

재귀신경망을 이용한 다중모델/제어기 IMC의 과도 응답 개선

Transient Response Improvement of Multiple Model/Controller IMC Using Recurrent Neural Networks

오 원 근, 조 성 언, 소 지 영
(Won-Geun Oh, Sung Eon Cho, and Ji-Young So)

Abstract : The Multiple Model/Controller IMC(MMC-IMC) is a model-based control method which uses a set of model/controller pairs rather than a single model/controller to handle all possible operating conditions in the IMC control structure. During operation, one model/controller pair that best fit, for current plant situation is chosen by the switching algorithm. The major drawback of the switching controller is the bad transient performance due to the model error and the use of linear controller for nonlinear plants. In this paper, we propose a method that can improve the transient response of the MMC-IMC using two recurrent neural networks. Simulation result shows that the proposed method represents better performance than the usual MMC-IMC's.

Keywords : neural networks, internal model control, gain scheduling, switching controller

I. 서론

선형 제어를 사용하여 비선형 시스템을 제어하는 방법은 실제로 산업계에서 널리 사용되고 있는 비선형 제어기 설계법이다. 이러한 설계법은 비선형 시스템이라 하더라도 일정한 동작 영역에서는 선형 시스템으로 근사적으로 표현할 수 있다는 점을 이용한 것으로써 관심있는 동작점(operating point)에서 강인(robustness)한 선형 제어를 설계한다면 제어 목적을 충분히 달성할 수 있다. 그러나 비선형 시스템이 넓은 영역에서 동작하는 경우에는 하나의 선형 제어기로는 충분하지 않으며 각 동작 영역별로 분리된 다수의 선형 제어를 사용하여 시스템을 제어하는 방법을 사용하게 된다. 이렇게 비선형 플랜트의 제어에 다수의 선형 제어를 사용하는 설계법은 게인 스케줄링(Gain Scheduling)과 제어기 스위칭으로 크게 나눌 수 있으며 이중 게인 스케줄링의 일반적인 설계 절차는 다음과 같다[1].

1 단계 : 비선형 플랜트의 여러 동작점에서 선형 모델을 얻은 다음 각각에 대한 선형 제어를 설계한다.

2 단계 : 각 선형 제어기의 계수를 플랜트의 동작 상태를 반영하는 신호(scheduling variable)에 대한 함수가 되도록 매개 변수화하여 하나의 선형 시변 제어기로 만든다.

3 단계 : 컴퓨터 모의실험을 통해서 성능을 확인한다.

반면에 제어기 스위칭 방법은 각 동작점에서 설계한 여러개의 선형 제어를 이용한다는 점에서는 게인 스케줄링과 동일하나, 각 선형 제어를 스케줄링 변수에 대하여 매개 변수화하지 않고 상황에 따라 적절한 제어를 선택하여 해당 제어기로 스위칭한다는 점이 다르다. 이러한 스위칭 제어기는 다수의 선형 제어기와 적절한 제어를 선택

하기 위한 스위칭 알고리즘으로 구성되며, 게인 스케줄링과 비교했을 때 매개 변수화 과정이 필요없다는 간편함이 있고, 이론적인 접근이 용이하다는 장점이 있어 근래에 많은 연구가 이루어지고 있다[2]-[4]. 한편 이 방법의 단점으로는 선형 제어기의 수를 적게 하였을 경우에는 모델링 오차와 선형 제어기 사용에 따른 오차로 인해서 스위칭이 일어났을 때 전체 시스템의 과도응답특성이 좋지 않다는 점을 들 수 있다.

한편 IMC(Internal Model Control) 구조는 일반적인 제어 시스템과는 달리 페루프 내에 플랜트의 공칭 모델(nominal model)이 내장되어 있는 구조를 가지고 있으며, 선형 IMC의 경우에 현대 제어이론에서 중요하게 거론되는 내부 안정성(internal stability), 안정 강인성(stability robustness), 성능 강인성(performance robustness), H^∞ 노음 최소화등이 모두 고려된 조직적인 설계 방법으로 특히 공정제어(process control) 분야에서 많은 연구가 이루어지고 있다[5]. 이 설계법의 장점은 직관적으로 이해하기 쉬운 구조와 제어기 설계의 용이함등이며, 특히 스위칭 제어기 설계시 스위칭 알고리즘을 구현하기가 쉽다는 점을 들 수 있다.

IMC 구조에서 제어기 스위칭 설계법을 이용하여 비선형 플랜트를 제어한 예로는 다중 모델/제어기 IMC(MMC-IMC ; Multiple Model/Controller IMC)가 있으며 [6], 시뮬레이션 결과 set point가 크게 변하는 경우에도 발산하지 않고 출력이 입력을 잘 추종함을 보였다. 그러나 과도응답 특성은 그리 만족스러운 편은 아니었으며, 이는 설계법 자체의 문제가 아니라 제어기 스위칭 방법이 갖는 본질적인 한계에서 기인한 것이었다. 본 논문에서는 제어기 스위칭 방법이 갖는 두 가지 한계인 모델링 오차와 선형 제어기의 사용에 따른 성능을 보완하기 위하여 MMC-IMC에 재귀신경망(RNN ; Recurrent Neural Network)을 사용하여 전체적인 제어 성능을 향상시키는 방법을 제

접수일자 : 2001. 1. 30., 수정완료 : 2001. 4. 30.

오원근, 조성언, 소지영 : 순천대학교 정보통신공학부(owg@sunchon.ac.kr/chose@sunchon.ac.kr/drsojy@sunchon.ac.kr)

※ 본 논문은 1998년도 순천대학교 공과대학학술재단 학술연구비의 지원을 받아 연구되었습니다.

안하였다. 제안된 방법에는 2개의 RNN이 사용되었으며, 하나는 모델링의 정확성을 높이기 위한 것이고 다른 하나는 선형 제어기의 성능을 향상시키기 위한 신경망이다.

본 논문의 구성은 다음과 같다. 2장에서는 일반적인 비선형 IMC 설계법과 IMC 구조에서의 스위칭 제어기 설계법에 대해서 설명하였고, 3장에서는 RNN의 구조와 온라인 학습법에 대한 내용이며, 4장은 RNN을 사용한 MMC-IMC 제어기의 성능 향상법이고, 5장은 모의실험, 6장은 결론으로 구성되어 있다.

II. IMC(Internal Model Control) 구조와 다중 모델/제어기 IMC

1. IMC 구조와 동작

1.1 IMC의 구조 및 특징

IMC 제어계의 일반적인 구조를 그림 1에 나타내었다. 여기에서 P 는 제어 대상인 비선형 플랜트, \hat{P} 는 플랜트의 공칭(nominal) 모델, \hat{Q} 는 공칭 제어기, 그리고 F 는 강인성 필터(robustness filter)이며, $Q = \hat{Q}F$ 의 관계가 있다. 이 시스템의 동작을 입력력 표현을 사용하여 기술하면 다음과 같다.

$$\begin{aligned} z &= (P - \hat{P})Qe \\ e &= r - z \\ y &= PQe = P\hat{Q}Fe \end{aligned} \tag{1}$$

이때 $P, \hat{P}, \hat{Q}, F: L_{pe} \rightarrow L_{pe}$ 는 causal, L_p -안정인 비선형 연산자이고 $r, e, u, y, \hat{y}, z \in L_{pe}$ 라 가정한다. 여기에서 L_{pe} 는 확장 L_p -공간(extended L_p -space)을 의미한다.

이와 같은 IMC 구조의 가장 큰 특징은 제어계에 플랜트의 모델이 포함되어 있으며, 플랜트와 모델간의 차이인 $z = P - \hat{P}$ 가 피드백된다는 것이다. 따라서 완벽한 모델링이 이루어진 이상적인 경우 즉, $P = \hat{P}$ 인 경우에는 피드

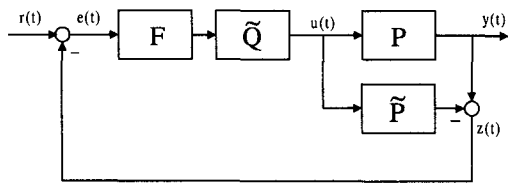


그림 1. 일반적인 IMC 구조.
Fig. 1. General IMC structure.

백 신호 $z=0$ 가 되므로 전체 IMC 시스템은 개루프(open loop)처럼 동작하게 되고 이때 (1)은 다음과 같이 쓸 수 있다.

$$y = PQr = P\hat{Q}Fr \tag{2}$$

1.2 IMC 제어기 설계법

(2)에서 $PP^{-1} = I$ (I 는 단위 연산자)와 같이 정의되는 P 의 우측 역(right inverse) 연산자 P^{-1} 를 구하여 이를 제어기 \hat{Q} 로 선택하고, $F = I$ 로 놓으면 (2)는 $y = r$ 이 되어 완벽한 입력 추종 성능 즉, 완전제어(perfect control)를 달

성할 수 있게 된다. 그러나 이와 같은 역모델제어기(model inverse controller)는 일반적으로 대단히 큰 제어기 이득을 필요로 하며, 또한 $P = \hat{P}$ 와 같이 완벽한 모델을 얻지 못했을 경우에 전체 시스템이 불안정하게 될 수 있다는 점 등으로 인하여 실제로 구현하기는 힘들다[5][7].

이러한 단점들은 역 모델 제어기 \hat{Q} 에 이득을 조절할 수 있는 필터를 덧붙여서 해결이 가능하며, 이러한 필터의 조건은 다음과 같이 구할 수 있다. 먼저 $P \neq \hat{P}$ 가 아닌 경우에 전체 시스템이 안정성을 유지하는 조건은 소이득정리(small gain theorem)[8]를 IMC 구조에 적용하여 다음과 같이 구할 수 있다[6][7].

$$\gamma_p(P - \hat{P})\gamma_p(Q) < 1 \tag{3}$$

여기에서 $\gamma_p(\cdot)$ 는 L_p -이득(L_p -gain)을 의미한다. 이 식은 전체 시스템의 안정성을 유지하기 위한 모델링 오차와 제어기 Q 사이의 tradeoff 관계를 나타내고 있다. 즉, 모델링 오차가 큰 경우에는 제어기의 이득이 작아야 하고, 모델링 오차가 작으면 큰 제어기의 이득이 가능하다는 것이다. 그러므로 제어기의 이득은 최대 모델링 오차를 고려해서 설계되어야 하며, 이때 제어기의 이득은 강인성 필터 F 를 이용해서 조절하게 된다. 즉, (3)에 $Q = \hat{Q}F$ 의 관계를 대입하면

$$\begin{aligned} \gamma_p(P - \hat{P})\gamma_p(\hat{Q}F) < 1 \\ \Rightarrow \gamma_p(P - \hat{P})\gamma_p(\hat{Q})\gamma_p(F) < 1 \end{aligned} \tag{4}$$

가 되므로 (4)를 만족하도록 필터 F 를 설계하면 모델링 오차에 강인한 제어기를 설계할 수 있다. 이상을 종합한 전체적인 제어기 설계 절차는 다음과 같다.

- 1단계 : 비선형 플랜트 P 의 공칭 모델 \hat{P} 를 구한다.
- 2단계 : 공칭제어기를 $\hat{Q} = \hat{P}^{-1}$ 와 같이 설계한다.
- 3단계 : 강인성 필터 F 를 (4)를 만족하도록 설계하여 공칭제어기에 덧붙인다.

이 설계 과정에서 어려운 점은 역 연산자를 구하는 문제와 강인성 필터 F 의 구체적인 형태를 설정하는 것으로, 선형 IMC의 경우 이 문제는 비교적 간단하게 해결할 수 있어 조직적인 설계 절차가 잘 정립되어 있으나[5], 비선형 IMC의 경우에는 특수한 경우가 아니고는 해석적으로는 해결하기 힘들기 때문에 IMC 구조를 비선형 플랜트에 적용하는데 난점이 되고 있다.

2. 다중 모델/제어기 비선형 IMC [6]

비선형 플랜트가 비교적 좁은 영역에서 동작하거나 또는 비선형성이 크지 않은 경우에는 비선형 제어기 대신에 선형 제어기를 사용하여 제어 시스템을 설계하는 방법도 자주 사용되는 방법이다. 그러나 비선형 플랜트가 넓은 영역에서 동작하거나 큰 비선형성을 갖는 경우에는 하나의 선형 제어기로는 제어가 불가능한 경우가 많으며, 이 경우에는 비선형 제어기를 사용하거나 게인 스케줄링(gain scheduling)과 같이 여러개의 선형 제어기를 사용하여야 한다. 다중 모델/제어기를 이용한 비선형 IMC (MMC-IMC; Multiple Model /Controller IMC)는 게인 스케줄링

처럼 여러개의 선형 IMC 제어기와 스위칭 알고리즘을 이용하여 비선형 IMC를 설계하는 방법으로써 비교적 간단한 설계 과정을 통해서 비선형 플랜트의 제어가 가능하다.

MMC-IMC의 구조를 그림 2에 나타내었다. 여기에서 $\{\hat{p}_i\}, i=1, \dots, N$ 는 비선형 플랜트 P 의 N 개의 동작점 (operating point)에서 얻은 선형 모델의 집합이며, $\{q_i\}, i=1, \dots, N$ 는 이에 대응하는 선형 IMC 제어기들이다. 이때 각각의 동작점에서 구한 선형 IMC 제어기는 플랜트가 해당 동작점 근방에서 운전되고 있을때는 전체 시스템을 안정화시키는 강인 제어기이다. 이때 하나의 선형 IMC 제어기 \hat{p}_i, q_i 가 담당하는 비선형 플랜트 P 의 동작 범위는 (3)을 이용하여 다음과 같은 식으로 표현할 수 있다.

$$\gamma_2(P - \hat{p}_i) < \gamma_2^{-1}(q_i) \quad (5)$$

여기에서 (5)를 만족하는 P 의 동작 영역을 R_i 라하고 P 의 전체 동작 영역을 $R(P)$ 라 하면 각각의 선형 IMC 제어기로 제어가 가능한 영역을 합하면 P 의 전체 동작 영역을

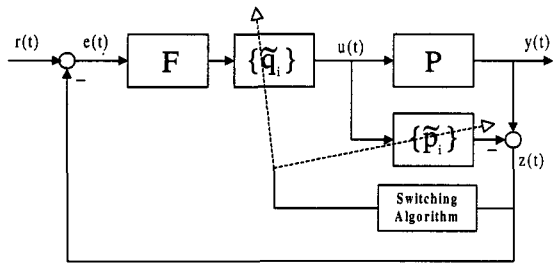


그림 2. 다중 모델/제어기 IMC 구조.
Fig. 2. MMC-IMC structure.

다음 수 있도록 하여야 한다. 즉,

$$R(P) = \bigcup_{i=1}^N R_i \quad (6)$$

이 성립하여야 하며, 이는 비선형 플랜트의 모든 동작 영역에 대해서 P 를 안정하게 제어하는 적어도 하나의 선형 IMC 제어기가 존재함을 의미한다. 하지만 실제로 제어기를 설계할 경우에 이 조건이 만족되었는지 아닌지를 해석적으로 알기는 힘들며, 모의실험을 통하여 선형 제어기의 수를 선택하거나 비교적 충분한 수의 선형 IMC 제어기를 사용함으로써 만족될 수 있다.

운전중에는 이러한 선형 IMC 제어기 $\{q_i\}, \{p_i\}$ 중 하나를 선택하여 사용한다. 이러한 선택은 현재 비선형 플랜트의 동작 상태가 속해 있는 영역 R_i 를 찾아내어 해당되는 \hat{p}_i, q_i 로 스위칭하는 스위칭 알고리즘을 이용하여 하게된다. 이러한 알고리즘에서 이상적으로는 스위칭시에 플랜트의 현재 동작 영역을 찾아내기 위해서 각 \hat{p}_i, q_i 에 대해서 (5)를 계산해야 하나, 이를 실시간으로 계산하는 것은 힘든 일이므로 근사적인 방법을 사용하여 P 의 현재 상태와 가장 '가까운' 선형 모델 \hat{p}_i 를 찾아내는 방법을 이용하였다.

사용한 방법은 기본적으로 플랜트와 각 모델의 출력을 비교하여 그 차이의 자승 오차가 가장 적은 모델을 선택하는 것이며, 이때 현재의 값만을 기준으로 하는 것은 잡음에 의한 오류가 생길 수 있으므로 과거의 값을 고려하여 다음과 같이 추정함수 $J_i(t)$ 를 설정하였다.

$$J_i(t) = \alpha \int_0^t e^{-\lambda(t-\tau)} z_i^2(\tau) d\tau + \beta z_i^2(t), \quad i=1, \dots, N \quad (7)$$

여기에서 $\alpha, \beta, \lambda \geq 0$ 이며, 각각 과거의 오차, 과거 오차의 중요도, 현재의 오차에 대한 가중치를 의미한다. 그러나 (7)을 그대로 사용하는 것은 잡음에 의한 채터링이 일어날 수 있으므로 최종적인 알고리즘은 다음과 같이 히스테리시스 스위칭 알고리즘을 사용하였다. 스위칭될 선형 제어기의 번호를 출력하는 히스테리시스 스위칭 함수 S_h 는 다음과 같다.

$$S_h = \begin{cases} i, & \text{if } J_i(t) \leq J_j(t) + h \\ j, & \text{if } J_i(t) > \min_{j, i \neq j} J_j(t) + h \end{cases} \quad (8)$$

이러한 스위칭 제어기는 잘 정립된 선형제어기 설계법을 그대로 이용하며 비선형 제어기를 직접적으로 구하지 않아도 되기 때문에 설계가 비교적 용이하다는 장점이 있다. 그러나 선형 모델과 비선형 플랜트 사이의 모델링 오차가 존재하게 되며, 선형 제어기의 사용에 따른 성능의 저하, 특히 과도 응답 특성이 좋지 않게 된다는 단점이 있다.

III. RNN 학습 알고리즘

본 논문에서는 MMC-IMC의 과도 응답 성능을 향상시키기 위해서 2개의 RNN을 사용하였다. RNN은 다층 신경망(Multilayer Neural Network)과는 달리 그림 3과 같이 되먹임(feedback) 구조로 되어 있으며, 내부적으로 동역학(dynamics)을 가지고 있다. 따라서 이러한 구조의 신경망은 동역학을 갖는 시스템에 다층 신경망보다는 더 적합할 것으로 생각되나 안정성 문제, 적절한 학습법의 부재등으로 많이 실용화되지는 않고 있다. 특히 학습 방법의 경우에 제어 시스템에 적용하기 위해서는 온라인 학습이 가능하면서도 계산량이 적당해야 하지만, 현재 알려진 BPTT (Backpropagation Through Time), RTRL(Real-Time Recurrent Network)등의 방법은 계산량이 많거나 온라인 학습이 불가능하다는 등의 단점이 있어, 제어 시스템에 응용하기에는 어려운 점이 있다. 본 논문에서는 이러한 점을 고려해서 학습법으로 [9]에서 제안한 방법을 사용하였다. 이 방법은 최적 제어 문제에 기반을 둔 학습 알고리즘으로 비교적 적은 계산량으로 온라인 학습이 가능하다는 장점이 있다.

1. RNN의 구조와 동작

본 논문에서 사용한 RNN의 구조는 그림 3과 같다. 여기에서 뉴런의 수를 N , 입력의 수를 M 이라고 하면, i 번째 뉴런의 동작을 나타내는 식은 다음과 같다.

$$x_i(k+1) = f \left[\sum_{j=0}^{M+N} w_{ij} z_j(k) \right], \quad i=1, \dots, N \quad (9)$$

여기에서 $f(\cdot)$ 는 시그모이드(sigmoid) 함수이고, $x_i(k)$ 는 시간 k 에서 i 번째 뉴런의 상태변수, w_{ij} 는 뉴런 x_j 와 x_i 사이의 무게값(weight), 그리고 $z_i(k)$ 는 다음과 같이 정의되는 값이다.

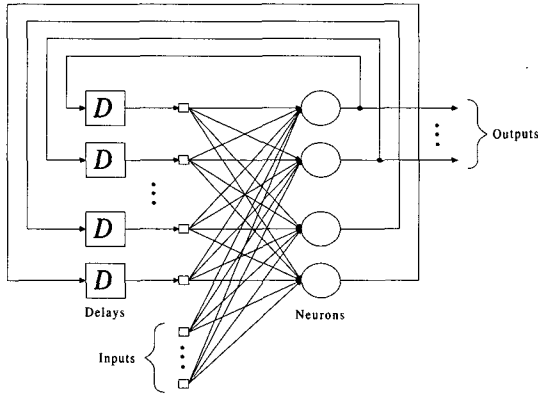


그림 3. RNN 구조.
Fig. 3. The RNN structure.

$$z_i(k) = \begin{cases} 1 & i=0 \\ u_i(k) & i \in [1, M] \\ x_{i-M}(k) & i \in [M+1, M+M] \end{cases} \quad (10)$$

이때 $u_i(k)$ 는 신경망에 들어오는 i 번째 외부입력이며 첫번째 요소 1은 bias 입력을 의미한다.

2. 학습 알고리즘

사용한 학습 알고리즘은 온라인 학습의 가능 여부와 실시간 학습에 따른 계산량을 고려하여 선택하였다. RNN의 학습법으로 널리 알려진 BPTT는 온라인 학습이 불가능하며, 실시간 학습이 가능하다고 알려진 RTRL은 많은 계산량을 필요로 하기 때문에 뉴런의 수가 많은 경우 제어 시스템에 적용하기 힘든 방법이다. 사용한 방법은 최적제어문제와 변분법을 이용한 방법으로 온라인 학습이 가능하며, RTRL 보다 적은 계산량을 필요로 한다. 이 방법은 RNN을 비선형 플랜트로, 신경망의 무게값(weight)을 제어입력으로 간주하여 신경망의 학습 문제를 최적제어문제로 변형한다. 그리고 이 문제에 변분법을 적용하여 최적의 무게값을 구한 다음, 최종적으로 이를 온라인 학습이 가능하도록 국소 구간에서 연속적으로 최적 무게값을 찾아가는 방식이다.

구체적인 학습 방법을 간단하게 요약하면 다음과 같다. 먼저 성능 지수(performance index)를 다음과 같이 정의한다.

$$J(k) \triangleq \frac{1}{2} \sum_{i=0}^k e_i^2(k) + \frac{1}{2} \sum_i \sum_j \alpha_{ij} w_{ij}^2 \quad (11)$$

(11)에서 $e_i(k) = d_i(k) - x_i(k)$ 로 원하는 출력과 신경망의 현재 출력간의 오차를 의미하며, α_{ij} 는 0과 1사이의 값으로 오차와 무게값의 크기 사이의 가중치를 의미한다. 이때 (9)를 비선형 플랜트로, 그리고 무게값 w_{ij} 를 플랜트의 입력으로 간주하고 (11)을 최소화하는 입력(즉, w_{ij})을 변

분법을 이용하여 구한다. 그러나 이렇게 변분법을 이용하여 구한 최적의 무게값은 온라인 학습에 적합하지 않은 형태를 하고 있어 다음 알고리즘과 같이 국소구간에서 최적값을 찾아가도록 알고리즘을 변형하였다. 최종적인 학습 방법을 정리하면 다음과 같다.

RNN 학습 방법[9]

Step 1 : 신경망의 무게값 w_{ij} 를 초기화한다.

Step 2 : 입력을 받아 신경망의 출력과 오차를 계산하고 버퍼에 z_i 와 e_i 를 저장한다.

Step 3 : 저장된 z_i 와 e_i 를 이용하여 다음과 같이 과거 T시간 동안의 정상상태 조건의 노름(norm)을 계산한다.

$$\sum_{l=k-T+1}^k \left\| \frac{\partial H^l}{\partial W} \right\| = \sum_{l=k-T+1}^k \sum_i \sum_j \left\{ \frac{\partial H^k}{\partial w_{ij}} \right\}^2 \quad (12)$$

여기에서 $\partial H^k / \partial w_{ij}$ 는 시간 k 에서의 정상상태 조건으로써 다음과 같이 정의된다.

$$\frac{\partial H^k}{\partial w_{ij}} = \lambda_i(k+1)z_j(k) + \alpha_{ij}w_{ij}(k) \quad (13)$$

$$\lambda_i(l) = \begin{cases} -e_i(l) & , l=k \\ \sum_j f_j' w_{ji} \lambda_j(l+1) - e_i(l) & , l=k-T+1, \dots, k-1 \end{cases} \quad (14)$$

Step 4 : 만약 $\sum_{l=k-T+1}^k \left\| \frac{\partial H^l}{\partial W} \right\| < E_{max}$ 이면 학습을 끝낸다. 그렇지 않으면 다음과 같이 무게 값을 갱신하고 Step2로 가서 다음 시간의 입력을 받는다.

$$w_{ij}^{new} = w_{ij}^{old} - \eta \sum_{l=k-T+1}^k \frac{\partial H^l}{\partial w_{ij}} \quad (15)$$

Step 5 : Step 2에서 Step 4를 정상상태 조건이 만족될 때까지 계속 반복된다.

IV. RNN을 이용한 MMC-IMC의 과도 특성의 향상

1. MMC-IMC의 성능 저하

MMC-IMC는 비선형 플랜트 P 와 선형 모델 $\{\hat{p}_i\}$ 간의 차이에 의해 생기는 모델의 부정확성과, 선형 제어기 $\{\hat{q}_i\}$ 를 비선형 플랜트의 제어에 사용하기 때문에 생기는 오차로 인한 성능의 저하요인을 가지고 있다. 이를 수식으로 표현하면 다음과 같다. 우선 스위칭 알고리즘에 의해 \hat{p}_i, \hat{q}_i 가 선택되었다고 했을 때 출력 y 는 다음과 같다.

$$y = P\hat{q}_i F(r-z) \quad (16)$$

여기에서 비선형 플랜트 P 를 다음과 같이 선형부와 비선형부를 분리할 수 있다고 가정하자.

$$P = P_L + P_N \approx \hat{p}_i + P_N \quad (17)$$

여기에서 P_L 은 플랜트 P 의 선형부이고 P_N 은 비선형부이

다. 이때 선형부 P_L 은 선형모델 \hat{p}_i 가 잘 근사하고 있어 $P_L \approx \hat{p}_i$ 이라 가정하고, $\hat{p}_i^{-1} = \tilde{q}_i$ 의 관계를 이용하면 (16)은 다음과 같이 쓸 수 있다.

$$\begin{aligned} y &= (P_L + P_N) \tilde{q}_i F(r - z) \\ &= P_L \tilde{q}_i F r - P_L \tilde{q}_i F z + P_N \tilde{q}_i F(r - z) \\ &\approx F r - F z + P_N \tilde{q}_i F(r - z) \end{aligned} \quad (18)$$

(18)의 첫 번째 항은 이상적인 IMC 구조의 입출력 관계 식이고, 두 번째 항은 비선형부로 인한 모델링 오차, 그리고 세 번째 항은 선형제어기를 사용함으로써 생기는 오차와 모델링 오차가 복합되어 있는 항이다. 이러한 오차 항들의 값이 커지게 되면 전체 제어 시스템의 성능은 저하되게 되며, 특히 스위칭 된 후 setpoint 로의 정착시간이 길어지거나 오버슈트가 커지는 등 과도응답 특성이 좋지않게 되는 결과를 낳는다.

이러한 비선형성으로 인한 오차를 없애기 위한 방법으로 MMC-IMC에 2개의 신경망을 삽입하는 방법을 제안하였다. 제안된 방법의 전체적인 구조는 그림 4와 같다. 두 개의 신경망은 각각 독립적으로 동작하며, 모델링의 정확도를 높이기 위한 신경망 N_m 과 선형 제어기를 보조하여 전체적인 제어 성능을 높이기 위한 신경망 N_c 로 구성되어 있다.

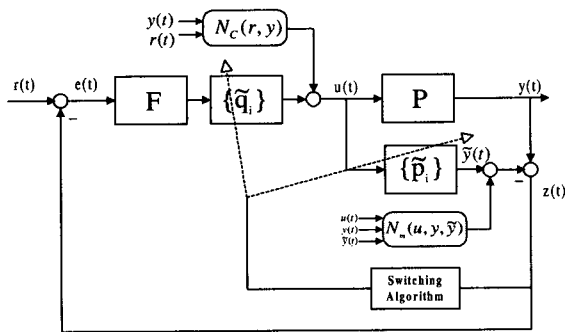


그림 4. 신경망이 삽입된 MMC-IMC.
Fig. 4. MMC-IMC with RNN.

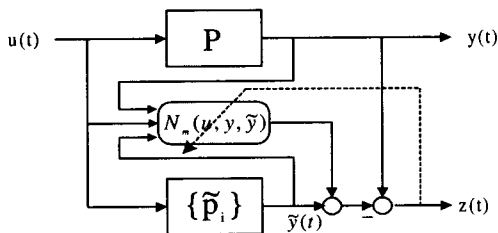


그림 5. 모델 신경망 N_m .
Fig. 5. Model network N_m .

2. 모델 신경망 N_m

이 신경망은 모델의 정확성을 높이기 위해서 사용된 것으로서 그림 5와 같이 제어 루프에 삽입된다.

이 때 $\phi(N_m)$ 을 신경망 N_m 의 출력이라 하면 출력오차

z 는 다음과 같다.

$$\begin{aligned} z &= y - (\hat{y} + \phi(N_m)) \\ &= Pu - (\hat{p}_i u + \phi(N_m)) \end{aligned} \quad (19)$$

(19)에서 플랜트와 모델간의 오차 $z=0$ 이 되기 위해서 N_m 은 $y - \hat{y} = Pu - \hat{p}_i u$ 를 출력해야 하므로, N_m 의 학습 목표로 $z = y - \hat{y}$ 를 설정하였다. 신경망의 구성은 신경망을 모델링 문제에 적용할 경우에 주로 사용하는 직-병렬 (series-parallel) 형태로 구성하였으며, 신경망의 입력으로는 비선형 플랜트와 선형 모델의 입력과 출력인 u, y, \hat{y} 를 인가하였다. 학습이 잘 이루어져 모델링 오차가 거의 없어진 경우에 출력 y 는 (18)에 $z \approx 0$ 을 대입하여 다음과 같이 구할 수 있다.

$$y \approx Fr + P_N \tilde{q}_i Fr \quad (20)$$

여기에서 모델 신경망 N_m 에 의해서 모델링 오차 z 에 의한 성능 저하 요인은 없어졌으나, 아직도 선형 제어기를 사용하는데 따른 오차(2번째 항)가 남아있는 상태이며 이를 없애기 위해 또 하나의 신경망 N_c 를 사용하였다.

3. 제어 신경망 N_c

이상적인 IMC의 경우 전체 제어 시스템의 입출력 관계 식은 $y = Fr$ 이 되어야 한다. 그러나 (20)에서와 보듯이 모델 신경망 N_m 을 사용하여 모델링 오차에 의한 성능저하는 제거하였으나, 선형 제어기를 사용하는데 따른 성능저하 요인은 남아있는 상태이다. 이를 보정하기 위하여 제어기 신경망 N_c 를 그림 6과 같이 연결하였다. 모델 신경망 N_m 의

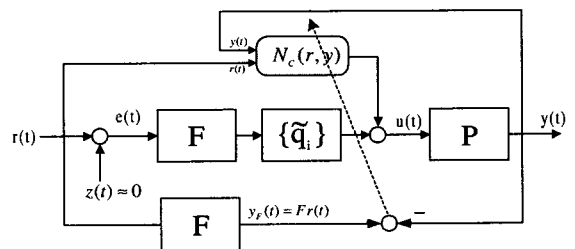


그림 6. 제어 신경망 N_c .
Fig. 6. Control network N_c .

사용으로 피드백 신호 $z(t) \approx 0$ 이라고 가정하면 (20)에서와 같이 전체 시스템은 근사적으로 개루프와 같이 동작하게 된다. 이때 제어 신경망 N_c 의 목표는 전체 시스템이 이상적인 경우인 $y = Fr$ 과 같이 동작하도록 하는 것이므로 필터 F 의 출력과 시스템의 출력 $y(t)$ 사이의 오차 $Fr - y$ 를 학습신호로 설정하여 신경망을 학습시켰다.

V. 모의실험

제안한 방법을 n 개의 동일한 크기의 상호 간섭이 없는 탱크(noninteracting tank)의 수위제어문제[10]에 적용하였다. 동역학 식은 다음과 같다.

$$\begin{aligned} \dot{h}_1(t) &= -\frac{c}{A} h_1^{1/2}(t) + \frac{1}{A} u(t) \\ \dot{h}_j(t) &= -\frac{c}{A} h_j^{1/2}(t) + \frac{c}{A} h_{j-1}^{1/2}(t), \quad j=2, \dots, n \\ y(t) &= h_n(t) \end{aligned} \quad (21)$$

여기에서 입력 $u(t) > 0$ 은 첫 번째 탱크로의 유입량, 출력 $y(t)$ 는 n 번째 탱크의 수위, 상태변수 $h_j(t)$ 는 j 번째 탱크의 수위. $A > 0$ 은 탱크의 단면적, 그리고 $c > 0$ 은 출력 흐름 저항(outflow resistance)을 의미한다. 이 플랜트에 상수 입력 u_o 를 인가했을 경우 setpoint는 $h_{i,o} = u_o^2/c^2$, $y_o = u_o^2/c^2$ 이며, (21)을 u_o 에 대해서 선형화한 식은 다음과 같다.

$$\begin{aligned} \dot{h}_{\delta i}(t) &= -K_0 h_{\delta i}(t) + A^{-1} u_{\delta}(t) \\ \dot{h}_{\delta i}(t) &= -K_0 h_{\delta i}(t) + K_0 u_{\delta, i-1}(t), \quad i=2, \dots, n \\ y_{\delta}(t) &= h_{\delta n}(t) \end{aligned} \quad (22)$$

여기에서 $K_0 = c^2/(2Au_o)$ 이고, 새로운 상태변수는 다음과 같다.

$$\begin{aligned} u_{\delta}(t) &= u(t) - u_o \\ h_{\delta i}(t) &= h_i(t) - u_o^2/c^2 \\ y_{\delta}(t) &= y(t) - u_o^2/c^2 \end{aligned} \quad (23)$$

이와 같은 플랜트에 대해서 하나의 선형 IMC제어기만을 사용한 경우, MMC-IMC를 사용한 경우, 그리고 MMC-IMC에 신경망을 사용한 경우에 대해서 기준 입력 $\kappa(t)$ 를 점점 크게 변화시켜 가면서 이에 따른 출력을 모의 실험 하였으며 $A=0.5$, $c=1$, $n=3$ 를 사용하였다. 단독으로 사용한 선형 제어기는 동작점 $u_o=3$ ($y_o=9$)에서 선형화한 모델을 이용하여 설계하였고, MMC-IMC를 구성하기 위한 선형 IMC 모델/제어기는 $u_o=3, 5, 7, \sqrt{70}$ 인 4가지 경우에 대해서 일반적인 선형 IMC 제어기 설계 절차 [5]에 따라 설계하였으며, 설계시 강인성 필터는 $F(s) = 1/(s+1)^4$ 를 공통으로 사용하였다. 여기에서 상수 입력 3, 5, 7, $\sqrt{70}$ 은 각각 setpoint $y_o=9, 25, 49, 70$ 에 해당되는 값이며, 각 setpoint에서 설계된 모델/제어기는 편의상 1, 2, 3, 4와 같이 번호를 붙였다.

과도 응답 특성을 개선하기 위해 사용한 신경망은 모델 신경망 N_m 의 경우 뉴런 수 13개의 RNN을 사용하여 학습율은 0.002로 학습하였으며, 제어 신경망 N_c 는 뉴런 수 17개의 RNN을 사용하여 학습율 0.02로 학습시켰다.

그림 7~9에 모의실험 결과를 나타내었다. 그림은 입력 $\kappa(t)$ 를 9에서 25, 45, 65로 점점 큰 계단 입력으로 변화시켜가면서, 하나의 선형 IMC제어기를 사용한 경우와 MMC-IMC제어기를 사용한 경우, 그리고 MMC-IMC에 신경망을 사용한 경우의 출력 $y(t)$ 를 보인 것이다. 그림에서 보듯이 $\kappa(t)$ 의 변화량이 커짐에 따라 선형 IMC 제어기를 하나만 사용한 경우에는 출력 특성이 점점 나빠지다가 결국 변화량이 커지는 $\kappa(t)=65$ 인 경우에는 출력이 수렴하지 못하고 진동함을 알 수 있다. 한편 MMC-IMC 제어기는 변화량이 큰 경우에도 setpoint를 추종하나, 입력의 변화량이 커질수록 과도 응답 특성은 점점 좋지 않게 되

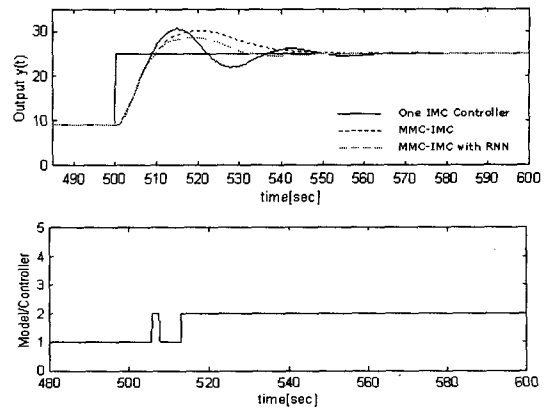


그림 7. Setpoint가 9에서 25로 변화한 경우의 플랜트 출력과 스위칭 추이.

Fig. 7. Plant output and switching sequence when setpoint changes from 9 to 25.

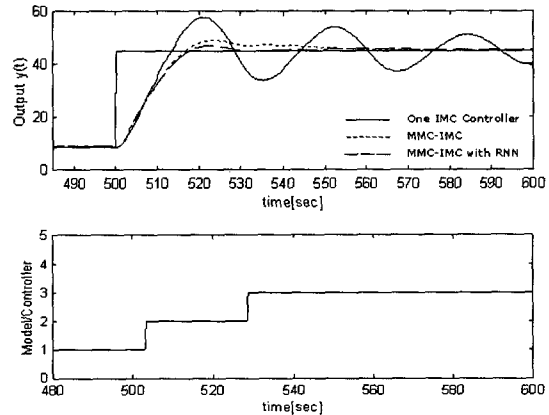


그림 8. Setpoint가 9에서 45로 변화한 경우의 플랜트 출력과 스위칭 추이.

Fig. 8. Plant output and switching sequence when setpoint changes from 9 to 45.

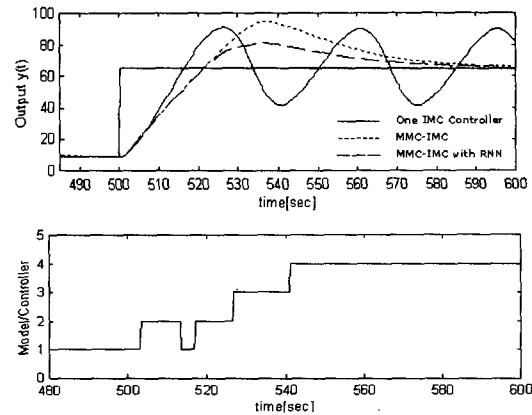


그림 9. Setpoint가 9에서 65로 변화한 경우의 플랜트 출력과 스위칭 추이.

Fig. 9. Plant output and switching sequence when setpoint changes from 9 to 65.

며, RNN을 사용한 제어기가 가장 좋은 과도응답 특성을 보임을 알 수 있다.

특히 $r(t)$ 가 9에서 65로 변화한 그림 9의 경우에 최종적으로 setpoint $y_0=70$ 에서 설계된 4번 모델/제어기로 스위칭이 일어나는 것을 볼 수 있다. 즉, 요구하는 setpoint는 65인데 스위칭 알고리즘에 의해서 그와 가장 가까운 모델/제어기인 4번으로 스위칭이 일어난 것이다. 이 경우에 모델/제어기가 설계된 setpoint인 70에서 벗어나서 전체 시스템이 동작하기 때문에 MMC-IMC의 과도 응답이 좋지 않은 것을 알 수 있다. 그러나 MMC-IMC에 신경망을 사용한 제한한 방법은 더 나은 과도응답특성을 보이고 있으며, 이는 IV장에서 설명한 대로 신경망이 모델링 오차와 비선형성을 감소시키고 있기 때문이다.

VI. 결론

IMC 구조에서 제어기 스위칭 기법을 이용한 MMC-IMC 방법의 과도 응답 특성을 개선하는 방법을 제안하였다. 제어기 스위칭 기법은 여러개의 선형 제어기를 미리 설계해 놓고서 비선형 플랜트의 동작 상황에 가장 적합한 제어기 하나를 선택하여 사용하는 방법으로써, 직접 비선형 제어기를 설계하는 것보다는 설계가 용이한 장점이 있으나 모델링 오차와 선형 제어기의 사용에 따른 성능의 저하 요인이 있으며, 특히 과도 응답 상태가 좋지 않게 된다는 단점이 있다.

본 논문에서는 IMC 구조에 적용된 스위칭 제어 시스템에 2개의 RNN을 사용하여 과도 응답 상태를 개선하는 방법을 제안하였다. 2개의 RNN은 각각 모델링 오차를 줄이기 위한 용도와 선형 제어기를 보조하기 위한 용도로 사용되었으며 시뮬레이션 결과 단순히 스위칭 제어기만을 사용한 경우에 비해 과도 응답 특성이 좋아짐을 확인할 수 있었다.



오 원 근

1967년 10월 6일생. 1989년 한양대학교 전자통신공학과 졸업. 동대학원 석사(1991), 동대학원 박사(1997). 1997년~현재 순천대학교 공과대학 정보통신공학부 조교수. 관심분야는 비선형시스템, 신경망, 신호처리, 마이크로프로세서 및 DSP 응용.



소 지 영

1960년 11월 20일생. 1983년 연세대학교 전기공학과 졸업. 동대학원 석사(1985), 동대학원 박사(1990). 1990년~현재 순천대학교 공과대학 정보통신공학부 부교수. 관심분야는 센서재료, 전자세라믹재료.

참고문헌

- [1] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, 36, pp. 1401-1425, 2000.
- [2] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems*, pp. 59-70, Oct., 1999.
- [3] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Trans. on Automatic Control*, vol. 42, no. 2, pp. 171-187, 1997.
- [4] Edited by R. Murray-Smith and T. A. Johansen, *Multiple Model Approaches to Modeling and Control*, Taylor & Francis, 1997.
- [5] M. Morari and E. Zafiriou, *Robust Process Control*, Prentice Hall, 1989.
- [6] 오원근, 서병설, "다중 모델, 제어기, 스위칭을 이용한 비선형 플랜트의 IMC 제어기 설계," *대한전자공학회지*, 33 권, 11 호, pp. 22-30, 11, 1996.
- [7] C. G. Economou, M. Morari, and B. O. Palsson, "Internal model control. 5. extension to nonlinear systems," *Ind. Eng. Chem. Process Des. Dev.*, 25, pp. 403-411, 1986.
- [8] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice Hall, 1993.
- [9] Won-Geun Oh and Byung-Suhl Suh, "An on-line learning algorithm for recurrent neural networks using variational methods," *Proc. of the 40th MWSCAS*, pp. 659-662, 1997.
- [10] W. J. Rugh, "Design of nonlinear PID controllers," *AIChE Journal*, vol. 33, no. 10, pp. 1738-1742, Oct., 1987.



조 성 언

1966년 7월 11일생. 1989년 한국항공대학교 통신정보공학과 졸업. 동대학원 석사(1991), 동대학원 박사(1997). 1997년~현재 순천대학교 공과대학 정보통신공학부 조교수. 관심분야는 무선통신, 디지털 통신, 신호처리.