

그리퍼 정밀 제어를 위한 이중 제어기 시스템의 구현 및 성능 평가

Implementation and Performance Evaluation of the Dual Controller System for Precision Control of Gripper

이 승 용¹ · 함 윤 형² · 박 영 우² · 정 일 균² · 임 선[†]

Seung-Yong Lee¹, Un-Hyong Ham², Young-Woo Park², Il-Kyun Jung², Sun Lim[†]

Abstract: This paper proposes a Dual Controller System for Precision Control (DCSPC) for control of the gripper. The DCSPC consists of two subsystems, CDSP (Controller based DSP) and CARM (Controller based ARM processor). The CDSP is developed on a DSP processor and controls the gripping motor and LVDT. In particular, the CARM is implemented using Linux and ARM processor according to recent research related to open-source. The robot for high-precision assembly is divided into the robot control and the gripper control section and controls CARM and CDSP systems respectively. In this paper, we also proposed and measured the performance of communication API. As a result, it is expected to recognize improvements in communication between CARM and the robot controller, and will continue to conduct relevant research among other commercial robot controllers.

Keywords: Open-source Control System, Dual Controller System, Real-time OS, Gripper

1. 서 론

근래에 로봇 기술의 개발과 더불어 다양한 분야에서 로봇에 대한 수요가 증가하고 있다. 특히 고령화 및 출산율 감소로 인한 노동력 부족 문제를 해결하기 위하여 산업 전반에서 로봇을 활용한 자동화가 꾸준히 진행되고 있다. 대표적으로 자동차산업의 경우 이송, 용접, 도장 등 대부분의 공정에서 로봇 도입을 통한 자동화를 구현함으로써 생산성을 극대화하고 있다. 최근에는 산업의 변화로 인해 제조로봇의 수요가 자동차 시장에서 각종 IT제품 등의 전기전자시장으로 확장되고 있으며, 기능적으로는 기존의 단순작업에서 보다 복잡한 작업 능력이 요구되는 조립 작업으로 변화하고 있다^[1].

로봇 기술의 발전에 따라 산업 전반에서 로봇 매뉴플레이

터를 적용한 제조공정의 활용성이 점점 높아지고 있다. 특히 모바일 IT 제품의 일체형 및 직접화가 고도화 됨에 따라 조립 공정에 고정밀 툴의 필요성이 제기되고 있으며, 이를 위해 다양한 로봇 그리퍼 모듈이나 그와 관련된 툴이 개발되고 있다^[1,2]. 특히, 모바일 IT 조립 공정에 특화되고 고난이도 조립 작업으로 구성된 생산라인의 변동에 신속한 대응이 가능해야 한다. 그러나 현재에는 다수의 부품 조립에 대한 솔루션을 제공할 수 있는 로봇 시스템이 요구되지만 단순 로봇만 갖고는 제한적으로만 운용 가능한 상태이다. 본 논문에서는 이런 단점을 최소화하며 유연성을 높이기 위해 전동 그리핑 모듈과 강성모듈을 통합하여 임베디드 시스템까지 통합된 그리퍼 시스템 모듈화 설계 및 그 제어에 관한 것을 제시한다.

본 논문에서는 2장에서는 본 연구의 배경을 서술하며, 3장에서는 그리핑 모듈 시스템에 대한 전반적인 소개를 4장은 성능 검증을 위한 실험과 결과를 분석하며, 마지막으로 5장에서는 결론과 향후 연구계획에 대해서 논한다.

2. 연구 배경

협소공간을 갖는 모바일 IT제품의 다수 부품 조립에 유연

Received : Nov. 11. 2017; Revised : Jan. 2. 2018; Accepted : Jan. 3. 2018

※ This project was funded by Korea Electronics Technology Institute (KETI) and is currently supported by the publication grant. This research was supported by the Ministry of Trade, Industry & Energy and the Korea Evaluation Institute of Industrial Technology (KEIT) with the program number of "10060110".

1. Computer Engineering, KETI, Seoul, Korea (michalen2006@keti.re.kr)

2. Computer Engineering, KETI, Seoul, Korea (mickey3d@keti.re.kr)

† Corresponding author: Electricity and Electronic Engineering, KETI, Pyeongcheon-ro, Bucheon, Korea (sunishot@keti.re.kr)

하게 대응하기 위해 필요한 고정밀 조립 작업을 수행하는 로봇의 제어운용 기술은 크게 로봇 시스템을 능동적으로 조립 반력을 제어하는 힘제어기반 로봇제어 방식과 로봇의 말단에 수동적인 추가 컴플라이언스를 부착하여 조립하는 방식으로 구분할 수 있다. 첫 번째로 힘제어 기반의 조립의 경우, 실시간으로 조립 상황을 파악하여 로봇을 제어함으로써 정밀 부품에 대한 조립이 가능한 반면 시스템 구성에 높은 비용(힘제어 기능을 갖춘 로봇 시스템 필요)이 소요되며 로봇과 환경의 접촉 작업 시에 로봇 시스템의 발산과 같은 예기치 않은 위험한 상황이 초래될 가능성이 있다. 그와 별개로 말단에 컴플라이언스를 부착하여 로봇 전체의 움직임을 제어하는 방식은 필요 작업에 비해 과도한 관성이 발생하여 작업 속도가 느리고 이에 따라 제품 생산성이 저하되므로, 현재 제조업에는 거의 사용하지 않고 있다.

최근에는 기존의 그리퍼 제어 기술이 가지는 한계를 극복하기 위해 수동 강성모듈과 자세 변환을 측정할 수 있는 센싱 모듈을 포함한 그리퍼 모듈이 개발되고 있으며 이를 실제 말단에 탑재하여 로봇 제어기와 연동한 동작을 통해 고정밀 작업을 수행이 가능함을 확인하였다. 이를 위해서는 외부 로봇제어기와 시리얼 또는 이더넷 인터페이스를 통해 축단위의 정보, 직교좌표계의 정보, 스크립트 기반의 잡 파일 등 다양한 지령 및 정보를 교환해야 하며 이를 위해 외부 로봇 제어기 인터페이스를 포함하고 있다^[4-6]. 그리고 자세 변환을 측정하기 위해 6채널의 LVDT (Linear Variable Differential Transformer)가 이용되며 고속 고분해능을 갖는 ADC를 통해 실시간으로 자세 변환을 출력할 수 있어야 한다

오픈 시스템의 정의를 다양하게 말할 수 있지만 다른 시스템과의 재사용이 용이하고 전체 시스템이 유연하게 개발이 가능하며 시스템에 의존하지 않고 개발할 수 있는 시스템으로 정의할 수 있다. 최근의 오픈 소스의 모션 제어 시스템은 범용 하드웨어와 소프트웨어를 이용하여 개발하고 연구되고 있다. 그 중 x86, ARM 프로세서와 범용 리눅스 등 기반의 모션 제어 시스템 개발이 활발히 진행되고 있으며 Xenomai, RTAI, Linux-RT 등이 여러 프레임워크가 제안되고 성능평가가 진행되어 왔다^[7-11].

3. 이중 제어기 시스템

그리퍼 정밀 제어를 위한 이중 제어기 시스템(Dual Controller System for Precision Control, DCSPC)은 최근 개발 연구 흐름에 맞춰 오픈소스를 이용하여 DSP 및 ARM 프로세서 기반의 제어 시스템을 개발하였다^[13]. DSP 제어기(Controller based DSP, CDSP)는 그리핑과 관련된 모터, LVDT 그리고 ARM 프로세

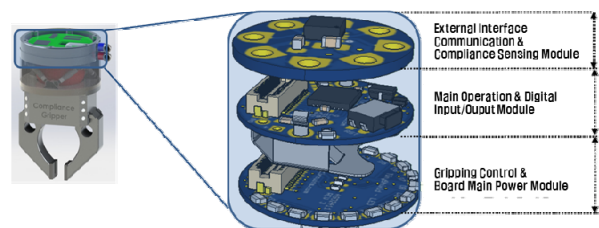
서와의 통신 등을 담당하게 된다. ARM 프로세서 기반의 제어기(Controller based ARM processor, CARM)는 정밀 부품 조립을 위한 조립 알고리즘과 외부 카메라와의 통신, 그리고 로봇 제어기와의 제어 통신 등의 인터페이스를 담당한다.

3.1 CDSP 하드웨어 설계

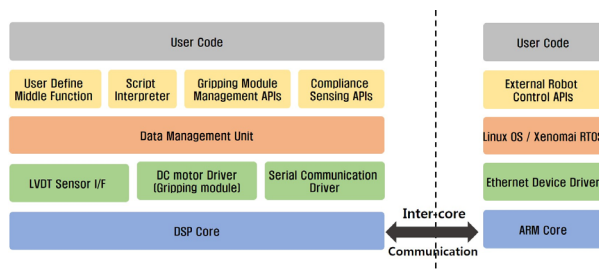
DCSPC 시스템은 그리핑 제어를 위한 CDSP와 조립 알고리즘, 로봇 제어기 통신 등을 담당하는 CARM으로 구분할 수 있다. CARM은 범용 하드웨어 및 소프트웨어를 기반으로 개발되기 때문에 임베디드 하드웨어 개발 필요성이 없다. 그러나 CDSP인 경우, 그리핑 모터를 위한 모듈, 그리핑 자세를 제어하기 위한 컴플라이언스 모듈 등을 위한 하드웨어 개발이 필요하며 실제 개발된 하드웨어는 [Fig. 1]과 같다. DCSPC의 하드웨어는 [Fig. 1]과 같이 총 3개의 층으로 구성된다. 첫 번째 가장 하단에는 그리핑 모터 제어와 관련된 하드웨어와 전원 모듈이 이식되었다. 중간 계층에는 DSP 프로세서와 그와 관련된 여러 하드웨어가 이식되었으며 부가적으로 디지털 I/O와 CARM과의 통신 모듈이 함께 개발되었다. 마지막으로 가장 상단에는 LVDT의 신호를 디지털로 변환하는 ADC 회로와 이후에 CARM 임베디드 보드가 장착되기 위한 주변 회로 등이 구성되어 있다.

3.2 CDSP 및 CARM 소프트웨어 설계

[Fig. 2]는 DCSPC 시스템의 이중으로 구성된 소프트웨어 구조도를 나타낸다. DCSPC는 모터 제어, 조립 알고리즘, 로봇



[Fig. 1] Hardware architecture of DCSPC



[Fig. 2] Software architecture of DCSPC

제어기 통신 등 정밀한 실시간 성능이 요하는 시스템이다. 이를 위해 CDSP는 DSP의 하드웨어 타이머 인터럽트를 이용하여 시스템 태스크가 실행되고, 하나의 태스크만으로 시스템 제어가 되므로 정밀한 실시간 성능을 만족하면서 동작되도록 설계하였다. 그러나 CARM 시스템은 범용 Linux를 이용하여 개발하였기 때문에 별도의 실시간 운영체제(Real-Time Operating System, RTOS) 이식이 필요하며 이는 아래에서 자세히 설명한다.

[Fig. 2]의 좌측 소프트웨어 구조도는 CDSP의 전반적인 소프트웨어 구조도를 나타낸다. CDSP는 3가지의 디바이스 드라이버를 필요로 하는데 첫 번째로 그리핑 자세를 위한 LVDT 센서 인터페이스이다. LVDT 센서 모듈은 DSP에 설정된 주기마다 호출되어 현재 LVDT 아날로그 값을 수신한다. 이 값은 외부 ADC칩과 DSP간에 SPI 통신을 이용하여 18bits 값을 수신한다. 두 번째로 외부로부터 인가되는 명령어를 입력 받기 위한 시리얼 디바이스 드라이버이다. DSP 프로세서는 모터 제어, LVDT 송신, LED 제어 등의 명령어가 수신되면 즉시 최우선인터럽트 핸들러가 호출되고 관련 명령어를 처리하게 된다.

마지막으로 그리핑 메커니즘을 구동과 관련된 DC 아날로그 모터를 제어하기 위한 디바이스 드라이버이다. 모터 제어 알고리즘은 모터 위치 명령어가 입력되면 PID와 같은 선형제어기를 통해 해당 모터 드라이버의 입력인 PWM 신호를 인가하여 모터를 구동하게 된다. 예를 들어, [Fig. 3]은 위의 3가지 주요 기능이 1 ms 주기에서 실행되는 CDSP의 실행 예를 나타낸다. DSP는 매 주기마다 현재 모터의 엔코더 값을 읽어온 후 PI제어를 통해 실시간으로 모터 제어 태스크를 처리한다. 만약 CARM에서 LVDT 읽기 요청, 모터 제어 등의 명령어가 수신되면 즉시 인터럽트를 이용하여 처리하며 그 외 시간에는 LVDT 센서에서 값을 업데이트 한다.

시스템은 기본적으로 Linux 범용 OS 기반으로 설계되었다. 범용 Linux는 Best-effort 방식의 태스크 스케줄링으로 처리되므로 실시간 성능을 만족시킬 수 없다. 그러나 CARM 시스템은 로봇 제어기와 통신, 조립 알고리즘, CDSP와의 통신 등 실시간 성능이 매우 중요한 응용프로그램들이 동작된다. 따라서 CARM의 실시간 성능 보장을 위해 Xenomai RTOS^[12] 소프트웨어를 이용하여 시스템에 이식하였다. Xenomai RTOS는 오픈 소스를 이용하여 개발된 범용 리눅스 기반의 실시간 운

영체제로서 기존 Linux에서 가지는 실시간 성능의 한계를 Adeos 마이크로 커널을 이용하여 보장되도록 개발되었다. 범용 Linux인 경우, Interrupt에 대한 우선순위 보장이 없으며, 태스크 또는 프로세스에 대한 우선순위가 없기 때문에 정주기 태스크 실행을 보장 받지 못한다. 이에 반해 Xenomai 태스크는 태스크 설정 시, 우선순위, 주기 등에 대한 정보 설정이 가능하며 이를 위해 실시간 스케줄링 또한 지원을 하고 있다.

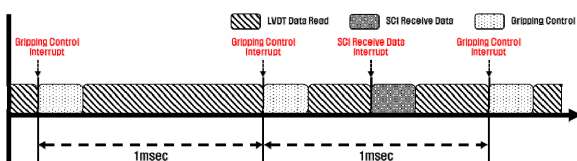
3.3 외부 통신 인터페이스 설계

CDSP는 CARM 시스템과 SCI (Serial Communications Interface) 이용하여 통신을 수행하며 기본적으로 슬레이브의 역할로 개발되었다. 3.2절에서 제시한 것과 같이 CDSP는 CARM 시스템에서 LVDT 읽기, 모터 제어, I/O 모듈 제어 등의 외부로부터의 요청이 수신될 수 있다. 또한 [Fig. 2]에서 제시한 것과 CARM은 외부 로봇 제어기와 통신 인터페이스를 제공한다. 전통적인 그리퍼 제어 시스템은 로봇 축 가장 말단에 부착되어 제어되는 방식으로 동작되었다. 그러나 본 논문에서는 조립 정밀도를 높이고 실시간으로 그리퍼 제어를 하기 위해 외부 로봇 제어기와 통신하기 위한 명령어를 개발하였다. [Table 1]은 현재 CDSP, CARM에서 제공되는 모든 API를 제시한다.

CARM 시스템은 외부 로봇 제어기 및 CDSP의 DSP 프로세

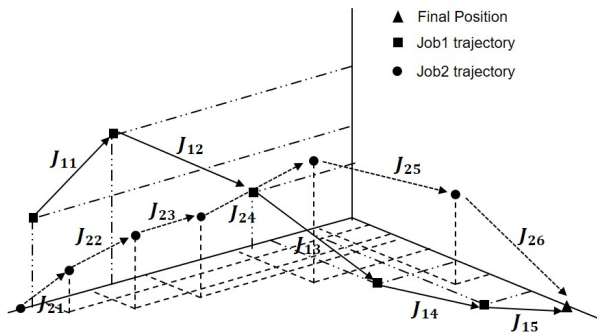
[Table 1] API of CDSP and CARM for external communication

Function Name	Description
RC_Initialize	Function for communication with robot controller (Robostar N1 or Denso VS-060 or Universal Robot UR3)
RC_GetCurrJ	Return actual joint value of robot
RC_GetCurrC	Return X, Y, Z, Rx, Ry and Rz of robot
RC_MoveJ	Move to target position (joint value)
RC_MoveC	Move to target position (X, Y, Z, Rx, Ry and Rz)
RC_SetJobPointJ	Set joint value of nIdx Job Point
RC_SetJobPointC	Set X, Y, Z, Rx, Ry and Rz of nIdx Job Point
RC_GetJobPointJ	Return joint value of nIdx Job Point
RC_GetJobPointC	Return X, Y, Z, Rx, Ry and Rz of nIdx Job Point
RC_GetInteger	Return integer value of nIdx
RC_SetInteger	Set integer value of nIdx
LVDT_Function	Return 6-channel LVDT of compliance module
DIO_Function	Request command of LED, button or etc.
MC_Function	Request motor command (Move, Stop, PowerOn)

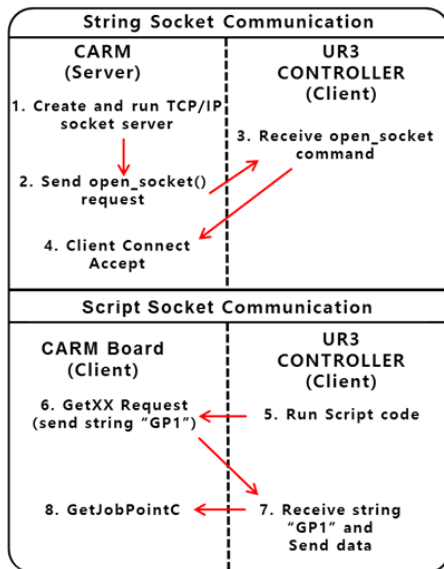


[Fig. 3] Execution sequence of CDSP

서 사이에 물리적인 인터페이스로 연결되어 통신을 수행한다. 그 중 외부 로봇 제어기와는 로봇의 현재 위치 읽기 명령어(RC_GetCurrX), 로봇 이동 명령어(RC_MoveX), Job Point 읽기(RC_GetJobPointX) 및 쓰기(RC_SetJobPointX) 명령어와 정수 변수 읽기 명령어(RC_GetInteger)를 통해 통신한다. 이때 X 인덱스는 로봇 관절 단위(Joint, J)와 직각 좌표계 단위(Cartesian, C)로 구분된다. 여기서 Job의 개념은 로봇의 공정을 하나의 집합을 의미하며 한 개의 Job은 복수의 JobPoint(이하 JP)로 구성된다. 이때 로봇 관절 단위 일 때 JobPoint 요소는 $JP_{jp} = [J_1, J_2, \dots, J_6]$, 직각좌표일 때 JobPoint 요소는 $JP_{jp} = [x, y, z, \theta_x, \theta_y, \theta_z]$ 의 형태로 최종적으로 정의된다. Job에 대한 예는 [Fig. 4]와 같다. 여기서 JP_{jp} 의 아래 첨자 jp는 Job번호 j와 JobPoint 번호 p로 정의될 수 있다. 즉 하나의 경로 j에 대해 임의의 개수만큼 JobPoint를 구성하여 p로 구성되며 이에 대한 정보는 로봇 제어기에 저장되어 관리된다.



[Fig. 4] Example of two job trajectory

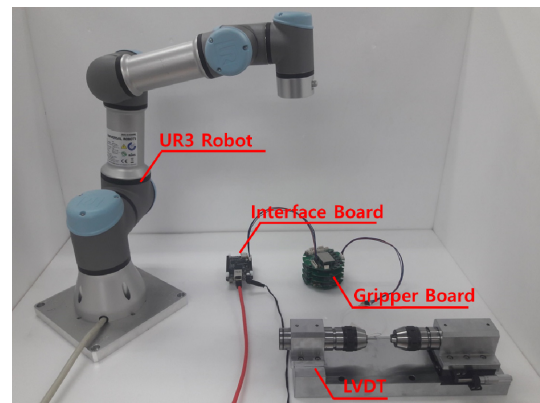


[Fig. 5] Socket communication of w.r.t. robot in UR (ex. RC_GetJobPointC)

본 논문에서는 DCSPC 시스템의 유효성 검증을 위해 Denso, UR, Robostar 등의 로봇 제어기를 채택하였다. 그러나 로봇 제어기와 CARM 프로세서 사이에 공유 변수가 정의되고 사용될 수 있어야 하는데 UNIVERSAL ROBOTS社は 이에 대한 기능을 현재 제공하지 않고 있다. 따라서 본 논문에서는 이를 지원하기 위해 UR 로봇 제어기에 별도의 TCP/IP 소켓 통신을 할 수 있는 UR 스크립트 소켓 프로그램을 개발하였으며 이에 대한 내용은 [Fig. 5]와 같다. 스크립트 소켓 프로그램은 시스템 초기에 UR 제어기에서 실행되고 필요한 전역변수를 선언한다. 그 이후 지속적으로 CARM의 전역변수 읽기 및 쓰기 명령어 수신 여부를 검사하며 해당 시점에 즉시 처리하게 된다. 위의 추가 기능으로 2개의 제조사와 다르게 TCP/IP 소켓 통신의 이중 구조로 인한 시스템 오버헤드가 가장 취약함을 알 수 있어 본 논문에서는 UR 로봇 제어기를 채택하여 실험 및 검증을 진행하였다.

4. 실험환경 및 성능평가

본 논문에서 제안하는 DCSPC의 성능 평가를 위해 [Fig. 6]과 같이 실험 환경을 구성하였다. CARM의 하드웨어인 Interface Board는 TI社의 AM335x ARM 프로세서로 개발된 Beagle Bone Black 상용 임베디드 보드를 사용하였다. 위 임베디드 보드에 Linux 3.10.37와 Xenomai 2.6.3 RTOS 패치하였다. Gripper Board는 TI社의 TMS320F-28335 DSP 프로세서와 34k*16 SARAM, 256*16 flash 메모리를 이용하여 자체 제작하였다. Interface Board와 통신하는 상용 로봇은 UNIVERSAL ROBOTS의 UR3 상용 로봇을 사용하였다. 4.1절에서는 CARM의 실시간 정주기성 실험을 진행하여 RTOS 성능 검증을 확인하였다. 4.2 및 4.3절에서는 DCSPC의 주요 통신에 소요되는 오버헤드를 측정하였으며 4.4절에서는 3개의 제어기가 가상의 작업을 수행될 때 소요되는 시스템 오버헤드를 측정하여 본 실험을 마친다.



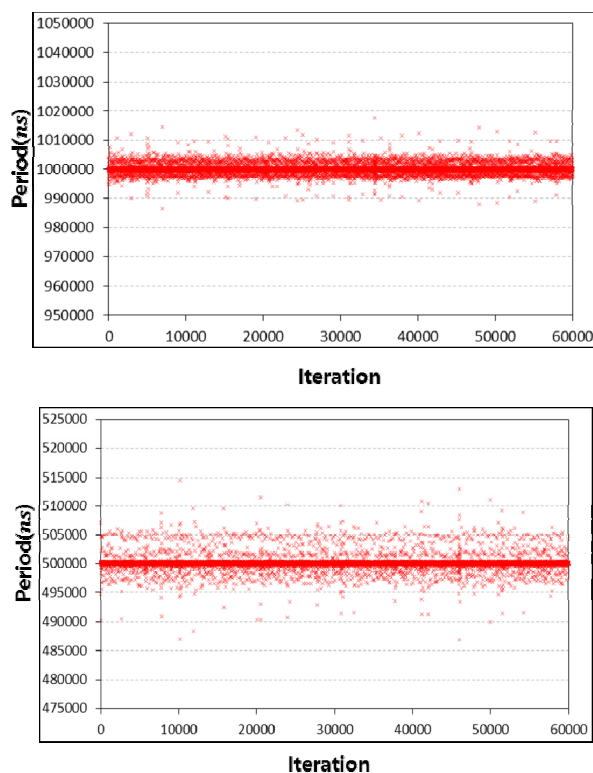
[Fig. 6] Experiment environment

4.1 CARM 실시간 성능 평가

본 논문에서 제안하는 DCSPC의 CARM은 상용 임베디드 보드를 이용하여 개발하였기 때문에 실시간 검증이 매우 중요하다. CARM은 CDSP와 외부 로봇 제어기와 통신을 처리하기 때문에 정주기성이 엄격하게 지켜져야 하며 따라서 본 절에서는 Beagle bone black의 정주기성 성능 평가를 진행하였다. 실험을 위해 Xenomai Task 주기는 500 μ s와 1 ms로 설정하였으며 총 60,000개의 샘플링 데이터를 취득하였고 그 결과는 [Fig. 7]과 같다. 그 결과, 주기가 1 ms에는 999999.92(ns) 평균 주기와 765.64(ns)의 편차를 보였으며 500 μ s일 때 499999.91(ns) 평균 주기와 편차 560.71(ns)가 측정되었으며 매우 정밀한 정주기성을 확인할 수 있었다.

4.2 CDSP 및 CARM 통신 시간 성능 평가

[Table 1]에서 제시한 API에 대한 통신 시간 측정은 시스템 제어 주기를 결정하기 위한 실험이다. 본 절에서는 CDSP 및 CDSP 통신 간에 사용되는 LVDT_Function, DIO_Function, 그리고 MC_Function 실험을 진행하였다. 두 시스템간의 SCI 통신 속도는 1.875 Mbps이며 3분 시간 간격으로 1회에 3개의 데

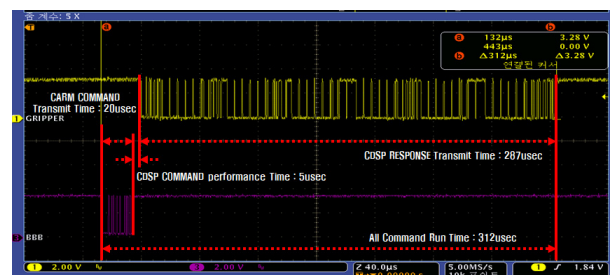


[Fig. 7] Experiment result of task period (1 ms period (up) / 500 μ s period (down))

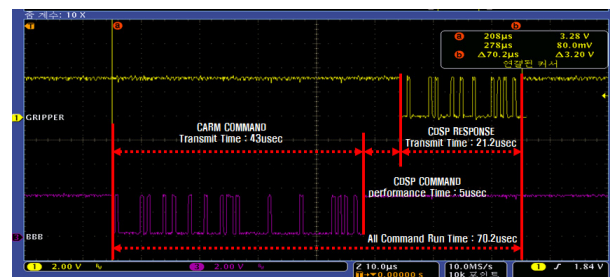
이터를 취득, 총 30개의 데이터를 취득하였다. 첫 번째로 LVDT_Function 명령어는 CARM에서 4 byte 요청 메시지를 송신하고 CDSP는 즉시 52 byte의 응답 메시지를 송신한다. 위와 동일한 방법으로 DIO_Function은 5 byte 요청 메시지와 4 byte 응답 메시지, 그리고 MC_Function은 20 byte의 요청 메시지와 4 byte의 응답 메시지를 서로 교환한다. 실험 결과, 3개의 API는 순차적으로 평균 312.66 μ s, 53.75 μ s, 137.21 μ s 통신 시간이 소요되었으며 [Fig. 8]과 같다. 전체 통신 시간은 요청 응답 메시지의 크기와 통신 속도에 의존적임을 알 수 있다. 실제로 3개의 API를 처리하기 위한 DSP 처리 시간은 평균 4.44 μ s, 5.71 μ s와 2.47 μ s 시간만이 소요되어 전체 실행시간에서 큰 비중을 차지하지 않음을 확인하였다.

4.3 CARM 및 외부 로봇 제어기 통신 시간 성능 평가

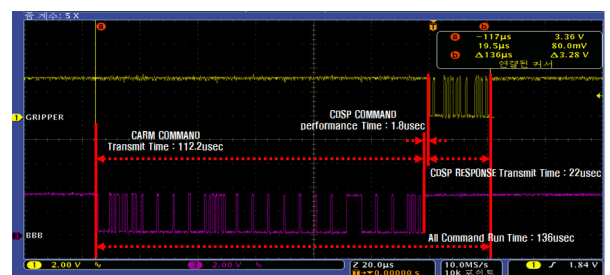
3.3절 [Table 1]의 API에서 CARM과 외부 로봇 제어기의 통신에 소요되는 시간을 측정하였다. 외부 로봇 제어기와 통신



(a) LVDT_Function

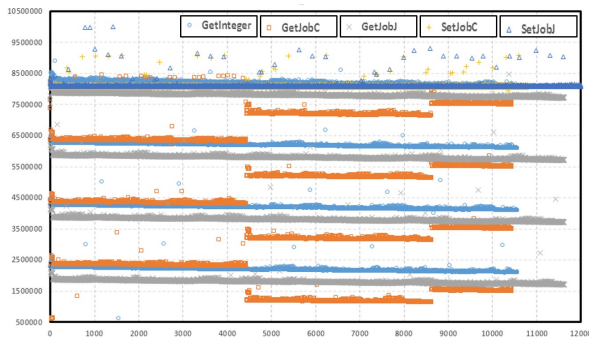


(b) DIO_Function



(c) MC_Function

[Fig. 8] Communication time of CDSP and CARM



[Fig. 9] Communication time of CARM and external robot controller (unit: nanosecond)

하는 명령어 중 이중 TCP/IP 소켓 통신을 필요로 하는 5개의 함수의 오버헤드가 더 높기 때문에 5개의 함수 실험 결과를 [Fig. 9]에 나타내었다. 현재 위치 읽기 및 로봇 구동 명령어 실험은 약 10,000개의 데이터 취득 결과 99% 이상 1 ms미만의 통신 시간이 측정되었다. 그러나 5개의 API인 경우 평균 6.15 ms와 2.44 ms의 편차를 보였다. 이는 UR3 로봇인 경우, 이중 TCP/IP 구조가 많은 시스템 오버헤드를 증가시킬 수 있었으며 추후 시스템 개선의 여지가 있음을 확인하였다.

4.4 DCSPC 시스템 오버헤드 성능 평가

본 절에서는 가상의 조립 작업을 수행하는 태스크를 수행되었을 때 DCSPC의 시스템 오버헤드를 측정하였다. 로봇은 임의의 두 지점을 왕복 운동하는 모션을 수행하는데 이때 이동 지점, 모션 시작 및 종료 명령은 CARM에 의해 요청되어 수행된다. 임의의 두 지점에서 CDSP는 가상의 제품을 조립하기 위해 특정 시간 동안 모터 구동을 수행 후 종료하게 된다. 위의 전체 동작을 수행하는 동안 CARM은 CDSP에 주기적으로 LVDT 값을 요청하여 지속적으로 현재 DCSPC의 위치를 모니터링을 수행한다. 위의 가상 조립 작업을 30회 수행하였으며 4회를 걸쳐 총 120회의 작업을 수행하였다. 실험결과, 각각의 작업 수행 시간은 1 ms이하의 적은 오차로 약 76초가 측정되었으며, 동일 시간으로 반복 작업이 수행됨을 확인할 수 있었다.

5. 결 론

본 논문에서는 제조로봇의 복잡한 작업능력이 요구되는 추세에 맞는 임베디드 시스템까지 통합된 그리퍼 시스템 모듈화 설계 및 제어에 대하여 서술하였다. 고정밀 작업수행을 위해 그리퍼에 자세 변환을 측정할 수 있는 센싱 모듈을 포함하여 실시간으로 자세변환을 측정할 수 있다. 정밀한 제어를 위해 그리핑과 자세변환 센싱을 담당하는 CDSP와 CDSP를 제어

하고 외부의 다른 로봇 인터페이스와의 API를 개발하여 로봇의 위치까지 제어할 수 있는 CARM을 결합하여 DCSPC 시스템을 구성하였다. CARM과 CDSP 사이의 통신에는 실시간성을 위해 고속 SCI통신을 사용하며 그리퍼의 변환된 위치 값, 그리핑 제어, 디지털 I/O를 제어하는 API를 개발하고 이를 실험을 통해 확인하였다. CARM과 외부 로봇 인터페이스 UR3와 실험을 위해 UR 로봇 제어기에 별도의 TCP/IP 소켓 통신을 할 수 있는 UR 스크립트 소켓 프로그램을 개발하였으며, 로봇의 현재 위치를 읽어내거나 이동시킬 수 있는 API를 개발하였다.

실시간성이 중요한 시스템이기 때문에 CARM에 Xenomai 2.6.3 RTOS패치를 하여 환경을 구성하고 CARM의 실시간성 성능 실험의 결과, 주기 500 μ s일 때 499999.91 ns의 평균 주기와 560.71 ns의 편차를 보여 정밀한 정주기성을 확인하였다. CARM과 CDSP와의 통신을 위해 개발한 API는 최대 312.66 μ s이 소요됨을 확인하였고, CARM과 외부 로봇과의 통신에 대해선 일부 API가 1 ms이상의 시간이 소요되었고, 추후 시스템 개선의 여지가 있음을 확인하였다.

추후에 DCSPC 시스템에 카메라와 관련된 연구 및 실험을 진행하여 그리퍼가 외부의 환경요인에 의한 상황판단 및 로봇과 그리핑의 제어를 보다 능동적으로 대응할 수 있을 것으로 기대된다. 이를 위해 카메라의 성능분석을 하고, 이를 확인하기 위해 내부적으로 조립 알고리즘 구동을 하여 최종적으로 로봇에 탑재해서 실제 조립 알고리즘을 구동할 예정이다. 또한 본 논문에서 개발한 DCSPC와 그리퍼를 실제 로봇 말단에 부착 후 조립 어플리케이션을 수행 후 관련 성능 분석을 진행할 예정이다.

Reference

- [1] D.H. Hong, J. An, S.M. Lee, S.Y. Jeong, and I.H. Lee, "Modular of controller for electric gripper," *KSPE Spring Conference*, Jeju, Korea, pp. 147, 2016.
- [2] S.U. Son, S.Y. Lee, M.B., and Kim, S.H. Kim, "Conceptual design on modularity of motor driven robot gripper" *KSPE Autumn Conference*, Jeju, Korea, pp. 305-306, 2014.
- [3] S.P. Cheon, S. Kim, G. Baek, and I.S. Lee, "Industrial roboto technical trend focused on actuator," *KSMTE Spring Conference*, Jeju, Korea, pp. 270-271, 2011.
- [4] Denso Manual, *Denso Manual Download*, [Online], <http://www.denso-wave.com/ko/robot/download/>, Accessed: Jan. 19. 2018.
- [5] Robostar Manual, *Robostar Manual Download*, [Online], <http://www.robostar.co.kr/kr/56/subview.do>, Accessed: Accessed: Jan. 9. 2018.
- [6] UR3 Manual, *UR3 Manual Download*, [Online], https://www.universal-robots.com/media/207442/ur3_user_manual_en_global.pdf, Accessed: Jan. 9. 2018.

- [7] A. Barbalace, A. Luchetta, G. Manduchi, M. Moro, A. Soppelsa, C. Taliercio, "Performance comparison of VxWorks, Linux, RTAI, and Xenomai in a hard real-time application," *IEEE Transactions on Nuclear Science*, Vol. 55, No. 1, pp. 435-439, 2008.
- [8] M.D. Marieska, A.I. Kistijantoro, and M. Subair, "Analysis and benchmarking performance of real time patch Linux and Xenomai in serving a real time application," *2011 International Conference on Electrical Engineering and Informatics*, Bandung, Indonesia, 2011, doi: 10.1109/ICEEI.2011.6021563.
- [9] J.H. Koh, and B.W. Choi, "Real-time performance of real-time mechanisms for rtai and xenomai in various running conditions," *International Journal of Control and Automation*, Vol. 6, No. 1, pp. 235-246, 2013.
- [10] W. Betz, M. Cereia, and I. C. Bertolotti, "Experimental evaluation of the Linux RT Patch for real-time applications," *IEEE Conference on Emerging Technologies & Factory Automation*, Mallorca, Spain, 2009, doi: 10.1109/ETFA.2009.5347056.
- [11] M. Chianzone, and G. Sulligoi, "Performance assessment of a motion control application based on linux RTAI," *2010 International Symposium on Power Electronics Electrical Drives Automation and Motion (SPEEDAM)*, Pisa, Italy, 2010, doi: 10.1109/SPEEDAM.2010.5542055.
- [12] Xenomai, *Home Page of Xenomai*, [Online], <https://www.xenomai.org>, Accessed: Jan. 9. 2018.
- [13] S.-Y. Lee, U.-H. Ham, Y.-W. Park, H.-S. Jung, I.-K. Jung, and S. Lim, "Implementation and performance evaluation open-source controller for precision control of gripper," *Second International Conference on Robotics and Machine Vision*, Kitakyushu, Japan, 2017, doi: 10.1117/12.2301196.



이 승 용

2014 서울시립대학교 기계정보공학과(공학사)
 2016 동대학원 기계정보공학과(공학석사)
 2016 ~ 현재 전자부품연구원 연구원 재직
 2017 ~ 현재 서울시립대학교 기계정보공학과 박사과정 재학

관심분야: 모션 제어 시스템, 오픈 소스 임베디드 시스템, 실시간 시스템



함 윤 형

2016 인천대학교 전기공학과(공학사)
 2016 ~ 현재 전자부품연구원 연구원 재직

관심분야: 임베디드 시스템, 실시간 시스템, 오픈 소스 시스템



박 영 우

2016 경기대학교 전자공학과(공학사)
 2015 ~ 현재 전자부품연구원 연구원 재직

관심분야: 임베디드 시스템, 실시간 시스템, 오픈 소스 시스템



정 일 군

1996 아주대학교 제어공학과(공학사)
 1998 동대학원 제어공학과(공학석사)
 2004 INP (National Polytechnic Institute) 컴퓨터 공학과(공학박사)
 2004 ~ 2006 LG전자 리빙센터 근무
 2006 ~ 현재 전자부품연구원 지능로봇틱스 센터장

관심분야: 모션 제어 시스템, 오픈 소스 임베디드 시스템, 실시간 시스템



임 선

2002 고려대학교 전기전자공학과(공학사)
 2005 동대학원 전기공학과(공학석사)
 2015 동대학원 전기전자공학과(공학박사)
 2005 ~ 2008 두산기술원 전자기술팀 근무
 2008 ~ 현재 전자부품연구원 지능로봇틱스 선임연구원

관심분야: 모션 제어 시스템, 오픈 소스 임베디드 시스템, 실시간 시스템, 비선형 제어 이론, 적응제어 시스템, 초음파 탐상검사 기법