

Tarantula I: 무인 감시 정찰을 위한 자율 주행 로봇

김성환, 홍인표, 김영효, 김영주, 김대영,[†] 이영진
한국과학기술원, (주)Autopower

Tarantula I: An Unmanned Ground Vehicle for Military Surveillance System

Sung-hwan Kim, In Pyo Hong, Young hyo Kim, Young-Joo Kim, Daeyoung Kim, Young jin Lee
KAIST, Autopower Corporation

e-mail: {ssshwan, inpyo, younghyo, yjkim73, kimd}@kaist.ac.kr, ceo@autopower.co.kr

요 약

무인 감시 정찰 시스템은 방재, 감시정찰, 국방감시 등에 응용하기 위하여 꾸준한 연구가 진행되고 있다. 무인 감시 시스템은 감시 범위의 확장 및 능동적인 대처라는 측면에서 로봇을 사용하는데, 이러한 로봇은 일반적으로 로봇에게 부착된 센서를 통해서 주변을 인식하는 구조이므로, 감시 정찰에 대한 인식 범위가 로봇의 주변이라는 지역적 한계를 가지고 있고 무인 감시 정찰 시스템의 요구에 대하여 능동적인 대처가 어렵다. 본 논문에서는 센서 네트워크를 통한 확장적인 인식 정보 획득을 위해 로봇에 부착된 싱크 노드를 통해서 감시지역 내의 센서 노드들과 통신하면서, 그 감시지역을 확장 할 수 있는 무인 자율 주행 로봇을 제안한다. 이 로봇은 무인 센서 노드 자율 설치를 통한 센서 네트워크 구성 시스템을 운영하고, 센서 네트워크로부터 인식 정보를 제공받아 로봇의 인식 범위를 보다 확장 하고, 자율 주행을 통해 시스템의 요구에 능동적으로 무인 감시 시스템에서 사용될 수 있다.

1. 서론

무인 감시 정찰 시스템은 방재, 감시정찰, 국방감시 등에 응용하기 위하여 꾸준한 연구가 진행되고 있다. 이에 더해, 로봇은 공장의 로봇 팔부터 시작하여 감시정찰 시스템의 일 부분으로써 이르기까지 많은 분야에 응용 되고 있다. 이는 사람을 대체할 수 있다는 것에 큰 의의를 두며, 특히 반복된 작업에 대해서 효율을 보인다. 무인 감시 정찰[1], 오지 탐사, 위험 지역 감시 정찰 및 침입자 추적 등 많은 부분에 로봇의 이용은 종전의 고정된 감시 정찰 시스템의 한계를 개선하는 방안에 있어서 효과적인 방법으로 인식되고 있다. 무인 감시 정찰 시스템은 감시 범위의 확장 및 능동적인 대처라는 측면에서 로봇을 사용하는데, 무인 감시 정찰 시스템에서의 로봇은 일반적으로 로봇에게 부착된 센서를 통해서 주변을 인식하는 구조이므로, 감시 정찰에 대한 인식 정보가 로봇의 주변이라는 지역적 한계를 가지고 있고 무인 감시 정찰 시스템의 요구에 따른 능동적인 대처가 어렵다. 이 문제를 해결하기 위해서 본 논문에서는 센서 네트워크를 통한 확장적인 인식 정보 획득을 위해 로봇에

부착된 싱크 노드를 통해서 감시지역 내의 센서 노드들과 통신하면서, 그 감시지역을 확장 할 수 있는 무인 자율 주행 로봇을 제안한다. 이 로봇은 센서 노드들과 통신 함으로써 자신의 센싱 및 감시 범위를 넓히며, 자율 주행 기능을 가지고 외부인 침입 추적 등의 요구의 상황에 대해서 더욱 능동적으로 대체할 수 있는 장점을 가진다. 또한 로봇 내에 구현되어 있는 센서 네트워크 구성 시스템은 센서 노드를 로봇이 미션에 따르거나 자율적으로 설치하며, 로봇의 센싱 및 감시 범위를 늘릴 수 있을 뿐만 아니라, 오지에 센서 노드를 설치하고 감시하며 종전의 사람이 해결 해야 했던 문제를 무인 로봇이 대신하여 해결하고, 이들을 통해 효과적으로 무인 감시 시스템에서 사용 될 수 있다.

본 논문의 구성은 다음과 같다. 2 절에서는 연구의 배경이 된 일반적인 무인 감시 시스템과 무선 센서 네트워크에 대해서 전반적으로 살펴보고 3 절에서는 무인 감시 정찰을 위한 Tarantula I 로봇 시스템, 로봇 인식 범위 확장을 위한 무선 센서 네트워크, 로봇 자율 주행을 위해서 구현된 자율 주행 소프트웨어에 대해서 기술한다. 4 절에서는 성능 검증을 위해서 준비 된 실험 환경과 시나리오 및 시뮬레이션을 통하여 구현된 시스템의 성능과 결과를 검증하며, 끝으로는 내용을 정리하고 보완하고 추가해야 할 부분에 대해서 논의하면서 결론으로 논문을 마무리 한다.

* 본 연구는 2007년도 지식경제부의 특구 보유기술 사업화 개발사업 과제 지원으로 수행되었습니다.

† 본 연구는 2009년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행되었습니다. (ROA-2007-000-10038-0)

† 교신저자: 한국과학기술원 전산학과 부교수

2. 연구 배경

본 절에서는 무인 감시 정찰 시스템에 대한 개요 설명과 무선 센서 네트워크 기반의 무인 감시 정찰 시스템에 대해서 설명한다.

2.1 무인 감시 정찰 시스템 개요

무인 감시 정찰 시스템은 컴퓨팅 시스템 및 여러 센서 및 카메라를 이용하여, 정해진 지역을 감시 정찰 하는 시스템을 말한다. 그러나 현재의 무인 감시 정찰 시스템은 몇 가지의 단점을 가지고 있다. 첫째, 대부분의 시스템에서 사용되는 장비들은 상당히 고가이고 크기가 크며, 많은 전력을 필요로 하는 동시에 관찰할 수 있는 지역이 매우 한정되어 있다. 따라서 이러한 장비들을 가지고, 충분히 넓은 지역을 동시에 감시 할 수 있는 시스템을 개발하기 위해서는 그 설치 및 유지 비용이 매우 크고, 고가의 장비들을 많이 필요로 하게 되는 단점이 있다. 둘째, 시스템 대부분이 배치된 인력에 의해서 운영되어지기 때문에, 시스템의 상시 실시간 모니터링에 대한 한계를 가지고 있다. 또한 카메라가 특정 지역에 고정됨으로써, 건물 혹은 외곽 펜스에 사각 지역이 발생하는 단점을 가지고 있다. 마지막으로, 현재의 무인 감시 정찰 시스템은 대부분이 센싱 능력은 있지만 특정 위치에 고정되고 이동 능력이 없는 단순 센서만을 사용하기 때문에, 사람의 개입 없이는 능동적인 대처가 불가능하다. 예를 들어, 특정한 센서에 외부의 침입이 탐지되었을 시 재빠르고 신속한 대처가 이루어져야 하는데, 실제 사람의 개입 없이 시스템 자체만으로 대처할 수 있는 방법이 현재의 무인 감시 정찰 시스템에는 존재하지 않는다. 이러한 한계들로 인해서, 무인 감시 정찰시스템은 여전히 많은 인력을 필요로 하고 설치 및 유지 비용이 큰데 반해서, 그 효율성과 신뢰성은 충분히 보장되지 않고 있다. 2.2 절에서는 이런 문제를 일부 해결 하기 위한 방법인 무인 감시 정찰을 위한 무선 센서 네트워크에 대해서 소개한다.

2.2 무선 센서 네트워크

무선 센서 네트워크[2,3]는 일반적인 무인 감시 정찰 시스템의 단점을 보완하기 위해, 저전력, 저가 및 실시간 상시 모니터링을 위해서 디자인 되었다. 무선 센서 네트워크는 크기가 작고 저가의 센서 노드를 사용 한다. 이 센서 노드들은 무선으로 통신 하기 때문에 간단하게 설치 할 수 있다는 장점이 있다. 또한 센서 노드의 비용도 저가 이기 때문에 센서 노드에 의한 감시 범위를 비약적으로 확장 할 수 있다는 장점을 가지고 있다. 또한 이렇게 넓은 지역에 설치된 센서 노드가 수집한 정보들을 무선 네트워크를 통해서 실시간으로 싱크 노드가 정보를 수집하여 실시간 모니터링이 가능하다는 장점이 있다.

그러나 여전히 무인 감시 정찰을 하는데 있어서 무선 센서 네트워크는 문제를 가지고 있다. 첫째, 센서 노드 설치를 위해서 사람의 인력이 필요하고

유지 보수 또한 사람이 직접 해야 한다. 두번째, 무선 센서 네트워크는 센서 노드가 고정된 지역에 설치되어 있으므로, 센싱 범위의 확장 문제 및 추적 감시에 대해서 능동적인 대체를 하기가 어렵다. 이러한 단점을 보완하기 위해서 자율 주행이 가능한 로봇이 필요하다. 현재까지의 감시 정찰용 로봇들은 일반적으로 로봇에게 부착된 센서를 통해서 주변을 인식하는 구조이므로, 감시 정찰에 대한 인식 범위가 로봇의 주변이라는 지역적 한계를 가지고 있다. 따라서 지역적 한계를 극복하기 위해서 센서 네트워크와 연동하고 센서 네트워크의 변화에 따른 능동적 대체가 가능한 자율 주행 로봇이 필요하다.

3. 자율 주행 로봇: Tarantula I

본 절에서는 자율 주행 로봇인 Tarantula I 의 구성 요소와 로봇을 제어하는 로봇 제어부를 설명하고, 그리고 로봇의 인식 범위를 확장하기 위한 센서 네트워크 시스템 및 구성과 자율 주행을 위한 소프트웨어 부분에 대해서 설명한다.

3.1 로봇 구성

Tarantula I 은 [그림 1]에서와 같이 기본 프레임과 싱크 노드를 가지고 있다. 로봇은 전방에 모터 및 감속기, 로봇 구동부의 제어를 담당할 하위 제어부와 자율 주행을 담당할 상위 제어부를 가지고 있다.



[그림 1] 로봇 외부 및 싱크 노드 사진

로봇 상판에 자율 주행을 위한 센서들이 설치되어 있는데 Laser Range Finder 2 개는 로봇의 전방에, 카메라 및 GPS 는 로봇의 중앙에 IMU 는 로봇의 후방에 설치하였다. 또한 이 로봇은 센서 노드와 통신 및 정보 수집을 위한 싱크 노드와 외부 모니터링 시스템과의 통신을 위한 유무선 공유기를 가지고 있다. 각각의 센서들은 자율 주행 소프트웨어 구현을 위해 사용 되는데, 위치 인식을 위한 센서들로는 DGPS, IMU, Encoder 가 사용 되었으며, 장애물 인식을 위한 센서들로는 2 개의 Laser RangeFinder 와, Camera 가 사용되었다.

로봇은 2 축의 6 개 바퀴를 가지고 2 개의 모터를 이용하여 움직이며, 조향방식은 스키드 스티어링 방식을 사용한다. 로봇은 도심지역뿐만 아니라 험난한 지형에서도 운행이 가능하도록 설계되었으며, 스키드 스티어링 방식을 사용함으로써 제자리 360° 회전이 가능하도록 설계되었기 때문에 좁은

공간에서도 방향전환이 자유로워 다양한 환경에서 폭넓게 사용될 수 있다.

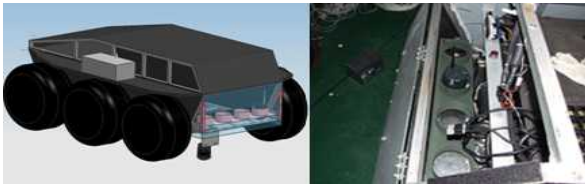
3.2 로봇 제어부

본 절에서는 로봇을 제어 하기 위한 하위 제어부 및 상위 제어부의 구성과 요소에 대해서 설명한다.

3.2.1 하위 제어부

모터 드라이버는 구동제어기와 CAN 통신을 한다. 구동제어기는 DSP 를 사용하였으며, 모터의 전류 제어와 속도 제어를 위하여 PI 제어기를 구현하였다. 로봇의 위치 인식을 위하여 모터와 바퀴에는 절대치 Encoder 가 각각 설치 되었으며, 이는 GPS 와 IMU 와 함께 로봇의 위치를 구하는데 사용된다. 하위 제어기는 상위 제어기와 시리얼 통신을 하며, 상위 컴퓨터로부터의 구동 명령에 대한 기구학 해석을 통해 각각의 바퀴 속도 및 조향 각도를 산출한다. 전체 시스템에 대한 상태 진단 및 비상 정지 기능을 가지고 있으며, 전조등과 방향 지시등 등의 부속 기기를 제어한다.

구현된 로봇에는 [그림 2]에서와 같이 로봇 후방에 센서 네트워크 구성 시스템이 설치되어 있다. RMDF(Route and Mission Data File)에서 주어진 센서 노드를 떨어뜨릴 위치에 로봇이 도달하면, 센서 노드가 탑재되어 있는 패널을 열어서 센서 노드가 아래로 자연스럽게 탈거해 자율 설치 되도록 하였다.



[그림 2] 센서 네트워크 구성 시스템

이때, 센서 노드의 안테나가 위로 향하는 것을 보장하기 위해 센서 노드 밑에 무게 중심 기구를 제작하여 항상 센서 노드의 안테나가 위로 향하게 제작하였다. 떨어진 센서 노드는 곧바로 네트워크에 참여하여 로봇의 인식 범위를 넓히게 된다. 이 시스템 통해 무인 자율 주행 로봇이 센서 노드를 인식 범위 확장에 사용할 수 있도록 하였다. 센서가 설치 되는 위치에 관해서는 현재 RMDF 에 표기 하는 방식뿐만 아니라 로봇이 자율적으로 최적의 센서 설치 위치를 구해 설치할 수 있도록 구현 할 예정이다.

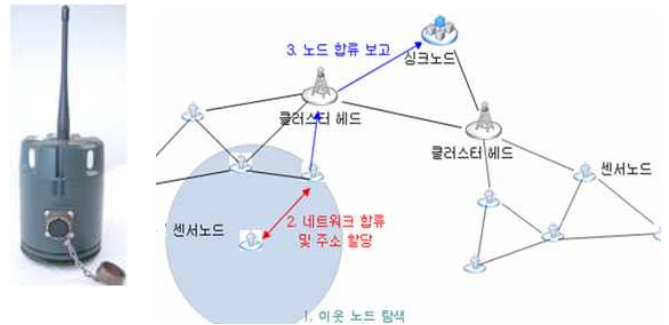
3.2.2 상위 제어부

상위 제어부는 산업용 컴퓨터가 사용되었고, 추가적으로 1 대의 PC-104 규격의 임베디드 컴퓨터가 설치되었다. 산업용 컴퓨터는 상위 제어부에 필요한 모든 데이터들의 통신을 중앙에서 처리해주는 역할을 담당하며, 임베디드 컴퓨터는 주로 센서 인터페이스를 위해서 주로 사용하였다. 모든 컴퓨터는 유무선 공유기를 통해서 내부 및 외부와 통신이 가능하다. 험한 지형에서 로봇이 운행할 시, 그 충격이 크기 때문에 컴퓨팅 시스템을

충격으로부터 보호하는 충격완화장치 위에 컴퓨팅 시스템을 위치하였다.

3.3 무인 감시 정찰을 위한 센서 네트워크

현재 구현된 무인 감시 정찰 시스템은 민간 보안 지역의 무인 감시 시스템을 위한 PIR, 진동, 음향, 마그네틱, 온도, 습도, 카메라, 레이저 센서들을 이용하여 다양한 환경 요인과 그 변화들을 탐지 인식할 수 있도록 하고, 각 센서로부터 수집 및 처리된 정보를 구성된 네트워크를 통해 로봇 과 공유한다. 현재의 네트워크 시스템은 3 계층으로 디자인 되어있는데 일차적으로 센서 노드에서 수집된 자료는 클러스터 헤드 노드로 전달하고, 클러스터 헤드 노드들은 센서 노드로부터 수집한 데이터를 주기적으로 로봇에 있는 싱크 노드에서 모니터링 함으로써 무인 감시 정찰 시스템이 이루어지게 된다. 로봇에 부착되어 있는 싱크 노드로부터 로봇은 센서 네트워크의 정보를 수집 함으로써 로봇의 인식 범위를 확장 시켜 준다. 센서 노드는 저전력을 위한 알고리즘과 운영 주기의 최적화를 이루었으며, 취약한 전장 환경에서도 적용 가능하도록 방수 및 방진 패키징이 되어 있고, 이동성이 보장되도록 소형화할 수 있도록 디자인 하였다.



[그림 3] 센서 노드 및 네트워크 형성 과정

무인 감시를 위한 무선 센서 네트워크는 넓은 지역에 걸쳐 평균 수십에서 수백 개에 이르기까지 많은 노드를 필요로 하기 때문에 이를 효과적으로 설치하고 유지할 수 있는 방법이 필요하다. 즉, 노드가 임의의 감시될 필드에 설치될 경우, 그 설치 과정은 가능한 빠르고 쉬워야 하며, 설치 후에는 가용성을 극대화하여 감시 역할을 수행해야 한다. 따라서, 저전력을 고려한 WOR (Wake On Radio)기반의 센서 전력 운용과, 효율적인 멀티 홉 네트워크는 필수이다. 이를 위하여 WOR 기능과 야외 환경에 적합한 UHF 대역을 지원하는 RF 송수신기를 이용하여, 채널 스캔 기능 및 이웃발견기능을 지원하는 CSMA/CD 기반의 MAC Layer 와 멀티 홉 통신기반의 계층 구조의 Network Layer 를 포함한 저전력 무선 통신 프로토콜을 사용 하였다. [그림 3]에서는 우리가 구현한 센서 노드와 그 센서 노드들의 센서 네트워크 형성 과정을 그림으로 나타낸 것이다.

3.4 소프트웨어 아키텍처 및 자율 주행 알고리즘

본 절에서는 자율 주행 소프트웨어의 구조와 컴포넌트 사이의 데이터 플로우와 무인 자율 주행을 위한 알고리즘 및 그 구현에 대해서 설명한다.

3.4.1 자율 주행 소프트웨어 구조

소프트웨어 프레임워크를 디자인 하는데 있어서, 재사용성, 독립성, 그리고 확장성 등 많은 장점을 가지고 있는 컴포넌트 기반의 개발 방식을 사용하였다. 전체 소프트웨어의 구조는 여러가지 로봇 디자인 방법 중에 컴포넌트 기반의 시스템 디자인에 알맞게 기능이 분화되어있는 Hybrid Reactive/ Deliberative Robot Architecture 인 Three Layered Architecture [4]에 기반을 두고 있다. Kubuntu-8.04 를 운영체제로 사용하였으며, 통신 미들웨어로는 유명한 미들웨어어인 CORBA 를 수정 개선하여 성능의 개선과 사용 인터페이스를 개선한 ZeroC, Inc. 에서 개발한 객체 지향 미들웨어인 ICE (The Internet Communication Engine)[5]을 사용하였다. 로봇 미들웨어는 컴포넌트 기반의 로봇 소프트웨어 프레임워크인 Orca[6], Player/Stage[7], Carmen[8]등의 기반 위에 구현되었다. 기본 컴포넌트 모델은 간단하고 재사용성과 간단한 구현에 초점을 둔 Orca 기반의 컴포넌트 모델을 사용하였으며, 그 위에 자율 주행에 필요한 Perception, Localization, Planning, Navigation 을 담당하는 컴포넌트 기능 아래에 ICE 의 Publish/Subscribe 통신 방법을 이용하여 각 컴포넌트 사이의 통신을 지원하였다. 또한 ICE 로부터 IPC(Inter Process Communication)를 구현하고, 이를 통해 네트워크를 기반으로 IP 와 등록된 프록시(Object reference)를 이용하여

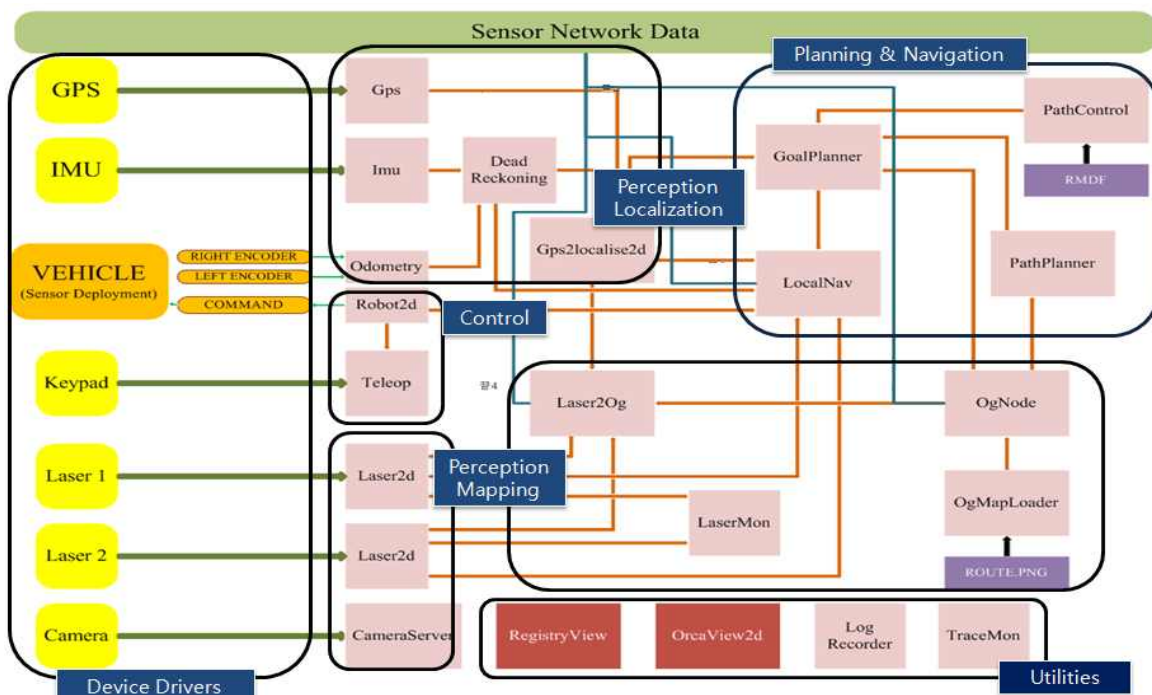
컴포넌트들의 컴포지션을 구성할 수 있는 분산 컴퓨팅 환경을 제공한다.

3.4.2 소프트웨어 플로우

[그림 4]는 무인 자율 주행 로봇에 사용된 전체 소프트웨어의 플로우를 보여준다. 각각은 구현 레벨의 독립적 모듈을 의미한다. Device Driver 파트는 실제 디바이스 장치를 의미한다. Control 파트는 로봇을 조절하기 위한 명령어를 내려주기 위한 자율/수동 컨트롤을 하는 부분의 기능을 가진 컴포넌트를 나타낸다. 위치 인식 (Perception & Localizaion), 계획 및 주행(Planning & Navigation), 상황 인식(Perception & Mapping)파트는 실제로 자율 주행을 위해 동작하는 컴포넌트들을 의미한다. 소프트웨어에 있는 모든 컴포넌트들은 로봇에 부착된 싱크 노드가 센서 네트워크 시스템과 통신한 센싱 데이터를 분산 미들웨어를 통해서 접근할 수 있다. RMDF 블럭은 로봇에게 주어지는 미션 및 경로정보를 나타낸다. 로봇에게 주어지는 정보는 텍스트 형태의 파일이며, 외부파일인 ROUTE.PNG 는 png 파일로 만들어진 지도다. RMDF 는 주행 관련 정보로서, 로봇이 반드시 지나쳐야 할 waypoints(위도, 경도 정보)와 센서 노드를 떨어뜨리라는 미션을 수행할 checkpoints(위도, 경도 정보)에 대한 내용을 포함하고 있다. 마지막으로, Utilities 파트에서는 컴포넌트들의 상태를 모니터링하는 기능과 (RegistryView) 시뮬레이션 (OrcaView2d) 하는 기능의 서비스 프로그램으로 현재 컴포넌트의 실행 상태와 시뮬레이션 환경을 제공한다.

3.4.3 무인 자율 주행

무인 자율 주행을 위해서, 로봇은 기본적으로



[그림 4] 소프트웨어 플로우 다이어그램

자신의 위치 인식을 하며, 실시간으로 맵을 생성하고, 자신의 위치를 맵에 반영 적용한다. 로봇은 적용된 위치와 경로 계획 알고리즘에 의해서 계획된 다음 waypoint 와의 위치를 비교해서 주행 정보를 계산해 Robot2d 컴포넌트를 통해서 시리얼로 로봇을 컨트롤 한다. 이 전체의 구현을 위해서 센서 정보를 사용하는데, 센서들의 정보는 크게 위치 인식과 맵핑을 위한 정보로 나뉜다. 또한 무인 자율 주행을 위해 주행, 장애물 탐지 및 회피, 경로 계획등의 알고리즘을 사용한다. 이 기능들은 일반적인 무인 자율 주행 플랫폼에 있어서 핵심 기술들이며, 각각 몇 개의 컴포넌트들의 데이터 공유 및 전달로 기능적인 부분을 갖는다. 각각 컴포넌트들은 크게 경로 계획, 위치 인식, 맵핑, 주행으로 나뉜다.

계획(Planning)은 RMDF 포맷으로 정의된 경로 및 미션 정보를 PathControl 이 GoalPlanner 에게 정보를 제공한 뒤, 그 정보를 PathPlanner 가 계획설정의 재설정이 빈번할 할 때 효과적인 Sparse Skeleton Path Planning Algorithm 을 통해 경로를 설정한다. GoalPlanner 에서는 현재 위치 정보, 맵 정보를 이용하여 PathPlanner 를 통해서 경로 계획을 끝낸 뒤, LocalNav 컴포넌트에게 waypoints 를 제공하는 형태로써 로봇 주행을 명령하게 된다. 이에 더해 센서 노드 설치 및 침입자 감시를 위해서 많은 미션이 센서 네트워크를 통해서 실시간으로 들어오는데, 이는 우선순위 기반의 Mission ordering 알고리즘을 이용하여 순서를 설정하여 계획 하게 된다.

위치인식(localization)은 IMU, Encoder, GPS 로부터 센싱된 정보를 받아서 이루어진다. IMU 와 Encoder 로부터 받아온 데이터를 Dead Reckoning 컴포넌트가 Encoder 로부터 받아온 바퀴의 회전 수 와 바퀴의 크기, IMU 로부터 방향을 실시간으로 받아와 바퀴의 크기와 회전 수를 곱해 위치를 계산한 알고리즘을 사용하여 현재 로봇의 상대적인 위치와 방향을 결정하게 된다. GPS 정보는 위도, 경도 값을 추가위성을 사용하여 일반 GPS 정보를 보정한 DGPS 장비로부터 받아오며, 그 이외에 위치 정보데이터들이 Gps2Localise2d 의 컴포넌트에서 EKF (Extended Kalman Filter)를 통해 보정되어 최종의 로봇의 위치가 결정된다. 이때, 센서들의 인식 오차를 최대 줄이고 보정하기 위하여 널리 오차 데이터 보정 및 예측 필터인 EKF 가 사용되고 있는데, 이는 정확도를 높이는 역할을 한다.

맵핑(Mapping)은 각각의 셀을 세 가지 형태의 정보(Unknown, Occupancy, and Free)로 표현하는 구현이 간단하고 다른 알고리즘을 적용하기 비교적 간단한 OgMap(Occupancy Grid Mapping) 맵 표시방법을 이용하며, 카메라와 2 개의 Laser RangeFinder 와 센서 네트워크 데이터로부터 센싱된 정보를 받아서 이루어진다. 시뮬레이션을 위해 외부 파일(ROUTE.PNG)로부터 가상의 맵을 받아오기 위한 OgMapLoader 와 실제 맵핑에 쓰이는 데이터를

받아올 Laser2d, 센서 네트워크 데이터 인터페이스 그리고 CameraServer 로 구성 되어 맵위에 표현되어 OgNode 에서 퓨전 후 최종 적인 맵 데이터를 가지게 된다.

주행(navigation)의 경우는 LocalNav 컴포넌트에서 담당을 한다. 이는 GoalPlanner 컴포넌트로부터 다음 waypoint 의 위치를 받아, 다음 waypoint 까지 가는 것을 목적으로 한다. 이 도중 장애물 탐지 및 회피 알고리즘을 사용하여 주행하는데, 장애물 탐지 및 회피 알고리즘은 맵핑에서 퓨전된 맵 정보와 위치인식 파트에서 계산된 현재 위치로부터 LocalNav 에서는 장애물 회피 알고리즘을 수행 한다. 여기서는 빠른 속도에서도 적용 되고 OgMap 맵 표현 방식과 쉽게 호환 가능한 VFH+ 알고리즘[9]을 이용하여 로봇이 다음 waypoint 까지 장애물을 피해 갈 수 있는 주행 경로를 장애물을 인식할 때마다 재결정한다. 이를 통해 로봇이 가야 할 방향과 그에 맞는 속도를 Robot2d 컴포넌트를 통해 로봇 상위제어기에 제공함으로써 센서를 통한 정보 인식과 계획에 따른 주행이 완료 되게 된다.

4. 결과 및 성능

본 절에서는 구현 된 자율 주행 로봇의 자율 주행 성능 시험 및 인식 범위 확장의 효과성을 입증하기 위해서 설정한 실험 및 시뮬레이션에 대해서 소개 및 분석한다.

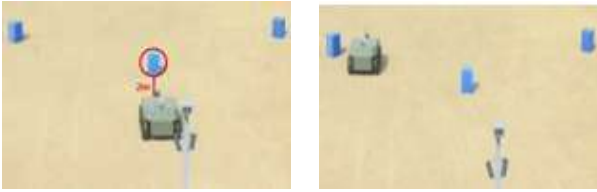
4.1 무인 자율 주행 실험 환경

로봇의 성능 평가를 위해서 우리는 실험 세트를 마련하였다. 대학교의 잔디 운동장을 실험 필드로 하였으며, 5 개의 waypoints 를 정의하였다. [그림 5]와 같이 이 waypoints 들은 맵 위에 분홍색으로 표시되어 있다. 이는 반 시계 방향 순서로 1 번부터 5 번까지 방문할 계획이다. 연두색 라인은 로봇이 이동해야 할 가장 짧은 경로를 나타내며, 노란색 사각형은 checkpoints 로서 지정된 센서 노드를 설치할 위치이다. 장애물 탐지 및 회피 능력을 테스트하기 위해서 검은색 사각형의 장애물이 존재하며, 빨간 경로는 주행 알고리즘의 장애물 회피 알고리즘으로 인해 바뀌게 될 로봇의 예상 이동 경로이다.



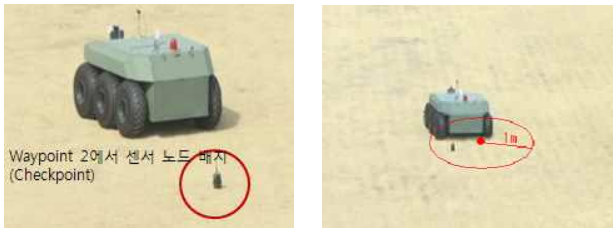
[그림 5] 실험 환경 표시

4.2 무인 자율 주행 실험 결과



[그림 6] 장애물 우회

실험 결과 예상 했던 대로, [그림 6]에서 볼 수 있는 것과 같이 VFH+ 알고리즘과 그 설정에 근거하여 2m 이내 에서 장애물을 인식한 후 우회하였다.

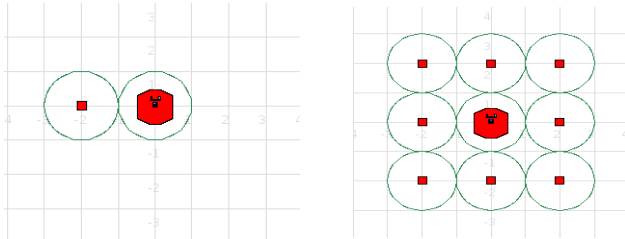


[그림 7] 센서 노드 설치 및 위치 인식 정확도

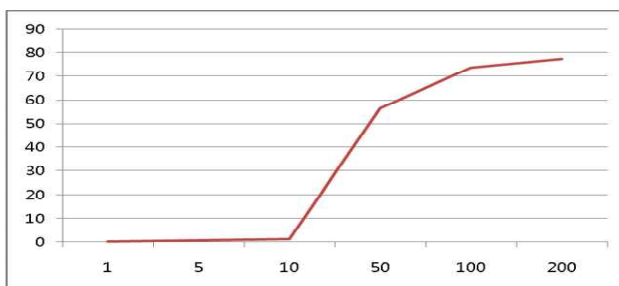
또한 [그림 7]에서와 같이 위치 인식의 정확도를 측정하고자 실제 설치 되기를 원했던 좌표와, 센서 노드가 실제로 설치된 위치를 비교해 보니 1m 안으로 매우 정확함을 알 수 있었다.

4.3 센서 범위 시뮬레이션

로봇이 센서 네트워크 구성 시스템을 사용함으로써, 원하는 위치에 센서 노드를 설치 할 수 있는데, 이가 실제로 얼마나 로봇의 센싱 범위 확장에 도움을 줄 수 있을지 시뮬레이션 해보았다. 시뮬레이션 도구로는 로봇 시뮬레이션 프로그램으로 많이 쓰이는 Player/Stage 가 이용 되었으며, 시뮬레이션 Factor 로는 센서 노드를 Optimal Coverage Location[10]에 추가 시켰을 경우 늘어나는 센싱 범위에 값을 구해 보았다.



[그림 8] Player/Stage 시뮬레이션 결과



[그림 9] 확장 노드수에 따른 센싱 범위

[그림 9]에서는 확장 노드수에 따른 센싱 범위의 시뮬레이션 결과를 나타낸 것이다. x 축은 확장 노드의 개수를 나타내고 y 축은 확장된 센싱 범위를 %로 나타낸다. 설정한 필드가 1000*1000(m)이고 센서 노드당 센싱 범위는 반지름 20m 의 원의 범위를 센싱 한다고 설정하였다. 결과와 같이 일정 수준까지는 급격히 센싱 범위가 증가하는 걸 볼 수 있다. 일정 노드 이상에서는 센싱 필드가 센싱 범위에 비해 좁아져서 센싱 범위가 접치는 경우가 생긴 것으로 예상 된다.

5. 결론

본 논문에서는 로봇이 자율 주행을 통해서 확장적인 인식 정보 획득을 위해 로봇에 부착된 싱크 노드를 통해서 감시지역 내의 센서 노드들과 통신하면서, 그 감시지역을 확장 할 수 있는 무인 자율 주행 로봇을 제안하였다. 제안한 로봇은 자율 주행 기능, 센서 네트워크 구성 시스템의 기능 및 싱크 노드으로써의 특징을 가지고 있다. 실험을 통하여 시스템의 자율주행 기능, 센서 노드 설치 기능 및 위치 인식의 정확도를 확인 하였으며, 시뮬레이션을 통하여 센서 네트워크 구성 시스템의 효과성을 확인하였다.

향후 자율 주행 소프트웨어 알고리즘의 개선과 로봇의 무게 및 배터리 문제를 개선하여 더욱 지속 가능하고 견고한 자율 주행 로봇을 만들 고자 한다.

참고문헌

- [1] Derek Armstrong et al, "A Controller-Based Autonomic Defense System," *DISCEX'03*, 2: 22-24, April 2003.
- [2] Vijay Kumar, Daniela Rus, Sanjiv Singh, "Robot and Sensor Networks for First Responders," *IEEE Pervasive Computing*, 3(4): 24-33, Oct. 2004.
- [3] Tian He et al, "Energy-Efficient Surveillance System Using Wireless Sensor Networks," *MOBISYS'04*, 270 - 283, 2004
- [4] Erann Gat, On Three-Layer Architectures, *Artificial Intelligence and Mobile Robots*, 195 -210, 1998
- [5] Internet Communication Engine, <http://www.zeroc.com>
- [6] Orca, <http://orca-robotics.sourceforge.net>
- [7] PlayerStage, <http://playerstage.sourceforge.net>
- [8] Carmen, <http://carmen.sourceforge.net>
- [9] Iwan Ulrich, Johann Borenstein, "VFH+ : Reliable Obstacle Avoidance for Fast Mobile Robots," *ICRA98*, 2:1572-1577, 1998
- [10] Seapahn Meguerdichian et al, Coverage Problems in Wireless Ad-hoc Sensor Networks, *INFOCOM01*, 3:1380-1387, 2001