International Journal of Control, Automation and Systems 17(11) (2019) 2850-2861
http://dx.doi.org/10.1007/s12555-019-9091-y
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Obstacle Avoidance Path Planning based on Output Constrained Model Predictive Control

Ji-Chang Kim, Dong-Sung Pae, and Myo-Taeg Lim*

Abstract: Image processing and control technologies have been widely studied and autonomous vehicles have become an active research area. For autonomous driving, it is essential to generate a safe obstacle avoidance path considering the surrounding environment. This paper devised an algorithm based on a real-time output constrained model predictive control for obstacle avoidance path planning in high speed driving situations. The proposed algorithm was compared with the normal model predictive control algorithm by simulation, including operation times to verify robustness for high speed driving situations. We used the ISO 2631-1 comfort level standard to quantify driver comfort fo r both cases.

Keywords: Comfort level, model predictive control, obstacle avoidance, path planning, vehicle dynamics.

1. INTRODUCTION

Ongoing development of image processing technologies, vehicle-to-vehicle communication and control technologies have strongly supported advanced driver assistance system (ADAS) and autonomous driving system (ADS) research. Consequently, various lane keeping assistance, active cruise control and autonomous emergency braking are in ADAS related technologies. ADAS and ADS provide users with convenience and comfort while enhancing user safety and efficiency. The most important ADAS and ADS consideration is safety. Therefore, this paper discusses obstacle avoidance path planning for autonomous driving situations and evaluates safety assurance for generated obstacle avoidance paths.

An autonomous driving vehicle must recognize obstacle vehicles or pedestrians on the global path and either steer or brake to generate a safe path to the final target point. Thus, ADAS and ADS algorithms that can control vehicle steering and braking are essential. Many path planning algorithms have been proposed for ADASs and ADSs, with the A-star and potential field algorithms being the most common currently employed.

Carroll *et al*. [1] proposed an A-star theory based path planning algorithm to avoid obstacles such as reefs in the water. Wang *et al*. [2] proposed a path planning algorithm incorporating an improved A-star technique for automated guided vehicles (AGV). They suggested to remove the edge of the global map to improve performance. Consequently, the proposed algorithm generated obstacle avoidance paths much faster than the normal A-star algorithm for multiple AGVs. However, the A-star algorithm has the disadvantage that a complete new path is generated for each dynamic obstacle avoidance.

Hwang *et al*. [3] proposed a potential field approach to generate an obstacle avoidance path by assigning an electrostatic potential to each obstacle. The path generated by potential field algorithm does not reflect the vehicle dynamics. Rasekhipour *et al*. [4] incorporated a vehicle dynamics model into the potential field algorithm. The resultant model predicted vehicle behaviors and obtained an optimal path considering vehicle dynamics. In contrast to the potential field algorithm, obstacles and road environment were considered as constraints. However, potential field algorithms may fail to generate the whole path due to local minima problems. When the vehicle drives the global path and reaches a local minimum, it cannot get out of that position, misjudges the position as the destination, and stops control.

Several previous obstacle avoidance path planning studies have used model predictive control (MPC) to overcome these disadvantages that occur in A-star and potential field algorithm. [5–8]. The basic MPC process is as follows. First, configure the system model to be controlled. Optimized outputs are obtained at each step from previously defined MPC input and state variables, and then assigned

Manuscript received February 8, 2019; revised May 24, 2019; accepted June 7, 2019. Recommended by Associate Editor Changsun Ahn under the direction of Editor Keum-Shik Hong. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. NRF-2016R1D1A1B01016071).

Ji-Chang Kim is with the Department of Automotive Convergence, Korea University, 1 Anam-dong, Sungbuk-gu, Seoul 02841, Korea (e-mail: jichang0619@korea.ac.kr). Dong-Sung Pae and Myo-Taeg Lim are with the Department of Electrical Engineering, Korea University, 1 Anam-dong, Sungbuk-gu, Seoul 02841, Korea (e-mails: {paeds915, mlim}@korea.ac.kr).
* Corresponding author.

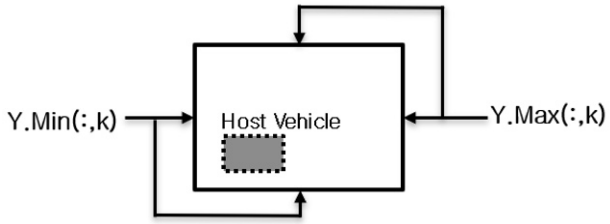


Fig. 1. Output constraint box for the host vehicle.

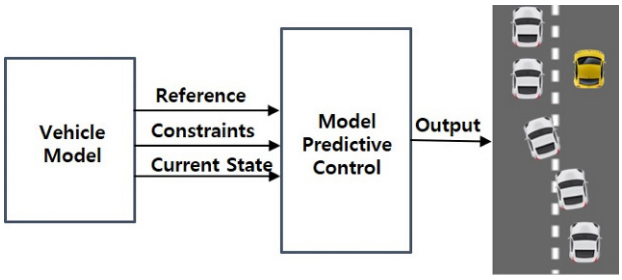


Fig. 2. Generalized model predictive control architecture.

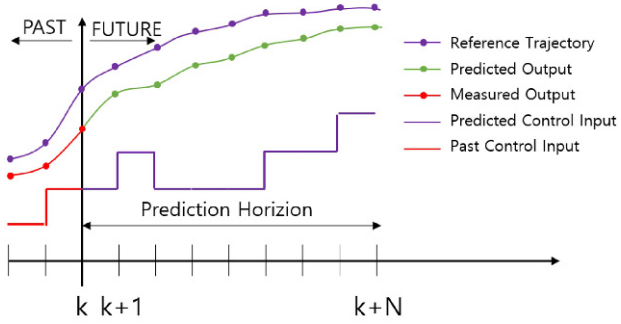


Fig. 3. Model predictive control.

to the system model. Predicted values for the next step are obtained based on these output values and input and state values.

By incorporating feedback control, MPC is robust against time delay [9–11], and has been widely used to model chemical process plants because it can control multiple variables and impose variable constraints [12]. Several studies have applied MPC to vehicle steering control, but conventional MPC computation speed is slow in multi-input multi-output system, such as vehicle models [13–16]. Slow computation makes it difficult to obtain timely output values in high speed driving situations, and hence difficult or impossible to generate timely obstacle avoidance paths.

To solve conventional MPC disadvantages, this paper proposes a method that increases computation speed by imposing constraints on output values and removing paths that do not satisfy the conditions. An output constraint box is formed in the host vehicle by applying output limits, as shown in Fig. 1, and input and state values that do not satisfy the output constraint are not calculated. Thus, the solver can transform non-convex problems into convex problems, significantly improving computation speed.

We compared operation speed for the proposed and conventional model predictive control algorithms using MATLAB®simulation and CarSim. Safety of the generated path was evaluated by applying ISO 2631-1, which evaluates comfort level based on vehicle accelerations in lateral, longitudinal, and vertical directions when driving the generated path [17, 18].

The remainder of this paper is organized as follows: Section 2 introduces MPC principles and designing a vehicle model and MPC for path planning. Section 3 presents the comfort level concept to evaluate safety and time to collision for decision making. Section 4 presents simulation results from MATLAB®and CarSim. Finally, Section 5 summarizes and concludes the paper.

2. MODEL PREDICTIVE CONTROL FOR OBSTACLE AVOIDANCE

This section introduces MPC concept and basic theory, as well as revised MPC methods for high speed driving situations and vehicle modeling to generate an obstacle avoidance path.

2.1. Model predictive control concept

Model predictive control is relatively robust to time delay and can simultaneously process multiple input and state values. Thus, MPC has become a standard technique for multiple variable control and economically controls large scale processes. Plant models for multivariate processes have mutual variable interferences, which can significantly limit performance in PID control, a representative modern control technique. However, MPC allows variable upper and lower limit to be set as constraint, allowing plant processes to be economically executed.

Basically, MPC aims to generate optimal control inputs for a vehicle model [19]. Thus, MPC constructs the system model, satisfies constraint conditions until the prediction horizon, and then generates a control input that minimizes the weighted objective function [20]. Fig. 3 shows control input from point k to $k+N$, calculated to drive to the destination [21]. We designed the path planning algorithm and vehicle models to obtain the final output value as the obstacle avoidance path.

The MPC system allocates vehicle velocity, acceleration, and obstacle vehicle position as inputs, and then obtains the host vehicle velocity and position. The obtained outputs are substituted into the vehicle model to generate an obstacle avoidance path.

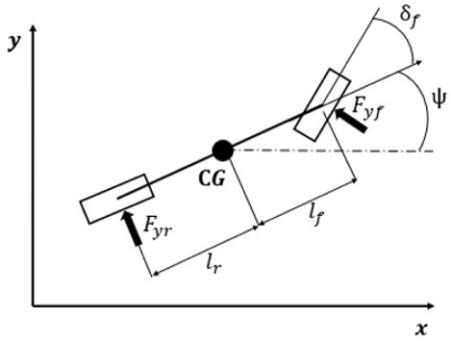


Fig. 4. Two degrees of freedom bicycle model.

2.2. Host vehicle modeling

We used a 2 degrees of freedom (2DOF) bicycle model for the MPC host vehicle system [22–24], with the intention to develop an obstacle avoidance path for high speed driving situations. Since linearized models have approximately 2-fold speed advantage over non-linearized models, without significant differences in the generated path, we employed a linearized vehicle model to ensure real-time obstacle avoidance path calculation. The host vehicle MPC model can be expressed by the state space equation,

$$\dot{X} = AX + BU, \tag{1}$$

where $X = [x,\ v_x,\ y,\ v_y]^T$ is state of vehicle and $U = [a_x\ a_y]^T$ is acceleration of vehicle.

Since the host vehicle model must predict lateral and longitudinal vehicle behavior up to the prediction horizon, we need to convert the continuous time model into discrete time state space with sampling time T_s. The discrete time model is

$$X_{k+1} = A_d X_k + B_d U_k, \tag{2}$$

where X_k is state of vehicle at k-th, $A_d = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and $B_d = \begin{bmatrix} \frac{1}{2}T_s^2 & 0 \\ 0 & T_s \\ \frac{1}{2}T_s^2 & 0 \\ 0 & T_s \end{bmatrix}$.

The obtained state space equation is executed in the vehicle model and calculates final outputs from the inputs obtained from MPC at the given prediction horizon,

$$Y_k = C_d X_k, \tag{3}$$

where $Y_k = [x_k,\ y_y]^T$ is the predicted obstacle avoidance path from the current point, k, to the prediction horizon, $k+N$, and

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T.$$

Fig. 5. Obstacle vehicle safety margin.

2.3. Obstacle vehicle modeling

An obstacle vehicle model is also required for MPC when setting collision avoidance path constraints for the host vehicle. Obstacle vehicle information is assumed to be acquired through a camera or sensor attached to the host vehicle, with relative distance and velocity, and obstacle available with no error or disturbance. We constructed the obstacle vehicle model as a constant speed model.

Fig. 5 shows the obstacle vehicle safety margin to ensure minimum safe distance in actual driving. The safety margin is proportional to the obstacle vehicle size.

2.4. Real-time output constraints

Normally, MPC constructs the system model and predicts future system operation. MPC has advantages for multiple variable input and output systems and has often been used for chemical plant systems. To improve controller performance, MPC feeds back the prediction outputs to obtain an optimal output, which is relatively robust to time delay since MPC can relatively easily adjust variable values by setting input and state constraints.

This paper proposes a real-time output constrained model predictive control (ROCMPC) that imposes output variable constraints to compensate computation speed, the main disadvantage of conventional MPC.

The basic ROCMCP theory for operation time enhancement is as follows. When the obstacle avoidance path algorithm is executed for conventional MPC, the MPC solver objective function (J) is

$$\min_{a=\{1,\cdots,P\}} \min_{u_1,\cdots,u_N} J_a(y,u) = \sum_{k=1}^{N} \|y_k - y_{a,ref,k}\|_2^2 + \|u_k\|_2^2, \tag{4}$$

where P denotes possible paths of obstacle avoidance within MPC horizon step N and $y_{a,ref}$ is optimal path of a-th path. MPC solver objective function is non-convex, i.e., two or more solutions may exist with suboptimal output.

Fig. 6 shows there are two avoidance paths that MPC could select for obstacle avoidance on a straight road (the simulation environment). Thus, the host vehicle calculates both paths in its entirety and selects one (usually the shortest path) to proceed.

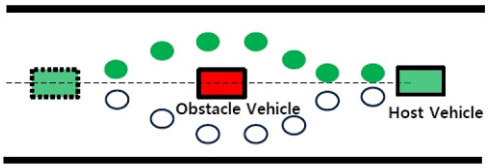


Fig. 6. Host vehicle non-convex problem.

We propose an additional output constraint to convert the problem to convex. Hence a single control input is produced when solving the MPC objective function. After the solution is put into the model of the host vehicle, the position and velocity of the host vehicle for the collision avoidance up to the prediction horizon are calculated as the output values.

$$\min_{u_1,\cdots,u_N} J(y,u) = \sum_{k=1}^{N} \|y_k - y_{ref,k}\|_2^2 + \|u_k\|_2^2. \tag{5}$$

Various vehicle constraints can be applied, including maximum acceleration or deceleration, maximum or minimum velocities, i.e.,

$$u_{min} \leq u_{k+i} \leq u_{max}, \tag{6}$$

and

$$\Delta u_{min} \leq \Delta u_{k+i} \leq \Delta u_{max}, \tag{7}$$

respectively, depending on the actual driving situation.

Adding the constraint condition increases the required level of MPC solver calculation to solve the objective function. However, the output constraints can exclude possible driving paths, hence reducing the required computation.

This is the main ROCMPC concept to improve MPC operation time. Host vehicle output constraints are added every step depending on the obstacle vehicle position and velocity. Upper and lower lateral and longitudinal direction limits are applied to the real-time feedback from the MPC, hence the output constraint box is constructed according to the host vehicle position and velocity. Thus, host vehicle position remains within the output constraint box at every step.

$$Y_{\min,k} = \begin{bmatrix} x_k - C_{\min,x} \\ y_k - C_{\min,y} \end{bmatrix}, \quad Y_{\max,k} = \begin{bmatrix} x_k + C_{\max,x} \\ y_k + C_{\max,y} \end{bmatrix}, \tag{8}$$

where $C_{\min}$ and $C_{\max}$ are constrained conditions. The process to construct the output constraint box is shown in Fig. 7. The initial box size only considers the host vehicle size, setting initial output constraint box to 2.5× host vehicle width and 3.5× length. The host vehicle then follows the reference path generated previously. the obstacle

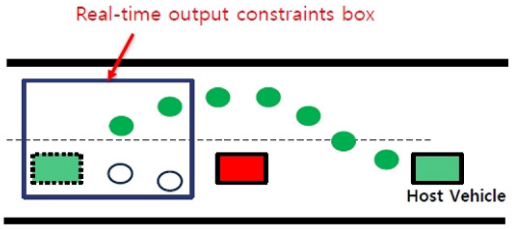


Fig. 7. Host vehicle real-time output constraints on avoidance path.

vehicle enters the prediction horizon, the host vehicle executes the obstacle avoidance path algorithm by changing longitudinal and lateral velocities.

The constraint box size can change depending on the obstacle vehicle position and velocity. Conventional MPC would calculate both obstacle avoidance paths from the solver. However, the proposed ROCMPC considers the obstacle vehicle location and modifies the output constraint box, which excludes paths from calculation within the ROCMPC solver unless the constraints are satisfied.

The other paths are not calculated since it would require the host vehicle to violate the constraint conditions. Therefore, the output constraint box converts the non-convex problem to a convex problem, considerably reducing the required MPC solver computation to obtain the host vehicle control input. The proposed method adds constrained conditions to the result of the conventional MPC, and does not hinder the inherent stability of MPC and determines solution. The ROCMPC output comprises control input for the previously established vehicle model to obtain control outputs for the host vehicle, i.e., host vehicle lateral and longitudinal velocity and position to avoid the obstacle vehicle.

We developed an algorithm to implement ROCMPC based obstacle avoidance for high speed driving situations. Section 4 compares performance for the proposed algorithms with normal MPC algorithms.

3. DETERMINING AVOIDANCE PATH SAFETY

The primary purpose of ISO 2631 is to define methods of quantifying whole-body vibration in relation to human comfort. Vibration is often complicated, containing many frequencies, occurring in several directions simultaneously, and changing over time [26, 27]. The primary quantity of vibration magnitude is acceleration.

3.1. Comfort level

Frequency is first evaluated to define the comfort level. Vibration measurement includes weighted root-

Table 1. Comfort level based on ISO 2631-1.

Overall acceleration (m/s^2)	Perception
$a_w < 0.315$	Comfortable
$0.315 < a_w < 0.63$	A little uncomfortable
$0.5 < a_w < 1$	Fairly uncomfortable
$0.8 < a_w < 1.6$	Uncomfortable
$1.25 < a_w < 2.5$	Very uncomfortable
$2.5 < a_w$	Extremely uncomfortable

mean-square (RMS) acceleration (a_k) of k-aixs,

$$a_{wk}^2 = \frac{1}{T}\int_0^T a_k^2(t)\,dt, \tag{9}$$

where T is the measurement duration in seconds.

We used the frequency weighted curve obtained from the experiment. Two frequency weightings, W_k (z direction) and W_d (x and y directions) relate to comfort. Frequencies in all directions are combined once the weights are obtained to provide the total vibration of weighted rms acceleration,

$$a_w = \sqrt{W_d^2 a_{wx}^2 + W_d^2 a_{wy}^2 + W_k^2 a_{wz}^2}, \tag{10}$$

where a_{wx}, a_{wy}, and a_{wz} are the weighted RMS acceleration; and W_d, and W_k are constants for the (orthogonal) x, y, and z directions, respectively [18].

The frequency weightings are applied for seated persons since we are simulating driver situations. Thus,

- x axis: $W_d = 1.4$.
- y axis: $W_d = 1.4$.
- z axis: $W_k = 1.0$.

Table 1 shows the corresponding comfort levels for various a_w, indicating likely driver reactions to total vibration following ISO 2631-1 [18].

Although the avoidance path can be generated such that the host vehicle can avoid the obstacle, path safety is also extremely important. It is not easy to judge path safety objectively. In actual driving situations, the driver smoothly avoids the obstacle as much as possible when steering, i.e., they do not feel discomfort when steering. If a sharp steering change was made, the driver senses the acceleration and is able to mitigate the effect. However, this does not apply for ADAS or ADS.

Therefore, we assessed the safety of generated obstacle avoidance paths (from the proposed ROCMPC or conventional MPC algorithm) using the ISO 263-1 based comfort level index (see Table 1). Thus, the obstacle avoidance path stability was determined using the driver's comfort level index.

Table 2. ISO 2631-1 quantification.

Overall acceleration (m/s^2)	Perception	Score
$a_w < 0.315$	Comfortable	10
$0.315 < a_w < 0.63$	A little uncomfortable	8
$0.5 < a_w < 1$	Fairly uncomfortable	6
$0.8 < a_w < 1.6$	Uncomfortable	4
$1.25 < a_w < 2.5$	Very uncomfortable	2
$2.5 < a_w$	Extremely uncomfortable	0

3.2. Comfort Level Quantification

The comfort level discussed in Section 3.1 used quantitative a_w, but the comfort indexes are defined qualitatively. Therefore, scores were assigned to the comfort index, as shown in Table 2. The score difference of each section was set at two points since we want the weighting for each section to be the same.

In Section 4, we apply this scoring to the simulation. The comfort level score was assessed for each simulation step, e.g., 120 comfort level scores were obtained if the simulation ran for 120 steps. The comfort level score for a simulation case was taken as the mean score, and we compared average comfort level for the whole avoidance path using the normal MPC and proposed ROCMPC algorithms.

3.3. Braking control using time to collision and overlap

We also simulated obstacle collision prevention through braking as well as avoidance by steering [28]. The host vehicle control system must decide whether to steer or brake to avoid collision when it detects an obstacle, i.e., decision making. The most frequently used indicator for decision making is time to collision (TTC) [29, 30], which is the expected collision time considering the relative speed and distance between the host and obstacle vehicles. Small TTC means the collision will occur soon, and if the obstacle vehicle velocity exceeds the host vehicle, TTC$= -\infty$ since no collision would occur.

We also considered overlap in the decision making process. Overlap with the obstacle vehicle is an important value for steering avoidance. If the overlap range is wide, steering avoidance will be difficult in situations where collision is imminent. However, if TTC and overlap range are both small, steering avoidance may be possible.

Section 4.4.3 discusses TTC and overlap research, to determine if the host vehicle should steer or brake to avoid collision. Obstacle avoidance by steering is prioritized and braking control is implemented only if that is not possible. Table 3 shows a simple truth table.

Table 3. Decision making table.

	Narrow overlap	Wide overlap
Greater than TTC	Steering	Steering
Less than TTC	Steering	Braking

4. SIMULATION RESULTS

We implemented the proposed algorithm in MATLAB®, and simulations were performed on an Intel® Core™ i7-4770 CPU @ 3.40GHz, 4 GB RAM. The road environment was straight, 1 km long, 6 m wide two-lane. Obstacle vehicle behavior followed four representative scenarios similar to actual road driving situations.

4.1. Simulation environment setup and scenarios

The host vehicle model followed the 2DOF bicycle model (see Fig. 4). The linearized and non-linearized 2DOF bicycle model showed no significant differences between obstacle avoidance paths, hence we used the linearized model to generate the obstacle avoidance path for high speed driving situations, minimizing operation time for control input obtained from the MPC solver. Output values are the obstacle vehicle lateral and longitudinal velocity, and position. We used three vehicle size categories for the simulation, as shown in Fig. 8: passenger car, luxury sedan, and commercial van. Vehicle size details were modeled by assigning values extracted from CarSim.

We modelled the obstacle as constant velocity or stationary, with behavior for moving vehicles set as straight forward, cut in, or cut out. This reflected typical real obstacle vehicle behavior on a straight road. Thus, we considered four obstacle vehicle behavior scenarios, as shown in Fig. 9.

Fig. 8. Obstacle vehicle types considered.

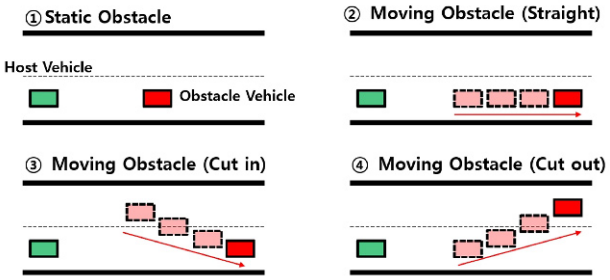


Fig. 9. Obstacle vehicle behavior patterns considered.

We also considered a curved road situation to test proposed ROCMPC algorithm robustness. We compared simulation operation time required to generate the obstacle avoidance path and comfort level scores for each scenario for conventional MPC and ROCMPC algorithms.

4.2. Operation time comparison

All simulation conditions were kept the same between the proposed ROCMPC and conventional MPC algorithms, to ensure a fair assessment of validity and performance. We implemented both algorithms in MATLAB®. Host vehicle, obstacle vehicle, and road environment were the same. We performed simulations for the four scenarios discussed above (see Fig. 9), first assuming a single obstacle vehicle, and Section 4.4 reports additional simulations with increased number of obstacle vehicles. All data, such as obstacle vehicle position and velocity, and road environment information details, were accurately obtained using equipment installed in the host vehicle, e.g., lidar or radar.

4.2.1 Scenario 1: stationary obstacle

Scenario 1 considers the case of a stationary obstacle in 1 km straight road. The MPC generates a control input for obstacle avoidance steering, which is input to the host vehicle model to obtain the obstacle avoidance path.

Fig. 10 shows that since ROCMPC uses a real-time output constraint box, it avoids relatively wide turns compared with conventional MPC, generating a path closer to the center line of the first lane when avoiding the stationary obstacle vehicle.

Lateral acceleration was limited to 1.98 m/s^2 for both MPC and ROCMPC algorithms, and the two path profiles are shown in Figs. 11 and 12. In the proposed method, there is no significant difference in the path that is created by adding the constrained conditions to the conventional MPC. However, when the obstacle is closed to the host vehicle, the proposed method generates the path faster than the conventional MPC due to the fast operation. Thus, the results also show a slightly quicker response path.

Table 4 compares computation times for the MPC and ROCMPC. Three replicate simulations were performed, and the presented data is the mean.

Operation speed for ROCMPC was approximately 27% improved over MPC. Solver operation speed to obtain the control input was halved by transforming the non-convex

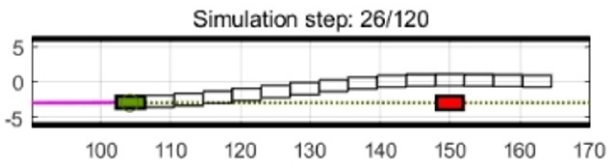


Fig. 10. Stationary obstacle vehicle.

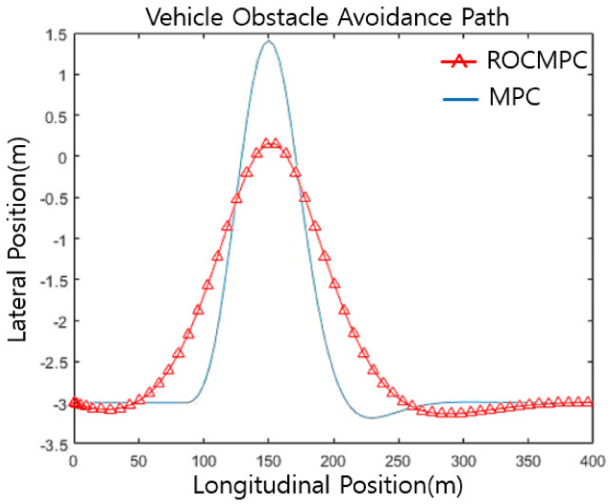


Fig. 11. Obstacle avoidance paths for scenario 1: stationary obstacle vehicle.

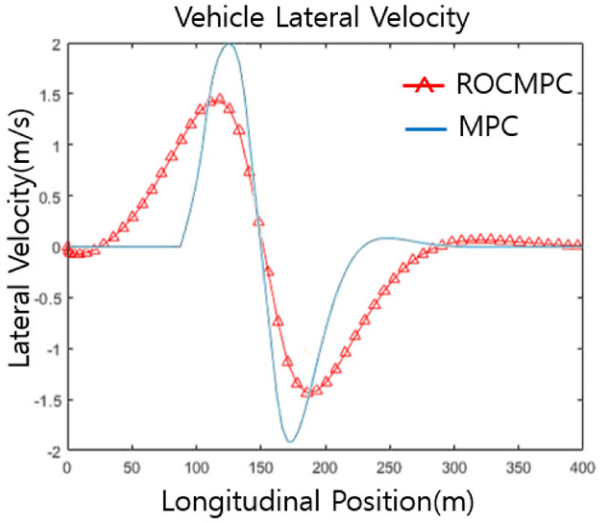


Fig. 12. Lateral velocity for scenario 1: stationary obstacle vehicle.

Table 4. Operation time for scenario 1: stationary obstacle vehicle.

	Passenger car	Sedan	Van
MPC	19.17 ms	18.78 ms	18.21 ms
ROCMPC	13.97 ms	13.55 ms	13.47 ms

problem for the ROCMPC algorithm, but the additional real-time output constraints required for that transformation meant the overall gain was ∼27%.

We also investigated different host vehicle sizes (data not shown). Computation time for the avoidance path was independent of host vehicle size and did not significantly affect the avoidance path. Only the safety margin changed with changing host vehicle size.

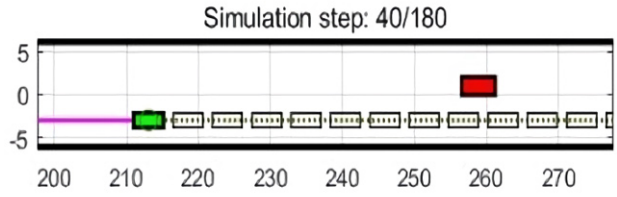


Fig. 13. Scenarios 2-4, dynamic obstacle vehicle.

Table 5. Operation time for scenario 2: dynamic obstacle vehicle.

	Passenger car	Sedan	Van
MPC(50 km/h)	16.17 ms	15.75 ms	15.49 ms
ROCMPC(50 km/h)	10.41 ms	10.30 ms	10.96 ms
MPC(60 km/h)	17.34 ms	17.19 ms	16.99 ms
ROCMPC(60 km/h)	11.62 ms	11.52 ms	11.58 ms
MPC(80 km/h)	18.10 ms	18.94 ms	18.01 ms
ROCMPC(80 km/h)	13.05 ms	12.81 ms	12.94 ms

Thus, the proposed ROCMPC algorithm was significantly superior to MPC for scenario 1: stationary obstacle vehicle.

4.2.2 Scenarios 2-4: dynamic obstacle vehicle

Similar to scenario 1, only the algorithms providing control inputs were changed between MPC and ROCMPC, all other details remained constant. In these scenarios, the obstacle avoidance path considers obstacle vehicle to be at constant 50, 60 and 80 km/h. Table 5 summarizes computation speed to produce an avoidance path, as shown in Fig. 13, for these scenarios.

Both MPC and ROCMPC exhibited longer computation time than for scenario 1, but ROCMPC is approximately 30% superior to MPC. Thus, ROCMPC is more efficient in dynamic vehicle obstacle situations.

Thus, the proposed ROCMPC algorithm was significantly superior to MPC for scenario 2: dynamic obstacle. Furthermore, ROCMPC produced an avoidance path closer to the lane center.

Section 4.4 extends this evaluation to the case of several obstacle vehicles. An experiment to increase the number of obstacle vehicles and an experiment on curved roads rather than on straight roads will be conducted in the Section 4.4.

4.3. Comfort level comparison

As discussed in Section 3, we considered passenger comfort level for the scenarios shown in Fig. 9.

4.3.1 MATLAB simulation results

The static obstacle avoidance path (scenario 1) simulation was set to 120 steps whereas the dynamic obsta-

Table 6. Comfort level score in MATLAB.

	Scenario 1		Scenario 2	
Comfort Level (score)	NMPC	ROCMPC	NMPC	ROCMPC
Comfortable (10)	46.67%	49.64%	60.00%	69.59%
A little uncomfortable (8)	8.89%	8.76%	4.50%	4.64%
Fairly uncomfortable (6)	15.56%	13.87%	10.50%	6.70%
Uncomfortable (4)	9.63%	13.14%	9.50%	7.33%
Very uncomfortable (2)	5.93%	6.57%	7.50%	4.64%
Extremely uncomfortable (0)	13.33%	8.03%	8.00%	6.70%
Average comfort level score	6.81	7.15	7.52	8.13

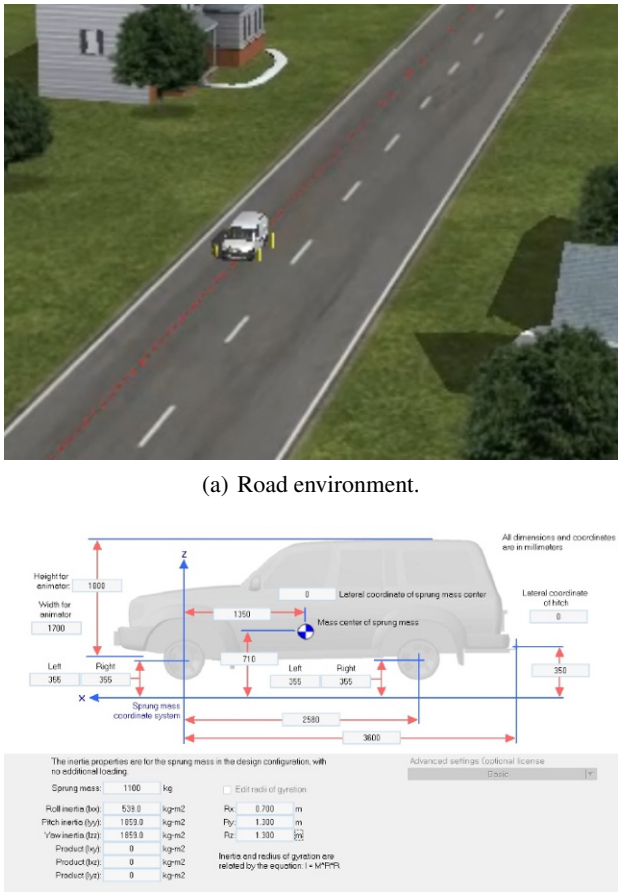
cle scenarios (2, 3, and 4) included 180 steps with 250 ms time length. Table 6 compares comfort level score between MPC and ROCMPC algorithm avoidance paths. Vertical acceleration, one of the inputs to derive a_w, could not be measured in the simulation. Therefore, we assumed zero acceleration in the vertical direction. The average comfort level score is calculated as the summation of each product of the comfort level score and the ratio shown during the simulation. The path generated by the ROCMPC has many comfortable moments and that the moment of extremely uncomfortable is less than that of the compared algorithm, as shown in Table 6. Also, the generated path is stable even at the average value.

ROCMPC avoidance paths exhibited significantly higher comfort level score than MPC paths. This arises for two main reasons:

1) ROCMPC output constraint box limits possible paths; whereas for MPC, steering starts rapidly when the obstacle vehicle is in the control horizon. Thus, the ROCMPC output constraint box ensures a smoother obstacle avoidance path, reducing lateral acceleration, which is highly weighted in the ISO 2631-1 calculation for comfort level.
2) Improved operation speed provides faster predictive values within the prediction horizon. Hence the path can be constructed faster to the same range, reducing deviation from the optimal path in the simulation.

Comfort level is also related to passenger safety. Higher comfort scores imply less rapid steering changes to avoid the obstacle.

Thus, the proposed ROCMPC algorithm produced higher comfort avoidance paths and hence safer paths than the MPC algorithm



(a) Road environment.

(b) Vehicle model.

Fig. 14. Simulation environment of CarSim.

4.3.2 CarSim results

In Section 4.3.1, we assumed vertical acceleration was zero to obtain the comfort level score, since vertical acceleration is not measured. Therefore, we used CarSim to estimate vertical acceleration from vehicle position and velocity obtained from the MATLAB®simulations, and hence estimate a more realistic comfort score as shown in Fig. 14. The road environment and vehicle models were constructed in the same way as for the MATLAB®simulations.

CarSim has 25 ms step length, producing 10 times the number of steps from MATLAB®simulations. Horizontal and vertical accelerations were obtained for each step, and a_w was derived subsequently. Comfort level score for the total obstacle avoidance path was then obtained from a_w for each step, as shown in Table 7.

4.4. Additional scenarios

This section considers curved road environments and several obstacle vehicles.

Table 7. Comfort level score from CarSim.

	Scenario 1		Scenario 2	
Comfort Level (score)	MPC	ROCMPC	NMPC	ROCMPC
Comfortable (10)	69.85%	70.71%	77.09%	73.16%
A little uncomfortable (8)	4.90%	4.35%	2.88%	8.76%
Fairly uncomfortable (6)	6.16%	6.45%	2.93%	5.10%
Uncomfortable (4)	9.17%	9.56%	4.64%	3.09%
Very uncomfortable (2)	6.60%	6.06%	8.90%	9.69%
Extremely uncomfortable (0)	3.34%	2.87%	3.57%	0.21%
Average comfort level score	8.24	8.31	8.48	8.63

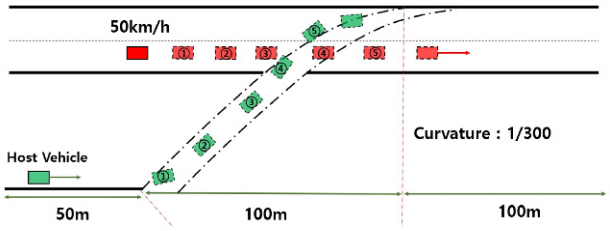


Fig. 15. Curved road environment.

4.4.1 Curved road

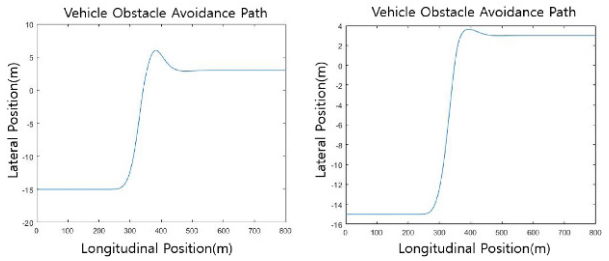
Similar to the straight road, road width= 6 m and the road was two lanes. Road curvature was set at 1/300, and obstacle vehicle behavior was straight-ahead, cut-in, and cut-out, similar to scenarios 2-4 for straight road.

Fig. 15 shows the curved road environment picture. Obstacle vehicle speed is 50km/h, and we considered the obstacle vehicle to be the commercial van.

Fig. 16 shows the generated avoidance paths, and Table 8 shows the computation speeds for MPC and ROCMPC. ROCMPC improved computation speed by 24.75% on average.

Table 9 shows the comfort level score based on the generated obstacle avoidance path.

Thus, the proposed ROCMPC algorithm was significantly superior to the MPC algorithm for both computation time and passenger comfort for all considered behavior scenarios for curved road.



(a) (b)

Fig. 16. Curved road obstacle avoidance path for (a) MPC and (b) ROCMPC.

Table 8. Operation time for curved road obstacle path.

	MPC	ROCMPC	Improvement
Curved scenario 2	15.06 ms	10.88 ms	27.78%
Curved scenario 3	14.58 ms	11.61 ms	20.35%
Curved scenario 4	14.84 ms	10.96 ms	26.14%

Table 9. Comfort level score for curved road obstacle path.

	MPC	ROCMPC
Curved scenario 2	15.06 ms	10.88 ms
Curved scenario 3	14.58 ms	11.61 ms
Curved scenario 4	14.84 ms	10.96 ms

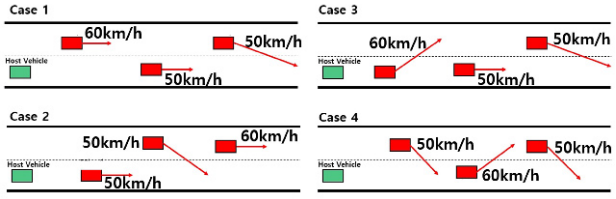


Fig. 17. Multiple obstacle vehicle scenarios.

4.4.2 Multiple obstacle vehicle scenarios

We considered the case of three obstacles vehicles rather than a single vehicle. As with the previous simulations conducted earlier, the road was 1 km long, 6 m wide, and two-lane.

Fig. 17 shows the 4 scenarios considered and Table 10 shows the operation time for each scenario.

Thus, the proposed ROCMPC algorithm exhibited significantly superior computation speed to the MPC algorithm considering several obstacle vehicles.

We also measured the comfort level for the various generated paths in MATLAB and CarSim.

Table 10. Operation times for multiple obstacle vehicles.

	MPC	ROCMPC	Improvement
Scenario 1	36.38 ms	28.98 ms	21.7%
Scenario 2	37.82 ms	26.37 ms	30.3%
Scenario 3	37.71 ms	25.93 ms	31.2%
Scenario 4	29.08 ms	19.58 ms	32.7%

Table 11. Decision making simulations.

Control type	Host vehicle speed	Obstacle vehicle speed	Decision
MPC	80 km/h	70 km/h	Steering
ROCMPC	80 km/h	70 km/h	Steering
MPC	80 km/h	60 km/h	Braking
ROCMPC	80 km/h	60 km/h	Steering

4.4.3 Braking control

This section considers simulations to prevent obstacle collision through braking. The host vehicle must determine steering or braking when it detects the obstacle using Table 11. Host vehicle velocity was 80 km/h, TTC= 1.00 s, and overlap threshold= 50%. Obstacle behavior was straight ahead, and obstacle vehicle initial position was adjusted to determine collision avoidance by steering or braking. If a steering avoidance strategy was not possible, the host vehicle should choose a strategy to minimize damage through braking.

Simulations confirmed that collision prevention using braking was chosen when the speed difference was large, since obstacle avoidance paths are difficult at faster relative speeds. However, at the same relative speed, ROCMPC tended to select collision avoidance using steering whereas MPC selected braking. This was because ROCMPC begins steering earlier than MPC due to the constraint box, enhancing steering avoidance opportunities.

5. CONCLUSIONS

This paper proposed the ROCMPC algorithm to avoid obstacles using MPC in high-speed driving situations. The proposed ROCMPC algorithm considers real-time output constraints regarding relative distance and velocity between the host and obstacle vehicle. The non-convex problem was transformed into a convex problem, significantly reducing operation time. ROCMPC sets a real-time output constraint box on the host vehicle to enhance computation speed and avoid sharp steering changes, consequently generating more comfortable and safer avoidance paths than conventional MPC.

We performed a number of simulations covering various scenarios, comparing the proposed ROCMPC and conventional MPC algorithms for operation time and comfort level, indispensable indicators for high speed autonomous driving situations.

Since vertical acceleration was not available from the MATLAB®simulations, we also evaluated comfort and safety for generated avoidance paths using CarSim. Simulations considered straight and curved roads, several obstructing vehicle behavior scenarios, several obstructing vehicle types, single and multiple obstacle vehicle cases, and two and three lane roads.

In all cases, the proposed ROCMPC algorithm exhibited significantly enhanced computation speed than MPC, and significantly improved passenger comfort, and hence safety.

This paper considered a relatively small number of limited scenarios evaluated in MATLAB®and CarSim simulations, rather than actual vehicles or road conditions. Therefore, further study will be pursued to verify ROCMPC for actual autonomous driving situations.

REFERENCES

[1] K. P. Carroll, S. R. McClaran, E. L. Nelson, D. M. Barnett, D. K. Friesen, and G. N. William, "AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones," *Proceedings of the IEEE 1992 Symposium on Autonomous Underwater Vehicle Technology, AUV'92*, pp. 79-84, 1992.

[2] C. Wang, L. Wang, J. Qin, Z. Wu, L. Duan, Z. Li, M. Cao, X. Ou, X. Su, W. Li, Z. Lu, M. Li, Y. Wang, J. Long, M. Huang, Y. Li, and Q. Wang, "Path planning of automated guided vehicles based on improved A-Star algorithm," *Proceedings of the 2015 IEEE International Conference on Information and Automation*, pp. 2071-2076, 2015.

[3] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation*, vol. 8 no. 1, pp. 23-32, 1992.

[4] Y. Rasekhipour, A. Khajepour, S. K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation System*, vol. 18, no. 5, pp. 1255-1267, 2017.

[5] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous System*, vol. 3, no. 2-4, pp. 265-291, 2005.

[6] E. Kim, J. Kim, and M. Sunwoo, "Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics," *International Journal of Automotive Technology*, vol. 15, no. 7, pp. 1155-1164, 2014.

[7] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transaction on Control Systems Technology*, vol. 25, no. 3, pp. 899-910, 2017.

[8] R. Quirynen, K. Berntorp, and S. Di Cairano, "Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control," *Proceedings of the 2018 IEEE Annual American Control Conference (ACC)*, pp. 3251-3256, 2018.

[9] T. Weiskircher, Q. Wang, and B. Ayalew, "Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic," *IEEE Transactions on Control System Technology*, vol. 25, no. 6, pp. 2034-2046, 2017.

[10] T. Wang, H. Gao, and J. Qiu, "A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 27, no. 2, pp. 416-425, 2016.

[11] D. A. Copp and J. P. Hespanha, "Nonlinear output-feedback model predictive control with moving horizon estimation," *Proceedings of the IEEE 53rd Annual Conference on Decision and Control (CDC)*, pp. 3511-3517, 2014.

[12] J. M. Maestre and R. R. Negenborn, *Distributed Model Predictive Control Made Easy*, Springer, vol. 69, 2014.

[13] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control System Technology*, vol. 15, no. 3, pp. 566-580, 2007.

[14] J. Ji, A. Khajepour, W. W. Melek, and Y, Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952-964, 2017.

[15] X. Qian, A. De La Fortelle, and F. Moutarde, "A hierarchical model predictive control framework for on-road formation control of autonomous vehicles," *Proceeding of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 376-381, 2016.

[16] H. J. Kim, D. H. Shim, and S. Sastry, "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles," *Proceedings of the IEEE American Control Conference*, vol. 5, pp. 3576-3581, 2002.

[17] I. O. for Standardization, Mechanical vibration and shock-Evaluation of human exposure to whole-body vibration-Part 1: General requirements. The Organization, 1997.

[18] X. Zhao and C. Schindler, "Evaluation of whole-body vibration exposure experienced by operators of a compact wheel loader according to ISO 2631-1: 1997 and iso 2631-5: 2004," *International Journal of Industrial Ergonomics*, vol. 44, no. 6, pp. 840-850, 2014.

[19] M. Nolte, M. Rose, T. Stolte, and M. Maurer, "Model predictive control based trajectory generation for autonomous vehicles-an architectural approach," arXiv preprint arXiv:1708.02518, 2017.

[20] H. Li and Y. Shi, "Robust distributed model predictive control of constrained continuous-time nonlinear systems: A robustness constraint approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1673-1678, 2014.

[21] E. F. Camacho and C. B. Alba, *Model Predictive Control*, Springer Science & Business Media, 2013.

[22] T. D. Gillespie, *Vehicle Dynamics*, Warren Dale, 1997.

[23] H. Pacejka, *Tire and Vehicle Dynamics*, Elsevier, 2005.

[24] R. Rajamani, *Vehicle Dynamics and Control*, Springer Science & Business Media, 2011.

[25] P. Holmlund and R. Lundstrom, "Mechanical impedance of the sitting human body in single-axis compared to multi-axis whole-body vibration exposure," *Clinical Biomechanics*, vol 16. S101-S110, 2001.

[26] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Engineering Practice*, vol. 61, pp. 307-316, 2017.

[27] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach-its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528-540, 2006.

[28] D. N. Lee, "A theory of visual control braking based on information about time to collision," *Perception*, vol. 5, no. 4, pp. 437-459, 1976.

[29] M. M. Minderhoud and P. H. Bovy, "Extended time to collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89-97, 2001.

[30] M. Essa and S. Tarek, "Full Bayesian conflict-based models for real time safety evaluation of signalized intersections," *Accident Analysis & Prevention*, vol. 129, pp. 367-381, 2019.

Ji-Chang Kim received his B.S. degree in Electrical Engineering from Korea University in 2017 and his Master degree in Automotive Convergence from Korea University in 2019. His research interests include model predictive control and vehicle dynamics.

Dong-Sung Pae received his B.S. degree in Electrical Engineering from Korea University, Seoul, Korea, in 2013, where he has been working toward a Ph.D. degree with the School of Electrical Engineering since 2013. His current research interests include computer vision, feature extractor, video stabilization, artificial intelligence, and their applications to intelligence vehicle systems.

Myo-Taeg Lim received his B.S. and M.S. degrees in Electrical Engineering from Korea University, Seoul, Korea, in 1985 and 1987, respectively. He also received his M.S. and Ph.D. degrees in Electrical Engineering from Rutgers University, NJ, USA, in 1990 and 1994, respectively. He was a Senior Research Engineer with the Samsung Advanced Institute of Technology and a Professor in the Department of Control and Instrumentation, National Changwon University, Korea. Since 1996, he has been a Professor in the School of Electrical Engineering at Korea University. His research interests include optimal and robust control, vision based motion control, and autonomous vehicles. He is the author or coauthor of more than 80 journal papers and two books (Optimal Control of Singularly Perturbed Linear Systems and Application: High-Accuracy Techniques, Control Engineering Series, Marcel Dekker, New York, 2001; Optimal Control: Weakly Coupled Systems and Applications, Automation and Control Engineering Series, CRC Press, New York, 2009). Dr. Lim currently serves as an Editor for International Journal of Control, Automation, and Systems. He is a Fellow of the Institute of Control, Robot and Systems, and a member of the IEEE and Korea Institute of Electrical Engineers.