이동로봇의 전역 경로계획에서 Self-organizing Feature Map의 이용

차 영 엽†(원광대학교 기계공학부) · 강 현 규*(원광대학교 대학원)

The Using of Self-organizing Feature Map for Global Path Planning of Mobile Robot

Young-Youp Cha, Hyon-Gyu Kang

Key Words: Global Path Planning(전역경로계획), Mobile Robot(이동로봇)

Abstract

This paper provides a global path planning method using self-organizing feature map which is a method among a number of neural network. The self-organizing feature map uses a randomized small valued initial weight vectors, selects the neuron whose weight vector best matches input as the winning neuron, and trains the weight vectors such that neurons within the activity bubble are move toward the input vector. On the other hand, the modified method in this research uses a predetermined initial weight vectors, gives the systematic input vector whose position best matches obstacles, and trains the weight vectors such that neurons within the activity bubble are move toward the input vector. According to simulation results one can conclude that the modified neural network is useful tool for the global path planning problem of a mobile robot.

1. 서 론

이동로봇의 전역 경로계획은 형상공간 방법(configuration space method), 포텐셜 방법(potential approach), 그리고 퍼지(fuzzy), 신경회로망(neural network), 유전자(genetic)에 기초한 인공지능 알고리즘이 이동로봇의 경로계획에 적용되었다. 형상공간 방법의 경우, Lozano-Perez[1] 는 V-그래프(visibility graph)에 의한 다각형들로 이루어진 환경에서 이동로봇을 한점으로 간주한 경로문제를 처음으로 다루었다. 그러나 이러한 V-그래프는 이동로봇의 주위환경뿐만 아니라 그 크기에도 영향을 받는다. 이러한 단점을 해결하기 위하여 Noborio[2] 는 환경을 quadtree로 모델링 하는 효과적인 방법을 제안 하였다. 그러나 이러한 quadtree 는 주위환경의 근사적인 표현일 뿐만 아니라 설정된 좌표계에 의존한다. 또한, 이러한 형상 공간 방법은 계산시간을 많이 요구한다.

포텐셜 방법의 경우, Brooks[3]과 Adams[4] 는 반력(repulsive force)을 장애물과 상사(identify) 시키고, 여기에 목표점 쪽으로 인력(attractive force)을 첨가하여 경로생성을 하였다. 따라서 이동로봇은 그들의 합력벡터(resulting vector) 방향으로 운동을 한다. Borenstein[5] 은 반력과 인력의 합력벡터 뿐만 아니라, 환경을 사각형으로 분할하고 각 사각형을 장애물로 판단될 확률(probability)로 결합하였다. 이 확률에 기초하여, artificial 힘을 계산하였다. 그러나 이 artificial 힘의 방향과 크기는 실시간으로 구현하였지만, 실재로 인간이 장애물을 피해서 목표점에 도달하는 경험적 방식과는 큰 차이가 있다.

다른 한편으로 인공지능 알고리즘을 로봇의 전역 경로계획에 사용하려고 하는 노력이 최근에 있었다. 퍼지(fuzzy), 신경회로망(neural network), 유전자(genetic) 알고리즘 또는 그들을 함께 사용하는 것이 그것인데, Qunjie[6]는 퍼지 알고지즘을 사용하여 지역 경로계획

† 원광대학교 기계공학부
E-mail : ggypcha@wonkwang.ac.kr
TEL : (063)850-6693 FAX : (063)850-6691
* 원광대학교 대학원

문제[7]를 다루었고, Zhu[8]는 신경회로망과 함께 cost 함수로 충돌 에너지함수를 이용하였고, Bourbakis[9]는 skeletonization과 신경회로망을 이용하였으며, Chaiyaratana[10]는 유전자 알고리즘을 이용하여 경로계획 문제를 해결하였다. 그러나 이러한 노력에도 불구하고 아직도 지역 또는 전역 경로계획문제는 완전히 해결이 되지 않은 실정이다.

본 연구에서는 이동로봇[11]의 자율주행을 위하여 신경회로망 중에서 self-organizing feature map을 이용한 전역 경로계획 알고리즘을 제안한다. self-organizing feature map은 초기에 가중벡터를 작은 랜덤(random)값으로 초기화하고, 학습계수와 이웃관계 함수를 초기화한다. 그리고 한 개의 입력이 가해진다. 가해진 입력에 가장 가까운 가중벡터를 갖는 승자뉴런(winning neuron)을 찾아서, 승자뉴런 근방의 뉴런을 입력벡터 방향으로 움직이게 함으로서 가중벡터를 재 계산한다. 이와 같은 과정을 반복하여, 뉴런의 위치를 재배치함으로서, 주어진 입력에 따른 특징을 구현할 수 있다.

또한 본 연구에서는 이러한 self-organizing feature map을 전역 경로계획에 적용하기 위하여 수정한다. 즉, 초기의 가중 벡터를 이미 결정된 값으로 초기화하고, 입력을 장애물 외부에 가하여, 가해지는 입력에 가장 가까운 가중벡터를 갖는 승자뉴런을 찾아서, 승자뉴런 근방의 뉴런을 입력벡터 방향으로 움직이게 함으로서 가중벡터를 재 계산한다. 이와 같은 과정을 반복하여 뉴런의 위치를 재배치함으로서 주어진 입력에 따른 이동로봇의 경로를 생성할 수 있다. 제안된 전역 경로계획 알고리즘의 효율성을 입증하기 위하여 장애물이 있는 환경에서 모의실험을 통하여 결과가 보여진다.

2. Self-organizing Feature Map

본 연구는 이동로봇의 자율주행을 위하여 신경회로망 중에서 self-organizing feature map을 이용한 전역 경로계획 알고리즘에 대한 것이다. 그 중에서도 튜보 코호넨(Tuevo Kohonen)에 의해 개발된 Kohonen의 self-organizing feature map을 기본으로 하였다[12].

self-organizing feature map 네트워크에서는 기대된 출력이 없다. 대신에 신경 회로망은 자기 조직화 특성에 의해 임의의 추상적인 관계를 추론할 수 있으며, 더 많은 입력이 인가 될수록 네트워크는 그 학습을 개선하고 변화된 입력들에 적응하여 출력을 내보낸다. 이러한 구조의 한 이점으로는 변화하는 상태와 입력에 대해 대처할 수 있다는 것이다. 그러므로 이 네트워크는 입력을 다른 카테고리(category)로 분류할 때와 음성 인식, 로봇 모터 제어 등에 사용된다.

self-organizing feature map의 목적은 N-차원의 입력 공간을 의미 있는 지형학적인 순서로 1차원 또는 2차원의 출력 공간(출력 뉴런)에 맵핑(mapping)할 수 있게 하는 것이다. 이러한 목적을 성취하기 위해 경쟁학습(competitive learning)에 의한 승자독점(winner-take-all)원리와 측면 제어(lateral inhibition)가 이용된다[12].

입력 벡터 X 와 j 번째 출력층 뉴런과 상응하는 가중치 벡터 W_j 를

$$X = [x_1, x_2, \ldots, x_p]^T \tag{1}$$

$$W_j = [\omega_{j1}, \omega_{j2}, \ldots, \omega_{jp}]^T, \quad j = 1, 2, \ldots, N \tag{2}$$

로 표시한다면, 입력 벡터 X 와 가중치 벡터 W_j 를 이용한 학습 법칙은 다음과 같다.

$$W_j^{new} = W_j^{old} + \eta (X_i - W_j^{old}) \tag{3}$$

여기서 η 는 학습율(learning rate)을 나타내고, i 는 1에서 p 까지로 입력 뉴런(neuron)의 수를 나타내며, j 는 1에서 N 까지로 출력 뉴런의 수이다. 그리고 w_{ji} 는 i 번째 입력 뉴런과 j 번째 출력 뉴런을 연결하는 가중치를 나타낸다.

입력 벡터의 분류를 위한 출력 층의 승자 뉴런의 결정은 입력값 X 와 가장 비슷한 가중치 W_j 를 갖는 출력 뉴런을 선택하는 것과 같다. 이러한 출력 뉴런을 선택하는 방법은 두 가지가 있다. 첫째는

$$I_{j\ max} = \sum_{i=1}^{p} \omega_{ji}\, x_i \tag{4}$$

과 같은 I_j 를 선택하는 것이고, 두 번째는 입력 벡터와 최소의 Euclidean norm을 갖는 가중치를 선택하는 것이다. 즉, $i(X)$ 가 승자 뉴런이라 한다면 이 식은 다음과 같다.

$$i(X) = k, \quad where\ \|W_k - X\| < \|W_j - X\| \tag{5}$$

위와 같이 출력 뉴런을 선택하여 승자가 된 뉴런만이 “1”이 되고 나머지는 “0”이 되는데 이러한 방법이 경쟁학습에 의한 승자독점 원리이다. 보통 가중치 벡터와 입력 벡터는 정규화(normalization) - 가중치 벡터와 입력 벡터의 최대 크기가 “1”로 설정됨 - 시키는데, 그 이유는 학습규칙이 입력 벡터로부터 가중치 벡터를 뺀 값을 사용하기 때문이다.

그리고, 좀더 효율적인 패턴 분류를 위하여 측면 제어를 이용한다. 측면 제어의 대표적인 방법은 이웃관계 함수(neighborhood function), $\Lambda_{i(X)}(n)$ 의 사용으로, 승자 뉴런이 선택되면 승자 뉴런의 이웃하는 거리에 따라 연결강도를 달리하는 방법으로 연결 강도는 거리에 반비례한다. 이웃관계 함수를 이용한 학습 법칙은 다음과 같다.

$$W_j(n+1) = \begin{cases} W_j(n) + \eta(n)\,[X - W_j(n)], & j \in \Lambda_{i(X)}(n) \\ W_j(n), & otherwise \end{cases} \tag{6}$$

여기서 $\eta(n)$ 은 n 시간에서의 학습율이다.

3. 전역 경로계획

3.1 수정된 Self-organizing Feature Map

본 연구에서는 이러한 self-organizing feature map을 전역 경로계획에 적용하기 위하여 수정한다. 즉, 초기의 가중 벡터를 이미 결정된 값으로 초기화하고, 입력을 장애물 외부에 규칙적으로 가하여, 가해지는 입력에 가장 가까운 가중벡터를 갖는 승자뉴런을 찾아서, 승자뉴런 근방의 뉴런을 입력벡터 방향으로 움직이게 함으로서 가중벡터를 재 계산한다. 이와 같은 과정을 반복하여 뉴런의 위치를 재 배치 함으로서 주어진 입력에 따른 이동로봇의 경로를 생성할 수 있다.

이와 같이 수정된 Kohonen의 self-organizing feature map은 다음과 같은 단계로 표현된다.

Step 1. 초기화

초기 가중치 벡터, $W_j(0)$ 을, 기존의 self-organizing feature map에서는 아주 작은 랜덤(random)값으로 초기화 시키는데 반하여, 이동로봇의 작업영역에 격자모양으로 배치시킨다. 그리고 학습계수, $\eta(0)$ 와 이웃관계 함수 $\Lambda_{i(X)}(0)$ 를 초기화한다. 두 값 모두 초기에는 큰 값을 부여한다.

Step 2. 각 장애물들에서 입력 벡터, X 의 경우에 다음의 Step 2a, 2b, 2c를 수행한다.

Step 2a. network의 입력층에 입력벡터, X 를 위치시킨다.

Step 2b. Similarity matching

입력벡터, X 에 가장 근접한 가중치 벡터를 갖는 뉴런을 선택하여 승자뉴런으로 한다. 이는 식 (5)를 사용하여 구할 수 있다.

Step 2c. Training

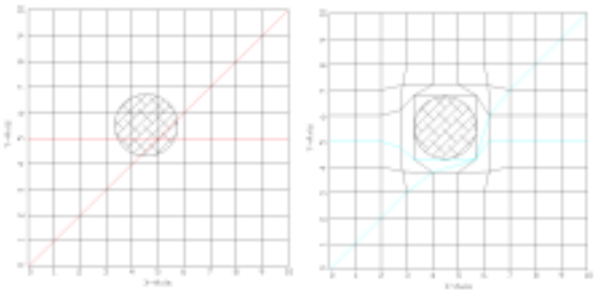
식 (6)과 같이 activity bubble 내의 뉴런들을 입력벡터 방향으로 가중치 벡터를 훈련(training)시킨다.

Step 3. 학습율, $\eta(n)$ 의 갱신

학습율의 선형감축은 만족할만한 결과를 얻도록 해야 한다.

Step 4. 이웃관계 함수, $\Lambda_{i(X)}(n)$ 의 감축

Step 5. 정지 조건의 확인



(a) (b)

Fig. 1 Example of global path planning by using self-organizing feature map (a) before and (b) after training

feature map에 식별할 수 있는 변화가 일어나지 않는 경우에 반복계산을 정지하고, 그렇지 않으면 Step 2로 간다.

3.2 전역 경로계획 알고리즘

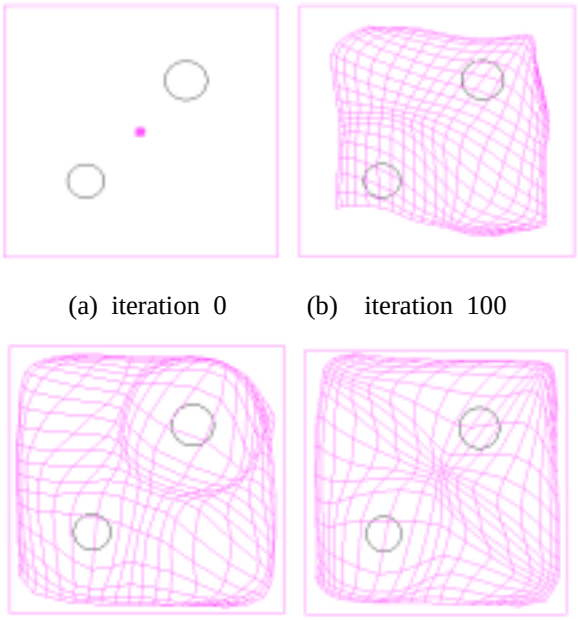
전역 경로계획은 이미 주어진 장애물 지도를 기본으로 출발점에서 목표점까지 장애물과의 충돌을 피하면서 가장 빠르게 갈 수 있는 경로를 찾는 것이다. 즉 이동로봇의 작업 영역에서의 장애물에 대한 정보를 미리 가지고 있으므로 이를 이용하여 이동로봇의 최단경로를 찾는 것이다.

본 연구에서는 이와 같은 전역 경로계획에 앞에서 거론한 수정된 self-organizing feature map을 사용한다. 이를 위하여 초기에 Fig. 1(a)와 같이 이동로봇의 작업영역에 위치가 결정된 초기 mesh를 설정한다. 여기서 이동로봇의 작업영역은 사각형이라고 가정하고, 원은 장애물을 나타내고 있다. 작업영역의 수평축을 X축, 그리고 수직축을 Y축이라 하면, mesh상의 한 점은 (X, Y)로 나타낼 수 있다.

만약 원으로 나타낸 장애물이 없다고 가정하면, 작업영역 위의 한점 (0, 5)에서 다른 한점 (10, 5)로 갈 수 있는 최단 경로는 mesh 위의 (0, 5), (1, 5), (2, 5), (3, 5), (4, 5), (5, 5), (6, 5), (7, 5), (8, 5), (9, 5) 그리고 (10, 5)를 지나는 선이 될 것이다. 같은 방법으로, 작업영역 위의 한점 (0, 0)에서 다른 한점 (10, 10)으로 갈 수 있는 최단 경로는 mesh 위의 (0, 0), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8), (9, 9) 그리고 (10,10)을 지나는 선이 될 것이다.

그리고 앞의 3.1절에서 기술한 수정된 self-organizing feature map을 이용하여 network을 training 하면, 장애물에서부터 일정한 거리만큼 떨어진 위치에 규칙적으로 입력이 주어지고, activity bubble 내의 뉴런은 가중치 벡터가 입력벡터 방향으로 주어지므로, 그 내부에 있는 mesh 들은 점차 장애물 바깥쪽으로 빠져나오게 된다. 그 결과가 Fig. 1(b)와 같다고 하면, 실제 이동로봇을 위한 전역 경로는 training 하기전의 좌표 순서를 따라서, training 후의 좌표순서대로 지정하면 된다. 그리고 training 전에 해당 경로위에 장애물이 없었다면, training 후에도 그 경로는 거의 변화가 없음을 알 수 있다. 즉, 작업영역 위의 한점 (0, 1)에서 다른 한점 (10, 1)로 갈 수 있는 최단 경로 점의 위치는 training 전이나 후에도 거의 변화가 없음을 알 수 있다.

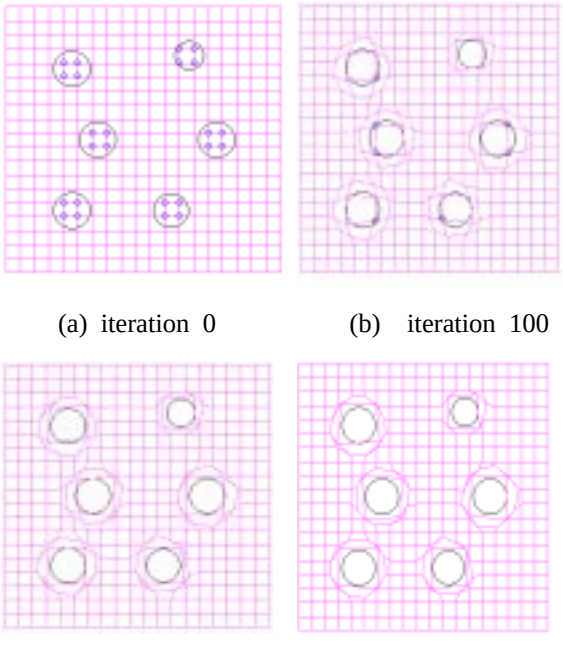
또한 작업영역 위의 한점 (0, 5)에서 다른 한점 (10, 5)로 갈 수 있는 경로를 보면, 출발과 도착부근에서는 mesh의 위치변화가 거의 없지만, 장애물이 있는 중간부의 위치변화는 상당한 것으로 보여 진다. 즉 출발부의 (0, 5), (1, 5), (2, 5)와 도착부의 (7, 5), (8, 5), (9, 5), (10, 5)의 위치는 training 전과 후에 거의 변화가 없지만, 장



(a) iteration 0 (b) iteration 100

(c) iteration 200 (d) iteration 400

Fig. 2. Sequential results of global path planning by using original self-organizing feature map.



(a) iteration 0 (b) iteration 100

(c) iteration 200 (d) iteration 400

Fig. 3. Sequential results of global path planning by using modified self- organizing feature map.

애물이 있는 중간부, 즉 (3, 5), (4, 5), (5, 5), (6, 5)는 mesh의 위치변화가 명백히 나타난다. 이렇게 함으로서 이미 주어진 장애물 지도를 기본으로 출발점에서 목표점까지 장애물과 충돌을 피하면서 가장 빠르게 갈 수 있는 최단경로를 찾는 것이 가능하다.

4. 실험결과

Fig. 2는 앞의 2장에서 설명한 Kohonen의 self-organizing feature map을 전역 경로계획에 적용한 순차적인 결과를 보여주고 있다. 가로와 세로의 뉴런 개수가 각각 20개인 2차원 네트워크를 구성하였다. 이동로봇의 작업영역을 사각형으로 가정하고 두개의 원형 장애물을 배치하였다. 초기에 가중치 벡터를 작은 랜덤값으로 주어짐으로서, 중앙에 몰려있는 것이 Fig. 2(a)에 나와 있다. Fig. 2(b)는 중앙에 몰려 있는 뉴런을 전체 작업영역으로 펼치기 위하여 입력을 전체 작업영역으로 랜덤하게 준 경우의 결과를 보여주고 있다. 여기서 보면 net가 전체 작업영역에 균일하게 펼쳐지지 못하고 있다. 이는 작업영역의 경계에 주어지는 입력벡터는 net를 넓게 펼치려고 하지만, 내부에 주어지는 입력벡터는 더 안쪽의 net는 펼쳐주지만 바깥쪽의 net는 입력방향으로 잡아당기기 때문이다. 그런 후에 장애물인 원의 내부에 있는 net를 원 바깥으로 잡아당기기 위하여 원의 외부에 집중적으로 입력벡터를 준 결과가 Fig. 2(c)에 나와 있다. 이 결과에서 보면 입력벡터의 영향으로 장애물 주위에 있는 net의 모양이 장애물을 확대한 형상으로 나타날 뿐만 아니라, 작업영역 가장자리의 net 생성이 불완전 한 것을 알 수 있다. 마지막으로 장애물인 원 주위의 net를 펼치기 위하여 작업영역 경계에 입력을 집중적으로 가한 결과가 Fig. 2(d)에서 보여 진다. 결과적으로 이와 같은 방법은 net가 작업영역 전체에 골고루 펼쳐지지 못하고, 장애물의 크기에 따라서 입력을 주는 방법을 다르게 해야 하며, training 중간 중간에 입력을 주는 방법을 바꾸어야하는 단점이 있다. 뿐만 아니라 이러한 복잡한 입력 생성과정을 거치더라도 net가 완전히 장애물 바깥으로 나오는 것을 보장하지 못한다.

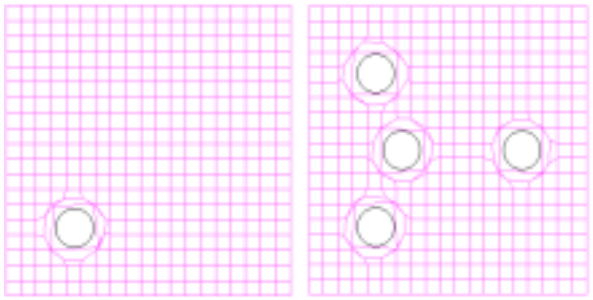
Fig. 3은 앞의 3장과 같이 본 연구에서 제안한 수정된 self-organizing feature map을 전역 경로계획에 적용한 순차적인 결과를 보여주고 있다. 가로와 세로의 뉴런 개수는 Fig. 2와 같이 각각 20개로 하고, 이동로봇의 작업영역에 있는 장애물은 6개를 배치하였다. 기존의 self-organizing feature map과 비교하여, 초기에 가중치 벡터를 작은 랜덤(random) 값으로 주는 대신에 전체 작업영역에 일정한 간격으로 배치한 것이 Fig. 3(a)에 나와 있다. 여기서 장애물 안쪽의 작은 점은 net의 교차점인 node 중에서 장애물 안쪽에 있는 것을 나타내고 있다. Fig. 3(b)는 각 물체의 바깥에 입력벡터를 원주방향으로 일정한 간격만큼 가하고, 승자뉴런 주위의 가중치벡터를 입력벡터 방향으로 끌어당기는 결과를 보여주고 있다. 이때 작업영역의 맨 가장자리에 있는 뉴런은 입력벡터의 영향을 무시하도록 하여, 작업영역의 경계를 나타내는 사각형 바깥으로 net가 나가는 것과 안쪽으로 이동하는 것을 제한한다.

점차적으로 입력벡터를 가하는 횟수를 증가시키면, 장애물인 원 내부에 있는 net가 원 바깥으로 밀려나가는 결과가 얻어지는데 200회와 400회 반복의 결과가 각각

Fig. 3(c)와 3(d)에서 보여 진다. 결과적으로 이와 같은 방법은 net가 전체의 작업영역에 골고루 펼쳐지고, 장애물의 크기에 관계없이 입력을 주는 방법이 일정하며, net의 교차점이 완전히 장애물 바깥으로 나오는 것을 보장해 줄 수 있다. 이 결과는 Fig. 2와 같은 Kohonen의 self-organizing feature map을 전역 경로계획에 적용한 결과와 비교하면 훨씬 더 효율적임을 알 수 있다.

Fig. 4는 장애물이 1개와 4개인 경우에 수정된 self-organizing feature map을 적용한 후의 전역경로계획 결과를 보여주고 있다. Fig. 3의 6개 장애물들에서 1개 또는 4개의 장애물에 대한 결과이며, net이 장애물들을 잘 피해서 생성된 것을 보여준다.

Fig. 5는 Fig. 4와 Fig. 3의 장애물이 1개, 4개, 그리고 6개인 경우에 수정된 self-organizing feature map의 적용에 따른 iteration 회수와 전체 node의 이동거리 총합을 보여주고 있다. 장애물이 많을수록 이동거리의 총합은 커지고, 장애물이 1개, 4개, 그리고 6개인 경우 모두 25 iteration 까지는 이동거리의 총합이 증가하다가, 그 이후



(a) (b)

Fig. 4. Results of global path planning by using modified self-organizing feature map in environment with (a) 1 obstacle and (b) 4 obstacles.

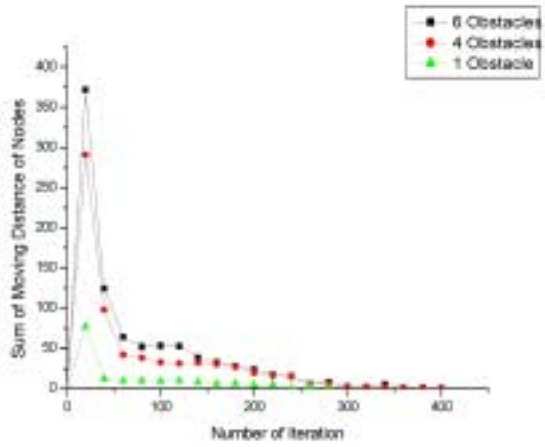


Fig. 5. Sum of moving distance according to the number of iteration in environment with 1, 4, and 6 obstacles by using modified self-organizing feature map.

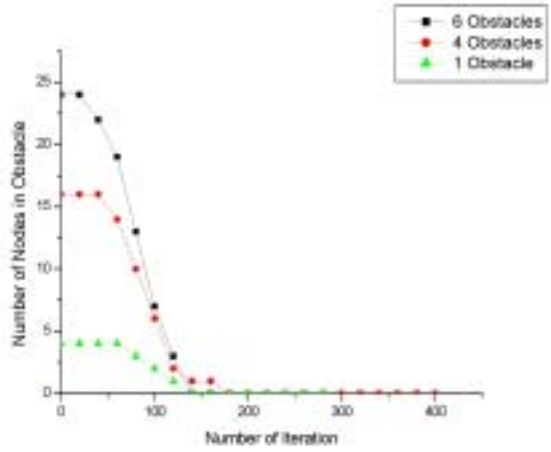


Fig. 6. The number of remained nodes in obstacles according to the number of iteration in environment with 1, 4, and 6 obstacles by using modified self-organizing feature map.

에는 감소하는 것을 알 수 있다. 300 iteration 이후에는 이동거리의 총합이 세 경우 모두 0으로 saturation되는 것을 볼 수 있다.

Fig. 6은 Fig. 5와 같이 장애물이 1개, 4개, 그리고 6개인 경우에 수정된 self-organizing feature map의 적용에 따른 iteration 회수와 장애물 내에 있는 node의 총 개수를 보여주고 있다. 초기에는 장애물의 개수가 많을수록 장애물 내에 있는 node의 총 개수는 큰 값을 갖고, 150 iteration 이후에는 세 경우 모두 0으로 saturation되는 것을 알 수 있다.

Fig. 7은 Fig. 4(a)와 같이 장애물이 1개인 경우에 수정된 self-organizing feature map을 적용한 후, 원점에서 각 node들까지의 총 이동거리를 3차원으로 보여주고 있다. 이 그림에서 음영의 변화는 전체 이동거리를 등고선 형태로 보여주고 있다. 장애물에 영향을 받지 않는 좌, 우측의 경우는 self-organizing feature map을 적용하기 전과 같이 아무런 변화가 없다. 그러나 장애물 바로 앞의 node까지 거리는 mesh의 변형에 따라서 최초의 상태보다 총 이동거리가 줄어들고, 장애물 뒤에서는 늘어나는 것을 알 수 있다.

Fig. 8은 장애물이 1개인 경우에 수정된 self-organizing feature map을 적용한 후와 적용 전의 원점에서 각 node들까지의 총 이동거리 차를 3차원으로 보여주고 있다. 앞의 Fig. 7에서 거론한 것처럼 장애물 앞의 node까지 총 이동거리 차는 음수를 갖고, 장애물 뒤는 양수를 갖는 것을 알 수 있다. 또한 장애물 좌, 우의 node에서는 그 차가 0이 되는 것을 알 수 있다. 결과적으로 본 연구에서 제안한 수정된 self-organizing feature map은 이동로봇의 전역 경로계획에 효과적으로 사용될 수 있음을 알 수 있다.

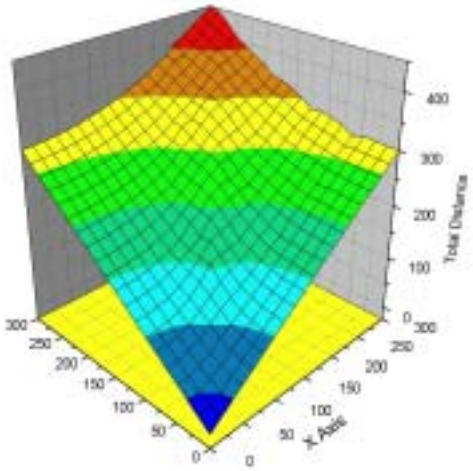


Fig. 7. Total distance from origin point to each node by using modified self-organizing feature map in environment with 1 obstacle.

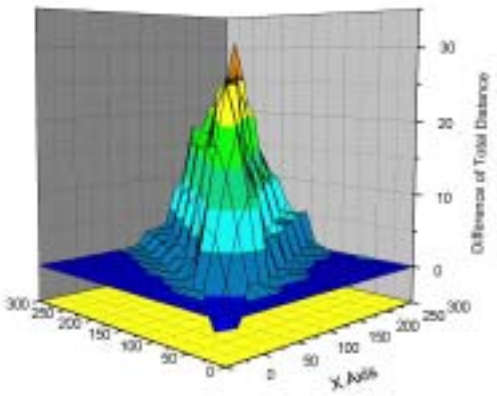


Fig. 8. Difference of total distance from origin point between applied to modified self-organizing feature map and before.

5. 결 론

본 연구에서는 신경회로망의 하나인 Kohonen의 self-organizing feature map을 전역 경로계획에 적용하기 위하여 수정하였다. 즉, 초기의 가중 벡터를 이미 결정된 값으로 초기화하고, 입력을 장애물 외부에 가하여, 가해지는 입력에 가장 가까운 가중벡터를 갖는 승자뉴런을 찾아서, 승자뉴런 근방의 뉴런을 입력벡터 방향으로 움직이게 함으로서 가중치벡터를 재 계산하였다.

이와 같은 과정을 반복하여 뉴런의 위치를 재 배치함으로서 주어진 입력에 따른 이동로봇의 경로를 생성할 수 있었다. 제안된 전역 경로계획 알고리즘의 효율성을 입증하기 위하여 장애물이 있는 환경에서 모의실험을 통하여 수정전의 self-organizing feature map을 이용한 결과와 비교하였고, 우수한 성능을 보여주었다.

참고문헌

(1) Lozano-Perez, T. and Wesley, M. A., 1979, "An algorithm for planning collision-free paths among polyhedrial obstacles", Commun. ACM, pp.560-570.

(2) Noborio, H., Naniwa, T. and Arimoto, S., 1988, "A fast path planning algorithm by synchronizing modification and search of its path-graph", Proc. IEEE Intern. Workshop on Artificial intelligent for Industrial Application, pp.351-357.

(3) Brooks, R., 1983, "Solving the find path problems by good representation of free space", IEEE Trans. Syst. Man Cybern., Vol.SMC-13, No.3, pp.190-197.

(4) Adams, M. D. and Probert, P. J., 1990, "Towards a real-time navigation strategy for a mobile robot", Proc. of the IEEE Intern Workshop on Intelligent Robots and Systems, pp.743-748.

(5) Borenstein, J. and Koren, Y., 1991, "The vector field histogram-fast obstacle avoidance for mobile robots", IEEE Trans. on Robotics and Automation, No.3, pp.278-298.

(6) Qunjie, D., Mingjun, Z., 2001, “Local Path Planning Method for AUV Based on Fuzzy-neural Network”, SHIP ENGINEERING, Vol. 1, pp.54-58.

(7) Cha, Y. Y., 1997, "Navigation of a free ranging mobile robot using heuristic local path planning algorithm", Robotics and Computer Integrated Manufacturing, Vol. 13, No. 2, pp.145-156.

(8) Zhu, Y., Chang, J., Wang, S., 2002, "A new path-planning algorithm for mobile robot based on neural network", TENCOM '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, v.3, pp.1570-1573.

(9) Bourbakis, N. G., Goldman, D., Fematt, R., Vlachavas, I., Tsoukalas, L. H., 1997, "Path Planning in a 2-D Known Space Using Neural Networks and Skeletonization", Conference proceedings : IEEE International Conference on Systems, Man, and Cybernetics, Vol.3, pp.2001-2005.

(10) Chaiyaratana, N., Zalzala, A. M. S., 2002, "Time-Optimal Path Planning and Control using Neural Networks and a Genetic Algorithm", International Journal of Computational Intelligence and Applications, Vol.2 No.2, pp.153-172.

(11) Cha Y. Y. and Gweon, D. G., 1998, "The development of a free ranging mobile robot equipped with a structured light range sensor", Intelligent Automation and Soft Computing, Vol. 4, No. 4, pp. 289-312.

(12) Kohonen, T., 1990, "The self-organizing map", Proc. of the IEEE, Vol. 78, No. 9, pp.1464-1480.