Fuzzy Logic Based Look-Up Table Regulator

Manuel Olivares [1] olivares@aii.upv.es

Pedro Albertos pedro@aii.upv.es

Antonio Sala asala@isa.upv.es

Universidad Politécnica de Valencia
Dept. of Systems Engineering and Control, Apdo. 22012
E-46071 Valencia, Spain. Phone: +34 96 3879570, Fax: +34 96 3879579

Abstract

In this paper, a look-up table control strategy derived from a fuzzy logic controller is developed for output regulation of a SISO linear plant with delay subject to load changes and transfer between different operating points. The strategy considers an off-line trained look-up table for rejection of an unity load step change and another one trained for unity reference step tracking combined in a switched LUT regulator. Also it is shown that any load or reference step change can be rejected or tracked by properly scaling the pre and post processing gains of its corresponding LUT controller. In this way, a non linear regulator is implemented.

Keywords: LUT control, self-organising controllers, disturbance rejection.

1 INTRODUCTION

Adaptation techniques for fuzzy systems have been extensively studied with the goal of developing practical learning controllers. Here, an iterative learning control strategy that gradually improves the reference and load step response is presented. This problem has been studied by several authors (see for example [2]). The approach in this paper utilises a PI fuzzy look-up table controller [5,6], enhancing the adaptive algorithm [1,3,4] to deal with the regulator problem (disturbance compensation).

The Self Organising mechanism yields an improved controller starting from a linear one with poor disturbance rejection performance, by repeated off-line iterations. After a number of learning runs, a satisfactory performance LUT is chosen.

The paper is organised as follows: The SOC structure with first order reinforcement learning is presented. Good performance look-up tables (LUT) for unity step reference and unity step load changes are obtained by experimental training. Then the disturbance detection and the scaling method to compensate any step like reference and load change are discussed. The switching between LUTs is easily determined by the a priori knowledge of the set point changes. Finally, some simulating results and conclusions complete the document.

2 SOC STRUCTURE

To improve the closed loop response by means of a learning algorithm, a LUT based discrete PI & SOC structure, is implemented. It consists of two main blocks (fig. 1), a discrete dynamic LUT and a SOC algorithm. Details about the SOC structure can be seen in [1]. In addition to the usual 2 input 1 output discrete static LUT, the discrete dynamic LUT has a third input and a second output.

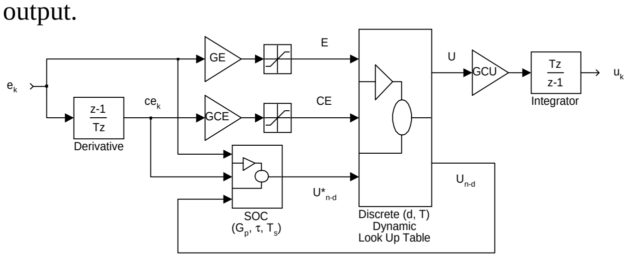

Figure 1. Discrete PI & SOC structure

The third input U^*_{n-d} is used to update the content of the table element taken "*d*" samples before, and the second output U_{n-d} supplies a copy of the delayed control action signal, sampled each T_S *sec*. The regular U output is computed from the E and CE inputs using linear interpolation. The central element of the table has been kept invariant, and *zero* to assure a good steady state behaviour. The SOC block produces U^*_{n-d} by the reward/penalty p_n, the reinforcement required to correct any bad performance detected from e_n and ce_n.

$$p_n = G_p\left(e_n + \tau ce_n\right)T_s \qquad (1)$$

$$U^*_{n-d} = p_n + U_{n-d} \qquad (2)$$

where G_p is the reinforcement gain and τ the dynamic balance between the error and its change.

[1] On leave from Dept. of Electronics, Universidad Técnica F. Santa María, Valparaíso, Chile.

3 LINEAR PLANT WITH DELAY

A linear third order plus delay process subject to input disturbances is considered

$$G(s) = e^{-9s}\frac{1}{s(s+1)^2} \quad (3)$$

and a fuzzy-PI discrete controller has been designed. The suitable control gains for unity set point transfer and unity step load change responses, shown in eq. (4) and eq. (5), have been chosen to visit 80% of the standard [-100,100] input universe ranges. Sampling time $T=1sec$ is chosen.

$$GE = 80, GCE = 1600, GCU = 3.125 \cdot 10^{-5} \quad (4)$$

$$GE = 3.9440, GCE = 78.8804, GCU = 6.3387 \cdot 10^{-4} \quad (5)$$

Thus, the initial linear LUT controller is equivalent to a discrete PI controller with $K_c=0.05$ and $T_i=20sec$. The set point transfer and load change error trajectories are depicted in fig. 2.

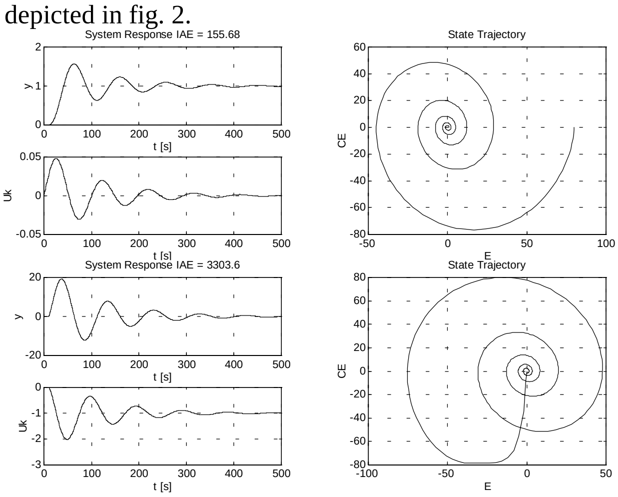

Figure 2. Unity Reference and Load Responses

4 SET POINT TRANSFER

The training schema is shown in fig. 3.

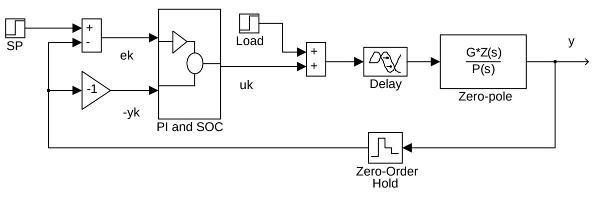

Figure 3. SOC training schema

The PI & SOC block, fig. 1, has to improve continuously the LUT control transfer for unity set point change, with zero load. Notice that for the derivative input, "$-y_k$" is taken instead of "e_k", to avoid the initial derivative kick. Starting with a linear LUT control, the discrete SOC block makes changes on table elements according to the reinforcement calculated every T_s *sec* during several learning experiments. In that way, the IAP index is reduced, indicating that better performance is obtained on each learning run, until over-learning occurs.

$$IAP = T_s \sum_{n=0}^{100} |p_n| \quad (6)$$

The results obtained using $d=4$, $T_s=5sec$, $G_p=2$ and $\tau=60sec$ are shown in table 1 and figure 4.

Table 1. SOC and LUT Servo Results

Table	IAP	IAE
Linear	-	155.68
T1	4551.6	185.98
T2	3125.0	110.72
T3	2408.0	102.53
T4	1467.3	180.71
T5	1530.5	112.90
T6	1011.3	138.52
T7	1227.8	100.47
T8	707.8	141.26
T9	1223.8	121.02
T10	827.2	113.33

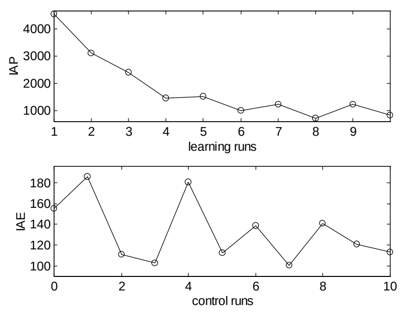

Figure 4. IAP and IAE Servo Trends

The leftmost column indicates the table that has been saved on each learning run, where *T8* appears to be the best option, considering the IAP index. To confirm that, each table is used in a static LUT control scheme, and the IAE performance index is evaluated.

$$IAE = T \sum_{k=0}^{500} |e_k| \quad (7)$$

The IAE results don't match exactly those of the IAP obtained during the training. This is due, in part, because of the learning equation considers the error and its derivative, but the IAE index takes into account just the error. From the IAE column, *T7* is the best table. Figure 5 shows the control surface and the control response to a unity reference step using *T7*.

4.1 MULTIPLE STEP TRACKING

The trained LUT for reference step can be utilised for any step input, if GE, GCE and GCU are accordingly varied. It can be easily proved that for a linear plant (with/without delay) and a LUT trained for a unity step reference change, this LUT gives an equivalent response for a twice

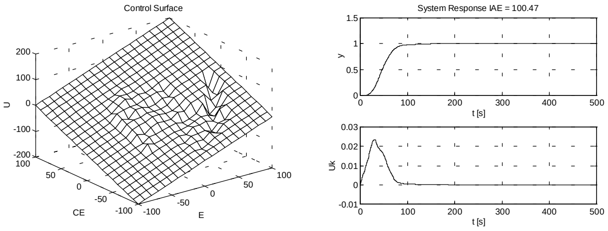

Figure 5. Control Surface for Reference Step Response

step reference change if GE and GCE are half reduced and GCU is doubled. In this way, the state "walks" over the same trajectory as for a unity step. Considering the step signal definition (eq.(8)), any step like reference change can be described as a weighted sum of *N* delayed steps.

$$u(t-\delta)=\begin{cases}1 & \forall t-\delta \geq 0\\ 0 & \forall t-\delta < 0\end{cases} \tag{8}$$

$$SP(t)=\sum_{i=1}^{N} a_i \cdot u(t-\delta_i) \tag{9}$$

Then, in order to use the same selected LUT for non unity step reference changes, GE, GCE and GCU have to be modified according to:

$$GE_i = GE / a_i\ , GCE_i = GCE / a_i\ ,\quad GCU_i = GCU \cdot a_i \tag{10}$$

Here, the satisfactory table obtained above is used for multiple transfer between set points varying GE, GCE and GCU at the proper time, with a_i={1,1.5,-2.5} as shown in in figure 6.

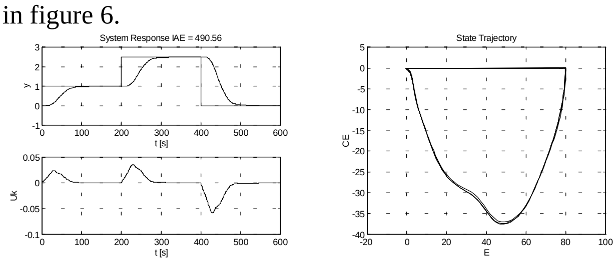

Figure 6. Multiple Reference Results

The interval between changes, $\Delta\delta_i$, has been considered larger than the plant settling time. This constraint could be also removed.

5 LOAD STEP REJECTION

The same training schema used for reference step tracking (fig. 3), is now used to obtain the best LUT for unity load step rejection, with zero reference. Starting with a linear LUT, a new table is generated on each learning run. The results obtained using *d=4*, T_s*=5sec*, G_p*=0.1* and τ*=30sec* are shown in table 2 and figure 7. From the IAP column, *T7* appears to be the best option, but from the IAE column, *T3* is the best table. As before, and for the same reasons, the IAE and IAP results don't match exactly.

Figure 8 shows the control surface and the control response to a unity load step using *T3*.

Table 2. SOC and LUT Regulation Results

Table	IAP	IAE
Linear	-	3303.6
T1	3238.8	2923.0
T2	2668.2	2540.1
T3	2274.4	2178.4
T4	1998.1	2533.2
T5	2079.7	3075.4
T6	2205.6	2770.3
T7	1950.7	2576.9
T8	2170.9	3372.6
T9	2368.8	3212.2
T10	2342.9	4474.8

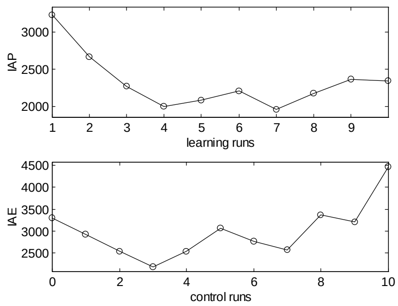

Figure 7. IAP and IAE Regulation Trends

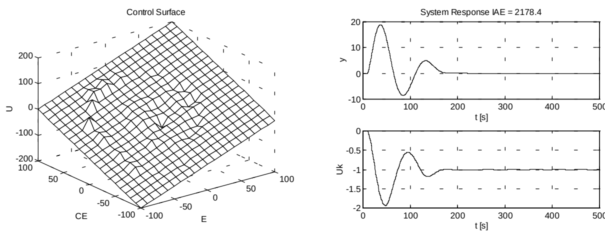

Figure 8. Control Surface for Load Step Response

5.1 LOAD MAGNITUDE ESTIMATION

Due to the nature of disturbances, it is impossible to know its magnitude previously, even if a simplified load step type disturbance is assumed. Thus, a load step magnitude estimator is needed to do a proper gain scaling, as seen in section 4.1, in order to make the state "walks" over the same trajectory as for a unity step. Similarly to the step like reference signal, any step like load change can be described as a weighted sum of *N* delayed steps.

$$P(t)=\sum_{i=1}^{N} a_i \cdot u(t-\delta_i) \tag{11}$$

For linear systems is valid that the load step magnitude a_i is proportional to the first non zero plant output change, after a disturbance occurs., that is

$$a_i=\begin{cases}\alpha(y_l - y_{l-1}) \text{ if } d=1 \text{ and } y_l - y_{l-1} \neq 0\\ 1 \text{ if } d=0\end{cases} \tag{12}$$

where "α" is a proportional factor, that can be easily determined from an experiment, and "l" denotes the memorised "k" sampled time corresponding to the first non zero plant output after a disturbance is detected by means of the "d" detector flag. If there is a reference change, its detection is depicted in section 6.

$$d = \begin{cases} 1 \text{ if } \left| e_k - e_{k-1} \right| > \varepsilon \\ 0 \text{ if } \left| e_k - e_{k-1} \right| < \varepsilon \end{cases} \qquad (13)$$

Thus, the satisfactory table obtained above for unity load step is used for multiple load step rejection, varying GE, GCE and GCU accordingly to eq. (10). For that, a scale factor signal is generated considering eq. (12) and eq. (13). Figure 9 shows some results.

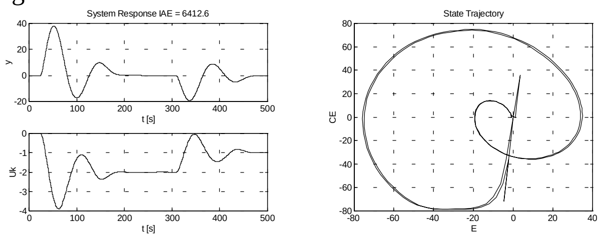

Figure 9. Multiple Load Results

6 COMBINED REGULATOR

Once having selected the tables, *T7* for reference step (LUTr) and *T3* for load step changes (LUTp), they can be used in a combined structure, as suggested in figure 10.

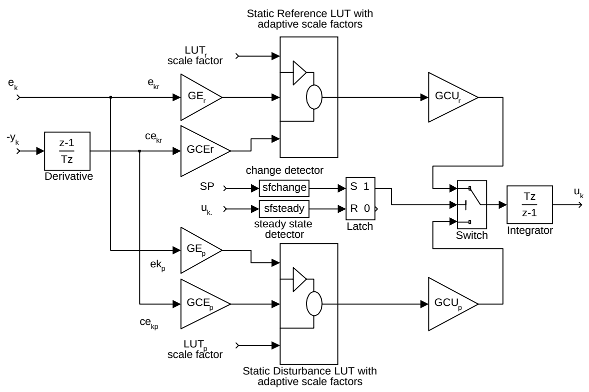

Figure 10. Combined LUT Regulator Structure

The strategy considers that reference changes are user demanded and load changes are process demanded, in such a way, LUTp control will be normally used unless a reference change is required. In that case, a switching from LUTp to LUTr control is made. In both cases, a scaling factor adaptation is needed if the magnitude of the step changes are different from unity.

The switching algorithm is as follows: LUTr control must be activated when a reference change occurs, and must be deactivated when the system reaches the steady state. The reference change signal condition is easily detected evaluating its discrete derivative:

$$\text{if } \left| SP_k - SP_{k-1} \right| > \varepsilon \text{ then SP has changed} \qquad (14)$$

Due to the plant delay, the system steady state condition is detected evaluating the controller output signal derivative, instead of the plant output

$$\text{if } \left| u_k - u_{k-1} \right| < \varepsilon \text{ then Steady State Reached} \qquad (15)$$

The first condition triggers the "set" condition for LUTr control and the second one triggers the "reset" condition for LUTp control. Figure 11 shows the results obtained with the combined regulator.

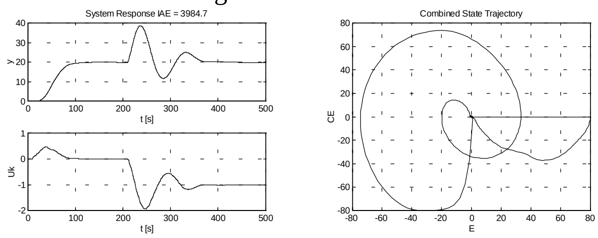

Figure 11. Combined Regulator Results

Conclusions

In this paper, a combined reference and load fuzzy look-up table based regulator has been designed. This regulator can cope with step type reference and load magnitude changes, varying the corresponding LUT scale factors. The tables itself have been obtained by means of a first order learning algorithm, trained for a specific linear plant with delay, giving 35% IAE reduction from the linear PI IAE. Also, with this work it is shown that satisfactory LUT elements are system structure dependent and LUT scale factors are system input dependent.

References

[1] Jantzen, J.. The Self-Organising Fuzzy Controller, TU of Denmark, Department of Automation, http://www.iau.dtu.dk/~jj/pubs/soc.pdf, *Technical report* no. 98-H 869, 1998.

[2] Moore, K. L.. Iterative Learning Control for Deterministic Systems. *Springer Verlag*, 1993.

[3] Olivares, M. and J. Jantzen. Fuzzy Self-Organising Control of a Pendulum Problem, *European Symposium on Intelligent Techniques ESIT'99*, Chania-Crete, Greece, 1999.

[4] Olivares, M., P. Albertos and A. Sala. Fuzzy Logic Based Look-Up Table Servo Control, *Advanced Summer Institute ASI'99*, Leuven, Belgium, 1999.

[5] Procyk, T.J. and E.H. Mamdani. A Linguistic Self Organizing Process Controller, *Automatica*, vol. 15, pp.15-30, 1979.

[6] Shenoi, S., K. Ashenayi and M. Timmerman. Implementation of an On-line Adaptive Fuzzy Controller in Low-end Hardware. *Engng. Applic. Artif. Intell.*, vol. 7(5), pp. 533-543, 1994.