

Matlab 과 USB DAQ 장치를 이용한 Rapid Control Prototyping System 개발

Development of a Rapid Control Prototyping System Based on Matlab and USB DAQ boards

이영삼*, 양지혁, 김석윤, 김원식, 권오규
(Young-Sam Lee¹, Ji-Hyuk Yang¹, Seuk-Yun Kim¹, Won-Sik Kim¹, and Oh-Kyu Kwon¹)

¹Inha University

Abstract: In this paper, we propose a new and cost-effective RCP (Rapid Control Prototyping) system based on Matlab/Simulink and a DAQ (Data Acquisition) unit with the high speed USB communication interface. The proposed RCP system has a feature that a computer on which Simulink is running acts as a realtime controller and a DAQ unit performs data acquisition, transmission of the data to and from a computer, and the application of control data received from the computer. For its implementation, we develop 10 communication blocks each of which is constructed by using S-function. In order to increase the data communication speed and thus to reduce the sampling period of the overall control system, we propose to use a batch transfer strategy through the USB interface. The proposed RCP system has several advantages over existing methods such as good maintainability, portability due to the USB interface, low cost, and no necessity for C-code generation even though it can only be applied to control systems with moderate sampling rates. It is expected that the proposed RCP system can be useful in teaching control-related topics to undergraduate and graduate students.

Keywords: high speed USB interface, RCP (Rapid Control Prototyping), block libraries, DAQ unit

I. 서론

RCP (Rapid Control Prototyping) 시스템이란 제어시스템 개발자가 단기간에 제어 알고리즘을 효율적으로 설계, 개발, 검증하기 위해 사용하는 개발환경을 말한다. 전통적으로 RCP 시스템은 Simulink와 같은 블럭다이어그램 방식의 모델링 프로그램, 하드웨어 입/출력 장치를 제어하기 위해 사용하는 라이브러리 블럭, C-code 생성기, 실시간 타겟(target) 컴퓨터, 타겟 컴퓨터와 통신하는 호스트(host) 컴퓨터 등으로 구성된다. Simulink를 이용하여 제어기를 설계하고 모의실험등을 통해 테스트 해본 후 만족스러운 결과가 나올 경우 입/출력 제어 라이브러리 블럭을 사용하여 실제 시스템을 제어 할 수 있도록 제어기를 구성한다. 자동 C-code 생성기를 이용하여 구성된 제어기 모델에 대한 C-code를 생성하고 컴파일 한 후 실시간 타겟 컴퓨터에 다운로드하여 실시간 제어 실험을 수행하게 된다. 제어 실험이 수행되는 도중 타겟 컴퓨터와 연결된 호스트 컴퓨터를 이용하여 관찰하고자 하는 신호를 실시간으로 확인해 보고 특정 파라미터를 실시간으로 튜닝할 수도 있다. 하드웨어 입/출력 장치를 제어하기 위한 Simulink용 라이브러리 블럭이 제공되고 C-code 생성 기능이 있기 때문에 제어 시스템 개발자는 지루하고 오류가 발생하기 쉬운 수

작업 코딩(coding)을 하지 않아도 되고 하드웨어 장치를 제어하는 것에 대한 부담없이 오로지 제어 알고리즘의 개발이라는 본연의 목적에만 집중할 수 있게 된다. RCP 시스템은 특히 ECU (Engine Control Unit) 개발과 같이 자동차 산업에서 활발하게 사용되고 있다.

RCP 시스템과 관련된 몇몇 소프트웨어가 상업적으로 판매되고 있는데 Matlab/Simulink는 가장 잘 알려지고 널리 사용되는 프로그램이다. Simulink의 추가 제품인 RTW (Real-time Workshop)은 Simulink로 구성된 블럭 다이어그램 모델에 대해 자동으로 ANSI-C Code를 생성해 준다[1]. Simulink의 또다른 추가제품인 Embedded Coder는 특정 내장형 프로세서에 특화된 C-code를 생성해 줌으로써 제품 생산 기간을 단축해주는 기능을 제공해준다[2]. Matlab/Simulink와 RTW는 개방형 구조를 취함으로써 직접 개발한 하드웨어 장치에 대해서도 RCP 소프트웨어를 적용할 수 있도록 하고 있다. 이런 개방성 때문에 자체개발된 하드웨어에 Matlab/Simulink/RTW를 사용한 RCP 시스템들이 학술적인 연구로써 다수 제안되었다 [3-7]. 더불어 RCP 시스템의 적용에 대한 연구들도 다수 발표되고 있다[8,9].

최근 국내의 상당수 공과대학에서는 공학교육의 내실을 기하기 위해 공학교육인증 교과과정을 채택하고 있다. 공학교육인증 교과과정의 특징 중 하나는 실험 수업과 설계 수업에 큰 비중을 두고 있다는 것이다. 일반적으로 공과대학에서 이루어지는 학생들의 실험/실습 수업은 학교의 실험실내에서만 이루어지고 학생들의 집에서 개인적인 실험/실습은 이루어지고 있지 않다. 이는 수업에 사용되는 장비와 소프트웨어가 학교에만 설치되어 있기 때문이다. 만약 학생들이 학교의

* 책임저자(Corresponding Author)

논문접수: 2012. 6. 21., 수정: 2012. 8. 22., 채택확정: 2012. 9. 19.

이영삼, 양지혁, 김원식, 권오규: 인하대학교 전기공학부

(lys@inha.ac.kr/zero1zuck@hanmail.net/nedjam220@gmail.com/okkwon@inha.ac.kr)

김석윤: 인하대학교 로봇공학전공(silentwind.kim@gmail.com)

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급인력 과정 지원사업과 로봇 특성화 대학원 지원사업의 연구결과로 수행되었음(NIPA-2012-H0401-12-1007, C7000-1001-0009).

실험/실습실 이외에도 집에서 실험 및 설계를 수행해 볼 수 있다면 좀더 빠르게 수업에 익숙해 질 수 있게 되어 수업 효율을 크게 향상시킬 수 있을 것이다. 많은 공과대학에서 제어 관련 교과목이 설강되고 있는데 이 과목은 실험 및 설계를 포함하는 형식으로 운영되고 있다. RCP 시스템을 활용한 제어 시스템 설계 및 실험은 학생들에게 수작업 코딩의 번거로움을 덜게 하고 제어 알고리즘 설계라는 본연의 목적에 충실할 수 있게 함으로써 과목에 대한 이해와 설계 및 실험에 대한 학생들의 흥미를 크게 증진할 수 있을 것이다. 하지만 상업적으로 판매하는 RCP 시스템이나 학술적인 연구로 개발된 RCP 시스템 모두 고가의 하드웨어 장치와 소프트웨어를 필요로 하거나 휴대성이 떨어지는 하드웨어 장치를 기반으로 하기 때문에 학생들이 집에서 설계 및 실험을 해볼 수 있게 하기에는 현실적으로 많은 어려움이 따른다. 예를 들어 PC를 타겟컴퓨터로 사용하고 DAQ 장치를 사용해서 입출력을 처리하는 RCP 시스템의 경우 DAQ 장치가 ISA 또는 PCI 버스 방식으로 인터페이스 되기 때문에 특정 PC에 장착되어 있어야 한다. 휴대가 가능한 임베디드 보드를 타겟으로 사용하는 RCP 시스템의 경우 Embedded coder를 사용하고 특정 프로세서용 컴파일러를 필요로 한다. 모두 고가의 소프트웨어이기 때문에 학생들이 개별적으로 소유할 수는 없다. 본 논문은 기존 RCP 시스템이 갖는 이러한 문제점(휴대성, 비용)을 해결하고 학부과정이나 대학원 학생들이 학교 뿐만 아니라 집에서 손쉽게 설계 및 실험을 해볼 수 있도록 하는 RCP 시스템을 제안하는 것을 목표로 하고 있다.

휴대성이 좋은 하드웨어 장치를 만들기 위한 하나의 방법은 장치를 USB로 인터페이스 하는 것이다. 제어가 목적이 아니라 단순히 데이터 수집이 목적인 경우 이미 USB를 이용한 DAQ 장치를 사용하는 방법들이 많이 시도되고 있다[10-12]. USB는 Plug-and-play를 지원하고 장치를 PC에 접속할 때 PC의 전원을 끌 필요가 없기 때문에 하드웨어 장치의 휴대성을 극대화시킬 수 있다[13]. 하지만 기존 RCP 시스템중에서 PC가 제어기 역할을 직접 수행하는 구조인 xPC Target과 같은 시스템에서는 USB 통신 방식의 하드웨어 장치를 사용하는 방법을 사용하지 않는다. 이는 USB 통신 방식이 실시간 제어에 필요한 소량 데이터의 고속 전송에 적합하지 않기 때문이다. 하지만 학부나 대학원 학생들의 설계 및 실험에는 아주 빠른 고속의 샘플링이 필요하지 않을 가능성이 높기 때문에 USB를 이용한 하드웨어 장치의 사용이 가능한 대안이 될 수 있다. 게다가 High speed USB는 Full speed USB에 비해 상대적으로 고속 통신이 가능하기 때문에 High speed 통신 방식을 사용한다면 통신속도도 크게 개선할 수 있을 것이다.

Embedded coder와 컴파일러 같은 고가의 소프트웨어 사용은 기본적으로 RCP 시스템이 C-code 생성을 기반으로 하고 있기 때문이다. 따라서 Code 생성을 사용하지 않고 Simulink만을 사용해서 RCP 시스템을 구성할 수 있다면 비용에 대한 문제를 해결할 수 있을 것이다. 실제로 추가 구입 제품을 포함하지 않는 기본 Matlab과 Simulink는 많은 학교에서 구비하고 있고 이 소프트웨어들은 서버에 접속해 특정 인원 내에서 동시에 사용이 가능한 형태로 사용하기 때문에 학생들이 집에서 사용해도 될 수가 있다. 따라서 소프트웨어에 대한 비용

문제를 해결할 수 있다.

본 논문에서는 기존 RCP 시스템이 가지는 휴대성 문제와 비용 문제를 해결할 수 있는 RCP 시스템 제안을 목적으로 한다. 이를 위해 본 논문에서는 USB 통신을 통해 인터페이스 되는 DAQ 장치를 사용하고 C-code 생성과정 없이 바로 PC측의 Simulink가 제어기가 되는 형태의 RCP 시스템을 제안한다.

II. 제안 RCP 시스템의 개요

제안되는 RCP 시스템은 그림 1과 같이 플랜트를 제외하고 크게 2개의 하부 시스템(subsystem)으로 구성된다. 첫번째 하부 시스템은 MS Windows 상에서 Matlab/Simulink가 실행되는 PC 시스템이고 연구실에서 자체 개발된 DAQ 장치가 두번째 하부 시스템을 구성한다. PC와 DAQ 장치는 High speed USB 통신을 이용하여 데이터(제어 데이터 및 센서 데이터)를 주고 받는다. Simulink를 이용하여 구성된 제어기 모델에 대해 C-code를 자동생성하고 컴파일하여 내장형 제어장치(embedded control unit)에 다운로드하여 제어기를 실행시키는 기존의 RCP 시스템과 달리 제안되는 RCP 시스템은 Code 생성과정 없이 바로 Simulink를 실행함으로써 제어기를 실행하는 점이 특징이다. 제안되는 RCP 시스템에서 제어가 수행되는 방식을 개략적으로 정리하면 다음과 같다.

1. DAQ 장치가 제어에 필요한 센서 데이터를 측정한 후 USB통신을 통해 PC 측으로 전송한다.
2. PC에서 수행중인 Simulink는 DAQ 장치에서 전송된 센서 데이터를 기반으로 제어 연산을 수행한다.
3. 계산된 제어 연산 결과를 DAQ 장치로 전송한다.
4. DAQ 장치에서는 PC 측에서 전송된 제어 데이터를 출력 장치(DAC, PWM 등등)를 통해 출력한다.

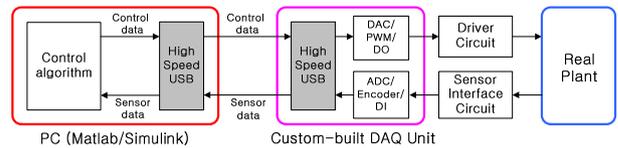


그림 1. 제안되는 RCP 시스템의 구성도.

Fig. 1. The conceptual diagram of the proposed RCP system.

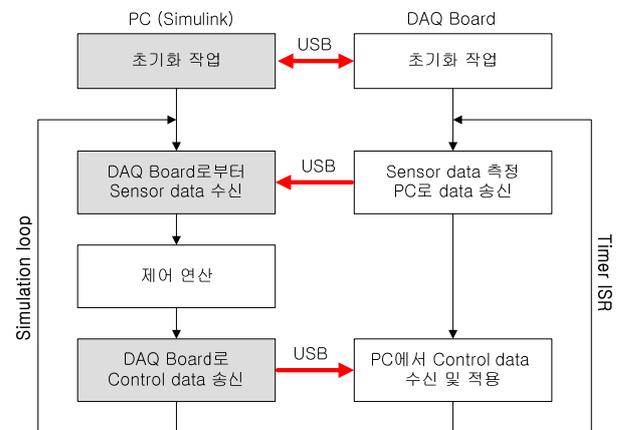


그림 2. 제안 RCP 시스템의 수행 방식에 대한 흐름도.

Fig. 2. The flowchart on the operation of the proposed RCP System.

그림 2는 위에서 설명한 개략적 수행 방식을 흐름도를 이용하여 나타내고 있다. 흐름도에서 음영진 부분에 대해서는 논문 뒷부분에서 좀더 다루기로 한다. DAQ 장치가 PC 측으로 송신해야 할 센서 데이터와 PC로부터 수신받을 제어 데이터의 종류는 Simulink 상에서 수행되는 제어기의 구성에 달려있다. 제어기는 Simulink에서 지원하는 여러가지 기능 블록과 본 RCP 시스템을 위해 개발된 입/출력 처리용 통신 블록을 이용하여 구성한다. 제안되는 RCP 시스템에서 제어 연산은 전적으로 PC에서 수행되며 DAQ 장치는 센서데이터의 측정과 구동 신호를 출력하는 기능을 수행한다. 제어연산을 일정한 주기로 실시간 수행하기 위해 DAQ 장치에 탑재된 마이크로컨트롤러의 timer interrupt 기능을 활용한다. 위에 주어진 개략적 수행방식의 1번과 4번은 DAQ 장치쪽에서, 그리고 2번과 3번은 PC 쪽의 Simulink에서 실행된다. 1번과 4번에 해당하는 부분을 timer interrupt service routine으로 처리하면 비록 2번과 3번이 PC 쪽에서 실행되지만 전체의 과정이 약간의 편차를 보이긴 하지만 일정한 주기로 실행되게 된다.

1. DAQ 장치

본 RCP 시스템에서 사용되는 DAQ 장치는 다양한 실험을 수행할 수 있도록 여러 종류의 입/출력 장치들을 탑재하고 High speed USB 통신 기능을 지원하도록 자체 개발되었다. 그림 3과 그림 4는 개발된 DAQ 장치의 실물 사진과 하드웨어 구성 개념도이다. 제안되는 RCP 시스템에서 DAQ 장치는 제어연산에는 관여하지 않고 제어연산에 필요한 센서 데이터를 측정하여 PC로 전송하고 PC측의 Simulink가 계산한 제어



그림 3. ATMEGA128기반의 자체 개발된 DAQ 장치.

Fig. 3. Lab-built ATmega128-based DAQ board.

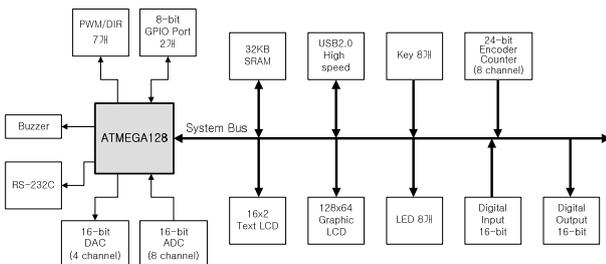


그림 4. DAQ 장치의 하드웨어 구성 개념도.

Fig. 4. The conceptual diagram on the hardware of the DAQ board.

데이터를 수신받아 출력장치를 통해 출력하는 기능을 수행한다. DAQ 장치에 탑재된 입/출력 장치와 통신 장치를 제어하기 위한 마이크로컨트롤러로는 ATMEGA128을 사용하였다. ATMEGA128은 16MHz의 클럭을 사용하는 8-bit 저가형 마이크로컨트롤러로써 GCC 컴파일러라는 무료 컴파일러가 지원되며 다양한 응용사례에 대한 자료들을 손쉽게 찾을 수 있기 때문에 개발에 용이한 장점이 있다. 제안되는 RCP 시스템에서 고속의 계산이 필요한 제어연산은 Simulink가 수행되는 PC 쪽에서 수행되므로 DAQ 장치에 사용되는 마이크로컨트롤러는 고속연산을 지원하지 않아도 시스템의 성능에 큰 영향을 미치지 않는다.

개발된 DAQ 장치는 High speed USB 통신기능을 지원한다. USB는 plug-and-play를 지원하므로 개발된 DAQ 장치가 특정 PC에 종속되지 않고 휴대가 가능하도록 하며, 데스크탑이나 랩탑 컴퓨터에 모두 사용할 수 있는 장점이 있다. 최근 공학 교육인중 교과과정이 실험이나 설계를 중요시하고 있는 상황에서 학생들에게 집에서라도 실험, 실습을 가능하게 하기 위한 DAQ 장치의 통신방식으로 USB가 최선의 선택이라고 할 수 있다. 여러가지 상용 USB 장치가 시판되고 있지만 그중에서 Cypress Semiconductor의 CY7C68016A를 사용하였다. CY7C68016A는 FX2라고 불리는데 FX2는 High speed를 지원하기 때문에 최대 480Mbps의 전송속도를 가진다. 참고로 Full speed USB 장치의 전송속도는 12Mbps이다. High speed USB가 통신속도가 빠르기는 하지만 이것은 큰 양의 데이터를 하나의 Frame에 실어서 전송하는 방식이기 때문이다. 제안되는 RCP 시스템에서는 작은 양의 데이터를 매 샘플링 주기마다 보내고 받는 형식을 취하므로 USB 통신이 실시간 제어를 위한 데이터 전송에는 그다지 적합한 방식은 아니다. 하지만 Full speed USB의 Frame rate가 1ms 인 것에 비해 High speed USB의 경우에는 Microframe이라는 것을 사용하며 Frame rate는 125us로서 Full speed에 비해 8배 빠르고 기존의 RS232와 같은 직렬통신에 비해서도 빠른 통신속도를 가지기 때문에 본 RCP 시스템에서는 High speed USB 통신방식을 사용한다.

그림 4에 도시된 다양한 입/출력 장치들중 주요 입/출력 장치들은 제안되는 RCP 시스템에서 자체 개발된 Simulink 라이브러리 블록을 이용하여 DAQ 장치로부터 값을 받거나 보낼 수 있도록 하였다. 라이브러리 블록이 지원되는 입/출력 장치들의 세부사항은 표 1과 같다.

표 1. 입/출력 장치의 세부사항.

Table 1. The details of input/output peripherals.

입/출력 장치	세부사항
PWM	6채널, 10KHz 주기
DAC	16 bit 해상도, 4채널, 출력범위: 0-5V, DAC8554 이용
Digital Output	16채널, 74LS574 이용
LED	8 bit 정보 On/Off 출력
ADC	16 bit 해상도, 8채널, 입력범위: 0-5V, ADS8344 이용
Digital Input	16채널, 74LS541 이용
Switch Input	8 bit 정보 입력, 74LS574 이용
Encoder	24 bit 해상도, 8채널, LS7566 이용

2. RCP 통신 블록 라이브러리

제안되는 RCP 시스템에서는 USB 통신을 통하여 DAQ 장치로부터 센서 데이터를 전송받거나 제어 데이터를 DAQ 장치로 전송하기 위해 Simulink 환경내에서 사용할 수 있는 10 개의 통신 라이브러리 블록을 지원한다. 그림 5는 개발된 10 개의 통신 블록들을 나타내고 있다. 2열로 구성된 라이브러리 블록 중에서 첫번째 열에 위치한 블록들은 DAQ 장치로부터 데이터를 전송받기 위한 Receive 블록이고 두번째 열에 위치한 블록들은 DAQ 장치 쪽으로 데이터를 전송하기 위한 Send 블록들이다. 블록 중에서 블록의 이름옆에 숫자가 있는 블록들은 여러 개의 채널을 지원하는 블록을 나타낸다. 그림 5에서는 모두 1번 채널이 선정된 경우의 블록들의 모양을 보여주고 있다. 여러개의 채널을 동시에 사용하고자 할 경우에는 블록을 더블클릭하여 대화상자를 통하여 사용하고자 하는 채널을 지정할 수 있다. 그림 6은 ADC 블록의 1번과 3번 채널의 정보를 USB 통신을 이용하여 수신하도록 대화상자를 설정하는 것과 대화상자 설정 후에 ADC 블록의 출력 포트가 2개로 변한 것을 보여주고 있다.

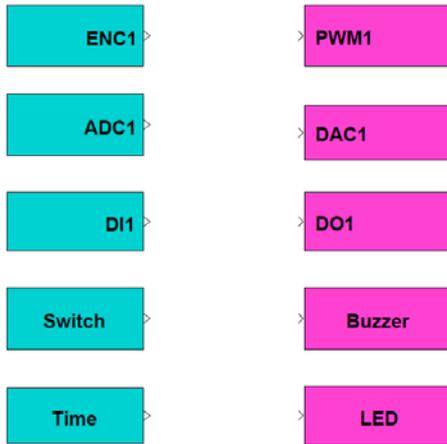


그림 5. 제안되는 RCP시스템의 기반이 되는 Simulink용 통신 라이브러리 블록.

Fig. 5. Communication blocks provided by the proposed RCP system for use with Simulink.

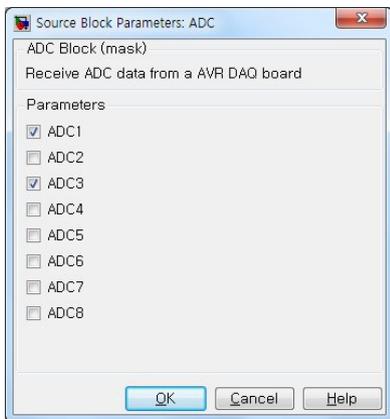


그림 6. 대화상자를 통한 채널의 지정과 블록의 모양 변경.
Fig. 6. Channel selection using a dialog box and the corresponding block shape change.

3. RCP 시스템을 이용한 제어기 구성사례

도립진자 시스템에 대한 스윙업(Swing-up) 제어 실험을 수행하기 위하여 제안된 RCP 시스템을 적용하였다.

그림 7은 도립진자의 실시간 제어 실험을 적용하기 위해 연구실에서 자체 개발한 도립진자의 실물 사진을 보여주고 있다. 실험에 사용된 도립진자 시스템은 봉이 직립상태를 기준으로 이루는 각과 카트(cart)의 수평방향 변위를 측정하기 위해 2개의 Incremental encoder를 센서로 사용한다. 그리고 카트를 좌우로 이동시키기 위한 구동기(actuator)로 DC 모터를 사용한다. DC 모터는 H-bridge 형태의 모터드라이버에 의해 구동된다. 따라서 도립진자 시스템을 제어하기 위해서는 그림 5에 나온 라이브러리 블록 중에서 Encoder 블록과 PWM 블록을 사용한다. Encoder 블록의 경우 2개의 채널을 처리해야 하므로 대화 상자를 이용하여 해당되는 채널을 사용하게끔 설정한다. 그림 8은 제안되는 RCP 시스템에서 제공하는 통신 라이브러리 블록(Encoder 블록과 PWM 블록)과 Simulink에서 제공하는 기능 블록들을 조합하여 구성한 도립진자 시스템의 스윙업 제어기 모델을 나타낸다. Encoder 블록의 1번 채널과 2번 채널이 활성화 되어 있는 것을 볼 수 있다. 또한 제어연산의 결과를 반영하기 위한 PWM 블록이 사용된 것을 볼 수 있다. 제어 알고리즘은 스윙업 제어를 수행하는 동안에는 비선형 제어기법을 이용하고 직립상태에 가까워지면 선형제어기로 교체하는 방식을 사용한다. 통신블럭 이외의 부분은 Simulink에서 제공되는 블럭라이브러리를 이용하여 구성하였다. 제어 연산 자체가 Simulink 내에서 이루어지고

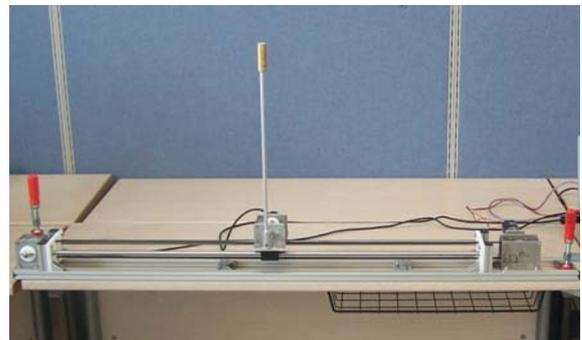


그림 7. 실시간 제어 응용을 위해 사용된 도립진자 시스템.
Fig. 7. An inverted pendulum system used for the application of realtime control.

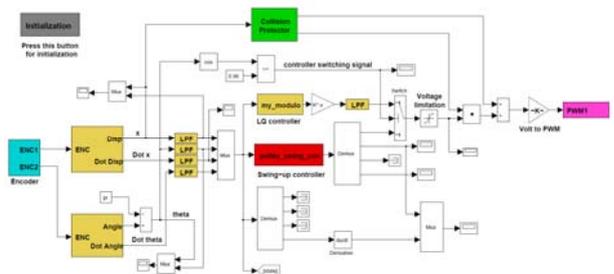


그림 8. 제안된 RCP 시스템을 이용하여 구성된 도립진자의 스윙업 제어기 모델.
Fig. 8. The swing-up controller model constructed using the proposed RCP system.

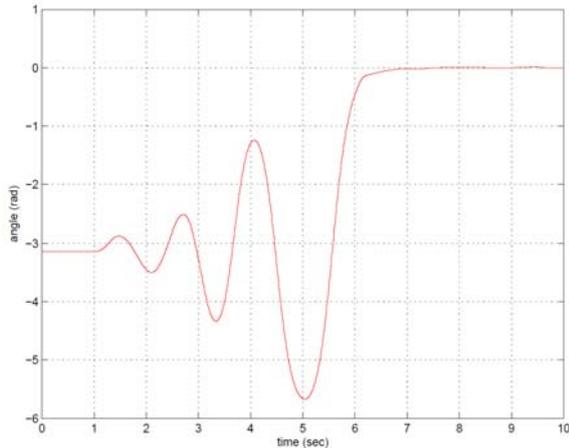


그림 9. RCP 시스템을 이용한 도립진자 스윙업 제어의 실험 결과.

Fig. 9. The result of the swing-up control experiment.

센서 데이터도 USB 통신을 통해서 Simulink 쪽으로 전송되기 때문에 모든 신호를 Simulink의 Scope 블록을 이용하여 실시간 관측할 수 있다. 또한 저장하고 싶은 데이터를 Matlab의 Workspace로 저장할 수도 있다. 이 제어모델의 샘플링 주기는 8 ms로 1초당 125회의 제어연산을 수행하게끔 설정하였다. 샘플링 주기는 해당 Simulink 모델의 샘플링 주기와 동일하므로 Simulink 모델의 파라미터 설정을 통해 샘플링 주기를 설정할 수 있다. 그림 9는 실험결과를 보여준다. 처음에 지면을 향하도록 위치해 있던 봉이 점점 흔들리면서 7초쯤 도립 상태로 안정화 되는 것을 볼 수 있다. 그림 9는 그림 7의 모델에서 Theta라고 이름이 붙은 신호에 Simulink의 To Workspace 블록을 활용하여 데이터를 저장한 후 Matlab의 stairs 명령을 이용하여 그린 그림이다. 이처럼 제안되는 RCP 시스템을 사용하면 실시간 제어 실험과 더불어 실시간 관측이 가능하고 데이터를 저장하여 후처리 할 수도 있다.

제안된 RCP 시스템은 다양한 시스템의 제어에 활용될 수 있다. 실제로 [14,15] 등에서 다루었던 제어 시스템은 제안된 RCP 시스템을 통하여 구현되어 있다.

III. 소프트웨어 구성

제안되는 RCP 시스템에서 지원되는 10개의 통신 라이브러리 블록은 S-Function을 이용하여 구현하였다. S-Function은 Simulink에서 기본적으로 지원하는 라이브러리 블록을 조합해서는 구현할 수 없는 블록을 사용자가 정의하여 사용할 수 있게끔하는 기능으로써 블록의 동작을 Matlab file 또는 C-code로 기술하도록 되어 있다. S-Function을 C-code를 이용하여 정의하면 실행속도를 크게 높일 수 있고 하드웨어 장치를 제어할 수 있는 장점이 있기 때문에 제안되는 RCP 시스템에서는 C-code 기반의 S-Function을 사용하였다. C-code 기반의 S-Function 작성은 미리 지정된 이름의 Callback function들을 정의하는 것으로 구성된다. 통신 라이브러리 블록을 구현하기 위해 다음과 같은 이름의 Callback function을 S-Function에 정의하였다: mdlInitializeSizes, mdlInitializeSampleTimes, mdlStart, mdlOutputs, mdlTerminate. S-Function에 기술된 Callback function들은 사용자가 호출하는 것이 아니라 Simulink가 적절한 순

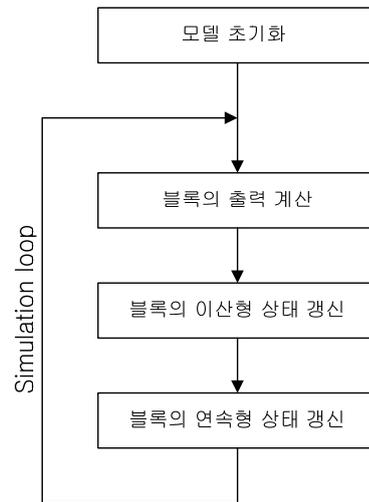


그림 10. Simulink가 simulation을 수행하는 흐름도.

Fig. 10. The flowchart on the operation of Simulink.

간에 호출하게 된다. 여기서 적절한 순간에 호출된다는 말의 의미를 이해하기 위해서는 Simulink의 동작방식에 대한 이해를 필요로 한다. 그림 10은 Simulink가 Simulation을 수행하는 방식을 흐름도로써 나타낸 것이다. Simulink 모델을 수행하면 Simulink에서는 그림 10에 주어진 흐름도에 따라 단계별로 Simulation을 수행한다. 처음 진행되는 단계는 모델 초기화 단계로 모델을 실행하기 위해 필요한 여러가지 초기화 작업을 수행한다. 이러한 초기화 작업중의 하나로는 모델에 포함된 블록들의 계산 시행순서(Execution order)를 정하는 과정이다. 초기화단계가 종료되면 Simulation loop 안으로 진입하여 그림 10에 주어진 순서대로 단계별 계산을 수행한다. 먼저 블록의 출력을 계산하는 단계를 수행한다. 이때 Simulink는 초기화 단계에서 구한 시행순서에 따라 각 블록의 출력계산 함수를 호출하게 된다. 출력계산 단계가 종료되면 Simulink는 이산형 상태의 갱신 단계와 연속형 상태의 갱신 단계를 차례로 수행한다. 이 단계들 역시 초기화 단계에서 구한 시행순서에 따라 수행되지만 블록이 이산형 상태나 연속형 상태를 가졌을 때만 수행된다.

만약에 S-Function을 이용하여 구성한 블록이 Simulink 모델에 사용되었다면 해당 S-Function 블록도 모델 전체내에서 특정 시행순서를 가지게 된다. Simulink는 출력계산 단계에서 S-Function 블록의 순서가 되었을 때 S-Function 블록의 출력을 계산하는 Callback function인 mdlOutputs를 호출하게 된다. 물론 S-Function 블록이 이산형 상태나 연속형 상태를 가질 경우 그 블록의 시행 순서가 되었을 때 상태를 갱신하기 위한 Callback function이 호출되게 된다. S-Function 블록은 모델 내에서 여러개의 Instance를 가질 수 있다. 즉 같은 S-Function을 사용하는 블록을 여러개 사용할 수 있다. 각각의 블록들은 대화상자를 통해 다른 파라미터를 전달 받을 수 있다. 예를 들어 S-Function을 이용하여 PID 제어 블록을 구성하였다면 모델 내에 여러개의 PID 블록을 사용할 수 있고 각각의 PID 블록들은 자기 고유의 파라미터(예를 들어 P계수, I계수, D계수)를 가질 수 있다.

제안되는 RCP 시스템에서 제공하는 10개의 통신 라이브러리

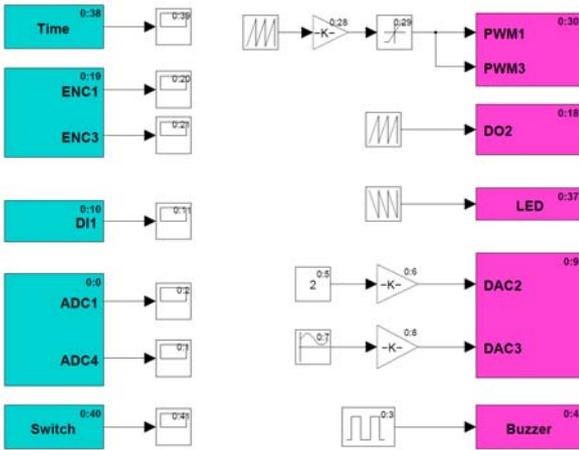


그림 11. RCP용 통신 블록 테스트를 위한 Simulink 모델.
Fig. 11. Simulink model constructed to test all RCP communication blocks.

리 블록들은 하나의 S-Function을 이용하여 정의되며 각 통신 블록의 기능 구별은 내부적으로 전달 받는 파라미터를 통해 결정된다. 예를 들어 PWM 블록은 '0x80'을, ADC 블록은 '0x20'을 파라미터로 넘겨받는다. 물론 이러한 파라미터를 전달하는 과정에 일반 사용자들은 개입할 필요가 없고 단지 해당 라이브러리 블록을 모델 내에 포함시키기만 하면 된다.

통신 라이브러리 블록을 구현하는데 있어서 중점을 두었던 점은 데이터를 일괄적으로 전송하고 일괄적으로 수신하도록 처리하는 것이었다. 만약 하나의 Simulink 모델에 여러 개의 Send 블록과 Receive 블록이 사용될 경우 USB를 통한 데이터의 전송이 해당 통신 블록의 시행 순서가 되었을 때 이루어지는 방식으로 처리한다면 하나의 통신 블록의 데이터를 처리하기 위해 USB에서는 하나의 Microframe이 소요된다. 따라서 통신 블록이 여러개일 경우 통신에 소요되는 시간이 늘어나게 되고 결국 제어시스템의 샘플링 주기를 짧게 하는데 저해 요소가 된다. 그림 11의 Simulink 모델을 이용하여 이를 좀더 설명하기로 한다. 그림 11의 모델은 제안하는 RCP 시스템에서 제공하는 10개의 통신 라이브러리 블록을 모두 사용하여 동작을 테스트하기 위해 구성한 Simulink 모델이다. 각 블록의 우측 상단에 보이는 숫자는 블록의 시행 순서이다. 예를 들어 ADC 블록은 시행순서 0, Encoder 블록은 시행순서 19번을 갖는다. 참고로 시행순서가 보이지 않는 블록은 여러 개의 블록으로 구성된 하부시스템이다. 만약 데이터 통신을 일괄 전송 방식이 아닌 블록별 전송 방식을 택한다면 ADC 블록은 0번 시행순서 때에 데이터를 수신 받고 Buzzer 블록은 시행순서 4번에 DAQ 장치 쪽으로 데이터를 전송하게 된다. 모두 10개의 통신 블록이 있으므로 10개의 microframe이 소요된다. 이것은 1.25ms에 해당한다. 이 시간은 가장 이상적인 상황을 가정했을 경우에 USB 통신에만 소요된 시간이다. 실제로 모델 내의 다른 블록들을 계산하는데 소요된 시간을 포함하고 Windows 운영체제가 Multitasking 인 점을 고려하면 모델내의 모든 블록을 한번 계산하는데 소요되는 시간은 이것보다 훨씬 더 길어서 샘플링 주기가 5ms 또는 10ms가 넘는 경우에만 적용할 수 있게 될 것이다. 따라서 이러한 문제점을 해결하고 빠른 샘플링 주기를 갖는 제어

표 2. 그림 11의 모델에 대한 RxDataInfo table 및 TxDataInfo table.

Table 2. RxDataInfo and TxDataInfo for the Simulink model given in Figure 11.

RxDataInfo	TxDataInfo
ADC CH1	Buzzer
ADC CH4	DAC CH2
Digital Input CH1	DAC CH3
Encoder CH1	Digital Output CH2
Encoder CH3	PWM CH1
Time	PWM CH3
Switch	LED

시스템에도 사용할 수 있도록 일괄 전송 방식을 지원하게끔 S-Function의 Callback function을 구현하였다. S-Function에 구현한 다섯개의 Callback function 중에서 가장 중요한 mdlStart와 mdlOutput을 중심으로 S-Function에 기술된 callback function의 동작을 설명하기로 한다.

통신 블록을 사용했을 경우 일괄적으로 센서 데이터를 DAQ 장치로부터 수신 받고 일괄적으로 제어데이터를 DAQ 장치로 전송하기 위해서는 먼저 어떠한 데이터가 통신의 대상이 되는지에 대한 정보를 DAQ 장치 측으로 넘겨주어야 한다. 이러한 정보는 2개의 Table(Code 상에서는 Array로 구현된다)을 통해서 전달되는데 하나는 RxDataInfo table이고 또 하나는 TxDataInfo table이다. RxDataInfo table은 DAQ 장치로부터 PC 측으로 전송해야 할 데이터의 종류를 순서대로 가지고 있으며 TxDataInfo table은 PC 측에서 DAQ 장치로 전송해야 할 데이터의 종류를 순서대로 가지고 있다. 예를 들어 그림 11에 주어진 모델에 대한 표 2와 같이 구해진다. 참고로 Table에 적힌 데이터의 순서는 Simulink 모델에 포함된 통신 블록들의 시행순서를 따르는 것을 알 수 있다. Simulation loop가 실행되기 전에 DAQ 장치는 이 정보를 PC (Simulink)로부터 전달받게 되고 Simulation loop가 시작되면 매번 Timer interrupt service가 시행될 때마다 RxDataInfo table에 있는 정보를 이용하여 해당 데이터를 모두 측정한 후 일괄적으로 USB 통신을 이용하여 PC 측으로 전송하고 PC측의 Simulink가 제어 연산을 수행하고 난 결과를 일괄적으로 DAQ 장치에 전송하면 TxDataInfo table에 있는 정보를 이용하여 받은 데이터를 해당 출력장치에 출력하게 된다. 통신 블록을 정의하는 S-Function에 정의되어 있는 mdlStart라는 Callback function에서 수행하는 여러 작업중 가장 중요한 2가지 작업은 바로 이 2개의 Table을 생성하는 작업과 모델내에 사용된 통신 블록중에서 시행순서가 가장 빠른 블록과 가장 늦은 블록의 종류를 알아내는 작업이다. 그림 11에 주어진 모델의 경우 ADC 블록이 통신 블록중에서 시행순서가 가장 빠르고 Switch 블록이 시행순서가 가장 늦은 블록이다. Simulink가 모델 초기화 작업 단계를 수행할 때 모델내 블록들의 시행순서에 따라 각 블록의 초기화 함수를 호출하게 되는데 RCP용 통신 블록의 경우에는 mdlStart가 호출된다. 따라서 각 통신 블록의 mdlStart가 호출되면 각 통신 블록이 다루는 데이터 정보를 전역변수로 잡혀있는 RxDataInfo 또는 TxDataInfo table에 순서대로 채워넣게 된다.

통신 블록의 mdlOutput이라는 Callback function에서는 다음

과 같이 5가지 조건을 순서대로 검사하고 조건을 만족할 경우 할당된 작업을 수행한다. 여기서 5가지 조건은 상호배타적인 조건이 아니어서 2개 이상의 조건이 만족될 수 있다.

- $t = 0$ 이고 통신 블럭중 시행순서가 가장 앞서는 블럭일 경우: mdlStart를 통해 구한 Table 정보를 DAQ 장치 측으로 넘겨주고 DAQ 장치의 Timer interrupt 기능을 활성화 시킨다. (DAQ 장치는 이 단계에서 PC로부터 넘어온 정보를 이용하여 PC로 전송해야 될 데이터의 종류와 PC로부터 전송받아야 될 데이터의 종류를 알 수 있게 된다. 또한 DAQ 장치의 Timer interrupt가 활성화 되기 때문에 PC로 전송해야 할 센서 데이터를 모두 측정 한 후에 PC측으로 일괄 전송하게 된다.)
- 통신 블럭중 시행순서가 가장 앞서는 블럭일 경우: 해당 블럭이 Send 블럭이든 Receive 블럭이든 상관없이 무조건 DAQ 장치로부터 USB 통신을 통해 일괄 전송된 센서 데이터를 하나의 배열 RxBuffer에 저장한다. 그림 11의 모델의 경우 ADC 블럭이 이 경우에 해당한다.
- Receive 블럭일 경우: RxBuffer에 저장된 일괄 센서 데이터 중에서 해당 Receive 블럭에 해당되는 데이터를 찾아 Receive 블럭의 출력을 갱신한다.
- Send 블럭일 경우: 블럭의 입력 데이터를 TxBuffer라는 배열에 채워넣는다. Send 블럭이 여러 개 있을 경우 배열에는 여러 개의 데이터가 채워지게 된다.
- 통신 블럭중 시행순서가 가장 늦는 블럭일 경우: 해당 블럭이 Send 블럭이든 Receive 블럭이든 상관없이 무조건 TxBuffer에 저장된 데이터를 USB 통신을 이용하여 DAQ 장치 측으로 전송한다. 그림 11의 모델의 경우 Switch 블럭이 이 경우에 해당한다. (DAQ 장치에서는 이 단계에서 넘어온 일괄 데이터를 TxDataInfo table에 있는 정보에 맞춰 해당 출력장치로 데이터를 출력한다.)

제안되는 RCP 시스템은 위에서 기술한 방법을 통하여 하나의 Simulink 모델에 여러 개의 통신 블럭이 있어도 일괄 전송과 일괄 수신이 가능하게 하였다. 이러한 방법을 통하여 제안되는 RCP 시스템은 샘플링 주기가 1 ms 정도인 제어 시스템에도 안정적으로 적용할 수 있다.

IV. 실시간 성능 및 장단점 논의

일체의 실시간 연산을 내장형 연산 장치에서 수행하는 기존의 RCP 시스템과는 달리 제안되는 RCP 시스템의 경우 제어 연산 자체는 고속계산이 가능한 PC에서 이루어지고 있기 때문에 제어연산에 소요되는 계산량 자체는 제어시스템의 샘플링 주기 선정에 그다지 영향을 못 미친다. 제안되는 RCP 시스템에서 샘플링 주기 선정에 영향을 주는 것은 USB 통신에 의해 소요되는 시간이다. 이 때문에 제안되는 RCP 시스템에서는 USB 통신에 소요되는 시간을 줄이기 위해 일괄 전송 방식을 제안하였다. 하지만 그렇다고 하더라도 기존 방식의 RCP 시스템에서 가능한 것처럼 수십 KHz에서 수백 KHz에 이르는 고속 샘플링은 제안되는 RCP 시스템에서는 불가능하다. 물론 기존 RCP 시스템에서도 그 정도의 고속 샘플링은 제어 연산량이 작은 경우거나 정수형 연산만을 사

용한 경우에 가능하다. 그렇다면 제안되는 RCP 시스템을 이용할 경우 제어 시스템의 샘플링 주기를 어느 정도까지 낮출 수 있는지에 대해 논의해 보자.

High speed USB를 사용할 경우 Full speed USB에 비해 Frame 속도가 8배 빠르다. High speed USB에서는 Microframe을 사용하며 하나의 Mframe은 125us 동안 지속된다. 따라서 DAQ 장치로부터 PC로의 데이터 전송이 하나의 Microframe 내에 이루어지고 PC로부터 DAQ로의 데이터 전송이 또 하나의 Microframe에 이루어 진다면 USB 통신에 소요되는 시간은 가장 이상적일 경우 250us 이내가 될 수 있다. PC에서 이루어지는 제어 연산이 하나의 Microframe 내에 이루어진다면 통신과 제어연산을 포함한 모든 시간이 375 us 보다 작을 수 있다. 만약에 PC에서 이루어지는 제어 연산의 연산량이 2개의 Microframe 정도 이내에서 이루어진다면 전체 소요시간은 500us 보다 작을 수 있다. 이러한 분석을 토대로 한다면 2KHz 정도의 샘플링 주파수도 가능할 수 있다. 이같은 분석이 과연 실제 시스템의 제어 실험에서도 적용할 수 있는 합리적인 분석인지를 살펴보기 위해 앞서 다루었던 독립진자의 제어 실험을 예로 들어 살펴보자. 독립진자 제어 시스템의 샘플링 주기는 8ms (따라서 샘플링 주파수는 125Hz) 이다. 한번의 루프 계산이 완전히 종료되는데 소요되는 시간을 측정하기 위하여 ATMEGA128의 Timer interrupt가 시작되는 시점에 Port D의 2번 Port에 Logic 1을 출력하고 Timer interrupt 안에서 지정한 작업이 모두 종료되는 순간에 역시 Port D의 2번 Port에 Logic 0로 출력하였다. 그리고 실험이 진행되는 동안 해당 Port의 신호값을 디지털 오실로스코프를 이용하여 측정한 후 저장한 데이터를 Matlab을 이용하여 그림으로 도시하였다(그림 12참조). 그림 12에서 T 는 샘플링 주기(Timer interrupt의 주기와 같다)이고 T_1 은 Timer interrupt에 진입했다가 빠져 나오는데 걸리는 시간을, T_2 는 아무것도 하지 않고 다음 Timer interrupt에 진입할 때까지 기다리는 시간이다. 당연히 $T = T_1 + T_2$ 가 성립한다. 그림 12를 통해서 알 수 있듯이 샘플링 주기를 8 ms로 하여 실시간 제어

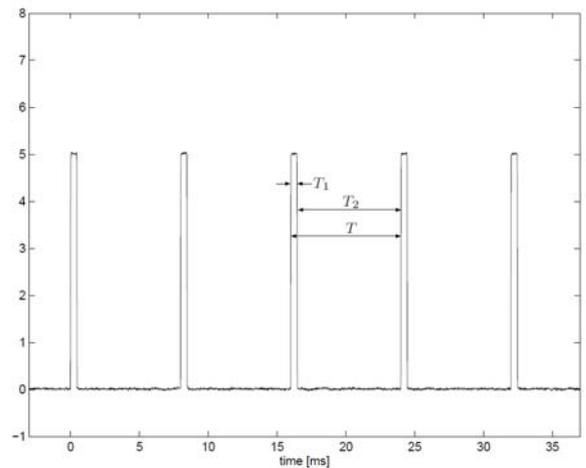


그림 12. 오실로스코프로 확인한 RCP 시스템의 실시간 수행 성능 (샘플링 주기 8ms).

Fig. 12. The oscilloscope-measured realtime performance of the RCP system (sampling period of 8ms).

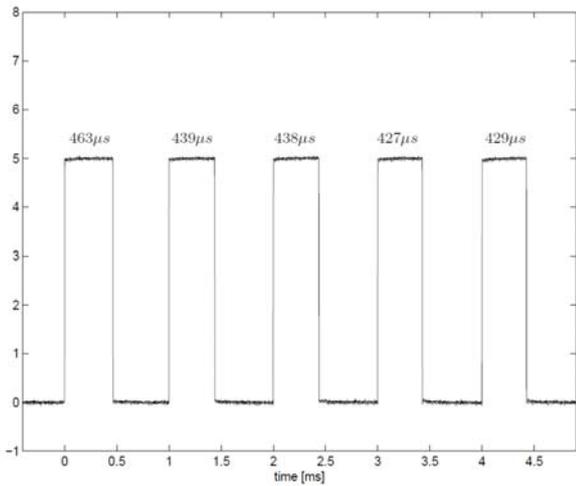


그림 13. 오실로스코프로 확인한 RCP 시스템의 실시간 수행 성능 (샘플링 주기 1 ms).

Fig. 13. The oscilloscope-measured realtime performance of the RCP system (sampling period of 1 ms).

실험을 하였을 때 여전히 시간적 여유가 많이 있는 것을 볼 수 있다 (T_1 이 T 에 비해 현저히 작다). 따라서 도립진자 시스템의 경우 샘플링 주기를 8 ms보다 더 작게 선정해도 실시간 성능에 문제가 없을 것이라는 것을 예상할 수 있다. 실제로 샘플링 주기를 1ms로 낮게 조정하면 제어실험을 수행해 보았고 실시간 성능을 오실로스코프로 확인해 보았다(그림 13참조). 그림 13에서 보이는 숫자는 Timer interrupt에 진입했다가 빠져나 오는데 까지 소요된 시간을 us 단위를 이용하여 나타낸 것이다. 소요된 시간이 모두 동일하지는 않지만 비교적 균일하게 나타나는 것을 볼 수 있다. 가장 길게 소요된 시간도 500us를 넘지 않는 것을 볼 수 있다. 따라서 2 KHz까지 샘플링을 높여보는 것도 가능하다. 하지만 Windows 환경이 Multitasking을 지원하고 실시간 운영체제가 아닌 점을 고려한다면 제안되는 RCP 시스템을 이용하여 실시간 제어실험을 수행할 경우 1KHz 정도의 샘플링이 적당할 것으로 생각된다. 실제로 이러한 1KHz 정도의 샘플링 속도로 제어 실험을 해볼 수 있는 시스템은 상당히 많기 때문에 학부나 대학원 과정의 실험을 수행하는데 무리 없이 사용될 수 있다.

위의 분석에서 지적한 바와 같이 제안되는 RCP 시스템을 이용할 경우 고속의 샘플링을 지원할 수는 없다. 그리고 기존의 RCP 시스템에서 가능한 실시간 파라미터 튜닝이 지원되지 않는 단점이 있다. 반면 기존 RCP 시스템에 대비되는 제안되는 RCP 시스템의 장점은 다음과 같다.

- 유지, 보수의 편리성을 고려하여 개발된 시스템이다. 대학교에서 사용하는 학부 또는 대학원용 실험 실습 기자재들이 대부분 외부에서 구매한 것 이어서 유지, 보수에 어려움을 겪는다. 이런 점을 고려해 본다면 유지, 보수의 편리성은 큰 장점이라고 할 수 있다.
- 저가의 마이크로컨트롤러(ATMEGA128)를 이용하고 Realtime Workshop이나 Embedded coder와 같이 기존 RCP 시스템에서 필요로 하는 고가의 소프트웨어가 필요 없이 가장 기본적인 Matlab/Simulink만으로 수행 가능하다. 많은 대학교

에서 실험 실습용으로 기본적인 Matlab/Simulink를 가지고 있으므로 제안되는 RCP 시스템을 적용할 수 있는 여건은 충분하다.

- 기존 RCP 시스템과는 달리 Simulink 자체가 제어기 기능을 수행하고 C-code를 생성하는 과정과 컴파일의 필요 없다. 따라서 제어기를 변경하고 실험해보는데 소요되는 시간이 크게 단축된다.
- DAQ 장치가 USB 통신을 사용하기 때문에 장치가 특정 PC에 종속될 필요가 없다. 따라서 학생들의 경우 학교에서 뿐만 아니라 집에서 실험이 가능하며, 이것은 학생들의 학습 효율향상에 크게 기여할 수 있다.

V. 결론

이 논문에서는 High speed USB 통신기능을 갖춘 DAQ 장치와 Matlab/Simulink를 이용한 RCP 시스템을 제안하였다. 제안되는 RCP 시스템은 10개의 라이브러리 블록을 기반으로 Simulink를 이용하여 손쉽게 제어 시스템을 구성할 수 있도록 설계되었다. 제공되는 라이브러리 블록을 구현하고 있는 S-function의 구조를 기술하였고 DAQ 장치에서 연동되어 수행되는 소프트웨어의 구조에 대해서도 살펴보았다. 제안되는 RCP 시스템을 이용할 경우 C언어와 같은 개발언어에 익숙하지 않은 사람도 Simulink 환경하에서 빠른 시간에 제어기를 설계하고 검증해 볼 수 있다. 제어 알고리즘이 PC 측에 설치된 Simulink에서 수행되므로 Simulink의 대부분의 기능을 활용할 수 있고 또한 Matlab에서 제공하는 함수들도 활용할 수 있어 복잡한 제어알고리즘의 구현도 가능하다. 제안되는 RCP 시스템은 시간지연, 실시간성, 샘플링 주기 등의 측면에서 기존의 RCP 시스템에 비해 약점을 가지지만 USB 기반의 DAQ 장치에 기반하므로 휴대성이 좋고, C-code 생성 및 컴파일이 필요하지 않고 바로 실행 할 수 있는 간편성, 또한 고가의 장비 및 소프트웨어를 필요로 하지 않는 경제성 등의 측면에 있어서 기존의 RCP 시스템에 비해 강점을 가진다. 제안된 RCP 시스템을 실제 도립진자의 스윙업 제어기 설계에 적용한 사례를 통하여 제안된 시스템의 편리성을 예시하였다. 향후 다양한 주변장치에 대한 통신블럭들을 추가함으로써 제안된 RCP 시스템의 기능을 더 향상시키는 연구를 수행할 예정이다.

참고문헌

- [1] The Mathworks Inc., "Real-time workshop user's guide (rtw ug.pdf)," Version 6, 2005.
- [2] The Mathworks Inc., "Real-time workshop embedded code user's guide (ecoder ug.pdf)," Version 4, 2005.
- [3] S. Rebeschief, "MIRCOS-Microcontroller-based real time control system toolbox for use with Matlab/Simulink," Proc. IEEE International Symposium Computer Aided Control System Design, pp. 267-272, 1999.
- [4] K. H. Hong, W. S. Gan, Y. K. Chong, K. K. Chew, C. M. Lee, and T. Y. Koh, "An integrated environment for rapid prototyping of DSP algorithms using and Texas Instruments' TMS320C30," Microprocessors and Microsystems, vol. 24, no. 7, pp. 349-363, 2000.
- [5] W. Lee, M. Shin, and M. Sunwoo, "Target-identical rapid

- control prototyping platform for model-based engine control," *Proc. InstnMech. Engrs Part D, J. Automobile Engineering*, vol. 218, pp. 755-765, 2004.
- [6] D. Hercog and K. Jezernik, "Rapid control prototyping using Matlab/Simulink and a DSP-based motor controller," *International Journal Engng Ed.*, vol. 21, no. 3, pp. 1-9, 2005.
- [7] R. Bucher and S. Balemi, "Rapid controller prototyping with Matlab/Simulink and Linux," *Control Engineering Practice*, vol. 14, pp. 185-192, 2006.
- [8] C. F. Lin, C. Y. Tseng, and T. W. Tseng, "A hardware-in-the-loop dynamics simulator for motorcycle rapid controller prototyping," *Control Engineering Practice*, vol. 14, pp. 1467-1476, 2006.
- [9] R. Kennel, "Improved direct torque control for induction motor drives with rapid prototyping system," *Energy Conversion and Management*, vol. 47, pp. 1999-2010, 2006.
- [10] M. I. Khalil, "A USB-based data acquisition system for neutron TOF measurements," *Meas. Sci. Technol.*, vol. 17, pp. N1-N7, 2006.
- [11] V. Fanti, R. Marzeddu, G. Piredda, and P. Randaccio, "A portable acquisition system based on USB standard for the Medipix2 X-ray detector," *IEEE Trans. Nuclear Science*, vol. 53, no. 5, pp. 2578-2583, 2006.
- [12] H. Jiang, M. Ojaruega, F. D. Becchetti, H. C. Griffin, and R. O. Torres-Isea, "A USB-based portable data acquisition system for detector development and nuclear research," *Nuclear Instruments and Methods in Physics Research A*, vol. 652, pp. 483-486, 2011.
- [13] Universal Serial Bus Specification, www.usb.org.
- [14] J. H. Yang, S. Y. Kim, Y. S. Lee, and O. K. Kwon, "Development of magnetic force modeling and equipment for magnetic levitation systems," *Journal of Institute of Control, Robotics, and Systems (in Korean)*, vol. 17, no. 4, pp. 321-327, 2011.
- [15] D. H. Kwak, T. H. Kim, and Y. S. Lee, "Correction method of tracking for astronomical telescope using recursive least square method," *Journal of Institute of Control, Robotics, and Systems (in Korean)*, vol. 18, no. 3, pp. 224-229, 2012.

이 영 삼

제어 · 로봇 · 시스템학회 논문지 제15권 제4호 참조.

양 지 혁

제어 · 로봇 · 시스템학회 논문지 제15권 제4호 참조.

김 석 윤

제어 · 로봇 · 시스템학회 논문지 제17권 제4호 참조.



김 원 식

2012년 인하대학교 전자공학과 졸업.
2012년~현재 인하대학교 대학원 전기공학과 석사과정 재학중. 관심분야는 Rapid Control Prototyping, 임베디드 시스템, 이동로봇.

권 오 규

제어 · 로봇 · 시스템학회 논문지 제15권 제7호 참조.