

동적 재구성이 가능한 퍼스널 로봇 플랫폼

Dynamically Reconfigurable Personal Robot Platform

노세곤, 박기흥, 양광웅, 박진호, 오기용, 김홍석, 이호길, 최혁렬*

(Se-gon Roh, Kiheung Park, Kwangwoung Yang, Jinho Park, Ki Yong Oh, Hongseok Kim, Hogil Lee, and Hyoukryeol Choi)

Abstract : In this paper, the framework for accelerating the development of personal robots is presented, which includes the technology such as modularization with its own processing and standardization open to the other developers. Its basic elements are Module-D(*Module of DRP I*) characterized functionally and VM-D(*Virtual Machine of DRP I*) arbitrating Module-Ds. They can suggest the effective ways for integrating various robotic components and interfacing among them. Based on this framework, we developed a fully modularized personal robot called DRP I(*Dynamically Reconfigurable Personal robot*). Its hardware components are easily attached to and detached from the whole system. In addition, each software of the components is functionally distributed. For the materialization of the proposed idea, we mainly focus on the dynamically reconfigurable feature of DRP I.

Keywords : personal robot, standardization, reconfigurable robot, virtual machine, module

I. 서론

최근 들어 우리는 통신과 집적 회로 분야와 같은 기술이 급진적으로 발전됨에 따라 삶에 있어서 광범위한 변화를 경험하고 있다. 이러한 변화는 사회구조를 발전시키고 특히 가정에서의 개인의 삶의 영역을 확장시키게 한다. 예를 들어, 사람들은 집안의 유지[1], 보안, 교육, 청소[2] 등에 있어서, 디지털 가전기기, 홈네트워크(home network)[3], 이동 통신 기기 등을 이용하는 환경에 익숙해져 가고 있다. 홈오토메이션(home automation)의 기술이 이미 진부한 아이디어로 간주되고 있지만 이제는 최신의 기술 또는 앞선 분야로 다시 평가되면서 많은 관련 산업 분야에서 상품화가 진행되고 있다. 이렇듯 개인영역의 삶이 중요시됨에 따라, 많은 사람들은 가까운 미래의 로봇은 산업용이 아닌 개인적인 사용에 목표를 두어야 한다고 말하고 있으며[4], 이미 퍼스널 로봇에 관한 연구는 세계적인 추세로 나타나고 있다. 그러나 퍼스널 로봇 시장이 매우 매력적인 분야라 하더라도 장래성 있는 사업이 되기에는 아직 많은 문제를 내포하고 있다. 예를 들어, 퍼스널 로봇을 매우 많은 곳에서 개발해오고 있으나, 각각의 로봇개발자들은 그들 나름대로의 설계방식과 경험을 바탕으로 자신만의 로봇에 알맞도록 부품들을 적절하게 조합하여 제작하고 어플리케이션(application) 또한 그 로봇에 적합하도록 개발하고 있기 때문에 각각의 로봇들 사이에는 공유할만한 점이 그렇게 많지 않다. 따라서 로봇들은 독립된 시스템으로서 각각이 상당한 가치와 능력을 가졌다고 하더라도 서로 간에 확장 가능성과 호환성은 갖고 있

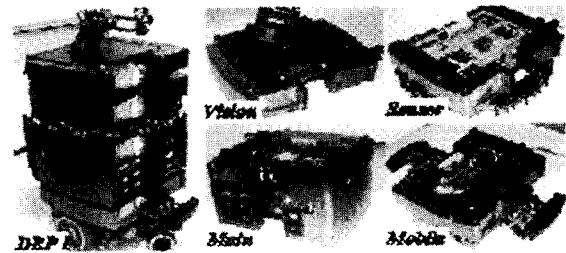


그림 1. DRP I과 Module-D.

Fig. 1. DRP I and Module-D.

지 않으며, 아직 사용자의 다양한 요구를 만족시킬 수 없고 상품으로서 개발의 신속함과 비용적인 경쟁력을 갖추고 있지 않다. 그러므로 여전히 개발, 상품화, 보수유지, 안전, 작업수행 등에 있어서의 비용 효과를 높이기 위해서는 현재의 제품들은 여전히 개선할 점이 많이 남아 있다. 이와 같은 문제점을 극복하기 위한 근본적인 해결책은 퍼스널 로봇에 관련된 기술들의 기본 구조를 표준화(standardization) 하고 모듈화(modularization) 하는 것이다.

이 논문에서, 이러한 개념을 바탕으로 한 퍼스널 로봇 플랫폼의 소프트웨어와 하드웨어 구조에 대하여 논하며, 이를 로봇 시스템으로 구체화 시킨 DRP I(*Dynamically Reconfigurable Personal robot ver. I*)을 소개한다. 그림 1에서 나타나는 바와 같이 Module-D(*Module of DRP I*)라 불리는 로봇의 각 하드웨어 모듈은 전체 시스템에 있어서 쉽게 조립과 분해가 가능하다. 각 모듈의 소프트웨어는 기능적으로 분산되어 있고 언제든지 재구성(reconfigure) 될 수 있다. 보다 자세히 말하자면, 이러한 소프트웨어 구조는 고수준(high level) 작업(task)과 저수준(low level) 작업간의 연결 해법을 제공한다. 이런 환경을 기반으로, 소프트웨어 개발은 몇 개의 단계를 가지게 되고, 그 결과 저수준 작업 개발자들은 고수준 작업 개발자들에게 각 부분의 디바이스 드라이버(device driver)와 같은 저수준 소프트웨어를 제공할 수 있다. 이러

* 책임저자(Corresponding Author)

논문접수 : 2004. 6. 6., 채택확정 : 2004. 7. 18.

노세곤, 박기흥, 박진호, 오기용, 최혁렬 : 상균관대학교

(robotian@me.skku.ac.kr/khpark@me.skku.ac.kr/ottaking@me.skku.ac.kr/kyoh@me.skku.ac.kr/hrchoi@me.skku.ac.kr)

양광웅, 김홍석, 이호길 : 한국생산기술연구원

(ykgwg@kitech.re.kr/hskim@kitech.re.kr/leehg@kitech.re.kr)

※ 본 연구는 산업자원부 퍼스널 로봇 기반기술 개발 과제 지원으로 수행하였음.

한 것이 가능한 것은 VM-D(Virtual Machine of DRP I)라 불리는 가상머신이 독립성, 호환성, 이식성 등을 지원하기 때문이다. 로봇의 이러한 기능은 중추신경시스템으로 비유될 수 있는 IEEE1394 네트워크를 통하여 동적으로 구성되어 있다.

II장에서, 본 연구자들이 개발한 기본구조의 개념과 아이디어를 소개한다. III장은 Module-D, VM-D와 RPL을 설명한다. IV장에서는 개발한 기본구조를 기반으로 한 DRP I의 실험과 구현에 대하여 동적(dynamically)으로 재구성이 가능한 기능을 중심으로 설명한다. V장에서는 현재와 미래의 연구에 대해 논하고 본 논문을 결론지을 것이다.

II. 문제정의 및 아이디어

이 장에서는 본 연구자들이 개발한 퍼스널 로봇의 구조의 개념 및 아이디어에 대하여 설명하고자 한다. 이를 위해 퍼스널 컴퓨터와 같은 시스템의 구조와 비교하고 있으며 퍼스널 로봇의 개발을 가속시키기 위한 개선된 구조에 관하여 소개할 것이다.

퍼스널 로봇은 최근 몇 년간 자주 소개되고 있는데 대부분의 퍼스널 로봇은 바퀴를 가지고 있다[5-9]. 비록 몇몇이 팔과 다리를 가지고 있는 구조이기는 하지만 자동차 같은 바퀴달린 이동체(vehicle)로 보이기 때문에 퍼스널 로봇을 개발한다는 자체는 단순하게 보일 수도 있다. 하지만, 네비게이션(navigation)과 지능(intelligence) 같은 높은 수준의 기술을 필요로 하기 때문에 이러한 로봇개발이 결코 쉽지만은 않다. 또 다른 관점에서 볼 때, 퍼스널 로봇 개발이 어려운 이유는 '퍼스널(personal)'이라는 단어가 의미하는 것처럼 많은 사용자들의 특정 지을 수 없는 다양한 요구를 만족시켜야 함에 있다.

퍼스널 컴퓨터와 비교해 볼 때 퍼스널 로봇의 특징은 양쪽 모두 전산적(computational) 기술을 요구할 뿐만 아니라 여러 부류의 사용자를 만족시켜야 한다는 점에서 유사하다. 그러나 퍼스널 로봇은 퍼스널 컴퓨터와 구별되는 점이 있다.

첫째, 퍼스널 컴퓨터는 이미 널리 퍼져 있다. 퍼스널 컴퓨터의 대중화는 호환성과 확장성에 바탕을 둔 표준화에서 비롯되었다. 이러한 표준화는 퍼스널 컴퓨터 개발에 전문적이고 높은 수준의 기술을 요구함에도 기업들이 신속하게 개발할 수 있게 해주었다. 이것은 현재까지 표준화가 이루어지지 않고 있는 퍼스널 로봇의 발전과 대중적 보급은 표준화에 달려있다는 것을 의미한다.

둘째, 퍼스널 컴퓨터는 메인보드의 CPU와 VGA 카드, 사운드 카드, 모니터 같은 기능적 부품들로 이루어진 중앙제어구조를 가지고 있고, 부품들 간 데이터 전송을 위한 비실시간(time-delayed) 통신을 허용하고 있다. 그렇지만, 퍼스널 로봇의 센서와 같은 부품들은 주변 환경과 접촉을 요하기 때문에 물리적으로 분리되어 있고, 로봇은 많은 양의 데이터를 전송받아 처리해야 하기 때문에, OPEN-R, OpenHRP, BERRA, PETER[10-14]과 같은 분산/다중처리 제어구조를 지향하고 있으며, 감지와 반응을 요하는 운동성 측면에서 볼 때 실시간성을 갖추어야 한다.

앞서, 퍼스널 컴퓨터와의 비교에서 언급한 것처럼, 많은

퍼스널 로봇들은 그 나름대로 이점을 가지고 있지만 표준화와는 여전히 거리가 있다. 예를 들어, Sony사의 AIBO는 유연하고 융통성 있는 플랫폼(platform)이다[5]. AIBO는 소프트웨어 구조로 OPEN-R과, 기계적 하드웨어로 모듈화 부품을 선택했다. AIBO의 ERS-210 모델의 다리와 머리는 ERS-220 과도 호환 가능하며, 이 로봇은 OPEN-R 기반의 프레임워크(framework)를 가지고 새로운 모델로 끊임없이 업데이트 될 수 있다. 그러나 OPEN-R은 애완용 강아지 로봇을 위해 설계되어 있으며 아직까지는 다른 퍼스널 로봇 회사들을 위한 것이 아니라 Sony 자사제품을 위한 모델로 보여진다. 반면에, 본 연구자들이 개발한 퍼스널 로봇 플랫폼의 구조는 이러한 로봇관련 회사들이 특정 모듈 또는 부품들을 개발하는데에만 노력할 수 있도록 하기 위해 카메라, 적외선 센서, 구동장치, 관절과 같은 범용적인 제품들의 효과적 통합을 위해 설계 되었다. 바꾸어 말하면, 퍼스널 로봇의 호환성과 모듈화를 포함하는 표준화에 부합한다고 할 수 있다.

이처럼 본 연구자들이 개발한 구조는 퍼스널 컴퓨터처럼 로봇 개발의 표준이 되는 것을 목표로 하고 있다. 그러나 퍼스널 로봇의 표준화는 퍼스널 컴퓨터와는 다른 구조적 차이를 가진다. 퍼스널 컴퓨터는 고정된 장소에서 사용자의 지시에 따라 수동적으로 작동하지만, 퍼스널 로봇은 장애물 같은 주변 환경에 반응해야 하고, 사용자의 명령 없이도 여러 장소를 능동적으로 움직일 수 있어야 한다. 이러한 퍼스널 로봇 기술은 유연한 이동성과 감지기능을 처리할 수 있는 지능에 기반을 두고 있다. 이러한 의미로 퍼스널 로봇은 정적(static)인 것이 아니라 동적(dynamic)이어야 하고, 일반화하기 어려운 다양한 전략(strategy)과 알고리즘을 쉽게 반영할 수 있는 지능 기술이 개발되어야만 한다. 이러한 퍼스널 로봇의 동적인 특성은 장애물 등에 실시간으로 반응하여 주행하는 동작을 가능하게 하며, 새롭고 다양한 인공지능에 스스로 적용하게 하는 것을 의미한다. 다시 말해서, 이 논문에서 소개하는 동적특성(dynamic feature)이란 두 가지를 의미한다. 하나는 기능적으로 분산된 처리를 하더라도 실시간 기반의 작업을 지원해야 하고, 또 다른 하나는 재구성(reconfigurability) 능력, 즉 새로운 조합과 구성에 대한 적용이 가능한 유연하고 쉬운 결합 구조를 지원하는 것을 의미한다.

III. DRP I 의 구조

이전 장에서, 본 논문에서 소개하는 로봇의 구조적인 특징으로 동적특성을 강조하였다. 이번 장에서는, 이러한 프레임워크의 주요 구성요소를 설명한다. 그 구성요소는 기능적으로 특징지어진 Module-D와 각 Module-D의 호환성을 지원하는 VM-D와 RPL이다.

1. Module-D

이 장은 퍼스널 로봇의 설계, 제어, 구현과 같은 다양한 분야의 개발자들을 위한 새로운 방법을 Module-D에 초점을 두고 설명한다. 로봇 어플리케이션의 다양한 소스(source)는 Module-D를 통하여 통합된다. Module-D는 몇 가지 다른 플랫폼에서, 다양한 언어로 작성되어 실행 될 지도 모른다. 그럼에도 불구하고, 이러한 Module-D가 같은 인터페이스

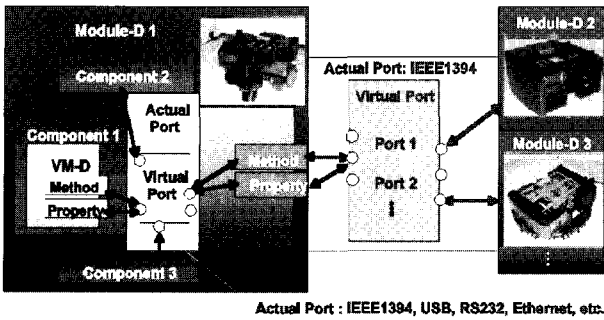


그림 2. Module-D의 인터페이스와 포트의 구성.
Fig. 2. Module-D interface and port configuration.

(interface)에 같은 수준의 특성을 제공하는 한, Module-D는 재구성, 교체 또는 교환, 재프로그램(reprogram) 될 수 있다. 이런 기능은 전체 시스템에 매우 큰 유연성을 부여한다. 따라서 Module-D는 회사들이 그들의 저수준 구성 요소를 개발할 수 있게 하는 통합 환경과 플랫폼을 제공한다. 이러한 능력은 Module-D가 VM-D에 의해서 중재되기 때문에 실현될 수 있다. VM-D는 Windows와 Linux 같은 OS의 상위 레벨에서 Module-D에 적용되며 Module-D에 속한 컨트롤러와 같은 구성요소들을 구조적으로는 간접적이지만 내부적으로는 직접적으로 조종하게 되고, 다른 Module-D의 구성요소들에게도 쉽게 접근할 수 있다.

그림 2는 Module-D들의 공통 인터페이스를 보여 주고 있다. 각 Module-D는 자신의 프로퍼티(property)와 메소드(method)를 가진다. 프로퍼티란 Module-D에 한정된 값을 의미하는데 Module-D의 특징을 결정하고, Module-D에 특정 값을 할당하거나 Module-D로부터 할당받기 위해 사용된다. 메소드는 각 Module-D의 어떠한 작동 또는 행동의 통제를 위한 절차를 의미한다. 예를 들어 로봇에게 이동하라고 지시할 때, '이동하라'라는 명령은 Mobile Module-D에 속하는 메소드 중 하나이다. 프로퍼티, 메소드와 함께, Module-D의 내부 상태(internal state)와 객체(objects)는 다른 Module-D들에 의해서 접근될 수 있다.

그림 1에서는 각각의 Module-D의 물리적인 하드웨어를 보여주고 있으며 각 Module-D의 내부적 기능과 특징은 다음과 같다.

- Main Module-D는 전체 시스템의 상황과 다른 Module-D 상황의 분석 정보 또는 저수준 데이터(low data)를 종합적으로 다루고 있다. 또한 과방전 방지(over-discharge protection)와 원격 파워 스위칭 기능(remote power switching)과 같은 전원관리와, Module-D간의 네트워크 중재와 작업자와의 무선 LAN 접속 연결과 같은 통신관리를 담당한다.
- Sensor Module-D는 사람의 감각 기관과 같은 다양한 감지와 탐색을 위한 부품, 요소들과 제어 컨트롤러로 구성되어 있다. 즉, 초음파 센서, 적외선 센서, 인체감지 센서 등의 정보를 통합 관리하도록 설계되었다.
- Mobile Module-D는 DRP I의 이동을 담당하며 자체 설계된 현가장치를 장착한 차동구동방식을 택하고 있다.
- Vision Module-D는 스테레오 카메라와 효율적인 주변 탐

As a part of DRP I	As an assembly of components
Integrated into DRP I	Integrating its components
Combination of Module-Ds	Decision by Module-D itself
Dependent on DRP I	Independent of DRP I

Deliberative control	Reactive control
Slower response	Real-time response
High-level intelligence	Low-level intelligence
Variable latency	Simple computation
Representation-dependent	Representation-free
Predictive capabilities	Reflexive capabilities

그림 3. Module-D의 이중적 특성.
Fig. 3. A dual characteristic of Module-D.

색을 위한 3 자유도의 팬틸트(pan-tilt)를 장착하고 있으며 비전 이미지 계산, 처리를 위해 설계되었다.

이 논문에서 서술하는 Vision과 Sensor, Mobile같은 Module-D는 DRP I를 구성하기 위해 설계 되었다. 따라서 어떤 다른 Module-D, 예를 들어 음석인식, 로봇 팔, 전방향 이동, 청소를 담당하는 Module-D도 장착할 수 있는 구조로 설계되었다. 실제로 또 다른 버전의 Module-D 기반의 로봇은 DRP I과는 다른 Module-D로 구성되어 있다.

2. Module-D 제어의 이중적 특성

일반적으로, 모듈(module)이란 특정 기능 또는 작업을 수행하는 독립된 소프트웨어와 하드웨어를 가지는 하나의 개체로 정의된다. 제안하는 Module-D는 전기적, 기계적 동일한 규격에 따라, 전체 시스템에 설치되거나, 다른 Module-D로 교체될 수도 있고, Module-D 자체만으로도 독립적으로 사용될 수 있다. 이러한 Module-D의 기능들은 일반적인 모듈의 의미와 비슷하다. 각 Module-D는 비슷한 특징을 가지거나 특정 요구들을 처리 할 수 있게 그룹 지워진 하위모듈(sub-module)들로 구성된다. 다시 말하면, 다른 Module-D에게 비종속적인 하나의 Module-D는 전체 시스템의 일부분(part)라 할 수 있고, 다양한 하위모듈을 통합하는 통합체(assembly)라고도 할 수 있다. 그러나 이러한 Module-D의 개념은 일반적인 모듈의 의미와는 구별되는 특징을 가지고 있다. 그림 3에서 설명하듯이 Module-D는 숙고형(deliberative)와 반사형(reactive) 제어를 만족시키기 위해 계획되고 설계되었다. 어떤 확실한 임무(mission) 또는 시나리오(scenario)를 실행하기 위해 각각의 Module-D는 모든 Module-D를 통합하여 제어하는 상위수준 지능의 통제 하에 로봇의 일부분으로서 조합되어 작동한다. 이것은 이 전 장에서 언급했던 다양한 조합과 구성에 대한 적응성을 갖춘 결합기능을 지원하는 것을 의미한다. 반면에 각 Module-D는 그 자신만이 관리하는 저수준 지능 제어 하에서 스스로 내부 구성요소인 하위모듈을 결합하여 제어하는 타 Module-D에 독립적인 통합체이기 때문에, 긴급 상황에 단독으로 상위수준 통합지능의 통제 없이 즉시 행동할 수 있다. 예를 들면, 로봇이 '거실로 가서 TV를 켜라'라는 명령을 들었을 때, 부여 받은 임무를 분석하고, Module-D들의 기능들이 조합되어 임무를 수행한다. 반면에 필요한 Module-D의 기능들을 조합하여 현재의 임무를 수행하고 있을 때, 로봇의 움직이는 경로에 갑작스러운 어린이가 출현한다면 상위수준의 통합지능의 관여

없이 즉시 Mobile Module-D는 다른 Module-D나 임무에 상관없이 아이를 피하기 위해 구동모터를 제어한다.

이러한 행위기반 제어(behavior-based control)를 위해 필요한 저수준 데이터(low-level data)와 숙고형 제어(deliberative control)를 하기 위한 명령들을 전송하기 위해 실시간성을 보증할 수 있는 IEEE1394[16]가 Module-D간 통신 네트워크에 사용된다.

3. VM-D의 개요

가상머신(virtual machine)이란 컴파일러(compiler) 또는 해석기(interpreter)가 통합된 임베디드(embedded) 시스템에 이식성을 부여하는 일반적 기법을 말한다. 1995년에 Sun Microsystems는 자바기술(Java technology), 자바언어(Java language), 자바가상머신(Java virtual machine)을 발표했다. 이러한 자바 환경은 서로 다른 종류의 OS나 하드웨어에서도 소프트웨어의 작동을 용이하게 만든다. 이 기술은 제품 개발 방향에 많은 영향을 미치고 있다. 자바 가상머신이 설치된 후에는 많은 시스템들이 높은 이식성을 제공받게 된다. 그러나 자바가 네트워크상의 분산된 로봇 모듈을 개발하는데 사용될 때, 개발 시간, 실행 수행 능력에 손실을 가져온다. 다시 말해서, 자바는 하드웨어 제어를 위한 저수준 명령(low-level commands) 처리와 자체의 해석기로는 바이트 코드(byte codes)를 빠르게 실행할 수 없기 때문에 실시간제어가 필요한 로봇 시스템에는 부적당하다.

이 논문에서 소개하는 VM-D(Virtual Machine of DRP I)라 불리는 새로운 로봇용 가상머신은 자바 가상머신과 마찬가지로 이식성을 가지며 시스템에 구매 받지 않는 장점을 가지고 있다. 더불어 실행 속도는 보다 빠르며, 로봇 개발에 꼭 필요한 기능들과 자주 쓰이는 알고리즘들이 제공 된다.

4. VM-D의 구조

VM-D는 다른 종류의 하드웨어 플랫폼과 Windows, Linux, RTLinux같은 여러 종류의 운영체제에 설치 될 수 있다. VM-D는 Module-D에서 제어 프로그램의 바이트 코드를 운영하기 위해 사용되며 Module-D상의 VM-D는 가상 포트(virtual port)를 통하여 다른 Module-D들과 서로 통신할 수 있다(여기서 가상 포트란 point-to-point 연결을 위한 통신 채널을 말한다). 물리적인 통신을 위해 설치된 실제 포트(actual port)는 그림 2에 나타나는 바와 같이, IEEE1394(FireWire), USB, RS-232, TCP/IP Ethernet 등이며 Module-D 사이의 메인통신은 IEEE1394를 채택하고 있다. 이러한 VM-D는 스택(stack) 기반으로 구성되어 있는데 VM-D는 현재 사용되는 프레임(frame)의 오퍼랜드(operand) 스택에서 하나 또는 그 이상의 오퍼랜드를 가져오고, 그 결과 값을 돌려준다. 새로운 프레임은 매번 생성되면서 메소드가 발생하게 되고 이러한 프레임은 새로운 오퍼랜드 스택과 메소드가 사용하는 지역 변수(local variables)를 생성하게 된다. VM-D는 레지스터(register), 메모리(memory), 실행자(executor)로 이루어져 있고, 어플리케이션 프로그램 실행 동안의 런타임(runtime) 메모리 영역을 정의한다. 이 영역은 다시 말해서, 바이트코드(byte code), 전역변수(global variable), 함수(function), 포인터(pointer), 스택(stack), heap 영역을 말한

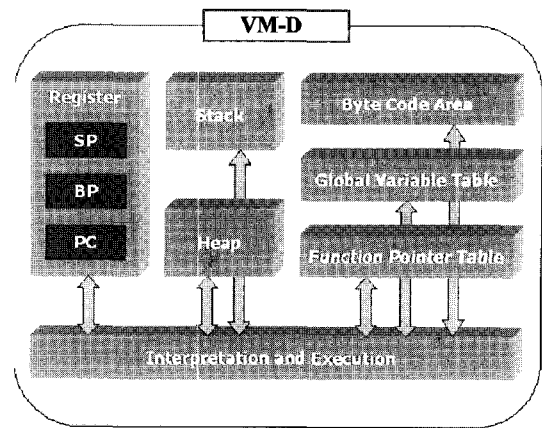


그림 4. VM-D의 구성.

Fig. 4. Configuration of VM-D.

다. 이와 같은 VM-D의 구조는 그림 4와 같고, 그 구성요소는 대표적으로 다음과 같다.

• 레지스터

VM-D는 세 개의 32-bit 레지스터를 가지고 있다. 현재 실행되는 VM-D 명령 주소(address)를 저장하거나 스택에 접근하는데 사용된다. SP(Stack Point)는 메모리에서 한 스택의 상위요소(top element) 주소를 기억한다. BP(Base Pointer)는 메모리내의 한 스택의 베이스 포인트(base point) 주소를 기억한다. PC(Program Counter)는 다음에 실행될 바이트코드의 주소를 저장한다.

• 스택

VM-D의 각 작업은 그 작업과 동시에 생성되는 한 개의 개별 스택(private stack)을 가진다. 스택은 C언어처럼 보편적으로 사용되는 언어와 유사하다. 즉, 스택은 지역(local) 변수와 부분적인 결과와 메소드에 대한 매개변수(parameter)와 메소드에 대한 결과를 받는다. VM-D의 모든 명령은 오퍼랜드 스택으로부터 오퍼랜드를 가져오고, 실행하고, 그 결과 값을 스택에 돌려준다.

• Heap

힙(heap)은 런타임 데이터 영역이다. 모든 스트링(string)과 배열(array)은 이 영역으로부터 할당되며, heap은 VM-D가 실행되면 자동으로 생성된다.

• 바이트 코드 영역

바이트 코드 영역(area)은 일반적으로 쓰이는 언어나 텍스트 세그먼트(text segment)에서 컴파일 된 저장소와 유사하다. 이 영역에는 바이트 코드와 기호 테이블(symbol table)이 축적 된다.

• 전역 변수 테이블

전역 변수 테이블은 VM-D의 모든 작업에서 공유된다.

• 함수 포인터 테이블

함수 포인터 테이블(function pointer table)은 함수본체의 주소를 저장한다.

• Instruction Set

바이트 코드라고도 통용되는 인스트럭션(instruction)은 1 바이트의 오퍼코드(opcode)와 오퍼랜드로 구성된다. 오퍼

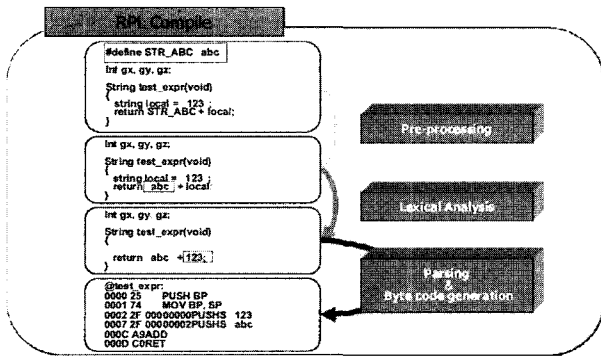


그림 5. RPL 컴파일 과정.
Fig. 5. RPL compile process.

코드는 수행될 작업을 정의하고 오퍼랜드는 작업에 사용될 데이터, 매개변수를 제공한다. 인스트럭션은 오퍼랜드를 가지지 않거나, 한개 또는 두개의 오퍼코드를 가진다. 바이트코드 스트림(stream)은 단지 바이트단위 결합을 가지게 되고 이러한 결합은 저장량으로 컴파일된 프로그램을 위한 VM-D의 코드를 만들게 된다.

5. RPL

본 연구자들은 VM-D가 적용되는 퍼스널 로봇의 어플리케이션 프로그램을 더욱 쉽게 작성하기 위해, RPL(Robot Programming Language)을 개발하였다. VM-D는 각기 다른 OS나 Mini-OS를 제어하고 조정하여 프로그램들이 원활히 동작하게 하기 위해 사용되고, 반면에 RPL은 VM-D에 의한 다양한 OS들의 통합을 기반으로 한 퍼스널 로봇 어플리케이션 프로그램의 개발자들을 지원하기 위하여 개발되었다. RPL은 C++, FORTRAN, PASCAL같은 프로그램 언어이며 RPL로 작성된 프로그램은 그림 5와 같은 바이트 코드로 변환되기 위해 RPL 컴파일러에 의해서 컴파일 된다.

RPL은 데이터 타입(data type)을 가지고 있으며 변수의 데이터 형은 변수가 포함할 수 있는 값을 결정하거나 그 값들이 수행될 동작을 결정한다. RPL은 이러한 정수형, 부동소수형, 문자열 등의 데이터 타입 모두를 지원한다. 컴파일 하는 동안 RPL 소스코드는 연속적 토큰(token)으로 표현되며 RPL은 다섯 종류의 토큰인 식별자(identifier), 키워드(keyword), 문자(literal), 연산자(operator), 다양한 분리기호(separator)를 인식한다. 탭(tab), 캐리지리턴(carriage return), 라인피드(line feed), 주석과 같은 여백(white space)은 토큰들을 분리할 수 있게 지원해 준다. 그림 5에서 지시하듯이 선행처리기(pre-processor)와 컴파일러는 소스코드를 바이트코드로 컴파일 하는 동안 선행처리(pre-processing), 사전적 분석(lexical analysis), 문법분석(parsing), 연결(link)의 네 단계를 거친다. 이러한 바이트코드는 내부적인 가상 포트를 통해 Module-D로 다운로드 되고, Module-D의 VM-D에서 기계어로 변환, 실행된다. Module-D를 제어하기 위해, VM-D는 'mobile.goto(x,y)'와 같은 Mobile Module-D와 Sensor Module-D의 내부명령들이 조합된 특정 작업 명령을 지칭하는 함수를 호출하거나 'mobile.readEncoder(motor1)'와 같이 Mobile Module-D에서만 제공하는 내부적인 기능성 함수를 호출한

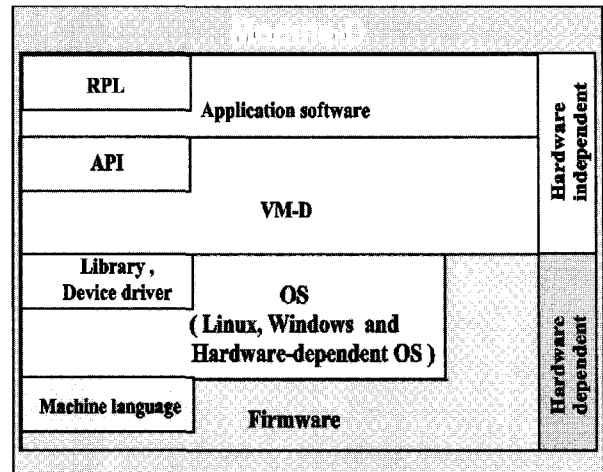


그림 6. Module-D의 소프트웨어 구조.
Fig. 6. Software structure of Module-D.

다. 이러한 함수는 API(Application Programming Interface)라 명명하였다. 다시 말하면, API는 저수준 함수들을 조합한 집합체를 의미하기도하며, 저수준 함수 그 자체를 의미하기도 한다.

그림 6에서 설명하는 바와 같이, API와 VM-D 레이어(layer)는 어플리케이션 소프트웨어를 하드웨어 종속으로부터 분리하여 하드웨어에 무관하게 동작 시킬 수 있도록 한다. 이러한 기능은 VM-D가 Module-D에 기본적으로 설치되어 다양한 하드웨어에 이식될 수 있도록 개발되었기 때문에 가능하며, 매우 유용한 함수들의 집합체인 API는 로봇의 어플리케이션 소프트웨어를 쉽게 개발 할 수 있도록 해준다. 예를 들어, 네비게이션과 이미지 프로세싱(image processing)과 같은 종속적 또는 독립적인 Module-D기반의 소프트웨어 프로그램들은 VM-D상에서 운용되기 때문에 하나의 Module-D는 다른 Module-D의 API를 쉽게 작동 시킬 수 있다.

IV. 구현 및 실험

이 장에서는, 본 논문에서 제안한 프레임워크의 구현에 대하여 설명한다. 제안한 프레임워크 기반의 DRP 1의 기본 메커니즘은 모듈형(modular) 구조이다. 이러한 구조의 동적 재구성 기능을 보장하기 위해, Module-D간의 연결과정에서의 안전성과 연결된 상태에서 작동정지 없이 성능이 유지되는 능력이 고려되어야 한다. 각 Module-D간의 결합을 통해 전체 시스템이 완성되고 전체 시스템에서 각 Module-D의 분리할 수 있는 기능은 기계적, 전기적인 하드웨어의 재구성과 소프트웨어적으로 재구성이 가능한 연결이라는 두 가지 모두에 적합한 방식으로 구현하였다. 근본적으로, 퍼스널 로봇의 모듈화와 호환성을 포함하는 표준화 달성이 고려된 것이다.

1. 하드웨어측면의 동적 재구성

동적 재구성 능력은 그림 7과 같이 Module-D들을 결합하여 다양한 로봇을 구성하는 것을 가능하게 한다(그림에 나타나는 로봇들은 선행되어 개발된 로봇들이며 각각은 호환되는 Module-D로 구성되어 있다)[16, 17]. 이러한 표준화된

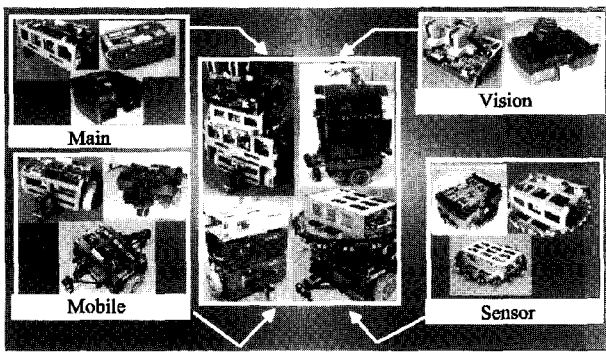


그림 7. 표준화를 기반으로 한 Module-D의 통합.
Fig. 7. Integration of Module-D based on standardization.

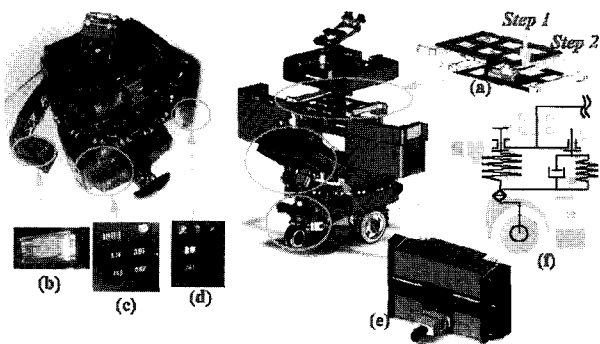


그림 8. DRP I의 기계적 구조. (a) Module-D의 연결 과정, (b) Module-D의 상태 감시 LCD, (c) 전원 로직 감시 제어 장치, (d) 구동 전원 감시 제어 장치, (e) 모듈 배터리의 충전 장치, (f) 다중 완충장치 메커니즘.
Fig. 8. Mechanical construction of DRP I. (a) Module-D connection process, (b) LCD for monitoring of Module-D state, (c) Monitoring and controlling device of power logic, (d) Monitoring and controlling devices of driving power, (e) Charger device with Modular battery pack, (f) Multilayered suspension mechanism.

구조는 Module-D 내의 각종 센서, 모터, 회로와 같은 하위 모듈에도 적용되었다. 그림 8에 나타나는 바와 같이, DRP I은 프레임과 케이스가 쉽게 여닫히는 착탈 방식 구조로 설계되었기 때문에 Module-D 자체에 종속되어 있는 하위 모듈들은 전체 시스템의 해체나 분리 없이 수정, 유지보수, 재조립이 가능하다.

그림 8 (a)에 보이듯이, Module-D간의 전원과 IEEE1394 통신은 자체 설계한 Module-D 전용 커넥터를 통하여 연결된다. Module-D 전용커넥터는 Module-D가 기계적으로 안정적으로 고정된 후에야 전기적으로 접속되기 때문에 Module-D간의 안전한 연결을 보장한다. 이러한 물리적으로 안전한 연결은 IEEE1394와 같은 고속통신과 핫플러그 기능이 노이즈(noise)에 민감하기 때문에 간과할 수 없다. 이러한 기계적으로 안전한 연결과 더불어, 그림 8 (c), (d)에 나타난 장치는 전기적으로 발생하는 문제를 고려하여 고안되었다. 전원은 예상하지 못한 충격과 Module-D의 착탈 시 발생할

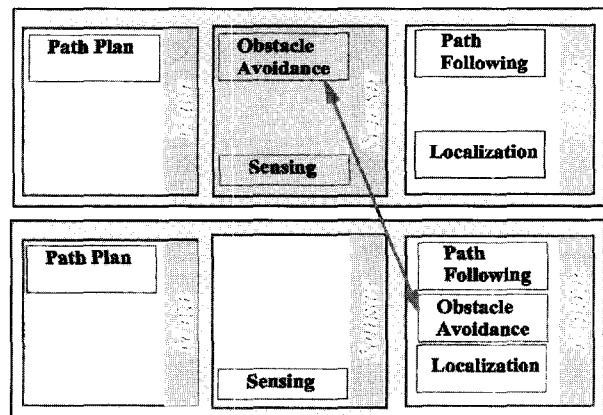


그림 9. Module-D 간의 행위 유연 이동.
Fig. 9. Flexible movement of behavior between Module-D.

수 있는 과전압, 과전류, 전압강하, 잡음 등을 막기 위해 소프트웨어적으로 스위칭 된다. 로봇의 각각의 Module-D의 상태를 안정적으로 유지시키기 위해 전압, 전류, 온도, 로봇 작동 시간은 그림 8 (b)처럼 모니터링 되고 있다(자동적인 제어와 복구는 현재 부분적으로 지원한다). 충전 시 로봇의 작동 정지 없이 전원을 유지하기 위해, 특별히 고안된 충전, 방전장치가 장착되었으며 과방전, 과충전, 과전압, 과전류, 온도 등이 감시되어 로봇의 전체전원을 안정적으로 유지시켜준다. 그림 8 (f)에서 지시하는 바와 같이, 로봇이 이동하는 동안 물리적인 흔들림이나 충격으로부터 시스템을 보호하고 안정된 시스템을 유지하기 위해 자체개발한 차동구동 방식에 적합한 현가장치를 설계하였다[18].

2. 소프트웨어측면의 동적 재구성

동적 재구성 능력은 행위 기반 제어에 있어 유연한 알고리즘 개발을 지원한다. 비록 Module-D 내부의 저수준 작업 수행은 하드웨어와 OS에 종속적이고 작업 처리나 특정 행동 같은 정보는 담당하고 있는 다른 Module-D에 요구하지만 하드웨어와 OS에 비종속적인 VM-D는 이러한 작업을 위해 강력한 호환성을 지원하고, 중앙 통신 네트워크로 채택된 IEEE1394는 분산된 데이터를 각 Module-D에 빠른 속도로 전송한다. 이러한 특성으로 인해 그림 9에서 보여주는 바와 같이 장애물 회피 같은 행위(behavior) 정보는 Module-D 사이에 쉽게 이동한다. 이러한 이동능력을 행위 스와핑(behavior swapping)이라고 한다(여기서 행위이란 API로 구성되거나 API들의 조합을 말한다). 이러한 개발방식은 각각의 Module-D 개발자들이 필요하다고 생각되는 알고리즘을 다른 Module-D의 하드웨어나 저수준 소프트웨어에 대한 특별한 지식이 없더라도 쉽게 개발 할 수 있는 유연성을 제공한다.

그림 10은 Module-D가 핫플러그에 기반을 두고 있다는 측면에서 강조한 DRP I의 동적 특성을 보여주고 있다. 핫플러그는 USB와 마찬가지로 IEEE1394 통신의 노드 감지(node detection)에 의해 생성되는 버스리셋(bus-reset) 기능을 통해 수행된다. 이러한 통신 방법은 로봇의 모듈식 설계에서 가장 중요한 문제 중 하나이다. 각 Module-D의 재구성과의 갑자기 폭주하는 데이터(burst data)의 교환을 가능하게 하기

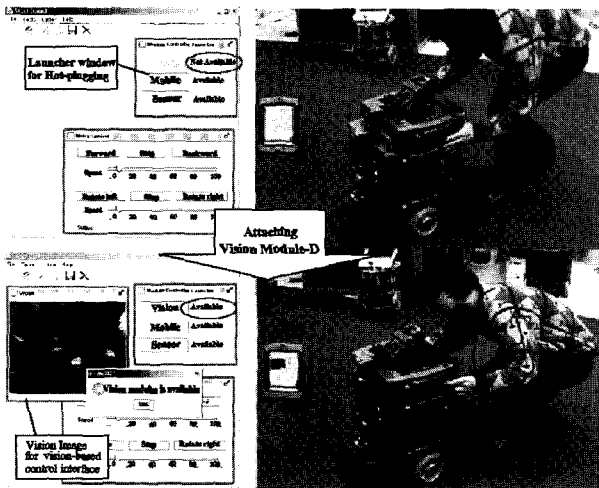


그림 10. 핫플러그 테스트.
Fig. 10. Hot plugging test.

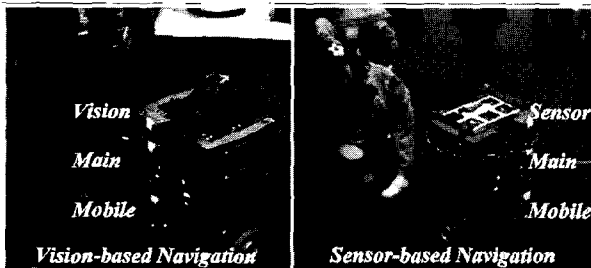


그림 11. 영상 기반 네비게이션과 센서 기반 네비게이션.
Fig. 11. Vision-based and Sensor-based navigation.

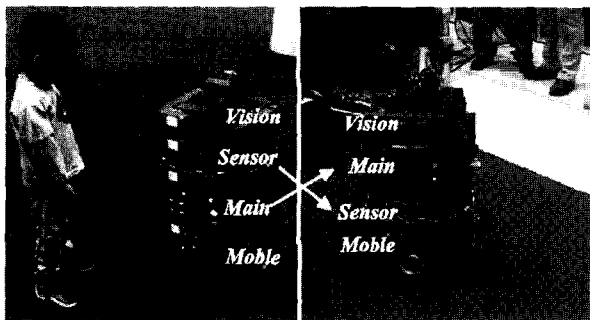


그림 12. 모듈 스와핑.
Fig. 12. Module-swapping.

위해, IEEE1394는 핫플러그와 충분한 대역폭(400Mbps)을 지원하며, 영상과 음성 데이터를 전송하는 비동기 전송을 지원한다. IEEE1394 구성망의 피어투피어(peer-to-peer)는 방식은 각 모듈이 자신의 특성을 유지할 수 있게 해주고, 시스템 작업의 손상 없이 시스템으로부터 분리를 가능케 한다. USB 또한 IEEE1394와 유사한 기능을 가졌으나, 중앙 집중 네트워크 구조(central network architecture)를 요구하는 마스터-슬레이브 인터페이스(master-slave interface)를 가지고 있으며, 노드(node)의 독립성을 유지시키지 못하기 때문에 전체 시스템의 독립적 모듈화 메커니즘을 구축하기에는 부적당한 것으로 보인다.

IEEE1394에 의한 로봇의 모듈화 메커니즘에 있어서, Module-D는 자체 OS를 가지고 있고 IEEE1394에서 전송할 수 없는 대용량의 독립적인 전원을 사용하고 있기 때문에, 핫플러그 기능이 제대로 동작하기 위해서는 소프트웨어적으로 Module-D가 초기화와 종료되어야 하며, 또한 소프트웨어적으로 전원 스위칭 기능이 지원되어야 한다. 다른 만약 특정 Module-D의 결합 시 버스리셋이 발생한다면, Main Module-D는 해당 Module-D의 전원을 키고 초기화 시킨다. 마찬가지로 특정 Module-D가 분리된다면 해당 Module-D를 종료하고 전원을 끈다. 이 과정은 IEEE1394 네트워크를 통해 이루어지도록 설계되었다. 이 처리를 완성하기 위해 SSR(Soild State Relay)과 자체 개발한 전원 소프트웨어 스위칭 회로를 사용한다. 이러한 Module-D들 간의 핫플러그 연결 구조를 바탕으로 DRP I의 주행 알고리즘은 물리적으로 Vision Module-D와 Sensor Module-D를 교체함에 따라 소프트웨어적으로도 자동적으로 센서 기반 네비게이션(sensor-based navigation)이나 영상 기반 네비게이션(vision-based navigation)으로 재구성될 수 있다. 예를 들어, Sensor Module-D가 없는 Vision Module-D만의 영상 기반 주행에서 카메라의 위치와 교정(calibration)은 재구성되고, 새로운 파라미터가 주행에 적용된다. 이와 유사하게, 그림 12에서 나타나듯이, Module-D의 배치 상황에 따른 Sensor Module-D의 각 센서 위치, 높이는 새롭게 주행에 반영된다. 이러한 Module-D의 배치변경을 모듈 스와핑(module-swapping)이라 명명했다.

V. 결론

이 논문에서, 본 연구자들은 새로운 퍼스널로봇의 프레임워크를 소개하고 통합과 인터페이스 기술로서 하드웨어와 소프트웨어의 주된 요소인 Module-D, VM-D, RPL을 소개하였다. 이러한 아이디어는 DRP I의 구조를 동적 재구성이 가능하도록 개발함으로써 구체화, 현실화 시켰다. DRP I와 같은 기본적인 퍼스널로봇 플랫폼은 로봇 팔과 같은 또 다른 물리적인 모듈의 추가에 대해 하드웨어와 소프트웨어적으로 재구성을 쉽게 가져갈 수 있으며, 보안, 안내, 교육 등과 같은 인공지능에 주안점을 둔 로봇으로도 좀 더 쉽게 재구성하여 확장될 수 있다. 향후 본 연구자들의 목표는 다양한 Module-D를 개발하고, 세부적인 기술적요소를 더욱 부가하여 다른 개발자와 공유할 수 있는 보다 효율적이고 적용이 용이한 프레임워크 기반의 퍼스널 로봇 플랫폼을 개발하는 것이다.

참고문헌

[1] P. Fiorini, E. Prassler, "Cleaning and Household Robots: A Technology Survey," *Autonomous Robots*, vol. 9, no. 3, pp. 227-235, December 2000.
 [2] E. Prassler, A. Ritter, C. Schaeffer, P. Fiorini, "A Short History of Cleaning Robots," *Autonomous Robots*, vol. 9, no. 3, pp. 211-226, December 2000.
 [3] B. J Lee, H. G. Lee, J. H Lee, G. T. Park, "New architecture for mobile robots, in home network environment using Jini," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 1, pp.

- 471-476, May 2001.
- [4] G. Dudek, M. Jenkin Computational Principles of Mobile Robotics, Cambridge United Kingdom : Cambridge University Press, 1998.
- [5] T. Fong, C. Thorpe "Vehicle Teleoperation Interfaces," *Autonomous Robots*, vol. 11, no. 1 pp. 9-18, July 2001.
- [6] R. Alami, M. Herrb, B. Morisset, R. Chatila, F. Ingrand, P. Moutarlier, S. Fleury, M. Khatib, T. Simeon, "Around the lab in 40 days," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 88-94, April 2000.
- [7] A. Goeurquiev, P.K. Allen, E. Gold, P. Blaar, "Design, Architecture and Control of a Mobile Site-Modeling Robot," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3266-3271, April, 2000.
- [8] F. Michaud, J. Audet, D. Letourneau, L. Lussier, C. Theberge-Turmel, S. Caron, "Experiences with an Autonomous Robot Attending AAAI," *IEEE Intelligent Systems*, vol. 16, no. 5, pp. 23-29, Sept 2001.
- [9] M. Beetz, T. Arbuckle, T. Belker, A.B. Cremers, D. Schulz, M. Bennewitz, W. Burgard, D. Hahnel, D. Fox, H. Grosskreutz, "Integrated, plan-based control of autonomous robot in human environments," *IEEE Intelligent Systems*, vol. 16, no. 5, pp. 56-65, Sept 2001.
- [10] M. Fujita, "Digital Creatures for Future Entertainment Robotics" *Proc. of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 801-806, April 2000.
- [11] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, K. Yamane, "Open Architecture Humananoid Robotics Platform" *Proc. of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 24-30, May 2002.
- [12] A. H. Jones, G. N. DeSouza, A. C. Kak, "A Multi-processing Software Infrastructure for Robotic System," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 193-198, May 2001.
- [13] M. Lindstrom, A. Oreback, H. I. Christensen, "BERRA: A Research Architecture for Service Robots," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3278-3283, April 2000.
- [14] E. Guglielmelli, C. Laschi, P. Dario, "Robots for personal use: humanoids vs. distributed systems," *International Symposium on Humanoid Robots*, October 1999.
- [15] Linux 1394 Library. <http://www.linux1394.org>.
- [16] S. G. Roh, S. M. Baek, D. H. Lee, K. H. Park, T. K. Moon, H. S. W. Ryew, J. Y. Kim, T. Y. Kuc, H. S. Kim, H. G. Lee, and H. R. Choi, "Development of Personal Home Robot Platform : Designed Approach for Modularization", *International Conference on Control, Automation and System(ICCAS)*, pp. 2313-2318, 2002.
- [17] S. G. Roh, K. H. Park, K. W. Yang, J. H. Park, H. S. Kim, H. G. Lee, and H. R. Choi, "Dynamic Infrastructure for Personal Robot : DynI", *International Conference on Control, Automation and System(ICCAS)*, pp. 2039- 2044, 2003.
- [18] J. H. Park, S. G. Roh, K. H. Park, H. S. Kim, H. G. Lee, and H. R. Choi, "Design of Multilayerd Suspension Mechanism for Differential Type Mobile Robot", *International Conference on Control, Automation and System(ICCAS)*, pp. 859- 864, 2003.
- [19] S. G. Roh, K. H. Park, K. W. Yang, H. S. Kim, H. G. Lee, and H. R. Choi, "Development of Dynamically Reconfigurable Personal Robot", *IEEE International Conference of Robotics and Automation (ICRA)*, pp. 4023-4028, 2004.



노 세 곤

2000년 성균관대 기계공학과 졸업. 동대학원 석사(2002). 2002년~현재 성균관대학교 대학원 메카트로닉스공학 박사과정. 관심분야는 퍼스널로봇, 배관 검사로봇.



양 광 응

1996년 인하대학교 자동화공학과 졸업. 석사졸업(1998). 1998년~2002년 두산 메카텍 근무. 2002년~현재 한국생산기술연구원 근무. 관심분야는 로봇 소프트웨어 구조, 가상머신, 컴파일러.



박 기 흥

2002년 성균관대학교 기계공학과 졸업.
2004년 동대학원 석사(2004). 2004년~
현재 삼성전자 근무. 관심분야는 모바일로
봇, 로봇 프로그래밍.



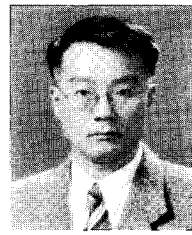
박 진 호

2003년 성균관대학교 기계공학과 졸
업. 2004년~현재 동대학원 석사과정.
관심분야는 모바일로봇, 환경인식.



오 기 용

2004년 명지대학교 기계공학과 졸업.
2004년~현재 성균관대 기계공학과 대
학원 석사과정. 관심분야는 모바일로
봇, 로봇프로그래밍.



김 흥 석

1980년 서울대학교 전기공학과 졸업.
서울대학교 제어계측공학과 석사(1983).
동대학원 박사졸업(1990). 1983년~
1991년 한국과학기술연구원 근무. 1991
년~현재 한국생산기술연구원 근무. 관
심분야는 제어이론, 제어기 설계 및 평
가, 시뮬레이션, 로보틱스 및 S/W 개발환경.



이 호 길

1980년 한양대학교 기계공학과 졸업.
오사카대학교 기계공학과 석사(1986).
1989년~1991년 일본 ASTEM 연구소
근무. 동대학원 박사(1998). 1991년~현
재 한국생산기술연구원 근무. 관심분야
는 로봇제어, 신호처리. 로봇환경기술.



최 혁 렬

1984년 서울대학교 기계공학과 졸업.
한국과학기술연구원 석사(1986). 포항
공대 기계공학과 박사(1994). 1986년~
1989년 LG Electronics 근무. 1993년~
1995년 일본 교토대학 객원연구원.
1999년~2000년 일본 AIST 객원 연구
원. 1995년~현재 성균관대학교 기계공학부 교수. 관심분야
는 이동메커니즘, 인공근육 액츄에이터, 다지 관절 로봇 손.