반복학습제어 명령의 간단화와 스케일링

채 회 창*, 이 상 훈*, 박 명관**, 서 일 홍*
*한양대학교 정보통신대학원, **동양대학교 컴퓨터제어공학과
(+82-31-408-5802;FAX: 82-31-408-5803; E-mail:ihsuh@hanyang.ac.kr)

Simplification and Scaling of Iterative Learning Control Command

Hui-chang Chae*, Sang-hoon Lee*, Myung Kwan Park**, Il Hong Suh*
*The Graduate School of Information and Communication, Hanyang University
**Department of Computer Control Engineering, Dongyang University
(+82-31-408-5802;FAX:82-31-408-5803; E-mail:ihsuh@hanyang.ac.kr)

Abstract - ILC(Iterative Learning Control: 이하 ILC)는 현재 기계, 전기, 화학 등 많은 분야에 널리 적용되고 있다. ILC는 특히 반복적인 trajectory tracking Control 문제에 아주 효과적인 방법 중의 하나이다. 하지만 ILC는 메모리 기반의 scheme로서 trajectory tracking을 위해서는 많은 메모리를 요구하게 된다.

한편, 자세한 관찰에 의하면 인간의 팔, 다리 등의 관절의 움직임은 아주 정확하지가 않다. 이러한 사실로 미루어 인간이 정확한 모션을 취하는데 드는 비용을 줄이고자 모션 명령을 간단히 한다는 가정을 추론 해 낼 수 있다. 이러한 가정에 기초하여 우리는 ILC 명령을 간단히 하기 위해서 약간의 trajectory tracking의 정확성을 희생하는 메커니즘을 제안한다. 간단해진 ILC 명령은 적은 메모리 공간에 저장될 것이다. 또한, 로봇의 trajectory tracking을 위한 기존의 방법들은 아주 복잡할 뿐만 아니라 하나의 task의 수행만이 가능할 뿐 어떤 일반화의 방법도 제시하지 못하고 있다. 그래서 본 논문에서는 ILC 명령의 scaling에 대한 메커니즘을 제공하여 하나의 trajectory에 대해서 비슷한 모양이지만 다른 크기와 속도를 가지는 trajectory를 구현 할 수 있도록 하였다.

1. 서 론

ILC는 과거의 제어 정보-tracking error, 제어 입력 등-를 현재의 제어 action 생성에 여러 형태로 적용한다는 견지에서 현존하는 대부분의 제어 방법과는 다르다. ILC에는 두개의 부분이 존재하는데 하나는 메모리 부분으로 과거의 제어 정보를 저장하고 하나는 저장된 정보를 시스템의 수렴성과 안정성을 만족하는 방법에 이용하는 부분이다. ILC의 수렴성과 안정성은 컨트롤러의 디자인 시에 더 많은 사전 지식을 필요로 하는 여타 제어 방법에서는 쉽게 얻을 수 있는 것이 아니다[1]. ILC는 시스템에 대한 훨씬 적은 정보로도 바람직한 결과를 얻어 낼 수 있는 방법이다. ILC의 이러한 간단하고 효과적인 특징으로 인해서 ILC는 과거 수십 년간 많은 분야에서 관심의 대상이 되었다.

한편, 로봇을 인간의 곁으로 가져오고 인간과 상호 작용하는 임무를 수행하도록 하는데 있어서 human motor control strategy를 이용하는 것이 자연스러울 것이다. 기존의 많은 연구들은 human motor control strategy를 성공적으로 적용한 결과를 보여주고 있다[3]. 인간의 모션은 아주 정확하지가 않다는 관측 결과에서 모션의 정확성을 희생하는 대가로 간단화된 learning space를 가진다는 가정이 있다[4]. 또한 비슷한 모션의 그룹을 다룰 때 human motor control system은 각 모션을 위해서 learning process를 실행시키는 것이 아니라 그룹에서 하나의 모션에 대한 명령을 선택하여 그것을 스케일링 함으로써 다른 모션을 수행한다고 한다. 실제 human motor control system이 이러한 정확성과 learning space간의 trade off를 수행하는지 의심의 여지는 남아있지만 여러 가지 실제 연구는 이러한 가정을 뒷받침 해주고 있다[5]. 이러한 Simplification과 Scaling에 대한 메커니즘이 Equilibrium-point hypothesis에 기초 한 human motion command에 성공적으로 접목 되었다[2].

우리는 본 논문에서 trajectory tracking 문제에 효과적인 ILC command 에 Simplification과 Scaling에 대한 메커니즘을 접목하여 ILC 명령에 일반화의 방법을 제공하였다.

2. Iterative Learning Control

이전의 모션의 정보를 이용하고자 하는 아이디어는 Uchiyama(Uchiyama, 1978)에 의해서 제안되었다[6]. 하지만 learning control에 대한 정확한 이론은 Arimoto et al 과 Casalino and Bartolini, Craig등에 의해서 거의 동시에 만들어 졌다.

2.1 Iterative Learning Control Schemes

ILC 문제는 출력 $y(t)$가 미리 입력된 출력 $y_d(t)$에 수렴하도록 하는 적당한 제어 방법을 디자인 하는 것이다. 가장 간단한 것은 D-type ILC scheme 로 아래와 같다.

$$u_{i+1}(t)=u_i(t)+L\dot{e}_i(t) \quad \cdots\cdots\cdots (1)$$

여기서, L 은 ILC gain이고 i는 iteration을 나타낸다. 또 다른 ILC scheme는 previous cycle feedback을 이용하는 D-type외에 P-type으로 current cycle feedback (CCF)을 이용한 것이다.

$$u_{i+1}(t)=u_i(t)+L\dot{e}_{i+1}(t) \quad \cdots\cdots\cdots (2)$$

이 두개의 간단한 scheme는 복잡한 dynamic system이나 initial tracking error가 있는 문제에서는 그리 효과적이지 못하다[1][7]. 가장 최근의 scheme는 이러한 문제를 풀기 위해서 P-type과 D-type을 혼합한 PD-type scheme이다.

$$u_{i+1}(t)=u_i(t)+L_D\dot{e}_i(t)+L_P e_i(t) \quad \cdots\cdots\cdots (3)$$

3. Simplification and Scaling

3.1 Human motion command Simplification and Scaling scheme [2]

우리는 우선 human motion command에 Simplification과 Scaling mechanism을 성공적으로 접목한 논문의 내용을 살펴보겠다. simplification과 scaling에 앞서 먼저 정확히 모션을 tracking하는 command를 생성하는 과정이 필요하다. 아래 그림1은 정확한 motion tracking을 위한 scheme를 보여주고 있다.

알고리즘을 살펴보면 먼저, desired 한 motion은 그림 4와 같은 Two-link robot arm의 좌표계(joint space)로

변환되고 FNN을 통과하여 motion command($EP_d(t)$)를 생성한다. 이 $EP_d(t)$는 robot arm의 출력과 함께 robot arm에 걸리는 토크 $\tau = K_p(EP_d - q_t) - K_d\dot{q}_t$ 를 발생시키는데 사용 된다. 또한 robot arm의 출력은 FNN의 rule을 update하는데 사용된다[8].

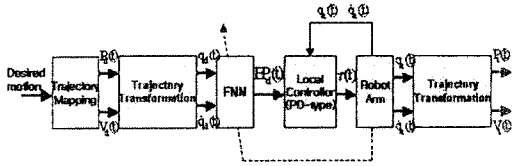


그림1. Learning scheme for accurate motion tracking

이렇게 생성된 motion command는 그림 2와 같은 scheme 로 simplification 된다. 여기서는 빗금 친 부분이 중요한 역할을 한다. 먼저 EB(error bound)는 임의의 작은 값으로 초기화된다. 각 sampling time마다 robot arm 의 출력과 desired 한 motion과의 cartesian error를 계산 E(error)에 계속 더한다. Decision maker에서는 E<EB이면 update gate를 닫아서 이전의 motion command를 $EP_s(t)$에 계속 넣게 되고 E>EB보다 커지면 update gate를 열고 현재의 motion command를 $EP_s(t)$에 넣는다.

또 하나의 scheme는 motion command scaling이다. scaling에 대한 scheme는 앞의 simplification의 scheme와 거의 유사한다. 다른 점이 있다면 error계산이 다음과 같이 scaled된 reference trajectories에 기초한다는 것이다.

$$r = c_1 t \quad \text{.....(4)}$$

$$\hat{q}_d(r) = (1 - c_1)q_{d0} + c_1 q_t(t) \quad \text{.....(5)}$$

그림3에 scaling에 대한 learning scheme가 나와 있다.

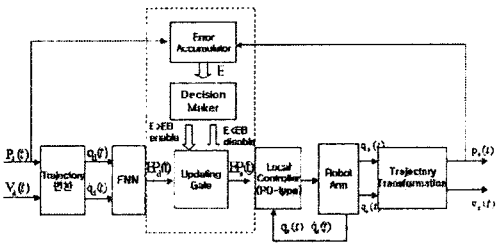


그림2. Learning scheme for motion command simplification.

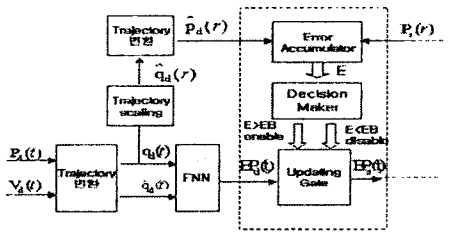


그림3. Learning scheme for motion command scaling.

3.2 ILC command Simplification and Scaling scheme

앞 절에서 우리는 human motion command를 성공적으로 simplification하고 scaling한 결과를 보았다. 하지만 앞의 scheme는 motion command 생성을 위해서 FNN이라는 복잡한 과정을 거치게 된다. 본 논문에서는 이 부분을 trajectory tracking에 아주 효율적인 ILC를 접목하여 정확한 motion을 구하는 ILC command를 생성 하고 여기에 simplification과 scaling에 대한 scheme를 접목하였다. ILC command의 Simplification 과 Scaling에 대한 scheme에 앞서서 trajectory를 정확히 tracking하는 ILC command을 생성해야 한다. 그림 5는 식 (3)의 PD-type ILC scheme의 Block Diagram을 보여주고 있다. Reference로 $q_d, \dot{q}_d$이 주어지고 시스템에서의 출력 $q, \dot{q}$ 와 $q_d, \dot{q}_d$ 사이의 의 에러에 linear controller가 연결되어 있다. ILC scheme는 linear controller에 적절한 gain을 사용할 경우 몇 번의 iteration안에 아주 만족할 만한 결과를 얻을 수 있다.

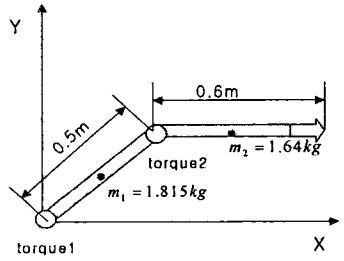


그림4. Two link robot arm.

Trajectory를 정확히 tracking하는 ILC command를 생성한 후 이를 simplification scheme에 사용한다. 이는 human motion command simplification scheme와 거의 유사하다. 다른 점은 간단화된 human motion command는 local controller에 입력되어서 torque를 발생시키지만 simplification된 ILC command는 PD-type ILC scheme를 통과하여 torque를 발생 시킨다는 점이다. 즉 simplification 된 ILC command $u_{sk}(t)$를 linear controller를 거쳐서 $u_{sk+1}(t)$를 구한 후 이것을 robot arm에 가해준다. 아래 그림6은 ILC command의 simplification에 대한 scheme를 나타낸다.

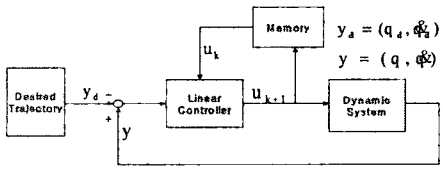


그림5. Block Diagram of PD-type ILC scheme

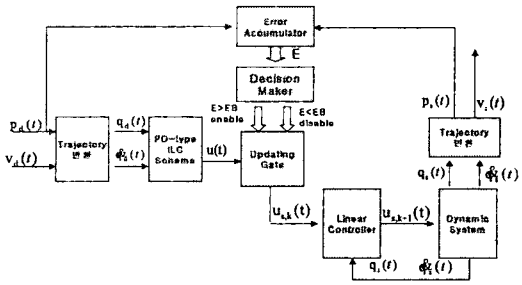


그림6. Learning scheme for ILC command simplification

Scaling scheme는 simplification과 거의 동일하다. 다른 것은 error accumulator에서 식 (6),(7)과 같이 scaleded 된 reference $\hat{P}_d(r)$과 출력 $P_t(r)$ 사이의 Cartesian error를 사용 한다는 것이다.

$$r = c_1 t \quad \text{.....(6)}$$

$$\hat{P}_d(r) = P_{d0} - c_1(P_{d0} - P_d(t)) \quad \text{.....(7)}$$

식 (7)은 (5)보다 더 정확한 스케일링 방법이다.
아래 그림7는 ILC command의 scaling에 대한 scheme를 나타낸다.

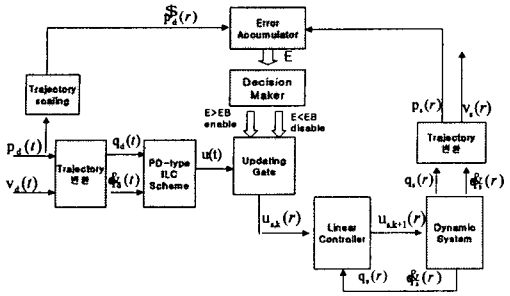


그림7. Learning scheme for ILC command scaling.

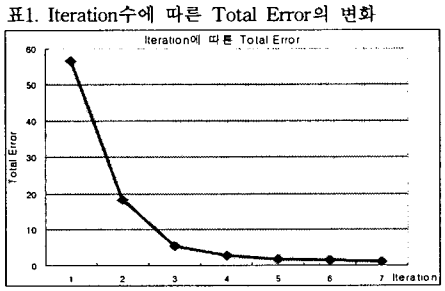
4. Simulation

본 논문에서 제안한 ILC 명령의 Simplification과 Scaling에 대한 mechanism 중에서 Scaling에 대한 시뮬레이션을 수행하였다. 시뮬레이션은 그림 5와 같은 Two link manipulator를 시스템으로 사용하였고 각 링크에 걸리는 토크를 입력(ILC 명령)으로 하였다.
아래 그림8 은 스케일링 시뮬레이션에 쓰인 원을 보여주고 있다. 작은 원은 원래 크기의 원이고 실선의 큰 원은 (7)식으로 1.5배 스케일링된 원이고 큰 점선의 원은 식 (5)로 스케일링된 원이다.

그림 8. 시뮬레이션에 reference원과 스케일된 원

스케일링에 앞서서 정확히 원을 추적하는 ILC 명령을 생성하여야 한다. 아래 표1은 reference원과 learning한 원과의 total 에러를 보여주고 있다.

표1. Iteration수에 따른 Total Error의 변화



Iteration 4후로는 거의 reference와 비슷해짐을 알 수 있다. 우리는 Iteration 4의 결과를 원을 정확히 추적하는 ILC command로 생각하였다.
아래 그림 9는 EB를 달리하였을 때의 스케일링 결과를 보여주고 있다. EB가 클수록 motion초기에는 에러가 작다가 motion이 끝나는 지점에서 에러가 많아지는 경향을 볼 수 있다. 이러한 경향은 EB를 크게 하더라도 총 에러는 증가하는 결과를 낳게 된다.
표2 는 스케일링 시에 EB(error bound)의 영향을 나타낸 것이다. EB가 증가 할수록 total error를 감소하다가 어느 정도 이상이 되면 다시 증가하고 있다.

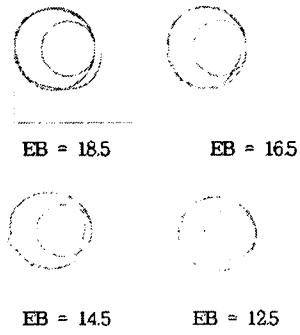
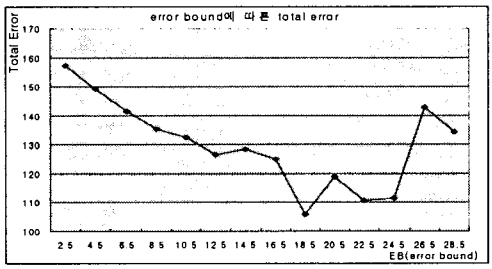


그림 9 여러 가지 EB에 따른 스케일링 결과

표2. EB(error bound)에 따른 Total Error



5. 결 론

본 논문에서는 ILC 명령의 Simpilification과 Scaling mechanism을 제안하고 Scaling에 대한 시뮬레이션을 성공적으로 보였다. 제안된 ILC 명령의 Simplification과 Scaling은 먼저 ILC 명령을 생성한 이후에 수행될 수 있다. ILC명령은 몇 번의 Iteration으로도 성공적으로 구할 수 있었다. 본 논문에서 제안된 Scaling mechanism은 큰 정확성을 요구하지 않는 태스크의 경우에는 효과적인 수행결과를 보였다. 우리는 앞으로 좀 더 정확한 스케일링이 가능한 mechanism에 대한 연구를 계속할 필요가 있다.

[참 고 문 헌]

[1] Zeungnam Bien and Jian-Xin Xu, "Iterative Learning Control -Analysis, Design, Integration and Applications," Kluwer Academic Publishers
[2] Kuu-Young Young and Shi-Huei Liu, "Robot Motion Command Simplification and Scaling," IEEE trans. on Systems, Man, And Cybernetics-Part B: Cybernetics, vol 32, No. 4, August 2002
[3] H.Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during mutijoint movements," science, vol 272, no.5258, pp.117-120,1996
[4] K.Y.Young and C.C.Fan, "An approach to simplify the learning space for robot learning control," Fyzzy Sets Syst., vol. 95, pp.23-38, Apr. 1998
[5] R.A.Schmidt and T.D.Lee, "Motor Control and Learning : A behavioral Emphasis, 3rd ed.," Champaign, IL: Human Kimetics, 1999
[6] Uchiyama, M.(1978). Fomation of high-speed motion pattern of a mechanical arm by trial, Trans. SICE(Society for Implementation and Control Engineers(in Japanese), 14: 706-712)
[7] Lee, K. H. and Bien, Z. (1991). "Initial Condition problem of learning control," IEEE Proceedings, Part-D, Control Theory and Applications, 138(6): 525-528
[8] H.R.Berenji and P. Khedkar, "Learning and tunning fuzzy logic controllers through reinforcement," IEEE Trans. on Neural Networks, vol. 3, pp. 724-740, Sept 1992