

# 자동조립 시스템을 위한 진단기능을 갖는 구조적 관리제어

○ 이 재혁\*, 유 범 제\*, 변 증 남\*, 오 상 록\*\*

\* 한국과학기술원 전기및전자공학부      \*\* 한국과학기술원 전기제어연구실

## Structured Supervisory Control with Diagnosis Capability for Automatic Assembly System

Jae-Hyeok Lee, Bum-Jae You, Zeung-Nam Bien, Sang-Rok Oh  
KAIST Dept. Electrical Eng. and KAIST Electrical Control Lab

### ABSTRACT

In this study, a structured supervisory control for automatic assembly system is developed. And also diagnosis and fault-recovery capability of this supervisory control are discussed. This structured supervisory control is actually applied to Die Bonding Machine and is proved to be useful and to work well.

### 1. 서 론

최근 산업계에서 공장 자동화에 관심을 가지면서 그 분야에 대한 연구가 활발히 추진중에 있다. 그중에서 조립 공정의 자동화는 일찍 부터 주목을 받고 있으며 가장 그 필요성을 느끼고 있는 분야이다. 특히 최근에는 비전시스템을 갖춘 고속정밀 자동조립시스템이 산업계에서 많이 요구되고 있는데 이러한 시스템의 경우에 처리해야 할 데이터량이 방대해 지기 때문에 빠른 시간내에 작업이 끝나야 하는 경우 여러개의 프로세서를 써서 시스템을 구성하여야 한다.

이러한 멀티프로세서시스템은 기능의 확장과 속도 향상을 생각하면 단일 프로세서시스템보다 가격대 성능비에서 더 우수한 평가를 받고있으나 전체 시스템 운영을 위한 소프트웨어가 복잡해지고, 테스트 및 고장 진단등이 어려워져 시스템을 총괄 관리 할 수 있는 관리제어가 중요한 비중을 차지하게된다.

본 논문에서는 프로세서를 여러개 사용하는 자동조립 시스템을 위한 관리제어를 좀더 구조적으로 구성하여 기존의 관리제어보다 유연성과 확장성이 증가되고 또 테스트하기 쉽고 디버깅(Debugging)이 쉽게하였다. 또 구조적 설계를 바탕으로 기계의 각 모듈의 진단과 전자 제어부의 각 모듈의 자기진단 기능을 추가하여 좀더 신뢰성 높은 관리제어를 구현하였으며 이를 실제 다이본딩 머신에 적용하여 그 유용성을 확인하였다.

### 2. 자동 조립 시스템의 구성

#### (1) 기능적 구조

본 논문은 다음과 같은 기능을 갖는 자동조립 시스템을 대상으로 하였다. 먼저 물체를 검사하고 위치를 결정하는 기능을 하는 비전시스템, 조립 대상인 부품들을 자동으로 운반해오고 조립된 부품을 다시 다음 공정으로 이송하는

이송 시스템 (Feeding System), 부품들을 직접 조립하는 매니퓰레이터 시스템, 마지막으로 위의 시스템을 총괄 관리하여 전체 시스템이 원활하게 운영될 수 있게 하는 관리제어 시스템이다.

#### (2) 하드웨어 구조

멀티프로세서 시스템의 구조는 여러가지가 있으나[1] 그 중에서 현재 가장 많이 쓰이고 있는 시분할(Time Shared)/ 공동버스(Common Bus) 구조를 본 논문의 대상으로 하였다.

이런 구조의 시스템의 가장 큰 단점은 데이터 전송율이 낮은 것이다[1] 실제로는 자동조립 시스템에서는 마스터(Master)/ 슬레이브(Slave) 프로세서간의 몇가지 데이터 교환을 빼고는 슬레이브(Slave) 프로세서마다 개별 메모리가 있어 자신의 소프트웨어와 데이터를 이 개별 메모리에 저장하므로 버스 경쟁 (Bus Contention)이 문제가 될 정도로 전송할 데이터 양이 많지는 않다[2].

#### (3) 예 : 다이본딩 머신

앞절의 구조를 갖는 자동조립 시스템의 한 예로 본 실험실에서 제작 실험을 완료한 다이본딩 머신을 예로 들 수 있다. 다이본딩 머신은 웨이퍼상의 작은 칩(Die)을 금속 리드프레임에 붙이는 기계이다[2]. 전체 구조는 (그림 1)과 같다. 3개의 프로세서(Super, Vision, Display)와 공동 메모리, 버스 아비터(Bus Arbiter)등과 여러 하드웨어들로 구성하였다.

### 3. 구조적 관리 제어

#### (1) 관리 제어의 구성

본 관리제어는 자동조립 시스템이 일반적으로 갖추어야할 기능을 분석하여 (그림 2)와 같이 6개의 스테이트(state)로 나누어 구현하였다.

##### 1) 주 스테이트(Idle State)

처음 전원을 넣을때 시스템의 각 하드웨어, 소프트웨어들을 초기화 시키고 주 메뉴를 보여준다.

##### 2) 자동조립 스테이트(Auto-Assembly State)

이 스테이트는 어떤 기계의 고유기능을 수행하는 스테이트이다. 이 스테이트의 특징은 기계의 동작 속도가 빠르고 기계의 모든 모듈이 전체 작업에 관여하므로 관리제어 프로세서는 자기 자신이 수행해야 되는 일 외에도 각 슬레이브 프로세서가 해야 할 일을 각각 정확히

시퀀스와 알맞은 시간내에 지시하고 확인해야 하므로 가장 중요한 스테이트라고 할 수 있다.

### 3) 파라미터 스테이트(Parameter State)

이 스테이트는 자동 조립 시스템의 유연성위한 스테이트이다. 즉 한가지 부품에만 고정되어 버린 시스템이 아니라 여러가지 다른 변수에 대해서도 시스템이 원하는 기능을 할 수 있게 파라미터를 조정할 수 있다.

### 4) 조정 스테이트(Adjust State)

이 스테이트는 부품의 조립을 좀 더 원활히 할 수 있게하기 위해 각 기계 부위들을 미세 조정하는 스테이트이다. 기계부 뿐 아니라 주위 환경에 따라 영향을 많이 받는 비전 처리를 위한 조명등도 조정한다.

### 5) 진단 스테이트(Diagnosis State)

이 스테이트는 시스템이 오동작을 하거나 고장이 났을 경우 고장난 부위의 진단을 용이하게 하기 위해 구현하였다. 먼저 전자 제어부의 각 모듈의 자기진단 (Self Diagnosis) 기능을 수행시켜 전자 제어부에 이상이 있는가를 찾아보고 다음 각 센서의 현재 상태를 나타낸 후 센서를 직접 손으로 작동시켜 그 상태 변화를 보고 각 센서의 정상 동작 여부를 확인한 수 있게 하였고 또 각 기계부들을 움직여 봄으로써 기계적인 고장 유무를 체크할 수도 있게 하여 기계의 고장이나 오동작시 그 원인을 빨리 찾아내어 수리 시간을 줄일 수 있게 하였다.

### 6) 긴급 스테이트(Emergency State)

이 스테이트는 예기치 않은 잘못, 고장 등으로 기계가 정상 동작 범위를 벗어나므로 해서 잘못하면 기계 자체가 망가지는 긴급한 사태가 일어날 때 기계를 멈추고 그 원인을 알아내게 구현하였다.

## (2) 계층(Layer) 구조

본 관리 제어는 운영 체제(Operating System)의 개념을 많이 도입하여 계층 구조[3]로 설계하였다(그림 3).

먼저 관리 제어 가장 내부에 시스템 하드웨어 계층이 있고 그 외부에 이 하드웨어를 직접 구동시키는 하드웨어 구동 소프트웨어 계층이 있다. 다음 이 계층밖에는 각 프로세서들 보다 필요한 기본동작을 만들어 놓은 계층이 있다. 이를 기능 계층(Function Layer)라 불렀으며 자동조립 시스템의 경우 어떤 작업을 위해 필요한 동작의 수는 일반적인 로보트 시스템과는 달리 한정되어 있어 필요한 기본 동작 들을 미리 데이터 베이스로 만들어 놓았다. 다음이 기본 동작들을 이용하여 원하는작업을 수행하는 가장 상위 계층인 스테이트 계층(State Layer)이 있다. 여기에는 파라미터, 조정, 진단, 긴급, 자동조립 스테이트가 각각 구현되어 있다.

이러한 계층 구조는 처음 각 기계부의 기본 동작을 정의하고 필요한 것을 뽑아낸 후 하드웨어 구동 소프트웨어 로 하나씩 구현하는 작업이 시간이 들지만 각 기본 동작들 마다 모듈별로 구현된 상태에선 원하는 기능등을 쉽게 구현할 수 있고 구조적인 소프트웨어가 이루어질 뿐만아니라 유연성, 확장성이 증가하여 요구되는 시스템 기능이 늘어나고 점점 소프트웨어가 방대해져도 쉽게 첨가나 삭제가 되어 시스템 개발에 도움을 줄 수 있다.

## (3) 관리제어의 운용 방식

본 관리 제어는 소프트웨어적으로 크게 두가지 부분으로 나누어 구현하였다. 즉 각 슬레이브의 상태와 키 입력을 체크하는 백그라운드(Background) 부분과 전체 제어 시스템의 제어를 수행하는 포그라운드(Foreground) 부분으로

나뉘어 시스템이 동작중에도 사용자의 요구에 즉각 응답하게 하였다. 이는 주기적인 타이머 인터럽트에 의해 운용된다.

## (4) 슬레이브 시스템과의 통신 방식

관리제어를 담당하는 프로세서는 다른 슬레이브 프로세서 들과 공동 메모리를 사용하여 통신을 수행하게 하였다. 이때 최악의 경우 여러 프로세서들이 동시에 같은 메모리 영역을 액세스하여 한 프로세서가 넣어 둔 데이터를 다른 프로세서가 지워버려 시스템이 교착(Dead Lock) 될 수가 있다. 이러한 문제는 여러 프로세서를 사용한 시스템에서는 자주 일어나는 문제로 동기 문제(Synchronization Problem) 이라 불린다[3].

본 관리 제어에서는 이 문제를 해결하기 위해 TAS (Test And Set) 기능을 이용하여 읽고 쓰는 규칙을 구현하여 통신 방식을 좀더 신뢰성있게 구성하였다[4].

## 4. 진단 기능및 예러 처리

고속 정밀 서보 시스템에 의해 동작되는 자동 조립 시스템의 생산성을 높이기위해선 한 싸이클 시간을 줄이는 것도 중요하나 또 하나의 중요한 기능은 시스템의 안정성을 높이고 고장 수리 시간을 줄일 수 있는 진단기능(Diagnosis Capability) 이라 할 수 있다. 본 관리제어는 자동조립 시스템의 각 프로세서 모듈은 자기 진단할 수 있게 하였으며, 입출력모듈과 각 기계부는 자기 진단이 불가능하여 사용자가 동작을 확인하여 진단하게 하였다 [4]

### (1) 각 프로세서의 모듈별 자기 진단

#### 1) 각 프로세서의 자기 진단

각 프로세서의 레지스터및 내부의 데이터 버스, 연산자(Arithmetic Logic Unit), 예외(Exception) 처리 기능등을 자기 진단한다.

#### 2) 메모리부 자기진단

각 프로세서 모드의 롬(ROM)은 모니터 롬으로 테스트가 불가능하므로 램(RAM) 부분만 테스트하였다. 메모리 테스트는 메모리가 큰 경우 시간이 많이 걸리기는 하지만 처음 전원 공급시 한번만 수행하므로 큰문제는 되지않는다. 단, 메모리 테스트 프로그램이 있는 곳은 테스트가 불가능하므로 그 블록은 그냥 넘어 가도록 하였다. 이 블록은 테스트 프로그램을 수행하면서 저절로 된다고 할 수 있다.

#### 3) 버스부및 타이머부 자기 진단

버스는 데이터 bus와 어드레스 bus로 나누어 테스트하였고 타이머는 10 msec 마다 인터럽트를 걸어와 다른 프로세서사이의 통신과 키 입력 여부를 검사하는 일을하는데 이 타이머가 정상적으로 인터럽트를 거는지 진단한다.

### (2) 입출력 모듈의 진단 기능

입력 모듈은 키 입력과 센서 입력 부분으로 나뉘 수 있다. 키 입력 진단은 각 키의 배열 모양을 모니터에 "H" 를 찍어 표시하고 사용자가 키를 누르면 눌린 키는 "L" 로 바뀌게 하여 키입력을 테스트한다. 센서 입력은 각 기계장치별로 그곳에 설치되어 있는 센서의 상태를 모니터에 나타내고 사용자가 직접 센서들을 하나씩 작동하게 만든후 그 센서의 현재 상태가 모니터에서 변하는 가를 보아 쉽게 고장 유무를 알 수 있게 하였다.

출력 모듈에는 온-오프(On-Off) 제어용 릴레이 구동부가 있다. 고장 여부는 각 릴레이 구동부를 사용자가 모니터의 지시대로 하나씩 구동시켜 그 동작을 직접 확인함으로써 진단한다.

### (3) 각 기계 부위의 기능 확인

자동 조립 시스템의 대부분의 구동 부위들이 모터를 사용하므로 각 기계 부위의 진단은 모터 제어 모듈의 진단과 함께 이루어 진다. 즉 각 기계 부위를 구동 시키면서 각 부위가 제대로 작동하는가와 함께 그 모터를 직접 구동시키는 서보 모듈도 진단 하게된다. 진단 방법은 각 구동 부위를 정해진 동작을 시키면서 사용자가 직접 확인한다.

### (4) 에러 처리

자동조립 시스템의 에러를 자기 진단시 발생한 에러, 자동조립 과정에서 발생하는 에러, 리미트 센서에 의해 발생하는 위급한 에러로 나누어 처리 하였다. 각 에러에 대한 처리순서는 다음과 같다. i) 에러 감지(Error Detection) 과정 : 에러의 발생을 알아내는 과정으로 자기진단이나 센서 상태의 이상등으로 알아낸다. ii) 에러 분리(Error Isolation) 과정 : 한번 발생한 에러가 다른 영역으로 확산되지 않게 하는 것으로 자동 조립 시스템의 경우 에러가 난 영역과 관련된 모든 기계 장치를 세워야 한다. iii) 에러 복구(Error Recovery) 과정 : 에러로 인한 에러의 영향을 제거하고 에러가 발생하기 이전의 상태로 전환시킨다. 각 에러에 대한 처리는 다음과 같다.

#### i) 자기 진단 에러

이 에러는 앞장에서 설명한 시스템의 진단 기능에 의한 에러 즉 프로세서 모듈의 에러는 자기 진단 할 때 감지하고 입력 센서나 출력 릴레이등과 각 기계부위의 진단은 사용자가 하나하나 확인하여 진단하는데 이 진단은 처음 시스템을 셋팅할 때 한번만 하는 것이며 미리 에러를 찾아 내는 기능이므로 레벨을 가장 낮은 0으로 하였다.

#### ii) 자동조립 과정중의 에러

이 에러는 자동조립을 수행하는 중에 운영자의 조치가 필요하게 되는 사건을 말한다. 예를 들어 자동조립 수행중 동작이 이상하거나, 부품이 떨어져서 더 이상 작업을 못할 경우등이 있다. 부품이 떨어진 경우는 업릴한 의미의 에러는 아니지만 시스템의 정상 작업을 막고 있는 원인이 되어 운영자가 처리를 해주어야 하므로 에러의 개념에서 처리 하였다. 이 레벨 1의 에러는 사용자가 그 조치를 취해주면 바로 계속 정상 작업으로 들어갈 수 있게 구성하였다.

#### iii) 리미트 센서 에러

이 에러는 시스템이 오동작으로 인해 각 기계 부위마다 허용 동작 범위를 벗어난 경우 동작 범위 밖에 설치되어 있는 리미트 센서에 의해 인터럽트가 걸려 긴급 상태(Emergency State) 에서 수동 또는 자동으로 복구한다. 이 에러는 아주 위급한 에러이므로 레벨 2로 구분하고 인터럽트 레벨을 가장 높게 하여 빨리 시스템을 안전하게 만들 수 있게 하였다.

## 5. 결론 및 추후 연구 과제

본 논문에서는 여러개의 프로세서를 사용하는 자동 조립 시스템의 관리 제어를 C 언어를 이용하여 구조적으로

구현하고, 진단기능을 갖게하였다. 또 실제로 현재 개발 중에 있는 다이 본딩 시스템에 적용하여 본 논문에서 제시한 구조의 관리 제어가 작성하기도 쉽고, 확장성과 디버깅 기능면 에서 뛰어남을 확인하였으며 특히 계층 구조는 큰 시스템의 개발에 있어 여러 사람과 공동 작업을 할때 유리하고 새로운 기능의 추가나 삭제등이 쉽게 되어 점점 기능이 확장되어 가는 자동 조립 시스템 분야의 관리 제어로서 적합함을 확인하였다.

또 이러한 관리 제어의 구조성을 바탕으로 진단 기능을 강화하여 시스템을 쉽고 안전하게 쓸 수 있는 관리 제어 시스템을 구현하였다. 특히 프로세서 모듈의 자기 진단 기능을 포함하여 시스템의 모든 모듈을 진단 가능하게 하였다.

본 논문은 본문에서 제시한 구조의 자동조립 시스템에는 모두 적용 가능하여 공장 자동화의 일원으로 산업계에서 추진중에 있는 자동조립 시스템 분야의 발전에 이용될 수 있다고 생각한다. 앞으로는 자동조립 시스템이 하나의 독립된 시스템으로 보다는 전체 생산 라인의 한 요소로 존재하게 되어 여러 자동조립 시스템의 네트워크화 요구되므로 지금의 구조위에 공장 자동화용 네트워크로 가장 널리 알려져 있는 MAP(Manufacturing Automation Protocol)에 부응할 수 있는 네트워크기능의 구현이 미래의 관리 제어에 필요하다고 말할 수 있다.

### 참고 문헌

- [1] P.H. Enslow JR., "Multiprocessor Organization - A Survey", Computing Surveys, Vol9, No.1, Mar, 1977
- [2] 과학기술처 "반도체 소자용 자동 다이본딩 장치의 개발", 2차년도 최종보고서, 1988
- [3] Comer, "Operating System Design : The XINU Approach", Prentice-Hall, 1984
- [4] 이재혁, "자동 조립 시스템을 위한 진단 기능을 갖는 구조적 관리 제어에 관한 연구", 한국과학기술원, 1988
- [5] Dhanajay Brahme, Jacob A. Abraham, "Functional Testing of Microprocessors", IEEE Trans. on Computer, Jun, 1984, pp475-485
- [6] EliT. Fathi and Moshe Krieger, "Multiple Microprocessor Systems : What, Why and When", Computer, Mar, 1983, PP 23-32
- [7] 김덕진, 김영천, 박석천, "빌 결합 멀티 프로세서 시스템의 구현 및 성능 평가", 대한 전자 공학회 논문지, 24권, 5호, 9월, 1987
- [8] G.D. Bergland, "A Guided Tour of Program Design Methodologies", Computer, Oct, 1981, pp13-37
- [9] Randall W. Jensen, "Structured Programming", Computer, Mar, 1981, pp31-48
- [10] Edward Yourdon, "Techniques of Program Structured and Design", Prentice-Hall, 1975
- [11] 이 현, "Fault Tolerant Computing System", 전자 교환 기술, 1권, 1호, 1985, pp 46-61.

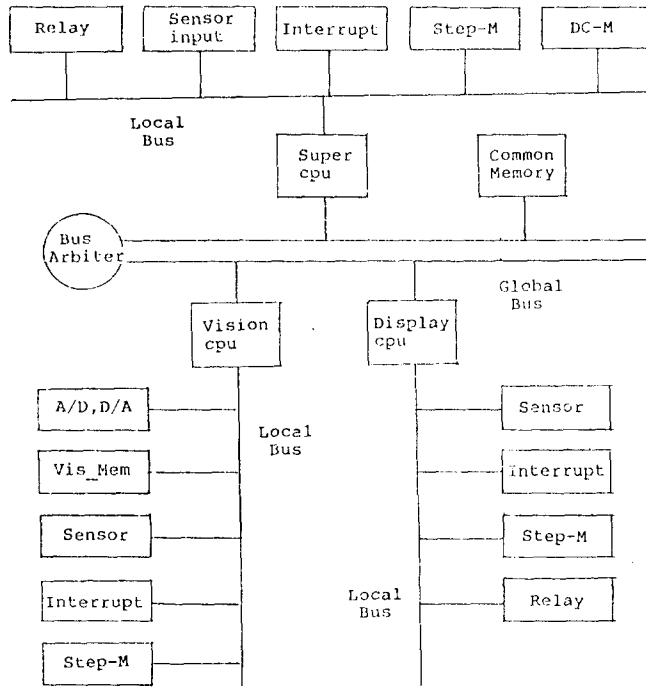


그림 1 다이본딩 머신의 하드웨어 구조  
 Fig 1 Hardware Structure of Die Bonding Machine

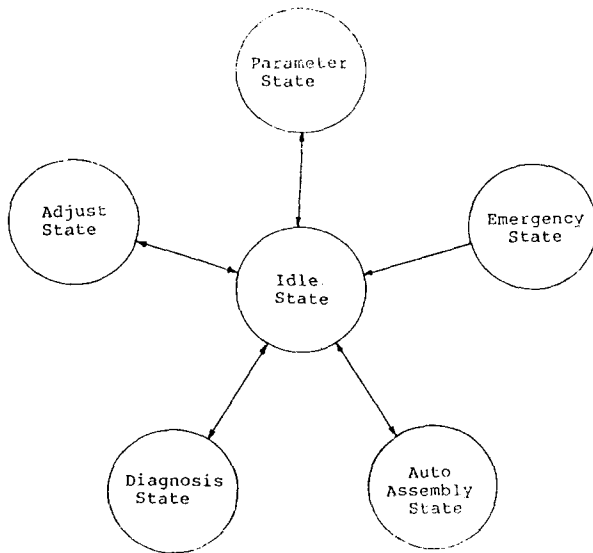


그림 2 관리제어의 6가지 스테이트  
 Fig 2 6 States of Supervisory Control

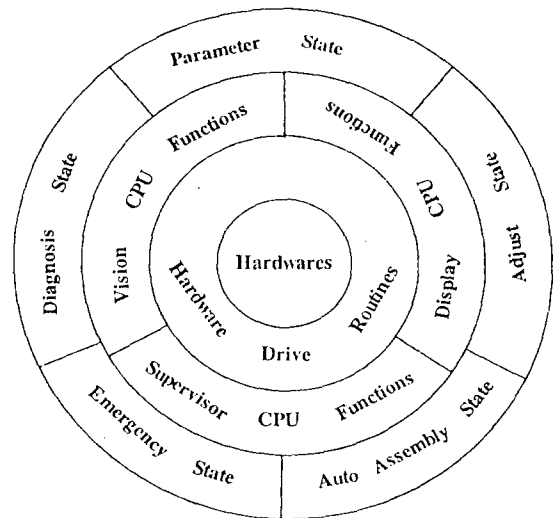


그림 3 관리제어의 계층 구조  
 Fig 3 Layer Structure of Supervisory Control