alone without the NN system. From the results, it can be inferred
that the performance of the NN controller is impressive even though the NN controller knows none of the dynamics a priori.

V. CONCLUSIONS

Discrete-time controller design for closed-loop control applications using CMAC neural networks was presented. This CMAC neural network-based controller which linearizes the MIMO nonlinear system through feedback achieves a desired tracking performance. The structure of the CMAC NN controller was derived using a filtered error notion. A PE condition was defined for the CMAC NN inputs and used to show the boundedness of CMAC NN weight estimates. A uniform ultimate boundedness of the closed-loop system was given in the sense of Lyapunov.

Note that it is extremely difficult to verify or guarantee the PE condition even for linear systems. Therefore future work will involve in the modification of the update laws so that the PE condition is relaxed. Finally, simulation results are provided to demonstrate the theoretical investigations.

APPENDIX

Proof of Theorem 3.3: Let the Lyapunov function candidate be given by

\[
V = v^T(k)r(k) + \frac{1}{\alpha} \text{tr}(\hat{W}_f(k)^T\hat{W}_f(k)) + \frac{1}{\beta} \text{tr}(\hat{W}_g(k)^T\hat{W}_g(k)),
\] (A.1)
The first difference of (A.1) will be investigated in two mutually exclusive regions and is given by

$$\Delta V = \Delta V_1 + \Delta V_2 + \Delta V_3. \tag{A.2}$$

where

$$\Delta V_1 = r^T(k+1)r(k+1) - r^T(k)r(k) \tag{A.3}$$

$$\Delta V_2 = \frac{1}{\alpha} \text{tr} \left( \tilde{W}_f(k+1)\tilde{W}_f(k+1) - \tilde{W}_f(k)\tilde{W}_f(k) \right) \tag{A.4}$$

and

$$\Delta V_3 = \frac{1}{\beta} \text{tr} \left( \tilde{W}_g(k+1)\tilde{W}_g(k+1) - \tilde{W}_g(k)\tilde{W}_g(k) \right). \tag{A.5}$$

**Region I:** $\|\hat{g}(x(k))\| \geq g$ and $\|u_c(k)\| \geq s_r$

The filtered error dynamics can be rewritten as

$$r(k+1) = k_\alpha r(k) + (f(x(k)) - \hat{f}(x(k)))$$
$$+ (g(x(k)) - \hat{g}(x(k)))u_c(k) + d(k)$$
$$+ gu_d(k). \tag{A.6}$$

where $u_d(k) = u(k) - u_c(k)$.

Substituting (25) and (26) in (A.6), one obtains

$$r(k+1) = k_r r(k) + \tilde{W}_f^T(k)\phi_f(k) + \tilde{W}_g^T(k)\phi_g(k)$$
$$\cdot u_c(k) + \varepsilon(k) + d(k) + gu_d(k) \tag{A.7}$$

where

$$\varepsilon(k) = \varepsilon_f(k) + \varepsilon_g(k)u_c(k). \tag{A.8}$$
Equation (A.7) can be rewritten as
\[
\begin{align*}
r(k+1) &= k_r r(k) + \bar{W}^T_f(k)\phi_f(k) + \bar{W}^T_g(k)\phi_g(k)u_c(k) \\
&\quad + \varepsilon(k) + d(k) + gu_{d}(k) \\
&= k_r r(k) + \bar{z}^T_f(k) + \bar{z}^T_g(k) + \varepsilon(k) + d(k) \\
&\quad + gu_{d}(k) \\
&\quad + (A.9)
\end{align*}
\]
where
\[
\begin{align*}
\bar{z}_f(k) &= \bar{W}^T_f(k)\phi_f(k), \\
\bar{z}_g(k) &= \bar{W}^T_g(k)\phi_g(k)u_c(k) = \bar{W}^T_g(k)\phi_f(k).
\end{align*}
\]
Similarly, the error in dynamics for the weight update laws are given for this region as
\[
\begin{align*}
\bar{W}_f(k+1) &= (I - \alpha_0\phi_f(k)\phi^T_f(k))\bar{W}_f(k) - \alpha_0\phi_f(k)k_r r(k) \\
&\quad + \bar{z}_f(k) + gu_{d}(k) + \varepsilon(k) + d(k) \\
&\quad + (A.12)
\end{align*}
\]
\[
\begin{align*}
\bar{W}_g(k+1) &= (I - \alpha_0\phi_g(k)\phi^T_g(k))\bar{W}_g(k) - \alpha_0\phi_g(k)k_r r(k) \\
&\quad + \bar{z}_g(k) + gu_{d}(k) + \varepsilon(k) + d(k). \\
&\quad + (A.13)
\end{align*}
\]
Substituting (A.6)–(A.13), combining the three terms and rewriting, completing the squares and simplifying, one obtains
\[
\begin{align*}
\Delta V &= -(1 - a_4)\|r(k)\|^2 + 2a_5\|r(k)\| + a_6 - (1 - \eta) \\
&\quad \cdot \left( \left\| \varepsilon_f(k) + \varepsilon_g(k) \right\| \right)^2 \\
&\quad + \frac{\eta}{1 - \eta}(k_r r(k) + gu_{d}(k) + \varepsilon(k) + d(k)) \\
&\quad + (A.14)
\end{align*}
\]
where
\[
\eta = \alpha_0\phi^2_{\text{max}} + \beta_0\phi^2_{\text{max}}.
\]
and
\[
\begin{align*}
a_4 &= 1 + \eta + \frac{\eta}{1 - \eta}, \\
a_2 &= \eta + \frac{\eta}{1 - \eta}, \\
a_3 &= \left(1 + \eta + \frac{\eta}{1 - \eta}\right).
\end{align*}
\]
Now applying the condition on the function \(g(x)\) by taking \(g(x)\) in the compact set, one can write
\[
\|g(x)\| \geq C_{01} + C_{12}\|r(k)\| \\
\]
with \(C_{01}, C_{12}\) domain specific constants.

Now in this region, the bound on \(u_d\) can be obtained as
\[
\|u_d(k)\| \leq \|u(k) - u_c(k)\| \\
\leq \left\|u(k) - u_c(k) - \varepsilon(\|u(k)\| - \alpha) \right\|.
\]
\[
\|
\]
In this region since \(\|u_c(k)\| \leq s\), and the other input \(u_r\) is given by (32), the bound in (A.20) can be obtained as a constant since all the terms on the right side are bounded and this bound is given by
\[
\|u_d(k)\| \leq C_2.
\]
Now the bound for \(gu_d(k)\) is given by
\[
\|gu_d(k)\| \leq C_2(C_{01} + C_{12}\|r(k)\|) \\
\leq C_0 + C_1\|r(k)\|. \\
\]
Using the bound for \(gu_d(k)\) in (A.14), the first difference of the Lyapunov function (A.2) is rewritten as
\[
\Delta V = -(1 - a_4)\|r(k)\|^2 + 2a_5\|r(k)\| + a_6 - (1 - \eta) \\
\quad \cdot \left( \left\| \varepsilon_f(k) + \varepsilon_g(k) \right\| \right)^2 \\
\]
with the bound for \(\varepsilon(k)\) can be obtained as
\[
\|\varepsilon(k)\| \leq \|\varepsilon_f(k)\| + \|\varepsilon_g(k)\| \\
\leq (\varepsilon_{\text{NL}} + \delta_{\text{NL}}) \\
\leq \varepsilon_N
\]
\[
\]
where
\[
\begin{align*}
a_4 &= a_4\phi^2_{\text{max}} + 2a_5\phi_{\text{max}} + a_6C_1, \\
a_5 &= a_5\phi^2_{\text{max}}(\varepsilon_N + d_M + C_0) + a_6C_1(\varepsilon_N + d_M) \\
&\quad + a_6C_0C_1, \\
a_6 &= a_6C_0^2 + 2a_5C_0(\varepsilon_N + d_M) + (\varepsilon_N + d_M)^2.
\end{align*}
\]
The second term in (A.23) is always negative as long as the condition (37b) holds. Since \(a_4, a_5,\) and \(a_6\) are positive constants, \(\Delta V \leq 0\) as long as (37c) holds and
\[
\|r(k)\| > \frac{1}{(1 - a_4)}[a_3 + \sqrt{a_3^2 + a_6(1 - a_4)}] \\
\|r(k)\| > \delta_{\text{NL}}.
\]
\[
\|
\]
Finally, the error in weight updates given by (35) and (36) are used to show the boundedness of weights. Here the PE condition is necessary but not sufficient to prove the boundedness of the error in weight updates. It is further assumed that the initial weight estimation errors and for both CMAC NN \(\bar{W}_f\) and \(\bar{W}_g\) are bounded.

Normally, the error in weight updates given by (35) and (36) are used to show the boundedness of weights. Here the PE condition is necessary but not sufficient to prove the boundedness of the error in weight updates. It is further assumed that the initial weight estimation errors are bounded. Then, the CMAC weight estimates bounded using the updates (35) and (36).
Region II: $$\| \dot{g}(x(k)) \| < g$$ or $$\| u_c(k) \| > s,$$

Since the input $$u_c(k)$$ may not be defined in this region, because of notional simplicity we will use it in the form of either $$\dot{g}(k)u_c(k)$$ or $$u_c(k)e^{-\gamma l_{u_c}(k)}\|u_c(k)\|^{-s}$$. Therefore, in this region, the tracking error system in (A.7) is rewritten as

$$r(k+1) = k_{r}r(k) + \bar{e}_{f}(k) + g(k)u(k) + \dot{g}(k)u_c(k) + \varepsilon_{f}(k) + d(k)$$

$$= k_{r}r(k) + \bar{e}_{f}(k) + \bar{u}_{d}(k) + \varepsilon_{f}(k) + d(k)$$  \hspace{1cm} (A.30)

where

$$\bar{u}_{d}(k) = g(k) = \dot{g}(k)u_c(k).$$  \hspace{1cm} (A.31)

Note the extremum of $$y e^{-\gamma y}$$ for $$\forall y > 0$$ can be found as a solution to the following equation:

$$\frac{\partial(y e^{-\gamma y})}{\partial y} = (1 - \gamma y)e^{-\gamma y} = 0$$  \hspace{1cm} (A.32)

which is $$y = (1/\gamma)^{\frac{1}{\gamma}}$$, and it is a maximum. Evaluating the function at this point yields an upper bound $$1/\gamma^{2}$$. This bound is used in the forthcoming set of equations.

Let us compute the bound for $$g(k)$$ and $$\dot{g}(k)u_c(k)$$. Take the cases in this region when $$\| u_c(k) \| \leq s$$ and $$\| u_c(k) \| > s$$. The bound on $$u(k)$$ from (30) can be written for this region as

$$\| u(k) \| \leq \frac{u_p(k) - u_c(k)}{2} e^{-\gamma l_{u_c}(k)} - s.$$  \hspace{1cm} (A.33)

Using (32) for $$u_c(k)$$, (A.33) can be written as

$$\| u(k) \| \leq \frac{1}{2} \left( \frac{\mu}{\varepsilon} \| u_c(k) \| + u_c(k) \right) e^{-\gamma l_{u_c}(k)} - s.$$  \hspace{1cm} (A.34)

The input is bounded above by some positive constant in both the cases. Note here for simplicity we denote the constant by the same variable $$d_1$$. Now the bound for $$g(k)$$ can be obtained using (A.22) as

$$\| g(k) \| \leq (C_0 + C_1 \| r(k) \|) \| u(k) \| \leq C_0 + C_1 \| r(k) \|. \hspace{1cm} (A.35)$$

Similarly, the bound for $$\dot{g}(x(k))u_c(k)$$ can be obtained as

$$\| \dot{g}(x(k))u_c(k) \| \leq g s, \text{ when } \| u_c(k) \| \leq s$$

$$\leq \frac{g}{\gamma}, \text{ when } \| u_c(k) \| > s$$

and

$$\leq C_2. \hspace{1cm} (A.36)$$

for some positive constant in both cases.

Using the individual upper bounds of $$g(x(k))u(k)$$ and $$\dot{g}(k)u_c(k)$$, the upper bound for both can be obtained as

$$\| g(k)u(k) \| = \| g(x(k))u(k) - \dot{g}(x(k))u_c(k) \| \leq C_0 + C_1 \| r(k) \| + C_2$$

$$\leq C_3 + C_4 \| r(k) \|$$ \hspace{1cm} (A.37)

where $$C_3 = C_0 + C_2$$, and $$C_4 = C_4$$.

The first difference of the Lyapunov function can be obtained after substituting for $$g(k)u(k)$$ as

$$\Delta V = -\left( 1 - \lambda_{0} \right)\| r(k) \|^2 + 2b_2 \| r(k) \| + b_2 - (1 - \eta) \left( \frac{\bar{e}_{f}(k)}{(1 - \eta)}(k_{r}r(k) + g(k)u(k) + \varepsilon_{f}(k) + d(k)) \right)^2$$

$$+ (d(k))^2$$ \hspace{1cm} (A.38)

where $$\eta$$ is given in (37b) and

$$b_0 = k_{r}^2 + 2C_4(C_4 + k_{r}^2) + C_4 + k_{r}^2$$

$$b_1 = C_3(k_{r} + k_{r}^2) + C_3(C_4 + k_{r}^2) \frac{\varepsilon_{f}(k)}{(1 - \alpha_{f}^2)}$$

$$+ (C_3 + 2C_3)\varepsilon_{f}(k) + d_M) \frac{\varepsilon_{f}(k)}{(1 - \alpha_{f}^2)}$$

$$b_2 = 2C_3^2 + 2C_3\varepsilon_{f}(k) + d_M) + (\varepsilon_{f}(k) + d_M)^2$$

$$+ (C_4^2 + 2C_3\varepsilon_{f}(k) + d_M) + (\varepsilon_{f}(k) + d_M^2) \frac{\varepsilon_{f}(k)}{(1 - \alpha_{f}^2)}$$  \hspace{1cm} (A.40)

and

$$\| \varepsilon_{f}(k) \| \leq \varepsilon_{f}(k). \hspace{1cm} (A.42)$$

The second term in (A.38) is always negative as long as the condition (37) holds. Since $$b_0, b_1, b_2$$ are positive constants, $$\Delta V \leq 0$$ as long as

$$\| r(k) \| > \frac{1}{1 - b_0} \left[ b_1 + \sqrt{b_2^2 + b_2(1 - b_0)} \right]$$

$$\| r(k) \| > \delta_2. \hspace{1cm} (A.43)$$

$$\|
\sum_{k=0}^{\infty} \Delta V(k) \| = \| V(\infty) - V(0) \| < \infty \text{ since } \Delta V \leq 0 \text{ as long as (37) hold. This implies that the tracking error } r(k) \text{ is UUB for all } k > 0 \text{ and it remains to show that the weight estimation errors } \tilde{W}_f \text{ or equivalently } \hat{W}_f \text{ are bounded.}$$

To show the boundedness of the weight estimates in region II for the function $$f(x(k))$$, the dynamics relative to error in weight estimates for the function $$f(x)$$ using (35) for this region are given by

$$\tilde{W}_f(k+1) = (I - \alpha_{f}(k)\phi_f^2(k))\tilde{W}_f(k) - \alpha_{f}(k)(k_{r}r(k) + g(k)u(k) + \varepsilon_{f}(k) + d(k))$$

$$= (I - \alpha_{f}(k)\phi_f^2(k))\tilde{W}_f(k) - \alpha_{f}(k)(k_{r}r(k) + C_3 + C_4\| r(k) \| + \varepsilon_{f}(k) + d(k))$$ \hspace{1cm} (A.44)

where the tracking error $$r(k)$$ is shown to be bounded. Applying the PE condition (34), and Lemma 3.1, the boundedness of $$\tilde{W}_f(k)$$ and hence $$\hat{W}_f(k)$$ are assured. Let us denote this bound as $$\delta_2$$. The weight estimate bounds for $$f(x(k))$$ and $$g(x(k))$$ in region I are denoted here as $$\delta_{f_1}$$ and $$\delta_{g_1}$$, respectively, and the value can be obtained from (35) and (36) in Section IV.
Reprise: Combining the results from regions I and II, one can readily set
\[
\delta_r = \max \{\delta_{r1}, \delta_{r2}\}, \quad \delta_f = \max \{\delta_{f1}, \delta_{f2}\}, \quad \text{and} \quad \delta_g.
\]  
(A.45)
Thus for both regions, if \[|r(k)| > \delta_r\], then \[\Delta V \leq 0\] and \[u(k)\] is bounded.

REFERENCES


S. Jagannathan was born in Madurai, India. He received the Bachelor’s degree in electrical engineering at Anna University, Madras, India, in 1986, the Master of Science degree in electrical engineering from the University of Saskatchewan, Saskatoon, Canada, in 1989, and the Ph.D. degree in electrical engineering from the Automation and Robotics Research Institute, University of Texas at Arlington in 1994.

During 1986 to 1987, he was a Junior Engineer at Engineers India Limited, New Delhi, acted as a Research Associate between 1990 and 1991 at the University of Manitoba, Winnipeg, Canada, and worked at Systems and Controls Research Division, in Caterpillar Inc., Peoria, as a Consultant during 1994 to 1998. He is now an Assistant Professor in the Department of Electrical Engineering at the University of Texas at San Antonio, Texas. He has coauthored more than 45 refereed conference and journal articles and several book chapters on neural networks and fuzzy logic systems. His research interests include adaptive control, nonlinear control, intelligent control, robotics, autonomous systems, machine learning, manufacturing systems control, and prognostics diagnostics.

Dr. Jagannathan received several gold medals and scholarships during his bachelor’s, master’s, and doctoral degree programs. He was the recipient of the University Alumni Doctoral Fellowship during 1992 to 1994 and received the Sigma Xi “Doctoral Research Award” in 1994. He currently holds more than 15 patents and he has won the “Patent Award” from the Automation and Robotics Research Institute in 1996. He is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.