ELectronics EXpress

IEICE Electronics Express, Vol.7, No.13, 892–898

A novel self-tuning scheme for fuzzy logic elevator group controller

Nasrudin Abd. Rahim, Hew Wooi Ping, and Jafferi Jamaludin[a)]

Center of Research for Power Electronics, Drives, Automation and Control, University of Malaya (UMPEDAC),
Lembah Pantai, 50603 Kuala Lumpur, Malaysia
a) *jafferi@um.edu.my*

Abstract: In this paper, an intelligent controller based on a novel self-tuning fuzzy logic framework for elevator control applications is presented. The proposed controller utilizes the measured performance data of the elevator control system i.e. average waiting time (AWT), to initiate self-tuning through the adjustment of the membership functions, and the modification of the rule set. The controller's adaptation to the system's environment that is based on the system's measured performance eliminates the heavy dependence on the predicted traffic patterns for adaptation. From computer simulation, the controller's effectiveness is satisfactorily verified.

Keywords: elevators, self-tuning, fuzzy logic control, performance evaluation

Classification: Electronic instrumentation and control

References

[1] C. B. Kim, K. A. Seong, H. L. Kwang, and J. O. Kim, "Design and implementation of a fuzzy elevator group control system," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 3, pp. 277–287, May 1998.

[2] R. H. Crites and A. G. Barto, "Improving elevator performance using reinforcement learning," *Advances in Neural Information Processing Systems 8*, eds. D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, MIT Press, Cambridge, MA, pp. 1017–1023, 1996.

[3] A. Fujino, T. Tobita, K. Segawa, K. Yoneda, and A. Togawa, "An elevator group control system with floor-attribute control method and system optimization using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 44, no. 4, pp. 546–552, Aug. 1997.

[4] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu, J. Hu, and S. Markon, "A double-deck elevator group supervisory control system using genetic network programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 4, pp. 535–550, July 2008.

[5] M. L. Siikonen, "Elevator group control with artificial intelligence," *Systems Analysis Laboratory*, Helsinki University of Technology, Helsinki, Finland, Research Rep. A67, 1997.

[6] R. Gudwin, F. Gomide, and M. Andrade Netto, "A fuzzy elevator group controller with linear context adaptation," *Proc. FUZZ-IEEE98, IEEE*

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

IEICE Electronics Express, Vol.7, No.13, 892–898

World Congress Comput. Intell., Anchorage, Alaska, USA., pp. 481–486, 1998.
[7] J. Koehler and D. Ottiger, "An AI-based approach to destination control in elevators," *AI Magazine*, vol. 23, no. 3, pp. 59–78, Fall 2002.
[8] R. K. Mudi and N. R. Pal, "A robust self-tuning scheme for PI- and PD-type fuzzy controllers," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 1, pp. 2–16, Feb. 1999.

1 Introduction

An elevator group control system basically comprises four components [1]; namely hall call buttons at the elevator foyers, car call buttons inside elevator cars, car controllers, and a group controller. Among the four components, the group controller plays the most important role in the system i.e. to select the most suitable cars to serve hall calls, in a way that improves the overall performance of the system. The system's performance is evaluated based on several criteria, in which the most common is the passenger waiting time whose values are to be minimized as much as possible.

Artificial intelligence techniques have been explored to search for viable solutions to improve the performance of the control system, and to enhance its effectiveness and efficiency. Among the techniques employed include neural networks [2], genetic algorithms [3], genetic network programming [4], and fuzzy logic [1, 5, 6]. These techniques are inefficient and impractical because they are computationally expensive, and slow. Although fuzzy logic is considered more efficient in the use of computational resources than the other techniques, it suffers from failing to produce desirable dispatching decisions at all time, owing to the inaccurate predictions made on the passenger traffic patterns. Since predictions accurate enough to provide useful information about traffic flows are impossible [7], it is the purpose of this paper to introduce a new way to employ fuzzy logic for elevator control applications with the addition of a self-tuning scheme based on measured performance evaluation.

2 The proposed group controller

The proposed group controller employs fuzzy logic with a self-tuning scheme to adjust its control strategy based on the current performance of the elevator control system which is widely evaluated using the passenger waiting time. The waiting time is defined as the time taken starting from the activation of the hall call signal when the hall call button is pressed, until the deactivation of the same signal when a car arrives at the floor concerned. Fig. 1 shows a schema of the elevator group control system's structure with the use of the proposed self-tuning fuzzy logic group controller. The proposed controller utilizes the number of stops the cars have to make before reaching the hall call floor (s), the cars' corresponding distance traveled (d), and the cars' corresponding response time spent (t) as the input variables

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

IEICE Electronics Express, Vol.7, No.13, 892–898

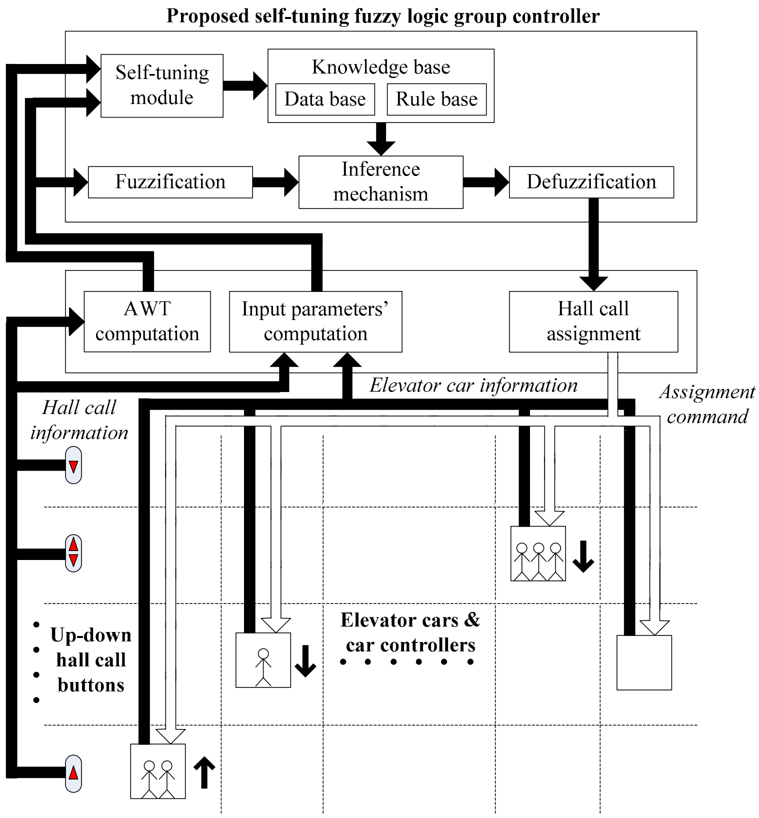

Fig. 1. Structure of the elevator group control system with the proposed self-tuning fuzzy logic group controller.

for fuzzy assessment. By fuzzification, fuzzy values of the variables are generated, before they are evaluated through the use of fuzzy rules in max-min inference scheme. The result of the inference engine is then converted into a performance index value (p) by centroid defuzzification method. Each car is given a performance index value after fuzzy evaluation, and the one with the highest is then selected to answer the hall call in evaluation.

3 Self-tuning scheme

A self-tuning fuzzy logic controller is a type of an adaptive controller that is able to adapt to changes in the system's environment [8]. In the proposed controller, self-tuning acts as a corrective action carried out to direct the actual performance to be as close as the desired performance. A correction is realized by adjusting the membership functions of the input variables, and by modifying the fuzzy rule set employed. Fig. 2 details the self-tuning mechanism of the proposed group controller shown in Fig. 1 earlier.

3.1 Adjustment of membership functions

The purpose of adjusting the membership functions is to specify the relevant range in the universe of discourse that can best describe the fuzzy values at a particular time. This is necessary as the passenger traffic behavior is changing all the time. As an illustration, a car's response time of 20 s may

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

IEICE Electronics Express, Vol.7, No.13, 892–898

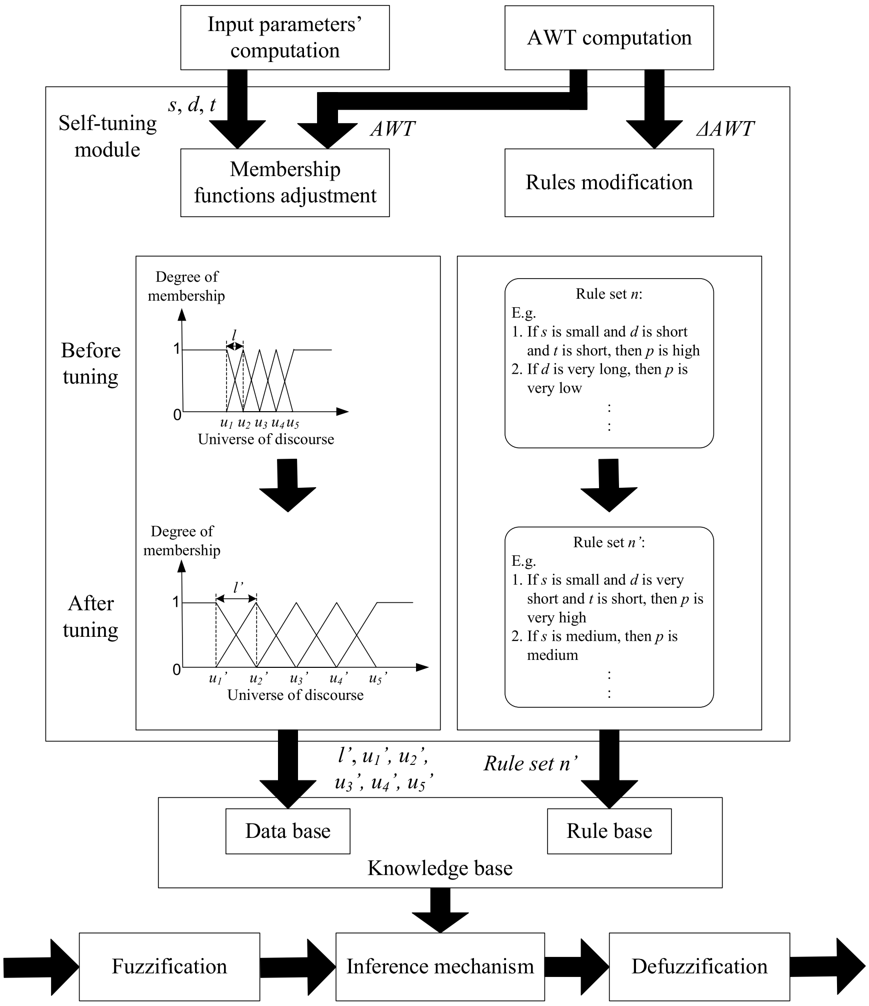

Fig. 2. Self-tuning mechanism of the proposed group controller.

be defined as long during light traffic, but can be defined as short when the traffic intensity is heavy.

In Fig. 2, to realize membership functions adjustment, four input parameters are used: s (s_1, s_2, ..., s_i), d (d_1, d_2, ..., d_i), t (t_1, t_2, ..., t_i), and AWT where subscripts 1, 2, ..., i represent the cars' numerical label. For example, car 1 corresponds to s_1, d_1, and t_1 and so on. s, d, and t have their own triangular membership functions in which five fuzzy values define the membership functions of each parameter. For instance, in case of s, fuzzy values *very short*, *short*, *medium*, *long*, and *very long* describe the membership functions.

Adjustment of the membership functions of s, d, and t is done in two ways: by changing the length of two adjacent vertices of the triangular membership functions l, and by shifting the middle value u_3 (l and u_3 are shown in Fig. 2, and l' and u_3' denote the results after adjustment). The former applies to parameters s, d, and t while the latter only involves t since for s and d, u_3 are made constants. $u_{3,s}$ (corresponds to s) and $u_{3,d}$ (corresponds to d) take the middle value of the total number of floors and the building's height respectively.

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

IEICE Electronics Express, Vol.7, No.13, 892–898

l is determined every time a hall call is needed to be assigned. It is derived using the following for the case of d:

$$l_d{}' = \frac{|d_{max} - d_{ave}|}{2} \quad \text{if } d_{max} - d_{ave} > d_{ave} - d_{min}, \text{ and} \tag{1}$$

$$l_d{}' = \frac{|d_{min} - d_{ave}|}{2} \quad \text{if otherwise,} \tag{2}$$

where $l_d{}'$ is l' corresponding to d, d_{max} is the maximum, d_{min} is the minimum, and d_{ave} is the average value, calculated by dividing the sum of all d (d_1, d_2, ..., d_i) with the number of cars (i). The same apply for $l_s{}'$ (corresponds to s) and $l_t{}'$ (corresponds to t). It should be noted that the l' obtained must be above a certain threshold that can be set as a constant to ensure that the membership functions generated do not cease to exist.

The proportional relationship between t and the waiting time i.e. short t also reflects short waiting time, can be exploited to derive new $u_{3,t}$ (corresponds to t), as represented by $u_{3,t}{}'$ in Fig. 2. By comparing the currently-measured AWT with the desired AWT during a certain traffic intensity level, $u_{3,t}{}'$ can be acquired. To illustrate further, consider a heavy traffic situation whose desired AWT is given by $AWT_{desired}$. Taking the currently-measured AWT as $AWT_{current}$ and the maximum allowable tolerance between $AWT_{desired}$ and $AWT_{current}$ as $limit$, $u_{3,t}{}'$ can be obtained using the following:

$$u_{3,t}{}' = AWT_{current} \text{ if } |AWT_{current} - AWT_{desired}| > limit, \tag{3}$$

indicating that a correction has been made and,

$$u_{3,t}{}' = u_{3,t} \text{ if otherwise,} \tag{4}$$

signaling for no correction is necessary, where $u_{3,t}$ portrays the middle value u_3 used previously. Unlike l which is continuously computed, u_3 is determined every five minutes. Once l and u_3 are obtained, u_1, u_2, u_4 and u_5 can be found using the following:

$$u_j = u_3 - (3 - j) \cdot l \qquad \text{for } j = 1, 2, 3, 4, 5 \tag{5}$$

3.2 Modification of rule set

Owing to the dynamic passenger traffic flows, the rule set employed has to be constantly evaluated to suit the current traffic condition. This is vital as in the actual traffic situations, a varying mix of different traffic intensity within a particular length of time is normally observed. For example, during a peak period, while heavy traffic intensity is regarded as the dominant traffic, there are always intances whereby medium and light traffic also exist. Therefore, to handle this traffic fluctuation, the rule set applied has to be assessed every five minutes.

Each rule in the rule set is designed in the following form: if A, and B, and C, then D. A, B, and C are the antecedents representing the statements about the three input variables (s, d, and t), while D is the consequent giving

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

ELectronics EXpress
IEICE Electronics Express, Vol.7, No.13, 892–898

the statement about the performance index (p). An example of the rules is: if s is very small, and d is very short and, t is very short, then p is very high. Six rule sets have been designed for the proposed controller to choose from for every five-minute period. As displayed in Fig. 2, to change a rule set with another, ΔAWT is referred to. ΔAWT is defined as the difference between the present AWT, and the previous AWT measured five minutes earlier. When ΔAWT exceeds certain limit, a different rule set is selected from the six rule sets for use within five minutes. During the evaluation, suitability of the six rule sets are judged, and the one with the highest p is considered as the most appropriate one, thus next employed for five minutes. After five minutes end, ΔAWT is reevaluated either to keep the current rule set, or to replace it with a different rule set for the next five-minute interval. Referring to Fig. 2, *rule set* n' will have different rule statements from those of *rule set* n which is used in the previous five minutes if correction is made i.e. rule set replacement has taken place and vice versa.

4 Performance analysis

Computer simulation is carried out as a method to test, study, and evaluate the performance of the proposed controller with the self-tuning scheme. A software simulator is developed in Visual C# platform, supported with a graphical user interface to provide virtual environment for a real elevator system. Simulation is conducted for three traffic patterns with varying traffic intensity i.e. up peak, down peak, and lunch peak. During up peak, majority passengers arrive at the ground floor to go to upper floors, while down peak mirrors the opposite to up peak. In lunch peak, heavy traffic exists in both up, and down directions.

Besides the self-tuning fuzzy logic algorithm applied in the proposed controller, two widely-implemented algorithms in the conventional group controllers are also simulated to act as the benchmarks. The *nearest car* algorithm favors the car with the shortest d, or the nearest car is always assigned to respond, while the *earliest car* algorithm chooses the car with the shortest t, or the earliest car is selected to answer the hall call. The self-tuning fuzzy logic algorithm considers the optimized balance of the shortest d, the shortest t, and the fewest s for the selection of the best car. It is worth mentioning that the conventional non-adaptive fuzzy logic algorithm is not used in the comparison due to its obvious weaknesses in dealing with the constant traffic fluctuation as the optimum operating point of the conventional fuzzy logic controller cannot be varied to suit the various traffic situations.

Table I presents simulation data for three different cases. Among the three cases, Case 3 shows the biggest improvements particularly during down peak traffic, namely 87.0% when compared to *nearest car* algorithm, and 64.6% when taking the *earliest car* algorithm as the reference. Case 2 generally portrays modest improvements, while lesser improvements are seen in Case 1. In Case 1, the improvement during down peak records a negative number when considering the *earliest car* algorithm as the benchmark. This figure

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010

ELectronics EXpress
IEICE Electronics Express, Vol.7, No.13, 892–898

Table I. Simulation results

Simulation case	Traffic pattern	Performance criterion & improvement percentage	Algorithm STFC	NC	EC
Case 1	Up peak	AWT (s)	14.7	14.9	15.8
Number of floors: 17		Improvement (%)	-	1.3	7.0
Number of cars: 4	Down peak	AWT (s)	32.3	44.7	32.0
Car speed: 2.5 m/s		Improvement (%)	-	27.7	-0.9
Car acceleration: 0.7 m/s^2	Lunch peak	AWT (s)	73.9	75.5	75.6
Car jerk: 1.4 m/s^3		Improvement (%)	-	2.1	2.2
Case 2	Up peak	AWT (s)	22.2	24.0	22.4
Number of floors: 31		Improvement (%)	-	7.5	0.9
Number of cars: 8	Down peak	AWT (s)	55.4	231.6	63.6
Car speed: 3.0 m/s		Improvement (%)	-	76.1	12.9
Car acceleration: 0.7 m/s^2	Lunch peak	AWT (s)	87.6	92.6	92.3
Car jerk: 1.4 m/s^3		Improvement (%)	-	5.4	5.1
Case 3	Up peak	AWT (s)	25.9	27.0	26.1
Number of floors: 35		Improvement (%)	-	4.1	0.8
Number of cars: 8	Down peak	AWT (s)	50.3	388.0	142.0
Car speed: 5.0 m/s		Improvement (%)	-	87.0	64.6
Car acceleration: 1.2 m/s^2	Lunch peak	AWT (s)	103.3	116.9	104.4
Car jerk: 1.6 m/s^3		Improvement (%)	-	11.6	1.1

STFC: self-tuning fuzzy logic, NC: *nearest car*, EC: *earliest car*

is insignificant because it only counts for a difference of 0.3 s as compared to the positive number in which the highest point (87.0%) represents a 337.7 s difference.

It is observed that the self-tuning fuzzy logic algorithm actually integrates the strong attributes of the two benchmark algorithms in order to improve the effectiveness of the proposed controller by producing better results than those of the benchmarks, or by maintaining the results of the better-performed benchmark algorithm in cases whereby better outcomes are not attainable. This clearly justifies the significant contribution offered by the proposed self-tuning scheme in building a strong link among the useful traits of the benchmark algorithms.

5 Conclusion

The work presented in this paper proposes an approach to improve the elevator group controller's performance by self-tuning fuzzy logic framework. The self-tuning scheme measures the elevator system's performance via the AWT, and ΔAWT to initiate self-tuning actions. By self-tuning, the measured performance results are translated into the proper adjustment of the membership functions of the input variables, and the suitable selection of the fuzzy rule set. With the proposed self-tuning fuzzy logic group controller, not only predictions on the traffic behavior are eliminated, but also the controller's sensitivity to the desired performance is improved.

DOI: 10.1587/elex.7.892
Received March 27, 2010
Accepted June 03, 2010
Published July 10, 2010