

다수 로봇간 공간계적 동기화를 위한 모션계획 알고리즘

A Motion Planning Algorithm for Synchronizing Spatial Trajectories of Multi-Robots

임현규*, 정영도, 김성락, 이충동

(Hyun-Kyu Lim, Young-Do Jeong, Sung-Rak Kim, and Choong-Dong Lee)

Abstract : Recently the need for cooperative control among robots is increasing in a variety of industrial robot applications. Such a control framework enhances the efficiency of the real robotic assembly environment along with extending the robot application. In this paper, an ethernet-based cooperative control framework was proposed. The cooperative control of robots can multiply the handling capacity of robot system, and make it possible to implement jigless cooperation, due to realization of trajectory-synchronized movement between a master robot and slave robots. Coordinate transformation was used to relate among robots in a common coordinate. An optimized ethernet protocol of HiNet was developed to maximize the speed of communication and to minimize the error of synchronous movement. The proposed algorithm and optimization of network protocol was tested in several class of robots.

Keywords : synchronized motion, industrial robot, robot cooperation, ethernet

I. 서론

로봇은 산업 현장에서 다양한 작업을 수행한다. 최근에는 생산 공정이 고도화됨에 따라 로봇이 독자적인 작업을 수행할 뿐만 아니라 주위 로봇과 협조하는 제어방법이 대두되어 오고 있다.

로봇 간 협조제어는 크게 4가지로 분류할 수 있는데[1], 첫째로 로봇이 동일한 작업공간에 배타적으로 작업하게 하는 방법, 둘째로 동일한 작업공간에 특정한 작업시간의 관계를 이용해 작업하게 하는 방법, 셋째로 두 대 이상의 로봇이 동일한 작업물에 대해 동시에 운반하는 작업, 넷째로 한 대의 로봇이 들고 있는 작업물 위에서 다른 로봇 들이 용접이나 조립과 같은 작업을 하는 방법이다. 첫 번째 방법과 두 번째 방법은 주로 로봇 제어기의 입출력 신호를 이용한 공정 제어를 통해 이루어지며, 세 번째와 네 번째 방법은 로봇 간의 공간계적 동기화 할 수 있는 알고리즘을 이용하여 좌표변환과 기구학 연산을 통해 이루어 질 수 있다. 지금까지 로봇간의 협조제어는 단일 제어기에 의한 방법이 주로 이루어져 왔는데, 단일 제어기는 동기화된 지령을 출력하기 쉽다는데 그 이유가 있다. 이러한 이유로 포지셔너와 로봇간의 협조 동기 작업은 하나의 제어기에 의해 수행된다[2]. 그러나 포지셔너와 같은 단순한 매니플레이터가 아닌 다수(N대)의 로봇을 하나의 제어기로 제어하는 기존방식은 N대 분의 제어에 필요한 계산시간이 필요하지만 한 대의 로봇을 제어하도록 설계된 제어기에서는 적용이 불가능하다. 또한 계산 성능이 높은 한 대의 제어기가 여러 대의 로봇을 제어할 수 있다하더라도 생산 공정의 바뀜에 따른 로봇의 용도의 변경 및 재배치가 용이하지 못한 단점이 있다. 이에 따라 본 논문에서는 다수의 로봇 제어기 간에

이더넷(ethernet)에 기반한 통신을 통하여 궤적을 동기 제어 하는 방법과 기구학적인 계산식을 제시하고자 한다. 이 방법을 이용하면 기존 제어기의 성능을 개선하지 않고도 적용이 가능하며, 동기 성능 또한 한 대의 제어기로 제어할 때와 별 차이 없이 가능하다는 장점이 있다.

또한 공간 궤적 동기화를 통해 그림 1, 2와 같은 다수 로봇

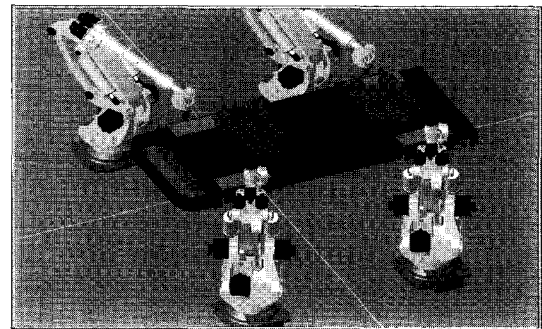


그림 1. 다수 로봇의 협조 핸들링.

Fig. 1. Cooperative handling among robots.

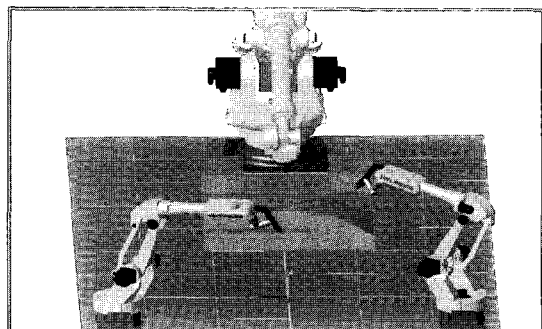


그림 2. 지그리스 협조제어.

Fig. 2. Jigless cooperative control.

* 책임저자(Corresponding Author)

논문접수 : 2004. 10. 25., 채택확정 : 2004. 11. 11.

임현규, 정영도, 김성락, 이충동 : 현대중공업(주)

(hklim@hhi.co.kr/jyd@shinbiro.com/ksr@hhi.co.kr/cdleec@hhi.co.kr)

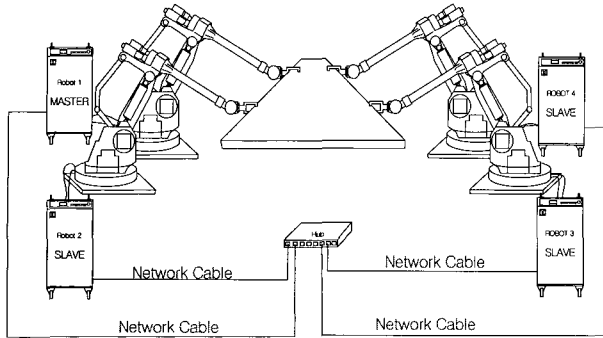


그림 3. 이더넷 통신 기반의 협조제어 개요.
Fig. 3. Outline of cooperative handling among robots.

을 이용한 핸들링 작업이나 지그리스 협조 작업이 가능해진다. 이더넷 기반 통신을 이용한 로봇 협조는 그림 3과 같이 각 로봇 제어기를 이더넷으로 연결하도록 구성한다. 협조 제어를 위해서는 기준 위치 궤적 지령을 생성하는 마스터 로봇과(master robot) 기준위치 궤적을 동기화하여 추종하면서 자신의 작업을 수행하는 슬레이브 로봇들로(slave robots) 구성된다. 당사에서는 마스터와 슬레이브간 통신 수단으로서 자체 개발한 이더넷을 기반으로 하는 전용통신 프로토콜(HiNet)을 사용하였다. HiNet을 간단히 소개하면 제어기가 초기화 할 때 로봇을 제어기에서 설정한 일련번호로 인식하도록하고, 로봇 간의 현재 제어 상태 및 지령 위치 정보를 주고받을 수 있도록 개발되어 있다. HiNet에 대해서는 자세한 구조적 설명은 생략하고 성능에 대한 소개와 구현 방식에 대해서는 4장에서 후술하도록 하겠다.

본 논문에서는 다수 로봇간의 공간제어를 동기화하는 알고리즘을 제시하기 위해서, 로봇 상호간의 좌표계 변환 방법, 마스터 로봇과 슬레이브 로봇의 위치 지령 생성 및 동기화 방법을 제시하고, 전용 통신 프로토콜의 성능을 소개하며, 이를 이용한 적용 결과에 대해 소개하고 결론을 맺는다.

II. 로봇간 좌표계 설정

1. 로봇 기구학

로봇 기구학은 각 관절의 링크 변위를 이용하여 TCP (Tool Center Point)의 위치와 자세를 구하는 순기구학(forward kinematics)과 로봇이 도달가능한 공간상의 위치와 자세를 각 관절의 변위로 환산하는 역기구학(inverse kinematics) 있다. 시리얼 링크 구조를 가지는 로봇의 순기구학은 Denavit-Hartenberg 파라미터를 이용하여 변환 행렬의 곱으로 쉽게 구할 수 있다. 회전관절로 이루어진 6축 다관절 로봇의 경우 로봇 TCP의 위치와 자세에 대한 4x4 동차변환행렬(homogeneous transform matrix) H 는 인접한 관절간의 좌표계 변환 동차변환행렬 T 의 곱으로 다음과 같이 계산된다[3].

$$H = {}^0T_1 \cdots {}^{i-1}T_i \cdots {}^5T_6 T_{tool} \quad (1)$$

여기서, ${}^{i-1}T_i = {}^{i-1}T_i(\theta_i) = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}d_i \\ 0 & 1 \end{bmatrix}$ 의 동차

변환 행렬이며 ${}^{i-1}R_i$ 는 $i-1$ 조인트 좌표계에서 바라본 i 조인트의 3x3 자세변환 행렬이고, ${}^{i-1}d_i$ 는 위치에 대한 3x1 위치 벡터이며 각 회전관절의 변위 θ_i 의 함수이다.

역기구학은 (2)와 같이 로봇의 TCP의 위치 (x, y, z) 와 자세 (w_x, w_y, w_z) 에 대해 관절의 변위 θ_i 를 구하는 것인데, 순기구학처럼 단순하게 완전해로 결정되지 않을 뿐 아니라 형상에 따라 복수개의 해가(solution) 나올 수 있는 복잡함이 있다.

$$(\theta_1, \theta_2, \dots, \theta_6)^t = f_{inv}(x, y, z, w_x, w_y, w_z) \quad (2)$$

그러나 관절 4, 5, 6축의 좌표계를 동일한 중심점을 기준으로 설정할 수 있는 6축 수직 다관절 로봇은 기구학적으로 4, 5, 6번째 관절과 1, 2, 3번째 관절이 디커플링이(decoupling) 되어 역기구학을 반복해기(iterative solution) 아닌 완전해로(closed form solution) 구할 수 있으며 로봇의 형상을 지정하면 하나의 해로 결정된다[3]. 일반적인 로봇은 서보모터의 제어를 위한 엔코더나(encoder) 리졸버기(resolver) 유일한 로봇의 위치 측정 수단이 되는데, 로봇의 TCP위치가 제어하고자 하는 위치에 도달했는지를 검사하기 위해서는 엔코더 위치를 TCP의 위치로 환산이 필요하며, 직교 좌표계로(cartesian coordinate) 계획된 지령을 로봇의 위치제어를 위해 각 관절의 위치로 환산이 필요하기 때문에 두 기구학은 로봇의 피드백 제어에 필수적인 요소가 된다.

수학적인 기구학 모델은 CAD데이터에 의한 이상적인 값으로 실제 기구에는 가공오차, 설치오차, 변형오차로 인해 모델과의 차이가 발생한다. 특히 기구학 파라미터의 오차는 로봇의 정확도에 가장 크게 영향을 미친다[4]. 따라서 다수 로봇 간 궤적동기화를 위해서는 로봇 자체의 정확도를 높이는 캘리브레이션(calibration) 과정이 선행되어야 한다. 로봇 캘리브레이션은 모델링(modeling), 파라미터의 선정범위 결정(parameter identification), 데이터 수집방법(measuring), 목적함수(cost function)의 구성 및 최소화하는 알고리즘에 이르기까지 다양한 방법이 제시되어 오고 있으나[4][5][6], 실제 산업 현장에서는 고가의 측정 장비를 이용한 캘리브레이션은 실용적이지 못하기 때문에, 현장에서 측정 장비 없이 로봇 관절의 엔코더값을 순기구학과 역기구학을 이용하여 기구학 파라미터를 캘리브레이션하는 실용적인 방법이 사용된다[7].

2. 로봇간의 좌표계 관계

다수 로봇간의 좌표계의 관계는 마스터와 슬레이브 로봇의 좌표계 상관관계로 일반화된다. 그림 4에서 마스터 로봇의 베이스 좌표계 {M}에서 엔드이펙터(end-effector) 좌표계 {MT}의 위치와 자세는 (1)에서 구해지고 위치를 P_m , 마스터 베이스 좌표계 기준 엔드이펙터의 자세를 ${}^MR_{MT}$ 로 표시한다. 한편 슬레이브의 베이스 좌표계를 {S}라고 할때, 마찬가지로 슬레이브 베이스 좌표계 기준 슬레이브의 엔드이펙터 좌표계 위치는 P_s , 자세는 ${}^SR_{ST}$ 로 표시할 수 있다. 마스터와 슬레이브 사이의 궤적 동기화를 위해서는 마스터의 엔드이펙터 좌표계를 기준으로 한 슬레이브의 엔드

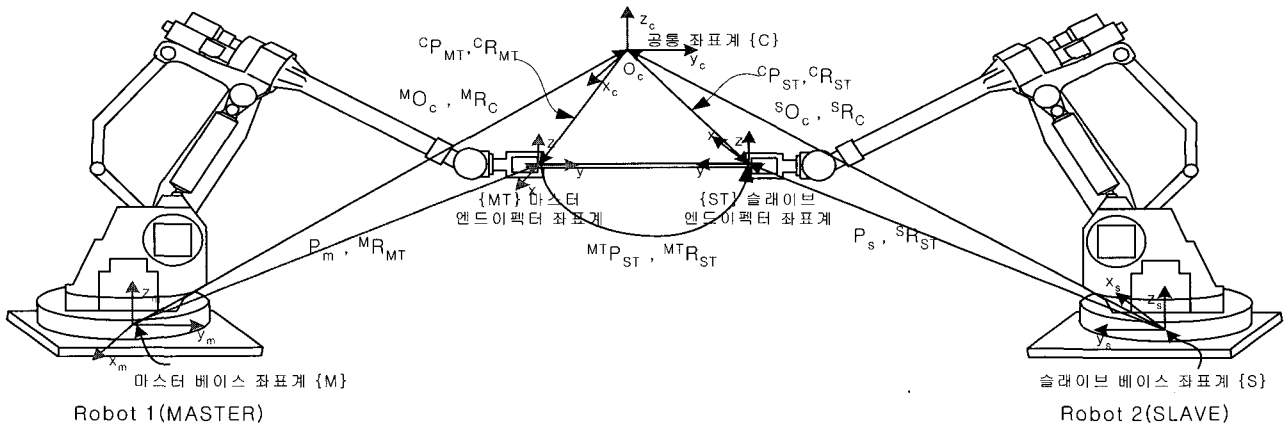


그림 4. 로봇간의 좌표계 관계.
Fig. 4. Coordinate relations between robots.

이펙터 좌표계의 위치 ${}^M P_{ST}$ 와 자세 ${}^M R_{ST}$ 관계가 필요하며 두 관계를 알기 위한 방법으로 여기서는 마스터와 슬레이브가 공통의 좌표계{C}를 설정한다. 공통 좌표계가 있으면 공통 좌표계 기준으로 마스터와 엔드이펙터 좌표계와 슬레이브의 엔드이펙터 좌표계가 환산될 수 있어 ${}^M P_{ST}$ 와 ${}^M R_{ST}$ 관계를 구할 수 있다. 마스터 베이스 좌표계 기준으로 본 공통 좌표계의 위치를 ${}^M O_c$ 로하고 자세를 ${}^M R_c$, 그리고 슬레이브와 공통좌표계의 위치 ${}^S O_c$ 와 자세를 ${}^S R_c$ 로 표시하며, 본 논문에서 P와 O는 3x1 위치 벡터를 나타내고, R은 3x3회전 행렬을 나타낸다.

3. 공통좌표계의 생성

마스터 로봇에 의해 생성된 궤적을 슬레이브로 전송하기 위해서는 공통좌표계가 필요하고 공통좌표계와 베이스 좌표계 상호간의 좌표계 변환 관계식을 각 제어기가 가지고 있어야 한다. 앞에서 설명한 바와 같이 각 로봇의 베이스 좌표계에서 TCP까지의 관계는 순기구학을 통해 구해지므로 공통좌표계와 베이스 좌표계와의 관계식을 정해 놓으면 공통좌표계와 엔드이펙터 좌표계까지의 관계를 구할 수 있으며, 두 로봇의 툴 좌표계의 관계는 공통 좌표계{C}를 이용하여 환산이 가능하다. 로봇의 좌표계를 설정방식은 공간상의 3점을 교시하여 그 위치를 이용해 하나의 좌표계를 구성하는 방법이 주로 사용된다. 그림 5는 두 로봇의 TCP를 일치 시켜 공통 좌표계를 설정하는 방법을 나타낸다. 마스터와 슬레이브가 서로 도달할 수 있는 작업 공간을 최대한 크게 잡아 3점을 동일한 순서로 마스터 제어기에 교시하면 로봇은 각축의 엔코더(encoder)를 이용해 각 관절의 위치를 환산할 수 있고 순기구학을 통해 TCP의 위치가 구해진다. 슬레이브 또한 마스터의 TCP 위치와 동일한 위치에 동일한 순서로 슬레이브의 위치를 교시하면 공간상의 동일한 점에 대해서, 마스터는 마스터 베이스 좌표계{M} 기준의 3점의 위치가 구해지고 슬레이브도 마찬가지로 구해진다. 마스터 좌표계{M}를 기준으로 설명하면, 공통 좌표계의 관계를 구성하는 방법으로는 다음과 같은 방법이 가능하다. 우선 로봇 좌표계에서 본 공통 좌표계의 중심을 3

점의 중심으로 정의한다.

$${}^M O_c = \frac{{}^M P_a + {}^M P_b + {}^M P_c}{3} = \{x_c, y_c, z_c\}^T \quad (3)$$

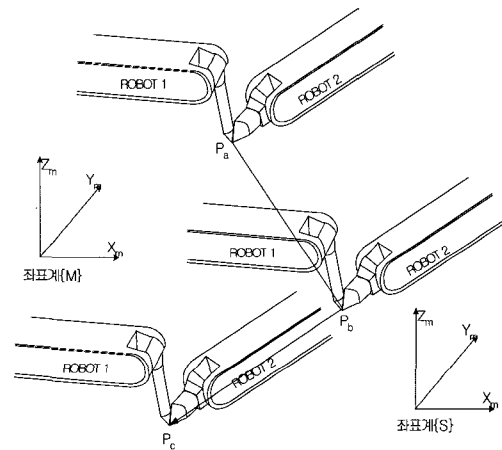


그림 5. 공통 좌표계 설정.
Fig. 5. Process of setting a common coordinate between robots.

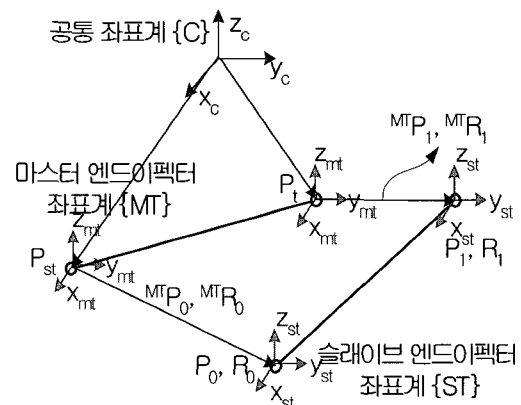


그림 6. 마스터와 슬레이브의 위치 관계.
Fig. 6. Coordinate relation between end-effector of the master and the slave.

공통 좌표계의 자세를 정의하기 위해서 Z 방향 벡터를 정의한다. Z 방향은 P_a, P_b, P_c 로 만들어지는 평면에 수직하고 중심점을 기준으로 P_a 에서 P_b 를 오른손 나사 방향으로 나오는 단위 벡터를 Z방향으로 정의한다.

$$\overline{Mz_C} = \frac{({}^M P_a - M O_C) \times ({}^M P_b - M O_C)}{\| {}^M P_a - M O_C \| \| {}^M P_b - M O_C \|} \quad (4)$$

X 방향은 3점의 중심에서 P_a 으로 향하는 단위벡터로 정의한다.

$$\overline{Mx_C} = \frac{{}^M P_a - M O_C}{\| {}^M P_a - M O_C \|} \quad (5)$$

공통 좌표계의 Z방향과 X방향이 결정되었으므로 Y방향 벡터를 구할 수 있다.

$$\overline{My_C} = \overline{Mz_C} \times \overline{Mx_C} \quad (6)$$

상기의 간단한 (4), (5), (6)의 과정으로 구한 각 방향 3×1 벡터를 방향벡터로 구성되는 행렬을 만들면 로봇 좌표계에서 본 공통 좌표계의 3차원 직교행렬 $SO(3)$ (special orthogonal group of order 3)의 형태로 자세변환 행렬이 구해진다.

$${}^M R_C = \{ \overline{Mx_C}; \overline{My_C}; \overline{Mz_C} \} \in SO(3) \quad (7)$$

마찬가지 방법으로 협조제어를 하는 모든 슬래이브 로봇에 대해서도 공통좌표계의 원점 ${}^S O_C$ 와 자신의 베이스 좌표계에서 본 공통좌표계의 자세 ${}^S R_C$ 를 구할 수 있다.

III. 보간계획

협조제어의 보간 계획을 위해 마스터가 P_{st} 위치에서 P_t 위치로 궤적 이동시간을 t_f 로 계획하여 직선 이동하고, 슬래이브는 마스터가 P_{st} 위치에 있을 때 P_0 위치로부터 마스터의 위치가 P_t 일 때 슬래이브는 P_1 위치를 직선 이동할 경우를 예로 들어 설명하고자 한다.

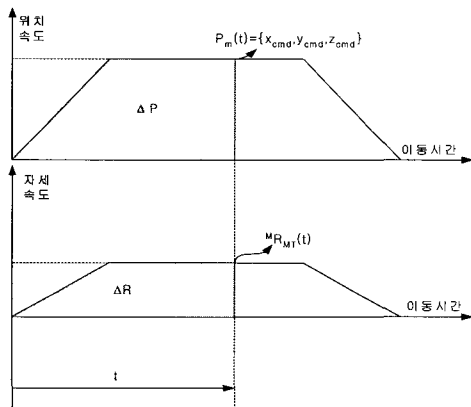


그림 7. 위치와 자세의 속도 프로파일.
Fig. 7. Velocity profiles of positions and orientations.

그림 6에 마스터와 슬래이브 이동위치에 따른 관계를 나타내었다. 마스터 로봇은 P_{st} 에서 P_t 까지의 직선 궤적을 계획하여 위치와 자세의 속도 프로파일을 만든다. 마스터 엔드이펙터 좌표계는 모션 지령 생성주기마다 만들어지는 속도프로파일의 위치에 따라 위치와 자세가 변하게 되는데 그림 7에서와 같이 속도 프로파일로 부터 일정시간 (θ)이 지났을 때 시작위치로부터 이동거리는 속도프로파일의 면적으로 계산되고 시작위치에 이 면적을 더하여 이번 주기에 실행할 이동할 위치가 계산된다.

일정 시간(θ)이 경과 되었을 때 생성된 마스터 로봇의 지령위치 $P_m(t)$ 와 로봇의 자세 ${}^M R_{MT}(t)$ 는 다음과 같은 식으로 구할 수 있다.

$$P_m(t) = P_{st} + \Delta P_m \quad (8)$$

$${}^M R_{MT}(t) = {}^M R_{st} \cdot \Delta R_m \quad (9)$$

단, $0 < t < t_f$, $P_m(0)$ 는 P_{st} , $P_m(t_f)$ 는 P_t

한편 슬래이브의 보간 계획은 마스터 엔드이펙터와 상대적인 관계를 계획하여야 하므로 마스터 엔드이펙터 좌표계 기준으로 수행되어야 한다. 따라서 P_0, R_0 와 P_1, R_1 를 마스터의 엔드이펙터 좌표계{MT} 기준으로 환산한 ${}^{MT} P_0, {}^{MT} R_0, {}^{MT} P_1, {}^{MT} R_1$ 로 슬래이브 보간 계획에 사용한다. 슬래이브의 시작위치

$${}^{MT} P_0 = ({}^C R_{st})^{-1} \cdot ({}^C P_0 - {}^C P_{st}) \quad (10)$$

$${}^{MT} R_0 = {}^C R_{st}^{-1} \cdot {}^C R_0 \quad (11)$$

슬래이브의 종료위치

$${}^{MT} P_1 = ({}^C R_t)^{-1} \cdot ({}^C P_1 - {}^C P_t) \quad (12)$$

$${}^{MT} R_1 = {}^C R_t^{-1} \cdot {}^C R_1 \quad (13)$$

슬래이브는 마스터와 시작 및 종료 시점을 동기화하기 위해서 (11)~(13)에 의한 시작위치와 자세를 이용하여 마스터와 동일한 궤적 이동시간 t_f 를 가지는 속도 프로파일을 계획하여 마스터와 마찬가지로 마스터 엔드이펙터 좌표계 기준의 직선 궤적을 시간 t 에 대한 위치와 자세를 생성할 수 있다.

$${}^{MT} P_{ST}(t) = P_0 + \Delta P_s \quad (14)$$

$${}^{MT} R_{ST}(t) = {}^{MT} R_0 \cdot \Delta R_s \quad (15)$$

IV. 지령생성

마스터와 슬래이브 로봇 간에 역할을 간략화한 흐름도가 그림 8에 나타나있다. 마스터는 매 지령 생성 주기마다 속도 프로파일에 의해 생성된 (8), (9)로 나타난 위치와 자세를

공통 좌표계로 환산하여 슬래이브로 송신하고 역기구학을 풀어 서보에 출력하는 과정을 되풀이하고, 슬래이브 쪽에서는 마스터의 엔드 이펙터의 위치를 수신하여 (14), (15)로 나타낸 슬래이브의 보간 계획에 의한 위치와 자세를 더한 위치를 생성하고, 공통 좌표계와 슬래이브 베이스 좌표계의 상관관계를 이용하여 슬래이브 좌표계로 환산하여 역기구학을 풀어 서보에 각 관절의 지령을 출력하는 과정을 되풀이한다.

1. 마스터 로봇

마스터와 슬래이브의 좌표계 변환은 공통 좌표계를 매개로 한 관계에 의해 변환되며 우선 마스터 로봇 좌표계를 공통 좌표계로 변환한다. (3), (7)에서 설정한 마스터 로봇의 베이스 좌표계에서 공통 좌표계까지의 상대 위치 ${}^M P_C$, 자세 ${}^M R_C$ 를 통해 마스터의 TCP 위치를 공통좌표계의 기준으로 나타낼 수 있다. 속도 프로파일에서 일정 시간 t 가 소요되었을 때 공통 좌표계에서 본 마스터 TCP의 위치를 ${}^C P_{MT}(t)$ 로 정하면 다음 식과 같이 계산된다.

$$\begin{aligned} {}^C P_{MT}(t) &= {}^C R_M(P_m(t) - O_c) \\ &= {}^M R_C^{-1}(P_m(t) - O_c) \end{aligned} \quad (16)$$

또한 공통 좌표계에서 본 마스터 TCP의 자세는 다음 식과 같다.

$${}^C R_{MT}(t) = {}^C R_M \times {}^M R_{MT}(t) \quad (17)$$

위의 (16), (17)와 같이 변환된 마스터의 지령은 전용 이더넷 통신을 통해 슬래이브 제어기로 송신된다.

2. 슬래이브 로봇

한편 슬래이브 제어기는 (16), (17)의 마스터에서 지령이 들어온 지령에 (14), (15)에서 구한 슬래이브의 보간 계획을 더하여 ${}^C P_{ST}$, 와 ${}^C R_{ST}$ 를 계산한다.

$${}^C P_{ST}(t) = {}^C P_M(t) + {}^M R_C^{-1} \cdot {}^M P_{ST}(t) \quad (18)$$

$${}^C R_{ST}(t) = {}^C R_M(t) \cdot {}^M R_{ST}(t) \quad (19)$$

슬래이브는 역기구학 풀이를 위해 공통좌표계의 TCP 위치를 슬래이브 베이스 좌표계로 환산하며 다음과 같이 계산된다.

$$P_s(t) = {}^S O_C + {}^S R_C \times {}^C P_{ST}(t) \quad (20)$$

$${}^S R_{ST}(t) = {}^S R_C \times {}^C R_{ST}(t) \quad (21)$$

슬래이브의 위치와 자세가 (20), (21)에서 결정되면 역기구학을 풀어 서보에 지령을 출력하며 마스터와 슬래이브 지령 생성에 대한 과정이 이루어진다.

3. 지령 동기화

지령 동기화를 위해서는 궤적 계획 및 지령생성은 마스터와 슬래이브가 별개의 제어기에 의해 제어되므로 경과시

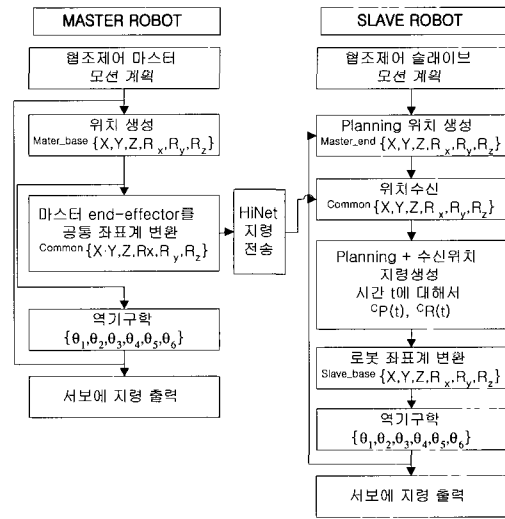


그림 8. 마스터와 슬래이브의 지령생성 흐름도.

Fig. 8. The flowchart transferring command from a master to slaves.

표 1. HiNet의 데이터 송수신 소요시간.

Table 1. HiNet's data transmission time.

	평균(Avg.)	최대(Max.)
송신함수 소요시간 (transmitting time)	96.5 (us)	135 (us)
수신함수 소요시간 (receiving time)	108.6 (us)	154 (us)
송수신 시간 (rotation time)	960.0 (us)	1091 (us)

간 t 에 관한 동기화와 t 일 때 동일한 모션 지령 시퀀스를 처리하는 동기화가 함께 이루어져야 한다. 이를 위해 당사에서는 이더넷 기반의 전용 프로토콜 HiNet을 개발하여 사용하였다. HiNet은 슬래이브가 마스터의 지령을 실시간으로 수신하기 위한 수단으로 IEEE802.3 규약[8]을(10~100Mbps) 사용하며 마스터가 슬래이브에 지령을 송신할 때 전송방식은 브로드캐스팅(broadcasting) 방식을 사용하여[9], 마스터는 일방적으로 지령을 송신하고 슬래이브는 실시간으로 데이터를 수신하기만 하면 되는 구조를 채택하였다. 따라서 마스터의 송신 시점과 슬래이브의 수신시점의 동기화는 마스터의 송신 속도와 슬래이브가 수신 인터럽트 주기에 의해 좌우된다. HiNet의 통신 성능은 10Mbps 미디어 환경일 때 표 1과 같이 측정되었다. 시험 환경은 제어기 2대를 스위칭 허브로(switching hub) 연결하고 310 바이트(byte)의 데이터에 대해 1000번의 데이터 송수신 시험을 10회 수행하였다. 데이터를 송신하여 수신하는데 까지 걸리는 송수신 시간(rotation time)은 평균 960 us, 최대 1091 us정도 측정되었고, 송수신 시간을 동일하다고 가정하면 간단한 산술계산으로 최대 505 us의 송신 시간이 소요되어, HiNet의 실질 데이터 송수신 속도는 약 4.9 Mbps이다. 궤적의 연속적인 동기화를 위해서는 지령 데이터의 연속성 보장되어야 하는데 72시간 이상의 네트워크 시험에서 단 한번의 송

수신 에러도 발생하지 않는 안정성을 보였다. 슬레이브가 마스터의 움직임에 따른 모션을 구현하기 위해 마스터가 슬레이브에 반드시 넘겨주어야 하는 데이터는 (16)과 (17)에서 만들어지는 위치와 자세로 6개의 실수형(real type) 데이터가 필요하므로 실제로는 310바이트보다 훨씬 적은 48 바이트 정도의 데이터 량이 요구된다. 안전율을 충분히 고려하여 지령데이터의 송신시간을 0.5msec라고 하더라도 모션지령 생성주기가 데이터를 송신하는데 걸리는 시간보다 크면 문제가 없도록 설계할 수 있다. 마스터와 슬레이브의 궤적 동기화를 위해 두 개의 인터럽트 주기를 사용하는데, 마스터와 슬레이브의 모션지령 생성주기를 Δt_{ic} 와 슬레이브가 마스터의 최초 동작 개시 신호를 검사하는 인터럽트 주기 Δt_r 이다. 마스터는 슬레이브에 Δt_{ic} 의 주기마다 슬레이브에 지령을 송신하고, 슬레이브는 마스터의 지령송신 주기보다 빠른 $\Delta t_{ic}/n$ (n 은 정수) 주기로 지령 수신 여부를 확인한다. 궤적의 동기화를 위해 마스터와 슬레이브는 지령을 곧바로 서보에 출력하지 않고 버퍼에 쌓아두고 개시신호가 입력될 때 까지 대기한다. 마스터는 버퍼에 1개 이상의 지령이 입력되면 개시 신호를 슬레이브에 송신함과 동시에 버퍼의 지령을 순차적으로 서보로 보내기 시작하고, 슬레이브도 개시신호 수신 주기 인터럽트에서 신호를 감지함과 동시에 버퍼의 지령을 서보에 순차적으로 출력하기 시작한다. 마스터의 개시 신호는 1비트(bit)로도 가능하므로 데이터 량이 작아 전송속도가 빠르다.

따라서 슬레이브의 개시 신호 수신 인터럽트 주기 Δt_r 은 짧아져야 하며, 개시 신호 수신 인터럽트 주기가 짧아질수록 좋은 동기 성능을 발휘할 수 있다. 마스터와 슬레이브 사이의 궤적 동기오차 (T_{err})는 슬레이브의 개시 신호의 수신 인터럽트 주기 (Δt_r)와 마스터 로봇의 이동 속도 (v_{cmd})와의 관계로 나타낼 수 있다.

$$T_{err} = \Delta t_r \times v_{cmd} \quad (22)$$

모션이 서보에 지령을 생성하는 주기 Δt_{ic} 는 Δt_r 보다 긴 간격으로 이루어지기 때문에, 일단 개시 신호를 송수신한 이후부터 궤적 지령이 완료될 때 까지 슬레이브는 Δt_r 주기로 개시 신호 수신 여부를 감시하지 않아도 된다. 만약 모션 지령의 생성 주기 Δt_{ic} 를 10msec이고 Δt_r 이 1msec 라면 모션 지령의 계산과 송수신은 매 10msec마다 이루어지고 서보에 지령을 출력하는 타이밍은 Δt_r 로 최초 한번만 동기화 시키면 되므로, 본 알고리즘의 모션지령 계산이 Δt_{ic} 이내에 이루어지면 되어 기존 제어기에 구현할 때에도 연산시간에 부담이 적은 장점이 있다. 다수의 로봇 제어기는 인터럽트가 독립적으로 실행되기 때문에 제어기 간의 동기화는 제어기 CPU의 최소 인터럽트 주기에 의해 제한을 받게 된다. 따라서 이더넷의 통신 속도가 빨라지고 연산이 빠른 CPU를 사용할 경우 Δt_r 를 줄일 수 있어 동기 궤적에 대한 성능은 향상될 것으로 예상된다.

V. 실험

본 알고리즘의 타당성 및 HiNet의 성능을 검증하고자 마스터 로봇에서 생성된 궤적에 따른 슬레이브 지령 슬레이브 지령의 추종 시험을 수행하였다. 이 실험은 본 알고리즘을 적용할 때 기인하는 계산 오차의 정도를 파악하는데 목적이 있다. 시험에 사용된 로봇과 제어기 주요 사양은 표 2에 표시하였다.

본 실험에서는 (14)에서 나타나는 ${}^{MT}P_{ST}$ 벡터의 크기를 시간에 따라 변하지 않도록 ΔP_s 를 0으로 고정하여 마스터와 슬레이브 엔드이펙터 간의 상대 거리를 4.49mm로 고정하고 마스터가 지령을 생성하여 송신한 지령을 슬레이브가 지령을 수신하고 추종 지령을 생성하는 과정을 통해 알고리즘을 검증하였다. 시험궤적은 공통좌표계 기준으로 환산하여 그림 9에 나타내었다. 그림 10, 11, 12는 마스터가 Δt_{ic} 마다 송신한 지령과 슬레이브가 Δt_{ic} 마다 수신한 공통좌표계 기준 위치 데이터를 표시하였다. 식별이 쉽도록 매 30 샘플 점당 마스터 기호(■:마스터 송신위치, ○:슬레이브 수신위치)를 삽입하였다. 그림에서 보는 바와 같이 송신하는 x,y,z방향 좌표 값과 슬레이브가 Δt_{ic} 마다 수신한 좌표값이 누락과 지연 없이 정확하게 일치함을 보여준다.

표 2. 로봇 및 제어기의 제원.

Table2. Specification of a robot and a controller.

로봇 모델/제어기	HYUNDAI HX165-02/Hi4
가반중량	165 kg
자유도	6
구동 방식	AC SERVO MOTOR
위치 반복정도	±0.3mm
CPU	Intel i960JT (32bit RISC, 100MIPS)
서보 DSP	TMS320C31(50 MFLOPS)
제어기 최대 제어축수	12

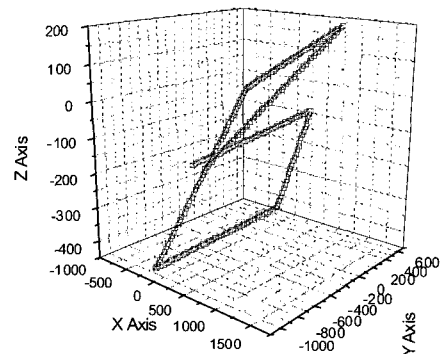


그림 9. 3차원 시험 궤적.

Fig. 9. 3D plot of trajectory.

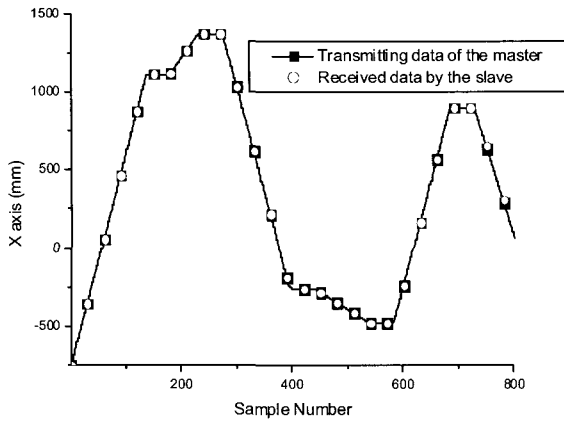


그림 10. 마스터 슬레이브간 송수신 지령(x 방향).
Fig. 10. Transmitting and received data between the master and the slave(x dir).

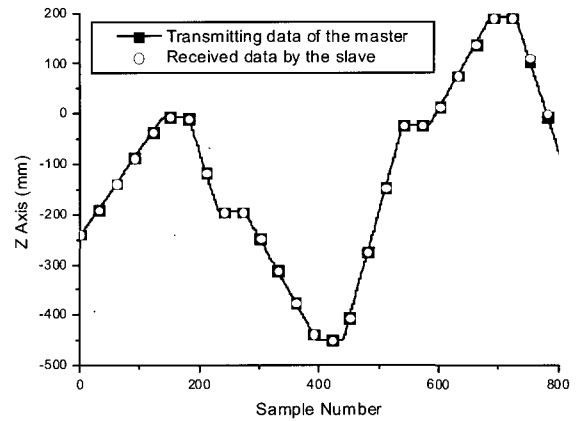


그림 12. 마스터 슬레이브간 송수신 지령(z 방향).
Fig. 12. Transmitting and received data between the master and the slave(z dir).

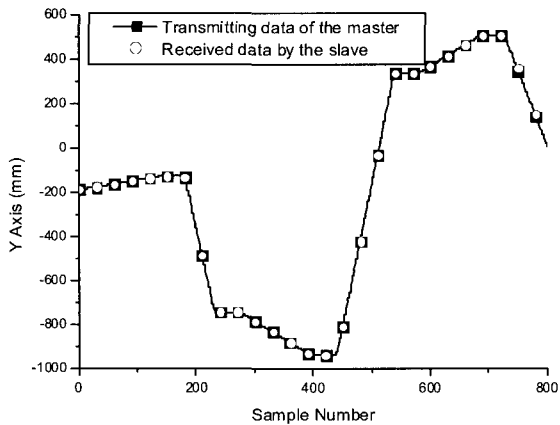


그림 11. 마스터 슬레이브간 송수신 지령(y 방향).
Fig. 11. Transmitting and received data between the master and the slave(y dir).

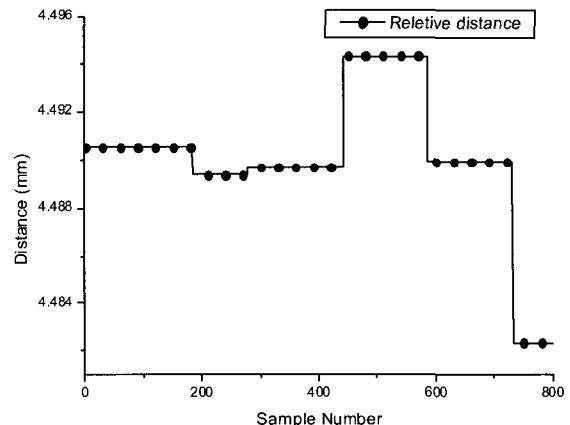


그림 13. 마스터와 슬레이브의 TCP간의 상대거리.
Fig. 13. Relative distance between TCPs of the master and the slave.

그림 13은 마스터가 슬레이브 사이의 상대 거리를 나타내며 기구학 계산, 좌표변환 계산, 엔코더의 분해능, 좌표값의 유효숫자의 범위 등에 의해 기인하는 계산오차로 (truncation error) 0.015mm 수준으로 확인되었다. 로봇의 정밀도가(repeatability) 0.3mm 수준이므로 동기 정확도에 영향을 미치지 않을 작은 크기이다. 두 로봇간의 동기 성능에 가장 큰 영향을 미치는 마스터와 슬레이브의 첫 번째 궤적 지령에 대한 동기 타이밍은 슬레이브의 개시 신호 수신 주기 Δt_r 을 1msec로 설정했을 때, 그림 14에서 보는 바와 같이 평균 0.4~0.5msec이고 최대값이 1msec를 넘지 않음이 확인되었다. 따라서 (22)에 의해 1000mm/s의 로봇 이동속도에서 본 알고리즘을 통해 평균 0.4~0.5mm수준의 동기 성능이 예상된다. 다만 실제 동기 성능은 로봇의 캘리브레이션에 의한 위치결정 정확도와(pose accuracy) 궤적추종 정확도의(path accuracy) 영향을 고려해야 한다.

VI. 결론

본 논문에서는 다수 로봇 제어를 이용하여 기존 제어기의 계산 성능을 향상하지 않고도 적용이 가능한 궤적 동기화를 위한 방법을 제시하였다. 이를 위해 마스터와 슬레이브 간의 좌표계 관계를 정의하고, 마스터와 슬레이브 제어가 각각 한번의 보간 계획과 역기구학 계산을 풀도록 하여 각 제어기에 계산량을 균등하게 분배하였다. 또한 마스터 로봇은 슬레이브 로봇에게 공통 좌표계로 환산한 TCP 좌표 값만을 넘겨주는 것만으로 충분하기 때문에 통신 데이터 량이 최소화되는 방법이 제시되었다. 본 알고리즘을 현대중공업 Hi4제어기에 구현한 결과, 그림 15와 같이 이동속도 600mm/s로 HX165 두 대의 로봇이 동시에 자동차 루프(roof)를 옮기는 자동차 생산 공정에 성공적으로 적용되었다. 실제 적용에서 마스터와 슬레이브 간의 궤적 동기화는 본 논문에서 제시한 마스터와 슬레이브의 지령 데이터를 정

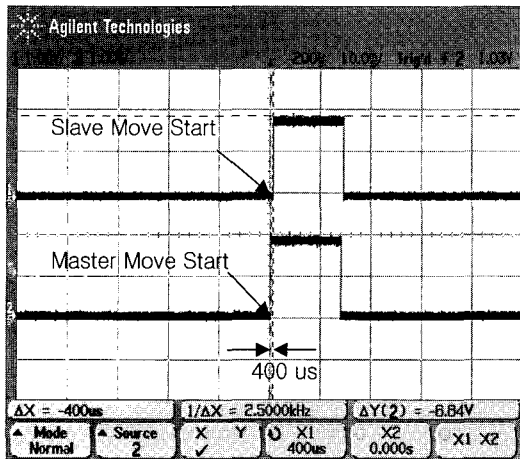


그림 14. 마스터와 슬레이브의 동기 시작타이밍.
Fig. 14. Start timing between the master and the slave.

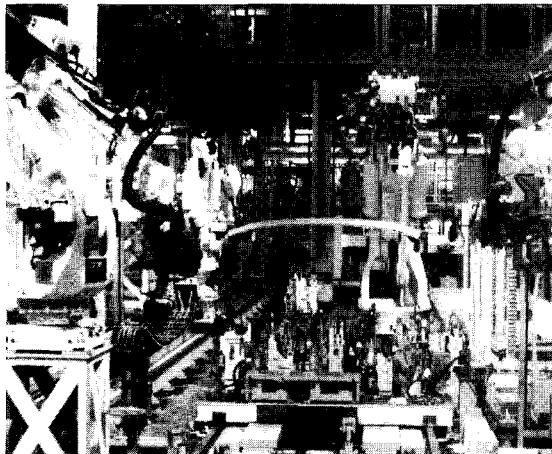
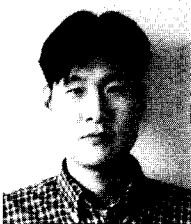


그림 15. 현장 적용 사례.
Fig. 15. An example of field application.

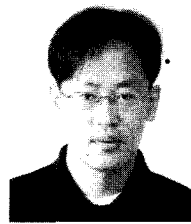
확히 동기화 시키는 개시 신호 타이밍의 정확도뿐만 아니라, 로봇 자체의 위치 결정 정확도(pose accuracy)와 궤적 정확도(path accuracy)에 의해 더 큰 영향을 받는다. 따라서 향후 통신 속도의 향상과 개시 신호 수신 인터럽트 주기가 짧아짐뿐만 아니라 로봇의 정확도 향상 및 궤적 정확도를 높일 수 있는 연구가 이루어져 성능 향상이 현실화됨에 따라 실질적인 다수 로봇 간 공간 궤적의 동기성능 또한 향상될 것으로 기대된다.

참고문헌

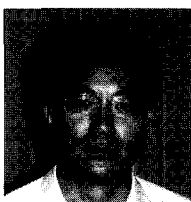
- [1] K. Stoddard, "Multi-controller robot cooperation," *35th International Symposium on Robotics*, 2004.
- [2] 임현규, 강태선, 김동혁, 김성락 "산업용 로봇의 포지셔너 동기기능 개발," 기술현대, vol. 21, no. 3, 2001.
- [3] M. W. Spong, M. Vidyasagar, *Robot Dynamics and Control*, 1989.
- [4] R. Bernhardt, S. L. Albright, *Robot Calibration*, Chapman & Hall, 1993.
- [5] H. Zhuang, Z. S. Roth, "Camera-aided robot calibration," CRC press, 1996.
- [6] D. Bennet and J. M. Hollerbach, "Autonomous calibration of single loop closed kinematic chains formed by manipulators with passive endpoint constraints," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 597-606, 1991.
- [7] 박재성, 김성락, "Robot tool calibration 기능 개발," 기술현대, vol. 18, no. 4, 1998.
- [8] W. Richard Stevens, *TCP/IP Illustrated Volume1 The Protocols*, Addison Wesley, 1994.
- [9] G. Pardo-Castellote, S. Schneider, and M. Hamilton, "NDDS: the real-time publisher subscribe middleware," *Proceedings of the IEEE Real-Time Systems Symposium*, 1997.



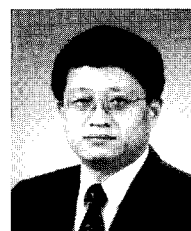
임 현 규
1998년 고려대학교 기계공학과 졸업.
2000년 동대학원 석사. 현대중공업(주) 기계전기연구소 로봇개발실 연구원.



정 영 도
1993년 서울대학교 제어계측공학과 졸업. 1995년 동대학원 석사. 현대중공업(주) 기계전기연구소 로봇개발실 선임 연구원.



김 성 락
1984년 서울대학교 전기공학과 졸업. 1986년 KAIST 전기 및 전자공학과 석사, 동대학원 박사(1993). 현대중공업(주) 기계전기연구소 로봇개발실 실장.



이 충 동
1977년 서울대학교 조선해양공학과 졸업. 1979년 9월~1985년 5월 University of California, Berkeley 석·박사(구조역학). 1985년 6월 이후 현대중공업(주) 연구소 근무. 현재 동사 기계전 기연구소 소장 및 테크노 디자인 연구 소 소장, 상무.