

# Microcomputer를 이용한 엘리베이터 모델의 制御 (Control of Elevator System Model Using Microcomputer)

宋 炯 彬\*, 卞 增 男\*\*

(Song, Hyung Bin and Bien, Zeung Nam)

## 要 約

엘리베이터 시스템은 位置 및 速度制御를 동시에 요구하는 시스템으로 하드웨어로 構成된 從來의 엘리베이터 시스템의 制御裝置를 최근에 발달된 micro-computer의 도움으로 간단화시킬 수 있는 可能性을 研究 報告하였다.

本 論文은 microprocessor를 이용한 controller hardware의 구성과 本 hardware에 따른 制御方式을 설명하였고, 이 방식을 이용하여 本 시스템이 주어진 速度曲線에 따라 동작하는 것을 實驗을 통하여 보였다. 사용된 엘리베이터 모델의 상태방정식을 사용한 制御프로그램을 適用하여 實驗한 結果, 一般的 엘리베이터 시스템에 micro-computer가 사용될 수 있는 可能性이 있음을 보였다.

## Abstract

A conventional elevator system, which requires simultaneous control of the speed and the position, contains complicated analog hardwares as the control system. Recent advances in LSI technology, however, suggest that the control of such an elevator system may be realized by incorporating digital devices and microcomputer. In this paper, such a possibility is investigated.

In this paper, the digital controller, which is implemented around an IMSAI 8080 microcomputer is designed for the control of model elevator system. Experiments show that this control system tracks the given velocity curve as well as it brings the elevator to the exact point.

## 1. 序 論

엘리베이터 시스템은 位置 및 速度의 制御를 同時에 行해야 하는 시스템으로, 既存의 많은 셔보기구(servo mechanism)가 위치제어나 속도제어의 어느 한 쪽만을 요구하는 경우와는 달리<sup>[7]</sup> 그 制御方法이 비교적 複

雜多様な 것으로 알려져 있다.<sup>[3]</sup> 즉, 실용되고 있는 典型的인 엘리베이터 시스템에서는, 시스템의 位置가 주어진 速度曲線을 좇아 制御되도록 하기 위하여 애널로그 반도체소자를 사용한 함수발생기 및 기타 전자회로, 릴레이와 타이머(timer) 등을 이용한 論理回路, 그리고 엘리베이터의 位置情報를 간접적으로 검출하게 하는 셀렉터(selector or floor controller) 등의 機械의 장치 등으로 制御裝置를 構成하고 있다. 여기서 速度曲線은 실험에 의하여 사람이 可速度에 對

\* 正會員, 홍릉기계 연구원 대전기제창

\*\* 正會員, 韓國科學院 電氣 및 電子科  
(Dept. of Electrical Science, KAIS)

接受日字; 1979年 4月 6日

한 快速性限界에 根據하여 결정된 것으로 一切가 하드웨어로 具現되고 있다.

本 論文에서는 이러한 在來式 엘리베이터 시스템의 制御裝置를, 최근에 소개되고 있는 micro-processor 및 digital components를 이용한 제어장치로서 代替할 수 있는가 하는 可能性을 조사하기 위한 실험연구 결과를 수록하였다. 이를 위하여 실제로 엘리베이터 시스템의 物理的 모델을 제작하고 본 엘리베이터 모델을 구동시키기 위한 電動機설치 및 구동회로의 설계 제작, 속도 및 위치제어를 위한 micro-computer 시스템의 interfacing, 제어이론을 이용한 control program의 작성 등을 行하였다.

2. 既存 엘리베이터 시스템

micro-computer를 이용한 엘리베이터 시스템의 제어장치를 기존 시스템과 비교하기 위하여, 在來式 엘리베이터 시스템에 대하여 간단히 설명하기로 한다.

엘리베이터의 구조는 使用目的과 모터, 運轉方式에 따라 여러가지로 분류될 수 있다.<sup>[6]</sup> 運轉方式에 따라서 단독설치방식과 병렬설치방식의 경우가 있으며, 다시 단독설치의 경우를 collective control, button switch control, signal collective control 등으로 구분하여 구별한다.

collective control이란 비교적 혼잡한 경우에도 능률적으로 운전할 수 있도록 하는, 고도의 수송능력을 발휘하는 全自動 運轉方式이다. 이 방법에서는 엘리베이터 內의 階層 버튼이, 등록된 層 및 운전방향과

同方向의 HALL 부름에 應하여 階層順으로 順次 停止하며, 운전방향의 부름이 없으면 최고층의 부름에 응한후, 방향을 反轉하고 逆방향의 부름에 서비스한다.

button switch control은 CAR 內의 floor button에 의한 운전을 HALL 부름에 우선시키는 專用運轉을 특징으로 하는 방법이다. 운전중에는, "使用中"-lamp를 켜서 다른층의 승객에게 사용중임을 신호로 표시한다. CAR에 타고 行先層의 Button을 누르면 즉시 출입문이 닫히고 운전이 시작된다. 사용이 끝나면 버튼을 눌러 door를 닫는 방식을 취한다.

signal collective control은 정상시에는 collective 방식으로, 승객이 혼잡시에는 signal 방식으로 운전수가 직접 운전하는 自動과 手動的의 복식 운전방식이다.

엘리베이터 시스템에서 주로 고려되어야 할 점은 운전중의 가속속이 유연해야 하고, 주행중에는 속도변화가 적어서 승객에게 안락한 운행이 되도록 해야 하며, 또 정지할 때는 정확한 위치에 도달해야 하는 점이다.

엘리베이터 제어 시스템의 주요부분을 block diagram으로 보인 결과를 그림 1에 실렸다.<sup>[3]</sup> 이 그림에서, 속도지령 장치에 부착되어 있는 selector는

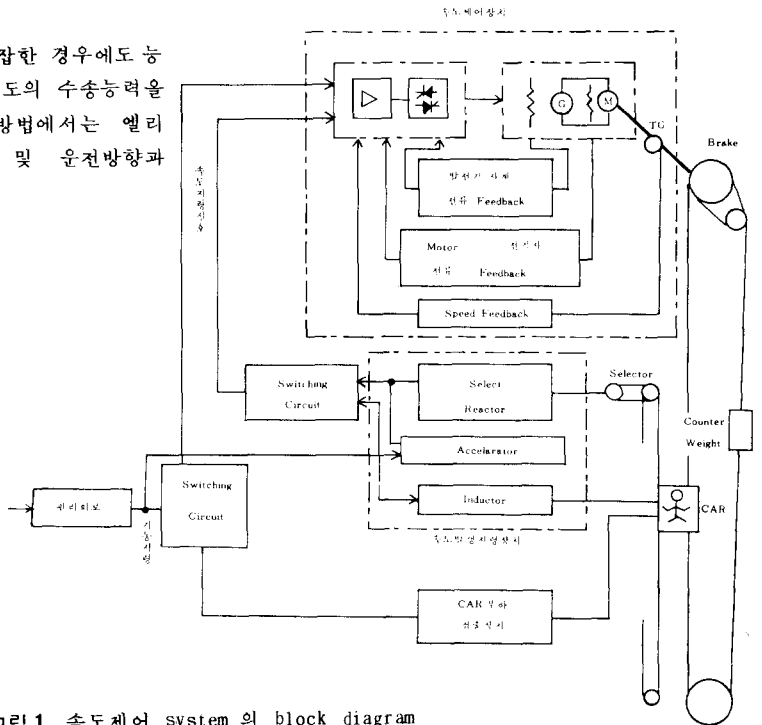


그림 1. 속도제어 system의 block diagram  
Fig.1. Block diagram of speed of control system.

건물 위치정보의 保持장치로 엘리베이터 및 그 주변 건물의  $\frac{1}{100} \sim \frac{1}{150}$  의 축소모형이다. 그것은 실제의 엘리베이터와 기계적으로 링크되어 있다. selector 에는 CAR 와 더불어 같은 시간에 움직이는 同期移動臺가 있고, CAR 가 주행을 개시하면 동시에 CAR보다 빨리 전진하여 정지해야 될 층이 어디에 있는가를 탐색하는 前進移動臺 및, 전진이동대와 동기이동대의 거리차를 연산하는 差動移動臺가 있다.

accelerator 는 엘리베이터가 起動時 얼마간은 CAR 內의 負荷에 依한 기초 torque 를 이기기 위해서 속도지령을 내지만 그 시간을 넘어서면 최고속도에 이르기까지 가속되는 속도지령을 만들어내는 속도지령장치이다.

select reactor 는 차동이동대에 부착되어 있고 감속개시점에 다다르면 남은 거리에 대해서 일정감속도의

속도지령을 발생하는 함수발생기이다.

$\frac{1}{100} \sim \frac{1}{150}$  정도의 축소모형인 Selector 에 의한 제어밖으로는 着床의 정확성을 기하기 어렵기 때문에 着床時에는 inductor 라고 불리는, 승강로에 부착되어 있는 착상지시장치를 병용하는 시스템도 있다.<sup>13)</sup>

### 3. Micro -processor 를 이용한 Controller Hardware 의 구성

그림 2에 본 논문연구를 위하여 사용된 실제적인 시스템의 모양과 시스템의 block diagram 을 보였다.

主 제어장치에 해당하는 micro - computer 시스템은 INTEL 8080 을 CPU로 하는 IMSAI micro computer 시스템을 사용하였고 입력장치로서 ADM 에서 개발된 CRT - terminal 을 사용하기 위해 MIO ( multiple I/O interface card ) 를 사용하였다. 본 논문에서는 엔

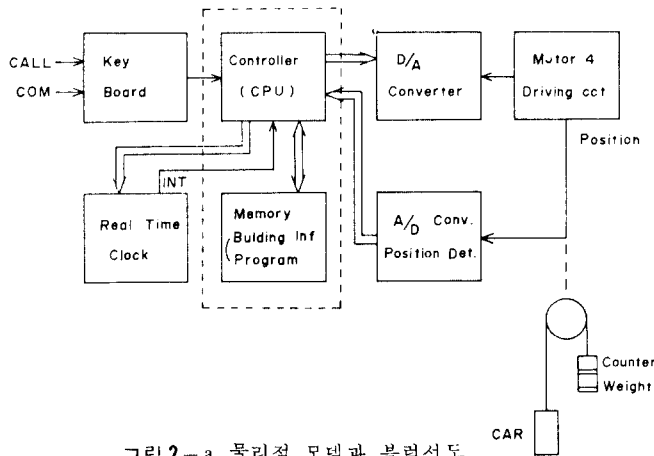


그림 2-a. 물리적 모델과 블럭선도  
Fig-2-a Physical model & block diagram

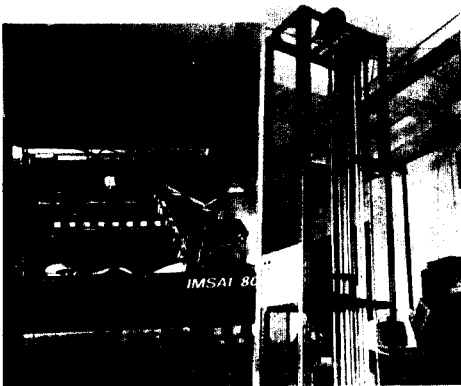


그림 2-b. 물리적 모델사진  
Fig. 2-b. Physical Model.

리베이터 운전 program 을 RAM 에 넣어서 사용 하였으며 language 는 BASIC 과 8080 assembler language 를 병용하였다.

이 micro - computer 시스템이 하는 일은 다음에 기술할 RTC (real time clock) 와 DAC (digital to analog converter), ADC (analog to digital converter), 모터의 방향을 제어하는 릴레이를 총괄하는 일로서 주 제어장치의 역할을 한다.

sampling interval 의 결정은 소프트웨어에 의한 방법과 하드웨어로 하는 방법이 있다. 그러나 소프트웨어에 의한 방법은 sampling time 들 사이에 다른 일을 computer 가 처리하기 힘들기 때문에 거의 불가능하다. 따라서 RTC 의 필요성이 생긴다. 즉, RTC

는 시간을 재는 programmable timer 의 역할을 한다.

전체 시스템의 개략적인 동작은 다음과 같은 순서를 거친다.

어떤 층에 관한 명령이 ADM-3A terminal을 통해 들어오면 그 입력은 micro-computer에서 프로그램의 도움을 받아 운행에 필요한 정보를 만들어 내고, RTC로 sampling interval을 잡아서 program(memory)에 미리 기억시켜둔 속도곡선에 따라 속도를 만들어 낸다. 이때 각 sampling time마다 속도를 계산하여 DAC를 통하여 모터 구동회로에 전달시키는데, 이 계산과정에서 사용되는 위치정보는 모터 shaft에 연결되어 있는 potentiometer에서 ADC를 통해 얻는다. 이때에 DAC에 가해 주는 값을 micro-computer에서 계산할 수 있도록 모터의 discrete-time state equation은 미리 memory에 기억되어 있다. 앞의 동작순서에서 본 바와 같이 엘리베이터 시스템에서 요구하는 HALL부름에 관한 제어나 속도곡선에 따른 속도 및 위치제어 등의 엘리베이터 시스템 고유의 제어는 모두 software로 구성되어 있다.

#### 4. 제어방식

##### 4-1 엘리베이터 모델의 Discrete-Time State Equation

DC motor의 position에 대한 transfer function은 다음과 같이 Modeling 할 수 있다.<sup>[1]</sup>

$$\frac{Y(s)}{R(s)} = \frac{G}{s(s+\tau)} \quad (\text{Eq-1})$$

여기서 Y(s)는 position이고 R(s)는 모터회로에 가해주는 입력을 표시한다. Y를  $X_1$ ,  $\dot{y}$ 를  $X_2$ 로 표시하면 Eq-1은 다음의 continuous-time state equation으로 표시 된다.

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\tau \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} Y \quad (\text{Eq-2})$$

$$Y = [1 \ 0] \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (\text{Eq-3})$$

여기서 Y =  $X_1$ 은 position,  $X_2$ 는 속도를 나타낸다. 본 시스템에서 입력 Y은 DAC를 통해 모터에 가해지는 입력으로 sampling interval T동안 입력값이 일정하다고 볼 수 있으므로 다음과 같은 discrete-time state equation을 얻을 수 있다.

$$\begin{bmatrix} X_1(k+1)T \\ X_2(k+1)T \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{\tau}(1-e^{-\tau T}) \\ 0 & e^{-\tau T} \end{bmatrix} \begin{bmatrix} X_1(kT) \\ X_2(kT) \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{G}{\tau} \left( T + \frac{1}{\tau} e^{-\tau T} - \frac{1}{\tau} \right) \\ \frac{G}{\tau} (1 - e^{-\tau T}) \end{bmatrix} r(kT) \quad (\text{Eq-4})$$

$$Y(kT) = [1 \ 0] \begin{bmatrix} X_1(kT) \\ X_2(kT) \end{bmatrix} \quad (\text{Eq-5})$$

##### 4-2 Stae Estimator

본 연구에서 사용된 시스템에서는 속도와 위치를 전부 검출하는 것이 아니고 위치만을 검출하기 때문에 complete state feedback이 아닌 경우이다. 그러나 위치정보만을 가지고는 앞에서 유도한 discrete-time state equation에서 요구되는 속도곡선을 따라가게 하는 입력 r을 계산할 수 없다. 따라서 불충분한 state 정보로부터 complete state vector값을 얻기 위해서 state estimator의 필요성이 생긴다.

estimated state  $\hat{X}(k) = \begin{bmatrix} \hat{X}_1(kT) \\ \hat{X}_2(kT) \end{bmatrix}$ 는 다음과 같은

과정을 통해 정할 수 있다.<sup>[2]</sup> 즉, Eq. 4, 5에서 kT를 k로 쓰면,

$$\hat{X}(k+1) = A \cdot \hat{X}(k) + B \cdot r(k) + K[Y(k) - S \cdot \hat{X}(k)] \quad (\text{Eq-6})$$

여기서  $\hat{x} = \begin{bmatrix} \hat{x}_1(kT) \\ \hat{x}_2(kT) \end{bmatrix}$ ,  $S = [1, 0]$ ,

$$A = \begin{bmatrix} 1 & \frac{1}{\tau}(1 - e^{-2T}) \\ 0 & e^{-2T} \end{bmatrix} \quad B = \begin{bmatrix} \frac{G}{\tau} \left( T + \frac{1}{\tau} e^{-\tau T} - \frac{1}{\tau} \right) \\ \frac{G}{\tau} (1 - e^{-\tau T}) \end{bmatrix}$$

이며 K는 feedback gain으로서 원하는 시스템 동작 특성을 갖도록 정할 수 있다.<sup>[1]</sup>  $x_2(k)$  estimation에서  $y(k) - y(k-1)$ 을 사용하지 않은 것은 Continuous System에서 noise에 sensitive한 이유등으로 pure differentiator를 이용한 state estimation을 하지 않은 이유와 유사하다.<sup>[8]</sup>

만약 측정 장치등에 noise가 있는 경우에는 noise 영향을 줄이기 위해서 고정된 이득 vector K 대신에 kalman filtering technique으로 K(k)를 정하여 사용한다.<sup>[2]</sup>

##### 4-3. 제어방식

앞절에서 구한 狀態方程式 및 state estimator를 사용해서, 주어진 속도곡선에 따라서 모터의 위치 및 속도제어를 행하는 실제의 제어순서는 다음과 같이 정할 수 있다. 즉, Eq-6으로부터,  $B = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$ 라 놓으면

$$r(k) = \frac{1}{G_2} [X_2(k+1) - A_2 \cdot X_2(k)] \quad (\text{Eq-7A})$$

그런데 Eq-7 에 사용되는  $X_2(k+1)$ 은 미리 기억시켜 둔 속도곡선에서의 값을 사용하고  $X_2(k)$ 는 Eq-6 로 추정된  $\hat{X}_2(k)$ 를 사용한다.

$$r(k) = \frac{1}{G_2} [X_2(k+1) - A_2 \cdot \hat{X}_2(k)] \quad (\text{Eq-7B})$$

K 번째 sampling time에서의 제어순서를 정리하면 다음과 같다 :

- STEP 1  $r(k)$ 를 ADC 에 加한다.
- STEP 2 위치출력  $Y(k)$ 를 DAC 로부터 읽어온다.
- STEP 3 STEP 2에서의  $Y(k)$ 로 Eq-6 을 사용하여  $\hat{X}(k+1)$ 을 계산한다.
- STEP 4 STEP 3에서 계산한  $\hat{X}_2(k+1)$ 과 기억시킨 속도  $X_2(k+2)$ 로 Eq-7B 를 사용하여  $r(k+1)$ 을 계산한다.
- STEP 5 sampling time 이 k+1 번째가 되기를 기다린 다음 STEP 1으로 가서 위의 과정을 반복한다.

#### 4-4. Minicomputer Simulation

simulation 은 sampling interval 이 0.1 이고 모터의 위치에 대한 transfer function 이

$$\frac{Y(s)}{R(s)} = \frac{1}{s(s+1)}$$

인 경우에 대해 NOVA 830 minicomputer 으로 행하였다.

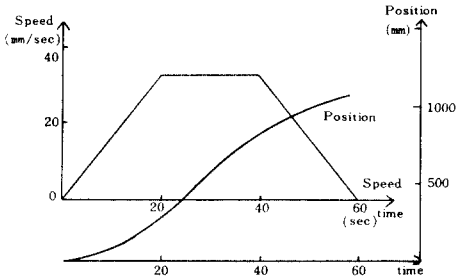


그림 3. 위치 및 속도 곡선  
Fig. 3. Position and speed curve without noise.

그림 3은 noise 가 없는 경우의 complete state feedback 를 통한 위치 곡선이며 그림 4는 gaussian noise 를 첨가 시켰을 경우, 위에 제시한 제어 순서를 쫓아 simulation 한 결과이다.

이상의 simulation 결과로 보아 4-3 절에서 제시된 제어 순서는 충분히 실용 가능함을 확인 하였다.

#### 4-5. G 및 $\tau$ 의 결정

앞에서 제시한 제어방식을 사용하려면 실제 시스템

의 parameter 값을 정할 필요가 있음을 알 수 있다. 따라서 G 및  $\tau$ 를 step input response 실험에 의하여 결정하였다.

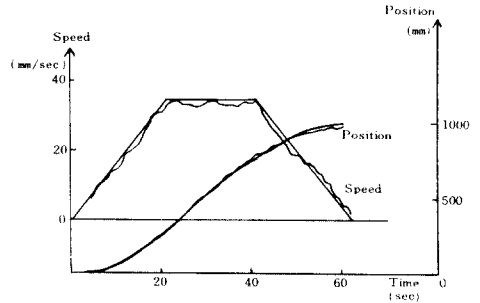


그림 4. 잡음과의 위치 및 속도곡선  
Fig. 4. Position and speed Curve under noise.

실험에 依하면  $\tau = 0.63$ ,  $G = 0.055$  이고 이에 대한 OFFSET 이 130 정도 (DAC Value) 존재함을 알 수 있다.

#### 4-6. 속도곡선에 따른 실험

앞의 4-3 에 따른 제어순서를 IMSAI micro-computer 로 계산하는데 약 0.6초가 걸리므로 sampling time 을 1초로 잡았다.

4-5 절에서 구한 G 및  $\tau$ 로 sampling interval T 가 1인 discrete-time state equation 의 characteristic equation 의 root 는 절대값이 1 이하로 되어 Stable 하다. 그 결과는 그림 5에 그려져 있다. 실선으로 표시된 것이 이론값이고 점선이 실험값이다.

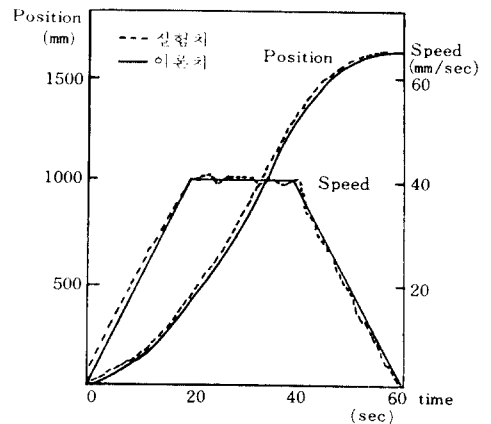


그림 5. 이론적 및 실험적 곡선의 비교  
Fig. 5. Comparison of theoretical and empirical curves.

이 결과는 기억된 속도곡선에 의하여 본 시스템의 DC 모터를 4-3에 기술된 제어순서에 따라 제어할 수 있음을 보여준다.

5. 엘리베이터 제어 프로그램

5-1. Button Switch Control

앞에서 설명한 것 같이 이 경우는 CAR가 운전중에는 使用中 lamp를 켜서 다른 사용자가 사용을 할 수 없게 되어 있다. 사용을 끝내면 door를 닫아서 다른 사용자의 부름에 응할 수 있게 하는 방식이다.

이 프로그램에서 앞부분에서는 discrete-time state Eq의 상수를 미리 계산하고 그 상수로 state estimator kmatrix를 미리 계산하고, 아울러 속도곡선도

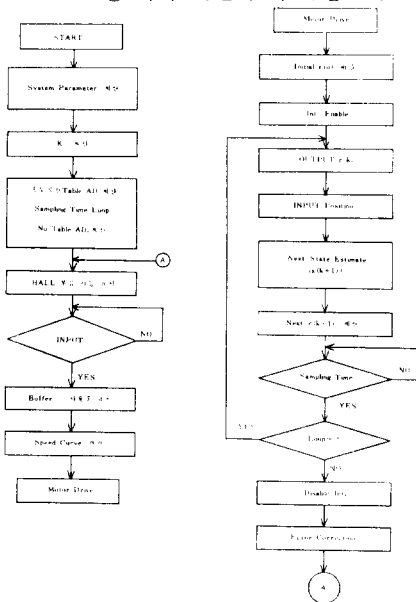


그림 6. Button switch control의 flow chart  
Fig. 6. Flow chart of button switch control.

계산한 다음 실제의 프로그램으로 들어가게 되어 있다.

실제의 프로그램 중에서 계산이 필요한 부분은 BASIC language로 프로그램이 구성되어 있고 ADC, 릴레이 제어 등의 입력, 출력에 관한 프로그램은 INTEL 8080 assembler language로 구성되어 BASIC과 연결 동작된다.

이 프로그램에서 어려운 점은 BASIC에서 Interrupt를 직접처리를 할 수가 없는 점이다. 그래서 Interrupt가 걸리면 A000H라는 memory location으로 Jump하여 interrupt event pointer(software) "1"로 만들어 준다. 따라서 BASIC 프로그램에서는

interrupt event pointer를 조사하여 interrupt의 유무를 알아낸다.

이 프로그램의 flow chart는 그림 6에 실려있다.

6-2. Collective Control

이 경우는 앞의 경우보다 복잡한 제어동작이 요구된다. 대부분은 앞의 button switch control의 경우와 같으나 이 경우는 운전중에도 서비스가 가능한 HALL 부름에 대하여는 서비스를 하여야 하므로 HALL 부름이나 명령에 대한 제어과정이 필요하다. 특히 운전중에 HALL 부름이 들어오는 경우를 생각하면 이 경우의 프로그램은 현재 sampling time과 다음 sampling time 사이에 모든 처리를 하여야 한다.

입력 r을 만들어 내는 계산은 0.6초씩 걸리는데 sampling interval은 1초이므로, 0.4초동안에 HALL 부름이 전의 sampling interval에서 없었으면, HALL 부름과 또 sampling time이 됐는가를 반복하여 조사한다. 만일 HALL 부름이 들어오면 그 sampling interval에서는 HALL 부름이 있는 층과 서비스를 원하는 방향만 기억시킨 다음 sampling time을 기다린다. 다음 sampling interval에서는 HALL 부름을 조사하지 않고 전의 sampling interval에서 들어온 HALL 부름의 서비스의 가능성을 조사하여 서비스가 가능하면 속도곡선을 다시 찾고 다음 sampling time을 기다리고, 서비스가 불가능하면 그 HALL 부름을 서비스 buffer에 기억시키고 다음 sampling time을 기다린다.

앞과 같이 두 개의 sampling interval에 나누어서 HALL 부름을 처리하는 이유는 sampling interval마다 0.4초의 여유분이 없기 때문이다.

이 프로그램의 flow chart는 그림 7에 실려있다.

6. 結 論

앞에서 본대로 시스템과 같이 micro-computer를 사용하여 엘리베이터 시스템을 구성하면 다음과 같은 장점이 있을 것으로 기대된다.

첫째, 기존의 엘리베이터 시스템은 제어방식이 hardware로 구성되어 있기 때문에 제어방식을 바꾸는 것이 어려우나 본 시스템은 제어방식을 소프트웨어로 구현하기 때문에 보다 더 나은 제어방식이 발견되었을 때 hardware를 고치지 않고도 소프트웨어의 수정만으로 가능하다. 단, noise가 있는 경우에 kalman filter 등의 technique을 사용할 수 있다.

둘째, 본 시스템의 하드웨어로 엘리베이터 시스템이 아닌 다른 시스템에 적용할 경우 속도곡선을 다르

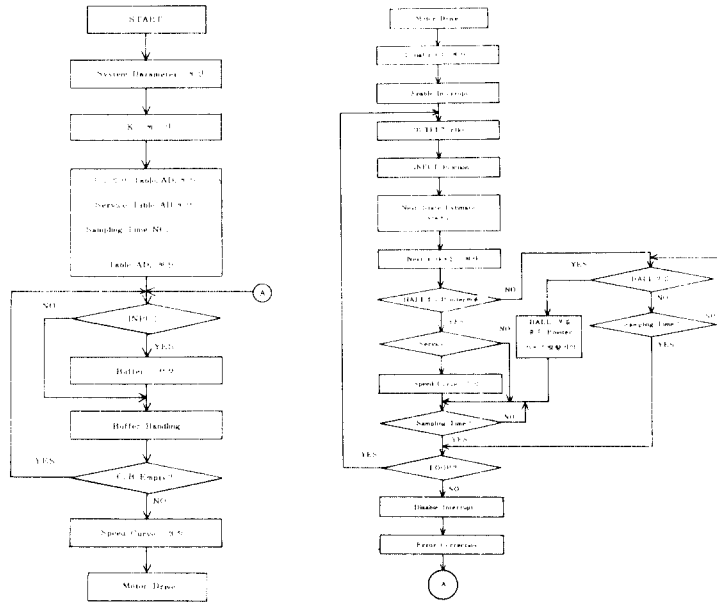


그림 7. Collective control 의 Flow chart  
Fig.7. Collective control flow chart.

게 적용할 필요가 있다면, 단지 기억된 속도곡선을 수정하기만 하면 사용자가 원하는 대로 이용할 수 있다.

셋째, 기존 엘리베이터 시스템에서 건물의 모형인 selector 대신 위치 검출회로만이 필요하고, 관리회로등의 하드웨어가 소프트웨어로 대체되므로 회로가 간단하여 진다.

넷째, 기존의 엘리베이터 시스템보다 싼 값으로 만들기가 가능할 것으로 생각된다.

다섯째, 모터와 모터구동회로를 제외한 거의 모든 부분이 digital 회로이므로 보수 고장 발견이 용이하다.

本 논문 연구의 하드웨어를 구성하여 실험을 하면서 발견된 사실은 엘리베이터의 위치정보를 검출하는 것과 micro-processor의 processing time 이 문제가 될 수 있다는 점이다. 本 실험에서는 위치를 검출하기 위하여 10-turn potentiometer 를 사용하였는데, 이런 type 을 실제의 엘리베이터 시스템에 적용하기는 어렵다. 보다 더 정확하고 넓은 범위의 위치를 검출하기 위해서는 incremental shaft encoder, 등의 모터의 회전 수의 범위를 제한하지 않는 장치가 바람직하

다. 또 유의 되어야 하는 것은, sampling interval 이 제어방식에 제한을 받는 점이다. 즉, 제어방식이 복잡해져서 계산식이 많아지면 micro-computer 의 계산능력 때문에 sampling interval 을 짧게 잡을 수 없다. 이 문제점은 multiprocessor 를 이용한 distributed control 을 하거나 속도가 빠른 micro-processor 를 사용하고 계산능력을 향상시키기 위해 floating point arithmetic 회로 등을 부착하여 해결될 수 있을 것으로 기대된다.<sup>[4]</sup>

사람을 태우는 실제의 엘리베이터에 micro-processor 를 이용한 제어장치를 사용하는 것은, 본 실험 연구를 통해 보여준 가능성외에 여러가지 부속장치의 설계 및 안전도와 신뢰도시험 등이 행해져야 할 것으로 생각되며, 선진국에서는 현재 field test 중이라고 알려져 있고, 우리나라에서도 elevator maker 인 금성사에서<sup>[6]</sup> 이러한 type 의 제어장치를 구성하기 위한 planning 이 행해지고 있다고 한다.

參 考 文 獻

1. James A. Cadzow & Hinrich R. Martens, "Discrete-time and Computer Control Systems," Prent-

- ice-Hall, 1970.
2. Karl J. Astrom, "Introduction to Stochastic Control Theory," Academic Press, New York & London, 1970.
  3. 渡邊英起, 滿原接行, "高速 Elevator 의 速度制御와 群管理," System & Control, Vol 22, No. 6, 1978.
  4. "IMSAI USER'S MANUAL," IMSAI Corporation, 1977.
  5. 손상권, "高速 Elevator 의 全自動 群管理 운전 방식" 대한 전기학회 제측제어 분과 위원회, 제 7 회 학술연구발표회 1978. 7.
  6. W. W. Koepsel, "Microprocessor Motor Control," presented at ROC-ROK Workshop on Microprocessor Devices and Application, Taipei, Tai Dec, 1978.
  7. A. Lin and W. W. Koepsel, "Microprocessor Speed Control," IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI - 24, No 3, Aug. 1977, pp. 241-247.
  8. C. T. Chen, "Intro. to Linear System Theory," Holt, Rinehart, & Winston, 1970.

