

FMS 구현을 위한 컴퓨터 제어 체계

(The Computer Control Hierarchy for the Implementation of FMS)

조 광 문 *
김 태 운 **

Abstract

FMS is the system which improves the productivity by making production lines flexible. FMS is composed of manufacturing system, material handling system, and softwares to control them. FMS is a complicate system; therefore, computer is required to control FMS efficiently. In this paper the concepts of FMS are shortly reviewed and the computer software and programming strategy to control FMS are presented.

1. 서 론

오늘날 세계 시장 경제의 여러가지 제약점들로 인하여 각 제조 회사들은 이에 대한 대응책에 고심하고 있다. 여기에 호응하여 생산 공장의 자동화 개념이 등장하게 되었다. 공장 자동화의 실현과정에서 부분적인 간이 자동화가 진행되게 되었고 나아가 전 공장의 자동화의 필요성을 인식하게 되었다. 그렇게 하여 FMS(flexible manufacturing system)의 개념이 나오게 되었다.

FMS란 생산 제품에 따라 유연성 있게 제조 라인을 운영하여 생산성을 높일 수 있는 시스템으로서 자동화된 자재 반송 시스템(automatic material handling system)에 의해서 연결된(interconnected) NC 기계(또는 자동화된 워크스테이션)들의 모임이다. 여기에서 사용되는 모든 기계들과 작업 조작 시스템(work handling system)들을 제어(control)하기 위하여 컴퓨터의 사용이 요구된다. FMS는 그 활동(activity)들을 감시(monitor)하고 제어하는 데 아주 많은 컴퓨터 하드웨어와 소프트웨어를 요구하는 복잡한 시스템이다. 즉, FMS란 합리화된 유연성(flexibility) 또는 다양성이 있는 가공 시스템으로서 핵심 부분은 자동화 된 가공 시스템, 이

* 고려대학교 전산학과 대학원

** 고려대학교 전산학과 조교수

시스템과 유기적으로 연결된 자동 반송 시스템 그리고 이들을 제어하는 통합적인 소프트웨어 기능이 요구된다.

FMS의 복잡성 때문에 이 시스템을 구현하기 위해서는 시스템이 제대로 동작하도록 조정하는 제어 체계가 구성되어야 한다. 제어 체계는 보통 몇 단계로 구성되는 데, 이 단계들 사이의 정보와 제어의 흐름(flow)이 FMS의 구축에 매우 중요한 요소가 된다.

본 연구에서는 먼저 FMS의 전반적인 부분을 살펴보고, 그 중에서 FMS를 구현하는 데 있어서 필수 요소인 정보와 제어의 흐름을 조정하는 컴퓨터의 소프트웨어 제어(software control) 기법을 제시하였다.

2. FMS의 정의와 구현 효과

FMS의 개념이 출현하게 된 가장 큰 배경은 각 산업체의 경제적인 동기라고 할 수 있다. 효과적인 생산 시스템(manufacturing system)을 구축함으로써 생산성의 향상은 물론 산업체의 장기적인 경쟁력의 향상을 지속시킬 수 있다는 것이다.

생산 시스템은 크게 다음 세가지의 속성으로써 정의될 수 있다.

- a) 적응력(adaptability : A) : 시스템이 새로운 제품의 생산이나 새로운 생산 기법에 적응하고 그 생산 기법의 잇점을 얻을 수 있는 능력
- b) 재구성력(reconfigurability : R) : 생산 제품의 수요 변화에 대응(react)할 수 있는 시스템의 능력
- c) 프로그램 능력(programmability : P) : 인간의 관여 없이도 시스템을 제어할 수 있는 능력

이 성분들은 3차원의 공간(A-R-P space)을 형성하고, 이 공간 내에서 생산 시스템은 속성 벡터(attribute vector)로 표시될 수 있다. 모든 생산 시스템이 어느 정도의 적응력과 재구성력 성분을 갖고 있지만, 모두다 프로그램 능력 속성을 갖는 것은 아니다. 현재 대부분의 생산 전략이 시스템의 무인화 능력을 강조하기 때문에 요즘 설계되는 대부분의 생산 시스템은 어느 정도의 프로그램 능력 속성을 갖는다. 컴퓨터와 통신, 그리고 각종 센서 기술의 발달로 인하여 생산 시스템의 프로그램 능력 성분의 정도가 높아질 수 있다. 일반적으로, 생산 셀을 프로그래머블(programmable)하게 할 경우에, 셀이 매우 작거나 가공 공정이 단순하지 않는 한, 적응력이 감소하게 되는 반비례 관계가 성립한다. 그러나, 프로그램 능력 속성에 따른 재구성력 성분의 변화는 반송 기기에 따라 다르게 된다.

이러한 FMS의 출현은 배치 생산 기법(batch manufacturing)에서의

전통적 제조 방법에 중요한 변화를 가져왔고 생산성의 향상을 비롯하여 다품종 생산의 효율화, 생산 관리의 합리화, 납기 단축, 재고 삭감, 노동력 감소, 좋은 기계 효율 등의 뚜렷한 잇점을 제공한다.

3. FMS의 유연성

생산 시스템의 유연성이란 변화에 따른 능동적인 대응 능력을 일컫는다. 2절에서 소개한 생산 시스템을 정의하는 세가지 성분은 생산 시스템의 유연성 공간(flexibility space)으로 간주될 수 있다. 즉, 이 세가지의 성분으로 주어진 생산 시스템의 유형과 유연성의 정도를 측정할 수 있는 것이다. 이러한 유연성은 크게 다음 세가지로 분류된다.

- 1) 상태 유연성(state flexibility) : 어떠한 상황의 변화에도 효과적인 기능을 수행할 수 있는 능력
- 2) 작업 유연성(job flexibility) : 진행되는 작업의 변화에 대응할 수 있는 능력
- 3) 기계 유연성(machine flexibility) : 기계나 워크스테이션들의 변화나 고장을 다룰 수 있는 시스템의 능력

이러한 유연성을 갖는 FMS를 구성하는 요소들은 다음 네가지로 구분된다.

- 1) 머신 툴과 관련 기기(machine tool and related equipment)
- 2) 자재 반송 시스템(material handling system)
- 3) 컴퓨터 제어 시스템(computer control system)
- 4) 인간의 노동력(human labor)

또한 이러한 FMS를 구현하기 위해 주어져야 할 요건들과 주요 기술들이 있다. 먼저 갖추어야 할 요건들로는 작업 착수 스케줄(schedule)의 작성, 자동적인 공작물 착탈 및 반송, 가공 정보로서의 NC 자료 유지, 공구의 준비와 가공의 진행에 따른 자동 교환, 기계의 작동 상태 감시 등의 사전 준비가 필요하게 된다. FMS에 필요한 주요 기술로는 공작물의 자동 착탈, 공구의 자동 착탈 및 관리, CNC(computer numerical control)에 의한 적응 제어, 자동 계측, 자동 자재 반송 시스템과 정보 처리 시스템 등이 필요하게 된다.

4. FMS의 단계적 제어 기능

FMS를 제어하는 제어 기능(control function)들은 크게 다음 네가지로 분류할 수 있다.

- 1) CNC 기기에 대한 제어
- 2) 자재 반송 기구(material handling equipment)에 대한 제어

- 3) 시스템 내에서의 부품의 움직임(part movement)에 대한 제어
- 4) 시스템 수행도(performance)에 관련된 정보의 제어

그리고 이러한 기능들을 수행하기 위한 소프트웨어 제어 시스템의 작업(task)들은 다음과 같이 분류된다.

- 1) 시스템 자료 획득(system data acquisition)
- 2) 시스템 자료의 저장과 회수(system data storage and retrieval)
- 3) 시스템 자료의 번역(system data interpretation)
- 4) 시스템의 상태 결정과 번역(system status determination and interpretation)
- 5) 의사결정(decision making)과 구현(implementation)

이러한 제어 작업들은 일반적으로 3단계의 제어 단계(control level)를 갖는 계층 망(hierarchical network)으로 구성된다.

1단계 제어 컴퓨터는 통신(communication)과 제어 능력을 갖는 공정 제어 컴퓨터(process control computer)로서 공정과 직접 통신하고 대부분의 공정 제어 작업에 직접적으로 연결된다. 즉, 공작 기계나 조립 로봇 등에 직접적으로 연결되어 그 활동을 제어한다. 이 1단계 제어는 PC(programmable controller)나 CNC 등을 포함하는 가장 기본적인 제어 단위이기 때문에 생산 시스템의 구조와 각 기기들에 의해 많은 영향을 받는다.

2단계 제어 기기(controller)는 정보와 제어의 흐름을 효과적으로 구현 가능하도록 특별히 제작된 소프트웨어를 갖는 범용 미니 컴퓨터(general purpose minicomputer)로서 1단계 제어 컴퓨터와 통신하여 1단계 제어를 관리(supervise)하고, 세부적인 의사(tactical decision)를 결정하며, 지역 데이터베이스(local database)를 이용하여 시스템 자료를 얻고 관리, 유지하며, 시스템의 상태를 관찰하고 의사를 결정하고 구현하는 일 등 FMS의 전반적인 작업을 관리한다. 2단계 제어 기기에 사용되는 지역 데이터베이스에 포함되는 내용으로는 작업 순서 자료나 NC 자료 등이 있다.

3단계 제어 컴퓨터는 보통 FMS의 제어에 직접 연관되지 않은 호스트 컴퓨터(host computer)로서 각 워크스테이션에 명령을 제공하고 그 상태를 파악한다. 즉 워크스테이션으로부터 되돌아 오는 상태를 평가하여 그 작동 상태에 따라서 수행하여야 할 작업의 명령들을 제공한다. 그래서 이 3단계의 제어는 간접적인 제어를 나타내며, 총괄적인 의사(strategic decision)를 결정하고 완전한 데이터베이스를 유지한다. 이 데이터베이스에 포함될 내용으로는 제품의 정의, 소요량 계획 자료, 작업 순서 자료, NC 자료 등이 있다.

5. 제어용 소프트웨어

각 제어 요소의 실질적인 작업은 제어 실행 프로그램에 의해서 행해진다. 이 실행 프로그램은 고급 언어나 어셈블리어로 작성되며 소프트웨어 제어 작업들을 관리하고 시스템 상태의 변화를 탐지(detect)하고 대응하는 방식을 결정한다. 각 제어 단계의 각 공정마다 그 자신의 실행 프로그램을 가지며, 어떤 제어 단계에서의 모든 실행 프로그램들은 기능적으로 유사하다.

FMS를 구현하는 데 꼭 필요한 것 중의 하나가 바로 각 제어 계층간의 제어와 정보의 흐름이다. 제어의 흐름은 정해진 사이클 시간(cycle time) 동안에 행해진다. 이러한 실행 프로그램에 의해서 어떠한 출력이 내보내질 것인가의 결정은 일정한 상태 표(state table)에 의해서 이루어진다. 이 상태 표는 주어진 입력에 대응하여 출력이 정해져 있는 결정적이고 동기화 된 시스템이다. 각 사이클 동안에 제어 실행 프로그램들은 몇 가지의 입력 정보를 받아서 상태 표에 주어진 출력을 제공한다. 입력 정보로는 입력 명령어, 감각 정보(sensor information), 상태 피이드백 그리고 내부의 상태 변화가 있다. 여기에 대하여 발생하는 출력 정보로는 출력 명령어, 상태 출력 그리고 내부의 상태 변화가 있다.

5.1 1단계 제어 프로그램

5.1.1 2단계 제어 기기와의 통신

2단계 제어 기기와의 통신에 관계된 활동들을 보면 다음 네가지로 분류할 수 있다.

- 통신에서의 확인 신호(acknowledgment)
- 2단계 제어 기기를 위한 프로그램의 전송
- 두 제어 기기들 사이의 정보 전송
- 활동의 완료 확인

2단계 제어 기기와의 통신을 위하여 우선 순위의 개념이 사용된다. 기억 장치나 전원의 손상, 기계의 고장, 팰렛의 위치 부적성, 공구의 파손 등이 발생될 때 인터럽트(interrupt)를 사용한다. 다중 네트워크 시스템(multidrop networking system)은 프로세서들 간의 통신을 매우 단순화 시켜 준다. 즉, 네트워크 하드웨어 및 소프트웨어는 많은 통신 특성을 제공하며, 사용자 자신이 프로토콜(protocol)이나 통신 서브루틴을 개발해야 하는 필요성을 없게한다.

5.1.2 프로그램의 전송

프로그램의 전송이란 새로운 실행 프로그램과 NC 프로그램을 1단계

프로세서(processor)에 전송하는 것을 말한다. 전송되어지는 정보로는 기계의 상태, 공구의 상태, 현재의 작동 상태 등이다. 하나의 활동이 완료되면 2단계 제어 기기는 기계의 상태가 변경된 것을 알게 된다. NC 기계나 로봇들을 제어하는 1단계 제어 실행 프로그램은 상위 단계에서 전송되어 오는 명령을 분해(decomposition)해서 각 NC 기계나 로봇에 할당하고 여기에서 수행된 결과와 각각의 상태들을 상위 단계로 피드백하는 지능적인 장치가 필요하게 된다. 이러한 예를 보면, 1단계 제어 기기는 공구의 마모 정도를 측정하고 가공 공정을 감시한다. 또한 잔여물의 처리(waste desposal), 픽스춰(fixture) 제어 그리고 자가 감시(self-monitoring) 등의 기능을 수행한다.

5.1.3 기계 작동의 감시

2단계 제어 기기에 의한 1단계 제어 실행 프로그램의 수정(modification) 능력은 보다 빠른 시스템의 재시동과 초기화(restart/initialization)를 위해서 사용되며 1단계 제어 기기에서 소프트웨어가 쉽게 갱신(update)될 수 있게 한다. 또한 1단계 제어 기기들 중 일부는 자재 반송 시스템과 관련이 되어 부품의 움직임과 부품의 확인(part identification) 활동을 관리한다. 이러한 활동들과 그에 관련된 작업들은 자재 반송 시스템의 형태와 시스템 내에서 부품의 움직임을 확인하는데 사용되는 기법에 의존한다.

5.2 2단계 제어 프로그램

2단계 프로세서는 1단계 프로세서보다 더 다양한 활동들을 담당하고 보다 많은 작업의 처리를 수행한다. 이 실행 프로그램들은 소프트웨어 제어 시스템이 수행해야 할 작업들의 수행도를 관리하고, 또한 1단계 프로세서로부터 정상적 통신과 높은 우선 순위 통신을 구별할 수 있게 하고, 1단계 프로세서가 다른 공정을 수행 중인 2단계 프로세서에 인터럽트할 수 있는지를 결정하며, 현재의 요구나 또는 다른 방식의 우선 순위에 의해 1단계 프로세서가 서비스를 받을 수 있는지의 여부를 결정하게 된다. 또한 2단계 제어 실행 프로그램은 3단계 프로세서와 통신하는 작업을 수행하여야 한다. 이처럼 2단계 제어 기기에 요구되는 다양한 활동들을 다루기 위해서 2단계 제어 실행 프로그램이 1단계 제어 실행 프로그램보다 더 복잡해지는 것은 당연한 일이다.

2단계 제어 기기는 FMS의 전반적인 사항을 관리한다. 그래서 FMS 관리 컴퓨터라고도 불리운다. 이 FMS 관리 컴퓨터에 의해 수행되는 2단계 제어 실행 프로그램의 활동들은 크게 시스템 제어, 통신, 데이터베이스 관리 등

새가지 역할로 분류된다. 시스템 제어 활동은 FMS 내로 또는 외로의 부품의 이동이나 시스템 내에서의 부품의 이동에 대한 제어를 담당한다. 통신 활동은 모든 시스템 통신의 제어와 1단계 및 3단계 제어 기기로의 데이터링크(datalink)의 보유, FMS 관리자를 위한 보고 능력의 보유 등의 역할이다. 데이터베이스 관리 활동은 시스템의 상태 자료 저장이나 NC 프로그램의 저장, 공정과 기계, 부품과 펠렛의 유형, 기계와 가공법에 관련된 할당 표의 유지나 시스템의 손상으로부터 체계적인 회복을 위한 오딧 트레일(audit trail) 기능들을 갖는다.

그리고 이러한 활동들을 수행하기 위한 시스템을 구현할 때 클럭-드리븐(clock-driven) 시스템, 시퀀셜리-드리븐(sequentially-driven) 시스템, 인터럽트-드리븐 시스템, 멀티태스킹을 기초로 한 실행 프로그램(multitasking-based executive program) 등의 방법을 이용한다.

클럭-드리븐 실행 프로그램에서 각각의 작업은 컴퓨터의 실시간 클럭에 의해서 제공되는 주기적인 인터럽트의 시간 간격에 할당된다. 이러한 접근 방식은 각 작업에 대한 시간을 미리 결정해야 한다는 문제점이 따르며, FMS처럼 복잡한 시스템에서는 이 일이 매우 어렵게 된다. 또한 높은 우선 순위의 기구와 활동들이 우선적인 서비스를 받을 수 없어서 효율적이지 못하며 고장 등이 발생한다. 그래서 클럭-드리븐 시스템은 거의 사용하지 않는다.

시퀀셜리-드리븐 실행 프로그램은 소프트웨어 폴링 시스템(software polling system)을 사용한다. 실행 프로그램은 미리 정해진 순서에 따라서 시스템 상태 입력 포트(input port)와 통신 채널을 폴링한다. 폴링의 순서는 입력 매개 변수(input parameter)에 의하여 변하지만 우선 순위가 높은 기구에 우선적인 서비스를 제공하지는 못한다. 시퀀셜리-드리븐 시스템의 잇점은 고급 언어로도 구현하기가 쉽다는 데에 있다. 실제로 FORTRAN으로 FMS를 구현한 예가 있다.

가장 효율적인 실행 프로그램은 인터럽트-드리븐 시스템을 사용하고 이 인터럽트-드리븐 시스템은 프로세서를 필요로 한다. 외부 장치가 프로세서를 정지시킬 수 있고, 그 프로세서는 현재의 프로그램 정보를 저장하고 미리 지정된 기억 장소로 분기(branch)하여 그 기억 장소에 내재된 인터럽트를 수행하여야 한다. 인터럽트의 수행이 완료되면, 프로세서는 인터럽트의 발생 전으로 회귀한다. 인터럽트 개념의 사용으로 각 기구들에 우선 순위를 구현할 수 있다. 그래서 높은 우선 순위의 기구가 인터럽트된 프로그램을 선점(preempt)할 수 있고, 낮은 우선 순위의 기구는 대기 행렬(waiting queue)에서 기다리게 된다.

이러한 인터럽트를 구현하는 많은 대기 행렬 방법이 있지만, 작업의

우선 순위에 기반을 둔 법칙이 우선 순위 인터럽트 시스템에 가장 적합하다. 이 인터럽트-드러브 시스템의 단점은 그 구현에 있어서 필요한 소프트웨어 오버헤드(overhead)가 많다는 것이다.

복잡한 제어 환경에서 인터럽트의 처리를 지원하기 위해서는 운영 체제가 실시간 제어를 위해서 설계되고 개발되어야 한다. 멀티태스킹(multitasking) 시스템에서는 각 활동을 감시하기 위해서 서로 다른 작업들이 생성되어야 한다. 각 작업은 비동기적(asynchronous)이고, 실시간으로 특정한 기능을 수행하는 논리적으로 완전한 프로그램으로서 다음 네가지 상태 중의 하나로 존재하게 된다. 수행(executing), 준비(ready), 대기(suspended), 정지(dormant) 상태이다.

각 작업들마다 우선 순위가 주어지고, 운영 체제는 정지와 대기 상태를 제외한 활성화된 작업(active task)들 중에서 가장 높은 우선 순위를 갖는 작업 만을 수행한다. 작업들은 각자 자신의 우선 순위를 변경할 수 있으며 서로 통신할 수 있는 것처럼 다른 작업의 우선 순위도 변경시킬 수 있다. 작업이 대기 상태가 되면 이 작업은 어떤 시스템 콜(system call)의 발생을 기다리거나 다른 실시간 동작의 완료를 기다리게 된다. 작업이 정지 상태가 되면 운영 체제는 그 작업을 무시하게 된다.

실제로 각 1단계 공정 제어 기기들을 감시하는 작업, 높은 우선 순위 통신의 각 유형을 다루는 작업들을 생성할 수 있는데 이러한 작업들은 1단계 제어 기기에 의해서 통신이 시작될 때까지 대기 상태에 있게 된다. 그래서 감시 작업이 활성화 되면 2단계 프로세서의 제어를 받아서 우선 순위 통신의 유형을 파악하게 된다.

FMS를 제어하기 위한 실행 프로그램을 구축하는 데 멀티태스킹 운영 체제를 사용함으로써 복잡한 인터럽트 관리 시스템의 필요 없이 인터럽트를 사용하여 우선 순위에 따른 서비스를 제공할 수 있다. 그러나 그렇게 구현된 시스템도 아직 많은 소프트웨어 오버헤드가 있다. 그러나 여러가지의 기술적인 진보로 이러한 오버헤드에 의한 커다란 제한을 받지 않게 되었다.

이러한 작업들을 수행하는 시스템 제어 구성 요소를 구현하는 데에는 프로그램 구조, 자료 구조, 의사 결정 알고리즘 등이 요구된다. 의사 결정 알고리즘들은 멀티태스킹 프로그램 구조를 사용하여 구현되며, 각 작업들이 필요로 할 때에만 활성화된다. 또한 알고리즘들은 서로 독립적이고 논리적으로 완전한 프로그램 세그먼트로 구현되기 때문에 작업을 교체하면 알고리즘이 교체된다. 그리고 이 제어 알고리즘들은 2단계 제어 기기가 1단계 실행 프로그램을 대치하는 식으로 3단계 제어 기기에 의해서도 대치될 수 있다.

5.3 자료 구조와 데이터베이스

FMS를 구현하는 데 필요한 자료 구조들은 정적(static) 또는 동적(dynamic) 정보들을 모두 포함한다. 정적 정보들은 시스템이 특정한 구조로 한 번 초기화되면 변화되지 않는다. 반면에 동적 정보들은 시스템 내의 각 사건 발생마다 변하게 된다. 대부분의 정보들은 현재의 시스템 상태를 유지할 수 있게 하는 자료들이고 그 외의 자료들은 데이터베이스에 접근하는 데 필요한 키워드(key word)들을 제공한다. 이러한 자료 구조는 크게 부품의 확인 자료, 기계의 확인 자료, 자재 반송 구성 요소의 확인 자료, 펠렛 확인 자료 등의 네가지 유형으로 분류할 수 있다.

데이터베이스는 운영 체제의 파일 조작 루틴(file manipulation routine)이나 데이터베이스 관리 시스템(DBMS)을 사용하여 생성될 수 있다. DBMS의 사용은 얼마 정도의 오버헤드가 필요하지만 소프트웨어 프로그래밍 작업을 단순화시키고, 제어 시스템을 위한 자료의 요구가 쉽게 확장되고 수정될 수 있게 한다. 빠르고 계속적인 접근을 필요로 하는 항목들은 데이터베이스 내에 사본(backup copy)을 가지며 주기억 장치 내에도 내재될 수 있다.

DBMS를 사용함으로써 3단계 제어 기기가 다른 작업을 통하여 데이터베이스에 접근할 수 있고, 시스템 계층을 통한 관리를 위해서 시스템의 상태와 성능에 대한 질의가 허용된다. 각 제어 작업들은 파일 조작을 위해서는 시스템 콜을 요구할 수는 없지만, 표준 질의어를 통하여 데이터베이스와 인터페이스할 수 있다.

6. 결 론

이상에서 오늘날의 세계 경제의 어려움 속에서 그 자구책으로 개발되어 온 공장 자동화의 일환으로서의 FMS의 개념을 고찰하였다. 아울러 FMS를 구현하는 과정에서 컴퓨터의 사용과 이에 따르는 소프트웨어의 기능에 대하여도 알아보았다. 실제로 FMS를 구현하는 데에는 많은 숫자의 컴퓨터와 소프트웨어가 필요하고 그만큼 많은 경비가 소요된다.

FMS는 그 중심 부분이 가공 시스템으로 이루어져 있고 이를 연결하는 자동 자재 반송 시스템 그리고 이들을 제어하는 종합 소프트웨어로 구성되어 있다. FMS의 구축에 따른 경제적인 문제와 컴퓨터의 보다 효율적인 사용을 위해서 FMS 소프트웨어 제어 작업들은 몇 단계의 계층 구조로 구현된다. 1단계 제어는 공정과 직접적으로 통신하는 공정 제어 컴퓨터이고, 2단계는 FMS에 직접적인 제어를 행하는 FMS 관리 컴퓨터, 그리고 3단계는 FMS에 대한 종합적인 인터페이스이다.

FMS를 제어하는 소프트웨어는 시스템의 작동을 관리하고 시스템의 상태

변화에 대한 대응 방안 등을 결정한다. 이러한 제어용 소프트웨어를 구현하는 몇가지의 방법이 있는 데 그 중에서 멀티태스킹을 기초로 한 시스템이 가장 강력한 수단이다. 제어용 소프트웨어의 기능은 시스템 제어, 통신, 데이터베이스 관리 등의 세가지로 분류할 수 있다. 이러한 기능들을 수행하기 위해서는 각 제어 단계에서 필요한 여러가지 의사 결정 알고리즘들과 그에 필요한 자료 구조와 데이터베이스의 구축이 요구된다.

이러한 생산 시스템을 구축함으로써 얻을 수 있는 효과는 많이 있다. 하지만 실제 구현 과정에서 컴퓨터의 보다 적절한 사용 등의 문제점들과 효과적인 제어 체계의 구현 등은 앞으로 더 연구하여야 할 과제이다.

< 참고 문헌 >

- [1] MIKELL P. GROOVER and EMORY W. ZIMMERS, JR., CAD/CAM Computer-Aided Design and Manufacturing, PRENTICE-HALL Inc, 1984.
- [2] MIKELL P. GROOVER, Automation Production Systems and Computer-Integrated Manufacturing, PRENTICE-HALL Inc, 1986.
- [3] Robert E. Young, "Software Control Strategies For Use In Implementing Flexible Manufacturing Systems", Industrial Engineering, 1981.
- [4] U. Rembold and R. Dillmann, Computer-Aided Design and Manufacturing : Methods and Tools, Springer-Verlag Berlin Heidelberg, 1986.
- [5] Kerry L. Elkins and Oded Z. Maimon, "Information Flow between a Workstation Controller and its Robotic Sub-System", Computers in Industry, Vol. 9, No. 3, 1987.
- [6] 한국 공장 자동화 연구소, FMS/FMC의 구성, 1988.
- [7] 한국 공장 자동화 연구소, 공장 자동화 기술 개론 : 공장 자동화의 추진 방향, 투자, 설비 기술, 1988.