STRATEGIES FOR AN ELEVATOR DISPATCHER SYSTEM

***Jana Horejsi*[1], *Petr Horejsi*[1], *Muhammad Latif*[2]**
[1]Department of Industrial Engineering
University of West Bohemia, Czech Republic
[2]School of Engineering, Manchester Metropolitan University, UK

e-mail: tucnak@cbox.cz

To provide for an efficient elevator system, a variety of diverse and sometimes conflicting constraints have to be solved. This paper focuses on using discrete event simulation as a means to model and explore elevator dispatching strategies. Arena simulation software has been used as a test-bed for model building, simulation and some experimentation. Model building using Arena building blocks are described in detail followed by four different elevator call strategies. The mains steps in the methodology are described with reference to a single elevator servicing a four floor office building. The elevator call strategies are simulated and results compared. It has been found that the data set significantly skews the results and overshadows any efficiency gains that might be possible from the different dispatching strategies. The paper concludes with the need to carefully selecting the data set as the basis for simulation comparisons and outlines future work required.

Keywords
Elevator, Elevator Dispatcher System, Discrete Event Simulation, Simulation, Arena

1. Introduction

Major cities in the UK have experienced a national programme of redevelopments often resulting in tall buildings with small footprints [Yuan 2008]. Existing buildings in many cases have been affected by urban generation programmes often with a change of use from warehousing to commercial/residential. All these buildings new or old require an efficient floor transportation system which traditionally is an elevator. Buildings often experience elevators congestion as a result of their heavy traffic, complex user types, and relatively slow-moving elevators (due to safety concerns) [Sydney 2003, Nagatani 2003]. Yet waiting for an elevator can be one of the main annoyances in one's experience with tall buildings [Berbeglia 2010, Sutton 1998].

How long we wait depends on the dispatching strategy the elevators use to decide where to go. Not surprisingly, the times of greatest traffic and greatest challenge to the dispatching algorithm are the morning and evening rush hours [Lee 2009]. Dispatchers are generally designed primarily for these difficult periods. Despite good designs, practically dispatchers have not achieved the efficiency levels that society expects often resulting with the most common complaint "waiting time was for too high" [Tebbenhof 2000].

Research into elevator system is quite recent and has followed the development of technology. The late eighties and the nineties can be considered as the start point, especially in USA and Japan [Robert 1988, Thangavelu 1989]. The focus of research during the last two decades has been on controls, mechanisms, safety, etc whilst using simple dispatching algorithms. However in more recent times some researchers are focusing on utilising artificial intelligence in elevators [Siikonen 1997, Tanaka 2005].

Although some researchers have explored the use of simulation in modelling an elevator [Cortes 2004], they have not directly addressed the effectiveness of dispatching algorithms within in-service elevators. Without an appropriate computer model it is difficult to develop and test the performance of an elevator dispatcher algorithm.

2. Aims of Research

The aim of the research was to model and simulate an Elevator Dispatcher System (EDS) to monitor the performance of different dispatching algorithms and explore strategies to reduce average *waiting time, average system time.*

The simulation tool selected for this work was Rockwell Automation's Arena package, which is part of the new generation of visually interactive simulator software. Arena is an easy-to-use, powerful tool that allows user to create and run experiments on models of particular systems. By testing out ideas in this computer virtually, you can predict the future with confidence and without disrupting your current real (e.g. business) environment. To build models with Arena user uses modelling shapes (called modules). The model consists of those modules and connections among them. There is also implemented the possibility Visual Basic of Application scripts development.

3. Case Study: Office Building, Manchester, UK

An office building was selected for this case study comprising of four floors serviced by a single elevator. The floors have the notation: Ground (G), Floor 1 (F1), Floor 2 (F2), Floor 3 (F3).

The elevator travelling time per floor was 15 seconds comprising of a 5 seconds for opening, 5 seconds for closing of the carriage doors and 5 seconds for lift elevation.

Elevator traffic data was obtained through rigorous observations for a full working day. Inter-floor movement is not considered for this particular case study. The elevator traffic data was converted into a series of schedules with hourly-based durations to represent each travel direction.

4. The Baseline Elevator Call Strategy (Scenario 1)

There were a variety of logical issues that needed to be considered to enable the elevator movement. Some of the important ones were how the elevator will:

- Respond to floor calls?
- Take people within its capacity?
- Take people to their desired floors?
- Enable FIFO loading logic?
- Remain on the floor if people are getting in or out?

A solution that attempts to answer the above issues was devised and developed by establishing three different control modes. In each mode the elevator is controlled in a different way as follows:

- Mode 0 (QUIET MODE) – Elevator is parked (stationary) and is waiting for a call;
- Mode 1 (CALL MODE) – Elevator is responding to a call whilst empty;
- Mode 2 (OPERATING MODE) – Elevator is in motion and occupied.

In the baseline strategy the elevator moves in cases when a call is made either empty (Mode 1) or occupied (Mode 2). In all other cases, the elevator simply parks and waits at the last drop off floor (Mode 0). If the elevator is in Mode 2 and people are queued at the call point (in the correct direction of travel) then the elevator is allowed to pick up people, within its capacity (max 8 people). If the elevator is in Mode 1 and some people wait at a floor call point in direction of move- ment then priority exists for those people who have called the elevator first. Next people who wait at other call points will be picked up in direction of movement (max 8 people). People are dropped off in the sequence depending on who arrived first i.e. person who made the first call, this person will be dropped off first (FIFO) then followed be the second person, etc.

Figure 1 illustrates the Arena modelling screen displaying the control logic and animation of the baseline model as described above. This

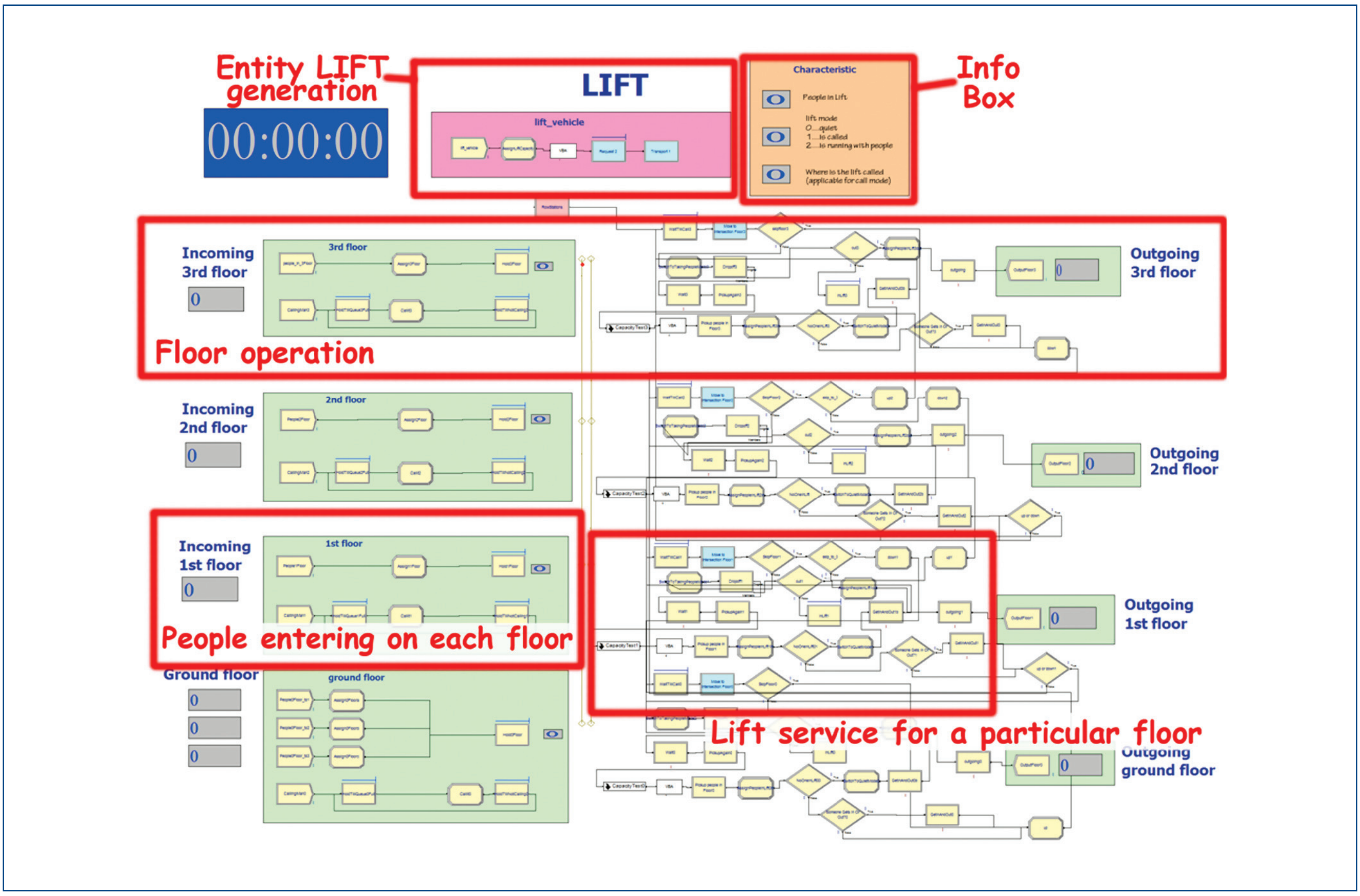

Figure 1. Baseline Arena Model

figure shows the ARENA model on the background and the red boxes illustrate the basic logical model structure (do not concern the unreadable labels in the background).

The developed model consists of the following discrete stages:

- People entering the model;
- Making an elevator call;
- Elevator control logic;
- Elevator movement between floors.

4.1. People Entering the Model

Figure 2 illustrates the control logic enabling entities (people) to enter the model and initiate an elevator call at a specific floor. Whilst the Arena modules shown are designated for the third floor the control logic is generic and applicable to all floors. The logic is captured in Arena using three modelling elements as follows:

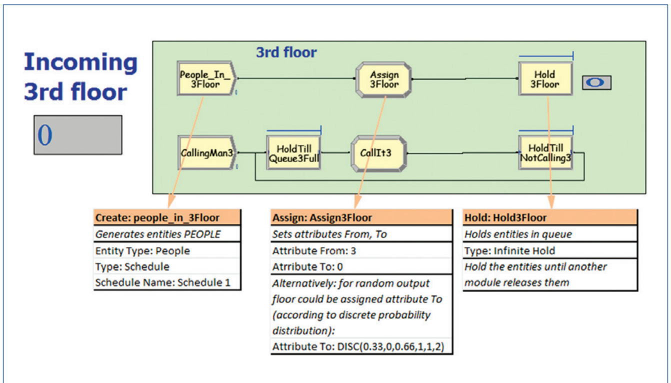

Figure 2. Control logic enabling people to enter the model

The Create block is used to generate entities in the system (entities are People) who arrive and intend to use the elevator. The arrival profile is based on schedule derived for each travelling direction according to case study specification.

The Assign block is used to set global variable and entity attributes to particular values. In this case attribute From with a value of 3 means that people arrive in to the system on the Third floor. Attribute To with a value of 0 means that people have a destination of the ground floor.

The Hold block is used to hold the people in a queue (i.e. Hold3floor.Queue) and wait for a specified condition to become true before the people can continue to move through the modelled system.

4.2 Making an Elevator call

Figure 3 illustrates the control logic for making an elevator call. Whilst the Arena modules shown are designated for the third floor the control logic is generic and applicable to all floors.

This part of the control logic is used for calling the elevator if the system is in mode 0 (i.e. stationary) and if people are in a queue waiting for the elevator at the respective floor. A global variable called Where called is used to capture from which floor the call for the elevator is made, applicable only when the system is in mode 1 (call mode). A dummy control entity is created at each floor level and is generated just once at each floor level. The dummy entity is named as Calling Man and is used for calling the elevator. This dummy entity is grounded (in Hold till queue 3 full block) until there are some people in the queue at given floor and the elevator is in the quiet mode (mode 0). After calling the elevator, the variable mode changes its value from mode 0 to mode 1 and runs to the floor where the first call was made. The entity Calling Man is waiting in HoldTillNotCalling block now. When the people get in to the lift (the

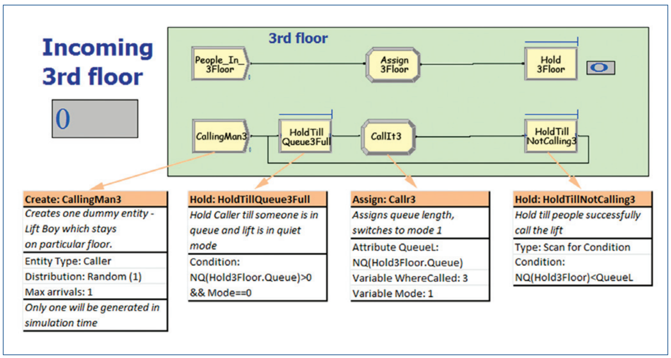

Figure 3. *Control logic for making an elevator call*

waiting queue changes), the entity Calling Man is moved back to hold block (Hold till queue 3 full) and is free for new requests.

4.3 Elevator Control Logic

The elevator control logic is shown in Figure 4. The control logic starts with the creation of a single but important entity to represent the elevator carriage labelled as *lift vehicle*, it is the main control entity and it is generated just once. The lift vehicle entity has the following attributes:

- Floor – the floor, where the elevator is now;
- LiftCapacity – the elevator capacity (currently set to 8 people);
- PeopleToTake – affects, how much people could be taken according to given floor queue length and the lift capacity;
- Down – has a value of 1 if the elevator is moving down and a value of 0 when the elevator is moving up;
- oldPeopleInLift – assistant attribute (for the control is the count of number of people in the elevator after drop-offs);
- nextfloor – where to go next (applicable for mode 2).

There was a need for all entities to access some important global variables i.e. Mode, WhereCalled which were established. This has the effect that when the simulation is running a dynamic counter will display the current mode, how many people are in the elevator and where the elevator was first called.

After the proper initialization is completed the elevator is moved to the respective floor i.e. third floor by utilizing the Request and Transport modules.

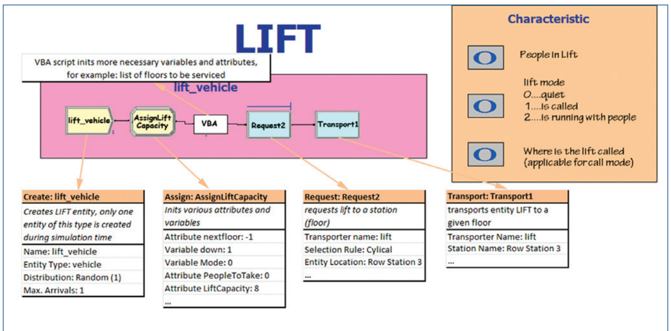

Figure 4. *Elevator control logic*

4.4 Elevator Movement between Floors

Figure 1 depicts a view of the control logic governing the elevator movement between floors. Module Stations were used to model the floors and enable movement of the elevator between floors. Details of the control logic for elevator movement between floors are not discussed here and will be the subject of another publication. It is sufficient to assume the model has been validated and verified according to the baseline strategy described earlier.

5. Simulation Results for Scenario 1

Results obtained after the model was simulated for 9 hours (i.e. 0900-1700 hrs) are shown in tables 2 and 3. The time units are seconds. Waiting time at call points are depicted in table 1. Essentially how long the people wait at the floor call points is important and considered to be a good measure of service levels and system efficiency. Clearly the smaller wait time the better the experience. Table 1 indicates that the maximum waiting time (62 seconds) was experienced at the ground floor and on average it was 9,4 seconds.

	Waiting Time (s)		
	Average	Minimum	Maximum
Ground floor	9,4	0,1	61,9
Floor 1	5,4	3,1	11,7
Floor 2	8,1	0,1	43,8
Floor 3	10,3	0,1	38,5

Table 1. *Scenario 1 Elevator call waiting time (units in seconds)*

Another useful measure to consider is how long the elevator was parked and not on call during the simulation period. This information is obtained by capturing how long the elevator waits in blocks wait till call i.e. essentially lift free time.

From table 2, it is evident that the elevator is parked for significant amounts of time in the upper floors.

	Waiting Time (s)		
	Average	Minimum	Maximum
Ground floor	128,2	0,2	517,8
Floor 1	237,7	1,9	1132,1
Floor 2	198,4	3,8	1075,7
Floor 3	132,9	5,8	618,1

Table 2. *Scenario 1 Elevator stationary periods (units in seconds)*

Recognising opportunities for improvements it was decided to explore alternative scenarios using the same data set (schedules).

6. Other Scenarios

Three further scenarios were devised. The actual modifications to the model control logic are not detailed in this paper. All other conditions were maintained as the baseline model (scenario 1) and the model simulated for the same duration i.e. 9 hours.

Scenario 2

In this scenario the control logic governing movement of the elevator from the baseline model was modified such that the elevator never parks (does not use mode 0). This scenario is paternoster based, but only with one carriage. The lift stops in floors where the call was made or someone wants to drop off. It is running up and down all the time.

Scenario 3

In this scenario the control logic governing movement of the elevator from the baseline model was modified such that the elevator will if not on call always park on the third floor. Secondly, the elevator when travelling empty (mode 1) and on call, is now able to stop and pickup people on interim floors, in the direction of movement, whilst travelling to its primary call point.

Scenario 4

This scenario is identical to scenario 3 except the elevator will, if not on call, always park on the ground floor.

Scenarios 2 to 4 were developed and the resulting models were simulated producing the tabulated results of call waiting times shown in table 3.

It is evident that the data set used to drive the simulation model will strongly reflect the behaviour of the model and subsequent results produced. However recognising the short simulation runs and limited iterations it is probably inappropriate to form trends without good confidence levels. Further work will address these shortcomings.

Realising the above shortcomings some observations can still be observed. Table 3 statistics show that average waiting times are smaller in scenario 1 in comparison to the other scenarios. If average waiting times is considered an important measure of performance then scenario 1 has produced the best results.

		Holding Floor Queue			
		Ground	**Floor 1**	**Floor 2**	**Floor 3**
Avg Value	**Scenario 1**	9.1	5,4	8,1	10,3
	Scenario 2	13,5	6,3	7	14,1
	Scenario 3	13,4	7,9	9,7	10,7
	Scenario 4	9,4	6,5	10,2	14,1
Min Value	**Scenario 1**	0,1	3,1	0,1	0,1
	Scenario 2	0,3	0,1	0,1	0,01
	Scenario 3	1,6	2,5	1,3	0,1
	Scenario 4	0,1	3,1	1,8	1,1
Max Value	**Scenario 1**	61,9	11,7	43,8	38,5
	Scenario 2	47,7	19,7	36,3	46
	Scenario 3	50,4	10,9	43,8	43,6
	Scenario 4	63,6	19,9	37,7	44,6

Table 3. *Comparison of Scenario Results (units in seconds)*

Minimum values in table 3 would essentially be disregarded as they only represent cold starting conditions which in reality are not representative of steady state conditions.

The maximum waiting times is often most important as this will lead to worst case conditions that are essentially what simulation is all about. From Table 3 scenario 3 appears to have the lowest range of maximum waiting times. This is probably explained by the fact the data is skewed towards more people making elevator calls on floor 3 which is where the elevator is parked.

Overall on balance table 3 would indicate that scenario 1 is more cost effective in reality this could also possibly lead to minimum operating costs.

7. Conclusions

The use of a visually interactive simulator has been shown to effectively and dynamically simulate the behaviour of a single elevator system. The technique of representing people as a series of entities enables the use of high quality animation in the simulation which improves the display at the human/computer interface.

A simple elevator control strategy has been developed and simulated using Arena software with reasonable success. Various variants of the simple baseline strategy were devised, developed and simulated producing interesting results. However it became evident the data set obtained that drives the model effects the simulation results in a very serious way.

The EDS become the test-bed for experimentation and enabled the dispatching scenarios to be dynamically evaluated and compared. During this work it became clear that using some lift call strategies is very individual, user and purpose depended. The dispatcher algorithm for the elevator decision making would need to be chosen on the nature and kind of building the elevator is being used for.

In this work we were unable to include artificial intelligence techniques for the decision making within the scenarios implemented. The decision making module for example could not consider the distances between (or among) the floors requiring calls. The elevator also could not skip from Call mode to Operation mode which would have been useful.

Clearly a good developed simulation model for a particular building in the construction design phase can save many unexpected expenses (especially more when we can predict and alter volume traffic.

The developed simulation model is designed strictly for the specific building with one ground floor and three floors above. So the developed model has one disadvantage: it is not parametrically built. The number of floors could not be variable. One of the possible solutions of this problem is to develop a universal parameterized floor in a submodel or more properly to use the ARENA template and design a user-defined block. It has been realised that the data set obtained has to some extent skewed the results and overshadowed any efficiency gains that might be possible from the different dispatching strategies.

The performance of any elevator dispatching system operating in high volume buildings will be increasingly important to building management as visitors come to expect higher service levels in the facilities being provided by modern elevators, having little regard for the complexity of the tasks involved in getting the decision making to be optimal.

This work acts as the starting case study for a more general work in this area. The first effort is to make the count of floors (and possibly count of elevators in the building) as the input parameter. There is also a need for adding the economical parameters and evaluation to the model. Next step is to include the optimization using conventional OptQuest ARENA tool or user developed parallel simulation tool PAARE. The result will be the complex system which will tell the user (based on quality of given data) which strategy is the best one in balance of costs and waiting times. There has to be included not only the elevator type costs, but also the energy and maintenance costs to the economic evaluation.

8. Further Work

The results and overall experience of this preliminary study has provided the necessary stimulus to continue the work to include high traffic volume, consider other scenarios i.e. minimum travelling distances, and to extend the experimentation to achieve 95% confidence levels.

Other work will involve optimisation for computational efficiency, incorporate intelligent decision making techniques, increase the problem domain to deal with multiple and banked elevators, and quantify the benefits gained from these methods.

Furthermore some optimization could be performed. The criteria function could consider average waiting time and time of the person within the system. For economical evaluation we should also consider operation costs. The criteria function can balance both of these factors (maximize the transport quality and minimize the costs).

Acknowledgement

This article was prepared with support of the Internal Science Foundation of the University of West Bohemia SGS-2010-065.

References

[Berbeglia 2010] Berbeglia, G., Cordeau, JF. and Laporte, G.: Dynamic pickup and delivery problems, *European Journal of Operational Research,* April 2010, Volume 202, Issue 1, pp. 8–15, ISSN 0377-2217

[Cortes 2004] Cortes, P., Larraneta, J. and Onieva, L.: Genetic algorithm for controllers in elevator groups: analysis and simulation during lunch peak traffic. *Applied Soft Computing,* May 2004, Volume 4, Issue 2, pp. 159-174, ISSN 1568-4946

[Lee 2009] Lee, Y., Kim, T.K et al.: Performance analysis of an elevator system during up-peak. *Mathematical and Computer Modelling,* February 2009, Volume 49, Issues 3-4, pp. 423-431, ISSN 0895-7177

[Nagatani 2003] Nagatani, T: Complex behaviour of elevators in peak traffic.: *Physica A: Statistical Mechanics and its Applications,* August 2003, Issues 3-4, pp. 556-566, ISSN 0378-4371
[Robert 1988] Robert M.C., and Abrego E.: *Coincident call optimization in an elevator dispatching system,* 1988, Westinghouse Electric Corp. U.S. Patent No. 4 782 921.
[Siikonen 1997] Siikonen, M.L.: *Elevator group control with artificial intelligence,* Research report: A67. Helsinki: University of Technology, Systems Analysis Laboratory, 1997
[Sutton 1998] Sutton, R.S. and Barto, A.G.: *Reinforcement Learning: An Introduction: Elevator Dispatching.* Massachusetts: The MIT Press, 1998, ISBN 0-262-19398-1
[Sydney 2003] Sydney, C.K., Chu, C.K.Y., Lin, C.K.Y. and Lam, S.S.: Hospital lift system simulator: A performance evaluator–predictor. *European Journal of Operational Research,* April 2003, Volume 146, Issue 1, pp. 156–180, ISSN 0377-2217
[Tanaka 2005] Tanaka, S., Uraguchi, Y., and Araki, M.: Dynamic optimization of the operation of single-car elevator systems with destination hall call registration. *European Journal of Operational Research*, December 2005, Volume 167, Issue 2, pp. 550-573, ISSN 0377-2217
[Tebbenhof 2000] Tebbenhof, A., and Dekker, R.: Econometrics in the elevator, *International Journal: The Medium for Econometric Applications,* 2000, Volume 9, Issue 3, pp. 26-30, ISSN 1389-9244
[Thangavelu 1989] Thangavelu and Kandasamy: *Queue based elevator dispatching system using peak period traffic prediction,* 1989, Otis Elevator Company. U.S. Patent No. 4 838 384.
[Yuan 2008] Yuan, X., Busoniu, L. and Babuska, R.: Reinforcement Learning for Elevator Control. *In*: Chung, Myung Jin, ed. *Proceedings of the 17th International Federation of Automatic Control (IFAC) World Congress,* Seoul, Korea, 6-11 July, 2008. ISBN 978-3-902661-00-5

Contacts:

Dipl. Ing. Petr Horejsi, Ph.D.
Department of Machining Technology, University of West Bohemia, Univerzitní 8, 306 14 Pilsen, Czech Republic
tel.: +420 608 152 357, fax.: +420 377 638 402
e-mail: tucnak@cbox.cz

Dr. Muhammad Latif, BSc (Hons), PhD, MIMechE, CEng
Dept of Engineering and Technology, Faculty of Science and Engineering
Manchester Metropolitan University, Chester St
Manchester M1 5GD, United Kingdom
tel.: +44 (0) 161 247 6264, fax: +44 (0) 161 247 1633
e-mail: M.Latif@mmu.ac.uk