

비선형 모델 예측 제어를 이용한 차동 구동 로봇의 경로 추종

Path Tracking with Nonlinear Model Predictive Control for Differential Drive Wheeled Robot

최재완¹·이건희¹·이치범[†]

Jaewan Choi¹, Geonhee Lee¹, Chibum Lee[†]

Abstract: A differential drive wheeled robot is a kind of mobile robot suitable for indoor navigation. Model predictive control is an optimal control technique with various advantages and can achieve excellent performance. One of the main advantages of model predictive control is that it can easily handle constraints. Therefore, it deals with realistic constraints of the mobile robot and achieves admirable performance for trajectory tracking. In addition, the intention of the robot can be properly realized by adjusting the weight of the cost function component. This control technique is applied to the local planner of the navigation component so that the mobile robot can operate in real environment. Using the Robot Operating System (ROS), which has transcendent advantages in robot development, we have ensured that the algorithm works in the simulation and real experiment.

Keywords: Model Predictive Control, Differential Drive Wheeled Robot, Path Tracking, Navigation

1. 서 론

지난 수십 년 동안 모바일 로봇 분야에서 급속한 발전이 있었다. 최근에는 서비스 로봇 및 물류 로봇과 같은 산업 응용 분야에서 큰 발전을 이루어지고 있다. 차동 구동형 로봇은 다양한 로봇 구동 유형 중에서 간단한 구조와 제자리 회전이 가능하다는 장점을 가지고 있으므로, 좁은 공간, 장애물이 많은 복잡한 공간에서 효율적으로 동작할 수 있다. 이로 인해 실내 로봇으로 널리 사용되고 있다^[1]. 차동 구동형 모바일 로봇의 두 구동륜은 개별적인 모터에 의해 독립적으로 구동하여 직진 주행과 회전을 수행한다. 본 연구에서는 차동 구동형 모바일 로봇의 네비게이션을 위한 비선형 모델 예측 제어(Model Predictive Control)에 기반한 경로 추종(path tracking) 알고리즘을 제안한다.

이동 구간(receding horizon)에 기반한 최적 제어 기법인 MPC는 화학 공정을 비롯한 다양한 영역에서 널리 활용되고 있다. MPC는 MIMO (Multi-Input Multi-Output) 시스템을 쉽게 설계하고 제약조건을 쉽게 적용할 수 있으며, 예측 능력 (preview capability)를 가져 성능을 향상시킨다. MPC는 시간 간격(time step)동안 실시간 최적화를 수행하기 때문에, 연산량이 많이 필요하다. 그러나 최근에 프로세서 기술의 발전으로 인해 저가의 프로세서로도 고성능 연산을 가능하게 되어 자동차와 로봇틱스 등의 분야에서도 많이 사용되고 있다. 이 미, 모바일 로봇에도 MPC 기법들이 많이 적용되어 왔다. 특히 Ackermann 조향 메커니즘으로 구동되는 바이시클 모델에 적용되었다^[2-7]. 차동 구동 모델에도 연구되었지만, 원, 무한루프 형태의 완만한 곡선의 경로 추종 실험만 진행 되었고, 급격한 변화를 가진 경로에 대한 실험과 실제 환경에서의 네비게이션 실험은 진행되지 않았다^[8-10].

본 연구에서는 차동 구동 모바일 로봇에 MPC를 접목시켜 경로 추종을 수행하고, 시뮬레이션 및 실제로 모바일 로봇을 구동할 수 있는 환경을 구성하여 실험을 진행하였다. 환경 구성을 위해 ROS (Robot Operating System)^[11]의 네비게이션 스택(navigation stack)을 활용하여, MPC로 경로 추종을 수행하도록 로컬 플래너(local planner)를 구성하였다. 2장에서는 연

Received : May. 20. 2020; Revised : Jun. 29. 2020; Accepted : Jul. 14. 2020

※ This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008473, HRD Program for Industrial Innovation)

1. Graduate Student, Mechanical Design and Robot Engineering, Graduate School, Seoul National University of Science and Technology, Seoul, Korea (wodhks7968, rjsgml6392@naver.com)

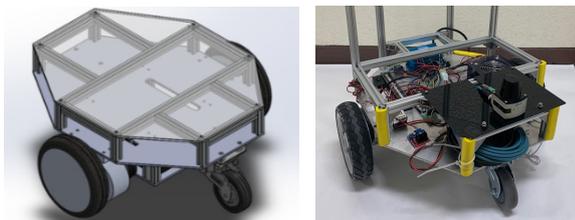
† Associate Professor, Corresponding author: Mechanical System & Design Engineering, Seoul National University of Science and Technology, Seoul, Korea (chibum@seoultech.ac.kr)

구에 사용된 차동 구동 로봇을 설명하고, 모델 파라미터를 기술하였다. 3장에서는 MPC 최적화에 필요한 모델, 비용함수(cost function,) 제약조건(constraints)에 대해 기술하였고, 네비게이션을 구성하는 글로벌 경로 플래너(global path planner)와 비선형 MPC로 경로 추종을 수행하는 로컬 플래너(local planner)에 대해 설명하였다. 4장에서는 시뮬레이션과 실제 환경 실험을 통해 MPC를 적용한 경로 추종 성능을 pure pursuit 알고리즘과 비교하여 설명하였다.

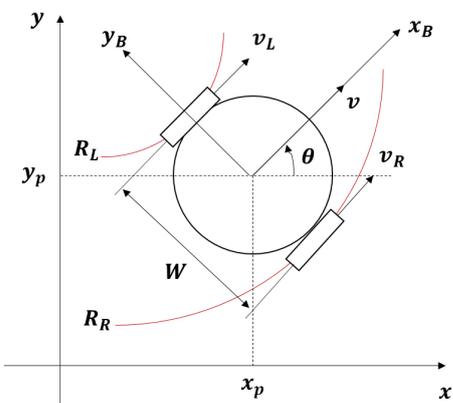
2. 차동 구동형 로봇

본 연구에 사용된 차동 구동형 로봇[Fig. 1]은 휠이 결합된 BLDC 모터 2개(PEWM-01, Pei Scooter사, China)를 구동부로 사용하고, 앞/뒤에 캐스터가 부착되어 있다. 평상시에는 구동륜과 후륜 캐스터를 통하여 3륜 접촉이 이루어진다. 차동 구동형 로봇은 구동륜의 각도나 각속도 정보를 측정하기 용이하며, Ackermann 조향 모델에서는 불가능한 제자리 회전이 가능하다는 장점을 지닌다.

차동 구동형 로봇의 기구식은 [Fig. 2]에서 표시한 2차원 좌표로 표시된다. 요우각(yaw) θ 는 두 바퀴의 중심에서 x축에 대한 로봇의 진행 방향으로 표시된다^[12]. 좌륜과 우륜은 각각 v_L , v_R 의 속도를 지니며, 사이드 슬립이 없다는 가정 하에, 모바일 로봇의 진행 방향 속도는 $v = (v_L + v_R)/2$ 로 주어진다.



[Fig. 1] Differential drive wheeled robot



[Fig. 2] Coordinate system for a differential drive robot

전역 좌표에서 대한 운동 방정식은 다음과 같다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

요우각 회전속도는 $\omega = (v_R - v_L)/W$ 이다. W 는 윤거(wheel tread)이다. 로봇의 입력인 전방 속도 v 와 회전속도 ω 로부터 모터의 요구 회전 속도 w_R , w_L 을 식 (2)와 같이 계산할 수 있다. R 은 구동륜의 반경이다.

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{W} & -\frac{1}{W} \end{bmatrix}^{-1} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

로봇의 전체 크기는 620 mm × 550 mm × 370 mm, 구동륜의 반지름 R 은 101.5 mm, 윤거 W 는 530 mm이다. ROS 구동을 위하여 Intel Celeron J4105을 장착한 Odroid H2(하드커널사, 대한민국)를 사용하였고, 라이더는 Sick사 Tim561을 장착하였다.

3. MPC 경로추종과 네비게이션

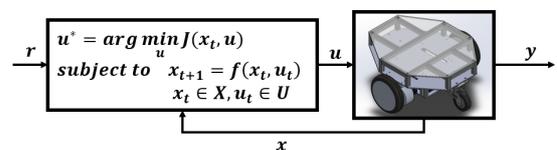
3.1 MPC 경로 추종을 위한 최적화 문제

MPC는 매 업데이트 시간 간격마다 이동예측공간(receding prediction horizon)에 대하여 비용 함수를 최소화하는 제어 값을 찾는 최적화 문제를 푼다. PID 제어^[13] 같은 전통적인 경로 추종과는 달리, MPC를 이용한 경로 추종은 로봇의 운동 방정식과, 상태 변수와 입력에 따른 비용 함수 및 제약조건을 필요로 한다. MPC의 장점은 비용 함수에 포함된 가중치를 다르게 조절함으로써, 의도하는 행동을 구현할 수 있다.

본 연구에서는 주어진 경로에 대해서 온라인으로 경로를 수정하면서 궤적(trajjectory)을 생성하도록 계층 이동공간 제어기(hierarchical receding horizon controller)로서 MPC를 활용하였다^[14].

3.1.1 경로 추종 동역학 운동 방정식

로봇의 2차원 평면 상에서의 위치 (x, y) 및 전방 속도 v , 요우각 θ 를 상태 변수로 지정하고, 전방가속도 a 와 회전 각속도 ω 를 제어 입력 u 로 지정하였다. [Fig. 3]은 MPC 제어가 포함된 제어 시스템을 보여준다.



[Fig. 3] Closed-loop control system with MPC

경로 추종 문제에 적용하기 위해서, 횡방향 트랙 오차(cross track error) d , 방향 오차(orientation error) η 를 상태 변수에 추가하였다. 횡방향 트랙 오차 d 는 2차원 좌표상에서 로봇 위치와 경로 사이의 유클리드 거리를 나타내며, 방향 오차 η 는 모바일로봇의 전방 방향과 경로의 접선 방향의 차이를 측정한다.

상태 변수 $x = (x \ y \ \theta \ v \ d \ \eta)^T$ 를 사용하여 이산시스템으로 모델링한 로봇의 비선형 운동방정식 $x_{t+1} = f(x_t, u_t)$ 은 식 (3)과 같다.

$$x_{t+1} = x_t + v_t \cos(\theta) \cdot dt \quad (3a)$$

$$y_{t+1} = y_t + v_t \sin(\theta) \cdot dt \quad (3b)$$

$$\theta_{t+1} = \theta_t + \omega_t \cdot dt \quad (3c)$$

$$v_{t+1} = v_t + a_t \cdot dt \quad (3d)$$

$$d_{t+1} = g(x_t) - y_t + v_t \sin(\eta) \cdot dt \quad (3e)$$

$$\eta_{t+1} = \theta_t - \theta_t^* + \omega_t \cdot dt \quad (3f)$$

여기서, dt 는 이산화에 사용된 시간 간격이고, $g(x_t)$ 는 다항식으로 표현된 곡선 접합(curve fitting) 함수로서, x_t 를 입력으로 받아 y 좌표를 출력한다. θ_t^* 는 곡선 접합 함수 $g(x_t)$ 의 기울기를 의미한다.

3.1.2 비용 함수와 제약조건

MPC의 최적화 문제는 식 (4)의 비용 함수와 식 (5)의 제약 조건으로 표현된다.

$$\begin{aligned} \min_{x, u} \{ & \sum_{t=0}^N [w_v \| v_t - v_t^{ref} \|^2 + w_d \| d_t \|^2 + w_\eta \| \eta_t \|^2] \\ & + \sum_{t=1}^{N-1} [w_\omega \| \omega_t \|^2 + w_a \| a_t \|^2] \\ & + \sum_{t=2}^N [w_\omega \| \omega_t - \omega_{t-1} \|^2 + w_a \| a_t - a_{t-1} \|^2] \} \quad (4) \end{aligned}$$

subject to

$$x_t = x(t) \quad (5a)$$

$$x_{k+1} = f(x_k, u_k), \quad \forall \in [t_1, \dots, t_{N-1}] \quad (5b)$$

$$x_m \leq x_k \leq x_M, \quad \forall \in [t_2, \dots, t_N] \quad (5c)$$

$$u_m \leq u_k \leq u_M, \quad \forall \in [t_1, \dots, t_{N-1}] \quad (5d)$$

여기서, N 은 이동 구간의 길이(prediction horizon length)이며, $W_v, W_d, W_\eta, W_\omega, W_a, W_\omega, W_a$ 는 개별 성능에 대한 가중치이고, x_m, u_m 은 각각 상태 변수와 제어입력에 대한 하한이고, x_M, u_M 은 상한이다. 식 (5a)는 최적화 과정에서 현재 상태 변수 $x(t)$ 을 이동 예측 구간의 초기값으로 부여한 것이고, 식 (5b)는 이동 구간에서 운동방정식을 따르도록 한다. 식 (5c)는

상태 제약조건을 나타내고 식 (5d)는 모바일 로봇을 제어하는데 사용되는 가속도 및 각속도 입력에 대한 제한 범위를 의미한다. 식 (4), (5)의 최적화 문제는 이차 프로그래밍(Quadratic Programming)을 통하여 해결할 수 있다.

3.2 ROS를 활용한 MPC 경로 추종과 네비게이션 구현

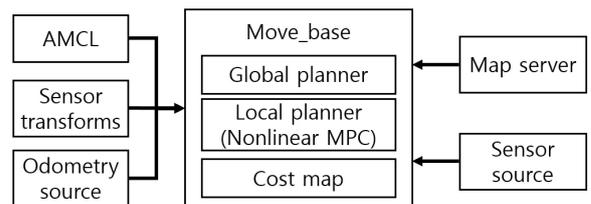
본 연구에서는 ROS에서 제공하는 네비게이션 스택을 활용하여 구성하였다^[15]. 비선형 MPC로 경로 추종을 구현하기 위해 로컬 플래너 부분을 MPC 경로 추종기로 [Fig. 4]와 같이 구현하였다. 이렇게 구현함으로써, 기존의 네비게이션 스택이 제공하는 위치 결정(localization), 지도 서비스와 글로벌 플래너 등을 그대로 사용할 수 있다.

모바일로봇의 네비게이션을 구현하는 방법은 다양하게 존재한다. 일반적으로 로컬 플래너의 역할은 DWA (Dynamic Window Approach)^[16]와 같이 일정 범위 내의 동적 장애물을 회피하여 새로운 경로를 만들고 추종하는 것에 초점을 둔다^[17]. 그러나 본 연구에서는 비선형 MPC 기반의 로컬 플래너가 글로벌 플래너가 생성한 경로를 추종하면서 최적조건을 따르는 로봇의 입력을 생성하도록 하였다.

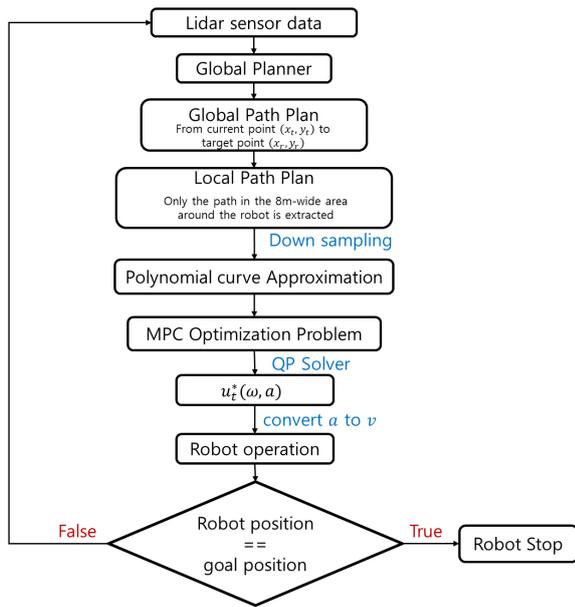
본 연구에서의 MPC를 포함한 로컬 플래너는 장애물을 인식하고 회피하는 문제를 고려하지 않았기 때문에 글로벌 플래너가 A* 알고리즘^[18]을 사용하여 장애물이 반영된 지도에 대해서 목표지점까지의 경로를 찾는 간격(1초)으로 생성한다. 로컬 플래너는 글로벌 플래너가 생성한 경로 중 로봇이 이동하는 위치를 기준으로 전방 5m까지의 경로를 통해 궤적을 생성하고, 로봇의 가속도 a 와 회전속도 ω 를 산출한다. 실제 로봇 제어에 필요한 입력인 진행방향 속도 v 는 가속도 a 를 적분하여 사용하였고, 최종적으로 각 구동륜의 속도는 식 (2)를 이용하여 모터 제어기에서 조절된다.

네비게이션 과정을 정리하면 [Fig. 5]와 같다.

MPC의 최적화 가중치는 [Table 1]과 같이 설정하였다. 경로 추종 성능을 우선 시하여 우선순위를 횡방향 트랙 오차, 방향 오차에 중점을 두었으며, 빠른 주행을 위하여 진행 방향 속도 오차를 중요하게 두었다. 또한 급작스러운 변화를 방지하기 위하여 요우각 가속도와 저크(jerk)에 대한 가중치를 고려하였다.



[Fig. 4] Navigation stack architecture



[Fig. 5] Navigation flow chart

[Table 1] MPC optimization weight

Weight	Value
Weight for forward velocity error, w_v (m/s) ⁻²	100
Weight for cross track error, w_y (m) ⁻²	2000
Weight for orientation error, w_a (rad) ⁻²	100
Weight for yaw rate, w_ω (rad/s) ⁻²	0.0
Weight for acceleration, w_a (m/s^2) ⁻²	0.0
Weight for yaw acceleration, w_ω (rad/s^2) ⁻²	1000
Weight for forward jerk, w_a (m/s^3) ⁻²	50

정해진 PC 성능에서 주어진 MPC 최적화의 연산 시간은 예측하는 구간의 시간 간격 dt 와 개수 N 에 의해 결정된다. 로봇의 실시간 제어를 0.1초 간격으로 수행하기 위하여, dt 는 실시간 제어와 동일한 0.1초로 정하였고, N 에 따른 실제 연산 시간을 확인하여, 충분한 여유가 있도록 $N=20$ 으로 설정하였다. 즉, 예측 시간은 2초이다.

Ubuntu 16.04에서 ROS kinetic 버전을 이용하였고, MPC 비선형 최적화 문제는 오픈 소스 소프트웨어 패키지인 IPOPT (Internal Point Optimizer)^[19]를 사용하여 해결하였다.

4. 시뮬레이션 및 실험 검증

MPC 경로 추종 실험을 시뮬레이터를 이용한 가상 환경과 실제 로봇을 이용하여 실내에서 진행하였다. 본 연구에서는 경로 추종에 많이 사용되는 pure pursuit 알고리즘^[20]을 대조군으로 선정하고, 횡방향 트랙 오차, 방향 오차, 전방 속도 변화로 추종 성능을 판단하고, 추종한 경로, 주행 방식 등을 비교하여

서비스나 물류 모바일 로봇에서 MPC 경로 추종이 적합한지에 대해 평가한다. 참고로, pure pursuit 알고리즘은 별도의 속도 프로파일 생성기가 없으며, 모델의 운동방정식을 이용하지 않는다.

4.1 시뮬레이션

가상 환경은 ROS 기반 Gazebo (7.0.0) 시뮬레이터에서 구현하고 진행하였다^[21]. 가상 환경에서 경로 추종과 네비게이션 시뮬레이션을 각각 수행하였다.

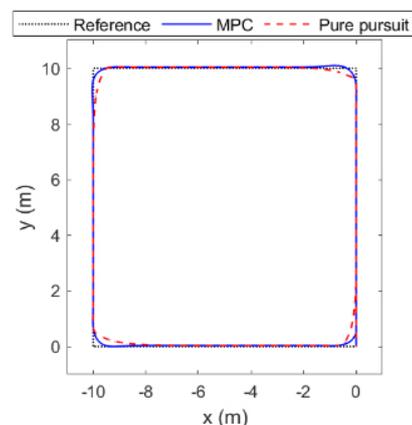
4.1.1 경로 추종

네비게이션 성능을 살펴보기 전에 단독으로 경로 추종 성능만을 파악하기 위해서, 비선형 MPC와 pure pursuit 만을 사용하여 미리 지정해 놓은 사각형의 경로를 잘 추종하는지 확인하였다. 경로 추종에 사용되는 MPC와 pure pursuit의 파라미터는 직진 구간을 동일한 시간에 주파하도록 [Table 2]와 같이 설정하였다.

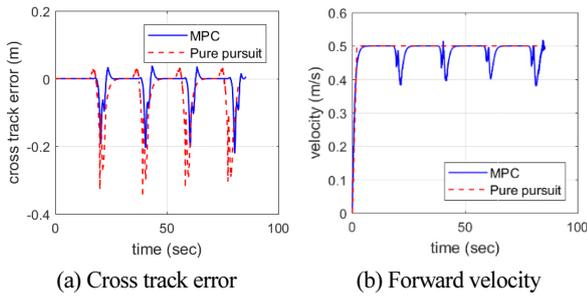
[Fig. 6]은 각 제어기가 사각형 경로를 추종한 결과를 보여준다. 흑색 선은 정확한 기준(reference) 경로이며 각 제어기를 통해 추종하는 경로이다. 청색 선은 MPC로, 적색 선은 pure pursuit 추종 제어로 로봇이 실제 지나간 경로이다. [Fig. 7]은 사각형 경로 추종하는 과정에서 횡방향 트랙에러와 전방 속도를 나타낸 것이다. [Fig. 7(b)]를 보면 MPC는 급격한 코너 구간에 진입하기 전에 미리 감속을 하는 것을 볼 수 있다.

[Table 2] Tracking test parameter MPC and Pure pursuit

Parameter	MPC	Pure pursuit
Ref velocity, v_{ref} (m/s)	0.5	0.5
Acceleration, a (m/s^2)	-	0.25
Max angular velocity, ω (rad/s)	1.0	-
Look ahead distance, L_f (m)	-	0.8



[Fig. 6] Square path tracking



[Fig. 7] Comparison of tracking performance of MPC and Pure-pursuit in square path

[Table 3] Mean of absolute value of tracking performance of MPC and Pure-pursuit in Square-shape

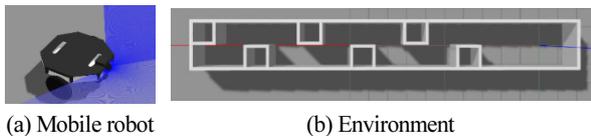
	MPC	Pure pursuit
Cross track error, d (m)	0.0174	0.0437
Orientation error, η (rad)	0.0432	0.0787

[Table 3]에 따르면 MPC가 횡방향 트랙 오차 및 방향 오차가 더 작은 값을 갖는다. MPC의 경우, 코너 구간 진입전에 미리 속도를 줄이고, Out-In-Out으로 코너를 통과하는 반면 pure pursuit은 전방 주시 거리(look ahead distance) 만큼 앞을 보고 경로를 추종하기 때문에 코너를 완벽하게 추종하지 못한다.

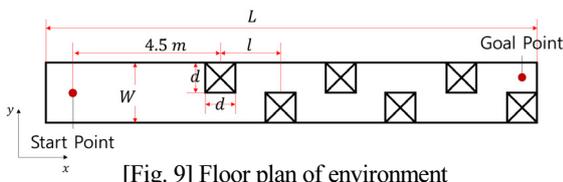
4.1.2 네비게이션

GAZEBO를 이용하여 [Fig. 8]과 같은 형태의 가상 로봇과 주행 환경 환경을 구성하였다. 주어진 가상 환경에 대해 SLAM 알고리즘인 G-mapping을 활용하여 네비게이션에서 사용할 지도를 생성하였다^[22].

환경은 구체적으로 [Fig. 9]의 형상을 가지며, [Table 4]와 같이



[Fig. 8] Virtual mobile robot and environment in GAZEBO



[Fig. 9] Floor plan of environment

[Table 4] Geometric parameters in navigation environment

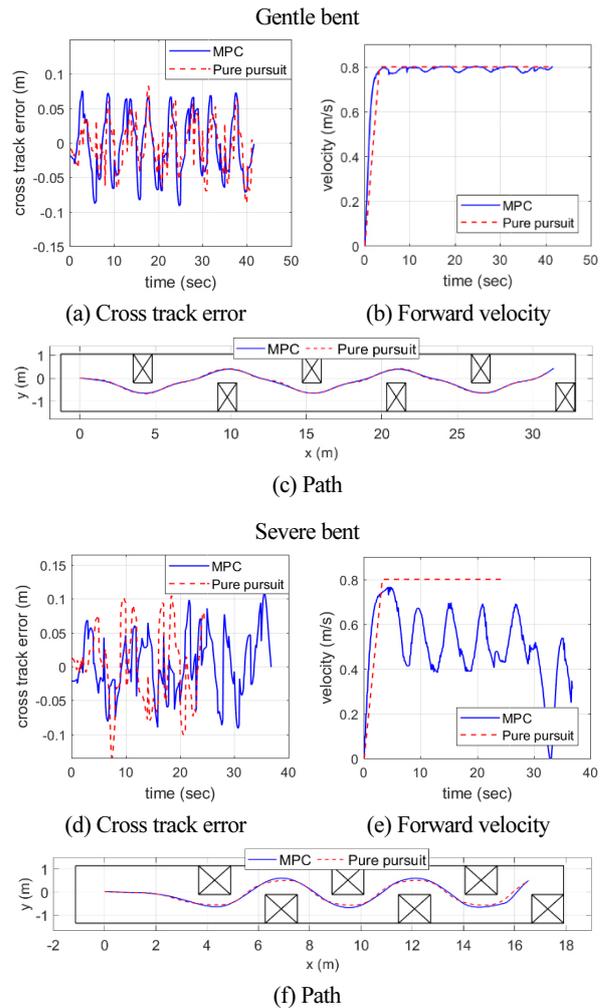
	L (m)	l (m)	W (m)	d (m)	n
Gentle bent	34.1	3.0	2.5	1.25	6
Severe bent	19.0	1.5	2.5	1.25	6

굴곡이 심한 환경과 완만한 환경 두가지로 구성하였다. n 은 장애물의 개수이다. 목표 속도가 저속($v_{ref}=0.8$ m/s)인 경우, 목표 속도가 고속($v_{ref}=1.6$ m/s) 실험, pure pursuit 알고리즘의 전방 주시 거리 변화에 따른 시뮬레이션을 수행하였다.

Case 1. 저속($v_{ref}=0.8$ m/s) 네비게이션

[Table 4]에서 주어진 두가지 환경에 대해서 목표 속도를 0.8 m/s로 지정하여 굴곡이 완만한 환경(gentle bent)와 굴곡이 심한 환경(severe bent)에서 진행하였다. 완만한 환경에서 주행 시간은 MPC, pure-pursuit 모두 42초, 굴곡이 심한 환경에서는 MPC는 37초, pure-pursuit 은 24초 소요되었다.

저속 네비게이션에서는 [Fig. 10(a,d)]의 횡방향 트랙 오차를 볼 때, 굴곡이 심하거나 완만한 경우에 대해서 MPC와 pure-pursuit 두 제어기 모두 경로를 유사하게 잘 추종하는 것을 볼 수 있다. [Table 5]와 [Fig. 10(b,e)]를 보았을 때 복잡해지는



[Fig. 10] Comparison of tracking performance of MPC and Pure-pursuit in low speed navigation

[Table 5] Mean absolute value of tracking performance of MPC and Pure-pursuit in low speed navigation

	Gentle bent		Severe bent	
	MPC	Pure pursuit	MPC	Pure pursuit
Cross track error, d (m)	0.0192	0.0214	0.0371	0.0456
Orientation error, η (rad)	0.0536	0.0428	0.1042	0.0832

경로에 따라서 MPC는 감속을 통하여 경로를 더 정확하게 추종하는 특징을 볼 수 있다.

Case 2. 고속($v_{ref}=1.6$ m/s) 네비게이션

고속 네비게이션은 목표 속도 값을 1.6 m/s로 변경하여 저속 네비게이션과 동일하게 진행하였다. 완만한 환경에서 주행 시간은 MPC, pure-pursuit 모두 23초, 굴곡이 심한 환경에서는 MPC는 18초 소요되었고 경로 추종 결과는 [Table 6], [Fig. 11]와 같다.

Pure pursuit의 경우의 굴곡이 심한 환경에서 1.6 m/s의 속도로 주행을 요구할 경우, [Fig. 12]와 같은 위치에서 충돌이 발생하였다. 전방 주시 거리가 0.8 m인 경우 굴곡이 심한 환경에서 목표점에 도달하지 못하였다.

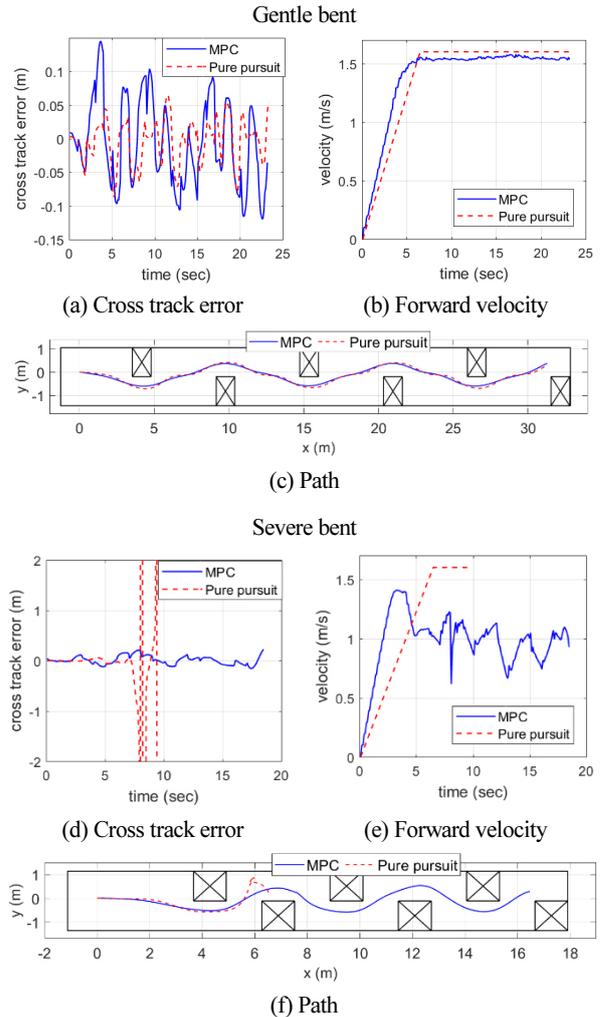
반면 MPC는 목표 속도가 1.6 m/s 인 경우, 횡방향 트랙 오차가 크게 발생하게 되면, 속도를 줄여서 주행하는 경향을 보인다. 비록 목표 속도까지 도달하지 못하더라도 안정적으로 주행을 완료하여 목표점에 도달하였다.

Pure-pursuit 경로 추종의 경우, 전방 주시 거리가 가장 중요한 파라미터가 된다. 따라서, pure pursuit의 전방 주시 거리만 0.4-1.6 m까지 0.2 m 단위로 변경하여 굴곡이 심한 환경에서 고속($v_{ref}=1.6$ m/s) 네비게이션을 수행하였으며, [Fig. 13]은 이에 따른 이동경로를 보여준다.

전방 주시 거리가 1.0 m인 경우에만 로봇이 목표 지점에 도달하였고, 1.0 m보다 짧을 경우 속도에 비해서 너무 가까운 경로를 보게 되어 경로 변화에 빠르게 반응하지 못하고 바깥쪽 벽에 충돌하였다. 전방주시거리가 1.0 m 보다 길 경우 너무 먼 위치를 봐서 너무 이른 시점에서 회전을 시작하여 경로의 안쪽 벽에 충돌하였다. 이는 추종해야 하는 경로의 굴곡 정도와 전방 주시 거리, 요구 목표 속도가 상관 관계를 가질 수밖에 없

[Table 6] Mean absolute value of tracking performance of MPC and Pure-pursuit in high speed navigation

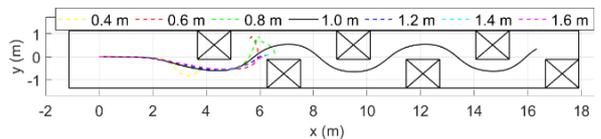
	Gentle bent		Severe bent	
	MPC	Pure pursuit	MPC	Pure pursuit
Cross track error, d (m)	0.0428	0.0223	0.0691	-
Orientation error, η (rad)	0.1162	0.0643	0.2324	-



[Fig. 11] Comparison of tracking performance of MPC and Pure-pursuit in high speed navigation



[Fig. 12] Collision location of high speed pure pursuit tracking



[Fig. 13] Path with respect to look ahead distances

는 pure pursuit 알고리즘에서는 당연한 결과일 수 있으며, 이를 개선하는 연구들도 존재한다^[23].

시뮬레이션 결과를 요약하면, pure pursuit은 주어진 속도로 주행하므로, 경로를 완주한 경우에는 시간이 적게 소요되었지만, 다양한 조건에서 고정된 파라미터로 효과적으로 주행하지 못

하였다. 반면에 MPC의 경우, 목표 속도를 높게 설정하면 지도에 따라 안정된 구간에서만 높은 속도로 주행하고 굴곡이 심한 구간에서는 저속으로 안정적으로 주행하므로 고정된 과라미터로 다양한 조건에서 효과적으로 주행할 수 있다.

4.2 실제 환경 네비게이션

[Fig. 14], [Fig. 15]와 같이 실제 공간에 장애물 3개를 부등간격으로 배치하고 이 환경에서 네비게이션을 수행하였다.

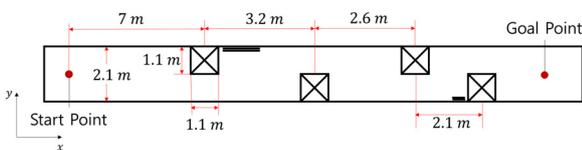
실제 주행 네비게이션은 목표 속도를 0.8-1.4 m/s까지 0.2 m/s 단위로 변경하여 속도에 따른 주행가능성 및 MPC와 Pure pursuit의 성능을 비교하였다.

실제 환경에서의 로봇 구동은 가상 환경에서 구동하는 것과 차이가 발생한다. 주행하는 노면에 상태, 구동륜의 슬립, 캐스터로 인한 회전 마찰 등 다양한 외란이 발생하였다. 시뮬레이션에서 기준으로 설정했던 pure pursuit의 전방 주시 거리 0.8 m로는 특정 속도에서 주행이 불가능한 경우가 많이 발생하여, 실제 실험 시에는 1.0 m로 설정하였다.

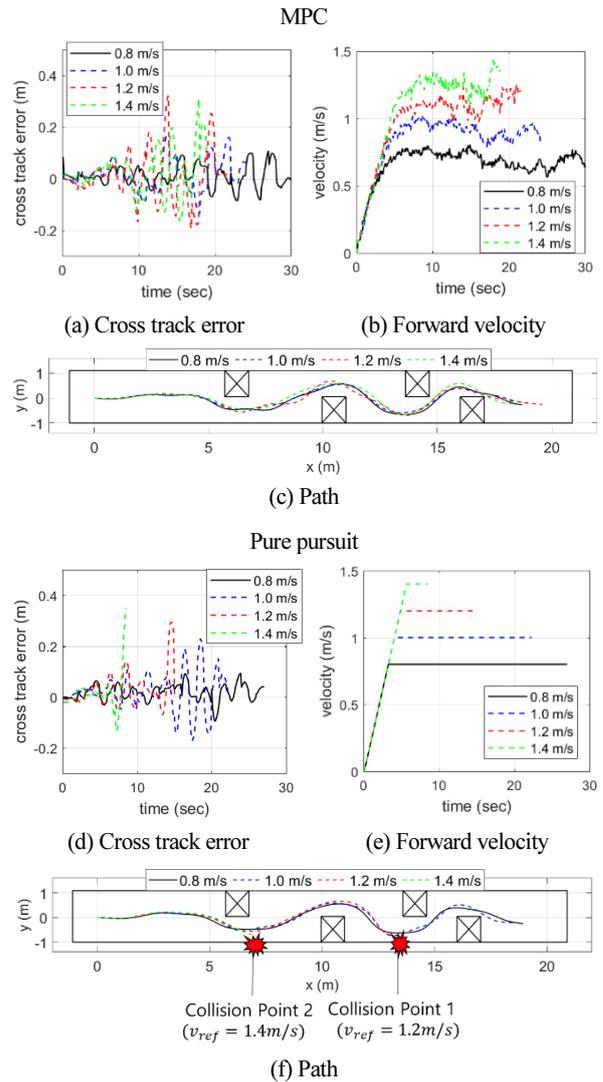
Pure pursuit는 가상 환경에서의 시뮬레이션과 동일하게, 주행을 완료할 경우 MPC보다 빠른 시간 내에 목표지점에 도달하였으나, 속도가 증가함에 따라 추종 성능 저하가 크게 나타나며 [Fig. 16(c)]에서 보는 바와 같이 1.2 m/s와 1.4 m/s에서는 주행을 완료하지 못하였다. 1.2 m/s의 속도로 주행시에는 [Fig. 16(f)]의 지점 1에서 충돌이 일어났고 1.4 m/s의 속도로 주행시에는 지점 2에서 충돌이 일어났다. 추가적으로 1.3 m/s의 속도로 실험해 본 결과 지점 1과 지점 2 중간에서 충돌이 일어났다.



[Fig. 14] Real experiment environment



[Fig. 15] Floor plan of real environment



[Fig. 16] Tracking performance in real experiment

[Table 7] Mean absolute value of MPC tracking performance in real experiment

	0.8 m/s	1.0 m/s	1.2 m/s	1.4 m/s
Cross track error, d (m)	0.0316	0.0465	0.0668	0.0685
Orientation error, η (rad)	0.1001	0.0938	0.1623	0.1630

[Table 8] Mean absolute value of Pure-pursuit tracking performance in real experiment

	0.8 m/s	1.0 m/s	1.2 m/s	1.4 m/s
Cross track error, d (m)	0.0288	0.0585	0.0757	-
Orientation error, η (rad)	0.0744	0.1245	0.1642	-

[Table 7], [Table 8]을 정리하면, MPC의 경우는 목표 속도가 변함에 따라 추종 성능의 저하가 발행하지만, pure pursuit에 비해 크지 않았으며 실험한 모든 목표 속도에서 충돌없이 경로를 잘 추종하는 것을 볼 수 있다. 또한 시뮬레이션에서 살펴본 것처럼 MPC는 비인과성(non-causality)을 지니므로, 코너 전에 미리 감속하여 횡방향 트랙 오차를 최소화하는 경향을 보였다. 이러한 경향은 비용 함수의 가중치를 어떻게 설정하는지에 따라 달라질 수 있다. 본 연구에서는 복잡하고 유동적인 환경에서 경로를 잘 추종하기 위해 목표 속도와 관련된 가중치보다 횡방향 트랙 오차와 관련된 가중치 크게 주었기 때문에 위와 같은 결과가 도출되었다고 볼 수 있다.

5. 결 론

본 연구는 차동 구동형 모바일 로봇의 네비게이션을 위해 경로 추종 알고리즘으로 비선형 MPC를 적용하였을 때의 장점을 보이기 위해 가상 환경과 실제 환경에서 네비게이션 실험을 진행하였다. 경로 추종 문제에 MPC를 적용하기 위해서 횡방향 트랙 오차, 방향 오차를 추가한 상태 변수를 지정하여, 로봇의 비선형 운동방정식을 도출하였다. 또한, 다양한 성능을 포함한 비용 함수와 제약 조건을 통해 최적화 문제를 정의하였다. 본 연구에서는 ROS의 네비게이션 스택을 활용하여 로컬 플래너 부분을 MPC 경로 추종기로 대체하였다. MPC의 최적화 가중치 중 횡방향 트랙 오차와, 방향 오차에 우선순위를 두고 가상 환경과 실제 환경에서 pure pursuit 경로 추종 알고리즘과 비교하여 실험을 진행하였다. 실험은 저속과 고속, 완만한 환경과 굴곡이 심한 환경에서 진행하였다. 두 알고리즘 모두 저속과 고속에서 완만한 환경 경로를 잘 추종했으나, 굴곡이 심한 환경을 고속으로 주행한 경우 MPC는 굴곡의 정도에 따라 완만한 경우 목표 속도로, 심할 경우 저속으로 속도 조절하여 안정적으로 경로를 통과하는 반면 pure pursuit은 모든 경로에 전방 주시 거리만큼의 지점을 바라보며 목표속도로 주행하기 때문에 경로 추종 중 충돌이 발생하였고, 이는 전방 주시 거리를 변경함으로 해결할 수 있다. 하지만 변경한 전방 주시 거리로 저속일 때 혹은 완만한 환경에서 주행 시 충돌이 발생하였다. 즉, MPC는 다양하고 유동적인 환경에서 설정 값들을 변경하지 않고 최적화된 주행이 가능함을 알 수 있다. 이는 실제 모바일 로봇이 쓰이는 사람이 많은 곳과 물류창고와 같은 유동적으로 변하는 환경에서 안정적인, 효과적인 주행이 가능하다는 것을 의미한다. 또한 경로 추종 성능이 우수하고, 연산 시간의 일관성, 감축이 있으며 최적화 가중치 설정이 비교적 직관적이고 쉬워 모바일 로봇의 원하는 성능을 유도하기 용이하다.

References

- [1] Y. Zhang, D. Hong, J. H. Chung, and S. A. Velinsky, "Dynamic model based robust tracking control of a differentially steered wheeled mobile robot," *1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, Philadelphia, USA, pp. 850-855, 1998, DOI: 10.1109/ACC.1998.703528.
- [2] F. Dušek, D. Honc, and P. Rozsival, "Mathematical model of differentially steered mobile robot," *18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, pp. 221-229, 2011, [Online], <https://www.uiam.sk/pci11/data/abstracts/023.html>.
- [3] H.-S. Choi, D.-I. Kim, and J. B. Song, "Simultaneous path tracking and orientation control for three-wheeled omnidirectional robots," *Journal of Korea Robotics Society*, vol. 10, no. 3, pp. 154-161, Sept., 2015, DOI: 10.7746/jkros.2015.10.3.154.
- [4] C. Woo, M.-U. Lee, and T.-S. Yoon, "Robust Trajectory Tracking Control of a Mecanum Wheeled Mobile Robot Using Impedance Control and Integral Sliding Mode Control," *Journal of Korea Robotics Society*, vol. 13, no. 4, pp. 256-264, Dec., 2018, DOI: 10.7746/jkros.2018.13.4.256.
- [5] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566-580, May, 2007, DOI: 10.1109/TCST.2007.894653.
- [6] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628-647, July, 2015, DOI: 10.1002/oca.2123.
- [7] H. LIN, *HyphaROS MiniCar (1/30 Scale MPC Racing Car)*, [Online], https://github.com/Hypha-ROS/hypharos_minicar, Accessed: Aug 16, 2018.
- [8] F. Kuhne, W. F. Lages, and J. G. da Silva Jr, "Model predictive control of a mobile robot using linearization," *APS/ECM*, pp. 525-530, 2004, [Online], http://www.ece.ufrgs.br/~fetter/mechr-ob04_553.pdf.
- [9] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-Tracking Control of Mobile Robot Systems Incorporating Neural-Dynamic Optimized Model Predictive Approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 740-749, June 2016, DOI: 10.1109/TSMC.2015.2465352.
- [10] I. Maurović, M. Baotić, and I. Petrović, "Explicit Model Predictive Control for trajectory tracking with mobile robots," *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Budapest, Hungary, pp. 712-717, 2011, DOI: 10.1109/AIM.2011.6027140.
- [11] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," *ICRA Workshop on Open Source Software*, June, 2009, [Online], <https://www.semanticscholar.org/paper/ROS%3A-an-open-source-Robot-Operating-System-Quigley-Conley/d45eace8b2e047306329e5dbfc954e6dd318ca1e>.

- [12] P. Corke, "Mobile robot vehicle," *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*, 2nd ed., Springer Publishing, 2017, ch.4, pp.109-111, DOI: 10.1007/978-3-642-20144-8_4.
- [13] J. E. Normey-Rico, I. Alcalá, J. Gómez-Ortega, and E. F. Camacho, "Mobile robot path tracking using a robust PID controller," *Control Engineering Practice*, vol. 9, no. 11, pp. 1209-1214, Nov, 2001.
- [14] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical Model Predictive Control framework for autonomous ground vehicles," *2008 American Control Conference*, Seattle, USA, pp. 3719-3724, 2008, DOI: 10.1109/ACC.2008.4587072.
- [15] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, USA, pp. 300-307, 2010, DOI: 10.1109/ROBOT.2010.5509725.
- [16] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, Mar., 1997, DOI: 10.1109/100.580977.
- [17] A. Koubaa, H. Bennaceur, I. Chaari, S. Trigui, A. Ammar, M.-F. Sriti, and M. Alajlan, "Introduction to mobile robot path planning," *Robot Path Planning and Cooperation Studies in Computational Intelligence*, Springer, 2018, ch.1. pp. 3-12, DOI: 10.1007/978-3-319-77042-0_1.
- [18] S. M. LaValle, "Search for feasible plans," *Planning algorithms*, Cambridge university press, 2006, ch. 2, pp.35-39, DOI: 10.1017/CBO9780511546877.
- [19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25-57, Mar., 2006, DOI: 10.1007/s10107-004-0559-y.
- [20] J. Morales, J. L. Martinez, M. A. Martinez, and A. Mandow, "Pure-Pursuit Reactive Path Tracking for Nonholonomic Mobile Robots with a 2D Laser Scanner," *EURASIP Journal on Advances in signal Processing*, Mar., 2009, DOI: 10.1155/2009/935237.
- [21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Sendai, Japan, pp. 2149-2154, 2004, DOI: 10.1109/IROS.2004.1389727.
- [22] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34-46, Feb., 2007, DOI: 10.1109/TRO.2006.889486.
- [23] H. Wang, X. Chen, Y. Chen, B. Li, and Z. Miao, "Trajectory Tracking and Speed Control of Cleaning Vehicle Based on Improved Pure Pursuit Algorithm," *2019 Chinese Control Conference*, Guangzhou, China, pp. 4348-4353, 2019, DOI: 10.23919/ChiCC.2019.8865255.



최재완

2019 서울과학기술대학교 기계시스템디자인 공학과(학사)

2019~현재 서울과학기술대학교 기계설계 로봇공학과(석사)

관심분야: 로보틱스



이건희

2018 서울과학기술대학교 기계시스템디자인 공학과(학사)

2020 서울과학기술대학교 기계설계로봇 공학과(석사)

2020~현재 시스콘 연구원

관심분야: 로보틱스, 강화학습, 딥러닝



이치범

1998 서울대학교 기계설계학과(학사)

2000 서울대학교 기계설계학과(석사)

2010 Mechanical Science and Engineering, University of Illinois, Urbana-Champaign(박사)

2011~현재 서울과학기술대학교 부교수

관심분야: Mobile robots, Autonomous driving, Robust control