

# 유전자 알고리즘을 이용한 축구 로봇의 속도 제어 함수 파라미터 최적 해 탐색

김지윤, 김현영, 심귀보

중앙대학교 전자전기공학부 인공생명 및 지능정보시스템연구소

## 1. 서 론

일반적으로 로봇 축구 시스템은 크게 CCD 카메라로부터 들어오는 영상 신호를 라벨링하고 잡음을 제거하여 각 물체의 좌표를 입력받는 비전처리 부분과 비전으로부터 입력 받은 좌표를 이용하여 원하는 목표를 수행하기 위해 신속하고 정확한 동작을 만들어내는 속도제어 부분, 그리고 시시각각 동적으로 변하는 상황을 판단하여 어떤 동작을 취할지를 판단하는 전략 부분으로 나누어 질 수 있다[2]. 본 논문에서는 로봇의 가장 기본적인 움직임을 만들어 내는 속도 제어 부분에 관하여 논의한다. 제어 부분은 크게 목표를 달성하기 위해 거리오차와 각도 오차에 따라 각각의 바퀴에 독립적인 속도를 전송하여 원하는 움직임을 만들어 내는 속도 제어 부분과 몇 가지 속도 제어 함수를 조합하여 구성되어진 행동 제어 부분으로 구성되어 있으며 이것은 실제 축구 경기에 있어서 Dribble, Kick, Tackle등과 같은 개인기에 해당한다. 이때 가장 하위 레벨의 움직임이 얼마나 신속하게 이동하면서 정확한 자세를 유지하느냐 하는 것은 상위 레벨의 모든 움직임에 영향을 주며 게임에 있어서 승패를 좌우하는 열쇠가 된다. 따라서 본 논문에서는 거리 오차와 각도 오차의 반영 비율을 파라미터를 이용하여 조절하도록 구성된 속도 제어 함수와 속도 제어 함수의 적절한 파라미터 조합을 유전 알고리즘을 이용하여 국소 해에 해당하는 파라미터 해를 피해 빠른 시간 내에 가장 적합한 조합을 탐색하는 방식을 제안하고자 한다.

## 2. 속도 제어 전략

### 2.1 속도 제어 알고리즘

축구 로봇을 원하는 지점까지 이동시키는 알고리즘은 비전으로부터 입력되어진 각 물체의 좌표와 각도들을 바탕으로 목표지점까지의 거리 오차와 각도 오차를 줄여가도록 각 바퀴의 DC모터 속도를 변화시켜주는 것이다. 공에 근접했을 때의 속도를 로봇의 중심 속도라 하면 이러한 움직임을 바탕으로 공 근처에서의 중심 속도를 크게 해 주면 Kick동작이 이루어지며 다음 동작으로 적절한 가속도를 가지고 이동하게 되면 Dribble 또한

가능하게 되므로 Move 함수를 잘 설계하는 것은 전체적인 게임에 있어서 공격력의 향상에 결정적인 영향을 미치게 된다. 잘 알려진 가장 기본적인 Move 함수는 다음과 같이 구성된다.

$$V_R = V_C + K_a \theta_e \quad (1)$$

$$V_L = V_C - K_a \theta_e \quad (2)$$

여기서  $V_C$ 는 거리 오차에 따른 로봇의 이동속도를  $\theta_e$ 는 각도 오차를  $K_a$ 는 각도 오차에 따라 양 바퀴의 속도 차이를 크게 만들어 주는 상수 파라미터로써 각도 오차에 대한 민감도를 결정한다. 그러나 이러한 방식은 각 상황에 따라  $K_a$ 값을 실험을 통하여 정확히 설정하여야 하며 항상 같은 비율로 각도 오차를 수정하므로 유연한 움직임을 만들어 내는 데 제한이 있게 된다. 특히 거리에 따라 각도 오차에 대한 반응 비율이 달라져야 가감속을 통한 효율적인 목표점 수렴이 가능하므로 실험으로 거리에 따라 다른  $K_a$ 값을 구하여 사용하여야 정교한 움직임을 제어하기에 적합하게 된다. 이 외에도 벡터장 방법, 비 선형 함수의 Limit cycle을 이용한 방법과 Fuzzy 제어를 이용하는 방법, 강화 학습을 이용한 방법 등이 있는데[2] 대부분이 다양한 움직임을 만족시키기 위하여 그만큼 많은 상수들을 오랜 시간동안 실험을 통하여 최적화 시켜야 하므로 변화된 다른 경기장 환경에서 필요한 대체 해를 찾기가 매우 어렵다. 이런 점들에 착안하여 '공 사냥꾼'팀에서는 속도 함수에 거리 오차를 이동 속도에 반영하는 정도를 제어하는 D 파라미터, 각도 오차를 회전 속도에 반영하는 정도를 제어하는 K 파라미터 외에 거리 오차와 각도 오차를 동시에 고려하여 이동 중에도 오차 각을 쉽게 수정할 수 있도록 속도를 감속 시켜주는 정도를 제어하는 A 파라미터를 사용하여 세 가지 파라미터 값만으로 다양한 속도와 경로 패턴을 만들 수 있도록 하였으며 적절한 D, A, K 세 가지 파라미터의 조합을 유전자 알고리즘을 이용하여 탐색함으로써 국소 해를 피하면서 변화된 환경에서도 빠른 대체 파라미터 조합을 찾을 수 있도록 하였다.

## 2.2 속도 제어 함수의 설계

'공 사냥꾼'팀에서 사용한 속도 제어 함수는 크게 두 가지로 나누어진다. 공격과 방어를 위해 유리한 위치로 신속하게 이동하면서도 정확한 Kick과 Dribble을 위한 유연한 감속 제어를 목표로 설계되어진 DecelMove 함수, 공을 놓치지 않으면서도 어느 정도의 방향을 수정할 수 있는 가속 제어를 위해 고안되어진 AccelMove 함수, 그리고 원하는 반경의 원을 그리면서 회전하도록 설계되어진 Circle 함수가 있다. 즉 감속 직선 주행, 가속 직선 주행 그리고 곡선 주行的 세 가지 이동 함수들로 구성되어 있는데 본 논문에서는 유전자 알고리즘을 실세계 모델 속에서 실험하는데 사용되어진 DecelMove 함수를 논의한다.

DecelMove 함수는 기본적으로 위치 이동을 위한 함수이다. 이 함수의 알고리즘은 거리 오차에 따라 로봇의 이동 속도를 얼마나 크게 해 줄 것인가를 결정하는 D 파라미터와 각도 오차에 따라 양 바퀴에 속도 차를 줌으로써 얼마나 빨리 각도를 수정하게 할 것인가를 결정하는 K 파라미터 그리고 이동 속도 항에 곱해져서 각도 오차에 따라 이동 속도를 줄여주어 이동 중에도 유연하게 각도를 수정하기 위한 A 파라미터로 구성되어 있다. 이 파라미터들을 어떤 값으로 설정하는가에 따라 로봇은 다양한 경로 패턴을 만들 수 있다. 먼저 거리 오차에 따라 속도를 조절하는 항을 살펴보면

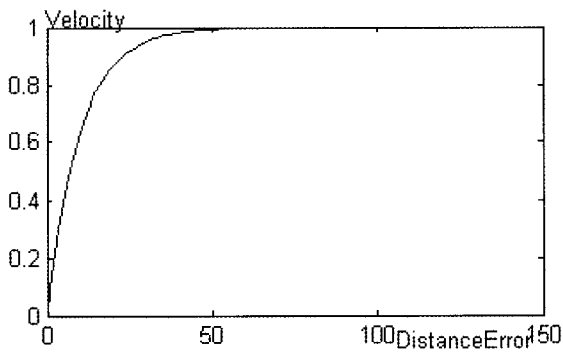


그림 1. 거리오차에 따른 속도 분포 곡선

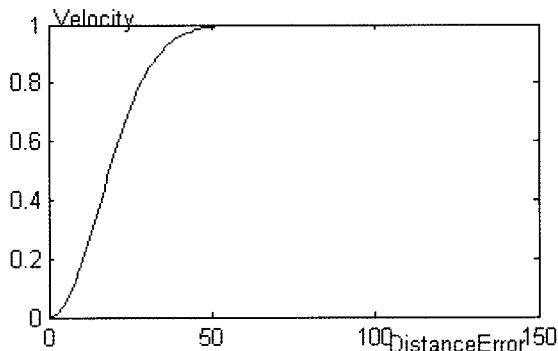


그림 2. Gaussian 분포를 이용한 속도 곡선

$$V = 1.0 - \exp\left(-\frac{D_e}{D}\right), \text{ (normalized)} \quad (3)$$

식 3의  $D_e$ 는 거리 오차를 나타내며 D는 속도의 감속 곡률을 결정하는 상수로써 이 값이 클수록 속도 곡선은 거리 오차에 따라 더욱 완만한 감속을 하게 된다. V 값은 0에서 1사이의 값으로 normalize되어있으므로 실제 적용 시에는 이동 중 요구되는 최고 속도를 곱하여 scaling하면 된다. 그림 1에서 알 수 있듯이 지수함수를 이용함으로써 일정 거리오차까지는 최고 속도를 유지하다가 부드러운 감속을 할 수 있다는 장점이 있으나 이 함수의 단점은 속도 곡선과 시간에 대한 미분의 관계를 가지는 가속도 곡선과 jerk 곡선을 생각해보면 거리 오차가 0일 때 불연속점을 만듦으로써 Dribble과 같은 정교한 제어를 필요로 하는 동작으로 이어질 때 충격을 주게 되고 이것으로 인하여 공을 빠뜨리는 경우를 발생시킨다. 이런 문제를 해결하기 위하여 그림 2와 같이 Gaussian 분포 함수를 이용하였다.

$$V = 1.0 - \exp\left(-\frac{D_e^2}{D}\right), \text{ (normalized)} \quad (4)$$

다음으로 이동 중에 각도 오차에 따라 속도를 조절하는 부분을 살펴보면 각도 오차가 작을수록 가능한 최고 속도를 내며 각도 오차가 크면 속도를 낮추어주어 각도 오차를 쉽게 줄일 수 있도록 하여야 한다. 이를 위해서 속도 함수에 또한 그림 3.과 같은 분포를 가지는 함수를 곱하여 주어야 한다.

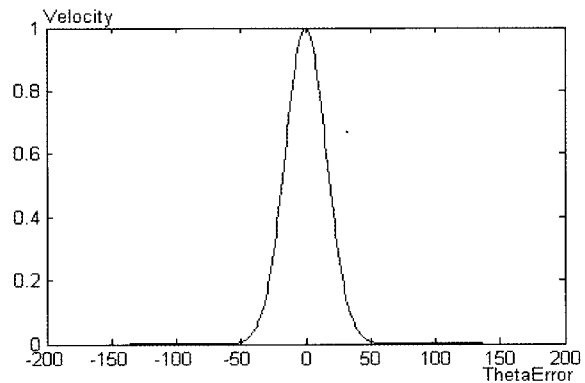


그림 3. 각도 오차에 따른 속도 분포 곡선

$$\exp\left(-\frac{\theta_e^2}{A}\right), \text{ (normalized)} \quad (5)$$

그림 3에서 알 수 있듯이 이 식은 각도 오차가 0일 때 최대 값 1이 되고 각도 오차가 커질수록 0의 값을 가지게 되어 각도에 따라 중심 속도를 감속 시켜주는 역할을 하게 된다. 따라서 식 4와 식 5를 곱하면 원하는 이동 속도 분포가 나오는데 그림 2와 그림 3 두 가지의 속도 분포를 만족시킬 수 있는 전체적인 속도 분

포와 '공 사냥꾼'팀에서 사용한 속도 제어 함수는 다음과 같다.

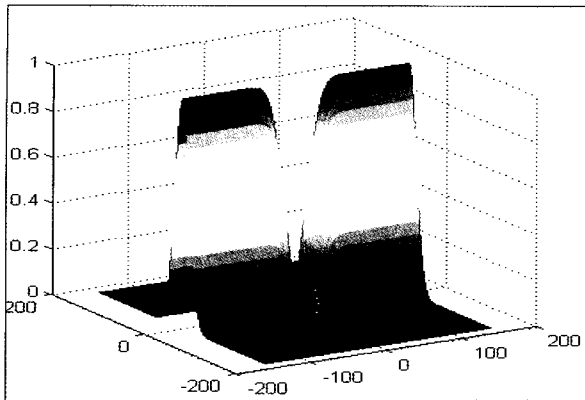


그림 4. 거리와 각도 오차에 따른 속도 분포

$$V = (1.0 - \exp(-\frac{D_e^2}{D}))(\exp(-\frac{\theta_e^2}{A})),$$

(normalized) (6)

그림 4에 실제 경기장에서 생길 수 있는 거리 오차 범위 (+150 ~ -150)와 각도 오차 범위 (+180 ~ -180) 내에서 전체적인 제어 속도 분포를 나타내었다. 이렇게 구성된 DecelMove 함수에 원하는 동작을 실험하면서 각 상황에 맞게 D와 A라는 파라미터와 두 바퀴의 속도 차를 만들어서 방향을 수정하게 하는 항의 K 파라미터를 적절하게 조절해주면 다양한 경로 패턴을 제어 할 수 있다.

■ 완성된 속도 제어 함수

$$V_L = 125(1.0 - \exp(-\frac{D_e^2}{D}))(\exp(-\frac{\theta_e^2}{A})) + K \theta_e$$

(7)

$$V_R = 125(1.0 - \exp(-\frac{D_e^2}{D}))(\exp(-\frac{\theta_e^2}{A})) - K \theta_e$$

(8)

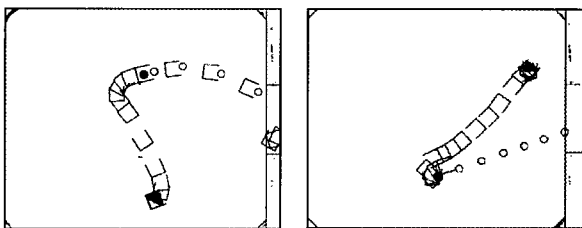


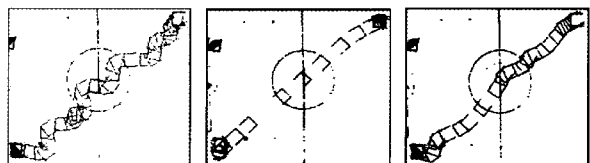
그림 5. 실제 경기장에서 속도 함수의 적용

이렇게 설계되어진 속도 함수를 이용하여 실제 로봇

의 여러 가지 움직임을 구현 해 보았다. 그림 5는 이 속도 함수를 적용하여 Dribble Kick과 Circle Kick을 하는 모습이다. 이전의 속도 제어함수에 비해 개선되어진 점은 다양한 경로 패턴이 요구되어질 때 각각의 패턴에 대해 한 쌍의 파라미터 값만이 요구되어진다는 것이다. 다시 말하면 위에서 언급한 이전의 속도 함수는 곡선 경로가 요구되어질 때 오랜 실험을 통해 거리 오차에 따라 다른 K값을 일일이 구하여야 원하는 곡선 경로가 나올 수 있다. 또 다른 경로에 대해서도 오랜 실험 기간을 필요로 한다. 본 논문에서 제안한 속도 함수는 세 값으로 구성된 파라미터 쌍 하나로 한 개의 경로를 제어하는 것이 가능하므로 실험을 통하여 요구되는 경로를 만들어 내는 것이 훨씬 수월하다. 그러나 각도 오차와 거리 오차를 줄여 가는 속도 제어 방식을 이용하므로 여러 가지 비선형적인 환경적 요소와 역학적인 요소들로 인해 실험을 통하여 이 값들을 조절해 갈 수밖에 없으며 변화된 새로운 환경에서는 다른 파라미터 쌍을 찾아내어야 한다. 실제 경기에서도 이런 문제들을 접하게 되지만 적절한 움직임에 수렴하는 파라미터 조합 해를 찾는 시간이 오래 걸리고 특히 주변의 해들을 고려하지 못할 경우 Localminima를 구별해내기가 어려우므로 유전 알고리즘을 적용하여 변화된 환경 속에서 최적화된 파라미터 쌍을 가능한 짧은 시간에 탐색하는 알고리즘을 도입하였다.

3. 유전자 알고리즘을 적용한 최적해 탐색

가장 하위레벨에 속하는 이동 함수는 임의의 지점까지 가장 짧은 시간 내에 도달하면서도 다음 명령을 위해 안정된 자세를 유지하여야 하므로 위에서 유도되어진 속도 함수를 이용하여 로봇을 지정된 두 좌표로 이동을 시키면서 목표지점에 도달한 후 다음 행동을 하기 위해 정해진 각도에 맞추어 정지하기까지의 시간을 제어 적합도의 기준으로 이용하였다. 처음에는 원하는 경로와 이동경로 사이의 거리 편차까지 고려하였으나 위의 두 가지 조건만으로도 정확한 직선 경로와 안정된 파라미터 쌍을 찾을 수 있었다. 우선 각 파라미터 값에 따른 몇 가지 로봇의 이동 패턴을 보면 그림과 같다.



a. 수행시간: 8143(ms) b. 수행시간: 4251(ms) c. 수행시간: 5954(ms)

그림 6. 파라미터 값에 따른 로봇의 경로 패턴

그림 6을 경로 패턴과 파라미터 값 사이의 관계를 고려하여 설명하면 a. 그림은 D 파라미터 값보다 K 파라미터 값이 상대적으로 너무 커서 전체적인 경로가 진동하는 경우이고 b. 그림은 D값이 너무 커서 목표점에 바로 수렴하지 못하고 지나치는 경우이며 c. 그림은 A값이 너무 커서 진행 중에 조금만 오차 각이 커져도 감속을 하는 경향을 보이며 D값에 비해 K값이 상대적으로 너무 작아서 진동하는 것을 볼 수 있다. 즉 D, A, K 파라미터는 로봇의 경로를 결정하는데 있어서 측정된 오차 값이 각 바퀴 속도에 얼마나 영향을 줄지를 결정해주는 파라미터임을 알 수 있다. 이처럼 원하는 경로의 패턴을 주어진 환경 속에서 가장 잘 만들어 내는 파라미터 쌍을 탐색하기 위해 다음과 같이 유전자 알고리즘을 설계하였다[1].

■ 유전자 알고리즘으로 적절한 파라미터 쌍을 탐색하기 위해서는 우선 각각의 파라미터 값들을 2진 비트로 Encoding해야 하는데 이때 최적해에 성공적으로 수렴하기 위해서는 파라미터의 영역과 구간 간격을 잘 설정해야한다. 본 논문에서는 이전의 실험에서 얻은 경험치를 바탕으로 각 파라미터의 정의역을 다음과 같이 정했다. D : [0 1500] 50, A : [35 480], K : [0.05 0.36] 각 파라미터 별로 분해능(resolution)을 32구간으로 동일하게 나누었으며 각 파라미터당 5bit씩 총 15bit의 염색체(chromosome)를 하나의 개체(individual)로 설정하였다. 예를 들어 어떤 개체가 010110011010010의 염색체를 가진다면 표현형(phenotype)으로 Decoding 할 경우  $D = 50 \times (01011) = 50 \times 11 = 550$ ,  $A = 35 + 15 \times (00110) = 35 + 15 \times 6 = 125$ ,  $K = 0.05 + 0.01 \times (10010) = 0.05 + 0.01 \times 18 = 0.23$ 이다.

■ 초기 개체군(population)은 random하게 생성된 일개의 개체들로 구성하였으며(두 번째 실험부터는 좀더 빠른 시간에 수렴하기 위하여 20개의 개체로 실험하였다.) 적합도(fitness)는 한 점에서 출발하여 목표점과의 거리 오차 3cm 이내, 각도 오차 0도의 자세로 안정되게 정지한 상태까지 걸리는 시간을 기준시간(15초)에서 백수의 역수를 0에서 1까지의 값으로 스케일링하였다. 또 실제로 로봇을 움직이다 보면 같은 파라미터 쌍을 가지고도 다른 경로 패턴을 보이는 파라미터 쌍이 존재하는데 이와 같은 불안정한 해를 피하기 위하여 종료 조건으로 각 세대의 개체군 중 최고의 적합도를 가지는 개체가 0.79이상이고 2세대 이상 이 상태를 유지하면서 두 세대의 적합도 편차가 0.02이내인 개체가 나타나면 이를 적합한 개체로 판정하여 종료하도록 구성하였다. 그리고 선택 방법에 있어서는 주어진 환경 속에서 최적해를 찾는데 실패했다 하더라도 원하는 움직임을 어느 정도 만족시킬 수 있는 대체 해를 보존하기 위해 엘리트 보존을 적용하였으며 아무리 안정된 해라하더라도 역학적으로 복잡한 실세계 구조 속에서 구현된 움직임으로 fitness를 평가받기 때문에 적절한 해가 불안정한 결과를 보일 수도 있으므로(비전의 프레임 손실, DC 모터의 비선형 특성, 이동 방향에 따른 관성의 변화, 바닥의 굴곡 등) 선택 방법에 있어서 엘리트 보존 선택을 적용하되 엘리트 개체 수를 2개체로 설정하였다.

■ 교배와 돌연 변이율은 각각 교배율 0.6, 돌연 변이율 0.033으로 설정하였으며 교차 방법은 적절한 파라미터 값들임에도 불구하고 파라미터의 쌍이 적절하지 못할 경우가 많으므로 각 파라미터 세그먼트를 기준으로 하여 2점 교차를 하였다.

이렇게 설계되어진 유전자 알고리즘의 타당성을 검증하기 위하여 같은 로봇으로 정식 경기장에서 두 번 반복하여 실험을 한 후 수렴 결과를 비교하여 유전자 알고리즘을 이용한 탐색법이 안정되게 수렴함을 보이고 굴곡이 심하고 다른 재질의 도료를 이용하여 바닥의 마찰력이 다른 자체 제작한 경기장에서 다른 로봇을 가지고 반복 실험하여 변화된 환경에서도 적절한 대체 파라미터 쌍이 구해짐을 검증하였다.

## 4. 실험 결과

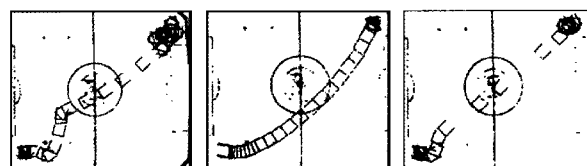
### 4.1 정식 경기장에서의 실험

#### 4.1.1 첫 번째 실험 결과

3장에 의거하여 임의로 발생시킨 10개의 개체(표 1)에 대해서 로봇의 경로 패턴을 그림 7에 나타내었다.

표 1. 랜덤 발생한 개체군의 파라미터 값과 적합도

개체 번호	염색체 (유전형)	표형형			수행시간 (ms)	적합도
		D	A	K		
0	000011011010000	50	380	0.20	10011	0.332600
1	110111101000011	1350	440	0.07	5286	0.647600
2	010001101110110	400	455	0.26	5802	0.613200
3	100111101011110	950	440	0.34	6309	0.579400
4	111011000110111	1450	305	0.27	5274	0.648400
5	011101010100111	700	365	0.11	4591	0.693933
6	010111101111000	550	455	0.28	4788	0.680800
7	110101011111001	1300	395	0.29	4756	0.682933
8	100100000100001	900	65	0.05	9540	0.364000
9	010001000000101	400	290	0.09	4423	0.705133



0번 개체                      1번 개체                      2번 개체

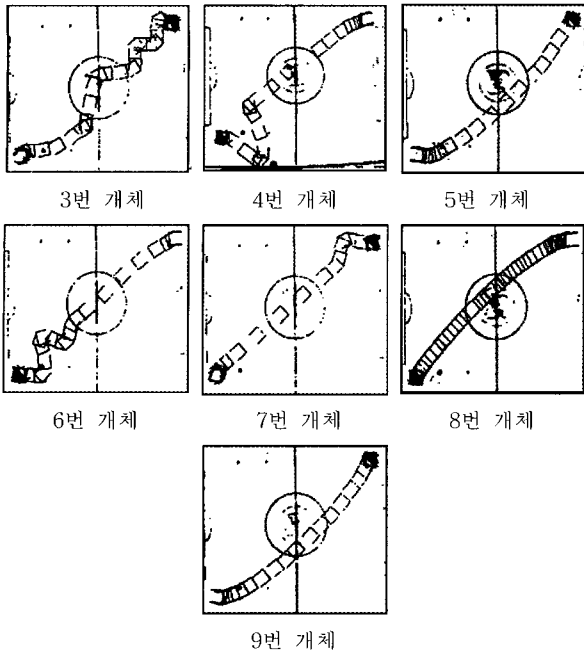


그림 8. 실험이 끝난 세대 개체들의 경로 패턴

진화의 과정을 보기 위해서 각 세대수에 따라 개체들의 평균 적합도와 최대 적합도를 정리하면 표 3과 같고, 이를 그래프로 나타내면 그림 9와 같다.

표 2. 각 세대별 개체군의 평균 적합도와 최대 적합도 값

세대수	세대수 0-9		세대수 10-19	
	평균적합도	최대적합도	평균적합도	최대적합도
0	0.594800	0.705133	10	0.613007
1	0.656333	0.784133	11	0.653913
2	0.583767	0.749867	12	0.713847
3	0.636067	0.785133	13	0.732587
4	0.554167	0.727533	14	<b>0.717113</b>
5	0.581687	0.795333	15	0.711960
6	0.606440	0.760800	16	0.710513
7	0.591880	0.717467	17	0.700140
8	0.659660	0.761067	18	0.694287
9	0.597473	0.762867	19	0.702567

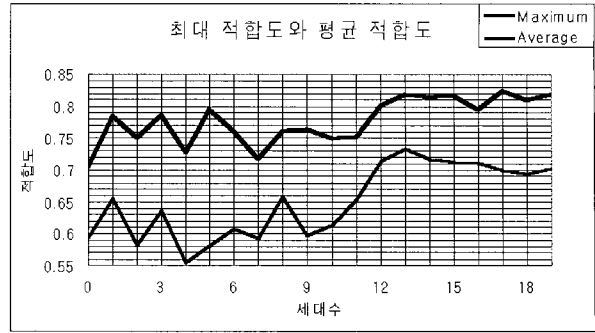
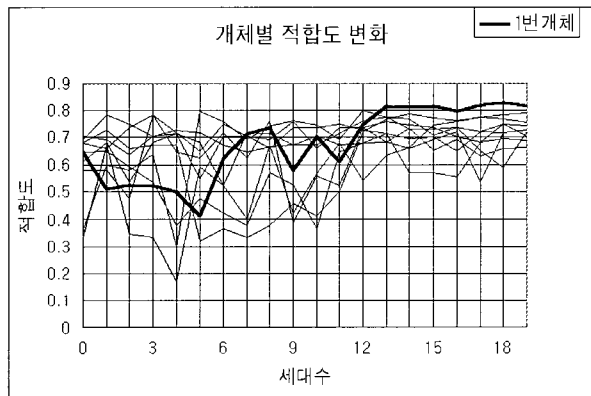


그림 9. 각 세대별 개체군의 적합도 변화

그림 7과 그림 8의 이동 경로 패턴을 보면 그림 7의 초기 세대보다 세대가 지난 후 안정된 경로를 보이는 개체가 많아졌음을 알 수 있다. 또 그림 9의 최대 적합도 변화를 보면 12세대까지는 비교적 적합도는 높으나 상당히 불안정한 파라미터 조합을 가진 개체들이 엘리트로 선택되다가 곧 도태되어 버리나 종료 조건을 만족시키는 해에 수렴하는 세대로 갈수록 최대 적합도 변화가 안정되어 가는 것을 볼 때 안정된 최적해를 찾기 위하여 2세대에 걸쳐 적합도가 유지되는 것을 종료 조건에 포함시킨 것이 적절한 결과를 도출함을 알 수 있다. 마지막으로 종료 조건을 만족시킨 최적해에 해당하는 파라미터를 살펴보면 13세대에서 종료 조건을 만족시키는 개체가 발생되어 14 세대에 종료하게 되며 탐색되어진 적합한 파라미터 쌍은 표 2에 의해  $D : 650, A : 395, K : 0.20$ 이 된다. 뿐만 아니라 이후에도 실험을 계속하여 추이를 지켜본 결과 이때 나타난 엘리트 개체가 7세대 동안 최고의 적합도를 유지하고 표 2와 표 3을 비교해 보면 불안정한 실세계 모델에서의 실험임에도 불구하고 상당히 안정되어 있으면서도 높은 적합도를 가진 파라미터 해가 탐색되었음을 알 수 있다.

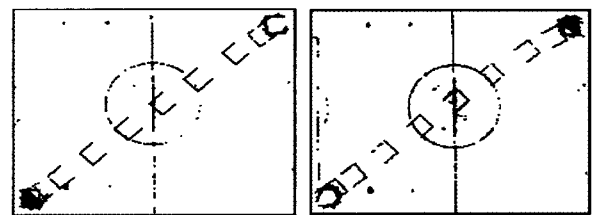


그림 10. 안정된 직선 경로를 보이는 엘리트 개체들

그림 10의 왼쪽 그림은 진화 종료 후의 최고 엘리트 개체에 의해 수렴되어진 최적해에 해당하는 파라미터 쌍을 이용하여 로봇의 이동 경로를, 오른쪽은 그 세대에 두 번째 엘리트 개체의 이동 경로를 나타낸 것이다. 이것으로써 적합도를 시간만으로 평가하더라도 신속하면서도 안정된 최적의 주행 경로를 나타내는 파라미터 쌍을 찾아내는 것이 가능함을 확인 할 수 있다.

### 4.1.2 두 번째 실험 결과

앞의 실험에서 찾은 해가 타당한 해임을 검증하기 위하여 반복하여 실험하였다. 이때 신뢰성을 높이고 좀더 빠르게 수렴하기 위하여 개체 수를 20개체로 하였으며 K값의 resolution을  $K : [0.05 \ 0.298]$ 로 세분화하고 나머지 경우는 모두 동일한 방법으로 같은 경기장과 로봇을 이용하여 실험한 결과이다.

표 3. 랜덤하게 발생시킨 초기 세대 개체군의 파라미터 값과 적합도

개체 번호	염색체 (유전형)	표현형			수행시간 (ms)	적합도
		D	A	K		
0	011011011000110	650	365	0.088	6292	0.580533
1	000111100001010	150	245	0.120	5789	0.614067
2	110000010100011	1200	110	0.048	7144	0.523733
3	011111001100001	1550	320	0.048	6461	0.569267
4	010110111000010	550	425	0.056	5123	0.658467
5	000110001101111	150	80	0.160	3757	0.749533
6	011011101000011	650	290	0.048	4787	0.680867
7	111101000101101	1500	320	0.144	4289	0.714067
8	000011001110011	50	335	0.192	10383	0.307800
9	100110111000110	950	245	0.088	4242	0.717200
10	00000001110101	400	50	0.208	5475	0.635000
11	111111011100111	1550	380	0.192	4758	0.682800
12	000110000111110	150	50	0.280	6469	0.568733
13	110000001010011	1200	65	0.192	4068	0.728800
14	111010110000011	1450	215	0.064	4424	0.705067
15	011011110011010	650	455	0.248	5752	0.616533
16	011100111111000	600	260	0.232	5423	0.638467
17	110010110000111	1250	215	0.064	5620	0.625333
18	010001111110110	400	500	0.216	5292	0.647200
19	110101010011010	1300	395	0.248	5774	0.615067

표 4. 진화 종료(15세대) 후의 개체군의 파라미터 값과 적합도

개체 번호	염색체 (유전형)	표현형			수행시간 (ms)	적합도
		D	A	K		
0	011011100100110	650	35	0.088	7144	0.523733
1	100111011001010	950	365	0.120	2887	0.807533
2	011010000011110	650	410	0.248	7296	0.513600
3	111010110001000	1450	215	0.104	5057	0.662867
4	111111100000110	1550	365	0.088	4418	0.705467
5	000111100101011	150	410	0.128	4385	0.707667
6	011111011000100	750	215	0.072	4769	0.682067
7	111100110100101	1500	230	0.080	5271	0.648600
8	001000111010110	200	365	0.216	6806	0.546267
9	110101011000100	900	500	0.072	5573	0.628467
10	001000010000100	200	95	0.072	5621	0.625267
11	101111010101101	1150	350	0.272	4937	0.670867
12	010111010110010	550	350	0.216	6476	0.568267
13	111100101010000	1300	185	0.168	4088	0.727467
14	111010100001011	1450	155	0.128	4257	0.716200
15	110110100110010	1350	170	0.184	4423	0.705133
16	100110110001000	950	215	0.104	3572	0.761867
17	100010010001111	850	95	0.160	3405	0.773000
18	010001010110001	400	335	0.168	5439	0.637400
19	011011100101010	<b>650</b>	<b>410</b>	<b>0.120</b>	<b>2882</b>	<b>0.807867</b>

진화의 과정을 보기 위해서 각 세대에 다른 개체들의 평균 적합도와 최대 적합도를 정리하면 표 6과 같고, 이를 그래프로 나타내면 그림 11과 같다.

표 5. 각 세대별 개체군의 평균 적합도와 최대 적합도 값

세대 수	평균적합도		최대적합도	
	0	1	9	17
0	0.628927	0.749533	0.641227	0.762933
1	0.605217	0.772000	0.656140	0.793200
2	0.622677	0.773000	0.677407	0.750667
3	0.627197	0.751800	0.652483	0.784600
4	0.645927	0.749667	0.688110	0.806667
5	0.649723	0.770800	0.671943	0.807533
6	0.613267	0.786533	<b>0.670980</b>	<b>0.807867</b>
7	0.659983	0.807467	0.695043	0.807600
8	0.656350	0.794667	0.701823	0.806667

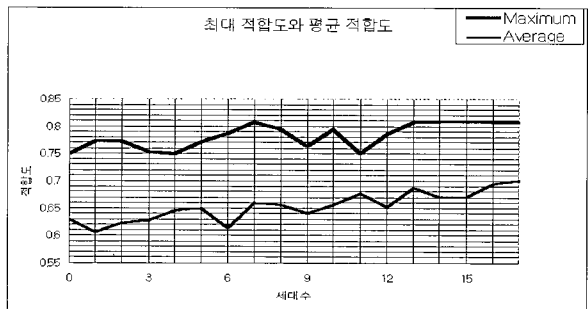
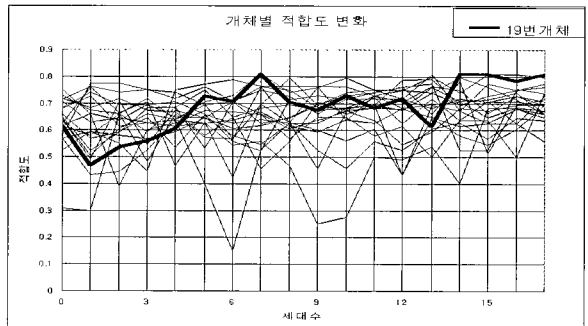


그림 11. 각 세대별 개체군의 적합도 변화

앞의 그래프에서도 알 수 있듯이 14세대에 종료 조건을 만족시키는 개체가 발생하였으며 15세대에 종료하게 된다. 첫 번째 실험과 두 번째 실험에서 종료조건을 만족시킨 개체들의 표현형을 살펴보면 첫 번째 실험에서  $D : 650, A : 395, K : 0.20$ , 수행시간 : 2721(ms), 적합도 : 0.814133이고 두 번째 실험에서  $D : 650, A : 410, K : 0.12$ , 수행시간 : 2887(ms), 적합도 : 0.807533에 수렴하였다. 이 두 실험의 파라미터 값과 수행 시간 그리고 적합도와 주행 경로를 비교해 볼 때 적합한 해가 수렴성이 있음을 확인 할 수 있으며 수렴 시 파라미터 값이 비교적 근사한 값을 나타내므로 실제 경기에서 적용 가능한 최적의 이동 패턴을 만들어내는 최적해라고 판단할 수 있다.

### 4.2 자체 제작한 경기장에서의 실험

또 변화된 경기장에서도 대체 해를 탐색해 낼 수 있음을 검증하기 위하여 다른 도료를 이용하여 바닥의 마찰력이 다르고 약간의 굴곡이 있는 자체 제작한 경기장에서 다른 로봇으로 두 번 반복하여 실험하였다. 실험 결과 중 랜덤하게 생성되는 초기 개체의 데이터는 지면 관계상 생략하였고, 파라미터 해를 보여주는 종료 세대의 데이터만 표시하였다.

#### 4.2.1 첫 번째 실험의 결과

표 6. 진화 종료(15세대) 후의 개체군의 파라미터 값과 적합도

개체 번호	염색체 (유전형)	표전형			수행시간 (ms)	적합도
		D	A	K		
0	110000111100100	1200	320	0.072	3758	0.749467
1	111001001101000	1400	185	0.104	4088	0.727467
2	011101010111111	700	350	0.288	5256	0.649600
3	100011001100110	850	320	0.088	6670	0.555333
4	010011001110000	450	320	0.168	3751	0.749933
5	001101000100110	300	290	0.088	5590	0.627333
6	101001011000111	<b>1000</b>	<b>365</b>	<b>0.096</b>	<b>3406</b>	<b>0.772933</b>
7	110111100110110	1350	410	0.200	4104	0.726400
8	101111011111011	1150	380	0.256	4424	0.705067
9	110101011011001	1300	125	0.240	4586	0.694267
10	000100001100111	100	80	0.096	8128	0.458133
11	011100110101111	700	230	0.152	5093	0.660467
12	100010001011000	850	65	0.232	4587	0.694200
13	010010011001000	450	125	0.104	7824	0.478400
14	000010111100111	50	260	0.096	7311	0.512600
15	011011110010101	650	455	0.208	5123	0.658467
16	100110011001111	950	125	0.160	3933	0.737800
17	010000101111010	400	200	0.120	5625	0.625000
18	100111011101001	950	380	0.112	6125	0.591667
19	100100011011111	900	260	0.288	4392	0.707200

위의 실험을 통하여 얻어진 모든 세대에 걸친 Data들을 정리하여 도표로 나타내었다.

표 7. 각 세대별 개체군의 평균 적합도와 최대 적합도 값

세대 수	평균적합도		최대적합도	
	0	15	0	15
0	0.63823	0.741	0.67076005	0.750667
1	0.6640267	0.764467	0.61928335	0.761
2	0.64326995	0.770867	0.68029005	0.7852
3	0.6361567	0.706333	0.6884233	0.772067
4	0.62828005	0.738533	0.6771434	0.761867
5	0.62318	0.7596	0.66999005	0.783067
6	0.61684665	0.740467	<b>0.6540867</b>	<b>0.772933</b>
7	0.64967005	0.772133	0.67536665	0.773133
8	0.66057665	0.761867		

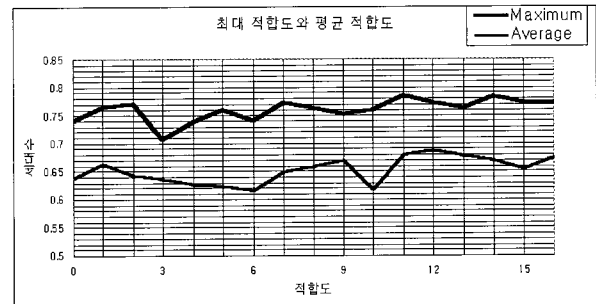
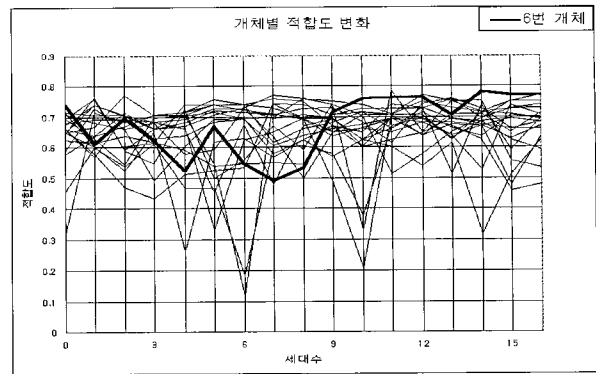


그림 12. 각 세대별 개체군의 적합도 변화

#### 4.2.2 두 번째 실험의 결과

표 8. 진화 종료(15세대) 후의 개체군의 파라미터 값과 적합도

개체 번호	염색체 (유전형)	표전형			수행시간 (ms)	적합도
		D	A	K		
0	101001110100110	1000	290	0.088	4076	0.728267
1	000101000100001	100	425	0.112	5440	0.637333
2	001101000111111	100	290	0.288	4425	0.705000
3	110101101000101	1200	95	0.08	6123	0.591800
4	101110010011111	1150	500	0.288	4040	0.730667
5	010010100101100	450	170	0.136	4087	0.727533
6	011011111111100	650	290	0.264	4256	0.716267
7	110000100010100	1200	170	0.20	4087	0.727533
8	001111100111100	350	455	0.208	4086	0.727600
9	001101000111101	300	290	0.272	4922	0.671867
10	011000111000000	250	245	0.04	7488	0.500800
11	010110111010100	550	245	0.20	5657	0.622867
12	100101100000111	<b>900</b>	<b>395</b>	<b>0.096</b>	<b>3069</b>	<b>0.795400</b>
13	100011001001001	850	305	0.112	9666	0.355600
14	000011000001101	50	275	0.144	10042	0.330533
15	111000001111011	1400	200	0.256	4575	0.695000
16	100101001110100	900	320	0.20	3890	0.740667
17	100101101000100	900	425	0.072	4604	0.693067
18	000010001000110	50	305	0.088	5608	0.626133
19	010011000100001	450	230	0.048	6624	0.558400

위의 실험을 통하여 얻어진 모든 세대에 걸친 Data들을 정리하여 도표로 나타내었다.

표 9. 각 세대별 개체군의 평균 적합도와 최대 적합도 값

세대수	0	1	2	3	4	5
평균적합도	0.68738345	0.66130995	0.67301005	0.64410670	0.66174000	0.6441670
최대적합도	0.77206700	0.79640000	0.78400000	0.77306700	0.79613300	0.79540000

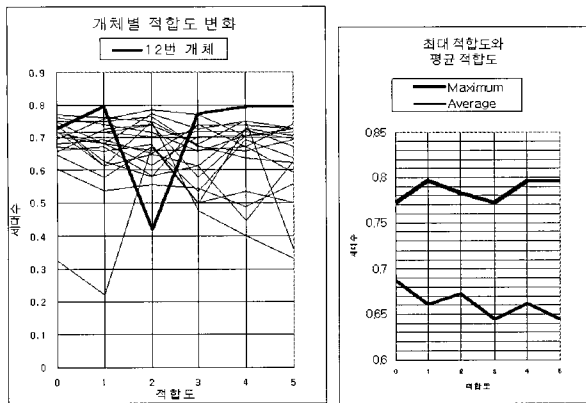


그림 13. 각 세대별 개체군의 적합도 변화

앞의 실험 결과들로부터 알 수 있는 것은 각 세대별 엘리트 개체들이 바닥의 마찰력이 커진 이유로 인해 앞의 실험보다는 조금 안정되어 있음을 알 수 있고 이 때 두 실험에서 구해진 각각의 파라미터 값들은 첫 번째 실험에서  $D : 1000$ ,  $A : 365$ ,  $K : 0.096$ , 수행 시간 :  $3406ms$ , 적합도 :  $0.772933$ 으로 수렴되었고 두 번째 실험에서는 각각  $D : 900$ ,  $A : 395$ ,  $K : 0.096$ , 수행 시간 :  $3067ms$ , 적합도 :  $0.7954$ 으로 수렴되었다. 이 두 파라미터 쌍을 비교해 보면 적합도가 조금 낮아지긴 하였으나 정식 경기장에서의 실험과 마찬가지로 최적해에 수렴함을 알 수 있으며 이로써 변화된 환경에서도 적절한 대체 해를 탐색할 수 있음을 알 수 있다.

### 5. 결 론

본 논문에서 제안 한 방식을 실제 축구 로봇으로 수 차례 실험 해 본 결과 이전 지역 대회에 참가 할 때 필자가 직접 파라미터 값들을 구했던 시간들에 비해 훨씬 짧은 시간에 파라미터 값들을 구할 수 있었으며 그 당시 구했던 파라미터들의 대부분이 Localminima였음을 알 수 있었다. 특히 정식 경기장에서의 첫 번째 실험에서 찾은 파라미터 값은 지역 예선 대회에서 적용했던 파라미터 값들보다 훨씬 안정되어있으면서도 더욱 빠른 시간에 목표점에 수렴함을 보여주었고 모든 실험이 20세대 이전에 최적화된 해를 탐색해냄으로써 실제 경기에서 게임 시작 전에 주어지는 세팅 시간 내에 대체 해를 찾아 낼 수 있음을 입증하였고 실제 경기에서의 적용 가능성을 보여주었다.

### 참고문헌

1. 심귀보, 인공지능의 방법론, 드림 미디어, 2000. 10.
2. 김종환, 로봇 축구 시스템, 대영사, 2000. 1.
3. 임영도 이상부, 퍼지·신경망·유전 진화, 인솔 미디어, 1999. 1.
4. 기타노 히로아키 편 조성배 역, 유전자 알고리즘, 대청 정보 시스템(주), 1996. 10.
5. 박성수 박혜영, C++로 구현한 유전자 알고리즘, 한울 출판사, 2001. 1.