

absolute values of the capacitive coupling coefficients of the calibration inputs differ at each stage due to variable total capacitance between stages.

The measured results with a ramp signal  $f_{va} = 500\text{Hz}$  at the input are shown in Fig. 4b (only the 4th bit is shown). The power consumption of the present circuit was only  $67\mu\text{W}$ , when  $V_{DD} = 2\text{V}$  and  $f_{va} = 10\text{kHz}$ . The measured results indicate that the circuit operates properly, and due to the calibration the bit number can be increased.

© IEE 1999

16 October 1998

Electronics Letters Online No: 19990110

A. Rantala, S. Franssila, M. Åberg and P. Kuivalainen (VTT Electronics, PO Box 11012, FIN-02044 VTT, Finland)

K. Kaski and J. Lampinen (Laboratory for Computational Engineering, Helsinki University of Technology, Miestentie 3, PO Box 9400, FIN-02015 HUT, Finland)

P. Kuivalainen: Also with Electron Physics Laboratory, Helsinki University of Technology, Otakaari 7A, FIN-02150 Espoo, Finland

## References

- 1 SHIBATA, T., and OHMI, T.: 'A functional MOS transistor featuring gate-level weighted sum and threshold operations', *IEEE Trans. Electron Devices*, 1992, **39**, pp. 1444-1455
- 2 KOTANI, K., SHIBATA, T., and OHMI, T.: 'Impact of high-precision processing on the functional enhancement of neuron-MOS integrated circuits', *IEICE Trans. Electron.*, 1996, **E79 C**, pp. 407-414

## Improved CMAC neural network control scheme

C.C. Lin and F.C. Chen

The cerebellar model articulation controller (CMAC) neural network control scheme is a powerful tool for practical real-time nonlinear control applications. However, despite its initial outstanding performance, the CMAC control system can suddenly diverge after a long period of stable tracking. The authors propose a modification to the CMAC control scheme in order to improve stability while preserving the original good performance.

**Introduction:** The cerebellar model articulation controller (CMAC) was proposed by Albus [1]. This neural network is capable of learning nonlinear functions extremely quickly due to the nature of its weight updating, so it is a powerful and practical tool for real-time control. Miller *et al.* [2, 3] proposed a feasible control scheme to combine the CMAC and a traditional controller (e.g. PID) for robot manipulator control. However, in our recent paper [4], it was shown that the Miller scheme can be unstable. The purpose of this Letter is to use the simple example in [4] to explain a modification that can be made to significantly improve the stability of the CMAC control system.

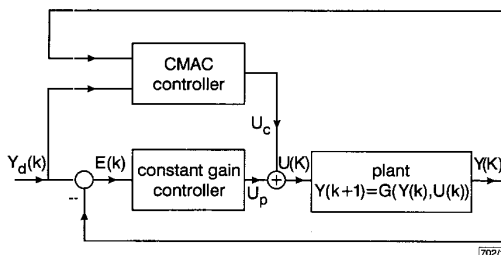


Fig. 1 CMAC control system

**Control problem and original scheme:** The CMAC control system is shown in Fig. 1. The simple system to be controlled is

$$Y(k+1) = G(Y(k), U(k)) \quad (1)$$

The control  $U(k)$  is generated by

$$U(k) = U_c(k) + U_p(k) \quad (2)$$

where  $U_p(k) \equiv P(Y_d(k) - Y(k))$  is the output of the traditional constant gain controller,  $P$  being the proportional gain and  $Y_d$  being the desired system output.  $U_c(k) \equiv \text{CMAC}(Y_d(k+1), Y(k))$  is the output of the CMAC module for the input vector  $(Y_d(k+1), Y(k))$ . Then the CMAC is updated by the following gradient-type learning rule:

$$W_i(k+1) = W_i(k) + \frac{\mu}{g}(U(k) - \text{CMAC}(Y(k+1), Y(k))) \quad (3)$$

where  $g$  is the size of generalisation,  $W_i$  is the content of the  $i$ th memory location (there are  $g$  locations to be updated), and  $\mu$  is the learning rate. In essence, in this scheme, the constant gain controller helps the CMAC to learn the inverse of the plant. However, these two controllers are independent, and the conflicts between them are the source of the instability [4].

**Proposed modification:** Instead of having the system inverse solely learned by the CMAC, the constant gain controller and the CMAC are integrated, and will be made to approximate the system inverse together. In other words, the CMAC will be used to learn the difference between the system inverse and the constant-gain control. To carry out this modification, the constant gain control has to be calculated based on the same input vector as is applied to the CMAC. In eqn. 2, since the input vector to the CMAC is  $(Y_d(k+1), Y(k))$ , the proportional control  $U_p$  is modified to be  $U_p(k) \equiv P(Y_d(k+1) - Y(k))$ . After the control  $U(k) = U_p(k) + U_c(k)$  is applied to the plant, the CMAC goes through the learning process according to

$$W_i(k+1) = W_i(k) + \frac{\mu}{g}(U(k) - U_p(k) - U_c(k)) \quad (4)$$

Note that, in eqn. 4,  $U_p(k)$  is calculated by  $P(Y(k+1) - Y(k))$ , and  $U_c(k)$  is generated from the input vector  $(Y(k+1), Y(k))$ , where  $Y(k+1)$  is the actual system output at time step  $k+1$ .

**Simulation results:** As in [4], the nonlinear system used in the simulation is  $Y(k+1) = 0.5Y(k) + \sin(Y(k)) + U(k)$ . This system is expected to track the command  $\sin(2\pi k/400)$ , with each cycle containing 400 time steps. The results using the old and the modified schemes are shown in Figs. 2 and 3, respectively.

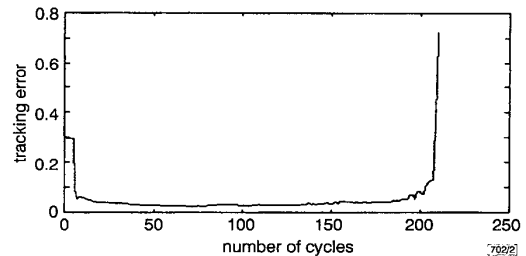


Fig. 2 Tracking error with original scheme

Tracking error reduces significantly, but then diverges. Horizontal axis: number of sinusoidal cycles, each cycle containing 400 time steps; vertical axis: largest tracking error in each cycle

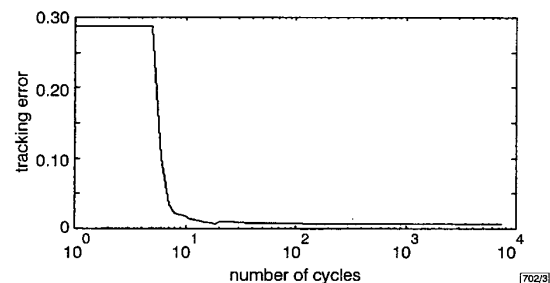


Fig. 3 Tracking error with modified scheme

Tracking error reduces significantly and keeps stable. Horizontal axis: number of cycles in log scale, each cycle containing 400 time steps; vertical axis: largest tracking error in each cycle

In Fig. 2, the system starts to diverge around the 200th cycle, whereas in Fig. 3 the system shows no tendency to instability when the program stops at the 7500th cycle. We are very confident that the stability can be maintained forever because we observe that all aspects in the modified closed-loop control system are very stationary. Note that, in Figs. 2 and 3, the CMAC start to function at the fifth cycle. Before that, only the constant gain controller is in charge. One can observe that the CMAC quickly and dramatically reduces the track error once it takes effect.

**Conclusions:** The better stability in the new scheme is due to the fact that the destabilising interactions (as described in [4]) between the CMAC and the constant gain controller in the original scheme have been eliminated. Although the new CMAC scheme is demonstrated by simple examples, the ideas can be extended to sophisticated and practical nonlinear control problems. This might open a range of new research areas. In addition, the usefulness of the CMAC neural networks justifies the need for parallel-processing CMAC IC chips, which we have been developing for several years, and publish details of elsewhere.

© IEE 1999

9 November 1998

Electronics Letters Online No: 19990083

C.C. Lin and F.C. Chen (Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, Republic of China)

E-mail: fchen@cc.nctu.edu.tw

## References

- ALBUS, J.S.: 'A new approach to manipulator control: The cerebellar model articulation controller (CMAC)', *J. Dyn. Syst., Meas. Control*, 1975, pp. 220-227
- MILLER, W.T., GLANZ, F.H., and KRAFT, L.G.: 'Application of general learning algorithm to the control of robotics manipulators', *Int. J. Robot. Res.*, 1987, 6, pp. 84-98
- MILLER, W.T., HEWES, R.H., GLANZ, F.H., and KRAFT, L.G.: 'Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller', *IEEE Trans. Robot. Autom.*, 1990, 6, pp. 1-9
- CHEN, F.C., and CHANG, C.H.: 'Practical stability issues in CMAC neural network control systems', *IEEE Trans. Control Syst. Technol.*, 1996, 4, pp. 86-91

## 10Gbit/s all-optical regenerative memory using single SOA-based logic gate

R.J. Manning, I.D. Phillips, A.D. Ellis, A.E. Kelly, A.J. Poustie and K.J. Blow

The authors report an all-optical regenerative memory operating at 10Gbit/s using a semiconductor optical amplifier-based interferometer as a regenerator in an optical fibre loop. Error-free operation was measured after more than 30 000 circulations of the stored data.

**Introduction:** An all-optical memory is one of the possible building blocks for future ultra-high speed photonic networks, to be used, for example, in the buffering and queuing of high bit rate (100Gbit/s) packets for subsequent processing. Considerable progress has been made recently in the development of a stable, all-optical regenerative memory [1]. In this approach, two semiconductor optical amplifier (SOA)-based loop mirror interferometers, using the TOAD/SLALOM [2-4] topology, were combined in an optical fibre loop and regenerated an input data 'word' twice every circulation. Two different wavelengths were used to distinguish 'probe' from 'switching' pulses. The second TOAD was used to wavelength convert 1 Gbit/s data switched from the first TOAD so that it was the correct wavelength to be regenerated in the first TOAD. Here, we describe a simpler architecture which uses a single all-optical logic gate where 'pump' and 'probe' pulses are distinguished by propagation direction. We use an ultrafast nonlinear interferometer (UNI) [5, 6] and demonstrate an all-optical regenerative memory operating at 10Gbit/s. The advantages of regenera-

tion upon each circulation are retained in this configuration. In principle it is also possible to use polarisation diversity for single gate applications [7].

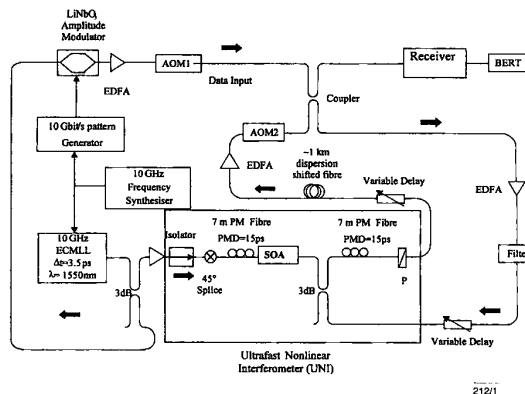


Fig. 1 Schematic diagram of experimental setup

Single gate memory comprises UNI, which acts as a single gate regenerator, and ~1km of dispersion shifted fibre, which is the storage ring

**Experiment:** A schematic diagram of the experimental arrangement is shown in Fig. 1. The pulse source used was a 10GHz external cavity modelocked semiconductor laser (ECMLL), which produced 3.5ps pulses (FWHM) at a wavelength of 1550nm. These pulses were split using a 3dB coupler; one output being used for the probe pulse input to the UNI. The other output from the coupler was passed through a lithium niobate amplitude modulator driven by a 10Gbit/s, pseudo-random data sequence. These data pulses were amplified in erbium-doped fibre amplifier (EDFA) and passed through an acousto-optic modulator (AOM1), which was usually non-transmitting. Another coupler allowed 20% of each data pulse into the memory loop. Data pulses in the loop were amplified using an EDFA, filtered to remove unwanted spontaneous emission, and introduced as switching pulses into the UNI. The switched probe pulses from the UNI passed through ~1km of dispersion shifted fibre (having a dispersion zero at a wavelength of ~1550nm), a further EDFA and an acousto-optic modulator (AOM2) (usually transmitting). After the coupler, these pulses became switching pulses for the next circulation of the loop memory, 20% passing to a bit error rate test (BERT) set via a high-speed optoelectronic detector.

The UNI was used in a counter-propagating configuration similar to that described in [5, 6]. Probe pulses input to the UNI passed through an EDFA, to control the probe power, and were launched at 45° to the axes of polarisation maintaining (PM) fibre, thus splitting into orthogonally polarised pulse pairs. The pulse pair were separated by 15ps after propagation through 7m of fibre due to polarisation mode dispersion (PMD), and were input to a polarisation insensitive SOA with a mean power of ~-1dBm. After the SOA the pulse pair was launched at 45° into a second 7m length of PM fibre, so the pulse pair suffered a delay of the opposite sense and hence recombined. The resultant pulse passed through a fibre polariser *P*. Switching pulses were input via the 3 dB coupler with a mean power of ~-2dBm and were counter-propagating to the probe pulse pair. The switching pulse causes a change in the gain and hence the refractive index of the SOA and this affects the relative phase of the pulse pair, a phase difference of  $\pi$  resulting in a polarisation rotation of 90° at the polariser of the UNI. The SOA used here had a gain recovery time of 80ps at a current bias of 400mA, and an alpha parameter of ~-9 at 1550nm [8]. Error-free operation at 10Gbit/s was easily achievable by adjustment of the probe pulse pair power, which acted as a 'holding beam' to reduce the effective lifetime of the SOA [4, 6].

The memory operation was started by simultaneously switching AOM1 to transmission mode and AOM2 to rejection mode (both acousto-optic modulators had rise and fall times of ~20ns). This allowed pulses to be loaded into the loop and prevented pulses already in the loop from being recirculated. The loop was filled with the required data pulses and the modulators were then both returned to their initial state, after which time the data in the memory was allowed to continuously recirculate up to a maximum time of ~185ms. This limit was imposed by the timing electronics