

정렬 네트워크를 이용한 엘리베이터 군 제어기 설계

Design of Elevator Group Controller using Sorting Network

윤한얼, 박창현, 심귀보
중앙대학교 전자전기공학부

Han-ul Yoon, Chang-hyun Park, and Kwee-bo Sim
School of Electrical and Electronic Engineering, Chung-Ang University
E-mail: huyoon@wm.cau.ac.kr

ABSTRACT

현재 복수개의 엘리베이터를 최적 제어하기 위해서 엘리베이터 군 제어 시스템이 사용되고 있다. 일반적인 엘리베이터 군 제어 시스템에서는 임의의 층에서의 호출(Hall Call)에 대해 복수개의 엘리베이터로부터 정보를 받는다. 다음으로 각 엘리베이터의 매력함수값을 산출, 매력함수들을 비교한 후 가장 큰 값을 갖는 엘리베이터를 호출한 층으로 보내는 작업을 수행한다. 본 논문에서는 16개의 엘리베이터와 정렬 네트워크(Sorting Network)를 이용하여 각 엘리베이터들의 매력함수값의 비교를 수행하는 엘리베이터 군 제어부를 설계한다. 이를 통해 제어부에서 매력함수값들의 비교 연산에 소요되는 처리속도를 향상시킬 수 있다.

Key words : 정렬 네트워크, Elevator Bits Representation, 엘리베이터 매력함수, Hall Call, Car call, 엘리베이터 군 제어부

1. 서 론

대부분의 고층 건물들은 복수개의 엘리베이터를 가지고 있다. 각 엘리베이터 마다 상/하향 호출 버튼을 가진 기존의 제어 방식은 복수개의 엘리베이터 제어에 사용될 경우, 불필요한 이동에서 오는 에너지 손실에 대해 유연하게 대처할 수 없었다. 이에 해결하기 위해 엘리베이터 최적 제어 방식에 대한 활발한 연구가 수행되었다. 엘리베이터 제어 시스템의 발전 과정을 보면, 1980년대 후반까지 전문가 & 지식 기반 시스템(Expert & Knowledge-Based System)이 주를 이루었다. 이후 1990년대 초에 만들어진 퍼지 & 신경망(Fuzzy & Neural Networks) 제어 시스템은 현재까지도 사용되고 있으며, 1990년대 후반에 등장한 해석적(Analytic Solution) 제어 시스템의 과정을 거쳐, 현재는 엘리베이터 군 제어시스템이 널리 쓰이고 있다[1].

엘리베이터 군 제어 시스템이란 임의의 층에

서의 호출이 있을 때, 복수개의 엘리베이터의 정보를 받아 매력함수를 산출한 후, 매력함수값이 가장 높은 엘리베이터를 호출층으로 보내는 시스템이다. 매력함수값 산출을 위해 어떤 층에서 호출이 오면 각 엘리베이터 들은 자신이 가지고 있는 정보를 보낸다. 이 정보들을 통해 각 엘리베이터 들의 매력함수값이 구해지고 가장 매력이 높은(즉 해당 층까지 이동했을 때 시간비용, 공간비용, 에너지 손실 등이 가장 적은) 엘리베이터를 찾기 위해 값들을 비교하게 된다.

본 논문에서는 각 엘리베이터들의 매력함수값들을 비교할 때, 소프트웨어적 정렬 알고리즘을 이용하여 프로세서가 처리를 하는 방식 대신 하드웨어적 정렬 네트워크를 이용하는 방식을 이용한 16개의 엘리베이터와 엘리베이터 군 제어기를 구현한다. 구현에는 Altera사의 Max-Plus II를 사용하고, Waveform Editor를 이용한 시뮬레이션을 통해 위 방식을 적용했을 때의 성능을 평가한다.

II. 정렬 네트워크(Sorting Network)

정렬 네트워크(Sorting Network)란 n개의 입력값을 정렬해 낼 수 있는 k개의 비교기(comparator)로 구성된 네트워크를 말한다. 많은 수학자와 공학자들은 버블정렬이나 삽입정렬 등과 같은 루프의 반복적인 수행을 필요로 하는 방법과는 다른, n개의 입력값들이 k개의 비교기를 거쳐 정렬이 되는 네트워크 방식에 대해 연구를 해왔다. 유명한 것으로는 16개의 입력에 대해 가장 최적의 해인 60개의 비교기를 이용한 Green의 정렬 네트워크(그림1), 공진화 모델에 기초한 유전 알고리즘으로 16개의 입력을 정렬하는 61개 비교기의 구성을 찾아낸 Hillis의 공진화 정렬 네트워크(co-evolving sorting network)(그림2), 최근에는 END(Evolving Non-Determinism) search algorithm을 이용하여 최적화인 60개 비교기로 구성된 END algorithm 정렬 네트워크(그림3) 등 여러 연구 결과들이 나와있다.

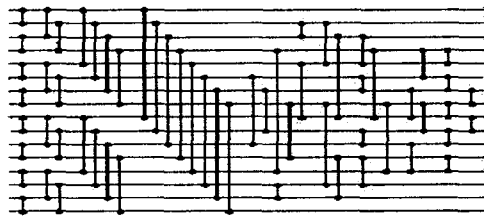


그림1. Green 모델: 60 비교기, 10 평행 스텝, 16 입력 정렬 네트워크

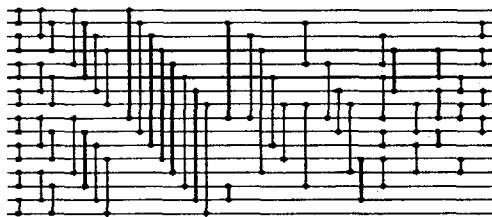


그림2. Hillis 모델: 61 비교기, 11 평행 스텝, 16 입력 정렬 네트워크

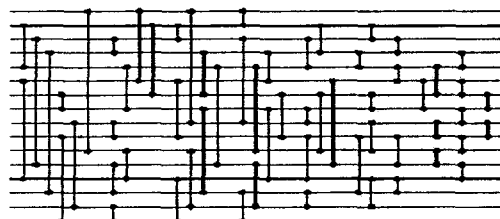


그림3. END 모델: 60 비교기, 10 평행 스텝, 16 입력 정렬 네트워크

정렬 네트워크의 장점은 정렬 알고리즘을 시각적으로 나타내므로 이를 회로제작에 직접 적

용할 수 있다는 것이다. 또한 표1를 기준으로

	comparator stages	comparators
Odd-even transposition sort	$O(n)$	$O(n^2)$
Bubblesort	$O(n)$	$O(n^2)$
Bitonic sort	$O(\log(n)^2)$	$O(n \cdot \log(n)^2)$
Odd-even mergesort	$O(\log(n)^2)$	$O(n \cdot \log(n)^2)$
Shellsort	$O(\log(n)^2)$	$O(n \cdot \log(n)^2)$

표1. 여러 정렬 알고리즘들의 시간·공간 복잡도

n=16 일 때를 비교해 볼 때 정렬 네트워크는 Time Complexity = 10×tick, Space Complexity = 60×the area occupied by comparator로 매우 효율적임을 알 수 있다(Green's construction 기준).

III. 엘리베이터 구현

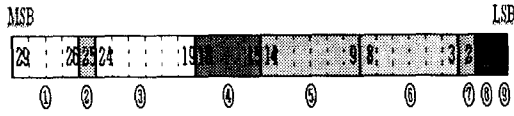
엘리베이터 군 제어 시스템을 구성하기 위해 16개의 엘리베이터를 구현한다. 우선 두 가지 용어를 정의한다[2].

- Hall Call - 고객이 임의의 층에 도착하여 상향 또는 하향 버튼을 누르는 것
- Car Call - 고객이 엘리베이터에 탑승한 다음 목적층의 버튼을 누르는 것

앞에서 언급했던 엘리베이터 군 제어 시스템을 위의 정의된 두 용어와 구현을 위해 설정한 환경을 적용하여 다시 표현하면 다음과 같다.

임의의 층에서 Hall Call이 발생하면, 16개의 엘리베이터로부터 정보를 받아 각 엘리베이터의 매력함수값을 산출한 후, 16개의 값을 비교하여 그 값이 가장 높은 엘리베이터를 호출층으로 보낸다.

임의의 Hall Call에 대해 제어부로 자신의 정보를 보내기 위해 엘리베이터는 자신의 정보를 저장할 레지스터를 필요로 한다. 또 임의의 엘리베이터가 동작 또는 고장(Enable or Disable), 이동 중 또는 정지 중(Move or Stop) 등의 정보를 가져야 한다. 그 외에도 환자 수송 등의 긴급한 상황을 위한 긴급 호출 역시 유용한 기능일 것이다. 본 논문에서는 이러한 정보들을 담기 위해 1개의 엘리베이터가 30bit의 정보 데이터를 갖게 하였다. 각 정보들을 표현하기 위해 할당된 bit(또는 bits)의 자세한 내용은 다음과 같다.



Total 30bits = Elevator ID#(9,0)

그림4. 엘리베이터의 bit 할당별 정보 데이터

<Bit representation>

- ① BIT[29..26] = Elevator_ID# - 1 to 16
- ② BIT[25] = each elevator Enable/Disable
- ③ BIT[24..19] = Current Floor
- ④ BIT[18..15] = Crowd Rate in the elevator
- ⑤ BIT[14..9] = Car Call memory Register File
- ⑥ BIT[8..3] = Next Destination Floor Register File (Control Logic will send this value!!)
- ⑦ BIT[2] = Move/Stop
- ⑧ BIT[1] = Up/Down Moving Direction
- ⑨ BIT[0] = Emergency Call

<Environments & Capacities>

- Total number of elevators: 16
- Total floor: 0 to 63
- Elevator velocity: 1.5m/s
- Height of each floor: 3m
- Limit of person loading: 15personnel

그림5는 엘리베이터와 제어부간의 데이터 흐름을 나타낸 것이다. 제어부에서 Hall Call을 보냈을 때 Enable=1인 모든 Elevator는 자신의 정보를 보낸다. 그 뒤 제어부로부터 16개의 매력함수값이 계산된다. 이 값들은 제어부 내부의 정렬네트워크를 통해 작은값→큰값 순으로 정렬 되어, 보낸 Hall Call에 대해 가장 적절한 Elevator의 Next destination floor Register File을 Hall Call floor가 추가된 새로운 데이터로 업데이트 한다. Current floor (BIT[24..19])는 Move(BIT[2])=1 이고 Up(BIT[1])=1 이면 +1씩 증가하고, Move (BIT[2])=1 이고 Up(BIT[1])=0 이면 -1씩 감소, Move (BIT[2])=0 즉 Stop이면 변화가 없도록 설계하였다. 엘리베이터 또한 Hall Call이 발생한 층으로 이동 시 중간에 서게 되는 층 수를 계산하기 위해 정렬 네트워크를 가진다. 이 부분에 대한 자세한 설명은 다음장의

제어부에서 하기로 한다.

Emergency Call(BIT[0])는 임의의 층에서 환자발생과 같은 긴급상황을 위해 설정되었다. 이 BIT이 ON되면 Emergency Call이 발생한 층에서 가장 가까운 층에 있는 엘리베이터는

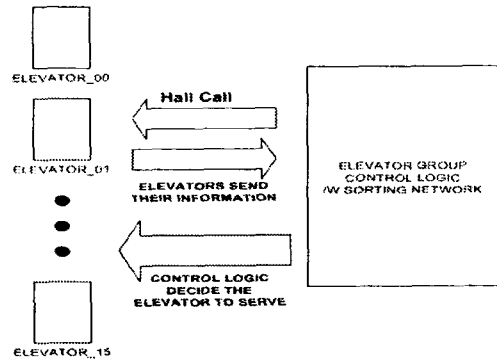


그림5. 엘리베이터와 제어부 간의 데이터 흐름

이동 중 다음 층에 정지하여 승객들에게 긴급 상황으로 인해 내려줄 것을 요구하는 방송을 내보낸다. 이후 환자발생의 Emergency Call이 발생한 층으로 이동한다.

IV. 엘리베이터 군 제어부 구현

엘리베이터 군 제어부는 Hall Call signal을 각 엘리베이터로 보내고, 정보를 받아 엘리베이터 들의 매력함수값을 산출, 가장 매력이 높은 엘리베이터의 Next Destination Floor를 Hall Call이 발생한 층으로 만들어 주는 기능을 수행한다. 엘리베이터의 매력 함수 모델에는 여러 종류들이 있지만 본 논문에서는 다음의 식을 사용하였다.

$$f(i,j) = w_1h(i,j) + w_2g(i,j) + w_3c_i + w_4t(i)$$

$h(i,j)$ 는 엘리베이터 i 의 현재 위치에서 j 층까지의 물리적 거리를 반영하는 함수이다. $g(i,j)$

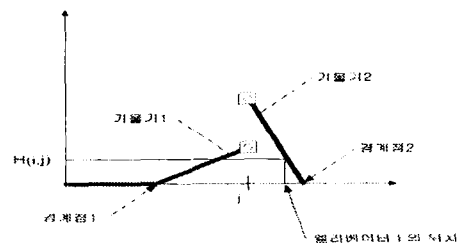


그림 6. 함수 H(i,j)의 모양

는 엘리베이터 i 가 j 층 근방에 있는 층에 이미 정지 계획이 있을 때 보상을 해주는 함수이다. c_i 는 엘리베이터 i 의 탑승인원 수를 %로 나타낸다. $t(i)$ 는 엘리베이터 i 의 현재 위치와 j 층 사이의 층들 중 엘리베이터 i 가 정지하기로 계획되어 있는 층들의 수이다. w_1, w_2, w_3, w_4 는 각 함수들의 weight들을 나타낸다[2]. 본 논문에서는 $h(i,j)$ 와 $g(i,j)$ 가 매우 유사한 형태라는 점을 감안, 두식을 $w_H H(i,j)$ 로 근사화 해주었다(그림6 참조, $w_H = w_1 + \alpha w_2$, α 는 $g(i,j) \neq h(i,j)$).

매력함수 식을 보면 알 수 있듯이 제어부는 엘리베이터들로부터 받은 정보를 가지고 ① (엘리베이터의 현재층 - Hall Call이 발생한 층)을 계산, ② 엘리베이터에 내장된 카메라나 무게 센서로부터 탑승인원을 파악, ③ 만약 Hall Call이 발생한 층으로 이동한다면 중간에 서게 되는 층 수를 계산 ④ 이들의 총합을 그 엘리베이터의 매력 함수 값으로 하고, ⑤ 16개의 값을 비교하여 ⑥ 가장 높은 매력함수값을 가지는 엘리베이터의 Next destination floor Register File의 적절한 위치에 Hall Call이 발생한 층을 삽입해야 한다.

ELEVATOR GROUP CONTROL LOGIC

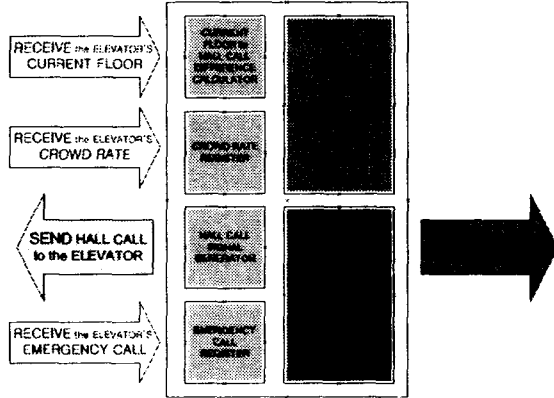


그림7. 엘리베이터 군 제어부의 구성

현재 층과 Hall Call이 발생한 층과의 거리를 계산해 주는 Current floor to Hall call floor Difference Calculator 부분은 Carry Look Ahead Adder를 이용한다[3]. 두 입력 중 Hall call floor을 1과 XOR하여 넣어주면 아래 수식

$$\begin{aligned} & \text{Current floor} + (\text{Hall call floor} \oplus 1) + C_{in}(=1) \\ & = \text{Current floor} + \overline{\text{Hall call floor}} + 1 \\ & = \text{Current floor} - \text{Hall call floor} \end{aligned}$$

이 성립하므로 두 층간의 Difference를 구할 수 있다[4]. 이 연산에서 발생하는 Borrow는

Up/Down moving direction register; BIT[1]에 저장한다. BIT[1]=1일 때(current floor - hall call floor is negative value) Current floor < Hall call floor 이므로 엘리베이터가 UP해야 한다는 것을 나타내고, 이와 반대로 BIT[1]=0일 때(that is positive value) Current floor > Hall call floor 이므로 DOWN 해야 한다는 것을 나타낸다.

탑승객수/정원 비율	BIT[18..15]
0/15	0000 (empty)
1/15	0001
2/15	0010
3/15	0011
...	...
14/15	1110
15/15	1111 (full)

표2. 탑승객수/정원 비율에 따른 BIT 표현

탑승인원을 파악하는 부분은 엘리베이터에 내장된 무게 센서를 통해 이루어 진다. 총 중량 값을 센싱하여 <현재중량+수용가능중량>을 0%일 때에 0000, 100%일 때 1111을 제어부로 보내게 된다(표2 참조).

Hall Call 발생 층으로 이동한다고 가정 시 중간에 서게 되는 층 수 계산 부분에서 이뤄지는 과정은 다음과 같다. 먼저 Hall Call floor를 엘리베이터의 Car Call Memory Register File로 보낸다. 다음으로 Current floor 값을 Car Call Memory Register File로 보낸다. 그 값을 앞에서 구현한 엘리베이터의 내부에 있는 정렬 네트워크를 통해 정렬한다. 결과적으로 정렬 된 값들에서 Current floor와 Hall Call floor 사이에 있게 되는 층들의 개수가 원하는 값이 된다(예를 들어 정렬 후 Current floor의 위치 위에서 2번째, Hall Call floor 위에서 5번째 라면 중간에 서게 되는 층 수는 2개가 된다).

이 값들에 매력함수값 계산부는 weight 를 배정하여 각 엘리베이터에 대한 매력함수 값을 계산 후, 제어부에 내장된 정렬 네트워크로 보낸다. 제어부 정렬 네트워크를 통과한 데이터 들은 매력 함수값 순으로 정렬이 되고 가장 높은 매력을 가진 엘리베이터(selected elevator by attractive function)의 Next Destination Register File에 Hall Call floor를 저장한다. 방법은 층들의 개수를 구할 때 엘리베이터 내부의 정렬 네트워크를 거친 데이터

(Current floor & Hall Call floor와 함께 정렬된 데이터)를 Current Floor를 제외하고 Next Destination Register File에 업데이트 해주면 된다(그림8 참조).

System Design and Architecture, Prentice Hall, 1997
 [4] J. M. Yarbrough, *Digital Logic Applications and Design*, PWS, 1997

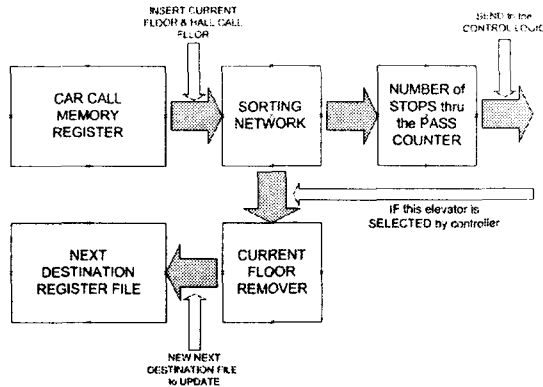


그림8. Next Destination Register File을 업데이트 하는 과정

Emergency Call 발생 시 그 층에서 가장 가까운 곳에 있는 엘리베이터는 다음 층에 일단 정지한다. 긴급 상황이므로 내려달라는 방송을 보내고 고객들이 모두 내리면 Current floor를 제외한 모든 Car Call과 Next Destination 데이터를 Clear하고 Emergency Call이 발생한 층을 Next Destination으로 한다.

V. 결론

본 논문에서는 매력함수값들을 비교할 때, 하드웨어적 정렬 네트워크를 이용하는 모델로 엘리베이터 군 제어를 설계하였다. 결과적으로 임의의 Hall Call 발생부터 엘리베이터의 Next destination register file을 업데이트 하는 과정을 약 8step으로 간소화 할 수 있었고, 매력함수값들의 비교 연산을 10tick으로 줄여 매우 빠른 성능을 보임을 확인 할 수 있었다.

VI. 참고문헌

[1] Young Cheol Cho, Zavarin Gagov, and Wook Hyun Kwon, "Elevator Group Control with Accurate Estimation of Hall Call Waiting Times", *IEEE International Conference on Robotics & Automation*, pp. 1-3, 1999.
 [2] 문병로, *유전알고리즘*, 두양사, 2003
 [3] V. P. Heuring and H. F. Jordan, *Computer*