

입력의 크기를 고려한 비선형 시스템의 반복학습 제어

Iterative Learning Control of Nonlinear Systems with Consideration on Input Magnitude

최종호, 장태정
(Chong-Ho Choi and Tae-Jeong Jang)

Abstract : It is not desirable to have too large control input in control systems, because there are usually a limitation for the input magnitude and cost for the input energy. Previous papers in the iterative learning control did not considered on these points. In this paper, an iterative learning control method is proposed for a class of nonlinear systems with consideration on input magnitude by adopting a concept of cost function consisting of the output error and the input magnitude in quadratic form. We proposed a new input update law with an input penalty function. If we choose a reasonable input penalty function, the two control objectives, good command following and small input energy, can be achieved. The characteristics of the proposed method are shown in the simulation examples.

Keywords : iterative learning control, nonlinear systems, input penalty function

I. 서론

반복학습 제어는 어떤 시스템의 유한한 시간 구간에서의 실제 출력 신호가 우리가 원하는 출력 궤적을 정확하게 추종하도록 하는 입력 신호의 궤적을 반복되는 학습을 통하여 구해내는 제어 기법이다. 즉, 주어진 시간 구간에서의 시행을 반복하면서 매 시행마다 원하는 출력 궤적과 측정된 출력 궤적 사이에 발생한 오차 궤적을 이용하여 현재 시행시 입력한 입력 궤적을 수정하여 다음 시행시 인가할 새로운 입력 신호를 만들어 주는 것이다. 이러한 반복학습 제어의 기본 아이디어는 1978년 Uchiyama[1]가 일본 국내에서 처음 제안한 이후 1984년 Arimoto 등[2]이 국제 학회에 논문을 발표함으로써 여러 연구자들의 관심을 끌기 시작했고 최근까지 많은 관련 논문들이 발표되고 있다[3-16].

지금까지의 반복학습 제어는 원하는 출력 궤적을 정확하게 추종하게 하는 입력 궤적을 구해내는 것이기 때문에 이러한 원하는 출력 궤적을 만들어내는 입력이 실제로 시스템에 인가할 수 있는 입력의 범위를 벗어나서는 안된다. 따라서 반복학습 제어 이론에서는 주어진 원하는 출력 궤적을 만들어낼 수 있는 원하는 입력이 항상 존재한다는 가정이 필요하다. 그리고, 실제 시스템에 반복학습 제어를 구현할 때는 그 시스템에 인가할 수 있는 입력으로 생성 가능한 궤적을 원하는 출력 궤적으로 설정하여야 한다.

그러나 정확한 시스템 모델과 파라미터가 알려져 있지 않고 시스템에 불확정성이 존재하는 등 시스템에 관한 정보가 제한적일 때, 시스템의 성능을 최대한 발휘할 수 있는 원하는 출력 궤적을 구하는 것은 쉽지 않다. 왜냐하면 시스템의 성능을 최대한 발휘한다는 것은 대개 시스템에 인가할 수 있는 최대 입력 또는 최대 입력에 가까운 입력 범위에서 동작하는 것을 의미하며 이 경우 자칫 원하는 출력을 내게하는 원하는 입력이 실제로 시스템에 인가할 수 있는 범위를 벗어날 수 있기 때문이다.

이와같은 문제점 때문에 지금까지의 반복학습 제어 이론에서는 앞서도 지적했던 바와 같이 우리가 설정하는 원하는 궤적은 시스템에 인가할 수 있는 입력의 범위에서 구현

가능(achievable)해야한다는 가정을 필요로 한다. 이런 가정과 더불어 입력 궤적과 이에 대응되는 출력 궤적 사이의 일대일 대응 관계를 추가로 가정함으로써 반복학습에 의해 학습되는 입력이 어떤 궤적으로 수렴한다면 그 궤적은 반드시 원하는 출력 궤적을 내게하는 원하는 입력 궤적이 된다는 것을 보장한다.

본 논문에서는 기존의 반복학습 제어에서의 기본적인 목표인 실제 출력이 원하는 출력을 정확하게 추종하게 하는 입력을 구해내는 조건을 완화하여, 실제 시스템이 추종하기에는 좀 무리한 원하는 출력 궤적이 주어졌다 할지라도 그 궤적에 근접하게 따라가면서 입력의 실제 크기는 작은 입력 궤적을 학습하는 방법에 대해 알아본다. 이를 가능하게 하는 방법으로서 본 논문에서는 원하는 출력 궤적에 대한 추종성과 입력의 크기 사이에 가중치를 두고 이 가중치의 크기를 조정함으로써 입력의 크기에 대한 보상을 행한다.

입력과 상태변수, 또는 입력과 출력 사이에 가중치를 도입하는 것은 이미 선형 최적제어(linear optimal control)의 LQR(linear quadratic regulator) 문제 등에서 이미 가중치 행렬(weight matrix)을 쓰고 있기 때문에 새로운 것은 아니다. 그러나 기본적으로 최적제어에서는 시스템의 모델을 정확히 알고 있다는 가정에서 최적의 제어를 설계하는 문제이기 때문에 잘 알려져 있지 않은 시스템의 어떤 주어진 출력 궤적을 가장 잘 추종할 수 있는 입력 궤적을 반복적인 학습으로 구해내는 반복학습 제어 문제와는 근본적인 차이점이 있다. 오히려 이러한 가중치의 도입은 기존의 반복학습 제어 이론을 최적제어 이론과 연관시켜 좀 더 일반화시킨 관점이라 볼 수도 있을 것이다.

II. 시스템 정의 및 가정

다음과 같은 비선형 시스템을 생각하자.

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + G(x(t))u(t), \\ y(t) &= h(x(t)), \end{aligned} \quad (1)$$

단, $u = [u_1, \dots, u_m]^T \in \mathbb{R}^m$, $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, 그리고 $y = [y_1, \dots, y_m]^T \in \mathbb{R}^m$ 는 플랜트 입력, 시스템의 상태 변수, 그리고 플랜트 출력을 각각 나타낸다. 함수 $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $G = [g_1, \dots, g_m]: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$, 그리고 $h = [h_1, \dots,$

$h_m]^T: R^n \rightarrow R^m$ 는 그들의 정의영역에서 각각 해석가능(analytic)한 함수¹들이다. 원하는 출력 궤적 $y^d(t) = [y_1^d(t), \dots, y_m^d(t)]^T$ 가 유한한 시간구간 $t \in [0, T]$ 에서 주어졌다고 하자. 그러면 원하는 출력과 실제 출력 궤적 사이의 오차는 다음과 같이 표현될 수 있다.

$$e(t) = y^d(t) - y(t), \quad \forall t \in [0, T]. \quad (2)$$

여기서 $e = [e_1, \dots, e_m]^T \in R^m$ 이다. 입력 포화기(input saturator)를 가지는 다음과 같은 되먹임 제어를 생각하자.

$$\dot{z}(t) = a(z(t)) + B(z(t))e(t), \quad (3)$$

$$u_{fb}(t) = c(z(t)) + D(z(t))e(t), \quad (4)$$

$$v(t) = u_{ff}(t) + u_{fb}(t), \quad (5)$$

$$u(t) = \gamma(v(t)), \quad (6)$$

단, $z \in R^p$, $u_{fb} \in R^m$, $u_{ff} \in R^m$, $v = [v_1, \dots, v_m]^T \in R^m$, 그리고 $u \in R^m$ 는 제어기의 상태변수, 되먹임 제어 입력, 앞먹임 입력, 제어 입력, 그리고 플랜트 입력을 각각 나타낸다. 함수 $a: R^p \rightarrow R^p$ 와 $c: R^p \rightarrow R^m$, 그리고 $B: R^p \rightarrow R^{p \times m}$ 와 $D: R^p \rightarrow R^{m \times m}$ 는 그들의 정의영역에서 해석 가능하고 $a(0) = 0$, $c(0) = 0$ 이다. 입력 포화 함수 $\gamma: R^m \rightarrow R^m$ 는 $\gamma(v(t)) = [\gamma_1(v_1(t)), \dots, \gamma_m(v_m(t))]^T$ 와 같이 나타낼 수 있고, 이 때

$$\gamma_i(v_i(t)) = \begin{cases} v_i(t), & \text{if } |v_i(t)| \leq \bar{v}_i, \\ \frac{v_i(t)}{|v_i(t)|} \bar{v}_i, & \text{if } |v_i(t)| > \bar{v}_i \end{cases} \quad (7)$$

와 같이 정의된다. 단, $\bar{v}_i > 0$, $i = 1, \dots, m$ 는 포화 한계값들이다.

앞으로 학습의 시행 횟수를 표시하기 위해 윗첨자 k 를 사용하겠다. 따라서 $x^k(t)$, $y^k(t)$, $z^k(t)$, $u_{fb}^k(t)$, $u_{ff}^k(t)$, $v^k(t)$ 및 $u^k(t)$ 는 k 번째 시행시의 각각의 벡터를 나타낸다. 그리고 앞으로 사용하기 위해 다음과 같은 몇 가지 기호들과 정의를 기술하겠다.

먼저 n 차원의 벡터 $x = [x_1, \dots, x_n]^T$ 에 대하여 벡터 노름(norm)을 $\|x\| = \max_{1 \leq i \leq n} |x_i|$ 와 같이 정의한다. 또한 $n \times m$ 행렬 $W = \{w_{ij}\}$ 에 대하여 벡터 노름에서 유도(induced)된 행렬 노름을 $\|W\| = \max_{1 \leq i \leq n} (\sum_{j=1}^m |w_{ij}|)$ 와 같이 정의하자. 그리고 함수 $q: [0, T] \rightarrow R^m$ 에 대하여 λ -노름을

$$\|q(t)\|_\lambda = \sup_{t \in [0, T]} e^{-\lambda t} \|q(t)\| \quad (8)$$

와 같이 정의하자[2].

스칼라 함수 $\varphi(x)$ 의 벡터 $f(x) = [f_1(x), \dots, f_n(x)]^T$ 에 따른 미분은 다음과 같이 나타낸다.

$$L_f \varphi(x) = \frac{\partial \varphi}{\partial x} f(x) = \sum_{i=1}^n \frac{\partial \varphi}{\partial x_i} f_i(x), \quad (9)$$

단, $x = [x_1, \dots, x_n]^T$ 이다. 그리고 $\varphi(x)$ 를 먼저 $f(x)$ 에 따라 미분한 다음 다시 어떤 벡터 $g(x)$ 에 따라 미분하는 것을

$$L_g L_f \varphi(x) = \frac{\partial (L_f \varphi)}{\partial x} g(x) \quad (10)$$

와 같이 나타낸다. 또한 $\varphi(x)$ 를 $f(x)$ 에 따라 j 번 미분한 것을 $L_f^j \varphi(x)$ 와 같이 나타낸다. 이 때, $L_f^0 \varphi(x) = \varphi(x)$ 이다.

정의 2.1[17] : 앞의 (1)의 형태로 주어지는 비선형 시스템은 다음과 같은 조건을 만족할 때 x^0 에서 (벡터) 상대차수 $\{\sigma_1, \dots, \sigma_m\}$ 을 가진다.

(i) 모든 $1 \leq j \leq m$, $1 \leq i \leq m$, $k \leq \sigma_i - 1$, 그리고 x^0 의 어떤 부근(neighborhood)에 있는 모든 x 에 대하여 $L_{g_i} L_f^k h_i(x) = 0$ 이다.

(ii) $m \times m$ 비간섭(decoupling) 행렬

$$J(x) = \begin{bmatrix} L_{g_1} L_f^{\sigma_1 - 1} h_1(x) & \dots & L_{g_m} L_f^{\sigma_1 - 1} h_1(x) \\ L_{g_1} L_f^{\sigma_2 - 1} h_2(x) & \dots & L_{g_m} L_f^{\sigma_2 - 1} h_2(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{\sigma_m - 1} h_m(x) & \dots & L_{g_m} L_f^{\sigma_m - 1} h_m(x) \end{bmatrix} \quad (11)$$

는 $x = x^0$ 에서 비특이(nonsingular) 행렬이다.

여기서 σ_i 는 시간 $t = t_0$ 에서 i 번째 출력 $y_i(t)$ 에 입력 벡터 $u(t_0)$ 의 항들 중 적어도 하나가 나타나기 위해 $y_i(t)$ 를 미분해야 하는 최소 횟수와 일치한다[17].

또한 되먹임 시스템 (1)-(2)에 대하여 다음과 같은 성질을 가정하겠다.

(A2.1) 시스템 (1)에 대하여 Φ_s 를 $(x(0), u(t), t \in [0, T])$ 에서 $(x(t), t \in [0, T])$ 로의 사상으로, 그리고 Ψ_s 를 $(x(0), u(t), t \in [0, T])$ 에서 $(y(t), t \in [0, T])$ 로의 사상이라고 하자. 그러면, 각 $x(0)$ 에 대하여 상태변수 사상 Φ_s 와 출력 사상 Ψ_s 는 각각 일대일 대응이다. 또한 되먹임 제어기 (3)-(6)에 대하여 Φ_c 를 $(z(0), e(t), t \in [0, T])$ 에서 $(z(t), t \in [0, T])$ 로의 사상으로, 그리고 Ψ_c 를 $(z(0), e(t), t \in [0, T])$ 에서 $(u_{fb}(t), t \in [0, T])$ 로의 사상이라고 하자. 그러면, 각 $z(0)$ 에 대하여 제어기 상태변수 사상 Φ_c 와 제어기 출력 사상 Ψ_c 는 각각 일대일 대응이다.

(A2.2) 시스템 (1)의 상대차수는 모든 $x \in R^n$ 에 대하여 $\{\sigma_1, \dots, \sigma_m\}$ 이다.

(A2.3) 정칙(positive definite) 행렬 함수 $\Lambda: [0, T] \rightarrow R^{m \times m}$ 를 $\Lambda(t) = \text{diag} \{\lambda_1(t), \dots, \lambda_m(t)\}$, $t \in [0, T]$ 라 정의하자. 또한 행렬 함수 $S: R^n \rightarrow R^{m \times m}$ 를 정의영역 R^n 에서 해석 가능한 유계의(bounded) 함수라고 하자. 되먹임 시스템 (1)-(6)에서 $x(0)$, $z(0)$, 그리고 $y^d(t)$, $t \in [0, T]$ 가 주어졌을 때, 주어진 $\Lambda(t)$, $t \in [0, T]$ 에 대하여 시스템 관계식 (1)-(6)을 만족하며 또한 다음 관계식을 만족하는 $x^*(t)$, $y^*(t)$, $u_{ff}^*(t)$, $u_{fb}^*(t)$ 가 유일하게 존재한다.

$$S(x^*(t)) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} - \Lambda(t) u_{ff}^*(t) + (I - \Lambda(t)) u_{fb}^*(t) = 0. \quad (12)$$

1) 어떤 스칼라 함수 $\varphi: R^n \rightarrow R$ 가 해석 가능하기 위해서는 그 함수가 무한히 미분 가능하고, 즉 $\varphi \in C^\infty$ 이고, 모든 $x^0 \in R^n$ 각각에 대하여 x^0 의 어떤 부근(neighbourhood) W 가 존재하여 φ 의 x^0 에서의 Taylor열 전개가 모든 $x \in W$ 에서 $\varphi(x)$ 에 수렴해야 한다[17]. 어떤 벡터 (또는 행렬) 함수가 해석 가능하기 위해서는 그 함수의 각 항들이 각각 해석 가능해야 한다.

여기서 출력 $y^*(t), t \in [0, T]$ 는 입력 포화 한계 내의 입력에 의해 구현가능(achievable) 하다. 다시 말해서 출력 궤적 $y^*(t)$ 를 생성하는 플랜트 입력 $u^*(t)$ 는 시간 $t \in [0, T]$ 에서 $u^*(t) = \gamma(u^*(t))$ 를 만족한다. 이 때 $u_{ff}^*(t)$ 와 $u_{fb}^*(t)$ 는 플랜트 입력이 $u^*(t)$ 일 때의 앞먹임 입력과 되먹임 제어 입력을 각각 나타내며, $u_{ff}^*(t) + u_{fb}^*(t) = u^*(t), \forall t \in [0, T]$ 이다.

(A2.4) $U \subset R^m$ 를 $U = \{u \in R^m \mid u = \gamma(u)\}$ 를 만족하는 유계(bounded)의 플랜트 입력 공간으로 생각하자. 그러면 주어진 $y^d(t), t \in [0, T]$ 및 초기 상태변수 $x(0)$ 와 $z(0)$ 에 대하여 $U \times [0, T]$ 에 속하는 모든 입력 궤적에 의해 생성될 수 있는 모든 플랜트의 상태변수 $x(t)$ 와 제어기의 상태변수 $z(t)$ 를 각각 포함하는 콤팩트(compact) 집합 $X \subset R^n$ 와 $Z \subset R^p$ 가 존재한다. 즉, 모든 $t \in [0, T]$ 에 대하여 $x(t) \in X$, 이고 $z(t) \in Z$ 이다.

주 2.1 : 만약 $y^*(u^*)$ 를 알면 시스템 (1)과 가정 (A2.1)로부터 $u^*(y^*)$ 를 알 수 있고, 그러면 제어기 (3)-(6)과 가정 (A2.1)로부터 u_{fb}^* 와 u_{ff}^* 가 유일하게 결정된다.

주 2.2 : 위의 식 (12)를 만족시키는 $u_{fb}^*(t)$ 가 알려졌다고 하고 다음의 목적함수를 생각하자.

$$V(t) = \frac{1}{2} \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{(\sigma_m)}(t) \end{bmatrix}^T Q(t) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{(\sigma_m)}(t) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} u(t) \\ u_{fb}^*(t) \end{bmatrix} \begin{bmatrix} \Lambda(t) & -I \\ -I & P(t) \end{bmatrix} \begin{bmatrix} u(t) \\ u_{fb}^*(t) \end{bmatrix} \quad (13)$$

여기서 $Q(t) \in R^{m \times m}$ 및 $P(t) \in R^{m \times m}$ 는 정칙(positive definite)인 대칭 행렬들이며 특히 $P(t)$ 는 입력 벡터 $[u^T(t) \ u_{fb}^{*T}(t)]^T$ 의 가중치 행렬인

$$R(t) = \begin{bmatrix} \Lambda(t) & -I \\ -I & P(t) \end{bmatrix} \quad (14)$$

이 정칙 행렬이 되도록 정한다. 예를 들면 $P(t) = p\Lambda^{-1}(t)$, $p > 1$ 로 설정하는 것도 한 방법이 될 수 있다. 이 목적함수가 최소값을 갖기 위하여 $u^*(t)$ 는

$$\left. \frac{\partial V(t)}{\partial u(t)} \right|_{u(t) = u^*(t)} = \begin{bmatrix} -J^T(x(t)) Q(t) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{(\sigma_m)}(t) \end{bmatrix} \\ + \Lambda(t)u(t) - u_{fb}^*(t) \end{bmatrix} \Bigg|_{u(t) = u^*(t)} = 0 \quad (15)$$

를 만족시켜야 한다. 목적함수 $V(t)$ 를 최소화시키는 것은 $y_i^{(\sigma_i)}$ 가 $y_i^{d(\sigma_i)}$ 로 부터 멀어지는 것과 입력 $u(t)$ 가 커지는 것을 동시에 방지하자는 것이다. 그리고 (15)를 다시 정리하면

$$-J^T(x^*(t)) Q(t) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} + \Lambda(t)u_{ff}^*(t) - (I - \Lambda(t))u_{fb}^*(t) = 0 \quad (16)$$

가 되어 이는 $S(x^*(t)) = J^T(x^*(t))Q(t)$ 라고 할 때 (12)와 같

은 형태이다. 따라서 (12)의 관계식을 위의 목적함수와 연관시킬 수 있다.

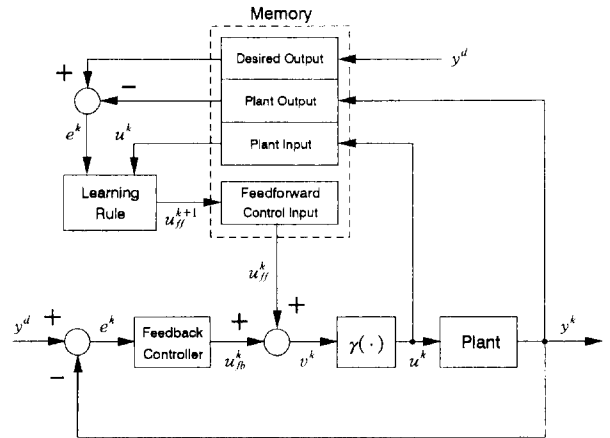


그림 1. 되먹임 제어를 가지는 반복학습 구조.
Fig. 1. Iterative learning scheme with feedback controller.

주 2.3 : 가정 (A2.4)는 플랜트나 제어기가 시간 $t \in [0, T]$ 에서 유한탈출시간(finite escape time)을 가지고 있지 않다면 만족될 수 있다. 즉, 플랜트나 제어기의 어떤 상태변수도 시간 $t \in [0, T]$ 에서 무한대로 발산하지 않는다는 것이다. 실제 세계에 존재하는 기계 시스템은 에너지 보존 법칙에 의해 유한한 시간 구간에서 발산하지 않는다. 따라서 (A2.4)는 실제 기계 시스템에서는 항상 만족될 수 있다.

III. 입력 갱신식 및 수렴 조건

되먹임 시스템 (1)-(6)에 대하여 다음과 같은 반복학습 제어를 위한 입력 갱신식을 제안한다.

$$u_{ff}^{k+1}(t) = [I - \Lambda(t)]u^k(t) + S(x^k(t)) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{k(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{k(\sigma_m)}(t) \end{bmatrix}, \quad t \in [0, T], \quad (17)$$

단, $S: R^n \rightarrow R^{m \times m}$ 는 가정 (A2.3)에서 정의된 함수이다. 이 식은 (12)와 관련시켜서 생각할 수 있다. 즉, (12)를 다시 쓰면

$$u_{ff}^*(t) = [I - \Lambda(t)]u^*(t) + S(x^*(t)) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix}, \quad t \in [0, T], \quad (18)$$

인데 위의 첨자 * 대신에 좌변은 $(k+1)$ 을 우변은 k 를 쓰면 (17)이 된다.

반복학습 제어의 구조는 [13]의 경우와 동일하며 그림 1에 나타내었다. 제한한 학습 구조에서는 다음 시행시의 앞먹임 제어 입력을 이전 시행시의 앞먹임 제어 입력이 아닌 실제 플랜트 입력으로부터 갱신한다. 만약 원하는 출력 $y^d(t)$ 가 구현 가능하고 플랜트와 제어기의 초기 조건 $x(0)$ 와 $z(0)$ 가 각각 $y^d(0) = h(x(0))$ 와 $z(0) = 0$ 를 만족한다면 $\Lambda(t) = 0, \forall t \in [0, T]$ 로 설정하여 반복학습을 행함으로써 [13]의 경우와 같이 $y^k(t) \rightarrow y^d(t), \forall t \in [0, T]$ 가 되는 완전한 학습 결과를 얻을 수 있을 것이다. 만약 원하는 출력 $y^d(t)$ 가 구현 가능하지 않다면 출력 $y^*(t)$ 가 구현가능하도

록 $\Lambda(t), t \in [0, T]$ 를 설정하여야 한다.

다음의 정리에서 입력 갱신식 (17)을 사용하는 제안한 학습 방법을 적용했을 때, 어떤 조건 하에서 주어진 $\Lambda(t)$ 에 대하여 실제 출력이 출력 $y^*(t)$ 를 추종하는 것이 보장된다는 것을 보여준다.

정리 : 앞의 (1)-(6)으로 주어지는 되먹임 시스템을 생각하자. 되먹임 시스템이 가정 (A2.1), (A2.2), 그리고 (A2.4)를 만족한다고 하고, 입력 갱신식 (17)을 사용하자. 원하는 출력 궤적 $y^d(t), t \in [0, T]$ 가 주어지고, 시스템 및 제어기의 초기 상태는 모든 k 에 대하여 항상 일정하다고 하자. 즉 모든 k 에 대하여 $x^k(0) = x(0), z^k(0) = z(0)$ 라 하자. 또한 어떤 주어진 $\Lambda(t)$ 에 대하여 (A2.3)을 만족하는 출력 $y^*(t), t \in [0, T]$ 가 존재한다고 하자. 그러면 행렬 함수 $S(x^k(t)) \in R^{m \times m}$ 가 모든 $k \geq 0$, 그리고 모든 시간 $t \in [0, T]$ 에서 부등식

$$\|I - \Lambda(t) - S(x^k(t))J(x^k(t))\| \leq \rho < 1 \quad (19)$$

을 만족한다면 $k \rightarrow \infty$ 함에 따라 매 시간 $t \in [0, T]$ 에서의 출력 $y^k(t)$ 가 평등하게(uniformly) 출력 $y^*(t)$ 로 수렴한다. 다시 말해서

$$\lim_{k \rightarrow \infty} \|y^*(t) - y^k(t)\| = 0, \quad \forall t \in [0, T] \quad (20)$$

가 된다.

증명 : 가정 (A2.4)에 의하면, 주어진 $y^d(t), t \in [0, T]$ 와 초기 조건 $x^k(0) = x(0), z^k(0) = z(0)$ 에 대하여 어떤 콤팩트 집합 $X \subset R^n$ 와 $Z \subset R^p$ 가 존재하여 모든 k 에 대해 $x^k(t) \in X, z^k(t) \in Z, \forall t \in [0, T]$ 를 만족한다. 증명을 위하여 우선 콤팩트 집합 X 와 Z 의 존재로부터 간단하게 유도될 수 있는 몇 가지 성질들을 먼저 기술한다.

(P3.1) 함수 $f(\cdot), G(\cdot)$, 그리고 $h(\cdot)$ 는 R^n 에서 해석 가능하기 때문에, 어떤 상수들 (local Lipschitz constants) f_0, g_0, h_0 가 존재하여 $\|f(x') - f(x'')\| \leq f_0 \|x' - x''\|, \|G(x') - G(x'')\| \leq g_0 \|x' - x''\|$, 그리고 $\|h(x') - h(x'')\| \leq h_0 \|x' - x''\|, \forall x', x'' \in X$ 를 만족한다.

(P3.2) 함수 $a(\cdot), B(\cdot), c(\cdot)$, 그리고 $D(\cdot)$ 는 R^p 에서 해석 가능하기 때문에, 어떤 상수들 a_0, b_0, c_0, d_0 가 존재하여 $\|a(z') - a(z'')\| \leq a_0 \|z' - z''\|, \|B(z') - B(z'')\| \leq b_0 \|z' - z''\|, \|c(z') - c(z'')\| \leq c_0 \|z' - z''\|$, 그리고 $\|D(z') - D(z'')\| \leq d_0 \|z' - z''\|, \forall z', z'' \in Z$ 를 만족한다.

(P3.3) 함수 $B(\cdot)$ 와 $D(\cdot)$ 가 R^p 에서, 그리고 함수 $G(\cdot)$ 가 R^n 에서 각각 해석 가능하므로 어떤 상수들 M_b, M_d , 그리고 M_g 가 존재하여 $\|B(z)\| \leq M_b, \|D(z)\| \leq M_d, \forall z \in Z$, 그리고 $\|G(x)\| \leq M_g, \forall x \in X$ 를 만족한다.

(P3.4) 함수 $S(\cdot)$ 는 R^n 에서 해석 가능하므로 어떤 상수들 s_0 와 M_s 가 존재하여 $\|S(x') - S(x'')\| \leq s_0 \|x' - x''\|, \forall x', x'' \in X$, 그리고 $\|S(x)\| \leq M_s, \forall x \in X$ 를 만족한다.

가정 (A2.1)에 의하면, 주어진 $y^d(t), x(0)$, 그리고 $\Lambda(t), t \in [0, T]$ 에 대해 출력 $y^*(t)$, 입력 $u^*(t)$, 그리고 상태변수 $x^*(t)$ 는 시간 $t \in [0, T]$ 에서 유일하게 결정된다. 상대차수의 정의와 가정 (A2.2)에 의해 $y_i^*(t) - y_i^k(t), 1 \leq i \leq m$ 의 미분들은

$$y_i^{*(\sigma)} - y_i^{k(\sigma)} = L_f^{\sigma} h_i(x^*) - L_f^{\sigma} h_i(x^k), \quad j < \sigma, \quad (21)$$

$$\begin{aligned} y_i^{*(\sigma)} - y_i^{k(\sigma)} &= L_f^{\sigma} h_i(x^*) - L_f^{\sigma} h_i(x^k) \\ &+ [L_{g_1} L_f^{\sigma-1} h_i(x^*), \dots, L_{g_m} L_f^{\sigma-1} h_i(x^*)] u^* \\ &- [L_{g_1} L_f^{\sigma-1} h_i(x^k), \dots, L_{g_m} L_f^{\sigma-1} h_i(x^k)] \gamma(v^k) \end{aligned} \quad (22)$$

와 같이 나타낼 수 있다. 입력 갱신식 (17)을 되먹임 시스템 (1)-(6)에 대입하고 (22)와 (11), 그리고 가정 (A2.3)의 (12)를 이용하면

$$\begin{aligned} &u_{jf}^*(t) - u_{jf}^{k+1}(t) \\ &= u^*(t) - u^k(t) - \left\{ S(x^*(t)) \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} \right. \\ &\quad \left. - \Lambda(t) u_{jf}^*(t) + (I - \Lambda(t)) u_{j0}^*(t) \right\} \\ &+ [S(x^*(t)) - S(x^k(t))] \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} \\ &- \Lambda(t) [u^*(t) - u^k(t)] \\ &- S(x^k(t)) J(x^k(t)) [u^*(t) - u^k(t)] \\ &- S(x^k(t)) [J(x^*(t)) - J(x^k(t))] u^*(t) \\ &- S(x^k(t)) \begin{bmatrix} L_f^{\sigma_1} h_1(x^*) - L_f^{\sigma_1} h_1(x^k) \\ \vdots \\ L_f^{\sigma_m} h_m(x^*) - L_f^{\sigma_m} h_m(x^k) \end{bmatrix} \\ &= [I - \Lambda(t) - S(x^k(t)) J(x^k(t))] [u^*(t) - \gamma(v^k(t))] \\ &+ [S(x^*(t)) - S(x^k(t))] \begin{bmatrix} y_1^{d(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{d(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} \\ &- S(x^k(t)) [J(x^*(t)) - J(x^k(t))] u^*(t) \\ &- S(x^k(t)) \begin{bmatrix} L_f^{\sigma_1} h_1(x^*) - L_f^{\sigma_1} h_1(x^k) \\ \vdots \\ L_f^{\sigma_m} h_m(x^*) - L_f^{\sigma_m} h_m(x^k) \end{bmatrix}, \quad t \in [0, T] \end{aligned} \quad (23)$$

를 얻을 수 있다. 함수 $f(\cdot), G(\cdot)$, 그리고 $h(\cdot)$ 는 R^n 에서 해석 가능하므로 $J(x)$ 와 $L_f^{\sigma_i} h_i(x), i = 1, \dots, m$ 의 각 항들은 R^n 에서 연속이다. 또한 $x^*(t)$ 와 $x^k(t), t \in [0, T]$ 는 (A2.4)에서 가정하였듯이 콤팩트 집합 X 에 속한다. 그러므로, 모든 $t \in [0, T]$ 와 k 에 대하여 다음을 만족하는 m_1 과 m_2 가 존재함을 알 수 있다.

$$\|J(x^*(t)) - J(x^k(t))\| \leq m_1 \|x^*(t) - x^k(t)\|, \quad (24)$$

그리고

$$\left\| \begin{bmatrix} L_f^{\sigma_1} h_1(x^*) - L_f^{\sigma_1} h_1(x^k) \\ \vdots \\ L_f^{\sigma_m} h_m(x^*) - L_f^{\sigma_m} h_m(x^k) \end{bmatrix} \right\| \leq m_2 \|x^*(t) - x^k(t)\|. \quad (25)$$

가정 (A2.3)에 의하면 $u^*(t)$ 는 항상 입력 포화 한계 내에 존재하고, $|u_i^*(t) - \gamma_i(v_i^k(t))| \leq |u_i^*(t) - v_i^k(t)|, \forall i$ 가 성립하므로, 다음의 부등식이 항상 성립한다.

$$\|u^*(t) - \gamma(v^k(t))\| \leq \|u^*(t) - v^k(t)\|, \quad \forall t \in [0, T]. \quad (26)$$

앞먹임 입력 오차를 $\delta u_{jf}^k(t) = u_{jf}^*(t) - u_{jf}^k(t)$, 그리고 되먹임 입력 오차를 $\delta u_{j0}^k(t) = u_{j0}^*(t) - u_{j0}^k(t)$ 라 하고, 시스템 상태변수 오차를 $\delta x^k(t) = x^*(t) - x^k(t)$, 그리고 제어기 상태변수

오차를 $\delta z^k(t) = z^*(t) - z^k(t)$ 라 정의하자. 우선 (23)의 양변에 노음을 취하고, (1)-(6), (P3.1)-(P3.4), (19), (24), (25), 그리고 (26)을 이용하면 다음과 같이 된다.

$$\begin{aligned} & \|\delta u_{ff}^{k+1}(t)\| \\ & \leq \|I - A(t) - S(x^k(t))f(x^k(t))\| \|u^*(t) - \gamma(v^k(t))\| \\ & \quad + \|S(x^k(t))\| m_1 \|\delta x^k(t)\| \|u^*(t)\| \\ & \quad + \|S(x^*(t)) - S(x^k(t))\| m_{ye} + \|S(x^k(t))\| m_2 \|\delta x^k(t)\| \\ & \leq \rho \|u^*(t) - v^k(t)\| + M_s m_1 m_{u^*} \|\delta x^k(t)\| \\ & \quad + s_0 m_{ye} \|\delta x^k(t)\| + M_s m_2 \|\delta x^k(t)\| \\ & \leq \rho \|\delta u_{ff}^k(t)\| + \rho \|c(z^*(t)) - c(z^k(t))\| \\ & \quad + \rho \|D(z^*(t)) - D(z^k(t))\| \|y^d(t)\| \\ & \quad + \rho \|D(z^*(t)) - D(z^k(t))\| \|y^*(t)\| \\ & \quad + \rho \|D(z^k(t))\| \|y^*(t) - y^k(t)\| \\ & \quad + (s_0 m_{ye} + M_s m_1 m_{u^*} + M_s m_2) \|\delta x^k(t)\| \\ & \leq \rho \|\delta u_{ff}^k(t)\| + \rho (c_0 + d_0 m_{yd} + d_0 m_{y^*}) \|\delta z^k(t)\| \\ & \quad + (\rho M_d h_0 + s_0 m_{ye} + M_s m_1 m_{u^*} + M_s m_2) \|\delta x^k(t)\| \\ & \leq \rho \|\delta u_{ff}^k(t)\| + d_1 (\|\delta x^k(t)\| + \|\delta z^k(t)\|), \end{aligned} \tag{27}$$

단, $m_{ye} = \max_{1 \leq i \leq m} [\sup_{t \in [0, T]} \|y_i^{k(\sigma)}(t) - y_i^{*(\sigma)}(t)\|]$, $m_{u^*} = \sup_{t \in [0, T]} \|u^*(t)\|$, $m_{yd} = \sup_{t \in [0, T]} \|y^d(t)\|$, $m_{y^*} = \sup_{t \in [0, T]} \|y^*(t)\|$ 이고, $d_1 = \max\{\rho(c_0 + d_0 m_{yd} + d_0 m_{y^*}), \rho M_d h_0 + s_0 m_{ye} + M_s m_1 m_{u^*} + M_s m_2\}$ 이다. 모든 $k \geq 0$ 에서 $x^k(0) = x(0)$ 이고 $z^k(0) = z(0)$ 이므로, 되먹임 시스템 (1)-(6)의 상태변수들이 다음을 만족한다.

$$\begin{aligned} & x^*(t) - x^k(t) \\ & = \int_0^t [\{f(x^*(\tau)) - f(x^k(\tau))\} \\ & \quad + \{G(x^*(\tau)) - G(x^k(\tau))\} u_{ff}^*(\tau) \\ & \quad + G(x^k(\tau)) (u_{ff}^*(\tau) - u_{ff}^k(\tau)) \\ & \quad + \{G(x^*(\tau)) - G(x^k(\tau))\} c(z^*(\tau)) \\ & \quad + G(x^k(\tau)) \{c(z^*(\tau)) - c(z^k(\tau))\} \\ & \quad + \{G(x^*(\tau)) - G(x^k(\tau))\} D(z^*(\tau)) y^d(\tau) \\ & \quad + G(x^k(\tau)) \{D(z^*(\tau)) - D(z^k(\tau))\} y^d(\tau) \\ & \quad - \{G(x^*(\tau)) - G(x^k(\tau))\} D(z^*(\tau)) y^*(\tau) \\ & \quad - G(x^k(\tau)) \{D(z^*(\tau)) - D(z^k(\tau))\} y^*(\tau) \\ & \quad - G(x^k(\tau)) D(z^k(\tau)) \{h(x^*(\tau)) - h(x^k(\tau))\}] d\tau, \end{aligned} \tag{28}$$

$$\begin{aligned} & z^*(t) - z^k(t) = \int_0^t [\{a(z^*(\tau)) - a(z^k(\tau))\} \\ & \quad + \{B(z^*(\tau)) - B(z^k(\tau))\} y^d(\tau) \\ & \quad - \{B(z^*(\tau)) - B(z^k(\tau))\} y^*(\tau) \\ & \quad - B(z^k(\tau)) (y^*(\tau) - y^k(\tau))] d\tau. \end{aligned} \tag{29}$$

위의 (28)과 (29)의 양변에 노음을 취하고 성질 (P3.1)-(P3.3)을 이용하면

$$\begin{aligned} \|\delta x^k(t)\| & \leq \int_0^t \{ [f_0 + g_0(m_{uff^*} + m_c) + g_0 M_d(m_{yd} + m_{y^*}) \\ & \quad + M_g M_d h_0] \|\delta x^k(\tau)\| \\ & \quad + M_g(c_0 + d_0 m_{yd} + d_0 m_{y^*}) \|\delta z^k(\tau)\| \\ & \quad + M_g \|\delta u_{ff}^k(\tau)\| \} d\tau, \end{aligned} \tag{30}$$

$$\begin{aligned} \|\delta z^k(t)\| & \leq \int_0^t \{ (a_0 + M_b m_{yd} + M_b m_{y^*}) \|\delta z^k(\tau)\| \\ & \quad + M_b h_0 \|\delta x^k(\tau)\| \} d\tau \end{aligned} \tag{31}$$

를 얻을 수 있다. 단, $m_{uff^*} = \sup_{t \in [0, T]} \|u_{ff}^*(t)\|$ 이고, $m_c = \sup_{t \in [0, T]} \|c(z^*(t))\|$ 이다. 여기서 (30)과 (31)을 서로 더하면

$$\begin{aligned} & \|\delta x^k(t)\| + \|\delta z^k(t)\| \\ & \leq \int_0^t \{ d_2 (\|\delta x^k(\tau)\| + \|\delta z^k(\tau)\|) + M_g \|\delta u_{ff}^k(\tau)\| \} d\tau \end{aligned} \tag{32}$$

이 얻어진다. 단, $d_2 = \max\{f_0 + g_0(m_{uff^*} + m_c) + g_0 M_d(m_{yd} + m_{y^*}) + M_g M_d h_0 + M_b h_0 M_g(c_0 + d_0 m_{yd} + d_0 m_{y^*}) + a_0 + M_b m_{yd} + M_b m_{y^*}\}$ 이다. 앞의 (32)에 Gronwall의 보조정리를 적용하면

$$\|\delta x^k(t)\| + \|\delta z^k(t)\| \leq \int_0^t M_g \|\delta u_{ff}^k(\tau)\| e^{d_2(t-\tau)} d\tau \tag{33}$$

와 같이 된다. 이 때 (27)과 (33)의 결과를 서로 조합하면

$$\|\delta u_{ff}^{k+1}(t)\| \leq \rho \|\delta u_{ff}^k(t)\| + d_1 M_g \int_0^t \|\delta u_{ff}^k(\tau)\| e^{d_2(t-\tau)} d\tau \tag{34}$$

를 얻을 수 있다. 위의 (34)의 양변에 $e^{-\lambda t} (\lambda > d_2)$ 를 곱하고 (8)의 λ -노음의 정의를 이용하면 다음과 같이 된다.

$$\begin{aligned} & \|\delta u_{ff}^{k+1}(t)\|_\lambda \\ & \leq \rho \|\delta u_{ff}^k(t)\|_\lambda \\ & \quad + \sup_{t \in [0, T]} \left[d_1 M_g \int_0^t e^{(d_2 - \lambda)(t-\tau)} d\tau \right] \|\delta u_{ff}^k(t)\|_\lambda \\ & \leq \left[\rho + \frac{d_1 M_g (1 - e^{-(\lambda - d_2)T})}{\lambda - d_2} \right] \|\delta u_{ff}^k(t)\|_\lambda \\ & \leq \bar{\rho} \|\delta u_{ff}^k(t)\|_\lambda. \end{aligned} \tag{35}$$

단, $\bar{\rho} = \rho + \frac{d_1 M_g (1 - e^{-(\lambda - d_2)T})}{\lambda - d_2}$ 이다. $\rho < 1$ 이므로, $\bar{\rho} < 1$ 이 되게 하는 $\lambda > d_2$ 가 항상 존재한다. 이런 λ 에 대해 $k \rightarrow \infty$ 함에 따라 모든 $t \in [0, T]$ 에서 $\|\delta u_{ff}^k(t)\|_\lambda \rightarrow 0$ 가 됨을 쉽게 보일 수 있다. 그러므로, λ -노음의 정의로부터 $k \rightarrow \infty$ 함에 따라 $\sup_{t \in [0, T]} \|\delta u_{ff}^k(t)\| \rightarrow 0$ 가 됨을 알 수 있다. 따라서, (33)으로부터, $k \rightarrow \infty$ 함에 따라 $\sup_{t \in [0, T]} \|x^*(t) - x^k(t)\| \rightarrow 0$ 이고 $\sup_{t \in [0, T]} \|z^*(t) - z^k(t)\| \rightarrow 0$ 가 된다. 그런데 $\|y^*(t) - y^k(t)\| \leq h_0 \|x^*(t) - x^k(t)\|$ 이므로, 출력 오차 $y^*(t) - y^k(t)$ 는 $k \rightarrow \infty$ 함에 따라 모든 $t \in [0, T]$ 에서 평등하게 (uniformly) 0으로 수렴한다. ■

주 3.1 : 참고문헌 [13]에서는 원하는 출력 $y^d(t)$ 가 시간 구간 $t \in [0, T]$ 에서 주어졌을 때 이를 완전하게 추종하게 하는 입력 $u^d(t)$, $t \in [0, T]$ 가 존재한다고 가정하고 이를 구하는 문제를 다루었다. 이 때 플랜트의 초기 조건 $x^k(0)$ 는 $y^d(0) = h(x^d(0))$ 를 만족하는 $x^d(0)$ 와 매 시행마다 일치해야 한다. 본 논문의 결과는 [13]의 확장이라고 할 수 있

는데, $\Lambda(t) = 0, \forall t \in [0, T]$ 이고 $x^k(0) = x^d(0)$ 인 경우 [13]의 결과와 거의 일치한다. 본 논문에서는 플랜트의 초기 조건 $x^k(0)$ 가 $x^d(0)$ 와 동일해야 할 필요는 없고 단지 $x^k(0) = x(0)$ 와 같이 매 시행시마다 동일하기만 하면 된다. 만약 원하는 출력 $y^d(t), t \in [0, T]$ 가 구현가능하다면 입력 가중치를 $\Lambda(t) = 0, \forall t \in [0, T]$ 로 설정할 수 있는데, 이 때 (12)는

$$S(x^*(t)) \begin{bmatrix} y_1^{\alpha(\sigma_1)}(t) - y_1^{*(\sigma_1)}(t) \\ \vdots \\ y_m^{\alpha(\sigma_m)}(t) - y_m^{*(\sigma_m)}(t) \end{bmatrix} + u_{fb}^*(t) = 0 \quad (36)$$

와 같이 표현된다. 위 식의 $u_{fb}^*(t)$ 가 식 (3)과 (4)에서 보듯이 입력이 $e^*(t) = y^d(t) - y^*(t)$ 인 제어기의 출력이므로 (36)은 최종 학습된 출력 오차의 운동방정식이라 할 수 있다. 따라서 (36)이 안정하다면 $x^k(0) = x^*(0) = x(0) \neq x^d(0)$ 이어서 생기는 출력 오차 $y^d(t) - y^*(t)$ 는 시간이 경과함에 따라 줄어들게 될 것이다.

주 3.2 : 되먹임 제어식 (4)를 다음과 같이 바꿈으로써 사용 가능한 되먹임 제어기의 범위를 좀 더 넓힐 수 있다 [13].

$$u_{fb}(t) = c(z(t)) + \sum_{i=1}^m D_i(z(t)) \hat{e}_i(t), \quad (37)$$

단, 함수 $D_i: R^p \rightarrow R^{m \times \sigma_i}, i = 1, \dots, m$ 는 각각의 정의영역에서 부분적으로(piecewise) 연속이고, $\hat{e}_i = [e_i, \dot{e}_i, \dots, e_i^{(\sigma_i - 1)}]^T, i = 1, \dots, m$ 와 같이 정의된다. 이것을 (4)와 비교할 때 (37)은 우변의 두 번째 항에 약간의 변화가 있는데, 위의 정리의 증명 과정 중 일부만 수정하면 된다. 따라서 위의 정리는 (4)가 (37)로 바뀌어도 여전히 성립한다. 이러한 수정으로 시스템 (1)의 상대차수가 1보다 클 때, 되먹임 제어 입력을 발생시키는데 출력 오차의 미분항들을 이용할 수 있게 된다. PD 제어기 등이 이러한 되먹임 제어기의 예가 될 수 있다.

IV. 시뮬레이션 예제 및 결과

이 절에서는 제안한 반복학습 제어 방법이 로봇 제어에 효율적으로 적용될 수 있음을 보이기 위해 로봇트 머니플레이터의 궤적 추종 제어 문제를 생각해 보겠다. 자유도가 m 인 로봇트 머니플레이터의 운동 방정식은 다음과 같이 주어지는 Lagrange-Euler 방정식으로 나타낼 수 있다.

$$M(\theta(t))\ddot{\theta}(t) + c(\theta(t), \dot{\theta}(t)) + g(\theta(t)) = \tau(t), \quad (38)$$

단, $\theta(t)$, $M(\theta)$, $c(\theta, \dot{\theta})$, $g(\theta)$, 그리고 $\tau(t)$ 는 로봇트 팔의 축 변수, 관성 행렬, 코리올리(Coriolis)력 및 원심력 벡터, 중력에 의한 힘벡터, 그리고 제어 토크를 각각 나타낸다. 상태변수를 $x(t) = [\theta^T(t) \ \dot{\theta}^T(t)]^T$ 로, 플랜트 입력을 $u(t) = \tau(t)$ 로, 그리고 플랜트 출력을 $y(t) = \theta(t)$ 로 놓으면 위의 운동방정식 (38)을 (1)과 같은 상태변수 방정식 꼴로 바꿀 수 있다. 이 때

$$f(x(t)) = \begin{bmatrix} \dot{\theta}(t) \\ -M^{-1}(\theta(t))\{c(\theta(t), \dot{\theta}(t)) + g(\theta(t))\} \end{bmatrix},$$

$$G(x(t)) = \begin{bmatrix} 0 \\ M^{-1}(\theta(t)) \end{bmatrix}, \text{ 그리고 } h(x(t)) = \theta(t)$$

이다. 단, $M(\theta)$ 는 비특이(nonsingular) 행렬이다. $y(t)$ 를 두 번 미분하면

$$\dot{y}(t) = \dot{\theta}(t) = \begin{bmatrix} L_j h_1(x) \\ \vdots \\ L_j h_m(x) \end{bmatrix}, \quad (39)$$

$$\ddot{y}(t) = \ddot{\theta}(t) = -M^{-1}(\theta(t))\{c(\theta(t), \dot{\theta}(t)) + g(\theta(t))\} + M^{-1}(\theta(t))u(t)$$

$$= \begin{bmatrix} L_j^2 h_1(x) \\ \vdots \\ L_j^2 h_m(x) \end{bmatrix} + \begin{bmatrix} L_{g_1} L_j h_1(x) & \dots & L_{g_m} L_j h_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_j h_m(x) & \dots & L_{g_m} L_j h_m(x) \end{bmatrix} u(t) \quad (40)$$

와 같이된다. 이 때, 시스템의 상대차수는 $\{\sigma_1, \dots, \sigma_m\} = (2, \dots, 2)$ 이고 비간섭(decoupling) 행렬은 정의에 의해 $J(x(t)) = M^{-1}(\theta(t))$ 로 나타난다.

되먹임 제어기로는 비례 이득이 K_p 이고 미분 이득이 K_d 인 PD 제어기를 사용한다. 그러면 제어 입력 $u(t)$ 는

$$u(t) = u_{fb}(t) + u_{ff}(t) = K_p(\theta^d(t) - \theta(t)) + K_d(\dot{\theta}^d(t) - \dot{\theta}(t)) + u_{ff}(t) \quad (41)$$

와 같이 주어진다. 여기서 $\theta^d(t) (= y^d(t))$ 는 로봇트 머니플레이터의 각 축이 따라가야할 원하는 출력 궤적이다. 유한한 시간구간 $t \in [0, T]$ 에서 원하는 출력 궤적 $\theta^d(t)$ 와 가중치 함수 $\Lambda(t)$ 가 주어지고 이 $\Lambda(t)$ 에 대해 가정 (A2.3)의 (12)에서와 같이

$$S(x^*(t))(\ddot{\theta}^d(t) - \ddot{\theta}^*(t)) - \Lambda(t)u_{ff}^*(t) + (1 - \Lambda(t))u_{fb}^*(t) = 0 \quad (42)$$

을 만족하는 구현 가능한 출력 $\theta^*(t), t \in [0, T]$ 가 존재한다고 하자. 제어의 목적은 충분히 작은 임의의 ϵ 오차 한계가 주어졌을 때

$$\|\theta^*(t) - \theta(t)\| < \epsilon, \forall t \in [0, T] \quad (43)$$

를 만족하는 앞먹임 입력 토크 $u_{ff}(t)$ 를 구하고자 하는 것이다. 그러나 실제로는 $\theta^*(t)$ 를 미리 알 수 없으므로 임의의 큰 수 N 이 주어졌을 때

$$\|\theta^k(t) - \theta^{k+j}(t)\| < \epsilon, \forall t \in [0, T], \forall j = 1, 2, \dots, N \quad (44)$$

을 처음으로 만족하는 k 에 대해 $\theta^* \approx \theta^k$ 로 근사화하기로 하겠다.

입력갱신식 (17)에 기초한 학습 법칙은 다음과 같다.

$$u_{ff}^{k+1}(t) = (I - \Lambda(t))u^k(t) + \alpha \hat{M}(\theta^k(t))(\ddot{\theta}^d(t) - \ddot{\theta}^k(t)), t \in [0, T], \quad (45)$$

단, $\hat{M}(\theta^k(t))$ 는 관성함수의 추정치이고, α 는 학습률을 나타내는 상수이다. 입력 갱신식 (17)과 비교하면 $S(x^k(t)) = \alpha \hat{M}(\theta^k(t))$ 이고 따라서 (42)에서 $S(x^*(t)) = \alpha \hat{M}(\theta^*(t))$ 임을 알 수 있다. 위의 (43)을 실현할 수 있게 하는 수렴 조건은 다음과 같이 주어진다.

$$\|I - \Lambda(t) - \alpha \hat{M}(\theta^k(t))M^{-1}(\theta^k(t))\| < 1, \quad \forall t \in [0, T], \forall k. \quad (46)$$

컴퓨터 시뮬레이션을 위해 각 축의 실제 길이와 무게가 각각 $\ell_1 = \ell_2 = 0.5\text{m}$ 와 $m_1 = m_2 = 2.0\text{kg}$ 인 그림 2와 같은 2축 로봇트 머니플레이터를 생각하겠다.

시스템을 모델화할 때는 모델링 오차가 존재하게 되는데, 여기서는 각 축의 길이와 무게를 각각 $\hat{\ell}_1 = \hat{\ell}_2 = 0.6\text{m}$ 와

$\hat{m}_1 = \hat{m}_2 = 1.8\text{kg}$ 으로 추정하였다고 하자. 즉, 실제보다 10~20%의 오차를 가지는 것으로 가정하였다. 그리고 되먹임 이득 K_p 와 K_d 를

$$K_p = \begin{bmatrix} 60 & 0 \\ 0 & 10 \end{bmatrix}, K_d = \begin{bmatrix} 8 & 0 \\ 0 & 2.5 \end{bmatrix} \quad (47)$$

로 설정하였다. 그리고 최대 허용 오차 한계를 $\epsilon = 0.001$ [radian] 으로, 학습률을 $\alpha = 0.75$ 로 정하였다. 또한 10초 동안에 대한 원하는 출력 궤적을 그림 3의 실선과 같이 임의의 모양을 가지도록 발생시키고, 최초의 앞먹임 입력은 모든 시간 $t \in [0, T]$ 에 대하여 $u_{ff}^0(t) = 0$ 이 되도록 인가하였다.

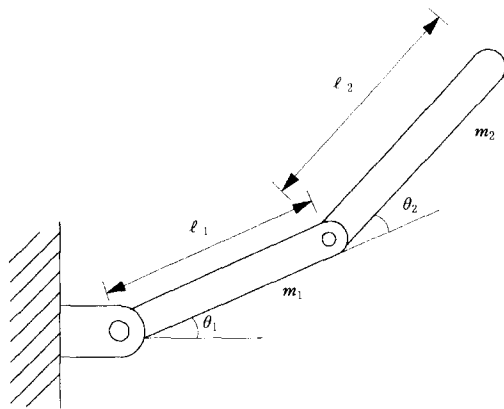
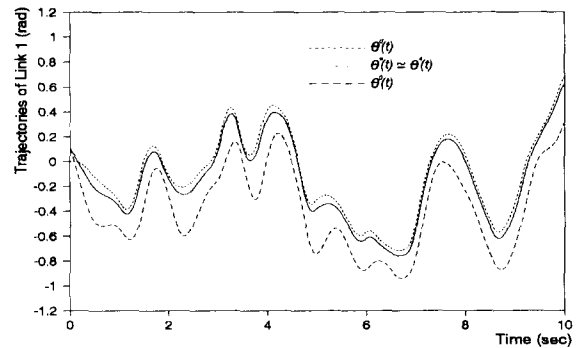


그림 2. 2축 로봇트 머니플레이터.
Fig. 2. A 2-link robot manipulator.

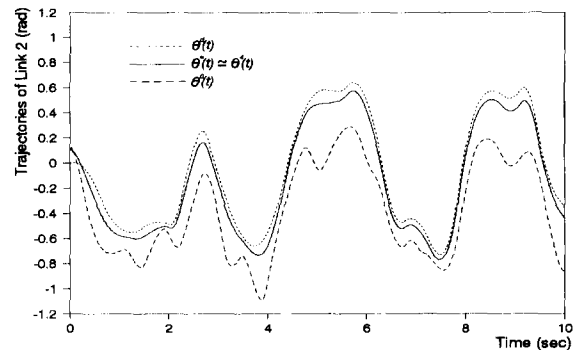
이상의 조건은 [13]의 예제에서와 동일한데, [13]의 경우에는 입력 포화 한계를 충분히 크게 설정함으로써 원하는 출력 $\theta^d(t)$ 가 구현 가능하도록 하였다. 그러나 본 예제에서는 원하는 출력 $\theta^d(t)$ 가 구현 가능하지 않은 경우에도 $\Lambda(t)$ 를 도입함으로써 구현 가능한 출력 $\theta^*(t)$ 로 실제 출력이 수렴한다는 것을 보여주기 위한 것이므로, 입력 포화 한계를 적당히 작은 값을 가지면서 $\theta^d(t)$ 가 구현 가능하지 않게 되는 값으로서 $\bar{v}_1 = 29$ 와 $\bar{v}_2 = 8.5$ 로 설정하였다. 또한 본 논문에서는 매 시행시의 초기 조건 $x^k(0)$ 가 $x^d(0)$ 와 같을 필요가 없고 단지 모든 k 에 대해 $x^k(0) = x(0)$ 이기만 하면 되므로 여기서는 링크 1의 초기 상태 조건을 $x_1(0) = \theta_1(0) = 0.1$ 로, 그리고 링크 2의 초기 상태 조건을 $x_2(0) = \theta_2(0) = 0.1$ 로 설정하여 항상 일정한 초기 오차를 가지는 것으로 가정하였다.

현재의 상태에서는 어떤 입력 가중치 함수 $\Lambda(t)$ 를 선택해야 출력 $\theta^*(t)$ 가 구현 가능할 지를 알 수 없다. 그러나 $\Lambda(t)$ 가 너무 크면 수렴 조건 (46)을 만족하기가 어려워진다. 그림 3에 $\Lambda(t) = 0.16I$ 로 설정하였을 때 반복학습을 통한 출력의 수렴 과정을 보였다. $N = 10$ 으로 놓았을 때 $k = 4$ 에서 (44)를 만족하는 수렴 결과가 확인되었다. 따라서 $\theta^*(t) \approx \theta^k(t) = \theta^d(t)$ 로 근사화한다. 점선은 원하는 출력 $\theta^d(t)$ 이며 실선은 실제로 학습된 출력 $\theta^*(t)$ 이다. 그림 4의 실선은 이 때의 입력 $u^*(t)$ 이며 점선은 입력 포화 한계가 충분히 클 때에 학습되는 원하는 입력 $u^d(t)$ 이다. 입력 $u^*(t)$ 가 $u^d(t)$ 보다는 크기가 충분히 작으면서도 그림 3과

그림 4에서 보듯이 원하는 출력 $\theta^d(t)$ 와 실제 수렴한 출력 $\theta^*(t)$ 사이의 오차가 그리 크지 않으면서 입력의 크기는 상당히 작아진 것을 알 수 있다.



(a)



(b)

그림 3. $\Lambda(t) = 0.16I$ 일 때의 출력의 수렴 과정
(a) 링크 1 (b) 링크 2.

Fig. 3. The converging trends of outputs when $\Lambda(t) = 0.16I$ (a) Link 1 (b) Link 2.

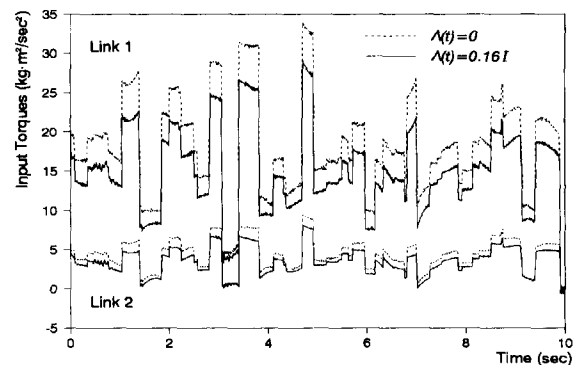


그림 4. 학습된 입력의 궤적.

Fig. 4. The trajectories of trained inputs.

위의 예에서는 $\Lambda(t)$ 를 모든 시간구간 $t \in [0, T]$ 에서 일률적으로 동일하게 설정하였지만 $u^d(t)$ 가 입력 포화 한계를 넘지 않는 시간 구간에 대해서는 개별적으로 그 구간에서 $\lambda_i(t) = 0$ 로 설정하여 $\theta_i^*(t)$ 가 $\theta_i^d(t)$ 를 최대한 추종하도록 하는 것이 좋다. 그림 4을 참고로 하여 입력 포화 한계가 각각 $\bar{v}_1 = 25$, $\bar{v}_2 = 7.5$ 를 넘지 않도록 하면서 가능한 많은 영역에서 $\theta_i^*(t)$ 가 $\theta_i^d(t)$ 를 정확히 추종하도록 하기 위해 그림 5에 보인 것과 같이 $\Lambda(t)$ 를 선정하였다. 이 값은

미리 알 수 있는 것은 아니며 시뮬레이션을 통해 시행착오를 거치면서 얻어진 결과이다. $N=10$ 으로 놓았을 때 $k=5$ 에서 (44)를 만족하는 수렴 결과가 확인되었다. 따라서 $\theta^*(t) \approx \theta^k(t) = \theta^5(t)$ 로 근사화한다. 원하는 출력과 실제 출력 사이의 오차 $\theta^d(t) - \theta^*(t)$ 와 실제로 플랜트에 인가된 입력 $u^*(t)$ 를 그림 6과 그림 7에 각각 나타내었다.

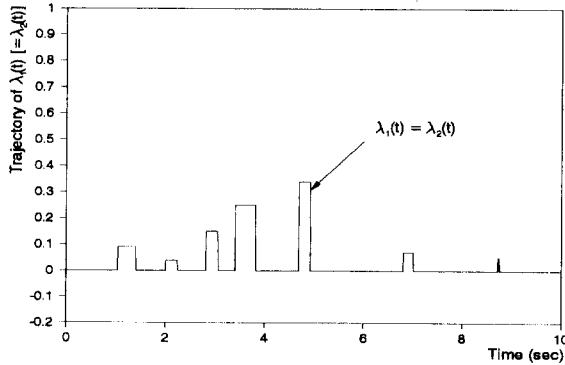


그림 5. 입력 가중치 함수 $\Lambda(t) = \text{diag}[\lambda_1, \lambda_2]$.

Fig. 5. Input penalty function $\Lambda(t) = \text{diag}[\lambda_1, \lambda_2]$.

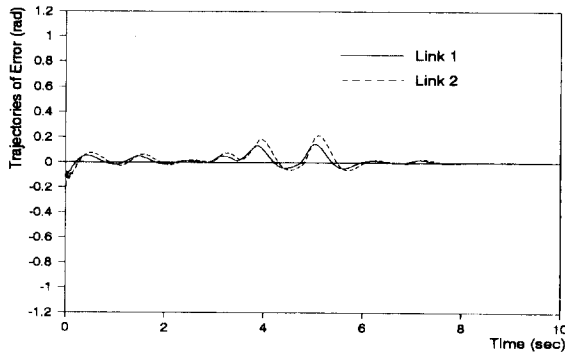


그림 6. $\Lambda(t)$ 가 그림 5와 같이 주어졌을 때 학습 후의 출력 오차.

Fig. 6. The trajectories of outputs after learning when $\Lambda(t)$ is given as in Fig. 5.

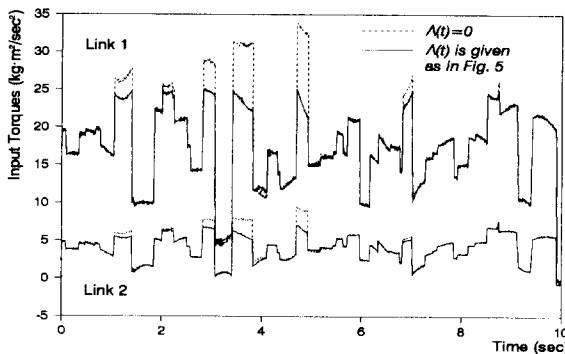


그림 7. $\Lambda(t)$ 가 그림 5와 같이 주어졌을 때 학습된 입력 궤적.

Fig. 7. The trajectories of trained inputs when $\Lambda(t)$ is given as in Fig. 5.

그림 6에서 초기 조건 불일치에 의한 출력 오차는 시간이 감에 따라 급격히 감소하고 있는 것을 볼 수 있으며, 또한 $\Lambda(t)=0$ 가 오래 지속되는 부분에서는 출력 오차가 거의 없음을 알 수 있다. 그리고 그림 7을 보면 학습된 입력의 크

기가 모두 입력 한계값 내에 들어온 것을 확인할 수 있는데, 따라서 만약 $\Lambda(t)$ 를 보다 더 잘 설정한다면 입력의 한계값 내에서 최대의 성능을 발휘하도록 학습이 가능하다는 것을 짐작할 수 있다. 이 시뮬레이션 결과로부터 본 논문에서 제안한 반복학습 방법에 의해 실제로 구현가능하지 않은 원하는 출력이라 하더라도 $\Lambda(t)$ 를 잘 설정함으로써 가능한 범위에서 그 출력을 추종하는 입력을 학습해 낼 수 있음을 확인하였다. 그러나 이러한 $\Lambda(t)$ 를 설정하는 방법에 대해서는 앞으로 더 많은 연구가 필요할 것으로 보인다.

V. 결론

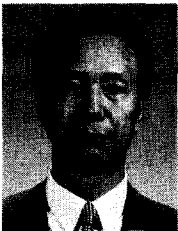
본 논문에서는 기존의 반복학습 제어에서의 기본적인 목표인 실제 출력이 원하는 출력을 정확하게 추종하게 하는 입력을 구해내는 조건을 완화하여, 실제 시스템이 추종하기에는 좀 무리한 원하는 출력 궤적이 주어졌다 할지라도 그 궤적에 근접하게 따라가면서 입력의 실제 크기는 작은 입력 궤적을 학습하는 방법에 대해 알아보았다. 이를 가능하게 하는 방법으로서 본 논문에서는 원하는 출력 궤적에 대한 추종성과 입력의 크기 사이에 가중치를 두고 이 가중치의 크기를 조정함으로써 입력의 크기에 대한 보상을 행하였다. 이렇게 함으로써 원래의 목적을 달성함을 물론 플랜트의 초기 상태가 원하는 초기 상태 출력을 낼 수 있도록 맞추어져 있지 않더라도 본 논문의 결과를 적용할 수 있게 되는 부수적인 결과도 얻었다.

시뮬레이션을 통하여 구현가능하지 않은 원하는 출력 궤적이 주어졌더라도 입력에 대한 가중치를 적당히 설정함으로써 실제 출력이 구현 가능한 출력으로 수렴함을 보였다. 이때 학습된 입력은 원하는 출력을 정확히 추종하기 위해 필요한 입력보다 크기는 작으면서 출력의 추종 오차는 그리 크지 않은 것을 확인할 수 있었다. 입력에 대한 가중치의 설정 방법은 앞으로 연구해야될 과제이다.

참고문헌

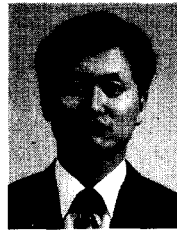
- [1] M. Uchiyama, "Formulation of high-speed motion pattern of a mechanical arm by trial," *Trans. Soc. Instrum. and Contr. Eng.* (in Japanese), vol. 14, no. 6, pp. 706-712, 1978.
- [2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robotic Systems*, vol. 1, no. 2, pp. 123-140, 1984.
- [3] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method," *IEEE Trans. System, Man, Cybern.*, vol. 18, pp. 126-134, 1988.
- [4] C. G. Atkeson, and J. McIntyre, "Robot trajectory learning through practice," *Proc. IEEE Conf. Robotics Autom.*, pp. 1737-1742, 1986.
- [5] J. E. Hauser, "Learning control for a class of nonlinear systems," *Proc. 26th IEEE Conf. Decision Contr.*, Los Angeles, CA, pp. 859-860, Dec., 1987.
- [6] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial conditions," *IEEE Trans. Automat. Contr.*, vol. 37, no. 1, pp. 110-114, 1992.
- [7] T. Kuc, K. Nam, and J. S. Lee, "An iterative lear-

- ning control of robot manipulators," *IEEE Trans. Robotics Autom.*, vol. 7, pp. 835-842, 1991.
- [8] T. Kuc, J. S. Lee, and K. Nam, "An iterative learning control theory for a class of nonlinear systems," *Automatica*, vol. 28, pp. 1215-1221, 1992.
- [9] S.-R. Oh, Z. Bien, and I.-H. Suh, "An iterative learning control method with application for robot manipulator," *IEEE Trans. Robotics Autom.*, vol. 4, no. 5, pp. 508-514, Oct., 1988.
- [10] T. Sugie and T. Ono, "An iterative learning control law for dynamical systems," *Automatica*, vol. 27, no. 4, pp. 729-732, 1991.
- [11] H.-S. Ahn, C.-H. Choi, and K.-B. Kim, "Iterative learning control for a class of nonlinear systems," *Automatica*, vol. 29, no. 6, pp. 1575-1578, Nov., 1993.
- [12] T.-J. Jang, H.-S. Ahn, and C.-H. Choi, "Iterative learning control for discrete-time nonlinear systems," *Int. J. Systems Sci.*, vol. 25, no. 7, pp. 1179-1189, July, 1994.
- [13] T.-J. Jang, C.-H. Choi, and H.-S. Ahn, "Iterative learning control in feedback systems," *Automatica*, vol. 31, no. 2, pp. 243-248, Feb., 1995.
- [14] T. Ishihara, K. Abe, and H. Takeda, "A Discrete-Time Design of Robust Iterative Learning Controllers," *IEEE Trans. System, Man, Cybern.*, vol. 22, pp. 74-84, 1992.
- [15] D.-H. Hwang, Z. Bien, and S.-R. Oh, "Iterative learning control method for discrete-time dynamic systems," *IEE Proc. Pt. D*, vol. 138, no. 2, pp. 139-144, 1991.
- [16] 황동환, 변증남, 오상록, "비선형 시스템의 반복 학습 제어 알고리즘에 관한 연구," 대한전기학회 논문지, 제 42권, 제 10호, pp. 112-119, 10, 1993.
- [17] A. Isidori, *Nonlinear Control Systems: An Introduction*, vol. 72, Springer-Verlag, Berlin, 1989.
- [18] H. Nijmeijer and Arzan van der Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, New York, 1990.
- [19] K. S. Fu, R. C. Gonzalez, and C. G. S. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York, pp. 98-102, 1987.



최 중 호

1970년 서울대학교 기계공학과 학사. 1975년 Univ. of Florida(미국) 산업 및 시스템공학과 석사. 1978년 동대학교 전기공학과 박사. 현재 서울대학교 전기공학부 교수. 주관심 분야는 학습 제어, CNC 제어기, 신경회로망 등임.



장 태 정

1986년 서울대학교 제어계측공학과 학사. 1988년 동대학원 제어계측공학과 석사. 1994년 동대학원 제어계측공학과 박사. 현재 강원대학교 제어계측공학과 전임 강사. 주관심 분야는 제어이론 및 응용, 신경회로망, 필드버스 등임.