

공장 자동화를 위한 컴퓨터 구조

朴 圭 皓

韓國科學技術院 電氣 및 電子工學科 教授 (工博)

I. 概 要

최근의 공장 자동화는 대량생산 위주에서 다품종 소량생산으로 바뀌고 있다. 앞으로의 생산 제품의 대부분은 중소형 기업에서 불규칙하고 다양한 소비자의 요구에 따라서 생산되어야 하므로 이에 적합한 생산 시스템이 필요하다. 이같은 목적으로 컴퓨터는 공장의 자동화에 있어서 각종의 생산도구들을 융통성있게 운용하는데 있어서 가장 중요한 요소이다.

일반적으로 부품들이 대량생산 될때는 고정된 자동화 설비를 이용하는 것이 경제적으로 유리하나, 소량생산인 경우는 이와같은 설비를 가설하는 비용에 비하여 비경제적이다.

융통성을 갖는 자동화 시스템은 두가지 방식을 절충하는 것으로서 제품의 확실성을 증가시킬 수 있을 뿐만 아니라, 수요에 따라서 여러가지의 스타일을 선택할 수 있는 다양성까지 갖는다.

이같은 목적으로 소프트웨어의 대처로서 기존 line의 큰 변화없이 다품종 소량생산에 대처 할 수 있는 FMS (flexible manufacturing system)이 필요하다.

FMS는 자동화된 작업의 흐름으로 연관되는 workstation들의 그룹과 이들을 연관시키고, 각 workstation에서 작업을 하는 로봇, 그리고 부품의 제조 및 조립을 관리하는 종합제어 컴퓨터로 구성된다.

자동화 시스템을 이루기 위해서는 다음 4가지 요소들이 갖추어져야 하며 시스템 구성을 위해서 반드시 고려해야 한다.

- ① 통신
- ② 부계환 제어
- ③ 실시간 처리
- ④ 사람과 기계와의 접속장치

앞으로의 제조 시스템, 자동화된 job shops, 그리고 산업 로봇들에서의 컴퓨터 하드웨어는 분산된 센서

들과 actuator들이 부착된 다중 프로세서들이 서로 상호 연관을 갖는 형태로 구성된다. 이같은 처리능력의 분산은 신뢰도의 증가와 실시간 응답이 빨라지는 반면에 소프트웨어 설계 및 구성의 문제가 아직도 많이 남아 있다.

본 원고에서는 공장 자동화에서 필요한 컴퓨터 구조를 기술한다. 컴퓨터의 5가지 중요한 구성요소는

- ① 생산 시스템 제어
- ② 분산 데이터 관리
- ③ 분산 컴퓨터 구조
- ④ 시스템 소프트웨어
- ⑤ 사용자 인터페이스

으로 나눌 수 있다.

II. 생산 시스템 제어¹⁾

일반적으로 제어 시스템은 제어대상과 컴퓨터 제어로 나눌 수 있다. 컴퓨터 제어기는 다음의 3가지 단계의 파이프 형태로 수행된다.

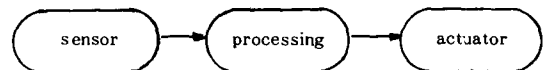


그림 1. 컴퓨터 제어기 파이프라인

위의 각 단계들은 병행으로 연결된 계층적 구조를 갖는다. 또한 컴퓨터 제어기는 제어 대상을 통하여 각 각의 계층 형태에서 부계환된다.

생산 시스템의 제어 구조는 state machine으로서 구성되는 부계환 제어기의 계층적인 형태이다. 각 부분 시스템의 모든 입력, 출력, 상태, 그리고 상태천이은 상태그래프로 나타낼 수 있으며, 제어 시스템에 의해서

처리될 수 있는 상태 테이블로 발전 시킬 수 있다. 시간간격 또는 제어 cycle은 각 제어 부분 시스템에 대하여 정의되며, 어느 정도로 자주 테이블이 처리되어야 하는가를 결정한다.

상태 테이블의 처리는 상태 변수의 샘플링, 테이블의 현재 상태 위치, 그리고 procedure의 수행 및 현재 상태에 관련된 출력의 발생으로 이루어진다.

각 수준에서 제어 cycle은 시스템 안정성을 유지할 수 있도록 짧아야 한다. 즉, 각 프로세서는 시스템의 동작이 허용 범위를 벗어나기 전에 현재 상태를 알고 적절한 출력을 발생시켜야 한다.

제어 시스템이 task들을 다루기 위해 준비하는데 소요되는 시간의 양은 planning horizon으로 정의된다. 시스템은 planning horizon을 넘어서 발생하는 사건들을 다룰 수 없다. 제어 수준이 낮을 수록 보다 짧은 planning horizon을 정하여, 컴퓨터 시스템은 각 제어 cycle동안 처리 능력을 증가시킬 수 있다. 각 수준에서 제어 시스템은 높은 수준에 의해서 설정된 한계 범위 내에서 병행으로 처리할 수 있다. 생산 시스템에서의 제어 계층은 일반적으로 5 개의 주요한 수준으로 나눌 수 있다. 각 수준은 몇가지의 제어를 가지며, 또한 부분 단계나 모듈들로 나누어진다.

다음 그림에서 제어 기능은 사각형으로 나타내고, 점

선은 상부 계층에서 하부 계층의 작업 흐름을 나타낸다. 각 기능 상자는 내부 제어를 위한 자신의 제어기들을 갖는다. 모든 기능들은 실선으로 나타나 있는 통신 라인과 각 숫자로 표시된 통신 node를 통하여 정보를 교환한다.

1. facility

제어 계층의 가장 높은 수준은 생산공학, 정보관리, 및 생산관리의 3 가지 주요한 부분 시스템으로 이루어진다.

생산공학은 부분 기기들의 설계를 위한 CAD 및 생산 공정의 계획등을 다룬다. 정보관리는 가격 및 투자 평가, 소비자 요구등에 대한 정보들을 관리한다. 생산 관리는 중요한 project를 계획하고, 장기간의 스케줄 구성 및 생산 자원 요구사항등을 다룬다. 이 수준에서 만들어진 계획은 다음 단계인 shop 제어 시스템으로 넘어간다.

2. shop

이 수준에서는 두가지의 주요한 모듈들을 통하여 shop floor상에 있는 작업들과 task 및 자원관리이다. Task 관리는 작업 순서의 스케줄, 장비의 유지와 같은 일을 한다. 자원관리는 cell수준의 제어시스템과 생산 작업들에게 작업장, 작업을 위한 장비, 그에 필요한 재료들을 할당한다.

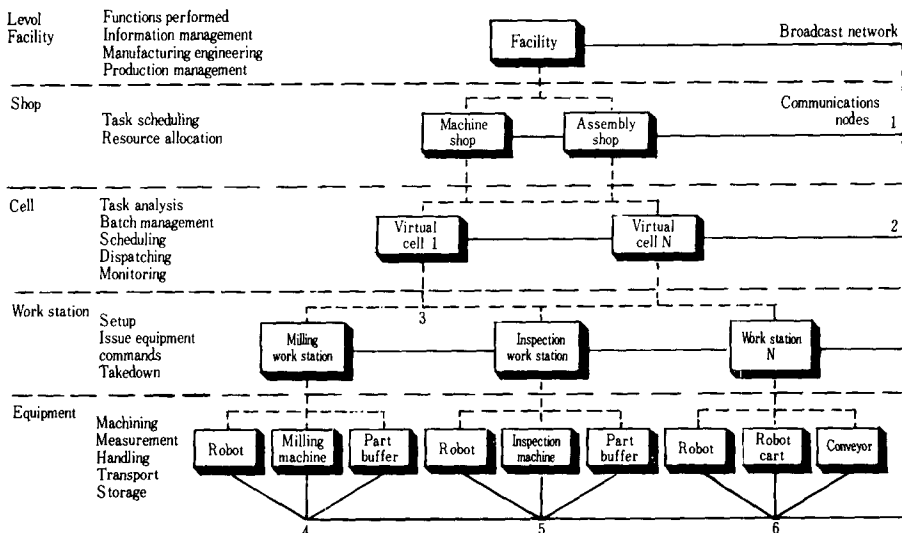


그림 2. 자동화 생산 시스템의 계층적 구조

3. cell

이 수준에서의 제어기는 비슷한 부분들의 batch 작업들을 순서화 한다. Cell은 동적인 생산 제어구조에서 workstation 처리 시스템의 시분할을 허용할 경우에 가상적인 cell들의 모임으로 볼 수 있다. Cell내의 모듈들은 시스템 task의 성능을 제어하고, 자원의 가용도를 분석하며, 작업의 진행도를 보고하고, 작업들의 스케줄 설정, 자원 준비, task감시등을 담당한다.

4. workstation

이 수준은 shop floor상의 장비들에 대한 그룹을 설정하고, 서로간의 일을 조정한다. 전형적인 workstation은 로봇트, 기기 장비, 재료 창고, 그리고 제어 컴퓨터로 구성된다.

5. equipment

이들 제어기들은 shop floor상에 있는 로봇트나 수차제어 기계등에 직접 관련된다. 이들 시스템들은 재료 보관, 운송, 재료처리 등의 기본적인 수준의 기능을 담당한다.

Ⅲ. 분산 데이터 관리

정보를 공유하기 위한 제어 시스템을 위해서 데이터 베이스가 필요하다. 이는 데이터 구조를 정하는 데이터 정의 언어와 데이터 dictionary를 위한 scheme, 그

리고 데이터 베이스를 사용하고 수정하기 위한 데이터 조작 언어로 이루어진다.

데이터 베이스 관리 시스템에 있는 기능과 데이터가 분산된 경우에는 요구되는 사항들이 제어 계층의 각 수준에 따라 다르다. 일반적으로 데이터의 양은 많으며, 허용 응답시간은 제어 계층의 수준이 높아질수록 길어진다. 각 제어 수준에서 데이터 전송을 위한 응답시간은 제어 cycle내의 수행에 대한 우선순위에 의해 결정된다.

데이터 사용과 수정이 매우 빠른 속도로 수행되어야 할 경우 디스크에서 보다 컴퓨터 주 기억장치에 데이터 베이스가 있어야 한다.

공장 자동화를 위한 데이터 관리 시스템은 그림3과 같이 데이터 dictionary 와 directory 시스템 그리고 데이터 베이스 관리 시스템으로 구성된다.

데이터 베이스 관리 시스템은 logical view processor와 실제적인 데이터 관리 시스템으로 나누어 질수 있다.

분산구조의 데이터 dictionary와 dictionary 시스템은 각 노드에서 데이터 세트를 가리키고 정의 하도록 각 통신 노드에 있어야 한다. Dictionary의 directory 시스템내에 있는 정보는 시스템 운용중에도 데이터가 발생 및 삭제되므로 동적인 상태로 유지된다. 또한, 전체적인 데이터 베이스의 공통된 개념적 모델은 중앙에

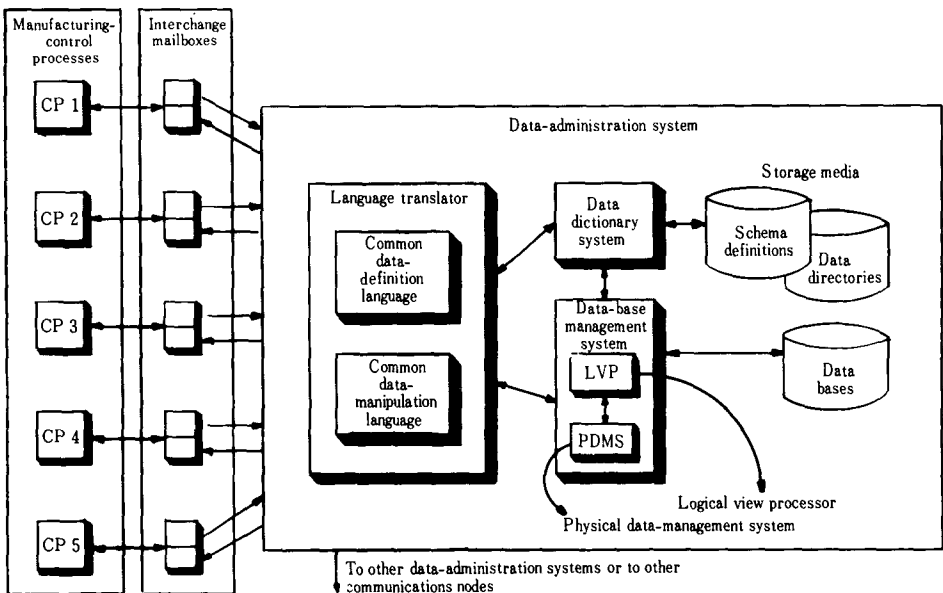


그림 3. 데이터 관리 시스템

서 관리해야 한다. 중심적인 모델은 분산에 관한 정보와 데이터 set들의 논리적인 구조와 여러 node에 분산된 데이터 set들의 레코드 사이의 관계를 포함하고 있다.

언어 번역기는 국지 제어 프로그램 및 통신망에 있는 다른 제어 프로그램 모두가 사용할 수 있어야 한다. 번역기는 요청이 국지에서 처리가능한 것인지를 결정하기 위해 국지 데이터 directory를 사용한다. 가능하면 query는 번역되어 국지의 데이터 베이스 관리 시스템으로 전해진다. 만약에 데이터가 국지에 없으면, query는 통신 시스템을 통하여 관련된 다른 node에 있는 데이터 관리 시스템으로 보낸다.

각 시스템은 logical view processor를 통하여 데이터를 사용한다. Physical data manager는 기억장치에서 실제적인 할당, 저장, 및 수정등의 일을 수행한다.

IV. 분산 컴퓨터 구조

1. 외부와의 접속장치

외부와 접속을 위한 장치는 가장 낮은 수준의 환경에 따라서 결정된다. 이는 컴퓨터가 처리할 수 있는 신호와 외부 프로세스가 처리하는 신호를 서로 변환시켜 주는 역할을 한다.⁶⁾ 일반적으로 디지털(또는 binary)신호, 펄스신호, analog신호로 나눌 수 있다. 디지털 신호는 컴퓨터가 처리할 수 있는 신호이다. 그러

므로 펄스 신호는 카운터를 통하여, analog신호는 A/D변환기로 각각이 디지털 신호로 변환시켜 컴퓨터와 접속시킨다. 이같은 접속장치는 주로 제어 계층에서 equipment 수준에서 필요로 하며 상부 계층에서는 디지털 신호로 처리한다.

2. 컴퓨터 구조

공장 자동화를 위한 컴퓨터는 로봇과 같이 계산량이 많은 경우를 위한 다중 프로세서 구조로부터 job/shop, CAD/CAM 등과 같이 서로 관련된 시스템들을 연결하기 위한 분산 시스템 구조까지의 넓은 범위를 포함한다.

여러개의 컴퓨터들이 연결된 분산 시스템은 융통성을 갖고 병행처리가 가능하다. 또한 응답시간과 신뢰도의 향상을 얻을 수 있다.

일반적인 실시간 처리 응용분야는 기능이 계층적으로 정해지며, 비교적 작은 여러가지의 일을 동시에 수행한다.¹⁾ 공장 자동화에서도 그림 1 과 같이 제어 기능이 계층적으로 나누어지므로 분산 시스템이 적합하다. 제어 시스템은 대부분 중형 컴퓨터와 소형 컴퓨터가 각각에 여러개의 제어 프로세스를 갖고 공용의 통신망에 의해 연결되는 형태를 갖는다.

제어 process 간의 상호 데이터 교환이 실시간을 요구하며 많은 양의 데이터를 처리하는 단위를 그룹으로 하여 그림 4 와 같이 통신 node에서 수행 시킨다.

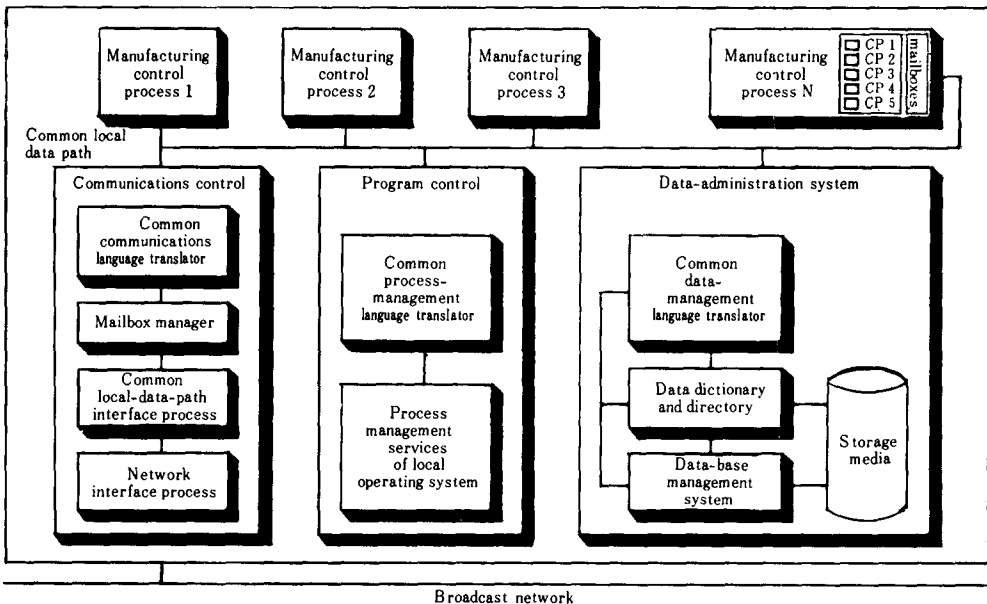


그림 4. 통신 node를 갖는 컴퓨터 시스템

예를들면, 그림 4가 그림 1에서 노드 6을 나타낸다면, 생산 제어 프로세스 1은 로봇트 제어 기능을 담당하는 프로세스이며, 생산 제어 프로세스 2는 로봇트 cart를, 그리고 생산 제어 프로세스 3은 컨베이어를 제어하는 프로세스이고, 생산 제어 프로세스 N은 워크스테이션 N을 나타낸다.

한 node에서 계산량이 많고 시간 제약이 많을 경우 다중 processor를 사용할 수 있다. 그림 4에서 제어 프로세스 1, 2, 3들이 시간 제약이 큰 경우에, 각각의 전용 processor를 둘 수가 있다. 이 경우 mailbox는 shared 메모리에 있어야 한다. 제어 계층의 하부일 수록 전용 processor를 갖는 다중 processor 구조가 적합하다. 생산 제어 소프트웨어의 상부 계층일수록 외부 환경과의 접속장치에 관련이 적고, 하부 계층에서 보다 긴 응답 시간을 갖는다. 분산 제어에서 제어 컴퓨터의 위치는 접속하는 외부 환경에 따라서 결정된다. 이같이 하여 장거리에서 제어신호의 감쇄 문제를 극복할 수 있다. 제어 프로세스들 사이의 데이터 교환이 통신망을 통해서 교환할 수 있는 시간보다 짧은 경우 공용 메모리를 사용해야 한다.

하나의 processor 상에서 필요하지 않을지라도 고속의 데이터 교환을 갖는 경우 이들의 그룹화가 필요하다.

3. 통신망 구조⁽⁴⁾

공장의 자동화와 같은 시스템은 제어 대상이 지역적으로 분산되어 있으므로 국지 통신망이 많이 사용된다. 하드웨어 가격이 낮아지고 통신 기술의 발전으로 최근에는 분산 시스템으로 실시간을 요구하는 응용분야에서 여러가지의 요구사항들을 만족시켜 나간다.

국지 통신망의 구조는 여러가지가 있으나 공장 자동화와 같은 응용 분야에서는 신뢰도, 응답시간들이 중요한 요소이므로 사무 자동화를 위한 통신망에 비하여 선택 범위가 제한된다.

일반적으로 버스 형태와 ring 형태가 널리 사용될 수 있다. 두가지의 특징은 서로 상반되는 면이 있다. 버스 형태는 하나의 케이블에 여러가지의 기능을 갖는 컴퓨터들이 접속장치를 통하여 연결된다. 이 형태의 특징은 접속장치가 수동 소자로 구성되므로 구성이 간단하며 신뢰도가 높다.

통신망을 사용하기 위한 프로토콜은 CSMA(carrier sense multiple access) 방식이 주로 사용되고 있다. CSMA 방식은 통신망 사용권이 불규칙하게 결정되므로 응답시간을 결정하기 힘들다.

버스 형태에서 다른 프로토콜은 token-passing 방식으로서, 최대 지연 시간을 보장 할 수 있으나 신뢰성에서 CSMA 방식에 비하여 낮다.

통신망 사용 프로토콜은 에러의 영향과 긴급 메시지 송수신, 응답 시간 제약등을 고려하여 결정해야 한다.

Ring 형태는 여러개의 컴퓨터들이 ring의 모양으로 연결된다. 이 형태는 전송로에서 데이터의 충돌이 없으므로 통신망 사용에 대한 지연시간을 보장할 수 있다. 또한 잡음의 영향을 받지 않고 대역폭이 넓은 광섬유를 사용 할 수 있는 장점이 있다. 그러나 통신망 접속장치가 능동소자로 구성되어 사고에 따라서는 통신망 전체에 영향을 줄 수 있으므로 신뢰도가 낮은 단점이 있다.

V. 시스템 소프트웨어

계층구조에서 제어 수준이 통신망에 연결된 여러 컴퓨터에 분산 되고, 개별 시스템은 각 수준에서 작은 여러가지의 task들로 수행된다. 대부분 제어 프로그램은 서로 통신하는 다른 프로그램이 어디에 있는지 알 필요가 없어야 한다. 그렇게하면 두개의 제어 프로세스 사이의 데이터 교환은 그들의 위치에는 서로가 무관해질 수 있다. 하나의 프로세스 간의 통신을 위한 기능으로 가능하다. 프로그램은 이름에 의해 통신 경로를 열면 통신 관리 프로그램에서 이름으로부터 실제 프로세스의 주소로 바꾸어 적절한 통신 방법을 찾으면 된다. 이같이 하여 제어 프로그램은 서로 격리되어 있어도 단지 표준화된 통신 방식으로 메시지 교환을 할 수 있다. 외부 장치에 직접 연결되지 않은 제어 모듈들은 다른 프로세서로 옮겨질 수 있다.

이같은 환경을 제공하기 위해서는 통신망과 이를 기반으로 분산 시스템을 운용할 수 있는 분산 운영 시스템이 필요하다. 시간 제약을 받는 여러 프로세스를 효율적으로 수행시킬 수 있는 프로세스 관리 기능과 분산 시스템에서의 메시지 교환을 원활히 할 수 있는 통신기능이 중요한 위치를 차지한다. 일반적인 오퍼레이팅 시스템이 컴퓨터가 갖는 자원들을 효율적으로 관리하는 데에 역점을 두는 반면에 공장 자동화와 같은 분야에서는 실시간 처리가 가능하도록 하며, 시스템 신뢰도 향상에 주안점을 둔다.

실시간 서비스는 deadline내에 서비스가 제공되지 못했을 경우에 야기되는 문제의 크기에 따라서 실시간 작업수와 우선순위를 정하여 제공한다. 일반적으로 심각한 문제를 야기시킬 경우, 심각하지는 않으나 dead-

line이내에 수행되어야 할 경우, 응답시간과 관련이 없는 작업으로 나누어 우선순위를 정한다. 심각한 문제를 야기시킬 작업이 있는 경우 응답 시간을 확신할 수 있는 범위 내에서 작업량을 결정해야 한다.

일반적으로 신뢰성을 높히는 문제와 빠른 응답 제공에는 서로 상반된 일이 필요하다. 신뢰성을 높이기 위해서는 복잡한 운용 시스템이 되기 쉽고, 빠른 응답을 위해서는 시스템의 단순화가 필요하다. 그러므로 실시간 시스템에서 신뢰성을 높이기 위한 시간적인 손실과 요구되는 응답 시간은 응용 분야에 따라서 적절한 수준에서 결정해야 한다.

그림 4에서와 같이 각 제어 프로그램과 이들을 운용하는 통신, 데이터 관리, 프로그램 관리 시스템들 간의 통신을 위하여 공통된 언어가 필요하다. 통신 시스템에서의 언어는 가상적인 통신 경로의 이름 설정, 사용권, 사용 방식등과 같은 기능에 대한 것들이다. 프로세스 관리 언어는 분산 시스템 전반에 걸쳐서 요구되는 프로세스 관리의 공통된 syntax들이다.

VII. 사용자 인터페이스

공장 자동화를 위한 분산처리 시스템을 유지 및 관리를 위하여 오퍼레이터, 프로그래머, 관리인들이 시스템과의 접속장치를 가져야 한다. 간단한 그래픽스와 메뉴 방식, 그리고 자연어와 비슷한 고급 언어가 공장 자동화 시스템을 조작하는 숙련되지 않은 사용자에게 필요하다. 인간의 특성을 연구한 결과에 의하면, 사람은 한번에 단지 7 가지 정도의 상이한 정보를 인식할 수 있으며 그림에 의한 정보가 문자나 테이블 보다 쉽게 이해된다.

미국의 NBS (national bureau standard)에서는 도형에 의한 심볼들을 시스템안에 있는 대상들과 시스템의 상태를 나타내기 위해 개발하고 있다. 이들 심볼들은 사용자가 배우기 쉽고 사용하기 간편하며 쉽게 시스템을 인식할 수 있어야 한다. 이같은 수단은 사용자들이 제어 시스템에게 도식적으로 새로운 task들을 어떻게 수행할 것인가를 가르칠 수도 있다.

VIII. 結 論

컴퓨터가 공장 자동화에 있어서 중요한 위치를 차지하고 있다. 현재 공장 자동화에 있어서 난점은 개발 가격이 높고, 시스템이 복잡하며, 각종의 기술이 미성숙한 단계에 있기 때문이다. 또한 표준화된 생산 시스

템 구조가 부족한 점도 요인이 된다.

공장 자동화를 위한 컴퓨터 시스템 구성에서 문제점은 하드웨어와 통신망에서 보다는 소프트웨어에 있다. 현재 고비능을 갖는 마이크로 프로세서, 외부 접속장치등은 공장 자동화에서 요구되는 컴퓨터를 쉽게 구성할 수 있다.

소프트웨어에서는 아직도 해결되지 않은 문제들이 많다. 병행처리가 가능하도록 하며 신뢰도와 성능에 따라서 재구성성이 가능한 분산 운용시스템이 개발되어야 한다. 이같은 특성을 갖는 운용 시스템은 기존의 시스템과 다른 개념에서 고려되어야 한다. 기본적으로 고려되어야 할 사항들은 신뢰도, 융통성, 보장된 응답들이다.

공장 자동화를 위한 컴퓨터 구성을 위한 제어 특성과 이에 적합한 구성 요소들을 고찰하여 보았다.

공장 자동화가 널리 보급되기 위해서는 컴퓨터와 생산 시스템 구성 요소들이 종합적으로 연구 개발되어야 하며, 표준화된 기능 및 접속방식이 이루어져야 한다.

參 考 文 獻

- [1] C. McLean, M. Mitchell and E. Barkmeyer, "A Computer architecture for small-batch manufacturing," *IEEE Spectrum*, pp. 59-64, May, 1983.
- [2] Hiromu Tanimoto, "Factory Automation: An Automatic Assembly Line for the Manufacture of Printers," *IEEE Computer*, vol. 17, no. 12, Dec., 1984.
- [3] N. Komoda, K. Kera & T. Kubo, "An Autonomous, Decentralized Control System for Factory Automation," *IEEE Computer*, Dec., 1984.
- [4] Mellichamp, *Real-Time Computing*, Van Nostrand Reinhold, 1983.
- [5] 김명환, 최은호, "실시간 처리 컴퓨터 시스템" 전기학회지 제33권 제9호 1984년 9월.
- [6] James D. Schoeffler, "Distributed Computer Systems for Industrial Process Control," *IEEE Trans. Computer*, Feb., 1984.*