

이동 로봇을 위한 실시간 충돌 회피 궤적 계획과 제어

A Real-Time Collision-Free Trajectory Planning and Control for a Car-Like Mobile Robot

이수영, 이석한, 홍예선
(Soo-Yeong Yi, Suk Han Lee, and Yeh Sun Hong)

Abstract : By using the conceptual impedance and the elasticity of a serial chain of spring-damper system, a real-time collision-free trajectory generation algorithm is proposed. The reference points on a trajectory connected by the spring-damper system have a mechanism for self-position adjustment to avoid a collision by the impedance, and the local adjustment of each reference point is propagated through the elasticity to a real robot at the end of the spring-damper system. As a result, the overall trajectory consisting of the reference points becomes free of collision with environmental obstacles and efficient having the shortest distance as possible. In this process, the reference points connected by the spring-damper system take role of virtual robot as global guidance for a real robot, and a cooperative optimization is carried out by the system of virtual robots. A control algorithm is proposed to implement the impedance for a car-like mobile robot.

Keywords : collision-free trajectory, impedance control, reference adaptatin, elacticity of the spring-damper system

I. 서론

로봇 작업 계획에 있어서 충돌 회피 궤적 계획은 미리 주어진 시점과 종점을 지나면서 궤적 상에 장애물과의 충돌이 없는 한 연속 궤적을 찾아내는 것이다. 기존의 충돌 회피 궤적 계획 방법들은 대개 경로의 최적성에 초점을 맞춘 오프라인 경로 계획과 계획된 경로에 시간 정보를 설정하는 시간 계획으로 분리되어 이루어진다. 오프라인 방식에 의한 궤적 계획은 장애물등의 주변 상황 정보가 미리 알려져 있어야 하므로, 정보가 완전하지 않거나, 혹은 장애물의 움직임 때문에 발생할 수 있는 주변 상황 모델의 불확실성에 대한 적응성이 없다. 또한 오프라인 방식에서는 대개 대상 시스템의 동특성을 고려하지 않고 궤적을 계획하므로, 실제의 액츄에이터 궤환 제어에서 추종하기 어려운 궤적이 생성될 수도 있다. 그러므로 궤적의 최적성은 다소 희생하더라도, 불확실한 환경에 대해 실시간으로 적용할 수 있고 액츄에이터 궤환 제어시에 추종하기 쉬운 궤적을 얻기 위해서는 대상 시스템의 동특성을 고려한 온라인 방식의 궤적 계획 알고리즘이 필요하다. Khatib 등은 일반적으로 상위의 오프라인 궤적 계획부에서 수행되던 충돌 회피 궤적 계획을 구동기 제어 문제로 재편하므로써, 제어 대상 시스템의 동특성을 고려한 실시간 충돌 회피 궤적 계획 및 제어 알고리즘을 제안한 바 있다[3]. 이는 충돌 회피 위치 조정을 위해, 장애물과 로봇 사이의 거리에 의존하는 인공의 위치 에너지 함수를 이용하여 충돌력을 생성하고, 이를 구동기의 제어 신호로 변환해준 것이다. 그러나 위치 조정을 위한 충돌력은 주로 로봇 바로 주변의 장애물에 의해서 생성되므로, 전체적으로는 여러 국부 최소점

(local minimum)이 존재하게 되고, 따라서 전체 궤적의 효율성이 떨어지게 된다.

임피던스 제어 방식은 불확실한 환경에 대해서 실시간으로 측정된 힘 정보에 따라 로봇의 현재 위치를 조정해나가는 로봇 힘 제어의 한 방식이다[7]. 임피던스에 의한 힘 제어는 불확실한 환경에서 로봇과 주변 환경간의 간섭을 피하기 위한 하나의 실시간 위치 조정 알고리즘으로 이해될 수 있다. 따라서 로봇 궤적 계획에 있어서 장애물과 로봇간의 거리를 가상의 힘으로 정의하면, 불확실한 주변 상황에서의 실시간 충돌 회피 궤적 계획에 임피던스 제어 개념을 적용할 수 있다[8]. 또한 임피던스 제어는 스프링과 댐퍼 계수의 조정과 resolved motion 제어 알고리즘을 이용하여 실제 제어 대상 시스템의 동특성을 반영할 수 있다. 그러나 임피던스 제어 방식에 있어서 충돌 회피를 위한 위치 조정도 장애물과 로봇이 상호 작용하는 부근에서 국부적으로 이루어지게 되므로, 필연적으로 로봇의 움직임에 시간 지연과 요동(jerk)이 발생하게 되고, 따라서 결과적인 전체 궤적은 비효율적이며 추종하기 어려운 것이 되기 쉽다.

임피던스의 조정 개념을 실제 로봇 뿐만 아니라, 로봇이 따라가야할 궤적상의 모든 참조점들에 적용하므로써 전체 궤적의 위치를 적응, 조정해 나가는 방식을 생각해 볼 수 있다. 이 경우 모든 참조점들의 위치 조정은 실제 로봇 궤적을 미리 길안내하는 역할을 하므로, 단순 임피던스 개념에서 로봇의 위치 조정이 국부적이기 때문에 발생했던 시간 지연과 요동 문제를 피하므로써, 전체적으로 보다 효율적인 충돌 회피 궤적을 얻을 수 있다. 본 논문에서는 임피던스의 개념을 로봇이 따라가야할 궤적상의 모든 참조점들로 확장하기 위해 직렬 스프링-댐퍼 시스템의 탄성 개념을 이용한, 효율적인 실시간 충돌 회피 궤적 계획 알고리즘을 제시한다. 또한 비선형성이 일

접수일자 : 1998. 3. 16., 수정완료 : 1998. 10. 28.

이수영, 홍예선 : 한국과학기술연구원

이석한 : 삼성종합기술원

는 자동차형의 이동 로봇에 본 임피던스를 구현하기 위하여, 궤환 선형화 기법(feedback linearization)에 기반을 둔 제어 알고리즘도 제안한다. 2, 3절에서는 직렬 스프링-댐퍼 시스템의 특성과 함께 제안한 궤적 계획 알고리즘, 그리고 자동차형 이동 로봇에 대한 임피던스 구현 제어를 각각 설명하고, 4절에서는 제안한 알고리즘의 컴퓨터 시뮬레이션에 의한 결과와 그에 대한 분석을 제시한 후, 5절에서 결론을 맺는다.

II. 제안한 충돌 회피 궤적 계획 알고리즘

1. 임피던스 제어

임피던스 제어는 로봇과 주변 환경간의 간섭의 크기를 의미하는 힘과 위치 관계를 임피던스로 정의하고, 측정된 힘의 크기에 따라 다음과 같이 로봇의 위치를 조정하는 것이다.

$$k_a(\bar{p} - \bar{p}_d) + k_v(\dot{p} - \dot{p}_d) + k_p(p - p_d) = f(p) - f_d(p) \quad (1)$$

여기서 p 와 f 는 각각 위치와 힘 벡터를 나타내며, 아래 첨자 a 는 미리 주어진 양임을 의미한다. 이제 충돌 회피 궤적을 얻기 위해서, $f_d = 0$ 으로 하자. 그러면, (1)은 로봇과 주변 환경과의 간섭에 의한 힘이 없을 때, 즉 $f(p) = 0$ 일 때는 $p \rightarrow p_d$ 이므로 미리 주어진 궤적을 추종하지만, 어느 시점 $t = t_k$ 에서 $f(p(t_k)) \neq 0$ 인 경우, 즉 간섭이 있는 경우는 실제 위치 p 가 p_d 로 부터 측정된 양 f 에 따르는 편차를 가짐을 의미한다. 즉 불확실성이 있는 주변 환경에서, 미리 설계된 p_d 가 주변 환경과의 간섭이 있을 때, 간섭에 의한 힘 $f(t)$ 정보를 이용하여, 간섭을 피하는 위치 p 를 결정하는 것이다[7]. 이제 액츄에이터 제어부에 대한 위치 명령은 미리 주어진 p_d 가 아닌 조정된 p 가 된다. 이동 로봇의 경우에는, 충돌에 의한 힘 f 를 한 예로서 이동 로봇의 거리 센서로부터 측정되는 장애물과의 거리에 따라 반비례하는 가상적인 힘으로 정의한다면, 미리 주어진 p_d 에 충돌이 있을 때, 충돌력 f 에 따라 (1)식에 의하여 편이된 p 는 충돌이 없는 것이 된다.

그러나 임피던스에 의한 위치 조정은 충돌이 발생하는 부근에서만 일어나므로 로봇의 움직임에 시간 지연과 요동이 발생할 수 있고, 따라서 결과적인 전체 궤적은 추종하기 어렵고 비효율적인 것이 될 수 있다. 그러나, 그림 1에서 설명하는 것 처럼 현재 위치 p 뿐만 아니라 주어진 궤적상의 앞으로 추종해야할 모든 참조점, 즉 $p_d(t), t \geq t_k$ 자체를 얻어진 힘 정보에 따라 조정한다면, 궤적 계획에 있어 보다 광역적인 접근을 할 수 있을 것이다. 그림 1에서 흰점들은 검은 점으로 표현된 실제 로봇이 따라가야할 궤적상의 참조점들을 나타내며, 회색 점들은 실제 로봇의 자취를 나타낸다.

2. 직렬 스프링-댐퍼 시스템

물리적으로 임피던스 (1)은 p_d 와 p 사이에 스프링-댐퍼 시스템이 있는 것으로 이해될 수 있다. 그리고 p_d 는 미리 정해진 궤적을 갖는다. 이제 이를 확장하여 그림 1에서 보이는 것과 같이 p 와 p_d 사이 뿐만 아니라 미리

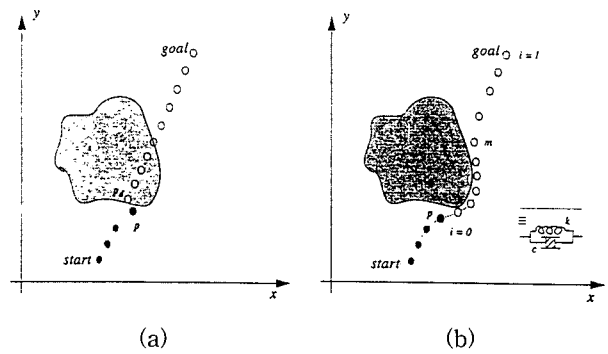


그림 1. 충돌 회피를 위한 위치 조정 (a) 단순 임피던스의 경우, (b) 직렬 연결된 스프링-댐퍼 시스템의 경우.

Fig. 1. Position adjustment for collision avoidance (a) in case of the pure impedance, (b) in case of the serial chain of spring-damper system.

주어진 궤적 상의 모든 참조점 p_d 들 사이에도 스프링-댐퍼가 연결되어 있는 시스템을 생각하자.

위 시스템의 뉴턴 운동 법칙에 따른 수학적 모델은 다음과 같다:

$$\begin{aligned} \ddot{p}_i + 2\frac{c}{m}\dot{p}_i + 2\frac{k}{m}p_i &= \frac{c}{m}(\dot{p}_{i+1} + \dot{p}_{i-1}) + \frac{k}{m}(p_{i+1} + p_{i-1}) + \frac{1}{m}f_i(p_i) \quad i = 1, \dots, I-1 \\ \ddot{p}_i + \frac{c}{m}\dot{p}_i + \frac{k}{m}p_i &= \frac{c}{m}\dot{p}_{i+1} + \frac{k}{m}p_{i+1} + \frac{1}{m}f_i(p_i) \quad i = 0 \end{aligned} \quad (2)$$

여기서 m, c, k 는 각각 로봇의 질량, 댐퍼의 마찰계수, 스프링 상수를 나타내며, 중점에 놓인 점의 속도 및 가속도는 당연히 $\dot{p}_I = \ddot{p}_I = 0$ 이다. 이제 p_0 을 실제 로봇의 위치 벡터, 그리고 $p_i = [x_i, y_i]^T$ 들을 실제 로봇이 참조해야할 가상 로봇들의 위치 벡터라고 생각하자. 각 가상 로봇은 스프링-댐퍼 연결에 의해 앞뒤의 다른 가상 로봇과 임피던스 관계를 갖고 있다. 따라서 가상 로봇이 주변 환경과의 간섭으로부터 느끼는 힘 $f(p_i)$ 은 가상 로봇 자신의 위치 조정에 영향을 줄 뿐만 아니라, 이것이 스프링-댐퍼 시스템의 탄성(elasticity)을 통해 전달되어 실제 로봇의 위치 조정에도 영향을 주게 된다. 결과적으로 실제 로봇의 궤적은 (1)과 같은 단일 임피던스에 의한 국부 위치 조정 보다 광역적인 충돌 회피 위치 조정이 이루어지며, 또한 탄성력은 항상 가능한 한 거리가 짧은 궤적을 따르는 경향이 있으므로 궤적 계획의 광역 최적성을 추구할 수 있다.

3. 장애물 모델링을 위한 신경망을 이용한 위치 에너지장
가상의 로봇들은 궤적 계획 알고리즘 내에만 존재하는 것이므로 결국 주변 환경에 대한 정보를 얻는 것은 실제 로봇의 센서 시스템에 의한다. 실제 로봇은 거리 센서등을 통하여 실시간으로 주변 환경의 장애물에 대한 기하학 모델 정보를 얻어내고, 가상 로봇들의 충돌 회피

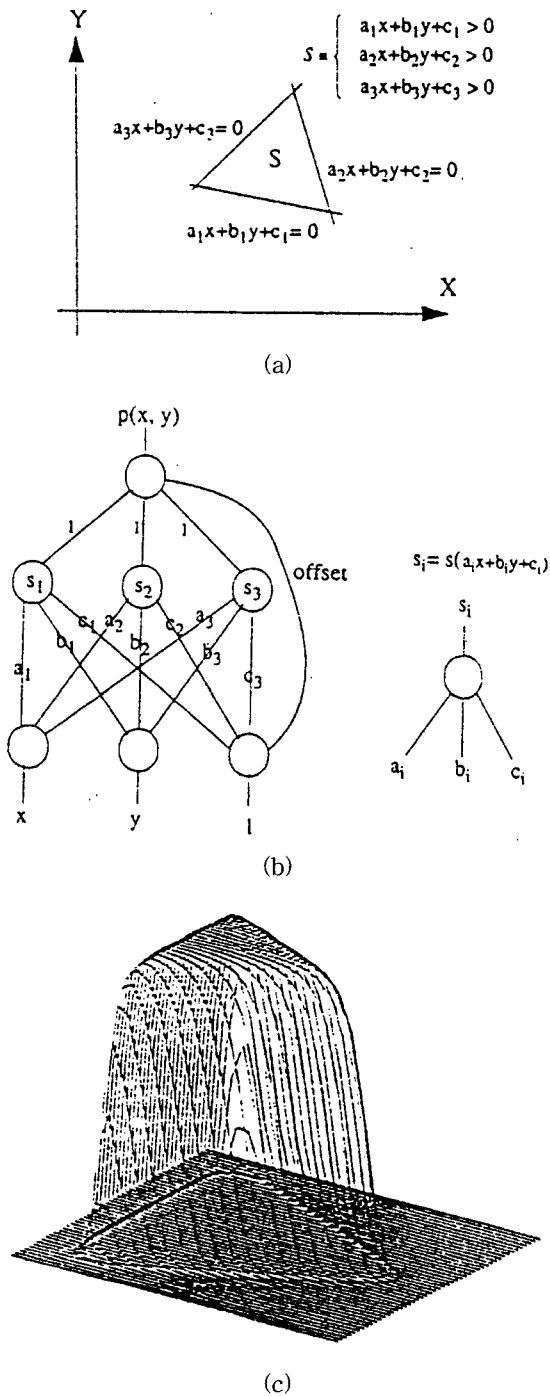


그림 2. 인공 신경망에 의한 충돌 영역의 모델 (a) 충돌 영역, (b) 인공 신경망 모델, (c) 충돌의 위치 에너지.

Fig. 2. Modeling of collision region by using the artificial neural network (a) region of collision, (b) neural network model, (c) potential energy for collision.

위치 조정은 이들 정보를 이용하여 이루어지는 것이다. 로봇과 장애물간의 충돌을 표현하기 위한 장애물 모델링에 관한 방법으로는 다음과 같은 것들이 있다:

1) 장애물을 다각형으로 모델링하고 이것과 로봇간의 최단 거리에 의해서 충돌을 표현하는 방법[3]

2) 작업 영역을 2차원적으로 이산화하여 각 셀에 장애물의 존재 유무에 관한 코드를 부여하고, 장애물이 존재하는 셀과 로봇 간의 거리에 의해서 충돌을 표현하는 방법[1]

3) 작업 영역을 B-spline patch 등에 의한 연속 벡터 함수로 모델링하고, 높이를 나타내는 요소에 의해서 장애물을 표현하는 방법[2]

4) 충돌 영역에 인공의 위치 에너지 함수(artificial potential field)를 설정하고, 위치 에너지의 크기에 따라 충돌을 표현하는 방법[3][4]

특히 [4]에서는 인공의 위치 에너지를 설정하기 위한 함수로, 필요한 데이터 베이스의 크기에 비해 표현 능력이 뛰어난 인공 신경망(artificial neural network)을 사용하는 방법을 제시하였다. 또 일반적으로 많은 계산량의 전처리(pre-processing)가 필요한 학습 과정을 피하고, 인공 신경망의 내재된 특성을 이용함으로써 체계적인 방법으로 실시간에 장애물의 인공 위치 에너지를 표현하는 방법을 개발하였다. 본 논문에서는 실시간 알고리즘의 개발이라는 목적에 부합하는 [4]에서의 방법을 채택하고 이를 간단히 설명한다. 먼저 그림 2 (a)에서 보듯이 장애물이 불록 다각형으로 모델링된다고 하면, 다각형의 내부 영역 S는 그림 2 (a)의 오른쪽에 있는 부등식들을 만족한다. 여기서는 미리 장애물의 위치와 모양이 주어져 있다고 가정하지만, 전술한 바와 같이 실제 로봇에 장착된 거리 센서와 데이터 처리 알고리즘을 이용하여 실시간으로 장애물의 다각형 모델을 구할 수 있다. 이제, 신경망의 활성화 함수를 (3-1)과 같은 단순한 부호 함수(signum function)이라고 하면, 그림 2 (b)의 구조를 갖는 인공 신경망의 은닉층의 출력이 각각의 부등식에 대응되므로, 신경망의 출력, $P(p_i)$ 는 모든 부등식을 만족할 때에만, 즉 점 p_i 가 영역 S내에 있을 때에만 1이 되고, 그렇지 않은 경우에는 0이 된다. 그러므로 활성화 함수를 (3-2)와 같은 연속적인 함수(sigmoid function)로 하면 출력 $P(p_i)$ 는 그림 2 (c)에 보이는 것과 같은 연속적인 양을 갖게 되고, 이는 어떤 점 p_i 가 장애물 영역에 속하는지 여부를 판단할 수 있는 지수가 되므로 이를 충돌의 위치 에너지로 이용할 수 있다.

$$s(z) = \begin{cases} 1 & z \geq 0 \text{ 일 때} \\ 0 & z < 0 \text{ 일 때} \end{cases} \quad (3-1)$$

$$s(z) = \frac{1}{1 + e^{-z/T}} \quad (3-2)$$

4. 시스템에 관계된 에너지 표현

그림 1 (b)에 보인 직렬 스프링-댐퍼 시스템에 관련된 물리적 에너지는 다음과 같다:

$$E = \sum_{i=0}^{n-1} \frac{1}{2} m \dot{p}_i^t \dot{p}_i + \sum_{i=0}^{n-1} \frac{1}{2} k (p_{i+1} - p_i)^t (p_{i+1} - p_i) + \sum_{i=0}^{n-1} P_i(p_i) \quad (4)$$

(4)의 첫 번째와 두 번째 항은 각각 점 질량의 운동 에너지와 스프링의 탄성 에너지를 의미하며, 마지막 항은

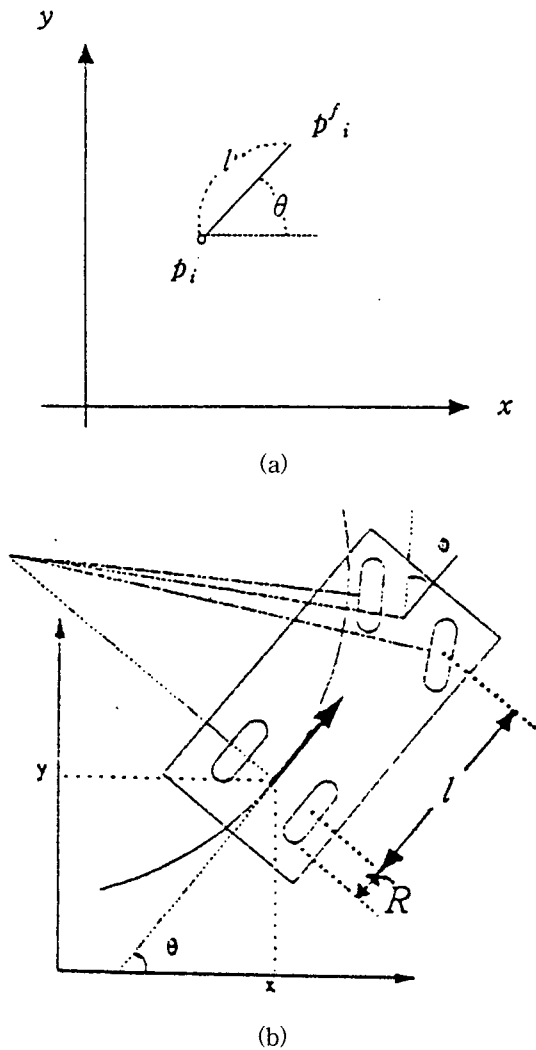


그림 3. 방향을 갖는 이동 로봇의 모델 (a) 자동차 형태의 이동 로봇, (b) 기하학적 모델.
 Fig. 3. Geometric model of a mobile robot with heading (a) a car-like model, (b) geometric model.

충돌을 나타내는 위치 에너지를 의미한다. 만일 이동 로봇을 일정한 크기를 갖는 다각형으로 모델링한다면, 다각형내의 모든 관심 점, 즉 한 예로서 모든 꼭지점들의 위치 에너지의 합으로 충돌을 정의할 수 있다. 본 논문에서는 이동 로봇이 그림 3(a)에서 처럼 방향을 갖는 일정 길이의 선분이라고 단순 가정하고 충돌 여부의 관심 점을 선분의 양 끝점으로 한다. 그러므로 충돌의 위치 에너지는

$$P_i(\mathbf{p}_i) = P(\mathbf{p}_i) + P(\mathbf{p}_i^f) \quad (5)$$

이다. 여기서 선분의 다른 한 끝점 \mathbf{p}_i^f 는 방향 θ_i 에 대해 $\mathbf{p}_i^f = R(\theta_i) \mathbf{L} + \mathbf{p}_i$ 로 구해진다. 이 때 $R(\theta_i)$ 는 회전 행렬, $\mathbf{L} = [l, 0]^t$, l 은 이동 로봇의 길이를 나타낸다. (2)로부터 간단한 연산에 의해 (4)의 시간 미분은 $\mathbf{f}_i(\mathbf{p}_i) = -\frac{\partial P_i(\mathbf{p}_i)}{\partial \mathbf{p}_i}$ 로 할 때, 즉 충돌력 벡터를 충돌 위치 에너지의 음 경사 방향이 되도록 할 때,

$$\frac{dE}{dt} = -c \sum_{i=0}^{n-1} (\dot{\mathbf{p}}_{i+1} - \dot{\mathbf{p}}_i)^t (\dot{\mathbf{p}}_{i+1} - \dot{\mathbf{p}}_i) \leq 0 \quad \forall t \quad (6)$$

임을 구할 수 있다*. 이는 직렬 스프링-댐퍼 시스템에 관련된 에너지가 항상 감소하는 방향으로 로봇들이 움직임을 의미하는데, 시스템에 관련된 에너지가 가장 작은 $E=0$ 인 경우는 $\mathbf{p}_0 = \mathbf{p}_1$, 즉 실제 로봇이 종점에 이르렀을 경우이다. 여기서 한가지 주의해야 할 것은 (6)의 결과가 $t \rightarrow \infty$ 에 따라 $E \rightarrow 0$ 임을, 즉 실제 로봇의 위치가 늘 $\mathbf{p}_0 \rightarrow \mathbf{p}_1$ 임을 의미하지는 않는다는 것이다. 이는 장애물과의 충돌력과 스프링의 탄성력이 평형을 이루는 국부 최소점들이 단순 임피던스의 경우 처럼 여전히 존재하기 때문이다. 그러나 직렬 스프링-댐퍼 시스템에서는 여러 가상 로봇들의 협동에 의한 최적화가 수행되므로, 단순 임피던스에 비해 국부 최소점을 벗어날 수 있는 가능성이 많고, 따라서 특별한 경우를 제외하고는 대개 $\mathbf{p}_0 \rightarrow \mathbf{p}_1$ 를 얻을 수 있다. 또한 스프링-댐퍼 시스템의 탄성은 최단 거리를 추구하는 경향이 있으므로 탄성 계수 및 마찰 계수의 적절한 선택에 따라 전체적으로 효율적인 궤적을 생성할 수 있다.

III. 자동차형의 이동 로봇에 대한 제어

1. 이동 로봇의 동역학 모델

구동과 방향 조절을 위한 동력으로 각각 전기 모터를 갖는 그림 3(b)의 자동차 형태의 이동 로봇 동역학은 다음 식으로 표현된다[5]:

$$\dot{x} = \cos \theta R \omega \cos \alpha_1 \quad (7-1)$$

$$\dot{y} = \sin \theta R \omega \cos \alpha_1 \quad (7-2)$$

$$\dot{\theta} = \frac{R}{l} \omega \sin \alpha_1 \quad (7-3)$$

$$\dot{\omega} = -H_2 \omega + u_1 \quad (7-4)$$

$$\dot{\alpha}_1 = \alpha_2 \quad (7-5)$$

$$\dot{\alpha}_2 = -H_1 \alpha_2 + u_2 \quad (7-6)$$

여기서 R 은 구동 바퀴의 반지름, l 은 앞 뒤 바퀴사이의 거리, H_1, H_2 는 두 모터-부하 시스템의 마찰 계수, ω 는 구동 바퀴의 각속도, α_1, α_2 는 각각 방향 조절 모터의 각도 및 각속도, u_1, u_2 는 각 모터의 제어 입력이다.

2. 이동 로봇의 제어

액튜에이터 제어부의 목적은 (7-1)-(7-6)으로 표현되는 이동 로봇의 비선형 동역학을 (2)의 스프링-댐퍼 시스템 처럼 움직이도록 만드는 데 있다. 구동 및 방향 조절 모터의 속도 변수에 대해 직교 좌표 공간상에서의 운동만을 기술한 이동 로봇의 단순 운동식에 대해 제한 선형화 기법에 기반을 둔 한 제어 알고리즘이 제안된 바 있는데[6], 여기서는 이를 확장하여 모터들의 동역학을 포함하는 전 시스템에 대해서 이동 로봇의 운동이 스프링-댐퍼 시스템 처럼 움직이도록 하는 제어를 제안, 설계한다.

* (2)와 (4)로부터 (6)에 이르는 과정을 부록에 서술하였다.

먼저 구동 모터와 방향 조정 모터의 속도 변수 v_1, v_2 를 (9-3), (9-4)와 같이 정의하면 (7-1)-(7-6)으로 이루어진 시스템은 (9), (10), (11)과 같이 크게 세부분으로 나누어 질 수 있다:

$$\dot{x}_i = \cos \theta v_1 \quad (8-1)$$

$$\dot{y}_i = \sin \theta v_1 \quad (8-2)$$

$$\dot{\theta} = v_2 \quad (8-3)$$

$$\dot{\alpha}_1 = \alpha_2 \quad (9-1)$$

$$\dot{\omega} = -H_2 \omega + u_1 \quad (9-2)$$

$$v_1 = R \omega \cos \alpha_1 \quad (9-3)$$

$$v_2 = \frac{R}{l} \omega \sin \alpha_1 \quad (9-4)$$

$$\dot{\alpha}_2 = -H_1 \alpha_2 + u_2 \quad (10)$$

여기서 x_i, y_i 의 아래 첨자 i 는 i 번째 가상 로봇을 의미하며, 나머지 제어 변수들은 모두 i 번째 로봇의 것에 해당한다. (8-1), (8-2)에 잘 알려진 케환 선형화 기법을 적용하므로써, 다음과 같은 직교 좌표 공간상에서 로봇의 운동식을 구할 수 있다[6]:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta v_1 \\ \sin \theta & \cos \theta v_1 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ v_2 \end{bmatrix} \quad (11)$$

여기서 행렬 $M_1 = \begin{bmatrix} \cos \theta & -\sin \theta v_1 \\ \sin \theta & \cos \theta v_1 \end{bmatrix}$ 의 $\det(M_1) = v_1$ 이고, 운동중에는 $v_1 \neq 0$, 즉 $\det(M_1) \neq 0$ 임에 주목하자. (11)을 (2)와 같은 스프링-댐퍼 시스템처럼 움직이도록 만들기 위해서는 $[\dot{v}_1, v_2]^t$ 가

$$\begin{bmatrix} \dot{v}_1 \\ v_2 \end{bmatrix} = M_1^{-1} \begin{bmatrix} g_x \\ g_y \end{bmatrix}, \quad (12-1)$$

$$\begin{bmatrix} g_x \\ g_y \end{bmatrix} =$$

$$\begin{bmatrix} -2\frac{c}{m} \dot{x}_i - 2\frac{k}{m} x_i + \frac{c}{m} (\dot{x}_{i+1} + \dot{x}_{i-1}) + \frac{k}{m} (x_{i+1} - x_{i-1}) + \frac{1}{m} f_x(\rho_i) \\ -2\frac{c}{m} \dot{y}_i - 2\frac{k}{m} y_i + \frac{c}{m} (\dot{y}_{i+1} + \dot{y}_{i-1}) + \frac{k}{m} (y_{i+1} - y_{i-1}) + \frac{1}{m} f_y(\rho_i) \end{bmatrix} \quad (12-2)$$

를 추종하도록 하면 된다. 즉 (12)는 (9)-(10)으로 이루어진 부 시스템에 대한 추종 명령이 된다.

이제 제어기 설계 문제는 (9)-(10)으로 구성된 제어 대상 시스템이 (12)로 주어지는 궤적을 추종하기 위한 제어 입력 $[u_1, u_2]^t$ 을 구하는 것이다. 비슷한 케환 선형화 방법으로 (9-3), (9-4) 식을 미분하고 (9-1), (9-2)를 대입하면 $[v_1, v_2]^t$ 에 관한 다음의 미분 방정식을 얻을 수 있다:

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = - \begin{bmatrix} RH_2 \omega \cos \alpha_1 \\ \frac{R}{l} H_2 \omega \sin \alpha_1 \end{bmatrix} + \begin{bmatrix} R \cos \alpha_1 & -R \omega \sin \alpha_1 \\ \frac{R}{l} \sin \alpha_1 & \frac{R}{l} \omega \cos \alpha_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (13)$$

$$= - \begin{bmatrix} H_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \frac{v_1}{\omega} & -l v_2 \\ \frac{v_2}{\omega} & \frac{v_1}{l} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

마찬가지로 운동중에 $M_2 = \begin{bmatrix} \frac{v_1}{\omega} & -l v_2 \\ \frac{v_2}{\omega} & \frac{v_1}{l} \end{bmatrix}$ 의 $\det(M_2) =$

$\frac{R^2}{l} \omega (\neq 0)$ 이므로 (13)의 상태 방정식이 (12)로 주어지는 $[\dot{v}_1, v_2]^t$ 를 추종하기 위해서는 (13)의 제어 입력 $[u_1, u_2]^t$ 이

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = M_2^{-1} \left\{ \begin{bmatrix} H_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} - K_1 \begin{bmatrix} v_1 - v_{d1} \\ v_2 - v_{d2} \end{bmatrix} \right\} \quad (14)$$

이면 된다. 여기서 K_1 는 $diag[k_{11}, k_{22}]$, $k_{ij} > 0$ 이다. (14)의 첫 번째 행은 구동 모터에 필요한 제어 입력 u_1 을 결정한다. 그런데 (14)의 두 번째 행, u_2 는 다시 방향 조정 모터 운동식 (10)에서 제어 입력 u_2 의 종속 변수이다. 그러므로 (10)의 상태 변수 α_2 가 (14)의 두 번째 행으로 주어지는 α_{d2} 를 추종하기 위해 필요한 제어 입력, u_2 는 (10)에 같은 방법을 적용하므로써

$$u_2 = H_1 \alpha_2 + \dot{\alpha}_{d2} - k_2(\alpha_2 - \alpha_{d2}), \quad k_2 > 0 \quad (15)$$

로 구할 수 있다. 각 부시스템 (8), (9), (10)으로 구성되는 전체 제어 시스템의 구성은 다음 그림 4와 같다.

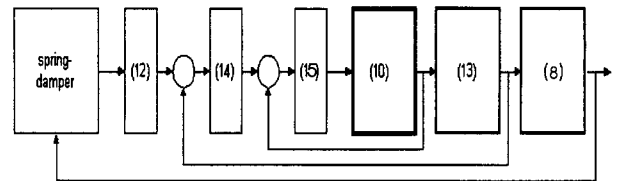


그림 4. 전체 제어 시스템의 신호의 흐름.
Fig. 4. Block diagram of the control system.

참고 1 : 변환된 제어 대상식 (13), (10)과 (14)의 첫 번째 행과 (15)로 주어지는 제어 입력 $[u_1, u_2]^t$ 로 이루어지는 폐루프 제어 시스템의 안정성은 다음과 같이 설명할 수 있다. 여기서 제어 시스템의 상태 변수는 $y = [v_1, v_2, \alpha_2]^t$ 로 이루어지며, (14)의 두 번째행은 제어 규칙을 구하기 위한 매개 변수, α_{d2} 이다. 편의를 위하여 다시 쓰면 제어 입력은

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = M_2^{-1} \left\{ \begin{bmatrix} H_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} - K_1 \begin{bmatrix} v_1 - v_{d1} \\ v_2 - v_{d2} \end{bmatrix} \right\} \quad (16)$$

$$u_2 = H_1 \alpha_2 + \dot{\alpha}_{d2} - k_2(\alpha_2 - \alpha_{d2}), \quad k_2 > 0 \quad (17)$$

이다. 이제 제어 시스템의 안정성을 보이기 위한 Lyapunov 함수 후보로 간단히

$$V = \frac{1}{2} e_v^t e_v + \frac{1}{2} e_{\alpha_2}^2 \quad (18)$$

를 취하자. 여기서 $e_v = [v_1 - v_{d1}, v_2 - v_{d2}]^t$ 이고 $e_{\alpha_2} = \alpha_2 - \alpha_{d2}$ 이다. (18)을 시간에 대해 미분하고, 상태(13)과 제어 입력 (16), (17)을 대입하면 다음과 같다:

$$\begin{aligned} \dot{V} &= \begin{bmatrix} e_{v_1} \\ e_{v_2} \end{bmatrix}^t \left\{ - \begin{bmatrix} H_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + M_2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} \dot{v}_{d1} \\ \dot{v}_{d2} \end{bmatrix} \right\} + e_{a_2} (H_1 a_2 + u_2) \\ &= \begin{bmatrix} e_{v_1} \\ e_{v_2} \end{bmatrix}^t \left\{ - \begin{bmatrix} H_2 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + M_2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} \dot{v}_{d1} \\ \dot{v}_{d2} \end{bmatrix} + M_2 \begin{bmatrix} 0 \\ e_{a_2} \end{bmatrix} \right\} \\ &\quad + e_{a_2} (H_1 a_2 + u_2) \\ &= \begin{bmatrix} e_{v_1} \\ e_{v_2} \end{bmatrix}^t \left\{ -K_1 \begin{bmatrix} e_{v_1} \\ e_{v_2} \end{bmatrix} + M_2 \begin{bmatrix} 0 \\ e_{a_2} \end{bmatrix} \right\} - k_2 e_{a_2}^2 \\ &= -e_v^t Q e_v \end{aligned} \tag{19}$$

여기서 $e_v = [e_{v_1}^t, e_{v_2}^t]^t$ 이고, $Q = \begin{bmatrix} k_{11} & 0 & m_{12} \\ 0 & k_{22} & m_{22} \\ 0 & 0 & k_2 \end{bmatrix}$ 이며,

m_{ij} 는 (13)의 입력 이득 M_2 의 (i, j) 요소이다. Q 는 양한정 행렬이므로 결과 (19)는 제어 대상 시스템 (13), (10)과 제어 입력 (16), (17)로 이루어지는 폐루프 제어 시스템이 임의의 궤적을 추종할 수 있음을 의미한다.

참고 2 : (7)로 주어지는 자동차형 이동 로봇의 동력학은 다음과 같은 nonholonomic 제한 조건을 갖는다:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \tag{20}$$

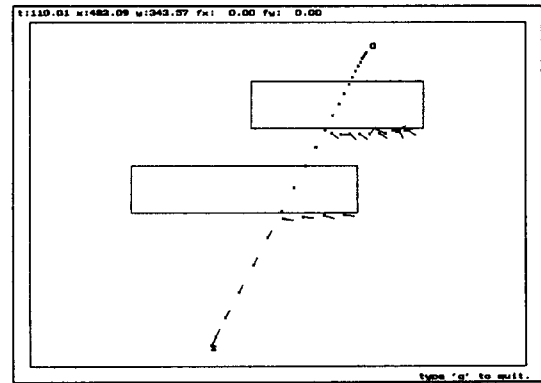
이는 어떤 방향각 θ 에서 움직일 수 있는 방향의 속도 벡터 $[\dot{x}, \dot{y}]^t$ 에 제한이 있음을 의미한다. 이 때문에 3차원의 상태 변수 $[x, y, \theta]^t$ 를 동시에 임의의 원하는 값으로 추종하도록 하는 것은 궤환 제어 알고리즘 자체만으로는 불가능하고 결국 궤적 계획 알고리즘과의 연계를 해결해야 할 문제가 된다[10]. 그러나 본 논문에서는 3차원의 상태 변수중 위치에 해당하는 $[x, y]^t$ 만의 추종 문제를 고려하고 있기 때문에 궤환 제어기 설계가 가능하다.

또 4륜의 자동차형 이동 로봇은 가능 회전 반경에 제한이 있다. 이는 결국 방향 조정 모터 입력 변수의 크기 제한 조건이 된다. 본 논문에서는 비록 명시적으로 가능 회전 반경 제한 조건을 다루지는 않았지만, 여러개의 가상 로봇으로 연결된 직렬 스프링-댐퍼 시스템은 각각의 급격한 움직임을 구속하고, 가상 로봇이 실제 로봇의 움직임을 미리 안내하므로 전체적으로는 본 제한 조건을 만족하려는 경향이 있다.

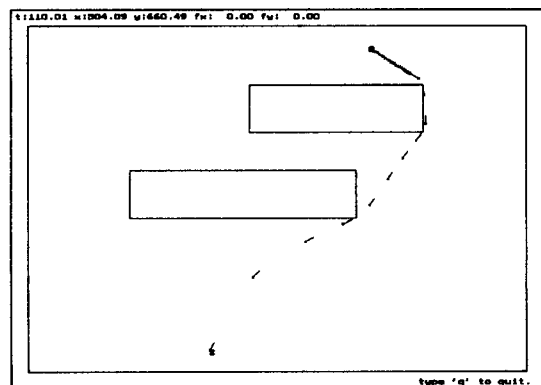
IV. 컴퓨터 시뮬레이션 및 결과 분석

제안하는 충돌 회피 궤적 계획 알고리즘 및 궤환 제어 알고리즘의 성능을 확인하기 위하여, 여러 상황의 주변 장애물에 대해서 컴퓨터 시뮬레이션을 행하고, 결과 궤적의 효율성을 단순 임피던스 알고리즘에 의한 것과 비교하였다. 제한한 알고리즘의 경우 모두 30개의 가상 로봇을 설정하였는데, 초기에는 장애물 유무에 상관없이 시점과 종점을 잇는 직선상에 등간격으로 배열하였다. 그리고 마지막 단에 있는 로봇만을 실제의 로봇으로 가정하여 (8), (9)의 동력학과 (14), (15)의 궤환 제어기를 적용하였고, 나머지의 가상 로봇들은 단순히 (2)로 주어지는 스프링-댐퍼 시스템의 동력학을 따르게 하였다. 스프링-댐퍼 시스템의 매개 변수들은 $m = 0.25$, $c = 0.5$, $k = 0.3$ 으로 하였고, (9-1), (10)의 모터 댐핑 상수 H_1 ,

H_2 는 각각 1.0으로 하였으며, 이동 로봇 기하 모델 변수 $R = 1.0$, $l = 10.0$ 으로 하였다. 그리고 (14), (15)의 제어기 상수는 단순히 $K_1 = \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix}$, $k_2 = 1.0$ 으로 설계하였다. 단순 임피던스 알고리즘의 경우는 (1)에서 $f_d(p) = 0$ 으로 하였고, 이 때 주어진 궤적 $p_d = [x_d, y_d]^t$ 는 시점부터 종점을 잇는 직선상을 등속으로 움직이는 것으로 하였다. 단순 임피던스 알고리즘의 경우 로봇의 운동에 실제 동력학 (8)-(9)를 적용하지 못하고 (1)로 주어지는 운동만을 시뮬레이션 하였는데, 이는 결과에 보이는 것처럼 궤적에 많은 요동이 생겨서 큰 제어 입력을 필요로 하고, 또 어느 경우 수치적으로 오버 플로우가 생기기도 하였기 때문이다. 그림 5, 7, 9에서 2차 평면상의 사각형들은 장애물을 의미하는데 II.3절에서 설명한 것처럼 장애물 영역에 인공 신경망을 이용한 위치 에너지를 형성하였고, 작은 'x' 점들은 실제 로봇이 따라가야 할 궤적 상의 참조점들을 나타내며, 실제 로봇의 방향과 위치를 짧은 선으로 표현하였다. 그림 5, 7에서 각각 단순 임피던스 알고리즘에 의한 시뮬레이션 결과에는 주어



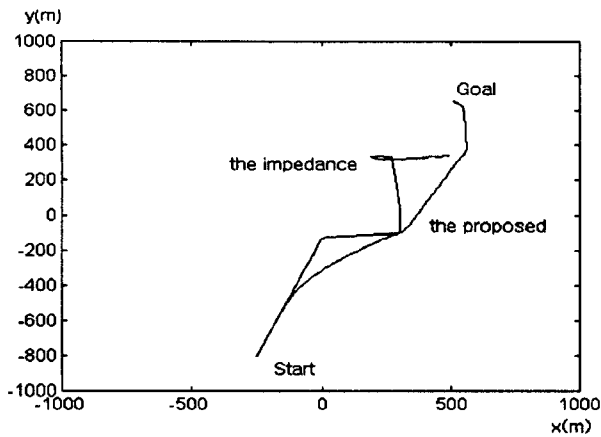
(a)



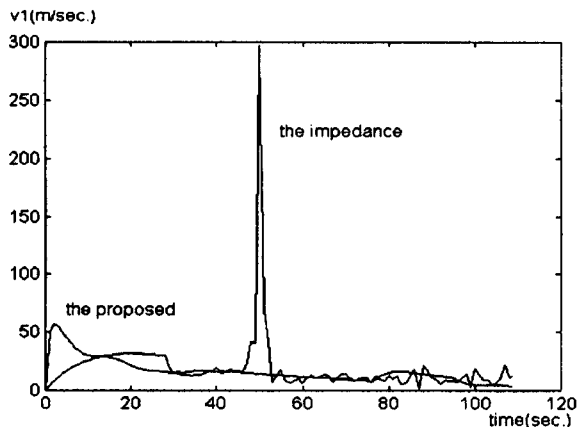
(b)

그림 5. 결과 궤적의 비교 : Time-slack, jerk, dead-end (a) 단순 임피던스 알고리즘, (b) 제안한 알고리즘.

Fig. 5. Comparison of the resulting trajectories: Time-slack, jerk, and dead-end (a) from the pure impedance algorithm, (b) from the proposed algorithm.



(a)

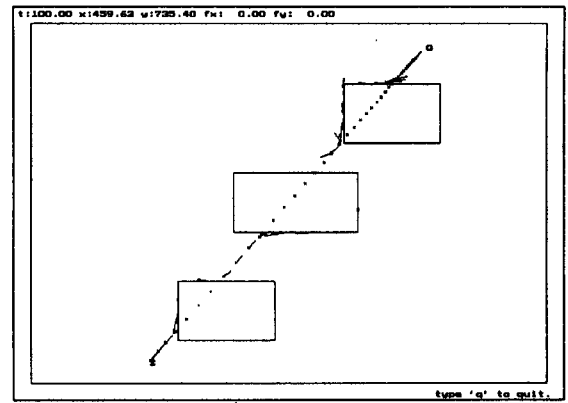


(b)

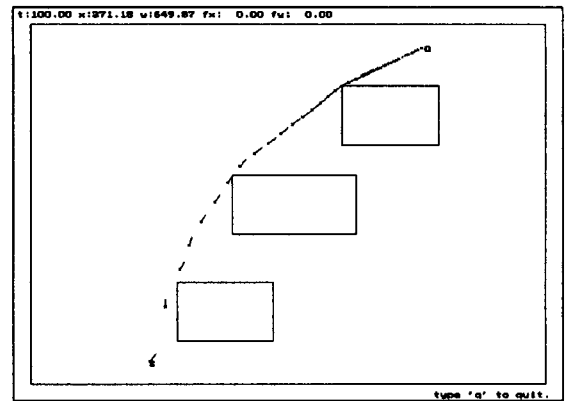
그림 6. 그림 4에 보인 결과 궤적의 시간에 따른 위치 및 속도 (a) 직교 좌표계상에서의 위치, (b) 속도.

Fig. 6. Position and velocity of the resulting trajectories in Fig. 4 (a) position trace in cartesian space, (b) velocity profile.

진 직선 궤적 d_d 와 함께 이동 로봇의 실제 자취를 나타 내었고, 제안한 알고리즘에 의한 시뮬레이션 결과에는 직렬 스프링-댐퍼 시스템의 종단에 있는 실제 이동 로봇의 자취만을 제시하였다. 그리고 그림 9에서는 제안한 본 알고리즘에서의 시간에 따른 가상 로봇들의 위치 조정 경향을 보이기 위해 실제 이동 로봇 뿐만아니라 가상 이동 로봇의 자취도 함께 나타내었다. 그림 5 (a)와 그림 6 (b)에서 보면 단순 임피던스 알고리즘에 의한 결과 궤적에 시간 지연과 요동이 있음을 알 수 있다. 그리고 그림 6 (b)에서 보인 속도의 피크는 단순 임피던스 알고리즘에 의한 결과 궤적이 실제로 제한 제어에 의해서 추종하기 어려운 것임을 의미한다. 한편 제안한 알고리즘에 의한 결과 궤적은 그림 5 (b)와 그림 6 (b)에서 보는 것처럼 장애물 부근에서의 시간 지연이나 요동이 없고, 따라서 추종하기 쉬운 궤적이 된다. 그림 6 (a)에서는 또 단순 임피던스 알고리즘에 의한 궤적이 목표점에 이르지 못하였음을 보여주는데, 이는 종말점(dead end)에서 목



(a)



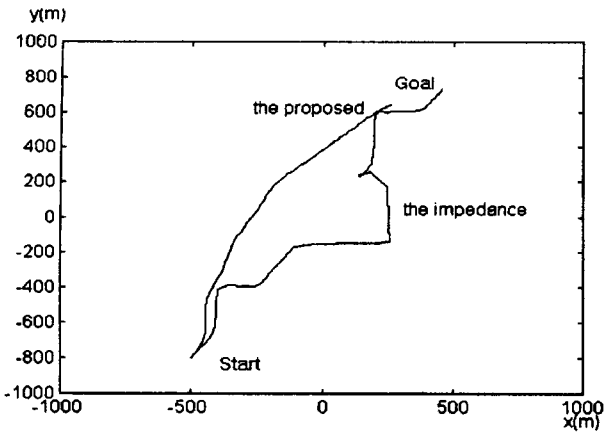
(b)

그림 7. 결과 궤적의 비교 : 협동 최적화 (a) 단순 임피던스 알고리즘, (b) 제안한 알고리즘.

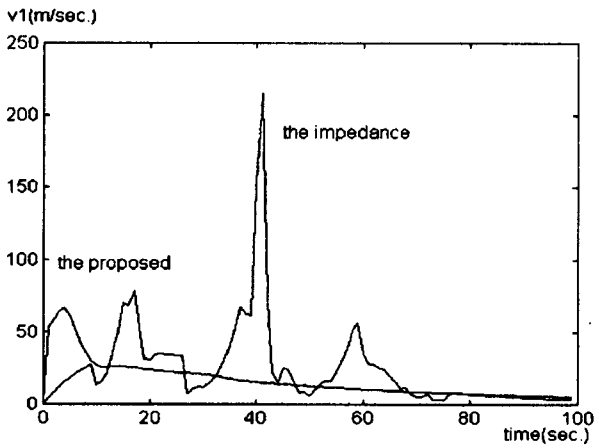
Fig. 7. Comparison of the resulting trajectories : cooperative optimization (a) from the pure impedance algorithm, (b) from the proposed algorithm.

표점이 당기는 힘과 장애물이 미는 힘이 평형을 이루는 국부 최소점에 빠졌기 때문이다. 그러나 같은 그림에서 보듯이 제안한 알고리즘에 의한 결과 궤적은 목표점에 이르렀다.

그림 7과 그림 8은 주변 장애물을 다르게 배치했을 때의 시뮬레이션 결과를 보여준다. 여기서 주목할 것은 단순 임피던스 알고리즘에 의한 결과 궤적의 길이보다 제안한 알고리즘에 의한 것이 짧으므로, 주행에 필요한 구동 에너지면에서 훨씬 효율적이라는 것이다. 이는 제안한 알고리즘에서 스프링-댐퍼 시스템의 탄성력에 의한 각 가상 로봇들의 협동 최적화의 결과이다. 이러한 탄성 시스템의 협동 최적화 특성은 각 가상 로봇들의 시간에 따른 움직임을 보여주는 그림 9에서 잘 나타나는데, 각 가상 로봇들은 각각이 따로 움직이는게 아니라 앞뒤의 다른 가상 로봇들과 같이 연계되어 움직이기 때문에 한 가상 로봇이 국부 최소점에 빠지려해도 다른 가상 로봇과의 연계에 의해서 국부 최소점을 벗어날 수 있는 가능성이 있다.



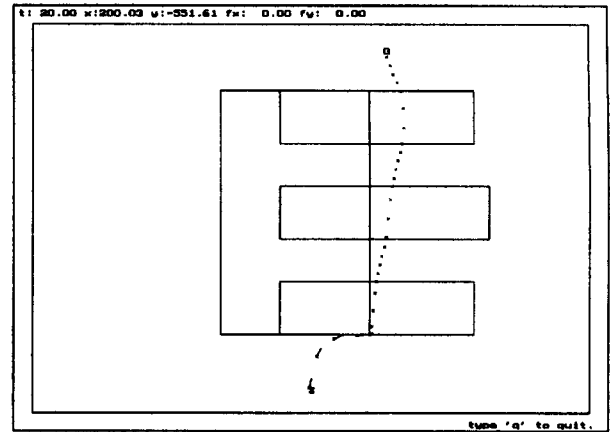
(a)



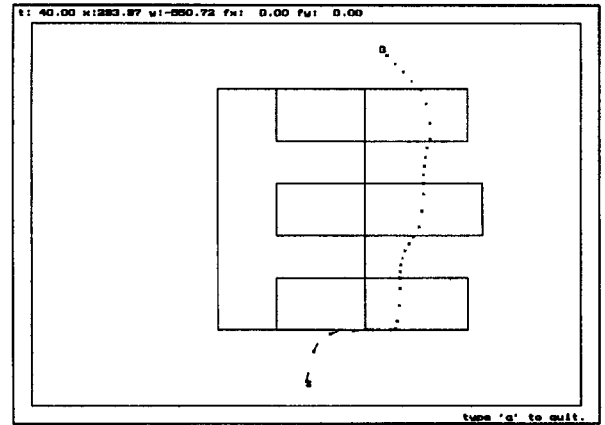
(b)

그림 8. 그림 6에 보인 결과 궤적의 시간에 따른 위치와 속도 (a) 직교 좌표계 상에서의 위치, (b) 속도.

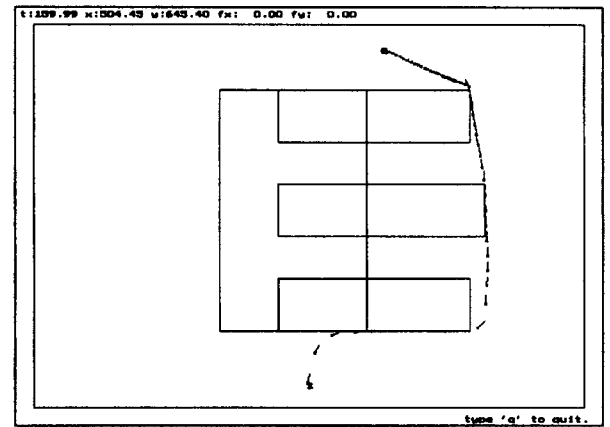
Fig. 8. Position and velocity of the resulting trajectories in Fig. 6 (a) position trace in cartesian space, (b) velocity profile.



(b)



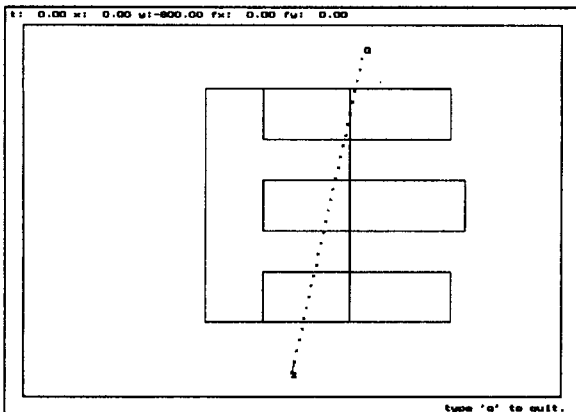
(c)



(d)

그림 9. 제안한 알고리즘에서 가상 로봇들의 시간에 따른 위치 조정 (a) 초기 배치 ($t=0.0$ 에서), (b) $t=20.0$ 에서, (c) $t=40.0$ 에서, (d) $t=160.0$ 에서.

Fig. 9. Position adjustment of the virtual robots according to time in the proposed algorithm (a) Initial configuration (at $t=0.0$ sec.), (b) At $t=20.0$ sec., (c) At $t=40.0$ sec., (d) At $t=160.0$ sec..



(a)

V. 결론

광역적으로 최적인 궤적을 찾아내는데는 항상 오프라인 계산이 필요하다. 오프라인 계산은 로봇 주변 환경에 대한 확실한 정보를 필요로 하므로, 오프라인 계산 결과에 의한 궤적의 최적성은 주변 환경의 불확실성에 의해서 크게 좌우될 수 있다. 그러므로 어떤 한 궤적 계획 및 제어 알고리즘의 평가는 결과 궤적의 최적성 뿐만 아니라 주변 장애물 환경의 변화나 불확실성에 대한 적응성도 함께 고려해야 한다. 본 논문에서는 직렬 스프링-댐퍼 시스템의 탄성에 개념적 바탕을 두고, 온라인 계산에 의한 적응성과 탄성에 의한 궤적의 효율성을 동시에 추구할 수 있는 실시간 충돌 회피 궤적 계획 알고리즘을 제안하였다. 본 알고리즘은 고전적으로 잘 알려져 있는 백트래킹이 필요한 피아노 이동 문제나 엄밀한 의미에서 nonholonomic 제한 조건을 만족시키는 궤적 계획 문제를 풀지는 못한다. 그러나 그러한 문제들을 풀기 위한 기존의 오프라인 접근 방식과는 달리 기본적으로 온라인 계산 구조를 가지고 있기 때문에 실시간 적용이 가능하며, 따라서 전술한대로 주변 환경의 불확실성에 대한 적응성이 있고, 또한 궤환 제어시에 추종하기 쉬운 궤적을 생성할 수 있다.

본 논문에서는 궤적 계획 알고리즘에 초점을 맞추기 위하여 주변 장애물의 형태나 위치등 기하학 모델이 미리 주어진 것으로 가정하였지만, 카메라 비전 센서나 링 구조의 초음파 거리 센서 배열등을 이용하여 실시간으로 주변 장애물에 대한 정보를 얻을 수 있다. 이 경우, 보다 현실적이고 실용성이 있는 실시간 궤적 계획 알고리즘을 구현할 수 있을 것이다.

참고문헌

[1] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179-1187, 1989.
 [2] Z. Shiller and Y. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Trans. Robotics and Automation*, vol. 7, no. 2, pp. 241-249, 1991.
 [3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The Int'l. Jour. of Robotics Research*, vol. 5, no. 1, pp. 396-404, 1986.
 [4] S. Lee and J. Park, "Neural computation for collision-free path planning," *Jour. of Intelligent Manufacturing*, vol. 2, pp. 315-326, 1991.
 [5] W. L. Nelson and I. J. Cox, "Local path control for an autonomous vehicle," *Proc. of 1988 IEEE conf. on Robotics and Automation*, pp. 1504-1510, 1988.
 [6] J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, 1991
 [7] N. Hogan, "Impedance control : an approach to manipulation: part I-theory," *ASME Journal of Dynamic Systems, Measurement and Control*,

vol. 107, pp. 1-7, 1985.

[8] T. Arai, H. Ogata and T. Suzuki, "Collision avoidance among multiple robots using virtual impedance," *Proc. IEEE/RSJ Int. Workshop Intel. Robotics and Systems' 89*, pp. 479-485, 1989.
 [9] S. Lee, S. Yi, J. Park and C. Lee, "Reference adaptive impedance control and its application to obstacle avoidance trajectory planning," *Proc. IEEE/RSJ Int. Workshop Intel. Robotics and Systems' 97*, vol. 2, pp. 1158-1162, 1997.
 [10] J. Laumond, P. Jacobs, M. Taix and R. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Robotics and Automation*, vol. 10, no. 5, pp. 577-593, 1994.

부록: (6)의 유도

(4)의 시간에 대한 미분은

$$\frac{dE}{dt} = \sum_{j=0}^I \frac{\partial E}{\partial \mathbf{p}_j} \frac{d\mathbf{p}_j}{dt} + \sum_{j=0}^I \frac{\partial E}{\partial \dot{\mathbf{p}}_j} \frac{d\dot{\mathbf{p}}_j}{dt} \quad (21)$$

이고, 각각

$$\frac{\partial E}{\partial \mathbf{p}_j} = \begin{cases} -k(\mathbf{p}_1 - \mathbf{p}_0) + \frac{\partial P_0(\mathbf{p}_0)}{\partial \mathbf{p}_0} & j=0 \\ -k(\mathbf{p}_{j+1} - \mathbf{p}_j) + k(\mathbf{p}_j - \mathbf{p}_{j-1}) + \frac{\partial P_j(\mathbf{p}_j)}{\partial \mathbf{p}_j} & j=1, \dots, I-1 \\ k(\mathbf{p}_I - \mathbf{p}_{I-1}) + \frac{\partial P_I(\mathbf{p}_I)}{\partial \mathbf{p}_I} & j=I \end{cases} \quad (22)$$

$$\frac{\partial E}{\partial \dot{\mathbf{p}}_j} = m \dot{\mathbf{p}}_j, \quad j=0, \dots, I \quad (23)$$

이다. 이제 $\frac{\partial E}{\partial \mathbf{p}_j}$ 와 $\frac{\partial E}{\partial \dot{\mathbf{p}}_j}$ 를 (21)에 대입하면

$$\begin{aligned} \frac{dE}{dt} &= \left\{ -k(\mathbf{p}_1 - \mathbf{p}_0) + \frac{\partial P_0(\mathbf{p}_0)}{\partial \mathbf{p}_0} \right\}^t \dot{\mathbf{p}}_0 \\ &+ \sum_{j=1}^{I-1} \left\{ -k(\mathbf{p}_{j+1} - \mathbf{p}_j) + k(\mathbf{p}_j - \mathbf{p}_{j-1}) \right. \\ &+ \left. \frac{\partial P_j(\mathbf{p}_j)}{\partial \mathbf{p}_j} \right\}^t \dot{\mathbf{p}}_j \\ &+ \sum_{j=0}^I m \dot{\mathbf{p}}_j^t \ddot{\mathbf{p}}_j \\ &= \left\{ m \dot{\mathbf{p}}_0 - k(\mathbf{p}_1 - \mathbf{p}_0) + \frac{\partial P_0(\mathbf{p}_0)}{\partial \mathbf{p}_0} \right\}^t \dot{\mathbf{p}}_0 \\ &+ \sum_{j=1}^{I-1} \left\{ m \dot{\mathbf{p}}_j - k(\mathbf{p}_{j+1} - \mathbf{p}_j) + k(\mathbf{p}_j - \mathbf{p}_{j-1}) \right. \\ &+ \left. \frac{\partial P_j(\mathbf{p}_j)}{\partial \mathbf{p}_j} \right\}^t \dot{\mathbf{p}}_j \end{aligned} \quad (24)$$

를 얻을 수 있다. 여기서 $\dot{\mathbf{p}}_I = \mathbf{0}$ 이 이용되었다. 마지막으로 (24)에 운동 (2)를 대입하고 $f_j(\mathbf{p}_j) = -\frac{\partial P_j(\mathbf{p}_j)}{\partial \mathbf{p}_j}$ 임을 이용하면,

$$\begin{aligned} \frac{dE}{dt} &= -c(\dot{\mathbf{p}}_0 - \dot{\mathbf{p}}_1)^t \dot{\mathbf{p}}_0 \\ &+ c \sum_{j=1}^{I-1} \left\{ -2 \dot{\mathbf{p}}_j + (\dot{\mathbf{p}}_{j+1} + \dot{\mathbf{p}}_{j-1}) \right\}^t \dot{\mathbf{p}}_j \quad (25) \\ &= -c \sum_{j=0}^{I-1} (\dot{\mathbf{p}}_{j+1} - \dot{\mathbf{p}}_j)^t (\dot{\mathbf{p}}_{j+1} - \dot{\mathbf{p}}_j) \end{aligned}$$

임을 얻을 수 있다.

이 수 영

제어·자동화·시스템공학 학회지 제3권, 제3호, 참조.

홍 예 선

제어·자동화·시스템공학 학회지 제3권, 제3호, 참조.

**이 석 한**

1948년 11월 23일생. 1982년 Purdue Univ. Ph. D. 1983년-1997년 Dept. of CS, Univ. of Southern California, 교수. 1990년-1997년 NASA Senior Technical Staff. 1998년-현재 IEEE A Fellow Member. 1998년-현재 삼성종합기술원 Sector 장(상무).