Fuzzy logic based controller for peak traffic detection in elevator systems

Pablo Cortés[1†], Joaquín R. Fernández[1], José Guadix[1], Jesús Muñuzuri[1]

[1] University of Seville
Grupo Ingeniería Organización.
Escuela Superior Ingenieros, Camino de los Descubrimientos s/n.
Sevilla 41092. SPAIN
[†]E-mail: pca@esi.us.es
[†] URL: http://io.us.es/P.Cortes/

Abstract

Vertical transportation refers to the problem that arises when a passenger wishes to travel by lift between the floors of a building. Controllers are installed in lifts that aim to maximise specific criteria such as passenger waiting time, the energy consumption, or the handling capacity. However, in order to make a correct car-call allocation, a traffic analysis must be previously carried out in the building. In fact, peak traffic can lead to dramatic waiting times during busy hours in the case of no modification of the car-call allocation rules. This paper presents a fuzzy logic based controller for peak traffic detection in elevator systems. The controller is validated with the most authorised traffic patterns in the industry for office buildings, namely the CIBSE Guide, and the Strakosch and Siikonen traffic patterns. The fuzzy logic controller demonstrate a suitable performance providing adequate traffic pattern identification for all cases. The controller was proved to be robust and reliable when tested with patterns depicting sudden or smooth changes, and with high or low maximum and minimum peaks.

Keywords- vertical transportation, elevator, fuzzy logic, peak traffic

1. Introduction

Vertical transportation refers to the problem that arises when a passenger wishes to travel by lift between the floors of a building. The situation takes place when the passenger requests a lift by pressing a landing call button installed on every floor or selects its destination by pressing the corresponding button located inside every deck. The elevator controller receives the call and identifies which car of the elevator group would be most suitable for such call. The system goal to be solved through allocation is to select a lift for each call that will minimise a preselected cost function representing a criteria or group of criteria such as the passenger waiting time, the total percentage of large waiting times, the ride times or the energy consumption amongst others.

Most of the controller algorithms are preconfigured and make use of performance rules that are applied depending on the type of traffic in the building at that time. The elevator group control system therefore needs to comprise some form of traffic-type detector. This can be carried out from using a simple timer (which usually leads to a bad performance due to the lack of flexibility) up to the use of most sophisticated IA methodologies and technological devices.

Traffic patterns represent the demand and behaviour of passengers wanting to use a lift to travel from their origin to their destination. Depending on the specific form of the traffic pattern, different behaviours are usually characterised. Figure 1 illustrates a typical traffic pattern in an office building, following the classic theory (Barney, 2003). It shows the number of up and down landing calls that are registered during the working day.

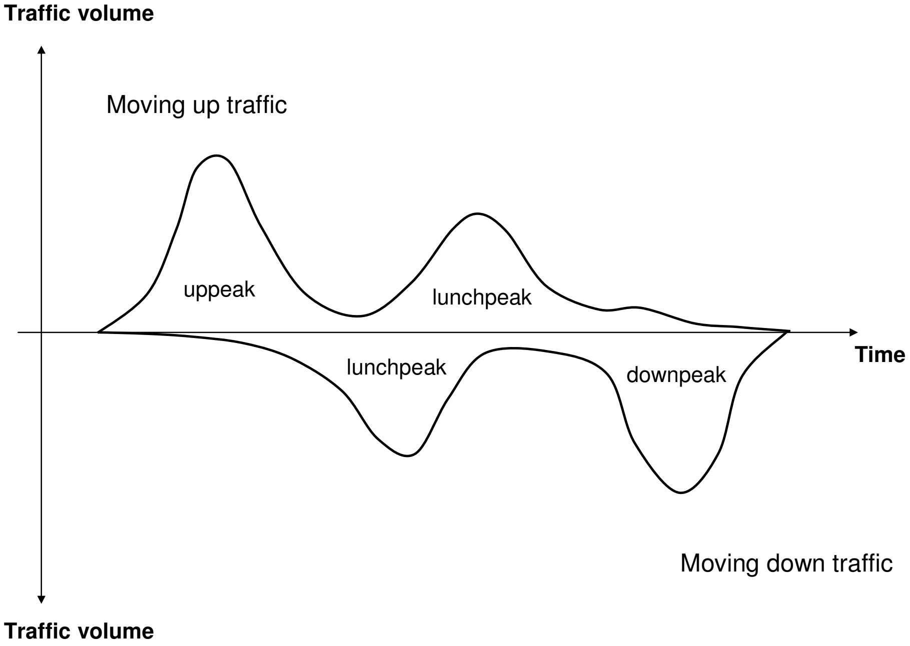

Figure 1. Traffic pattern in an office building

Normally, at the start of the day there is a larger than average number of up landing calls. These are due to the building's workers arriving to start work. This stage is called uppeak traffic. Later in the day there is the opposite phenomenon, and a larger than average number of down landing calls takes place. This corresponds to the building's population wanting to go home after the working day. This traffic pattern is called downpeak. In the middle of the day there are two joint phenomena, due to the appearance of up and down peaks. Figure 1 depicts a situation of people leaving the building for lunch and people coming back after lunch. This period has been called midday or lunchpeak traffic. Finally, the rest of the day can be characterised for a constant low demand (usually around 4% of the total population) in both directions. This period has been called the interfloor traffic. Interfloor traffic can be either balanced or unbalanced depending on whether the demand and/or the destination are heavily located only in a few floors or not.

The correct determination of the period of traffic pattern being experienced by the building is a key factor, because most of the calculations to estimate the passengers' average waiting time, the round trip time and other performance measurements, depend on the identification of the corresponding peak period [references for a quick revision on performance indexes in vertical transportation are (Barney, 2003) and the (CIBSE Guide, 2004; chapters 3 and 4].

For example, if uppeak traffic situations are not identified quickly, long queues can build up in the main entrance floor of the building (typically the ground floor) and passenger waiting times will become longer. Long waiting times cause dissatisfaction with the operation of the lifts. However, the uppeak mode should not be activated unnecessarily, as a real practice rule consisting of the direct return of lifts to the entrance floors would be activated, resulting in long waiting times for passengers who are waiting at landings other than the ground floor. In such a case, calls issued from floors other than the main entrance floor obviously take longer to be served than during normal traffic.

Most elevator designs are currently based on uppeak calculations because this is the most problematic period in terms of *handling capacity* (Barney, 2003), that is in terms of quantity of passengers being transported. However when we attend to the quality of the transportation system (waiting times, ride times, etc...) uppeak period is not the most appropriate choice for the analysis because the lunchpeak traffic can in fact lead to more complex situations and has been considered the most difficult period for dispatching landing calls (see for example Barney and Peters, 2005; and Cortés *et al*. 2004).

For example, a classic widely implemented algorithm such as the *Estimated Time of Arrival* (ETA) algorithm provides an adequate performance for interfloor or uppeak situations, but its results are poor for lunchpeak or downpeak situations. Also, a common procedure consists of dividing high-rise buildings in different zones to be served by subgroups of the elevator group system. In this sense, the *Adaptive Call Allocation* (ACA) algorithm determines the building's sub-zones, depending on the peak period. In summary, when the complexity of the algorithm is increased, a quick and correct identification of the period becomes a key factor for the success of the algorithm.

So the identification of an incoming peak traffic condition involves two objectives: a quick identification, and a correct identification.

The paper continues with the second section, which includes a literature review of methods used to detect different types of traffic both in industry (patents) and scientific literature. The third section is devoted to the detailed description of the fuzzy logic controller, and the fourth section presents the experimental results using real data. The main aspects are then reviewed in the conclusion.

2. Literature review

The traditional procedure for traffic pattern identification has been the use of passenger traffic surveys. In manual surveys, observers count passengers entering and exiting the lifts. Manual surveys are normally based on one of two approaches. The first is a

Survey from main terminal, when the observers count passengers in and out of the lifts of the main terminal floor (typically the ground floor). This method is only suitable for uppeak designs that include the problems addressed previously. An alternative method is *In-car survey*, when observers are situated in the lift car and count passengers in and out at every floor. This method generates a more comprehensive survey.

Manual surveys are discussed in detail in Barney and dos Santos (1985), and the Elevator World's Guide to Elevatoring (1992). The main problem of such methods is the statistical nature of such data. Due to data being collected during one specific point in time and the fact that complex real life buildings are continuously varying and their population habits are always changing, static surveys do not provide reliable permanent data. Similarly, the human observers' method becomes impracticable for heavy traffic conditions. The identification of peak periods based on such procedures can consequently lead to incorrect dispatching algorithm policies.

An alternative to manual observers consists of the use of historical data in order to carry out a statistical analysis together with a set of rules based on threshold values. Thangavelu and Kandasamy (1993) make use of the load per car during the day to predict the traffic behaviour of the following day. The data is managed and compared to a set of threshold values that allow the peak period to be identified, taking into account the relative population of the building. Data collection is carried out during small time intervals. The proposal consequently consists of an expert rules system rather than an authentic artificial intelligence approach.

KONE has been developing several approaches based on calculus with statistical series for historical data. This is the case of the single exponential smoothing and the adaptive response rate single exponential smoothing (ARRSES) that was used by the Traffic Master System 9000. See Siikonen (1997), where the predictions are updated for each floor and journey direction. More recently, Tyni (2005) has patented an algorithm for KONE that includes statistical analysis and fuzzy rules to predict future traffic volumes.

In a similar way, Luo *et al.* (2005) presented a elevator traffic flow prediction based on statistical learning theory. They predict elevator traffic flow using least squares support vector machines, which are a kind of support vector machine with quadratic loss function.

The statistical methods include the same problem: there is a delay between the data collection and the prediction for following days or periods, meaning that no real-time peak identification is possible.

Nowadays, sophisticated equipment that includes video cameras and infrared or computer vision tracking systems allows the capture of data in real time and in a dynamic manner. This data must however be real-time managed in order to identify the traffic pattern and the peak period of the building. In this context, Fuzzy Logic plays a relevant role when dealing with control methods in cases of uncertainty, whether for input data or output decision. Siikonen and Leppala (1991) proposed a fuzzy logic algorithm to classify the traffic pattern; however features such as the ratio of incoming passengers and the ratio of outgoing passengers were not used as input variables in their work. This paper presents a fuzzy logic controller that takes into account these aspects and other significant characteristics of vertical transportation systems. The fuzzy logic algorithm is capable of processing real-time data very quickly and identifying the corresponding peak period for the subsequent controller action.

Moreover, the use of complex approaches providing outstanding simulation results are never installed in real practice due to hard-designs difficulties because they are not economically worthy. This aspect leaded us to develop a model capable of constituting a real practice implementable solution. So, the lack of need for a complex memory and the use of fuzzy logic are decisive factors that make the model easy and economically worthy to be implemented.

3. Fuzzy Logic controller model

In real life, traffic patterns can show variations and a certain level of inconsistency between *a priori* similar days. Even rapid fluctuations from one instant to another require fast peak traffic identification. Nowadays the peak traffic identification of highly populated buildings cannot be based on static predictions constructed from surveys or static statistical analysis. Even more, the use of a historical memory does not necessarily implicate a better performance and sometimes can significantly deteriorate it.

Here we propose the use of Fuzzy Logic (FL) for the dynamic identification of peaks. FL is especially useful when quick and sudden fluctuations are merged with other slow and smooth stages. In this sense, FL is used to deal with reasoning that is approximate, rather than precise. In fuzzy logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values {true (1), false (0)}, as in classic predicate logic. When linguistic variables are used, these degrees may be managed by specific functions.

FL is a robust method that does not need much input information. It can be described in the following three steps: (i) fuzzification, where the values from the inputs are converted into fuzzy values; (ii) inference process based on logic rules; (iii) defuzzification, when fuzzy variables are reconverted and a decision is taken.

The FL model we developed follows the flowchart in figure 2. The controller provides an accurate pattern forecasting just starting from some simple initial data:

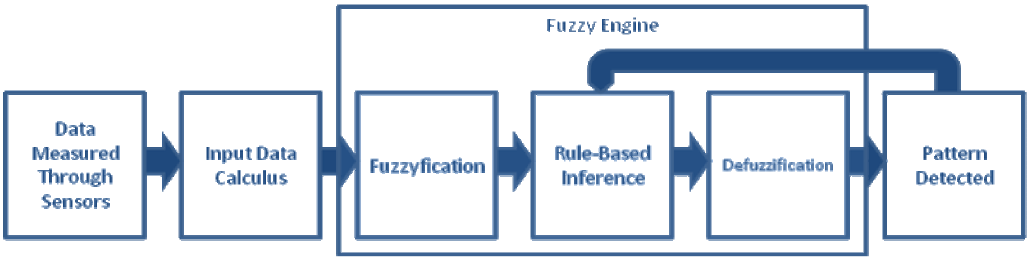

Figure 2: Flow Chart of the traffic detection approach

The use of feedback increases the robustness of the controller by aiding it to recognise more accurately the demand curve which leads to a better forecasting. It works as a reinforced recent memory. Even more, using such simple feedback avoids tedious and/or heavy computational load designs.

3.1. Data and variables of the model

The controller analyses the situation of the traffic in the building every Δt seconds, which is known as the time interval for the analysis.

The controller receives information from input sensors related to the car load each time an event (change of load or new call appears) is detected. Data related to the movement (upwards, downwards or idle) of the lift is also collected.

The management variables of the model are the following:

m_u Total weight (car load) moving up during the interval Δt.

$m_u^{'}$ Variation of the weight (car load) moving up between Δt and Δt-1.

m_d Total weight (car load) moving down during the interval Δt.

$m_d^{'}$ Variation of the weight (car load) moving down between Δt and Δt-1.

T_prev Type of traffic detected in the previous period.

Values for m_u and m_d are calculated by weighting the registered total car load in a trip during its duration (1).

$$\begin{aligned} m_u\big|_{\Delta t} &= \sum_i m_{u_i} \cdot \Delta t_{u_i} \\ m_d\big|_{\Delta t} &= \sum_j m_{d_j} \cdot \Delta t_{d_j} \end{aligned} \qquad (1)$$

Figure 2 illustrates a typical trip process in a vertical transport system where up calls and down calls take place and the cars have to serve those calls. The case corresponding to one car is represented here.

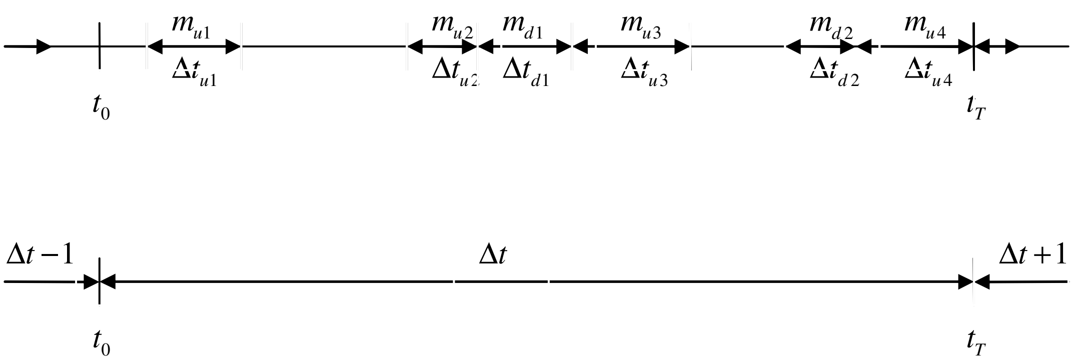

Figure 2. Car serving up and down calls

Where:

m_{ui} Car load moving upwards during trip, *i*

m_{d1} Car load moving downwards during trip, *j*.

Δt_{ui} Time period for trip *i* (upwards direction).

Δt_{dj} Time period for trip *j* (downwards direction).

Δt Time interval of analysis.

t_0 Initial instant for time interval.

t_T Final instant for time interval.

Values for m_u and m_d are calculated as (2):

$$\begin{aligned} m_u\big|_{\Delta t} &= m_{u_1}\cdot\Delta t_{u_1} + m_{u_2}\cdot\Delta t_{u_2} + m_{u_3}\cdot\Delta t_{u_3} + m_{u_4}\cdot\Delta t_{u_4} \\ m_d\big|_{\Delta t} &= m_{d_1}\cdot\Delta t_{d_1} + m_{d_2}\cdot\Delta t_{d_2} \end{aligned} \qquad (2)$$

It has to be taken into account that the total weight must include the sum of the car load for all the cars, in the case of more than one.

The variation of weight being transported is calculated as the percentage variation between two consecutive periods (3).

$$
\begin{aligned}
m_u^{'}\Big|_{\Delta t} &= \frac{m_u\big|_{\Delta t} - m_u\big|_{\Delta t-1}}{m_u\big|_{\Delta t-1}} \\
m_d^{'}\Big|_{\Delta t} &= \frac{m_d\big|_{\Delta t} - m_d\big|_{\Delta t-1}}{m_d\big|_{\Delta t-1}}
\end{aligned}
\qquad (3)
$$

3.2. Fuzzification of the variables

The car load moving up and down is caught by the sensors and is then grouped in one of the following three groups:

m_s A small value of car load

m_m A medium value of car load

m_b A big value of car load

Similarly, the variation in car load between consecutive periods is classified according to:

m_p A positive variation in car load

m_z A stabilisation in car load

m_n A negative variation in car load

Triangular membership functions are constructed in order to state the ranges of variation for each fuzzy variable. The membership function identifies the participation of each input, associating a weight with each of the inputs processed, defining overlaps between inputs and determining an output response (Figure 3).

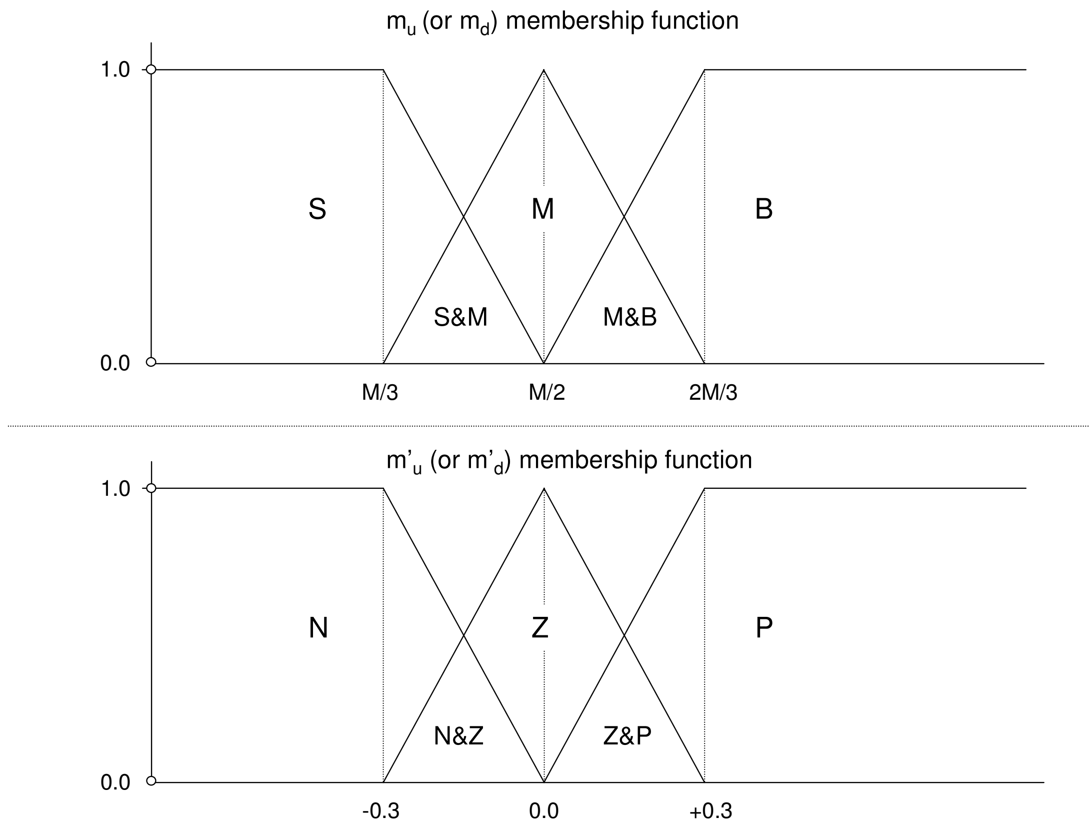

Figure 3. Input variables membership functions

The expected maximum capacity moving up (or down) during the time interval is calculated as:

$$M = 80\% \times (\text{car capacity}) \times \left(\frac{\Delta t - t_{stops}}{2} \right) \tag{4}$$

Formula (4) is calculated taking into consideration that rarely the total weight transported by the car exceeds an eighty percent of the maximum allowed amount of weight, and also it considers that a car that is not stopped spends half the time moving upwards and half the time moving downwards if we consider a time interval long enough.

3.4. Inference process

Fuzzy variables (input for the FL controller) are processed to get the corresponding fuzzy output. To do so we follow the rules set out in Table 1.

I. ***SI*** (*T_prev* = UPPEAK || *T_prev* = LUNCHPEAK):

- R1u: m_{us} ***&&*** m'_{un} ***THEN*** Output = NON_UP_FLOW
- R2u: m_{us} ***&&*** m'_{uz} ***THEN*** Output = UP_FLOW
- R3u: m_{us} ***&&*** m'_{up} ***THEN*** Output = UP_FLOW
- R4u: m_{um} ***&&*** m'_{un} ***THEN*** Output = UP_FLOW
- R5u: m_{um} ***&&*** m'_{uz} ***THEN*** Output = UP_FLOW
- R6u: m_{um} ***&&*** m'_{up} ***THEN*** Output = UP_FLOW
- R7u: m_{ub} ***&&*** m'_{un} ***THEN*** Output = UP_FLOW
- R8u: m_{ub} ***&&*** m'_{uz} ***THEN*** Output = UP_FLOW
- R9u: m_{ub} ***&&*** m'_{up} ***THEN*** Output = UP_FLOW

II. ***SI*** (*T_anterior* = DOWNPEAK || *T_anterior* = INTERFLOOR):

- R10u: m_{us} ***&&*** m'_{un} ***THEN*** Output = NON_UP_FLOW
- R11u: m_{us} ***&&*** m'_{uz} ***THEN*** Output = NON_UP_FLOW
- R12u: m_{us} ***&&*** m'_{up} ***THEN*** Output = NON_UP_FLOW
- R13u: m_{um} ***&&*** m'_{un} ***THEN*** Output = NON_UP_FLOW
- R14u: m_{um} ***&&*** m'_{uz} ***THEN*** Output = UP_FLOW
- R15u: m_{um} ***&&*** m'_{up} ***THEN*** Output = UP_FLOW
- R16u: m_{ub} ***&&*** m'_{un} ***THEN*** Output = UP_FLOW
- R17u: m_{ub} ***&&*** m'_{uz} ***THEN*** Output = UP_FLOW

- R18u: m_{ub} ***&&*** m'_{up} ***THEN*** Output = UP_FLOW

III. ***SI*** (*T_anterior* = DOWNPEAK || *T_anterior* = LUNCHPEAK):

- R1d: m_{us} ***&&*** m'_{un} ***THEN*** Output = NON_DOWN_FLOW
- R2d: m_{us} ***&&*** m'_{uz} ***THEN*** Output = DOWN_FLOW
- R3d: m_{us} ***&&*** m'_{up} ***THEN*** Output = DOWN_FLOW
- R4d: m_{um} ***&&*** m'_{un} ***THEN*** Output = DOWN_FLOW
- R5d: m_{um} ***&&*** m'_{uz} ***THEN*** Output = DOWN_FLOW
- R6d: m_{um} ***&&*** m'_{up} ***THEN*** Output = DOWN_FLOW
- R7d: m_{ub} ***&&*** m'_{un} ***THEN*** Output = DOWN_FLOW
- R8d: m_{ub} ***&&*** m'_{uz} ***THEN*** Output = DOWN_FLOW
- R9d: m_{ub} ***&&*** m'_{up} ***THEN*** Output = DOWN_FLOW

IV. ***SI*** (*T_anterior* = UPPEAK || *T_anterior* = INTERFLOOR):

- R10d: m_{us} ***&&*** m'_{un} ***THEN*** Output = NON_DOWN_FLOW
- R11d: m_{us} ***&&*** m'_{uz} ***THEN*** Output = NON_DOWN_FLOW
- R12d: m_{us} ***&&*** m'_{up} ***THEN*** Output = NON_DOWN_FLOW
- R13d: m_{um} ***&&*** m'_{un} ***THEN*** Output = NON_DOWN_FLOW
- R14d: m_{um} ***&&*** m'_{uz} ***THEN*** Output = DOWN_FLOW
- R15d: m_{um} ***&&*** m'_{up} ***THEN*** Output = DOWN_FLOW
- R16d: m_{ub} ***&&*** m'_{un} ***THEN*** Output = DOWN_FLOW
- R17d: m_{ub} ***&&*** m'_{uz} ***THEN*** Output = DOWN_FLOW
- R18d: m_{ub} ***&&*** m'_{up} ***THEN*** Output = DOWN_FLOW

The input degree of membership is calculated by means of different subsets operated by IF conditions together with the use of the logical sum OR (||) and the logical product AND (&&).

The fuzzy inference gathers the conditional rules around four fuzzy variables meaning how are the strength of the up and down flow, and the strength of the non-up and non-down-flow. The rules include not only the former variables but also the previous pattern detected so recent past is examined.

Once the rules have been processed, the firing strength of each rule is calculated. The logical products for each rule are then combined by means of the root-sum-square (RSS) method, which combines the effects of all applicable rules, scales the function at their respective magnitudes, and computes the fuzzy centroid of the composite area.

The RSS method was chosen to include all contributing rules, since there are few member functions associated with the inputs and outputs for the respective uppeak/downpeak at this stage. The non-uppeak/uppeak phenomenon is computed separately, as is the non-downpeak/downpeak phenomenon, according to the rules (5-6). Thus:

$$\begin{aligned}&\text{"Non\_Up\_Flow"} = \sqrt{\left(R1u^2 + R10u^2 + R11u^2 + R12u^2 + R13u^2\right)}\\ &\text{"Up\_Flow"} = \sqrt{\begin{pmatrix} R2u^2 + R3u^2 + R4u^2 + R5u^2 + R6u^2 + R7u^2 + R8u^2 \\ +R9u^2 + R14u^2 + R15u^2 + R16u^2 + R17u^2 + R18u^2 \end{pmatrix}}\end{aligned} \qquad (5)$$

$$\text{"Non\_Down\_Flow"} = \sqrt{\left(R1d^2 + R10d^2 + R11d^2 + R12d^2 + R13d^2\right)}$$

$$\text{"Down\_Flow"} = \sqrt{\begin{pmatrix} R2d^2 + R3d^2 + R4d^2 + R5d^2 + R6d^2 + R7d^2 + R8d^2 \\ +R9d^2 + R14d^2 + R15d^2 + R16d^2 + R17d^2 + R18d^2 \end{pmatrix}} \quad (6)$$

3.5. Defuzzification

The next step consists of the defuzzification of the data into a crisp output by means of the fuzzy-centroid algorithm, combining the results of the inference process and computing the fuzzy-centroid area.

To do so, the following output membership functions (for downpeak and uppeak) are considered. See Figure 4.

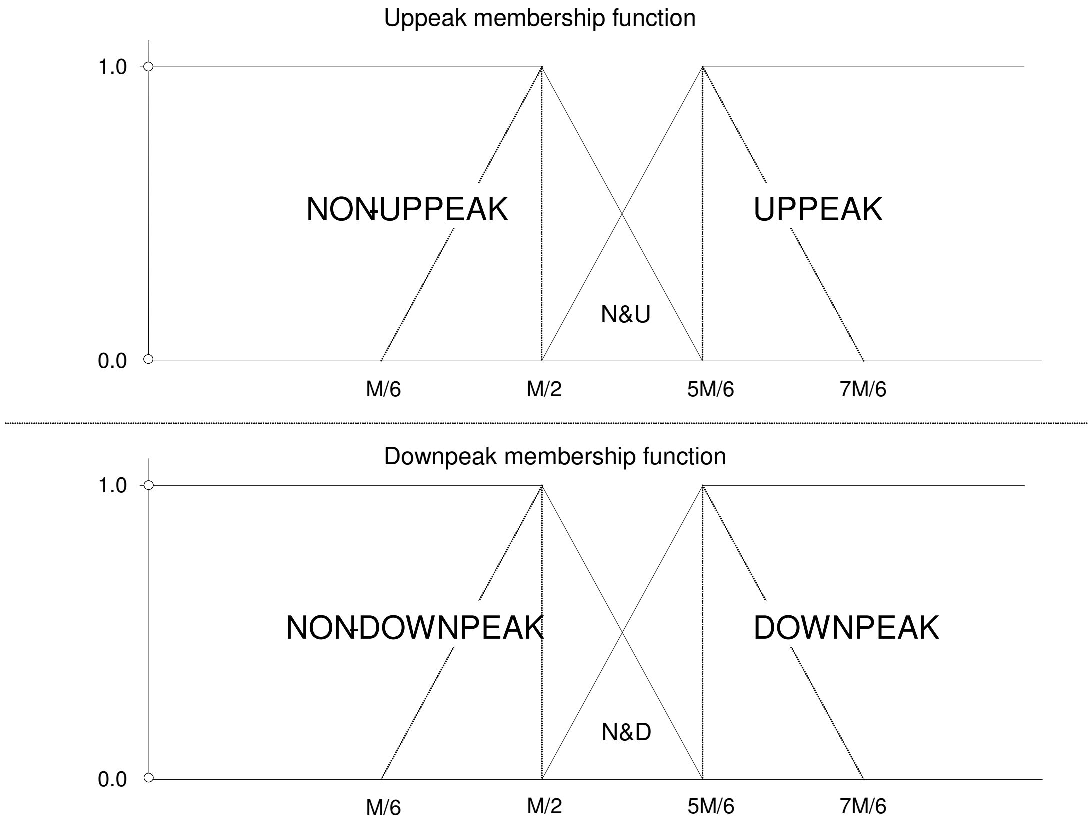

Figure 4. Output membership functions

(Habría que cambiar tb los nombres en estas gráficas…)

Where M/6 and 7M/6 represent the non-down-flow and non-up-flow, and up-flow and down-flow centroids respectively. Using the fuzzy-centroid algorithm, the up-flow and down-flow analyses are set out below with the calculation of the crisp values (7).

$$\begin{aligned}\text{crisp downpeak} &= \frac{[\text{No-down.center}]\times[\text{No-down.strength}]+[\text{Down.center}]\times[\text{Down.strength}]}{\text{No-down.strength}+\text{Down.strength}}\\ \text{crisp uppeak} &= \frac{[\text{No-up.center}]\times[\text{No-up.strength}]+[\text{Up.center}]\times[\text{Up.strength}]}{\text{No-up.strength}+\text{Up.strength}}\end{aligned} \tag{7}$$

Once the crisp numbers have been obtained, the final peak identification rules are processed using the logical product operator (Table 2).

Table 1. Inference rules

Rf1: IF Non-uppeak AND Non-downpeak THEN INTERFLOOR
Rf2: IF Uppeak AND Non-downpeak THEN UPPEAK
Rf3: IF Non-uppeak AND Downpeak THEN DOWNPEAK
Rf4: IF Uppeak AND Downpeak THEN LUNCHPEAK

Finally, in order to select the peak proposal, the maximum method is used, which selects the output at which the fuzzy subset has it maximum truth value (the higher probability).

4. Experimental results

In order to test the performance of the proposed controller, simulations using Elevate^{TM} software have been carried out. Elevate^{TM} is a vertical transport simulation software release from Peters Research Ltd. (a detailed description about Elevate^{TM} can be found in Caporale, 2000) and is mainly based on Dr. Peters' developments. Its main characteristics are introduced in Peters (1998).

The proposed building has been designed to cope with the demand in terms of handling capacity, as it is described in Barney (2003). The building has five floors with 41 people on each floor, so a single 600 kg capacity car ensures an acceptable transport capacity by means that 15% of the population can be served during the busiest five minutes of uppeak traffic.

The detection of the controller is analysed using different traffic patterns that have been especially created by expert researchers in vertical transport. These three different types of traffic are a well-known basis of analysis that is managed by the most relevant vertical traffic planning and simulation tools (Peters, 1998; Carporale, 2000; Cortés *et al*. 2006). The patterns employed for the simulations are the CIBSE Guide pattern (2004), the Strakosch pattern (Strakosch, 1998) and the Siikonen pattern (see Siikonen, 1993 and Siikonen, 2000). They all correspond to the daily traffic in an office building, from the start of the working day to the end, and consider all the possible types of traffic.

The following figures represent the daily hour and the vertical transport demand (the amount of upward and downward traffic as a percentage of the total building population). Different colours and tones represent the different traffic patterns forecasted by the controller. With regards to the computational time, the FL controller response time was always less than one millisecond.

The CIBSE Guide traffic pattern in Figure 5 depicts a typical day in an office building, as well as the FL controller detection.

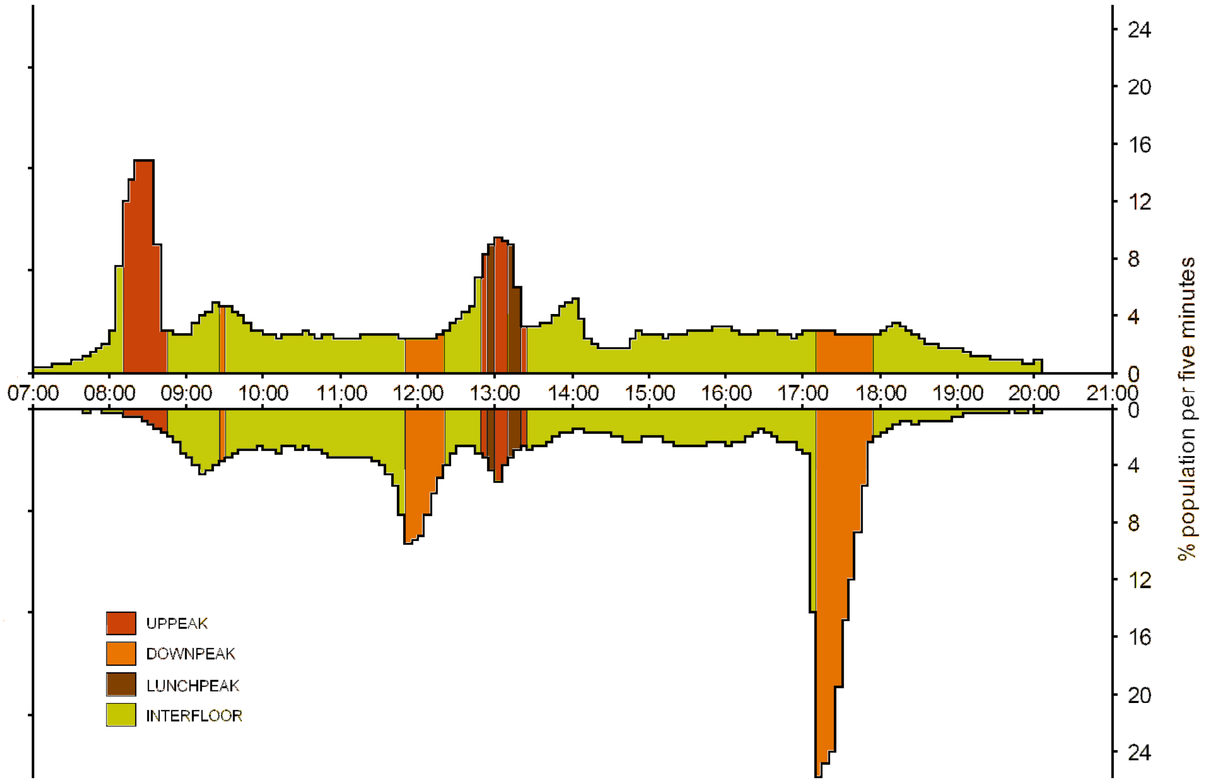

Figure 5: CIBSE Guide traffic pattern and FL controller output

The CIBSE Guide model describes the typical expected behaviour, showing a clear uppeak period at the start of the day and a downpeak period in the latter part of the day. Midday shows a typical lunchpeak with a moderate downpeak following a subsequent moderate uppeak.

The Strakosch pattern represents a more atypical pattern with smoother peaks, smaller maximums and minimums, and light uppeak and downpeak periods taking place both in the morning and in the afternoon. Figure 6 depicts the controller response when analysed by the Strakosch pattern.

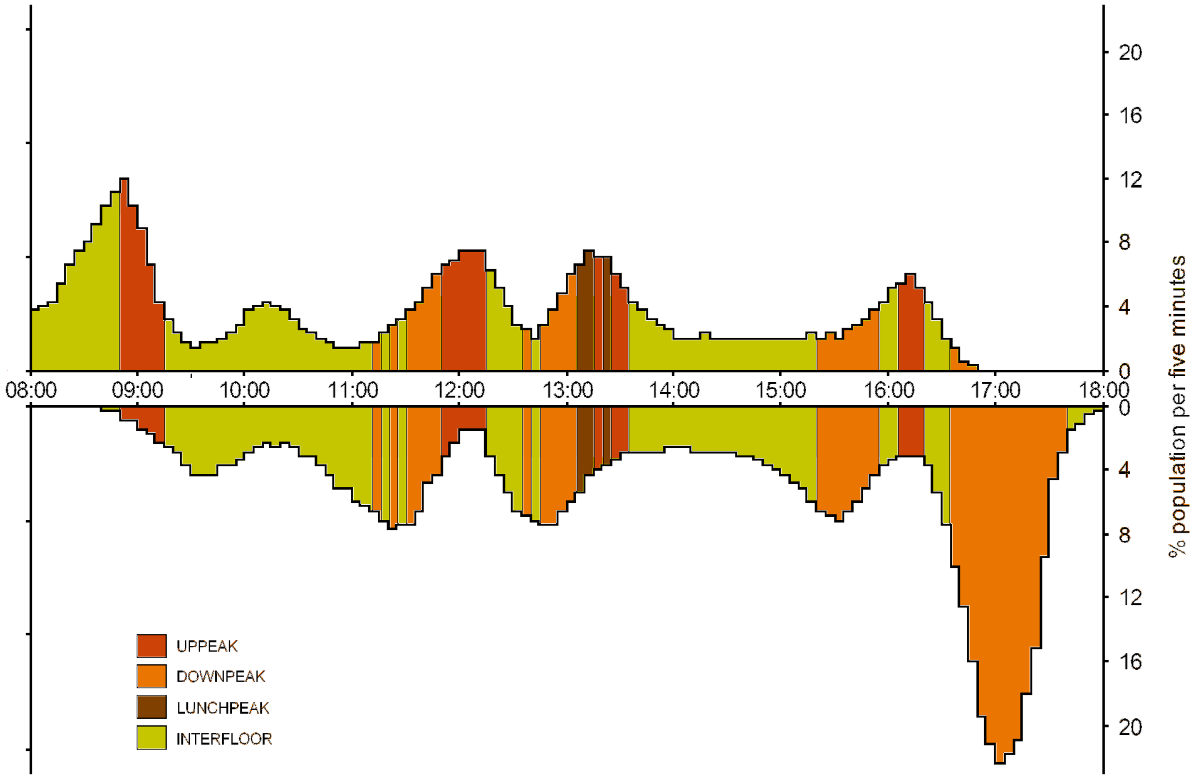

Figure 6: Strakosch traffic pattern and FL controller output

Finally, we present the Siikonen pattern. This traffic pattern has been created by Kone Corporation and is used by the company as a basis of analysis for traffic analysis in office buildings. It is claimed to be the most realistic traffic pattern in literature. Figure 7 shows the pattern and the results offered by the FL controller.

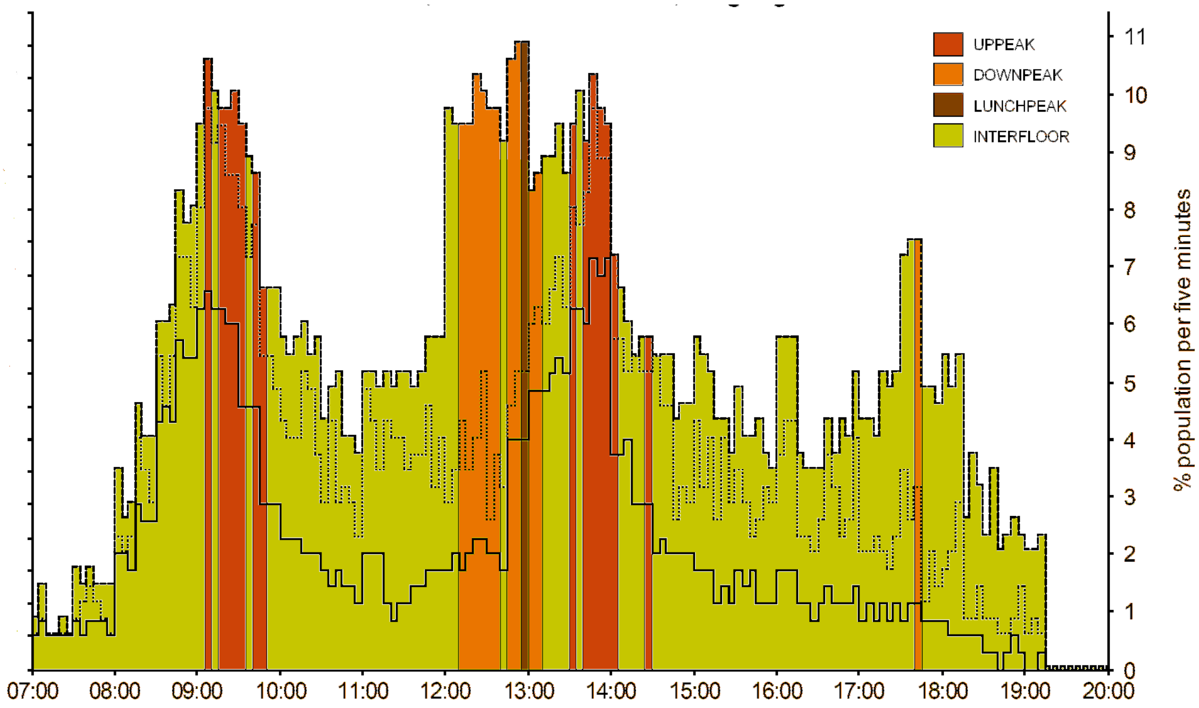

Figure 7: Siikonen traffic pattern and FL controller output

The three patterns displayed above show significant differences. This aspect provides an idea about the robustness of the controller and its reliability, independently following the pattern of the input source.

Occasionally, the controller shows a quick change in the identified peak (e.g. Strakosch pattern between 11:15 and 11:30). These rarely fast variations on the detected pattern occur because slight modifications in the real demand shake sometimes the working point between the theoretical frontiers of the predefined patterns. However, there is no need to provide the controller with a hysteresis mechanism: when the demand is moving between the bounds of two different patterns, it is not very important to classify the traffic pattern as one type or the other because in fact it is possible to state that both or any of the two traffic pattern are occurring.

.

Moreover, neither the selected car dispatcher nor the number of elevators in the system condition the simulation results as long as the system is correctly designed to attend passengers' demand employing less time than the length of the predefined time interval (Δt). Taking into account that the traffic controller is generally designed after the elevator system and regarding the mentioned aspect, the time interval (Δt) should be chosen at least greater than two times the *Round Trip Time* in order to make the prediction only once the passenger demand is accurately calculated. Different simulations were carried out concerning this aspect and it could be observed that for time intervals lower than this threshold sometimes the behaviour was the not desired.
Note: Round Trip Time is the average time each lift employs since it opens its doors for gathering the passengers at the main floor until it opens them again for gathering the new batch of passengers.

5. Conclusions

This paper presents a fuzzy logic based controller for traffic peak detection in elevator systems. The controller has been tested with the most authorised traffic patterns in the industry for office buildings, which are the CIBSE Guide, Strakosch, and Siikonen (Kone Corporation) traffic patterns. The FL controller showed a more than suitable performance providing adequate traffic pattern identification. The controller is robust and reliable when tested with patterns depicting sudden or smooth changes, and with high or low maximum/minimum peaks.

Changes are never detectable before the end of the period as the controller only puts into exam the traffic happened once the time interval is over. Theoretically the average time employed in the detection of a change in the traffic pattern is $\Delta t/2$ minutes. E.g. in a typical 5 minutes interval the average control reaction takes place every 2.5 minutes once the traffic has changed. This figure gives an approximate idea of how fast the controller detects a change in the traffic pattern. Besides, in this sense it also exists a compromise between a fast detection and a reliable detection: Time interval could be reduce so average detection is prompter but never less than twice the round trip interval as working under this threshold could lead to undetermined and unreliable performance.

Fuzzy logic based controllers usually have a problem based on memory faults, affecting the performance of such controllers. To avoid such problems, fuzzy logic controllers are combined with neural networks, thus increasing the complexity of the controller implementation. In the case of dispatching lifts this factor is crucial. The computation availability does not allow complex implementations as neural networks could require it when combined with fuzzy logic. The use of feedback allows avoiding such an inconvenience: Through the parameter *T_prev*, which provides information about the traffic pattern detected in the previous period, the system is able to "remember" what was happening in the immediately previous period. This parameter significantly improves the controller's performance.

Our further research is making use of this research in the attempt to further the design of a new fuzzy logic controller that is integrated into a multiagent system, governing a sub-zoning based dispatching algorithm.

Acknowledgements

The authors would like to thank MACPUAR, S.A. (Spanish elevator company) for the support of this research line since 2000. Also, the authors acknowledge the financial support given by the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía) (project ref. P07-TEP-02832), Spain.

References

Barney, G., 2003. Elevator Traffic Handbook: Theory and practice, Taylor & Francis Group.

Barney, G.C., dos Santos S.M., 1985. Elevator traffic analysis design and control second ed. London Peter Peregrinus.

Barney, G. and Peters, R. 2005. Traffic planning and selection of lift equipment and performance, in: Transportation System in Buildings CIBSE Guide D The chartered Institution of Building Services Engineers London / Cibse Pub. London.

Caporale, R. S., 2000. Elevate traffic analysis software (eliminating the guesswork). Elevator World, June, 118-21.

Cortés, P., Larrañeta, J., Onieva, L., 2004. Genetic Algorithm for Controllers in Elevator Groups: Analysis and simulation during lunchpeak traffic. Applied Soft Computing. 4, 159-174.

Cortés, P., Muñuzuri, J., Onieva, L., 2006. Design and analysis of a tool for planning and simulating dynamic vertical transport. Simulation: Transactions of the Society for Modelling and Simulation International. 82 (4), 255-274.

Elevator World's Guide to Elevatoring 1992, Mobile AL: Elevator World.

Luo, F., Xu, Y-G., Cao, J-Z., 2005. Elevator traffic flow prediction with least squares support vector machines, in: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 18-21.

Peters, R.D., 1998. Simulation for control system design and traffic analysis, in: Barney, G.C. (Ed.), Elevator Technology 9Proceedings of ELEVCON '98, IAEE Publications, Kendal, England, pp. 226–235.

Siikonen, M.L., 1993. Elevator Traffic Simulation. Simulation: Transactions of the Society for Modelling and Simulation International. 61(4), 257-267.

Siikonen, M.L., 1997. Elevator group control with artificial intelligence, Helsinki University of Technology, Systems Analysis Laboratory, Research Reports A67.

Siikonen, M.L., 2000. On traffic planning methodology, in: Lustig, A. (Ed.), Elevator Technology 10, IAEE, International Congress on Vertical Transportation Technologies Elevcon Berlin.

Siikonen, M.L., Leppala, J., 1991. Elevator traffic pattern recognition, in: Proceedings of the 4th IFSA Congress, pp. 195–198.

Strakosch, G.R., 1998. The vertical transportation handbook 3rd Edition. John Wiley & Sons, Inc.

Thangavelu, S. and Kandasamy, G., 1993. “Artificial intelligence", based learning system predicting "peak-period" times for elevator dispatching. Otis Elevator Company, U.S. Patent No. 5 241 142, 1993.

The Chartered Institution of Building Services Engineers London, 2004. CIBSE Guide D: Transportation systems in buildings. Norfolk, UK: CIBSE Publications Department.

Tyni, T., 2005. Identification of incoming peak traffic for elevators. KONE Corporation patent WO/2005/000726 (application number PCT/FI2004/000232).