

김 대원, 고 명삼, 이 범희

서울대학교 공과대학 제어계측공학과 로봇틱스 및 지능 시스템 연구실

An Investigation of Robot Programming Language with the Capabilities of Sensory Information Processing

Dae Won Kim, Myoung Sam Ko, Bum Hee Lee

Robotics & Intelligent Systems Lab., Dept. of Control & Inst. Eng. College of Engineering, Seoul National University

Abstract

In this paper, among the robot programming languages that enable processing of sensory information, eight exemplary languages are chosen, and investigated in terms of their characteristics, why they are designed the way they are, and the kind of sensors each language can use and apply to. In addition, the characteristics of each language is compared with one another from the sensor point of view and the flow of each language is analyzed from the robot language classification point of view. Finally, we investigate the progress and the requirements of the sensor-based robot programming languages for further developments.

1. 서 론

최근 컴퓨터 산업의 발달과 함께 급속하게 성장되어 전 세계적으로 주목을 받고 있는 연구가 진행되고 있다. 이와 함께 로봇 매니폴레이터가 어떤 복잡한 다양한 작업에 하드웨어의 변형없이 수행하도록 하기 위해 소프트웨어 즉, 로봇 프로그래밍 시스템에 의한 해결 방법이 여러 각도에서 연구되어 지고 있다. 로봇 프로그래밍 시스템의 목적은 하나 또는 그 이상의 로봇의 동작과 센싱 기능을 규정해 주는 언어를 제공하고, 그 언어를 해석하는데 있다. 그러나, 이때 정의되는 언어는 두가지의 상반되는 점을 안고 있다. 다시 말하면, 그 언어 자체가 간단하거나 아닌 일만 사용자에게 쉽게 수용될 수 있어야 한다는 점이고, 그와 반대로 아무리 복잡한 작업의 지라도 전문가에 의해 프로그래밍될 수 있을 정도의 작업 표현력을 지녀야 한다는 점이다. 이러한 두가지의 상반되는 필요에 따라 각기 다른 접근 기호를 갖는 다양한 종류의 로봇 프로그래밍 언어가 개발되게 되었다[1].

과거 산업용 로봇의 응용에 있어서는 응용 영역이 제한되고, 단순해서 단지 교시와 반복에 의한 방법만으로도 충분하였지만 자동차 산업이나 우주 산업과 같은 경우에는 각지점들을 개별적으로 교시한다는 것이 몹시 어렵고, 지루하며, 많은 에러를 발생시키는 작업이 된다. 이에 따라 교시와 반복에 의한 방법을 능가하는 교과서적인(textual) 로봇 프로그래밍 언어의 개발이 이루어졌는데, 이는 CAD/CAM 데이터베이스를 이용할 수 있어서 오프-라인(off-line) 프로그래밍을 가능하게 해주며, 이로 인해 위의 문제점을 해결할 수 있게 된다. 또한 힘(force)이나 비전(vision) 센서의 정보를 실시간에 처리할 수 있는 장점을 지닌다.

또 다른 관점으로 보면, 로봇 프로그래밍 언어의 설계에 대한 두가지의 접근 방법이 있다. 한가지 방법은 로봇 제어 시스템의 어의론(semantics)에 초점을 맞추는 것이다. 여기서 각 구문(syntax)은 로봇이 어떻게 동작하는가를 묘사하기 위한 것과 기타 부수적인 언어의 구조를 포함하게 되는데, 여기에는 WAVE[2], AL[3], PAL[4], AUTOPASS[5] 같은 것들이 있다. 이 방법의 장점은 다양한 매니폴레이션(manipulation)을 묘사하기에 충분한 도구를 제공한다는 것이다. 그러나, 실제 로봇의 응용에 있어서는 단순한 동작보다는 그 이상의 많은 것들을 포함하게 되는데, 이는 기존의 프로그래밍 언어의 특성을 의미한다. 그래서 또 다른 방법으로 대두되는 것이 현존하는 일반 언어에 필요한 로봇 어의론을 부가해서 그 언어를 확장하는 것이며, 여기에는 RPL[6], MACLISP[7] 등이 있다[8].

이와 같이 로봇 언어에 대한 연구는 여러가지 접근 방법으로, 그리고 지속적으로 이루어져 왔지만 그러한 연구가 특정한 로봇 또는 그 응용의 필요를 충족시켜 주기 위하여 임기응변식으로 만들어져 왔기 때문에[9] 좀더 일반적인, 로봇에 독립적인(robot-independent) 소프트웨어의 개발이 필요하다. 다시 말해 많은 종류의 로봇이나 여러대의 로봇을 동시에 인터페이스(interface)시킬 수 있는 언어 및 시스템이 필요하다. 그러나 과거, 로봇 언어를 만드는 사람들은 그들 로봇과의 인터페이스 관계에 대한 자료를 문서화하지 않았다[10]. 그러나 궁극적으로는 로봇과 외부 시스템과의 인터페이스가 표준화 되고, 또 로봇에 독립적인 언어 시스템의 개발이 진행 될 것이며, 현재 일부에서 진행되고 있는 실정이다.

또한 로봇은 이제 그 로봇 자체에 대한 관심보다는 외부와의 인터페이스를 통한 전체 시스템의 구성 요소로서의 의미가 있다. 이러한 추세에 따라 로봇의 센싱 및 외부 기기와 인터페이스에 초점을 맞추고 로봇 프로그래밍 언어에 대하여 조사해 본다는 것은 의미있는 일이라 생각된다.

본문에서는 2장에서 로봇 언어의 역사적 변천에 대하여 살펴보고, 3장에서 센서 정보에 관련된 8가지 로봇 언어에 대하여 서술하며, 4장에서는 센서와의 관계를 중심으로 본 로봇 언어의 특성을 비교하고, 마지막 장에서 앞으로의 연구 방향을 제시한다.

2. 로봇 언어의 변천

1960년대 초창기 산업 로봇에서는 단지 교시와 반복의 방법에 의하여 단순한 반복 작업에 응용하였으나, 1960년대 후반에 들어서면서 산업용 로봇에 사용되는 오프-라인 프로그래밍 언어의 필요성이 대두되었고, Paul은 MIT의 Hand-eye 프로젝트에서 교시없이 로봇을 제어하는 방법을 처음 시도하였다[11,12,13].

1970년대에 접어들어, 미국 Stanford 연구 지능 연구소에서는 연구 목적의 실험적인 언어인 WAVE[2]를 개발하였고, 1974년 ALGOL에 기초를 둔 AL[3]을 개발하였다. AL은 힘 검출 능력이 있고, 기존의 컴파일러에 고시를 위한 프로그램이 추가되었으며[14] 로봇의 동작과 대상물(object) 묘사를 구제화하였다는 점에서 오늘날 로봇 언어의 한류가 되었다[15].

IBM T. J. Watson 연구소에서도 1976년 이레로 직각 좌표형(Cartesian) 미니플레이터를 제어하기 위한 몇가지의 언어를 개발하였다. 여기에는 EMILY와 ML이 있으며 조립 작업에 이용되었고, 작업-레벨(task-level)[9]의 언어인 AUTOPASS[5] 또한 개발하였다. 또 1982년에는 상업성을 띤 언어인 AML[8]을 발표하였다.

한편 최초의 상업적인 언어는 1979년 Unimation사에 의해 발표된 VAL이다[16,17,18]. 이는 BASIC 언어의 확장으로 PUMA 미니플레이터와 함께 제공되었다[11]. 그 이후 1982년에 관리(supervisory) 컴퓨터 및 비전을 비롯한 다양한 센서와의 상호 관계의 필요성에 따라 VAL-II[19]도 개발되었다.

Purdue 대학에서는 AL 이후 PAL[4]에 대한 연구를 시작하였는데, PAL에서는 모든 작업이 구조적 좌표(structured Cartesian coordinate)로 표현되며, 미니플레이터의 불필요한 정지를 배제하기 위하여 동작의 연속성과 컴플라이언스(compliance)를 허용하는 능력을 제공하였다.

그 밖에도 1981년 비전에 의한 검색(inspection)과 조립, 아르 용접을 위해 Automatrix에서는 RAIL[25]을 개발하였고, CAD 데이터를 이용하기 위해 NC 공작 기계의 언어인 APT의 확장으로 McDonnell-Douglas에서는 MCL[13]을 발표하였다.

1983년 이레로는 Seiko, Machine Intelligence, American Robots 그리고 Intelledex 등에서 BASIC을 기초로 하는 로봇 언어를 발표하였다[11].

3. 센서 정보 처리에 관련된 로봇 언어

이 장에서는 센서로부터의 정보 처리에 관련된 8 가지의 로봇 언어를 선정하여, 그 언어의 기능과 특성 그리고 설계 목적에 대하여 서술하고, 어느 레벨에 속하는가, 이용 가능한 센서와 그 응용, 그리고 사용 명령어에 관하여 서술한다.

(1) AL : 1974-현재 : SAIL(Stanford A. I. Lab.)

AL[3]은 ALGOL, PASCAL과 같은 상위 레벨 언어의 프로그래밍 특성 뿐만 아니라 로봇 프로그래밍을 위해 필요한 기능을 포함하는 상위 레벨의 언어로서 현재에 이르기까지 개발된 로봇 언어중 가장 완전한 언어중의 하나일 것이다. 이는 로봇 언어의 분류[9]에 의하면 구조 프로그래밍 (structured programming) 레벨에 속하나, 후에 발표된 Pointy를 통하여 작업-레벨의 특성도 포함한다. 이용이 가능한 센서로는 비전, 힘/토크(force/torque) 센서, 근접 리미트 스위치(proximity limit switch)가 있어서 실시간 언어의 기능 즉, 여러 프로세스의 동시 실행, 동기화(synchronization)와 조건 모니터링 등이 가능하며, 컴플라이언트 제어도 가능하고, affixment, 매카니즘을 이용한 대상물의 모델링도 가능하다[12]. 외부와의 I/O를 위해서 'SIGNAL', 'WAIT', 접촉 센서를 위한 'CENTER' 명령어, 힘 제어를 위해 'MOVE'와 함께 'WITH FORCE FRAMB', 'WITH FORCE', 'WITH TORQUE'를 이용하며, 조건 모니터를 위해 'ON < > DO < >' 명령어가 사용된다.

(2) AML : 1982-현재 : IBM

IBM에서 10년간 연구되었던 프로그래머를 자동화 시스템(programmable automation system) 연구의 경험을 기초로 상당히 진보된 프로그래밍 인터페이스 특성을 갖춘 새로운 언어인 AML[8]이 개발되었다. 이 언어는 구조가 훌륭한 언어로서, 풍부한 경험을 가진 프로그래머에 의해 사용이 편리하게 만들어진

서브루틴을 마치 시스템 명령어처럼 사용할 수 있도록 한 점이 이 언어를 설계한 목적이 된다. 로봇 언어의 본류에 따르면 구조 프로그래밍 레벨에 속하며, 이용 가능한 센서로는 힘/토크 센서, 손가락(finger)상의 light-beam-presence 센서, solid-state TV 카메라, 피이더(feeder)상의 empty indicator가 있다. 센서와의 I/O를 위해서 'SENSIO', 'DEPIO', 'DELIO'이 사용되고, centering grasp 서브루틴이 제공되며, 실시간 모니터링을 위한 명령어로 'MONITOR', 그리고 힘 제어에 관계된 'PINCH-FORCE' 서브루틴, 그 밖에 AML이 제공하는 서브루틴으로 'APPROACH MOVE', 'FINAL MOVE'가 있다.

(3) MCL : 1979-현재 : McDonnell-Douglas

MCL[13]은 NC 공작 기계의 언어인 APT(Automatic Programmed Tool)의 확장으로 로봇과 각종 센서, 공작 기계 및 매뉴팩چ어링 작업 단위(work cell) 내에 사용되는 각종 기기들을 제어할 수 있다. 이 언어는 로봇의 동작 규정, 영상 모델링의 기능을 가지며, 여러 종류의 토폴로지를 제어하고, 실시간의 사물 검정 기능과 비전 프로세싱 기능을 가진다. 로봇 언어의 본류에 따르면 구조 프로그래밍 레벨에 속하고, 이용 가능한 센서로는 비전과 근접 리미트 스위치가 있다. 여기서 실시간 의사 결정은 WHEN-BLSB 구조를 통하여 행하여지며, 비전 프로세싱에서는 프로그램 내의 모델과 카메라에 조각된 영상을 비교하여 실시간 의사 결정에 이용한다. 외부와의 I/O를 위해 'SEND', 'RECRIVE'가 사용되고, 이는 접촉 센서의 정보 이용을 위해서도 쓰이며, 비전 명령어에는 'REGION', 'PROJECT', 'LOCATE', 'INSPEC'이 있다.

(4) VAL-II : 1982-현재 : Unimation

VAL-II[19]는 이 언어의 전신인 VAL[16,17,18]의 기능을 훨씬 능가하는 회로망 통신, 확장된 센서 인터페이스 기능, 실시간 궤적 변경(real-time trajectory modification) 기능과 함께 상위 레벨의 컴퓨터 프로그래밍 언어의 수학적 기능을 갖추고 있다. 이 언어의 설계 목적은 회로망 내에서의 통신 기능과 복잡한 응용, 유연한(flexible) 로봇의 경로 제어, 그리고 일반적인 센서 인터페이스의 실현에 있었다. 로봇 언어의 본류에 의하면 구조 프로그래밍 레벨에 속한다. 이용할 센서로는 비전과 힘/토크 센서, 그리고 근접 리미트 스위치 등이며, 이들을 이용하여 컴플라이언트 제어와 비전에 의한 서보잉(visual servoing) 등이 가능하다. 접촉 센서를 위해 'REACT', 'REACTI', 'REACTE', 'SIGNAL', 'WAIT' 그리고 외부와의 I/O를 위해 'IGNORE', 'IFSIG', 'REACT', 'SIGNAL' 명령어가 사용되며 실시간 경로 변경을 위해서는 'alter mode'가 사용되고, 사용자의 알고리즘에 따라 경로 변경이 가능한 'procedural motion'도 가능하다.

(5) RCCL : 1984-현재 : Purdue 대학

RCCL[20,21]은 로봇 언어가 아니고, 로봇을 제어하기에 적합한 시스템 콜(call)의 집합이라고 할 수 있다. 이 시스템 콜은 미니플레이터의 작업 묘사, 센서와의 인터페이스, 광범위한 응용, 추적(tracking), 힘 제어 등에 있다. 이 시스템은 또한 모듈화되어 있고, 유연성이 높으며, 하드웨어 독립적인 장점을 지니는 반면, 컴파일하는 시간이 길어서 편집-테스트의 주기가 길어진다 단점도 있다. 이용 가능한 센서로는 비전, 근접 리미트 스위치, 접촉 센서, 힘/토크 센서 등이며, 비전 센서와 같이 계산 속도가 긴 경우는 'wait'를 이용하여 프로그램 실행과 로봇의 운동을 동기시켜 주며, 빠른 처리 속도를 갖는 센서 정보는 직접 되읽시켜 이용한다.

(6) RAIL : 1982-현재 : Automatrix

RAIL[25]은 상위 레벨의 데이터 구조와 프로그램 제어 구조를 갖춘 PASCAL의 많은 부분을 포함하는 언어로서, 비전과 로봇의 제어에 위한 언어이다. 이는 Automatrix의 주요 생산 공정에 사용되는 검색과 아크-음절에 이용된다. 이 언어는 VAL 보다는 훨씬 복잡한 구조를 가지며, 다중 프로세싱이나 AL에서와 같은 'affixment'는 제공하지 않는다. 로봇 언어의 분류에 따르면 구조 프로그래밍 레벨에 속하고, 이용 가능한 센서로는 비전, 근접 리미트 스위치가 있으나, 힘/로오브 센서는 이용할 수 없다. 이 센서를 이용하면 유도 운동(guarded motion)이 가능하며, 비전의 경우는 비전에 의한 서보잉까지는 실현하지 못하고, 촬영과 대상물 분석이 가능한 정도이다. 비전 명령어로 'PICTURE', 'IF THEN' 구문을 이용하여 'ANALYZE' 그리고 'AREA' 지정 등이 있으며, 'WAIT', 'UNTIL' 도 사용한다.

(7) LM : 1981-현재 : LJFIA Lab.

LM[1]은 조립용 로봇을 위한 자동 프로그래밍 시스템의 개발을 목표로 연구하던 중 목적 언어(target language)로서 개발된 미니플레이터-레벨의 언어이다. 이 언어는 프로그램 개발 및 디버깅 기능까지 제공하는 완전한 로봇 프로그래밍 시스템으로서 AL이 갖고 있는 대부분의 미니플레이터 기능을 포함하고, 그래픽 시뮬레이터의 이용이 가능하며, CAD 정보를 이용하여 로봇의 기하학적 프로그래밍까지 발전시킬 수 있으며, 오프-라인 운동 계획이 가능하다. 이 언어 역시 구조 프로그래밍 레벨에 속하며, 이용 가능한 센서로는 비전, 힘/로오브 센서, 근접 리미트 스위치 등이 있고, 이들의 정보를 이용하여 자유 운동, 유도 운동, 컴플라이언트 운동이 가능하다. 실제 응용에 있어서는 비전 센서를 이용하여 전자 부품의 삽입 공정에 이용된다. 유도 운동을 위해서는 'MOV'에 'UNTIL' 구문을 이용하여, 컴플라이언트 운동을 위해서는 여러가지 유도 운동과 'NOWAIT', 'IMMEDIATELY'를 조합하여 이용한다.

(8) RPL : 1981-현재 : SRI International

RPL[6]은 몇대의 로봇, 센서, 그리고 보조 기구들로 구성된 자동 매뉴팩처어링 시스템을 위한 제어 알고리즘의 개발과 시험, 그리고 디버깅에 이용되는 로봇 언어이다. 이 언어는 서브루틴 콜로서 설계되었고, 미숙련자도 쉽게 사용할 수 있으며, 프로그램 수정이 쉽고, 새로운 시스템에의 적응이 쉬운 장점을 지닌다. 이 언어는 원시 운동(primitive motion) 레벨에 속하고, 이용 가능한 비전과 접촉 센서가 있으며, 이를 이용하여 매우 단순한 이산 접촉 센싱과 매우 복잡한 비전 명령을 통하여 대상물의 회전과 위치 벡터를 파악해 낸다. 그러나 명령 실행이 불가능한 단점도 가지고 있다. 외부와의 I/O를 위해 'DTLRD', 'DTLWR', 'SETOX', 'CLRO', 'RDDXV', 'RELAYC', 'RELAYC', 접촉 센서를 위한 'DTLRD', 'DTLWR', 'DTLINI', 그리고 비전 정보를 위해 'NIVIS', 'PICTURE', 'GETFA', 'BLINK', 'DELB', 'LJ', 'RECOGN' 등의 명령어가 있다.

4. 센서와의 관계를 중심으로 본 로봇 언어의 특성 비교

로봇 언어에 있어서 센서 정보를 어느 정도까지 이용 가능하느냐 하는 문제는 로봇 프로그래밍의 문제점[1] 중 불확실성을 어떻게 비제할 수 있느냐 하는 문제에 직접적으로 연관되기 때문에 매우 중요한 논점이라고 사려된다. 로봇 언어에서 센서 정보를 이용하기 위해서는 대부분의 상위 레벨 컴퓨터 언어에서 이용 가능한 I/O 메카니즘과 힘 제어와 같은 다양한 제어 운동 메카니즘이

제공되어야 하는데, 후자의 경우에는 센서 정보에 의한 운동과 센서 정보에 따른 조건 분기(또는 의사 결정 기능)을 필요로 한다[12].

센서는 로봇 프로그래밍에서 주로 내가치의 목적을 위해서 사용된다;

(1) 운동의 시작과 종결

가장 많이 사용되는 기능으로 대부분의 로봇 언어가 제공하는 기능이다. 이 기능의 응용 시로는 피더로부터 물건을 받아 들일 때 종 센서에 의해 로봇 손가락 운동의 시작을 알리는 것과 리미트 스위치나 힘/로오브 센서를 이용한 유도 운동에 이용된다. 이 유도 운동이 가능한 언어로는 AL, AML, VAL-II, RCCL, LM 등이 있다[1,3,11,19,21,22].

(2) 의사 결정에 의한 분기 기능

이 기능은 대상물을 파악하기 위한 비전 시스템에서 어느 정도 가능하며, 비전을 이용한 서보잉이나 컨베이어 주력이 가능한 AL, MCL, RAIL, RPL VAL-II, RCCL, LM 등이 가능하다[1,3,13,20,21,22]. 그러나 그 기능은 매우 초보적인 단계이다. 이런 기능을 위해서는 상위 레벨의 컴퓨터 프로그래밍 언어의 제어 구조를 갖추고 있어야 하고, 비전을 포함해서 힘/로오브 센서, 위치 센서 등과 같은 센서의 정보를 모니터링하는 기능이 필요하다.

(3) 대상물의 위치, 방향, 복성 획득

이 기능은 비전 센서를 이용하여 수행되는데, 보통 대상물의 중심 위치를 계산해 내고, 그것의 회전 벡터를 알아 내어 로봇 손 끝으로 그 대상물을 잡을 위치에 대한 정보를 알아 내기도 하고, 데이터 베이스에 이미 저장해 놓은 정보와 실제 비전 센서에 의해 측정된 데이터와의 비교를 통한 검색 기능에도 이용된다. 전자에 주로 응용되는 언어로는 AL, RAIL, RPL, VAL-II, RCCL 등이 있고, 후자의 경우에는 MCL이 있다[3,6,11,19,20,21,22].

(4) 외부 제약 조건에 따른 응력

이 기능은 힘, 근접 또는 비전 센서로부터의 정보에 의하여 대상물과의 관계를 원하는 대로 유지하기 위해서 로봇의 운동을 변형시켜 주는데, 그 예로는 그랭크 속을 돌리는 일이나, 삽입 공정을 들 수 있다. 이 기능이 가능한 언어에는 AL, RCCL 등이 있다[3,11,21,22].

한편 로봇 언어의 레벨이라는 관점에서 살펴 보면, 상위 레벨로 갈수록 센서와의 관계는 더욱 밀접해 지고, 다양한 기능을 갖게 됨을 알 수 있다. 3장에서 서술한 8 가지 언어 중 RPL을 제외하고는 구조 프로그래밍 레벨 또는 그 이상(AL이나 LM의 경우는 작업 레벨의 속성도 일부 포함)에 속함을 볼 때 센서 정보를 이용하여 보다 지능적인 로봇의 제어를 위해서는 상위 레벨의 컴퓨터 프로그래밍 언어의 기능 및 구조를 포함해야 하고, 병렬 실행 기능과 외부와의 다양한 통신 기능 등이 부가되어야 할 것을 볼 수 있다.

그러나, 로봇 언어는 구조 프로그래밍이 가능해짐에 따라 복잡해지고, 사용자의 자각이 제한되며, 그 응용이 어려운 단점을 포함하게 된다. 과거 이러한 점으로 인해, 조급은 단순한 공정에 적용되었던 로봇 언어는 원시 운동 레벨의 언어로도 충분하였지만, 오늘날 보다 복잡하고, 다양한 작업 공정에 응용하기 위해서는 센서와의 인터페이스 및 병렬 실행이 불가피하게 되었고, 초보 단계의 사용자도 어렵고, 많은 경험을 필요로 하는 작업을 쉽게 프로그래밍할 수 있게 하기 위해서 사용자 서브루틴 기능도 크게 확대되고 있다.

5. 앞으로의 연구 방향

로봇 프로그래밍 언어는 앞서 서론에서 언급한 사용자의 편의와 충분한 표현력이라는 상반된

두 개념을 만족시키면서 인간의 지능에 접근하는 레벨로 상승되어 가고 있다. 즉, 앞서 살펴 본 8가지 로봇 언어와 그 역사적 변천을 통해서 로봇 언어는 점진적으로 작업 레벨을 향해 진보하고 있고, 부분적으로는 실현되고 있음을 보았다. 앞으로는 로봇의 종류와는 무관하고, 외부 환경과의 인터페이스가 자유로우며, 사용자의 자기에 제한이 없는 범용 로봇 언어의 개발을 추구하는 방향으로 연구가 진행되어야 할 것이며, 이를 위해서는 그 무엇보다도 센서 정보의 이용이 필수적이며, 이에 대한 수많은 연구와 부자가 필요 하리라 생각된다.

다음은 앞으로 로봇 언어가 갖추어야 할 기능 중 센서와 연관되어 필요한 기능으로 앞으로의 연구 대상이라 할 수 있겠다;

- 센서 정보 처리에 충분한 프로그래밍 언어의 구조
- 센서 정보의 모니터링을 위한 I/O 기능
- 센서 정보 처리를 위한 외부 컴퓨터와의 회로망 기능
- 병렬 실행 기능
- 센서 정보를 이용한 판단 기능
- 다중 센서 정보를 이용한 실시간 제어 기능
- 위험 상황이나 예외 발생에 따른 보상 기능
- 센서 정보 간의 연관성

REFERENCES

[1] Latombe J.C. et al, "The IM Robot Programming System", Robotic Research; The Second International Symposium, 1985.

[2] R.P. Paul, "WAVE: A Model - Based Language for Manipulator Control", Technical Paper MR 76-615, SMC, 1976.

[3] R. Finkel et al, "An Overview of AI, A Programming System for Automation", Proceedings of the 4th Inter. Joint Conf. on AI, 1975.

[4] Kunikatsu T. et al, "A Structured Approach to Robot Programming and Teaching", IEEE, Trans. on SMC, Vol. SMC-11, No. 4, 1981.

[5] L.I. Lieberman et al, "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly", IBM J. of Research & Development, Vol. 21, No. 4, July 1977.

[6] W.T. Park, "The SRI Robot Programming System", Proceedings of the 13th ISIR, 1983.

[7] D. Moon, MACLISP reference manual, ver. 0 Cambridge, Mass.: M.I.T. Lab. for Computer Science, 1974.

[8] R.T. Taylor et al, "AML: A Manufacturing Language", The Inter. Journal of Robotics Research, Vol. 1, No. 3, Fall 1982.

[9] S. Bonner et al, "A Comparative Study of Robot Languages", IEEE, Computer, 1982.

[10] B.I. Soroka, "What Can't Robot Language Do?", Proceedings of the 12th ISIR, 1982.

[11] W.A. Gruver et al, "Industrial Robot Programming Languages: A Comparative Evaluation", IEEE, Trans. on SMC, Vol. 14, No. 4, July/Aug., 1984.

[12] T. Lozano-Perez, "Robot Programming", IEEE, Proceedings, Vol. 71, No. 7, July, 1983.

[13] B.O. Wood et al, "MCL, The Manufacturing Control Language", Proceedings of the 13th ISIR, 1983.

[14] David D. Grossman et al, "Interactive Generation of Object Models with a Manipulator", IEEE, Trans. on SMC, Vol. 8, No. 9 Sept., 1978.

[15] 奥田実夫, "＜解説＞ロボットの制御システムと 言語", オートメーション誌, Apr., 1982.

[16] Unimation Inc., "User's Guide to VAL, Ver. 12, June, 1980.

[17] B. Shimano, "VAL: A Versatile Robot Programming and Control", Proceedings of COMPAC 79, 1979.

[18] C.S.G. Lee et al, "Hierarchical Control Structure Using Special Purpose Processor For the Control of Robot Arms", Proc. of Pattern Recog. and Image Proc. Conf., 1982.

[19] B.E. Shimano et al, "VAL-II: A Robot Programming Languages and Control System", Robotics Research Vol. 2, pp. 917-940, MIT Press, 1984.

[20] V. Hayward, "Introduction to RCCL: A Robot Control 'C' Library", Technical Report, TR-EE 83-43, Purdue Univ., Oct., 1983.

[21] V. Hayward, "Robot Real Time Control User's Manual, Technical Report, TR-EE 83-42, Purdue Univ., Oct., 1983.

[22] J.C. Latombe and E. Mazer, "IM: A high-level language for controlling assembly robots", Proc. of 11th ISIR, Oct., 1981.

[23] S.M. Mujtaba, "Current Status of the AL Manipulator Programming System", Proc. 10th ISIR, Milan, 1980.

[24] S.M. Mujtaba and R. Goldman, "The AL User's Manual, STAN-CS-79-718, Stanford Univ., 1979.

[25] J.W. Franklin et al, "Programming Vision and Robotics Systems With RAIL", Technical Paper MS82-216, SME, 1982.