자동차형 로봇의 도시 실외에서의 효율적인 위치 추정 및 네비게이션 시스템의 구현

An Efficient Urban Outdoor Localization and Navigation System for Car-like Mobile Robots

윤 건 우*, 김 진 백, 김 병 국
(Gun Woo Yoon¹, Jin Baek Kim¹, and Byung Kook Kim¹)

¹Electrical Engineering, Korea Advanced Institute of Science and Technology

Abstract: An efficient urban outdoor localization and navigation system is proposed for car-like robots. First an accurate outdoor localization method is suggested using line/arc features and 2.5D map matching with LRFs (Laser Range Finders), which can reduce the number of singular cases and increase accuracy. Also, path generation, path tracking, and path modification algorithms are proposed for navigation. All these algorithms are implemented on an electric scooter to construct an autonomous urban outdoor localization and navigation system. Experiments reveal the practicality of the proposed system.

Keywords: urban outdoor, localization, car-like mobile robot, LRF, navigation

1. 서론

자율 이동 로봇의 위치 추정 및 네비게이션(navigation)은 기본적으로면서도 중차대한 문제이다. 이동 로봇은 그 쓰임새와 중요성이 점점 높아지면서 많은 연구가 진행되고 있다. 노인이나 장애인의 편의를 위해 도심 지역에서 목적지까지 안전하게 자동으로 운행하는 무인 주행 로봇, 이 전의 체르노빌 지역 등 원전 사고가 발생한 지역과 같이 사람이 직접 갈 수 없는 곳에서의 탐사 로봇 등이 그러한 예이다. 이와 같은 용도로 사용되는 로봇들의 경우 주어진 이동 경로를 탐색하거나 최종 원하는 목적지까지 자동으로 도착해야 하는 등의 네비게이션 임무 수행에 있어 가장 우선시 되어야 하는 부분은 바로 로봇 자신의 위치 추정(localization)이다. 자신의 위치를 파악하지 못했을 경우 어떠한 경로나 목적지가 주어진다 하더라도 정확한 경로 추종이 되지 않아 목적지까지 이동 할 수 없기 때문이다.

다양한 센서들을 활용한 로봇의 위치 추정에 대한 많은 연구가 진행되어 왔다. 위치 추정을 하는 가장 간단한 방법 중의 하나는 초기 위치 정보가 주어진 상태에서 이동 로봇의 바퀴에 달린 엔코더 데이터를 누적하여 로봇의 위치를 계속적으로 계산해나가는 dead reckoning 이다. dead reckoning을 이용한 지역 위치 추정 방법의 경우 로봇의 현재 위치를 알아내는 과정이 단순하고 계산 과정이 복잡하지 않아 쉽고 빠르게 자신의 위치를 추정할 수 있지만, 엔코더의 미세한 오차 누적, 그리고 로봇 지면과의 미끄러짐 등으로 인하여 장거리를 이동할수록 위치 추정 결과의 신뢰도가 떨어질 수밖에 없

다. 이러한 오차를 최소화 하기 위해 여러 센서들을 통한 오차 보정 방법이 제안되고 있고, 그 중에 레이저레이인지파인더(Laser Range Finder) 센서의 경우 높은 정확도의 데이터, 빠른 주기의 데이터 획득, 간단한 데이터 처리, 광범위한 영역의 스캔 등 많은 장점을 지니고 있어 이를 이용한 많은 연구가 진행되어왔다. 관련 기존 연구들은 다음과 같다.

Zhang은 레이저레이인지파인더로부터 스캔된 환경 정보를 선분 개체들로 특징을 추출하여 선분들만으로 구성된 2D 기반 지도의 제작 방법을 제안하였다. 또한 추출된 선분 개체들을 이용한 대응 관계를 통하여 지역 위치 추정을 하는 방법을 제안하였다[1]. 정직환은 레이저레이인지파인더를 이용하여 실내를 스캔 한 데이터에서 선분 개체를 Split and Merge segmentation 방법을 이용하여 추출해 내고 이를 통해 3D 장애물의 발견 및 위치 추정을 하는 방법을 제안하였다[2]. Takeshi는 2개의 레이저레이인지파인더를 서로 수직 방향을 스캔 할 수 있도록 장착하여 3D 데이터를 획득한 후, 이를 2D grid map으로 투영하고 각 그리드에 높이 정보를 알 수 있게 하여 정확도를 높이는 방법을 제안하였고 particle filter를 이용한 위치 추정을 수행하였다[3]. 손희진은 레이저레이인지파인더를 통해 얻은 스캔 데이터들을 순차적 최소 제곱 알고리즘을 통해 효율적으로 벡터를 추출해 내고 이를 통한 벡터 대응 기법을 기반으로 지역, 전역 위치 추정과 SLAM (Simultaneous Localization And Mapping)을 수행하는 방법을 제안하였다[4]. 배상훈은 레이저레이인지파인더 스캔 데이터로부터 순차적 최소 제곱 알고리즘을 이용하여 선분과 원을 추출하고 선분과 원으로 구성된 2D 지도를 이용한 실외 위치 추정 방법을 제안하였다[5].

이동 로봇의 자율 주행을 위해 위치 추정뿐만 아니라 실외 지도와 이동 경로가 주어져야 한다. 본 논문에서는 실외 지도를 2.5D로 표현하여 2D의 간단함과 3D의 많은 정보를 절충하여 효율적으로 표현하는 방법을 사용하였고[6] 이동

* 책임저자(Corresponding Author)

Manuscript received February 12, 2013 / revised April 29, 2013 / accepted May 30, 2013

윤건우, 김진백, 김병국: 한국과학기술원 전기 및 전자공학과
(gunwooy@kaist.ac.kr/jbkim@rcl.kaist.ac.kr/bkkim@kaist.ac.kr)

※ 본 연구는 지식경제부의 융복합형 로봇전문인력양성사업(NIPA-2012-H1502-12-1002)의 지원에 의하여 연구되었음.

경로는 자동차형 로봇의 회전 제한 각도를 만족하는 선분과 호로 이루어진 송의규가 제안한 경로를 이용하였다[7].

지도와 이동 경로가 주어졌다면 이를 추종하기 위한 경로 추종 알고리즘도 필요하다. 여러 경로 추종 알고리즘 중에서 효율적이면서도 간단한 기하학적 경로 추종(Geometric path tracking)에 대한 연구가 특히 오래 전부터 진행되어 왔다. 이러한 기하학적인 풀이를 통한 조향각을 계산하는 연구는 pure pursuit [8]이라는 알고리즘을 시작으로 여러 가지 변형된 형태의 알고리즘들이 개발되고 있다. pure pursuit 방법은 차량의 현재 위치와 look-ahead distance라는 개념을 이용하여 look-ahead distance 앞의 경로상의 지점까지 가는 데 필요한 조향각을 기하학적으로 구하는 방식이다. 하지만 look-ahead distance를 어느 정도로 정하는지에 따라 많이 휘어진 곡선 경로 등을 제대로 추종하지 않고 가로질러가는 문제가 발생할 수가 있다. 이 후에 pure pursuit 알고리즘을 변형하여 차량의 위치를 포함하여 방향까지 고려한 알고리즘인 vector pursuit [9] 방법이 제안되었다. 이 방법은 경로를 선분들의 연속으로 표현하여, pure pursuit 알고리즘과 같이 look-ahead distance 라는 개념을 사용하되 look-ahead point에 도착할 때의 방향까지 고려하게 된다. 최근의 기하학적인 풀이를 통한 자동차형 로봇의 조향각을 계산하는 방법으로는 송의규가 제안한 2-arc를 이용한 알고리즘이 있다[7]. 2-arc 알고리즘의 경우 경로를 선분과 호의 연속으로 구성하고 경로상에서 일정 거리 이상 떨어졌을 서로 접하는 최소 회전 반경의 2개의 원호를 이용하여 주어진 경로로 복귀한다. 하지만 이러한 방법은 항상 2개의 원호를 통해 주어진 경로로 복귀해야 함으로써 효율적으로 경로에 들어가지 못하고 멀리 떨어진 위치에서 경로 추종 시작 시 경로 추종이 불가능하다는 단점이 있다. 이러한 단점들을 보완하여 본 논문에서는 제안된 실외 위치 추정, 경로 표현 방법을 이용하여 동작하면서도 실시간으로 이동 로봇을 제어할 수 있는 간단하면서도 효율적인 경로 추종 방법을 제안한다. 또한 경로 추종 과정에서 주어진 경로상에 사전에 주어지지 않은 물체 또는 장애물이 포착될 경우 안전하게 목적지까지 이동하기 위해 된 다른 알고리즘과 연동되어 동작할 수 있는 경로 수정 연구도 수행을 하였다.

본 논문의 구성은 다음과 같다. I 장은 서론으로 연구의 배경, 목적 및 범위에 대하여 설명하였다. II 장에서는 실험을 위해 구현된 자동차형 로봇 시스템의 하드웨어와 소프트웨어에 대하여 설명하였다. III 장에서는 레이저레인지파인더와 오도메트리를 이용한 실외 위치 추정 알고리즘을 설명하였다. IV 장에서는 경로 생성, 경로 추종 및 수정 등 제안된 네비게이션 알고리즘에 대하여 설명하였다. V 장에서는 시뮬레이션과 실험 결과에 대하여 설명을 하였고 마지막 VI 장에서는 본 연구의 결론과 추후 과제에 대한 기술을 하였다.

II. 자동차형 로봇 시스템

이 장에서는 본 논문에서 구현된 자동차형 로봇 시스템을 하드웨어와 소프트웨어로 나누어 설명한다.

1. 하드웨어

1.1 기구학

본 논문에서 제안된 알고리즘들의 검증에 사용된 로봇은 상용 4바퀴 전동 스쿠터를 바탕으로 제작되었다. 자동

그림 1. 장착된 엔코더들[10].

Fig. 1. Encoders [10].

그림 2. 장착된 레이저레인지파인더 모듈.

Fig. 2. Laser Range Finder module.

주행 시의 조향 제어를 위해 조향 핸들과 앞 바퀴의 기계적 연결을 분리하고 조향 제어를 위해 바퀴 회전 축에 조향 모터 및 엔코더 장착하였다. 수동 모드에서 사용자의 조향 의도 파악은 조향 핸들 축에 장착된 조향 핸들 엔코더로 각도를 읽어들이어 조향 모터 엔코더가 PID제어를 통해 조향 핸들의 각도를 따르도록 하였다. 또한 뒷바퀴의 구동을 통한 dead reckoning을 위해 주행 모터 엔코더가 뒷바퀴에 장착되었다[10].

레이저 센서로써 HOKUYO사의 UTM-30LX를 사용하였다. 레이저레인지파인더의 roll, pitch를 구현하기 위해 다음의 그림과 같이 두 개의 서보 모터를 사용하였다. 서보 모터는 TTL레벨의 신호를 사용하기 때문에 MAX232칩을 이용하여 RS-232신호를 이용하는 임베디드 보드를 통하여 제어할 수 있도록 하였다. 이를 통하여 레이저레인지파인더를 지면으로 비스듬히 기울일 경우 지면과 주변까지 센서 하나로 모두 스캔이 가능하며 센서 하나로도 3차원 데이터를 얻을 수 있다. 이 데이터는 USB를 통하여 노트북으로 전송되어 실외 위치 추정 및 네비게이션 알고리즘을 구현하는데 이용된다.

1.2 기구학 모델

자동차형 로봇의 기구학 모델은 다음과 같다. 그림 3에서 (x, y) 는 자동차형 로봇 뒷바퀴 축의 중심 위치, v 는 자동차형 로봇의 선속도, ψ 는 자동차형 로봇의 조향각, l 은 앞바퀴 축과 뒷바퀴 축 간의 거리이다. 이를 이용하여 자동차형 로봇의 기구학 식을 컴퓨터를 이용하기 위한 이산시간으로 표현하면 식 (1)과 같다.

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_k \cdot \cos(\psi_k) \cdot \cos(\theta_k) \\ v_k \cdot \cos(\psi_k) \cdot \sin(\theta_k) \\ v_k \cdot \sin(\psi_k) / l \end{bmatrix} \quad (1)$$

그림 3. 자동차형 로봇의 기구학.
Fig. 3. Kinematics of car-like mobile robots.

1.3 제어를 위한 임베디드 보드

자동차형 로봇을 제어하기 위한 임베디드 보드로 EZ-X5 보드를 사용하였다. EZ-X5는 Intel PXA255 프로세서[11]를 탑재한 FA리눅스 사의 제품이다.

임베디드 보드는 조향 바퀴 엔코더, 조향 핸들 엔코더, 뒷 바퀴 구동 엔코더, 레이저 레인지 파인더 등 센서 데이터를 읽고 이 값들을 노트북으로 전송하는 등의 역할을 수행하기도 하고, 반대로 노트북으로부터 읽어 들인 조향각이나 속도 값을 읽어 들여 전동스쿠터를 제어하는 역할을 하기도 한다. 이러한 기능들을 수행하기 위해 임베디드 보드 위에 확장보드가 추가되었다. 각 엔코더 신호의 경우 ALTERA사의 EPM7128S를 통해 값을 카운트하도록 하였다. 주행 모터는 기존 설치되어 있는 모터 드라이버를 이용하여 임베디드 보드에서 DAC를 통해 속도를 조절할 수 있도록 하였다. 또한 그 밖의 필요한 I/O를 위해 GPIO를 이용하였다[10].

노트북과 보드간의 통신은 STUART를 이용하여 통신을 하고 레이저 레인지 파인더 위치 제어를 위한 서보 모터와의 통신은 BTUART를 통해 이루어지도록 하였다. 이를 통합한 전체적인 구조는 그림 4과 같다.

2. 소프트웨어

본 연구에서는 시뮬레이션을 위해 Matlab을 이용해 제작한 시뮬레이터와 ROS (Robot Operating System) stage 시뮬레이터 등 두 가지를 이용하였다. 실험을 위해서는 ROS를 이용한 소프트웨어를 제작하였다. 여기서 ROS는 로봇 관련 연구를

그림 4. 하드웨어 전체 구조도.
Fig. 4. Hardware structure.

그림 5. 전체 소프트웨어 구조.

Fig. 5. Software structure.

위한 일종의 framework이자 meta-operating system 으로 다른 연구원들과의 호환성과 제공되는 라이브러리 등의 편리성을 이용하기 위해 사용하였다[12]. 노트북은 제안 된 위치 추정 알고리즘, 경로 추종 알고리즘, 경로 수정 알고리즘과 경로 생성, 지도를 제공하는 부분 등을 각각 하나의 노드 단위로 구성되도록 하였다. 이와 같은 노드들의 연동을 통하여 노트북은 주요 알고리즘들을 실시간으로 계산하여 자동 주행을 위한 조향각과 속도를 임베디드 보드로 전송한다. 각각의 알고리즘들이 하나의 노드들로 구성되어 있어 각 노드를 동작 시키거나 정지시키면서 여러 가지 모드로 동작을 할 수 있게 하였다. 임베디드 보드는 노트북으로부터 전송 받은 조향각과 속도 값을 이용하여 이동 로봇을 10 ms 주기로 제어하고 또한 10 ms마다 읽어들이는 조향 모터 엔코더, 드라이브 모터 엔코더 값 등을 노트북으로 전송하여 dead reckoning이 가능하도록 동작 한다. 이를 전체적으로 표현한 그림은 다음과 같다.

III. 실외 위치 추정 알고리즘

실외에서의 위치 추정을 수행하는데 있어 본 연구에서는 레이저레인지파인더로 주변을 스캔 한 정보를 이용하여 오도메트리를 통한 부정확한 위치로부터 정확한 위치로의 지속적인 업데이트를 통해 오도메트리의 오차가 누적되는 것을 방지하고 정확한 위치를 추정하는, 이전에 본 연구팀이 제안 한 알고리즘을 개선하여 이용하였다[13].

1. 실외 환경의 표현

실외 위치 추정에 앞서 실외 환경을 표현한 지도로써 2.5D의 실외 환경 지도를 사용하였다. 2D 지도의 경우 간단하게 주위 환경을 포함하여 적은 메모리로 빠르게 실외 환경을 표현할 수 가 있지만 실외의 정확한 위치 추정을 하기에 정보가 부족할 가능성이 많다. 3D 지도의 경우 실외 환경을 정확하게 표현할 수가 있지만 이를 위해 많은 양의 메모리가 소요되고 또한 그 환경 정보를 얻어내는 과정이 복잡하게 된다.

본 연구에서는 2D처럼 적은 메모리 소요와 간단하게 환경을 표현할 수 있는 방법과 3D 지도의 높이 정보를 활용할 수 있는 장점을 합쳐 효율적인 2.5D의 실외 환경 지도를 사용하였다[6].

2. 레이저레이저파인더 스캔 데이터의 분류 및 선분/호 개체 추출

제작된 실외 지도와 레이저레이저파인더 스캔 데이터를 비교하기 위해 이를 분류하고 선분/호 개체를 추출해 내는 과정이 필요하다. 본 연구에서 사용한 UTM-30LX의 경우 25ms마다 한번씩 270도 범위를 스캔 한 거리 데이터들을 내 보낸다. 본 연구에서는 이러한 스캔 데이터들을 매 주기마다 선분과 호 개체로 분류하기 위해 양방향 선분 및 호 추출 방법[14]을 사용하였다. 양방향 선분 및 호 추출 방법은 기존의 순차적 최소 제곱 알고리즘[4]을 확장한 방법이다.

한 방향으로 스캔 데이터를 하나씩 검사해 가며 선분 또는 호로 근사를 해나가는 방법이 순차적 최소 제곱 알고리즘이라면 양방향 선분 및 호 추출 방법은 한 방향으로 알고리즘 수행 후 다시 반대방향으로 알고리즘을 수행하면서 더욱 정확하게 선분과 호 개체들을 추출해 낼 수 있게 된다[14].

3. 추출된 개체들과 지도상의 개체들의 대응

3.1 추출된 개체들과 지도 상의 개체들의 표현 법

레이저레이저파인더 스캔 데이터로부터 추출 된 선분 및 호 개체들을 지도 상의 선분 및 호와 비교를 하기 위해 서로 동일한 표현 방법이 필요하다. 각 개체들의 대응은 최종적으로 2.5D 상에서 이루어진다. 이를 위해 먼저 2D 상으로 개체들을 투영시킨다. 그리고 선분은 전역(global) 좌표계 상에서 선분의 양 끝점과 normal form의 ρ, θ 로 표현, 호는 호의 양 끝점과 중심점 그리고 반지름으로 표현, 원은 원의 양 끝점과 중심점 그리고 반지름으로 표현한다[13]. 추출 된 개체들은 센서 좌표계 상의 데이터이기 때문에 이를 전역 좌표계 상으로 변환시키고 이 과정이 수행 되었다면 스캔 데이터로부터 추출된 선분은 지도상의 선분과 대응과정을 거치고, 추출된 호는 지도상의 호 또는 원 개체와 대응과정이 이루어지게 된다.

3.2 선분 대응

전역 좌표계로 변환된 데이터들을 이용하여 다음과 같이

그림 6. 2.5D 대응을 위한 높이 값 비교.

Fig. 6. Checking height for 2.5D matching.

선분 대응을 한다. 먼저 추출된 선분 개체와 지도 선분 개체의 ρ, θ 를 비교하여 그 차이가 일정 이하인지 판단한다. 첫 번째 단계를 통과한 경우 두 선분간의 거리를 구하고 그 거리가 일정 허용 범위 안에 포함되는지 검사한다. 두 번째 단계까지 통과한 경우 지도 선분과 스캔 선분 개체 양 끝점의 높이 값을 비교하여 2.5D 대응을 위한 높이 검사를 한다.

기존의 알고리즘은[5] 추출된 스캔 선분과 지도 개체를 2D 상에서 대응시키기 때문에 건물 벽 등으로부터 추출된 선분이 경우 높이 값이 낮은 둔덕 등에 대응 되는 문제가 발생하였지만 2.5D 대응을 통하여 이를 방지하고 더욱 정확도 높은 위치 추정 결과를 얻을 수 있다.

$$z_{map,low} \leq z_{scan,low}, \quad z_{scan,high} \leq z_{map,high} \quad (2)$$

3.3 호/원 대응

전역 좌표계 상으로 변환 된 스캔 호 개체들은 지도 상의 호/원 개체들과 다음과 같이 대응시킨다. 먼저 스캔 호의 중심점과 지도 상 호/원 중심점의 거리 차가 일정 이하인지 판단한다. 두 번째로 스캔 호의 반지름과 지도 상 호/원의 반지름의 차이가 일정 이하인지 판단한다. 위 두 단계를 통과하였다면 호/원 개체의 대응도 선분 대응의 경우와 같이 지도상 개체의 높이 범위 안에 스캔 개체 양 끝점의 높이가 포함되는지 검사하여 2.5D로 대응이 이루어지게 된다.

이 과정에서 추출된 호 개체와의 반지름과 중심점 위치 차이가 일정 오차 이내인 지도 상의 개체가 원 일 경우 높이 값 비교를 마지막으로 대응이 완료되지만, 지도 상의 개체가 호로 판정될 경우 지도 상의 호의 범위 안에 추출된 호가 포함되는지 추가로 판단한다. 모든 지도 상의 호/원 개체에 대하여 위 과정을 반복하고 최종 대응 된 개체들 중 중심점의 위치 차이가 가장 작은 개체와 대응 시킨다[13].

4. 대응되는 개체들을 통한 위치 추정

4.1 선분 비용함수

스캔 된 선분, 원/호와 지도 상의 선분, 원/호의 대응 관계가 모두 이루어졌다면 이를 바탕으로 선분과 호/원에 대한 비용함수를 각각 정의하여 최적의 위치로 업데이트를 시킨다. 먼저 선분 비용함수는 손회진의 논문으로부터 다음과 같이 정의 된다[4].

ϕ 만큼 회전되고, T 만큼 이동 된 스캔 선분의 양 끝점으로부터 대응 된 지도 상 선분까지의 수직 거리를 v_{k1}, v_{k2} 라고 하고 N_k 는 스캔 선분을 구성하는 데이터의 개수, l_k 는 스캔 선분의 길이라 할 때 다음과 같이 비용함수가 정의 된다[4].

$$\begin{aligned} E_{line}(\Phi, T) &= \sum_{k=1}^N N_k l_k \sum_{j=1}^2 (v_{kj})^2 \\ &= \Phi^T A_{line} \Phi + 2T^T B_{line} \Phi + T^T C_{line} T \\ &\quad - 2U_{line}^T \Phi - 2V_{line}^T T + \tau_{line} \end{aligned} \quad (3)$$

$$\begin{aligned} A_{line} &= \sum_{k=1}^N N_k l_k \sum_{j=1}^2 \Delta_{kj}^T \Theta_k \Theta_k^T \Delta_{kj}, \quad B_{line} = \sum_{k=1}^N N_k l_k \sum_{j=1}^2 \Theta_k \Theta_k^T \Delta_{kj} \\ C_{line} &= 2 \cdot \sum_{k=1}^N N_k l_k \Theta_k \Theta_k^T, \quad U_{line} = \sum_{k=1}^N N_k l_k \sum_{j=1}^2 \rho_k \Delta_{kj}^T \Theta_k \\ V_{line} &= 2 \cdot \sum_{k=1}^N N_k l_k \rho_k \Theta_k, \quad \tau_{line} = 2 \cdot \sum_{k=1}^N N_k l_k \rho_k^2 \end{aligned} \quad (4)$$

4.2 호/원 비용함수

서로 대응 되는 지도상의 호/원 과 스캔 호/원 들에 대하여 ϕ 만큼 회전되고, T 만큼 이동 된 스캔 호/원 개체의 양 끝점 과 중간점으로부터 지도 상 호/원 개체까지의 수직거리를 d_{l1}, d_{l2}, d_{l3} 라 할 때 다음과 같이 호/원 비용함수가 정의된다[13].

$$\begin{aligned} E_{arc}(\Phi, T) &= \sum_{l=1}^N N_l l_i \sum_{m=1}^3 \|d_{lm}\|^2 \\ &= \sum_{l=1}^N N_l l_i \sum_{m=1}^3 ((\Delta_{lm} - R_l \Theta_s) \Phi \\ &\quad + T - c_l)^T ((\Delta_{lm} - R_l \Theta_s) \Phi + T - c_l) \\ &= \Phi^T A_{arc} \Phi + 2T^T B_{arc} \Phi \\ &\quad + T^T C_{arc} T - 2U_{arc}^T \Phi - 2V_{arc}^T T + \tau_{arc} \end{aligned} \quad (5)$$

$$A_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 (\Delta_{lm} - R_l \Theta_s)^T (\Delta_{lm} - R_l \Theta_s) \quad (6)$$

$$B_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 (\Delta_{lm} - R_l \Theta_s), \quad C_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 I$$

$$U_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 (\Delta_{lm} - R_l \Theta_s)^T c_l, \quad V_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 c_l$$

$$\tau_{arc} = \sum_{l=1}^N N_l l_i \sum_{m=1}^3 c_l^T c_l$$

4.3 선분 비용함수와 호/원 비용함수를 더한 전체 비용함수

앞의 과정에서 구한 선분 비용함수와 호/원 비용함수를 모두 이용하여 위치 추정을 할 수 있도록 전체 비용함수를 구성하게 된다. 전체 비용함수가 구해지면 비용함수를 최소화시키는 ϕ 와 T 를 구하여 오차가 비교적 큰 오도메트리를 통한 직전 위치로부터 새로운 위치로 업데이트가 이루어지게 된다[5]. 선분 비용함수와 호/원 비용함수는 모두 동일한 형태의 quadratic form을 지니고 있기 때문에 서로 더하는 것으로 전체 비용함수를 구할 수 있다. 전체 비용함수가 최소화 되도록 하는 조건은 다음과 같다.

$$\frac{\partial E_{tot}}{\partial \Phi} = 0, \quad \frac{\partial E_{tot}}{\partial T} = 0 \quad (7)$$

이와 같은 과정을 통해 ϕ 와 T 가 구해졌다면 다음 식을 통해 오도메트리를 통한 직전 위치 X_k 로부터 보다 레이저 센서를 통한 정확한 위치 X_{k+1} 으로 위치 업데이트를 수행한다[5].

$$X_{k+1} = R_{\phi_k} X_k + T \quad \text{where} \quad R_{\phi_k} = \begin{bmatrix} \cos(\phi_k) & -\sin(\phi_k) & 0 \\ \sin(\phi_k) & \cos(\phi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$T = \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix}$$

IV. 자동차형 로봇의 네비게이션 알고리즘

위치 추정 결과를 바탕으로 목적지까지 이동 로봇이 자율 주행 하기 위해서는 경로의 표현 및 생성방법, 실외 위치 추정 알고리즘과 함께 실시간으로 동작하도록 빠르면서도 제안된 경로 표현을 이용하는 정확한 경로 추종 방법 그리고 경

로를 따라 이동 시 발견되는 장애물의 회피 방법이 필요하다. 본 연구에서는 다음과 같은 경로 추종 알고리즘들을 정립하였다.

1. 경로의 표현 및 생성

자동차형 로봇은 두 바퀴 이동 로봇 또는 omnidirectional 로봇과는 다르게 최소 회전 반경이 존재한다. 이러한 제한 조건을 만족하는 경로의 생성을 위해서 선분과 호로 구성 된 경로를 이용하였다[7]. 하지만 선분과 회전 제한 조건을 만족 하는 호로 경로를 구성하기 위해서는 각 경로 구간의 끝점들이 정확하게 연결되어야 하므로 모든 경유점을 수동적으로 지정하기에는 많은 노력이 필요하게 된다. 이러한 점을 개선 하여 좀 더 편리하면서도 제한 조건을 만족하는 경로 생성을 할 수 있도록 한다. 먼저 아래의 그림과 같이 자동차형 로봇의 회전 제한 조건을 고려하지 않은 상태로 원하는 이동 경로 상의 최소 개수의 Path points들을 입력한다.

다음과 같이 주어진 Path points들을 연결하면 그림 7과 같이 직선으로만 구성된 경로가 형성 된다. 주어진 Path points를 최대한 가까이 지나면서 자동차형 로봇의 회전 제한 각도를 만족시키는 경로 생성을 위해서 그림 8과 같이 자동차형 로봇의 최소 회전 반경을 반지름으로 갖는 직선 구간과 다음 직선 구간에 동시에 접하는 원을 생성한다. 이러한 과정을 통하여 편리하면서도 자동차형 로봇의 회전 제한 조건을 만족시키는 선분과 호로 구성된 경로(via points)를 생성할 수 있다.

2. 벡터를 이용한 경로 추종 알고리즘

자동차형 로봇의 자율 주행 시스템을 구현하기 위해 모든 알고리즘의 동작은 실시간으로 이루어질 필요가 있다. 제안된 실외 위치 추정을 실시간으로 수행하면서도 또한 정확도 높은 경로 추종까지 실시간으로 진행하기 위해 현재 위치에서 경로 추종을 위해 필요한 즉각적인 조향각과 속도를 결정하는 효율적이면서도 정확도 높은 알고리즘을 제안한다.

2.1 직선 구간

먼저 그림 9의 현재 추종하고 있는 경로구간의 시작점 P_0 및 끝 점 P_f 를 로봇 좌표계로 이동시킨다. 그 후 로봇 좌표계 상에서 현재 추종하고 있는 경로 구간의 끝 점 P_f 와 로봇 좌표계 원점을 잇는 벡터 V_{goal} 과 로봇 좌표계 원점으로부터 현재 추종하고 있는 경로까지의 수직 방향을 나타내는 V_{path} 를 구하고 이를 이용하여 로봇이 진행해야 할 방향 벡터인

그림 7. 주어진 경로 점들.

Fig. 7. Path points.

그림 8. 자동차형 로봇이 추종할 수 있는 경로 생성.

Fig. 8. Via points.

그림 9. 직선구간 경로 추종.

Fig. 9. Path tracking for line path segment.

$V_{heading}$ 을 구한다.

$$\begin{aligned} V_{goal} &= (x_{P_f}, y_{P_f}) - (x_c, y_c) \\ V_{path} &= (x_a, y_a) - (x_c, y_c) \\ V_{heading} &= c1 \cdot V_{goal} + c2 \cdot V_{path} \end{aligned} \quad (9)$$

이 때, $c1$ 과 $c2$ 는 는 각 벡터에 가중치를 주는 상수로서 $c1$ 이 클수록 현재 이동 경로의 끝점까지 부드럽게 이동하게 되고 $c2$ 가 클수록 이동 중인 경로에 빠르게 접근한다. $V_{heading}$ 벡터와 로봇 좌표계의 X 축과 이루는 각도를 통하여 이동로봇의 현재 위치에서 필요한 조향각과 속도를 결정한다.

$$\begin{aligned} \phi_k &= \text{Angle between } V_{heading} \text{ and } X_{robot}, \quad |\phi_k| < \phi_{max} \\ v_k &= v_{max} - c \cdot \frac{1}{|\phi_{max}|} \cdot |\phi_k| \end{aligned} \quad (10)$$

2.2 곡선 구간

곡선 구간의 경우는 다음과 같다. 그림 10과 같이 이동로봇 좌표계 원점으로부터 현재 추종하고 있는 곡선구간의 중심점을 잇는 벡터 V_{center} 를 생성한다. V_{center} 벡터가 생성되었다면 이를 현재 이동 중인 방향에 맞도록 회전시켜 현재 추종하고 있는 곡선구간을 접하는 벡터 $V_{tangent}$ 를 생성한다.

$$V_{center} = (x_{center}, y_{center}) - (x_c, y_c) \quad (11)$$

그림 10. 곡선구간 경로 추종.

Fig. 10. Path tracking for arc path segment.

$$V_{tangent} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot V_{center} \quad (\theta = 90^\circ \text{ or } -90^\circ) \quad (12)$$

그 후 직선 구간과 비슷한 방법으로 이동 로봇의 현 위치로부터 이동 중인 곡선구간의 최단 거리의 점을 잇는 V_{path} 와 앞서 구한 $V_{tangent}$ 를 이용하여 이동로봇이 진행해야 할 방향 벡터인 $V_{heading}$ 을 구한다.

$$\begin{aligned} V_{path} &= (x_a, y_a) - (x_c, y_c) \\ V_{heading} &= c1 \cdot V_{tangent} + c2 \cdot V_{path} \end{aligned} \quad (13)$$

이 때, 직선구간과 마찬가지로 $c1$ 과 $c2$ 는 는 각 벡터에 가중치를 주는 상수이고 $V_{heading}$ 벡터와 이동로봇 좌표계의 X 축과 이루는 각도를 통하여 이동로봇의 현재 위치에서 필요한 조향각과 속도를 결정한다.

이 방법은 제안 된 위치 추정 알고리즘과 함께 실시간으로 동작할 수 있도록 간단하면서도 현 위치에서 즉각적으로 필요한 조향각과 속도를 지속적으로 계산하기 때문에 높은 정확도로 경로를 추종한다. 최근에 제안 된 송의규의 2-arc 알고리즘과의 정확성 비교결과는 그림 11과 같다. 또한 2-arc 알고리즘을 이용했을 때 2-arc가 닿지 않는, 경로상에서 멀리 떨어진 위치에서 출발할 경우 최소 회전 반경인 두 개의 원호를 이용하기 때문에 두 개의 원호의 범위를 넘어서는 위치에서는 경로를 추종할 수 없게 되는 경우가 발생하고, 현 위치가 주어진 경로로부터 두 개의 최소 회전 반경 원호 범위 안에 들어오더라도 두 개의 원호 생성과정을 거쳐야 하기 때문에 비교적 비효율적인 경로를 통하여 주어진 경로에 진입하게 되지만, 본 논문에서 제안 된 알고리즘의 경우 벡터를 통한 조향각을 이용하기 때문에 그림 12의 결과와 같이 먼

그림 11. 2-arc 알고리즘과의 정확성 비교.

Fig. 11. Compare tracking accuracy to 2-arc algorithm.

그림 12. 경로에서 멀리 떨어진 위치에서 경로 추종.

Fig. 12. Path tracking from far starting point.

그림 13. 장애물의 원으로의 근사 과정.

Fig. 13. Obstacle closing by a circle.

그림 14. 물체를 둘러싼 원과 이동 중인 경로와의 거리.

Fig. 14. Distance between the closing circle and the given path.

위치에서 출발하더라도 경로를 추종할 수 있고, 더욱 효율적으로 경로에 접해 들어와 추종할 수 있다.

3. 경로 수정 알고리즘

본 논문에서는 자동차형 로봇의 자율 주행 시스템을 위한 정적 장애물 회피까지를 목표로 하였고 이를 위한 경로 수정 알고리즘은 장애물의 발견과 최소 제곱 원 근사를 통한 closing 그리고 이를 이용한 경로의 수정으로 이루어져있다.

3.1 장애물의 발견 및 원을 이용한 closing

먼저 장애물은 레이저 스캔 영역의 일정 범위로 제한한 장애물 발견 영역 안에서 발견되는 개체들 중에 지도 상의 개체와 대응 되지 않으면서 현재 이동중인 경로에 방해를 발생시킬 수 있는 물체로 정의한다. 레이저레이저파인더를 통한 장애물 발견 범위 안에 물체가 발견되었다면 그림 13(b)과 같이 스캔 데이터 점들 사이의 거리 간격 d 가, 이동로봇의 이동가능 폭보다 좁을 경우 하나의 물체로 인식을 한다. 보행자의 두 다리 가 감지되었을 경우 이 범위에 들어가므로 하나의 장애물로 간주하게 되는 장점이 있다.

그 후 최소 제곱 원 근사방법을 이용하여 해당 물체를 구성하는 스캔데이터들을 통하여 원을 생성하고 해당 원의 중심점을 구한다. 생성된 원의 중심점으로부터 물체를 구성하는 스캔 데이터들 중 가장 먼 데이터까지의 거리를 반지름으로 갖는 원을 생성한다. 이와 같은 방법을 통하여 원형이 아닌 물체가 인식되더라도 그림 13(c)와 같이 원 안으로 closing이 가능하게 된다. 물체를 closing하는 원이 생성되었다면 그림 14와 같이 해당 원으로부터 현재 이동중인 경로의 수직거리 d_p 를 구한다. 이는 해당 원의 중심점으로부터 이동중인 경로와의 수직거리를 구하고 원의 반지름을 빼는 방법으로 구할 수 있다. 이 거리 값 d_p 가 이동로봇의 폭보다 작을 경우 해당 물체는 진행중인 경로에 방해하는 물체로 판정되어 장애물로 인식이 된다.

3.2 장애물 회피 경로의 생성

장애물 회피 경로의 생성은 다음과 같은 단계들로 설계하였다. 먼저 발견된 장애물을 회피하기 위한 경로 생성을 위해 먼저 장애물을 둘러싼 원의 반지름에 이동 로봇의 폭 w

그림 15. 선분과 호로 구성 된 수정된 경로.

Fig. 15. Modified path with lines and arcs.

가 더해진 크기의 반지름을 갖는 원 O_1 과 로봇의 현재 위치로부터 최소 회전 반경을 반지름으로 갖는 원 O_2 를 생성한다. 이 때 O_1 의 중심점의 위치가 이동 중인 경로의 왼쪽에 있을 경우와 오른쪽에 있을 경우를 나누어 O_2 의 중심점의 위치가 다르게 결정된다. 그 다음 원 O_1 과 원 O_2 에 동시에 접하는 선분을 생성한다. 두 원에 동시에 접하는 선분은 중내접선 2개와 외접선 2개로 이루어 지는데 내접선의 경우가 최단거리를 형성할 수 있기 때문에 외접선은 고려하지 않는다. 남은 두 개의 내접선 L_1, L_2 중에서 이동 로봇이 진행하는 방향으로부터 원 O_1 과 원 O_2 가 각각 반시계 또는 시계 방향인지 판단하여 최단거리를 구성하는 선분을 구할 수 있다. 그 다음 원 O_2 를 생성할 때와 마찬가지로 현재 경로구간의 끝점 (v_x^{i+1}, v_y^{i+1}) 으로부터 이동로봇의 최소 회전 반경을 반지름으로 갖는 원 O_3 를 생성하고, 원 O_1 과 O_3 원에 동시에 접하는 선분을 생성한다. 동시에 접하는 선분 L_3, L_4 중 최단 경로를 구성하는 선분을 택하는 방법은 이전의 경우와 동일하게 각 원의 생성 방향을 통해 결정된다. 이 과정을 통하여 그림 15와 같이 경로 표현법에 맞고 이동 로봇의 제한 회전 각도를 만족하는 선분과 호로 이루어진 경로가 생성된다.

V. 시뮬레이션 및 실험

본 논문에서 제안된 시스템을 검증하기 위해 연구를 진행하였던 한국과학기술원 전자동과 반도체 동 근처를 2.5D 표현법에 맞추도록 지도를 작성하고 이를 이용하여 시뮬레이션과 실험을 진행하였다[15].

1. 시뮬레이션

그림 16. 주어진 경로.

Fig. 16. Given path.

ROS의 Stage simulator를 이용하여 한국과학기술원 전자동 교수주차장 입구에서 시작하여 전자동 입구 방향으로 진행 되는 ‘ㄱ’자 구간을 경로로 설정하고 장애물이 있는 상황을 가정하여 시뮬레이션을 진행하였다. 주어진 Path points는 순서대로 (60 m, 40 m), (20 m, 40 m), (20 m, 25 m)이고 시작 위치는 (60 m, 40 m)로 주었다.

그림 17에 위치 추정 알고리즘이 동작하지 않을 경우의 결과를 도시하였다. 오도메트리만을 이용한 위치 추정을 통하여 경로 추종과 경로 수정 알고리즘을 수행하면서 부정확한 위치로 인해 장애물을 회피하지 못하고 충돌하는 것을 확인할 수 있다.

다음으로 위치 추정 알고리즘을 수행하면서 자율 주행을 수행하였을 경우의 결과의 전/후반부를 그림 18에 도시하였다. 그림 17의 결과와 달리 레이저레이인지파인더를 통한 정확

그림 17. 위치 추정 알고리즘이 동작하지 않을 경우의 장애물과의 충돌.

Fig. 17. Autonomous navigation without localization algorithm.

그림 18. 위치 추정 알고리즘이 동작할 경우 전반부(좌측)과 후반부(우측).

Fig. 18. Autonomous navigation with the proposed localization algorithm.

그림 19. 수동 주행시 위치 추정 실험.

Fig. 19. Localization experiment.

한 위치 업데이트가 이루어지면서 장애물을 회피하여 주어진 경로를 정확히 추종하고 목적지까지 도달하는 것을 확인할 수 있다.

2. 실험

실외 실험은 한국과학기술원의 전자동과 반도체동 근처에서 진행하였다. 실외 실험은 수동 주행 시의 위치 추정 실험, 실외 위치 추정을 통한 복잡한 경로의 자율 주행 실험, 장애물 회피 실험으로 구성되어 있다.

2.1 수동 주행시의 위치 추정 실험

그림 19의 실험결과와 같이 위치 추정 알고리즘을 통한 결과는 실제 직진 경로와 같이 표시되었지만 오도메트리만을 이용한 위치 추정을 하였을 경우 오차 누적으로 인하여 점점 목적지로부터 멀어지는 것을 알 수 있다. 곡선 구간이 포함될 경우 더욱 큰 차이가 발생한다. 이 역시 위치 추정 알고리즘 수행을 한 경우 레이저레이인지파인더를 통한 정확한 위치 업데이트가 이루어져 최종 목적지에서의 오차가 약 3cm 이하로 복잡한 경로를 이동하면서도 정확한 위치 추정이 이루어짐을 확인하였다.

2.2 자동 주행 실험

실외 위치 추정의 경우 실제 정확한 위치를 추정하기에 어려움이 따르기 때문에 정확한 위치 추정과 자율 주행이 이루어짐을 확인하기 위해 복잡한 경우점을 지나 다시 출발한 지점으로 돌아오는 경로를 설정하였다. Path points는 순서대로 (11.5 m, 22 m), (11.5 m, 15.0 m), (27.0 m, 15.8 m), (27.0 m, 15.8 m), (27.0 m, 12.3 m), (11.5 m, 12.3 m), (11.5 m, 22 m)로 주었다.

실험 결과 그림 20과 같이 오도메트리만을 이용한 경우 실제 로봇의 이동 궤적과는 매우 다르게 측정이 된 반면에 위치 추정 알고리즘을 통해 정확한 위치로 업데이트를 지속적으로 수행한 경우 실제 이동 위치와 거의 같고 그림 20(c)

그림 20. 복잡한 구간 자율 주행 실험.

Fig. 20. Autonomous navigation experiment on a complex path.

그림 21. 장애물 회피 실험.

Fig. 21. Obstacle avoidance experiment.

와 같이 처음 나갔던 경로로 다시 돌아왔을 때 경로와의 오차가 10 cm 이하로 실외 자율 주행이라는 점을 고려하였을 때 매우 정확하게 자율 주행이 수행되었음을 확인할 수 있다.

2.3 장애물 회피 실험

그림 21의 실험 결과와 같이 경로 수정 알고리즘을 통하여 장애물로부터 주어진 경로로부터 새로운 경로를 계획하여 이동 로봇이 지나갈 수 있는 폭을 두고 회피한 후 기존 경로로 다시 복귀하는 것을 확인할 수 있다. 위 실험에서도 dead reckoning 만 이용했을 경우 장애물을 회피하지 못할 뿐만 아니라 목적지와는 다른 방향으로 진행되는 반면 위치 추정 알고리즘이 수행된 경우 장애물 회피 경로 생성 및 추종이 정확하게 이루어짐을 확인할 수 있다.

VI. 결론 및 추후 과제

본 연구에서는 도심지역에서의 자동차형 로봇의 자율 주행을 할 수 있는 시스템을 설계하고 구현하였다. 이를 설계하는 과정에서 2.5D 실외 환경의 표현 및 제작, 레이저레인지파인더 데이터로부터 선분과 원/호를 추출하고 지도와의 2.5D 대응을 통한 실외 위치 추정 알고리즘을 통해 둔턱 등의 스캔 선분이 건물의 벽에 대응되는 문제 등을 해결하여 더욱 정확도 높은 위치 추정이 되도록 하였다. 또한 사용자의 편의를 고려한 선분과 호로 이루어진 경로 생성 알고리즘, 간단하면서도 효율적인 경로 추종 알고리즘, 앞서 제안된 알고리즘과 통합 가능한 정적 장애물을 회피할 수 있는 경로 수정 알고리즘을 제안하였다. 최종적으로는 제안한 모든 알고리즘을 구현 및 통합하여 진행한 시뮬레이션과 실험을 통하여 도심지역에서의 실시간으로 정확한 위치 추정과 자율 주행이 가능하다는 것을 검증하였다.

추후 과제로는 더욱 정확한 위치 추정을 위해 이동 중에 발생한 스캔 데이터의 오차 보정을 통한 더욱 정확도 높은 위치 추정, 동적 및 다수의 장애물의 회피 등을 추가로 진

행할 수 있다.

REFERENCES

- [1] L. Zhang, "Line segment based map building and localization using 2D laser range finder," *International Conference on Robotics & Automation 2000*, vol. 3, pp. 2538-2543, Apr. 2000.
- [2] J. J. Han, "Real-time indoor localization and 3d obstacle detection using laser range finder," M. S. Thesis, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, 2003.
- [3] T. Takahashi, "2D localization of outdoor mobile robots using 3D laser range data," M. S. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2007.
- [4] H. J. Sohn and B. K. Kim, "An efficient localization algorithm based on vector matching for mobile robots using laser range finders," *Journal of Intelligent and Robotic System*, vol. 51, no. 4, pp. 461-488, Nov. 2007.
- [5] S. H. Bae and B. K. Kim, "An efficient outdoor localization method using multi-sensor fusion for car-like robots," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 10, pp. 995-1005, Oct. 2011.
- [6] J. B. Kim, G. W. Yoon, and B. K. Kim, "Development of laser range finder simulator based on 2.5D map for mobile robot localization in urban outdoor," *Proc. of 2012 27th ICROS Annual Conference (in Korean)*, Seoul, Republic of Korea, pp. 198-199, Apr. 2012.
- [7] U. K. Song and B. K. Kim, "Development of a DGPS-based localization and semi-autonomous path following system for electric scooters," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 4, pp. 353-361, Apr. 2011.
- [8] O. Amidi, "Integrated mobile robot control," Tech. Rep., Carnegie Mellon Univ, Robotics Institute, 1990.
- [9] J. Wit, C. D. Crane, and D. Armstrong, "Autonomous ground vehicle path tracking," *Journal of Robotic Systems*, vol. 8, pp. 439-449, 2004.
- [10] W. C. Lim, "Development of a cost-effective safety system with curb detection and avoidance for electric scooter," M.S. Thesis, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, 2010.
- [11] Intel PXA255 Processor Developer Manual, Intel(<http://www.intel.com>), 2003.
- [12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," *International Conference on Robotics and Automation*, ser. Open-Source Software workshop, 2009.
- [13] G. W. Yoon, J. B. Kim, and B. K. Kim, "Outdoor mobile robot localization algorithm using line/arc features based on laser range finders and 2.5D map," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 18, no. 7, pp. 658-663, Jun. 2012.
- [14] B. K. Kim et al., "Method and apparatus for bidirectional line and arc segmentation of scan distance data for localization of mobile robots," KOREA Patent, Application Number 10-2012-0005646, 2012.
- [15] G. W. Yoon, "Implementation of an efficient urban outdoor localization and navigation system for car-like mobile robots," M.S. Thesis, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, 2013.

윤 건 우

2011년 성균관대 전기 및 전자공학과 졸업. 2013년 한국과학기술원 전기 및 전자공학과 석사. 2013년~현재 ETRI 부설연구소 연구원. 관심분야는 임베디드 시스템.

김 진 백

2005년 한양대 전기 및 전자공학과 졸업. 2007년 한국과학기술원 석사. 2007년~현재 동 대학원 박사과정 재학중. 관심분야는 이동 로봇, 제어 소프트웨어 구조, 임베디드 제어.

김 병 국

1975년 서울대 전자공학과 졸업. 1977년 한국과학기술원 전기 및 전자공학과 석사. 1981년 동 대학원 박사. 1981년~1986년 우진계기(주) 연구실장. 1982년~1984년 University of Michigan 방문연구. 1986년~현재 한국과학기술원 전기 및 전자공학과 교수. 관심분야는 실시간 시스템, 로보틱스, 임베디드 제어.