

신경망 제어에 의한 철심형 리니어모터의 추종성 향상 연구

송창규*(한국기계연구원 지능형정밀기계연구부), 김경호(한국과학기술원 기계공학과),
정재한(삼익 LMS 기술연구소)

Study for Improvement of Tracking Accuracy of the Feeding System with Iron Core Type Linear DC Motor by Neural Network Control

C. K. Song(KIMM), K. H. Kim(KAIST), J.H. Jung(Samick LMS)

ABSTRACT

The requirements for higher productivity call for high speed of the machine tool axes. Iron core type linear DC motor is growly accepted for a viable candidate of the high speed machine tool feed unit. LDM, however, has inherent disturbance force components: cogging and force ripple. These disturbance force directly affects tracking accuracy of the carriage and must be eliminated or reduced. Reducing motor ripple, this paper adapted the feedforward compensation method and neural network control. Experiments carried out on the linear motor test setup show that this control methods is usable in order to reduce the motor ripple.

Key Words : Linear Motor(리니어 모터), Iron Core(철심), Feeding System(이송계), Tracking Error(추적 오차), Feedforward(피드포워드), Neural Network(신경망제어), Cogging(코깅), Force Ripple(추력 리플)

1. 서론

현대 생산기술의 특징은 고생산성의 추구하고 대량 가공품 형상의 복잡성에 있다고 할 수 있다. CAD 로 설계된 복잡한 형상을 가공하며 고생산성을 달성하기 위해서는 생산 기계인 공작기계 각 축이 고속화되면서 동시에 높은 정도의 운동 정밀도를 갖는 것이 필연적이다.

고속 가공기의 이송계를 구성하는 기존의 방법은 회전모터와 볼스크류로 이루어지는 간접적 드라이브 방식이지만 속도와 가속도 및 정밀도에서 제한적이다. 반면 최근의 각종 국제 공작기계전시회에서 선보이고 있는 리니어모터에 사용한 직접구동 방식의 공작기계 축은 간접적 드라이브 방식에 비해 다음과 같은 장점을 갖고 있기 때문에 그 수요가 점점증하고 있다.

- 백래쉬가 없고 마찰이 작다.
- 가속도와 속도에 기계적 제한이 없다 : 속도는 엔코더의 밴드폭이나 전원 회로에 의해서만 제한된다.
- 기계적으로 간단하고 그로 인해 높은 신뢰성

과 높은 프레임 강성을 보인다.

공작기계 이송계로 가장 많이 사용되는 리니어모터는 리니어 DC 모터(linear dc motor, LDM)로서 그 중에도 철심에 코일을 감은 철심형이 무철심형에 비해 추력이 크기 때문에 공작기계에서의 채택 비율이 매우 높다. 하지만 철심형은 추력이 큰 대신에 이송계에 대해 외란이라 할 수 있는 모터 리플(motor ripple)이 발생하여 정밀도가 떨어지는 단점이 있다. 모터 리플에는 코깅(cogging)과 추력 리플(force ripple)의 2 가지가 있다. 코깅은 자기적인 외란이라 할 수 있고 추력 리플은 전자기적 영향에 의한 외란이라고 할 수 있다. 코깅은 배열된 영구자석에 대해 자성체인 철심의 위치가 자기적으로 안정된 상태를 찾아가려는 경향에 의해 나타난다. 이것은 모터 코일의 자석에 대한 상대적인 위치에만 관계가 있고 모터 전류와는 독립적이다. 추력리플은 코일에 전류를 흘렸을 때 코일에 발생하는 자기장이 영구자석과 결합하여 나타내는 힘으로서 무철심형에도 존재하는 힘이다. 추력 리플은 모터 전류가 0 이 아닌 때에만 발생하고 이것의 절대값은 요구되는 힘과 이송자의 상대적 위치에만 의존한다.

이러한 모터 리플은 이송자의 속도 변동을 일으켜서 추종성을 떨어뜨리므로 추적 오차를 발생시킨다 [1-3].

본 논문은 리니어 DC 모터(LDM)로 구성된 이송계에서 모터 리플에 의해 발생하는 추적오차를 피드포워드 제어와 신경망 제어에 의해 감소시켜 이송계의 추종성을 향상시키는 방법에 대해 논한다. 먼저 LDM 이 갖는 외란적 성격의 모터 리플의 특성을 파악하고 그 특성을 피드포워드 필터로 활용하여 피드포워드 제어를 실행한 결과를 제시하고 다음으로 신경망제어에 의하면 모터 리플의 측정 없이도 개선된 결과를 얻을 수 있음을 보인다. 피드포워드제어와 신경망제어 방식의 유용성을 검증하기 위하여 실험장치를 제작하고 실험결과를 제시한다.

2. 코깅 측정에 의한 피드포워드 제어

2.1 리니어 모터 이송계의 구성

추종성 향상을 위한 실험을 위해 Fig. 1 과 같은 리니어 DC 모터로 이송계를 설계 제작하였다. 전체적인 구성은 삼익 LMS 社가 중기거점사업으로 개발한 철심형 리니어모터 시제품 1 호기와 THK 社의 LM 가이드(HSR55R) 및 다사테크社의 모터 앰프 (IMS-J series)로 구성하였고 리니어스케일은 분해능 1 μm 인 Renishaw 社의 RGH-22 를 채택하였다. 실험대상인 리니어모터는 공작기계 이송계에 많이 사용되는 철심형 리니어 모터로 철심에 삼상의 권선이 감겨있고, 최대추력은 12,000 N, 연속추력은 6,000 N 이다.

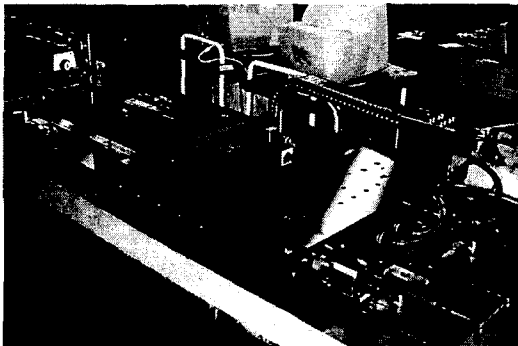


Fig. 1 Experimental setup for Linear DC motor feeding system

2.2 DSP 제어 구성

제어를 위해 사용한 DSP 보드는 TI 사의 TMS320C44 칩을 장착한 Innovative integration 사의 M44 보드이다. 이 칩은 60MHz 의 속도로 32bit 부

동소수점 연산을 수행할 수 있다. 모터 제어를 위해 모션 컨트롤 전용 옵션보드인 "MOT" 모듈을 장착하였으며, 이 모듈은 4 축으로 동시제어가 가능하다. DSP 보드에 다운로드되는 타겟 프로그램을 위해 CodeWriter 를 이용하였으며, 이송거리, 속도, 가속도, PID 게인, 앰프 ON/OFF 등을 설정할 수 있게 호스트 프로그램을 Visual C++로 작성하였다. 본 실험에서 엔코더 신호는 1 kHz 의 속도로 샘플링된다.

2.3 코깅 측정

리니어 모터의 코깅을 측정하기 위하여 Fig. 2 와 같이 로드셀과 레이저 간섭계를 이용하였다. 로드셀에 덧붙여 장착되어 있는 스크류를 이용하여 레이저 간섭계로 위치를 1 mm 씩 확인하면서 변화시키고 이때 로드셀에 가해진 힘을 측정하였다.

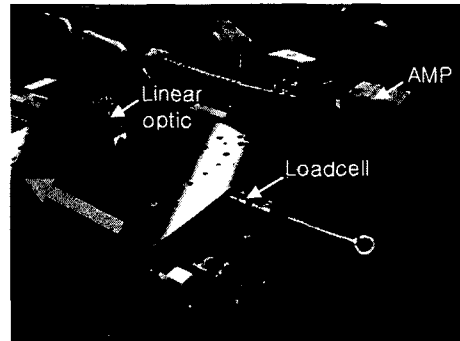


Fig. 2 Experimental setup for cogging measurement in linear motor feeding system

측정된 코깅값과 이를 사인 함수로 곡선맞춤(fitting)한 결과를 Fig. 3 에 나타내었다. 한편, 자석은 모두 동일한 것으로 가정하여 이송의 전구간이 아닌 자석의 한 피치 구간에서만 측정하였다. Fig. 3 에서 보면 코깅의 주기는 10 mm 로 나타났으며, 이는 설치된 자석간의 피치인 30 mm(Fig. 4)의 1/3 로써 각 상(phase)이 한 번 바뀔 때마다 코깅이 발생하는 것으로 보인다. 위치에 따른 코깅값을 사인 함수로 나타내면 식 (1)로서 표현할 수 있다.

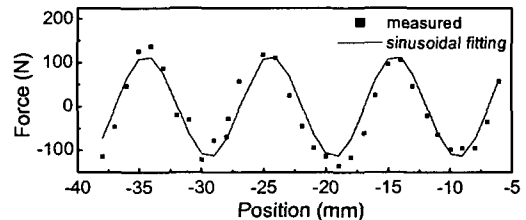


Fig. 3 Cogging force measured from setup of Fig. 1

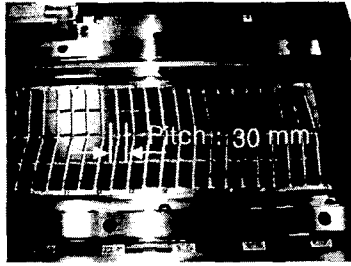


Fig. 4 Photograph of magnet placement

$$F = 116.6 \sin\left\{\pi \frac{x - 3.17891}{4.99615}\right\} N \quad (1)$$

측정된 코깅값을 이용하여 제어를 하기 위해서는 전압 대 추력 곡선을 알 필요가 있다. 즉, 특정 위치에서 이 코깅만큼 원하지 않는 힘이 발생하는 것이므로 이 코깅에 해당하는 전압을 구해 역으로 힘이 발생하도록 제어를 해 주어야 한다. Fig. 5 는 측정된 전압-추력 곡선을 보여주고 있다. 4V 이상에서는 포화현상이 발생하여 전압을 올려도 더 이상 추력이 증가하지 않으나, 일반적으로 사용하는 영역인 4V 까지는 어느 정도 선형적인 관계를 보여주고 있다.

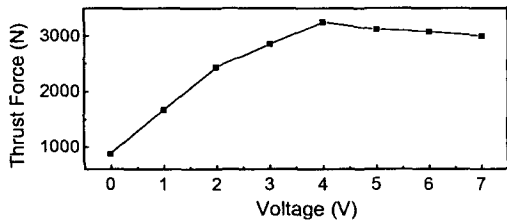


Fig. 5 Relation between voltage and thrust force

Fig. 5 의 관계를 이용하여 식 (1)을 전압값으로 환산하면 식 (2)를 얻을 수 있다.

$$F = 213.6 \sin\left\{\pi \frac{x - 3.17891}{4.99615}\right\} mV \quad (2)$$

Fig. 6 는 코깅을 제거하기 위한 제어도를 나타낸다. 일반적인 PID 제어에 추가로 코깅을 제어하기 위한 피드포워드 제어가 추가되었다.

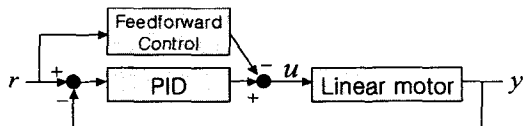


Fig. 6 Control method of feedforward control

Fig. 7 은 피드포워드 제어 없이 PID 제어만 한

경우와 코깅을 이용하여 피드포워드 제어를 한 경우를 비교한 결과이다. PID 제어만으로는 150 μm 정도의 추종오차를 발생하였으나 코깅 제어를 통해 75 μm 정도로 추종오차를 줄여들어 50% 정도의 향상이 있었음을 확인할 수 있다.

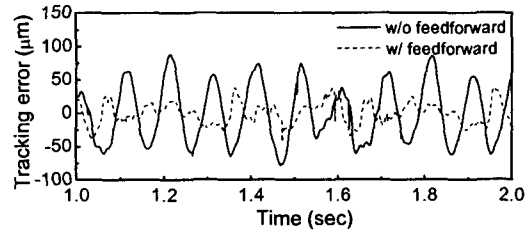


Fig. 7 Comparison of between feedforward and non-feedforward control

3. 신경망 PID 제어에 의한 향상

3.1 신경망 알고리즘

신경망 알고리즘은 학습을 통해 출력이 입력을 추종하도록 하는 것이며, 뛰어난 학습능력으로 인해 비선형 시스템의 제어에 매우 효과적이라고 알려져 있다. 학습은 목적함수를 설정하여 이 목적함수가 최소가 되도록 입력과 출력 사이의 가중치 값을 변화시키는 것에 의해 이루어진다.

신경망에서는 가중치값을 업데이트하기 위한 한 방법으로 역전파 알고리즘(Back propagation algorithm)을 사용한다. 역전파 알고리즘에서는 오차 신호가 시스템을 거쳐 신경망의 가중치값을 업데이트하므로 시스템의 입력과 출력간의 관계를 나타내는 시스템 자코비안 함수를 구해야만 한다. 그러나 시스템 자코비안을 평가하는 자체가 매우 어려우므로 여기서는 자코비안을 피하는 방법으로 목적함수를 설정한다. 이 방법은 Kawato 가 제시한 방법으로 목적함수를 시스템 출력과 입력의 차이로 하지 않고, 식 (2)와 같이 PID 제어기의 출력 전압으로써 목적함수를 설정하는 방법이다. 즉, Fig. 8 과 같은 제어도에서 오차신호를 $r-y$ 로 설정하지 않고 u_{pid} 로 설정하게 되면 자코비안을 구하지 않아도 되는 장점이 있다.

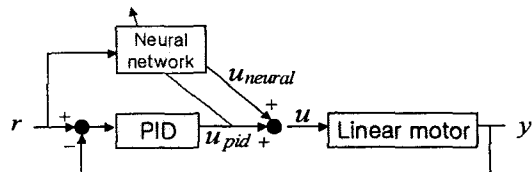


Fig. 8 Control method of neural network control

$$E = \frac{1}{2} u_{pid}^2 \quad (3)$$

Fig. 8의 제어도를 보면, 제어 초기에는 신경망이 학습되지 않았기 때문에 PID 제어기에 의해 전체 시스템이 제어되게 된다. 그러나 학습이 진행됨에 따라 목적함수로 하는 PID 제어기의 출력인 u_{pid} 의 영향이 작아지고 신경망 제어기의 출력인 u_{neural} 이 전 시스템을 지배하게 된다. 학습이 완료된 후에도 PID에 의한 피드백은 계속 이루어지므로 갑작스런 외란이 작용해도 시스템은 어느정도 안정성을 갖게 된다.

본 논문에서는 Fig. 9와 같은 신경망에서 입력을 원하는 위치(r), 출력을 신경망의 출력 전압값(u_{neural})으로 설정하고 은닉층으로는 세 개의 뉴론을 두었다.

각 층에서의 가중치 값을 업데이트하는 방법은 다음과 같다.

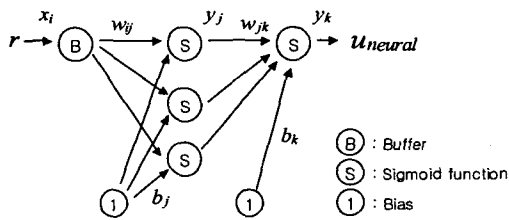


Fig. 9 Neural network

입력층의 입력 x_i 와 입력층과 은닉층 사이의 가중치값 w_{ij} 가 곱해져서 은닉층의 출력 s_j 가 된다. s_j 는 대표적인 비선형 함수인 시그모이드 함수를 통과하여 y_j 가 된다. 여기서 입력이 항상 1인 바이어스 가중치인 b_j 도 고려해야 한다.

$$s_j = \sum_{i=1}^{N_i} w_{ij} x_i + b_j \quad (4)$$

$$y_j = f(s_j) = \frac{1}{1 + e^{-s_j}}$$

같은 방법으로 은닉층에서 출력층으로의 출력 y_k 도 구할 수 있다.

$$s_k = \sum_{j=1}^{N_H} w_{jk} y_j + b_k \quad (5)$$

$$y_k = f(s_k) = \frac{1}{1 + e^{-s_k}}$$

$$= \frac{1}{1 + e^{-\sum_{j=1}^{N_H} w_{jk} [\frac{1}{1 + e^{-\sum_{i=1}^{N_I} w_{ij} x_i + b_j]}] + b_k}}$$

한편, 신경망의 출력전압(u_{neural})은 y_k 에 게인을 곱함으로써 구할 수 있다.

각 가중치의 변화를 나타내면 식 (6)과 같다.

$$\begin{aligned} \Delta w_{jk} &= \eta \delta_k y_j \\ \Delta b_j &= \beta \delta_j \\ \Delta w_{ij} &= \eta \delta_j x_i \\ \Delta b_k &= \beta \delta_k \end{aligned} \quad (6)$$

여기서 η 는 학습률을 나타내며, β 는 바이어스에 대한 학습률을 나타낸다. 또한 δ_k 와 δ_j 는 식 (7)과 같이 정의된다.

$$\begin{aligned} \delta_k &= e_k f'(s_k) \\ \delta_j &= \sum_{k=1}^{N_o} \delta_k w_{jk} f'(s_j) \end{aligned} \quad (7)$$

최종적으로 각 가중치는 다음식에 의해 업데이트된다.

$$\begin{aligned} w_{jk}(t+1) &= w_{jk}(t) + \Delta w_{jk}(t) + \alpha(\Delta w_{jk}(t) - \Delta w_{jk}(t-1)) \\ b_j(t+1) &= b_j(t) + \Delta b_j(t) + \alpha(\Delta b_j(t) - \Delta b_j(t-1)) \\ w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij}(t) + \alpha(\Delta w_{ij}(t) - \Delta w_{ij}(t-1)) \\ b_k(t+1) &= b_k(t) + \Delta b_k(t) + \alpha(\Delta b_k(t) - \Delta b_k(t-1)) \end{aligned} \quad (8)$$

여기서 α 는 모멘텀항으로써 이전 가중치 변화율과 현 가중치 변화율의 비율을 나타내는 것으로 학습속도를 빠르게 하고, 지역최소점에 빠지는 것을 어느 정도 방지할 수 있게 한다고 알려져 있다.

3.2 가중치 적용 방법

기본적으로 가중치를 적용하는 방법은 식 (8)과 같다. 그러나 이는 패턴인식문제와 같이 여러 입력 패턴 중에서 입력되는 패턴의 유형을 찾는 것과 같이 모든 패턴에 대해 동일한 가중치를 부여하는 것이다. 그러나 위치제어에서는 그 입력되는 패턴(위치정보)이 주어지는 것과 마찬가지로 모든 패턴에 대해 공통적인 가중치를 부여할 필요가 없다. 즉, 각 위치마다 각각의 가중치를 설정하여 다른 위치에서의 가중치에 관계없이 오직 그 위치에서만 최적의 가중치를 구하면 되는 것이다. 단, 이 경우는 각 가중치를 저장하기 위한 메모리가 많이 필요하다는 단점이 존재한다.

3.3 실험결과

Fig. 10은 동일한 가중치를 부여하여 학습시킨 결과를 보여주고 있다. 여기서 n 의 값은 학습시킨 회수를 나타낸다. 동일한 가중치를 부여시 학습이 15번이나 진행됐지만 추종오차는 학습전과 별 차이가 없는 것으로 나타났다. 부분부분 더 좋아진 곳도 있고, 오히려 더 나빠진 곳도 존재한다. 이것은 가중치가 모든 곳에서 적용되다 보니 특정위치에서는 잘 맞고, 다른 위치에서는 잘 맞지 않는 것

으로 생각된다. 결과적으로는 학습이 아직 덜 됐음을 의미한다.

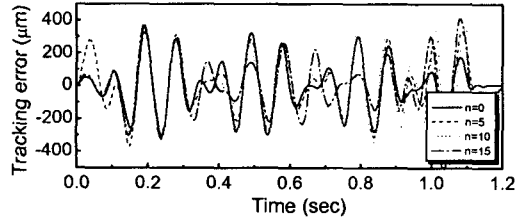


Fig. 10 Learning result in case of applying identical weight factor

Fig. 11 은 각 위치마다 각각의 가중치를 부여했을 경우로 학습이 진행됨에 따라 추종오차가 매우 작아지는 것을 알 수 있다. 15 번 학습결과 학습 전보다 10 배 이상 좋아져 60 μm 정도의 추종오차를 보이고 있다.

Fig. 12 와 13 는 신경망 PID 제어의 특성을 알아보기 위해 학습 전 후의 PID 제어기의 출력 전압인 u_{pid} 와 신경망 제어기의 출력전압인 u_{neural} 을 비교한 것이다. Fig. 12 에서 u_{pid} 는 학습 전에는 코깁에 의해 사인 형태로 존재했지만 학습이 진행됨에 따라 u_{pid} 는 목적함수로써 작아지게 되어 거의 0 에 가까워지는 것을 알 수 있다. 한편, Fig. 13 에서 u_{neural} 은 학습 전에는 초기값으로써 특정형태를 띄지 않고 있으나 학습이 진행된 후에는 처음에 u_{pid} 에 나타났던 형태가 그대로 나타나는 것을 알 수 있다. 단, P-V 값은 u_{pid} 에 비해 작아진 것을 알 수 있다.

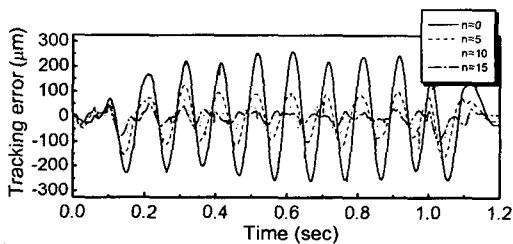


Fig. 11 Learning result in case of applying individual weight factor

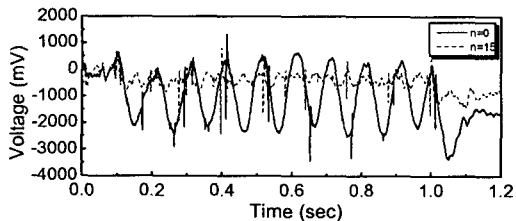


Fig. 12 Comparison of PID control voltage w/o and w/ learning

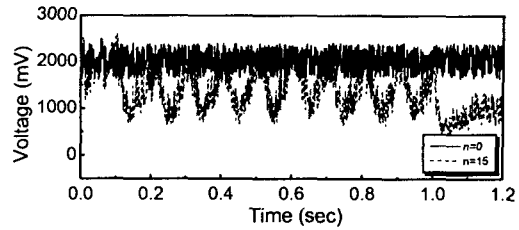


Fig. 13 Comparison of Neural network voltage w/o and w/ learning

PID 제어기에서 나오는 전압은 시스템이 약간이라도 불안정할 때 그 변동이 매우 심하다. 따라서 PID 제어기 출력 전압을 목적함수로 택하는 것이 약간의 무리가 따를 수도 있다. 목적함수를 원하는 출력과 실제 출력의 차로써 정의하고, 시스템의 특성을 나타내는 자코비안 함수를 실험적으로 구한다면 변동성을 줄여 더 나은 결과를 얻을 수 있을 것이므로 추후 연구과제로 추진할 예정이다.

4. 결론

철심형 리니어모터에 고유하게 내재하는 모터리플은 추종성 오차를 불러온다. 추종성 오차를 줄이기 위하여 코깁 측정에 의한 피드포워드 제어와 신경망 제어를 적용하였고 다음과 같은 결론을 얻었다.

- (1) 피드포워드 제어로 추적 오차를 50% 감소시킬 수 있었다.
- (2) 신경망 제어에 의하면 코깁의 측정이 불필요하며 피드포워드보다 향상된 결과를 15 번 학습에 의해 얻을 수 있었다.

참고문헌

1. P. van Den Braembussche, J. Swevers, H. van Brussel and P. Vanherck, "Accurate tracking control of linear synchronous motor machine tool axes," *Mechatronics*, Vol. 6, No. 5, pp. 507-521, 1996.
2. Gerco Otten, Theo J. A. de Vries, Job van Amerongen, Adrian M. Rankers and Erik Gaal, "Linear Motor Motion Control Using a Learning Feedforward Controller," *IEEE/ASME Transaction on Mechatronics*, Vol. 3, No. 3, pp. 179-187, 1997.
3. G. Pritschow and W. Philipp, "Research on the Efficiency of Feedforward Controllers in Direct Drives", *Annals of the CIRP*, Vol. 41, pp. 411-415, 1992.