

# 자율주행자동차 PHAROS

## Introduction to Autonomous Vehicle PHAROS

유 지 환\*, 박 장 식, Dmitry Ogay, Segey Bulavintsev, 김 혁, 송 영 욱, 윤 문 영, 김 재 석, 강 전 진  
 (Jee-Hwan Ryu<sup>1</sup>, Jang-Sik Park<sup>1</sup>, Dmitriy Ogay<sup>1</sup>, Segey Bulavintsev<sup>1</sup>, Hyuk Kim<sup>1</sup>, Young-wook Song<sup>1</sup>,  
 Moon-Young Yoon<sup>1</sup>, Jea-Seok Kim<sup>1</sup>, and Jeon-Jin Kang<sup>1</sup>)

<sup>1</sup>School of Mechanical Engineering, Korea University of Technology and Education

**Abstract:** This paper introduces the autonomous vehicle Pharos, which participated in the 2010 Autonomous Vehicle Competition organized by Hyundai-Kia motors. PHAROS was developed for high-speed on/off-road unmanned driving avoiding diverse patterns of obstacles. For the high speed traveling up to 60 km/h, long range terrain perception, real-time path planning and high speed vehicle motion control algorithms are developed. This paper describes the major hardware and software components of our vehicle.

**Keywords:** autonomous vehicle, unmanned driving, path planning, obstacle detection, terrain perception, vehicle motion control

### I. 서론

2011년 11월 자율주행자동차 대회가 현대기아차의 주최 아래 아시아 최초로 개최되었다[11]. 포장 및 비포장로 상의 4 km 트랙 완주를 목표로 진행된 이 대회는 트랙 안에 설치된 7개의 각기 다른 형태의 장애물을 통과하여야 하며, 장애물 미션 이외에 횡단보도 식별 및 정지 미션을 포함한다. 국내 20개의 대학이 예선에 지원하여, 11개 대학이 본선 대회에 출전하였다. 한국기술교육대학교의 차량인 파로스(PHAROS)는 8분 52초의 기록으로 완주 하였으며, 횡단보도 미션에서 별점 5분을 받아 최종 13분 52초의 기록으로 4위를 차지하였다.

최근 자율주행자동차 분야에서는 다양한 연구 활동이 활발히 이루어지고 있다. 특히 DARPA Grand Challenge [1,8]와 Urban Challenge [10]를 통하여 자율주행자동차 분야는 괄목할만한 성과를 이루어냈다. 하지만 고속 주행[9], 실시간 이동 장애물 회피 및 상황변화를 고려한 경로 계획[3,6] 그리고 고속 주행 시 차량의 자세 제어 분야[5] 등에서는 보다 진보된 연구가 필요하다.

표 1은 대회에 참여했던 타 대학 차량들 및 DARPA Urban challenge에 참여했던 차량들 중 대표적인 차량들과 PHAROS의 하드웨어를 비교하고 있다. 대회 미션의 난이도로 인하여 Stanley와 BOSS 차량은 3D laser scanner 등을 비롯하여 고가의 센서장비를 다수 사용하고 있다는 것을 알

표 1. 각 팀별 하드웨어 비교.

Table 1. Hardware specification of each team.

		PHAROS	A1 (한양대)	Boss (CMU)	Junior (Stan)
Vehicle		Hyundai SantaFe CM	Hyundai Tucson IX	chevrolet Tahoe	volkswagen passat
Engine		2.4L gasoline	2.0L diesel	5.3L gasoline	2.0L diesel
Transmission		Auto			
Percep-tion	Laser Scanner	LMS291*6	LMS291*3 LMS120*2	LMS291*6 ISF172 HDL-64E IBEO XT	LMS291*2 LD-LRS Riegl laser IBEO LUX HDL-64E
	Radar	NONE	UNKNOWN	ARS 300	BOSCH LRR2
	Camera	PointGrey Chameleon	UNKNOWN	Pointgrey firefly	NONE
GPS+HNS		Novatel Propak v3	Novatel Propak v3	Applanix POS-LV 220/420	Applanix POS-LV 420
Computer		Compact PCI Core2duo 1.6/2.16Ghz 6EA	UNKNOWN	Compact PCI Core2duo 2.16Ghz 10EA	Quad Core Server 2EA

수 있다. 일부 센서를 제외한다면 PHAROS도 Stanley와 BOSS 차량과 비교하여 하드웨어적인 측면에서는 유사한 사양을 구비하고 있음을 알 수 있다.

본 논문에서는 파로스에 적용된 자율주행 기술에 대하여 소개하고자 한다. 차량 개발 시의 주된 목표는 안정된 시스템 구축과, 고속주행시의 신뢰성 있는 장애물 감지와 안정적인 경로계획 및 주행 이었다. 이러한 사안들을 만족하기 위하여 장애물 감지, 지도 생성, 실시간 충돌 회피 및 경로 계획, 비포장도로에서의 안전한 차량 제어 등에 대한 새로운

\* 책임저자(Corresponding Author)

논문접수: 2011. 12. 5., 수정: 2012. 1. 17., 채택확정: 2012. 6. 27.

유지환: 한국기술교육대학교 기계정보공학부(jhryu@kut.ac.kr)

박장식: 한국기술교육대학교(ganggai@kut.ac.kr)

Dmitry Ogay, Segey Bulavintsev, 김 혁, 송영욱, 윤문영, 김재석,

강전진: 한국기술교육대학교 기계공학부

(espinoso@gmail.com/sergey@kut.ac.kr/perseus0616@korea.com/song923

@kut.ac.kr/htnlg@kut.ac.kr/kjs2122@kut.ac.kr/dolpals@kut.ac.kr)

※ 본 논문은 한국기술교육대학교 교육연구진흥비지원 프로그램의 지원에 의하여 수행되었음.

방법들이 제안되었다. 특히, 10 Hz로 빠르게 경로를 생성할 수 있는 실시간 경로계획 알고리즘의 제안, 60 km/h의 속도에서도 30 m 이내의 장애물을 15 cm 이내의 오차를 유지하며 감지할 수 있는 장애물 감지 알고리즘의 개발 그리고 고속주행 시에도 안정적으로 차량의 자세제어가 가능한 모션 제어 알고리즘의 구현이 고속자율주행을 가능케 하였다.

**II. 하드웨어 설계**

그림 1은 대회당일 자율주행자동차 파로스(Pharos)의 모습이다. 자율주행자동차 파로스는 현대자동차에서 생산, 판매되고 있는 산타페 CM 2.4L 가솔린 오토 사양의 차량을 기본으로 자율주행이 가능하도록 개조되었으며, 차량의 전면부에는 자율주행 시 발생할 수 있는 충격에 보호할 수 있도록 프레임 범퍼로 보강되었다.

횡방향 제어를 위해 조향축 하단에 DC 모터를 장착하였으며 모터와 조향축은 1:1기어비를 가지는 풀리를 백래쉬를 방지하기 위해 타이밍 벨트로 연결하였다. 조향에 필요한 토크는 차량에서 토크가 최대로 필요한 상황을 고려하여 4.41 N·m로 설정하였으며, 조향 속도는 사람이 조작할 수 있는 최대 속도를 기준으로 이보다 상회하도록 300 RPM으로 설정하였다. 조향을 위한 기구부는 자체 제작되었으며, 모터(Animatic)와 1:4 기어비를 가지는 기어헤드를 조합하여

24V에서 최대 5.8 N·m의 지속적인 토크와 400 RPM의 회전수를 낼 수 있도록 제작되었다. 그림 2는 횡방향 제어를 위한 조향기구의 3D CAD 모델이다. 1:1기어비의 풀리로 연결되었으며, 정확한 스티어링 각도를 얻기 위해 모터 출력축에 조향각 센서를 장착하였다.

브레이크 시스템은 조향제어기구에 사용된 동일한 DC 모터를 페달 옆에 장착하여 구현되었으며, 실험을 통해 비상시 필요한 제동력을 얻기 위해선 페달이 최소 60 mm 눌러야 하며 이를 위해선 최소 8 N의 힘이 필요하다. 이를 위하여 1:5기어비의 기어헤드를 사용하여 8 N 이상의 힘을 만들어 주었으며, 위치제어를 기반으로 제어 되게 하였다.

중방향 가속제어는 APS 센서의 출력되는 값을 모방하여 ECU로 입력되게 함으로서 아이들 상태부터 최대 가속까지 가능토록 구현하였다. 각각의 제어 과정에서 발생하는 차량의 조향각, 차량 속도, APS 센서 신호 등을 포함한 모든 데이터는 이더넷을 통해 자동적으로 컴퓨터에 저장된다.

그림 3은 자체적으로 제작한 루프랙의 모습이다. 4개의 SICK 레이저 스캐너는 차량 전방 확인에 용이하도록 상단 제일 전방에 설치되었으며, 차량의 진행 방향으로 각각 용도에 따라 다른 각도를 가지도록 설치되었다. 횡단보도 인식에 사용되는 CMOS 카메라도 차량 상단 가운데 설치되어 있다. 그 외 GPS 수신기, RF 모듈 그리고 비상정지장치 안테나 등은 상단 후방에 설치함으로 전방 센서의 간섭을



그림 1. 자율주행자동차 파로스(PHAROS).  
Fig. 1. Autonomous vehicle PHAROS.



그림 3. 지붕에 설치된 각각의 센서.  
Fig. 3. View of custom made roof rack with sensor.

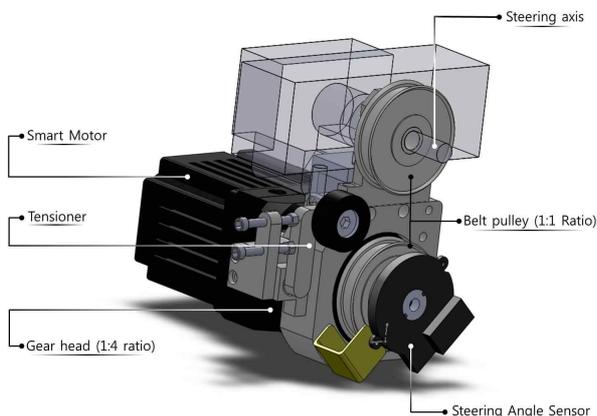


그림 2. 조향기구 3D 모델링.  
Fig. 2. Assembled 3D model of steering Actuation mechanism.



그림 4. 파로스의 트렁크 공간.  
Fig. 4. Controllers and communication system in the trunk of the vehicle.

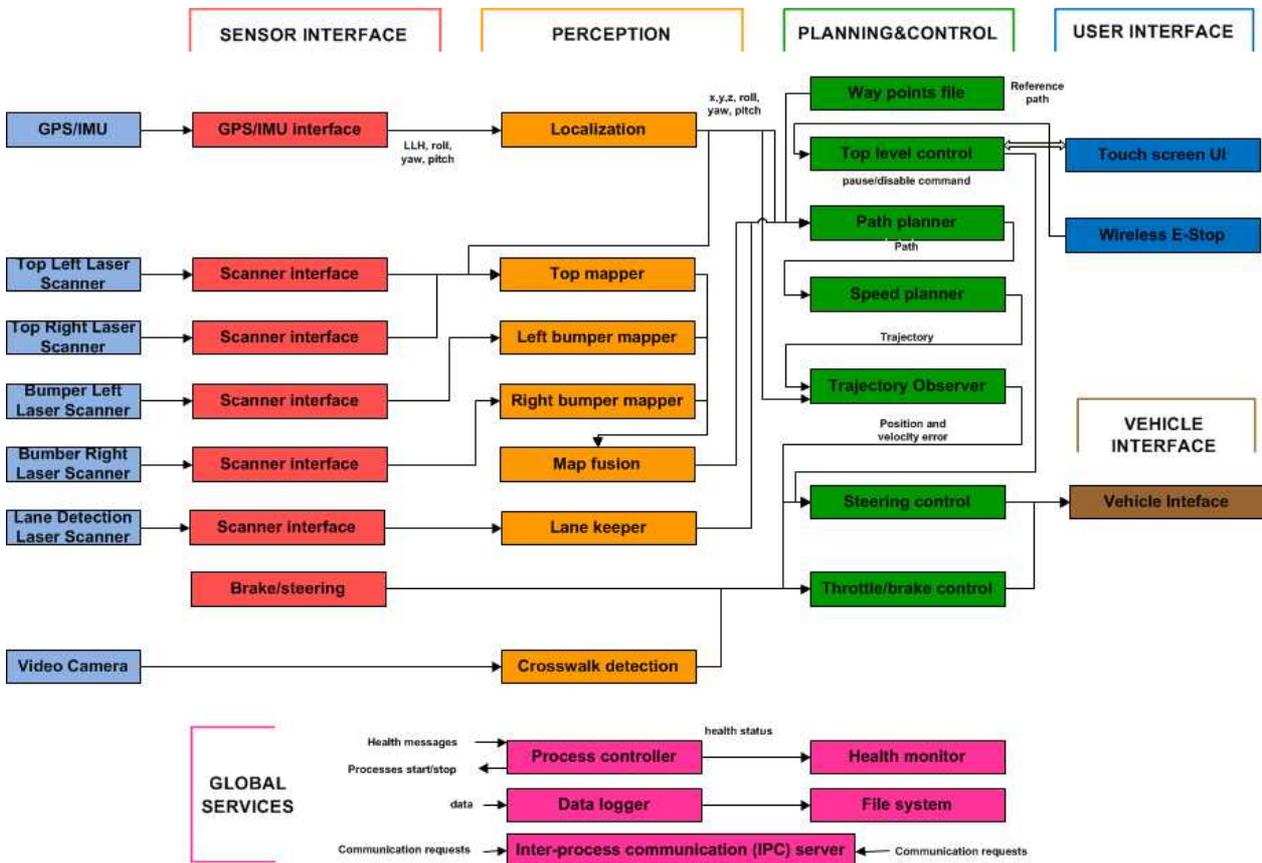


그림 5. 소프트웨어 아키텍처.

Fig. 5. Software architecture.

예방 하였다. 안전장치인 무선 비상정지장치는 주최 측에서 제공 하였으며 차량과 일정거리 안에서 무선 제어를 통하여 차량을 멈출 수 있다. 그 외 3개의 비상정지 스위치를 차량의 좌, 우 그리고 후방에 장착 하였다.

그림 4에 보이듯 파로스의 제어기와, 통신에 관련된 시스템은 모두 차량 내부 트렁크 공간에 설치되었다. 여섯 개의 Compact PCI, NI-CompactRIO, Gigabit Ethernet switches 그리고 센서와 구동장치를 위한 기타 여러 가지 장치들은 방진작업이 되어 있는 마운트에 장착 되었다. 자체 제작한 전장 시스템은 차량의 내부 트렁크 최하단에 장착 되어 차량 내 외부 모든 장치들에게 전원을 공급 한다. IMU는 흔들림이 없도록 차량 프레임에 이용하여 고정 되었다.

차량의 전체 시스템은 대략 2.4 kW의 전력이 필요하다. 직렬로 연결된 2개의 12 V 알터네이터를 차량 자체 동력과 연결함으로써, 24 V, 3.4 kW의 전력을 확보 하였으며, 이를 통하여 차량 자체 시스템을 제외한 모든 시스템이 외부 전원에 의해 충전 없이 독립적으로 장시간 작동이 가능하게 하였다.

### III. 소프트웨어 구조

파로스의 소프트웨어 시스템은 신뢰성과 손쉬운 설치가 라는 주된 원칙하에 설계되어졌다. 이 원칙을 충족시키기 위하여 각 소프트웨어는 각 독립된 컴퓨터들에서 독립된 모듈로 작성되었으며, 각 모듈간의 통신은 CMU에서 개발

된 IPC 모듈을 사용하였다. 각 프로그램을 관할하는 중앙 프로세스는 사용되지 않았다. 이는 중앙 프로세스가 중단되었을 때 신뢰도가 감소되는 것을 방지하기 위함이며, 또한 각각의 모듈을 각각의 컴퓨터에서 쉽게 실행 되도록 하는 장점과 서로 다른 운영체제를 사용할 수 있다는 장점을 활용하기 위함 이었다. 추가적으로, 입력과 출력만 제한된 상황에서 다른 모든 사항은 제한 없이 개발될 수 있다는 장점을 가지고 있다.

Watchdogs에 모든 모듈을 상주시켜 프로세스가 중단되었을 경우 자동으로 재 시작할 수 있으며 사용자 환경이나 health monitoring system을 통하여 원격으로 조종할 수도 있다.

시스템은 센서에서 데이터가 들어오면서부터 실제 핸들과 감, 가속 구동부에 데이터가 도달할 때까지 sensor interface, perception layer, planning layer와 control layer 등의 여러 개의 Processing layer로 구성되어있다. 그림 5를 보면 각 layer들 또한 몇 개의 모듈들로 구성되어 있음을 알 수 있다.

Sensor interface layer는 센서에서 센서 프로토콜을 이용하여 직접 데이터를 받아오는 모듈과 시스템 형식에 맞추어 데이터를 보내주는 모듈로 나누어져 있다. 이 모듈들은 각각 레이저 스캐너와 75 Hz 속도로 GPS/IMU와 100 Hz 속도로 통신한다. 데이터 동기화는 고려되지 않았으며, 가장 최근 데이터를 활용하여 알고리즘들이 수행되었다.

Perception layer는 차량의 localization, 레이저 스캐너로부터 장애물 지도를 그리는 laser mapping, 그리고 도로 표지판을 인식하는 vision mapping 파트로 나뉘어 있다. perception layer부터 경로 계획 모듈로는 10Hz의 속도로 데이터가 업데이트된다.

Planning layer는 2D 장애물 지도를 사용하여 운용 가능한 경로를 생성하는 경로 계획 모듈, 현재 차량의 속도와 경로의 모양에 따른 속도 제한을 기준으로 속도를 계획하는 속도 계획 모듈로 구성되어 있다.

Control layer는 실시간으로 계획된 경로와 현재 차량의 위치를 계산하여 모션 제어 모듈로 데이터를 보내주는 trajectory observer로 구성되어 있다. trajectory observer는 100 Hz로 동작하며, 차량의 위치를 모니터링 하여 경로 계획 모듈에서 데이터가 오지 않으면 차량을 멈추는 역할도 한다. 모든 모듈들은 프로세스가 정지되는 상황을 대비하여 health monitoring block에 상태 메시지를 전송한다.

#### IV. 장애물 인지와 맵핑

안전하게 장애물을 회피하기 위해서는 차량이 움직이기에 충분한 범위의 정확한 지형 지도를 만들 수 있어야 하며, 이 지형 지도에는 차폭을 고려한 협로에서의 유연한 움직임을 유지하기 위한 장애물의 정확한 위치정보를 포함하고 있어야 한다. 장애물 지도의 크기는 차가 고속으로 움직일 경우를 대비하여 저속일 때보다 더 멀리 인지할 수 있어야 한다. 레이저 스캐너는 75 Hz로 정보가 갱신되며 30 m거리에서 1 cm의 정확도를 가지고 있다. 총 6개의 레이저 스캐너 중 4개는 파로스의 지붕에, 나머지 2개는 파로스의 앞의 양쪽 범퍼에 각각 장착되어 있다.

장애물 인지를 위한 방법은 스캐너의 위치 및 역할에 의하여 달라진다. 레이저 스캐너는 각 스캐너가 위치한 고유 좌표를 기준으로 거리정보를 얻는데 얻어진 거리정보는 위치와 차량의 방향을 고려하여 보정하고 글로벌 좌표계의 3D 포인트 클라우드로 변형된다. 이 3D 포인트 클라우드는 X, Y, Z 성분으로 구성되어 있다. 이 3D 포인트 클라우드 데이터 중 인접한 두 개의 데이터들을 추출하여 높이 (Z 값)를 비교하는 방식을 사용하여 2D 기반의 occupancy grid map [2]이 작성되었다. 높이 차이가 일정 값 이상일 경우 해당 위치의 셀에 장애물이 있다고 판단하였으며, 이 경우에 해당되지 않거나 데이터 값이 존재하지 않는 경우 장애물이 없거나 모르는 구간으로 정의하였다. 각 셀은 글로벌 위치좌표를 사용하고 장애물의 유무에 대한 정보를 지니고 있다. 셀의 크기는 바랄 수 있는 최대거리인 30 m를 기준으로 결정하였다.

장애물의 인지를 위하여 각 레이저 스캐너의 데이터를 조합하여 장애물이 있는 지점은 셀을 채우고 그렇지 않은 지점은 셀을 비워 정의 하였다. 그림 6은 우리 대학 캠퍼스를 그린 occupancy grid map이다. 언급한 바와 같이 장애물이 있는 공간은 하얀 색으로, 나머지 빈 공간은 회색으로 정의하였다.

범퍼 레이저 스캐너의 레이저는 지면과 평행하게 진행하며 두 레이저 스캐너는 각각 좌, 우를 향하고 있다. 장애물

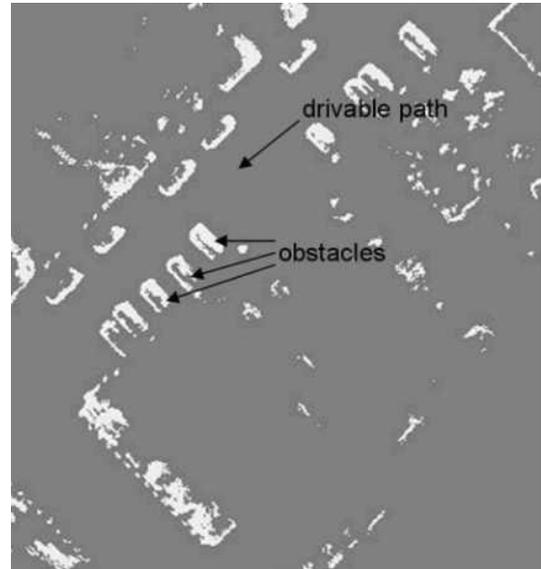


그림 6. Occupancy grid map의 예.

Fig. 6. An example of occupancy grid map.

인지를 위하여 데이터를 처리하는 과정에서 각 레이저 스캐너는 데이터에 대한 오류를 포함할 수 있으므로 여러 번 같은 위치에 장애물이 인지되면 장애물로 정의하는 방법을 사용하였다. 실험을 통하여 두 알고리즘의 신뢰성이 검증되었으며 제안된 알고리즘을 바탕으로 60 km/h의 속도에서 30m 앞을 바라보며 15 cm 이내의 위치 오차를 유지하며 장애물을 회피할 수 있다.

#### V. 도로 경계 판단

여러 요인에 의하여 DGPS의 좌표계는 고정되어 있지 않고 움직인다. 이로 인해 경로의 궤적 또한 변형 되어 자율주행 시 원하는 경로를 유지하는데 문제가 있다. 이러한 현상을 피하기 위하여 도로의 경계를 찾아내어 차선을 지키는 알고리즘이 필요하였다. 도로 위에 장애물이 없다면 차량은 우측 도로로 진행하여야 한다. 이것은 간단히 도로의 경계와 차량의 거리를 유지하는 것으로 해결 할 수 있다. 일반적으로 도로의 검정색 아스팔트는 다른 지형보다 레이저의 반사율이 낮다. 이 반사율을 기반으로 occupancy grid를 그릴 수 있다. 그리드의 한 셀이 특정 반사율 값보다 높게 측정되면 그 셀을 운용할 수 없다. 기준 경로는 도로의 경계와 평행이지만 편심 된 차량의 위치정보는 시간이 지날수록 편심양이 더 커지게 된다. 도로의 경계와의 거리를 찾기 위하여 일차원의 low-pass Kalman filter를 사용하였으며, Kalman filter의 상태 값은 도로의 경계와 참조 경로의 거리이다. Kalman filter는 기준 경로와 직각인 패턴을 쪼개어 가장 큰 편심을 찾아내었다. 그림 7에서 쪼개진 패턴을 볼 수 있다. 이 경우 도로의 경계는 천천히 변화한다. 특정 반사율보다 높은 값을 갖는 장애물이 나타나 갑작스런 반사율의 증가가 이루어지면 도로 경계에 가장 작은 영향을 주는 결과로 판단한다. 경로 계획 모듈이 사용하는 Kalman filter의 출력 값은 차선이탈을 방지하고 DGPS 좌표 값의 변형에 대한 영향을 줄일 수 있다.

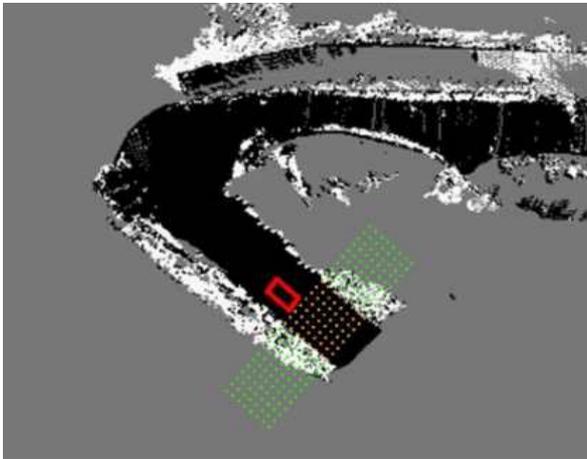


그림 7. 도로 경계 판단의 예.  
Fig. 7. An example of road boundary estimation.

**VI. 경로 계획 및 속도 계획**

이번 대회에서 가장 중요한 과제는 자동차가 미리 정해진 기준 경로를 주행하는 것이며 주행 중 만나게 되는 장애물을 차량이 스스로 인지 및 회피 하는 것이다. 여기서 기준 경로 추종 시 도로 끝 등이 장애물로 인식 될 수 있는데 이는 빈번히 발생하는 GPS Drifting 때문이다.

경로계획을 할 때는 접선 벡터와 곡률 등의 계산된 파라미터들에 의해 이루어지기 때문에 미리 작성되는 기준경로는 Smoothing(유연화작업)이 이루어져야 한다.

본 차량은 RRT [7]를 기초로 경로를 계획하며, 이 알고리즘에 기구학적인 요소들이 추가되어 차량은 좁은 경로까지도 주행이 가능 하지만 고속 주행을 위해 빠르게 변하는 환경에 스스로 대처하는 자율주행 차량을 제작하는 데는 연산시간문제[4]로 인하여 많은 어려움이 있었다. 일반적으로 환경 인지에 사용되는 센서의 범위가 한정되어 있기 때문에 이는 효율적으로 작동되는 경로 계획 모듈이 만들어 지면 해결 될 것이라 예상을 하였다.

이를 위해 RRT 알고리즘 일부를 수정하였으며 또한 경로 계획 모듈이 효율 적으로 작동하기위해 단일 쓰레드로 운영 하였으며, 이를 바탕으로 차량 속도는 최고 60 km/h, 경로 재생성은 100 ms 주기로 이루어 졌다.

기본적인 RRT 알고리즘을 무인자동차에 맞게 여러 번 수정되었으며, 사용된 차량의 상태 변수는 x, y 그리고 heading angle 이렇게 3성분으로 구성되었다. RRT의 트리 확장 시에 차량의 구속조건을 사용하였으며, 현재 속도 및 향후 속도의 변화까지는 고려하지 않았으므로, 선속도 및 조향 각은 상태변수에 포함되지 않은 상태이지만, 이를 고려하여 향후 추가적인 연구가 필요하다.

차량이 곡률을 가지는 경로에 진입 시, 안정적으로 주행 할 수 있는 최대 속도를 결정 하여야 하는데 이를 위해 차량의 기구학적 요소와 경로 상의 곡률을 고려하였다. 무인 차량이 가져야 할 가장 최적의 상태는 빠르고 자연스러운 운동이다. 그림 8은 인공적으로 만들어진 터널을 통과 시 경로 계획을 보여주고 있다. 8(a)는 일반적으로 쓰이는 RRT 알고리즘으로 추출된 값을 사용하고 있어 값들 중 일부

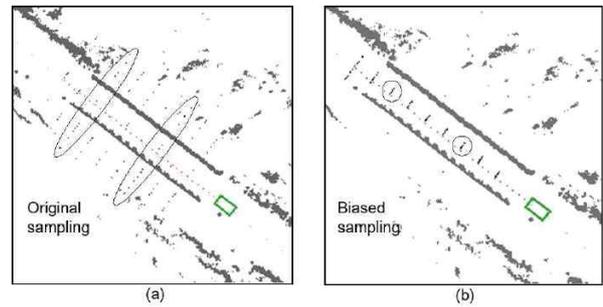


그림 8. (a) 기존 RRT의 경로생성 결과 (b) 수정된 RRT의 경로 생성 결과.

Fig. 8. Comparison of sampling distribution(black dots) of non-optimized algorithm (a), and optimized algorithm (b). Passing artificially made narrow tunnel.

은 차가 갈수 없는 곳까지 경로로 표시 하고 있다. 하지만 8(b)와 같이 수정된 RRT 알고리즘은 자동차가 갈수 없는 장소에 경로 생성을 하지 않아 터널 통과 시 경로 계획하는 시간을 줄일 수 있다.

계획된 경로로 자동차가 출발하기 전에 필터를 통해 경로가 부드럽게 변해야 하는데, 가장 중요한 것은 접선 벡터들과 모든 점을aaa 경로의 곡률반경으로 계산하는 것이다.

차량의 속도 계획을 하기 위해 해야 될 가장 중요한 점은 가장 빠른 속도로 제한된 범위 안에서 안전하게 자동차가 계획된 경로를 따라가는 것이다. 이때 충돌 위험이 생기면 브레이크를 최대로 사용해서 차를 감속시켜야 한다. 앞서 언급한 대로 안전한 속도를 내기 위해서는 계획된 경로의 곡률반경을 알아야 수행이 가능하고 어떤 이유로 갑자기 경로의 곡률반경이 변경되면, 속도 계획 모듈은 이와 동시에 변경된 곡률에 맞는 속도 값을 생성 한다.

**VII. 실시간 차량 운동 제어**

차량의 현재 위치와 기준경로간의 오차는 관측기에서 측정되며, 이 두 요소간의 오차 값이 감지되면 운동 제어기는 기준경로를 따라가기 위하여 조향, 가속, 제동 명령을 출력으로 내보낸다.

그림 9를 참고하면, 조향 제어기는 기준경로에서의 측면 오프셋, 각도 차이, 곡률이 입력으로 들어가며, 100 Hz의

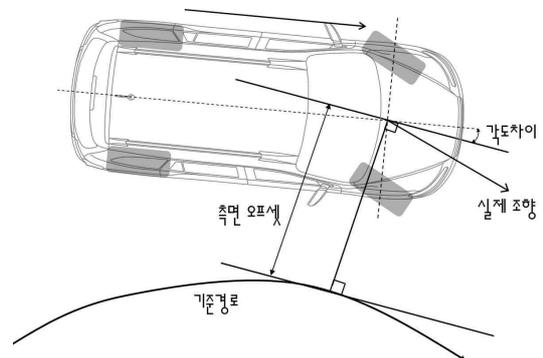


그림 9. 조향제어기 개념도.  
Fig. 9. Illustration of the steering controller.

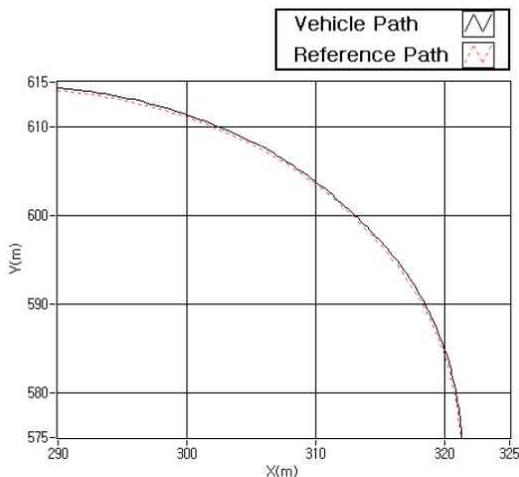
속도로 조향 모터에 출력을 내보낸다. 기본적인 조향각 제어법칙은 Look ahead consideration을 제외하고는 [8]과 거의 유사하다. 측면 오프셋에 대해서는 PD 제어기에 기반을 둔 간단한 Kinematic을 사용한다. 참고로, 측면 오프셋은 차량 앞바퀴 축 중앙 점과 기준경로 상의 가장 가까운 점간의 거리로 정의한다. 각도 차이는 앞의 기준경로 점의 접선 성분과 차량이 향하고 있는 방향의 상대적인 차이로 정의한다.

제어기 설계 시, 각도 차이만을 보상함으로써 일정 한도까지는 차량이 기준경로를 쫓아갈 수는 있지만, 측면 오프셋의 고려 없이는 정적 상태 오차를 완전히 보상할 수 없다. 따라서 본 차량은 측면 오프셋과 각도차이를 기본으로 하였고, 더 완벽한 보상을 위하여 곡률을 추가 입력으로 주었다. 각 성분에 대한 제어기 이득 값은 차량속도에 따라 유동적으로 변화도록 튜닝 하였다.

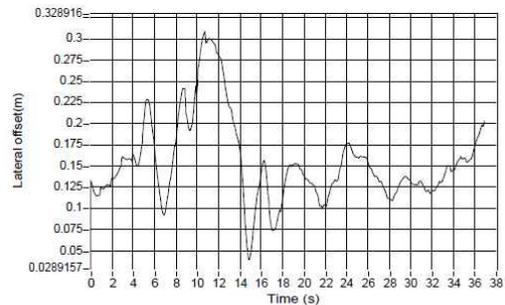
이러한 방법으로 제어기를 설계한 결과, 포장도로와 비포장도로는 물론 급커브 길에서도 안정적으로 참고궤적을 쫓아가는 차량의 움직임을 확인할 수 있었다.

그림 10은 파로스가 40 m 곡률의 커브 길을 도는 동안의 조향 출력을 보여준다. 이 때, 조향제어기 성능은 차속 60 km/h에서 30 cm 이내의 측면 오프셋을 유지한다.

속도 계획 및 횡단보도 감지 모듈은 Low-level speed controller에 명령을 보낸다. 속도 제어기는 각 모듈의 가속



(a) Tracking performance of the steering controller



(b) Lateral offset error of the vehicle

그림 10. 반경 40 m 커브에서 조향제어 결과.  
Fig. 10. Steering performance of the vehicle on 40 m radius curved road.

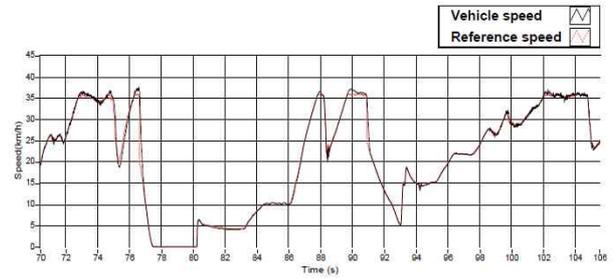


그림 11. 속도 입력과 실제 차량 속도 비교.  
Fig. 11. Control performance of the speed controller.

명령과 감속 명령을 속도 명령으로 바꾸어 보낸다. 일단 원하는 속도가 결정되면, 속도 제어기는 가속 명령이나 감속 명령을 선별하여 하나의 명령만을 처리하게 되는데, 이 명령은 차량의 진행 방향에 대하여 가속 시켜 주거나 감속 시켜 주는 스로틀과 브레이크를 제어하게 된다.

제어기는 속도 오차의 누적 합과 누적 적분 값을 더하여 계산을 수행한다.(PI제어기 사용) 이러한 계산 값에 기초하여, 가속 명령과 감속 명령 하나가 선택적으로 출력되며, 데드 존 가까이에서는 채터링(감, 가속에 번갈아 가며 나타남)을 피할 수 있도록 설계하였다.

그림 11은 고안된 속도 제어기의 제어 출력을 보여준다. 그래프를 살펴보면 입력 속도 값에 대하여 출력 속도 값이 매우 잘 추종하고 있음을 확인할 수 있다.

VIII. 결론

본 논문은 2010년 한국에서 개최된 자율주행자동차경진대회를 위해 개발된 파로스(PHAROS) 시스템의 개요를 설명 하였다.

기본적으로 파로스의 제어 소프트웨어는 무인자동차 분야에서 널리 사용되는 방법론들을 활용하여 구성되었지만, 많은 모듈들은 고속에서의 자율주행을 구현하기위해 특별하게 개선되었다. 원거리의 지형인식 및 장애물 인지 알고리즘, 실시간 경로 계획 및 장애물 회피 알고리즘, 그리고 고속에서의 차량 모션 제어 알고리즘 등은 신뢰성을 높이기 위해 광범위한 실험이 이루어졌다.

파로스는 성공적으로 레이스를 마쳤지만 아직도 개선해야 할 많은 이슈를 가지고 있다. 동적 장애물에 대한 대처방안, 최고 속도에선 30 m 이하로 제한된 전방 인지. 때로는 장애물 회피를 위해서 실제 경로에서 6 m까지 벗어나 주행하여야 하지만 장애물 인지 거리가 제한되어 있다면 장애물 회피를 위해서 차량의 속도를 줄여야 하기도 한다. 이 때문에 차량이 보다 먼 거리의 환경을 인식 할 수 있는 장애물 인식 알고리즘에 대한 개발이 요구된다. 또한, 제안된 알고리즘 및 하드웨어를 바탕으로 어느 정도의 안정성이 보장되는 지에 대한 엄밀한 해석이 향후 수반되어야 할 것이다.

참고문헌

[1] M. Buehler, K. Iagnemma, S. Singh, The 2005 DARPA Grand Challenge: the Great Robot Race, Springer Tracts in Advanced Robotics 36, Springer-verlag, Berlin; 2007.

- [2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol 22, no. 6, pp. 46-57, 1989.
- [3] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel. "Practical search techniques in path planning for autonomous driving," in *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics*, 2008.
- [4] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *ALAA Journal of Guidance and Control*, vol. 25, no. 1, pp. 116-129, 2002.
- [5] A. Kelly, A Partial Analysis of the High Speed Autonomous Navigation Problem, Project report of "Perception for outdoor navigation" and "Unmanned Ground Vehicle System," Carnegie Mellon University, 1994.
- [6] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using RRT," in *Proc. IROS*, pp. 1681- 1686, 2008.
- [7] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400, 2001.
- [8] S. Thrun and et al., "Stanley: The Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661-692, 2006.
- [9] C. Urmson and et al., "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 427-508, 2006.
- [10] C. Urmson and et al., "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425-466, 2008.
- [11] <http://www.hyundai-ngv.com/techcontest/>, in Korean.



**유지환**

2002년 한국과학기술원 기계공학과(공학박사). 2005년~현재 한국기술교육대학교 기계정보공학부 교수로 재직중. 관심분야는 Haptics, Teleoperation, Autonomous vehicle.



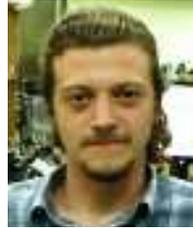
**박장식**

2010년 한국기술교육대학교 기계정보공학부 졸업. 2012년 현재 한국기술교육대학교 대학원 석사과정중. 관심분야는 로봇 기구설계, Autonomous vehicle.



**Dmitry Ogai**

2011년 한국기술교육대학교 기계공학부 대학원 석사. 관심분야는 Teleoperation, Exoskeleton robot, Haptics Autonomous vehicle.



**Sergey Bulacintsev**

2011년 한국기술교육대학교 기계공학부 대학원 석사. 관심분야는 Teleoperation, Exoskeleton robot, Haptics Autonomous vehicle.



**김혁**

2009년 한국기술교육대학교 메카트로닉스공학부 졸업. 2011년 한국기술교육대학교 기계공학부 대학원 석사. 관심분야는 Teleoperation, Haptics Control, Autonomous vehicle.



**송영욱**

2009년 한국기술교육대학교 기계정보공학부 졸업. 2012년 한국기술교육대학교 기계공학부 대학원 석사과정중. 관심분야는 Haptics Teleoperation, Autonomous vehicle.



**윤문영**

2010년 한국기술교육대학교 기계정보공학부 졸업. 2012년 한국기술교육대학교 기계공학부 대학원 석사과정중. 관심분야는 Haptics Teleoperation, Autonomous vehicle.



**김재석**

2011년 한국기술교육대학교 기계정보공학부 졸업. 2012년 한국기술교육대학교 기계공학부 대학원 석사과정중. 관심분야는 Teleoperation, Path planning, Autonomous vehicle.



**강전진**

2011년 한국기술교육대학교 기계정보공학부 졸업. 2012년 한국기술교육대학교 기계공학부 대학원 석사과정중. 관심분야는 기구부 설계, Control, Autonomous vehicle.