

머시닝 시스템의 실시간 제어를 위한 개방형 구조 제어기

An Open Architecture Controller for the Real-time Control of Machining Processes

°이 재 영*, 권 옥 현*, 박 재 현**

*서울대학교 전기공학부 (Tel: (02)880-7314; Fax: (02)871-7010; E-mail: {jylee, whkwon}@cisl.snu.ac.kr)

**인하대학교 자동화공학과 (Tel: (032)860-7713; Fax: (032)863-4386; E-mail: jhyun@rcsl.inha.ac.kr)

Abstract This paper presents an open architecture controller (OAC) for machining systems and describes the OAC testbed at Seoul National University. Because our OAC is designed for fully open systems, it does not depend on any specific hardware or software components. This openness includes software reusability which enables integration of a wide range of monitoring and control features. In addition to openness, our OAC system provides guaranteed real-time performance, an important requirement for advanced manufacturing.

Keywords Open Architecture, Machining Process, Openness, Real-time Controller, Distributed System

1. 서론

머신 제어기의 가동 유지 비용과 함께 제어 시스템 자체의 개발 비용을 절감할 수 있는 유연성있는 차세대 생산 시스템을 개발하기 위해서는 개방형 구조 (open architecture) 개념을 기반으로 하드웨어 및 소프트웨어의 구조를 정의해야 한다. 본 논문에서는 머시닝 시스템을 위한 개방형 구조 제어기 (Open Architecture Controller, OAC)를 제안하고, 서울대학교 제어계측신기술 연구센터에서 연구 중인 개방형 구조 제어기 테스트베드를 소개한다.

개방형 시스템에 대한 일반적인 정의를 내리기는 매우 어려운 일이나, 공급자 중립성 (vendor neutrality)과 구성요소 통합성 (component integrability)은 개방형 시스템의 가장 중요한 특성이라고 할 수 있다. 개방형 시스템은 특정 독점 공급자와는 무관한 표준안에 기초하여 설계될 수 있어야 하므로 공급자 중립성이 필요하다. 또한 구성요소 통합성은 개방형 시스템이 이식성 (portability)과 확장성 (expandability)을 가질수 있게 한다. 이밖에도 개방형 시스템에 대한 연구들에서는 자체의 독특한 제약 조건이나 요구 사항을 정의하고 있는데, 상호운용성 (interoperability), 비율성 (scalability), 상호교환성 (interchangeability), 모듈성 (modularity), 연장성 (extensibility), 재사용성 (reusability), 호환성 (compatibility) 등을 언급하고 있다. 응용 프로그램에 있어서 개방형 시스템은 제어나 감시 기능에 대한 새로운 요구 사항을 모듈화된 방식으로 통합할 수 있게 해야 한다. 한가지 예로서 온도 보상 제어를 수행시키고자 하는 경우, 시스템은 기존의 온도 보상 연구 결과 (온도 센서, 열적 모델, 보상 알고리즘 등)를 최소의 노력으로 제어기 내부로 통합시킬 수 있어야 한다. 따라서, 개방형 시스템은 공급자 중립성을 위해 호환성과 상호운용성을, 그리고 구성요소 통합성을 위해 모듈성과 재사용성을 가져야 한다.

현재까지의 개방형 구조 제어기에 대한 연구는 크게 두가지 부류로 구분될 수 있다. 그 한 부류는 상용 제품들간의 호환성을 위주로 하는 기업체들에 의해 주도되는 연구이고, 다른 부류는 기초 연구기관에 의해 주도되는 연구이다. 이 두가지 부류가 비록 그 접근 방식은 다르다 하더라도, 개방형 구조 제어기를 개발하는 최종의 목표는 개방형 생산 시스템을 구현하는 것이다. OSACA (Open System Architecture for Controls within Automation systems) [1, 2, 3]는 개방형 구조 제어기에 관한 대규모의 프로그램으로서

, 통신망, 응용 소프트웨어 및 하드웨어 전반에 대한 모든 표준화 작업을 포함한다. 이 프로젝트는 독일의 Stuttgart 대학의 연구소가 중심이 되어 유럽의 8개 기업, 3개 연구기관이 참가하여 실시되었다. 그 목적은 특정 벤더에 의존하지 않는 개방형 제어 장치의 참고 구조 (reference architecture)를 정하는 것이다. 여기서 제어 장치는 CNC 장치를 비롯하여 로봇, PLC, 셀의 제어 장치를 포함한다. 구조 (architecture)란 소프트웨어의 구조를 의미하고 하드웨어에 의존하지 않는 시스템을 고려하고 있다. 객체 지향형 접근 방식을 취하여 제어 장치의 기능을 언제나 재편성할 수 있는 구성 (configuration) 시스템을 마련하는 점이 특징이다. 연구 프로젝트로는 HOAM-CNC (Hierarchical Open Architecture Multi-Processor CNC system) [4]와 다이 몰드 (die and mold) 공정을 위한 개방형 제어기 [5], UMOAC (the University of Michigan Open Architecture Controller) [6] 등에서 다양한 연구결과를 발견할 수 있다. 특히 HOAM-CNC [4]에서는 PC-AT와 ISA 버스를 이용한 플랫폼을 사용하고, 측제어에는 DSP를 사용하여 적응 제어, 공구 결함 검출 및 체터 진동 회피의 기능을 동시에 실현하였다.

한편, 개방형 제어기는 이러한 개방성 요구뿐만 아니라 보장된 실시간 동작 성능을 제공해야 한다. 실시간 동작 성능은 자동화된 생산 시스템에서 기본적으로 요구되는 사항이기 때문이다. 생산 시스템에서 하나의 제어 작업은 기계 상태의 감지, 몇 단계의 제어 알고리즘, 구동기의 제어 등과 같은 다수의 하부 작업으로 구성된다. 이러한 작업 중 일부는 주기적으로, 나머지는 비주기적으로 수행된다. 그러나, 모든 작업은 어떠한 형태로든 시간 제약 (timing constraint)을 만족해야 한다. 생산 시스템과 같은 실시간 시스템에서는 시간 제약을 만족하지 않는 공정 감시 혹은 제어는 의미가 없게 된다. 그러나, 기존의 개방형 제어기에 관한 연구에는 이러한 실시간 제약에 대한 결과가 없다. 따라서 차세대용 생산 시스템의 제어기를 개발하기 위해서는 두가지의 목표를 모두 달성해야 한다. 첫번째 목표는 발달된 각종 감시 및 제어 기술을 모듈화된 방법으로 통합할 수 있는 유연한 개방형 시스템을 구성하는 것이다. 두번째 목표는 서로 다른 단계의 계층구조적 작업의 응답 시간을 보장하는 시스템을 구성하는 것으로, 이것은 생산 시스템의 공구 응용 작업들간의 실시간 인터페이스를 포함한다. 여기에 덧붙여 생산 공정의 무인화, 복합화, 시스템화 추세에 부응하기 위해서는 통신망의 이용이 필수로 되어 분산 시스템을 지원하는 구조이어야만 한다.

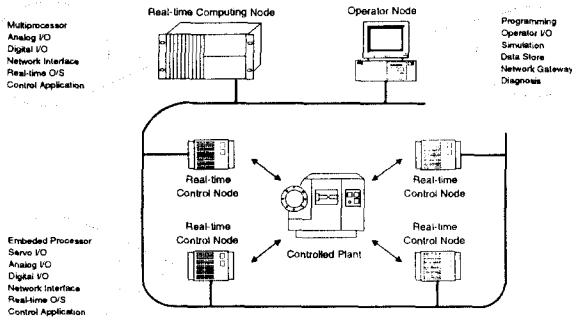


그림 1. 개방형 구조 제어기의 하드웨어 구성
Fig. 1. OAC Hardware Configuration

지금까지의 연구 결과를 종합하면, 이 두가지의 목표와 함께 분산 시스템을 지원하는 구조의 개방형 제어기로는 OSACA의 참고 구조를 제외하고는 UMOAC가 적절하다. 따라서, 기본적인 하드웨어 및 소프트웨어의 구조는 UMOAC를 참고로 한 개방형 구조 제어기를 설계하도록 한다. 본 논문에서는 이같은 개방형 제어기의 목표를 달성하기 위한 개방형 구조 실시간 제어기를 소개한다. 2장에서는 개방형 구조 제어기의 하드웨어 구조를 제시하고, 3장에서 소프트웨어의 구조를 설명한다. 4장에서는 서울대학교 제어계측신기술 연구센터에서 연구중인 개방형 구조 제어기의 테스트베드를 소개하고, 5장에서 결론을 맺는다.

2. 개방형 구조 제어기의 하드웨어 구성

본 논문에서 제안하는 개방형 구조 제어기는 다음의 2가지 기본 요구를 만족하도록 설계되었다. 2가지의 기본 요구는 첫째, 개방성으로서 다양한 머신 모니터링 기술과 머신 제어 기술을 모듈화시킨 방법으로 통합할 수 있게 하기 위해서이다. 둘째는 실시간 운전으로서 생산 시스템에서 우선 순위가 서로 다른 작업들의 동작 시간을 보장할 수 있어야 한다는 것이다. 이 두가지의 기본 요구를 만족하기 위한 개방형 구조 제어기의 기본 하드웨어 구성은 분산 시스템으로서, 각 연산 노드 (processing node)는 실시간 통신망을 통해 연결된다. 이러한 분산 시스템은 다양한 하드웨어 구성을 가능하게 한다. 즉, 실제의 시스템 구성에 있어 소형의 마이크로 제어기로부터 중형 컴퓨터에 이르기까지 어떠한 시스템이라도 연산 노드가 될 수 있다. 그러나, 그 크기나 기능성과는 관계없이 각 연산 노드는 통일된 소프트웨어 계층구조내에서 실시간성을 만족하는 통신 호환성을 유지하며 동작하게 된다. 공급자 중립성을 유지하면서 이종기간의 시스템 구성을 가능하게 하기 위해 본 개방형 구조 제어기는 특정 하드웨어를 정의하지 않는다. 그러나, 각 연산 노드는 VME 버스같은 산업 표준의 모듈식 장비들로 구성된다. 그림 1은 제안된 개방형 구조 제어기 하드웨어 구성의 한 예이다. 본 제어기의 하드웨어는 운영자 노드, 실시간 연산 노드, 실시간 제어 노드의 3가지 노드로 구성된다. 운영자 노드 (operator node)에서는 프로그래밍이나 비실시간 플랜트 감시 등의 비실시간 작업이 수행된다. 이와는 달리 실시간 연산 노드 (real-time computing node)는 실시간 데이터 수집, 실시간 진단, 실시간 스케줄링, 실시간 제어 등과 같은 실시간 제어 및 감시 작업을 수행한다. 또한, 실시간 제어 노드 (real-time control node)에서는 서보 단위의 제어나 데이터 수집같은 고정밀도 실시간 작업이 수행된다.

본 제어기와 같은 분산 시스템에서 각 연산 노드간의 통신 채널은 그 개방성과 더불어 실시간 성능에 중요한 영향을 미친다. 생산 자동화를 위한 통신 프로토콜 (예: Mini-MAP, Ethernet)에는

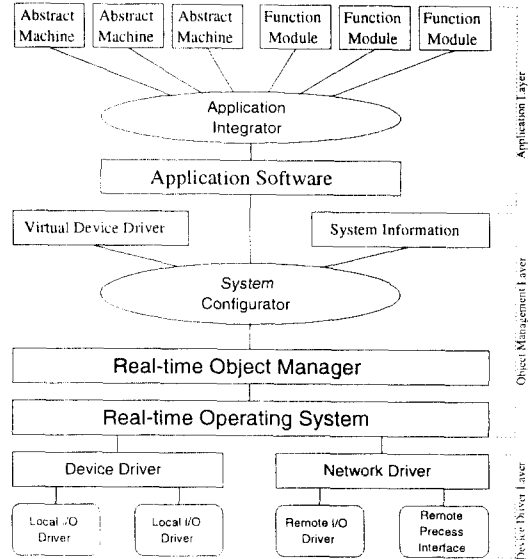


그림 2. 개방형 구조 제어기의 소프트웨어 계층구조
Fig. 2. OAC Software Hierarchy

여러가지가 있으나, 본 제어기에서는 연산 노드간의 실시간 통신 링크로서 FIP (Factory Instrumentation Protocol)를 채택하였다. 주기적인 실시간 메시지를 단일 통신망을 통해 제한된 시간안에 전송기에 적절하기 때문이다 [7]. 또한 FIP 버스는 기본적으로 재전송과 실시간 통신에 대한 응답을 하지 않아도 되고, 통신망을 통해 실시간으로 전송되어야 하는 변수들의 값과 전송 특성을 미리 알 수 있으며, 반응 시간이 제한되는 분산 버스 수집 체계를 제공하기 때문에 기존의 통신 프로토콜들보다 실시간 범위를 만족하는데 있어 보다 나은 성능을 보인다 [8, 9].

3. 개방형 구조 제어기의 소프트웨어 구조

3.1 소프트웨어의 계층구조

하드웨어 구성과는 무관하게 이식성 높은 응용 프로그램을 작성하는 것이 가능하게 하기 위해 제안된 제어기의 소프트웨어 계층구조는 3개의 계층, 즉 응용 계층 (Application Layer), 객체 관리 계층 (Object Management Layer), 그리고 장치 구동자 계층 (Device Driver Layer)으로 구성된다. 그림 2는 제안된 개방형 구조 제어기의 소프트웨어 계층구조를 나타낸다. 응용 계층은 추상적 머신 모델 (Abstract Machine Model), 기능 모듈 (Functional Module) 및 응용 프로그램 (Application Program)으로 구성된다. 응용 프로그램은 사용자 인터페이스, 프로그래밍, 모니터링을 포함하는 최상위 소프트웨어이다. 이 응용 프로그램이 이식 가능하게 하기 위해 하드웨어 구성과 독립적인 추상적 머신 모델과 모듈 구성이 가능한 기능 모듈을 사용한다. 기능 모듈과 가상 머신 모델은 응용 통합자 (Application Integrator) 소프트웨어에 의해 관리되는데, 응용 통합자 소프트웨어는 특정 응용 소프트웨어용으로 작성된 기능 모듈이 다른 응용 소프트웨어용으로 재사용되거나 확장될 수 있게 해준다.

추상적 머신 모델은 실제의 머신 하드웨어에 대응한다. 이 모델에는 머신 자체의 사양과 함께 실행 시간중 수집되어야 할 데이터의 사양이 포함되어 있다. 실행 시간 데이터는 실시간 객체 관리자 (Real-time Object Manager)에 의해 관리되고, 사전 정의된 응답 시간을 지킬 수 있다. 응용 프로그램과 기능 모듈은 추상적 머신 모델과 상호작용하므로 하드웨어와 분리되고, 따라서 각 모듈은 모

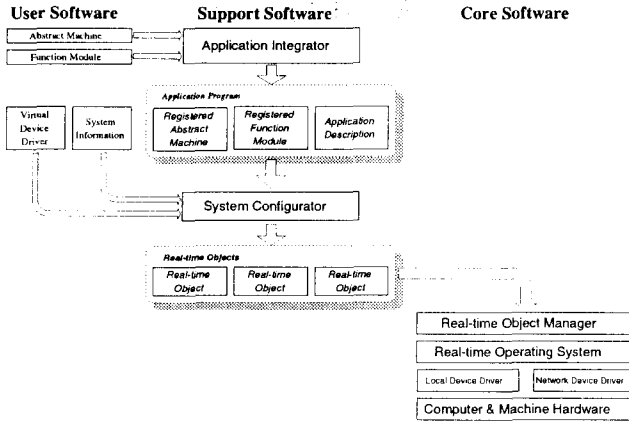


그림 3. 개방형 구조 제어기의 소프트웨어 구성요소
Fig. 3. OAC Software Structure

들성 (modularity)과 재사용성 (reusability)을 가질 수 있다.

소프트웨어 계층구조의 두번째 계층은 객체 관리 계층으로서, 가상 장치 구동자 (Virtual Device Driver), 시스템 정보 (System Information), 시스템 구성자 (System Configurator), 실시간 객체 관리자 (Real-time Object Manager), 실시간 운영 체제 (Real-time Operating System)으로 구성된다. 시스템 구성자의 주 기능은 기능 모듈과 추상적 머신 모델을 포함하는 하드웨어 독립적인 응용 소프트웨어와, 제어 대상 플랜트나 원격 연산 모듈 (remote processing module)같은 실제의 하드웨어를 연결시키는 것이다. 제어 대상 플랜트가 국지 입출력 (local I/O) 인터페이스 하드웨어와 연결된 경우, 먼저 가상 장치 구동자가 사용된 후 특정 하드웨어 장치 구동자가 인터페이스를 담당하게 된다. 제안된 개방형 구조 제어기에서 사용된 가상 장치 구동자 개념은 하드웨어 단계에서의 상호운용성 (interoperability)을 제공한다. 또한 제어 대상 플랜트가 원격 연산 모듈과 연결된 경우나 한 기능 모듈이 원격 연산 모듈로부터의 데이터를 사용하고자 할 경우, 시스템 구성자는 통신망 구동자 (network driver)를 사용한다. 이때 시스템 구성자에 의해 제공되는 이러한 시스템의 연결은 응용 프로그램의 작성과는 무관하므로, 소프트웨어 모듈성과 재사용성을 유지할 수 있다.

실시간 객체 관리자는 시스템 서비스를 객체 지향적 머신 제어 응용 프로그램 영역에 알맞게 조정한다. 이 서비스는 실시간 운영 체제 서비스를 확장하고, 작업과 자원의 특정 영역 스케줄링을 포함한다. 실시간 객체 관리자는 상용의 실시간 운영 체제의 상위에 접속될 수 있도록 설계되었다.

계층구조의 세번째 계층은 장치 구동자 계층으로서, 유일하게 하드웨어와 관련이 있는 부분이다. 국지 입출력 구동자나 원격 입출력 구동자가 동일한 인터페이스 프로토콜을 제공하기 때문에, FIP 버스를 통한 원격 데이터 역시 국지 데이터처럼 취급될 수 있다.

3.2 소프트웨어의 구성요소

제안된 개방형 구조 제어기의 소프트웨어는 다음의 그림 3과 같이 사용자 소프트웨어 (user software), 지원 소프트웨어 (support software) 및 핵심 소프트웨어 (core software)로 구성된다. 사용자 소프트웨어는 사용자가 작성해야 하는 소프트웨어로서, 추상적 머신 모델, 기능 모듈, 가상 장치 구동자 및 시스템 정보 파일이 여기에 해당된다. 추상적 머신 모델, 기능 모듈 및 가상 장치 구동자는 문서 파일 (text file) 형태로 작성되며, 응용 대상에 따라 재사용과 재구

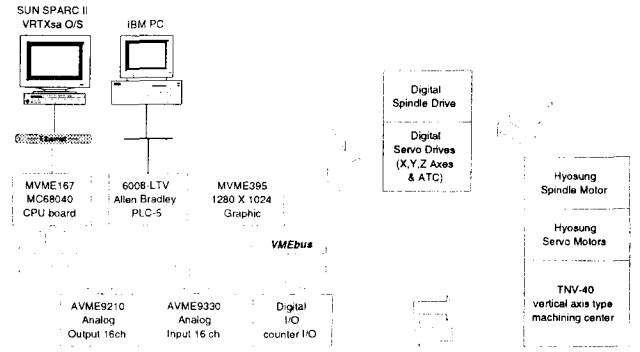


그림 4. 개방형 구조 제어기 테스트베드의 하드웨어 구성
Fig. 4. OAC Testbed Hardware Configuration

성이 가능하다. 그러나, 시스템 정보 파일은 제어 대상 시스템에 따라 변동이 있을 수 있으므로 재사용에는 제약을 받게 된다.

지원 소프트웨어에는 응용 통합자와 시스템 구성자가 포함된다. 응용 통합자는 등록된 추상적 머신 모델과 기능 모듈을 이용하여 동적 라이브러리 (dynamic library) 형태의 추상적 머신 모델 라이브러리 및 기능 모듈 라이브러리를 생성한다. 또한 응용 프로그램의 전반적인 동작을 기술하는 응용 기술 (application description) 파일을 생성하며, 동적 라이브러리들과 함께 응용 프로그램을 구성한다. 이 응용 프로그램은 가상 장치 구동자, 시스템 정보 파일과 함께 시스템 구성자에 의해 통합되어 실제의 작업 단위인 실시간 객체들로 분할된다. 이 실시간 객체들은 핵심 소프트웨어의 실시간 객체 관리자 자료로 전달된다. 여기서, 시스템 구성자는 응용 기술 파일에 등록된 실시간 작업 기준을 검토하여 시스템 정보를 기준으로 그 가능성을 판단하게 된다. 시스템 정보 파일에는 제어기의 하드웨어 구성, 성능, 연산 노드수 등의 설치 환경이 기록되며 시스템 구성자가 응용 프로그램을 최적의 실시간 객체들로 분할할 수 있게 한다.

핵심 소프트웨어에는 실시간 객체 관리자, 실시간 운영체제, 국지 장치 구동자, 통신망 장치 구동자가 해당된다. 핵심 소프트웨어는 시스템 구성자에 의해 분할된 실시간 객체들을 실시간 운영체제의 작업으로 등록하여 장치 구동자와 제어 하드웨어에 의해 수행될 수 있도록 한다.

4. 개방형 구조 제어기 테스트베드의 구성

본 설계를 평가하기 위해 그림 4와 같은 개방형 구조 제어기의 테스트베드를 구성하였다. 이 테스트베드는 3축 메시닝 센터, 상용의 CNC 제어기 (Fanuc사), PC 기반의 개방형 구조 제어기, VME 버스용 PLC, VME 버스 기반의 개방형 구조 제어기, 그리고 FIP 버스 기반의 개방형 구조 제어기로 구성된다. 현재의 테스트베드는 서울대학교 제어계측신기술 연구센터의 차세대 CNC 제어기에 대한 연구활동의 일환으로서 설치된 플랫폼이다. 효과적인 연구가 수행되기 위해서는 개방적이고 쉽게 수정가능한 제어 시스템이 필요하지만, 기존의 상용 CNC 시스템에서는 이러한 기능이 제공되고 있지 않다. 최근 공작기계의 기술이 고속화, 고정밀화, 복잡화 향상 가공 위주로 발전해 왔던 반면에 CNC 장치를 생산하는 CNC 장치 제조 메이커는 CPU의 32-비트화 및 RISC 엔진 탑재 등의 수단을 통해 그런 목적이 달성되도록 지속적인 개발에 막대한 예산과 인력을 투입하면서도 공작기계 사용자들로부터 가격 저가화 요청에

직면하게 되었다. 또한, 생산 현장에서는 기계나 CNC 장치가 보유하고 있는 기능을 쉽게 배우고 더욱 더 효율적으로 조작할 수 있도록 조작자를 위해서 다양한 기능을 갖춘 CNC 장치를 필요로 한다. 다양한 기능이란 (1) 가공 준비 시간을 단축해 주며, (2) CNC 장치를 배우기 쉽고 사용하기 쉽게 해주며, (3) 뛰어난 현장 조작성을 제공하며, (4) 기계나 조작 장치가 생산 공정에 적합해야 한다는 것이다. 기존의 CNC 장치가 공작기계 전용으로 각 메이커마다 소프트웨어의 전용기계어 및 CPU를 포함한 전용 하드웨어를 채택하고 있기 때문에, 상이한 CNC 장치간 접속이나 FMS 혹은 CIM에서와 같은 상위 컴퓨터 접속 통신을 위한 프로그램의 개발이 어렵고 많은 시간이 소요되는 등, 적용상의 제약때문에 CNC 장치의 개방화가 절실히 요구된다. 설치되어 있는 개방형 구조 제어기의 하드웨어 플랫폼은 PC 기반으로 구성되어 있으며, 추후 하드웨어 플랫폼을 VME 버스 시스템과 FIP 버스 시스템으로 확장시킬 예정이다.

일반적으로 CNC 제어기의 소프트웨어는 적응 제어 모듈, 오차 보상 모듈, 보간 모듈, 스핀들 제어 모듈, 서보 제어 모듈, 및 센서 인터페이스 모듈 등으로 구분된다. 또한 이러한 소프트웨어 모듈들은 관리 프로그램에서 통제되는데, 이 때 만족해야 하는 시간 제약은 보통 수 msec에서 수 초까지이다. 따라서, 개방형 구조 제어기 테스트베드의 소프트웨어는 하위 단계의 제어 모듈들과 상위 단계의 관리 소프트웨어로 구성된다. 또한 기존의 상용 CNC 제어기에 비해 개방적이므로, 보간 (interpolation), 적응 제어, 오차 보상 등의 다양한 서보 제어 알고리즘을 구현하고, 수정, 시험하기에 용이하도록 구성되었다.

5. 결론

지금까지 머시닝 시스템을 위한 실시간 개방형 구조 제어기의 구조와 설치 시험중인 테스트베드에 관해 언급하였다. 현재 프로젝트는 기초 단계에 있지만 머시닝 시스템의 개방형 구조 제어에 대한 본 연구는 몇가지 측면에서 중요한 의미를 갖는다. 먼저, 제안된 시스템은 개방적이므로 특정한 하드웨어나 소프트웨어 구성 요소의 영향을 받지 않는다. 그리고, 제안된 시스템은 생산 시스템에서의 중요한 요구 사항인 실시간 동작을 보장한다. 또한, 다양한 감시 및 제어 기능들을 모듈화된 방법으로 시스템에 통합할 수 있다. 앞으로 시험중인 테스트베드를 지속적으로 시험하는 한편, 현재 PC 기반인 연산 노드를 다양한 시스템으로 확장하는 작업이 필요할 것이다.

참고문헌

- [1] G. Pritschow, "Automation Technology - On the Way to an Open System Architecture," *Robotics & Computer-Integrated Manufacturing*, vol. 7, no. 1/2, pp. 103-111, 1990.
- [2] G. Pritschow, C. Daniel, G. Junghans, and W. Sperling, "Open System Controllers - A Challenge for the Future of the Machine Tool Industry," *Annals of the CIRP*, vol. 42, no. 1, pp. 449-452, 1993.
- [3] *Open System Architecture for Controls within Automation Systems EP 6379 & EP 9115 OSACA phase I and II Project Report*, Stuttgart, 1996.
- [4] Y. Altintas and W. K. Munasinghe, "A Hierarchical Open-Architecture CNC System for Machine Tools," *Annals of the CIRP*, vol. 43, no. 1, pp. 349-354, 1994.
- [5] T. Bailey, Y. Ruget, A. Spence, and M. A. Elbestawi, "Open Architecture Controller for Die and Mold Machining," *Proceedings of ACC 95*, vol. 1, pp. 194-199, 1995.
- [6] J. Park, S. Birla, K. G. Shin, Y. Shan, G. Ulsoy, and Y. Koren, "An Open Architecture Testbed for Real-Time Monitoring and Control of Machining Processes," *Proc. of ACC 95*, vol. 1, pp. 200-204, 1995.
- [7] P. Pleinevaux and J. -D. Decotignie, "Time Critical Communication Networks: Field Buses," *IEEE Network*, vol.2, no. 3, pp. 55-63, 1988.
- [8] French Association for Standardization, *FIP Bus for Exchange of Information Between Transmitters, Actuators and Programmable Controllers*, NF C46, 601-607, 1990.
- [9] G. Cena, C. Demartini, L. Durante, and A. Valenzano, "A FIP Prototype Network for Numerical Control Applications," *Proc. of IECON*, vol.3, pp. 1196-1199, 1994.