

# C++Builder를 이용한 제어시스템 모니터링

## Control System Monitoring Using C++Builder

신 명 준\*, 손 영 익\*\*, 김 갑 일\*\*\*  
(Myong-Jun Shin, Young-Ik Son, Kab-Il Kim)

**Abstract** - When a control system is designed computer simulations are important because they provide a way to test the performances of the control law and determine some design parameters of the control system. Since most real situations are different from those of the simulations, however, it is required to manage uncertainties of the plant and analyze errors fast and correctly. For this we consider a convenient monitoring system which can display real systems responses exactly and check the performances of the control systems. In this paper, an approach to design a monitoring system is presented by using C++Builder program which is based on PC. Firstly we introduce the structure and functions of the monitoring system. Then, an inverted-pendulum control system is investigated by using the monitoring system.

**Key Words** : C++Builder, control system, real-time monitoring, inverted-pendulum

### 1. 서 론

일반적인 제어시스템에서 컨트롤러(예를들어 마이크로프로세서)는 센서로부터 값을 읽고 연산을 통하여 제어입력을 산출하며 적절한 제어입력을 플랜트에 인가하기 위한 목적으로 사용된다. 최근 PC의 성능이 향상됨에 따라 범용컴퓨터 또는 전용컨트롤러가 제어시스템에 사용되고 있으며 처리속도가 빠르기 때문에 보다 정확한 시스템 제어가 가능하게 되었다.

그러나 이러한 전용 컨트롤러는 가격이 비싸고 그 구조가 복잡하기 때문에 쉽게 접근하기가 어렵다. 게다가 컨트롤러와 함께 제공된 소프트웨어만을 사용해야 하므로, 시스템 개발에 필요한 부가적인 기능들의 추가가 어려워 국한된 인터페이스만을 사용하는 것은 여간 불편한 일이 아니다.

이 논문에서는 시스템 제어에 필요한 간단한 모니터링시스템 혹은 인터페이스를 구현하는 방법에 대해 기술하고 있다. 이를 역진자 시스템에 적용함으로써 마이크로컨트롤러와 PC의 역할 및 제어시스템의 구조에 대한 이해를 돕고 있으며 C++Builder로 작성한 인터페이스의 기능을 보여주고 있다. [1][2][3]

### 2. 본 론

#### 2.1 모니터링 시스템의 구조

모니터링 시스템은 그림 1과 같은 구조로 되어 있다. 마이크로컨트롤러가 플랜트에 설치된 센서로부터 값을 읽어오고

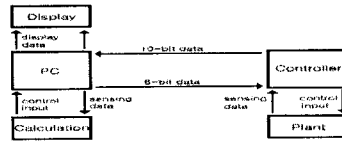


그림 1 모니터링시스템의 구성도

시리얼 통신을 통해 10bit 데이터 형태로 PC에 전송한다. PC는 수신된 데이터 일부를 처리하여 PC모니터에 디스플레이 하고 나머지는 연산을 통해 제어입력을 구하며 이 데이터를 다시 마이크로컨트롤러로 전송한다. 마이크로컨트롤러는 전송된 제어입력 데이터를 변환해서 플랜트에 인가하게 된다. 마이크로컨트롤러가 제어입력에 관련된 연산을 수행하며 동시에 여러개의 센서값을 읽어오는 과정에서 오차가 발생할 확률은 크다. 일반적으로 저가의 마이크로컨트롤러는 처리속도가 매우 낮기 때문이다. 데이터의 연산부분을 PC가 수행하도록 하여 컨트롤러의 연산량을 줄이고 데이터의 처리결과를 모니터 디스플레이를 통해 육안으로 판단할 수 있다는 것이 이 모니터링 시스템의 가장 큰 특징이다.

#### 2.2 모니터링 시스템의 설계 및 구현

본 논문에서 개발된 모니터링 시스템을 적용할 플랜트는 그림 2와 같은 역진자 시스템이다. 시스템의 물리량을 대입하면 다음의 상태방정식이 유도된다.

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0025 & 1.6081 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0059 & 27.0906 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 2.4768 \\ 0 \\ 5.8816 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (1)$$

이 시스템은 제어가 용이하지 않은 대표적인 불안정 시스템

\* 辛 明 俊 : 明知大學校 電氣工學科 碩士課程  
\*\* 孫 瑛 翼 : 明知大學校 電氣工學科 助教授 · 工博  
\*\*\* 金 甲 一 : 明知大學校 電氣工學科 教授 · 工博

이다. 시스템에 대한 더 자세한 내용은 다른 참고문헌을 참조한다.[4][5]

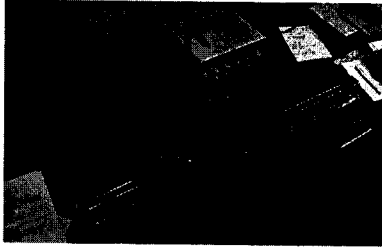


그림 2 역진자 시스템

마이크로컨트롤러는 센서로부터 위치값과 각도값을 입력받는다. 컨트롤러의 데이터 연산량을 줄이기 위해 위치값은 엔코더의 펄스값을 각도값은 포텐서미터의 A/D 컨버팅된 값을 그대로 통신에 사용한다. 컨트롤러는 그림 3과 같은 패킷 형태의 데이터를 PC로 전송한다.[6][7] 그리고 PC는 데이터를 가공하며 이를 통해서 매 샘플링 타임마다 속도와 각속도의 값을 저장하고 일부 연산된 값을 모니터에 디스플레이 한다. 제어입력  $u$ 는 다음식과 같이 표현되며

$$u = kx = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (2)$$

PC는 연산된 제어입력( $u$ )와 플랜트에 인가해야 할 방향 데이터를 그림 4의 패킷 형태로 마이크로컨트롤러에 전송한다.

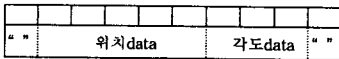


그림 3 PC로 전송되는 데이터 패킷

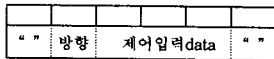


그림 4 컨트롤러로 전송되는 데이터 패킷

컨트롤러는 수신된 데이터를 방향성분과 제어입력 성분으로 나누고 제어입력 데이터를 PWM형태로 변환하여 모터에 인가한다. 역진자 시스템의 제어에 필요한 이상의 개념을 적용하여 C++Builder로 작성한 모니터링 프로그램이 그림 7에 있다.

### 2.3 하드웨어의 설계 및 구현

그림 5는 실험을 위해 제작된 역진자 시스템에 필요한 부품의 종류와 이 부품들의 결선을 나타낸 회로도이다. 컨트롤러는 AVR시리즈 중 ATMEGA128을 사용하였다. 16MHz로 동작하는 저가의 범용 마이크로컨트롤러이다. 포텐서미터와 엔코더는 각각 막대의 기울기와 수레의 위치를 측정하는 센서이다. 포텐서미터는 컨트롤러 내부의 A/D컨버터를 통해 막대의 절대 기울기를 출력하고, 엔코더는 모터의 회전수마다 일정한 펄스를 출력하므로 펄스의 개수를 이용해서 수레의 상대위치를 알 수 있다. 특히 엔코더의 경우 기준위치가 아닌 단순한 위치의 증가, 감소만을 센싱하므로 펄스를 놓치지 않도록 하는 것이 중요하며, 동작원리는 그림 6에 있다.

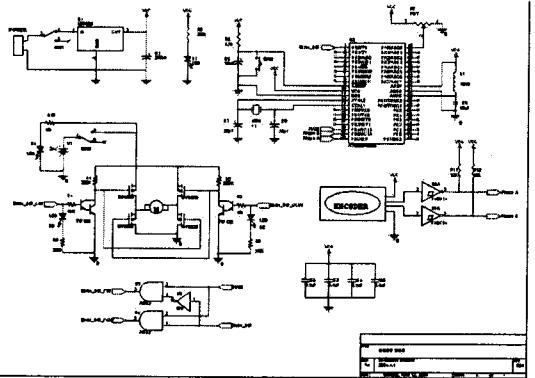


그림 5 역진자 시스템의 컨트롤러부 회로도

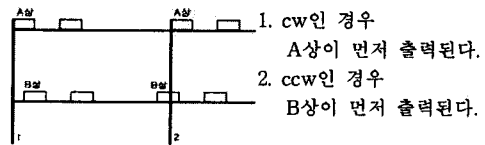


그림 6 엔코더의 동작원리에 대한 모식도

### 2.4 모니터링 시스템의 기능

PC기반의 모니터링 시스템이 마이크로컨트롤러의 데이터 연산량을 줄여준다. 그렇기 때문에 컨트롤러가 엔코더의 펄스를 놓친다거나 A/D 컨버팅 타임을 놓치는 등의 에러를 줄일 수 있다. 또 센싱된 데이터의 값을 PC모니터에 디스플레이 하기 때문에 센싱된 데이터의 잡음성분 유무를 판단할 수 있고 데이터의 흐름이나, 시스템의 문제점을 보다 쉽게 찾아 낼 수 있다. 마이크로컨트롤러는 프로그램의 다운로드를 통하여 그 동작 순서나 인터럽트의 사용여부를 결정하게 된다. 만약 우리가 사용한 이득값이 실제 시스템에 맞지 않는다면 매번 이득값을 바꾸고 다시 다운로드 해야하는 번거로움이 있다. 이 모니터링 시스템이 지닌 가장 큰 장점 중의 하나는 이러한 번거로움이 없다는 것이다. 사용자가 모니터링 시스템을 통해서 원하는 이득값을 컨트롤러로 보내면 컨트롤러의 수신 인터럽트에서 제어입력이 수신된 것인지 이득값이 수신된 것인지를 판단하게 된다. 만약 이득값이 수신된 것이라면 새로운 이득값을 적용하는 알고리즘이 컨트롤러에 있기 때문이다.

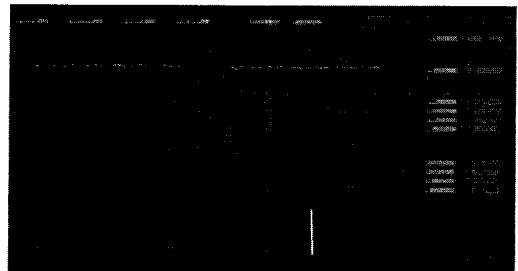


그림 7 역진자 시뮬레이션1

그림 7에서 각각의 위치, 속도, 각도, 각속도를 텍스트 뿐만 아니라 매 샘플링 시간에 대해 실시간 그래프로 보여주고 있

다. 스크롤바를 이용해서  $k_1, k_2, k_3, k_4$ 의 이득을 설정하고 확인버튼을 눌러 마이크로컨트롤러에 그 값을 전달할 수 있으며 플랜트로 입력되는 PWM의 크기와 모터의 동작 방향을 알 수도 있다. 실제 이러한 데이터를 이용하여 카트의 위치와 각도를 표현한 그림이 나타나 있다. 수ms단위로 동작하는 역진자 제어시스템에서 데이터의 흐름과 플랜트의 동작을 육안으로 쉽게 확인할 수 있다.

### 2.5 모니터링 시스템의 적용

실험을 통해 모니터링 시스템의 성능을 검증해 본다. 그림 8은  $k = [-97.4679 \ -49.4898 \ 127.7323 \ 26.2696]$  일때의 시뮬레이션 결과이다.

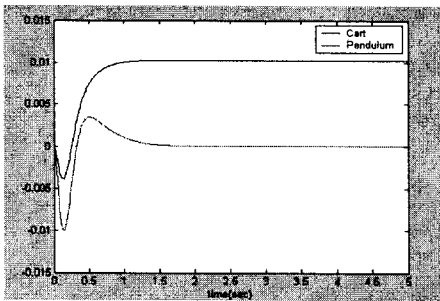


그림 8 역진자 시뮬레이션1

이를 실제 시스템에 적용하면 그림9과 같이 시스템이 불안정해진다.

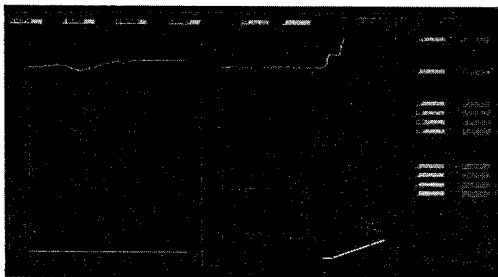


그림 9 역진자 시뮬레이션2

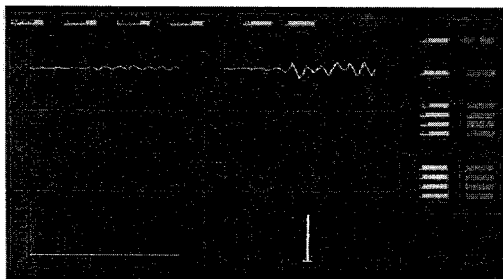


그림 10 역진자 시뮬레이션3

그림 10은 실험을 통해  $k_1, k_3$ 의 이득을 수정한 시뮬레이션이다. 위치와 각도의 진동이 나타나지만 그림 9보다는 시스템의 실제 응답이 개선되었음을 알 수 있다.

본 논문에서는 PC 기반의 모니터링 시스템 개발에 목표를 두고 있으므로, 이를 활용한 보다 정밀한 제어에 관한 연구는 차후에 진행할 계획이다.

### 3. 결 론

시스템을 정확하게 모델링하는 것은 불가능한 일이다. 그러므로 시뮬레이션 상에서 얻는 제어 결과가 모든 시스템의 경우에 대해 실제와 일치한다고 할 수 없다. 특히 수 ms이하의 샘플링 시간을 가지는 제어시스템의 경우 매 샘플링 시간마다 어떠한 값들이 입력되고 어떻게 처리되는지를 확인하는 것은 쉬운 일이 아니다.

본 논문에서 제안한 모니터링 시스템을 이용하면 데이터의 흐름과 플랜트의 상태를 쉽게 파악할 수 있다. 논문에서 제안한 시스템은 간단한 C++Builder 작성법과 시리얼 통신을 이력하면 손쉽게 꾸밀 수 있다.

PC 기반 제어입력에서는 통신으로 인한 지연현상이 발생할 수 있다. 통신을 통한 지연현상은 ms단위로 데이터 처리를 하는 제어시스템에 있어서는 치명적이다. 시간지연 현상이 제어시스템에 미치는 영향에 대한 규명과 대처 방법을 찾는 것은 앞으로의 과제이며, 활발히 연구되고 있는 주제 중의 하나이다.[7]

#### 감사의 글

본 연구는 과학기술부 및 한국과학재단의 ERC 프로그램을 통한 지원으로 이루어졌으며 이에 감사드립니다.

#### 참 고 문 헌

- [1] 조한우, "Borland C++ Builder: Visual development tool", 문화전사, 1997
- [2] 박영근, "Borland C++ Builder", 상조사, 1999
- [3] 정태영, "볼랜드 C++ Builder 정복", 가남사, 1997
- [4] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing Up Control of Inverted Pendulum", IECON'91, 1991.
- [5] Richard C. Dorf, Robert H. Bishop, "Modern Control Systems", Eighth Edition, Pearson Education, 2000
- [6] 한경호, 최천원, "Implementation of two wire RS232C Serial Communication Interface using CSMA protocol", 조명전기설비학회논문지, Vol.17, No.3, 12-17, 2003
- [7] 서석훈, 유정봉, 우광준, "Fault Detection of BLDC Motor Using Serial Communication Based Parameter Estimation", 조명전기설비학회논문지, Vol.16, No.5, 49-50, 2002