

모션 제어장치의 네트워크 보간 운전방법

Operating Method of Network Interpolation for Motion Control Device

곽 군 평

(Gun-Pyong Kwak)

Abstract : Motion controllers are essential components for operating industrial equipments. Compared with general industrial controllers, motion controllers allow motion control requiring greater speed and precision. This paper presents a method for controlling multi-axes motors via industrial networks. To achieve a line or arc interpolation, the master system delivers instructions to slave systems connected to the network. The network instruction transmitted from the master controller is re-interpolated by the individual slaves through sub-interpolators. The re-interpolated feedrate information is transmitted to the motion control loop in which the current position and the reference position are then calculated. In this way, the interpolation driving between control units is achieved via industrial networks.

Keywords : motion controller, network interpolation, interpreter, CAN, multi-axes, master, network

I. 서론

모션 제어기는 기계에 장착되어 있는 모터를 제어하여 기계의 움직임이 일정 궤적 및 속도를 가지도록 하는 장치로써 각종 전용기계, 공작기계, 반도체 장비등과 같은 기계 장비에 설치되어 기계의 동작을 제어하는 장비이다. 기존의 모션 제어기의 운영방법은 하나의 모션 제어기에서 모든 보간을 실행하여 actuator(motor drive)에 아날로그 또는 펄스로 이송값을 지령하는 일체형 제어방식으로 되어 있으나[1][2], 본 논문에서 제안되는 기법은 기존의 일체형 보간방식을 네트워크를 통하여 여러 모션제어기들을 분산 제어하는 방식으로, 네트워크를 통하여 운영되는 모션제어기들이 하나의 통합된 제어기가 보간 운전을 수행하는 것처럼 네트워크를 통하여 전체 제어기간동안 보간 운전을 수행하는 것이다. 본 논문에서 제시한 네트워크 보간용 통신 모듈은 CAN 버스를 사용하였다[3]. 모션제어기의 일종인 CAN(Control Area Network)통신기능을 갖춘 제어장치는 기존의 FA시장에서 판매되는 모션제어장치와는 통신 방법이 근본적으로 상이하다. 이것은 기존의 단순한 시리얼 통신 방법으로 각종 데이터를 주고 받는 기술을 넘어 표준 Field Bus를 이용한 산업용 통신기능을 내장한 단축 및 다축 제어장치이다. 특히 산업용 제어장치의 설계기술 인프라가 취약한 기계산업 육성을 위해 저가격의 경쟁력을 갖춘 고기능의 모터 구동용 제어장치의 기술 개발은 관련 기계산업의 대외 가격 경쟁력을 제고시킬 것으로 사료된다.

본 논문에서 제시된 네트워크 보간 운전방법이 가능한 CAN 제어장치는 소형, 저가를 목적으로 하는 제어기로서 기본적으로 보간이 필요치 않는 1축용 보조 장비에 장착되는 목적으로 개발된 제어장치이다[4]. 이 제어기는 서보 및

스텝핑 모터를 구동하기 위한 상위제어기로서 펄스를 출력하는 타입과 아날로그 전압을 출력하는 타입으로 구분된다. 또한 저가형 제품이면서 고급 제어기만이 보유하고 있는 고속 산업용 통신기능이 기본적으로 내장되어 있어 상위제어기와 인터페이스를 통하여 다축 제어 및 다축 보간 그리고 동기운전등의 기능이 구현됨으로써 일반 고급제어기와 같은 성능을 보유하게 될 것이다. 아울러 본 제어장치는 산업용 네트워크(CAN)기능을 내장한 I/O기기를 부착하여 축제어기와 기기의 동작을 연결시켜 하나의 시스템 통합 제어기의 구현도 가능하다. 산업용 네트워크(CAN)는 64개의 노드를 가질 수 있어 총 64개의 장비(모터)를 연결할 수 있다. 따라서 이 기능을 이용하면 최대 64개 제어기를 1Mbps의 통신속도로 분산 제어 및 감시가 가능하다.

II. 일체형 모션제어기의 보간기술

종래의 모션 제어기술은 하나의 모션 제어기가 보간에 관련된 모든 축들의 연산을 수행하고, 모터를 구동하기 위한 이송값을 직접 출력하는 방식으로서, 기술적 특징은 크게 하드웨어와 소프트웨어로 분류할 수 있다. 종래 모션제어기의 하드웨어적 기술특징은 그림 1과 같다. 여기서 서보는 전압 입력형으로 가정하여 설명한다. 3축 보간기능을 지원하는 모션제어기는 그림 1에서 보는 것처럼 3개의 포트에 각각의 서보 드라이버가 병렬 케이블을 통하여 연결되고 이 케이블을 통하여 속도 지령신호(아날로그 신호)를 출력하고, 모터의 엔코더 신호를 입력 받는다. 그림 1의 모션 제어기는 3축을 제어하기 위하여 엔코더를 통하여 각축의 현재 위치를 입력 받아 축 이송에 관계되는 연산을 수행하고 그 값을 아날로그 출력값으로 전환하여 각각의 드라이버에 출력한다. 이러한 동작을 일정 시간동안 주기적으로 반복 수행한다. 모션 제어기의 일반적인 동작은 그림 2와 같다.

그림 1의 모션제어기는 그림 2와 같은 동작을 수행한다. 사용자가 기계의 동작에 관련되는 이송궤적(직선, 원호, 급송이송등) 및 이송속도가 들어있는 이송 프로그램을 작성하

논문접수 : 2002. 3. 6., 채택확정 : 2002. 6. 21.

곽군평 : 창원대학교 전기공학과(kwak@sarim.changwon.ac.kr)

* 본 연구 학기술부 · 한국과학재단 지정 창원대학교 공작기계기술연구센터의 지원에 의한 것입니다.

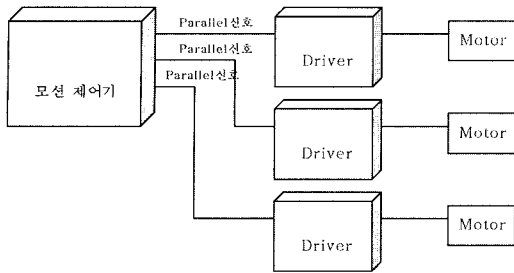


그림 1. 종래 모션제어기의 시스템 구성도.
Fig. 1. System architecture of conventional motion controller.

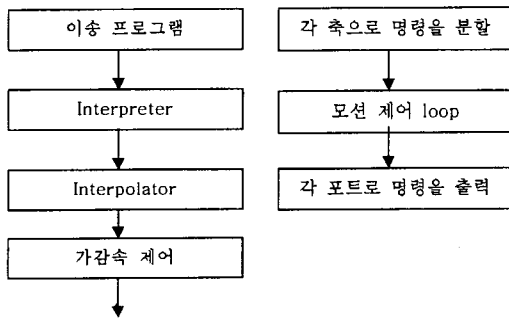


그림 2. 모션제어기의 동작순서.
Fig. 2. Operating sequence of motion controller.

여 모션 제어기에 이송을 시작하라는 지령을 하면, 해석기(interpreter)는 이송 프로그램을 해석하여 보간기(interpolator)가 이해할 수 있는 명령 코드로 전환하여 보간기에 전달한다. 보간기는 이송지령을 일정 주기의 시간에 관계되는 속도 함수로 해석하고 이 속도 함수에 가감속 제어를 추가하여 3축의 합성 속도 프로파일을 만든 후 합성 속도 프로파일을 각 축으로 분할하여 이송을 지령한다. 즉 보간기에서 3축의 움직임을 합성하여 만든 속도 프로파일을 각 축의 속도 프로파일로 분할하여 지령한다는 의미이다. 그림 2의 모션 제어 루프는 서보 드라이버의 엔코더로부터 받은 현재 위치 값과 보간기에서 분할하여 받은 이송위치의 차를 계산하여 각각의 서보 드라이버에 이송값을 지령한다. 종래의 보간방법은 그림 2의 동작을 하나의 모션제어기에서 수행하고 그 시스템 구성은 그림 1과 같은 구조로 되어있다. 즉 모든 모션 관련 동작을 하나의 제어기에서 계산하고 관리한다. 그림 2의 이송 프로그램은 사용자가 기계의 이송궤적과 이송속도를 지령하는 명령어로서 예를 들면 다음과 같이 작성할 수 있다.

```

LINEAR X100.0 Y50.0 F1000
CCW X0.0 Y0.0 I-100.0 J-100.0 F200
.....
    
```

첫번째 명령은 X, Y축이 현재 위치에서 X100.0 Y50.0의 위치로 속도 1000을 가지고 직선 보간을 수행하라는 지령이고, 두번째 명령은 X, Y축이 반시계 방향으로 속도 200으로 원호 보간을 수행하라는 지령으로 가정한다. 위의 프로그램을 수행하라는 지령을 모션 제어기에 내리면, 그림 2의 해석

기는 이송 프로그램을 순차적으로 읽어 들이고 첫번째 명령부터 순차적으로 해석하여 이송 위치에 해당하는 위치 합성 벡터를 그림 3과 같이 발생한다.

해석기에서 만든 위치 합성벡터에는 속도 개념이 포함되어 있지 않다. 이렇게 만든 위치 합성벡터를 보간기에서 속도의 개념을 포함하여 일정 시간동안 이송할 위치 합성벡터로 분할한다. 즉 모션제어기의 모션제어 루프는 일정 주기의 샘플링 시간으로 모션 제어를 실행하고 있으므로 이 샘플링 시간동안 이송해야할 위치 합성벡터로 분할하여 계산한다. 이와 같은 이송시간의 개념을 가지고 생성된 프로파일을 그림 4의 왼쪽에 나타내었다. 보간기에서 그림 4의 왼쪽과 같이 생성된 이송 프로파일에 가감속 제어를 적용하면 보간후 가감속인 경우 그림 4의 오른쪽과 같은 사다리꼴 이송 프로파일이 생성된다. 그림 4의 오른쪽과 같이 생성된 사다리꼴 이송 프로파일을 보간기에서 샘플링 시간을 기준으로 하는 이송거리로 분배한 다음 이 값을 각 축의 샘플링당 이송거리로 계산하고 이 결과 값을 그림 2와 같이 모션제어 루프에 전달한다.

그림 5는 모션제어 루프의 동작을 나타낸 것이다. 보간기에서 보간 및 가감속을 적용한 샘플링 시간당 각 축의 이송 위치 지령을 그림 5의 모션 제어 루프에 전달한다. 모션제어 루프는 보간기에서 지령된 위치 값과 서보 구동기의 엔코더로부터 피드백된 위치 값과의 오차를 계산하여 그 결과를 속

- S : 위치 합성벡터
- x : 이동해야될 X축 거리
- y : 이동해야될 Y축 거리

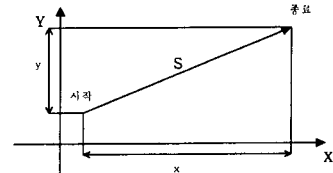


그림 3. 합성위치 벡터.
Fig. 3. Composition of position vector.

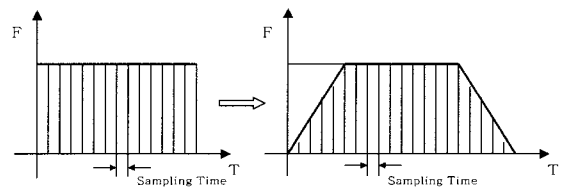


그림 4. 속도와 시간의 이송 프로파일.
Fig. 4. Feedrate profiling of velocity versus time.

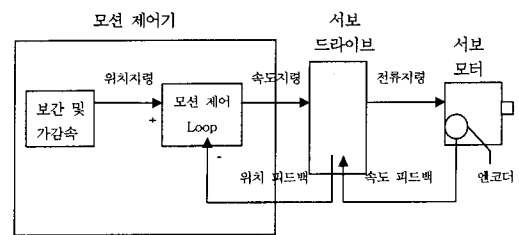


그림 5. 종래의 모션 제어 루프.
Fig. 5. Looping for conventional motion control.

도지령으로 변환한다. 이 속도 지령값을 그림 5와 같이 출력 포트를 통하여 서보 구동기로 출력한다. 기존의 모션제어기는 이상에서 설명한 방법으로 모터를 제어하고 또한 모든 제어기능(해석기, 보간기, 모션제어 루프등)이 하나의 모션제어기에 구성되어 있다. 종래의 통합된 모션제어기는 고정된 축 제어를 수행할 수 밖에 없다. 즉 단축 제어기는 1축, 2축 제어기는 2축 이하, 3축 제어기는 3축 이하의 축만을 제어하게 구성되어 있어, 제어축을 확장할 때는 모션제어기를 교체해야되는 단점이 있다. 이송축을 축소한다면 제어기의 기능 축소 현상이 발생되고 또한 두개 이상의 모션제어기가 하나의 네트워크상에 존재할 때 독립된 각각의 보간기가 시스템에 존재하여 다른 제어기들간의 축간 보간 운전이 불가능하여 제어기1의 X축과 제어기2의 Y축간의 보간 운전을 구현할 수 없다는 단점이 있다. 또한 두개의 모션 제어기가 기계상에서 하나의 축을 공유하는 기능이 필요할 때 기존의 3축 제어기로는 적용이 불가능하다. 이 기능을 구현하기 위해서는 5축 이상의 채널기능을 보유하고 있는 고가의 고급 제어기로만 구현이 가능하다. 그러므로 채널 개념을 가지고 있지 않은 3축 제어기에서는 각각의 보간기가 존재하여 하나의 축을 두 제어기가 공유한다면, 한 축에 두개의 모션제어 루프가 형성되어 모션제어 구성이 파괴되어 축제어가 되지 않는다. 따라서 하나의 축을 두개의 제어기가 공유하는 시스템 구성이 불가능하다.

III. 제안된 기법에 의한 네트워크 보간

1. 네트워크 시스템 구성

본 기술은 분리된 모션제어기간의 보간 운전 및 제어기간의 축 공유등의 문제점을 해소하기 위하여 모션제어기의 모션제어 루프를 네트워크를 통하여 구현 함으로써 모션제어기의 유연성을 창출하고, 네트워크를 통하여 분리된 제어기간의 보간을 구현하는 데 그 목적이 있다. 제안된 기술은 네트워크를 통하여 보간기능을 수행함으로써 제한된 축의 기능을 해소하여 제어축의 확장성을 구현할 수 있다. 최근에는 이와 같이 모션제어기에 유연성 및 네트워크의 계층적 구조를 추가하여 활성화되고 있는 산업용 필드버스(profibus, device-net, ethernet등)에로의 적용을 가능하게 한다[5]. 본 기법은 네트워크를 통하여 보간을 구현하는 방법으로써 시스템 구성은 그림 6과 같다.

신기술에 대한 설명은 3축 보간을 지원하는 네트워크 보간 시스템과 서보 모터를 예로 한다. 그림 6의 왼쪽에 있는

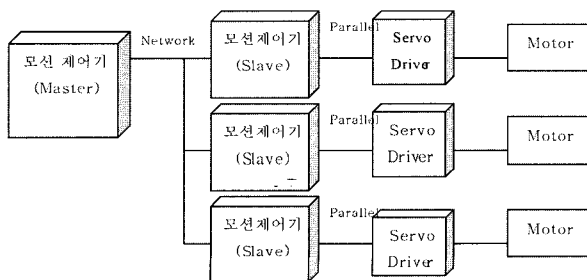


그림 6. 네트워크 보간의 시스템 구성도.
Fig. 6. System configuration of network interpolation.

모션제어기(master)는 네트워크에 연결되어 있는 모든 장비를 제어하기 위한 주 제어장치이다. 이 장치는 네트워크에 연결되어 있는 모든 모션제어기(slave)들에 대한 지령을 전송하고, slave들의 상태를 감시한다. 그림 6에서 네트워크로 표시되어 있는 통신방법은 특정 통신방법이 아니고 현재 사용되고 있는 모든 serial 통신 또는 향후에 적용될 무선 통신으로 구현이 가능하다. 적용 가능한 통신방법은 profibus, device-net, CAN등의 fieldbus와 일반적으로 사용되고 있는 ethernet으로 구현할 수 있다[5][6]. 그림 6의 slave는 master로부터 명령을 받아 기계의 구동 시스템(서보 모터, 스텝핑 모터등)을 제어하는 모션 제어기로서, 여기서는 이 모션 제어기를 하나의 축을 제어할 수 있는 단축 모션제어기로 가정한다. 아울러 slave 모듈은 1축 제어기 이상의 모션제어기로 구성이 가능하다. 그림 6의 master와 slave간의 동작은 다음과 같다.

그림 7은 그림 6의 모션제어기(master)에 대한 동작을 나타낸 것이다. 그림 7의 이송 프로그램은 시스템이 장착되어 있는 기계의 이동궤적과 이송속도를 정의한 사용자 프로그램이다. Master를 구동하라는 명령을 내리면, 그림 7의 해석기가 이송 프로그램을 해석하여 합성위치 벡터를 생성하고 그 결과와 합성 이송속도를 그림 7의 보간기로 전송한다. 합성위치 벡터와 이송속도를 받은 보간기는 속도와 시간에 대한 이송 프로파일(그림 4의 왼쪽)을 만든다. 여기서 샘플링 시간은 기존의 샘플링 시간이 아닌 네트워크 샘플링 시간이다. 보간기에서 만든 합성 이송 프로파일에 가감속 제어를 적용하여 사다리꼴 이송 프로파일(그림 4의 오른쪽)을 만들고 이 프로파일을 네트워크 샘플링 시간으로 분할하여 이 분할된 합성 이송거리를 slave에서 이송하기 위한 각 축의 이송거리로 계산한다. 이 각축의 이송거리를 네트워크를 통하여 해당 slave에 전송한다.

그림 8은 그림 6의 모션제어기(slave)의 동작을 나타낸 것이다. Master로부터 전송되는 네트워크 샘플링 시간동안의 이송지령을 그림 8의 네트워크 명령(network command)으로 전송받는다. Slave는 상위로부터 전송받은 네트워크 명령을 보조 보간기(sub interpolator)를 통하여 slave 샘플링 시간동안 이송할 지령으로 다시 보간을 한다. 이렇게 재 보간된 이송지령을 모션제어 루프로 전송하고 모션제어 루프에서는 현재 위치와 지령된 위치에 대한 계산을 수행한다. 계산된 결과는 서보 구동기에 출력할 신호(아날로그 또는 펄스)로 전환되어 포트로 출력한다. 제안된 기술은 종래의 모션제어기의 보간부분을 네트워크를 통하여 master의 주 보간(main

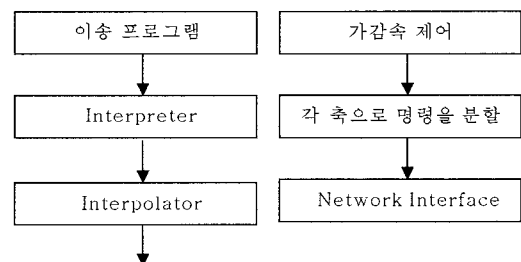


그림 7. 모션제어기(master)의 동작.
Fig. 7. Operation of motion controller(master).

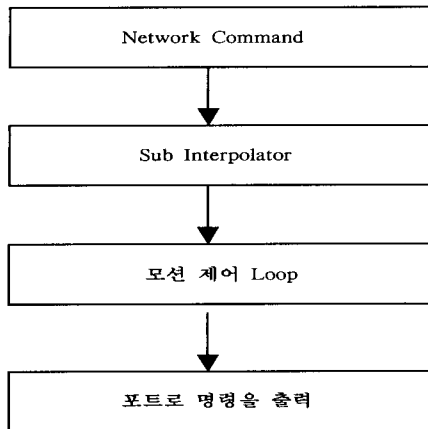


그림 8. 모션제어기(slave)의 동작
Fig. 8 Operation of motion controller (slave)

interpolation)과 slave의 보조 보간(sub interpolation)으로 분리하는 방법과 네트워크상에서 보간을 수행하는 시스템간의 동작을 동기화 시키는 방법에 주안점을 두고 있다. 먼저 네트워크를 구성하는 방법과 네트워크 보간을 구현하기 위한 기본 파라미터에 대하여 설명한다. 그림 6과 같이 하나의 네트워크상에 설치되어 있는 모션제어기들을 구분하기 위하여 각각의 제어기에 네트워크상의 이름을 부여해야 한다.

일반적인 방법은 각각의 모션제어기들에 독립된 네트워크 어드레스를 설정하는 것이다. 만약 적용되는 네트워크가 네트워크 어드레스 설정이외의 다른 방식으로 각 노드들을 구분한다면 네트워크의 사양에 의거하여 각 노드를 네트워크상에서 구분할 수 있게끔 설정하면 된다. 여기서는 네트워크 어드레스 설정방식으로 각 노드들을 구분하여 설명한다. 각각의 모션 제어기들의 네트워크 어드레스를 설정한 다음 어떤 모션제어기가 master 또는 slave인지를 설정한다. 그리고 master에 제어대상 slave들에 대한 내부 이름(X, Y, Z축등)을 설정한 후, 네트워크 보간에 참여하고 있는 모든 모션제어기에 네트워크 샘플링 시간을 설정한다. 그림 9는 네트워크 구성을 설정하는 방법을 나타내고 있다. 먼저 모든 시스템의 네트워크를 하드웨어적으로 구성하고 네트워크 설치가 완료되면 네트워크상에 연결되어 있는 모든 장비의 노드 어드레스와 특성을 설정한다. 그리고 네트워크 샘플링 시간을 설정한다. nst란 master의 주 보간기가 이송 프로파일을 설정할 때 기준이 되는 시간 단위이고 또한 네트워크를 통해 slave에 명령을 지령하는 기준 시간이다. nst는 slave 샘플링 시간(sst)의 2n배 이상이 되어야 한다.

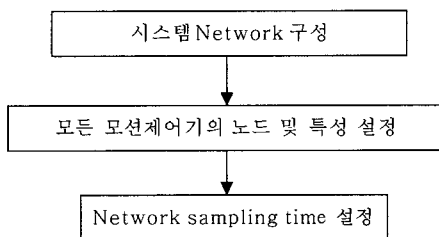


그림 9. 네트워크 구성 방법
Fig. 9 Network configuration method

$$nst = 2 \times n \times sst \tag{1}$$

nst : 네트워크 샘플링시간(모션제어기중 master의 보간 샘플링시간)

sst : 샘플링시간(모션제어기중 slave의 보간 샘플링시간)

n : 정수 (1 이상의 값)

또한 적용되는 네트워크의 특성을 만족하기 위하여 아래의 네트워크 특성도 만족해야 한다.

$$nst >= fbt \times fcount + ((fcount - 1) \times st) + rcv \tag{2}$$

fbt : 가장 긴 전송 프레임의 통신 시간

fcount : nst시간 안에 통신할 통신 프레임의 갯수

st : 연속 데이터 통신시 프레임간의 sleep time

rcv : 에러 발생시 복구시간 (fcount x rcv_one)

rcv_one : 하나의 에러 프레임에 대한 복구 시간 및 재 전송 시간

sst는 slave의 하드웨어 구조에 의해 고정되는 값으로써 nst를 결정하기 위해서는 네트워크 보간에 참여하고 있는 slave들중의 가장 큰 sst를 기준으로 하여 nst를 계산하면 안정된 nst값을 얻을 수 있다. (1)에 의해 설정된 nst값을 네트워크에 연결되어있는 모든 장비(master, slave)에 설정한다. nst값은 통신 매체와 통신 속도에 따라 그 값이 결정되므로 (1)에 의해 결정된 값이라 해도 (2)를 만족하지 않으면 nst동안에 안정된 네트워크 보간을 수행할 수 없다. 그러므로 nst를 설정할 때는 (1)과 (2)를 동시에 만족하도록 설정하여야 한다. 모션제어기중 master의 동작은 다음과 같다. 그림 7의 이송 프로그램을 해석한 후 그림 3과 같은 합성위치 벡터를 생성하여 그 결과와 이송속도를 보간기로 전송한다. 보간기는 nst를 기준 단위로 하여 그림 10의 왼쪽과 같은 이송 프로파일을 생성한다. 이 프로파일에 가감속 제어를 적용하여 그림 10의 오른쪽과 같은 합성 프로파일을 생성하고 nst에 대한 이송 거리로 프로파일을 분할한다. 이 값을 다시 각 축의 이송거리로 분할한 후 하나의 nst 시간안에 전체 slave에 이송할 거리를 전송하고, 전체 slave가 동시에 알 수 있는 글로벌 네트워크 명령(global network command)을 전송한다.

글로벌 네트워크 명령은 네트워크 보간에 참여하고 있는 전체 노드의 동기를 맞추는 신호이다. 모션제어기중 slave의 동작은 그림 8에 나타내었다. slave가 네트워크를 통하여 nst동안 이송할 거리를 master로부터 지령 받은 다음 글로벌 네트워크 명령을 받으면 이 신호와 동기되어 보조 보간기에서 보간을 수행한다.

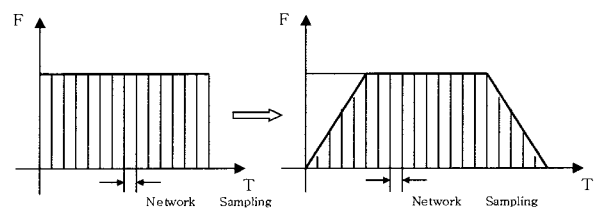


그림 10. 모션제어기(master)의 이송 프로 파일.
Fig. 10. Feedrate profile of motion controller(master).

slave의 보조 보간기동작은 그림 11과 같다. 여기서 T축은 시간축이고, P축은 위치축이다. 전체 nst동안 받은 위치값을 기준으로 하여 시간에 대한 위치 벡터(S)를 생성하고 이 위치 벡터를 기준으로 하여 sst의 위치 벡터(s)를 생성한다. 이송속도는 nst와 tp에 의해 계산되어 진다. 보조 보간기는 시간에 대한 위치 벡터(S)를 가지고 전에 생성된 이송 프로파일을 참조하여 새로운 이송 프로파일을 그림 12와 같이 연속으로 생성한다.

그림 12의 이송 프로파일에서 sst기간동안의 이송거리를 그림 8과 같이 모션제어 루프로 전달하면 그림 5와 같은 동작으로 모션제어 루프가 동작하여 그 결과 값을 출력 포트를 통하여 출력한다.

2. Master와 slave간의 통신 방법

그림 13은 모션제어기의 master와 slave간의 통신 흐름을 보여주고 있다. Master는 네트워크상의 slave의 보간 주기를 일정하게 유지하기 위해서 일정 주기(nst)로 전체 동기 명령(글로벌 네트워크 명령)을 내린다. 이때 이 동기 명령은 네트워크 보간에 관련되는 모든 slave들이 동시에 수신할 수 있는 명령이어야 한다. Slave들은 동기 신호를 받은 다음 일정 시간 경과후 일제히 동기 신호전에 들어온 위치 명령에 대하여 보조 보간을 수행하고 그 결과를 실행한다. 또한 slave들은 동기 신호를 받은 시점에서 이송축의 현재 위치를 저장하

- S : nst동안 이송벡터
- s : sst동안 이송벡터
- tp : nst동안 이송거리
- sp : sst동안 이송거리
- nst : 네트워크 보간시간
- sst : slave 보간시간

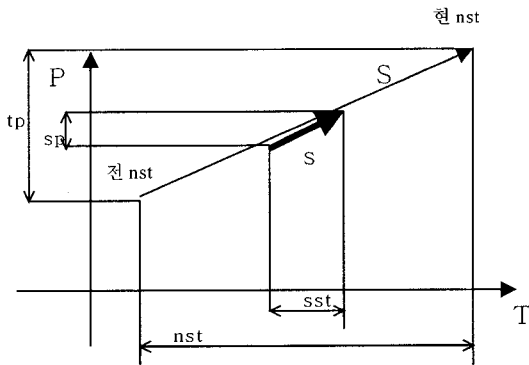


그림 11. Slave의 합성 벡터.
Fig. 11. Composition vector of slave.

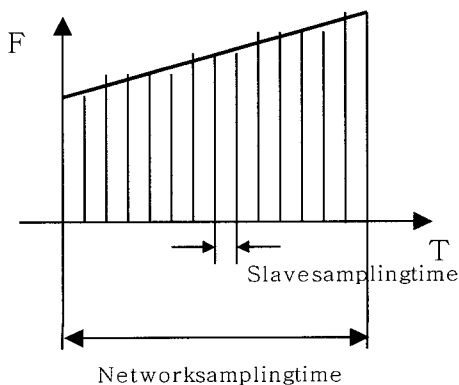


그림 12. Slave의 이송 프로파일.
Fig. 12. Feedrate profile of slave.

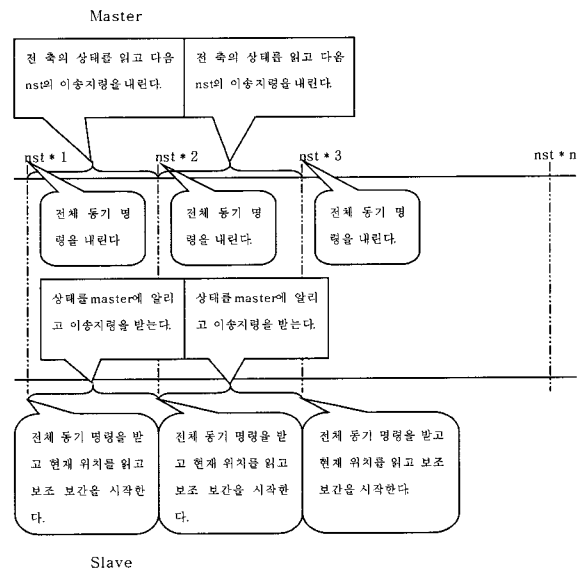


그림 13 네트워크 보간의 동작.
Fig. 13 Operation of network interpolation.

고 master에서 위치 정보 요청이 들어오면 동기 신호시 저장한 위치를 master에 송부하여 master에서 네트워크 보간에 대한 관리를 수행할 수 있게 한다. 그리고 master는 다음 동기명령을 지령하기 전에 다음 구간에서 보조 보간 할 위치를 slave에게 전송 완료하여야 한다. 이러한 동작을 일정주기(nst)로 반복한다. 네트워크를 통하여 동기 신호를 지령하다 보면 통신오류에 의해 정확한 주기로 명령을 지령할 수 없게 될 수도 있다. 이 오류를 보정하기 위해 정의되는 시간을 네트워크 보간 갭시간(network interpolation gap time : ngi)이라 한다. ngi는 slave에만 적용되는 시간으로써 그림 14에 나타내었다. 그림 14와 같이 직전에 발생된 동기신호(nst*-1)와 현재 발생될 동기신호(nst)에 대하여 ngi의 범위 안에 동기 신호가 slave에 입력되어야 한다. 여기서 ngi 값은 모든 모션 제어기의 slave에 일정한 값으로 인가하는 것이 정확한 네트워크 보간관리를 위하여 유리하다. 만약 ngi 시간동안 동기 신호가 입력되지 않는다면 모션제어기의 slave들은 네트워크 보간에 오류가 발생되어 동기가 파괴된 것으로 인지하고 현재 이송중인 축을 내부 가감속 값에 의하여 즉시 중지한다. 그리고 slave는 네트워크 보간이 파괴되어 더 이상 보간을 실행할 수 없음을 master에게 알린다. 여기서 master는 slave로부터 받은 정보로 네트워크 보간이 파괴되었음을 알 수 있을 뿐만 아니라 slave로부터 읽어 들이는 위치정보의 변화량으로도 slave의 상태를 감시할 수 있다. 또한 ngi시간은 동기신호가 들어온 다음 얼마만큼의 시간후에 보조 보간을 수행할 지를 결정하는 기준이 된다. 동기신호가 들어온

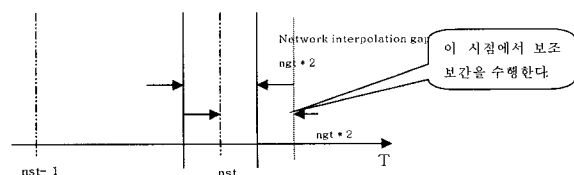


그림 14. 네트워크 보간의 갭 시간.
Fig. 14. Gap time of network interpolation.

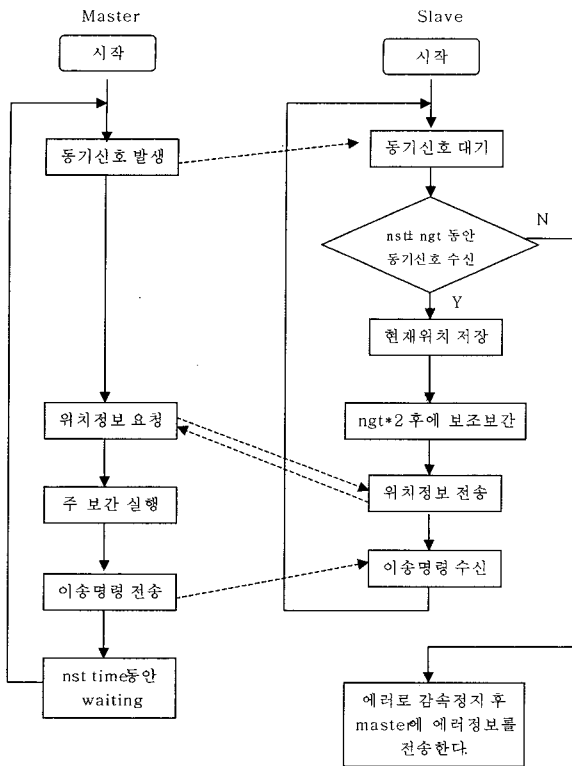


그림 15. Master와 slave 제어기간의 통신 흐름도.
Fig. 15. Communication flowchart between master and slave controller.

후 최소 $ngt*2$ 시간후에 보조 보간을 시작해야 만 네트워크 보간을 안정하게 수행할 수 있다. 이 시점은 nst 를 기준으로 $ngt*2$ 시간후이다. 그림 15는 이상에서 설명된 master와 slave 간의 통신 흐름을 나타내고 있다.

IV. 결론

제안된 기법은 일정 시간동안 주기적인 행위를 수행하는 모션 제어기의 보간 제어를 네트워크를 통하여 구현한 것으

로써 보간 동작을 master의 주 보간동작과 slave의 보조 보간 동작으로 나누어 master와 slave의 시간 동기를 네트워크를 통하여 실현하였다. Master는 slave에 일정주기로 동기명령을 지령한다. 이 지령을 받은 slave는 현재 위치를 저장하고 master에서 상태요청 명령이 지령 되면 현재 위치를 반환한다. Master는 전체 보간 시간을 구하고 이 결과값(위치합성 벡터, 이송속도, 가감속 시간, 전체 이송시간)을 slave에 전달한다. Slave는 master에서 받은 값을 참조하여 sst를 만들고 그 값을 포트출력한다. 여기서 nst 주기의 동기신호를 slave들간의 보정값으로 이용하여 slave들간의 보간 동기를 맞춘다. 만약 동기신호 위에서 규정된 시간안에 입력되지 않으면 네트워크 동기가 파괴된 것으로 간주하고 감속 정지 후 결과를 master에 송부한다. 이러한 모션제어기의 보간을 네트워크를 통해 수행하므로써 반도체 장비와 같이 많은 모타가 소요되는 장비를 위치제어하는 데 있어 단일 제어를 사용한 경우 보다 효율적인 제어가 가능하며 아울러 필요한 경우 대규모계통의 분산제어도 가능할 것이다. 기타 활용방안으로써는 복합기계의 동기제어기술 및 산업용 네트워크 시스템의 모니터링 기술에도 활용 가능할 것으로 사료된다.

참고문헌

- [1] LG산전(주), "CNC컨트롤러기술 기술개발에 관한 연구," G7 단계 보고서, 1996.
- [2] 이스택(주), "iM-series manual," 한국, 1999.
- [3] Robert Bosch GmbH, "Bosch CAN specification version 2.0," Bosch, 1995.
- [4] 신우자동화기술(주), "Motion Control Unit 사용자 설명서," 2001.
- [5] 월간자동제어계측, "FA 네트워크에서의 필드버스 기술," pp. 2-57, 1998. 4.
- [6] 김기압, 홍승호, "자동화시스템에서 profibus 네트워크 구현 및 성능평가," 제어자동화시스템논문지, 제4권, 1호, pp. 113-122, 1998. 2.



곽 군 평

1959년 3월 9일. 1982년 고려대 전기공학과 졸업. 동대학원 석사(1985), 동대학원 박사(1990). 현재 창원대학교 전기공학과 조교수 겸 신우자동화기술(주) 대표이사. 관심분야는 가변구조제어, 모션 컨트롤러.