

가상머신 드라이버를 이용한 윈도우즈 환경에서의 실시간 제어기법

The Real-Time Control Technique Over the Environment of Windows Using Virtual Machine Driver

장성욱, 이진걸

(Sung-ouk Chang and Jin-Kul Lee)

Abstract : We studied the technique which can control the real system without additional hardware drivers using virtual machine driver operated on the windows operating system. We showed the feasibility of the proposed scheme under the error and the delay of a sampling time on the multi task processing through the load test of the experiment using graphic user interface.

Keywords : virtual machine driver, graphic user interface, sampling time

I. 서론

최근 고성능의 마이크로 프로세서(microprocessor)의 개발은 개인용 컴퓨터 성능을 비약적으로 발달시켰으며, 이러한 하드웨어(hardware)의 발달을 바탕으로 종래에 텍스트(Text)환경에서의 거의 모든 작업이 사용자의 편의를 고려한 윈도우즈(Windows)환경으로 대체되고 있는 실정이다. 이러한 환경의 변화 때문에, 거의 대부분의 운영체제(Operation System) 및 응용 소프트웨어는 윈도우즈 환경에서 처리하도록 설계되고 있으며, 컴퓨터를 손쉽게 사용 가능하도록 하는데 큰 기여를 하고 있다. 그러나, 윈도우즈 환경은 막강한 그래픽 환경으로 인하여 일반 초급 사용자가 손쉽게 사용할 수 있도록 하기 위한 많은 편리함을 지니고 있는 반면에, 컴퓨터의 하드웨어를 직접 액세스(access)하는 것과 같은 고급기능에 대한 제약이 많도록 설계되어 있다. 이러한 문제점으로 인하여, 대부분의 기계 및 물리적 시스템을 실제로 제어하는 제어전공자들은 실 시간성 보장과 정확한 샘플링 시간(Sampling Time)을 위해서 실시간 운영체제를 제공하는 실시간 운영체제(Real-Time Operating System)를 별도로 구입하여 제어시스템에 적용하거나, 일반적으로 흔한 도스(Dos)와 같은 텍스트 환경하에서 시스템에 대한 제어를 행하고 있다[1][2].

따라서, 윈도우즈 95나 98과 같은 대표적인 GUI(Graphic User Interface)구현 운영체제의 경우, 실시간으로 사용자가 시스템의 시간을 멀티 작업 할 때도 영향을 받지 않고 분할하는 것이 매우 어렵기 때문에, 실제 장치를 제어하는 목적으로 사용하지 못하고, 뛰어난 그래픽 지원환경을 바탕으로 데이터의 획득 및 그래픽 디스플레이(graphic display)에 주로 사용되고 있다[3]. 또한 GUI 환경 하에서 실시간 제어를 위한 상용 프로그램의 경우, 실시간으로 샘플링(sampling)을 지원하기 위한 고가의 하드웨어적 인터페이스 카드(Interface card)가 부가적으로 설치 되어야만 실시간으로 실제시스템

을 제어 가능하도록 구성되어 있다[4].

본 연구에서는 GUI 환경하에서 실시간으로 실제 시스템의 제어를 위하여 윈도우즈 운영체제 기반하에서 작동하는 가상머신장치(VxD : Virtual Machine Device)를 이용하여 GUI 환경 하에서 별도의 하드웨어 장치 없이 실제장치를 제어할 수 있는 기법에 대하여 연구하였으며, GUI 특성상 다중 처리 과정시 발생할 수 있는 샘플링(Sampling)상에서 발생가능한 오차 및 지연 등의 문제점을 부하실험을 통하여 타당성을 고찰하였다.

II. 이론적 배경

기계시스템을 컴퓨터를 이용하여 제어하기 위해서는 시스템에 적절한 샘플링 시간(Sampling Time)의 선정이 매우 중요하며, 이러한 샘플링 시간이 어떤 요인에 의해서도 변화가 있어서는 안된다. 이러한 샘플링 시간을 컴퓨터 사양에 무관하게 정확히 맞추어 주기 위해서 제어전공자는 인터럽트 방식을 주로 사용하고 있다. 제어전공자들에게서 일반적으로 널리 사용되고 있는 도스(Dos)의 경우, 하드웨어 인터럽트가 발생하면 레지스터(register)를 푸쉬(push)하고 해당 인터럽트로 분기한다. 이러한 도스의 경우 80×86의 실체모드(real mode)에서 동작하며, 단일작업 운영체제(single task operating system)이기 때문에 시간 지연 없이 응용프로그램에서 사용할 수 있다. 그러나 윈도우즈는 기본적으로 다중작업 운영체제(multi task operating system)이기 때문에 많은 프로세스(process)를 동시에 혹은 산발적으로 처리하여야 하며, 하나의 프로세스로 하여금 인터럽트를 독점하도록 둘 수 없다. 이러한 원인으로 운영체제가 임의적으로 순간순간 발생하는 프로세서를 수행하는 과정에서 많은 시간 지연이 발생한다. 윈도우즈는 80x86 CPU의 보호모드(protected mode)에서 동작하는 운영체제이다. 보호모드에서는 프로세스별로 권한을 설정하여, 권한이 낮은(Ring-3) 프로세스가 권한이 높은(Ring-0) 하드웨어나 프로세스에 조작을 가하는 것을 금지하고 있다. 윈도우즈98에서의 응용프로그램은 그림 2에서 나타낸 바와 같이 Ring-3로서 수행되는 프로세스이다. 따라서, 그림 1에서 나타낸 윈도우즈 운영체제

접수일자 : 2001. 3. 30., 수정완료 : 2001. 9. 13.

장성욱 : 부산대학교 기계공학부 대학원(marble@pusan.ac.kr)

이진걸 : 부산대학교 기계공학부(leejik@pusan.ac.kr)

아래에서 윈도우즈가 허용하지 않는 하드웨어를 직접 조작할 수 없다. 즉 입출력조작, 메모리조작 명령 등으로 임의의 하드웨어를 조작하거나 액세스 할 수 없다. 따라서 윈도우즈 운영체제 혹은 다른 Ring-0 프로세스에 조작을 의뢰하여야 하며, 이러한 이유로 IBM호환 개인용 컴퓨터의 경우 키보드, 마우스, 타이머, 카운터 등에서 발생하는 인터럽트(Interrupt)를 직접 받을 수 없고, 운영체제를 거쳐 간접적으로 받게 된다. 이러한 과정에서 시간지연이 발생하고, 실시간성을 보장받을 수 없게 된다.

윈도우즈에서는 권한이 높은 하드웨어에 직접 조작을 하고자 할 경우, 하드웨어와 같은 레벨의 운영체제를 거치거나 Ring-0의 레벨에서 수행되는 가상머신장치에 이용할 수 있다[5]. 그림 3에서 나타난 바와 같이 윈도우즈 가상머신 장치(VxD)는 Ring-0에서 수행되는 프로세스로서 비록

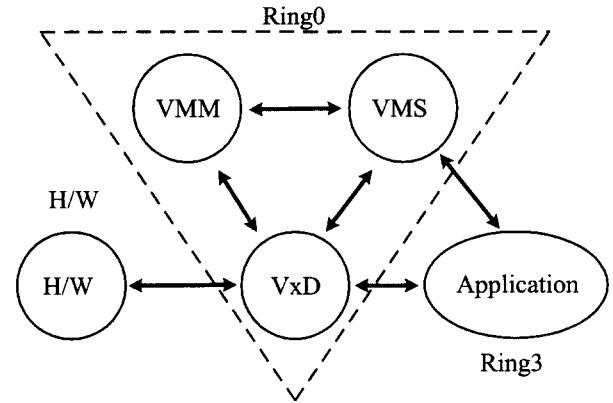


그림 3. 윈도우즈 운영체제에서 VxD의 작용.
Fig. 3. The performance of a VxD on the windows operating system.

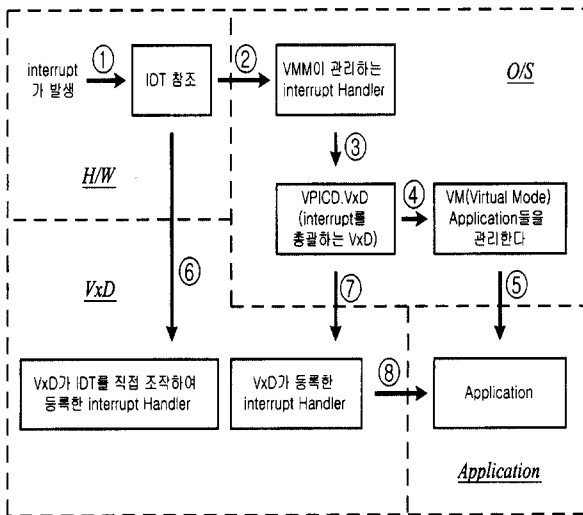


그림 1. 윈도우즈에서 인터럽트 발생시 제어 이행.
Fig. 1. The control process of the windows over the interrupt.

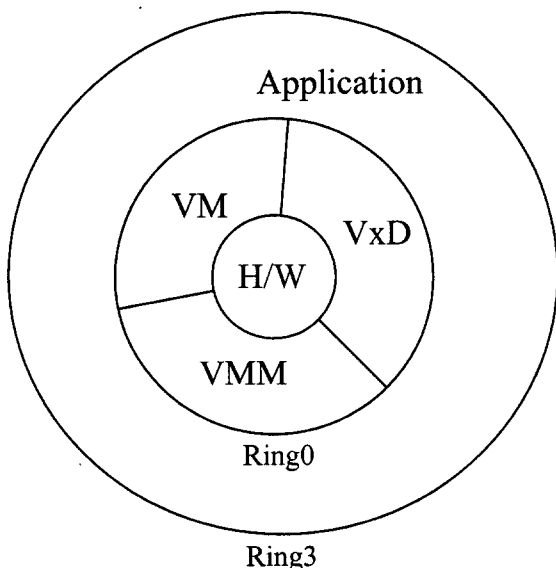


그림 2. 윈도우즈 운영체제.
Fig. 2. The schematic diagram of the windows operating system.

윈도우즈가 허용하지 않는 하드웨어라도 임의로 조작할 수 있으며, 하드웨어에서 발생하는 인터럽터를 윈도우즈 운영체제를 거치지 않고 우선적으로 받을 수 있다. 그러므로 기존의 도스와 같은 환경에서 주로 행하였던 임의의 하드웨어 조작이나, 정확한 샘플링 시간이 요구되는 물리시스템의 제어 및 실시간성이 요구되는 시스템에 가상머신장치를 적용할 수 있다. 그림 2에서 나타난 바와 같이 가상머신장치는 윈도우즈 운영체제를 구성하는 핵심 코드들이며, 윈도우즈 운영체제는 VMM(Virtual Machine Manager)를 위주로 하여 VxDs, VMs(Virtual Machine), 응용프로그램으로 구성된다.

III. 실험 장치의 구성 및 방법

가상머신장치를 이용하여 윈도우즈(Windows)환경에서 제어신호를 생성한 경우, 외부적·내부적 외란요인에 대한 안정적인 동작가능 여부를 검증하기 위하여 그림 4, 그림 5와 같이 실험장치를 구성하였으며, 시스템의 블럭선도를 그림 6에 각각 나타내었다.

그림 4, 5, 6에 나타난 바와 같이 본 시스템은 실제 제어신호를 발생시키기 위하여 펜티엄 II-450 [MHz]의 컴퓨터를 사용하였으며, 컴퓨터에 장착된 프린트 포트에 출력되는 제어신호의 정확한 측정용 위해서 150 [MHz]의 오실로스코프를 이용하였고, 측정된 데이터의 기록을 위한 프린터를 오실로스코프에 연결하였다[6].

가상머신장치를 이용한 제어신호는 PC 내부의 타이머를 이용하여 1[ms]를 주기로 내부적으로 High와 Low를 반복하

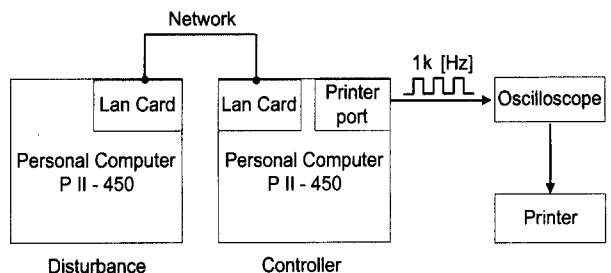


그림 4. 가상머신장치의 실험장치 계략도.
Fig. 4. The schematic diagram of the experimental setup of a VxD.

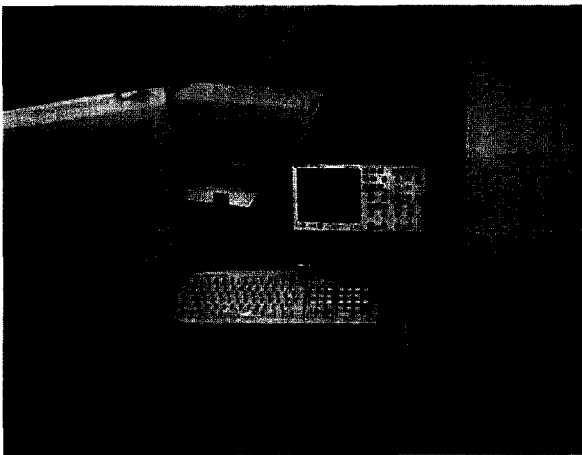


그림 5. 가상머신장치의 실험장치.
Fig. 5. The experimental setup of a VxD.

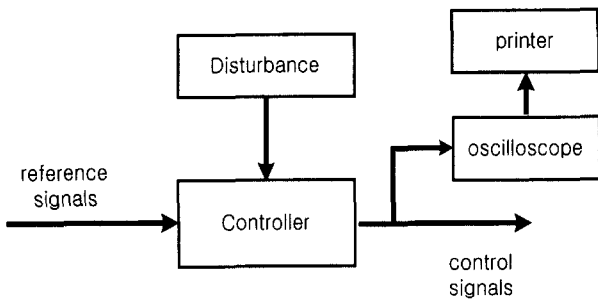


그림 6. 가상머신장치의 실험장치 블록선도.
Fig. 6. The block diagram of the experimental setup of a VxD.

여 PC의 프린터 포트로 일정한 출력을 지속적으로 발생시키도록 프로그램 하였으며, 제어신호를 발생시키는 도중에 윈도우즈 특성상 다중 프로세서를 처리하는 과정에서의 오차 및 시간지연 요소에 대한 부하 실험으로써 컴퓨터 내부에서는 음악파일 및 뮤직비디오의 상영(play) 등의 내부적 부하요소와 네트워크(network)상에서 공유파일을 복사하는 등의 외부적인 부하요소를 가하였을 때, 제어신호에 미치는 영향을 실험을 통하여 고찰하였다.

정밀한 제어를 요하는 고속의 시스템의 경우에도, 개인용 컴퓨터를 이용할 경우 컴퓨터 자체의 기준 클럭(Reference Clock)이 8k 이상을 지원하지 않으며, 도스와 같은 실제 모드환경에서도 최대 4k 이상의 적용이 힘든 실정이다. 또한 1[ms] 이상의 고속의 샘플링시간을 요하는 시스템의 경우 피드백 되는 센서로부터의 데이터를 저장하기 위한 방대한 메모리가 필요하므로 일반적인 개인용 컴퓨터를 이용한 제어시스템의 경우 1[ms] 이상의 속도로 샘플링 시간을 조정하는 경우는 극히 드물다.

따라서 일반적인 기계 및 물리적 시스템의 경우 시스템의 폐루프 샘플링 시간(Close loop sampling time)이 1[ms] 이하에서 널리 사용되고 있으며, 연속시스템이 이산시스템으로 근사화될 때 사용되고 있는 근사기법 및 샘플링 시간 선정시 고려되는 나이퀴스트 주파수(Nyquist frequency)를 고려하여 초 정밀도를 지니는 일부시스템을 제외한, 전체의 샘플링시간을 1[ms] 이하로 사용되는 기계 및 전기·전자시

스템에 적용하고자 샘플링시간 1[ms]를 기준으로 내·외적 부하를 가하였을 때 신호의 응답특성을 실험을 통하여 검토하였다[7]-[9].

IV. 실험 결과 및 고찰

그림 7은 VxD를 이용하여 1[ms]로 일정하게 제어신호를 발생시키고 동시에 뮤직비디오를 상영하여 그래픽과 음악을 동시에 부하를 걸면서 네트워크로 연결된, 다른 컴퓨터에서 1 Mbyte의 화일을 동시에 복사할 때의 신호를 나타낸 것이다. 그림 6에서 나타낸 바와 같이 컴퓨터 내·외적인 요인에 대해서 생성되는 제어신호가 정확하게 발생됨을 알 수 있다.

그림 8은 가상머신 장치를 사용하지 않고, 일반적인 윈도우즈 응용프로그램 영역에서 프로그램 하여 내·외적 부하를 가하였을 때의 결과를 나타낸 것이다. 그림 7에서 나타낸 바와 같이 윈도우즈 응용프로그램에서 VxD를 이용하지 않고 PC내의 타이머를 이용하였을 때 50[ms] 이상의 정밀도

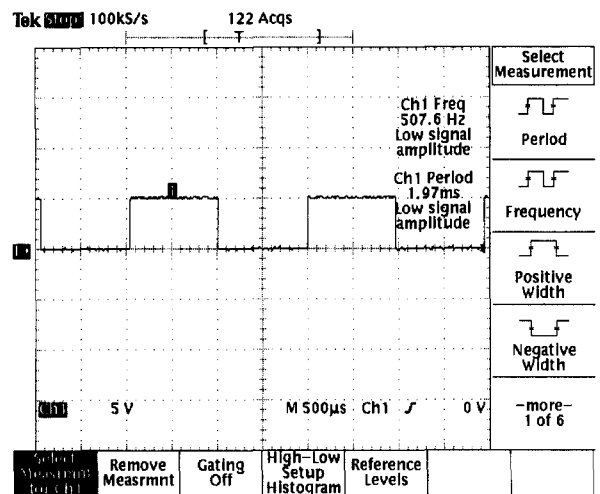


그림 7. 가상머신 장치를 이용한 실험 결과.
Fig. 7. The result of experiment using VxD.

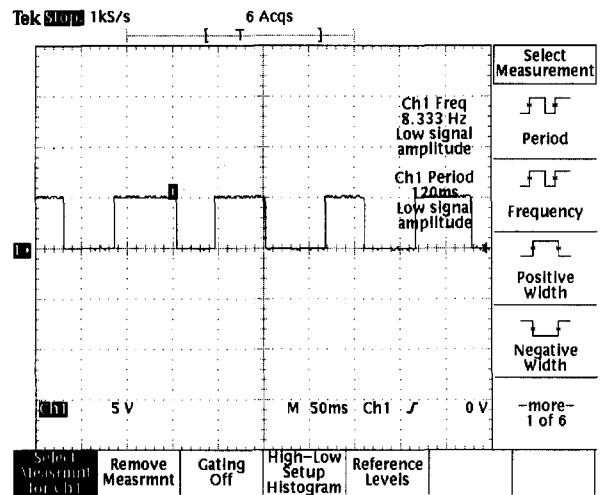


그림 8. 가상머신 장치를 사용하지 않은 실험 결과.
Fig. 8. The result of experiment without VxD.

로 샘플링시간을 조정하는 것이 힘들었다. 또한 가상머신장치를 사용하지 않았을 경우에는 제어 프로그램이 실행 중에도 다른 프로세서가 먼저 처리될 수 있으므로 외란으로 작용할 때 다른 작업이 처리되는 동안 제어신호가 내·외부적 요인에 의해 불안정하게 변동해 가는 것을 알 수 있다. 이러한 문제점은 제어시스템 자체의 안정성에 커다란 악영향을 미치므로, 이제까지 윈도우즈 환경에서 제어를 어떻게 하는 요인으로 작용되어 왔다. 이와 달리, 그림 7에서 나타낸 VxD를 이용한 제어 프로그램은 기존의 텍스트(text)환경에서의 특성을 샘플링 시간을 1[ms]를 기준으로 시험하였을 때 그대로 구현 가능함을 실험을 통해서 알 수 있다.

V. 결론

시스템을 제어하기 위한 운영체제로써 널리 사용되고 있는 도스환경은 8086 실제모드에서의 작동하므로 하드웨어의 액세스가 매우 용이하나, 텍스트 기반환경의 운영체제 특성상 그래픽 지원이 미약하고 사용자의 요구에 부응하는 막강한 그래픽환경을 지원하기에는 부족하다. 한편, 윈도우즈 환경은 뛰어난 그래픽 환경으로 일반 사용자가 손쉽게 사용할 수 있도록 하기 위한 많은 편리함을 지니고 있으며, 멀티 테스킹(multi tasking)과 같은 다중 프로세서를 지원하는 장점을 지니고 있다. 그러나 사용자가 컴퓨터의 하드웨어를 직접 액세스(access)하는 것이 어려우며, 이러한 문제점으로 인하여, 대부분의 기계 및 물리적 시스템을 실제로 제어하는 제어전공자들은 실시간 제어를 위해서 도스와 같은 텍스트 환경에서 제어를 행하고 있다. 그러나 기존의 도스 환경에서의 시스템의 제어는 사용자의 요구에 의해서 점차적으로 뛰어난 그래픽을 바탕으로 한 윈도우즈 환경에서의 실 시스템의 제어에 대한 요구가 증대되고 있다.

본 연구에서는 윈도우즈의 가상머신 장치를 이용하여 윈도우즈 환경에서 실제 시스템을 제어하기 위한 기법에 대해

여 연구하였으며, 가상머신 장치를 이용하였을 때와 일반적인 윈도우즈 응용프로그램에서의 실험 결과를 비교 고찰하였다. 또한 실제 시스템에 적용할 때 중요한 시스템의 샘플링 타임의 오차 및 지연을 시스템 사용 시 발생할 수 있는 여러 가지 부하요소를 가하였을 때의 결과를 고찰함으로써 실제 시스템에 적용 가능성을 확인하였다.

참고문헌

- [1] 장성욱, 이진걸, "윈보드 마이크로 프로세서 제어기 및 PC를 이용한 유압서보시스템의 실시간 시뮬레이터의 개발," 한국정밀공학회지, 제23권, 제10호, pp. 1702-1708, 10, 2000.
- [2] Toshihara Matsukuma, Junbo Song, Yoshihisa Ishida, and Shun-ichi Notoyama, "PID control of pneumatic cylinder using neural networks," *Journal of the Japan Hydraulics and Penematics Society*, vol. 30, no. 1, pp. 28-33, 1, 1999.
- [3] Akio Seko, Takayoshi Muto, Hironao Yamada, and Kstsunori Tanaka, "Development of simulation program for hydraulic systems," *Journal of the Japan Hydraulics and Penematics Society*, vol. 30, no. 2, pp. 420-427, 2, 1999.
- [4] User manual's, "Getting started with LabWindows / CVI," *National Instrument*, 1998.
- [5] Karen Hazzah, "Writing Windows VxDs and device drivers," *Miller Freeman Inc.*, Second Edition. 1997.
- [6] User manual's, "TDS 340A for Users," *Tektronix*, 1995.
- [7] 김종식, "선형 제어시스템 공학," 청문각, 1993.
- [8] Gene F. Frankin, J. David Powell, and Michael Workman, "Digital control of dynamic systems," *Addison Wesley*, 1997.
- [9] Sung-ouk Chang and Jin-Kul Lee, "Development of a hydraulic servo system real-time simulator using a one-board microprocessor and personal computer," *Proceedings of the 4th Asia-Pacific Conference on Control and Measurement*, pp. 128-132, July, 2000.



장 성 욱

1972년 9월 8일생. 1995년 부산대학교 정밀기계공학과 졸업. 1997년 동대학원 석사. 1998년~현재. 부산대학교 지능기계공학과 박사과정. 1998년~현재. 부산대학교 정밀정형 연구소 전임연구원. 관심분야는 지능 제어 및 유압제어, 실시간 학습제어, 강인제어.



이 진 걸

1940년 9월 10일생. 1969년 건국대 축산학과 졸업. 1973년 Kyoto대학 대학원 석사. 1976년 동대학원 박사. 1977년~현재 부산대학교 기계공학부 교수. 관심 분야는 유압제어 및 계측.