
Local sparse coding control of CVPSTs

S. Ozdemir

Artificial Intelligence & Design Laboratory, Mechanical Engineering
Department, Izmir Institute of Technology, 35430 Izmir, Turkey

Email: serhanozdemir@iyte.edu.tr

Abstract: This paper discusses simulations of a control scheme based on locally sparse coded networks (CMACs) for a novel previously proposed continuously variable transmission (CVT), a hybrid continuously variable power split transmission (CVPST) (Ozdemir and Schueller, 2002). Automotive transmissions match the speed and the torque of the power source to the speed and torque requirements of the load. Properly designed CVTs have shown potential to improve efficiency and performance. The main advantage of CMACs is fast computation because of their simple operational principles. Simulation results have shown that memory contents either reach a stable limit cycle or an attractor based on the selection of network parameters and the training method. Both online and offline training are possible.

Keywords: cerebellar model articulation controllers (CMACs), continuously variable transmissions (CVTs), hybrid continuously variable power split transmissions (CVPSTs), local sparse coding, power recirculation, power split, sparse coded networks.

Reference to this article should be made as follows: Ozdemir, S. (2005) 'Local sparse coding control of CVPSTs', *Int. J. of Heavy Vehicle Systems*, Vol. 12, No. 2, pp. 104-120.

1 Introduction

Interest in artificial neural networks have been growing in various fields of engineering as a result of theoretical development and the success of applications. Researchers have proposed many mathematical models of neural networks based on the function of biological neurons and their interconnections. The cerebellar model articulation controller (CMAC) was inspired by knowledge of the function of the human cerebellum. A theoretical model was used to explain the information processing characteristics of the cerebellum. David Marr in 1969 in Great Britain and James S. Albus in 1971 in the USA developed this model (Albus, 1981). It was a model of the structure and functionality of the various cells and fibres in the cerebellum. CMAC is a neural network architecture. Basically, a CMAC computes the desired output by taking inputs as an address to refer to a memory where the weights are stored. CMACs estimate a relationship between the input and output using supervised training techniques.

As for the transmissions, contemporary transmissions are comprised of either mechanical, hydrostatic, torque converter components, or a combination thereof. Mechanical transmissions, which form the basis of an earlier work (Ozdemir and Schueller, 2002), is the most efficient transmission, making it indispensable in cars and other machinery. It has been shown that CVTs can be successfully used in small and medium-size cars. It has also been shown that a CVT mechanism may be improved when the transmission contains multiple power flow paths, which are optimised for a given range of vehicle speeds. When a power split mechanism is combined with a CVT transmission, the new power split CVT has been proved to be not only a viable alternative to an ordinary CVT but also an improved version in performance (Mucino, 1997; Vahapzadeh and Linzell, 1991). Ozdemir *et al.* (2002) illustrated that a combination of a power recirculation, power split, and a regular CVT mechanism in a transmission box would refine the acceleration characteristics of a vehicle and enable a car to move even at creeping speeds, and achieve zero vehicle speed with an inherent clutch, characteristic to a recirculation design. A continuously variable power split transmission (CVPST) allows an engine to run over a range of speeds and loads independent of the speed and torque requirements placed on the wheels by the vehicle and the driver. An engine can produce a broad range of torque at any given speed demanded, or a broad range of engine speeds for any given torque demand. Because the CVPST allows an engine to run at this most efficient point virtually independent of the vehicle speed, a CVPST equipped vehicle may yield substantial fuel economy benefits when compared to a conventional transmission offering only a limited number of input/output ratios. The fact that the performance of a CVPST-equipped vehicle may be optimised means enhancement of the transmission unit as a whole.

The efficient solution of the control problem using conventional control techniques would require a thorough knowledge of the system behaviour, translated into a very accurate nonlinear mathematical model, which is typically very hard to obtain (Cembrano *et al.*, 1997). Neural network control schemes are suited to the control problem. In this case the approximation ability and learning capabilities make neural networks good alternatives. Instead of generating a complicated mathematical model of the system, a relationship between the input and the output is evaluated by the neural networks. The CMAC neural network has the advantage of much faster convergence and online learning ability than the other networks (Cembrano *et al.*, 1997).

In recent years, there have been numerous studies aimed at improving CVTs and the engine as a single power unit. Some of these have only dealt with individual component analyses as design enhancements, such as belt configurations, material selection, hydraulic actuation on CVT ratio selection, etc., which are outside the scope of this paper. The author has focused on the control aspects of such drive systems. One such work (Shen Shuiwen *et al.*, 2001) considered a hybrid driveline concept. So as to eliminate engine speed fluctuations in transient situations, an additional flywheel was added for optimal fuel economy without any restrictions on the driveability. Due to system uncertainties and unpredictable load conditions, a nonlinear robust controller was proposed. In spite of the fact that a fuel saving was reported, a relatively accurate system model has yet to be found. In another fuel-economy oriented control study (Pfiffner *et al.*, 2003), similar control policies were developed for transient regimes based on an optimisation software package. Optimum operating conditions were analysed and formulated based on the optimal control theory, which again necessitated explicit expressions of the system dynamics.

2 CMACs

The CMAC neural network, based on cerebellum's neuromuscular control, is basically a nonlinear lookup table technique which maps n-dimensional input to a corresponding output. The CMAC was first proposed by Albus (1971, 1975a, 1975b) and since then, it has been modified and improved. These studies focused on the development of algorithms (Abdelhameed Magdy *et al.*, 2002; Hsu Yuan *et al.*, 2002; Commuri and Lewis, 1997), improvements of CMAC structure, and applications (Cembrano *et al.*, 1997; Larsen *et al.*, 1995; Kim *et al.*, 2002; Pan *et al.*, 1996).

Lin and Chiang (1997) described the CMAC technique using a mathematical formulation and used this formulation to study the CMAC's convergence properties. Both information retrieval and learning rules are described by algebraic equations in a matrix form. Convergence characteristics and learning behaviours for the CMAC with and without hashing are investigated with the use of these equations. Thompson and Kwon (1995) studied the sequential neighbourhood training and random training techniques for CMACs. These techniques were used to generate mathematical functions. In the neighbourhood sequential training method, a strategy was devised for selecting points in the input space which would train CMAC systems in the most rapid manner. The random training method was found to converge on the training function with the greatest precision, although it requires longer training periods than the neighbourhood sequential training method. Commuri and Lewis (1997) developed novel weight update laws that guarantee the stability of the closed-loop system. The passivity properties of the CMAC under the specified tuning laws are examined and the relationship between passivity and closed-loop stability is derived. Abdelhameed *et al.* (2002) showed that the CMAC-based controller causes instability after a long period of real-time runs, whereby a new learning algorithm was proposed. They used their controller for the trajectory tracking control of a piezoelectric actuated tool post. The performance of the proposed controller is compared with conventional controllers. The experimental results showed that the performance of the CMAC-based controller using the proposed learning algorithm is stable and more effective than that of the conventional controllers.

3 The CMAC network

The detailed knowledge of the structure and the function of the various cell and fibre types in the cerebellum make it possible to form mathematical models to explain its information processing characteristics. Albus produced the version illustrated in Figure 1 (Albus, 1981).

The CMAC, as a controller, computes control values by referring to a memory look-up table where those control values are stored (Albus, 1975a). The memory table basically stores the relationship between the input and output or the control function. In comparison to other neural networks, CMAC has the advantage of very fast learning and the unique property of quickly training certain areas of memory without affecting the whole memory structure. Hence, this network is also known as the local sparse coding networks.

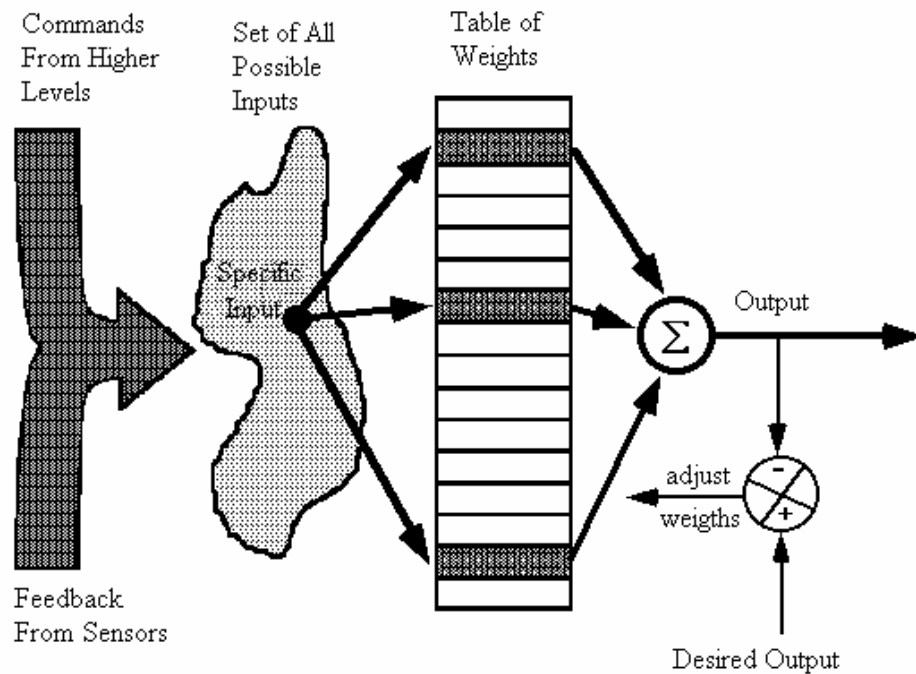


Figure 1 A schematic representation of CMAC (Albus, 1981).

The network architecture of the CMAC is illustrated in Figure 2. The input data of every state variable are quantised into discrete regions and mapped on to different memory areas. Each indexed block memory, called a hypercube, contains the input data of one quantised discrete state. The association memory mapping is implemented through hypercube to the actual memory as the mapping function of table look-up model. In addition to the association memory mapping function, the CMAC gives the feedback of the error of output to adjust the actual memory contents (Albus, 1981).

The output of this system is the summation of the contents of actual memory that is mapped by effective hypercubes. The error caused by the difference between the output summation and the desired output is processed as the feedback value for adjusting the contents of actual memory. The learning efficiency of the CMAC system depends largely on the division of the hypercube. Its technique can be explained with reference to Figure 3.

This example has two state variables (s_1 and s_2) with each quantised into four discrete regions, called blocks. For instance, s_1 can be divided into A, B, C and D and s_2 can be divided into a, b, c and d. Areas formed by quantised regions, named Bb, Gg, Kk, Oo are called hypercubes in the input state of $(s_1, s_2) = (7,7)$. If the quantisation for each variable is shifted by one element, different hypercubes will be obtained.

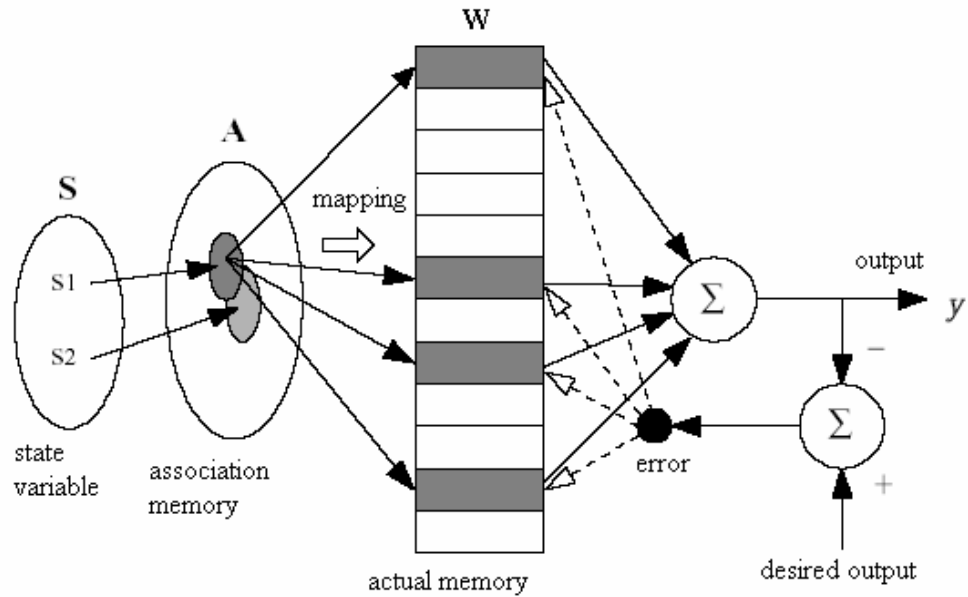


Figure 2 The learning architecture of CMAC (Albus, 1975a).

For example, E, F, G, H for s1 and e, f, g, h for s2 are shifted regions. Ee, Ff, etc. are new hypercubes from the shifted regions. Each state is covered by N_e different hypercubes, where N_e is the number of elements in a complete block. There are 64 ($= 4^2 \times 4$) hypercubes in this example. Each hypercube is taken as the corresponding address of the actual memory element. And the data of each state will be distributively stored in memory elements associated with hypercubes that cover this state. Assume \mathbf{a}_j represents an association vector of j th input space ($j = 1, 2, 3 \dots N_s$) where N_s indicates the total number of input states. 99th input state (state (7, 7)) is used to explain the actual memory how to be mapped by an association memory. Figure 3 shows the state (7, 7) is mapped by the hypercubes of Bb, Gg, Kk and Oo. If we give an index value for each mapped actual memory unit, then the state (7, 7) can be mapped to the memory locations of 6, 27, 43 and 59. We can use an association vector shown as Equation (2) to represent the mapping information.

$$\mathbf{a}_{99} = [0 \dots 0 \overset{a_6}{1} 0 \dots 0 \overset{a_{27}}{1} 0 \dots 0 \overset{a_{43}}{1} 0 \dots 0 \overset{a_{59}}{1} 0 \dots 0]_{1 \times 64} \quad (1)$$

Bb Gg Kk Oo

This is a 1×64 vector because there are 64 hypercubes needed (i.e. 64 actual memory units are used) in this case. In this vector, four 1s represent the mapped actual memory units that are used under this input state, and other 0s represent the mapped actual memory units are not used.

Therefore, the locations of 6, 27, 43 and 59 are recorded as 1 and everything else is recorded as 0. The actual output y_{99} of input state (7, 7) can be represented as:

$$y_{99} = \mathbf{a}_{99} \cdot \mathbf{W} \tag{2}$$

where \mathbf{W} is the weight vector of actual memory contents.

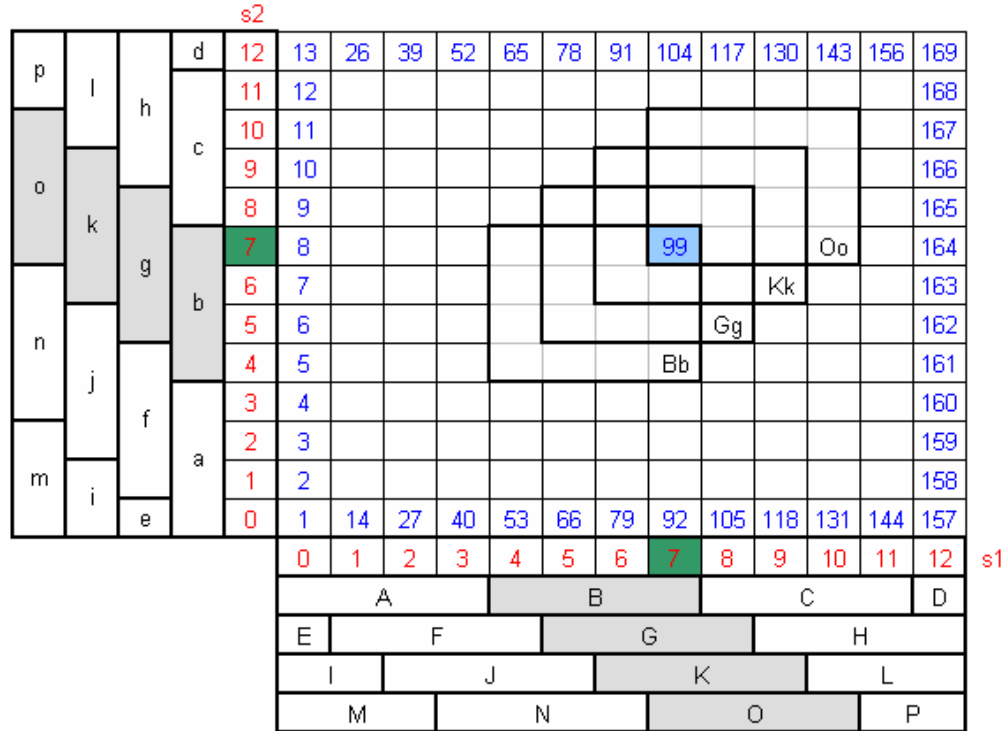


Figure 3 Block division of CMAC for a two-variable example.

4 CVPSTs

The following hybrid power split-power recirculation (PS-PR) gearbox (Ozdemir and Schueller, 2002) is intended to offer the advantages of both power split and power recirculation mechanisms. It has four distinguishable gears, one being the reverse gear. The reverse and drive 1 are governed by power recirculation equations. In this mode, the vehicle can back up, achieve the so-called 'geared neutral' phenomenon, and set the vehicle in forward motion. A characteristic power split CVT is shown in Figure 4.

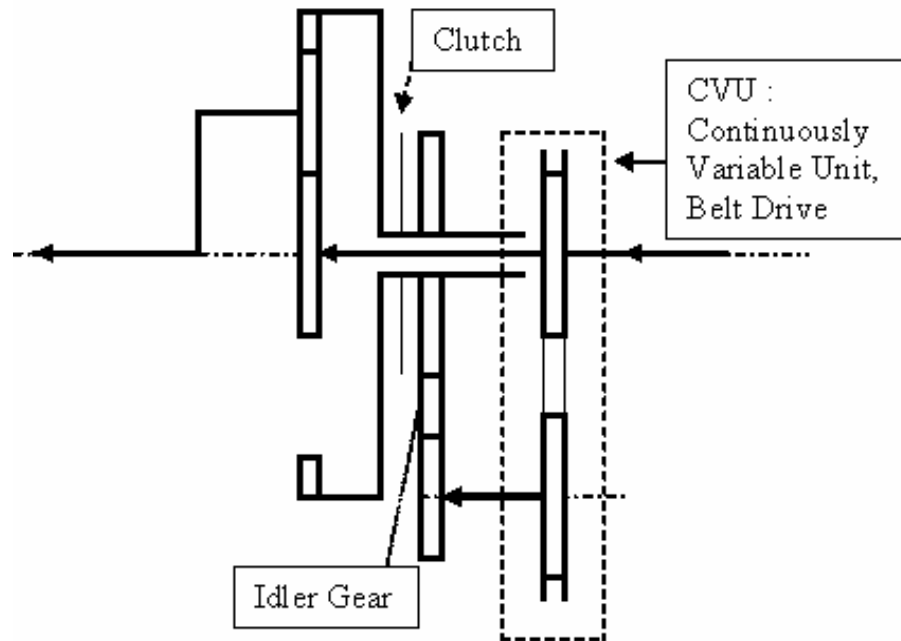


Figure 4 A generic power split CVT.

At a designated speed, which might be called first crossover, the motor vehicle switches to drive 2, or split mode. This mode allows the driver to achieve quicker acceleration, and to achieve relatively higher speeds. In this mode, all of the power is carried through the system's two belt units. The lessening of torques at high speeds permit the running of torques along continuously variable units. This feature is the key to rapid acceleration.

Further down the road, at even higher speeds, a third drive has been provided. At a second designated speed, another switch occurs. This shift from drive 2 to drive 3 might be called the second crossover. In this mode, to take advantage of ever-smaller torques, and to eliminate the inertia of masses of rotating gears, the path is now a single line, and no power split is needed.

The proposed design, as shown in Figure 5 in connection with Table 1, is the combination of a recirculation and a split design. The power is input into the system from the lower-right corner of the picture and output from the upper-left corner. The design contains two CVUs or two belts, and is composed of three stages. In each stage, the power follows a different path, in accordance with the control and the vehicle speed. The caption explains the symbols in the figure.

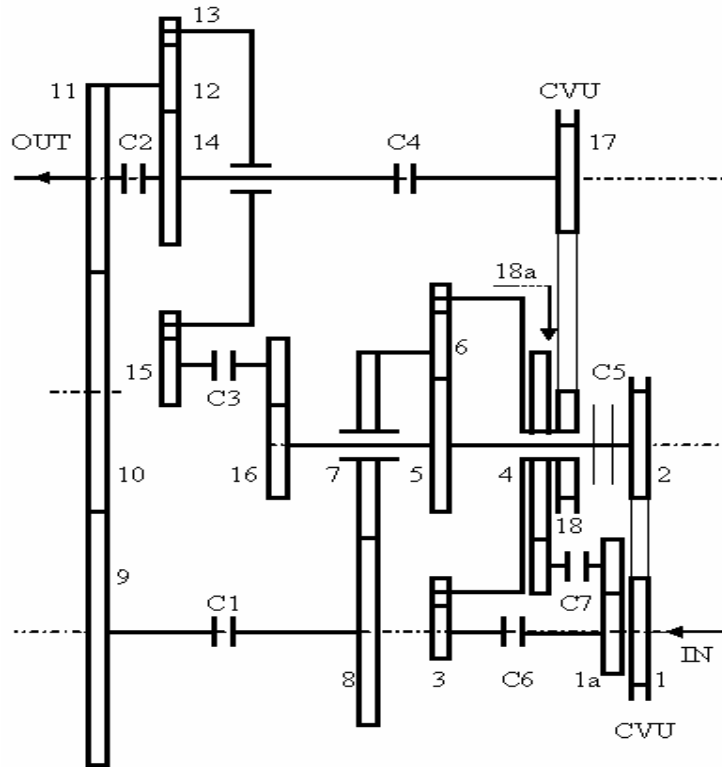


Figure 5 The CVPST to be controlled (Ozdemir and Schueller, 2002). C1..C7 : clutches; CVU: continuously variable units; 1, 2, 17, 18: variable pulleys; 3, 4..15, 16 : gears.

Table 1 Clutch layout table.

	C1	C2	C3	C4	C5	C6	C7
Reverse	ON					ON	
Drive 1	ON					ON	
Drive 2			ON	ON			ON
Drive 3		ON		ON	ON		
Neutral							

5 Inverse kinematics control with CMAC

The inverse kinematics control problem is to find quickly each individual input speed and requested acceleration. These entities could be calculated by the formulations. But rather than using formulations, CMAC is used here to evaluate what the input speed into the CVPST ought to be, given the desired vehicle speed. The task is motion control (point to point in the defined workspace). In Figure 6 Θ represents the accelerator pedal angle measured from a reference point. Thus not only the angular displacement but also the rate of change of the pedal is taken into consideration. CMACs output the necessary input angular speed and its rate of change into the CVPSTs by definition. The rate of change in the angular speed is provided by a continuously variable unit, in this case a belt drive. Now the problem is converted into the selection of input parameters based on the desired vehicle speed and a constant engine rpm. CMAC is trained in a supervised manner to find the training data set; the geometric equations can be used or approximate training values might be provided based on experience.

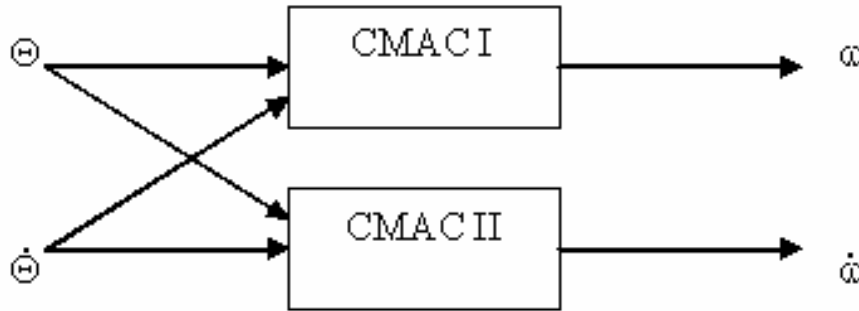


Figure 6 CMAC network for inverse kinematics of two-link manipulator.

The problem becomes much more complex as the number of DOF increases. The standard methodology for calculating the inverse kinematics is training the neural network offline for possible data to obtain solutions. Because of the generalisation property, neural networks can learn the associated patterns and recall the learned patterns. The trained network is then used to achieve the desired movements. This technique therefore involves two steps of operation as training phase and performing phase. From another point of view the advantage of using neural network for control problems is that neural networks have the ability of online learning and adaptive capabilities. The inverse kinematics of the proposed control scheme can be seen in Figure 7.

Initially, all the weights are equal to zero. The outputs of the CMAC are the input angular speed and its rate of change which are inputs to the CVPST. The error signals are calculated as the difference between the desired and the actual vehicle dynamic parameters. It is shown that the CVPST with CMAC can respond to any desired speed and acceleration within the working range.

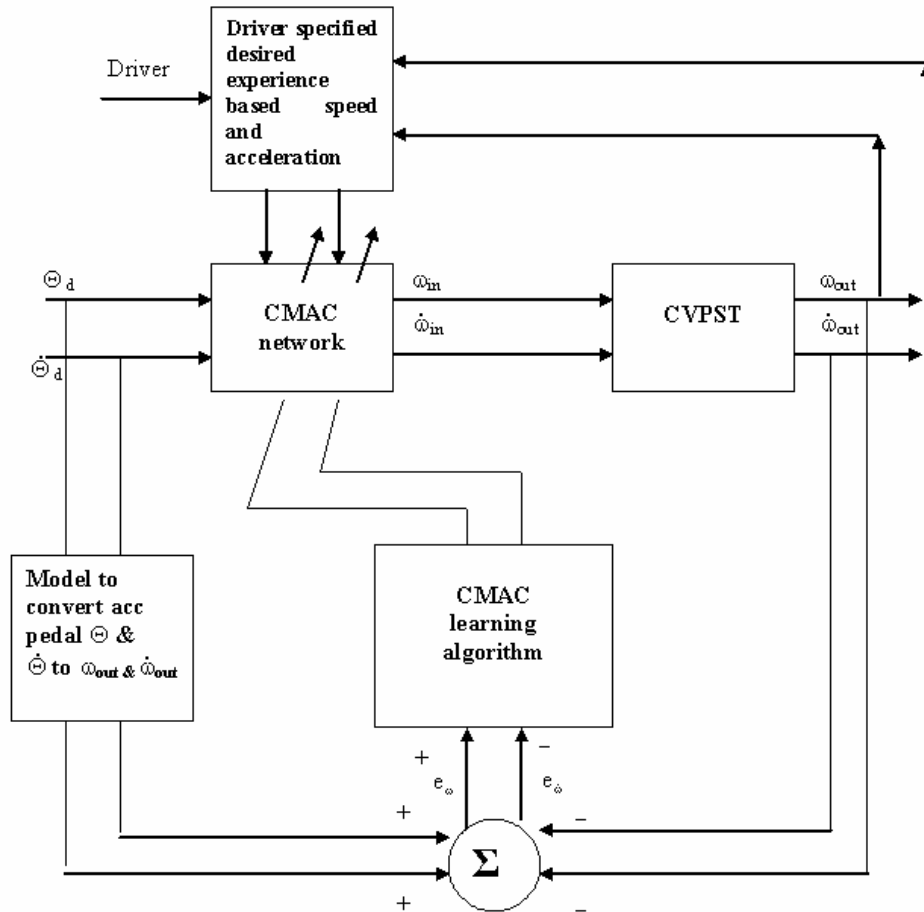


Figure 7 Block diagram for online learning of inverse kinematics of two-DOF manipulator.

6 Robot arm testbed

Before crediting any simulations, the CMAC code was tested on a realisable physical model. A robotic manipulator was used for this test. Even before the construction of the manipulator, simulations for the arm control were quite successful. On the test bench in Figure 9, CMAC was trained offline, and the training data were produced using the geometry of the two-link manipulator.

The desired x, y positions of the end effector are applied to the CMAC network, as in Figure 8. The network parameters of the CMAC are:

- generalisation width = 64
- total number of layers = 64

learning rate = 0.5

The desired x_d and y_d coordinates of the end effector are selected as:

$$x_d = [300 \ 100 \ 400], \quad y_d = [50 \ 250 \ 100].$$

The desired end effector positions are applied to CMAC network. The x and y positions of the end effector are shown in Figure 10, and the corresponding joint angle trajectories are shown in Figure 11.

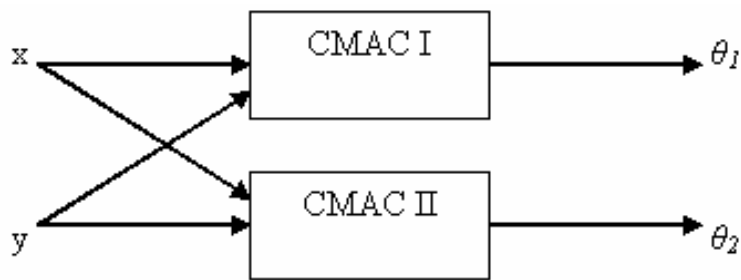


Figure 8 *CMAC network for inverse kinematics of two-link manipulator.*

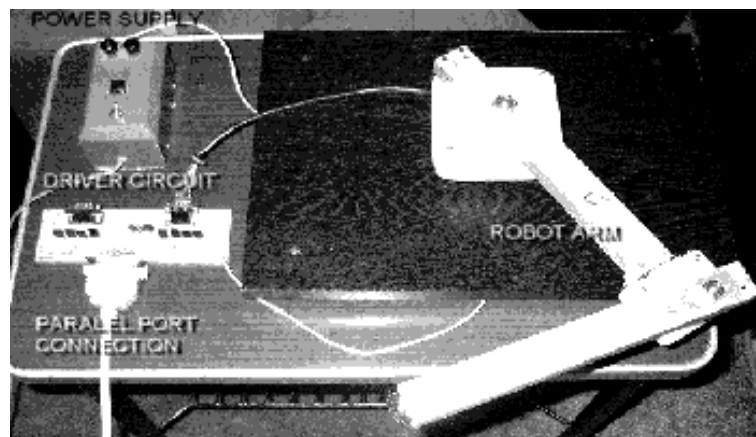


Figure 9 *Experimental hardware set-up.*

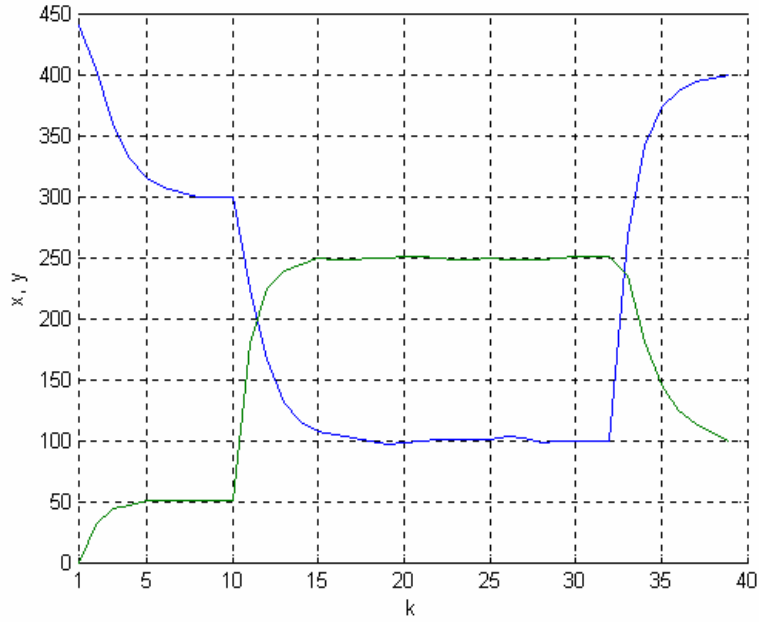


Figure 10 Simulation of the actual positions of the end effector of the two-linked robot.

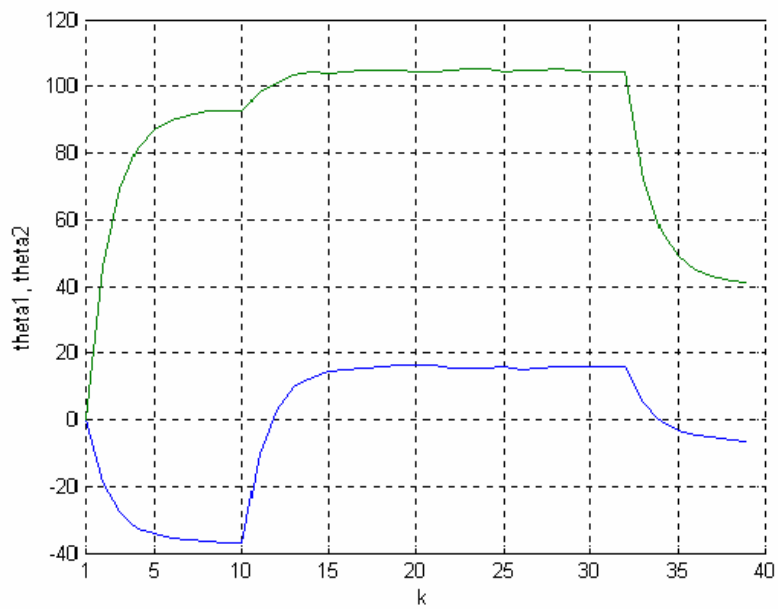


Figure 11 Joint angle trajectories of the two-linked robot.

Thirty-nine iterations are needed to reach the desired end-effector positions. The CMAC can be a fast solution for the inverse kinematics problem in control applications online. On the other hand, CMAC is trained offline even faster.

The network takes just seconds to train offline, and, even though the manipulator extends to the desired location with no errors in simulations, errors have occurred in hardware implementations to control a 2-DOF arm (Figure 9). This is not due to the code but rather the experimental set-up which was meant to be simple. For the feasibility of the CMAC, the whole control set-up was built in an open loop. No error correction is thus provided. For a given limb, angular misplacements did not exceed an absolute value of 10 degrees. Yet the CMAC algorithm is proven to work in a network topology, shown in Figure 8, which is the same for CVPST simulations, as in Figure 6.

7 Simulations to control CVPST with CMACs

The training data are selected in three different ways. First in a random fashion. Second, the neighbourhood sequential training technique is used. Finally, the training points are selected within the symmetry of the workspace. Figures 12, 13 and 14 present the performance of each mentioned training schemes. In Figure 12 the random training performance is seen. Generalisation width and the total number of layers are equal to 164. Memory converges after 10 cycles. In Figure 13 the neighbourhood sequential training method's performance is seen where the generalisation width and the total number of layers are 44. Finally in Figure 14 the output errors of the network are seen. Here the generalisation width and total number of layers are equal to 96. The desired numerical data are as follows:

$$\omega_d = [-33 \ -19 \ -23 \ -10 \ 20 \ 24 \ 33]$$

$$\dot{\omega}_d = [\ .1 \ .65 \ .25 \ .20 \ .3 \ .15 \ .7]$$

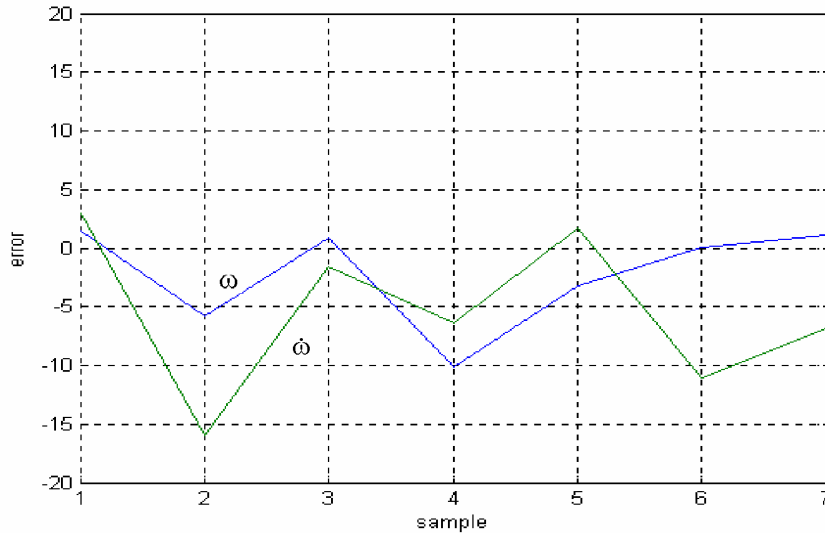


Figure 12 The output error for the random trained network.

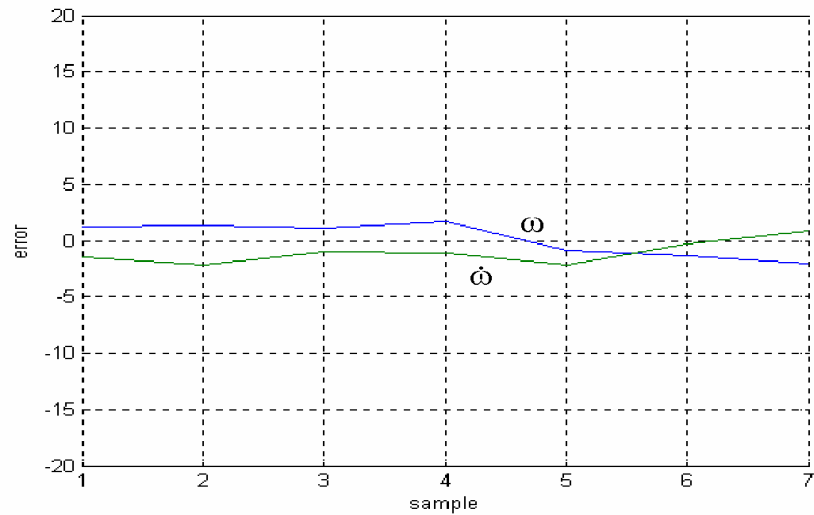


Figure 13 The output error for the sequentially trained network.

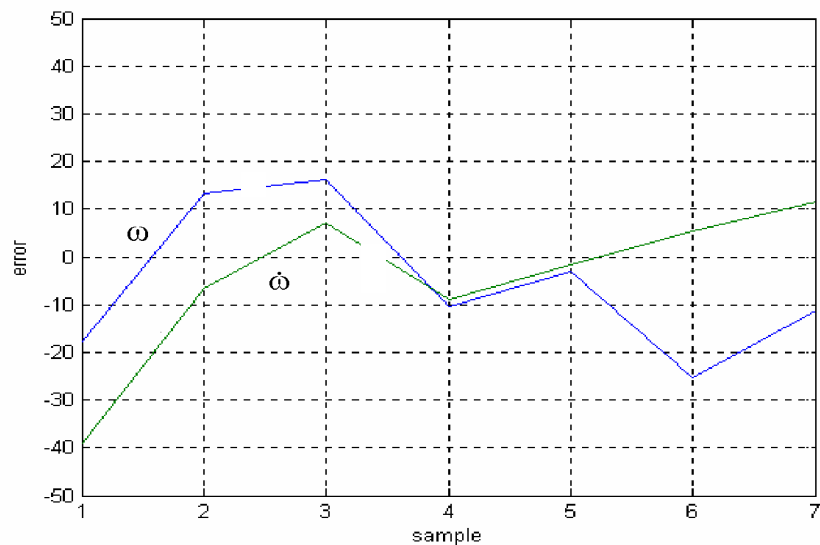


Figure 14 The output error for the symmetric trained network.

Tables 2 and 3 show a set of typical memory convergence graphs. The memory contents in Table 2 attain a stable limit cycle. In this training example, no hashing was used, and quite a small mean error was obtained. Table 3 also shows good error characteristics with no increase in memory, and again no hashing was used. In between these two extremes, various results were obtained with different training parameters.

Table 2 For the given CMAC parameters, memory contents reach a limit cycle.

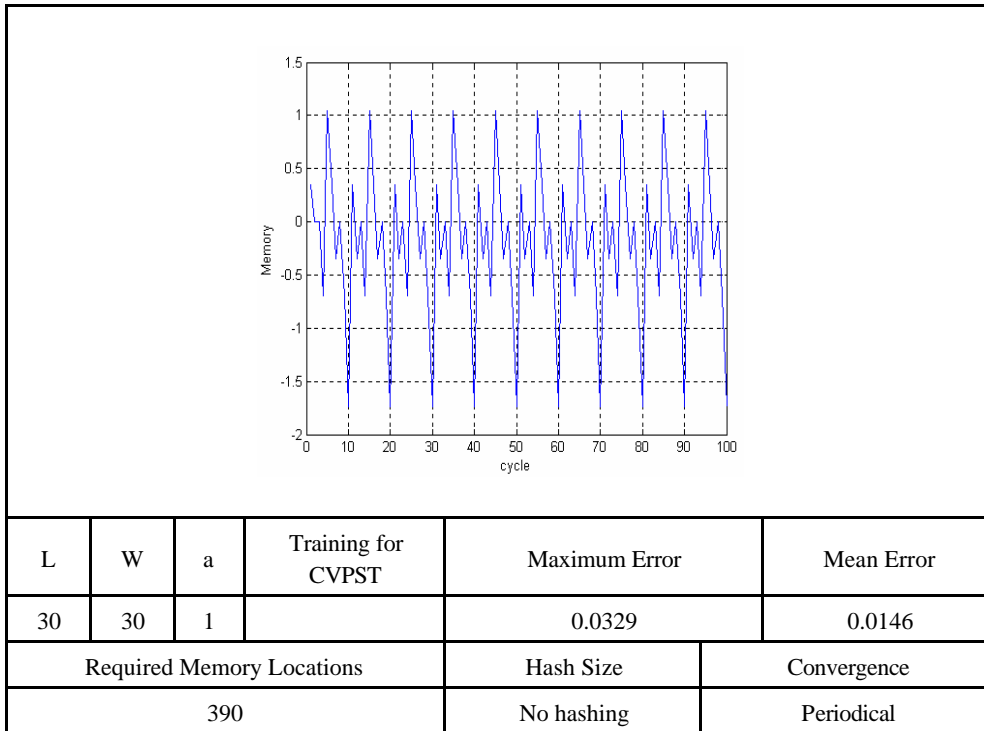
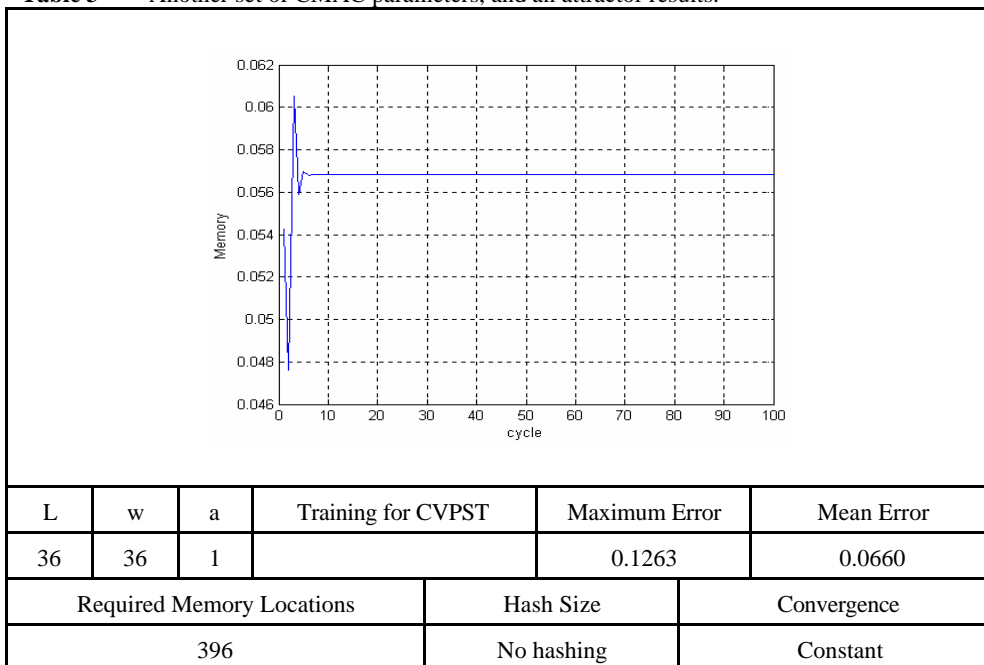


Table 3 Another set of CMAC parameters, and an attractor results.



8 Conclusions

Throughout the last decade various neural network architectures have been applied to numerous control applications. This has enabled intelligent control algorithms to take over the stationary control schemes that would not respond to any change in system dynamics. Problems with dynamic variations have been conventionally dealt with using rather robust control. One such application domain has been the CVT-equipped transmissions. An online learning intelligent model would not only update its rules of execution, provided that it is an inductive inference machine in varying situations, but also remove the necessity of proposing the exact system dynamics. One such alternative is the implementation of CMACs.

The main advantage of a CMAC over other neural networks is its fast computation, which results from the fact that it is more of a table look-up technique and less of a network. In the simulations for the test bed, and CVPST, it was seen that sequential neighbourhood training is a fast and efficient method for tackling the problem of inverse kinematics. Also the network parameters, such as generalisation width and total number of layers, are critical points. The optimum results are obtained when the generalisation width is in the neighbourhood of the total number of layers. And if the distance between the two training points is equal to the generalisation width, the CMAC produces a smooth curve. Also the other advantage of sequential neighbourhood training is the lower memory requirements. By using hash coding, the required memory can be decreased dramatically. As a result, for inverse kinematics calculations, a sequentially neighbourhood trained CMAC network's performance is a good alternative. In the test bed, a two degrees of freedom manipulator was analysed. In the case of many degrees of freedom systems, the analytic solutions are more complicated, and sometimes there are no formulations for the inverse kinematics. In such situations, one has to brave the complicated iterations. The ease with which the network was trained without a mathematical model was also shown.

During the analysis of CMAC network and hardware implementation, MATLAB software was used on a Pentium III PC. The simple nature of CMACs can be an advantage for the application areas. For instance, instead of a PC interface the same work could have been realised with micro controllers.

Any future study ought to deal with effective hash coding techniques where a demand for a higher resolution would require more memory, while still minding the hash collisions. Stability is another subject for CMACs. Some adaptive algorithms are developed for stability problems of CMACs. These adaptive algorithms need to be employed to state stability margins for any given work. This paper discussed only the simulations. Another test bed, this time with a scaled CVPST, should be tested for further suitability.

References

- Abdelhameed, M. M., Pinspon U., and Çetinkunt S. (2002) 'Adaptive learning algorithm for cerebellar model articulation controller', *Mechatronics*, Vol. 12, pp. 859-873.
- Albus, J. S. (1975a) 'A new approach to manipulator control: the cerebellar model articulation controller', *Journal of Dynamic Systems, Measurement and Control*, pp. 220-227.

- Albus, J. S. (1975b) 'Data storage in the cerebellar model articulation controller', *Journal of Dynamic Systems, Measurement and Control*, pp. 228-233.
- Albus, J. S. (1981) 'Brains, behavior, and robotics', BYTE Books, McGraw-Hill.
- Cembrano, G., Wells, G., Sarda, J., and Ruggeri A. (1997) 'Dynamic control of a robot arm using CMAC neural networks', *Control Eng. Practice*, Vol. 5, No. 4, pp. 485-492.
- Commuri, S. and Lewis, F. L. (1997) 'CMAC neural networks for control of nonlinear dynamical systems: structure, stability and passivity', *Automatica*, Vol. 33, No. 4, pp. 635-641.
- Hsu Yuan, P., Hwang, K. S. and Wang, J. S. (2002) 'An associative architecture of CMAC for mobile robot motion control', *J. Information Science and Engineering*, Vol. 18, pp. 145-161.
- Kim, D.-H., Oh, Ju-W. and Lee, In-W. (2002) 'Cerebellar model articulation controller for suppression of structural vibration', *J. Computing in Civil Engineering*, pp. 291-297.
- Larsen, G. A., Çetinkunt, S. and Dönmez A. (1995) 'CMAC neural network control for high precision motion control in the presence of large friction', *J. Dynamic Systems, Measurement, and Control*, Vol. 117, pp. 415-420.
- Lin, C.-S. and Chiang, C.-T. (1997) 'Learning convergence of CMAC technique', *IEEE Trans on Neural Networks*, Vol. 8, No. 6, pp. 1281-1291.
- Mucino, V. H. (1997) 'A continuously variable power split transmission for automobile applications', SAE Paper No 970687.
- Ozdemir, S. and Schueller, J. (2002) 'A new hybrid CVT design : CVPSTs', *Heavy Vehicle Systems*, Vol. 9, No. 4, pp.319-332.
- Pan, G., Xu, H., Kwan, C. M., Liang, C., Haynes, L. and Geng Z. (1996) 'Modelling and intelligent chatter control strategies for a lathe machine', *Control Eng. Practice*, Vol. 4 No. 12, pp. 1647-1658.
- Pfiffner, R. *et al.* (2003) 'Fuel-optimal control of CVT powertrains', *Control Engineering Practice*, Vol 11, pp. 329-336.
- Shen, S. *et al.* (2001) 'Coordinated control of a mechanical hybrid driveline with a continuously variable transmission', *JSAE Review*, Vol. 22, pp. 453-461.
- Thompson, D. E. and Kwon, S. (1995) 'Neighborhood sequential and random training techniques for CMAC', *IEEE Trans on Neural Networks*, Vol. 6, No. 1, pp. 196-202.
- Vahapzadeh, H. and Linzell, M. (1991) 'Modeling, simulation, and control implementation for a split-torque, geared neutral, infinitely variable transmission', *Automotive Transmissions Advancements*, SAE Special Publications, No. 854, Feb.