

## PC 인터페이스가 가능한 스카라형 로봇제어기의 개발

고경철\* 임태균\* 범희락\* 조형석\* 정용섭\*\*

\* 한국과학기술원 생산공학과 제어및 자동화 실험실

\*\* 주식회사 한독 자동화 사업부

## DEVELOPMENT OF A PC-Based SCARA ROBOT CONTROL SYSTEM

K.C. Koh\* T.G. Lim\* H.R. Beom\* H.S. Cho\* and Y.S. Jeong\*\*

\* Chonryang P.O.Box 150, Production Eng., KAIST, Seoul, Korea

\*\* Automation Dept. Handok Co. Ltd., Buchonsi kyungkido, Korea

## ABSTRACT

Recently, the robot control systems are required to be more flexible and intelligent in order to execute more complex and delicate tasks. As an approach to such system, a PC-based robot control system is presented in this paper. The axis controllers are independently designed for each joint of robot manipulator and are supervised by a personal computer. Therefore, the almost system program including the control algorithm can be easily developed and modified in the PC's environment. This proposed control system was successfully applied to 4 axis SCARA-type robot and the associated S/W modules were developed. We can construct sensory robot systems by simply connecting the sensing devices to the PC and the study on this now being progressed.

## 1. 서론

최근 산업 공정에 있어 생산 기술의 향상 및 원가 절감을 위해, 산업용 로봇을 이용한 생산 시스템의 자동화가 활발히 진행되고 있다. 특히 소형 부품의 조립 공정에 스카라형 로봇이 많이 사용되고 있으며, 그 응용 범위는 점차 확대되고 있다. 이러한 산업용 로봇은 단일 공정 또는 단일 작업 스테이션에서 작업하는 형태에서 점차 여러 스테이션을 총괄하는 일종의 cell controller의 역할까지 할 수 있도록 그 기능이 점차 확대되고 있다. 따라서 좀 더 복잡하고 지능이 요구되는 공정의 자동화를 위해서는 고도의 센서(시각, 접촉) 장치와의 연결이 용이하고, 이러한 주변의 system제어가 가능한 로봇시스템이 필요하다. 즉 기존의 로봇시스템도 단순 반복 기능보다는 어느정도의 지능과 감지능력을 갖도록 요구되고 있다. 이러한 기능의 추가는 로봇의 생산단계에서 이루어지는 방법과 시스템엔지니어 또는 사용자에 의해 이루어지는 방법이 있을 수 있다

그러나 센서시스템의 선정이나 이에 관련된 프로그램은 작업대상에 따라 좌우되므로, 사용자가 직접 작업공정에 맞는 로봇시스템을 재구성하는 경우가 빈번하다. 이때 필수적으로 발생하는 문제는 사용자가 임의의 센서시스템을 로봇에 연결할 때, 필수적인 센서정보의 처리 및 제어 알고리즘등의 S/W를 이미 주어진 로봇환경내에서 구성할 수 밖에 없는 제한성이다. 예를 들어 PUMA 로봇의 경우, 로봇에 관련된 모든 명령은 VAL-II

에 의해 제공되며, 외부시스템간의 통신은 이러한 제한된 명령만을 사용하여 이루어진다. 만약 비전시스템으로 부터 얻은 데이터에 의해 로봇의 경로를 변경하고자 할때, VAL-II가 제공하는 보간명령(MOVE 또는 MOVES)을 이용할 수 밖에 없으며, 새로운 경로의 설정이나 다양한 기능의 추가가 제약을 받는다.

이러한 문제점을 극복하기 위해, 기존의 로봇시스템에 유연성을 부여하려는 여러 연구결과들이 있었다. G. F. Mauer등[1]은 PC-based Robot Controller의 개발을 통해, 센서시스템의 접속을 PC에 하도록 하고, PC와 로봇간의 통신은 시리얼방식에 의해 VAL-II명령을 주고받도록 하였다. 이는 경로계획부분을 PC가 담당하도록 하여 새로운 경로설정 및 보간기능을 사용자가 직접 설계할 수 있으나, 안전한 유연성을 갖는다고 볼 수 없다. A. A. Godenberg등[2]은 자체 제작한 외부 컴퓨터를 PUMA의 조인트제어부에 직접 접속하여, VAL-II환경으로 부터 독립하였으며, 각조인트의 각도 계산을 위한 Inverse kinematics식을 포함한 모든 로봇관련 프로그램을 사용자가 직접 작성할 수 있도록 하였다. 또한 이에 따르는 계산적 부담을 줄이기 위해, 병렬프로세서를 이용하여 컴퓨터의 연산기능을 고속화하였다. 그러나 이 경우 기존의 PUMA에 있는 조인트 제어루트를 사용하므로 통신을 위한 별도의 회로가 필요하며, 제어알고리즘에는 유연성이 결여되는 단점이 있다. D. M. Patel[3]등은 AT&T사의 DSP-32를 사용하여 직접 조인트제어부를 구성하고 이를 Dual-port RAM을 통해 IBM-AT에 접속하는 구조를 제안한 바있다. 그러나 이방법은 PC와 조인트제어부가 버스를 공유하는 구조를 채택하여, 시스템의 설계면에서 다소 복잡해지고 시스템의 자유로운 변경이 어려운 단점이 있다.

본 연구에서는 이러한 문제점들을 해결하기 위해, 구조적으로 간단하고, 보다 큰 유연성을 갖는 PC-based 로봇 제어시스템을 제안하고자 한다. 기본적인 설계개념은 주제어부로서 PC를 사용하고 조인트 제어부를 병렬 입출력 포트를 통해 PC와 접속함으로써 H/W구조를 간단화하는 것이다. 이방법은 주제어시스템의 변경이 용이한 장점도 갖고있다. 이러한 설계방식의 효율성을 입증하기 위해 스카라형 4축 로봇을 위한 제어시스템을 제작하였으며, 이의 구동을 위한 시스템 프로그램을 개발하였다. 본 논문은 시스템 전체구성 및 설계 그리고 관련된 시스템 프로그래밍에 대해 기술한다.

## 2. 전체 구성

### 2.1 개요

일반적으로 로봇은 몸체부와 제어기로 구성되며, 제어기의 하드웨어는 대부분 멀티 프로세서로 구성되는 계층 구조 방식으로 설계된다. 이는 크게 주제어부와 축제어부로 나누어지는데, 주제어부는 전체 시스템을 총괄하고 위치 제어를 제외한 대부분의 로봇 제어에 필요한 기능을 수행한다. 축제어부는 조인트축마다 각각의 독립된 마이크로 프로세서를 채용하여 설계되고, 위치제어 및 주제어부와의 통신등을 담당한다.

### 2.2 PC-based 로봇시스템

“PC based Robot Control System”이란 현재 광범위하게 운용되고 있는 퍼스날 컴퓨터를 그대로 로봇제어시스템의 주제어부로 사용하는 것으로 전술 한 바와 같이 시스템의 유연성을 갖는 구조이다. 이는 로봇시스템을 구성하는데 필수적인 요소들중에서 범용 컴퓨터가 보유하는 Graphic display, Mass storage device 등의 하드웨어부와 고급언어체계, I/O, Screen editor 등과 같은 S/W 들을 그대로 활용할 수 있어, 로봇시스템에 국한된 S/W 및 H/W의 개발에 더 많은 연구와 투자를 집중할 수 있는 장점이 있다. 즉 PC에서 제공되는 고급언어를 사용하여 로봇제어 프로그램을 작성할 수 있으며, 모든 S/W의 개발이 단일 시스템상에서 이루어져 개발기간의 단축 및 S/W의 수정/확장이 용이해진다. 로봇의 성능평가에 있어서도 외부기기와의 접속이 자유로워, 성능평가 시스템의 구성이 가능하다. 또한 최근들어 PC접속이 가능한 각종 계측 장비 및 IBM-PC 슬롯용 확장보드들이 많이 개발되고 있으므로 이들을 이용한 시스템의 종합화가 가능하다. 그림.1은 이러한 종합적인 로봇시스템의 한예를 보이고 있다.

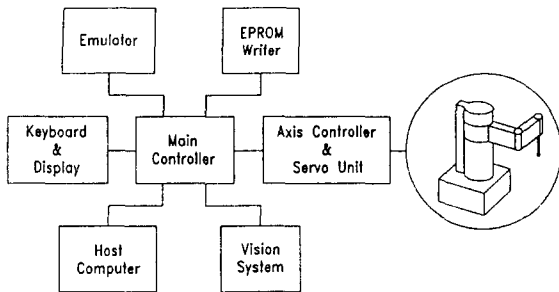


그림.1 PC를 주축으로한 종합적 로봇시스템의 일예

### 2.3 시스템 구성

본 연구에서 제안된 시스템은 그림. 2와 같이 범용 PC(IBM PC-AT)가 상위의 주제어부가 되고, 각축의 위치제어를 위해 독립적으로 설계된 여러 개의 축제어 보드들과 각각의 대응되는 모터 서보 Amplifier들이 하위의 축제어부를 구성하는 구조를 갖는다.

### 2.4 멀티 프로세서 간의 통신

제어시스템의 유연성있는 설계에 있어 중요한 것이 멀티 프로세서간의

통신방식이다. 이는 대부분 공용 버스를 통한 메모리 공유 방식이거나 Dual port RAM 방식을 이용하여 설계된다. 그러나 이방식은 축제어부를 주제어부가 채택하고 있는 버스방식에 따라 설계해야 하므로, 축 제어보드의 설계상의 유연성이 결여되는 단점이 있다. 이러한 문제점을 해결하기 위해 표준 통신방식의 하나인 Parallel I/O 방식을 채택하여 주 제어부와 축제어부의 통신이 이루어지는 방식을 제안한다. 이러한 방식의 장점은 하위의 축제어부와 상위 주제어부가 각각 독자 버스를 갖고 설계될 수 있으므로, 한번 개발된 축제어부를 주제어부의 변경시에도 계속 이용할 수 있는 것이다. 이때 단지 필요한 것은 새로운 주 제어부가 축제어부와 같은 표준 I/O 포트를 갖도록 하는 것이다.

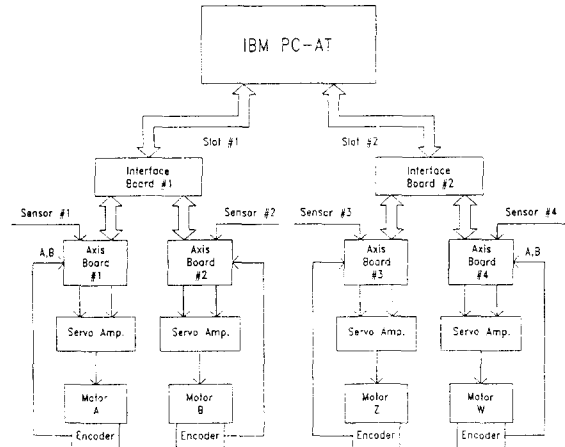


그림.2 제안된 PC-based 로봇제어시스템의 구조

인터페이스보드의 설계는 축제어부가 축마다 독립된 프로세서를 갖고 있으므로, 이들과 각각 통신을 하기위한 멀티통신 포트를 PC에 부여하기 위해 필요하다. 이와 아울러 PC가 외부센서등으로부터 다양한 데이터를 받을수 있도록 A/D 및 D/A 변환회로, 카운터등의 부가기능도 갖도록 인터페이스 보드를 설계하였다. 이러한 Interface Board의 블록구성도는 그림.3과 같다. 이 보드는 3개의 PIO확장회로를 갖고 있으며, 하나의 PIO가 조인트제어부의 한축 제어보드와의 양방향 통신을 위해 사용된다. 따라서 하나의 인터페이스 보드는 최대 3축과 접속할 수 있으며, 좀더 많은 제어축수를 증가는 이 보드의 갯수를 증가시키는 것으로 가능하다. 본 연구는 4축제어를 대상으로 하였으므로 2장의 인터페이스보드를 필요로 한다.

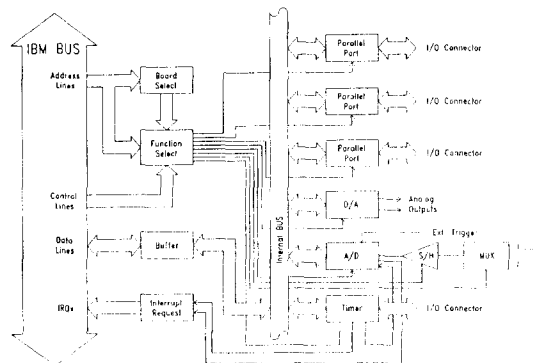


그림. 3 Block diagram of interface board

## 2.5 축제어부의 설계

축제어부는 로봇의 각 조인트의 위치제어를 수행하는 부분으로, 각 축마다 독립적으로 설계된다. 이는 크게 모터의 구동을 위한 아날로그 앰프와 디지털 위치제어기(축제어보드)로 구성된다. 이때 축제어보드는 주 제어부로 부터 매 샘플링 주기마다 위치지령값을 받아 이를 제어루프의 기준입력으로 사용하고, 제어되고 있는 모터의 위치값을 다시 주제어부에 보내주는 기능을 한다. 이러한 양방향 통신은 전용한 바와 같이 PIO를 통해 이루어지므로, 이 SPEC에 맞는 축제어보드를 새로이 설계할 필요가 있다. 이러한 축 제어기보드의 설계블록도는 그림.4와 같으며 CPU로서 INTEL MCS-51 계열의 8032를 사용하였다. 8032는 시리얼통신및 입출력포트 그리고 외부카운터등을 내장하는 8비트계열의 마이크로콘트롤러로서, 모터의 제어에 적합한 것으로 알려져 있다. 축제어에 필요한 S/W는 PLM/51을 사용하여 작성하였으며, 이를 외부 EPROM에 내장하였다.

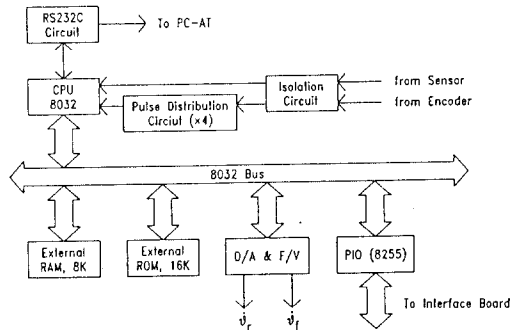


그림.4 축제어보드의 설계 블록도

## 3. 시스템 프로그램

### 3.1 개요

지금까지 설명한 시스템을 가지고 로봇을 제어하기 위해, 사용자는 로봇에 관련된 시스템 프로그램을 자유로이 작성할 수있다. 그러나 이는 사용자가 시스템의 하드웨어구조및 로봇자체에 대해 완전한 이해를 해야 가능한 것이다. 따라서 이에 관련된 기본명선을 모듈화하여 라이브러리로 구성하면, 일반 PC에만 익숙한 사용자도 원하는 소프트웨어의 개발을 용이하게 할 수 있다. 본 연구에서는 일제로써 현재 상용화 된 로봇이 갖는 기본적 기능에 대한 시스템프로그램을 Quick-basic언어를 이용하여 서브루틴들로 구성하였다. 이 루틴들은 크게 4개의 레벨로 분류되어지며 각각의 레벨별로 설명하기로 한다.

### 3.2 통신 관련 루틴 (1st level)

가장 하위의 루틴들로서, 인터페이스보드를 드라이빙하며, 축제어부와 의 통신을 수행한다. 이 루틴들의 추가및 변경을 위해서는 시스템의 하드웨어구조에 대한 이해가 필요하다.

- HARDSET: 인터페이스 보드상의 PIO들을 초기화
- TRANSC: 지정한 AXIS에 1-Byte의 데이터를 보냄
- TRANSV: 지정한 AXIS에 2-Byte의 데이터를 보냄

- RECIVC: 지정한 AXIS로부터 1-Byte의 데이터를 받음
- RECIVW: 지정한 AXIS로부터 2-Byte의 데이터를 받음

### 3.3 축제어명령 루틴(2nd level)

이는 위의 통신루틴들을 이용하여 작성되며, PC가 축제어부에 내리는 명령 프로토콜은 축제어보드와 PC간에 사전에 정의하여, 축제어부를 프로그램해야 한다. 따라서 사용자가 이 레벨에서 프로그램하기 위해서는 (새로운 명령 추가등) 축제어부를 구조및 S/W를 변경 할 수 있어야 한다.

- ZEROS: 지정한 AXIS에 원점복귀 명령
- MOVE: 지정한 AXIS의 기준위치값을 Update하고 현재의 제어된 모터의 위치변화 값을 읽어 들임.

### 3.4 로봇관련 루틴(3rd level)

이 레벨은 실제 로봇의 운동에 관련된 루틴으로 로봇에 대한 기구학및 경로계획법등이 포함되며, 또한 각 조인트부의 변수들(암길이, 감속비, 엔코더의 분해능, 모터의 최대 회전속도등)을 고려 해야한다.

(운동 관련 루틴)

- MOVEJOINT: 로봇을 조인트 좌표계에서 구동
- MOVELINEAR: 로봇을 직각좌표계에서 직선보간으로 연속 경로 제어
- MOVECIRCLE: 로봇을 직각좌표계에서 원호보간으로 연속 경로 제어

(연산 관련 루틴)

- INVKINEM: 로봇핸드의 직각좌표계의상의 위치로 부터 각축의 조인트각을 계산
- DIRKINEM: 각축의 조인트각으로 부터 로봇핸드의 직각좌표계의상의 위치를 계산
- CALCENCO: 조인트각도(Radian)로부터, 엔코더값의 계산(펄스 수)
- CALCANGLE: 엔코더값으로부터 조인트각의 계산
- RANGECHK: 주어진 조인트각이 로봇의 운동범위내인가를 검사
- VELOCHK: 조인트각의 속도가 정격속도범위내인가를 검사

### 3.5 타스크관련 루틴(4th level)

이 루틴은 실제 로봇이 해야할 작업내용을 묘사하고, 이를 수행하는 부분으로, 주로 Man/Machine interface에 관련되므로 프로그래머의 취향에 따라 다르게 작성 될 수 있다. 본 연구에서는 티칭플레이백방식으로 로봇을 운전하는 경우를 고려하여 다음과 같은 루틴들을 설계하였다.

(티칭 관련 루틴)

- TEACHING: 각축을 키보드를 이용하여 수동조작(속도, 보간, 이동량)
- RECORD: 현재의 로봇위치및 운동조건(속도, 보간모드, 가감속조건등)과 함께 티칭버퍼에 기억
- TESAVE: 티칭버퍼의 내용을 File로 저장

- TELOAD: File로부터 티칭데이타베이스를 티칭버퍼에 로딩
- DISTEACH: 티칭버퍼의 내용을 화면에 출력

(플레이백 관련 루틴)

- MOVESTEP: 티칭버퍼의 내용을 한 스텝씩 수행
- PLAYBACK: 티칭버퍼의 내용을 지정 횟수만큼 연속적으로 수행
- DISRESULT: 현재 운전되고 있는 상황(실제 위치, 스텝수등)을 화면에 출력

#### 4. 성능 테스트

##### 4.1 개요

현재 개발된 시스템의 성능을 평가하기 위해, 우선 기존의 플레이백/티칭 타입의 로봇기능을 재현할 수 있도록 프로그램을 작성하여 위치재현도 실험 및 연속경로제어실험을 수행하였다.

##### 4.2 위치 재현도 실험

로봇시스템을 새로운 하드웨어로 설계할 때, 가장먼저 확인해야 할 점은 첫째, 축제어보드가 모터의 엔코더신호를 분주하여 이를 카운터회로를 통해 카운트할 때, 노이즈의 영향을 받지 않고, 정확히 모터의 위치를 계산할 수 있는가이며 둘째, 주제어부와 축제어부의 통신이 한치의 오차없이 수행되는가이다. 만약 이들중 어느 하나의 문제가 있어도, 로봇의 위치정밀도에 심각한 타격을 주게 된다. 그러나 이를 확인할 구체적인 방법은 실제 무수의 반복실험을 통해 위치재현도를 측정하는 방법밖에 없다.

본 실험에서는 두점의 서로 다른 포인트를 조인트모드로 티칭하여, 이를 최대 속도로 플레이백동작을 시키므로써 시스템의 신뢰성을 평가하고자 한다. 이러한 신뢰성 평가의 구체적인 기준은 국내에는 아직 설정되지 않은것으로 보이며, 따라서 위치 반복실험은 가장 넓은 범위의 두점 사이를 최대속도로 조인트보간모드로 동작시키고, 양 끝점에 다이얼 게이지를 설치하여 로봇핸드의 위치측정을 통해 수행하였다. 이러한 방식으로 200번 실험한 결과, 본 시스템의 위치재현도는 +/- 0.025mm 이내임을 알 수 있었고, 위치제어 및 통신의 신뢰성을 확인하였다.

##### 4.3 연속경로 제어실험

연속경로모드(직선 및 원호보간)는 앞서의 조인트보간과 달리 로봇의 기구학적 방정식(Kinematics 및 Inverse kinematics)의 계산이 온라인으로 수행된다. 따라서 이모드의 실험을 통해 본시스템의 능력을 평가하고자 한다. 주어진 타스크는 그림.5와같이 4점을 서로 잇는 직선경로를 추종하는 것으로 하였다. 이때 경로상의 중간포인트의 계산 및 이점들에 대한 조인트각의 계산은 16mSEC내에서 수행되었으며, 매 샘플링 타임마다 계산된 각모터축의 위치를 축제어보드에 보내주게 된다. 그림.5(a)-(d)는 경로상의 속도를 증가시켜 가면서 주어진 경로에 대한 추종결과를 보여주고 있다. 여기서 저속운전시 좋은 경로제어 결과를 보이나, 속도가 높아질수록 추종정능이 떨어지는 것을 알 수 있다. 이 원인은 속도가 높아지면 그만큼 중간포인트의 갯수가 줄어들게 되어 치밀한 기준경로를 발생시키지 못

하는 것과 경로가 바뀔 때 급격한 속도변화로인해 시스템의 불안정성에 의한 경로오차가 커지는 것으로 설명할 수 있다. 고속에서의 경로오차를 줄이는 방법은 시스템의 고속화, 경로설계시 가감속을 고려하는 것과 제어알고리즘을 개선하는 방법이 있다. 즉 추종제어 향상을 위한 피드포워드설계의 추가등이 있을 수 있다.

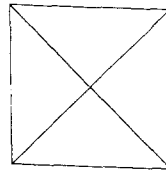


그림 5-A 속도 No.1 일때의 직선 보간

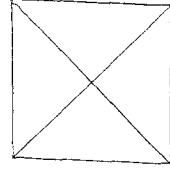


그림 5-C 속도 No.6일 때

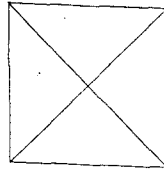


그림 5-B 속도 No.3 일때의 직선 보간

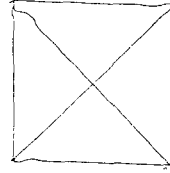


그림 5-d 속도 No.9 일때

#### 5. 결론

본 연구에서는 PC를 이용하여 유연성을 갖춘 로봇시스템을 제안 하였다. 비교적 간단한 하드웨어구조로써 시스템을 구성하였고, PC상에서 수행될 수 있는 시스템프로그램을 일반고급언어를 사용하여 개발하였다. 이는 4레벨의 서브루틴들로 구성되어 각레벨에 따라 사용자가 로봇을 위한 새로운 기능을 추가할 수 있도록 하였고, 주변기기와의 접속이 수반되는 S/W를 어느 레벨에서도 작성할 수 있도록 하였다. 이러한 시스템의 호용성을 알아보기 위해 위치반복 실험 및 경로 추종실험을 수행하였다.

현재 까지 진행된 연구는 우선 로봇이 갖는 기본적인 기능에 대한 것에 불과하나, 앞으로 이 시스템을 각종 센서시스템과 연결하여 종합적인 지능로봇시스템(Integrated Sensory Robot System)을 구현하고, DSP보드를 이용한 PC의 연산강화를 통해 좀더 진보된 제어알고리즘의 관한 연구를 할 예정이다.

#### REFERENCE

[1] G.F. Mauer, J. Skaggs and R.M. Turner, "A Flexible PC-Based Robot Controller", SME conf., 1989

[2] A.A Goldenberg and L. Chan, "An Approach to Real-Time Control of Robots in Task Space. Application to Control of PUMA 560 without VAL-II, IEEE Tran on Industrial electronics, Vol.35, No.2, 1988

[3] D.M. Patel, "High Speed Floating-point Robot Controller" conf. on Applied Motion Control, 1987

[4] J.D. Kossman and A. Malowany, "A Multi-processor Robot Control system for RCCL under iRMX", IEEE conf. on R & A, 1987

[5] Y.F. Zheng, J.Y.S. Luh and P.F. Jia, "A Real-Time Distributed Computer System for Coordinated-Motion Control of Two Industrial Robots", IEEE conf. on R & A, 1987