

## 휠구동방식의 이동로봇을 위한 제어시스템설계

고경철\* 조형석  
한국과학기술원 정밀공학과 제어및 자동화 연구실

## A Control System for Wheel-driven Mobile Robot

K.C. Koh and H.S. Cho  
Precision Engineering Department, KAIST

## ABSTRACT

Real-time mobile robot controllers usually have been designed with an emphasis on control theory ignoring the importance of system integration. This Paper demonstrates that useful mobile robots require a real time controller with a wide range of capabilities in addition to control theory. These capabilities include: path-planning, position estimation, path tracking control and wheel control. An architectural framework supporting these capabilities has been designed and implemented. Using this frame work, individual modules such as a path planner, a path tracking controller, position estimators, wheel controllers and other crucial elements have been successfully integrated into the control system for the LCAR robot which was developed as a proto-type mobile robot in our laboratory. The context of the research, the architecture, its implementation and performance results from experiments are discussed.

## 1. 서론

최근 고정 궤도식 무인운반차(AGV)의 결점을 보완하고 유연성을 높이기 위해 자유주행(free-ranging)방식을 채택한 자율 이동로봇(Autonomous mobile robot)에 관한 연구가 활발하다. 이 자유주행방식은 가야할 경로를 탑재된 메모리에 기억함으로써 경로변경이 용이하며, 충돌회피등 돌발적인 상황에 유연하게 대처할 수 있는 장점을 지니고 있다. 반면 이동로봇이 주어진 기준경로를 정확히 따라가도록 하기 위해서는, 현재 운행되고 있는 로봇의 자세(절대위치와 진행각)를 얻기위한 각종 센서시스템 및 처리알고리즘, 그리고 경로로부터 벗어난 현재의 자세로부터 다시 기준경로에 복귀하도록 하는 적절한 경로추종제어 알고리즘등이 필요하다. 지금까지 이러한 이동로봇을 위한 실시간 제어시스템은 시스템구성(System integration)의 중요성보다는 주로 제어이론에 강조를 두어 설계되어 왔다. [1]

최근 10년간 이동로봇을 위한 제어방법에 관한 많은 연구들이 발표된바 있다. Tsumura[2]는 기준경로를 일련의 기준점들로 표현하고, 이를 경로제어의 매 제어사이클에서 로봇의 기준점과 예측된 미래의 위치를 비교하여, 다음의 조향각(Steering angle)을 결정하는 방법을 제시하였고, Kanayma[3]는 이동로봇을 위한 경로로서 기준점대신 직선을 사용하는 방법을 소개 한 바 있다. Singh[4]은 경로궤적오차를 보상하기 위해, 현재점과 경로상의 기준점을 부드러운 곡선으로 연결하는 Quintic polynomials방법을 사용한 바 있다. 하드웨어구조에 대해서는 Crowley등[5]이 3개의 계층구조를 갖는 이동제어시스템(Locomotion control system)을 제시하였는데, 이 이방법에서는 가상의 이동체(virtual vehicle) 개념을 도입하여 직진과 선회 운동 각각의 독립적인 제어가 가능하고, 부드러운 궤적을 얻을 수 있도록 하였다. 이러한 각각의 연구들은 이동로봇의 제어시스템을 이루는 데 중요한 공헌이 되었지만 실제 시스템종합 기술의 측면에서의 폭넓은 연구는 아직 미흡하다고 볼 수 있다. 본 연구에서 기술되고 있는 이동로봇의 제어구조도 이러한 이전

의 연구결과들의 주요내용을 기반으로 하여, 보다 넓은 범위의 능력을 갖는 이동로봇의 실시간 제어를 설계하고 실제 구현을 통해 그성능을 알아보고자 한다. 즉 경로계획기, 경로추종제어 알고리즘, 위치추정기 그리고 휠제어기에 이르기 까지, 이동로봇제어에 필수적인 모듈들을 성공적으로 연결/구성하여, 본 연구실에서 연구용 Proto-type으로 개발된 이동로봇(일명:LCAR-90)에 적용한다.

## 2 LCAR-90의 설계배경

이동로봇 LCAR-90은 크게 몸체와 휠 구동시스템 그리고 제어시스템으로 구성되어 있으며, 그림.1과 같은 외관을 하고있다. 휠구동시스템은 몸체 하단부에 설치되어 몸체의 중양을 기준으로 하여 좌우 2개의 구동휠(Driving wheels)과 전후 2개의 캐스터휠(Caster wheels)로 이루어진다. 따라서 이동로봇은 2개의 구동휠의 속도차에 의해 직진과 선회운동을 하는 속도편차방식(Two wheel difference steering)의 구동방식을 갖는다. 이때 캐스터 휠은 몸체의 진행방향에 따라 수동적으로 회전하여 몸체의 균형을 유지하는 역할을 한다. 그러면 이러한 이동로봇을 위한 제어시스템을 그림.2의 구조로 설계하였으며, 편의상 주 제어 컴퓨터, 휠제어시스템 그리고 센서시스템으로 나누어 그 하드웨어적 구성에 대해 알아본다.

## 2.1 주 제어 컴퓨터

주 제어 컴퓨터는 그림.2에서 보는 바와 같이 전체 시스템을 총괄하는 기능을 한다. 즉 운전중 자동적인 경로계획, 각종 센서 처리알고리즘, 경로추종제어 그리고 Man-machine 인터페이스등을 담당하고 시스템의 상태를 점검하는 감시제어시의 기능까지 수행한다. 또한 하드웨어적으로는 각 하부 모듈들을 종합하여 정보를 전달하는 통신센터의 역할을 한다. 일반 산업용 로봇의 주 제어부는 보통 16비트 마이크로프로세서를 베이스로 하여 On-board 컴퓨터구조로 설계되고있으나, 이러한 주 제어 컴퓨터 구조는 시스템확장에 제한성이 있고, 소프트웨어의 변경이 자유롭지 못한 단점을 안고 있다. 본 제작로봇에서는 시스템확장과 유연한 프로그래밍을 을 고려하여 범용PC(CPU:80386, 24MHz)를 주 제어 컴퓨터로 채용하였다. 따라서 모든 주변장치들은 PC와 Compatible하도록 설계되었으며, PC 확장bus를 통해 연결 된다. 그리고 제어알고리즘이나 센서처리알고리즘의 실시간 수행을 위해 DSP보드(CPU:TMS320C30, 16MHz, 33MFLOPS)를 채택하여 PC의 연산능력을 강화하였으며, 실시간관련 소프트웨어모듈들은 DSP C언어로 프로그래머가 DSP에 다운로드하여 수행되도록 하였다.

## 2.2 휠 제어시스템

좌우의 구동휠은 벨트/풀리방식의 감속기를 통해 각각의 배터 리식 DC서보모터(40watts)에 의해 구동되며, 양구동휠축에는 회전당 2000펄스의 광학식 증가형엔코더(Optical incremental encoder)를 각각 부착하여, 휠의 회전양을 감지할 수 있도록 설계하였다. 구동휠을 제어하기위한 휠제어시스템은 위치제어보드, 통신인터페이스 그리고 구동드라이브등으로 구성된다. 위치

제어보드는 그림.2와 같이 8032마이크로컨트롤러를 이용하여 설계되었으며, 각 구동휠의 위치제어를 담당한다. 그러면 휠제어 시스템과 주제어시스템간의 통신을 포함한 휠제어시스템내의 각 구성요소간의 신호전달체계를 각구성부의 기능과 함께 살펴본다. 주제어시스템은 매 샘플링시간(제어샘플 주기는 상위 컴퓨터인 PC의 연산능력을 고려하여 20msec로 정하였다)마다 각 구동휠이 회전해야할 양(기준회전각)을 계산하고 이를 다시 각 구동모터의 엔코더의 펄스수로 변환하여, 통신인터페이스보드(병렬포트:8255×3개)를 통해 각 구동휠의 위치제어보드(CPU: 8032)로 전달한다. 위치제어보드에서는 PC로부터 받은 펄스데이터를 이용하여 위치명령치를 계산하고, 이를 기준입력을 하여 엔코더귀환을 통한 디지털 위치제어루프를 구성한다. 이 위치제어부의 출력은 아날로그 전압신호로 변환되어 다시 구동드라이브로 전달되며, 구동드라이브(PWM방식)에서는 입력신호를 기준 속도 신호로하여 모터의 속도를 제어한다. 속도귀환제어를 위해 제어모터축에 부착되어 있는 타코미터신호가 이용된다. 이때 속도 제어루프는 속도게인( $K_v$ )의 조절을 통해 제어플랜트의 감쇄효과를 키워주는 기능을 하고, 바깥의 위치제어루프는 비례제어게인( $K_p$ )의 조절을 통해 시스템의 안정과 위치오차를 줄이는 효과를 갖는다.

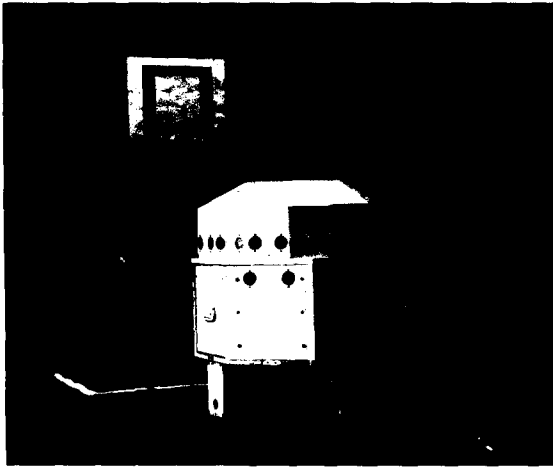


Fig.1 Overview of the mobile robot(LCAR)

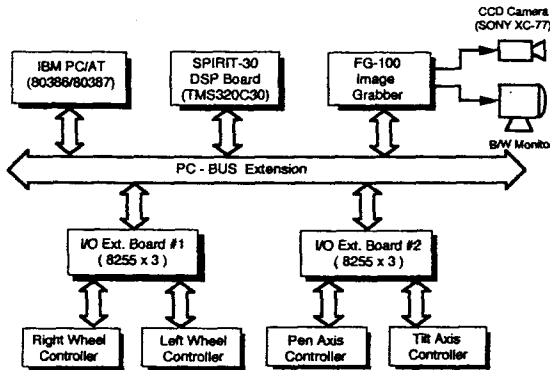


Fig.2 H/W block diagram of the control system

### 2.3 센서시스템

이동로봇은 매니퓰레이터와는 달리 보다 넓은 범위의 공간에서 작업을 하게되므로 될 수 있는한 주위환경에 대한 많은 정보를 필요로 한다. 이러한 목적을 위한 이동로봇의 센서는 측정한계와 특성을 고려하여 선정되어야 하며 본 시스템에서는 시각센서, 초음파센서 그리고 휠엔코더등이 이용되었다.

### 2.3.1 시각센서부

비전카메라를 이용하는 방법은 넓은 범위의 주위환경을 인식하고, 마크를 이용한 이동로봇의 절대 위치정보를 얻는 데 유용하다. 이러한 목적을 위해, LCAR-90의 시각센서부를 설계하였으며, 이는 핸드마크 인식용 CCD카메라 1대, 3차원 주위환경 인식을 목적으로 하는 스테레오비전용 CCD카메라 2대 등 총 3대의 카메라, 이 카메라들을 회전시켜 주는 팬/틸트장치(Pan/tilt device), 카메라영상을 처리하는 디지털타이저(Image digitizer) 그리고 팬/틸트 제어부등으로 구성된다. 팬/틸트장치는 시각영역을 넓히기 위해 사용되며 패닝(Panning)과 틸팅(Tilting)의 2자유도를 갖고 2대의 DC서보모터에 의해 구동된다. 사용된 카메라는 768(H)×493(H)의 분해능과 16mm의 초점거리를 갖고 이미지 디지털타이저는 512×512의 분해능과 256 gray-level을 갖는 PC인터페이스용으로 보드로 선정되었다.

### 2.4 초음파센서부

LCAR-90 이동로봇은 단면이 6각형인 몸체를 갖고 있으며, 그림.3과 같이 그 몸체면에 초음파센서(Ultra sonic sensor)를 둘레방향으로 일정한 간격(상단 18개, 하단 8개)으로 배치하였다. 이 센서들은 주행중에 진행방향을 중심으로 초음파감지영역내의 존재할 가능성이 있는 장애물들을 감지할 목적으로 사용되었다. 초음파센서들은 초음파센서보드(CPU: 8032)에 의해 제어되며, 이 보드에는 초음파를 발생시키는 Transmitter와 Echo신호를 받기 위한 Receiver/Gain제어기 그리고 측정방식에 따라 임의의 센서를 선정해 주는 스위칭모듈로 구성된다.

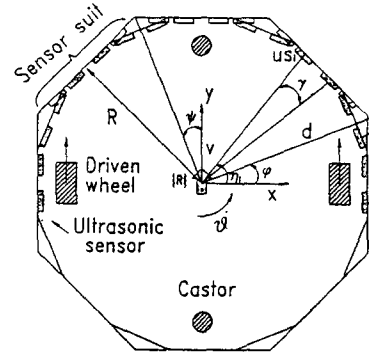


Fig.3 Arrangement of ultrasonic sensors

### 3. 제어시스템설계

이 장에서는 이동로봇의 제어시스템의 주요핵심이 되는 소프트웨어적인 구성에 대해 기술한다. 통상 자율이동로봇은 유용한 동작을 수행하기 전에, 아래의 질문들에 관한 대답을 필요로 한다.

- 현재 어디에 어떤 자세로 위치하고 있는가?
- 어디로 가야하는가?
- 그곳에 어떻게 가야하고 장애물과의 충돌등을 피할 수 있는가?

이러한 질문들은 이동로봇제어기의 구조적 설계를 위한 중요한 동기가 된다. 첫번째 물음은 이동로봇이 가능한 많은 센서정보들을 융합처리(Sensor fusion)하여 로봇의 현재상태에 관한 정확한 정보를 얻어야 함을 의미한다. 이러한 데이터의 중요성 때문에 이동로봇의 제어시스템을 구성하는데 있어 위치센서부의 구성과 그 처리알고리즘이 중요한 모듈이 된다. 두번째와 세번째 질문들은 경로계획기(Path planner)와 실시간주행제어기(Navigator) 필요성을 내포한다. 이동로봇의 제어시스템은 이러한 문제들의 해결을 기초하여 구성되며, 각부분에 대해 지금까지의 연구에서 LCAR이동로봇에 대해 실현되었던 연구결과들을 중심으로 간략히 서술한다.

### 3.1 위치 추정시스템

이 시스템은 될 수 있는 한 정확한 로봇위치추정을 위해 각종 센서들의 데이터들을 이용한다. LCAR의 위치센서들로는 휠의 회전량을 측정하는 구동휠축에 설치된 광학식 엔코더와 로봇의 절대위치를 추정하는 비전시스템들이다. 먼저 로봇은 2차원 평면 상에서 주행한다고 가정하여, 이 평면상에서 고정된 좌표계(또는 절대좌표계)를 정의한다. 그러면 이 절대 좌표계에 대해 이동로봇의 위치는 로봇의 회전중심점( $C_R$ )의 ( $X, Y$ ) 좌표값으로 표현되고, 방향은 로봇의 길이방향의 축과 X축 사이의 각(일명 heading angle)인  $\theta$ 로 표현된다. 이를 벡터로 표현한 것을 자세(posture)라 하며 이를 P로 정의한다.

$$P=(x, y, \theta)^T \quad (1)$$

#### 3.1.1 Dead Reckoning에 의한 위치추정

"Dead reckoning"이란 휠에 부착된 Odometry(엔코더 및 타코미터)의 정보를 이용하여, 이동로봇의 자세(posture)를 실시간으로 추정하는 방법이다. 먼저 로봇의 움직임을 곡선으로 표현할 때, 매 제어샘플마다 곡선의 곡률( $\gamma$ )과 진행거리의 증분( $dS$ )을 아래식과 같이 좌우구동휠의 회전량 ( $d\theta_R, d\theta_L$ )으로부터 계산할 수 있다.

$$dS=r_r(d\theta_R + d\theta_L)/2 \quad (2.a)$$

$$\gamma=2(d\theta_R - d\theta_L)/(d\theta_R + d\theta_L)/d_r \quad (2.b)$$

여기서  $r_r$ 는 휠의 반경이고,  $d_r$ 는 휠간의 거리이다. 일단 곡률( $\gamma$ )과 진행거리의 증분( $dS$ )이 주어지면, 절대좌표계(World coordinate)에서의 로봇의 위치 및 방향의 변화량을 계산할 수 있다. 즉 샘플링구간에서 로봇은  $r$ 의 회전반경을 갖고  $d\theta$ 의 회전각도만큼 원호운동을 한다고 가정한다. 이때 회전반경( $r$ )과 회전각도( $d\theta$ )는 아래의 관계식으로 부터 구해될 수 있다.

$$d\theta = \gamma dS \quad (3.a)$$

$$r = 1/\gamma \quad (3.b)$$

따라서 원호계산식과 좌표변환을 이용하여, 절대좌표계에서의 자세변화( $dx, dy, d\theta$ )는 아래의 최종식으로 부터 구해진다.

( $d\theta \neq 0$  일때): 원호운동

$$dx = dS [\sin(\theta+d\theta)-\sin(\theta)]/d\theta \quad (4.a)$$

$$dy = -dS [\cos(\theta+d\theta)-\cos(\theta)]/d\theta \quad (4.b)$$

$$d\theta = \gamma dS \quad (4.c)$$

만약 매우작은 간격의 샘플링구간을 갖거나 회전각도  $d\theta$ 가 0에 가까울 경우 직선운동으로 가정하여 구할 수 있다.

( $d\theta = 0$  일때): 직선운동

$$dx = dS \cos(\theta) \quad (5.a)$$

$$dy = dS \sin(\theta) \quad (5.b)$$

$$d\theta = 0 \quad (5.c)$$

이때 이 계산법의 반복주기는 최대허용곡률과 그곡률에 대한 최고속도의 값에 비례하여 결정되어야 이 방식에 의한 위치추정의 정확도를 주어진 허용범위내로 얻을 수 있다. 즉 경로의 모양이 복잡(sharp curve)하면, 좀더 정확하게 경로를 근사화하기 위해서는 될 수 있는 한 많은 구간의 원호 또는 직선 세그먼트(Segments)들이 필요해진다. 이방식은 비교적 간단한 센서의 이용과 빠른주기로 로봇의 위치정보를 얻을 수 있는 장점을 갖는 반면, 오차가 거리에 따라 계속 누적되고, 실제 몸체의 운동정보가 아닌 휠의 운동정보에 의존하므로 휠의 변형, 바닥면의 굴곡, 바닥면과 휠간의 미끄러짐등의 불확실성(Uncertainty)의 요인들로 인해 오차가 발생하는 단점을 지니고 있다. 그러나 이방법은 비교적 짧은 거리에 대해서는 좋은 위치추정 성능을 갖고, 휠서어보제에 사용되는 엔코더를 이용하므로 추가의 하드웨어가 필요없는 장점이 있어 위치추정시스템의 중요한 요소가 된다.

#### 3.1.2 비전시스템을 이용한 절대 위치 추정

이는 특수하게 제작된 마크들을 작업환경내의 미리 지정된 위

치에 설치하여, 이동로봇이 주행중에 카메라를 통해 얻은 마크의 영상데이터를 이용하여 로봇의 현위치를 추정하는 방법이다. 이때 문제가 되는 것은 카메라와 마크간에 위치적인 관계를 찾는 것이다. 지금까지 어떤 마크형태가 로봇의 위치를 얻기에 적합한 것인가에 대한 많은 연구가 있어 왔다. 예를 들어 Kabuka와 Arenas[6]등은 원형의 마크를 이용하여, 로봇이 마크로부터 떨어진 거리와 방향을 구하는 데 영상평면상의 타원정보를 사용하였다. 또한 실시간 처리를 위해 알고리즘을 하드웨어 회로로 구현하여 좋은 결과들을 보여준 바 있다. T.Tamura등[7]은 사각형태의 마크를 쥘렌즈를 사용하여 영상을 최대한 크게 한 후, 최소자승법(Least square method)을 사용하여 로봇의 위치를 결정하는 방법을 제시하였다. 그러나 이방법은 많은 계산량 때문에 실시간으로 처리될 수 없는 단점을 지니고 있다. 최근 J.H.Kim등은[8] 스트라이프패턴을 마크로 이용하고, 라인 스캐닝 카메라를 통해 마크의 영상데이터를 얻는 방법을 개발하였는데, 이는 영상데이터를 얻는데 걸리는 시간을 최소화하고, 위치관계식을 확정형(Explicit form)으로 얻을 수 있는 것이 특징이다. 이러한 접근들의 공통된 특징은 어떻게 하면 좀더 실시간으로 처리할 수 있는 알고리즘의 개발과 그에 따르는 마크의 디자인이다. 통상 로봇의 위치정보는 이동로봇이 주행중에 얻어야 하므로, 만약 아무리 정확한 위치정보를 얻을 수 있는 방법이 있더라도 이를 실시간으로 처리하지 못한다면 그방법은 실현성이 없게 된다. 최근 연구[9]에서 우리는 이러한 문제에 초점을 맞추어, 좀더 실시간으로 처리할 수 있는 방법을 시도하였다. 그림 4와 같이 사각형태의 마크를이용하고, 이 마크의 영상과 로봇의 자세의 상관관계를 그림 5와 같이 신경망모델로 구현한다. 그리고 이 신경망모델은 로봇을 마크주변의 기지의 포인트에 위치시켜 얻은 Training samples들을 사용하여 off-line 학습을 통해 구한다. 제안된 방법의 효율성을 알아보기 위해 마크에 대해 평행하게 설계된 5종류의 직선경로에 대해 실제 주행

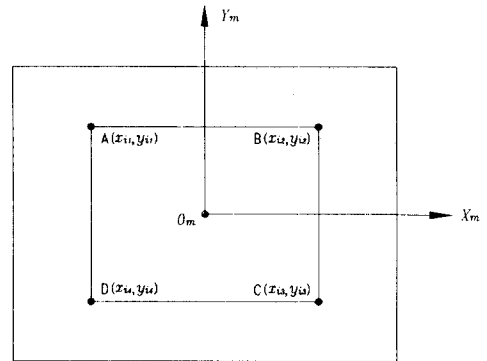


Fig.4 Figure of the used landmark

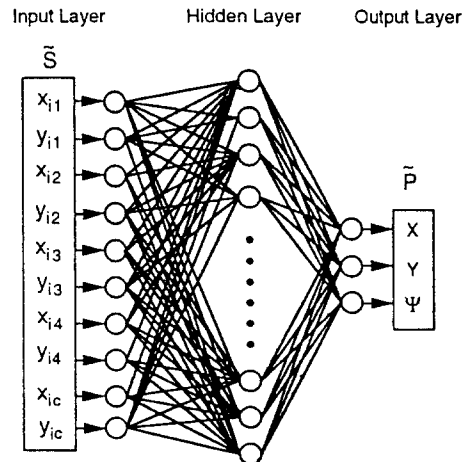


Fig.5 Neural net model for position estimation

실험을 수행하였다. 그림. 6은그 실험결과이며, 추정오차가 마크로부터 멀리 떨어진 지점에서 더욱커짐을 보여 준다. 이 방법은 카메라모델링 위치계산을 위한 수학적모델이나, 복잡한 카메라 Calibration과정등이 필요하지 않고, 향후 신경망계산기의 출현을 기대해 볼 때 병렬처리에 의한 실시간 처리를 할 수 있는 장점이 있다. 현재 이방법의 문제점과 성능개선을 위한 연구가 수행중이다.

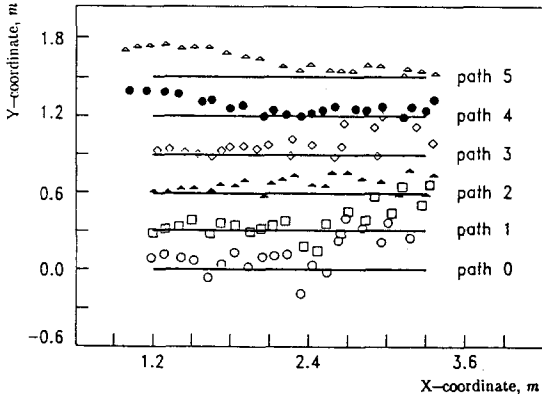


Fig.6 Experimental results of the proposed position estimation method

### 3.2 경로계획

이동로봇을 위한 경로계획은 그 경로가 주위환경에 대해 결정되는 전체경로계획(Global path planning)과 각 경로를 구간으로 나누어 수학적으로 표현하는 국부경로계획(Local path planning)으로 나눌 수 있다. 전체경로계획은 출발점과 최종목적지가 결정되면, 주위환경(예:복도, 문, 시설물등)에 관한 사전정보를 미리 이용하여 수행된다. 이전의 연구에서 H.R Beom 등[10]은 전체경로계획의 한 방법으로, 환경을 임의의 형상을 한 장애물들로 구성된 2차원 공간으로 가정하고, 환경에 대한 정보를 카메라로 부터 얻은 영상을 필터링과 스레쉬홀딩(Thresholding)을 거쳐 이진화하여 표현하는 방법을 제시한바 있다. 이 방법은 Thining을 통해 빈공간의 골격을 구하고, 이를 토대로 최단 경로를 찾는 방법으로, 특히 복잡한 장애물이 놓여 있는 환경에 대해 효율 적이다. 국부경로계획은 실제 이동로봇의 동적특성과 기구학적 한계를 고려하여 이동로봇이 추종 가능하도록 각 경로의 구간을 기하학적 식으로 표현하는 것이다. 이동로봇에 있어 부드러운 가속속이나 부드러운 회전은 미끄러짐과 과도한 충격을 피하는 데 필수적이다. 따라서 경로계획에 있어 이러한 부드러운 경로를 만들기 위해 곡률연속곡선(예:clothoid, spiral curve등)에 관한 연구결과[11,12]들이 발표된 바 있다. 그러나 대부분 이러한 복잡한 곡선에 의한 경로 설계는 곡선의 explicit한 해가 존재하지 않거나, 복잡한 계산을 필요로하여, 실시간 국부경로계획이 곤란한 단점이 있다. 따라서 본 연구에서는 이 문제를 해결 하기 위해 최근 소개한 바 있는 랜딩커브를 이용한 경로제어 알고리즘[13]을 적용한다. 이 방법에서는 경로를 단순히 직선경로로 표현되는 세그먼트들의 조합으로 설계하고, 인접한 세그먼트간에 어떤 제한조건(위치, 방향, 곡률등의 연속조건)이 필요하지 않는 간단한 경로계획 방법을 사용한다. 따라서 국부경로계획부가 충돌회피알고리즘과 같은 실시간 처리의 기능을 담당할 때, 최소한 계산부담을 덜어 줌으로써 보다 효율적인 제어시스템을 기대할 수 있다.

### 3.3 경로 추종 제어

이동로봇이 주어진경로를 따라갈때, 이 경로를 벗어나지 않도록 로봇의 선속도( $v_d$ )및 회전속도( $\omega_d$ )를 매 제어 샘플순간마다 결정하는 것을 경로제어라 한다. 이러한 경로제어의 한 방법으로 곡률연속적인 부드러운 움직임을 보장하는 랜딩커브를 이용한 경로제어방법[13]을 알아 본다. 사용되는 정보는 기준경로의 자세( $P_r$ )및 속도( $V_r, \omega_r$ )와 이동로봇의 현재자세( $P_c$ )및 속도( $V_c, \omega_c$ )이고 이를 이용하여 다음 제어샘플순간의 자세및 속도를 결정한다. 경로제어기는 선속도제어기와 회전속도제어기로 별개로 구성되고 이를 간략히 소개하면 다음과 같다.

이)고 이를 이용하여 다음 제어샘플순간의 자세및 속도를 결정한다. 경로제어기는 선속도제어기와 회전속도제어기로 별개로 구성되고 이를 간략히 소개하면 다음과 같다.

#### 3.3.1 최소시간제어기 (BANG-BANG control)

경로제어알고리즘의 설명에 앞서 이 제어의 기본을 이루고 있는 최소시간 BANG-BANG제어에 대해 설명한다. 우선 어떤 직선운동만을 하는 이동로봇이 주어지고, 그 속도( $v_c$ )및 가속도( $a_c$ )의 최대값이 아래와 같이 제한된다고 가정한다.

$$\begin{cases} |v| \leq v_{max} \\ |a| \leq a_{max} \end{cases} \quad (6.a) \quad (6.b)$$

여기서  $v_{max}$ 는 최대허용속도이고  $a_{max}$ 는 최대허용가속도이다. 그리고 초기에 이 이동체의 경로상에서 속도  $v_r$ 로 움직이는 가상의 목표점이 주어질 때, 이동로봇이 식(6)의 제한조건을 만족하면서 최소시간내에 현재의 위치및 속도( $x_c, v_c$ )로부터 목표점의 위치및 속도( $x_r, v_r$ )에 도달하기 위한 로봇의 가속도  $a_c$  를 구하는 문제를 고려한다. 먼저 이동체의 목표점의 위치및 속도( $x_r, v_r$ )에 대해, 현재 로봇의 위치  $x_c$ 에서 식(6.b)의 최대가속도로 감속했을 때 목표점에 도달할 수 있는 속도를  $v_s$ 라 할 때를 아래의 식으로 계산되어진다.

$$v_s = v_r + [2a_{max}|x_r - x_c|]^{1/2} \cdot \text{SGN}(x_r - x_c) \quad (7)$$

그리고 최대허용 속도조건 식(6.a)에 의해 그 크기는 아래와 같이 제한된다.

$$v_s = \mu(v_s, v_{max}) \quad (8)$$

여기서  $\mu(a, b)$ 는 a의 절대값이 b로 제한되는 Clamping함수이다. 이렇게  $v_s$ 가 구해지면, 샘플링주기를 dt라 할 때, 가속도( $a_c$ )는 다음과 같이 결정할 수 있다.

$$a_c = \mu[(v_s - v_c)/dt, a_{max}] \quad (9)$$

이러한 1차원 운동에 대한 운동체의 제어방법은 식(6)의 제한조건을 만족하며 최소시간 내에 목표점에 도달할 수 있다. 이때 식(6)의 제한 조건이 그 이동체가 얼마나 부드러운 가속속 운동을 하는 가를 결정하게 된다. 이를 직진과 회전의 2자유도를 갖는 이동로봇의 경로제어기에 확대적용한다.

#### 3.3.2 선속도 제어알고리즘

이동로봇의 선속도제어는 앞절의 최소시간방법을 그대로 적용하여 구현된다. 기준경로상에서의 목표점의 자세를  $P_r$ 로 정의하고, 이동로봇의 현재의 자세를  $P_c$ 로 정의할 때, 기준경로의 자세( $P_r$ )를 기준으로한 로봇의 자세오차( $P_{er}$ )를 다음식으로 구할 수 있다.

$$P_{er} = (x_{er}, y_{er}, \theta_{er})^T = T_{er}(P_r - P_c) \quad (10)$$

$$\text{여기서 } T_{er} = \begin{bmatrix} \cos(\theta_r) & \sin(\theta_r) & 0 \\ -\sin(\theta_r) & \cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ 이다.}$$

그러면  $x_{er}$ 은 목표점과 로봇의 현재위치를 직선경로의 수직방향으로 투사시킨 점간의 거리가 된다. 그리고 목표점의 기준속도는  $v_r$ 로 주어진다. 이와 같이 목표점까지의 거리( $x_{er}$ )및 목표점의 기준속도( $v_r$ )가 구해지면 앞절에서, 유도한 최소시간 제어방법을 그대로 적용할 수있고 이를 정리하면 다음과 같다.

$$v_s = v_r + [2a_{max}|x_{er}|]^{1/2} \cdot \text{SGN}(x_{er}) \quad (11.a)$$

$$v_s = \mu(v_s, a_{max}) \quad (11.b)$$

$$dv_c/dt = \mu[(v_s - v_c)/dt, a_{max}] \quad (11.c)$$

$$v_d = v_c + dv_c \quad (11.d)$$

이때 만약 기준경로가 n개의 직선구간으로 설계되고 각구간마다 서로 다른 기준속도( $v_{ri}, i=1, 2, \dots, n$ )를 설정하더라도 식(11)의 선속도제어법에 의해, 실제주행시 이동로봇은 식(6)의 제한조건을 만족하며 부드러운 가속속운동을 얻을 수 있다.

### 3.3.3 회전속도제어알고리즘

이번에는 이동로봇이 주어진 직선경로에 대해  $\theta_{er}$ 의 방향오차를 가지고 수직방향으로  $y_{er}$ 만큼 떨어진 거리에 위치할 때, 이 직선경로에 복귀하기 위한 이동로봇의 회전속도  $\omega_d$ 를 구하는 문제를 생각한다. 이 때 단지 회전속도의 제어만으로 방향오차( $\theta_{er}$ )와 거리오차( $y_{er}$ )를 모두 줄여야 하므로 로봇의 현재점에서 취해야 할 기준방향은 이 두 오차들을 모두 고려해 결정해야 하는 것이 앞서의 선속도제어와 다른 점이다. 이를 위한 하나의 접근으로 보통 곡부경로계획에서 사용하는 곡률연속곡선을 이용하는 방법을 제시한다. 먼저 기준경로상에 경로진행방향을 X축으로 하는 X-Y 좌표계를 설정한다. 그리고 이 좌표계의 원점은 이 직선경로상의 임의의 점에 놓고 직선경로를 기준하여 좌우에서 접하는 x에 대한 3차 Polynomials로 커브  $y_L(x)$ 를 아래의 조건들을 만족하도록 구한다.

- i)  $y_L(0) = 0$  (12.a)
- ii)  $dy_L(0)/dx = 0$  (12.b)
- iii)  $d^2y_L(0)/dx^2 = 0$  (12.c)
- iv)  $d^3y_L(0)/dx^3 = c_x$  (12.d)
- v)  $|x(x)| \leq x_{max}$  (12.e)

여기서  $x(x)$ 는 곡률함수로 X-Y좌표상의 커브의 곡률은 아래의 관계식을 갖는다.

$$x(x) = (d^2y/dx^2) / [1 + (dy/dx)^2]^{3/2} \quad (13)$$

여기서 i)-iii)은 곡선이 곡률의 연속성을 만족하면서 직선경로에 접하는 조건이고, vi)는 곡선의  $c_x$ 를 정하기 위한 조건으로 이 곡선의 최대곡률의 제한조건이다. 따라서 위의 조건들을 만족하는 커브식은 아래와 같다.

$$y_L(x) = c_x x^3 \quad (14)$$

이 곡선은 현재의 로봇지점에서 부드럽게 주어진 직선경로에 수렴할 수 있는 중간경로가 되며, 이를 랜딩커브라 정의한다. 따라서 이동로봇이 직선경로의 수직방향으로 임의의 거리( $y_{er}$ )에 위치할 때, 로봇의 중심을 통과하는 랜딩커브가 유일하게 존재하며, 이점에서의 랜딩커브의 접선과 로봇의 진행방향이 이루는 각이 로봇이 회전해야 할 각도로 간주된다. 이를 랜딩각도( $\theta_L$ )라 정의하며 아래의 식으로 표현된다.

$$\theta_L = \text{TAN}^{-1}[S_L] \quad (15)$$

$$S_L = c_x(y_{er}/c_x)^{2/3} \text{SGN}(y_{er}) \quad (16)$$

이 랜딩각도는 이동로봇이 선속도  $v_c$ 의 속도로 움직이고 있을 때 시간에 따라 변하게 된다. 따라서 랜딩각도의 변화율을 식(15)를 시간 t에 대해 미분하여 아래와 같이 구할 수 있다.

$$\dot{\theta}_L = (S_L) / (1 + S_L^2) \quad (17)$$

$$\dot{S}_L = 2c_x (|y_{er}|/c_x)^{-1/3} y_{er} \text{SGN}(y_{er}) \quad (18)$$

여기서  $y_{er}$ 는 이동로봇이 기준경로에 수직방향으로 접근하는 속도로써, 아래식으로부터 구할 수 있다.

$$\dot{y}_{er} = v_c \text{SIN}(\theta_{er}) \quad (19)$$

지금까지 구한 랜딩각 및 그 변화율로부터 이동로봇이 취해야 할 기준방향각도( $\theta_{rs}$ )와 그 기준각속도( $\omega_{rs}$ )를 아래와 같이 구할 수 있다.

$$\theta_{rs} = \theta_L + \theta_r \quad (20.a)$$

$$\omega_{rs} = \dot{\theta}_L + \dot{\theta}_r \quad (20.b)$$

따라서 조향방향각( $\theta_{rs}$ )과 기준각속도( $\omega_{rs}$ )가 구해지면, 앞절에서 소개한 최소시간 제어법을 적용할 수 있다. 이때 최대허

용 각속도( $\omega_{max}$ )와 최대 허용 각가속도( $a_{max}$ )의 조건은 로봇의 동적한계를 고려하여 결정하며, 최종적인 회전속도 제어알고리즘의 형태는 다음과 같다.

$$\omega_s' = \omega_{rs} + [2a_{max}|\theta_{rs} - \theta_c|]^{1/2} \text{SGN}(\theta_{rs} - \theta_c) \quad (21.a)$$

$$\omega_s = \mu(\omega_s, \omega_{max}) \quad (21.b)$$

$$d\omega_c/dt = \mu[(\omega_c - \omega_c)/\delta T, a_{max}] \quad (21.c)$$

$$\omega_d = \omega_c + [d\omega_c/dt]\delta T \quad (21.d)$$

여기서  $\omega_d$ 는 다음 제어샘플 순간에서의 이동로봇이 취해야 할 회전속도명령으로 휠서보제어기의 기준입력이 된다.

### 3.4 시스템구성(System integration)

지금까지 설명한 경로제어방법을 이용하여 이동로봇의 주행제어시스템을 다음과 같이 구성하였다.

#### (i) 이동로봇의 위치 및 속도 추정

일단 시작센서 영역밖의 경우 휠센서(엔코더 및 타코미터)정보를 가지고 Dead Reckoning을 통해 이동로봇의 현재의 자세( $x_c, y_c, \theta_c$ ) 및 속도( $v_c, \omega_c$ )를 계산하고, 시각영역내에서는 마크의 영상정보를 이용하여, 로봇의 자세를 추정한다. 이때 속도는 휠엔코더의 정보로부터 구한다. 이때 Dead Reckoning에서는 3.1.1절에서 제안된 식(4)와 식(5)를 이용한다.

#### (ii) 경로오차의 계산

식(10)을 이용하여 기준경로에 기준한 상대오차( $P_{er}$ )를 계산한다.

#### (iii) 경로추종제어

앞절에서 설명한바와 같이, 일단 상대위치오차( $P_{er}$ ) 및 기준경로에서의 속도( $v_r, \omega_r$ )가 주어지면, 식(11)과 식(21)의 경로추종제어 알고리즘을 수행하여 다음 제어샘플순간에서 이동로봇이 취해야 할 속도( $v_d, \omega_d$ )를 계산한다.

#### (iv) 로봇기준좌표계의 변환

앞의 단계 iii)에서 구한 경로제어기의 출력 즉  $v_d$  및  $\omega_d$ 를 Two-wheel Drive 방식의 이동로봇에 대한 역기구학적 해석을 통해 각 구동휠에 대한 기준속도로 변환한다. 이 변환식을 구하면 다음과 같다.

$$\omega_{an} = a/r_w * [v_d + d_w \omega_d/2] \quad (22.a)$$

$$\omega_{al} = l/r_w * [v_d - d_w \omega_d/2] \quad (22.b)$$

#### (v) 휠명령치의 계산

각 구동모터에 대한 기준속도는 샘플링 주기 dt를 고려하여 다음의 계산식을 통해 휠의 펄스수( $p_{ar}, p_{al}$ )로 변환된다.

$$p_{ai} = 2000(\omega_{ai} dt) / (2\pi), \quad \text{여기서 } i=R, L \text{ 이다.} \quad (23)$$

이 값들은 하부의 휠제어보드로 전달되어, 위치제어루프의 기준 입력이 된다.

이러한 (i)에서 (v)의 5단계는 매 제어샘플마다 반복되며, 그림.7은 이러한 제어알고리즘을 이용한 컴퓨터 모의실험 결과를 보여준다. 이 결과를 통해 알 수 있듯이, 제안된 경로제어방법은 이동로봇이 경로로부터 이탈된 위치에서 부드러운 운동을 하며 경로에 수렴하는 성능을 갖는다는 것을 확인할 수 있다.

### 4. 결론 및 향후과제

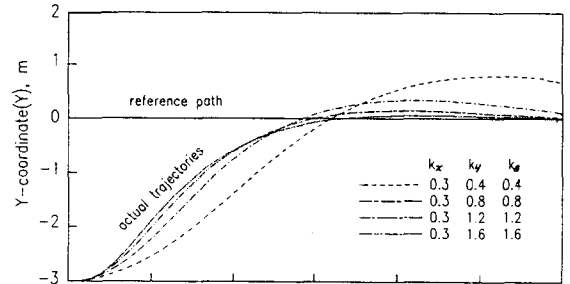
본 논문에서는 Proto-type 이동로봇 LCAR에 대한 제어시스템의 하드웨어적 구조와 각 하드웨어모듈에 대한 설계배경에 대해 전반적으로 서술하였다. 설계된 LCAR-90로봇의 제어시스템의 특징으로는 PC를 중심으로 비전시스템 및 휠제어부동 각모듈들이 연결구성되고, 휠제어부 및 초음파프로세서 그리고 카메라마운트

의 제어부를 각각의 마이크로프로세서를 사용하여 독립적 모듈로 설계함으로써, 계층적 구조가 되도록 하였고, 제어알고리즘의 실시간 연산을 위해 주제어컴퓨터인 PC에 DSP보드를 추가한 것이다. 또한 자유주행로봇의 제어시스템을 구성함에 있어 필수적인 요소들인 위치추정시스템, 경로계획기 그리고 경로추종제어기를 실시간처리의 개념에서 개발하였으며, 적용된 각각의 알고리즘들의 특징과 성능에 대해 간단히 서술하였다. 앞으로 이렇게 구성된 제어시스템의 각 소프트웨어모듈들을 계층적 구조로 종합구성하고, 복잡한 환경내에서의 자동 주행실험을 수행하여 봄으로써, 이 때 발생하는 실제적인 문제점들을 보완하고, 좀더 제어시스템이 지능적이 되도록 다양한 센서장치의 개발과 알고리즘의 개선에 관해 연구할 예정이다.

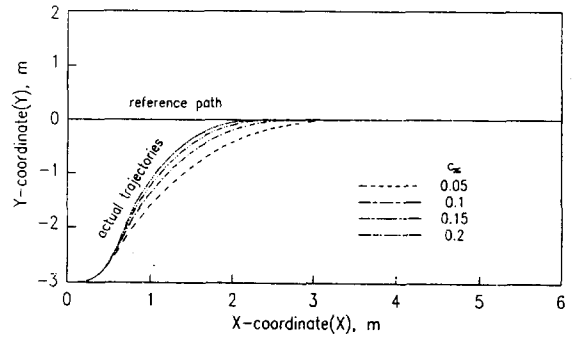
#### 참고문헌

- [1] Omead Amidi, "Integrated Mobile Robot Control", Maser Thesis, Dept. of Electrical and Computer Eng., CMU, 1990
- [2] T. Tsumura, "An Experimental System for AGV, Following the Route Stored in Memory", Proc. of 11th Int. Symp. on Industrial robots, 1981
- [3] Y. Kanayama and S.Y. Yuta, "Vehicle Path Specification by a Sequence of Straight Lines", IEEE J. of Robotics and Automation, Vol.4, No.3, 265-276, 1988
- [4] S. Singh and D.H. Shin, "Position Based Path Tracking for Wheeled Mobile Robots", Proc. of IEEE IROS'89, 1989
- [5] J. Crowley, "Asynchronous Control of Orientation and Displacement in Robotic Vehicle", Proc. of IEEE conf. on Robotics and Automation, 1989
- [6] M.R. Kabuka and A.E. Arenas, "Position Verification of a Mobile Robot Using Standard Pattern", IEEE J. of Robotics and Automation, RA-3, No.6, 505-516, 1987
- [7] T. Tsumura, "A System for Measuring Current Position and/or Heading of Vehicle", Bull. JSME 25, No.203, 821-826, 1982
- [8] J.H. Kim and H.S. Cho, "Real-time Determination of a Mobile Robot's Position by Linear Scanning of a Landmark", Robotica, Vol.10, 309-319, 1992
- [9] J.S. Kim, K.C. Koh and H.S. Cho, "A Neural Net-Based Position Estimation Method for Mobile Robots in Indoor Environments", submitted to ISMCR'92, 1992
- [10] H.R. Beom and H.S. Cho, "Path Planning Method for Mobile Robot", KACC, 1991
- [11] Y. Kanayama and B.I. Hartman, "Smooth Local Path Planning for Autonomous Vehicles", IEEE Int. Conf. on R & A, 1989

- [12] Y. Kanayama, et al., "A Stable Tracking Control Method for an Autonomous Mobile Robot", Proc. IEEE Conf. on R & A, 1990
- [13] K.C. Koh and H.S. Cho, "A Steering Control Method for Wheel-driven Mobile Robot", KACC, 1991



(a) linear path control algorithm



(b) The proposed control algorithm

Fig. 7 Tracking results for a straight line path