

## 현대 제어 이론의 구현과 D. S. P. 소자의 사용

金國憲, 權淳萬, 金鍾洙  
韓國電氣研究所 制御応用研究室

### I. 서 론

Control system은 IEEE 표준 전기전자 사전에서 "An assemblage of control apparatus coordinated to execute a planned set of controls"로 정의되고 있다. 실제로 "제어"라고 하는 문제는 제어시스템을 얘기하기도 하지만 협의의 제어 즉 control law만을 의미하는 경우도 대단히 많다. Control law는 "A function of the state of a plant and possibly of time, generated by a controller to be applied to a plant"라 설명되며, 이론제어 분야의 거의 모두가 control law의 해석과 규명에 집중되어 있다고 하겠다. 그러나 물리적인 방법에 의해서 구현될 수 없는 제어 법칙은 엔지니어의 관점에서는 그림의 떡과 다를 바가 없으며 1960년대 이전의 Kalman filter가 갖던 의미 밖에는 부여할 수 없을 것이다.

본고에서는 최근 10여년간에 걸쳐서 획기적인 발전이 있었던 분야 중의 하나인 적응제어 이론과 D. S. P.<sup>1)</sup> 소자를 사용한 구현 기법에 관하여 다루고자 한다.

"시변 또는 미지의 매개 변수를 갖는 시스템의 제어"를 위해 제안되었다고 생각할 수 있는 적응제어 시스템은 이론적으로 많은 논문들이 발표되었지만 실용용에서는 화학공정 등의 slow process에 대부분 사용되었다. 그러나 최근들어서는 electromechanical 또는 electromagnetic system 등의 fast system 제어에 응용되어 많은 좋은 결과들을 보여 주고 있다. 이것들이 가능해진 것은 최근의 범용 마이크로 프로세서 및 고속연산용 D. S. P. 소자 개발 기술의 획기적 발전에 힘입은 바 크다 하겠다.

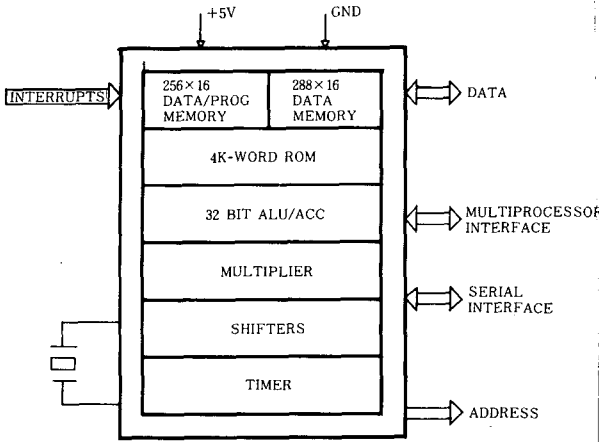
현재까지 개발된 구현기술 및 예, 그리고 현재 한

국전기연구소에서 계획 중인 연구내용의 도입을 위하여 2장에서 D. S. P. 소자에 대하여 개략적으로 기술하고 3장에서 제어대상 플랜트를 흡인식 자기부상 시스템(electromagnetic suspension system)<sup>13)</sup>으로 할 때의 모형화 및 적절한 적응제어 알고리즘을, 4장에서는 D. S. P. 소자를 이용한 구현에 관하여 기술하고 5장에서 결론을 맺었다.

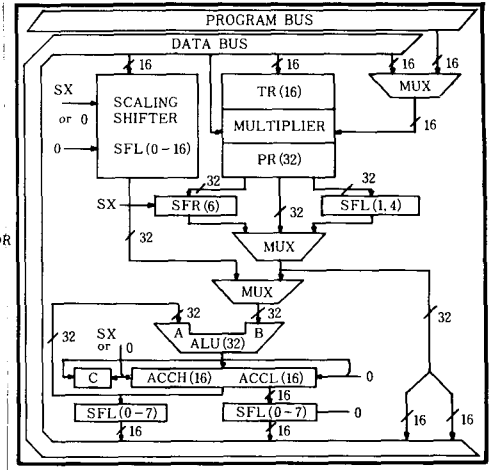
### II. D. S. P. 소자

최초의 D. S. P. chip이라 할 Intel의 2920은 승산기를 내장하지 않고 병렬 시프터와 누산기로 연산부를 구성하고 A/D, D/A 변환기를 내장함으로써 24×12bit의 곱셈을 2μsec 이내로 처리할 수가 있었고 이후 계속된 발전에 힘입어 16×16bit의 곱셈을 35nsec에 처리할 수 있게 되었다. D. S. P. chip은 기본 구성이 마이크로 컴퓨터와 흡사하여, 1chip의 LSI이면서도 내부에 프로그램 저장용 메모리, 데이터 저장용 메모리, 산술 연산회로, 제어회로를 가지며, 프로그램에 의해 디지털 신호 처리를 할 수 있는 1chip 형태의 마이크로 컴퓨터라 할 수 있다. 이러한 D. S. P. chip의 특징은 고속의 산술 연산 처리를 위하여 승산기를 내장하였으며, 승산기와 누산기에서 pipeline 구조를 채택하였고, Harvard-type structure라 불리는 병렬 처리 기법을 채택한 점 등이라 하겠다.

대표적인 D. S. P. 소자의 한 예로 TI(Texas Instrument)사의 TMS320C25를 다음 그림 1에 나타내었다. 이 TMS320C25는 TMS32020의 pin-for-pin 및 object-code 레벨에서 compatible하다. 그러나 CMOS technology를 이용하여 speed를 두배로 향상시켰고 또 24개의 새로운 추가명령, 8개의 보조 레



(a) 구성도



(b) CALU 구성도

그림 1. TMS320C25의 구성도

지스터, 8 레벨의 하드웨어 stack, on-chip ROM 등 여러가지를 추가시킨 version이다.

먼저 그림 1 (a)는 TMS320C25의 개략적인 구조도이다.

그림에서 256word의 데이터/프로그램 메모리는 명령에 의해 데이터메모리나 프로그램메모리 어느쪽으로 사용할 수 있는 메모리로, 액세스 타임이 느린 외부 프로그램 메모리에 저장된 프로그램을 메모리 이동 명령으로 이 메모리로 옮겨와 실행시킴으로써 대기 시간없이 프로그램을 고속으로 실행시킬 수 있어 반복처리되는 프로그램인 경우 이용하면 실행 속도를 증가시킬 수 있다. 데이터 메모리와 프로그램 메모리는 내장된 메모리를 포함하여 64Kword까지 간단하게 확장시킬 수 있다.

그림 1 (b)의 CALU(central arithmetic logic unit)는 ALU, ACC Multiplier SFL(shifter)등으로 구성되어 데이터의 산술연산, logic 연산을 행한다. 데이터 버스로 부터 fetch된 데이터는 SFL 또는 승산기를 거쳐 ALU에 입력되고 ALU에 의해 처리된 결과가 ACC에 저장된다. 32bit ALU는 1machine cycle로 동작되며 산술연산 뿐만 아니라 boolean operation에 의해 제어를 위한 비트 조작이 가능하고 SFL은 소숫점 계산등 scaling에 이용된다. 16x16bit 하드웨어 승산기는 1machine cycle로 승산을 완료하는

고속 승산기로 T 레지스터에 승수를, 피승수는 데이터 버스에서 fetch해 연산 결과를 P 레지스터에 저장하고 SFL에 의해 소숫점 연산을 행한다. SFL은 status 레지스터의 PM을 결정한다. 또한 산술연산이 가능한 보조 레지스터가 있으며 FFT를 효율적으로 처리할 수 있는 bit-reversed indexed addressing mode 기능이 있다. 이외에 주변회로와 interface를 위한 각종 제어 신호, 8개의 interrupt, serial port, multiprocessing과 DMA를 위한 동기 신호등으로 구성되어 있다.

이와같이 D. S. P.는 제어분야에서 요구하는 메모리 내장, 명령어의 종류, 사용하기 쉬운 것등의 범용성을 갖추고 있으며 특히 여러개의 sampling 주기를 처리할 수 있는 외부 interrupt 기능과 주변회로와의 interface 기능을 갖게됨으로써 마이크로프로세서로는 실시간 처리가 불가능한 현대의 알고리즘을 실현시킬 수 있게 되었다.

또한 D. S. P.의 제어분야에 대한 응용의 확대로 제어용 D. S. P.가 등장하고 있다. 제어용 D. S. P.의 한 예로 TI사의 TMS320C14/E14를 들 수 있으며 이 D. S. P.는 제 1 세대 D. S. P.인 TMS32010을 포함하고 내장 메모리가 강화되어 있으며 또한 제어용 주변회로를 chip상에 내장한 D. S. P. 마이크로 콘트롤러로, 시스템 구성시 부품수가 감소되어 소형화 및

경제성을 향상시킬 수 있게 되어 있다.

D.S.P.의 제어응용 예로는 N.C.공작기계, 서보 모터의 속도추종, A.C. motor의 벡터제어, 로봇트 경로제어, hard disk driver의 head 위치 제어 등이 보고되어 있다.

D.S.P.의 실질적인 제어알고리즘의 수행속도를 범용 CPU와 비교하기 위해 그 한 예로, D.C 모터의 속도제어를 PID 제어방식으로 구현했을 때의 사용소자에 따른 수행 시간을 그림 2에 나타내었다.

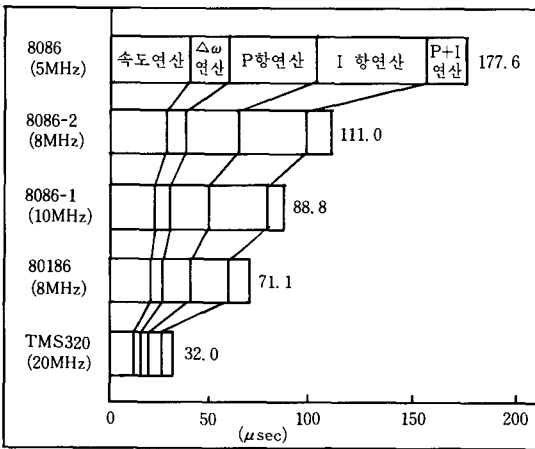


그림 2. D.C. 모터의 속도제어계의 처리시간 비교

D.S.P.에 있어서 제어 알고리즘 개발순서는 그림 3과 같이 C, Fortran, Assembly language등으로 source program을 개발하고 simulator, SWDS (software development system) 등으로 debug를 행하게 된다. SWDS 경우 1 step씩 실행이 가능해 각 step에 따른 레지스터, 메모리 내용을 검토할 수 있으며, test 또는 debug시에 이용할 I/O ports를 지정할 수 있게 된다.

Simulation에 의해 이상이 없으면 in-circuit emulator를 이용해 target board의 특성을 시험 할 수 있다. 즉 내부상태나 외부 상태에 따른 break point를 설정하고 이때 레지스터, 메모리 위치를 검색하고 외부상태를 점검하므로써 target board의 정상동작을 확인할 수 있게 된다. Target board가 정상적으로 동작하면 완성된 프로그램을 ROM에 programming함으로써 완료된다.

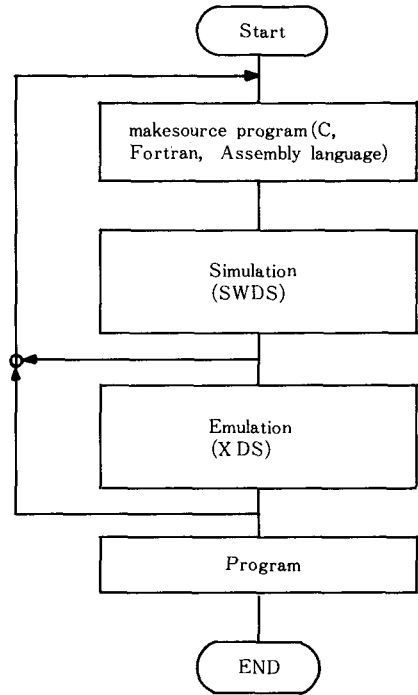


그림 3. D.S.P.의 소프트웨어 개발순서

### III. E.M.S 시스템과 적응 제어

#### 1. E.M.S 시스템 개요

많이 보고되지 않는 D.S.P.응용사례로서 흡인식 자기부상 시스템에의 적응제어 응용을 검토하여 본다.

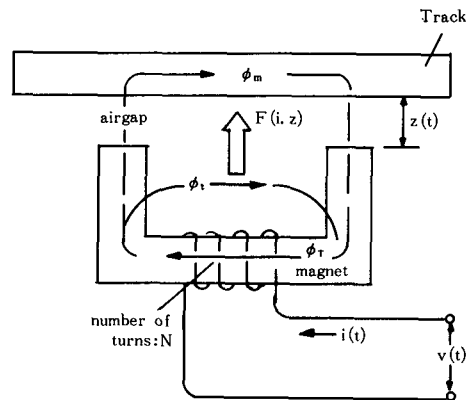


그림 4. E.M.S. 시스템의 구조

그림 4는 E.M.S. 시스템의 구조를 보여 주고 있다. 여기서의 문제는  $V(t)$ 의 제어에 의하여  $i(t)$ 가 제어되고, 이  $i(t)$ 에 의하여 발생된 흡인력  $F(i, z, t)$ 는 magnet가 고정된 track과 하방에서 일정거리(통상 10[mm] 정도)를 유지하도록 부상시키는 feedback 제어기를 설계하는 것이다. 이 시스템은 흡인력이 거리의 제곱에 반비례하고 전류의 제곱에 비례하므로 어떤 요인에 의해 주어진 평형점에서의 변화는 magnet를 track에 붙게 하거나 밑으로 떨어지는 open-loop unstable system임을 알 수 있다. 또한 거리의 제곱과 전류의 제곱에 반비례 및 비례하는 특성은 이 시스템을 비선형 시스템으로 하는 요소가 되며 일반적인 선형 시스템 제어 이론으로는 충분한 제어 효과를 기대할 수 없을 것이다.

그러나 기본적인 선형 모형의 유도를 위하여 주어진 operating point에서의 수식 전개를 하면 식(1)과 같다.

$$\begin{bmatrix} \Delta \dot{z}(t) \\ \Delta \ddot{z}(t) \\ \Delta \dot{i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{K_z}{m} & 0 & -\frac{K_i}{m} \\ 0 & \frac{K_z}{K_i} & -\frac{R}{L_o} \end{bmatrix} \begin{bmatrix} \Delta z(t) \\ \Delta \dot{z}(t) \\ \Delta i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_o} \end{bmatrix} \Delta v(t) \quad (1)$$

위 상태 방정식을 전달함수로 바꾸면

$$\Delta Z(s) = \left[ \frac{-K_i/mR}{L_o s^3 + s^2 - \frac{K_z}{m}} \right] \Delta V(s) \quad (2)$$

인데 계수  $K_i$ 는 open-loop 이득을 좌우하고  $K_z$ 는 극점과 관계있어 안정도와 직결된다. 전기적 시정수  $L_o/R \ll 1$  일때 open-loop 특성 방정식은

$$\left[ 1 + \frac{L_o}{R} s \right] \left[ s^2 + \frac{K_i}{mR} s - \frac{K_z}{m} \right] = 0 \quad (3)$$

이고 open-loop 극점은

$$S1 = -\frac{R}{L_o} \quad (4)$$

$$S2, 3 = -\frac{LoKz}{2mR} \pm \sqrt{\left[ \frac{LoKz}{2mR} \right]^2 + \frac{Kz}{m}}$$

이 되어 open-loop unstable 시스템이 되는데 이는 진상보상기를 달아 페루프 극점을 s평면의 좌반면에 위치시켜야 한다.

여기서  $K_i, K_z$ 는 식(5)와 같이 주어진다.

$$K_z = \frac{\partial F(i, z)}{\partial z} \Big|_{(i_o, z_o)} \quad K_i = \frac{\partial F(i, z)}{\partial i} \Big|_{(i_o, z_o)} \quad (5)$$

즉  $K_i$ 와  $K_z$ 는  $i$ 와  $z$ 의 함수로써 비선형 특성을 가지며 운전 중의 magnet 온도 변화에 의한 저항 변화 등도 E.M.S. 시스템의 매개 변수를 변화시키는 요인이 된다.

## 2. 적응제어 이론

적응제어의 기본 목적은 파라미터의 변화에 자동으로 제어 이득을 변환시킴으로써 양호한 제어 특성을 얻고자 하는 것이며 크게 기준모델 적응제어 방식과 자기동조 제어방식으로 나뉜다. 기준모델 적응제어 방식도 세분 가능하지만 자기동조 제어방식은 그 발전과정과 연대로서 더욱 확실히 구분된다 하겠다. 최소분산 적응제어 방식(MVC)에서부터 일반화된 최소분산 적응제어 방식(GMV), 극배치 적응제어 방식(APAC)<sup>[5,6]</sup>, 일반화된 예측제어 방식(GPC)<sup>[7]</sup> 등의 순으로 발전되어 왔다. 각각의 방식은 후에 제안된 것일수록 강인성이 큰 반면 연산량이 많이 소요되는 특징이 있다. 특히 MRAC나 MV 방식은 비최소 위상시스템의 제어에는 사용할 수 없으며 GMV도 극히 제한된 범위에서만 그 사용이 가능하다. 또한 GMV는 안정도 문제를 해결하기 위한 방법으로 detuning control 기법을 사용하므로 output bias가 생기는 등의 문제가 뒤 따른다.

시뮬레이션 상으로 가장 뛰어난 특성을 보이고 있는 GPC 방식은 이론적인 특성의 검토나 user chosen value의 선정에 관한 일반적인 합의는 이루어지지 못한 듯하며, 연산량의 부담은 상당히 크므로 여기서는 극배치 적응제어방식을 검토하기로 한다.

적응제어방식은 간접제어방식을 선택함으로써 식별과정과 제어과정을 분리시키고 각각의 연산량을 검토한다. 이는 어떤 경우에는 다중 식별 알고리즘을 처리하여 잘 선택된 파라미터로 제어를 구성하는 형태의 구현시에도 효과적으로 쓰일 수 있게 하기 위해서이다.

식별 알고리즘은 기본적으로 순환 최소자승법(R.L.S.E)와 사영법(projection)이 많이 쓰이나 여기서는 비교적 연산량이 많으나 뛰어난 수렴 특성을 보이는 R. L. S. E 방식을 기본으로 취한다.

이 경우 sampled plant model은 식(6)과 같다.

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) - a_3 y(k-3) + b_0 u(k-d) + b_1 u(k-d-1) + b_2 u(k-d-2) \quad (6)$$

여기서 다음과 같이 놓으면

$$\phi(k-1) = [-y(k-1) -y(k-2) -y(k-3) \ u(k-d) \ u(k-d-1) \ u(k-d-2)]$$

$$\theta^T = [a_1 \ a_2 \ a_3 \ b_0 \ b_1 \ b_2]$$

식(7)과 같이 간단히 표시된다.

$$y(k) = \phi^T(k-1) \theta \quad (7)$$

이 경우의 R. L. S. E. 알고리즘은 식(8), (9)와 같다.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-2) \phi(t-1)}{1 + \phi(t-1)^T P(t-2) \phi(t-1)} [y(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (8)$$

$$P(t-1) = P(t-2) -$$

$$\frac{P(t-2) \phi(t-1) \phi(t-1)^T P(t-2)}{1 + \phi(t-1)^T P(t-2) \phi(t-1)} \quad (9)$$

식(6), (7)에서의  $a_i, b_i$ 에 대하여

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}$$

$$B(q^{-1}) = q^{-d}(b_0 + b_1 q^{-1} + b_2 q^{-2})$$

와 같이 정의하면 제어 방식은

$$A(q^{-1}) L(q^{-1}) + B(q^{-1}) P(q^{-1}) = A^*(q^{-1}) \quad (10)$$

$$L(q^{-1}) u(t) = -P(q^{-1}) y(t) + M(q^{-1}) y^*(t+d) \quad (11)$$

와 같이 주어진다. 여기서

$$L(q^{-1}) = l_0 + l_1 q^{-1} + l_2 q^{-2}$$

$$P(q^{-1}) = p_0 + p_1 q^{-1} + p_2 q^{-2} \text{이고}$$

$A^*(q^{-1})$ 는 desired closed-loop pole 다항식이며  $y^*(t+d)$ 는  $(t+d)$  순간의 desired output이다.

식(10)을 푸는 과정은

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ p_0 \\ p_1 \\ p_2 \end{bmatrix} = Me^{-1} \begin{bmatrix} A^* \end{bmatrix}$$

이고  $A^*$ 는  $6 \times 1$  vector이고  $Me^{-1}$ 는  $6 \times 6$  matrix이다.

상기 알고리즘으로부터 계수 식별과정은 개략적으로 곱셈 126회, 덧셈 84회 정도가 소요된다. 제어법칙 과정에서는  $Me^{-1}$ 에서 주로 연산시간이 소요되며 식(11)의 완전한 구현을 위해서는 곱셈이 100회 정도이고 덧셈도 비슷한 정도가 소요된다. 즉 곱셈이 300회 이내, 덧셈이 300회 이내이고 이는 3세대 D. S. P. chip에서 32-bit floating point 곱셈이 적어도 200 nsec 이내에 이루어지고 덧셈은 60nsec 이내에 이루어지므로 적어도 12KHz 이상의 sampling frequency가 가능해진다. 모터 제어에 제 1세대 D. S. P. 소자를 사용하여 MV 제어방식으로 좋은 결과를 얻은 예<sup>[10]</sup>가 보고되었다. 이 경우에는 400  $\mu$ sec의 샘플링 time으로 처리하였으며 fixed point operation이었다. 그러나 현재는 같은 수준에서 floating point 연산이 가능해지며 충분히 좋은 효과를 거둘 수 있을 것으로 예상된다.

#### IV. D. S. P.를 이용한 E. M. S. 제어 시스템의 구현<sup>[8, 9]</sup>

3장 1절에서 설명한 E. M. S. 시스템에 대한 제어 시스템의 구성을 다음 그림 5에 나타내었다.

그림 5에서 E. M. S. 시스템부는 power amp와 제어대상인 magnet로 구성되며 점선으로 표시된 부분은 D. S. P. chip인 TMS320C25를 이용한 digital controller로 구성된다. 실제 제어시스템의 하드웨어 구성 예를 다음 그림 6에 표시하였다.

그림에서 알 수 있는 바와 같이 magnet에 가해지는 전압은 D. C chopper를 통해 제어된다. 그런데 이 전압의 제어는 gap sensor와 A/D 변환기를 통해 feedback된 gap 신호와 제어입력인  $u(t)$ 에 의해 먼저 parameter identification을 행한 후 수행된다. 실제 시스템에서의 gap output은 gap sensor에 의해 전압치로 나타나며 제어입력  $u(t)$ 는 PWM 형식인 chopper를 구동하기 위한 신호로서 3KHz-10KHz 정도인 chopping frequency를 가지게 된다.

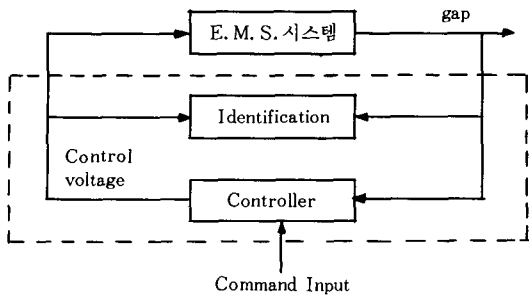


그림 5. E. M. S. 제어 시스템의 구성

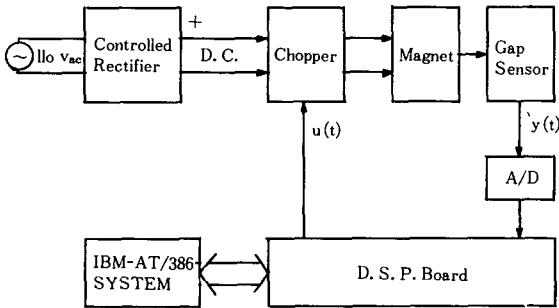


그림 6. E. M. S. 제어시스템의 하드웨어 구성 예

이 시스템은 현재 당 연구실에서 수행 중인 “자기 부상 및 추진시스템 개발 연구”중의 일부분이다.

### V. 결 론

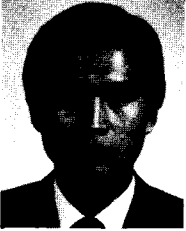
본고에서는 D. S. P. 소자를 이용한 현대제어 이론의 실제 시스템에서의 구현을 간략히 소개하였다. 기존의 범용 microprocessor를 사용했을 경우에 비해 D. S. P. chip을 사용함으로써 실행시간의 감소, 경비 절감 및 개발 시간의 단축 등의 이점을 얻을 수 있

다. 또한 기존의 100 nsec 이상의 cycle time의 D. S. P.로부터 이제 35nsec이하의 cycle time을 갖는 D. S. P.가 개발되고 있고 아울러 제어용 D. S. P.가 출현함으로써 현대제어 이론의 실용성의 관점에서 크게 고무적이며 따라서 현대제어의 실제 적용에 대한 각종 기술 개발의 급속한 확산이 예상된다.

### 參 考 文 獻

- [1] Second Generation TMS320 User's Guide, TI, 1987.
- [2] Third Generation TMS320 User's Guide, TI, 1988.
- [3] P.K. Sinha, *Electromagnetic Suspension Dynamics and Control*, Peter Peregrinus Ltd, United Kingdom, 1987.
- [4] B.V. Jayawant, *Electromagnetic Levitation and Suspension Techniques*, Edward Arnold Ltd, 1981.
- [5] Goodwin and Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, 1984.
- [6] K.J. Astrom, B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989.
- [7] D.W. Clarke, C. Mohtadi, "Generalized Predictive Control, Part 1, 2", *Automatica* 23, pp. 137-160, 1987.
- [8] M.B. Scudiere et al, "Digital Controller for a Magnetic Suspension System," *Rev. Sci. Instrum.* 57(8), 1986.
- [9] 崔田雅信, "DSP の 制御分野への応用", vol. 30, no. 11, *情報処理*, 1989.
- [10] K.H. Gurubasavaraj, "Implementation of a Self-Tuning Controller Using Digital Signal Processor Chips," *IEEE Control System Mag.* June 38-42, 1989.
- [11] 김국현, 권순만, 김석주, 김종문, "부상제어 시스템의 기본 원리와 응용", 대한전기학회 1990년도 춘계 Symposium 논문집. ㉔

筆者紹介



金 國 憲

1957年 3月 28日生  
1979年 2月 서울공대 전기공학과  
졸업  
1983年 8月 동대학 공학석사  
1987年 2月 동대학 공학박사

1987年 3月~1988年 12月 영국 Oxford대학 Self  
Tuning Control Group 박사후 연구원  
1989年 4月~1989年 11月 한국전기연구소 선임연구원  
1989年 12月~현재 한국전기연구소 제어응용 연구실  
실장

權 淳 萬

1958年 1月 26日生  
1982年 2月 부산대 전기공학과 졸업  
1984年 2月 동대학 공학석사  
1984年 3月~1989年 8月 한국전기연구소 연구원  
1989年 9月~현재 한국전기연구소 제어응용연구실  
선임연구원

金 鍾 洙

1959年 12月 26日生  
1982年 2月 서울대 공대 전기과 졸업  
1982年 3月~1989年 8月 한국전기연구소 연구원  
1989年 9月~현재 한국전기연구소 선임연구원