

## 오프라인 프로그래밍에서의 실시간 통신

### Real-Time Communication in an Off-Line Programming

°송 종 탁\*, 손 권\*, 이 민 철\*\*

\*부산대학교 기계공학과 (Tel: 051-510-2308; Fax: 051-512-9835; E-mail: kson@hyowon.cc.pusan.ac.kr)

\*\*부산대학교 기계공학과 (Tel: 051-510-2439; Fax: 051-512-9835; E-mail: mclee@hyowon.cc.pusan.ac.kr)

**Abstracts** An off-line programming, OLP, system is widely used in automation lines. To help an on-line robot system to carry out desirable tasks planned by the off-line simulation, an approach to the real-time communication is presented. The OLP system developed consists of a software, a host computer(PC), a SCARA robot body, four servo drivers, and four independent joint controllers. This study focuses on the software where real-time communication is included. The software can be used in teaching, trajectory planning, real-time running, and performance evaluation. The evaluation of different control algorithms is one of the merits of the software. The software can give servo commands for task running. A comparison of generated and corresponding actual trajectories provides the evaluation of task performance. The safety of the OLP system is ensured by alarming malfunctions of the system. The OLP system developed can reduce the teaching time and increase the user's convenience.

**Keywords** Off-Line Programming, Real-Time Communication, Teaching, Trajectory Planning, Performance Evaluation

#### 1. 서론

로봇의 작업과 환경의 변화에 능동적으로 대처할 수 있는 오프라인 프로그래밍(OLP) 시스템이 생산 자동화 라인에서 널리 이용되고 있고, 그에 대한 연구가 활발히 진행되고 있다.[1][2] 컴퓨터통합생산(CIM)과 유연생산체계(FMS)를 성공적으로 구현하기 위해서는 호스트 컴퓨터와 현장 로봇간의 실시간 통신이 필수적이다. 이에 본 연구에서는 실시간 통신이 가능한 스카라 로봇용 OLP 시스템을 개발하여 오프라인 상에서 행해진 시뮬레이션 결과에 따라 온라인 상의 로봇 시스템을 구동할 수 있게 하였다. 개발된 OLP 시스템은 소프트웨어, 호스트 컴퓨터, 4관절 스카라 로봇, 4개의 서보 드라이버, 그리고 제어보드에 연결된 4개의 독립적인 관절 제어기로 구성되어 있다. 본 논문은 관절 제어기와 실시간 통신 기능을 갖춘 소프트웨어에 초점을 맞춘다. C언어를 이용하여 PC용으로 개발된 소프트웨어는 교시, 궤적 계획, 실시간 구동, 3차원 애니메이션, 그리고 성능 평가의 기능을 갖고 있다. 서로 다른 제어 알고리즘의 평가는 개발된 소프트웨어의 두드러진 특징 중 하나이다. 소프트웨어 상에서 특정 제어 알고리즘을 선택하여 다운로드할 수 있고, 서보 온-오프, 원점 복귀, 리셋과 같은 서보 동작 명령을 수행할 수 있다. 매 샘플링 타임마다 기준 궤적을 관절 제어기에 넘겨주고 그에 대응하는 통과 궤적을 되돌려 받음으로써 작업 성능의 평가가 가능하다. 서보 드라이버와 관절 제어기의 동작 상태를 검사하여 오동작이 발생할 경우 이를 알려줌으로써 온라인 시스템의 안전성을 보장할 수 있다. 결과적으로 개발된 OLP 시스템은 교시 시간을 줄이고 사용자의 편의성을 증가시키는 데 이용될 수 있다.

그림 1은 개발된 OLP 시스템의 전체 구성을 나타내고, 그림 2는 제어 대상인 4관절 SM5 삼성전자 스카라 로봇을 나타낸다. 제어 보드는 호스트 컴퓨터인 PC와 4개의 관절 제어기간의 통신을 매개하는 64kbyte의 공통 메모리, 통신 회로, 그리고 버스의 충돌 방지 회로를 포함한다. 각 관절 제어기는 32k×16bit의

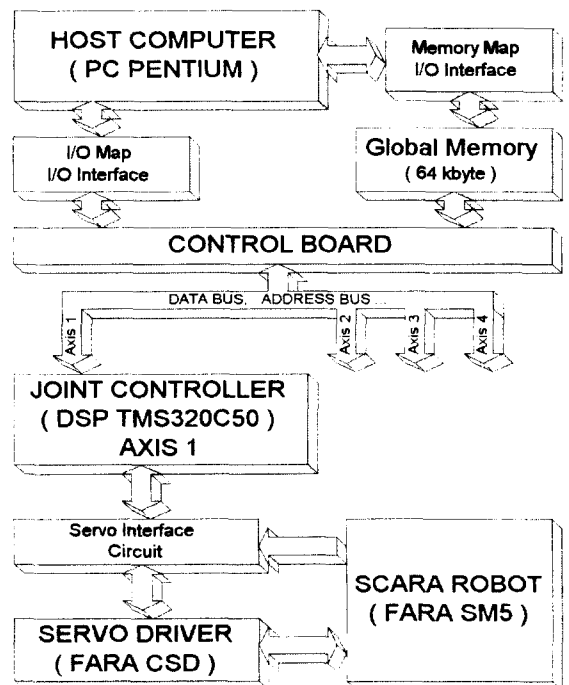


그림 1 OLP 시스템의 블록 선도

Fig. 1 Block diagram of the OLP system

#### 2. OLP 시스템의 구성

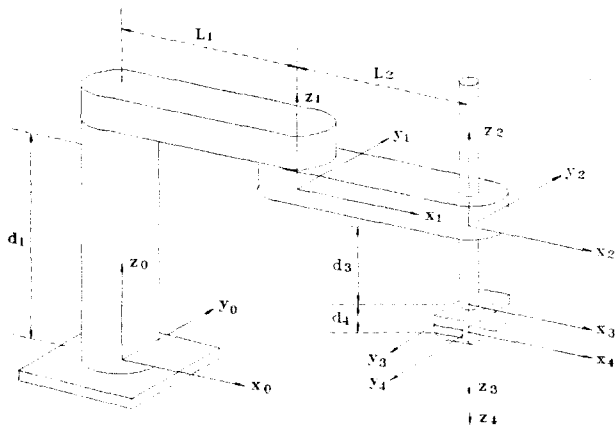


그림 2 스카라 로봇  
Fig. 2 SCARA robot

프로그램 메모리, 32k×16bit의 지역 메모리, 그리고 DSP TMS320C50을 포함한다. DSP는 각 관절 제어기의 메인 프로세서로 사용되며, 프로그램 메모리, 데이터 메모리인 공통 메모리와 지역 메모리, 그리고 서보 드라이버로의 접근이 가능하다. 관절 제어기와 호스트 컴퓨터간의 통신이 가능한 메모리 맵 I/O 인터페이스를 구성하여, 시스템 초기화 시에는 제어 프로그램을 공통 메모리로 보내고, 로봇의 운전 중에는 기준 궤적과 통과 궤적을 주고 받는다. 그림 3은 메모리 맵 I/O 인터페이스를 위하여 설정한 공통 메모리에 대한 PC측의 주소와 그에 대응하는 DSP측의 주소이다. 또한 I/O 맵 I/O 인터페이스를 구성하여 호스트 컴퓨터에서 직접 각 관절 제어기에 프로그램을 다운로드할 수 있게 하고, 관절 제어기의 초기화 동작을 제어할 수 있게 한다.[3]

PC		DSP
D000 0001	D000 0000	8000
D000 0003	D000 0002	8001
.	.	.
.	.	.
D000 FFFF	D000 FFFE	FFFF

그림 3 PC와 DSP의 공통 메모리 주소  
Fig. 3 Global memory address of PC and DSP

### 3. OLP 소프트웨어의 구조와 기능

그림 4는 개발된 OLP 소프트웨어의 전체 구조를 나타낸다. 본 OLP 소프트웨어는 시뮬레이션, 실시간 구동, 그리고 성능 평가의 세 부분으로 크게 나눌 수 있다. 시뮬레이션 부분에서는 여러 제어 알고리즘에 의한 동역학 및 제어 시뮬레이션, 교시, 그리고 궤적 계획이 수행된다. 실시간 구동 부분에서는 원하는 제어 알고리즘의 선택과 다운로드, 서보 동작 명령, 그리고 관절 제어기와의 실시간 통신이 수행된다. 마지막으로 성능 평가 부분에서는 기준 궤적 및 통과 궤적에 의한 제어 성능 평가와 3차원 애니메이션이 수행된다.

그림 5는 OLP 소프트웨어를 이용하여 로봇을 구동할 때 작업의 흐름도를 나타낸다. 먼저 동적 시뮬레이션, 궤적 계획을

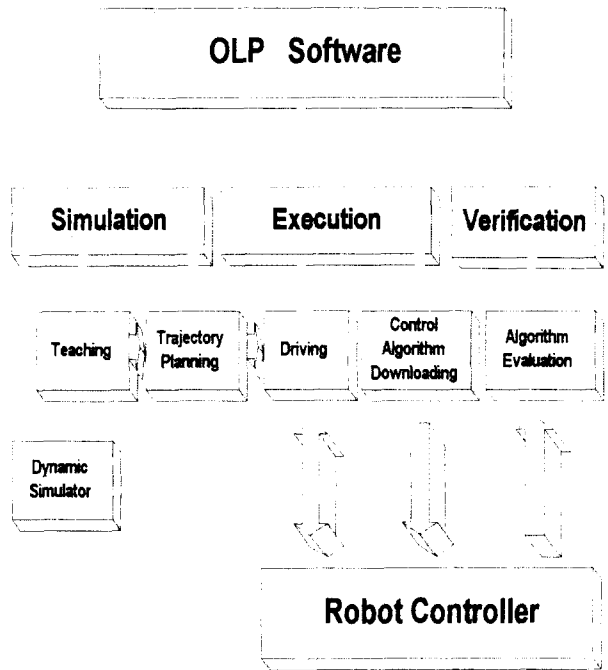


그림 4 OLP 소프트웨어의 블록 선도  
Fig. 4 Block diagram of the OLP software

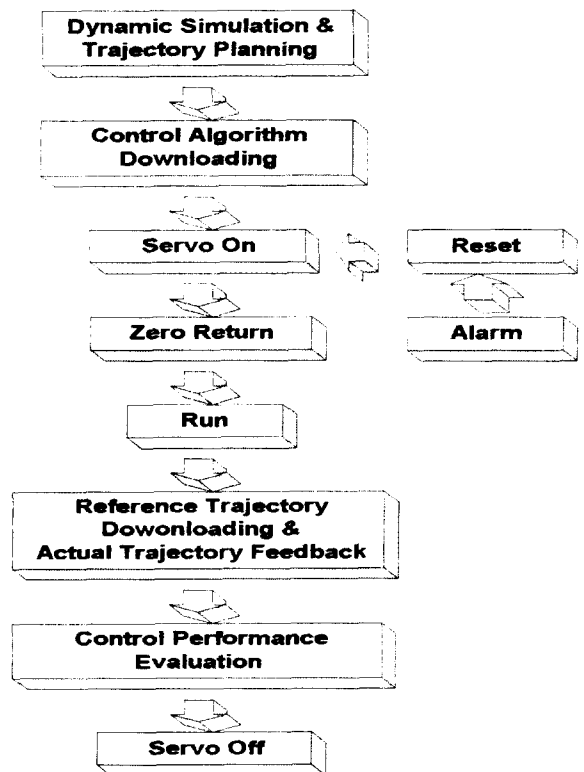


그림 5 소프트웨어에서의 작업의 흐름도  
Fig. 5 Flow chart of the software

수행하고, 원하는 제어 알고리즘을 다운로드한 다음, 서보-온, 원점 복귀의 서보 동작 명령을 수행하여 로봇의 실시간 구동을 위한 준비를 마친다. 실시간 구동에 의해 기준 궤적을 넘겨주고 통과 궤적을 되돌려 받아서 성능 평가를 행한 다음 서보-오프를 수행한다. 그리고, 로봇의 실시간 구동 중 OLP 시스템 내에 오

동작이 발생할 경우 이를 알려주고 리셋을 행한 후 일련의 동작을 다시 수행한다.

### 3.1 시뮬레이션 기능

그림 6은 OLP 소프트웨어의 로고 화면이다. 시뮬레이션 기능에 의해 교시, 궤적 계획, 동적 시뮬레이션이 가능하다.

교시는 관절 좌표계나 절대 좌표계에서의 로봇 경유점, 통과 시간, 경유점 사이의 궤적 계획법을 입력받음으로써 수행된다. 궤적 계획법으로는 3차 스플라인법, LFPB법, b-스플라인법, 직선 보간법, 원형 보간법이 준비되어 있다. 기존의 교시 상자를 통한 교시와는 달리 로봇의 자세를 컴퓨터 화면에서 확인하면서 교시할 수 있는 장점이 있고, 교시된 경유점이 작업 공간을 벗어날 경우 이를 미리 알려주므로 온라인 시스템의 안전성을 보장할 수 있다. 그림 7은 구간별로 다양한 궤적 계획법의 입력이 가능한 교시 화면이다.

그리고 궤적 계획에 이은 동적 시뮬레이션에 의해 수행하고자 하는 작업을 컴퓨터 화면 상에서 미리 볼 수 있으므로 로봇에 적용될 제어 알고리즘과 교시의 적합성을 판단할 수 있다.

### 3.2 로봇 구동 기능

제어 알고리즘의 다운로드, 기준 궤적의 선택, 서보 동작 명령, 온라인 작업 수행이 가능하다. 서보 동작 명령에는 서보 온-오프, 원점 복귀, 리셋이 포함된다. 그림 8은 서보-온을 수행

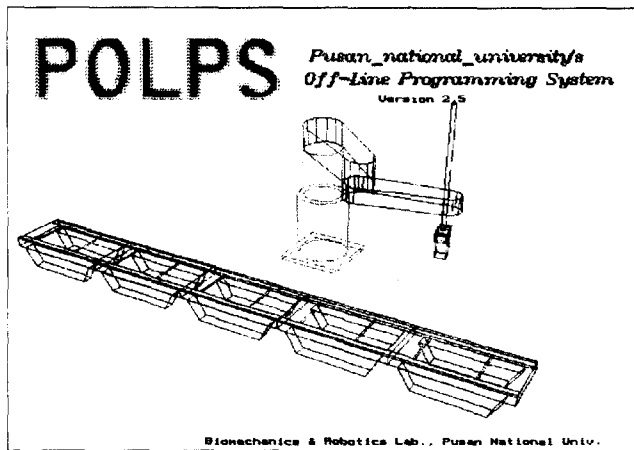


그림 6 OLP 소프트웨어의 로고 화면  
Fig. 6 Logo screen of the OLP

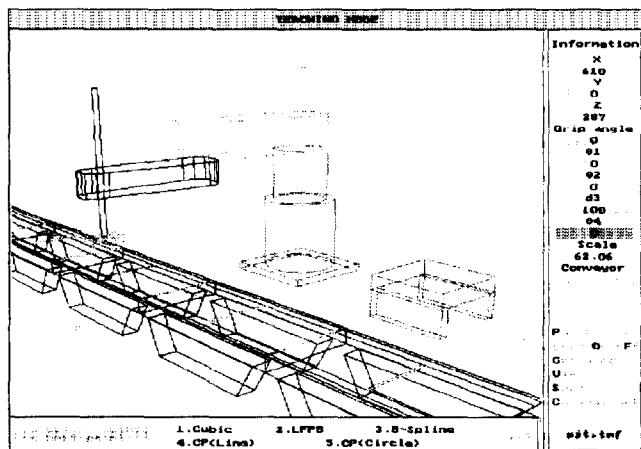


그림 7 교시 화면  
Fig. 7 Screen of teaching

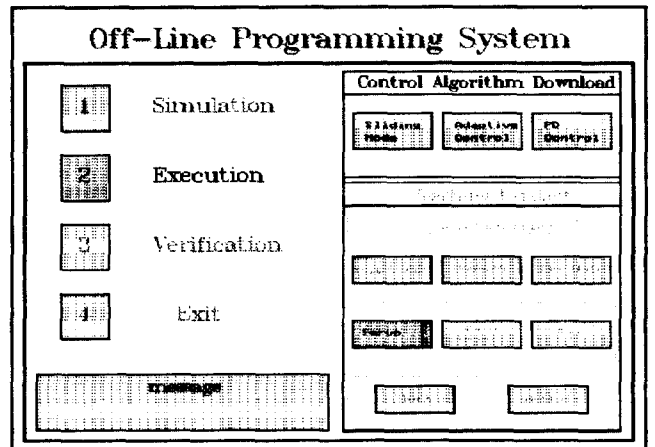


그림 8 서보 동작 명령 화면  
Fig. 8 Screen of servo commands

했을 때의 화면이다.

제어 알고리즘 다운로드를 행하면 메모리 맵 I/O 인터페이스 및 I/O 맵 I/O 인터페이스에 의하여 선택한 제어 알고리즘이 각 관절 제어기의 프로그램 메모리로 다운로드된다.

기준 궤적 선택을 행하면 온라인 작업 수행시 관절 제어기에 넘겨줄 기준 궤적이 지정된다. 이 때 각도 단위의 기준 궤적을 엔코더 펄스 단위의 중분치로 바꿈으로써 통신 중 발생하는 에러의 가능성을 줄인다. 온라인 작업 수행시 관절 제어기의 DSP는 16 msec 단위의 기준 궤적을 1 msec 단위로 보간하여 제어에 이용한다.

서보-온을 행하면 메모리 맵 I/O 인터페이스와 서보 인터페이스 회로를 이용하여 서보 드라이버가 모터를 구동할 수 있는 상태로 만들고, 반대로 서보-오프를 행하면 서보 드라이버에서 모터로의 전류를 차단한다. 서보-온에 이어 원점 복귀를 행하면 각 축 모터의 위치를 초기화하는 알고리즘이 수행된다. 온라인 작업 도중 서보 드라이버나 관절 제어기에 오동작이 발생할 경우 이를 알려주고, 리셋을 행하면 발생한 에러를 해제시킨다.

온라인 작업 수행을 행하면 매 샘플링 타임마다 기준 궤적을 호스트 컴퓨터에서 공통 메모리를 통하여 관절 제어기로 넘겨주고, 엔코더에 의해 읽어들이는 통과 궤적을 돌려 받는다.

### 3.3 성능 평가 기능

제어 성능의 평가와 3차원 애니메이션이 가능하다. 제어 성

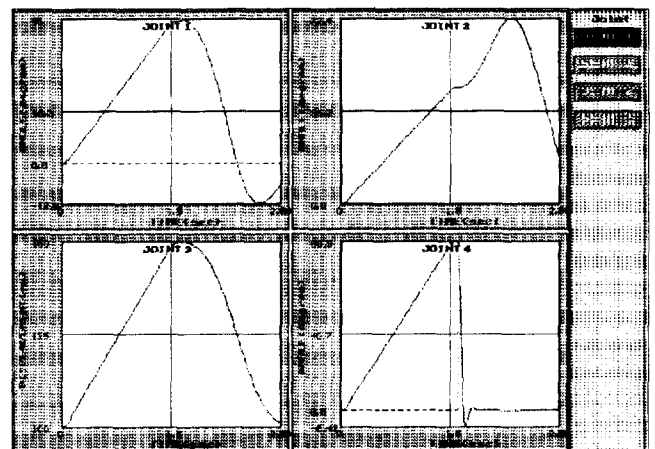


그림 9 성능 평가 화면  
Fig. 9 Screen of performance evaluation

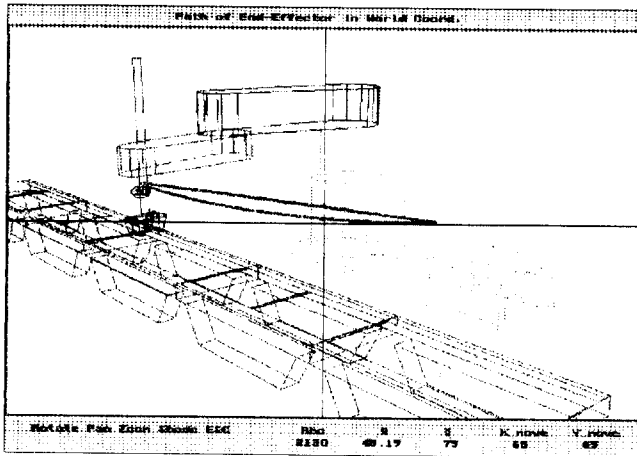


그림 10 3차원 애니메이션 화면  
 Fig. 10 Screen of 3-D animation

능의 평가는 시뮬레이션에 의한 기준 궤적과 관절 제어기로부터 넘겨받은 통과 궤적을 이용하여 관절 좌표계 또는 절대 좌표계에서 행해진다. 그리고, 3차원 그래픽을 이용하여 수행된 로봇의 작업을 시각적으로 재확인할 수 있다. 그림 9는 관절 좌표계에서의 성능 평가 화면이고, 그림 10은 수행된 작업에 대한 3차원 애니메이션 화면이다.

#### 4. 결론

빠른 시장 변화로 인하여 야기되는 로봇의 작업 및 환경의 변화에 능동적으로 대처할 수 있는 스카라 로봇용 OLP 시스템과 본 시스템에 적합한 소프트웨어를 개발하였다. 메모리 맵 I/O 인터페이스와 I/O 맵 I/O 인터페이스를 이용하여 소프트웨어가 수행되는 호스트 컴퓨터와 관절 제어기간의 실시간 통신을 가능하게 하였다.

개발된 소프트웨어를 사용하면 수행하고자 하는 온라인 작업을 오프라인 상에서 미리 행할 수 있고 호스트 컴퓨터에서 현장 로봇의 조작 및 상태 확인이 가능하다. 따라서 사용자의 편의성을 증가시키고, 시스템의 안전성을 높일 수 있다. 또한 기존 산업체에서는 대부분 하나의 알고리즘만으로 제어를 수행하는 데 반해 개발된 소프트웨어를 사용할 경우 여러 제어 알고리즘의 성능 비교 및 평가가 가능하다.

#### 참고문헌

- [1] N. Kircanski, M. Vukobratovic, B. Karan, M. Kircanski, and A. Timcenko, "Multiprocessor Control System for Industrial Robots", *Robotics & Computer-Integrated Manufacturing*, Vol. 8, pp. 77-86, 1991.
- [2] 정치연, 이기동, 이범희, "실시간 운영체제를 이용한 로봇 제어기 소프트웨어 디자인 및 구현", '94한국자동제어학술회의논문집(II), pp. 1247-1251, 1994.
- [3] 조순복, *제어 프로그래밍 ROM화 기법*, 기한재, 서울, 1993.