

스텝응답에 기반한 PID/PIDA 제어기의 자동동조

Auto-tuning of PID/PIDA Controllers based on Step-response

안 경 필, 이 준 성, 임 재 식, 이 영 일*

(Kyung-Pil Ahn, Jun-Sung Lee, Jae-Sik Lim, and Young-Il Lee)

Abstract: In this paper, a method of auto-tuning of PID (Proportional-Integral-Derivative) and PIDA (Proportional-Integral-Derivative-Acceleration) controllers is proposed that can be applied to a time-delayed second order model. The proposed identification method is based on step responses, but it can be easily automated using digital controller unlike the existing graphical identification methods. We provide a ways to yield parameter identifications which is independent to initial values of the plants. The tuning rule is based on the pole-placement strategy and is formulated so that it can be implemented using a digital controller with ease.

Keywords: auto-tuning, PID (Proportional-Integral-Derivative), PIDA (Proportional-Integral-Derivative-Acceleration), step-response, pole-placement

I. 서론

현대적인 고급 제어이론에 기초한 다양한 제어기의 형태가 있음에도 여전히 고전적인 PID 형태의 제어기가 산업용 제어기의 대부분을 차지한다. 그 이유는, 제어법칙이 단순하여 구현하기가 쉽고, 적분기를 가지고 있어서 일정한 설정치를 오차 없이 추종할 수 있으며, Ziegler-Nichols 식[1]과 같이 잘 알려진 동조법칙이 있으며, 무엇보다도 현장 기술자들에게 친숙하다는 점 등이다.

제어기를 온라인으로 동조하려면 플랜트에 대한 정보를 어떤 실험을 통해 구하고 이 정보에 기초하여 제어기의 설계한다. 자동동조(auto-tuning)는 적응제어와는 달리 플랜트의 식별과정과 설계과정은 분리되며 일단 동조된 제어기의 파라미터는 이후 고정되어 사용된다. 자동동조 기능을 갖춘 제어기는 비교적 가격이 저렴한 범용 제어기에 사용되기 때문에 계산이 복잡하지 않고 쉽게 구현할 수 있는 효과적인 식별법과 동조방법이 요구된다. 참고문헌 [2,3]에 이와 관련한 풍부한 내용이 수록되어 있다.

가장 잘 알려진 PID 동조 방법은 Zilger와 Nichols[1]가 제안한 스텝응답에 기초한 것과 주파수응답에 기초한 것이다. Åström과 Hägglund[4]는 릴레이 피드백(relay feedback)을 이용하여 Ziegler-Nichols의 주파수응답법을 자동화하는 방법을 제안하였다. 이 방법은 매우 간단한 실험으로부터 다양한 플랜트에 대해 건실한 제어기를 온라인으로 동조할 수 있다는 장점 때문에 산업 현장에서 많이 사용되고 있다. 하지만 릴레이 동조법을 시스템에 적용하기 위해서는 릴레이 피드백을 구성하였을 때 주기적인 진동이 발생해야 하는 조건이 만족되어야한다. 그리고 이때 발생된 진동이 대칭적이어야 안정적인 튜닝이 가능하며 진동이 비

대칭적일 때에는 대칭적 진동이 되도록 만들어 주는 과정이 필요하다[5]. 특히 열을 가할 수만 있는 히터의 경우 매우 비대칭적인 진동이 발생할 수 있는데, 이때 플랜트의 출력이 원하는 한도이상을 벗어날 수도 있다. 이런 경우는 릴레이 동조법이 바람직하지 않을 수 있다.

한편 Ziegler-Nichols의 스텝응답법은 스텝응답에서 시간지연을 갖는 적분기 모델을 도식적으로 구하여 이에 기초한 PID 파라미터를 경험적으로 구하여 놓은 것이다. 참고문헌 [2,6]에 스텝응답으로부터 시간지연이 있는 몇 가지 모델을 식별하는 방법이 소개되어 있다. 이들 대부분은 스텝입력을 인가하기 전의 초기조건을 고려하지 않고 있다. 즉 모든 초기조건이 0이라 가정하는데 이는 플랜트가 완전히 정지된 상태 또는 일정한 설정치로 안정화된 상태를 유지하고 있어야 하는 실제적인 측면에서의 제약이 있다. 한편, Methew와 Fairman[7]은 임의의 입력에 대해 초기조건을 고려하여 플랜트를 식별하는 방법을 제안하였다. 플랜트의 출력을 입력신호와 플랜트출력의 (중)적분의 선형조합으로 표현하여 최소자승법으로 플랜트의 파라미터를 식별한다. Wang과 Zhang[8], Ahmed 등[9]은 초기조건을 고려하여 스텝응답으로부터 플랜트를 식별하는 방법을 제안하였는데, 이는 [7]의 방법을 시간지연이 있는 플랜트로 확장한 것으로 볼 수 있다.

본 논문에서는 스텝응답으로부터 시간지연을 갖는 2차 적분기 모델을 식별되 모델의 특성을 분석하여 간단하면서도 효율적으로 초기치를 고려할 수 있도록 하였으며, 식별된 모델에 대하여 극점배치 방법을 통하여 PID 및 PIDA (Proportional-Integral-Derivative-Acceleration)제어기를 설계한다. 제안된 시스템 식별 및 제어기 자동 동조 방법은 마이크로프로세서로 구현하기 용이한 알고리즘으로 제시하였다.

폐-루프의 극점은 오버슈트를 일으키지 않을 조건을 만족하도록 선택된다. PID 제어기의 파라미터는 해를 찾기 위한 수치적인 알고리즘이 필요한 반면, PIDA 제어기의 파라미터는 설계 파라미터에 대한 간단한 대수식으로 주어진다. PIDA 제어기 구조는 참고문헌[10,11] 등에서 제안된

* 책임저자(Corresponding Author)

논문접수: 2009. 5. 4., 수정: 2009. 7. 8., 채택확정: 2009. 7. 31.

안경필, 이준성, 이영일: 서울산업대학교 제어계측공학과

(ahn8202@snut.ac.kr/gainus@snut.ac.kr/yilee@snut.ac.kr)

임재식: 서울산업대학교 나노아이티공학과(jslim@snut.ac.kr)

※ 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구원(KRF-2008-313-D00321).

바 있다.

제안한 식별 및 제어기 설계법을 MatLab의 M-file을 이용하여 구현하였으며 시뮬레이션을 통하여 제안한 시스템 식별과 PID/PIDA 동조 방법의 유효성을 입증하였다.

II. 시간지연이 있는 2차 적분기 모델의 식별

다음과 같은 시간지연이 있는 2차 적분기 모델을 생각한다.

$$G(s) = \frac{K}{s(Ts+1)} e^{-Ls} \quad (1)$$

식 (1)과 같은 시간지연이 있는 2차 적분기 모델은 화학 반응의 제어[12,13], DC 모터의 위치제어[14], 온도제어[15] 등에 유용하게 사용될 수 있다.

1. 도식적 식별 방법

본 논문에서 제안하는 식별방법을 소개하기 전에 [2]에서와 같이 도식적 식별 방법으로 플랜트의 파라미터 K , T , L 을 구하는 방법을 살펴보자.

그림 1과 같이 스텝 응답의 기울기가 일정한 부분에서 작도한 직선으로부터 $K(L+T)$ 값과 $L+T$ 값을 구하고, $K(L+T)$ 를 $L+T$ 로 나누어 파라미터 K 를 구한다. 그리고 $L+T$ 시점에서의 출력값 $y(L+T)$ 와 식 (1)로부터 파라미터 T 값을 다음과 같이 구할 수 있다[2].

$$T = \frac{y(L+T)}{K} e^1. \quad (2)$$

파라미터 L 은 작도해서 얻은 $L+T$ 값에 식 (2)를 통해 얻어낸 T 와의 차를 구함으로써 얻어낼 수 있다. 마지막으로, 구해진 파라미터 K , T , L 을 식 (1)에 대입함으로써 알아내고자 하는 플랜트의 전달함수를 얻을 수 있다.

스텝 응답의 그래프를 얻고 기울기가 일정해 지는 곳에 접선을 작도하여 파라미터를 계산하는 이 방법은 계산 과정도 간단하며, 매우 정확한 시스템 모델의 식별이 가능하다.

하지만 도식적 방법은 초기 값이 0인 경우에만 식별이 가능하다는 문제가 있다. 때문에 0 이 아닌 초기 값을 가지고 있을 때 스텝 응답의 기울기가 달라지므로 초기 값을 고려한 보정방법이 필요하다. 또 다른 문제는 이를 마이크로프로세서를 이용하여 구현할 때 발생한다. 식 (2)와 같이 파라미터 T 를 구하기 위해선 플랜트의 시간 $L+T$ 에서의

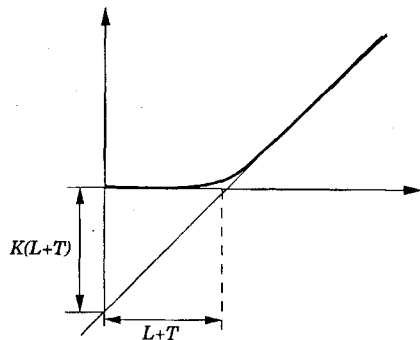


그림 1. 도식적 식별방법.
Fig. 1. Graphical method.

출력 값 $y(L+T)$ 를 먼저 알아야 한다. 하지만 기울기가 일정해지기 전에는 접선을 작도할 수 없기 때문에 정확한 $L+T$ 값을 알 수 없다. 이를 마이크로프로세서에서 구현하기 위해선 기울기가 일정해질 때까지 플랜트의 모든 출력 값을 기억장치에 저장하고 있어야한다. 이는 반응이 느린 플랜트의 경우나 저장 공간이 부족한 경우 문제가 발생한다. 본 논문에서는 이와 같은 문제를 해결하는 파라미터 식별 방법을 제안한다.

2. 제안하는 식별 방법

본 논문에서는 도식적 식별 방법을 보완하여 초기 값이 있는 스텝 응답으로부터 마이크로프로세서를 이용하여 효율적으로 식 (1)의 파라미터를 식별하는 알고리즘을 제안한다.

식 (1)에서 $L=0$ 인 경우에 스텝입력을 가하여 나타나는 출력의 라플라스 변환은 다음과 같이 주어진다.

$$Y(s) = \frac{aK}{s^2(Ts+1)}, \quad a = \text{스텝 입력 크기.} \quad (3)$$

식 (3)을 도함수의 라플라스 변환식에 초기값이 $y_0 \neq 0$, $y'_0 \neq 0$, $y''_0 \neq 0$ 이라고 고려하여 풀어쓰면 $Y(s)$ 는 다음과 같이 주어진다.

$$Y(s) = \frac{aK + (s^2Ty''_0 + s(Ty'_0 + y_0) + (y'_0 + Ty''_0))}{s^2(Ts+1)} \quad (4a)$$

$$= \frac{l_1}{s^2} + \frac{l_2}{s} + \frac{l_3}{Ts+1}.$$

$$\begin{aligned} l_1 &= Ty''_0 + y'_0 + aK. \\ l_2 &= Ty'_0 + y_0 - l_1T. \\ l_3 &= Ty_0 - l_2T. \end{aligned} \quad (4b)$$

식 (4a)을 라플라스 역변환을 통해 시간 영역에서의 출력 $y(t)$ 를 구하고, 그 미분인 출력의 변화율 $y'(t)$ 를 다음과 같이 구할 수 있다.

$$y(t) = l_1t + l_2 + \frac{l_3}{T}e^{-t/T} \quad (5a)$$

$$y'(t) = l_1 - \frac{l_3}{T}e^{-t/T}. \quad (5b)$$

시간이 지남에 따라 식 (5b)의 $\exp(-t/T)$ 성분이 0에 가까워지면

$$y'(t) \cong l_1 \quad (t \gg T) \quad (6)$$

되고, 출력은 일정한 변화율을 가지게 된다. 이는 그림 1에서 시간이 지남에 따라 $y(t)$ 의 그래프가 일정한 기울기를 가지게 되는 것과 일치 한다. 따라서 식 (4b)와 (6)으로부터 파라미터 K 를 다음과 같이 구할 수 있다.

$$K = \frac{(y'(\infty) - Ty''_0 - y'_0)}{a}. \quad (7)$$

식 (7)에서 T 를 고려하지 않고 파라미터 K 를 검출하기 위해서는 $y''_0 = 0$ 이라고 가정한다. 이 경우 K 는 초기 기울기 y'_0 를 고려하여 다음과 같이 주어진다.

$$K = \frac{(y'(\infty) - y'_0)}{a} \tag{8}$$

이제 파라미터 T 를 찾기 위해서 식 (5b)를 미분하면

$$y''(t) = \frac{l_3}{T^3} e^{-t/T} = (y''_0 + \frac{aK}{T}) e^{-t/T} \tag{9}$$

을 얻는다. 여기서 역시 $y''_0 = 0$ 이라 가정한다.

식 (9)에서 시간 $t = 0$ 일 때, $y''(t)$ 는 최대값을 가지며, $t = T$ 이면 $y''(t)$ 는 최대값의 e^{-1} 배의 값 즉, 최대값의 36.7%의 값을 가지게 된다. 따라서 $y''(t)$ 가 최대값의 36.7%까지 떨어지는 시간이 곧 파라미터 T 가 된다. 스텝 입력 인가 후 $y''(t)$ 의 최대값을 y''_{max} 라 둘 때 다음과 같은 관계를 얻을 수 있다.

$$y''(T) = 0.367 \times y''_{max} \tag{10}$$

$L \neq 0$ 인 경우에도 식 (8), (10)의 식별식은 그대로 적용 가능하다. 식 (9)에서 $y''_0 = 0$ 인 상태에서 스텝 입력을 가하면 스텝입력을 가하는 시점인 $t=0$ 에서 $y''(t)$ 는 최대값을 가지게 되지만 시간지연이 있다면 $t=L$ 에서 $y''(t)$ 가 최대값을 가지게 된다. 따라서 $t=0$ 에서 스텝 인가 후에 $y''(t)$ 가 최대가 되는 시간을 t_{max} 라 두면 지연시간은

$$L = t_{max} \tag{11}$$

이 된다.

지금까지 도출된 식 (8), (10), (11) 을 기반으로 한 식별 알고리즘을 다음 절에서 제안한다.

3. 식별 알고리즘

파라미터 식별과 PID 제어를 위한 시스템 구성은 그림 2와 같다. 잡음의 영향을 줄이기 위해서 사용한 저역 통과 필터를 거친 출력 $y(t)$ 를 이용해서 $y'(t)$, $y''(t)$ 를 구하고 이를 시스템 식별에 활용한다. 알고리즘에서 $y''(t) > 0$, $y''(t) = 0$ 인 상태를 검출하는 것이 필요한데 이 값들이 정확히 0이 되는 것을 기대하기 힘들므로 실제 플랜트에 적용하기 위해서는 적절한 문턱값(threshold) δ_1 , $\delta_2 (> 0)$ 를 사용한다. 그리고 저역 통과 필터의 차단 주파수(cutoff frequency)가 플랜트의 시정수에 비해 충분히 크다고 가정한다.

• 시스템 식별 알고리즘

시스템 식별 알고리즘은 2절에서 도출한 식 (8), (10), (11)을 기반으로 총 7단계로 구성 되어있다. 각 단계에서는

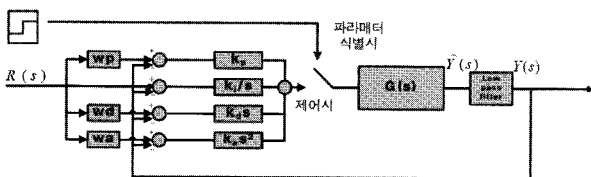


그림 2. 시스템 구조

Fig. 2. System structure.

마이크로프로세서로 식별 알고리즘을 구현 시 식별과정이 다음 단계로 넘어가는 조건 및 각 단계에서 해주어야 할 일들을 설명하였다.

Step 1: 시스템의 입력 $u=0$ 으로 두고 출력을 살펴면서 $|y''(t)| < \delta_1$ ($0 < \delta_1 \ll 1$)이 되기를 기다린다.

Step 2: Step 1의 조건이 만족되면 $t=0$ 으로 설정하고 y'_0 의 값을 저장 후, 크기가 a 인 스텝 입력을 시스템에 인가한다.

Step 3: 식 (11)에 의해서 스텝입력 후 최초로 $y''(t) > \delta_2$ ($\delta_1 < \delta_2 \ll 1$)가 되는데 걸리는 시간 t 를 파라미터 L 로 결정한다.

Step 4: $y''(t)$ 이 최대값을 가지게 되는 시간 t_{max} 와 출력 값 $y''(t_{max})$ 을 저장한다.

Step 5: 식 (10)의 관계에 따라 $y''(t)$ 이 최대값 $y''(t_{max})$ 의 36.7% 까지 떨어질 때 까지 기다리고, 그 시간과 Step 4에서 얻은 t_{max} 와의 차를 파라미터 T 로 결정한다.

Step 6: $|y''(t)|$ 값이 감소하여 $|y''(t)| < \delta_1$ 이 되는 시간의 $y'(t)$ 값을 구하고, Step 2에서 미리 구해놓았던 초기의 y'_0 값과 스텝입력의 크기 a 를 이용하여 식 (8)을 통해 파라미터 K 를 구한다.

Step 7: 스텝 입력을 해제하고, 식별된 파라미터를 이용하여 제어기를 설계한다.

각 Step에서 필요로 하는 $y(t)$, $y'(t)$ 및 $y''(t)$ 에 포함된 잡음을 제거하기 위해서 저역통과 디지털 필터를 사용한다.

III. 제어기 설계

지금부터 시스템 식별로 얻은 플랜트 정보를 바탕으로 식 (1)의 플랜트에 대해서 제어기를 설계하는 방법을 살펴본다. 제어기의 설계는 극배치 방법을 사용하며 오버슈트 발생하지 않는 조건을 만족하도록 페루프 극점의 위치를 선택한다. 그리고 이렇게 선택된 페루프 극점의 설계를 만족시키기 시키는 PID 및 PIDA 제어기의 설계 변수 결정 방법에 대해서 고찰한다.

1. 시스템 구성 및 제어기 설계

전체 시스템은 그림 2와 같이 PID (PIDA) 제어기와 플랜트를 페루프를 구성한다. 각 PID 이득은 k_p, k_i, k_d 로, 가속기(acceleration)의 이득은 k_a 로 표시하였다. 극배치 방법은 극점의 설계만을 만족시키므로 설계 이외의 요소인 영점의 영향을 최소화하도록 간단한 2자유도 구조를 구성하고 이에 대응하는 이득을 각각 w_p, w_d, w_a 로 표시하였다.

극 배치 방법을 사용하기위해서 식 (1)의 플랜트에 시간 지연 부분인 e^{-Ls} 을 $\frac{-Ls+2}{Ls+2}$ 형태로 1차 Pade 근사화시키면 플랜트 전달함수 $G(s)$ 는 식 (12)처럼 하나의 영점을 갖는 3차의 전달함수로 표시할 수 있다.

$$G(s) = \frac{-ls + 1}{d_3s^3 + d_2s^2 + d_1s + d_0} \tag{12}$$

여기서 $d_3 = (TL)/2K$, $d_2 = (2T + L)/2K$, $d_1 = 1/K$,

$l = L/2, d_0 = 0$.

그림 2에서 제어기를 포함한 전체 시스템의 페루프 전달 함수 $T(s)$ 는 다음과 같은 4차 모델로 주어진다.

$$T(s) = \frac{N(s)}{D(s)}. \quad (13)$$

$$\begin{aligned} N(s) &= (-ls + 1)(w_a k_a s^3 + w_d k_d s^2 + w_p k_p s + k_i). \\ D(s) &= (d_3 - lk_a)s^4 + (d_2 + k_a - lk_d)s^3 + \\ &\quad (d_1 + k_d - lk_p)s^2 + (d_0 + k_p - lk_i)s + k_i. \end{aligned} \quad (14)$$

극배치 방법을 사용할 경우 미리 설계한 극점과 $T(s)$ 의 극점만을 동일하게 맞추게 되고 $T(s)$ 의 영점은 제어기의 설계에 반영되지 않는다. 페루프 전달함수의 영점은 스텝 응답에 있어 오버슈트에 악영향을 미치므로 본 논문에서는 오버슈트가 없는 응답을 얻기 위해 $w_p = w_d = w_a = 0$ 으로 둔다.

극배치 방법을 위해 미리 선택한 페루프 특성 다항식 $\Delta(s)$ 을 다음과 같은 형태로 표시하자.

$$\Delta(s) = s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0. \quad (15)$$

그러면 $D(s) = \Delta(s)$ 를 만족하기 위해서 제어기 설계 변수는 아래의 등식을 만족하여야 한다.

$$\begin{aligned} d_2 + k_a - lk_d &= c_3(d_3 - lk_a). \\ d_1 + k_d - lk_p &= c_2(d_3 - lk_a). \\ d_0 + k_p - lk_i &= c_1(d_3 - lk_a). \\ k_i &= c_0(d_3 - lk_a). \end{aligned} \quad (16)$$

특성 다항식 $\Delta(s)$ 의 결정에 대해서 다음 절에서 생각해 보자.

2. 페루프 극점의 선택

1절에서 살펴본 바와 같이 전체 시스템의 페루프 전달함수는 4차의 모델이 되므로 극 배치 방법에서도 4개의 극점을 선택해야한다. 극점의 위치는, 시스템 식별에서 발생할 수 있는 오차 가능성과 극 배치 방법을 사용할 때 시간 지연의 근사화로 인한 오차를 감안하여 안정한 응답을 얻도

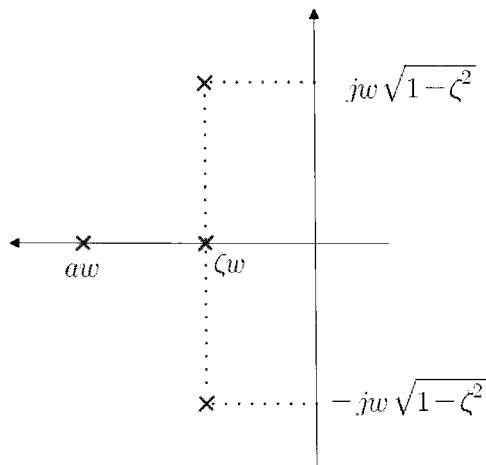


그림 3. 오버슈트가 없는 응답을 위한 극점 위치.

Fig. 3. Necessary pole location for the response without overshoot.

록 선택한다. 그리고 페루프의 스텝응답이 오버슈트가 없도록 극점의 위치를 선택한다. 그림 3과 같이 우세한 극점 3개를 동일한 실수부를 갖도록 두면 오버슈트가 없는 스텝응답을 얻을 필요조건을 만족한다[16,15].

이와 같이 선택한 극점의 위치로부터 극배치 방법에 사용할 특성 다항식 $\Delta(s)$ 를 식 (17)과 같이 표시할 수 있다.

$$\begin{aligned} \Delta(s) &= (s + \alpha w)(s + \zeta w)(s^2 + 2\zeta w s + w^2). \\ &:= s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0 \end{aligned} \quad (17)$$

여기서 선택 가능한 α, ζ, w 의 범위는 제어기의 구성에 따라 달라지게 된다.

3. PID 제어기 설계

그림 2에서 가속기(acceleration)를 사용하지 않으면($k_a = 0$) PID 제어기로 페루프를 구성한 것과 동일한 구조가 된다. PID 구조에서 극 배치 방법에 의해 제어기를 설계하는 자세한 방법은 참고문헌[15]에 나와 있으며 간단히 살펴보면 다음과 같다.

식 (16)에서 $k_a = 0$ 으로 놓으면 다음 수식을 얻을 수 있다.

$$d_2 - k_d l = c_3 d_3. \quad (18a)$$

$$d_1 + k_d - k_p l = c_2 d_3. \quad (18b)$$

$$d_0 + k_p - k_i l = c_1 d_3. \quad (18c)$$

$$k_i = c_0 d_3. \quad (18d)$$

식 (18d), (18c), (18a)에 의해 k_i, k_p, k_d 를 각각 다음과 같이 결정할 수 있다.

$$\begin{aligned} k_i &= c_0 d_3. \\ k_d &= \frac{d_2 - c_3 d_3}{l}. \\ k_p &= c_1 d_3 - d_0 + c_0 d_3 l. \end{aligned}$$

식 (18b)에 위에서 도출한 k_p, k_d 를 대입하면 2절에서 미리 결정한 $\Delta(s)$ 의 계수 c_0, c_1, c_2, c_3 는 다음 수식을 만족해야 한다. (즉 PID 제어기를 사용할 경우 극점의 선택에 제한이 가해진다.)

$$c_3 d_3 = d_2 + (d_1 - c_2 d_3)l + (d_0 - c_1 d_3)l^2 - c_0 d_3 l^3. \quad (19)$$

식 (17)에서 c_0, c_1, c_2, c_3 를 α, ζ, w 으로 표현하여 식 (19)에 대입하면 다음 수식을 얻는다.

$$\begin{aligned} \alpha \zeta d_3 l^3 w^4 &+ (\alpha + \zeta + 2\alpha \zeta^2) d_3 l^2 w^3 \\ &+ [1 + \alpha \zeta + 2(\alpha + \zeta)\zeta] d_3 l w^2 \\ &+ (\alpha d_3 + \zeta d_3 + 2d_3 \zeta) w \\ &- d_2 - d_1 l - d_0 l^2 = 0 \end{aligned} \quad (20)$$

비 우세극점과 우세극점인 α, ζ 를 먼저 결정하면 식 (20)은 w 에 대한 4차식이 되는데 모든 파라미터가 양수이면 위 식을 만족시키는 유일한 양의 해가 존재함을 Routh-Hurwitz 판별법을 적용하여 확인할 수 있다[15].

PID 설계변수는 대수식이므로 마이크로프로세서로 구현함에 있어서 어려움이 없다. 하지만 PID 설계 변수 결정에

앞서 미리 선택해야하는 극점의 경우 식 (20)의 4차 방정식을 풀어야한다. 여기서 4차 방정식의 실근을 구하는 것은 아래의 알고리즘[17]을 사용해서 마이크로프로세서에서 비교적 쉽게 구현이 가능하다.

• 다항식 $f(w) = 0$ 의 실근을 구하는 알고리즘

Step 1: 임의로 두 점 w_1, w_2 를 정하고 찾을 근의 허용 오차를 결정한다.

Step 2: 두 점 w_1, w_2 에서 다항식 값 $f(w_1), f(w_2)$ 를 계산하고 이 두 값의 부호를 비교한다. (다항식의 실근은 w 절편을 의미하므로 임의의 두 점에서 다항식 값의 부호가 다르다면 그 두 점 사이에 다항식의 근이 존재함을 알 수 있다.) 부호가 같다면 (iii)의 절차를 수행하여 두점 사이에 근이 포함되도록 간격을 넓히고 부호가 다르다면 이미 두점 사이에 근이 존재하므로 (iv)의 절차를 수행한다.

Step 3: 두 점 w_1, w_2 사이의 간격을 늘리고 다시 절차 (ii)를 수행한다. (식 (20)에서 양의 w 를 찾으므로 이 절차에서 두 점 w_1, w_2 를 양수로 제한시킨다.)

Step 4: 두 점 w_1, w_2 의 중간 값을 계산하여 w_3 에 넣는다.

Step 5: w_3 점에서 다항식 값 $f(w_3)$ 을 계산한다. 만약 이 값이 0이라면 다항식의 근을 찾은 것이므로 w_3 값을 근으로 표시하고 종료한다. 0이 아니라면 다음 절차 (vi)을 수행한다.

Step 6: $f(w_3)$ 의 부호를 두 점 w_1, w_2 에서의 다항식 값인 $f(w_1), f(w_2)$ 의 부호와 비교한다. $f(w_3)$ 의 부호가 $f(w_1)$ 와 같다면 w_1 에 w_3 값을 넣는다. 그렇지 않고 $f(w_3)$ 의 부호가 $f(w_2)$ 와 같다면 w_2 에 w_3 값을 넣는다. (즉 두 점 w_1, w_2 사이에 다항식의 근을 유지한 채 w_1, w_2 사이의 간격을 절반으로 줄인다)

Step 7: 두 점 w_1, w_2 사이의 간격이 허용오차 보다 크다면 다시 절차 (iv)를 수행한다. 그렇지 않고 작다면 w_1, w_2 의 중간 값을 근으로 표시하고 종료한다.

위 알고리즘은 다항식에서 실근이 존재 할 때 최소한 하나의 근을 찾을 수 있음을 보장 할 수 있다[17].

결국 PID 제어기로는 모든 극점을 자유로이 배치할 수 없고 그림 3에서처럼 α 와 ζ 를 임의로 정하면 w 는 (20)의 해로 결정된다. 반면에 PIDA 형태의 제어기를 사용하면 α, ζ 및 w 를 임의로 결정할 수 있다.

4. PIDA 제어기 설계

그림 2에서 가속기를 사용한 PIDA 구조를 사용하면 식 (16)은 다음과 같이 정리할 수 있다.

$$k_p = \frac{D}{c_0 l^4 + c_1 l^3 + c_2 l^2 + c_1 l + 1} \tag{21a}$$

$$k_i = \frac{c_0 k_p + d_0 c_0}{c_0 l + c_1} \tag{21b}$$

$$k_d = \frac{(c_1 l + c_2) k_p - c_2 l k_i + c_2 d_0 - c_1 d_1}{c_1} \tag{21c}$$

$$k_a = \frac{d_2 - d_3 c_3 - l k_d}{c_3 l + 1} \tag{21d}$$

여기서

$$D = [c_0 c_2 d_3 + c_0 (d_1 - d_3 c_2)] l^3 + [c_2 (c_1 d_3 - d_0) + c_0 c_3 d_3 + c_0 (d_2 - d_3 c_3) + c_1 (d_1 - d_3 c_2)] l^2 + [c_3 (c_1 d_3 - d_0) + c_0 d_3 + c_1 (d_2 - d_3 c_3)] l + c_1 d_3 - d_0$$

위 식 (21a) ~ (21d)를 순서대로 풀면 PIDA 설계 변수를 바로 결정할 수 있다.

임의의 $\Delta(s)$ 의 계수 c_0, c_1, c_2, c_3 에 대해서 식 (16)을 만족하는 4개의 PIDA 제어기 설계 변수 k_p, k_i, k_d, k_a 를 찾아 낼 수 있으므로 PID 제어기와 달리 배치할 수 있는 극점의 위치에 제한이 없어진다. 그러나 모델의 불확실성을 고려하여 적절한 극점의 위치를 선택하여야 한다.

IV. MatLab을 이용한 시뮬레이션

1. 식별 알고리즘 시뮬레이션

1.1 스텝입력에 대한 출력 데이터 생성

시스템 식별 알고리즘을 검증하기 위해서 그림 4와 같이 초기 스텝 입력과 외란에 의한 잡음을 갖는 블럭을 구성한다.

그림 4의 블록은 MatLab Simulink를 이용해서 구현하였다. 우선 크기가 $b (< a)$ 인 예비 스텝 입력을 사용해서 대상 모델이 초기값을 가지도록 하고, 모델의 출력에 외란으로 $\pm n (n=0, 1, 2)$ 범위의 난수를 잡음으로 추가 하였다.

예비 스텝 입력에 대한 응답이 정상 상태에 도달하고 초기값에 의한 응답이 $|y''(0)| < \delta_1$ 이 되는 시점에 크기가 a 인 스텝 입력을 가하여 플랜트의 출력을 생성하였다. 이를 0.1초 단위로 샘플링해서 y 벡터에 저장한다.

1.2 식별 알고리즘의 구현

식별 알고리즘은 MatLab 에서 제공하는 간단한 사칙연산 기능과 조건문 등을 이용하여 M-file로 간단하게 구현이 가능하다. 이는 마이크로프로세서에 C 언어를 이용하여 구현하는 방식과 흡사하여 M-file을 토대로 쉽게 이식하는 것이 가능하다.

미분을 수행하는 방법은 디지털 방식을 사용하였다. 즉 일정한 시간 간격으로 출력을 측정하고, 한 단계 전의 측정값과 현재 측정값과의 차이를 시간 간격으로 나눔으로써 미분값을 얻어내었다. 출력단의 저역 통과 필터는 차단 주파수(cutoff frequency) 값을 0.01로 갖는 Butterworth 방식의 디지털 2차 저역 통과 필터를 사용하였다. 알고리즘의 문턱

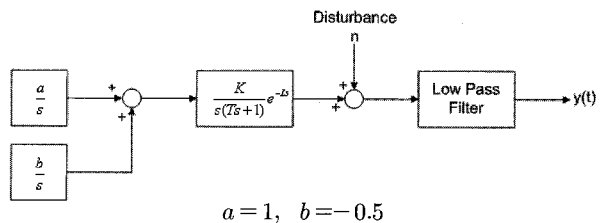


그림 4. 시스템 식별 알고리즘 시뮬레이션.
Fig. 4. System identification algorithm simulation.

표 1. 시스템 식별 결과.

Table 1. Result of system identification.

순번	대상 모델 파라미터			식별된 파라미터		
	K	T	L	K	T	L
1	5	2	4.3	5.233	3.55	3.9
2	4.6	2.9	1.5	4.758	3.65	2.5
3	3.8	4.2	7.5	3.866	4.2	5.75
4	1.5	2.7	5.8	1.558	3.8	5.5
5	0.8	7.2	2.7	0.756	4.15	5.75

표 2. 외란 크기 변경에 따른 시스템 식별 결과.

Table 2. Result of system identification about changing disturbance.

파라미터	외란 크기 (n)	대상모델	외란 크기		
			0	±1	±2
K		3.8	3.869	3.867	3.866
T		4.2	4.05	4.15	4.2
L		7.5	5.75	5.75	5.75

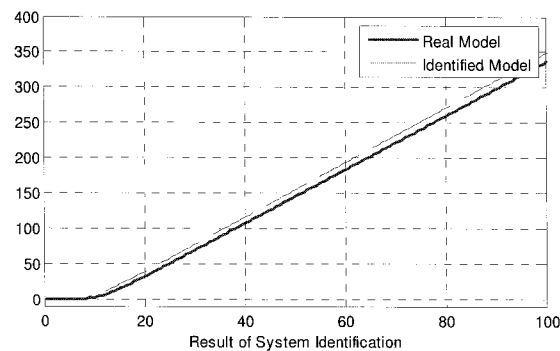
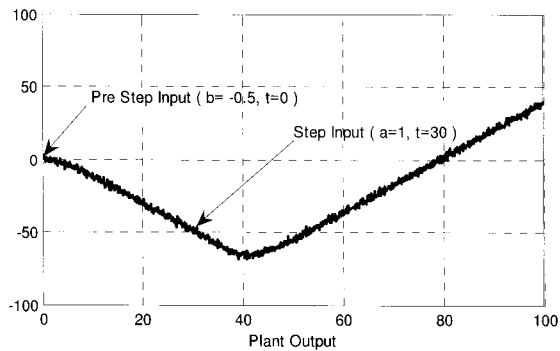


그림 5. 대상 모델의 출력과 시스템 식별 결과.

Fig. 5. Output of target model and result of system identification.

값(threshold) $\delta_1 = \delta_2 = 0.03$ 으로 정하였다.

이렇게 구현된 M-file을 1.1절에서 저장한 샘플링데이터 y 벡터에 적용하였다.

1.3 식별 알고리즘의 시뮬레이션 결과

표 1은 대상 모델의 파라미터를 변경하면서 시스템 식별 알고리즘을 적용한 결과를 나타낸다.

표 2에서는 파라미터 값을 고정하고 외란의 크기를 변경 하면서 시스템 식별 알고리즘을 적용한 결과를 나타낸다.

그림 5의 첫번째 그래프는 외란을 포함하고 초기값이 0 이 아닌 대상 모델의 출력을 보여준다. 두번째 그래프는 대상 모델의 출력에 식별 알고리즘을 적용시켜 식별된 모델과 실제 모델의 스텝 입력에 대한 반응의 차이를 나타낸다. 약간의 오차를 제외하고는 실제 모델과 식별된 모델이 근사한 응답을 보이는 것을 확인 할 수 있다.

2. 제어기 설계

1절에서 표 1의 식별된 플랜트 중에서 서로 다른 특성을 갖는 1, 5번 2 종류의 플랜트에 대해 설계 예제를 보인다.

극배치 방법에서 제어기 설계 시 목표응답을 나타내는 극점은 앞서 논의한 대로 그림 3과 같이 선택하되 본 시뮬레이션에서는 ζ, α 를 각각 0.8, 5로 고정하여 극점의 상대적인 위치를 결정하였다. ζ, α 의 결정에 따라 PID 구조에서는 w 가 식 (20)에 의해 결정되는 반면 PIDA 구조에서는 임의로 선택 가능하다. PIDA 제어기 설계에서는 w 를 식 (20)에 의해 결정되는 w 값에 각각 1.2 배와 0.8배를 하여 w 변화에 따른 응답 속도의 변화를 살펴 볼 수 있도록 하였다.

극점의 위치 ζ, α, w 를 선택하면 각 제어기 설계 변수는 III 장 2, 3절에서 논의한대로 대수식으로 주어지므로 MatLab의 기본 문법을 사용하여 바로 구할 수 있다.

다음 표 3와 표 4는 각각 1절 표 1의 1, 5번 식별 모델에 대해 설계된 제어기 설계 변수를 나타낸다.

3. 시뮬레이션 결과

시뮬레이션은 그림 2의 구조를 Simulink 상에서 구현하여 이루어졌다. 그림 6과 그림 7은 2절에서 설계한 PID 및 PIDA 제어기를 1절에서 각각 식별 대상인 실제 플랜트에 적용하였을 때의 결과 그래프를 나타낸다. 시스템 식별에 기반해 설계한 제어기를 실제 플랜트에 적용한 자동동조 결과가 설계 목표 응답(각각 $\zeta = 0.8, \alpha = 5\zeta, w = 0.1129$ 와 $\zeta = 0.8, \alpha = 5\zeta, w = 0.0839$)에 근접함을 확인할 수 있다.

표 3. 표 1의 1번 플랜트에 대해 설계한 각 제어기 설계 변수. Table 3. The parameters of the each controller that is designed for the first plant of the Fig. 1 ($\zeta = 0.8, \alpha = 5\zeta$).

	PID	PIDA	
ω	0.1129	0.0903	0.1355
k_p	0.0202	0.0124	0.0295
k_i	0.0007	0.0003	0.0012
k_d	0.0487	-0.0113	0.1068
k_a	0	-0.1450	0.113

표 4. 표 1의 5번 플랜트에 대해 설계한 각 제어기 설계 변수. Table 4. The parameters of the each controller that is designed for the fifth plant of the Fig. 1 ($\zeta = 0.8, \alpha = 5\zeta$).

	PID	PIDA	
ω	0.0839	0.0671	0.1007
k_p	0.0997	0.0618	0.1439
k_i	0.0025	0.0013	0.0043
k_d	0.2841	-0.1063	0.6560
k_a	0	-1.2610	0.9686

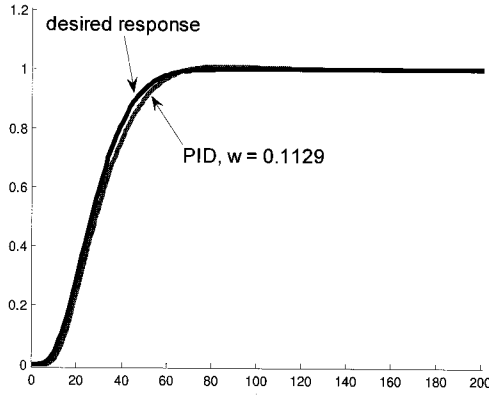


그림 6. 설계 목표 응답과 자동동조 결과 비교 (표3의 1번 플랜트).

Fig. 6. Comparison of the desired response with the result of the auto tuning (the first plant of the Fig. 3).

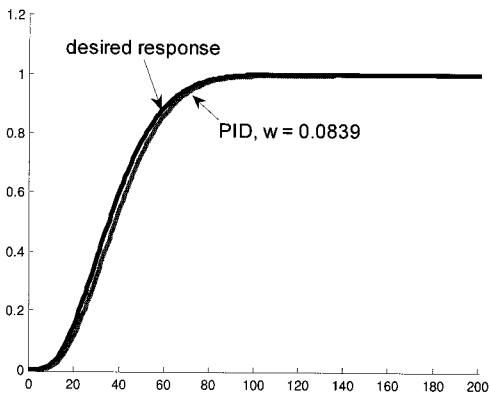


그림 7. 설계 목표 응답과 자동동조 결과 비교 (표3의 5번 플랜트).

Fig. 7. Comparison of the desired response with the result of the auto tuning (the fifth plant of the Fig. 3).

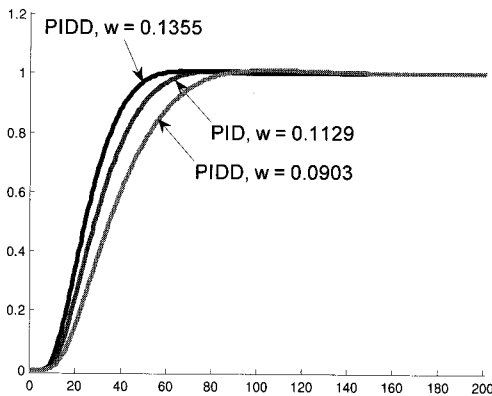


그림 8. w 의 변화에 따른 자동동조 결과 비교 (모델 1).

Fig. 8. Comparison of the auto tuning result as vary w (model 1).

다만 제어기의 설계가 실제 모델과 오차가 존재하는 식별 모델을 기준으로 설계된 점과 시간 지연 부분을 근사화시킨 모델을 기준으로 설계되었기 때문에 시스템이 불안정

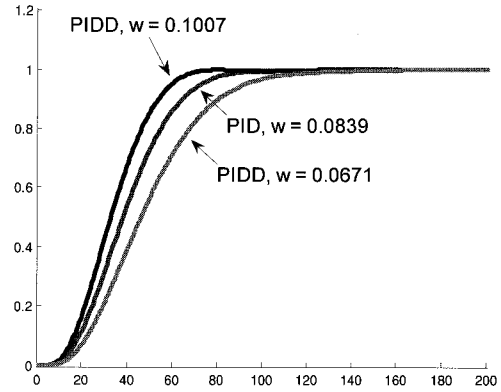


그림 9. w 의 변화에 따른 자동동조 결과 비교 (모델 5).

Fig. 9. Comparison of the auto tuning result as vary w (model 5).

해지는 w 의 구간이 존재하게 된다. 따라서 자동동조에 있어서 PIDA 제어기의 사용에는 주의가 필요하다. PID 제어기를 사용한 자동동조에서 응답 속도가 만족스럽지 않을 경우 PIDA 제어기의 사용을 고려해 볼 수 있으며 이때 w 의 선택은 PID 설계시 제한되었던 w 를 기준으로 삼아 조심스럽게 선택하여야 한다.

V. 결론

본 논문에서는 스텝응답에 기반한 자동동조 방법에 대해서 고찰하였으며 자동동조는 시스템 식별 부분과 제어기 설계 부분으로 나뉘어 논의되었다.

제안된 시스템 식별 방법은 그래프를 활용하는 방법으로부터 기본 아이디어를 얻어서, 시간 지연이 있는 2차 모델에 대해서 적용하였으며 마이크로프로세서로 구현이 용이하도록 알고리즘 형태로 도출하였다. 특히 제안된 시스템 식별방법은 플랜트의 스텝응답으로부터 플랜트의 초기값에 상관없이 플랜트의 파라미터에 대한 일정한 식별 값을 도출할 수 있으며 시뮬레이션을 통하여 외란에도 강인한 식별값을 도출해 냈을 보였다.

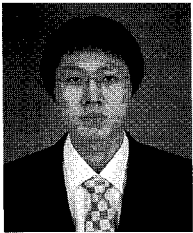
제어기 설계는 시스템 식별로 얻어진 플랜트를 대상으로 극 배치 방법이 사용되었으며 극 배치 방법에 사용할 극점을 선택하는 방법과 이를 만족시키는 PID 및 PIDA 제어기 설계 방법을 논의하였다. PIDA 제어기의 설계 변수는 대수식으로 도출되었지만 PID 제어기 설계에서는 극점 선택에 제한이 있게 되며 이때 다항식의 근을 찾는 과정이 수반된다. 제어기 설계를 마이크로프로세서에서 구현하기 쉽게 하기 위해서 다항식에서 실근을 찾는 알고리즘 역시 논의되었다.

마지막으로 이와 같은 자동동조 과정으로 설계된 제어기가 적용되었을 때 설계 목표 응답에 근접한 응답을 보임을 시뮬레이션을 통하여 확인하였다.

참고문헌

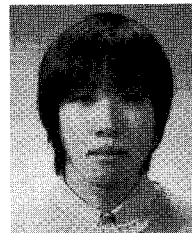
[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, pp. 759-68, 1942.

- [2] K. J. Aström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning* 2nd. Ed., Instrument Society of America, Research Triangle Park, NC, 1995.
- [3] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. on Control Systems Technology*, vol. 13, no. 4, pp. 559-576, 2005.
- [4] K. J. Aström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, pp. 645-6651, 1984.
- [5] C. C. Hang, K. J. Aström, and W. K. Ho, "Relay auto-tuning in the presence of static load disturbance," *Automatica*, vol. 29, no. 2, pp. 563-564, 1993.
- [6] H. Rake, "Step response and frequency response methods," *Automatica*, vol. 16, pp. 519-526, 1980.
- [7] A. V. Methew and F. W. Fairman, "Identification in the presence of initial conditions," *IEEE Trans. on Automat. Contr.*, vol. 17, pp. 394-396.
- [8] Q. G. Wang and Y. Zhang, "Robust identification of continuous systems with dead-time from step responses," *Automatica*, vol. 37, pp. 377-390, 2001.
- [9] S. Ahmed, B. Huang, and S. L. Shah, "Identification from step responses with transient initial conditions," *Journal of Process Control*, vol. 18, pp. 121-130, 2008.
- [10] S. Jung and R. C. Dorf, "Analytic PID controller design technique for a third order system," *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan, pp. 2513-2518, 1996.
- [11] D.-Y. Ha, I.-Y. Lee, Y.-S. Cho, Y.-D. Lim, and B.-K. Choi, "The design of pida controller with precompensator," ISIE 2001. Pusan. Korea.
- [12] E. Poutin and A. Pomerleau, "PID tuning for integrating and unstable processes," *IEE Proc -Control Theory App.* vol. 143, no. 5. Sep. 1996.
- [13] Y. H. Lee, J. S. Lee, and S. W. Park, "PID controller tuning for integrating and unstable processes with time delay," *Chemical Engineering Science*, vol. 55, pp. 3481-3493, 2000.
- [14] I. H. Cha and C. S. Han, "The auto-tuning PID controller using the parameter estimation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.
- [15] 임재식, "PID 제어기 지동동조에 관한 연구 및 온도제어에의 적용," 경상대학교 석사학위 논문, 2000.
- [16] S. K. Lin and C. J. Fang, "Nonovershooting and monotone nondecreasing step responses of a third-order SISO linear system," *IEEE Trans. on Automat. Contr.*, vol. 42, no. 9, pp. 1299-1303, 1997.
- [17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Press Syndicate of the University of Cambridge, pp. 350-354.



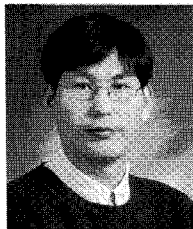
안 경 필

2008년 서울산업대학교 제어계측공학과 학사. 2008년 9월~현재 서울산업대학교 제어계측공학과 석사과정. 관심분야는 실시간 시스템.



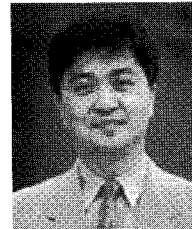
이 준 성

2009년 서울산업대학교 제어계측공학과 학사. 2009년~현재 서울산업대학교 제어계측공학과 석사과정. 관심분야는 실시간 시스템.



임 재 식

1996년 경상대학교 제어계측공학과 학사. 2000년 동 대학 대학원 석사. 2007년~현재 서울산업대학교 나노아이티공학과 박사과정. 관심분야는 Robust control, MPC 등.



이 영 일

1986년, 1988년, 1993년 서울대학교 제어계측공학과 학사, 석사, 박사. 1994년~2001년 7월 경상대학교 부교수. 2001년 8월~현재 서울산업대학교 제어계측공학과 교수. 1998년 2월~1999년 7월 Dept. of Engineering science, Oxford University, Visiting research fellow. 2007년 9월~현재 Editor of the International Journal of Control, Automatic, and system (IJCAS). 2009년 1월~현재 제어·로봇·시스템학회 국문지 편집장.