

전동휠체어의 자유주행을 위한 실시간 제어 구조의 개발

Development of Real-Time Control Architecture for Autonomous Navigation of Powered Wheelchair

김성진*, 김병국
(Seong-Jin Kim and Byung-Kook Kim)

Abstract : In this paper, an efficient real-time control architecture for autonomous navigation of powered wheelchair is developed. Since an advanced intelligent wheelchair requires real-time performance, the control software architecture of powered wheelchair is developed under Linux real-time extension Real-time Application Interface (RTAI). A hierarchical control structure for autonomous navigation is designed and implemented using real-time processes and interrupts handling of sensory perception based on slanted surface LRF, emergency handling capability, and motor control with 0.1msec sampling time. The performance of our powered wheelchair system with the implemented control architecture for autonomous navigation is verified via experiments in a corridor.

Keywords : real-time linux, software architecture, autonomous navigation

I. 서론

전동휠체어는 스스로의 힘으로 이동하기 어려운 노인이나 장애인을 위해 개발된 이동 보조 시스템으로, 그의 수요는 점차 늘어나고 있는 실정이다. 그러나, 일반적인 전동휠체어는 사용자의 세밀하고 정확한 수준의 제어를 계속적으로 필요로 하므로 감각 능력이나 조작 능력의 손상을 입은 사람에게 사용이 제한될 수 밖에 없다. 이러한 한계를 벗어나고자 다양한 기능을 갖춘 진보된 지능형 휠체어(Advanced Intelligent Wheelchair)의 개발은 필수 불가결하다.

진보된 지능형 휠체어는 기존의 조이스틱을 사용한 단순한 피드백(feedback) 제어기에 기반한 마이크로 콘트롤러를 사용하는 대신 사용자의 이동성을 강화시키기 위해 이동 로봇학(mobile robotics)의 기술 및 기술 체계들을 많이 사용하였다. 따라서, 지능형 휠체어는 자율주행을 비롯한 여러 가지 휠체어의 기능들을 자동 제어 시스템에 전가하는 소위 공유 제어 시스템(shared control system) 또는 센서에 기반한 제어 시스템(sensor-based control system)이다. 자동 로봇 시스템은 복합 시스템(hybrid system)으로 대부분 계층적인(hierarchical) 제어 소프트웨어 구조를 가지고 있다. 계층적인 구조는 하나의 행동을 하기 위해 각각의 계층에서 결정이 이루어지고 시스템 요소중 단지 그와 관련된 일부분만이 동작하므로 시스템 자원을 매우 효과적으로 사용한다는 것이 큰 장점이다 [9]. 따라서 진보된 지능형 휠체어는 보다 폭넓은 기능을 사용자에게 제공해야 하고, 강인성(robustness)과 안전성(safety)을 고려해야 하는 인간 친화적인 복합 시스템이므로 계층적인 소프트웨어 구조를 가져야 한다.

로봇식 휠체어인 Maid는 행동이 제약된 사람들을 이동성을 강화시켜 주고 약간의 자율 및 독립성을 제공하기 위해

개발되었다[6]. 넓고 혼잡한 지역에서의 자율 주행을 지원하기 위해 WAN(wide area navigation) 제어 구조를 가지고 있다. WAN은 기본 제어 레벨, 전술(tactical) 레벨, 전략(strategic) 레벨로 구성된 계층적인 주행 구조를 가지고 있다. 기본 제어 레벨에서는 속도 제어 모듈이 있고, 전술 레벨에서는 물체 인식, 물체 추적/추정, 장애물 회피 모듈이 있으며, 전략 레벨에서는 목표 선정 모듈로 이루어져 있다.

VAHM의 목적은 휠체어가 가능한 자율적으로 하는 것이 아니라, 사용자에게 너무 많은 부담을 지우지 않고 사용자의 능력을 최대한 이용하는 것이다[7]. VAHM의 소프트웨어 구조는 physical level, local level, global level의 계층적인 구조를 가지고 있다. Physical level에서는 센서와 엔진의 실행 제어 모듈이 있고, local level에서는 환경 인식 제어, 주행 제어, 통신 제어 모듈이 있고, global level에서는 global control 모듈로 이루어져 있다. 자율성의 정도는 사용자의 육체적, 인식 능력 정도에 따라 다르고, 원하는 기능에 연관된 모듈만이 활성화된다.

Bremen Autonomous Wheelchair, Rolland는 장애인과 노약자가 휠체어를 제어 할 수 있도록 도와주는 역할을 하기 위해 개발되었다[10]. Rolland의 시스템 구조는 sensor & act (SAM) module과 상위 응용 모듈로 구성된 계층적인 구조를 가지고 있다. SAM 모듈에서는 센싱과 상위 응용 모듈에서 오는 속도와 방향 각을 하드웨어에게 전달한다. 상위 응용 모듈에서는 Speedwizard로 휠체어의 속도를 관리하고, Doorwizard로 휠체어의 조타각을 관리한다. 최상위 레벨에 route navigation 모듈이 있어서 사용자가 원하는 목표지점을 선택한다. 사용자의 특별한 요구에 따라 다양한 지원 레벨이 활성화 된다.

전동휠체어는 대부분의 사용자가 거동하기 힘든 노인이나 장애인이므로 사용자의 안전(safety)에 대한 고려가 매우 중요한 설계 이슈이다[10]. 하나의 기능을 수행하기 위해 환경 센싱, 위치 추정, 모터 제어 등과 같은 많은 태스크들이 한꺼번에 또는 부분적으로 묶여지며 특정 시간 안에 수행된다. 이러한 태스크들의 잘못된 동작이나 외부의 간섭으로 인한 태스크의 미스(miss)는 예상하지 못한 결과 또는 치명적인 사

* 책임저자(Corresponding Author)

논문접수 : 2003. 9. 24., 채택확정 : 2004. 5. 27.

김성진, 김병국 : 한국과학기술원 전자전산학과 전기전자전공
(sjkim@rtcl.kaist.ac.kr/bkkim@ee.kaist.ac.kr)

※ 본 연구는 한국과학재단 인간친화 복지로봇 우수연구센터(HWRS-ERC)의 지원을 받아 연구되었음.

고로 이어질 수 있다. 따라서, 전동휠체어 시스템은 safety-critical 시스템, 실시간 시스템으로 생각할 수 있다[11]. 자율로봇 시스템의 제어 소프트웨어는 Windows OS같은 범용 운영체제에서 제작되어 왔다. Windows OS는 쉬운 개발 환경, 풍부한 디바이스 드라이버 지원 및 멀티태스킹 등과 같은 장점이 있다. 하지만 본래적으로 실시간 시스템을 위해 만들어진 운영체제가 아닐 뿐 아니라 그래픽 디스플레이에 대한 강한 의존성 때문에 실시간 성능을 갖지 못한다. 더구나 제어를 위한 디바이스 드라이버의 개발이 수월하지 않고, 상당한 개발 시간이 소요된다는 단점이 있다[3].

그래서, 실시간 제어 시스템의 소프트웨어를 구성하고 구현하는데 가장 기본적인 방법은 실시간 운영체제를 사용하는 것이다. QNX나 VxWorks 같은 상용 실시간 운영체제들은 원하는 경성 실시간 성능은 충분히 가지고 있지만 가격이 비싸고 자체 제작한 제어를 위한 디바이스 드라이버의 지원이 약해 확장성이 떨어진다는 단점이 있다[4].

본 논문에서는 개발중인 노약자 및 장애인을 위한 지능형 휠체어 시스템의 자율주행을 위한 효율적 실시간 제어 소프트웨어 구조의 설계방안을 제시한다. 실시간 성능을 내기 위해 운영체제로 Linux의 실시간 확장판인 RTAI 위에 자율주행을 위한 실시간 프로세스 및 인터럽트 등을 유효 적절히 설계 구현한 계층적인 제어 시스템을 구성 및 구현하여 개발 중인 전동휠체어 시스템에 적용, 그 성능을 입증한다. 특히, 실시간 프로세스로서 모터제어, 위치 추정 및 장애물 인식이 실시간으로 수행된 것에 대해 다룬다.

2장에서는 개발중인 휠체어의 하드웨어 플랫폼과 센서 시스템에 대해서 다루고, 3장에서는 실시간 제어 시스템으로서의 소프트웨어 설계를 다루고, 4장에서는 자율주행을 위한 제어 구조에 대해 다룬다. 5장에서는 설계된 구성 요소들을 실제 구현하여 전동휠체어 시스템에 응용, 실험 결과를 기술하고 마지막으로 6장에서 결론을 맺는다.

II. 하드웨어 플랫폼

본 논문에서 사용한 전동휠체어는 그림 1에서 보는 바와 같이 (주) 대세 엠 케어에서 제작된 상용 전동휠체어 PARTNER PW2002를 기반으로 하고 있다. 휠체어의 전원은 2개의 12V 전지를 이용하고, 2개의 구동바퀴와 2개의 보조바퀴를 가지고 있다. 구동바퀴는 DC 모터에 의해 구동되며 최대 속도는 7Km/h이다. 모터에는 32.5:1의 기어와 non-excited 자장 브레이크가 있다. 평상시에는 조이스틱으로 휠체어를 조정할 수 있다.

자율주행을 위해 휠체어의 뒤편에 추가로 설치한 제어 시스템은 산업용 PC를 기반으로 채택하였고 전체 하드웨어 구조는 그림 2와 같다. 기존의 조이스틱을 이용한 휠체어 제어기 외에 PC를 탑재하여 조이스틱을 이용한 사용자 제어 모드와 PC를 이용한 자동제어 모드의 전환이 가능하도록 구성하였다[12]. 제어 장치로는 ISA와 PCI bus를 갖고 있는 Pentium-III 550MHz 64Mbyte의 single on-board computer에 DC 모터의 제어를 위해 PWM 신호를 내고 초음파 센서의 구동을 수행하도록 설계 제작된 ISA 아날로그 및 디지털 인터페이스 카드가 있다.

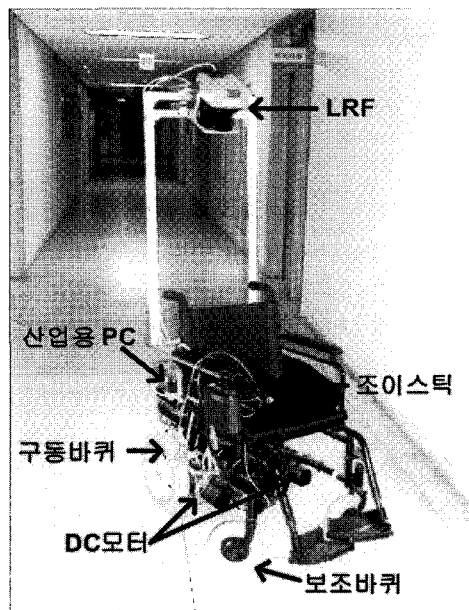


그림 1. 전동휠체어 시스템.
Fig. 1. Powered wheelchair system.

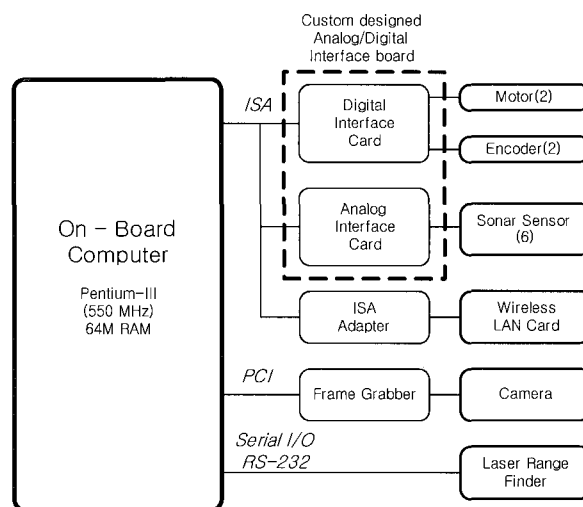


그림 2. 전동휠체어 시스템의 하드웨어 구조.
Fig. 2. Hardware architecture of powered wheelchair control system.

휠체어 시스템의 환경 인식, 장애물 회피, 위치 추정을 위한 센서시스템은 다음과 같다.

- 2D Laser Range Finder(LRF) : 독일 SICK사의 LMS200 모델로 180/100° 스캔 범위에 해상도는 0.25/0.5/1°가 가능하고, 최대 30m까지 물체와의 거리를 측정할 수 있으며 오차는 ±15mm 이내로 높은 정확도를 가진 거리 측정장치이다. 본 논문에서는 그림 3과 같이 LRF를 휠체어에 알루미늄 지지대를 이용하여 스캔 평면이 지면과 기울어지게 하고 휠체어의 축과 LRF의 축은 세로로 잘 정렬되게 설치하여 경사면 측정이 가능하도록 구현하였다. 이는 스캔 평면이 지면과 수평일 때처럼 설치한 높이 이외의 환경정보를 얻지 못하는 단점을 극복할뿐만 아니라, 2D를 이용해 3D 환경 정보를 얻을 수 있

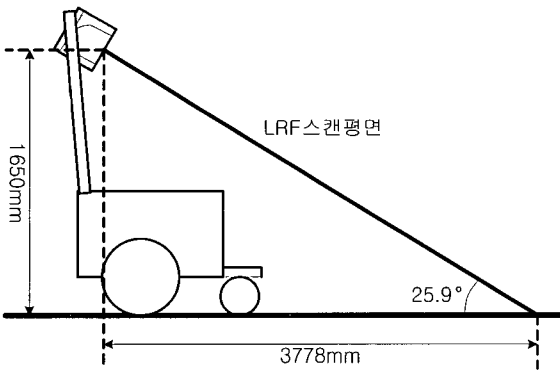


그림 3. LRF의 스캔 평면 및 범위.
Fig. 3. Scan plane and range of LRF.

는 장점이 있다. PC와의 통신은 RS-232 직렬포트를 사용하여 38.4Kbaud로 180° 범위에 1° 해상도 스캔 데이터(362byte)를 받아서 이용하였다. 그림 3에 도시한 바와 같이 LRF 센서 원점의 높이는 1650mm, 스캔 평면이 지면과 이루는 각도는 25.9° 이며, 스캔 평면을 지면과 기울어지게 설치했으므로 스캔 가능한 최대 수평거리는 3778mm이다.

- 초음파 센서 : 무라타 MA40B7 초음파 센서를 전후방에 한 개, 좌우측에 각각 2개씩, 총 6개를 설치하였다. ISA 디지털/아날로그 인터페이스 카드를 이용하여 각 샘플링 시간 마다 초음파를 쏜 후 되돌아오는 시간을 이용하여 휠체어와 벽이나 동적인 장애물과의 거리를 측정한다.
- Dead-reckoning 시스템 : 증가형 엔코더(2개)에 바퀴를 달아 제작된 엔코더 구조물을 이용하여 휠체어 하단에 부착하였다. ISA 디지털 인터페이스 카드내의 엔코더 카운터를 이용하여 휠체어의 방향 및 속도를 구한다.
- Vision Camera : SONY사의 DCR PC-100 카메라를 설치하여 PCI TV 카드를 이용하여 휠체어의 전방 시야를 캡처 한다. 캡처한 영상은 호스트 컴퓨터에서 휠체어의 전방을 관측할 수 있다.

Dead-reckoning 시스템으로 구하는 방향 및 속도를 통한 위치 정보와 방향 정보는 다소 부정확하므로 SICK LRF의 데이터를 이용하여 이를 보정한다. 속도 명령은 주행 시스템에 의해 계산되며 ISA 디지털 인터페이스 카드가 휠체어 모터를 구동시킨다.

통신 장치로 호스트 컴퓨터와 같은 외부 시스템과의 연계를 위해 Lucent사의 Orinoco 무선랜 카드를 이용하여 전동휠체어에 제어 명령을 내리거나 전동휠체어의 내부 시스템 상태 및 휠체어의 위치를 실시간으로 볼 수 있다. 또한 휠체어 제어 프로그램의 수정 및 실행, 확장도 원적으로 가능한 장점이 있다.

III. 자율주행을 위한 소프트웨어 플랫폼

1. 전동휠체어를 위한 실시간 시스템 소프트웨어

전동 휠체어에 안전성은 물론 다양한 기능을 추가하기 위해서는 휠체어의 기계적인 구조 향상도 중요하지만, 실시간 제어 시스템을 추가하여 다양한 종류의 소프트웨어 요소들을 구성하고 요소들간의 데이터 흐름을 정확하게 정의하고,

시스템의 자원을 효율적으로 분배하는 실시간 소프트웨어 구조의 설계 및 구현이 필요하다[2, 5]. 자율주행, 자동충전 등과 같은 진보된 상위레벨의 기능을 제공하기 위해서는 다양한 물리적 구조 하에서 동작하는 센서들로부터 오는 정보를 효과적으로 처리하고, 얻은 정보로부터 센서 융합을 통한 정확한 실시간 위치 추정, 물체 인식 및 추적 등의 하부레벨 요소들이 필요하다. 이러한 각 소프트웨어 구성 요소들이 시간 제약조건(timing constraint)을 만족하지 않으면 사용자는 정적이거나 동적인 물체에 부딪치는 위험한 상황이 발생하게 된다.

본 논문에서의 전동휠체어의 자율 주행을 위해서는 모터 제어, 센싱, 위치 추정, 경로 계획, 긴급 명령 등과 같은 실시간 태스크가 요구된다. 태스크는 각각 다른 주기와 우선 순위, 수행시간, 계산량을 가지고 있다. 이러한 다양한 태스크들은 센서와 모터에 대해 경성 실시간성(hard real-time)을 요구하기 때문에 이를 효율적으로 관리하고 실시간 제어를 위한 소프트웨어 구조가 필수적이다.

다양한 실시간 Linux 확장판 중에서 RTAI와 RT-Linux가 경성 실시간 성능을 제공한다. RT-Linux는 New Mexico Institute of Technology의 Victor Yodaiken 교수가 시작한 것으로, 기본적으로 RTAI와 비슷하다[8]. 하지만, RTAI가 RT-Linux보다 새로운 커널 버전에 대해 실시간 패치를 빠르게 보급하고 있고, 사용자 영역에서의 실시간 응용 프로그램 개발 기능을 가지고 있다. RTAI는 1996년 이탈리아 밀라노대학 항공우주 공학과(Dipartimento di Ingegneria Aerospaziale del Politecnico di Milano, DIAPM)에서 Linux에 경성 실시간 기능을 제공하기 위해 시작되었다. RTAI의 장점은 Linux와 같이 소스가 공개되어 있고 고속의 실시간 테스트 동작, 매우 낮은 jitter를 가지고 있으며 자유롭게 이용이 가능하다. 또한 LXRT(Linux Real-Time)라는 독특한 구조를 가지고 있어서 실시간 RTAI API를 사용자 영역에서 그대로 사용이 가능해 실시간 성능을 내도록 할 수 있을 뿐 아니라, 디버깅이 용이하여 개발에 편리하다[1].

RTAI는 그림 4와 같이 두 가지의 독특한 구조를 가지고 있는데 하나는 실시간 스케줄러이고 다른 하나는 two-level 인터럽트 핸들러이다. RTAI의 성능은 4 usec의 context switch time, 20 usec의 interrupt response time, 100 KHz periodic task, 30 KHz one-shot task rate를 가지고 있다. 이는 VxWorks, QNX 등과 같은 상용의 실시간 운영체제와 경쟁할 정도이다. 게다가 다양한 구조의 프로세서를 지원(UP, SMP, MUP)하고, POSIX 호환, 전통적인 실시간 운영체제의 IPC들, 동적 메모리 할당, FPU 지원 그리고 one-shot, periodic scheduler를 제공한다.

본 논문에서 RTAI를 선정함으로 경성 실시간에 좀더 접근할 수 있었고, 실시간 디바이스 드라이버의 작성으로 경성 실시간 태스크를 위해 하드웨어에 직접적인 접근이 가능하도록 하였고 소프트웨어 구조의 응답시간도 줄일 수 있었다.

2. 계층 제어 구조

그림 5는 본 논문에서의 자율주행을 위한 계층적인 제어 구조를 나타낸다. 이는 실행 주기 및 특성에 따라서 3개의 레벨로 나뉘어지며, 각각의 구성 요소들로 이루어진다. 상위 레벨은 주행목표(Task Goal), 경로계획(Path Planner), 경로발생

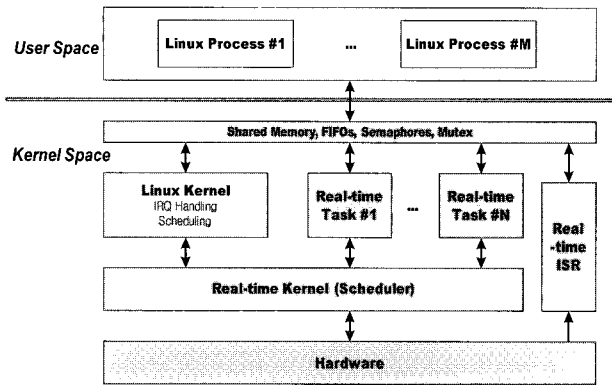


그림 4. RTAI의 구조.
Fig. 4. Structure of RTAI.

(Trajectory Generator)으로 이루어진다. 개발중인 전동휠체어의 주요 목적은 실내 복도 및 집에서 현재 위치로부터 어느 정도 거리에 있는 특정한 목적지까지 도달하는 것이다. 자율주행의 최종 목적지가 정해지면 이를 중간 목적지들로 경로를 계획한다. 그리고, 중간 레벨의 위치추정 결과와 다음 중간 목적지의 위치를 통해 휠체어의 운동학을 이용해 trajectory 모드와 trajectory 값을 계산하여 경로를 발생한다. 이는 산발적으로(sporadically) 일어난다. 주행목표는 오직 사용자의 선택으로만 정해진다. 이러한 상위 레벨은 Linux의 비실시간 프로세스(non-real-time process)들로 구현하였다.

중간 레벨은 위치추정(Localization)과 장애물 회피(Obstacle Avoidance)로 이루어진다. 앞에서 이미 언급했지만, 전동휠체어는 주변 환경을 계속적으로 모니터 하기 위해 LRF를 사용한다. LRF를 이용하여 실제 실험 환경인 실내 복도에서의 위치를 추정하고, 그 결과를 상위 레벨에 전달해 준다. 위치추정의 자세한 내용은 아래에서 다루기로 한다. 위치추정 프로세스의 실행주기는 200msec이며, 하위 레벨에 비해 매우 큰 계산량을 가진다.

3. 실시간 프로세스 설계

하위 레벨에서는 경로추종(Trajectory Following)을 위해 모터 제어 모듈과 모터 구동 모듈로 이루어진다. 모터 제어 모듈은 상위레벨에서 받은 trajectory 모드 및 trajectory 값에 대해 제어 파라미터의 값을 계산하고, 모터 구동 모듈은 실제 모터를 구동하기 위해 계산된 PWM을 내고 dead-reckoning 시스템인 엔코더에 의해 위치를 피드백 받는다. 이러한 모터 제어를 위한 실시간 프로세스의 실행 주기는 0.1msec이며, 작은 계산량을 가진다. 제어 실행 주기마다 타이머 인터럽트 서비스 루틴에 의하여 PWM 출력과 엔코더 값 입력이 신속히 이루어진다.

장애물 회피는 초음파 센서를 사용해 휠체어의 경로에 충돌이 예상되는 정적 또는 동적인 물체가 있는 경우에 동작한다. 장애물이 나타나면 긴급정지 명령을 하위 레벨에 내린다. 후에 LRF를 이용한 물체 인식 및 추적보다 향상된 장애물 인식 및 회피 기능을 수행할 것이다. 장애물 회피를 위해 초음파 Tx. 주기는 100msec이며, 초음파 Rx.는 인터럽트를 이용해 초음파의 TOF(time of flight)를 계산하여 위험 여부

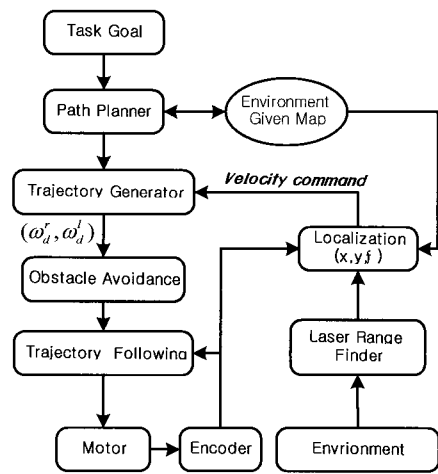


그림 5. 계층적인 주행 구조.
Fig. 5. Hierarchical navigation architecture.

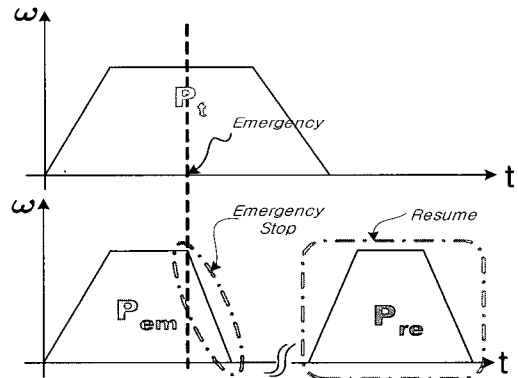


그림 6. 긴급정지후 경로발생.
Fig. 6. Trajectory generation after emergency stop.

를 판단한다. 이러한 장애물 회피 실시간 프로세스는 100msec 주기로 설계 구현하였다.

전동휠체어에 사용자의 안전성을 위해 실시간 제어 시스템을 추가하는 것 외에 안전을 위한 비상처리(emergency handling) 기능이 필요하다. 자율주행시 주변 상황에 대한 반응 시간이 상대적으로 느린 사용자를 위해 다양한 센서로부터 얻은 데이터로 충돌 등과 같은 위험상황일 때 휠체어를 안전하게 제어하기 위해서 최대한 빠른 시간 내에 비상처리 기능이 활성화 되어야 한다.

본 논문에서는 휠체어의 전후, 좌우에 설치한 초음파 센서를 이용하여 휠체어의 주행경로에 물체나 벽이 있어서 충돌이 예상되면, 긴급처리를 위해 긴급정지(emergency stop)를 인터럽트로 다른 태스크들 보다 우선적으로 처리하게 하여 신속한 응답을 보장하도록 하였다. 휠체어가 언제 긴급정지를 수행하는 것을 결정하기 위해 휠체어의 감속 및 브레이킹 능력과 가장 가까운 장애물과의 거리가 고려되었다.

전동휠체어가 위험한 상태에서 벗어나 안전한 상태가 되면 주어진 목표에 도달하기 위해서 이전에 발생된 경로를 따라가야 하므로 그림 6과 같이 속도패턴(reference command

velocity profile)을 변경한다.

경로발생에서 trajectory 모드에 따라 속도제어 모드에서는 긴급정지 명령을 받은 순간 지점의 속도에 최단 시간내 도달 하도록 trajectory를 발생시키고, 위치제어 모드에서는 주어진 기준 위치가 P_i 로 주어졌고 긴급상황이 발생하고 긴급정지를 수행하여 완전히 정지할 때까지의 거리가 P_{em} 이라 할 때, $P_{re} = P_i - P_{em}$ 의 trajectory를 발생한다. 이를 이용하여 이미 주어진 경로를 잘 따라가는 것이 가능하다. 이를 위해 휠체어의 최대 가속 능력이 고려되었다.

IV. 위치추정 알고리즘

자율주행을 위해서 위치추정(Localization)은 필수적이다. 본 논문에서는 전동 휠체어의 위치와 방향(orientation)을 얻기 위해 LRF와 두 개의 엔코더를 사용한다. 엔코더를 통해서 상대적인 위치 추정으로 에러가 쌓이는 단점이 있지만, 샘플링 타임이 짧은 장점이 있다. LRF로부터 얻은 데이터를 인식하여 위치 추정한 결과가 상당히 정확하고 신뢰성이 높지만 계산량이 많다.

Localization과정은 다음의 단계로 이루어지며, 주변 환경 모델은 그림 7과 같다. x_w, y_w 은 실세계 좌표(world coordinate)를 나타내며, x_r, x_l 은 휠체어와 우측과 좌측 벽까지의 측면 위치, ϕ 는 휠체어의 방향(orientation), y 는 휠체어의 세로 위치를 나타낸다.

단계 1(Segmentation) : LRF 데이터 프로세싱 과정으로서 LRF

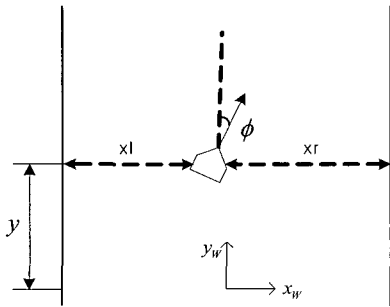


그림 7. 실내 복도 환경 모델.
Fig. 7. Indoor corridor environment model.

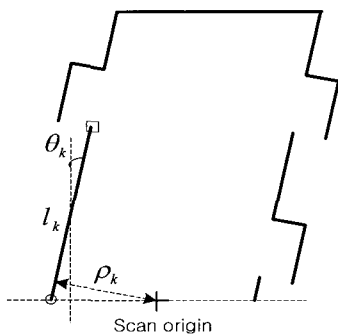


그림 8. 선분으로부터 얻을 수 있는 정보.
Fig. 8. Information from a segment.

로부터 얻은 데이터로부터 선분을 추출하고, 추출된 선분을 least square fitting을 이용하여 보정한다. 보정된 선분으로부터 그림 8과 같이 휠체어의 정면과의 기울어진 각도 θ_k , LRF의 중심과의 수직거리 ρ_k 와 선분의 길이 l_k 를 구한다.

단계 2(Grouping) : 추출된 선분들을 θ_k 에 의해 수직선 그룹과 수평선 그룹으로 나누고, 다시 수직선 그룹을 선분의 시작점과 끝점의 위치에 따라 우측 수직선 그룹과 좌측 수직선 그룹으로 나눈다.

단계 3(Position Correction) : 수직선 그룹으로부터 휠체어의 방향 ϕ 와 좌측과 우측 수직선 그룹으로부터 각각 좌우측 벽까지의 거리 x_r, x_l 를 다음과 같이 구한다.

$$\phi = \frac{\sum l_k \theta_k}{\sum l_k} \tag{1}$$

$$x_r = \frac{\sum l_k \rho_k}{\sum l_k}, x_l = \frac{\sum l_k \rho_k}{\sum l_k} \tag{2}$$

$$x = x_r - x_l \tag{3}$$

여기서 x 는 실세계 좌표에서의 휠체어의 측면위치이다.

위치 추정을 통해서 얻어진 위치 정보를 가지고 모터 제어 태스크에게 다음과 같이 주행을 위한 속도 명령을 준다.

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}_{d,k} = \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}_{d,k-1} + \begin{bmatrix} a_{11} & a_{12} \\ -a_{11} & -a_{12} \end{bmatrix} \begin{bmatrix} \phi \\ x \end{bmatrix} \tag{4}$$

여기서 $\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}_{d,k}$ 는 k번째 원하는 속도이고, $\begin{bmatrix} a_{11} & a_{12} \\ -a_{11} & -a_{12} \end{bmatrix}$ 는 비례상수이다.

주어진 복도 환경의 폭이 고정되어 있으므로 좌우측 벽까지의 거리를 이용하여 휠체어의 위치를 추정하여 휠체어를 제어할 수 있다. 또한 동적인 장애물이 있을 경우 장애물로 인해 나타나는 수직선을 벽으로 가정하여 동적인 장애물을 회피할 수 있다.

V. 시스템 구현 및 실험

지금까지 개발중인 전동휠체어의 하드웨어 구조와 자율주행을 위한 계층적인 주행 구조 및 각각의 요소, 이를 구현하기 위한 운영체제로 RTAI에 대해 살펴보았다. 자율주행을 위한 실시간 제어 구조를 구현하기 위해 3장에서 설명한 소프트웨어 요소들을 배치하고 각각의 실시간 태스크들을 우선순위(priority)와 주기(period)를 할당하여 실시간 스케줄러에 의해 실행이 되도록 RTAI에서 구현하였다. RedHat Linux 7.1을 기반으로 커널 버전 2.4.17에서 RTAI 24.1.10을 패치 및 설치하였다.

표 1은 소프트웨어 구성 요소중 주기적인 실시간 프로세스들의 파라미터를 나타낸 것이다. 주기와 수행시간의 단위는 msec이다. 모든 주기적인 실시간 프로세스들이 주어진 주기 이내, 즉 deadline 이내에 실행이 됨을 확인할 수 있고 CPU 점유율이 13% 이하로 매우 작음을 볼 수 있다. 주기적인 실시간 프로세스의 점유율이 낮으므로 산발적인 실시간

표 1. 각 실시간 프로세스들의 파라미터.

Table 1. Parameters of each real-time process(msec).

	Priority	주기	수행시간	Utilization
모터 구동 프로세스	1	0.1	0.01242	0.124
위치 추정 프로세스	2	200	0.706	0.004
초음파 Tx. 프로세스	3	100	0.109	0.001

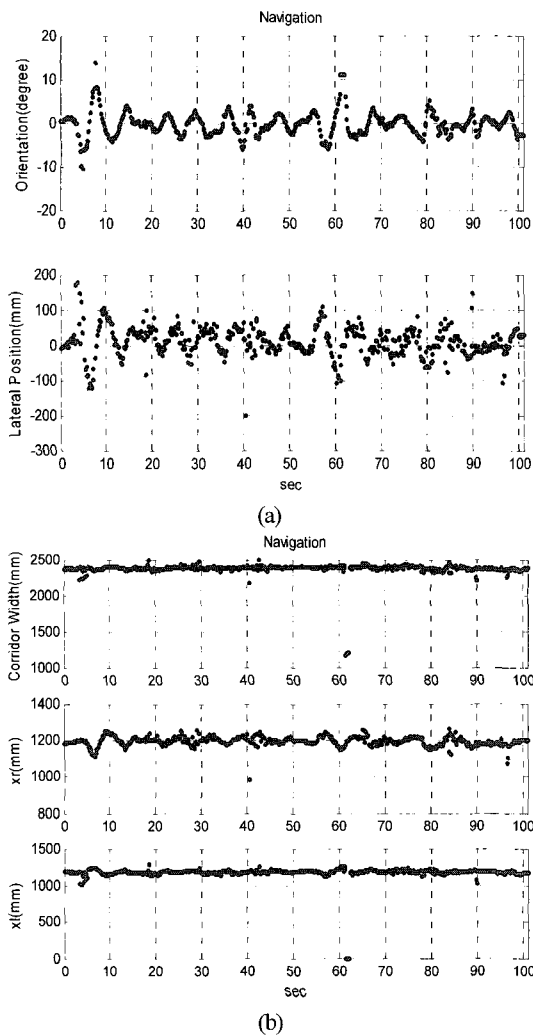


그림 9. 복도주행 실험결과 (a) 방향과 측면위치, (b) 계산된 복도폭과 좌우측 벽까지의 거리.

Fig. 9. Experiment result of corridor navigation (a) orientation and lateral position, (b) calculated corridor width and distance from left/right wall.

프로세스들과 네트워킹과 같은 비 실시간 프로세스들도 문제 없이 실행됨을 알 수 있다. 모터 구동 프로세스의 경우 0.1msec이라는 빠른 샘플링 타임으로 주행시스템의 안정성(stability)을 크게 하고 피드백 센서의 에러에도 강인성과 정확도를 향상 시켰다. 더구나 utilization도 매우 낮음을 볼 수

있다. 위치 추정 프로세스의 경우 초음파 Tx. 프로세스 보다 주기가 길지만 자율 주행을 위해서는 위치 추정이 더욱 중요하므로 우선 순위를 높게 두었다.

구현된 소프트웨어를 이용하여 실내 복도에서의 실험을 수행하였다. 시작 위치는 복도의 한쪽 끝이고 주행 목표는 복도의 맞은편 끝까지이다. 복도의 벽에 평행하게 복도의 중앙을 지나도록 하는 목표를 가지고 실험하였다.

그림 9의 주행 실험 결과로부터 전동 휠체어가 복도에 평행하게 주어진 경로인 복도의 중앙으로 벽에 부딪치지 않고 주행이 됨을 알 수 있다. 출발부터 10초 이내까지 방향과 측면위치의 에러가 큰 것은 휠체어의 초기 위치 에러 때문이다. 62초 근처에서 계산된 복도의 폭이 갑자기 좁아지는 결과를 나타내는데 이는 휠체어의 좌측에 장애물, 사람이 지나가서 나타난 것이다. 그밖에 방향은 $\pm 5^\circ$ 이하의 에러로, 측면위치는 $\pm 50\text{mm}$ 이하의 에러로 위치 추정이 매우 정확함을 알 수 있다. 그래서 복도의 폭은 2340mm로 주행하는 동안 거의 동일한 값으로 나오는 것을 볼 수 있다. 휠체어와 양쪽 벽까지의 거리 계산과 방향을 통한 위치 추정으로 실내 복도에서의 위치추정 성능을 보였고, 이를 이용하여 자율주행이 가능함을 실험적으로 보였다.

VI. 결론

본 논문에서는 개발중인 노약자 및 장애인을 위한 진보된 전동휠체어의 자율주행을 위한 실시간 제어 구조를 개발하였다. 실시간 제어 시스템인 전동휠체어의 하드웨어 설계와 아울러 소프트웨어 구조 설계를 실시간 운영체제인 RTAI 및 Linux를 이용하여 실시간 프로세스 및 인터럽트들로 설계 구현하였고, 인간 친화적인 복합시스템으로서 전동휠체어의 자율주행을 위해 계층적인 구조를 채택하고 위치추정 알고리즘을 제안하였다. 그리고 실제 복도에서의 자율 주행이 가능함을 실험적으로 보였다.

이와 같은 전동휠체어의 자율주행을 위한 실시간 제어 구조를 이용하여 기본적인 주행 기능뿐만 아니라, 주행중에 사용자의 안전을 고려하여 충돌 같은 위험 상황을 맞이 했을 때 휠체어가 이를 인식하여 긴급정지를 통해 긴급상황을 처리 할 수 있다. 게다가 실시간 성능이 뒷받침된 모듈화된 소프트웨어 구조를 이용하여 보다 다양한 센서를 이용하여 사용자가 요구하는 보다 상위 수준인 자동 충전이나 좁은 문과 좁은 복도의 이동 등의 기능을 제공하는 것이 가능하다.

향후 연구과제로써 주어진 맵에서 LRF와 엔코더를 이용한 정확한 위치 추정이 요구되고, 이를 통해 실내 직선 복도 뿐만 아니라 교차가 되는 복도에서의 회전 주행에 대한 trajectory planning이 요구된다. 그리고, 사용자와 휠체어의 긴밀한 상호작용을 위한 사용자 인터페이스와 관련된 연구도 절실히 요구된다

참고문헌

- [1] Lineo, Inc., "DIAPM RTAI programming guide 1.0," <http://www.aero.polimi.it/~rtai/>
- [2] I. S. Song and F. Karray, "Software architecture for real-time autonomous agents : a case study for digital train

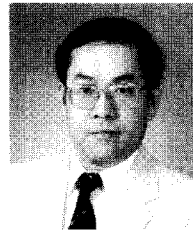
- system," *Proc. of IEEE International Symposium on Intelligent Control*, pp. 403-408, 2002.
- [3] C. H. Lee and C. Mavroidis, "PC based control of robotic and mechatronic systems under MS-windows NT workstation," *IEEE/ASME Trans. on Mechatronics*, vol. 6, no. 3, pp. 311-321, 2001.
- [4] W. F. Lages and E. M. Hemerly, "Linux as a software platform for mobile robots," *submitted to IEEE Transactions on Software Engineering*, 2000.
- [5] R. Brega, N. Tomantis and K. O. Arras, "The need for autonomy and real-time in mobile robotics: a case study of XO/2 and pygmalion," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1422-1427, 2000.
- [6] E. Prassler, J. Scholz and P. Fiorini, "A robotics wheelchair for crowded public environments," *IEEE Robotics & Automation Magazine*, pp. 38-45, Mar. 2001.
- [7] G. Bourhis et al, "An autonomous vehicle for people with motor disabilities," *IEEE Robotics & Automation Magazine*, pp. 20-28, Mar. 2001.
- [8] RT-Linux, <http://www.fsmlabs.com>
- [9] P. Pirjanian, "An overview of system architecture for action selection in mobile robotics," *Laboratory of Image Analysis, Aalborg Univ.*, Denmark, 1997.
- [10] A. Lankenau and T. Röfer, "A versatile and safe mobility assistant," *IEEE Robotics & Automation Magazine*, pp. 29-37, Mar. 2001.
- [11] J. W. S. Liu, *Real-Time Systems*, New Jersey, Prentice Hall, 2000.
- [12] C. H. Kim, J. H. Jung and B. K. Kim, "Design of intelligent wheelchair for the disabled," *Proc. of the International Conference on Rehabilitation Robotics*, 2003.



김 성 진

1976년 11월 28일생 2001년 연세대학교 전자공학과 졸업. 2003년 한국과학기술원 전기전자전공 석사. 2003년-현재 동대학원 전기전자전공 박사과정 재학중. 관심분야는 이동 로봇, 제어 소프트웨어 구조, 네

트워드 제어.



김 병 국

1952년 10월 5일생. 1975년 서울대학교 전자공학과 졸업. 한국과학기술원 전기 및 전자 공학과 석사(1975). 동대학원 박사(1981). 1981년~1986년 우진계기(주) 연구실장. 1982년~1984년 University of Michigan 방문연구. 1986년~현재 한국과

학기술원 전기 및 전자공학과 교수. 관심분야는 실시간 시스템, 로보틱스, 임베디드 제어.