

## 유전 알고리즘을 이용한 퍼지 슬라이딩 제어기 설계

### Fussy Sliding Controller Design using Genetic Algorithm

\*한종길, 유병국, 정사철, 함운철

전북대학교 전자공학과 (Tel : 0652-70-2409; Fax:652-70-2400; E-mail:Wooncul moak.chonbuk.ac.kr)

**Abstracts** In this paper, We present a fussy-siding controller dsign using genetic algorithm. We can suppress chattering and enhance the robustness of controlled system by using this controller and do that genetic algorithm can easily find out a nearly optimal fuzzy rule. performance of this controller is tested by simulation of car system with two pole.

**keywords** fuzzy logic control, fuzzy sliding mode control, genetic algorithm

#### 1. 서 론

퍼지 논리 제어기는 전통적인 제어 방법으로 해결하기 어려운 여러 가지 문제에 적용되어 뛰어난 성능을 보여 주고 있다. 그러나 퍼지 제어기를 설계하는 데에는 전통적인 제어 방식을 이용하여 설계하는 것에 비하여 여러 가지 어려움이 있다. 따라서 신경망이나 진화연산알고리즘, 유전알고리즘 등과 같은 학습알고리즘을 통하여 퍼지 논리 제어기를 자동으로 설계하고자 하는 노력이 여러 방면에서 진행되고 있다.[2,6,7,8]

진화 연산 알고리즘을 사용한 퍼지 제어기 설계의 큰 문제는 퍼지 논리 제어기의 입력변수나 입력 언어 레벨의 선형적인 증가에 따라 퍼지 규칙의 지수적인 증가를 유도하여 긴 스트링 길이와 높은 계산량의 원인이 된다.[7] 이것은 진화 연산 알고리즘을 사용하여 퍼지 논리 제어기의 최적의 퍼지 규칙과 소속함수를 찾는 데 큰 어려움이 된다. 이 문제의 해결방안으로서 제안한 퍼지 슬라이딩 모드 제어는 퍼지 논리 제어기와 슬라이딩 모드 제어기의 장점을 갖춘 알고리즘으로서 입력변수의 선형적인 증가에 대하여 퍼지 규칙의 선형적인 증가함수이다. 게다가 슬라이딩 모드에 퍼지 논리 제어기를 도입함으로써, 퍼지 슬라이딩 제어의 채터링 현상은 슬라이딩 모드 제어보다 현저하게 줄어들고, 퍼지 슬라이딩 제어의 퍼지 규칙 설계는 퍼지 논리 제어기보다 구현하기 쉽다.

두 개의 폴을 갖은 도립진자는 강한 비선형 제어 문제를 연구하기 위한 대표적인 대상으로 사용되고 있다. 대부분의 연구는 단지 한 개의 폴을 세우거나 두 개의 폴을 임의의 각도 영역에서 세우는 제어에 대하여 치중하고 있다.[9,10]

본 연구에서는 퍼지 슬라이딩 모드 제어에서 최적의 퍼지 규칙을 찾기 위해 진화 연산알고리즘을 사용하는 방법을 제안하고, 찾은 퍼지 규칙을 사용하여 제어기를 구성하여 두 개의 폴을 갖은 도립진자에 적용하여 퍼지 슬라이딩 제어기의 성능을 분석하여 본다.

본 논문의 구성은 다음과 같다. 2장에서는 비선형 방정식의 일반적인 선형화 방법을 소개하고, 3장에서는 reaching condition에 기반한 슬라이딩 제어기와 퍼지 슬라이딩 제어기를 합성하여 새로운 제어기를 제안하고 안정한 슬라이딩 함수를 설정하는 방법에 대하여 소개하고, 4장에서는 진화연산알고리즘의 최적 퍼지규칙을 찾는 과정을 소개하고 개체의 적합도함수를 제안하고, 5장에서는 시뮬레이션 하는데 사용된 두 개의 pole를 가진 도립진자의 매개변수를 정의하고 제안된 퍼지 슬라이딩 제어기의 퍼지 규칙을 탐색하여 시뮬레이션을 한다. 마지막으로 6장에서는 결론과 추후과제를 논한다.

#### 2. 비선형 방정식의 선형화

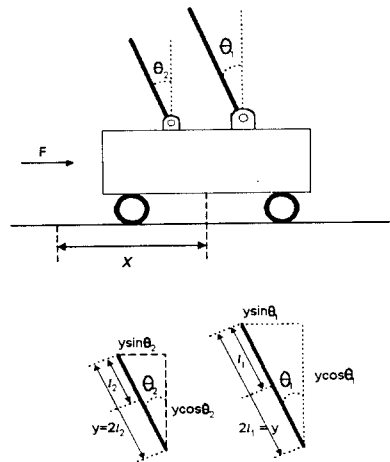


그림 1. 두개의 pole을 가진 도립진자

위와 같은 두개의 Pole를 갖은 도립진자의 동역학 방정식을 Lagrangian 방정식을 이용하여 구하면 다음과 같다.

$$\begin{aligned} \ddot{\theta}_1 &= \frac{3}{4} \frac{g \sin(\theta_1)}{L_1} + \frac{3}{4} \frac{\cos(\theta_1) \ddot{x}}{L_1} \\ \ddot{\theta}_2 &= \frac{3}{4} \frac{g \sin(\theta_2)}{L_2} + \frac{3}{4} \frac{\cos(\theta_2) \ddot{x}}{L_2} \\ \ddot{x} &= \frac{-\frac{3}{4} [m_1 \sin(\theta_1)(L_1 \dot{\theta}_1^2 - g \cos(\theta_1))] + F}{M + m_1(1 - \frac{3}{4} \cos^2(\theta_1)) + m_2(1 - \frac{3}{4} \cos^2(\theta_2))} \\ &\quad + \frac{-\frac{3}{4} [m_2 \sin(\theta_2)(L_2 \dot{\theta}_2^2 - g \cos(\theta_2))] + F}{M + m_1(1 - \frac{3}{4} \cos^2(\theta_1)) + m_2(1 - \frac{3}{4} \cos^2(\theta_2))} \end{aligned} \quad (1)$$

여기서

- $L_1$  : pole 1의 길이의 절반
- $L_2$  : pole 2의 길이의 절반
- $M$  : cart의 무게

- $m_1$ : pole 1의 무게
- $m_2$ : pole 2의 무게
- $x$ : cart의 위치
- $\dot{x}$ : cart의 속도
- $\theta_1$ : pole 1의 각도
- $\theta_2$ : pole 2의 각도
- $\dot{\theta}_1$ : pole 1의 각속도
- $\dot{\theta}_2$ : pole 2의 각속도이다.

시스템을 일반적인 불확실한 비선형 동역학 방정식으로 표현하면, 다음과 같다.

$$\dot{x}(t) = A(t, x)x(t) + B(t, x)u(t) + C(t, x)d(t) + f(t, x, u) \quad (2)$$

여기에서  $x(t) \in R^n$ 는 상태벡터이고,  $u(t) \in R^m$ 는 제어입력벡터이다.  $d(t) \in R^1$ 은 외부 잡음 벡터,  $f(\cdot): R^n \rightarrow R^n$ 는 비선형과 파라미터 불확실함을 고려한 벡터이며,  $A(t, x) \in R^{n \times n}$ ,  $B(t, x) \in R^{n \times m}$  그리고  $c(t, x) \in R^n$ 이다.  $A(t, x)$ ,  $B(t, x)$ 는 알고 있는 nominal 부분과 불확실한 부분으로 구분되어 질 수 있도록 표현될 수 있다. 즉

$$A(t, x) = A_0 + \Delta A(t, x), B(t, x) = B_0 + \Delta B(t, x) \quad (3)$$

여기에서  $\Delta A(\cdot)$ ,  $\Delta B(\cdot)$ ,  $f(\cdot)$ ,  $d(\cdot)$ 이 연속이라고 가정하고 다음과 같은 Matching condition을 고려한다.

A1. 모든  $(t, x) \in R_+ \times R^n$ 에 대해서 식(4)을 만족하는 다음과 같은 함수가 존재한다.

$$\begin{aligned} P(\cdot, \cdot): R_+ \times R^n &\rightarrow R^{m \times m} \\ Q(\cdot, \cdot): R_+ \times R^n &\rightarrow R^{m \times m} \\ R(\cdot, \cdot): R_+ \times R^n &\rightarrow R^m \\ S(\cdot, \cdot): R_+ \times R^n &\rightarrow R^{m \times 1} \\ \Delta A(t, x) &= B_0 P(t, x), \Delta B(t, x) = B_0 Q(t, x) \\ f(t, x) &= B_0 R(t, x), \Delta C(t, x) = B_0 S(t, x) \end{aligned} \quad (4)$$

위와 같은 가정 하에 모든 불확실한 항들은 한 항으로 묶여질 수 있으며, 따라서 식(13)은 다음과 같이 쓰여질 수 있다.

$$\dot{x}(t) = A_0 x(t) + B_0 [u(t) + \eta(t, x, u)] \quad (5)$$

여기에서  $\eta(t, x, u): R_+ \times R^n \times R^m \rightarrow R^m$ 는 불확실항을 나타내고  $\rho$ 가 최대값이 되도록 그 크기가 제한될 수 있다고 가정한다. 즉  $\rho = \max_{(t, x, u)} |\eta(t, x, u)|$ 이다.

### 3. 제어 범칙의 설계

#### 3.1 슬라이딩 제어기 설계

슬라이딩 함수는 다음과 같다.

$$s = Gx \in R^m \quad (6)$$

여기에서  $G \in m \times n$ 는 행렬  $GB_0$ 가 nonsingular가 되도록 하는 full rank 이어야 한다.

상태 궤적을 슬라이딩 평면에 계속 유지하게 하는  $u_{eq}$ 는 다음과 같다.

$$u_{eq} = -(GB_0)^{-1}GA_0x \quad (7)$$

$$\text{reaching condition : } s\dot{s} < -\alpha|s| < 0 \quad \alpha > 0 \quad (8)$$

슬라이딩 제어기는 식(8)의 reaching condition을 만족하여야 하는데 그에 따른 슬라이딩 제어기는 다음과 같다.

$$u_s = -(GB_0)^{-1}[GA_0x + Ks + \rho \text{sign}(s)] \quad (9)$$

#### 3.2 퍼지 슬라이딩 제어기 설계

입력변수나 입력 언어 레벨의 선형적인 증가는 퍼지 규칙의 지수적인 증가를 유도하는 문제를 해결하기 위해, 슬라이딩 함수를 퍼지화하는 슬라이딩 퍼지 제어기를 사용한다. 슬라이딩 변수에 할당된 입력 언어항은 PB, PS, PZ, NZ, NS, NB이 있고, s의 소속함수값은 상태 x로부터 슬라이딩 평면(s=0)까지 거리로 볼 수 있다. 따라서 퍼지 슬라이딩 퍼지 규칙 베이스는 다음과 같다.

$$R_j : \text{IF } s \text{ is } S_j(m_j, \delta_j), \text{ Then } u \text{ is } U_j(\theta_j)$$

여기에서 s는 입력변수이고 u는 출력변수이고,  $S_j$ 와  $U_j$ 는 각각 입력언어항과 출력언어항인 퍼지 집합이다. 비퍼지화는 입력으로 합성된 퍼지 집합의 무게 중심을 이용하는 방법인 무게 중심법은 다음과 같다.

$$u = \frac{\sum_{j=1}^N \theta_j \mu(R_j(x))}{\sum_{j=1}^N \mu(R_j(x))} \quad (10)$$

여기서  $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$ 는 출력 언어항  $U_j$ 의 중심이며, 다음 장에서 GA를 사용하여 찾는다.

#### 3.3 제어기 합성

슬라이딩 제어는 입력에서 채터링 현상이 발생하는데 이를 줄이기 위하여 경계영역부근에서 퍼지제어기를 사용함으로 채터링 현상을 줄이도록 제어기를 합성한다.

$$\begin{cases} u_s + (GB_0)^{-1}\rho(\text{sign}(s) - u_j) & |s| \leq \epsilon \\ u_s & |s| > \epsilon \end{cases} \quad (11)$$

여기서  $u_j$ 는 퍼지 제어기이고  $\epsilon$ 는 스위칭 상수이다.

#### 3.4 비선형 방정식의 선형화 및 슬라이딩 함수의 설정

시스템의 동역학 방정식을  $\theta = 0$ 에서 테일러급수 전개를 통하여 선형화하면 다음과 같다.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ gb_1 & 0 & 0 & 0 \\ 0 & gb_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_1 \\ b_2 \end{bmatrix} u$$

$$\dot{x} = Ax + Bu \quad (12)$$

여기에서

$$x_1 = \theta_1, \quad x_2 = \theta_2, \quad x_3 = \dot{\theta}_1, \quad x_4 = \dot{\theta}_2$$

$$b_1 = \frac{3}{4L_1}, \quad b_2 = \frac{3}{4L_2} \text{이다.}$$

시스템이  $\theta=0$  부근에서  $[B|A|B|A^2B|A^3B]$ 의 determinant가  $g^2 b_1^2 b_2^2 (b_2 - b_1)^2$ 이므로  $L_1 = L_2$ 인 경우에 제어 불가능하다. 선형화된 시스템이 제어 가능함으로 제어 가능한 canonical 형태로 변환할 수 있고 G를 좀더 쉽게 설정할 수 있다.

$$y = Qx \quad (13)$$

여기에서

$$Q = \begin{bmatrix} \frac{1}{b_1^2 g - b_1 b_2 g} & \frac{1}{b_2^2 g - b_1 b_2 g} & 0 & 0 \\ 0 & 0 & \frac{1}{b_1^2 g - b_1 b_2 g} & \frac{1}{b_2^2 g - b_1 b_2 g} \\ \frac{1}{b_1 - b_2} & \frac{1}{-b_1 + b_2} & 0 & 0 \\ 0 & 0 & \frac{1}{b_1 - b_2} & \frac{1}{-b_1 + b_2} \end{bmatrix}$$

식(13)에 의하여 식(12)을 다음과 같은 canonical 형태로 변환한다.

$$\dot{y} = A_1 y + B_1 u = Q A Q^{-1} y + Q B u \quad (14)$$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -b_1 b_2 g^2 & 0 & (b_1 + b_2)g & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

이 시스템 방정식을 통하여 다음과 같은 안정한 슬라이딩 함수를 잡을 수 있다

$$s = G_1 y = [c_1, c_2, c_3, 1] y \quad (15)$$

$|sI - [A_1 - B_1(G_1 B_1)^{-1} G_1 A_1]| = s^4 + c_3 s^3 + c_2 s^2 + c_1 s^1 = 0$ 의 근이 LHP상에 위치하도록  $c_1, c_2, c_3$ 를 설정한다.

$$s = G_1 y = G_1 Q x = G x \quad (16)$$

식(15)와같이 원래 시스템으로 역변환하여 안정한 슬라이딩 함수를 잡을 수 있다.

## 4. 진화 연산알고리즘

### 4.1 진화연산 알고리즘

진화연산 알고리즘에서는 생물학적인 유전인자인 염색체에 해당하는 문자열을 가지고 생물과 같은 재생산, 교배, 돌연변이를 거쳐 다음 세대의 자손을 만들어 낸다. 진화연산알고리즘으로 최적의 해를 구하는 순서는 다음과 같다.

- 단계 1 : 안정한 슬라이딩 함수의 설정
- 단계 2 : 퍼지화

퍼지제어 영역에서 슬라이딩 변수  $s$ 를 퍼지화한다.

단계 3 : 초기 퍼지 규칙 베이스

랜덤하게  $\theta_i$ 를 생성하여 초기 퍼지 규칙 베이스를 생성한다.

단계 4 : 초기 해집단

$\theta = [\theta_1, \dots, \theta_N]$ 를 코딩하여 직렬로 연결하여 하나의 개체를 만든다. 이 개체를 원하는 해집단수 만큼 랜덤하게 생성한다.

단계 5 : 평가

각 개체를 10진수로 디코딩하여 목적함수에 대한 적합도를 계산한다. 최적의 개체가 가장 큰 적합도를 가지도록 한다.

단계 6 : 복제

적자생존의 원리에 맞게 높은 적합도를 가진 개체일수록 다음세대로 복제될 확률을 높게 한다. 현재 세대에서 가장 큰 적합도를 가진 개체는 다음 세대에 항상 존재하도록 하는 엘리트 규칙을 주면 확실적인 진화 연산알고리즘의 성능을 향상시킬수 있다.

단계 7 : 교배

복제 과정에서는 새로운 개체를 형성하지 못한다. 교배 과정은 복제한 개체를 쌍으로 구성하여 개체 안에 있는 유전자 정보를 서로 교환하여 새로운 정보를 갖은 개체(후손)를 만든다.

단계 8 : 돌연변이

복제와 교배 과정은 집단의 개체들이 현재 가지고 있는 정보들을 이용하여 탐색하고 결합하는 과정이지만 돌연변이 과정은 현재 집단에 존재하지 않은 새로운 유전정보를 갖은 개체를 만드는 과정이다. 이 과정은 지역적 극값으로 향할 경우에 여기서 벗어나도록 도움을 준다.

단계 9 : 종료 상태의 조건이 만족하면 중단하고 그렇지 않으면 단계 5에서 단계 8과정을 반복한다.

## 4.2 목적함수

원하는 궤적에 수렴하는 최적의 퍼지 출력 언어항  $U_j$ 를 찾기 위해, 처음 슬라이딩 평면에 히팅한 시각  $t_h$ 로부터 최종시각  $t_f$ 까지 원하는 궤적에 수렴 정도를 나타내는 위치의 추정오차 벡터의 자승을 합한 항과 채터링의 정도를 나타내는 슬라이딩 함수의 시간 변화율을 자승한 항과 시스템의 입력의 시간 변화율을 자승한 항을 반영하도록 설정하며, 목적함수를 최소화하도록 tuning한다.

$$\text{목적함수} = \sum_{n=t_h}^{t_f/\Delta T} [v_1 \|x_i(n) - x_{id}(n)\|^2 + v_2 \|s_i(n-1) - s_i(n)\|^2 + v_3 \|u_{ii}(n-1) - u_{ii}(n)\|^2] \quad (17)$$

여기서  $\Delta T$ 는 샘플링 주기,  $v_1, v_2, v_3$ 는 양의 가중치이다. 이에 따른 적합도 함수는 다음과 같다.

$$F = \frac{1}{\text{목적함수} + 1} \quad (18)$$

## 5. 시뮬레이션 및 결과

### 5.1 도립 진자의 매개변수 및 제어기 파라미터

parameters	values	단위
$L_1$	1	m
$L_2$	0.4	m

표1. 두 개의 폴을 갖은 도립 진자의 매개변수

슬라이딩 함수는 다음과 같이 설정하였다. canonical 형태로 변환된 시스템에서 특성근이 LHP에 위치하도록  $s = [24, 26, 9, 1]y$ 을 잡았고, 이 슬라이딩 함수는 원래의 시스템에서는  $s = [24, 26, 9, 1]Qy = -10.9025x_1 - 9.161x_2 + 4.0333x_3 + 2.147x_4$ 이다. pole의 원하는 궤적은  $\theta_{1d} = 0, \theta_{2d} = 0$ 로, 초기치는  $\theta_1 = 0.2, \theta_2 = -0.1, \dot{\theta}_1 = 0, \dot{\theta}_2 = 0$ 로, 슬라이딩 제어기 파라미터  $K = 5, \rho = 1$ , 스위칭 상수( $\epsilon$ ) = 0.1로, 퍼지 제어기 파라미터는  $\delta_j = 0.03$ 로 정하였다.

### 5.2 진화 연산알고리즘의 파라미터

진화 연산 알고리즘은 해집단에서 최고의 적합도를 나타내는 개체는 다음 세대에서 항상 존재하도록 하는 엘리트 규칙을 사용하고 진화연산알고리즘에서 사용되는 파라미터는 다음과 같다. 모집단의 크기는 20개, 교배율은 0.7, 돌연변이율은 0.01 그리고 세대수는 1000이다.

슬라이딩 함수의 출력언어항의 중심  $[\theta_1, \dots, \theta_6]$ 를 진화연산알고리즘을 이용하여 tuning하기 위해 각  $\theta_j$ 를 5비트로 코딩하여 30비트 개체를 형성하였다. 그리고 퍼지 논리 제어기의 출력 언어항 중심의 탐색영역은  $[-10 \sim 10]$ 으로 주었다.

NB	NS	NZ	PZ	PS	PB
-5.238	-3.630	-1.852	1.569	3.604	5.783

표2. 출력 언어항의 중심의 추정치

### 5.3 추정된 6를 사용한 시뮬레이션

진화 연산 알고리즘을 사용하여 퍼지 논리 제어기의 출력 퍼지 집합 중심의 최적 추정결과는 표2와 같다. 그림 2는 두 개의 풀을 가진 도립진자가 원하는 각 0도에 잘 수렴하는 것을 보여주고 있으며, 그림 3.a는 슬라이딩 제어기를 사용하여 시뮬레이션할 때 제어입력이며 심한 채터링 현상을 볼 수 있으나, 3.b는 퍼지 논리 제어기의 출력 언어항의 추정값으로 시뮬레이션한 그림이며, 제어입력에서 채터링 현상이 슬라이딩 제어기를 사용한 것보다 상당히 적은 것을 볼 수 있다.

## 6. 결 론

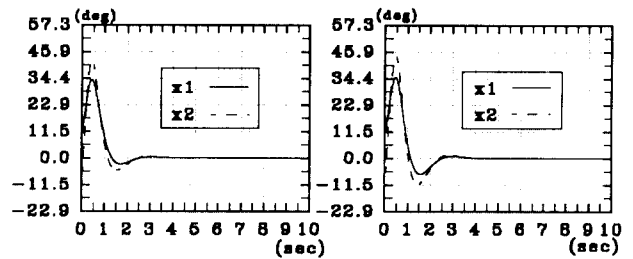
본 논문에서는 퍼지 슬라이딩 제어기의 출력 언어항을 진화 연산 알고리즘을 통하여 고찰하여 보았다. 컴퓨터 시뮬레이션을 통하여 슬라이딩 평면을 퍼지화함으로써 좀더 쉽게 진화 연산 알고리즘을 사용하여 최적의 퍼지 논리 제어기의 출력 언어항을 tuning하는데 활용될 수 있음을 알 수 있었고, 퍼지 슬라이딩 제어기를 nonminimum 시스템인 도립진자에 적용하여 채터링이 없고 원하는 궤적에 빠른 수렴을 보였고, 진화 연산알고리즘에서 좀더 빠른 수렴을 위한 연구가 진행되어야 하겠다.

## 7. 참고 문헌

[1] H.M. Tai and S.Shenoi, "Robust Fuzzy Controller," IEEE systems, Man and Cybernetics society, Vol.1, pp.86-90,1994.  
 [2] J.Kacprzyk, "Multistage Control of a Fuzzy System Using a Genetic Algorithm," Fuzz-IEEE/IFES'95, Vol, 3, pp. 1083 -1088, 1995  
 [3] J.Y.hung and J.C.James, "Variable Structure Control : A

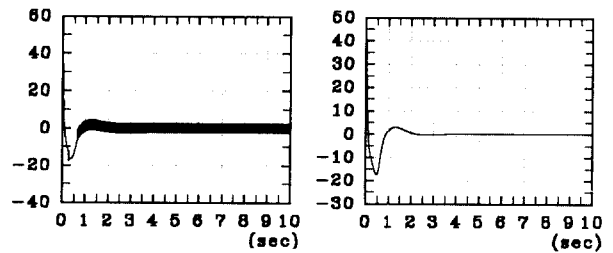
survey, "IEEE Transactions on Industrial Electronics, Vol.40, No.1, pp.2-22, 1993.

[4] J.Y.hung and J.C.James, "Variable Structure Control : A New approach, "IEEE Transactions on Industrial Electronics, Vol.40, No.1, pp.2-22, 1993.  
 [5] L.Dubois and J.Y. Dieulot and P.Borne, "Use of Fuzzy Logic in Multi-models Control," IEEE systems, Man and Cybernetics Society, Vol.1 pp.471-474, 1994.  
 [6] M.De Neyer and R.Gorez, "Use of fuzzy concepts in adaptive sliding mode control," Fuzz-IEEE/IFES'95, Vol.1, pp.443-447, 1995.  
 [7] S.C. Lin and Y.Y. Chen, "A Ga-based Fuzzy Controller with Sliding Mode, "FUZZ-IEEE/IFES'95, Vol.3, pp.1103-1100, 1995  
 [8] T.Culliere and A.Titli and Corrieu, "Neuro-fuzzy Modeling of Nonlinear Systems for Control Purposes, "FUZZ-IEEE /IFES'95, Vol.4, pp.2009-2016, 1995  
 [9] T.H. Li and C.Y. Tsai, "Parallel Fuzzy Sliding Mode Control of the Cart-Pole System," IEEE IECON, Vol.2, pp1468-1473, 1995



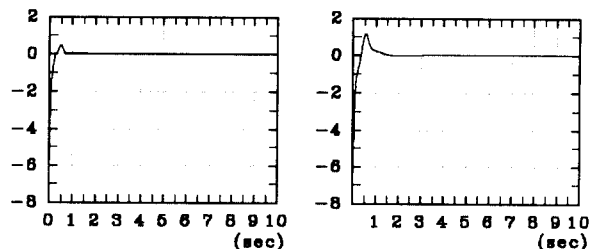
a. 슬라이딩 제어기 b. 퍼지 슬라이딩 제어기

그림2. 슬라이딩 제어기의 pole의 궤적



a. 슬라이딩 제어기 b. 퍼지 슬라이딩 제어기

그림3. 슬라이딩 제어기의 제어입력



a. 슬라이딩 제어기 b. 퍼지 슬라이딩 제어기

그림4. 슬라이딩 제어기의 스위칭 함수의 궤적