

# HALOGEN AUTOMATIC DAYLIGHT CONTROL SYSTEM BASED ON CMAC CONTROLLER WITH TRIANGULAR BASIS FUNCTIONS

**Horatiu Stefan Grif, Mircea Dulău**

”Petru Maior” University of Târgu Mureș, Romania  
hgrif@emgineering.upm.ro, mdulau@emgineering.upm.ro

## ABSTRACT

*In general, in an automatic daylight control application, the automatic lighting control system (ALCS) attempt (due to the controller presence) to maintain constant the illuminance at a desired level even the daylight illuminance is variable. The paper describes the design, the implementation and the tuning of a CMAC (Cerebellar Model Articulation Controller) type controller used in an automatic daylight control system, where the lighting process is implemented by a halogen lamp. After the tuning of CMAC controller, is presented the behaviour of the ALCS when the illuminance is disturbed by a supplementary electric light source (a halogen desk lamp). Even the applied control structure uses a gross approximation of the real inverse model of process the ALCS will have a good behaviour achieving the imposed performances for an automatic control system used in this specific application.*

**Keywords:** daylight control, automatic lighting control system, CMAC structure

## 1. Introduction

Nowadays, the artificial intelligence is widely used in research in the automatic process control domain. The Cerebellar Model Articulation Controller (CMAC) proposed by Albus in [1] as a model of the information processing activities within the cerebellum [13] has its special place in the researchers' attentions. This type of artificial neural network is preferred due to its local generalization, extremely fast learning speed and easy implementation in software and hardware [11],[15]. In the last decade different training algorithms for CMAC neural network as functions, data mapping approximator or in control applications was investigate. In [15] was proposed the Credit-Assignment CMAC (CA-CMAC) algorithm to reduce learning interference in conventional CMAC. In [10] is used the algorithm of CMAC-RLS, which applies recursive least square algorithm (RLS) to update the weights of CMAC and proved to be a good tool for on line modeling. In [11] a simplified algorithm of CMAC-RLS named CMAC-QRLS is proposed for reducing the computation time, reducing memory storage, and improving the numerical stability of the CMAC. In [12] was proposed a hybrid maximum error algorithm with neighborhood training for CMAC and proved in an inverse kinematics problem of a two-link robot arm In [14] was proved that a

CMAC can universally approximate a smooth function and its derivatives and in [2] is used the CMAC structure as nonlinear function approximator. In [8] the first author of the present paper used and studied the CMAC controller with Gaussian basis functions to control the same process used in the present paper.

In section 2 is presented the control structure, the experimental model of the process used by the control structure and the experimental stand used for experimentations. In section 3 is described shortly the general structure of the CMAC neural network and is presented the structure of the controller. In section 4 is presented experimental results related to the controller tuning and the behaviour of the automatic lighting control system (ALCS) in the presence of perturbations.

## 2. The control system configuration and the experimental stand

The control structure applied to the lighting process is depicted in Fig. 1 where, are denoted with:  $E_{desired}$  – the desired illuminance;  $E_{measured}$  – the measured illuminance;  $E_{real}$  – the illuminance;  $E_{daylight}$  – the daylight illuminance;  $E_{electric}$  – the illuminance due to electric light;  $\varepsilon$  - control error;  $\Delta\varepsilon$  - change in control error;  $U$  – control action;  $GE$  – scaling gain for the  $\varepsilon$  input of controller;  $GCE$  - scaling gain for

the  $\Delta\varepsilon$  input of controller;  $GU$  - scaling gain for the output of controller (variation in command  $\Delta U$ ).

Implementing the controller as incremental type [9] the control action is calculated by:

$$U(kT) = U(kT - T) + \Delta U(kT) \cdot GU \quad (1)$$

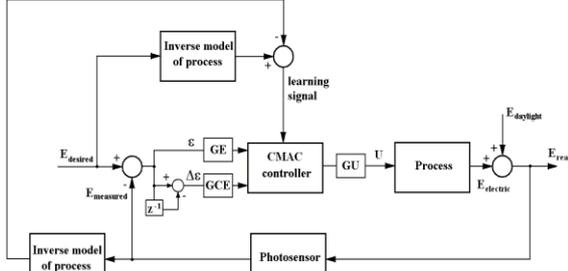


Fig. 1. The block diagram of the control system [8]

The  $\varepsilon$  and  $\Delta\varepsilon$  are given by:

$$\varepsilon(kT) = E_{desired}(kT) - E_{measured}(kT) \quad (2)$$

$$\Delta\varepsilon(kT) = \varepsilon(kT) - \varepsilon(kT - T) \quad (3)$$

where  $T$  is the sampling time.

The model of the process is unknown. For this purpose it was used an experimental model. The experimental direct model of process, a look up table (LUT) of measured data at the input and the output of process, is presented in Fig. 2.

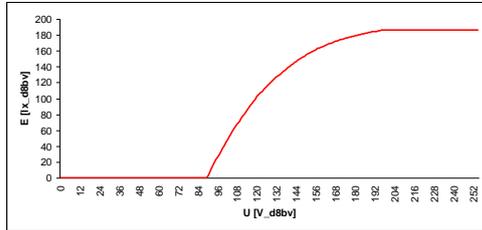


Fig. 2. The experimental model of the process (the direct model) [8]

The transformation given by the LUT is not single-valued. The experimental inverse model of process is the LUT of the output and the input data of the process which transform the interval  $[0, 186]$  ( $lx_{d8bv}$ ) in the interval  $[89, 196]$  ( $V_{d8bv}$ ). The meaning of notation  $d8bv$  is “digital 8 bits value”. A value followed by the unit which incorporate the notation “d8bv” represent the value of the measured signal converted by an 8 bits AD converter in case of measured illuminance and a digital 8 bits value which will be converted in an analogical signal by an 8 bits DA converter in case of the control action.

Fig. 3 shows the experimental stand composed of: (1) calculation equipment (IBM compatible, PI, 166MHz, 64Mb RAM), (2) execution element, (3) the technological installation based on one 40W halogen lamp, (4) light sensor, (5) data acquisition board with

two 8-byte conversion channels (an A/D channel, a D/A channel); (6) a halogen desk lamp used for generating the perturbations, during the night or constant daylight conditions.



Fig. 3. The experimental stand [8]

The illuminance on the desk surface represents the sum of the illuminances produced by the halogen lamp no. 3 (electric light) and the halogen lamp no. 6 (the perturbation).

### 3. The CMAC controller

#### 3.1 The CMAC algorithm

The CMAC algorithm can be decomposed into two separate mappings. The first is a nonlinear, topology conserving transformation that maps the network’s input into a higher dimensional space, in which only a small number of the variables have a non-zero output. Thus the CMAC produces a sparse internal representation of the input vector. The designer must specify a generalisation parameter,  $\rho$ , which determines the number of non-zero variables in the hidden layer, and also specifies the size of the network’s internal region that influences its response. The second transformation gives a linear combination of the non-zero variables from the hidden layer.[3]

The network’s  $n$ -dimensional input is denoted by  $x$  (Fig. 4), and network’s sparse internal representation is denoted by the  $p$ -dimensional vector  $a$  (Fig. 4); this vector is called the transformed input vector or the basis function output vector. The transformed input vector,  $a$ , has as elements the outputs of the basis functions in the hidden layer and the output,  $y$ , of the CMAC network is formed from linear combination of these basis functions. [3]

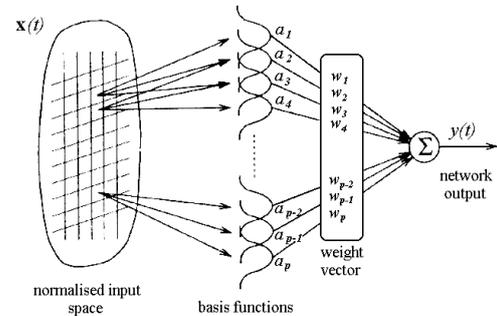
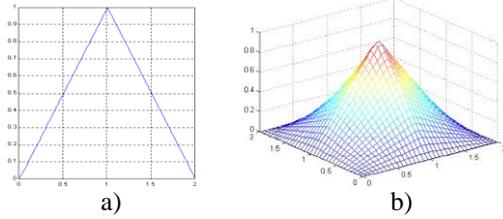


Fig. 4. General structure of CMAC[3]

### 3.2 The CMAC controller

The controller is implemented using a CMAC network with two inputs and one output. The design choices for CMAC controller are:

- control error ( $\varepsilon$ ) and the change in control error ( $\Delta\varepsilon$ ) are the input variables of controller;
- the variation in command ( $\Delta U$ ) is the output variable of the CMAC controller;
- the basis functions are implemented with triangular function type (Fig. 5a);

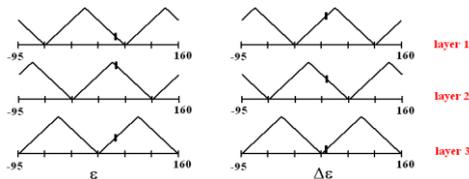


**Fig. 5.** Triangular basis function: a) one-dimensional; b) two-dimensional

- the generalization parameter of CMAC,  $\rho$  (which give the number of the over layers attached to the universe of discourse of each input and an important information about the width of the base of each basis function) is set to value 3;

- on each layer, the basis functions (one-dimensional functions) attached to  $\varepsilon$  and those attached to  $\Delta\varepsilon$  are together connected using the linguistic *and* resulting bi-dimensional basis functions. Using the product operator to implement the connector *and* the bi-dimensional basis functions will have the shape like the one depicted in Fig. 5b;

- each over layer attached to an input variable it was divide in equal intervals by 5 interior knots (Fig. 6); the width of each basis function was determined multiplying the width of an interval by the value of generalization parameter  $\rho$ ;



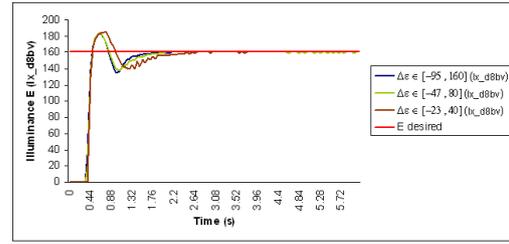
**Fig. 6.** The displacement of the basis functions on the overlays attached to the inputs of controller when  $d=(1,2)$  and  $\rho=3$  (the ticks mark represent the output of the basis functions when  $\varepsilon = 60 (lx_{d8bv})$  and  $\Delta\varepsilon = 40(lx_{d8bv})$ )

- the overlay displacement vector is set to  $d=(1,2)$ , the basis functions displacement for each input of controller are presented in Fig. 6;

- the universe of discourse for the input variable  $\varepsilon$  was settled to the range  $[-95,160] (lx_{d8bv})$ . Considering the desired illuminance on the desk surface settled to  $160 (lx_{d8bv})$  the universe of discourse of  $\varepsilon$  was determined as follow: replacing in (2) the value of desired illuminance and the minimum

converted value of measured illuminance with 8 bits D/A converter ( $0(lx_{d8bv})$ ) it is getting the maximum value  $160(lx_{d8bv})$  of universe of discourse and replacing in (2) the value of desired illuminance and the maximum converted value of measured illuminance with 8 bits D/A converter ( $255(lx_{d8bv})$ ) it is getting the minimum value  $-95(lx_{d8bv})$  of universe of discourse.

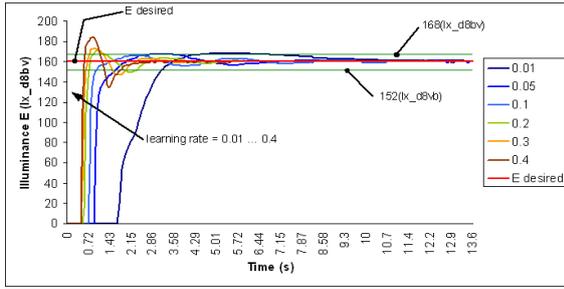
- the universe of discourse for the input variable  $\Delta\varepsilon$  are set experimentally to the range  $[-95,160] (lx_{d8bv})$ . The decrease of the universe of discourse range will produces an increase of the transient response duration (Fig. 7).



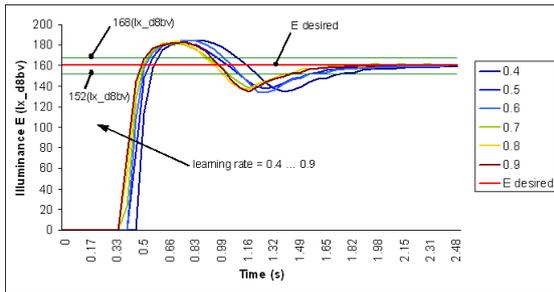
**Fig. 7.** Step response family of the ALCS when the range of the universe of discourse of  $\Delta\varepsilon$  is variable ( $GE=GCE=GU=1$ , learning rate  $\gamma = 0.9$ )

### 4. The controller tuning and experimentation

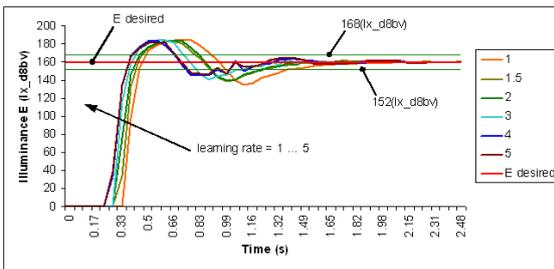
Before the tuning of CMAC controller the influence of the learning rate,  $\gamma$ , was studied. In Fig. 8, Fig. 9, Fig. 10 and Fig. 11 is presented the step response family of the ALCS when the learning rate is variable and the scaling gains are constant ( $GE=GCE=GU=1$ ). The desired illuminance is set to  $E_{desired} = 160 (lx_{d8bv})$ . In literature, for learning rate, are recommended values in the interval  $(0.0, 1.0)$ . In present paper, the range of the learning rate values is set to  $(0, 9]$ . Analyzing the step response family from Fig. 8, the increase of the learning rate from 0.01 to 0.4 will produce the decrease of the time delay and the increase of the overshoot. The increase of the learning rate, from 0.4 to 5 (Fig. 9 and Fig. 10), will produce the decrease of the time delay and the decrease of the transient response duration, but the overshoot is relative constant, having values around 12.5% of  $E_{desired}$ . If the learning rate  $\gamma$  is increased over 4 (Fig. 10 and Fig. 11) the ALCS instability signs appear, the ALCS will be totally instable for  $\gamma = 9$ . In order to avoid the big overshoot values and the instable behaviour of the ALCS, the range of the learning rate values is recommended to be  $(0, 0.3]$ .



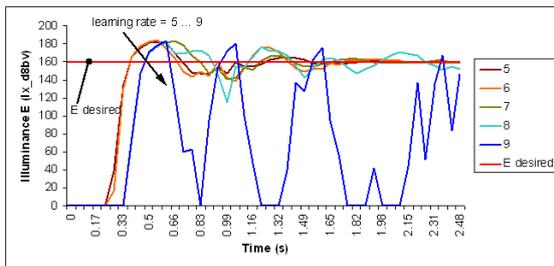
**Fig. 8.** Step response family of the ALCS ( $GE=GCE=GU=1$ , learning rate  $\gamma = 0.01 \div 0.4$ )



**Fig. 9.** Step response family of the ALCS ( $GE=GCE=GU=1$ , learning rate  $\gamma = 0.4 \div 0.9$ )



**Fig. 10.** Step response family of the ALCS ( $GE = GCE = GU = 1$ , learning rate  $\gamma = 1 \div 5$ )



**Fig. 11.** Step response family of the ALCS ( $GE = GCE = GU = 1$ , learning rate  $\gamma = 5 \div 9$ )

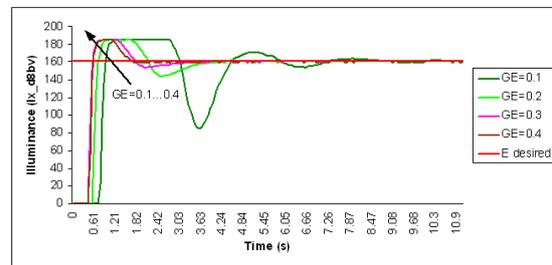
The tuning of the CMAC controller was done using the tuning via universes of discourse method presented in [9] and applied in lighting control in [6],[7]. This type of tuning method was selected due to the similarities of the CMAC network with a fuzzy system. In case of a fuzzy system the input variables

values after fuzzyfication (usual singleton fuzzyfication [5]) are mapped to membership degree given by membership functions.

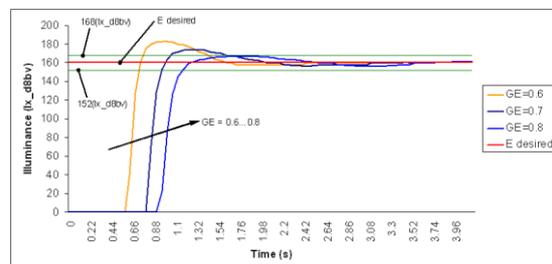
In case of a CMAC network the input variables values are mapped to the values given by the basis functions; these values can be viewed like the membership degree used in case of the fuzzy system.

The tuning procedure is applied as follow: generate a step response family of the ALCS keeping constant two scaling gains and modifying the third scaling gain. In this manner are generated three types of step response families (one for GE variable, one for GCE variable, and one for GU variable).

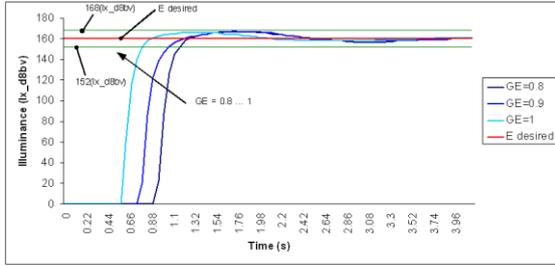
In Fig. 12, Fig. 13, Fig. 14, Fig. 15 and Fig.16 are presented the step response families when the scaling gain GE is variable and the scaling gains GCE and GU are constant. For all these step response families the scaling gain GU is set to 0.5. The scaling gain GCE is set to 0.1 for the step response families from Fig. 12, Fig. 13, Fig. 14, Fig. 15 and is set to 1 for the step response family from Fig. 16. The increase of the GE from 0.1 to 0.4 (Fig. 12) will reduce the time delay and the transient response duration; the increase of GE from 0.6 to 0.8 (Fig. 13) will increase the time delay and the transient response duration and, will reduce the overshoot; the increase of GE from 0.8 to 1 (Fig. 14) will reduce the time delay and the transient response duration and, will limit the overshoot to values smallest as 5%  $E_{desired}$ ; the increase of GE over unity (Fig. 15 and Fig. 16), will increase the overshoot.



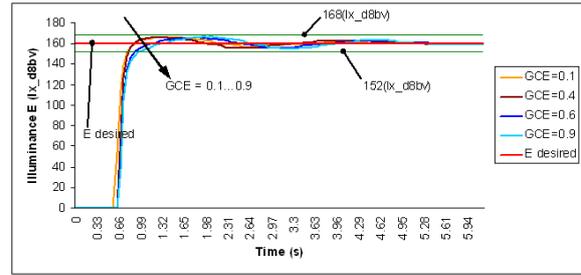
**Fig. 12.** Step response family of the ALCS ( $GE = 0.1 \div 0.4$ ,  $GCE = 0.1$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



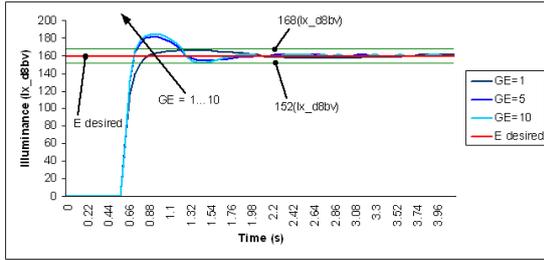
**Fig. 13.** Step response family of the ALCS ( $GE = 0.6 \div 0.8$ ,  $GCE = 0.1$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



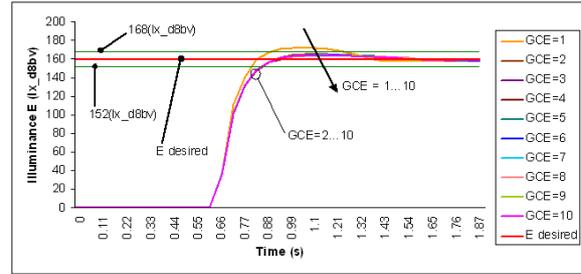
**Fig. 14.** Step response family of the ALCS ( $GE = 0.8 \div 1$ ,  $GCE = 0.1$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



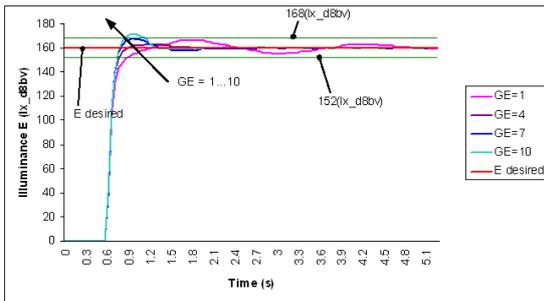
**Fig. 17.** Step response family of the ALCS ( $GE = 1$ ,  $GCE = 0.1 \div 0.9$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



**Fig. 15.** Step response family of the ALCS ( $GE = 1 \div 10$ ,  $GCE = 0.1$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



**Fig. 18.** Step response family of the ALCS ( $GE = 10$ ,  $GCE = 1 \div 10$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )



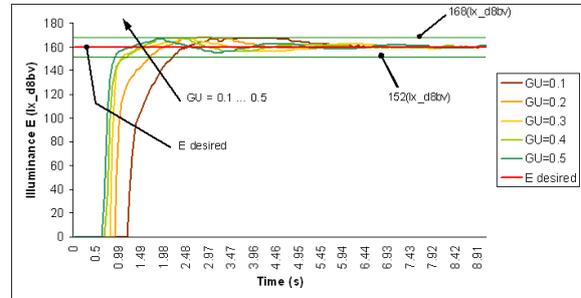
**Fig. 16.** Step response family of the ALCS ( $GE = 1 \div 10$ ,  $GCE = 1$ ,  $GU = 0.5$ , learning rate  $\gamma = 0.3$ )

In order to avoid big values for overshoot the proportion between scaling gains  $GE$  and  $GCE$  is recommended to satisfy the relation (Fig. 15 and Fig.16):

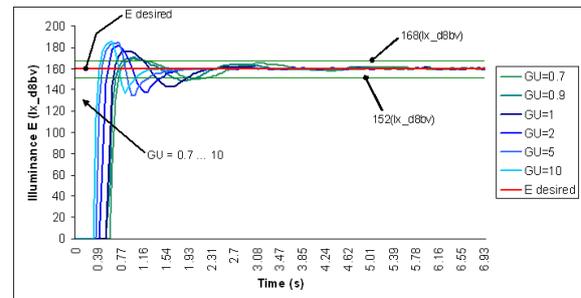
$$\frac{GE}{GCE} \leq 10. \quad (4)$$

In Fig. 17 and Fig. 18 are presented the step response families when the scaling gain  $GCE$  is variable and the scaling gains  $GE$  and  $GU$  are constant. Analyzing these step response families the increase of the  $GCE$  will decrease the overshoot and the transient response duration.

In Fig. 19 and Fig. 20 are presented the step response families when the scaling gain  $GU$  is variable and the scaling gains  $GE$  and  $GCE$  are constant. Analyzing the step response families from Fig.19 and Fig. 20, the increase of the  $GU$  will decrease the time delay and the transient response duration, and increase the overshoot.



**Fig. 19.** Step response family of the ALCS ( $GE = 1$ ,  $GCE = 1$ ,  $GU = 0.1 \div 0.5$ , learning rate  $\gamma = 0.3$ )



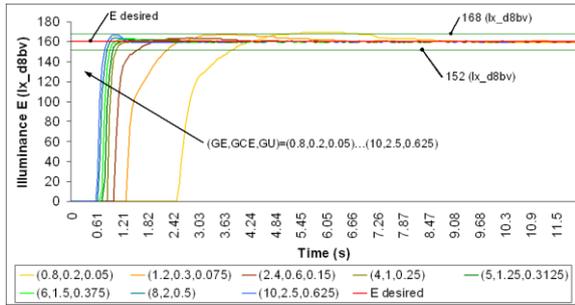
**Fig. 20.** Step response family of the ALCS ( $GE = 1$ ,  $GCE = 1$ ,  $GU = 0.7 \div 10$ , learning rate  $\gamma = 0.3$ )

Analyzing the step response of the ALCS when  $GE=4$ ,  $GCE=1$  and  $GU=0.5$  (Fig. 16) the overshoot was calculated to be  $1.875\%E_{desired}$ . Keeping  $GE=4$ ,  $GCE=1$  and selecting the  $GU=0.25$  a couple of proportions between the scaling gains were calculated to be:

$$\frac{GE}{GCE} = 4 \quad (5)$$

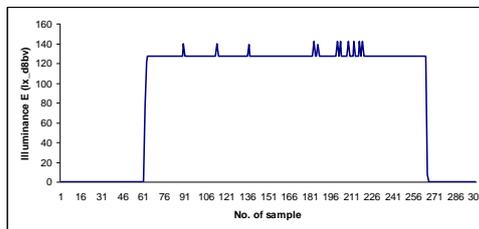
$$\frac{GCE}{GU} = 4. \quad (6)$$

Keeping constant the proportions (5), (6) and modifying  $GE$ , a step response family of the ALCS was acquired and depicted in Fig. 21. Analyzing the step response family the overshoot will be smaller or equal to  $5\%E_{desired}$  even the scaling gains are variable. The increase of scaling gains will reduce the time delay and the transient response period.

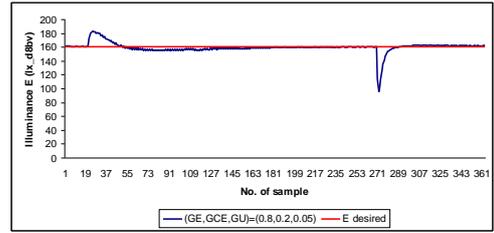


**Fig. 21.** Step response family of the ALCS  $((GE, GCE, GU) = (0.8, 0.4, 0.05) \div (10, 2.5, 0.625))$ , learning rate  $\gamma = 0.3$

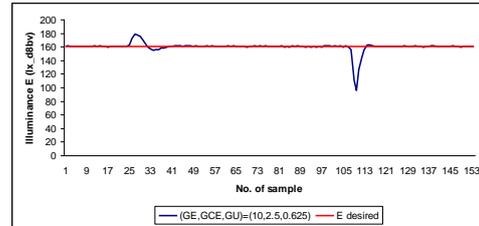
From Fig. 21 it was chosen two sets of values for  $(GE, GCE, GU)$ :  $(0.8, 0.2, 0.05)$  and  $(10, 2.5, 0.625)$ . For these sets of values was tested the stability of the ALCS when the illuminance on the desk surface was disturbed by the additional illuminance produced by the incandescent lamp denoted by 6 in Fig. 3. The shape of the disturbance signal is presented in Fig. 22. The ALCS is stable, according to Fig. 23 and Fig. 24.



**Fig. 22.** The shape of perturbation signal



**Fig. 23.** The behavior of the ALCS  $((GE, GCE, GU) = (0.8, 0.2, 0.05))$  when is applied the perturbation signal



**Fig. 24.** The behavior of the ALCS  $((GE, GCE, GU) = (10, 2.5, 0.625))$  when is applied the perturbation signal

## 5. Conclusions

Using proper proportions between scaling gains applied to the inputs and the output of CMAC controller, the ALCS meet the desired performances. The modifications of one scaling gain (maintaining the same proper proportion between the scaling gains) allows the user of the ALCS to set the own reaction of the system to the variation of daylight. This feature allows the use of the ALCS for two different types of applications. The first type, represent those applications where, from the human eye perception point of view, the illuminance must be constant (for example design laboratory). The second type, represent those applications where, the users need to feel the changes in the light environment due to the natural variations of daylight (for example office and home applications).

Unfortunately the finding of the proper proportions between the scaling gains requests time effort (the step response families was acquired during night conditions) and experience.

## References

- [1]. **Albus J.S.** - *New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)*, Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, September 1975, p. 220 – 227, 1975;
- [2]. **Almeida P.E.M., Simões M.G.** - *Parametric CMAC Networks: Fundamentals and Applications of a Fast Convergence Neural Structure*, IEEE Transactions On Industry Applications, Vol. 39, No. 5, p. 1551-1557, 2008;
- [3]. **Brown M., Harris C.** - *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall International (UK) Limited, 1994;
- [4]. **Fuller R.** - *Neural Fuzzy Systems*, Abo Akademi University, Abo, 1995;
- [5]. **Grif H.S.** - *Contribuții privind tehnici de inteligență artificială în conducerea sistemelor de iluminat interior. PhD. Thesis*, The Technical University Cluj Napoca, Romania, 2006;
- [6]. **Grif, H.S., Gligor A., and Bucur D.** - *Fluorescent Daylight Control System based on B-spline Like Network with Gaussian-Type Basis Functions*, The 4th International Conference ILUMINAT 2007, Cluj-Napoca, Romania, p. 161-166, 2007;
- [7]. **Grif H.S., Pop M.** - *Fluorescent Daylight Control System based on Neural Controller*, The 4th International Conference ILUMINAT 2007, Cluj-Napoca, Romania, p. 171-177, 2007;
- [8]. **H. S. Grif.** - *Halogen Daylight Control System Based on CMAC Controller*, 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2010), Cluj-Napoca, pp. 84-88, 2010;
- [9]. **Passino K.M., Yurkovich S.** - *Fuzzy Control*, Addison Wesley Longman, Inc, 1998;
- [10]. **Qin, T., Chen, Z., Zhang, H., Li, S., Xiang, W and Li, M.** - *A Learning Algorithm of CMAC Based on RLS*, Neural Processing Letters, No. 19, p. 49–61, 2004;
- [11]. **Qin, T., Zhang, H., Chen, Z., and Xiang, W.** - *Continuous CMAC-QRLS and Its Systolic Array*, Neural Processing Letters, No. 22, p. 1–16, 2005;
- [12]. **Sayil, S., Lee, K.Y.** - *A Hybrid Maximum Error Algorithm with Neighborhood Training for CMAC*, Proceedings of the 2002 International Joint Conference on Neural Networks, Hawaii, p. 165-170, 2002;
- [13]. **Tham, C.K.** - *Modular On-Line Function Approximation for Scaling Up Reinforcement Learning*, University of Cambridge, Cambridge, England, 1994;
- [14]. **Wang S., Lu H.** - *Fuzzy system and CMAC network with B-spline membership/basis functions are smooth approximators*, In Springer-Verlag, Soft Computing, No. 7, p. 566–573, 2003;
- [15]. **Zhang, L., CAO, Q., LEE, J., and ZHAO, Y.** - *A Modified CMAC Algorithm Based on Credit Assignment*, Neural Processing Letters, No. 20, p. 1–10, 2004.