

지능로봇을 위한 행위기반의 하이브리드 제어구조에 관한 연구

김광일*, 최경현⁺, 이석희⁺⁺

(논문접수일 2005. 7. 15, 심사완료일 2005. 8. 24)

A Study on Behavior-based Hybrid Control Architecture for Intelligent Robot

Kwang-Il Kim*, Kyung-Hyun Choi⁺, Seok-Hee Lee⁺⁺

Abstract

To accomplish various and complex tasks by intelligent robots, improvement is needed not only in mechanical system architecture but also in control system architecture. Hybrid control architecture has been suggested as a mutually complementing architecture of the weak points of a deliberative and a reactive control. This paper addresses a control architecture of robots, and a behavior representation methodology. The suggested control architecture consists of three layers of deliberative, sequencing, and reactive as hybrid control architecture. Multi-layer behavior model is employed to represent desired tasks. 3D simulation will be conducted to verify the applicability of suggested control architecture and behavior representation method.

Key Words : Intelligent Robot(지능 로봇), Hybrid Control Architecture(하이브리드 제어구조), Deliberative Control(심의 제어), Behavior-based Control(행위 기반 제어), Task Representation(작업 표현)

1. 서론

지능로봇 시스템이 원하는 기능을 수행하기 위해서는 많은 하드웨어와 소프트웨어 모듈을 필요로 하는데, 이들을 통일성 있게 구현 할 수 있는 것이 제어구조이다. 단순한 작업(task)만을 수행하던 이전의 로봇과는 달리, 복잡한 환경에서 다양한 행동이 요구되는 현대의 지능로봇에 있어서 주어

진 작업을 성공적으로 수행하기 위해서는 인간의 두뇌작용에 해당하는 지능제어시스템과 이를 바탕으로 한 자율기능, 높은 수준의 인간과 로봇과의 통신을 위한 적절한 인터페이스, 고기능 센서를 이용한 감각기능, 로봇의 활동범위를 넓혀주는 이동기능, 여러 가지 조작을 위한 조작기능, 그리고 임무수행을 위한 처리기능 등이 필요하다. 이러한 다양하고 복잡한 작업을 수행하기 위해서는 기계적인 향상 이외에 제

* 부산대 대학원 지능기계공학과

+ 교신저자, 제주대학교 기계에너지시스템공학부 (khchoi@cheju.ac.kr)

주소: 690-756 제주도 제주시 제주대학교 66

++ 부산대학교 기계공학과

어구조의 개선이 필요하다.

종래의 로봇 제어방식에는 크게 심의 제어(deliberative control)⁽¹⁾와 반응 제어(reactive control)⁽²⁾로 나눌 수 있다. 심의 제어는 지식베이스에 근거를 두는 추론기능을 가지고 궤적을 계획하고 수행하는 전통적인 계층적 제어구조를 가지며 복잡한 작업에서도 전반적인 최적화를 이룰 수 있지만 느린 반응속도와 외부환경모델에 의존하는 약점을 가진다.

반응 제어는 센서에 의한 구동기의 반응을 유한하게 정의하여 로봇의 동작에서 일어나는 행위(behavior)를 실시간으로 처리하지만, 전체적인 입장의 임무수행 측면에서 약점을 가지고 있다. 이러한 약점을 상호 보완하는 구조로서 심의/반응 하이브리드 제어구조^(3~6)가 필요하다. 이러한 하이브리드 제어구조는 주변 환경의 체계적인 표현으로 고차원적인 작업을 수행할 수 있으며, 오류 진단 및 회복이 빠르다. 또한, 환경의 불확실성과 비예측성에 대해서도 효과적으로 대처할 수 있다.

지금까지 연구된 하이브리드 제어구조는 조작을 제외한 구조, 군사용으로 설계 된 구조, 그리고 내부 인터페이스를 공개하지 않은 경우가 대부분이었다. 따라서 본 논문에서는 모바일/머니플레이터 형태의 지능로봇의 복잡한 작업을 효율적으로 수행하기 위한 작업 계획, 수행과 조작의 효과적인 지원, 그리고 기능별로 관련 모듈을 컴포넌트화 하여 재사용성이 가능한 심의, 순서계획(sequencing), 그리고 반응의 3개의 층으로 구성된 행위기반의 하이브리드 제어구조를 제안한다. 또한 작업 표현방법으로 제어로직을 시각화 할 수 있고, 오류 처리가 가능한 다층 행위 모델(multi-layer behavior model)을 제안한다. 마지막으로, 물론 나르기 심부름에 대한 3D 시뮬레이션을 통해 제안된 제어구조와 작업 표현 방법의 유용성을 검증한다.

2. 행위 기반의 하이브리드 제어구조

2.1 행위 기반의 하이브리드 제어

행위라는 개념은 심의 제어 방식인 감각-계획-행동(SPA, Sense-Plan-Act)의 작업 환경에 대해 많은 지식을 가지고 있어야 하는 점과 환경의 불확실성과 비예측성에 효과적으로 대처 할 수 없는 단점의 대안으로 Brooks의 포섭(subsumption) 구조에서 도출되었다⁽²⁾. 이 제어 방법은 구성 요소를 기능별로 분해한 기존의 방법과 달리 능력별로 나누었고, 이 개념이 확대되어 행위기반 제어가 탄생하였다. 그러나 행위기반 제어는 지능형 에이전트의 자율적인 환경 대응면에서는 신뢰성이 있으나 주어진 작업 수행 측면에서는 취약한 면이

있다. 이는 행위기반 제어에서 신중한 작업 계획 및 관리를 배제하고 있기 때문이다.

일반적으로 하이브리드 제어구조의 상위에는 사람-로봇-대화(HRI, Human-Robot-Interface) 관련 부분과 작업을 분해 및 계획을 담당하는 심의 층이 존재하고, 하위구조에는 실시간성이 필요한 로봇 제어 및 센싱에 관련된 반응 층이 존재한다. 그리고 이 둘 사이의 연결을 담당하는 층이 존재하기도 한다. 따라서 하이브리드 제어구조는 최소 2개의 층이 필요하며, 대체로 3개의 층으로 구성된다.

Fig. 1은 제안된 하이브리드 제어구조를 보이며 각종 컴포넌트들의 배치와 컴포넌트 사이의 연결 관계 및 흐름을 보여준다. 제어구조는 크게 심의, 순서계획, 그리고 반응의 3개의 층과 지식 데이터베이스(Knowledge Database)로 구성된다.

최상위 계층인 심의 층은 사용자와의 대화 및 작업 계획을 담당하고, HRI와 작업 계획기(Task Planner)로 이루어진다. 중간 계층인 순서계획 층은 심의 층과 반응 층 사이의 원활한 동작을 위한 인터페이스를 담당하며 2개의 부분으로 나뉜다. 첫 번째는 반응 층 제어부로 프로세스 감독기(Process Supervisor)와 프로세스 저장소(Process Storage)로 구성된다. 두 번째는 정보 생성부로 주행 및 조작 모듈들(Navigation/Manipulation Modules)과 음성 및 대상 인식 모듈들(Speech/Object Recognition Modules)로 구성된다. 최하 계층인 반응 층은 짧은 계산을 반복적으로 수행하며 환경에 민감하게 반응하고, 자원들(Resources), 행위 실행기(Behavior Executor), 그리고 제어기들(Controllers)로 구성된다.

지식 데이터베이스는 사용자 지식(User Knowledge), 작업 지식(Task Knowledge), 그리고 월드 모델(World Model)로 이루어진다. 사용자 지식은 사용자에 대한 정보, 작업 지

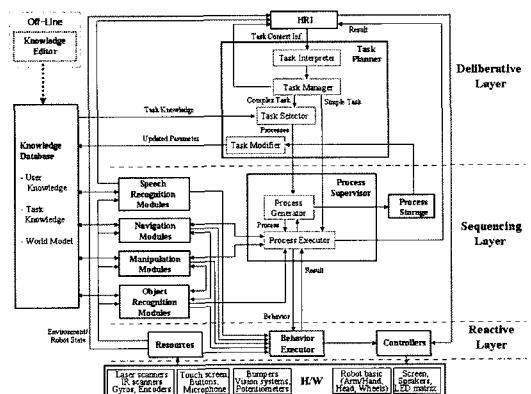


Fig. 1 Proposed hybrid control architecture

식은 요구되는 작업에 대해 다층 행위 모델로 미리 정의된 행위의 집합으로 구성된 프로세스들이 저장되어 있으며, 월드 모델은 여러 가지 지도, 이미지 등의 환경 정보가 저장되어 있다. 지식 데이터베이스는 지식 편집기(Knowledge Editor)를 통해 오프라인 상에서 정보를 수정 및 추가하여 업데이트 할 수 있다.

2.2 심의 층

심의 층은 사용자와의 인터페이스 및 작업을 계획하는 역할을 한다. HRI는 사용자와의 대화 및 터치 스크린 등을 통해 명령을 받는다. 작업 계획기는 사용자로부터 작업 명령을 받으면 관련 프로세스들 또는 행위를 순서계획 층에 보낸다. 내부 모듈은 작업 해석기(Task Interpreter), 작업 관리기(Task Manager), 작업 선택기(Task Selector), 그리고 작업 수정기(Task Modifier)로 구성된다.

작업 해석기는 HRI에서 내리온 작업명령 문맥(context)을 해석한다. 작업 관리기는 해석된 문맥 정보를 단일 작업(simple task)과 복합 작업(complex task)으로 분류하여, 단일 작업은 순서계획 층의 프로세스 실행기(Process Executor)에 보내고, 복합 작업은 작업 선택기로 보낸다. 여기서 단일 작업은 사용자 호출과 같은 프로세스를 필요로 하지 않는 작업을 말하고, 복합 작업은 물건 나르기 심부름과 같이 프로세스를 필요로 하는 작업을 말한다.

작업 선택기는 복합 작업에 대한 관련 프로세스들을 작업 지식 데이터베이스에서 선택하여 순서계획 층의 프로세스 생성기에 보낸다. 프로세스 수정기는 프로세스 수행 종료 후 프로세스 저장소에 저장되어 있는 재사용 가능한 경로 및 궤적을 작업 지식 데이터베이스에 업데이트 한다.

Fig. 2는 단일 및 복합 작업에 대한 작업 계획의 플로우-차트를 보인다.

2.3 순서계획 층

순서계획 층은 반응 층을 제어하고 프로세스 수행을 직접적으로 책임지며, 행위 수행에 관련되거나 사용자 음성 인식과 대상 인식에 관련된 알고리즘을 수행한다.

프로세스 감독기는 프로세스를 생성 및 변경하고 관련 모듈들에게 행위를 보내며, 프로세스 실행 결과를 HRI에 보고한다. 내부 모듈로 프로세스 생성기(Process Generator)와 프로세스 실행기(Process Executor)를 가진다.

프로세스 생성기는 작업 선택기에서 받은 프로세스들을 프로세스 저장소에 저장하고, 수행 프로세스를 생성하여 프

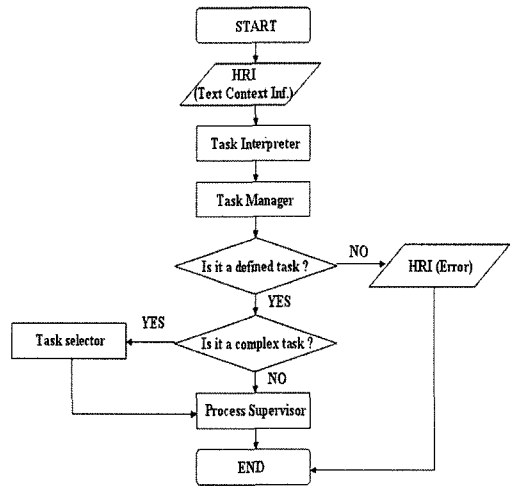


Fig. 2 Flow-chart of task planning for simple or complex task

로세스 실행기에 보낸다. 만약 프로세스 수행 중 프로세스 변경 요청이 들어오면, 프로세스 저장소에 저장되어 있는 프로세스를 기반으로 프로세스를 변경하고, 변경된 프로세스를 프로세스 실행기에 보낸다.

프로세스 실행기는 생성된 수행 프로세스 내부의 행위 또는 작업 관리기로부터 바로 내리온 행위를 수행 및 조작 모듈들에게 할당하고 생성된 경로 및 궤적을 행위 실행기에 보낸다. 그리고 프로세스 실행 결과를 HRI에 보고하며, 반응 층과 순서계획 층의 정보 생성부에서 보고된 이벤트와 경계 조건에 따른 프로세스 변경을 프로세스 생성기에 요청한다.

프로세스 저장소는 프로세스 수행 동안의 임시 저장소로서 작업 지식 데이터베이스로부터 선택된 프로세스들과 프로세스 수행 중에 생성되거나 변경된 경로 및 궤적을 저장한다. 프로세스 종료 후 저장된 데이터는 작업 수정기로부터 작업 지식 데이터베이스를 업데이트 하고, 저장된 데이터는 삭제된다.

주행 모듈들은 지도 작성을 통하여 월드 모델을 업데이트하거나 프로세스 실행기로부터 관련 행위가 내려오면 현재의 로봇 상태와 월드 모델을 참조하여 위치추정(localization) 및 경로계획(path planning)을 수행하고 생성된 경로는 프로세스 실행기 또는 반응 층의 행위 실행기에 보낸다. 그리고 방 청소와 같은 행위를 모듈로 저장하고 있다.

조작 모듈들은 프로세스 실행기로부터 관련 행위가 내려오면 현재의 로봇 상태와 월드 모델을 참조하여 지점간(point-

to-point) 또는 연속 궤적(continuous trajectory)과 목표 형상에 따른 파지 방법을 생성한다. 생성된 궤적 및 파지 방법은 프로세스 실행기 또는 반응 층의 행위 실행기에 보낸다. 음성 인식 모듈들은 자원들로부터 받은 사용자의 음성 데이터를 지식 데이터베이스를 참조하여 처리하고, 인식결과를 HRI에 보낸다.

대상 인식 모듈들은 자원들로부터 받은 대상 및 사용자 얼굴 인식 데이터를 지식 데이터베이스를 참조하여 처리하고, 인식 결과를 주행 또는 조작 모듈들에게 보내거나 생성 및 변경된 데이터를 지식 데이터베이스에 업데이트한다.

2.4 반응 층

반응 층은 매우 짧은 시간동안 반복적으로 수행하고 실제 로봇의 움직임을 제어하며, 센서와 액츄에이터와 같은 하드웨어 관련 모듈과 실시간 제어 명령을 생성하는 행위들로 이루어진다. 센서의 입력 데이터를 저장하고 있는 자원들로부터 정보를 받은 행위 실행기는 단 시간 내에 제어 명령을 생성하고 상위 명령과 취합된다. 취합된 제어 명령은 제어기들을 통해 하드웨어에 전달되고 실제 로봇을 구동시킨다. 그리고 장애물 회피(obstacle avoidance)⁽⁷⁾와 같은 행위 모듈을 저장하고 있다. Fig. 3은 행위 실행기의 행위 우선순위를 보인다.

3. 작업 표현 방법

작업은 Fig. 4와 같이 계층적 구조를 가지는 다층 행위 모델로 표현한다. 작업은 행위의 집합으로 표현되고 행위와 행위 사이의 전이(transition)는 이벤트와 경계조건으로 정의된다. 작업은 프로세스의 조합이고 프로세스는 다층 행위 모델로 표현한 행위들의 일련의 순서이며 오류 처리 로직을 포함한다.

Fig. 5, 6, 7, 그리고 8은 기본 작업인 물건 나르기 심부름

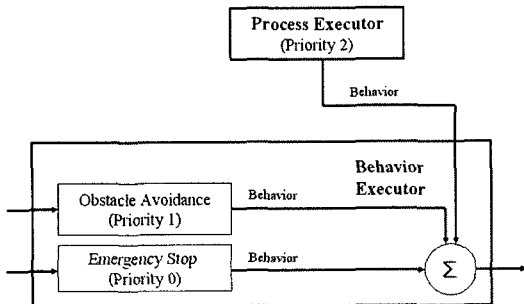


Fig. 3 Priority-based behavior executor

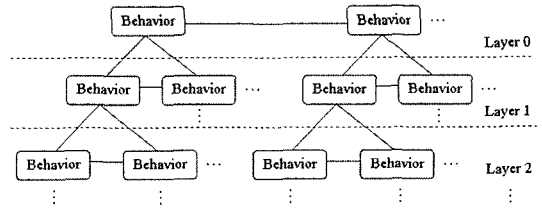


Fig. 4 Multi-layer behavior model

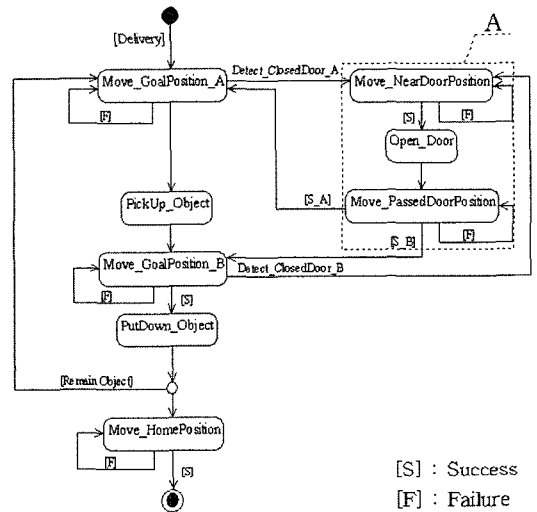


Fig. 5 State-chart diagram of delivery task

에 대해 다층 행위 모델을 기반으로 UML(Unified Modeling Language)의 상태-차트 다이어그램을 이용하여 표현한 것이며, 각 상태는 행위를 나타낸다. Fig. 5는 물건 나르기 심부름의 최상위 프로세스를 나타내고 있으며, A 부분은 로봇이 다른 방으로 이동 중에 닫힌 문을 발견하면, 프로세스를 변경하여 문을 열고 목적지로 이동하는 것을 보인다.

Fig. 6, 7, 그리고 8은 각각 Open_Door, PickUp_Object, 그리고 PutDown_Object 명령의 하위 프로세스를 보인다.

Fig. 6의 B 부분은 로봇이 누름 버튼(push button)으로 되어 있는 방문을 열기 위해 문 앞으로 이동, 버튼 위치로 팔을 뻗음, 그리고 한 손가락을 펴는 주행과 조작의 동기화된 연동을 나타낸다. 이런 연동 표현이 가능한 것은 로봇이 고정된 위치인 방문과 손잡이의 정보를 미리 인지하고 있기 때문이다. 이렇게 생성된 주행 및 조작 데이터는 프로세스 종료 후 지식 데이터베이스에 저장되고 재사용된다. 만약 문 또는 손잡이의 위치가 변경 되었을 경우에는 변경된 주행

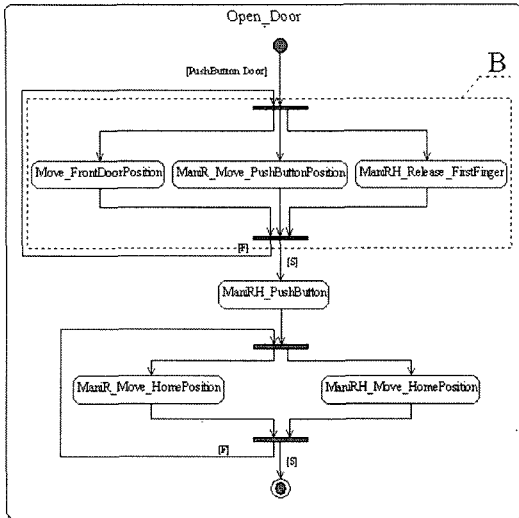


Fig. 6 State-chart diagram of sub-layer process for Open_Door command

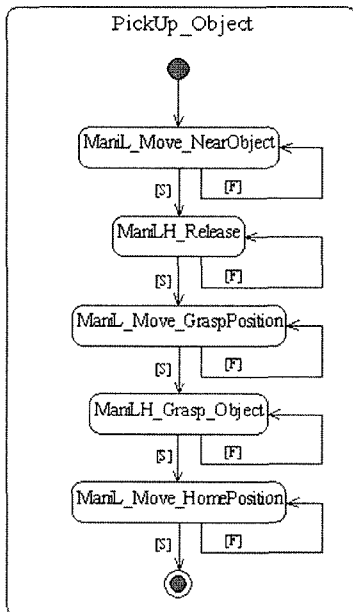


Fig. 7 State-chart diagram of sub-layer process for PickUp_Object command

및 조작 데이터를 업데이트 한다. 이 밖에 냉장고 문열기, 엘리베이터 버튼 누르기 등에 적용이 가능하고, 이것으로 로봇이 좀 더 자연스럽게 빠르게 작업을 수행 할 수 있다.

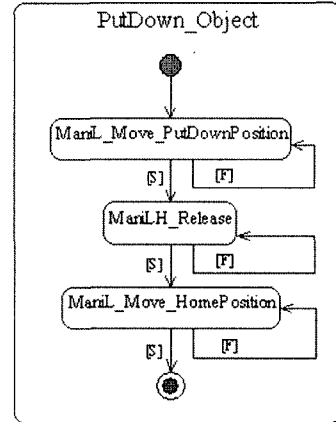


Fig. 8 State-chart diagram of sub-layer for PutDown_Object command

4. 로봇 시뮬레이션

로봇 시뮬레이션은 제안된 제어구조와 작업 표현 방법을 기반으로 주어진 작업을 성공적으로 수행함을 확인하며, 실제 시스템에서 발생할 수 있는 오차는 고려되지 않았다.

3D 시뮬레이션을 위한 로봇 사양 및 환경은 Table 1과 같다. 로봇은 2개의 구동 바퀴와 2개의 보조바퀴, 양 팔, 그리고 3개의 손가락을 가지는 양손으로 구성되어 있다. 환경은 2개의 방, 누름 버튼으로 되어있는 1개의 문, 장애물 역할을 하는 2대의 로봇, 사람, 음료수 캔, 테이블들로 구성되어

Table 1 Robot specification and environment for 3D simulation

Robot	Mobile	<ul style="list-style-type: none"> • 2 active cater type • $\Phi 680 \times 1174\text{mm}$ • 3 DOF
	2 Manipulators	<ul style="list-style-type: none"> • Each 7 DOF • Max. Reach : 1m
	2 Hands	<ul style="list-style-type: none"> • Each 3 Fingers • Each 6 DOF
Environment	<ul style="list-style-type: none"> • 2 rooms ($16400 \times 8400\text{mm}$) • Automatic door ($80 \times 1000 \times 2000\text{mm}$) • Push Button ($20 \times 100 \times 300\text{mm}$) • Obstacles ($\Phi 680 \times 1174\text{mm}$) • Tables ($500 \times 500 \times 500\text{mm}$) • Can ($\Phi 60 \times 120\text{mm}$) • Human ($620 \times 250 \times 1500\text{mm}$) 	

있다. Fig. 9는 작업을 수행하기 위한 환경 모델링을 보인다.

로봇 및 환경에 대한 초기 상태는 다음과 같다.

1. 사용자와 로봇은 한 방에 같이 있다.
2. 로봇은 문이 항상 열려 있는 상태라고 인지한다.
3. 로봇은 환경 지도를 가지고 있다.
4. 로봇은 Table_A와 Table_B의 위치를 알고 있다.
5. 로봇은 사용자 및 음료수 캔의 정보를 가지고 있다.

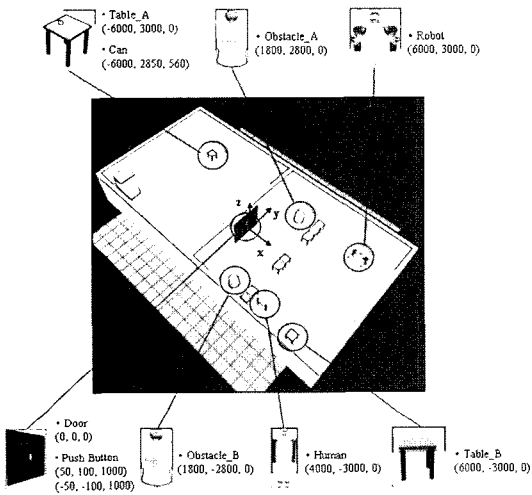


Fig. 9 Environment modeling

로봇 시뮬레이션은 Table_A 위에 있는 음료수 캔을 Table_B 위로 옮기는 물건 나르기 심부름에 대해 수행되었으며, Fig. 10과 Fig. 11은 각각 물건 나르기 심부름의 플로우-차트와 3D 시뮬레이션을 보인다.

Table 2는 문을 열기 위한 주행과 조작의 동기화된 연동과 비연동의 시뮬레이션 조건과 결과를 보이며, Fig. 12와 Fig. 13은 각각 연동과 비연동의 3D 시뮬레이션을 보인다. 각각의 시뮬레이션 수행 시간을 보면 연동된 경우가 약

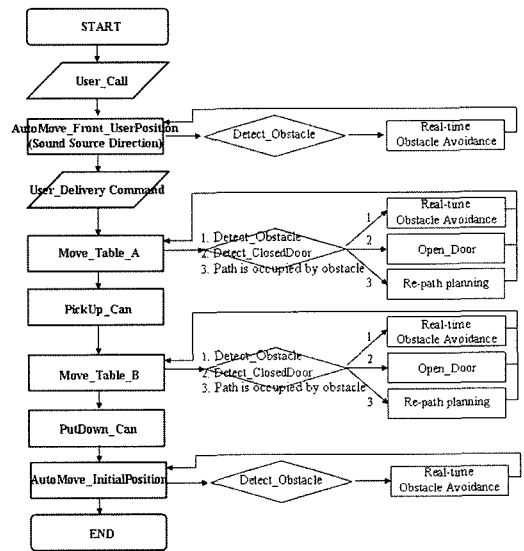


Fig. 10 Flow-chart for delivery task

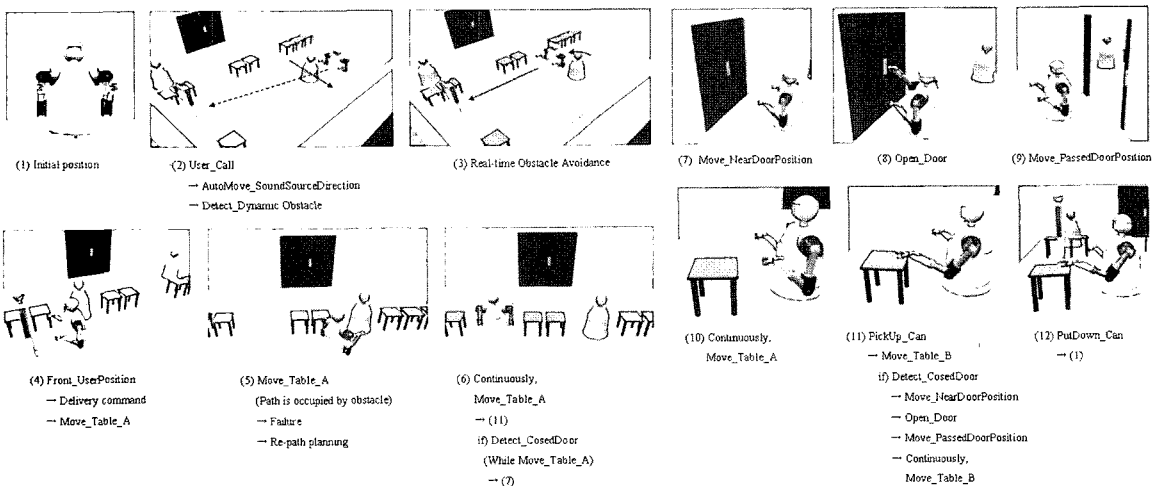


Fig. 11 3D simulation for delivery task

Table 2 3D Simulation condition and result of Open_Door command

	Command	Distance(from door)	Velocity	Total Performance Time
Synchronous	Move_NearDoorPosition	3000~1800mm	200mm/sec	15.7sec
	Move_FrontDoorPosition	1800~1100mm	100mm/sec	
	Mani_Move_PushButtonPosition		100deg/sec	
	ManiRH_Release_FirstFinger		90deg/sec	
	ManiRH_Push_Button	1100mm	45deg/sec	
Asynchronous	Move_FrontDoorPosition	3000~1100mm	200→100mm/sec	17.6sec
	ManiR_Move_PushButtonPosition	1100mm	100deg/sec	
	ManiRH_Release_FirstFinger		90deg/sec	
	ManiRH_Push_Button		45deg/sec	

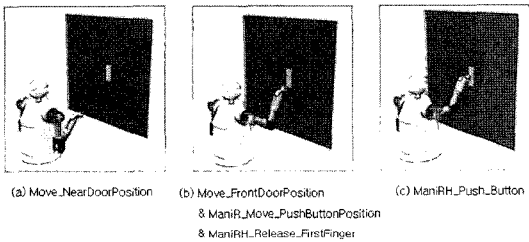


Fig. 12 Synchronized behaviors of navigation and manipulation

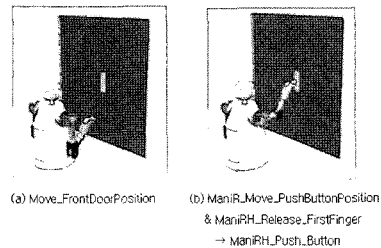


Fig. 13 Asynchronized behaviors of navigation and manipulation

11%정도의 시간이 단축됨을 확인 하였다. 만약 연동의 시작 위치와 속도를 최적화한다면 더 많은 시간을 단축할 수 있을 것이다.

5. 결론

본 논문에서는 지능로봇을 위한 행위기반의 하이브리드 제어구조와 작업 표현 방법인 다층 행위 모델을 제안하였다. 그리고 이를 기반으로 3D 시뮬레이션을 통해 주어진 작업을 성공적으로 수행함을 확인하였다. 따라서 다음과 같은 결론을 내릴 수 있다.

- (1) 행위기반의 하이브리드 제어구조는 효율적인 작업 계획과 주행과 조작의 효과적인 지원으로 다양하고 복잡한 작업을 수행할 수 있다. 또한 관련 모듈을 기능별로 컴포넌트화 하여 로봇 시스템 통합을 위한 가이드라인을 제시하였다.
- (2) 다층 행위 모델은 제어로직을 시각화하고 표현이 간단하며, 각각의 행위를 클래스화 하여 재사용성이 용이하다.

다. 또한 다층 행위 모델로 정의된 작업은 작업 지식 데이터베이스에 저장되어 사용자의 작업 명령에 따라 빠르게 수행할 수 있으며, 이벤트에 따른 프로세스 변경이 용이하다.

- (3) 주행과 조작의 동기화된 연동 표현으로 작업 실행 시간이 단축됨을 확인하였으며, 연동의 시작위치와 속도를 최적화한다면 더 많은 시간 단축이 예상된다.

후기

본 논문은 과학기술부의 21세기 프론티어 사업인 ‘인간기능 생활지원 지능로봇 기술개발 사업’에 의해 지원되었습니다.

참고 문헌

- (1) Nilson, N. J., 1980, *Principles of Artificial Intelligence*, Palo Alto: Tioga.
- (2) Brooks, R. A., 1986, “A Robust Layered Control System for a Mobile Robot,” *IEEE Journal of Robotics*

- and Automation*, Vol. 2, No. 1, pp. 14~23.
- (3) Arkin, R. C., 1998, *Behavior-Based Robotics*, The MIT Press, Cambridge, pp. 205~235.
- (4) Lindstrom, M., Oreback, A., and Christensen, H. I., 2000, "BERRA : A Research Architecture for Service Robots," *Proc. of the IEEE Conference on Robotics and Automation*, San Francisco, USA, pp. 3278~3283.
- (5) Albus, J. S., April 2002, "4D/RCS A Reference Model Architecture for Intelligent Unmanned Ground Vehicles," *Proc. of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, USA, pp. 303~310.
- (6) Kim, G. H., Chung, W. J., Kim, M. S., and Lee, C. W., 2004, "Implementation of Multi-Functional Service Robots Using Tripodal Schematic Control Architecture," *Proc. of the IEEE International Conference on Robotics and Automation*, New Orleans, USA, pp. 4005~4010.
- (7) Ko, N. Y., and Choi, J. S., 1998, "A New Approach to Moving Obstacle Avoidance Problem of a Mobile Robot," *Trans. of KSMTE*, Vol. 7, No. 1, pp. 9~21.