

퍼지 논리를 이용한 슬라이딩 모드 제어기의 인자 자동 튜닝

Auto-Tuning of Sliding Mode Control Parameters Using Fuzzy Logic

류 세 희, 박 장 현

(Se-Hee Ryu and Jahng-Hyon Park)

Abstract : Sliding mode control guarantees robustness in the presence of modeling uncertainties and external disturbances. However, this can be obtained at the cost of high control activity that may lead to chattering. As one way to alleviate this problem a boundary layer around sliding surface is typically used. In this case the selection of controller gain, control bandwidth and boundary layer thickness is a crucial problem for the trade-off between tracking error and chattering. The parameter tuning is usually done by trial-and-errors in practice causing significant effort and time. An auto-tuning method based on fuzzy rules is proposed in this paper. In this method tracking error and chattering are monitored by performance indices and the controller tunes the design parameters intelligently in order to compromise both indices. To demonstrate the efficiency of the proposed method a mass-spring translation system and a robotic control system are simulated and tested. It is shown that the proposed algorithm is effective to facilitate the parameter tuning for sliding mode controllers.

Keywords : sliding mode control, chattering, controller gain, control bandwidth, boundary layer thickness, auto-tuning, fuzzy logic

I. 서론

슬라이딩 모드 제어는 모델링 오차를 보상해 줄 수 있는 강인한 제어 방법이지만 채터링 현상을 유발할 수 있다[1]. 이 현상은 심한 잡음과 함께 시스템의 제어 정밀도 저하, 기계 부품의 마모 등을 유발하기 때문에 가능한 한 회피되어야 한다[2]. 이를 위한 연구로는 연속화 방법[3][4], 적응 슬라이딩 모드 제어 방법[5][6], 퍼지 슬라이딩 모드 제어 방법[7][8] 등이 시도된 바 있다.

연속화 방법에서는 슬라이딩 평면 주위에 경계층을 두고, 그 내부에서 적당한 연속 함수로써 불연속적인 제어 입력을 근사화하여 채터링을 방지한다. 이 경우 제어기 이득, 제어 대역폭, 경계층 두께 등과 같은 제어기 인자 값들의 설정이 매우 중요한데, 이 작업은 해석적으로 하기에는 어려운 면이 있다. 반면에 적응 슬라이딩 모드 제어 방법에서는 실시간으로 시스템 모델식을 보정하고, 이에 따라 오차 보상을 위한 제어 이득을 감소시키므로써 채터링을 방지한다. 그러나 이 방법은 모델식의 보정을 위해 시스템에 대한 많은 사전 정보를 필요로 한다. 한편 퍼지 슬라이딩 모드 제어 방법은 제어 입력을 퍼지 규칙으로부터 결정함으로써 채터링을 방지한다. 이 경우 퍼지 규칙 및 멤버십 함수의 선정이 매우 난해한데, 이를 위하여는 유전 알고리즘, 다구치 방법 등과 같은 최적화 알고리즘이 시도된 바 있다[9]. 하지만 이 방법들은 많은 계산량과 계산 시간을 필요로 하기 때문에 실제의 제어 문제에 적용하는 데는 한계가 있다.

이상에서 언급된 바와 같이 각각의 채터링 방지 방법은 고유한 단점을 갖는데, 본 논문에서는 그 중 연속화 방법의 문제점인 인자 튜닝 작업을 효율적으로 수행하기 위한 알고

리즘을 제시하고, 궁극적으로 채터링이 없는 슬라이딩 모드 제어를 가능하게 하고자 한다. 일반적으로 인자 튜닝 작업은 모델링 되지 않은 시스템 동특성을 고려하여 수행되어야 한다. 그러나 시스템에는 구조적 진동 모드, 시간 지연, 추종 궤적의 크기 및 주파수 등 매우 다양한 동적 성분들이 존재하므로[1][10], 이 작업은 실제에 있어서는 시행착오 과정을 통해서 이루어진다. 여기에서 제안되는 방법은 제어기로 하여금 시스템 응답 특성을 실시간으로 관찰하고 그 결과에 따라 인자값을 지능적으로 튜닝하도록 하는 것이다. 본 논문의 후반부에서는 제안된 튜닝 알고리즘을 적용한 슬라이딩 모드 제어기로써 1자유도 병진 시스템의 궤적 제어 시뮬레이션 및 2관절 로봇 매니퓰레이터의 궤적 제어 실험을 각각 수행하고, 그 결과를 토대로 제시된 방법의 효율성 및 타당성을 검증하고자 한다.

II. 슬라이딩 모드 제어기 설계

본 논문에서의 슬라이딩 모드 제어기는 연속화 방법을 적용하여 구성되었다. 이 경우 일반적인 n 차 시스템에 대하여 경계층 내에서의 슬라이딩 변수 s 의 미분치는 다음과 같이 전개된다[1].

$$s = -b \delta^{-1} k \frac{s}{\phi'} + (f - b \delta^{-1} \hat{f}) + (1 - b \delta^{-1}) (-x_d^{(n)} + (n-1)\lambda \hat{x}^{(n-1)} + \dots + \lambda^{n-1} \hat{x}) \quad (1)$$

여기에서 f , b 는 각각 시스템 동력학, 제어 이득을 나타내는 함수들이고, \hat{f} , δ 는 각 함수들의 추정치를 의미하며, λ 는 제어 대역폭, ϕ' 는 경계층 두께이다. 위의 식으로부터 s 는 모델 오차가 입력인 1차 필터의 출력으로 간주될 수 있다. 그러므로 개념적으로 오차 동력학의 구조는 그림 1과 같이 두 개의 블록을 이용하여 표현될 수 있다. 여기에서 두 번째 블

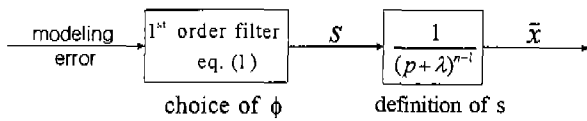


그림 1. 폐회로 오차 동역학 구조
Fig. 1. Structure of the closed-loop error dynamics.

특은 s 의 정의로부터 유도된 것으로서 λ 의 대역폭을 갖는 필터로 간주할 수 있다. 그러므로 만약 ϕ' 를 아래의 (2)와 같은 관계가 성립하도록 튜닝한다면, 두 개의 블록은 각각 λ 의 대역폭을 갖는 고주파 필터가 된다. 여기에서 만약 λ 값을 시스템의 모델되지 않은 동특성의 주파수보다 작게 선택하면 동특성이 자극되는 것을 막을 수 있고, 그 결과로서 채터링이 회피된다.

$$\frac{b \hat{b}^{-1} k}{\phi'} = \lambda \tag{2}$$

한편 이 경우 변형된 경계층 두께 ϕ 를 도입하면 제어기 이득과 튜닝 인자들 사이에는 다음과 같은 관계가 성립한다.

$$k = \lambda \phi \tag{3}$$

여기에서, $\phi = \phi' \hat{b} b^{-1}$

그리고 슬라이딩 모드 제어를 수행할 경우, 최종적인 시스템 출력의 유효 제어 오차의 크기는 다음과 같이 표현된다[1].

$$\varepsilon = \frac{b \hat{b}^{-1} \phi}{\lambda^n} \tag{4}$$

III. 제어 시스템 및 인자 자동 튜닝 알고리즘 구현

1. 제어 시스템 구성

본 논문에서 제안하는 제어 시스템의 구조는 그림 2에 도시된 바와 같다. 제어기는 제어를 수행하면서 일정 시간 간격마다 시스템의 제어 성능을 관측하여, 이를 바탕으로 제어기 인자값들을 퍼지 규칙을 이용하여 튜닝한다.

2. 제어 성능 관측

현재의 시스템 제어 성능을 관측하여 제어기에게 알려주기 위해서 제어 정확도 및 채터링의 정도를 나타내는 성능 지표들을 다음과 같이 각각 정의하였다.

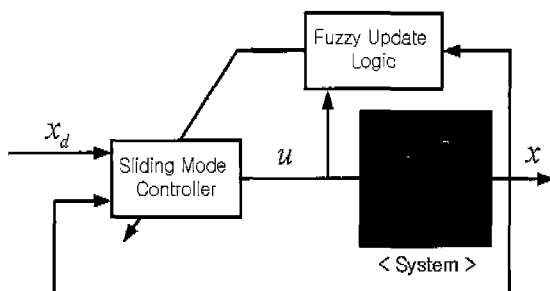


그림 2. 제어 시스템 구성도.
Fig. 2. Configuration of control system.

$$E = \frac{C_e \int_T (x - x_d)^2 dt}{\int_T |x_d| dt} \tag{5}$$

$$C = \frac{C_c \int_T \left| \frac{d^2 x}{dt^2} \right| dt}{T \tau_{max}} \tag{6}$$

여기에서 T 는 튜닝 시간 간격, τ_{max} 는 액추에이터의 최대 출력, C_e , C_c 는 각각 비례 상수이다. 위의 (5)의 E 는 제어 정확도를 나타내기 위한 성능 지표이다. 여기에서 분모는 시간 T 동안의 기준 입력 크기의 총합을 의미하고 분자는 제어 오차 크기의 제곱의 총합을 의미한다. 이는 오차 크기가 기준 입력의 크기에 상대적으로 평가되어야 함을 반영한 것으로 E 값이 작을 수록 정확한 제어가 이루어진 것으로 평가될 수 있다. 그리고 (6)의 C 는 시간 T 동안의 액추에이터 출력의 2계 도함수의 평균값을 액추에이터의 최대 출력값으로 나눈 채터링 판별 지표이다. 이는 채터링의 정도가 액추에이터의 성능 한계값을 기준으로 평가되어야 함을 반영한 것으로 그 값이 작을 수록 액추에이터 출력이 완만한 변화를 갖게 되어, 채터링 현상이 작은 것으로 평가될 수 있다. 한편 위에서 사용된 비례 상수 C_e , C_c 의 값은 성능 지수 계산에 있어서 가중치의 역할을 하며, 요구되는 제어 성능에 따라 조정된다. 예를 들어 고정밀도를 요구하는 제어에서는 C_e 값을 상대적으로 크게 설정해 준다.

3. 퍼지 규칙을 이용한 인자 튜닝

인자 튜닝은 어떤 성능 지표를 최소화하는 인자값을 탐색하는 작업으로 생각할 수 있다. 여기에서는 퍼지 규칙을 이용한 인자 튜닝 방법을 논하고자 한다.

그림 3은 어떤 성능 지표 y 값을 최소화하는 인자 x 값을 튜닝에 의해 찾는 과정을 도시한 것이다. 이 경우 x 값의 튜닝은 다음과 같이 표현할 수 있다.

$$x_{i+1} = x_i + \Delta x_i \tag{7}$$

여기에서 Δx_i 는 x 값 x_{sol} 에 보다 근접시킬 수 있는 값으로 선택해야 하는데, 이를 위해 다음과 같은 퍼지 규칙들을 이용할 수 있다.

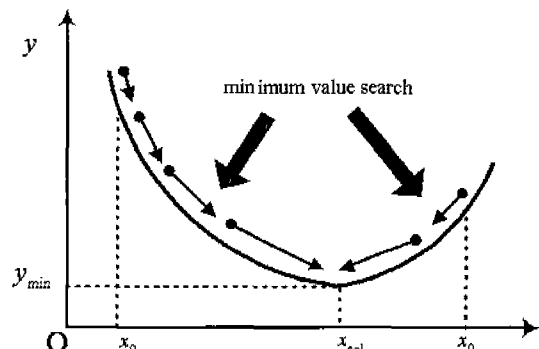


그림 3. 최소값 탐색 과정.
Fig. 3. Search process of the minimum value.

“만약 $\frac{\Delta y}{\Delta x}_{i-1}$ 값이 큰 음수이면, Δx_i 를 작은 양수로 택한다”

“만약 $\frac{\Delta y}{\Delta x}_{i-1}$ 값이 작은 음수이면, Δx_i 를 큰 양수로 택한다”

“만약 $\frac{\Delta y}{\Delta x}_{i-1}$ 값이 영이면, Δx_i 를 영으로 택한다”

“만약 $\frac{\Delta y}{\Delta x}_{i-1}$ 값이 작은 양수이면, Δx_i 를 큰 음수로 택한다”

“만약 $\frac{\Delta y}{\Delta x}_{i-1}$ 값이 큰 양수이면, Δx_i 를 작은 음수로 택한다”

위에서 사용된 $\frac{\Delta y}{\Delta x}_{i-1}$ 는 이전 단계에서의 x 값 튜닝폭과 이에 의한 y 값 변화량을 이용하여 계산되며, 이는 곡선상의 한 점에서의 기울기를 의미한다. 이 값은 부호와 크기를 갖는데, 퍼지 규칙에서는 각각을 이용하였다. 우선 부호의 경우 $\frac{\Delta y}{\Delta x}_{i-1}$ 값의 부호와 반대 부호를 갖도록 Δx_i 를 택하고 (7)을 이용하여 x_{i+1} 값을 구하면, 이 값은 y 값을 이전보다 감소시키는 x 값이 된다. 그리고 $\frac{\Delta y}{\Delta x}_{i-1}$ 값의 크기는 이전 단계에서의 x 값 튜닝에 의한 y 값 변화의 민감도를 의미하므로 그 크기가 크면 다음 단계에서의 x 값 튜닝의 폭을 작게 하고, 반대의 경우에는 크게 함으로써 해의 수렴 속도를 보다 증가시킬 수 있다. 위와 같은 퍼지 규칙을 이용한 x 값 튜닝을 반복하면 y 값을 최소화하는 x_{opt} 를 탐색할 수 있다. 이 방법은 최적화 방법 중에서 경사 강하법(steepest descent method)[11]과 유사하다.

4. 3 단계 인자 자동 튜닝 알고리즘

여기에서 튜닝하고자 하는 슬라이딩 모드 제어기 인자는 k, λ 및 ϕ 이다. 본 논문에서는 실제의 사용자가 슬라이딩 모드 제어기의 설계 인자들을 튜닝하는 경우를 모사하여 3 단계의 인자 자동 튜닝 알고리즘을 구성하였고, 그 순서도 및 과정을 그림 4, 5에 각각 도시하였다.

제1단계 튜닝은 제어를 시작한 후의 첫 작업 단계이므로 이 과정에서 우선적으로 시스템 안정성을 보장할 필요가 있다. 이를 위해서 이 단계에서는 ϕ 값을 고정하고, λ 값을 튜닝하면서 E 값을 유효 오차 범위 ϵ_1 이하로 감소시켜 시스템 안정성을 확보하도록 한다. 이 경우 k 값은 (3)에 의해 자동적으로 튜닝된다. 이를 수식으로 표현하면 다음과 같다.

$$\phi_{i+1} = \phi_i \tag{8}$$

$$\lambda_{i+1} = \lambda_i + \Delta\lambda_i \tag{9}$$

$$k_{i+1} = \lambda_{i+1} \cdot \phi_{i+1} \tag{10}$$

여기에서 채터링은 고려되지 않으며, 최종 결과로서 획득되는 k 값은 시스템 안정성 보장을 위해 다음 단계에서도 그대로 유지되어야 한다.

제2단계 튜닝은 시스템 안정성은 보장하면서 채터링을 방지하기 위한 작업 단계이다. 이 과정에서 가장 중요한 인자는 λ 이다. 왜냐하면 시스템 안정성 또는 제어 정확도 향상을 위해서는 λ 값을 크게 선택하는 것이 바람직하지만, 그

값이 지나치게 크면 모델링되지 않은 동특성을 자극하여 채터링이 발생되기 때문이다. 그러므로 여기에서는 제어 오차 지표 E 는 시스템 안정성을 위해 유효 오차 범위 ϵ_2 이하로 유지하면서 시스템의 채터링 오차 지표 C 를 유효 채터링 허용 한계 δ 이하로 감소시키기 위한 인자 값들을 탐색하도록 하였다. 이 과정에서 사용되는 ϵ_2 는 1단계에서 사용된 ϵ_1 보다 약간 큰 값으로 설정함으로써 제어 정확도가 다소 악화되더라도 안정성이 보장되는 한, 채터링을 감소시키기 위한 인자 튜닝을 허용한다. 구체적으로 제2단계 튜닝은 전 단계에서 결정된 k 값을 고정하고, ϕ 값을 튜닝하면서 C 값을 유효 오차 범위 δ 이하로 감소시키는 과정으로 정의될 수 있고, 이 경우 λ 값은 (3)에 의해 자동적으로 튜닝된다. 이를 수식으로 표현하면 다음과 같다.

$$k_{i+1} = k_i \tag{11}$$

$$\phi_{i+1} = \phi_i + \Delta\phi_i \tag{12}$$

$$\lambda_{i+1} = \frac{k_{i+1}}{\phi_{i+1}} \tag{13}$$

이 경우 최종적으로 획득되는 λ 값은 모델링되지 않은 시스템 동특성을 자극하지 않는 최대 제어 대역폭을 의미한다. 이 값은 제어기로 하여금 채터링을 방지하면서도 보다 강인하고 정확한 제어를 수행하도록 하기 때문에 다음 단계에서도 그대로 유지되어야 한다.

마지막으로 제3단계 튜닝은 채터링이 방지되는 가운데 제어 정확도를 보다 향상시키고자 하는 작업 단계이다. 이 과정은 전 단계에서 결정된 λ 값을 고정하고, 채터링 오차

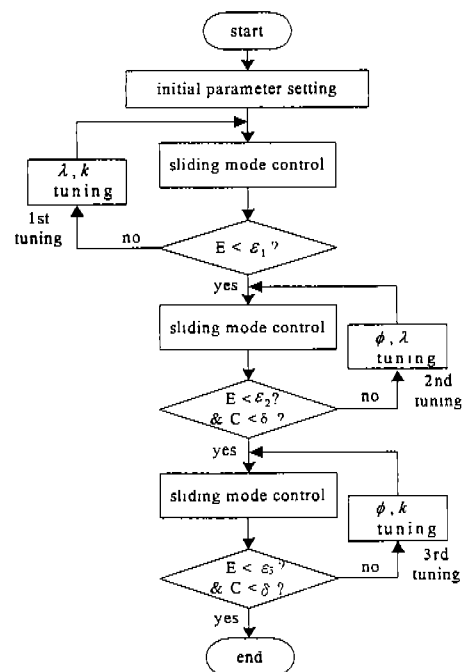


그림 4. 인자 튜닝 순서 흐름도.
Fig. 4. Flowchart of parameter tuning scheme.

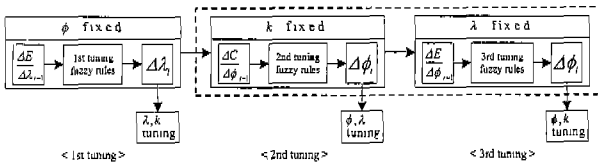


그림 5. 3단계 인자 자동 튜닝 과정.
Fig. 5. 3 stage parameter auto-tuning process.

지표 C 를 유효 채터링 허용 한계 δ 이하로 유지하면서, 시스템의 제어 오차 지표 E 를 유효 오차 범위 ϵ_3 이하로 감소시키기 위한 ϕ 값을 튜닝하는 작업으로 정의될 수 있다. 이 경우 k 값은 (3)에 의해 자동적으로 튜닝된다. 이 과정에서 성능 지표 평가 기준으로 사용되는 ϵ_3 는 안정성 보장을 위해 사용된 ϵ_1 보다 약간 작은 값으로 설정함으로써 제어 정확도를 더욱 향상시키는 인자 튜닝을 수행하도록 한다. 제3 단계 튜닝 작업은 최종 제어 오차가 (4)와 같이 ϕ 에 비례하는 점에 착안하여 시스템 안정성이 보장되는 한 가능한 한 ϕ 값을 감소시킴으로써 보다 정확한 제어를 수행할 수 있도록 한 것이다. 이 경우 각 인자의 튜닝을 위한 수식은 다음과 같다.

$$\lambda_{i+1} = \lambda_i \tag{14}$$

$$\phi_{i+1} = \phi_i + \Delta\phi_i \tag{15}$$

$$k_{i+1} = \lambda_{i+1} \cdot \phi_{i+1} \tag{16}$$

5. 튜닝 단계별 퍼지 규칙 설정

여기에서는 각 단계별 튜닝을 위한 퍼지 규칙들을 논하고자 한다. 튜닝 제1단계는 E 값을 최소화하기 위한 λ 값의 탐색, 제2단계는 C 값을 최소화하기 위한 ϕ 값의 탐색, 마지막으로 제3단계는 E 값을 최소화하기 위한 ϕ 값의 탐색으로서 정의되므로 앞서 언급했던 인자 튜닝을 위한 퍼지 규칙들을 동일하게 이용할 수 있다. 다만 퍼지 규칙의 입력으로 사용되는 민감도는 각 단계별로 $\frac{\Delta C}{\Delta\phi_{i-1}}$, $\frac{\Delta E}{\Delta\lambda_{i-1}}$, $\frac{\Delta E}{\Delta\phi_{i-1}}$ 값들을 각각 사용한다. 예로 제1단계 튜닝을 위한 퍼지 규칙의 구성은 다음과 같다.

“만약 $\frac{\Delta E}{\Delta\lambda_{i-1}}$ 값이 큰 음수이면, $\Delta\lambda_i$ 를 작은 양수로 취한다”

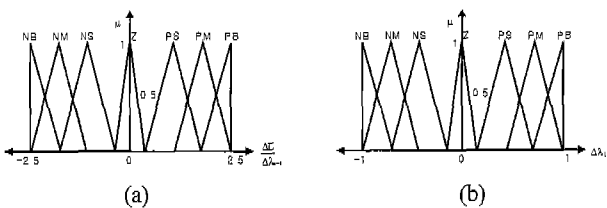


그림 6. 제1단계 튜닝을 위한 퍼지 멤버십 함수, (a) 입력 변수 : $\Delta E/\Delta\lambda_{i-1}$, (b) 출력 변수 : $\Delta\lambda_i$.
Fig. 6. Fuzzy membership functions for the first stage tuning, (a) input variable : $\Delta E/\Delta\lambda_{i-1}$, (b) output variable : $\Delta\lambda_i$.

“만약 $\frac{\Delta E}{\Delta\lambda_{i-1}}$ 값이 작은 음수이면, $\Delta\lambda_i$ 를 큰 양수로 취한다”

“만약 $\frac{\Delta E}{\Delta\lambda_{i-1}}$ 값이 영이면, $\Delta\lambda_i$ 를 영으로 취한다”

“만약 $\frac{\Delta E}{\Delta\lambda_{i-1}}$ 값이 작은 양수이면, $\Delta\lambda_i$ 를 큰 음수로 취한다”

“만약 $\frac{\Delta E}{\Delta\lambda_{i-1}}$ 값이 큰 양수이면, $\Delta\lambda_i$ 를 작은 음수로 취한다”

그리고 여기에서 사용되는 멤버십 함수는 그림 6에 도시된 바와 같은 형태로 구성된다.

IV. 예제를 통한 알고리즘 성능 검증

1. 1 자유도 병진 시스템의 궤적 제어

본 논문에서 제안된 자동 인자 튜닝 알고리즘을 적용한 슬라이딩 모드 제어기로서 1 자유도 병진 시스템의 궤적 제어를 시뮬레이션하였다.

효과적인 성능 검증을 위해서 시스템 내부에는 공진 모드가 존재하지만 엔지니어가 그것을 누락한 채 모델링을 했다고 가정하였고, 액츄에이터에 시간 지연이 0.01초 존재한다고 가정하였다.

이 경우 실제 시스템과 모델된 시스템은 그림 7에 도시된 바와 같고, 이에 대한 운동 방정식은 각각 아래의 (17), (18)과 같다.

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} -(k+k') & k' \\ k & -k' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \tau \\ 0 \end{bmatrix} \tag{17}$$

$$\ddot{x} = -\frac{k}{m}x + \frac{\tau}{m} \tag{18}$$

여기에서, x 는 시스템 출력, m_1 , m_2 , m 은 각각 질량(단, $m = m_1 + m_2$), k , k' 는 각각 스프링 상수, 그리고 τ 는 종방향 추력으로 작용되는 제어 입력을 나타낸다 이 경우 슬라이딩 모드 제어기의 제어 입력은 다음과 같다.

$$\tau = \hat{\tau} - k \cdot \text{sat}(s) \tag{19}$$

여기에서, $\hat{\tau} = kx + m(\ddot{x}_d - \lambda \dot{x})$

한편 시뮬레이션에 있어서는 5계 Dormand-Prince 적분 방법이 사용되었고, 적분 시간 간격은 0.001초로 설정되었다. 위의 문제의 경우 실제 시스템의 고유 주파수는 1Hz, 200Hz 이고, 모델된 시스템의 고유 주파수는 1Hz이다. 따라서 λ 값

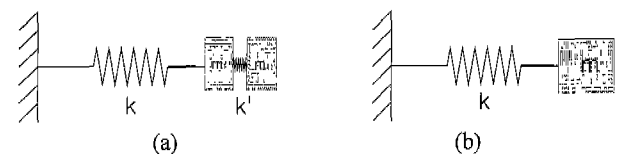


그림 7. 1 자유도 질량-스프링 시스템, (a) 실제 시스템, (b) 모델된 시스템.
Fig. 7. One d.o.f. spring-mass system, (a) real system, (b) modeled system.

이 지나치게 클 경우 모델링되지 않은 시스템 내부의 공진 모드가 자극되어 시스템이 불안정해질 수 있고, 그밖에 액츄에이터 시간 지연등도 고려되어 인자 튜닝이 수행되어야 한다. 시스템이 추종할 기준 궤적은 다음과 같이 주어졌다.

$$x_d(t) = 0.3 \sin(2\pi t) \quad (20)$$

위의 기준 입력은 주기가 1초이므로 1회 튜닝 시간 T 는 같은 값으로 설정하였고, 액츄에이터 출력 힘은 100N으로 제한하였다. 궤적 제어 시뮬레이션에 있어서 λ , ϕ 및 k 값은 오차 및 채터링 지표에 따라서 시간 T 간격으로 튜닝되었다. 한편 λ 값의 초기치는 최적값보다 크게 설정될 수도 있고, 작게 설정될 수도 있으므로 본 논문에서는 두 가지 경우를 모두 시도하였다. 제어기의 인자 튜닝 결과는 그림 8, 9에 각각 도시하였다. 그림 8에서 처음에는 시스템 안정성을 보

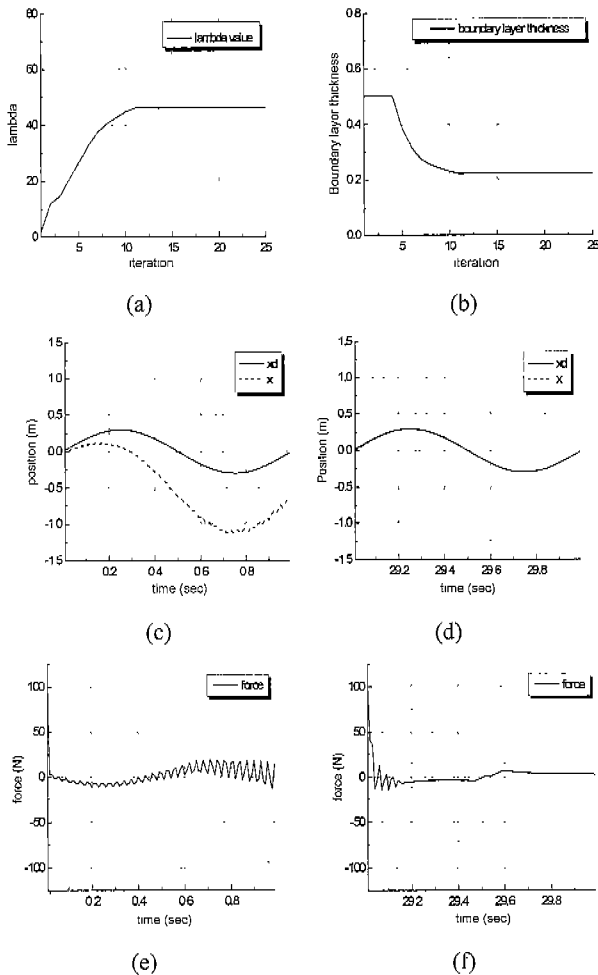


그림 8. λ 초기치가 작은 경우의 튜닝 결과, (a) 제어 대역폭 변화 과정, (b) 경계층 폭 변화 과정, (c) 튜닝 이전 궤적, (d) 튜닝 이후 궤적, (e) 튜닝 이전 액츄에이터 출력, (f) 튜닝 이후 액츄에이터 출력.

Fig. 8. Auto-tuning results for low initial λ case, (a) control bandwidth history, (b) boundary layer thickness history, (c) trajectory before tuning, (d) trajectory after tuning, (e) actuator output before tuning (f) actuator output after tuning.

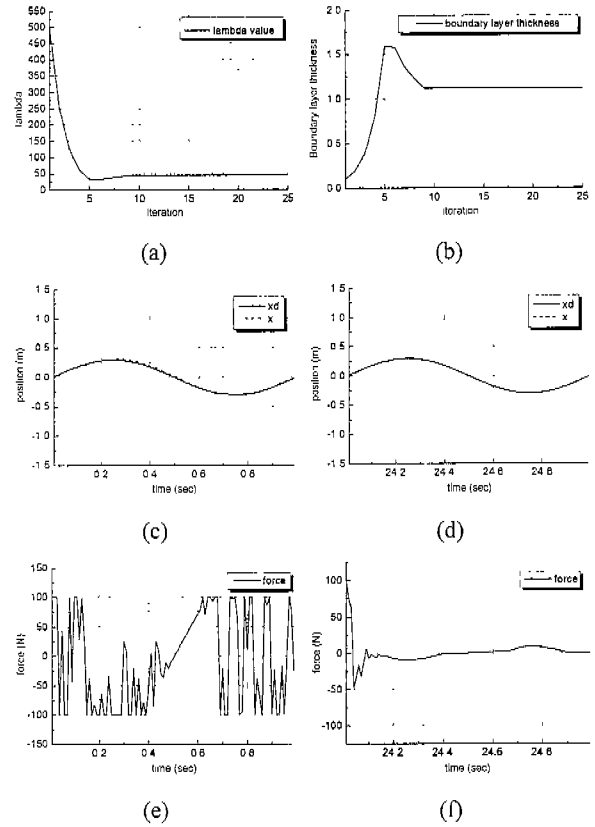


그림 9. λ 초기치가 큰 경우의 튜닝 결과, (a) 제어 대역폭 변화 과정, (b) 경계층 폭 변화 과정, (c) 튜닝 이전 궤적, (d) 튜닝 이후 궤적, (e) 튜닝 이전 액츄에이터 출력, (f) 튜닝 이후 액츄에이터 출력.

Fig. 9. Auto-tuning results for high initial λ case, (a) control bandwidth history, (b) boundary layer thickness history, (c) trajectory before tuning, (d) trajectory after tuning, (e) actuator output before tuning, (f) actuator output after tuning.

장하기 위해 ϕ 값은 고정된 채, λ 값만이 증가되어 제어기 이득을 증가시켰고, 이후에는 λ , ϕ 모두 튜닝되어 성능 지표 E , C 값을 감소시키고자 하였다. 그러나 그림 9에서는 처음부터 두 인자들이 모두 튜닝되었는데, 이는 λ 의 초기값이 매우 크게 설정되어 이미 시스템 안정 조건이 만족되었기 때문이다. 한편 튜닝 이전과 이후의 시스템 위치 궤적 및 액츄에이터 힘의 변화를 살펴 보면 그림 8의 경우, 튜닝 이전에는 λ 값이 매우 작아 채터링은 작지만 제어 오차가 매우 컸고, 튜닝 이후에는 채터링이 작으면서도 비교적 정확한 궤적 제어가 수행되었다. 그리고 그림 9의 경우에는 튜닝 이전에 λ 값이 매우 커서 제어 오차는 비교적 작았지만 채터링이 매우 심하였고, 튜닝 이후에는 채터링이 작으면서도 비교적 정확한 궤적 제어가 수행되었다. 한편 두 경우 모두 λ 값은 약 47 정도로 수렴하였고, 이로부터 제어기는 인자 초기 조건에 관계없이 인자들을 적절히 튜닝함을 알 수 있다.

기존의 연속화 방법에서는 우수한 제어 성능을 얻기 위해 많은 시행 착오를 거쳐 제어기 인자값들을 튜닝해 주어야 했지만, 본 알고리즘은 그러한 일련의 작업을 자동화, 지능화한데 그 의미가 있다. 그리고 시스템 및 외부 환경에 대

한 사전 정보나 가정없이 현재의 제어 목적에 가장 적합한 인자값들을 탐색할 수 있으므로 기존의 적응 슬라이딩 제어 방법보다 실용적이다.

2. 자유도 로봇 매니플레이터 시스템의 궤적 제어

제안된 자동 튜닝 알고리즘을 적용한 슬라이딩 모드 제어기로써 2자유도 로봇 매니플레이터 시스템의 궤적 제어 실험을 수행하였다. 사용된 시스템은 그림 10에 도시된 바와 같은 산업용 6관절 로봇이다. 본 알고리즘은 임의의 다자유도 시스템에 적용 가능하지만 여기에서는 실험의 단순화를 위하여 총 6축 중에서 제2, 3축만을 제어하고자 하였다. 한편 가변적인 시스템 외란을 주기 위하여 제4축의 브레이크를 풀어 주었고, 그 결과로서 로봇이 움직임에 따라 제2, 3축에는 각각 가변적인 외부 하중이 작용되었다. 한편 로봇의 운동 방정식은 마찰을 무시하고 다음과 같이 구성되었다.

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (21)$$

여기에서

$$\begin{aligned} H_{11} &= m_1 l_{c1}^2 + I_1 + m_2 (l_{c1}^2 s_1^2 + l_1^2 c_1^2) & C_{11} &= (m_2 l_{c1}^2 - m_2 l_1^2) \sin \theta_1 \cos \theta_1 \dot{q}_1 \\ H_{12} &= m_2 l_1 l_{c2} \cos \theta_1 \cos \theta_2 & C_{12} &= -m_2 l_1 l_{c2} \cos \theta_1 \sin \theta_2 \dot{q}_2 \\ H_{21} &= H_{12} & C_{21} &= -m_2 l_1 l_{c2} \sin \theta_1 \cos \theta_2 \dot{q}_1 \\ H_{22} &= I_2 + l_{c2}^2 m_2 & C_{22} &= 0 \end{aligned}$$

$$\begin{aligned} G_1 &= (m_1 g l_{c1} + m_2 g l_1) \cos \theta_1 \\ G_2 &= m_2 g l_{c2} \cos \theta_2 \end{aligned}$$

위 시스템에 대한 슬라이딩 모드 제어 규칙은 다음과 같다.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \widehat{H}_{11} & \widehat{H}_{12} \\ \widehat{H}_{21} & \widehat{H}_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_{r1} \\ \ddot{q}_{r2} \end{bmatrix} + \begin{bmatrix} \widehat{C}_{11} & \widehat{C}_{12} \\ \widehat{C}_{21} & \widehat{C}_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \widehat{G}_1 \\ \widehat{G}_2 \end{bmatrix} - \begin{bmatrix} k_1 \cdot \text{sat}(s_1/\phi_1) \\ k_2 \cdot \text{sat}(s_2/\phi_2) \end{bmatrix} \quad (22)$$

$$\begin{aligned} \dot{q}_{r1} &= \dot{q}_{d1} - \lambda_1 (q_1 - q_{d1}) \\ \dot{q}_{r2} &= \dot{q}_{d2} - \lambda_2 (q_2 - q_{d2}) \\ s_1 &= \dot{q}_1 - \dot{q}_{r1} \\ s_2 &= \dot{q}_2 - \dot{q}_{r2} \end{aligned}$$

한편 모든 λ 및 ϕ 값들은 각각 동일한 값을 갖는다고 가정하여 튜닝 작업을 수행하였다.

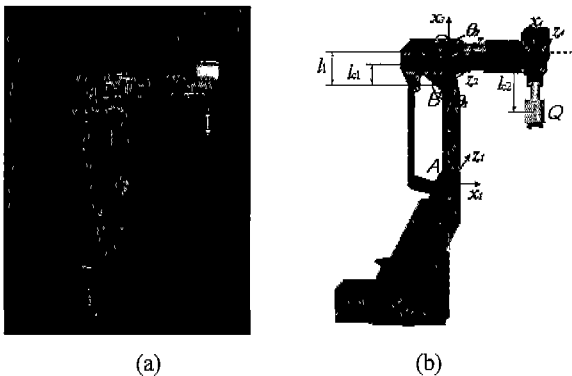


그림 10. 로봇 매니플레이터 시스템, (a) 실제 시스템, (b) 모델된 시스템.

Fig. 10. A robot manipulator system, (a) real system, (b) modeled system.

여기에서 주어진 시스템의 추종 궤적은 다음과 같다.

$$\begin{aligned} q_{d1}(t) &= 0.2 \cos(2\pi t) \\ q_{d2}(t) &= 0.4 \cos(2\pi t) \end{aligned} \quad (23)$$

위의 기준 입력은 주기가 각각 1초이므로 1회 튜닝 시간 T 는 이 값으로 설정되었고, 샘플링 타입은 1kHz로 설정되었다. 여기에서도 λ 의 초기치가 작은 경우 및 큰 경우를 각각 실험하였고, 그림 11에 그 중 초기치가 큰 경우의 인자 튜닝 과정을 도시하였다. 이 경우 초기 채터링이 지나치게 크기 때문에 λ 를 감소시키는 방향으로 튜닝이 진행되었음을 알 수 있다. 그리고 튜닝 이전과 이후의 각 관절의 제어 오차 변화 및 액츄에이터 토크 변화를 비교해보면 튜닝 이후에 채터링이 효과적으로 감소되었고, 제어 오차도 유효 오차 범위 내로 감소된 것을 알 수 있다. 한편 두 경우 모두 λ 값은 약

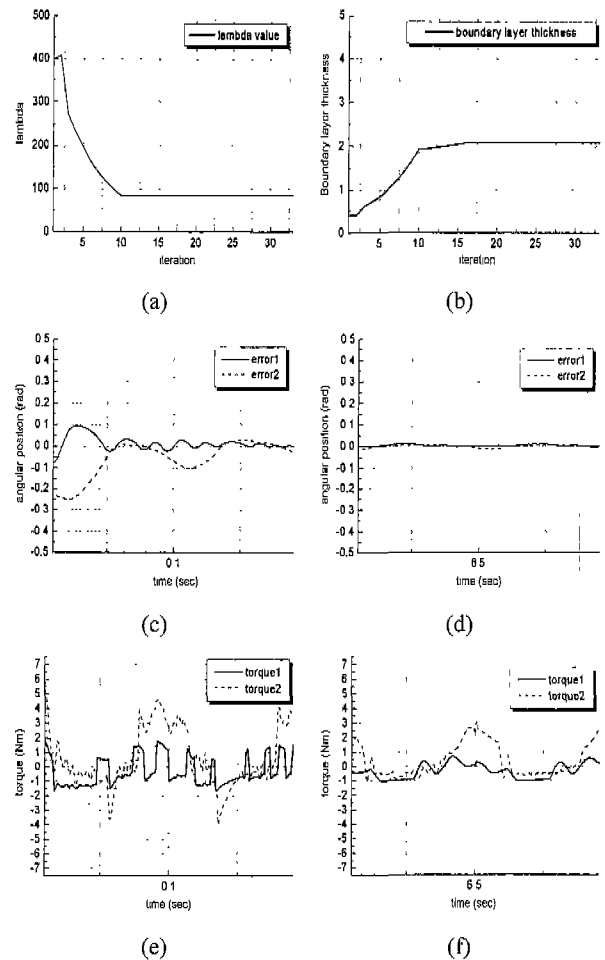


그림 11. λ 초기치가 큰 경우의 튜닝 결과, (a) 제어 대역폭 변화 과정, (b) 경계층 폭 변화 과정, (c) 튜닝 이전 궤적 제어 오차, (d) 튜닝 이후 궤적 제어 오차, (e) 튜닝 이전 액츄에이터 출력, (f) 튜닝 이후 액츄에이터 출력.

Fig. 11. Auto-tuning results for high initial λ case, (a) control bandwidth history, (b) boundary layer thickness history, (c) trajectory control error before tuning, (d) trajectory control error after tuning, (e) actuator output before tuning, (f) actuator output after tuning.

85 정도로 수렴하였고, 이로부터 제어기는 인자 초기 조건에 관계없이 인자들을 적절히 튜닝함을 알 수 있다. 결론적으로 자동 인자 튜닝 알고리즘을 적용한 슬라이딩 모드 제어기는 시스템의 동특성등을 고려하여 제어기 인자값들을 적절히 튜닝함으로써 효과적인 제어를 수행한다.

V. 결론

본 논문에서는 퍼지 규칙을 이용한 슬라이딩 모드 제어 인자의 자동 튜닝 알고리즘을 제안하였다. 이에 의하면 제어기는 시스템 제어 성능을 실시간으로 관찰하고 그 결과에 따라 설계 인자 값들을 지능적으로 튜닝한다. 이 과정에서 효과적인 시스템 제어 성능 관측을 위한 성능 지표가 도입되었고, 지능적 인자 튜닝을 위한 민감도를 이용한 퍼지 규칙들이 구성, 사용되었다. 제안된 알고리즘의 타당성 검증 을 위해서는 이를 적용한 슬라이딩 모드 제어기로써 1 자유도 궤적 제어 시뮬레이션 및 2 자유도 로봇 매니퓰레이터 시스템의 궤적 제어 실험을 각각 수행하였다. 특히 이 과정에서 다양한 형태의 모델링 오차가 시스템에 부여되었고, 슬라이딩 모드 제어기는 각각의 경우에 대하여 능동적으로 시스템의 응답 및 튜닝 과정을 고려하여 효과적으로 제어 인자들을 튜닝하여, 궁극적으로 제어 오차 및 액츄에이터 채터링면에서 만족할만한 슬라이딩 모드 제어가 이루어지도록 하였다. 이 알고리즘은 많은 시행착오가 요구되는 슬라이딩 모드 제어 인자 튜닝 작업을 효과적으로 수행하는데 적용될 수 있을 것으로 기대된다.

참고문헌

[1] Slotine, J. -J. and Li, W., *Applied Nonlinear Control*, Prentice-Hall Inc., New Jersey, 1991.

- [2] Utkin, V., Guldner, J., and Shi, J., *Sliding mode control in electromechanical systems*, Taylor & Francis Inc., Philadelphia, 1999.
- [3] Slotine, J. -J., "Sliding controller design for non-linear systems," *International Journal of Control*, vol. 40(2), pp. 421-434, 1984.
- [4] Utkin, V., "Variable structure systems with sliding modes," *IEEE Transactions on Automatic Control*, vol. AC-22(2), pp. 212-222, 1977.
- [5] Wang, Y. -P. and Sinha, A., "Adaptive sliding mode control algorithm for a microgravity isolation system," *Acta Astronautica*, vol. 43(7-8), pp. 377-384, 1998.
- [6] Wang, W. J. and Chen, J. Y., "A new sliding mode position controller with adaptive load torque estimator for an induction motor," *IEEE Transactions on Energy Conversion*, vol. 14(3), pp. 413-418, 1999.
- [7] Lo, J. -C. and Kuo, Y. -H., "Decoupled fuzzy sliding-mode control," *IEEE Transactions on Fuzzy Systems*, vol. 6(3), pp. 426-435, 1998.
- [8] Ha, Q. P., Rye, D. C., and Durrant-Whyte, H. F., "Fuzzy moving sliding mode control with application to robotic manipulators," *Automatica*, vol. 35, pp. 607-616, 1999.
- [9] Chen, C. -L. and Chang M. -H., "Optimal design of fuzzy sliding-mode control : A comparative study," *Fuzzy Sets and Systems*, vol. 93, pp. 37-48, 1998.
- [10] Choi, S. B., Park, D. W., and Jayasuriya, S., "A time-varying sliding surface for fast and robust tracking control of second-order uncertain systems," *Automatica*, vol. 30, pp. 899-904, 1994.
- [11] Burden, R. L. and Faires, J. D., *Numerical Analysis*, PWS Publishing Company, Boston, 1993.

류 세 희

제어 · 자동화 · 학회지, 제5권, 제 8호, 참조.

박 장 현

제어 · 자동화 · 학회지, 제5권, 제8호, 참조.