

퍼지 예측기를 이용한 비선형 일반 예측 제어기의 설계

Design of a Generalized Predictive Controller
for Nonlinear Plants Using a Fuzzy Predictor

안상철, 김용호, 권욱현

(Sang Chul Ahn, Yong Ho Kim, and Wook Hyun Kwon)

Abstract : In this paper, a fuzzy generalized predictive control (FGPC) for non-linear plants is proposed. In the proposed method, the receding horizon control is applied to the control part, while fuzzy systems are used for the predictor part. It is suggested that the fuzzy predictor is time-varying affine with respect to input variables for easy computation of control inputs. Since the receding horizon control can be obtained only with a predictor instead of a plant model, the fuzzy predictor is obtained directly from input-output data without identifying a plant model. A parameter estimation algorithm is used for identifying the fuzzy predictor. The control inputs of the FGPC are computed by minimizing a receding horizon cost function with predicted plant outputs. The proposed controller has a similar architecture to the generalized predictive control (GPC) except for the predictor synthesis method, and thus may possess inherent good properties of the GPC. Computer simulations show that the performance of the FGPC is satisfactory.

Keywords : Fuzzy, predictive, control, nonlinear, GPC

I. 서론

일반 예측 제어(Generalized Predictive Control, GPC) [3,4]는 플랜트 모델로부터 얻어진 예측기 모델을 이용하여 주어진 이동 구간 비용 함수를 최소화하는 제어이다. 이러한 일반 예측 제어는 그 좋은 성능으로 여러 가지 플랜트에 적용되어 왔다[2,5,6,11,15]. 그러나, 예측 제어 방법에서 비용 함수를 최소화시키는 문제는 플랜트의 예측기가 선형일 때 쉽게 풀리기 때문에 지금까지의 적용 플랜트는 대부분이 선형이거나 선형화된 플랜트들이었다.

비선형 플랜트에 일반 예측 제어를 적용하는 방법은 몇몇 연구에서 다루어져 왔는데[7,10,13,18], 그중 가장 기본적인 접근 방법은 플랜트의 선형화였다[10,13]. 그러나, 일반적으로 선형화된 플랜트 모델을 사용하게 되면 두 가지 문제점에 부딪치게 된다. 그 첫번째는 플랜트의 동작점의 변화에 따라 여러 개의 선형화된 모델이 필요하게 된다는 것이다. 두번째는 선형화된 모델로부터 얻어지는 출력 예측기가 미래 시점에서는 동작점이 바뀔 수 있기 때문에 미래 출력을 제대로 예측하지 못하는 경우가 있을 수 있다는 점이다. 만일 출력 예측기가 맞지 않는다면, 예측된 출력을 바탕으로 계산된 제어 입력은 플랜트를 제대로 제어할 수 없게 된다. 플랜트의 선형화와 다른 방법으로는 비선형 플랜트에 대해서 특별한 모델을 사용하는 방법들도 있다[7,18]. Zhu는 비선형 플랜트에 대해 Hammerstein 모델이라는 것을 사용하였다. 그러나, Hammerstein polynomial을 풀어서 제어 입력을 구하는 것이 쉽지 않다는 단점이 있다. Dumont은 비선형 플랜트에 대해 비선형 Laguerre 모델이라는 것을 사용하였다. 그러나, 제어 입력을 구하기 위해서는 비선형 최적화를 해야하는 어려운 점이 있다. 한편, 퍼지 시스템을 사용하여 비선형 플랜트의 예측 제어를 하는 연구도 있었다 [1,9,12]. 이 방법들에서는 플랜트에 대해서 규칙 형태의 예측기 모델을 사용하였고, 퍼지 제어 규칙을 사용하거나 궤환 선형화 제어를 사용하였다. 그러나, 이 방법들은 이동 구

간 개념을 사용하지 않았고, 제어기의 구조가 기존의 일반 예측 제어기와 다르기 때문에, 비최소위상 플랜트의 안정화 같은 일반 예측 제어의 장점을 가질 수 없다.

본 논문에서는 퍼지 이론을 이용하여 비선형 플랜트에 일반 예측 제어를 적용할 수 있는 퍼지 일반 예측 제어(Fuzzy Generalized Predictive Control) 방법을 제안한다. 이 방법에서는 기존의 일반 예측 제어의 좋은 성능을 유지하면서도 제어 입력을 쉽게 구하도록 하기 위해서 비선형 플랜트에 대해서 어파인(affine) 퍼지 예측기를 도입하고, 제어부에 이동 구간 제어를 적용한다. 따라서, 제안하는 제어기는 예측기 구성 방법을 제외하고는 일반 예측 제어기와 같은 구조를 갖는다. 이 방법을 사용하면 퍼지 예측기가 비록 시변이긴 하지만 플랜트의 미래 입력에 대해 어파인(affine) 하기 때문에 비용 함수의 비선형 최적화등을 할 필요가 없게 된다. 따라서, 제어 입력이 쉽게 구해질 수 있다는 장점을 가지게 된다.

본 논문의 2절에서는 비선형 플랜트에 대한 퍼지 예측기의 구성 방법을 제안하고, 3절에서는 퍼지 예측기의 정확도를 높이기 위해 퍼지 예측기의 파라미터를 추정하는 방법에 대해서 논한다. 4절에서는 제안된 퍼지 예측기를 이용하여 퍼지 일반 예측 제어를 적용하는 방법을 논하고, 5절에서는 시뮬레이션을 통해 제안된 퍼지 일반 예측 제어의 성능을 본다. 그리고, 6절은 본 논문의 결론이다.

II. 퍼지 예측기의 구성

이 절에서는 Wang이 제안한 퍼지 모델과 Takagi-Sugeno 퍼지 모델을 사용하여 비선형 시스템에 대한 퍼지 예측기를 구성하는 방법을 제안한다.

일반 예측 제어에서는 다음 형태의 이차 비용 함수를 사용한다.

$$J = \sum_{j=N_1}^{N_2} \{ \hat{y}(t+j) - r(t+j) \}^2 + \lambda \sum_{j=1}^{N_2} \{ \Delta u(t+j-1) \}^2, \quad (1)$$

$$\Delta u(t+j) = 0, \quad j = N_u, \dots, N_2,$$

여기서, N_1 은 최소 비용 구간이고, N_2 는 최대 비용 구간, N_u 는 제어 비용 구간이다. 또한, $\hat{y}(t+j)$ 는 플랜트의 출력 예측값, $y(t+j)$ 는 기준 명령 신호이고, λ 는 입력 증가분에 대한 비용 가중치이다. 일반 예측 제어에서는 이러한 비용 함수의 최소화 문제를 입력에 대하여 풀어서 이동 구간 제어 (receding horizon control) 입력을 구하게 된다. 이 비용 함수의 최소화 문제를 풀기 위해서는 플랜트의 미래 출력에 대한 예측값 $\hat{y}(t+j)$ 를 얻을 수 있는 예측기가 필요하게 된다. 그리고, 이 예측기의 구조가 플랜트의 미래 입력에 대하여 어파인(affine)하다면, 이 비용 함수 최소화 문제는 쉽게 풀리게 된다. 따라서, 만일 우리가 비선형 플랜트에 대해서 어파인(affine)한 예측기를 구성할 수 있다면, 일반 예측 제어에서 이동 구간 제어 입력은 쉽게 구해질 수 있다. 이것이 이 절에서 플랜트의 미래 입력에 대해 어파인한 퍼지 예측기를 구성하는 이유이다.

다음과 같이 나타내어지는 이산 시간 플랜트를 생각해 보자.

$$y(t+1) = h_1(y(t-n_y), \dots, y(t), u(t-n_u), \dots, u(t)), \quad t \geq 0, \quad (2)$$

여기서, $n_y \geq 0$, $n_u \geq 0$ 이고, $h_1(\cdot)$ 는 연속 함수이다. 이를 "플랜트 모델(plant model)"이라 하자. 또한 (2)로부터 다음을 알 수 있다.

$$y(t+2) = h_1(y(t-n_y+1), \dots, y(t), y(t+1), u(t-n_u+1), \dots, u(t+1)). \quad (3)$$

이때 (3)에서 $y(t+1)$ 를 (2)로 대체하면 다음이 얻어진다.

$$\begin{aligned} y(t+2) &= h_1(y(t-n_y+1), \dots, y(t), h_1(\cdot), \\ & \quad u(t-n_u+1), \dots, u(t+1)) \\ &= h_2(y(t-n_y), \dots, y(t), \\ & \quad u(t-n_u), \dots, u(t), u(t+1)), \end{aligned} \quad (4)$$

여기서, $h_2(\cdot)$ 는 적당한 함수이다. 같은 방법으로 다음을 얻을 수 있다.

$$\begin{aligned} y(t+3) &= h_1(y(t-n_y+2), \dots, y(t), y(t+1), y(t+2), \\ & \quad u(t-n_u+2), \dots, u(t+2)) \\ &= h_1(y(t-n_y+2), \dots, y(t), h_1(\cdot), h_2(\cdot), \\ & \quad u(t-n_u+2), \dots, u(t+2)) \\ &= h_3(y(t-n_y), \dots, y(t), \\ & \quad u(t-n_u), \dots, u(t), u(t+1), u(t+2)), \end{aligned} \quad (5)$$

여기서, $h_3(\cdot)$ 도 또 다른 적당한 함수이다. 이러한 과정을 연속적으로 적용하면 임의의 j 에 대해 플랜트는 다음과 같이 나타내어질 수 있다.

$$\begin{aligned} y(t+j) &= h_j(y(t-n_y), \dots, y(t), \\ & \quad u(t-n_u), \dots, u(t), \dots, u(t+j-1)) \\ &= h_j(\underline{\psi}(t)), \quad t \geq 0, \quad j \geq 1, \end{aligned} \quad (6)$$

여기서, $\underline{\psi}(t) = [y(t-n_y), \dots, y(t), u(t-n_u), \dots, u(t), \dots, u(t+j-1)]^T \in \Psi \subset \mathbb{R}^{n_y+n_u+j+1}$ 는 정보 벡터이고, 정보 벡터의 집합 Ψ 는 유계(bounded)이다. 또한 $h_j(\cdot)$ 는 $h_1(\cdot)$ 과는 다른 적당한 함수이다. 그리고 (6)을 잘 보면 플랜트의 미래 출력이 과거 출력과 과거 입력, 그리고 미래 입력에 의해 결정되는 예측기 형태를 가지고 있음을 알 수 있다. 이를 "예측기 모델(predictor model)"이라고 하자. 그러면, "플랜트 모델(plant model)"로 나타내어지는 플랜트는 전부 "예측기 모델(predictor model)"로 나타낼 수 있음을 알 수 있다. 이때, 이 두 모델 사이의 차이는 그림 1에서 볼 수 있다.

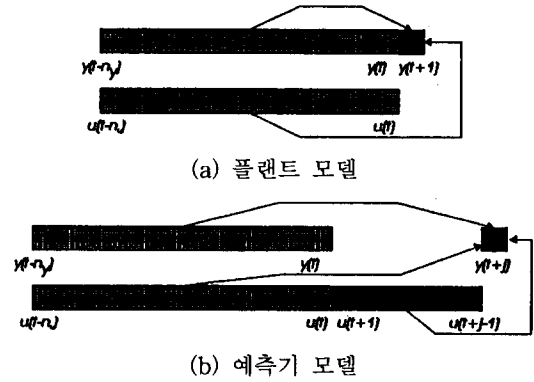


그림 1. 플랜트 모델과 예측기 모델.
Fig. 1. Plant model and predictor model.

한편, 함수 $h_j(\cdot)$ 는 다음과 같이 근사될 수 있다[8].

$$\begin{aligned} h_j(\underline{\psi}(t)) &= h_j(\underline{\psi}_N(t), \underline{\psi}_L(t)) \\ &\cong h_{j0}(\underline{\psi}_N(t)) + \sum_{k=1}^{l_2} h_{jk}(\underline{\psi}_N(t)) \psi_{L,k}(t), \end{aligned} \quad (7)$$

여기서, $\underline{\psi}_N(t)$, $\underline{\psi}_L(t)$ 은 $\underline{\psi}(t) = [\underline{\psi}_N, \underline{\psi}_L]^T \in \Psi$ 인 부분 벡터(sub-vector)들이고,

$$\begin{aligned} \underline{\psi}_N(t) &= [\psi_{N,1}(t), \dots, \psi_{N,l_1}(t)]^T \in \Psi_N \subset \mathbb{R}^{l_1}, \\ \underline{\psi}_L(t) &= [\psi_{L,1}(t), \dots, \psi_{L,l_2}(t)]^T \in \Psi_L \subset \mathbb{R}^{l_2} \end{aligned} \quad (8)$$

이다. $h_{jk}(\cdot): \Psi_N \rightarrow \mathbb{R}$, $0 \leq k \leq l_2$, 은 적당한 비선형 함수이다. 이러한 근사 방법은 근사하고자 하는 함수가 변수 중 일부에 대해서 선형인 특성을 가지게 되는 동작 영역을 나누고, 이 동작 영역 내에서 주어진 함수를 선형화하는 방법이다. 위의 근사 방법에서는 $\underline{\psi}_N(t)$ 가 주어진 함수의 동작 점을 나타내고, $\underline{\psi}_L(t)$ 가 함수를 선형화할 때의 대상 변수이다. 따라서, $\underline{\psi}_N(t)$ 은 함수의 비선형성을 잘 반영할 수 있도록 적절히 잡아야 한다. 만일 이 벡터를 잘못 잡게 되면 이러한 근사 방법에서는 함수의 비선형성의 일부를 반영하지 못하게 되어, 좋은 근사 방법이 될 수 없다. 이제 (6)의 플랜트 모델과 (7)에서

$$\begin{aligned} \underline{\psi}_N(t) &= [y(t-n_y), \dots, y(t), u(t-n_u), \dots, u(t-2)]^T \\ &\equiv [z_1(t), \dots, z_{n_y+n_u}(t)]^T = \underline{z}(t), \end{aligned} \quad (9)$$

$$\underline{\psi}_L(t) = [u(t-1), \dots, u(t+j-1)]^T \quad (10)$$

로 잡는다. 이렇게 잡는 이유는 첫번째로 벡터 $\underline{\psi}_N(t)$ 가 플랜트의 과거 입력과 출력을 포함하고 있어서 플랜트의 비선형성을 반영하는 동작점을 나타낼 수가 있기 때문이다. 물론, 여기서 과거 입력 $u(t-1)$ 이 빠져있지만, 대신 현재 출력 $y(t)$ 가 포함되어 있어서 이를 보완할 수 있게 된다. 두 번째는 이렇게 $\underline{\psi}(t)$ 를 나누어 잡으면 우리가 앞에서 언급한 대로 구성되는 예측기의 구조가 플랜트의 미래 입력에 대해 어파인(affine)한 적당한 구조가 되기 때문이다. 그러면, (7), (9), (10)으로부터 플랜트는 다음의 형태로 모델링될 수 있다.

$$\begin{aligned} y(t+j) &= h_{j0}(\underline{z}(t)) + \sum_{i=1}^j h_{ji}(\underline{z}(t)) u(t+i-1) \\ & \quad + \nu_0(t+j), \quad t \geq 0, \quad j > 0, \end{aligned} \quad (11)$$

여기서, $\nu_0(t+j)$ 는 근사 오차이다.

그런데, 퍼지 시스템은 임의의 연속 실수 함수를 근사할

수 있으므로[16,17], 함수 $h_{j0}(\cdot)$ 와 $h_{\kappa(i+1)}(\cdot)$ 은 다음과 같이 퍼지 시스템으로 근사될 수 있다.

$$\begin{aligned} h_{j0}(\underline{z}(t)) &\cong \underline{\theta}_{1j}^T \underline{\xi}(\underline{z}(t)), \\ h_{\kappa(i+1)}(\underline{z}(t)) &\cong \underline{\theta}_{2ji}^T \underline{\xi}(\underline{z}(t)), \end{aligned} \quad (12)$$

여기서, 파라미터 벡터들은

$$\begin{aligned} \underline{\theta}_{1j} &= [\theta_{1j,1}, \dots, \theta_{1j,M}]^T \\ \underline{\theta}_{2ji} &= [\theta_{2ji,1}, \dots, \theta_{2ji,M}]^T \end{aligned} \quad (13)$$

이고, 회귀 벡터는 $\underline{\xi}(\underline{z}(t)) = [\xi_1(\underline{z}(t)), \dots, \xi_M(\underline{z}(t))]^T$ 이다. 그리고, M 은 퍼지 규칙의 개수이다. 회귀자 $\xi_i(\underline{z}(t))$ 은 다음과 같이 정의된다.

$$\xi_i(\underline{z}(t)) = \frac{\prod_{i=1}^{n_y+n_u} \mu_{F_i'}(z_i(t))}{\sum_{j=1}^M \left(\prod_{i=1}^{n_y+n_u} \mu_{F_i'}(z_i(t)) \right)}, \quad (14)$$

여기서, F_i' 은 규칙 i 의 입력 변수 $z_i(t)$ 에 대한 퍼지 집합이다. $\mu_{F_i'}(z_i(t))$ 은 다음의 소속 함수이다.

$$\mu_{F_i'}(z_i(t)) = \exp\left[-\frac{1}{2} \left(\frac{z_i(t) - \bar{z}_i^i}{\sigma_i^i} \right)^2\right], \quad (15)$$

여기서, \bar{z}_i^i 은 i 번째 입력 공간에서 소속 함수의 값이 최대가 되는 점이고, $\sigma_i^i \in (0, \infty)$ 는 가우시안 소속 함수의 모양을 특징짓는 값이다. 이러한 근사 방법에서 파라미터 벡터, 퍼지 집합, 퍼지 규칙의 개수 등은 예측기의 정확도를 결정하게 되는 설계 파라미터들이다. 결국, 결과적으로 얻어지는 근사된 플랜트 모델은 다음과 같이 주어진다.

$$\begin{aligned} y(t+j) &= \underline{\theta}_{1j}^T \underline{\xi}(\underline{z}(t)) + \\ &\sum_{i=0}^j \underline{\theta}_{2ji}^T \underline{\xi}(\underline{z}(t)) u(t+i-1) + v(t+j), \quad t \geq 0, j > 0 \end{aligned} \quad (16)$$

여기서, $v(t+j)$ 은 근사 오차이다. 이제 일반적으로 (6)으로 주어지는 비선형 플랜트가 모델 (16)으로 나타내어질 수 있다는 것을 알 수 있다. 그런데 이는 과거의 입, 출력들과 앞으로 들어올 입력들로부터 미래의 출력이 만들어지는 예측기의 모양이다. 따라서 이를 플랜트 (2)의 예측기로 사용할 수 있다. 이 때, (16)의 근사 오차를 나타내는 마지막 항 $v(t+j)$ 를 0을 평균(mean)으로 가지는 백색 잡음(white noise)으로 가정하면 플랜트의 출력 예측기는 다음과 같이 구해진다.

$$\begin{aligned} \hat{y}(t+j) &= \underline{\theta}_{1j}^T \underline{\xi}(\underline{z}(t)) \\ &+ \sum_{i=0}^j \underline{\theta}_{2ji}^T \underline{\xi}(\underline{z}(t)) u(t+i-1), \quad t \geq 0, j > 0. \end{aligned} \quad (17)$$

N 스텝 미래까지의 예측기를 사용한다고 가정하고 이를 각각의 j 에 대해 풀어쓰면 예측기는 다음과 같이 된다.

$$\begin{aligned} \hat{y}(t+1) &= \underline{\theta}_{11}^T \underline{\xi}(\underline{z}(t)) + \underline{\theta}_{210}^T \underline{\xi}(\underline{z}(t)) u(t-1) \\ &+ \underline{\theta}_{211}^T \underline{\xi}(\underline{z}(t)) u(t), \\ \hat{y}(t+2) &= \underline{\theta}_{12}^T \underline{\xi}(\underline{z}(t)) + \underline{\theta}_{220}^T \underline{\xi}(\underline{z}(t)) u(t-1) \\ &+ \underline{\theta}_{221}^T \underline{\xi}(\underline{z}(t)) u(t) + \underline{\theta}_{222}^T \underline{\xi}(\underline{z}(t)) u(t+1), \\ &\vdots \\ \hat{y}(t+N) &= \underline{\theta}_{1N}^T \underline{\xi}(\underline{z}(t)) + \underline{\theta}_{2N0}^T \underline{\xi}(\underline{z}(t)) u(t-1) \\ &+ \dots + \underline{\theta}_{2NN}^T \underline{\xi}(\underline{z}(t)) u(t+N-1). \end{aligned}$$

이 예측기는 플랜트의 미래 입력에 대해서 어파인(affine)한

것을 알 수 있다.

이 예측기 (17)은 퍼지 시스템을 사용하여 비선형식을 근사함으로써 얻어졌지만, Takagi-Sugeno 퍼지 모델을 이용해서도 구할 수 있다. 이러한 예측기를 구성하게 하는 퍼지 규칙은 다음과 같다.

IF $z_1(t)$ is F_1^1 and ... and $z_{n_1}(t)$ is $F_{n_1}^1$, THEN

$$\hat{y}(t+j) = \theta_{1j,1} + \theta_{2j,1} u(t-1) + \dots + \theta_{2ji,1} u(t+j-1).$$

IF $z_1(t)$ is F_1^2 and ... and $z_{n_1}(t)$ is $F_{n_1}^2$, THEN

$$\hat{y}(t+j) = \theta_{1j,2} + \theta_{2j,2} u(t-1) + \dots + \theta_{2ji,2} u(t+j-1). \quad \vdots$$

IF $z_1(t)$ is F_1^M and ... and $z_{n_1}(t)$ is $F_{n_1}^M$, THEN

$$\hat{y}(t+j) = \theta_{1j,M} + \theta_{2j,M} u(t-1) + \dots + \theta_{2ji,M} u(t+j-1), \quad (18)$$

여기서 $[z_1(t), \dots, z_{n_1}(t)]^T = \underline{z}(t)$, $n_z = n_y + n_u$ 이고, F_i^i , $1 \leq i \leq n_z$, $1 \leq i \leq M$, 은 퍼지 집합이다. 이러한 M 개의 퍼지 규칙에 퍼지 추론을 거치고 무게 중심 비퍼지화를 하면 다음과 같은 퍼지 시스템의 출력이 만들어진다.

$$\begin{aligned} \hat{y}(t+j) &= \sum_{i=1}^M (\theta_{1j,i} + \theta_{2j,i} u(t-1) \\ &+ \dots + \theta_{2ji,i} u(t+j-1)) w_i(\underline{z}(t)). \end{aligned} \quad (19)$$

이제 싱글톤 퍼지화와 곱 추론을 사용하면 $w_i(\underline{z}(t))$ 는 앞에서 정의한 회귀자 $\xi_i(\underline{z}(t))$ 가 된다.

$$w_i(\underline{z}(t)) = \xi_i(\underline{z}(t)) = \frac{\prod_{i=1}^{n_z} \mu_{F_i^i}(z_i(t))}{\sum_{j=1}^M \left(\prod_{i=1}^{n_z} \mu_{F_i^i}(z_i(t)) \right)}. \quad (20)$$

따라서, (19)에서 $w_i(\underline{z}(t))$ 대신 $\xi_i(\underline{z}(t))$ 을 사용하고 이 식을 변형하면 다음과 같다.

$$\begin{aligned} \hat{y}(t+j) &= \sum_{i=1}^M \theta_{1j,i} \xi_i(\underline{z}(t)) + \sum_{i=1}^M \theta_{2j,i} \xi_i(\underline{z}(t)) u(t-1) \\ &+ \dots + \sum_{i=1}^M \theta_{2ji,i} \xi_i(\underline{z}(t)) u(t+j-1) \\ &= \underline{\theta}_{1j}^T \underline{\xi}(\underline{z}(t)) + \underline{\theta}_{2j0}^T \underline{\xi}(\underline{z}(t)) u(t-1) \\ &+ \dots + \underline{\theta}_{2jj}^T \underline{\xi}(\underline{z}(t)) u(t+j-1), \end{aligned} \quad (21)$$

여기서

$$\begin{aligned} \underline{\theta}_{1j} &= [\theta_{1j,1}, \dots, \theta_{1j,M}]^T, \\ \underline{\theta}_{2ji} &= [\theta_{2ji,1}, \dots, \theta_{2ji,M}]^T, \end{aligned} \quad (22)$$

$$\underline{\xi}(\underline{z}(t)) = [\xi_1(\underline{z}(t)), \dots, \xi_M(\underline{z}(t))]^T$$

이다. (21)을 다시 정리하면

$$\begin{aligned} \hat{y}(t+j) &= \underline{\theta}_{1j}^T \underline{\xi}(\underline{z}(t)) \\ &+ \sum_{i=0}^j \underline{\theta}_{2ji}^T \underline{\xi}(\underline{z}(t)) u(t+i-1), \quad t \geq 0, j > 0. \end{aligned} \quad (23)$$

과 같이 된다. 이 식은 (17)과 동일함을 알 수 있다. 따라서, 앞에서 구한 예측기 (17)은 퍼지 규칙을 (18)과 같이 잡고 곱추론 방법과 무게 중심 비퍼지화 방법을 사용하면 Takagi-Sugeno 퍼지 모델을 이용해서도 구현할 수 있게 된다.

III. 예측기 파라미터의 추정

앞 절에서는 퍼지 모델로 구성할 수 있는 예측기의 구조를 결정하였다. 이 퍼지 예측기에서 정보 벡터의 길이를 결

정하는 n_y, n_u 소속 함수를 결정짓는 \bar{z}_i^l, σ_i^l 등은 예측기의 정확도를 결정하는 설계 파라미터들이다. 이 파라미터들은 예측기의 성능이 최적이 되도록 미리 정해질 수 있는 것들이 아니고 적용되는 플랜트에 따라 달라지게 된다. 기본적으로 파라미터 n_y, n_u 는 적용될 플랜트의 기록된 입출력 데이터를 보고 얼마나 과거까지의 데이터가 현재 출력에 영향을 끼치는 가 하는 것에 따라 결정하여야 한다. 일단, 정보 벡터 $\mathcal{X}(t)$ 에 사용될 과거 입출력 데이터가 결정되면 퍼지 집합의 소속 함수를 결정짓는 \bar{z}_i^l, σ_i^l 를 정하여야 하는데, 기본적으로 퍼지 집합들이 퍼지화할 각 변수들의 전체 공간(universe of discourse)을 잘 구분하고, 모든 경우를 다 표현할 수 있도록 해야 한다. 또 한가지 중요한 것은 입출력 데이터의 분석을 통해서 퍼지 집합이, 플랜트의 특성이 입력에 대하여 선형이 되는 구간을 잘 반영할 수 있도록 하여야 한다는 것이다. 따라서, 전체 공간(universe of discourse)중에서 어느 곳은 퍼지 집합의 분포가 조밀해질 수 있고, 반면에 다른 곳은 덜 조밀해질 수 있다. 만일 이러한 분석이 어렵고 \bar{z}_i^l, σ_i^l 를 결정하기 어려울 경우에는 적은 수의 퍼지 집합을 전체 공간(universe of discourse)내에서 균등하게 배치한 뒤 예측기를 구성해 보고, 예측기의 성능이 만족할 만해질 때 까지 퍼지 집합을 나누거나 합쳐서 이를 결정할 수 있다. 이러한 방법은 퍼지 모델의 전체부 구조 인식(structure identification)과도 밀접하게 연결되어 있다. 이러한 구조 인식의 문제는 대표적으로 [14]에서 다루어졌고, 구조 인식 방법을 통해서 가능한한 규칙의 수를 줄인 퍼지 모델을 구할 수 있다.

전체부의 파라미터들이 정해지면 예측기의 특성은 파라미터 θ_{1j}, θ_{2j} 을 조절하므로써 쉽게 바꿀 수 있게 된다. 따라서, 여기서는 파라미터 벡터 θ_{1j}, θ_{2j} 를 제외한 설계 파라미터들을 미리 결정되었다고 보고 (17)의 파라미터 θ_{1j} 와 θ_{2j} 를 결정하는 방법에 대해서만 언급한다. 이들 파라미터는 전문가에 의해 결정된 퍼지 규칙 집합으로부터 결정될 수도 있고, 또는 파라미터 추정 방법으로부터 결정될 수도 있다. 그러나, 파라미터가 퍼지 규칙 집합에 의해 결정될 때에도 퍼지 규칙 집합이 전문가에 의해 결정되는 한 플랜트의 정확한 특성을 규칙으로 만들기는 쉽지 않으므로 파라미터 추정 기법은 여전히 필요하게 된다. 여기서는 경사 감소(gradient descent) 추정 방법을 고려한다. 퍼지 예측기 식을 다시 적어 보면 다음과 같다.

$$\hat{y}(t+j) = \sum_{k=1}^{M(j+2)} \varphi_{j,k}(t) \theta_{j,k}(t) \quad (24)$$

여기서, 회귀 벡터 $\varphi_j(t)$ 은 다음과 같이 정의되고,

$$\begin{aligned} \varphi_j(t) &= [\xi(\mathcal{Z}(t)), \xi(\mathcal{Z}(t))u(t-1), \dots, \\ &\quad \xi(\mathcal{Z}(t))u(t+j-1)]^T \\ &\equiv [\varphi_{j,1}(t), \dots, \varphi_{j,M(j+2)}(t)]^T \end{aligned} \quad (25)$$

파라미터 벡터 $\theta_j(t)$ 는 다음과 같이 정의된다.

$$\begin{aligned} \theta_j(t) &= [\theta_{1j}(t), \theta_{2j}(t), \dots, \theta_{2j}(t)]^T \\ &\equiv [\theta_{j,1}(t), \dots, \theta_{j,M(j+2)}(t)]^T \end{aligned} \quad (26)$$

이제, 예측 오차를

$$e_j(t) = y(t+j) - \hat{y}(t+j) \quad (27)$$

로 정의하고, 다음과 같은 Lyapunov 함수를 잡는다.

$$V_j(t) = e_j^2(t) \quad (28)$$

그러면, 다음을 알 수 있다.

$$\begin{aligned} \Delta V_j &= 2e_j(t) \left\{ \frac{\partial e_j(t)}{\partial \theta_{j,1}(t)} \Delta \theta_{j,1}(t) + \frac{\partial e_j(t)}{\partial \theta_{j,2}(t)} \Delta \theta_{j,2}(t) \right. \\ &\quad \left. + \dots + \frac{\partial e_j(t)}{\partial \theta_{j,M(j+2)}(t)} \Delta \theta_{j,M(j+2)}(t) \right\} \end{aligned} \quad (29)$$

여기서 $\Delta \theta_{j,k}(t), 1 \leq k \leq M(j+2)$, 는

$$\theta_{j,k}(t+1) = \theta_{j,k}(t) + \Delta \theta_{j,k}(t) \quad (30)$$

이다. 이제 파라미터 갱신값을 다음과 같이 놓으면 $\Delta V_j \leq 0$ 이 되고 파라미터는 수렴하게 된다.

$$\Delta \theta_{j,k}(t) = -\gamma_{j,k} \frac{\partial e_j(t)}{\partial \theta_{j,k}(t)} e_j(t), \quad (31)$$

여기서 $\gamma_{j,k}$ 는 양의 상수로 일종의 학습률을 나타내며 일반적으로 0과 1사이의 값을 사용한다. (24)에서

$$\frac{\partial e_j(t)}{\partial \theta_{j,k}(t)} = -\varphi_{j,k}(t) \quad (32)$$

이므로 파라미터 추정식은 결국 다음과 같이 주어지게 된다.

$$\theta_{j,k}(t+1) = \theta_{j,k}(t) + \gamma_{j,k} \varphi_{j,k}(t) e_j(t). \quad (33)$$

따라서, 이러한 방법을 사용하면 오프라인으로 파라미터를 추정할 수 있게 된다. 만일, 파라미터를 추정할 때 온라인으로 하게 되면 오프라인으로 추정할 때와는 또 다른 문제가 생기게 된다. 퍼지 예측기의 식을 다시 한번 보기로 하자.

$$\hat{y}(t+j) = \varphi_j^T(t) \theta_j(t), \quad t \geq 0, 1 \leq j \leq N. \quad (34)$$

우리가 퍼지 시스템으로 구성한 예측기는 j 스텝 후의 플랜트 출력을 예측하도록 구성하지만, 이 예측기의 파라미터를 추정하기 위해서는 역으로 j 스텝 후의 실제 출력 $y(t+j)$ 이 있어야 한다. 오프라인으로 추정을 할 때는 과거 플랜트의 입출력 데이터를 미리 알고 있기 때문에 j 스텝 후의 출력을 파라미터를 추정하는데 사용할 수 있었다. 하지만, 플랜트 예측기를 온라인으로 추정할 때는 실제로는 j 스텝 후의 출력을 미리 알 수는 없으므로 앞에서 설명한 오프라인 파라미터 추정 방법으로는 이 예측기의 파라미터를 구할 수 없다. 따라서 온라인 추정에서는 예측기 (34)의 파라미터를 추정하기 위해서 현재까지의 데이터와 j 스텝 후의 플랜트 출력 $y(t+j)$ 대신 과거 시점까지의 데이터와 현재의 출력 $y(t)$ 를 사용한다. 따라서, 오프라인으로 파라미터를 추정할 때는 j 스텝이 앞당겨진 다음과 같은 예측기를 대상으로 파라미터 추정을 하게 된다.

$$\hat{y}(t) = \varphi_j^T(t-j) \theta_j(t). \quad (35)$$

여기서는 (35)의 파라미터 벡터 $\theta_j(t)$ 가 (34)의 파라미터 벡터 $\theta_j(t)$ 와 같다고 가정한다. 이는 플랜트 모델 (17)이 플랜트의 특성을 충분히 반영하여 플랜트가 어떠한 비선형성을 가지더라도 현재 시점까지의 데이터로부터 얻을 수 있는 j 스텝 후 출력에 대한 예측기와 과거의 데이터로부터 얻을 수 있는 현재 출력에 대한 예측기가 같은 형태로도 출된다 것을 전제로 하고 있다. 이와 같은 전제는 이 예측기의 구조가 아주 일반적이어서 시간에 관계없이 플랜트의 비선형성을 그대로 반영하고 있다는 가정을 기반으로

하게 된다.

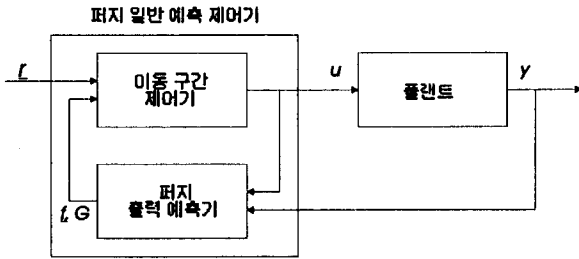


그림 2. 퍼지 일반 예측 제어기의 구조.
Fig. 2. The structure of FGPC.

IV. 퍼지 일반 예측 제어

본 논문에서 제안하는 퍼지 일반 예측 제어 알고리즘에서는 플랜트 모델이 주어지지 않다고 가정한다. 따라서 시스템 모델에서 구해지는 예측기도 얻을 수 없게 된다. 그러나, 이 때 앞에서 언급된 퍼지 예측기는 이의 해결책이 될 수 있다. 이제 제어에 사용되는 비용 함수 (1)은 다음과 같이 다시 쓸 수 있다.

$$J = (\hat{y} - r)^T (\hat{y} - r) + \lambda \tilde{u}^T \tilde{u}, \quad \Delta u(t+j) = 0, \quad j = N_u, \dots, N_2 \quad (36)$$

여기서

$$\begin{aligned} \tilde{u} &= [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T \\ \hat{y} &= [\hat{y}(t+N_1), \hat{y}(t+N_1+1), \dots, \hat{y}(t+N_2)]^T \\ r &= [r(t+N_1), r(t+N_1+1), \dots, r(t+N_2)]^T \end{aligned} \quad (37)$$

이다. 이 때 일반적으로 $N_1 \leq N_u \leq N_2$ 이 가정된다. 이제, 퍼지 예측기 식을 풀어서 다시 써 보면 다음과 같다.

$$\begin{aligned} \hat{y}(t+N_1) &= \theta_{1N_1}^T \xi(z(t)) \\ &+ \theta_{2N_0}^T \xi(z(t))u(t-1) + \theta_{2N_1}^T \xi(z(t))u(t) \\ &+ \dots + \theta_{2N_{N_1}}^T \xi(z(t))u(t+N_1-1), \\ \hat{y}(t+N_1+1) &= \theta_{1(N_1+1)}^T \xi(z(t)) \\ &+ \theta_{2(N_1+1)0}^T \xi(z(t))u(t-1) \\ &+ \theta_{2(N_1+1)1}^T \xi(z(t))u(t) \\ &+ \dots + \theta_{2(N_1+1)(N_1+1)}^T \xi(z(t))u(t+N_1), \\ &\vdots \\ \hat{y}(t+N_2) &= \theta_{1N_2}^T \xi(z(t)) \\ &+ \theta_{2N_0}^T \xi(z(t))u(t-1) \\ &+ \theta_{2N_1}^T \xi(z(t))u(t) \\ &+ \dots + \theta_{2N_{N_2}}^T \xi(z(t))u(t+N_2-1). \end{aligned} \quad (38)$$

여기서, 주목할 것은 비용 함수에서는 $u(t+N_u), \dots, u(t+N_2-1)$ 이 0으로 제한되고 있지만 퍼지 예측기의 파라미터 추정을 위해서는 이 항들을 포함하여야 한다는 것이다. 이 예측기 식을 간단하게 하면 다음과 같다.

$$\begin{aligned} \hat{y}(t+N_1) &= a_{N_1} + b_{N_1}u(t-1) \\ &+ c_{N_{11}}u(t) + \dots + c_{N_{1N_1}}u(t+N_1-1), \\ \hat{y}(t+N_1+1) &= a_{(N_1+1)} + b_{(N_1+1)}u(t-1) \\ &+ c_{(N_1+1)1}u(t) + \dots + c_{(N_1+1)(N_1+1)}u(t+N_1), \\ &\vdots \\ \hat{y}(t+N_2) &= a_{N_2} + b_{N_2}u(t-1) \\ &+ c_{N_{21}}u(t) + \dots + c_{N_{2N_2}}u(t+N_2-1), \end{aligned} \quad (39)$$

여기서, a_i 는 $\theta_{1i}^T \xi(z(t))$ 을 나타내고, b_i 는 $\theta_{2i0}^T \xi(z(t))$ 을, c_{ij} 는 $\theta_{2ij}^T \xi(z(t))$ 을 나타낸다. 비용 함수의 최소화 문제를 풀기 위하여 이 퍼지 예측기 식을 제어 입력의 증분

인 $\Delta u(\cdot)$ 를 포함하는 식으로 만들면 다음과 같다.

$$\begin{aligned} \hat{y}(t+N_1) &= a_{N_1} + b_{N_1}^o u(t-1) \\ &+ c_{N_{11}}^o \Delta u(t) + \dots + c_{N_{1N_1}}^o \Delta u(t+N_1-1), \\ \hat{y}(t+N_1+1) &= a_{(N_1+1)} + b_{(N_1+1)}^o u(t-1) \\ &+ c_{(N_1+1)1}^o \Delta u(t) + \dots + c_{(N_1+1)(N_1+1)}^o \Delta u(t+N_1), \\ &\vdots \\ \hat{y}(t+N_2) &= a_{N_2} + b_{N_2}^o u(t-1) \\ &+ c_{N_{21}}^o \Delta u(t) + \dots + c_{N_{2N_2}}^o \Delta u(t+N_2-1), \end{aligned} \quad (40)$$

여기서, 입력의 증분은 $\Delta u(t) = u(t) - u(t-1)$ 이고, 계수들은

$$\begin{aligned} b_j^o &= b_j + \sum_{i=1}^j c_{ji} \\ c_{jk}^o &= \sum_{i=k}^j c_{ji} \end{aligned} \quad (41)$$

이다. 비용함수에서 $\Delta u(t+N_u) = \dots = \Delta u(t+N_2-1) = 0$ 이 되도록 제한하고 있으므로 이동 구간 제어를 할 때는 퍼지 예측기 식에서 이에 해당하는 항들을 없애고 제어 입력을 구한다. 결국, 예측기 식은 다음과 같이 행렬식으로 나타낼 수 있다.

$$\hat{y} = f + G \tilde{u} \quad (42)$$

여기서, 행렬 G 는

$$G = \begin{bmatrix} c_{N_1,1}^o & \dots & c_{N_1,N_1}^o & 0 \\ c_{(N_1+1),1}^o & \dots & c_{(N_1+1)(N_1+1)}^o & \\ \vdots & & \vdots & \\ c_{N_u,1}^o & \dots & c_{N_u,N_u}^o & \\ \vdots & & \vdots & \\ c_{N_2,1}^o & \dots & c_{N_2,N_2}^o & \end{bmatrix} \quad (43)$$

로 주어지고, f 는

$$f = \begin{bmatrix} a_{N_1} + b_{N_1}^o u(t-1) \\ \vdots \\ a_{N_2} + b_{N_2}^o u(t-1) \end{bmatrix} \quad (44)$$

이다. 그러면 (36)과 (42)로부터 이차 최소화 (quadratic minimization) 문제는 선형 대수의 문제로 바뀌게 된다. 따라서, 문제의 해인 미래 제어 입력 증분 벡터 \tilde{u} 는 다음과 같이 구할 수 있다.

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (r - f). \quad (45)$$

따라서 (45)에 의해서 시간 t 에서 $t+N_u-1$ 까지의 미래 제어 입력 증분을 계산할 수 있다. 하지만, 실제 플랜트의 제어에서는 제일 처음의 제어 입력 증분 $\Delta u(t)$ 만을 사용하면 된다. 따라서, 제어 입력 증분은 다음과 같이 주어지게 된다.

$$\Delta u(t) = [1 \ 0 \ \dots \ 0] (G^T G + \lambda I)^{-1} G^T (r - f). \quad (46)$$

그리고, 다음 스텝에서는 시간 $t+1$ 에서 구할 수 있는 데이터로부터 앞의 방법을 다시 사용하여 제어 입력 증분을 다시 계산한다. 전체적인 제어기의 구조는 그림 2에 나타난 것과 같이 앞에서 설명한 퍼지 예측기와 (46) 식으로 주어지는 이동 구간 제어를 합쳐 놓은 것과 같다.

V. 시뮬레이션

예제 1 : 여기서는 비선형 플랜트의 경우 퍼지 일반 예측 제어의 결과를 기존의 일반 예측 제어를 사용한 것과 비교하여 본다. 여기서 대상으로 한 플랜트는 다음과 같다.

$$y(t) = 0.9y(t-1) - 0.6y^2(t-1)y(t-2) + u(t-1). \quad (47)$$

일반 예측 제어를 적용할 때는 플랜트를 동작점마다 선형화하고 각 선형화된 모델로 부터 출력 예측기를 구하였다. 이 경우 각 비용 구간은 $N_1=1, N_2=3,$ and $N_u=3$ 으로 하였고 입력에 대한 가중치 λ 는 0.1로 하였다. 그림 3은 이렇게 선형화된 플랜트 모델을 이용하여 일반 예측 제어를 적용한 결과를 나타내고 있다. 이를 보면 플랜트의 출력에 오버슈트가 크고 진동이 많음을 볼 수 있다. 이는 플랜트의 동작점이 크게 변하는 곳에서는 선형화된 모델에서 구한 예측기가 플랜트의 출력을 제대로 예측하지 못하기 때문이다. 한편, 이 플랜트를 퍼지 일반 예측 제어를 사용해서 제어할 때 퍼지 예측기는 앞에서와 같이 3 스텝 앞의 미래 출력까지 예측할 수 있도록 다음과 같이 구성하였다.

$$\begin{aligned} \hat{y}(t+1) &= \theta_{11}^T \xi(z(t)) + \theta_{210}^T \xi(z(t))u(t-1) \\ &\quad + \theta_{211}^T \xi(z(t))u(t) \\ \hat{y}(t+2) &= \theta_{12}^T \xi(z(t)) + \theta_{220}^T \xi(z(t))u(t-1) \\ &\quad + \theta_{221}^T \xi(z(t))u(t) + \theta_{222}^T \xi(z(t))u(t+1), \\ \hat{y}(t+3) &= \theta_{13}^T \xi(z(t)) + \theta_{230}^T \xi(z(t))u(t-1) \\ &\quad + \theta_{231}^T \xi(z(t))u(t) + \theta_{232}^T \xi(z(t))u(t+1) \\ &\quad + \theta_{233}^T \xi(z(t))u(t+2). \end{aligned} \quad (48)$$

그리고, 이는 다음과 같이 간단히 나타내어진다.

$$\begin{aligned} \hat{y}(t+1) &= a_1 + b_1^o u(t-1) + c_{11}^o \Delta u(t) \\ \hat{y}(t+2) &= a_2 + b_2^o u(t-1) + c_{21}^o \Delta u(t) + c_{22}^o \Delta u(t+1), \\ \hat{y}(t+3) &= a_3 + b_3^o u(t-1) + c_{31}^o \Delta u(t) + c_{32}^o \Delta u(t+1) \\ &\quad + c_{33}^o \Delta u(t+2). \end{aligned} \quad (49)$$

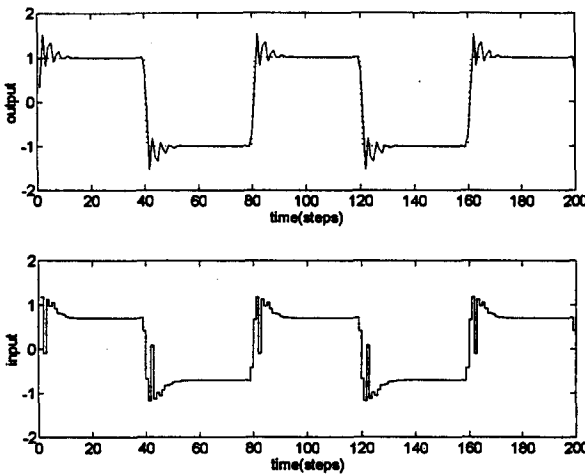


그림 3. 선형화된 플랜트에 일반 예측 제어 적용 결과. (예제 1)

Fig. 3. Simulation results using GPC for linearized plant. (example 1)

여기서, a_i, b_i^o, c_{ij}^o 는 정보벡터의 함수이다. 정보벡터는 $z^T(t)=[y(t-1) \ y(t)]$ 를 사용하였고, 각 변수에 대해 3개의 퍼지 집합을 사용하였다. 또한, 퍼지 예측기의 파라미터 $\theta_{1i}, \theta_{2ij}$ 는 오프라인으로 추정하였다. 그림 5는 퍼지 일반 예측 제어를 사용하였을 때의 결과를 보여주고 있다. 이를 보면 오버슈트가 작고, 진동도 덜한 것을 볼 수 있다. 따라서 퍼지 일반 예측 제어 방법이 이 경우에는 더 좋은 성능을 내는 것을 알 수 있다. 이는 퍼지 시스템으로 구성된 플랜트의 출력 예측기가 앞에서 선형화된 모델로부터 구한 예측기보다 더 정확한 예측을 하기 때문이다.

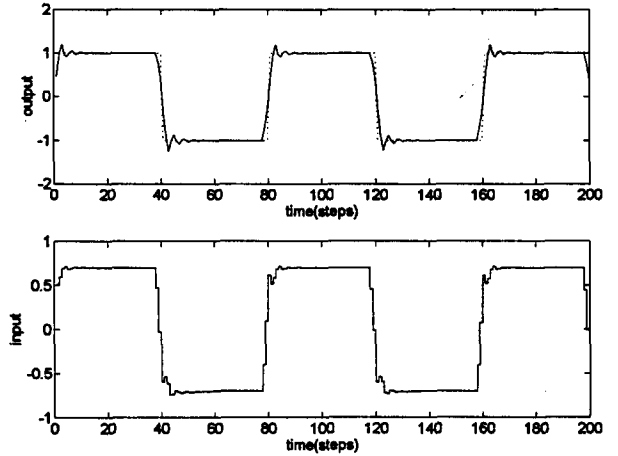


그림 4. 비선형 모델에 퍼지 일반 예측 제어 적용 결과, 변수 추정 종료시. (예제 2)

Fig. 4. Simulation results using FGPC for non-linear plant, parameter estimation finished. (example 2)

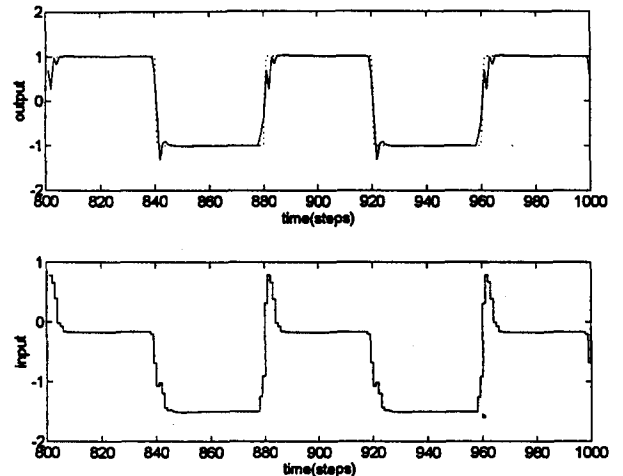


그림 5. 퍼지 일반 예측 제어 적용 결과. (예제 1)

Fig. 5. Simulation results using FGPC. (example 1)

예제 2 : 여기서는 비선형이 강한 플랜트에 대해 퍼지 예측기를 통한 예측 제어의 성능을 본다. 시뮬레이션에 사용된 플랜트의 모델은 다음과 같다.

$$y(t) = \frac{y(t-1)y(t-2)[y(t-1)+2.5]}{1+y^2(t-1)+y^2(t-2)} + u(t-1). \quad (48)$$

퍼지 예측기는 예제 1에서와 같이 구성하였다. 퍼지 시스템이 정의되는 전체 공간(universe of discourse)은 $[-1.5, 1.5]$ 가 되도록 하였고, 각 입력 변수에 대해 3개의 퍼지 집합을 사용하였다. 퍼지 예측기의 파라미터는 플랜트의 입출력 데이터로부터 온라인으로 추정하였고, 정보 벡터로는 $\hat{y}(t+1), \hat{y}(t+2), \hat{y}(t+3)$ 을 예측하는 각 예측기에 대해 각각 $[y(t-2), y(t-1)], [y(t-3), y(t-2)], [y(t-4), y(t-3)]$ 을 사용하였다. 그림 4는 퍼지 예측기 파라미터의 추정이 거의 끝났을 때 퍼지 일반 예측 제어에 의해 제어된 플랜트의 입력과 출력을 보여주고 있다. 이를 보면 퍼지 일반 예측 제어가 제대로 동작하는 것을 알 수 있다.

VI. 결론

본 논문에서는 비선형 제어 요소로서 점차 인정받고 있는

퍼지 시스템 이론을 사용하여 일반 예측 제어를 비선형 플랜트에 적용할 수 있는 새로운 방법을 제안하였다. 제안된 방법은 비선형 플랜트의 미래 출력을 퍼지 시스템으로 구성된 예측기를 통해 예측하고 이 예측된 값을 통해 이동 구간 최적화를 하는 일반 예측 제어를 적용하는 방법이다. 또한, 제안된 방법에서는 파라미터 식별을 통해서 전문가의 지식만으로 구성된 퍼지 예측기의 정확도를 높이도록 하였다.

제안된 방법의 큰 특징은 퍼지 시스템을 이용하여 비선형 플랜트의 어파인(affine) 예측기 모델을 만들기 때문에 기존의 비선형 일반 예측 제어 방법에서와는 달리 비용 함수의 비선형 최적화를 할 필요가 없고 쉽게 제어 입력을 구할 수 있다는 것이다. 제안된 방법의 또다른 특징은 플랜트를 퍼지 시스템으로 모델링할 때 플랜트 자체를 모델링하는 대신 플랜트의 미래 출력을 예측하는 예측기를 직접 모델링한다는 것이다. 이는 제어 입력을 구할 때는 예측기만 사용한다는 점을 이용한 것으로 플랜트 모델을 먼저 추정하고 이로부터 예측기를 구하는 기존의 방법과는 다른 점이다. 제안된 방법은 플랜트에 대한 예측기를 구성할 때 플랜트가 어떤 플랜트인가에 상관없이 예측기를 구성하기 때문에 선형 시스템이나 비선형 시스템의 구분 없이 적용될 수 있다는 장점이 있다. 또한, 제안된 방법에는 퍼지 시스템을 이용함으로써 플랜트에 대한 전문가의 지식을 그대로 이용할 수 있다는 장점도 있다. 이는 전문가가 플랜트에 대해 가지고 있던 기본 지식을 바탕으로 플랜트의 미래 출력 예측기를 구성할 수 있기 때문이다. 이 경우 전문가 지식의 정확성이 떨어져도 파라미터 식별 방법을 사용하여 퍼지 예측기의 정확성을 유지할 수 있게 한다. 본 논문에서는 비선형 플랜트에 대한 시뮬레이션을 통해 제안된 퍼지 일반 예측 제어의 성능을 확인하였다.

참고문헌

- [1] S. Aoki, S. Kawachi, and M. Sugeno, "Application of fuzzy control logic for dead-time processes in a glass melting furnace," *Fuzzy Sets and Systems*, vol. 38, pp. 251-265, 1990.
- [2] P. Bendotti and M. M'Saad, "A skid-to-turn missile autopilot design: the generalized predictive adaptive control approach," *Int. J. Adaptive Contr. and Sig. Proc.*, vol. 7, pp. 13-31, 1993.
- [3] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control-Part 1. Basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137-148, 1987.
- [4] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control-Part 2. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149-160, 1987.
- [5] D. W. Clarke, "Application of generalized predictive control to industrial processes," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 49-55, 1988.
- [6] J. M. Dion, L. Dugard, A. Franco, Nguyen Minh Tri, and D. Rey, "MIMO adaptive constrained predictive control case study: An environmental test chamber," *Automatica*, vol. 27, no. 4, pp. 611-626, 1991.
- [7] G. A. Dumont, Y. Fu, and G. Lu, "Nonlinear adaptive generalized predictive control and applications," *Advances in Model-Based Predictive Control*, Oxford University Press, pp. 498-515, 1994.
- [8] T. A. Johansen and B. A. Foss, "Constructing NARMAX models using ARMAX models," *Int. J. Contr.*, vol. 58, no. 5, pp. 1125-1153, 1993.
- [9] T. A. Johansen, "Fuzzy model based control: Stability, robustness, and performance issues," *IEEE Trans. Fuzzy Systems*, vol. 2, no. 3, pp. 221-234, 1994.
- [10] D. A. Linkens and M. Mahfouf, "Generalized predictive control with feedforward (GPCF) for multivariable anaesthesia," *Int. J. Contr.* vol. 56, pp. 1036-1057, 1992.
- [11] M. Mahfouf D. A. Linkens, A. J. Asbury, W. M. Gray, and J. E. Peacock, "Generalized predictive control (GPC) in the operating-theater," *IEE Proc. Part D* vol. 139, pp. 404-420, 1992.
- [12] S. Miyamoto S. Yasunobu, and H. Ihara, "Predictive fuzzy control and its application to automatic train operation systems," *Analysis of Fuzzy Information*, vol. II (Florida, CRC Press, 1990) pp. 59-72.
- [13] J. A. Rossiter B. Kouvaritakis, and R. M. Dunnnett, "Application of generalized predictive control to a boiler-turbine unit for electricity generation," *IEE Proc. Part D* vol. 138, pp. 59-67, 1991.
- [14] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15-33, 1988.
- [15] M. T. Tham F. Vagi, A. J. Morris, and R. K. Wood, "Multivariable and multirate self-tuning control: a distillation column case study," *IEE Proc. Part D* vol. 138, pp. 9-24, 1991.
- [16] L. X. Wang, "Fuzzy systems are universal approximators," *Proc. IEEE Int. Conf. Fuzzy Systems*, pp. 1163-1170, 1992.
- [17] L. X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*, Prentice-Hall, 1994.
- [18] Q. M. Zhu, K. Warwick, and J. L. Douce, "Adaptive general predictive controller for nonlinear systems," *IEE Proc. Part D*, vol. 138, no. 1, pp. 33-40, 1991.



안 상 철

1988년 서울대학교 제어계측공학과 졸업. 1990년 동대학원 제어계측공학과 석사. 1996년 동대학원 제어계측공학과 박사. 1996년 ~ 1997년 미국 Univ. of Southern California 방문연구원. 관심 분야는 퍼지시스템, 신경회로망 이론 및 응용, 제어이론, 3D sensing, 이미지 프로세싱 등, 현재 한국과학기술 연구원 근무.



김 용 호

1990년 서울대학교 제어계측공학과 졸업. 1992년 동대학원 제어계측공학과 석사. 1994년 동대학원 제어계측공학과 박사과정 수료, 관심분야는 퍼지시스템 이론 및 응용, 제어 및 시스템 이론, 산업용 통신망 등.



권 옥 현

1966년 서울대학교 전기공학과 졸업, 1972년 동대학원 석사, 1975년 미국 Brown University 제어공학 박사, 1966년 ~ 1968년 육군 통신 학교 중위, 1975년 ~ 1976년 Brown University 연구원, 1976년 ~ 1977년 University of Iowa 겸직조교수, 1981년 ~ 1982년 Stanford University 객원교수, 1977년 ~ 현재 서울대학교 전기공학부 교수, 1991 ~ 현재 서울대학교 제어계측신기술연구센터 소장, 관심분야는 제어 및 시스템 이론, 이산 현상 시스템, 산업용 통신망, 분산 공정 제어 등.