

예측제어기법을 이용한 PID 제어기 설계

The PID Controller for Predictive control Algorithm

김 양 환, 이 정 재, 이 정 용, 이 장 명*

(Yang-Hwan Kim, Jung-Jae Lee, Jung-Yong Lee, and Jang-Myung Lee)

Abstract : This paper is concerned with the design of a predictive PID controller which has similar features to the model-based predictive controller. A PID type control structure is defined, which includes prediction of the outputs and the recalculation of new set points using the future set point data. The optimal values of the PID gains are precalculated using the values of gains calculated from an unconstrained generalized predictive control algorithm. Simulation studies demonstrate the performance of the proposed controller and the results are compared with the conventional PID and fuzzy control algorithms.

Keywords : predictive PID controller, MPC(model predictive controller)

I. 서론

선형과 비선형 제어시스템을 통틀어서 가장 유용하게 사용되고 있는 제어 알고리즘중의 하나는 Proportional-Integral-Derivative(PID)제어기이다. 그 이유는 제어능력이 우수하고 또한 제어이득의 조정이 비교적 쉽기 때문이다. 그러므로 그 성능을 최대한 발휘시키는 것에도 큰 의미를 찾을 수 있는데, PID 파라미터의 튜닝을 위해서는 이미 많은 방법들이 제안되어 왔다.

일반적인 PID 파라미터 튜닝은 Ziegler-Nichols와 Cohen-Coon이 제안한 방법을 들 수 있다[1,2]. Astrom-Hagglund은 PID 제어기의 자동 튜닝을 위한 중계 피드백 기술을 개발하였다[3]. 중계를 기반으로 한 자동 튜닝방법은 그 간단성과 명료성으로 상업적인 제어기로 생산되었고, 많은 제어분야에서 성공을 거두었다[4]. 또 다른 방법으로서의 General Predictive Controller (GPC)은 Clarke가 제안하였고, 이것은 산업계와 학계에서 가장 유명한 것 중의 하나인 Model Predictive Controller(MPC)가 되었다[5,6]. GPC는 분석적인 해결 방법을 제시하고, 플랜트가 최소위상계가 아니고 불안정한 상태에서도 다룰 수 있는데, MPC는 GPC를 가역함수의 증가를 제어하기 위한 방법으로 영역제어의 개념을 최적 방법으로 포함시켰다. PID제어기의 폭넓은 활용으로 많은 연구자들이 최적 제어나 GPC 같은 발전된 제어기법을 사용하여 PID제어기의 구조를 구현하는 것을 시도하였다. Rivera는 1차 시스템에서 IMC를 기반으로 한 PID 제어기를 소개하였다[7]. Chien은 IMC-PID제어기를 디자인하여 2차 시스템에서 구현가능 하도록 확장하였다[8]. Morari-Zafriou는 산업에서 일반적으로 사용할 수 있는 IMC-PID제어기를 보였다[9]. Wang은 최소자승법을 사용하여 IMC 설계에 폐구간의 PID 제어기를 계산하였고 주파수 응답에 적용하였다[10]. 그러나 그 설계는 시간

지연과 불안정한 시스템에서는 효과가 없었다. Marques-Fliess는 출력에 기반한 선형 연속적인 PID 제어를 위한 간단한 접근법을 개발하였다[11]. MPC 방법의 중요한 점은 예측된 궤도에 기반한 플랫을 고려하여 PID제어를 한다는 것이다. Rusnak는 선형이차식(LQR)이론으로 공식화 하여 PID 제어기의 문제 해결을 보였다[12]. 이러한 공식화는 PID제어기의 좋은 영향에 대한 설명을 시스템적으로 접근하는 것을 회피하게 되었다. 또한 Rusnak는 일반적인 PID구조를 선보이며 5차 시스템에 적용하였다[13]. Tan은 GPC적으로 접근하는 것을 기반으로 PID 제어기를 설계하였다. 이것은 실시간으로 작동하여 적용 가능하였다[14,15].

본 논문에서는 일반적인 접근에 기반으로 하는 새로운 예측PID 제어방법을 구현하고자 한다. PID 제어기 설계의 목적은 저차, 고차 그리고 비 최소위상을 가진 시스템에서 사용이 가능해야 한다는 것이다. 기존의 예측 제어를 사용한 연구에서는 이론적인 면을 치우치고 실제 실험부분을 간과하는 경향이 있었다. 그렇기 때문에 본 논문의 목적은 새로운 PID 제어 설계로서 시간지연을 포함한 실제 불안정한 시스템에서도 적용 가능할 수 있는 제어기를 설계 한다.

본 논문에서 제시된 주된 아이디어는 GPC 제어 범칙으로부터 PID 파라미터의 환경을 계산하는 것에 기반을 둔다. 먼저 PID 파라미터의 초기값은 일반적인 튜닝방법인 Ziegler-Nichols방법으로 구하고 예측 시스템을 통하여 최적인 파라미터값을 구하는 것이다. 본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 예측 제어기에 관한 알고리즘을 정리하고, 제 3장에서는 예측 PID제어기의 이론적인 부분을 정리하여 설계 한다. 그리고 나서 제 4장에서 설계된 제어기를 실제 모터에 적용 및 실험하여 결과를 고찰하고, 제 5장에서 결론을 맺는다.

II. 예측 제어기

1. 예측 제어의 개념

예측제어의 기본적인 구조는 그림 1과 같이 나타낼 수 있다. 그림 1은 모델 예측제어의 구조를 표시한 것이다. 모델에는 전달함수가 사용되는 경우와 FIR동정으로 얻은 시계열계수형식이 사용되는 경우가 있다. 그림 1과 비교하여 본다면

* 책임저자(Corresponding Author)

논문접수 : 2004. 3. 3., 채택확정 : 2004. 7. 27.

김양환, 이정재, 이정용, 이장명 : 부산대학교 전자공학부

(hedang@pusan.ac.kr/swear21@pusan.ac.kr/leejyoung21@hanmail.net/jmlee@pusan.ac.kr)

※ 본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과로 수행되었습니다.

PID제어는 모델 예측제어의 간략한 형식이라고 할 수 있지만 엄밀히 말하자면 모델 예측제어가 아닌 모델 제어라고 할 수 있다. 예측의 기능은 다음의 제어주기를 보는 것뿐이고 장래의 움직임을 예측하는 기능은 없기 때문이다.

즉 입력과 출력의 편차는 제어연산주기에서만 검출해 그것을 기반으로 제어한다. 이 경우 프로세스의 진정한 응답은 예측부분에 약간이라도 차가 있으면 고유 진동수에 의한 진동이 발생해 서서히 정정하는 제어 형식으로 된다. 하지만 모델만 실제의 프로세스 응답과 같게 할 수 있으면 모델과 실제 프로세스와의 응답 차에 기인하는 고유진동은 발생하지 않는다. 그러나 사실상 프로세스의 스텝 응답을 동정해 모델화해도 약간의 차이는 생긴다. 즉 모델은 변화하지 않아도 프로세스의 응답이 그 부하나 원료성분의 변화 등으로 변화하는 것이다. 따라서 양자의 차에 의한 진동은 반드시 생긴다고 할 수 있다. 이러한 오차나 진동을 근본적으로 해소하기 위해서는 예측이라고 하는 기능이 필요하다. 즉 편차를 제어연산주기에서의 편차가 아니라 장래의 어떤 점에서 설정과의 차를 모델로 예측해 거기에 적합한 조작출력을 계산하는 방법이다. 예측제어의 알고리즘을 나타내면 그림 2와 같다. 시간 k 에서 주어진 목적함수의 최적화 문제를 풀어 얻은 해, 즉 현재와 미래의 제어입력을 (1)로 나타낼 수 있다.

$$u(k), u(k+1), \dots, u(k+M-1) \quad (1)$$

그러면 (1)에서 단지 첫번째 입력, 즉 현재 제어입력 $u(k)$ 만을 시간 $[k, k+1]$ 에서 플랜트를 제어하는데 사용한다. 다음 시점 $k+1$ 에서는 제어구간과 예측구간이 한 시점 앞으로 이동하게 되고 제어입력 $u(k)$ 와 시스템의 출력 $y(k+1)$ 로 부터 예측된 초기조건으로 하는 새로운 최적화 문제로부터 제어입력을 구한다.

2. 예측제어의 제어방법

예측 제어에서의 제어방법은 일반적으로 이동 구간 제어 (Receding Horizon Control, RHC)를 사용한다.

그림 3. 과 같이 RHC는 먼저 시간 k 에서 $[k, k+N]$ 구간을 고려하여 최적 N 개의 제어입력인 $u(k|k), u(k+1|k), u(k+2|k), \dots, u(k+N-1|k)$ 를 계산한다. 그 후 $u(k|k)$ 를 프로세스에 사용하여 예측한다. 그 다음으로 시간 $k+1$ 에서 $[k+1, k+N+1]$ 구간을 고려하여 최적 N 개의 제어입력인 $u(k+1|k+1), u(k+2|k+1), \dots, u(k+N|k+1)$ 를 계산하고 또다시 $u(k+1|k+1)$ 를 프로세스에 사용한다.

즉 현재 제어입력인 $u(k|k)$ 만을 시간 k 에서 프로세스를 제어하는데 사용하고, 다음 시점인 $k+1$ 에서 제어구간과 예측구간이 한 점 앞으로 이동하고 제어입력인 $u(k|k)$ 와 시스템의 출력 $y(k+1)$ 로 부터 예측된 초기조건으로 하는 새로운 최적화 문제로부터 제어입력을 구하는 것이다.

그리고 이중에서 단지 첫 번째 제어입력 $u(k+1|k+1)$ 만을 시간 $k+1$ 에서 프로세스를 제어하는데 사용한다. 그 이후의 매 시점에 대해서도 이와 같은 과정을 반복하는 것으로

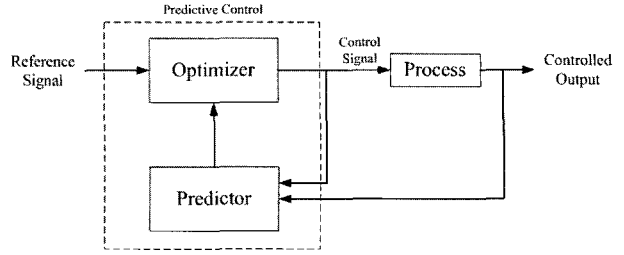


그림 1. 예측제어의 기본구조.
Fig. 1. Basic structure of predictive controller.

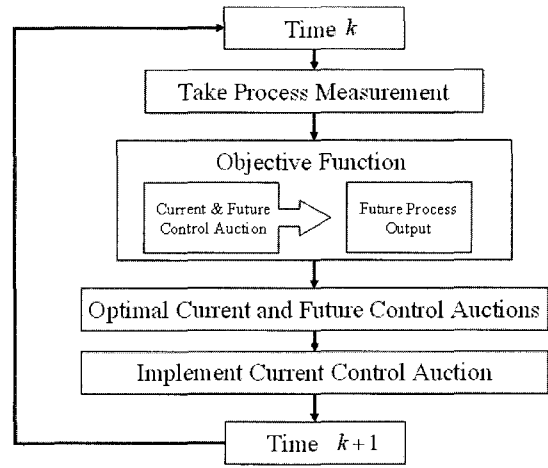


그림 2. 예측제어의 알고리즘.
Fig. 2. Algorithm of predictive control.

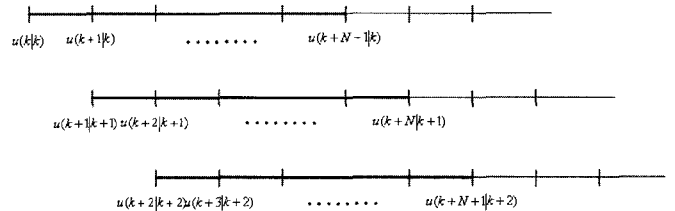


그림 3. 이동구간제어.
Fig. 3. Receding horizon control (RHC).

설정된 목적함수를 최소화하는 제어입력의 수는 설정된 목적함수에 따라서 달라진다.

III. 예측 PID제어기

1. 일반적인 형태의 PID

PID 제어기는 다음과 같이 표현한다.

$$\tilde{u}(k) = k_p e(k) + k_i \sum_{j=1}^k e(j) + k_d [e(k) - e(k-1)] \quad (2)$$

(2)의 양변에 $(k), (k+1)$ 의 스텝을 취하면

$$\begin{aligned} \Delta \tilde{u}(k) &= \tilde{u}(k) - \tilde{u}(k-1) \\ &= k_p [e(k) - e(k-1)] + k_i e(k) \\ &\quad + k_d [e(k) - 2e(k-1) + e(k-2)] \end{aligned} \quad (3)$$

와 같이 된다. (3)를 Z-domain에서 나타내면

$$\tilde{U}(z) = \frac{[q_0 + q_1 z^{-1} + q_2 z^{-2}]}{1 - z^{-1}} E(z) \quad (4)$$

여기서 : $q_0 = (k_p + k_I + k_D)$,
 $q_1 = -(k_p + 2k_D)$, $q_2 = k_D$

2. 예측 PID 제어기

예측 PID 제어기는 (5)와 같이 정의 된다.

$$u(k) = \sum_{i=0}^M \left(k_p e(k+i) + k_I \sum_{j=1}^k e(j+i) \right) + \sum_{i=0}^M (k_D [e(k+i) - e(k+i-1)]) \quad (5)$$

제어기는 그림 4에서 같이 예측횟수M에 대응되는 PID 제어기로 구성된다. M=0 일 때 제어기는 이상적인 보통의 PID 상태와 같다. M > 0이면 목적된 제어기는 PID 제어기의 M번의 예측을 수행하는 MPC와 유사하고 여기서 M은 GPC의 해를 구함으로써 최적의 값을 구할 수 있다. (2)을 이용하면 M시각 이후까지의 예측한 제어입력량은 다음과 같다.

$$u(k) = \tilde{u}(k) + \tilde{u}(k+1) + \dots + \tilde{u}(k+M) \quad (6)$$

(6)을 그림 4의 $\Delta u(k)$ 로 나타내면 다음과 같다.

$$\Delta u(k) = \Delta \tilde{u}(k) + \Delta \tilde{u}(k+1) + \dots + \Delta \tilde{u}(k+M) \quad (7)$$

여기서 (6)에서 임의의 시각에서의 제어 신호인 $\tilde{u}(k+i)$ 는 다음과 같다.

$$\tilde{u}(k+i) = k_p e(k+i) + k_I \sum_{j=1}^k e(j+i) + k_D [e(k+i) - e(k+i-1)] \quad (i=0, \dots, M) \quad (8)$$

여기서 $e(k) = w(k) - y(k+i)$ 을 대입하면

$$\tilde{u}(k+i) = k_p [w(k) - y(k+i)] + k_I \sum_{j=1}^k [w(k) - y(j+i)] + k_D [w(k) - y(k+i) - w(k) + y(k+i-1)] \quad (i=0, \dots, M) \quad (9)$$

여기서 $w(k) = 0$ 라 두면,

$$\tilde{u}(k+i) = k_p [-y(k+i)] + k_I \sum_{j=1}^k [-y(j+i)] + k_D [-y(k+i) + y(k+i-1)] \quad (i=0, \dots, M) \quad (10)$$

가 되므로 $\Delta \tilde{u}(k+i)$ 는 다음과 같다.

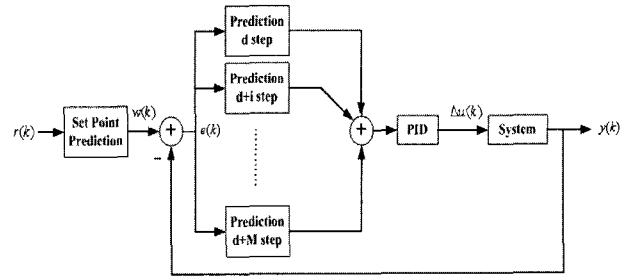


그림 4. 예측 PID 제어기의 기본 알고리즘.

Fig. 4. Basic Algorithm of a Predictive PID Controller.

$$\begin{aligned} \Delta \tilde{u}(k+i) &= \tilde{u}(k+i) - \tilde{u}(k+i-1) \\ &= k_p [y(k+i-1) - y(k+i)] + k_I \sum_{j=1}^k [y(j+i-1) - y(j+i)] \\ &\quad + k_D [2y(k+i-1) - y(k+i) - y(k-i-2)], \quad (i=0, \dots, M) \end{aligned} \quad (11)$$

(11)을 정리하면 다음과 같다

$$\Delta \tilde{u}(k+i) = -KY(k+i) \quad (12)$$

여기서 : $K = [k_D \quad -k_p - 2k_D \quad k_p + k_I + k_D]$

$$Y(k+i) = [y(k+i-2) \quad y(k+i-1) \quad y(k+i)]^T$$

(12)을 (7)에 대입하면

$$\begin{aligned} \Delta u(k) &= \Delta \tilde{u}(k) + \Delta \tilde{u}(k+1) + \dots + \Delta \tilde{u}(k+M) \\ &= -K\{Y(k) + Y(k+1) + \dots + Y(k+M)\} \end{aligned} \quad (13)$$

이것은 현재 제어신호는 예측된 미래의 출력의 선형조합이라는 것을 나타낸다. 따라서 i 번째 출력은 다음과 같은 방정식으로 나타낼 수 있다[16].

$$\begin{aligned} y(k+i) &= [g_{i-1} \quad g_{i-2} \quad \dots \quad g_0] \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \dots \\ \Delta u(k+i-1) \end{bmatrix} \\ &\quad + [f_{i1} \quad f_{i2} \quad \dots \quad f_{i(n_a+1)}] \begin{bmatrix} y(k) \\ y(k-1) \\ \dots \\ y(k-n_a) \end{bmatrix} \\ &\quad + [g'_{i1} \quad g'_{i2} \quad \dots \quad g'_{i(n_b)}] \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \dots \\ \Delta u(k-n_b) \end{bmatrix} \end{aligned} \quad (14)$$

(14)에서 미래의 제어입력 $\{\Delta u(k+i) \quad i=1:(N_u-1)\}$ 의 계산하는 것을 필요로 한다는 것을 주목해야 한다. 주어진 식을 다시 정리하면 다음과 같다.

$$Y(k+i) = G_i \hat{\Delta u}(k) + F_i y_0(k) + G'_i \Delta u_0(k) \quad (15)$$

여기서

$$G_i = \begin{bmatrix} g_{i-3} & \dots & g_0 & 0 & 0 \\ g_{i-2} & g_{i-3} & \dots & g_0 & 0 \\ g_{i-1} & g_{i-2} & g_{i-3} & \dots & g_0 \end{bmatrix}$$

$$F_i = \begin{bmatrix} f_{(i-2)1} & f_{(i-2)2} & \dots & f_{(i-2)(n_a+1)} \\ f_{(i-1)1} & f_{(i-1)2} & \dots & f_{(i-1)(n_a+1)} \\ f_{i1} & f_{i2} & \dots & f_{i(n_a+1)} \end{bmatrix}$$

$$G'_i = \begin{bmatrix} g_{(i-2)1} & g_{(i-2)2} & \dots & g_{(i-2)n_b} \\ g_{(i-1)1} & g_{(i-1)2} & \dots & g_{(i-1)n_b} \\ g_{i1} & g_{i2} & \dots & g_{in_b} \end{bmatrix}$$

$$\hat{\Delta u}(k) = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+N_u-1)]^T$$

$$\Delta u_0(k) = [\Delta u(k-1) \quad \Delta u(k-2) \quad \dots \quad \Delta u(k-n_b)]^T$$

$$y_0(k) = [y(k) \quad y(k-1) \quad \dots \quad y(k-n_a)]^T$$

$$Y(k+i) = [y(k+i-2) \quad y(k+i-1) \quad y(k+i)]^T$$

(15)에 (13)을 대입하면

$$\Delta u(k) = -K \sum_{i=0}^M Y(k+i)$$

$$= -K \left\{ \sum_{i=0}^M G_i \hat{\Delta u}(k) + \sum_{i=0}^M F_i y_0(k) + \sum_{i=0}^M G'_i \Delta u_0(k) \right\} \quad (16)$$

(15)를 PID 타입의 예측제어 형태로 다시 풀면

$$\Delta u(k) = -K \{ \alpha \hat{\Delta u}(k) + F_f y_0(k) + G_g \Delta u_0(k) \} \quad (17)$$

여기서

$$\alpha = \sum_{i=0}^M G_i \quad F_f = \sum_{i=0}^M F_i \quad G_g = \sum_{i=0}^M G'_i \quad (18)$$

3. 예측 PID 제어기 Gain의 최적값

이득의 최적값을 구하기 위해서는 GPC이론이 사용된다. 프로세서 제어를 하는동안 가격 범위 출력의 일반적 세팅은 $\{N_1 : N_2\} = \{1 : N\}$ 이고, 제어 가격 범위 $N_u = 1$ 는 GPC에서 사용할 수 있다. GPC는 다음의 가격 함수를 최소화 하는 제어 결과로 구성된다.

$$J(1, N) = \sum_{i=1}^N [y(k+d+i) - w(k+d+i)]^2 + \lambda \Delta u(k) \quad (19)$$

J(제어 신호에 어떠한 구속조건이 없다고 가정하면) 의 최소값은 다음 식과 같다[16].

$$\Delta u(k) = (G^T G + \lambda I)^{-1} G^T [w - F y_0(k) + G' \Delta u_0(k)] \quad (20)$$

미래 Set point $w(t+i) = 0$ 라고 가정하면

$$\Delta u(k) = -K_{GPC} [-F \quad G] \begin{bmatrix} y_0(k) \\ \Delta u_0(k) \end{bmatrix} = -K_0 \begin{bmatrix} y_0(k) \\ \Delta u_0(k) \end{bmatrix} \quad (21)$$

여기서

$$K_0 = -K_{GPC} [-F \quad G] \quad K_{GPC} = (G^T G + \lambda I)^{-1} G^T$$

$$y_0(k) = [y(k) \quad y(k-1) \quad \dots \quad y(k-n_a)]^T$$

$$\Delta u_0(k) = [\Delta u(k-1) \quad \Delta u(k-2) \quad \dots \quad \Delta u(k-n_b)]^T$$

$N_u = 1$ 일 때 예측 제어 PID 이득의 최적 값을 계산하면, PID 제어 신호는 GPC 제어기와 같이 될 것이다. 이것은 (17)을 이용하여 (21)을 풀면 다음과 같은 최적화 문제가 된다는 것을 의미한다.

$$\text{Min}_{K \in K_{PID}^s, M} J(K, K_0)$$

여기서

$$J(K, K_0) = \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] Z(k) - K_0 Z(k_0) \right\|_2$$

$K_{PID}^s =$ PID의 안정한 이득값

$Z = \begin{pmatrix} y_0(k) \\ \Delta u_0(k) \end{pmatrix}$ 는 제어 이득에 의존하므로 다음과 같다.

$$Z(k) = Z(k_0) + \Delta Z$$

$N_u = 1$ 을 (14)에 대입하면 최적화 문제는 다음과 같이 된다.

$$J(K, K_0) = \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] \{ Z(k_0 + \Delta Z) \} - K_0 Z(k_0) \right\|_2$$

$$\leq \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] - K_0 \right\|_2 \left\| Z(k_0) \right\|_2$$

$$+ \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] \right\|_2 \left\| \Delta Z \right\|_2$$

$$\leq \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] - K_0 \right\|_2 \left\| Z(k_0) \right\|_2$$

$$+ \left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] \right\|_2 \left\| \Delta Z \right\|_2$$

그러므로 정리하면 다음과 같다

① 최소 높음은 다음 식에서 구할 수 있다.

$$\left\| -(1+K\alpha)^{-1} K [F_f \quad G_g] - K_0 \right\|_2$$

이 식은 최소값이므로 다음과 같이 나타낼 수 있다.

$$-(1+K\alpha)^{-1} K [F_f \quad G_g] = K_0$$

② 적합한 K 를 K_0 에 근접시킬 수 있다고 가정하면 $\|\Delta Z\|_2$ 는 알맞게 작아진다.

K 에 관한 식은 K_0 의 식으로 바꿀 수 있다.

$$K_0 = -(1+K\alpha)^{-1} K [F_f \quad G_g] \rightarrow K_0 = -(1+K\alpha)^{-1} K S_0 \quad (22)$$

$$K_0(1+K\alpha) = K S_0 \rightarrow K(S_0 - \alpha K_0) = K_0 \quad (23)$$

여기서 $S_0 = [F_f \quad G_g]$

(22)의 해는 항상 존재하고 이를 K에 대해 다시 정리 하면 다음과 같은 식을 얻는다.

$$K = K_0(S_0 - \alpha K_0)^T [(S_0 - \alpha K_0)(S_0 - \alpha K_0)^T]^{-1} \quad (24)$$

이차 시스템에서 PID(M=0)의 차는 GPC의 동작으로 충분하다. 고차 시스템에서는 M이 값을 가지고 최적화된 값을 GPC를 풀어 선택한다.

4. Set Point 설정

GPC 알고리즘에서 Set point 의 N 영역에 대한 정보는 제어 결과를 계산해서 구한다. 새로운 Set point의 설정방법은 미래의 Set point에 대한 정보를 저장하여 발생시킨다. 새로운 Set point $r(k)$ 는 미래 Set point $w(k)$ 에서 계산되고 :

$$r(1) = \sum_{i=1}^N \frac{w(i)}{N} \quad (25)$$

$$r(i) = r(i-1) + w(i+N-1) - w(i-1) \quad i = 2, \dots, n$$

Set point의 재설정은 N 다음 단계에서 Set point의 평균이다. 설정을 하기 위해 사용하는 것은 GPC가 도달하는 Set point를 사용한다. Set point를 설정하기 위한 다른 방법은 :

$$r(k) = K_{GPC}W(k) \quad k = 1, \dots, n \quad (26)$$

여기서 : $W(k) = [w(k) \ w(k+1) \ \dots \ w(k+N)]$

K_{GPC} 는 (21)에서 계산된 GPC의 이득이고 $W(k)$ 는 시스템의 미래 Set point값이다.

5. 예측 PID 제어기의 알고리즘 구성단계

예측 PID제어 알고리즘은 다음과 같은 절차로 만들어 진다.

Step 1: 시작

- ① 시스템의 모델을 찾고 이산 다항식 A와 B를 계산
- ② 예측 범위, M 의 값을 고르고, 미래 set point 벡터인

$r(k)$ 를 공식화

Step 2: 오프라인에서 계산

- ① (15) 에서 행렬 α, F_f, G_g 를 설정
- ② (21)를 이용하여 GPC 이득 K_{GPC} 를 계산
- ③ (22)를 이용하여 예측 PID 이득의 최적값 계산
- ④ 가격 함수의 최소화를 위한 M의 값을 되풀이 하여 계산

Step 3: 온라인에서 계산

① 다음의 신호를 계산 $[F_f \ G_g]Z(k)$ (25)와 (26)에서 $R(k) = [r(k-2) \ r(k-1) \ r(k)]$

② 제어 증가신호를 계산

$$u(k) = u(k-1) + (I + K\alpha)^{-1} K [R(k) - [F_f \ G_g]Z(k)]$$

Step 4: 튜닝

- ① 제어 신호를 적용하여 페루프의 동작 확인
- ② 필요하다면 PID 이득을 좋게 튜닝

6. 안정도

Set point에서 식 (16)의 전달 방정식의 행렬은 :

$$\Delta u(k) = -K \{ \alpha \Delta u(k) + F_f \theta(k) + G_g \Delta U(k) \} \quad (27)$$

여기서:

$$G_i = \begin{bmatrix} g_{i-3} & \dots & g_0 & 0 & 0 \\ g_{i-2} & g_{i-3} & \dots & g_0 & 0 \\ g_{i-1} & g_{i-2} & g_{i-3} & \dots & g_0 \end{bmatrix}$$

$$F_i = \begin{bmatrix} f_{(i-2)_1} & f_{(i-2)_2} & \dots & f_{(i-2)(n_a+1)} \\ f_{(i-1)_1} & f_{(i-1)_2} & \dots & f_{(i-1)(n_a+1)} \\ f_{i1} & f_{i2} & \dots & f_{i(n_a+1)} \end{bmatrix}$$

$$G'_i = \begin{bmatrix} g'_{(i-2)_1} & g'_{(i-2)_2} & \dots & g'_{(i-2)n_b} \\ g'_{(i-1)_1} & g'_{(i-1)_2} & \dots & g'_{(i-1)n_b} \\ g'_{i1} & g'_{i2} & \dots & g'_{in_b} \end{bmatrix}$$

$$\theta(k) = [-e(k) \ -e(k-1) \ \dots \ -e(k-n_a)]^T$$

$$\Delta U(k) = [\Delta u(k-1) \ \Delta u(k-2) \ \dots \ \Delta u(k-n_b)]^T$$

$$\alpha = \sum_{i=0}^M G_i \quad F_f = \sum_{i=0}^M F_i \quad G_g = \sum_{i=0}^M G'_i$$

$$e(k+i) = w(k-i) - y(k-i) \quad i = (0, \dots, n_a)$$

제어 법칙을 간략화 하고 선형 방정식을 사용하면 :

$$A_c(z^{-1})y = w^* - B_c(z^{-1})\Delta u \quad (28)$$

여기서:

$$A_c(z^{-1}) = KF_f Z_y \quad Z_y = [1 \ z^{-1} \ \dots \ z^{-na}]^T$$

$$G' = [1 + K\alpha \ KG_g] \quad B_c(z^{-1}) = G'Z_u$$

$$Z_u = [1 \ z^{-1} \ \dots \ z^{-nb}]^T \quad w^* = k_{ref}w \quad k_{ref} = KF_f$$

CARIMA 모델은 다음과 같이 주어진다.

$$\tilde{A}(z^{-1})y(k) = z^{-d}B(z^{-1})\Delta u(k-1) = z^{-d-1}B(z^{-1})\Delta u(k) \quad (29)$$

$d \geq 1$ 일 때 시스템의 시간지연은

$$\tilde{A}(z^{-1}) = D(z^{-1})A(z^{-1}) = (1 - z^{-1})A(z^{-1})$$

(29)에 (28)을 대입하고 시스템의 페루프 전달함수는 다음과 같다.

$$y = \frac{B(z^{-1})z^{-d-1}}{A_c(z^{-1})B(z^{-1})z^{-d-1} + B_c(z^{-1})\tilde{A}(z^{-1})} w^* \quad (30)$$

여기서 페루프 극점은 특성 방정식의 근이된다.

$$A_c(z^{-1})B(z^{-1})z^{-d-1} + B_c(z^{-1})\tilde{A}(z^{-1}) = 0 \quad (31)$$

(27-31)에 의해서 페루프 시스템의 특성 방정식의 근이 PID 이득에 의존한다는 것을 보인다. 그리고 PID 이득은 주어진 M의 예측 범위의 선택에 영향을 받는다.

IV. 실험 및 결과

1. 시뮬레이션

먼저 그림 5 같이 예측 PID 제어기를 구성하였다. 본 논문

에서의 시뮬레이션은 예측 PID 제어기에 부가한 응답을 예측 횟수 M에 따라 어떠한 변화를 가지는지 상호 비교하는 것으로 시뮬레이션 하였다. 먼저 구성된 예측 PID 제어기의 성능을 측정해 보기 위해 시간 지연과 불안정한 시스템을 표현한 임의의 전달함수를 선택하였다.

$$G(z) = \frac{0.033z^{-4} - 0.0328z^{-3} - 0.0348z^{-2} + 0.0346z^{-1}}{0.0330z^{-4} - 0.9559z^{-3} + 2.8115z^{-2} - 2.8885z^{-1} + 1}$$

그림 6에서 볼 수 있듯이, M 값에 따라서 예측 PID의 출력은 여러 형태로 나온다. M이 증가할수록 오버슈트는 줄어들지만 계산량의 증가에 따른 시간 지연 또한 증가하여 시스템의 필요에 맞는 적절한 M의 선택이 필요하다. 이 시스템에서 PID의 파라미터의 첫번째 값은 일반적인 PID 조율 방법으로 사용되는 Zeigler와 Nichols 방법을 사용하였다[23].

2. 실험결과

본 연구에서 향후 적용하는 Throttle Valve제어를 위해 그림 7과 같이 실험 장치를 구성 하였다. 보드는 TMS320LF 2407을 사용하였고 DC모터와 이를 구동하기 위한 모터의 드라이버는 L 6203, 그리고 1000 Pulse의 엔코더로 구성하였다.

먼저 일반적인 모터의 수렴값을 확인 하여 시뮬레이션의 결과와 비교하였다. 그림 8은 수렴하기까지의 모터의 동작을 M값에 따라 나타내었다. 그림과 같이 M값이 점차 커짐에 따라 진동이 줄어들고 안정적인 수렴값을 나타내지만 예측 횟수의 증가에 따른 시간 지연현상이 나타나고 있다.

그림 9는 수렴도중 외란을 주었을 때 다시 안정화될 때까지의 값을 구하여 보았다. 앞선 실험과 마찬가지로 M값의

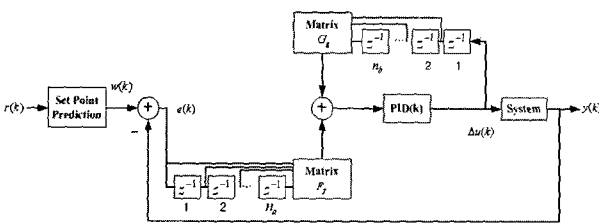


그림 5. 예측 PID 제어기의 구조.
Fig. 5. Structure of predictive PID controller.

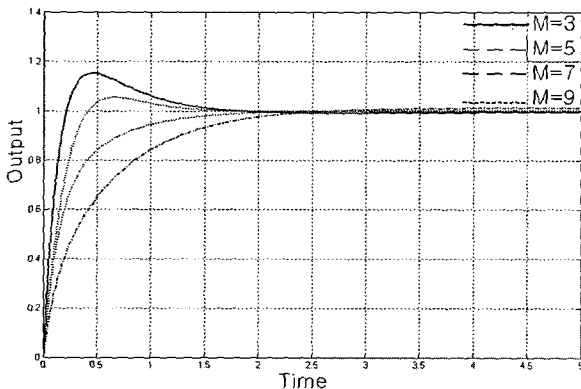


그림 6. 시뮬레이션 결과.
Fig. 6. Simulation results.

증가에 따른 파형의 변화를 볼 수 있다. M값이 증가함에 따라 시스템의 파형상에서 오버슈트는 거의 없어지나 제어기 자체의 시간지연현상이 증가하는 현상을 확인 할 수 있다.

그림 8과 그림 9에서 M값에 따른 여러 파형을 비교하였을 때 시스템의 시간지연현상과 오버슈트 현상을 고려하였을 때 M=5일 때 제어기의 성능이 가장 좋다고 할 수 있다. 제어기의 예측 회수는 모터의 특성에 따라 다르겠지만, 실험에서 사용한 모터의 특성은 M=5일 때 가장 좋은 성능을 보인다고 결정하고 4.3절과 같은 비교실험을 하였다.

3. 예측 PID제어기와 퍼지제어기와의 성능비교

예측 PID 제어기와 성능을 비교하기 위하여 일반적으로 사용되는 퍼지 제어기를 구성하여 실험 해 보았다.

표 2는 모터를 원하는 각도로 동작시키기 위한 오차 e와 오차의 변화량 ce에 대한 전체 25개의 퍼지 제어 규칙을

표 1. 예측 PID 게인.

Table 1. Predictive PID Gain.

M	Gain
PID(0)	[0.88 0.18 1.13]
3	[1.14 0.22 0.52]
5	[1.35 0.29 0.37]
7	[1.42 0.33 0.31]
9	[1.50 0.39 0.29]

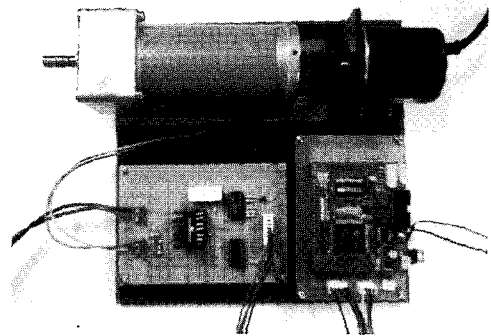


그림 7. 장비 구성.
Fig. 7. Hardware construction of controller.

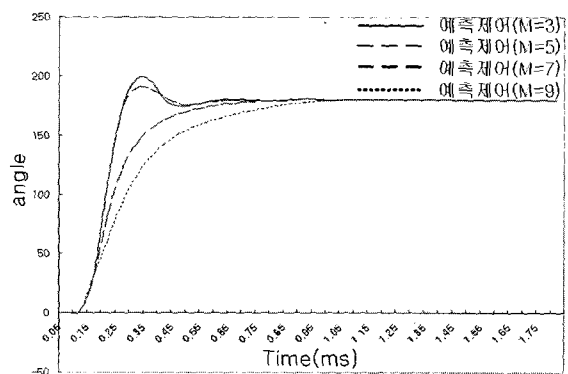


그림 8. M값에 따른 실험 결과치.
Fig. 8. Experiment results.

표 2. 퍼지 제어 규칙.

Table 2. Fuzzy control rules.

e \ ce	NB	NM	ZO	PM	PB
NB	NG	NB	NM	NS	ZO
NM	NB	NM	NS	ZO	PS
ZO	NM	NS	ZO	PS	PM
PM	NS	ZO	PS	PM	PB
PB	ZO	PS	PM	PB	PG

NG : Negative Great
 NM : Negative Medium
 ZO : Apporoximately Zero
 PM : Positive Medium
 PG : Positive Great

NB : Negative Big
 NS : Negative Small
 PS : Positive Small
 PB : Positive Big

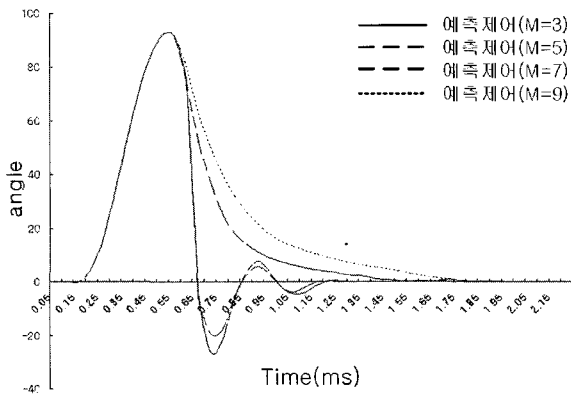


그림 9. M값에 따른 실험 결과치.

Fig. 9. Experiment results.

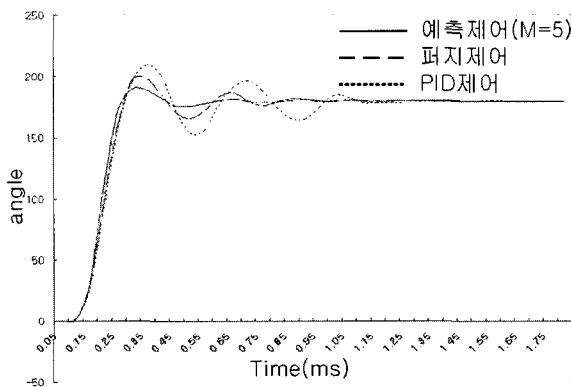


그림 10. 제어기별 실험 결과치.

Fig. 10. Experiment results.

나타낸다[18]. 먼저 PID 제어기, 퍼지제어기, 예측 PID 제어기의 일반적인 성능을 비교하였다. 그림 10에 세 제어기의 성능 비교를 나타내었다. 예측 제어기는 예측 횟수에 따라 성능의 차이를 보이지만, 4.2절의 실험에서 가장 좋은 성능을 보인 M=5를 비교하였는데 PID제어기와 퍼지 제어기에 비해 더 좋은 성능을 나타내었다. 퍼지제어기는 PID 제어기보다는 좋은 성능을 보이지만 오버슈트와 안정도 부분에서 예측 PID 제어기에 비해 성능이 떨어짐을 알 수 있다.

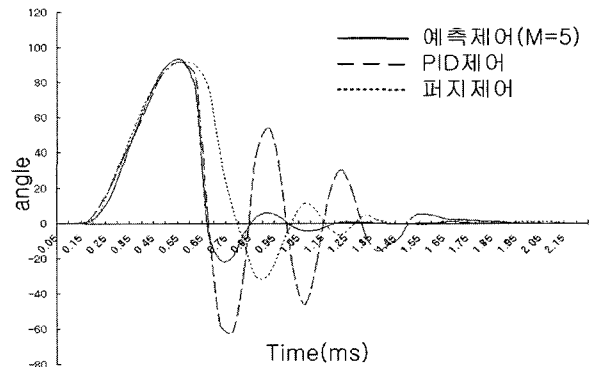


그림 11. 제어기별 실험 결과치.

Fig. 11. Experiment results.

그림 11에서는 외란의 강인성에 대한 실험을 하였다. 이 결과에서도 일반 PID제어기나 예측 제어기에 비해 예측 PID 제어기가 외란에도 강인함을 나타내고 있다. 일반 PID 제어기는 처음의 반응 속도는 예측 PID 제어기와 같이 빠르나 시스템 자체에 가지는 오버슈트 문제로 외란에 안정화 하기 까지의 시간이 필요하고, 퍼지제어기는 일반 PID 제어기에 비해 외란에 강인함을 보이지만 첫 반응 시간에서 PID 제어기보다 떨어짐을 알 수 있다. 이에 비해 예측 PID 제어기는 반응 시간면에서는 일반 PID 제어기와 같고 안정화 되는 시간도 다른 제어기에 비해 더 좋은 성능을 나타내고 있다.

V. 결론

우리는 일반적으로 알려져있는 PID 제어기를 예측 제어기를 사용하여 개선하는 방법을 제시하였다. 일반적인 형태의 PID 제어기는 예측 PID 제어기보다 몇가지 단점을 지닌다. 첫째, 프로세스 스텝 응답의 다양성에 대해 PID 제어기는 1 차 지연함수를 모델로서 제어하고 그 나머지는 피드백으로 제어되므로 진동이 생길수 밖에 없다. 둘째, 제어 연산의 기본이 되는 $u(k)$ 는 연산시에 얻은 편차만 보므로 피드백에 의한 수정 동작이 진동적으로 되며, 그 진동수는 PID의 고유값으로 된다. 이러한 단점들을 보완하여 예측 PID 제어기를 설계하여, 실험 결과 및 고찰을 통하여 다음과 같은 결론을 얻을 수 있었다.

예측 PID 제어기는 예측의 횟수에 따라 제어기의 성능이 달라지는데, 본 논문에서 제시한 제어기에서는 M=5에서 최대의 성능임을 알 수 있었고, M=7부터 시간 지연의 증가로 제어기의 성능이 오히려 떨어짐을 알 수 있었다. 이러한 결과는 시뮬레이션과 실험에서 같은 형태로 나타났는데, 실험에서는 하드웨어의 특성에 따라 다른 결과로 나타날 것이라고 생각된다. 또한 제어기의 성능측정을 위하여 퍼지제어기와 비교하였는데 M=5에서 더 좋은 성능을 보였다.

향후 개선해야 될 사항으로는 복잡한 예측 과정으로 인한 계산량의 증가를 어떻게 감소시킬수 있는 방향으로 좀더 연구가 진행 되어야 할 것이고, 원래 목적인 Throttle Valve 제어에 적용함으로써 차량에 가속 페달을 밟았을 때 Throttle Valve 까지 전해지는 시간 지연을 최소화 하여 보다 정확하고 빠른 제어를 할 수 있을 것이다.

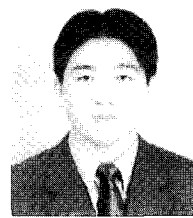
참고문헌

- [1] J. G. Ziegler, and N.B.Nichols, "Optimum setting for automatic controllers", *Trans. ASME* 64, 759-768.
- [2] G. H. Cohen and G. A. Coon, "Theoretical consideration of retarded control", *Trans. ASME* 75, 827-834, 1953.
- [3] K. J. Astrom. and T. Hagglund, "Automatic tuning of PID controllers", *Instrument Society of America, Research Triangle Park, NC*.
- [4] K. J. Astrom, and T. Hagglund, "PID controllers: theory, design and tuning", *Instrument Society of America, Research Triangle Park, NC, USA* 1995.
- [5] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalised predictive Control", *I & II. Automatica*, 23(2): 137- 160, 1987.
- [6] D. W. Clarke, and R. Scattoni, "Constrained receding-horizon predictive control", *IEE Proceedings Part D*, vol. 138, no. 4, July, pp. 347-354, 1991.
- [7] D. E. Rivera, S. Skogestad and M. Morari, "Internal model control 4. PID controller design", *Ind. Eng Chem. Proc. Des & Dev*, 25, pp. 252-265, 1986.
- [8] I. L. Chien, "IMC-PID controller design-An extension", *IFAC Proceeding Series*, 6, pp. 147-152, 1988.
- [9] M. Morari, and E. Zafiriou, "Robust process control", Prentice Hall, 1989.
- [10] Q. G. Wang, C. C. Hang and X. P. Yang, "Single -loop controller design via IMC principles", *In Proceeding Asian Control Conf. Shanghai*, P. R. China, 2000.
- [11] K. J. Astrom T. Hagglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers- a survey", *Control Engineering Practice*, vol. 1, pp. 699-714, 1993.
- [12] R. Gorez, and G. Calcev, "A survey of PID auto-tuning methods", *Proc. 11th Conference on Control Systems and Computer Science*, vol. 1. pp. 18-27, 1997.
- [13] R. Marques, and M. Fliess, "From PID to model predictive control a flatness-based approach", *IFAC Workshop PID'00*, Spain pp 534-539, 2000.
- [14] I. Rusnak, "Generalized PID controllers", *the 7th IEEE mediterranean conference on controller & automation*. MED 99 Haifa, Israel, 1999.
- [15] I. Rusnak, "The generalized PID controllers and its application to control of ultrasonic and electric motors", *IFAC Workshop PID'00*, Spain pp. 125-130, 2000.
- [16] K. K. Tan, T. H. Lee, S. N. Huang and F. M. Leu, "PID control design based on a GPC approach", *Personal communication*, 2000.
- [17] K. K. Tan, Q-G. Wang and C. C. Hang, "Advances in PID control", Springer Verlag London, ISBN-1-852333-138-0, 1999.
- [18] E. F. Camacho, and C. Bordons, "Model predictive control", Springer-Verlag London, 1999.
- [19] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized predictive control", *I&II. Automatica*, 23(2):137-160, 1987.
- [20] K. M.-S, "A Study on the control of SMA actuator for smart catheter", Dong-A University Pusan Korea.
- [21] M. R. Katebi, M. H. Moradi and M. A. Johnson, "Controller tuning methods for industrial boilers", *IECON*, Nagoya, JAPAN 2000.
- [22] M. R. Katebi, and M. H. Moradi, "Predictive PID controllers", *Submitted to IEE Control Application*, Part D, 2001.
- [23] B. Kouvaritakis, and J. A. Rossiter, "Constrained stable generalized predictive control", *Proc. IEE Pt-D*, 140(4): 243-254, 1993.
- [24] Levine, W, "The control handbook press", 1996.
- [25] 김종식, "선형 제어시스템 공학"
- [26] M. H. Moradi, and M. R. Katebi, "Predictive PID control for autopilot design", *IFAC CAMS 2001*, Glasgow, Scotland, UK.



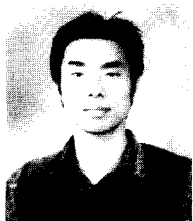
김 양 환

2002년 부경대학교 제어계측공학과 졸업.
2003년~현재 부산대학교 대학원 전자공학과 석사과정 재학중. 관심분야는 로버스트 제어, 네비게이션시스템.



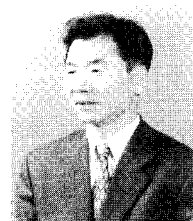
이 정 재

2003년 울산대학교 제어계측공학과 졸업. 2003년~현재 부산대학교 대학원 전자공학과 석사과정 재학중. 관심분야는 비전시스템, 최적제어.



이 정 용

2003년 부경대학교 제어계측공학과 졸업.
2003년~현재 부산대학교 대학원 전자공학과 석사과정 재학중. 관심분야는 예측 제어.



이 장 명

1980년 서울대학교 전자공학과 공학사.
1982년 서울대학교 전자공학과 공학석사.
1990년 University of Southern California 공학박사.
1992년~현재 부산대학교 전자공학과 교수. 관심분야 다관절 로봇 시스템 설계 및 제어, 협동제어, 햅틱 인터페이스, 마이크로 프로세서 응용 시스템 설계.