March-April 2020
ISSN: 0193-4120 Page No. 5456 - 5461

Efficient Algorithm for People Management in an Elevator

Jayanta Biswas[1], Joseph Varghese Kureethara[2], Debabrata Samanta[3], Sandhya M[4]
[1,3,4]Department of Computer Science, Christ (Deemed to be University), Bangalore, India
[2]Department of Mathematics, Christ (Deemed to be University), Bangalore, India

Article Info
Volume 83
Page Number: 5456 - 5461
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 28 March 2020*

Abstract:
High-rise buildings from ancient times to modern times were built on human aspirations to reach heights. Most of the multi-storied buildings are supported by vertical elevators. The need for efficient and intelligent vertical-elevator management systems requires efficient algorithms. This paper attempts to solve the problem of identifying the next floor to go while servicing the current floor while maintaining input queues corresponding to each floor. The paper adopts the concept of the highest response ratio (HRR) of the next algorithm from the operating systems to solve the problem. This proposal estimates the number of persons to be serviced for each floor based on the number of persons last served for the corresponding floor. The given proposal solves starvation for queue with HRR and is implemented with simple hardware logic.

Keywords*: Destination floor, HRRN, PID controller, Acceleration, Waiting time, Lift.*

INTRODUCTION

Human beings have always aspired to reach heights. At first, it was the branches of the trees. They constructed homes on the top of trees to save themselves from wild animals. Then they looked at the top of the mountains. Mountaineering is one of the oldest physical exercises of human beings. Humans set foot on every peak available in the world. Constructing high rise buildings was considered to be a great accomplishment of any ruler. From the pyramids to the Colosseum, from temples to churches, humans tried constructing imposing structures all around the world.

Climbing up is not an easy task. When the imposing structures were created all around the world by different generations and cultures, one of the pertinent questions was to lift the people up to the top of the mammoth building. Pyramids, temples, and churches were not for daily living and business activities. Hence, it was alright to permit only skilled people to go up to the top of such buildings. However, in the last part of the 19th century, because of the boom in construction activities with the industrialization in the US, high-rise buildings took birth to supply the demand of the office spaces required for the businesses.

LITERATURE REVIEW

2.1 The German inventor, Wener von Siemens made the first electric elevator way back in 1880. In the initial decades of the vertical elevators, they were powered by either electricity or by hydraulics. By the Second World War, many elevators were driven by electronic systems. Vertical elevators based on electronics systems employed powerful and efficient algorithms. There are many considerations in the making of an efficient vertical elevator serving several floors of a high rise building.

Some of them are given below:

- Speed
- The capacity of the car
- Delay at each floor
- Idle time

March-April 2020
ISSN: 0193-4120 Page No. 5456 - 5461

•Automated door-opening and door-closing while ensuring safe entries for persons entering the lift.
•Secure signals for opening and closing of doors
•Keys for various floors
•Intelligent stop and run
•Efficient stop and run
•Movement of people in a car
•Efficient queue in a car
•Efficient entry and exit of the passengers

2.2 Schroder has patented a method**[8]** that matches the demand and supply at the main floor for an elevator in a multi-storied building. The method described by the inventor is a very powerful mechanism to ease the traffic at busy times of a business day. There are effects of pressure changes among the passengers while traveling by a vertical elevator. Some of the issues related to this problem which can lead to the slowing down the elevator is solved by Rory and Peters [9].
2.3 Rush hour blues of the elevators were addressed by Powel [7]. The number of people arriving and leaving the car is still a complicated problem. Ovaska has approached this issue and tried to provide a sensor based solution using Doppler radars [6] and mechatronics [5].

Cirtes and Barto [1] tried presenting high performing elevator models with the help of a reinforcement learning model. Multi-floor management is also influenced by the layout of the floors [12]. Optimizing the number and location of the elevators must be a matter of concern for every layout designer. Most of the modern elevators are unmanned. Much research is happening in this area in an extensive manner considering the speed, security, and efficiency in transportation. Many alert and signal systems are put in practice. Many are under testing (Han et al., [3]: Teixeira et al., [10], Dash, [2]; Wang, [11]).

PROBLEM STATEMENT

The proposal tries to solve the next floor identification problem by taking the motivation of the operating system concept. In the case of an operating system, multiple jobs with different priorities are submitted to the processor, jobs are placed in the queue with the corresponding priority while maintaining the order. The processor remembers the service time for each of the queues and predicts the service time of the job while picking up the highest dynamic priority. The dynamic priority is calculated based on the predicted service time for the corresponding queue and waiting time for the first item in the queue.

Similarly, the proposal remembers the number of people who entered the lift from a specific floor while that specific floor was being served. When the lifts stop at a particular floor, people enter the lift till the lift is full. When a person enters the line time stamp is attached to the corresponding person and dynamic priority for the flor is calculated based on the timestamp of the first person in the queue and the predicted number of people entering the lift from the specific floor.

This algorithm is efficient because it ensures the minimum response time and the algorithm guarantees that starvation does not occur. Hence, people wait at the queue for the least amount of time and the implementation uses simple counter based logic.

PROPOSED METHODOLOGY

The algorithm is given below to calculate the destination floor. The current floor is initiated to zero. The condition is checked whether the internal key press list is zero. The process of reversing the direction value is applied, if the key press list is empty. The destination floor value is computed using the HRRN algorithm as per given in the Fig. 3.2. The destination floor value is sent to the PID controller, eventually the lift reaches to the destination floor. If the internal key press list is not

March-April 2020
ISSN: 0193-4120 Page No. 5456 - 5461

empty, then the nearest floor value from the head of the list is initialized to the destination floor value. The destination floor command is sent to PID controller and PID controller is activated.

CALCULATE DESTINATION FLOOR

The proposal calculates the next destination floor while serving the current floor in an efficient manner. The proposal ensures minimum waiting time for the person in the queue at any floor while avoiding starvation.
Hence, nobody is waiting indefinitely in a queue waiting for the lift.

Pseudocode for calculating destination floor

1. Process of Calculating Destination Floor
2. {
3. initialize the current destination floor to zero
4. initialize the current floor value to zero
5. if (internal keypress list is NULL)
6. {
8. reverse the direction value;
9. calculate the destination floor value using HRRN as depicted in Fig. 3.2
10. send the destination floor command to the PID controller to go to the destination floor
}
11. else
12. {
13. set the nearest floor value from the head of the list
14. set the destination floor value to nearest floor
15. send the command to PID controller to go to destination floor
16. }
17. }

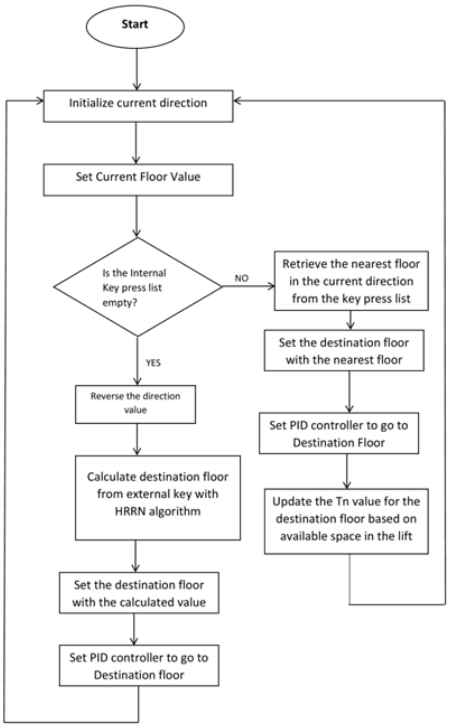

Figure. 3.1 Flowchart for calculating Destination Floor

3.2 CALCULATE HRRN(Highest Response Ratio Next)

HRRN algorithm finds the response ratio of all processes available and picks the one with the highest response ratio.
In the case of CPU scheduling, waiting time for each queue is calculated from the head of queue and processing time is predicted. Sn+1 denotes the predicted processing time for the (n+1)th job and it is calculated from T_1, T_2, T_3 to T_n values where $T_n$corresponds to the processor execution time for the $_n$th job in the queue.
Response Ratio for any process is calculated by using the formula[1]

HRRN = (W + S) /W[1]

Here, W denotes Waiting time for the queue and
S denotes Predicted Service Time for the corresponding queue.

March-April 2020
ISSN: 0193-4120 Page No. 5456 - 5461

$$S = (T_1 + T_2 + T_3 + \ldots\ldots + T_n)/n[2]$$

In this proposal, T_i indicates the number of persons entering the lift from the floor at the ith instant. Whenever the lift stops on a particular floor, people waiting at the floor enters the lift. The number of persons entered the lift is assigned to the T value at that instant for that floor.

Sn+1 can be expressed in the form of T_n and S_n. Hence, two multiplications are required to compute the predicted number of people waiting (Sn+1) for the corresponding floor. In the proposed algorithm HRRN is used to calculate the next destination floor. The algorithm calculates the waiting time of the timestamp of the head of the queue.

The assumption here is the total waiting time for any person is 0 to 511 seconds or less than 512 seconds as nine bit counter is used. When the lift stops at the destination floor, the empty capacity of the lift is filled from the waiting queue. Hence T_n is considered as the number of persons entering the lift when the lift stops at the destination floor.

Pseudocode for calculating HRRN

1) Process of Calculating HRRN (Highest Response Ratio Next)
2) {
3) initialize α(alpha) value to zero
4) initialize Tn value with respect to all floors
5) set Sn+1 value to (α, Tn)
6) set W to head of the queue
7) calculate W for each floor from the head of the queue time
8) calculate HRR for each floor
9) set HRR value to highest in the queue
10) set destination floor to the highest value
11) }

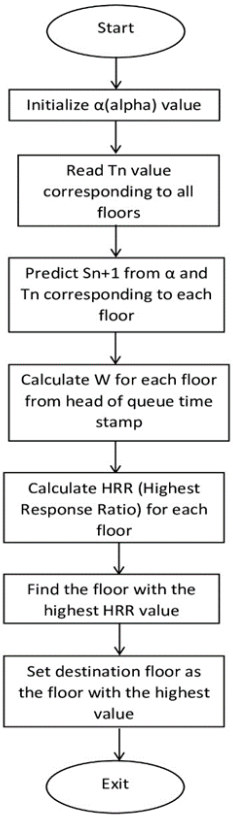

Fig. 3.2 Flowchart for Calculating HRRN

3.3 CALCULATE PID CONTROLLER TO REACH CONSTANT VELOCITY WITH INITIAL ACCELERATION

PI controller is used to reach constant velocity while accelerating and to reach sliding velocity while breaking. Acceleration is stopped after the lift reaches 80% of maximum velocity. The lift moves in constant velocity with the PI controller till it reaches within the threshold of minimum distance. A break is applied after the lift crosses the threshold of minimum distance and the PI controller is activated to reduce the speed to sliding velocity. When the acceleration is activated error value is considered to seven and similarly when the brake is applied error value is considered as minus four. The actual error value is calculated when acceleration is stopped.

Procedure to reach Constant Velocity with initial acceleration

March-April 2020
ISSN: 0193-4120 Page No. 5456 - 5461

1) initialize a, b and c value of PID controller value to zero
2) initialize direction value to zero
3) set previous error value to zero
4) calculate the remaining distance for the destination floor
5) if (remaining distance>threshold)
6) {
7) if (current velocity>0.8*maximum velocity)
8) {
9) set error to 7
10) calculate d[n+1] = d[n] + a * e[n] + b * e[n-1] + c * e[n-2]
11) set LED display to new value
12) }
13) else
14) set error to minus four value
15) set error value to current velocity and maximum velocity of the lift
16) calculate d[n+1] = d[n] + a * e[n] + b * e[n-1] + c * e[n-2]
17) }

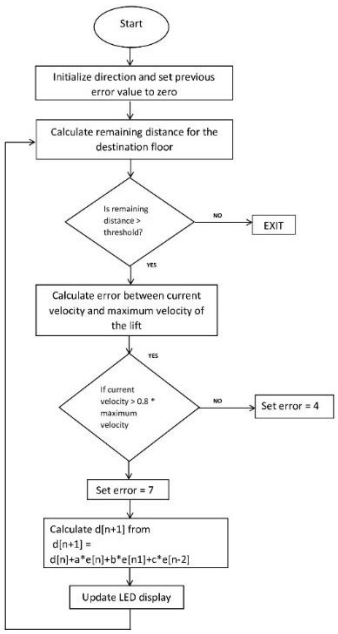

Fig. 3.3 Flowchart for calculating PID controller to reach constant velocity with initial acceleration

3.4 PID CONTROLLER TO REACH DESTINATION FLOOR WITH INITIAL BREAKING

Procedure to reach destination floor with initial breaking

1) initialize error value to minus four
2) set remaining distance value to destination floor
3) if (remaining distance > threshold)
4) {
5) d[n+1]=d[n]+a*e[n]+b*e[n1] + c*e[n-2]
6) if (d[n] is greater than two)
7) {
8) set d [n] = two
9) set error value based on remaining distance value
10) }
11) else
12) move to destination floor
13) set door value to OPEN
14) set LED display to the new value
15) }

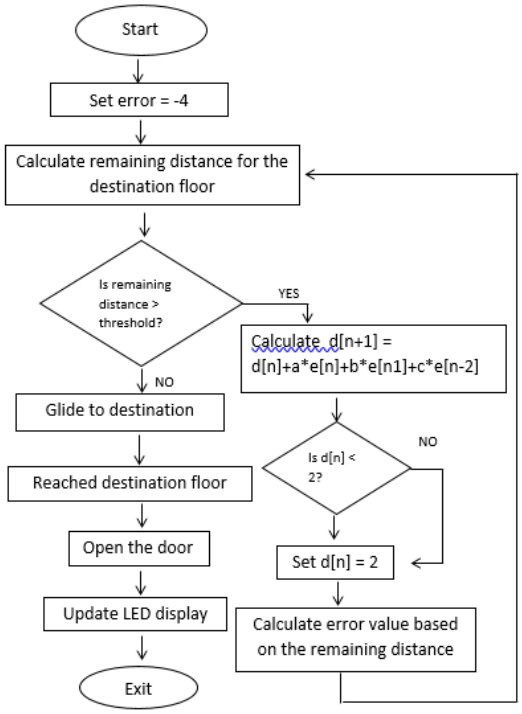

Fig. 3.4Flowchart for calculating PID controller to reach constant velocity with initial breaking

CONCLUSION

The proposal ensures minimum waiting time for a person in a queue while avoiding starvation for all the floors. Simple control logic is employed to detect the next destination floor while serving the current floor and PI control is used to reduce the time to reach the next destination floor. The proposal also ensures the lift is accelerating and decelerating in a smooth manner.

REFERENCES

1. A. G. Barto and R. H. Crites, “Improving Elevator Performance Using Reinforcement Learning,” Adv. Neural Inf. Process. Syst.pp. 1017-1023, (1996)
2. A. R. Dash, S. Kumar Sahu, and B. Kewal, “An Optimized Disk Scheduling Algorithm with Bad-Sector Management,” Int. J. Comput. Sci. Eng. Appl., (2019)
3. W. Han, F. Xu, and X. Ma, “Research on Alert Strategy of Unmanned surface Vessel Based on Typical Missions,” in 3rd International Symposium on Autonomous Systems, ISAS 2019, (2019)
4. K. Matsuzaki, T. Irohara, and K. Yoshimoto, “Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization,” Comput. Ind. Eng., (1999)
5. S. J. Ovaska, “Evolutionary modernization of large elevator groups: Toward intelligent mechatronics,” Mechatronics, (1998)
6. Ovaska, Seppo J. "Procedure for counting moving objects as they stop." IEEE transactions on instrumentation and measurement 3 (1986)
7. B. A. Powell, “Optimal Elevator Banking Under Heavy Up-Traffic,” Transp. Sci., vol. 5, no. 2, pp. 109–121, (1971)
8. Schroder, Joris. "Method and apparatus for the control of elevator cars from a main floor during up peak traffic." U.S. Patent 4,926,976, issued May 22, (1990)
9. Smith, Rory, and Richard Peters. "Control for limiting elevator passenger tympanic pressure and method for the same." U.S. Patent 8,534,426, issued September 17, (2013)
10. F. A. Teixeira, F. M. Q. Pereira, H. C. Wong, J. M. S. Nogueira, and L. B. Oliveira, “SIoT: Securing Internet of Things through distributed systems analysis,” Futur. Gener. Comput. Syst., (2019) page no. 1172-1186
11. Wang, K. C. "Block Device I/O and Buffer Management." In Systems Programming in Unix/Linux, pp. 357-376. Springer, Cham, 2018.
12. V Kureethara, J Biswas, DebabrataSamanta, N G Eapen, ,Balanced Constrained Partitioning of Distinct Objects, International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075(Online). 2019