

# M-Space를 이용한 자동 주차를 위한 주차 경로 생성

## Motion Planning of the Car-like Vehicle in the Parking Space by the Motion Space

김 달 형<sup>1</sup>, 정 우 진<sup>2</sup>

Dalhyung Kim<sup>1</sup>, Woojin Chung<sup>2</sup>

**Abstract** Automatic parking assist system is one of the key technologies of the future automobiles. Control problem of a car-like vehicle is not easy due to the nonholonomic constraints. In this paper, a practical solution for planning a car-parking path is proposed according to the proposed motion space (M-space) approach. The M-space is the extension of the conventional configuration space (C-space). A collision-free, nonholonomic feasible path can be directly computed by the M-space conversion and a back-propagation of reachable regions from the goal. The proposed planning scheme provide not a single solution, but also a candidate solution set, therefore, optimization of the parking path can be easily carried out with respect to performance criteria such as safety, maneuvering, and so on. Presented simulation results clearly show that the proposed scheme provides various practical solutions.

**Keywords** : Motion planning

### 1. 서 론

최근, 여러 지능화된 기술들이 자동차에 접목되고 있으며 이 중 자동 주차 시스템은 가장 주목 받는 기술이다. 자동 주차 시스템은 운전자가 주차를 수행할 때, 시스템이 자동차의 위치와 주차 목표 위치를 인식하여 적절한 주차 경로를 생성하고, EPS(Electronic Power Steering)을 이용하여 생성된 주차 경로를 따라가며 주차를 수행할 수 있도록 조향 장치를 자동으로 제어해 주는 시스템이다. 이 때, 자동차의 속도는 운전자에 의해 직접 제어되게 된다.

자동 주차 시스템을 위한 경로 생성 문제는, 자동차의 구름접촉에 의한 발생하는 논홀로노믹 구속조건을 만족시키며 주위의 장애물과 충돌을 하지 않고 주어진 경계조건인 초기 위치와 최종위치 연결하는 경로를 생성하는 것이다. 이러한 문제를 해결하기 위한 방법으로 [2,3]에 제안된 체인드롭 변환기법이 있다. 그러나 이 방

법은 변환식이 너무 복잡한 비선형식으로 표현되고, 주위의 장애물을 고려하지 않은 방법이기 때문에 주차 환경과 같은 좁은 공간에서의 경로 생성 방법으로 적합하지 않다. 또 다른 방법으로는 [8]에서는 운동을 원과 직선으로 제한하여, 이 운동들을 이용하여 두 위치를 연결하는 최소 거리의 경로를 연결하는 방법을 제안하였다. 하지만 이 역시 장애물을 고려하지 않은 상황에서 두 위치의 연결 관계만 고려하여 주차 경로 생성 방법으로는 적합하지 않다.

다른 방법으로 [9]에서는 주위의 장애물을 고려한 홀로노믹 경로 생성 단계와 생성된 경로를 논홀로노믹 경로로 만드는 두 단계로 나누어진 방법을 제안하였다. 이 방법은 홀로노믹 경로를 바탕으로 경로를 생성하기 때문에 주차에 적합한 경로를 찾는 방법으로는 부적합하다. 또한 [4]에서는 C-space를 격자 지도로 나누어 자동차의 국부 경로 계획기를 이용하여 경계조건이 포함된 격자를 검색하는 방법을 제안하였다. 이 방법은 격자의 크기의 설정의 어려움과 계산 비용이 큰 단점이 있다. 또한 격자 지도를 바탕으로 하기 때문에 정확한 해가 아닌 근사 해를 주어 주차 경로 계획기로는 부적

※ 본 연구는 (주) 만도의 VISS(영상기반 지능형 조향장치) 개발 과제 지원으로 수행되었음.

<sup>1</sup> 고려대학교 기계공학과 연구원 (E-mail : zsnail@korea.ac.kr)

<sup>2</sup> 고려대학교 기계공학과 교수 (E-mail : smartrobot@korea.ac.kr)

합하다. 또 다른 연구로는 국부 경로 계획기를 이용하여 로드맵을 구성하고, 이 로드맵을 이용하여 경로를 생성하는 방법[6,7]이 있다. 이 방법은 로드맵을 구성하는 것에 시간은 걸리지만 일단 완성하게 되면, 일단 완성하게 되면 빠른 시간 안에 경로를 생성 할 수 있다. 이 방법은 주로 복잡한 환경에서 적합한 방법으로, 주차 공간과 같은 좁은 환경인 경우에 생성된 경로가 주차에 적합한 최적의 해인지를 보장할 수 없어, 주차를 위한 경로 계획기로 부적합하다.

본 논문에서는 M-Space라는 개념을 이용하여 주차에 적합한 경로 계획기를 제안한다 M-Space는 모델의 모양과 운동을 모두 고려한 무 충돌 영역을 말한다. M-Space를 이용하여 제한된 모션의 연속적인 도달 가능 영역을 계산할 수 있으며, 연속적인 영역을 역 전과 방법을 이용하여 경로 생성 가능 영역(path set)를 계산 할 수 있다. 경로 생성 가능 영역에서 하나의 점을 선택하여 최종 경로를 생성할 수 있으며, 기존의 방법과 같이 하나의 경로만 주는 방법에 비해, 수 많은 경로를 쉽게 선택할 수 있는 장점이 있다. 또한 모션의 개수를 제한 하였을 때, 가능한 모든 경로 생성 가능 영역을 검색하여 더 많은 경로를 생성할 수 있다. 생성된 경로에 경로의 선호 조건인 안전, 조향 장치의 조작의 횟수, 전후진의 조작 빈도, 이동 거리 등을 고려하여 비교함으로써 주차에 적합한 경로를 선택할 수 있다. 시뮬레이션 결과를 통하여 제안된 방법이 주차 공간에서 유용함을 보이고, 생성된 경로가 선호 조건에 따라 주차 경로에 적합한 경로가 생성됨을 보인다.

## 2. Motion Space (M-Space)

### 2.1 주차 환경

주차 환경은 실제로 매우 다양하지만, 대부분 직각 주차 또는 평행 주차가 가능한 공간으로 나누어 분류할 수 있다. 이런 환경을 간단히 모델링 하기 위하여 주위

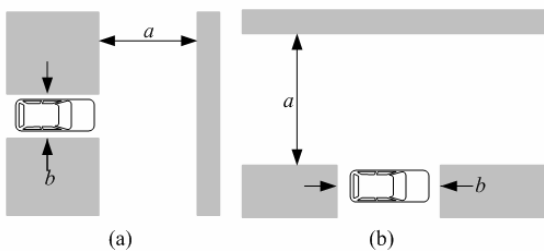


그림 1. (a) 직각주차 환경, (b) 평행주차 환경

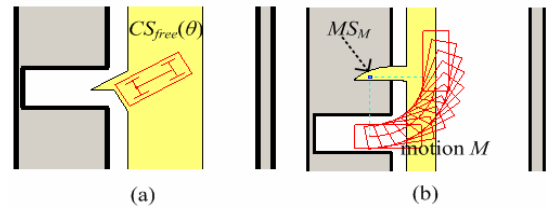


그림 2. (a) 무 충돌 영역  $CS_{free}(\theta)$  (b) Motion Space

의 주차된 자동차나 벽을 모두 장애물로 인식하여 그림 1-(a), (b)와 같이 나타낼 수 있다. 이 공간은 주차 도로의 폭 (a)와 주차 공간의 폭 (b)의 두 개의 파라미터로 표현될 수 있다. 이 두 파라미터의 크기에 따라 주차의 난이도가 결정된다.

먼저, 몇 가지 용어에 대해 정의가 필요하다. 자동차 모델의 위치를 configuration 또는 pose( $q$ )라 하며 위치  $x, y$ 와 방향  $\theta$ 로 표현된다. 이 때 자동차의 위치  $x, y$ 는 후륜 바퀴 축 중심으로 한다. 이러한 configuration의 모인 공간은 configuration space (CS)라 한다. 이러한 configuration 중 장애물과 충돌이 일어나지 않는 configuration만 모인 공간을 free configuration space( $CS_{free}$ )라 한다. 이  $CS_{free}$  중 방향( $\theta$ )이 같은 configuration만 모인 공간을  $CS_{free}(\theta)$ 로 표현한다. 그림 2-(a)는  $CS_{free}$  가운데  $\theta$ 가 같은 충돌이 없는  $CS_{free}(\theta)$ 를 나타낸 그림이다.

### 2.2 Motion Space

Motion Space(M-Space, MS)는 자동차 모델의 운동을 결합한 확장된 C-Space라 할 수 있다. 운동이 구체적으로 정해지면, 자동차 모델이 운동을 수행하면서 자동차 모델에 의해 점유되는 공간을 결정할 수 있다. 이 공간을 하나의 물체라 가정하고 장애물과의 무 충돌 영역을 구할 수 있는데, 이 영역을 M-Space라 한다.

정의된 운동에 원과 같은 곡선의 부분이 존재하면 그 운동에 의해 점유되는 영역을 이용하여 CS를 구하는 일반적인 방법인 minkowski sum[1, 10]에 적용하는 것은 쉬운 문제가 아니다. 그 이유는 곡선이 존재하는 경우 다각형으로 근사화 할 때, 많은 점들의 다각형을 계산해야 하므로 점에 개수에 비례하여 계산 비용이 증가하는 minkowski sum로는 한계가 있다. 하지만 자동차 모델에 대하여 계산된 CS를 이용하면 그 과정을 좀 더 쉽게 구할 수 있다. 자동차 모델은 다각형 모양(주로 사각형)이기 때문에 모델의 각 방향( $\theta$ )에 대한 CS는 minkowski sum을 이용하여 쉽게 구할 수 있다.

M-Space를 구하기 위해 정의된 운동의 경로를 하나

의 점으로 대표하기 위하여, 임의의 기준 점( $p_{ref}$ )를 선택한다. 각 경로 위의 위치에서는 모델 방향( $\theta$ )의  $CS_{free}(\theta)$ 가 각각 존재하는데, 기준 점과 경로 위의 점의 상대적인 위치 차이만큼 이동시켜 그 영역의 공통 영역을 구함으로써  $M$ -Space를 쉽게 구할 수 있다. 이 영역 안에서는 정의된 모션에 대해  $p_{ref}$ 가 존재하게 되면 충돌이 일어나지 않음을 알 수 있다. 알고리즘 1과 그림 2-(b)는  $M$ -Space의 계산 방법을 설명하고 있다.  $MS_M$ 은 모션  $M$ 에 의한  $M$ -Space를 의미한다.

```

Input: a motion  $M$  (a sequence of poses)

 $p_{ref} \leftarrow$  the selected point as a reference point
For all poses  $q \in M$ 
     $\delta \leftarrow$  the offset between  $q$  and  $p_{ref}$ 
     $S_{free,shift(\delta)}(\theta) \leftarrow$  the moved region from  $CS_{free}(\theta)$  by  $\delta$ 
End
 $MS_M \leftarrow \bigcap CS_{free,shift}(\theta)$ 
Return  $MS_M, \delta$ 

```

알고리즘 1. Compute Motion Space

### 3. 경로 계획

#### 3.1 기본 운동의 정의

주차 수행 시, 사람들은 좁은 공간에서 움직이기 위해 조향 각도를 최대한으로 유지한 상태로 대부분 이동한다. 이 때의 이동경로는 최소 회전 반경을 반지름으로 갖는 원 또는 호가 된다. 이런 점을 착안하여 두 가지 운동을 정의하였다. 하나는 차량의 방향을 유지한 채 움직이는 직선운동(translation motion,  $TM$ )이고, 다른 하나는 차량의 조향 각도를 최대한으로 유지한 채 최소 회전 반경을 갖고 움직이는 원 운동(arc motion,  $AM$ )이다.

##### 3.1.1 직선운동 (Translation Motion, $TM$ )

직선 운동( $TM$ )은 차량의 pose가 주어졌을 때, 그 위치의 방향( $\theta$ )을 유지한 채 움직이는 직선 운동을 말한다.  $TM$ 에 의한 도달 가능 영역( $RR_{TM}$ )은 그림 3과 같이 쉽게 계산될 수 있다.

도달 가능 영역은 초기 위치 그림-3(a)로부터 충돌이 일어나지 않는 영역  $CS_{free}(\theta)$  안에서 직선으로 도달 가능한 영역이 된다. 그림 3-(c)는 (a)의 초기 위치로부터 도달 가능한 영역( $RR_{TM}$ )을 보여주고 있다. 그림 3-(b)의 자동차 모델의 방향의  $CS_{free}(\theta)$ 를 나타내고 있다.

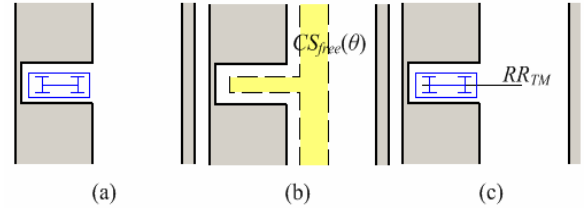


그림 3. 직선운동(TM)에 의한 도달 가능 영역(RR)  
(a) 입력 영역(초기 위치), (b)  $CS_{free}(\theta)$ , (c)  $RR_{TM}$

##### 3.1.2 원운동 (Arc Motion, $AM$ )

원 운동( $AM$ )은 자동차 모델의 최대 곡률 반경을 유지한 채 움직이는 운동으로 정의한다. 최대 곡률 반경을 유지하려면 최대 조향 각도를 유지해야 하며, 최소 회전 반경을 갖고 원 운동을 하게 된다. 자동차의 시작 방향( $\theta_{start}$ )과 최종 방향( $\theta_{final}$ )이 주어졌을 때, 이 두 방향 사이를 움직이는 운동은 그림 4와 같이 병진 속도와 조향 각에 따라 4가지 경우로 나누어 진다. 즉, 원 운동을 결정하려면 최초 및 최종 방향과, 자동차의 운동을 결정하는 입력인 병진 속도( $v$ )와 조향 각( $\phi$ )이 필요하게 된다.

일단 원 운동이 하나로 정해지면, 원의 회전 중심을 기준 점( $p_{ref}$ )으로 선택하여 정해진 원 운동의 무 충돌 영역( $MS_{AM}$ )을 알고리즘 1에 의해 구할 수 있다.  $MS_{AM}$ 을

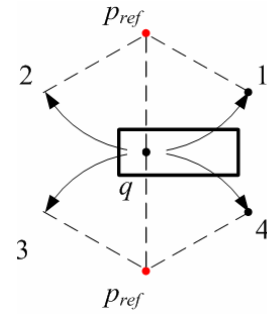


그림 4. 병진속도와 조향 각에 따른 4가지 경로 유형

$$\begin{aligned} \langle 1 \rangle v &= 1, \phi = \phi_{max} & \langle 2 \rangle v &= -1, \phi = \phi_{max} \\ \langle 3 \rangle v &= -1, \phi = -\phi_{max} & \langle 4 \rangle v &= 1, \phi = -\phi_{max} \end{aligned}$$

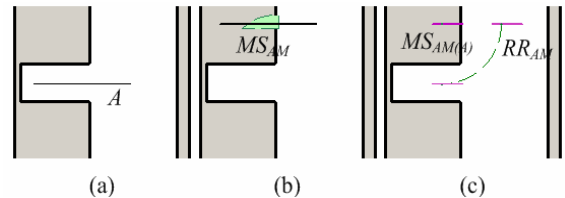


그림 5. 원 운동(AM)에 의한 도달 가능 영역  
(a) 입력 영역(초기 위치), (b)  $MS_{AM}$ , (c)  $RR_{AM}$

이용하여 도달 가능 영역( $RR_{AM}$ )을 구하는 과정이 그림 5에 표현되어 있다. 그림 5-(b)의  $MS_{AM}$  초기 및 최종 방향( $\theta_{start}, \theta_{final}$ )을 0도, 90도로 각각 가정하고, 병진 속도와 조향 각 모두 양수일 때의 원 운동을 가정한 결과이다. 초기 영역 중에 원 운동이 가능한 영역을 찾기 위하여  $MS_{AM}$ 과 비교하여 가능한 영역을 골라낸다. 이 영역은 그림 5-(b)와 같이 초기 영역을 기준 점과 초기 위치의 차이만큼 이동시킨 후  $MS_{AM}$ 과 공통 영역을 구하여 얻는다.  $RR_{AM}$ 은  $MS_{AM(A)}$ 를 최종 위치와  $p_{ref}$ 의 차이만큼 이동시켜 얻는다.  $MS_{AM(A)}$ 는 A로부터 AM이 가능한 MS를 말한다. 그림 5-(c)는 도달 가능한 영역( $RR_{AM}$ )을 구한 결과를 보여주고 있다.

### 3.1.3 One Set of Motion

경로를 계획하기 위해 모션 세트를 정의하였다. 모션 세트는 직선운동-원운동-직선운동(TM-AM-TM)의 순서로 정의하였다. 이렇게 정의된 모션 세트는 양쪽의 직선운동 때문에 서로 쉽게 연결할 수 있기 때문이다. 그림 6는 하나의 모션 세트에 의해 도달 가능한 영역을 계산

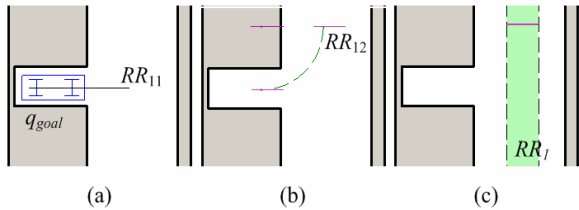


그림 6. 모션 세트에 의한 도달 가능 영역  
(a)  $RR_{TM}$ , (b)  $RR_{AM}$ , (c)  $RR_{1SET}$

Input:  $\theta_{start}, \theta_{final}$  (The start orientation and the end orientation of an arc motion),  $A$ : An input region with an orientation  $\theta_1, v, \varphi$ : A velocity input for a circle motion

$p_{ref}$  ← The instantaneous center for a circle motion  
 $RR_{i,1}$  ← A reachable region from  $A$  by a translation motion in  $CS_{free}(\theta_1)$ .  
 $AM$  ← the unique motion ( $\theta_{start}, \theta_{final}, v, \varphi$ )  
 $MS_{AM}, \delta$  ← Compute M-space ( $AM$ )  
 $MS_{i,2}$  ← An intersection region between  $RR_{i,1,shift}(\delta_{start})$  ( $\theta_{start}$ ) and  $MS_{AM}$   
 $RR_{i,2}$  ←  $MS_{i,2,shift}(\delta_{final})(\theta_{final})$  (shift  $MS_{AM(A)}$  by  $\delta_{final}$ )  
 $RR_i$  ← A reachable region from  $RR_{i,2}$  by a TM in  $CS_{free}(\theta_{final})$ .  
**Return**  $RR_i$

알고리즘 2. Compute Reachable Region  $RR_i$

하는 과정을 나타내고 있다. 그림 6-(a), (c)는 전 단계의 영역으로부터 직선운동에 의한 도달 가능 영역이며, 그림 6-(b)는 원 운동에 의한 도달 가능 영역이다. 알고리즘 2는 하나의 모션 세트에 의해 도달 가능한 영역을 계산하는 과정을 나타낸다.

### 3.2 도달 가능 영역의 확장

도달 가능 영역의 확장은 최종 위치로부터 시작된다. 목표 위치( $q_{goal}$ )로부터 모션 세트를 하나씩 연결하면서 도달 가능한 영역을 계속 계산한다. 이 도달 영역 안에 초기 위치( $q_{ini}$ )가 존재하게 되면 이 영역을 최종위치 도달 가능 초기영역(Goal-Reachable Start Region,  $GRSR$ )이라한다.

계산을 위하여 몇 개의 파라미터가 필요하다.  $Depth\ n$ 은 모션 세트의 개수를 의미하며, 이것은  $n$ 개의 모션 세트까지 연속적으로 연결하여  $GRSR$ 을 모두 검색할 수 있음을 의미한다.

최초에  $n=1$ 인 경우에는, 모션이 한 세트만 허용되기 때문에 AM의 최초 방향( $\theta_{start}$ )은 목표위치의 방향( $\theta_{goal}$ ), 최종 방향( $\theta_{final}$ )은 초기 위치의 방향( $\theta_{ini}$ )으로 정해지게 된다. 하지만  $n=2$ 인 경우에는 2개의 모션이 허용되기 때문에, 첫 번째 모션 세트의 AM의  $\theta_{start}$ 은  $\theta_{goal}$ 이 되고 두 번째 모션 세트의 AM의  $\theta_{final}$ 이  $\theta_{ini}$ 이 된다. 이 두 모션 세트를 연결을 위하여 또 다른 파라미터는 중간 경유 방향( $\theta_{inter}$ )이 필요하다. 중간의 모션 세트의  $\theta_{start}, \theta_{final}$ 은 이 중간 경유 방향( $\theta_{inter}$ )에 의해 정해지게 된다. 중간 경유 방향의 개수가 많을수록 다양한 모션을 만들 수 있지만, 계산의 부하가 큰 trade off가 있다. 실제 계산 중에는 모션 세트가 하나씩 연결되는 과정에서 TM이 중복되므로 생략되어 계산 가능하다. 그림 7은 최종 위치로부터 초기 영역까지 도달 가능 영역이 확장되며 초기 위치가 포함된 영역을 찾는 과정이며, 그림에서 영역  $RR_n$ 은 초기 위치를 포함하고 있기 때문에  $GRSR$ 이 된다.

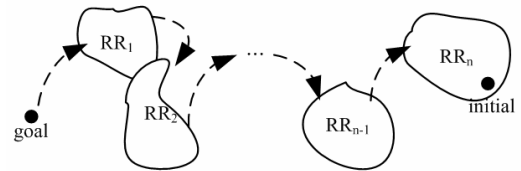


그림 7. 최종위치부터의 도달가능영역의 확장

### 3.3 도달 가능 영역의 역 전파 (Back-propagation)

일단 초기 도달 가능 영역( $GRSR$ )이 검색되면,  $GRSR$

로부터 역 전파 방법(Back-propagation method)를 이용하여 최종 경로를 결정할 수 있는 경로 생성 가능 영역(path set)을 구할 수 있다.

GRSR을 초기 위치( $q_{in}$ )로 대체하고 역으로 초기 위치로부터 도달 가능 영역을 구한다. 이 때, 초기 위치로부터 계산한 도달 가능 영역과 최종 위치로부터 계산한 도달 가능 영역이 겹치는 영역을 계산한다. 이 과정을 초기 위치부터 최종 위치까지 수행하면 초기 위치와 최종 위치 사이의 새로 계산된 영역은 작아지게 된다. 이렇게 구해진 모든 영역을 경로 생성 가능 영역(path set)이라 하고, 이 중 각 영역에서 하나의 점을 선택하여 경로를 생성한다. 그림 8은 역 전파에 의해 구해진 경로 생성 가능 영역의 개념적인 그림을 나타낸 것이다. 새로 구해진  $RR_1$ 부터  $RR_{n-1}$ 이 경로 생성 가능 영역(path set)이며, 각 영역에서 하나의 점을 선택함으로써 경로를 생성할 수 있다.

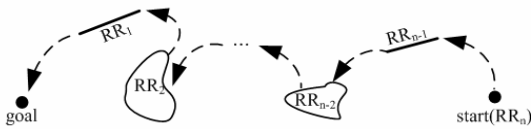


그림 8. 역 전파 방법에 의한 경로생성 가능 영역

### 3.4 경로 선택

모션 세트의 개수를  $depth$ 에 의해 제한하고 GRSR을 얻어내기 때문에,  $depth$ 가 커질수록 더 많은 GRSR을 얻어낼 수 있다. 각각의 GRSR로부터 서로 다른 경로 생성 가능 영역(path set)을 얻어낼 수 있어 더 많은 다양한 경로를 얻어낼 수 있다. 이 얻어낸 경로들 중 가장 주차에 적합한 경로를 선택하기 위해, 안전도, 조향 및 전후진의 빈도, 이동 거리 등과 같은 기준을 통하여 적절한 주차 경로를 경로의 선택하는데 이용하였다.

안전한 경로의 기준은 경로 중 가장 장애물과 가까운 거리로 표현될 수 있다. 주차의 특성상 목표 위치가 비교적 벽에 가까운 위치에 존재하기 때문에 목표 위치에서의 장애물과 거리와 경로 중 가장 장애물과 가까운 거리의 차이를 안전도의 척도로 사용하였다. 이 값이 작을수록 안전한 경로라 할 수 있다. 조향 조작 빈도는  $depth$ 가 모션 세트의 개수를 의미하기 때문에 이 값과 밀접한 관계가 있다. 전후진 조작 빈도는 경로 중 전진과 후진이 바뀌는 횟수로 나타내어진다. 조향 및 전후진 조작 빈도는 적을수록 더 선호되는 경로이다. 마지막으로 이동 거리는 경로의 총 길이로 이 값으로 작을

수록 더 선호되는 경로이다.

위에 제시된 기준 중 선호되는 기준을 고려하여 가중치를 부여하여 각 경로마다 경로 비용(cost)을 부여하여 가장 적은 비용을 갖는 경로를 최종 경로로 선택한다.

$$C_{total} = \alpha(D_{goal,obs} - D_{min,obs}) + \beta N_{steer} + \gamma N_{shifting} + \delta D_{travel}$$

$C_{total}$ : Total cost

$D_{goal,obs}$ [mm]: Distance with obstacles at the goal pose

$D_{min,obs}$ [mm]: Minimum distance with obstacles on the path

$N_{steer}$ : Number of steer maneuvering

$N_{maneuvering}$ : Number of changing a direction

$D_{travel}$ [mm]: Travel distance

$\alpha$ [1/mm],  $\beta$ ,  $\gamma$ ,  $\delta$ [1/mm]: Weights

## 4. 시뮬레이션

제안된 방법의 유효성을 판단하기 위해 실제 주차 환경과 자동차 모델에 대해서 시뮬레이션을 수행하였다. 그림 9은 자동차 모델의 치수를 나타낸 것이다. 최대 조향 각도는 30도로, 최소 회전 반경은 4746mm가 된다.

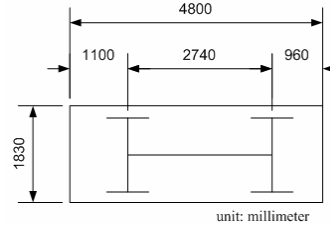


그림 9. Dimension of the real car

### 4.1 직각 주차

직각 주차의 시뮬레이션을 위한 환경은 표 1과 같다. 각각의 path set에서 임의로 3개의 경로를 생성하였다. 이 3개의 경로는 같은 path set에서 나왔으므로 원 운동은 같지만, 직선 운동이 서로 틀려 다소 다른 경로가 된다.

323개의 GRSR을 검색하여 967개의 경로를 생성하여 이들 사이의 비용을 구하여 최종 경로까지 생성하는 데 약 0.9sec의 시간이 걸렸다. 경로 비용(cost)을 구하는 파라미터는 표 2와 같다. 첫 번째 파라미터 set은 상대적으로 safety에 관련되는 파라미터인  $\alpha$ 를 크게 주어 가장 안전한 경로를 찾도록 설정한 것이다. 두 번째 파라미터 set은 maneuvering (the number of changing direction)의 파라미터인  $\gamma$ 를  $\alpha$ 의 효과와 비슷한 정도가 되도록 크게 주어, 안전하면서 전후진의 조작 빈도가 적은 경로를 찾아내도록 한 것이다.

표 1. 직각 주차 시뮬레이션 환경 및 파라미터

$\theta_{inlet}$	10°	Parking lot Env.	$a$	6.5m
$depth_{max}$	3		$b$	2.3m
Boundary Condition	$q_{goal}$	[-4m, 0m, 0°]		
	$q_{int}$	[2.5m, 10m, 90°]		

표 2. 직각 주차 시뮬레이션의 파라미터 값

Weights	$\alpha$ [1/mm]	$\beta$	$\gamma$	$\delta$ [1/mm]
Set 1	0.2	1	2	0.0001
Set 2	0.2	1	6	0.0001

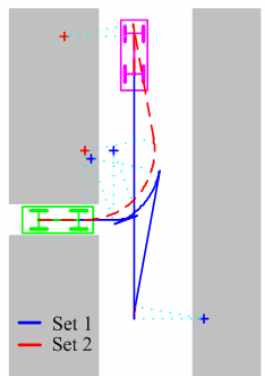


그림 10. 직각 주차 시뮬레이션 결과

그림 10은 표 1와 표 2를 바탕으로 한 시뮬레이션 결과이다. 그림 10에서 파라미터에 따라 달리 선택된 2개의 최종 경로가 나타나 있다. 실선의 경로는 파라미터 set 1에 의한 경로이며 점선 경로는 파라미터 set 2에 의해 선택된 경로이다.

실선의 경로는 상대적으로 안전한 경로를 구하기 위하여 파라미터를 설정하였기 때문에, 비록 많은 조작과, 전후진의 횟수, 그리고 긴 주행거리가 있지만, 보다 안전한 경로를 얻을 수 있었다. 반면 점선의 경로는 실선의 경로보다 더 많은 전후진의 조작을 피하기 위하여 maneuvering에 관계된 파라미터인  $\gamma$ 를 보다 높게 설정하였기 때문에, 적은 전후진의 조작으로 주차를 완료하는 경로를 주었다. 위의 시뮬레이션을 통하여 파라미터의 설정에 따라 선호되는 경로를 선택할 수 있음을 알 수 있다. 하지만 점선 경로의 경우는 그 이전의 경로보다 벽에 가깝게 주행하여 실선 경로보다 위험한 경로임을 알 수 있다. 위의 시뮬레이션 결과로 shifting과 safety는 상계관계(trade off)의 관계에 있음을 알 수 있다.

#### 4.2 평행 주차

평행 주차 시뮬레이션을 위해 표 3, 표 4와 같이 파

표 3. 평행 주차 시뮬레이션 환경 및 파라미터

$\theta_{inlet}$	10°	Parking lot Env.	$a$	6.5m
$depth_{max}$	3		$b$	6.5m / 6.3m
Boundary Condition	$q_{goal}$	[-4m, 0m, 0°]		
	$q_{int}$	[2.5m, 10m, 90°]		

표 4. 평행 주차 시뮬레이션의 파라미터 값

Weights	$\alpha$ [1/mm]	$\beta$	$\gamma$	$\delta$ [1/mm]
Set 3	0.2	1	10	0.0001

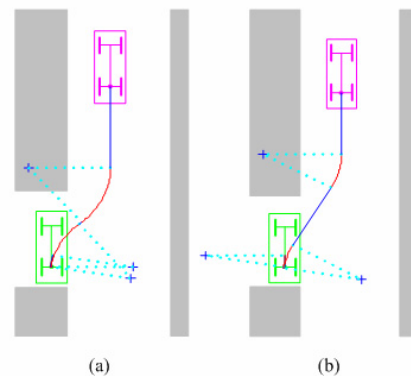


그림 11. 평행 주차 시뮬레이션 결과

라미터를 설정하였다. 다른 파라미터는 직각 주차의 파라미터와 같다. 평행 주차의 경우는 주차 공간의 크기가 주차의 난이도를 결정하는 큰 요소이기 때문에 이 폭(그림 1의 b)을 변화시키며 시뮬레이션 수행하였다. 평행 주차의 경우에는 주차 공간이 작게 되면 전후진의 조작 횟수가 직각 주차보다 더 빈번히 일어나게 된다. 이 경우에는 짧은 거리를 앞뒤로 움직이는 경우가 많기 때문에, safety보다 maneuvering의 횟수가 더 중요한 요소가 된다.

그림 11은 주차 공간의 크기가 6.5m, 6.3m 일 때 두 경우의 simulation 결과를 보여주고 있다. 두 경우 모두 약 20여 개의 path set을 찾아내었고 약 60여 개의 경로를 생성하였다. 파라미터를 maneuvering에 대해 높게 설정하였기 때문에, 전후진의 조작이 적으면서 안전한 경로를 최종 경로로 결과로 주고 있다. 그림 11-(a), (b) 두 경우 모두 주차 공간의 크기가 작아서 전후진의 조작 없이 빠져나가지 못하는 경우로, 2회 이상의 전후진 조작을 포함하고 있다. 이렇게 maneuvering이 나타나는 이유는 비록 주차 공간의 크기가 maneuvering이 없이 주차를 수행할 수 있는 크기가 되어도, 그 위치가 매우 제한적이기 때문에 대부분의 경우에 maneuvering이 나타나는 것이다.



표 5. 평행 주차 시뮬레이션 환경 및 파라미터

Parking lot Env.	$a$	6.5m	
	$b$	Garage	2.3m
Parallel		6.5m	
Boundary Condition	$q_{goal}$ [x[m], y[m], $\theta$ [°]]	Garage	[-4, 0, 0]
		Parallel	[-9.15, 2.1, 90]
	$\theta_{ini}$	90°	

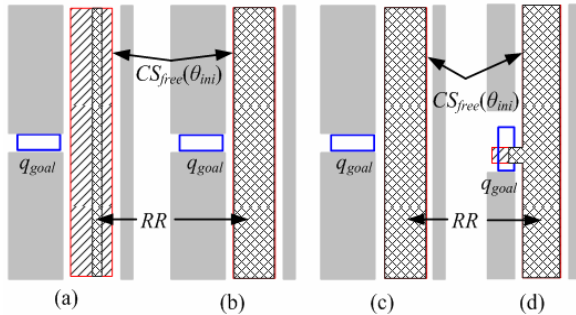


그림 11. Coverage area (diagonal cross-hatched area),  $CS_{free}(\theta_{ini})$  (45° diagonal hatched area), (a) garage parking case  $n_{max} = 1$  (22.7% of  $CS_{free}(\theta_{ini})$ ), (b)  $n_{max} = 2$  (95.5% of  $CS_{free}(\theta_{ini})$ ), (c)  $n_{max} = 3$  (97.8% of  $CS_{free}(\theta_{ini})$ ), (d) parallel parking case  $n_{max} = 3$  (95.2% of  $CS_{free}(\theta_{ini})$ )

#### 4.3 Coverage area

제안된 경로 계획기의 주차 공간에서의 유용성을 증명하기 위해, depth에 따른 Coverage area를 구하였다. Coverage area는 초기 방향( $\theta_{ini}$ )를 알고 있다고 가정하였을 때, 방향이 초기 방향( $\theta_{ini}$ )과 같은 도달 가능 영역을 합한 것이다. 이 도달 영역 안에 초기 위치가 있다면 이 영역은 *GRSR*이 되고 *path set*을 생성할 수 있다. Coverage area를 구하기 위한 다른 조건은 표 5에 나타나 있다.

그림 12-(a) ~ (c)는 직각 주차의 경우 Coverage area를 나타내고, 그림 12-(d)는 평행 주차의 Coverage area를 나타낸다. 직각 주차의 경우에 depth가 1인 경우(하나의 모션 세트만 허용) Coverage area가 22.7%였고, depth가 2(2개의 모션 세트까지 허용)인 경우 Coverage area가 95.5%였다. 위의 결과로써 직각 주차의 경우에는 2개의 motion set만 허용하여도 대부분 경로를 생성할 수 있음을 알 수 있다. 만약 depth를 3인 경우(3개의 모션 세트까지 허용) 약 97.8%의 영역에서 초기 경로를 생성할 수 있다. Depth 3까지 허용하는 것은 다양한 모션을 얻어내기 위해서는 좋은 일이지만, 계산 부하가 크다는 상계관계(trade off)가 있다. 그림 12-(d)의 평행 주차의

경우에는 depth가 3일 경우(3개의 모션 세트를 허용) 95.2%의 영역에서 초기 경로를 생성할 수 있음을 알 수 있다. 위의 결과를 통해 제안된 방법이 주차 공간에서 적은 수의 모션만으로 대부분의 영역에서 주차 경로를 생성할 수 있음을 알 수 있다.

## 5. 결론

본 논문에서는, M-Space를 이용한 자동차의 주차 경로 생성 방법을 제안하였다. M-Space는 자동차의 모양과 운동을 모두 고려한 무 충돌 영역으로 기존의 C-space를 확장한 개념이다. 이 개념을 이용하여 도달 영역의 계산과 역 확산 방법을 통하여 충돌이 없고 논홀로노믹 구속조건을 만족하는 자동차의 경로를 생성하였다. 또한 모션의 개수를 제한하여 가능한 모든 경우의 path를 얻어내고, safety와 shifting 등의 기준을 바탕으로 Cost를 경로에 부여하는 방법으로, 여러 경로 중 가장 주차에 적절한 경로를 생성하였다. Coverage area를 통하여, 주차의 특성을 고려하여 적은 depth로도 대부분의 주차 공간에서 경로를 생성할 수 있음을 보였다. 시뮬레이션 결과를 통해 제안된 방법의 유효성을 검증하였다.

## 참고 문헌

- [1] "Robot Motion Planning," J. C. Latombe, Kluwer Academic Publishers, 1993.
- [2] R. M. Murray, S. S. Sastry "Nonholonomic Motion Planning: Steering Using Sinusoids" *IEEE trans. on Automatic Control*, Vol. 38 No. 5, 1993.
- [3] D. Tilbury, R. Murray and S. Sastry, "Trajectory generation for the n-trailer problem using Goursat normal form", *IEEE Trans. on Automatic Control*, Vol. 40 (5), pp. 802-819, 1995.
- [4] J. Barraquand and J.C. Latombe, "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles", *Algorithmica*, Springer Verlag, Vol. 10, pp. 121-155, 1993.
- [5] "Computational Geometry in C", O'Rourke, J., Cambridge U. Press, New York, 1994.
- [6] Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, "M. Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces." *IEEE Transactions on Robotics and Automation*,

- 12(4):566–580, 1996.
- [7] “Robot motion Planning and Control”, J.-P. Laumond, Ed., Lectures Notes in Control and Information Sciences 229, Springer, ISBN 3-540-76219-1, 1998.
- [8] J. A. Reeds and R. A. Shepp, “Optimal paths for a car that goes both forward and backwards.” *Pacific Journal of Mathematics*, 145 (2), pp. 367-393, 1990.
- [9] J.-P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. “A motion planner for nonholonomic mobile robots.” *IEEE Trans. Robot. Autom.*, 10(5), October 1994.
- [10] “Computational Geometry: Algorithms and Applications”, Mark de Berg, M. van Krefeld, M. Overmars, O. Schwarzkopf, Springer, 1999



### 김 달 형

- 2005 고려대학교 기계공학과 (공학사)  
2007 고려대학교 기계공학과  
기계공학 전공 (공학석사)

2007~현재 고려대학교 기계공학과 연구원  
관심분야: Nonholonomic motion planning



### 정 우 진

- 1993 서울대학교 기계공학과 (공학사)  
1995 동경대학교 기계공학과 (공학석사)  
1998 동경대학교 기계공학과 (공학박사)

2005~현재 고려대학교 기계공학과 교수  
관심분야: 이동로봇 위치추정 및 주행제어, 논홀로노믹  
시스템 제어, 다지 로봇 손