A NEW EFFICIENT SELF-ORGANISING FUZZY LOGIC CONTROL (SOFLC) ALGORITHM USING A DYNAMIC PERFORMANCE INDEX TABLE

Q. Lu and M. Mahfouf

Department of Automatic Control and Systems Engineering
The University of Sheffield, Mappin Street,
Sheffield S1 3JD
The United Kingdom
Tel: +44 (0) 114 222 5607
Emails: q.lu@sheffield.ac.uk, m.mahfouf@sheffield.ac.uk

Abstract: The *"raison d'etre"* of Self-Organising Fuzzy Logic Control (SOFLC) algorithms is the Performance Index table which normally issues the adequate corrections to the low-level control given certain performance criteria. In the standard SOFLC architecture, the Performance Index table is generic, fixed *a priori*, and is of a 'grid-partition' structure making the whole scheme inefficient in terms of computational complexity and performance. In this paper we propose a new SOFLC architecture whereby the Performance Index table is 'dynamic', of a free structure, and starting from an empty table. The architecture includes 3 mechanisms for optimising the rules of the Performance Index table and enhancing the performance as well as the robustness of the algorithm in terms of disturbance rejection and noise. Results of experiments on a non-linear muscle relaxation process showed that the proposed control scheme was superior to the standard SOFLC algorithm in terms of performance and robustness against parameter variations, stochastic activity, and sensitivity to the selection of scaling factors.

Keywords: Fuzzy logic, self-organising, uncertainty, non-linear, performance, optimisation.

1. INTRODUCTION

The first Self-Organising Fuzzy Logic Controller (SOFLC) was proposed by Procyk and Mamdani (1979), and includes a control policy that can change with respect to the process it is controlling and the environment it is operating in. This is what is often called an adaptive or learning controller to stress that its operation relies on the acquisition of past experience, i.e. a suitable combination of past control actions and the effects they produced. One interesting characteristic of this controller is that it strives to improve its performance until convergence to a predetermined quality. In doing so, the SOFLC performs two tasks at the same time which are: a) observe the environment while ensuing the appropriate control action and b) use the results of these control actions to improve them even further. A considerable amount of work has been carried out using the SOFCLC, but perhaps, the most interesting application of the controller is Sugeno's fuzzy car (Sugeno and Murakami, 1985). The car has the ability to learn how to park itself.

Due to its on-line features and model-free architecture, the SOFLC algorithm has been used in many application areas, demonstrating that it possesses the advantage of controlling uncertain, mathematically ill-understood, non-linear and time-varying systems. However, it does suffer from several drawbacks, including the fact that the original fuzzy rules for the Performance Index measure has been left practically unchanged and expected to work with any system that exhibits second-order over-damped dynamics! The basic reason for this is that such a table was not easy to alter, either individually or in blocks of cells (Mahfouf *et al.*, 2000), leading therefore to problems relating to long convergence caused by inaccurate modification of rules and increased computation complexity with the increase of the dimensionality of the system under consideration.

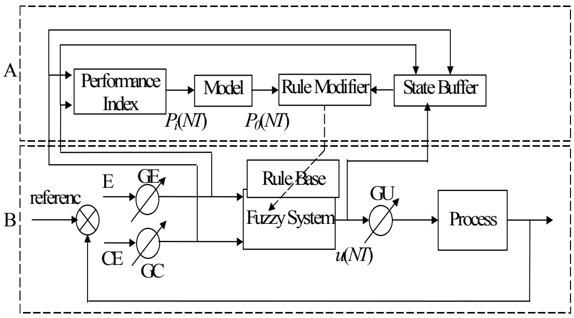
In this paper, we propose a new self-organising architecture that is similar to the standard SOFLC as proposed by Procyk and Mamdani except that: 1. The Performance Index table used for issuing the necessary corrections to the low-level simple fuzzy logic controller is of 'free-partitioning' rather than of 'grid-partitioning'. 2. The rules of this Performance Index table are elicited on-line via a computationally efficient Genetic Algorithm (GA) (Goldberg, 1989) using a new mechanism which also ensures that the solution is optimal. In the optimisation process, in addition to the error and its derivative, the change of the control effort is also included in the fitness evaluation to guarantee a constant control signal during the steady-state period. Using the credit assignment concept of the polarity of the trajectory error is also used as a criterion for reward and punishment of good and bad actions respectively. The remainder of this paper is organised as follows: Section 2 is a simple introduction to the standard SOFLC idea. The GA based rule elicitation procedure is reviewed in Section 3, where the on-line evaluation and credit assignment are both explored. Section 4 describes the simulation results carried-out on the non-linear muscle relaxation process, whereas Section 5 explores several extensions to the algorithm to enhance its performance and its robustness against noise and process parameter variability. Finally, Section 6 concludes this research study with several comments on the new proposed algorithm and plans for future research.

2. A BRIEF DESCRIPTION OF THE STANDARD SOFLC ALGORITHM

As shown in Fig. 1, the SOFLC architecture includes two basic levels: the first level 'B' is a simple fuzzy controller, whereas the second level 'A' consists of the self-organising mechanism, acting as a monitor and evaluator of the low-level controller performance. The self-organizing part 'A' consists of three main blocks: a performance measure, a model estimator, and a rule modification. The measure of system performance is represents a critical step in producing a successful 'learning' controller. Usually, two physical features, including the system output Error (E) and the Error Change (CE) are measured to establish a performance decision table. The model estimation block is used to find the relationship between the system output performance and the control input. The performance measure is then employed to calculate the correction value relating to each fuzzy rule based on the estimation model. As far as the rules modification procedure is concerned, it can be assumed that, the control action at sample time $'nT-mT'$ has contributed most to the process performance at the sampling instance $'nT'$. Thus, the rule included reads as follows:

$$E(nT-mT) \rightarrow CE(nT-mT) \rightarrow U(nT-mT) + P_i(nT) \qquad (1)$$

IF E is … and CE is … THEN *Control* is …
where $P_i(nT)$ is issued by the Performance Index table, E is the error and CE is the change of error.



A: 'The learning' Part
B: 'The simple fuzzy logic control' Part

Fig. 1 A schematic diagram depicting the architecture behind the standard SOFLC algorithm

The new rules with a new antecedent will be added into the fuzzy rule bank, while rules with a similar antecedent and a different consequent would be used to modify the existing rules in the rule bank.

In the self-organizing level 'A', obtaining $P_i(nT)$ represents the main task that would affect the performance of the closed loop system directly. As a result, several issues need to be considered, including:

1. How to measure the tracking performance accurately?
2. How to combine the operators' or the experts' knowledge?
3. What improvement can be made to obtain a good control performance, including a quick convergence,

a resistance to noise, and a low computational burden?

The original and classic method used to calculate $P_i(nT)$ consists of using a linguistic Performance Index table. The typical rule in this table should read:

IF change-in-error is *NB*, AND error is *NB*, THEN the performance index based change is *NB*.

As already stated, most performance index tables are fixed and are of a 'grid partition' configuration, but in our proposed new architecture, the Performance Index table is dynamically changed from instant $'nT'$ to instant $'nT+1'$ and its configuration is 'free'. The next section will outline the philosophy behind the new proposed scheme.

3. A DYNAMIC PERFORMANCE INDEX TABLE USING AN ON-LINE GENETIC ALGORITHM

The main purpose of the GA in the proposed algorithm is to generate a suitable Performance Index table output at each sampling instant with only one chromosome from a population being evaluated at that instant. The other chromosomes in the population are then estimated based on the relationships between the GA individuals. Since many of the processes involved have a stochastic influence from external as well as internal mechanisms, such 'imprecise' fitness estimation can however be deemed very reasonable.

3.1 The GA Process Encoding

In the proposed algorithm, a maximum of '*M*' sequences of the GA algorithm are used to optimise the Performance Index rules, where the number '*M*' is decided by the number of cells in the Performance Index table. Each GA sequence includes a small size of individuals, and does not depend on other GA sequences. At every sampling instant, only the individuals, which represent the consequence of the activated Performance Index cell, will carry out one iteration of the GA-based operations, i.e., evolution, selection, crossover and the creation of new individuals under certain conditions sequentially. The other *'M-1'* sets of GA individuals should be kept unchanged. Fig. 2 gives an example of a Performance Index table with 25 rules. In our case, there is a maximum of 25 sequences of independent GAs which attempt to optimise their corresponding rule cells and each GA sequence includes 5 individuals at the most. At the sampling instant $'nT'$, the GA in cell *'X44'*, which is indexed to produce the modification value $P_i(nT-mT)$, will be activated. Five individuals of **set B**, which correspond to the cell *'X44'* will experience one iteration of the GA–based operations, after which the estimated fitness values will provide the various rankings of these individuals. Once the cell *'X44'* is indexed by the SOFLC algorithm again, the individual '0000101' for instance, with the highest fitness, would be chosen to produce the modification value. Meanwhile, the cell *'X32'* is invoked to infer the modification $P_i(nT)$ for the low level simple fuzzy logic at this sampling instant. The shaded individual '0000111' in **set A** has the highest fitness value at that time, so it will be chosen to calculate the modification $P_i(nT)$.

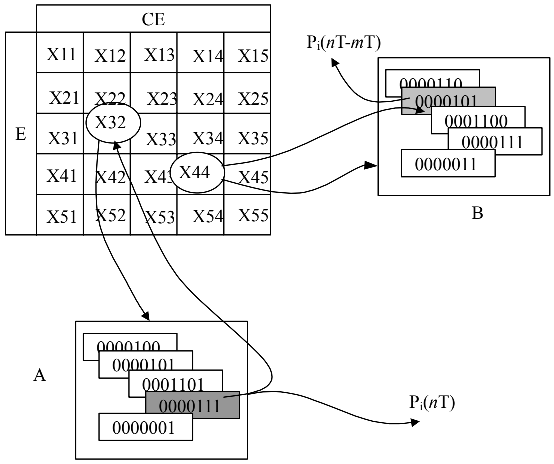


Fig. 2 The coding of the GA sets

3.2 Evaluation of the GA Population

3.2.1 Performance Evaluation

In order to improve the system's performance, the performance-adaptive learning controller needs information as to how well it is performing at each sampling interval. The idea of "reinforcement learning" is stated as follows: "If a particular action is associated with a satisfactory state of affairs then the tendency to reproduce that action in a similar situation should be enhanced". As shown in Fig. 3, all possible cases for the output threads in a particular sampling interval are classified into 3 Groups, A, B, and C which correspond to the position of the output in relation to the set-point as well as its gradient.

A simple expression for the predictive error estimation can be given as:

$$\widehat{e}(nT+kT) = e(nT) + KT\dot{e}(nT) \qquad (2)$$

where $e(t)$ is the process error in the current sampling instant $'nT'$, and k is the number of the intervals predicted ahead.

3.2.2. Credit Assignment

The performance evaluation in Section 3.2.1 represents a measurement for the individual I_i which is included within the real feedback environment. In order to compare the fitness values of all the individuals in this generation, it is crucial to rate the usefulness of the other '$N-1$' individuals in the activated set of GA X_{jl}. The credit assignment in this section carries out this task with the **reward/penalty** concept. Since different individuals in one generation correspond to different degrees of modification of the low-level fuzzy logic rules, one can infer the possible performances (*Good*, *Divergence* or *Over-Shoot*) of the individuals $I_k (k = 1, 2, \ldots, i-1, i, i+1, \ldots, N)$ from the individual I_i.

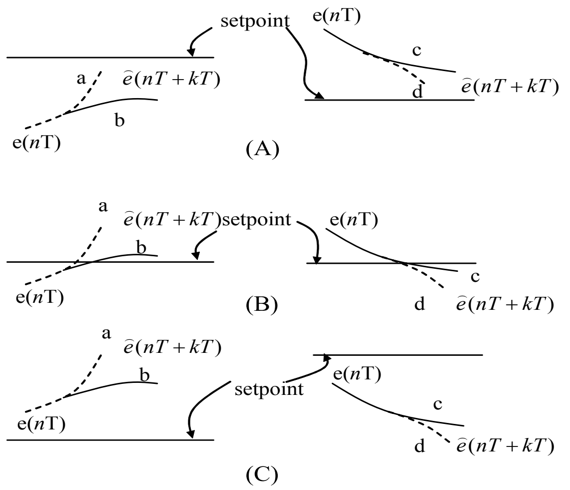


Fig. 3 Instant state performance classification

4. SIMULATION RESULTS

Simulation results of the implementation of the on-line GA and the acquisition of the rules relating to the dynamic Performance Index table on a non-linear medical process are presented here. The necessary transfer function components for the model used in these studies have been obtained via various ways. The drug considered in the model for humans is atracurium, which produces muscle relaxation. Muscle relaxation can be monitored continuously via evoked EMG responses. The overall nonlinear model can be described by the following Wiener nonlinear structure (Mahfouf and Linkens, 1998):

$$G(s) = \frac{X_E(s)}{U(s)} = \frac{K_1(1+T_4 s)e^{-s}}{(1+T_1 s)(1+T_2 s)(1+T_3 s)}$$
$$E_{eff} = \frac{X_E^{\,2.98}}{(X_E^{\,2.98} + 0.404^{2.98})} \qquad (3)$$

where,

$K_1 = 1, T_1 = 34.4\,\text{min}., T_2 = 4.8\,\text{min}., T_3 = 3.1\,\text{min}.,$
$T_4 = 10.6\,\text{min}.$

X_E is the drug concentration in the blood and E_{eff} is the effect the drug produces (muscle relaxation or actual paralysis).

The simulation study used a step length of 0.1 and the sampling interval of 1 minute. The initial condition is 0% relaxation. In our research, the low level simple fuzzy controller includes five fuzzy sets for the tracking error and the change of tracking error: negative big (NB), negative small (NS), zero (O), positive small (PS), positive big (PB), which were defined in a universe of discourse, which was equally partitioned. For consistency, the Performance Index table is also partitioned into 25 cells, each corresponding to one fuzzy rule. At the initial state, the Performance Index table is empty and then filled with 25×5 GA individuals randomly generated; each individual codes the consequence part of one Performance Index rule. A set point profile of 85%, 65% and then 85% is used and changed every fixed period. The controller used in this series of experiments is of an incremental type (linguistic Performance Index).

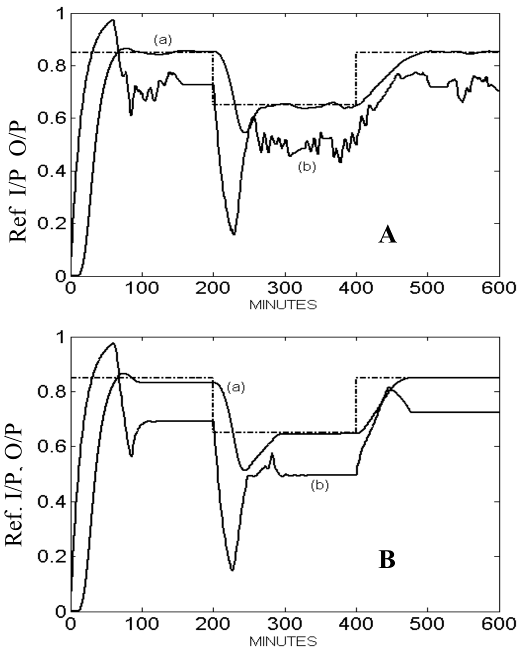


Fig. 4 Closed-loop control simulation results using the new SOFLC algorithm (A); and the standard SOFLC algorithm (B) (a): output signal ; (b) input signal

As shown in Fig. 4, two different SOFLC schemes are tested and compared, which are the standard SOFLC with a fixed Performance Index table and the proposed SOFLC with a dynamic Performance Index table respectively. The proposed dynamic SOFLC scheme starts with an empty Performance Index table at each run. From Fig. 4 it can be seen that, compared with a standard SOFLC, the proposed SOFLC with a dynamic Performance Index table can make the system track the set point with a faster rise time, especially during the time phase 200~300 min. This merit can be attributed to the proposed algorithm's ability of searching a more suitable Performance Index table output for the fuzzy rules modification block.

Furthermore, system (B) seems to generate an offset during the time phase 100~200 min despite the integral action and the choice of an optimal set of scaling factors already included in the algorithm. Despite this, it is reckoned that the proposed dynamic SOFLC still has its own limitation in that it seems to find it difficult to reach a steady level in the control signal at time phases of 300~400 min and 500~600 min. This is believed to be due to the fact that at that instant, the control effort modification was small and the populations of the invoked Performance Index cells were almost all identical. Section 5 will focus on extending the algorithm to improve its performance.

5. ALGORITHM EXTENSIONS FOR PERFORMANCE IMPROVEMENTS

5.1. Micro-GA and its restarting mechanism

The simulation results shown in Fig. 4 prove that, the new SOFLC algorithm makes the system work well under closed-loop conditions albeit with some limitations. Further analyses of the results reveal that, the small size of individuals (5 or 6 generally), which is consistent with the optimization technique known as Micro-Genetic Algorithm (MGA) (Krishnakumar, 1989), coupled with the credit assignment mechanism, make the system generate abrupt changes. In order to overcome this, an iterative strategy was included in which the resolution of the micro GA-based solution is improved significantly at each iteration. In order to avoid premature convergence, the MGA includes a distinctive feature to maintain a sufficient variety of genetic information, that is, when the population is about to converge closer to one solution, it is recomposed and restarted as shown in Fig. 5.

Once the decomposition of the populations is triggered in the MGA (iteration $k1$, iteration $k1+k2$ in Fig. 5), new individuals are generated randomly covering the new area, where the best individual is taken as the centre, and the boundary will be zoomed at. simple form is expressed as follows:

$$R_i = \begin{cases} R_0 * \alpha & \text{if } \textit{in steady state area} \\ R_0 & \text{elsewhere} \end{cases} \tag{4}$$

where R_0 is the initial search interval, R_i the newly constrained search interval, and $0 < \alpha < 1$ is a constant value.

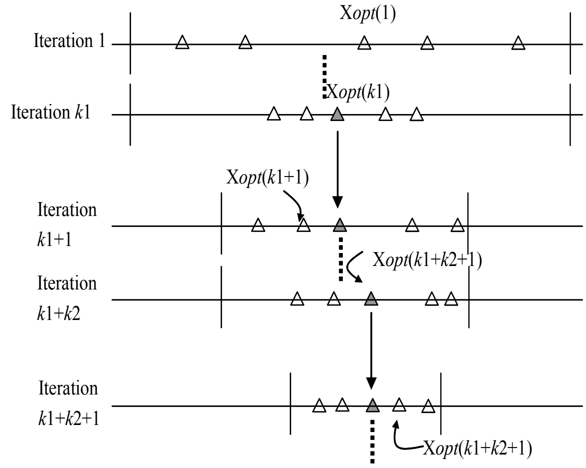


Fig. 5 Iterative search for the 'optimal' solution.

As shown in Figs. 6 and 7, our improved SOFLC with the dynamic Performance Index table performed better than the standard SOFLC, even when a sudden disturbance is added.

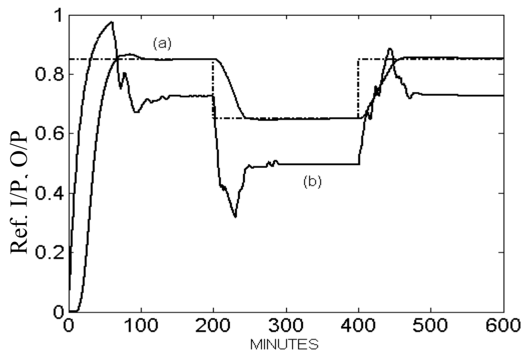


Fig. 6 Closed-loop control simulation results using the new improved SOFLC algorithm; (a): output signal ; (b) input signal

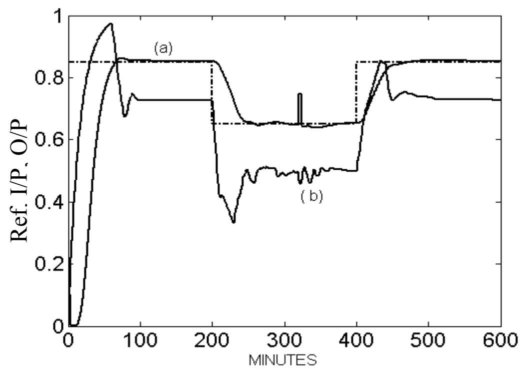


Fig. 7 Closed-loop control simulation results with added noise; (a): output ; (b) input

5.2. Robustness of the new SOFLC algorithm in the stochastic case

When a noise sequence of 5% is superimposed to the system output, the SOFLC with a dynamic

Performance Index table fails to control the output as shown in Fig. 8. This is mainly due to the fact that, the information on the output trend (Error Change) is corrupted and hence the fuzzy controller infers the wrong output. A common method for dealing with noise under closed loop control, is to include a band pass filter before the measured signal is passed to the controller. However, a new method based on fitting a function to a certain number of output data points is proposed here.

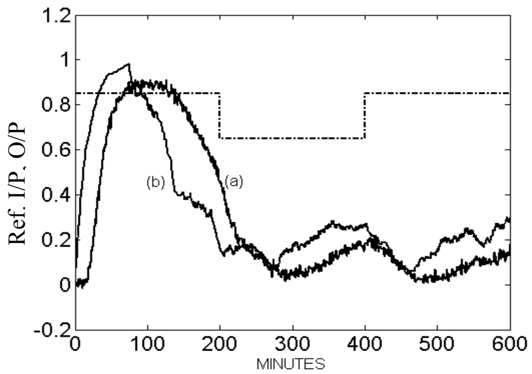


Fig. 8 Closed-loop control simulation results when noise is added to the output; (a): output signal ; (b) input signal

Such a function can take the format of a polynomial function of the following form:

$$p(x) = \sum_{k=1}^{m} c_k \varphi_k(x) \qquad (5)$$

where $\varphi_k(x)$ is the kth order polynomial function. According to the least square principle, the factors $c_1, c_2, c_3, \cdots, c_m$ are subject to the following least-square problem:

$$\frac{\partial E}{\partial c_j} = \sum_{k=1}^{n} 2[\sum_{i=1}^{m} c_i \varphi_i(x(k)) - z(k)]\varphi_j(x(k)) = 0 \qquad (6)$$

For the history tracking error data which deteriorate with the noise $z(k)$, $(k = 1, 2, \ldots, t)$, t is the current sample time iteration), n samples (that is, $z(k)$, $k = t-n+1, t-n+2, \ldots, t)$ are used to estimate the current tracking error $\hat{z}(t) = \sum_{i=1}^{m} c_i \varphi_i(x(t))$ with the polynomial curve fitting method, and then the tacking error change can be replaced by $\hat{z}(t) - \hat{z}(t-1)$. A 3rd order polynomial was found to be adequate in this case as shown in Fig. 9.

6. CONCLUDING REMARKS

This paper has outlined a new approach for real-time learning of the Performance Index table in the much-celebrated Self-Organising Fuzzy Logic Control (SOFLC) architecture. The variant of the GA optimisation technique, the Micro Genetic Algorithm (MGA), is applied to modify the consequent part of the Performance Index fuzzy rules. The results presented in this paper and others, which have not been included due to space constraints, have shown that the proposed scheme is superior to the standard SOFLC algorithm in terms of computational complexity, performance in the transient and steady-state phases, as well as robustness against disturbances, system parameter variations, and the choice of the scaling factors which are so crucial in fuzzy logic based control. Future research on this subject work will include extensions to the multivariable case.

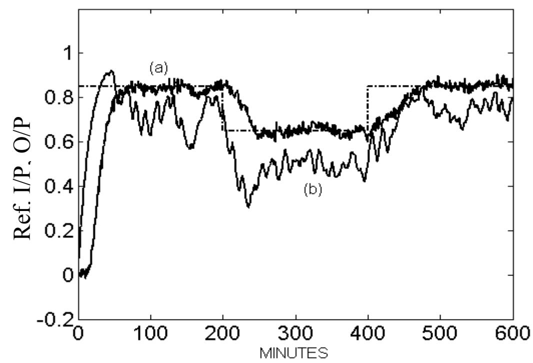


Fig. 9 Performance of the extended SOFLC algorithm including a 3rd-order filter; (a): output signal ; (b) input signal

REFERENCES

Goldberg, D.E. (1989), Genetic algorithms in search, optimisation, and machine learning, Wokingham, Addison-Wesley.

Krishnakumar. K (1989), Micro-genetic algorithms for stationary and non-stationary function optimisation, *SPIE, Intell. Contr. Adapt. Syst*, **1196**, pp 289-296.

Mahfouf, M. and Linkens, D.A. (1998), Generalised Predictive Control and Bioengineering, London: Taylor and Francis.

Mahfouf, M., Linkens, D.A., and Abbod, M.F. (2000), Muti-objective genetic optimisation of GPC and SOFLC tuning parameters using a fuzzy-based ranking method, *IEE Proceedings on Control Theory Application*, **147**(3), pp 344-354.

Procky, T.J., Mamdani. E.H. (1979), A linguistic self-organizing process controller, *Automatica*, **15**(1), pp 15-3.

Sugeno, M. and Murakami, K. (1985), An experimental study on fuzzy parking control using a model car, *in Industrial Application of Fuzzy Control, Sugeno, M. (Ed.), Amsterdam: North Holland*, pp 125-138.