

## 관절 공간에서의 GP 기반 진화기법을 이용한 4족 보행로봇의 걸음새 자동생성

### Automatic Gait Generation for Quadruped Robot Using a GP Based Evolutionary Method in Joint Space

서기성\*, 현수환

(Kisung Seo and Soohwan Hyun)

**Abstract** : This paper introduces a new approach to develop a fast gait for quadruped robot using GP(genetic programming). Planning gaits for legged robots is a challenging task that requires optimizing parameters in a highly irregular and multidimensional space. Several recent approaches have focused on using GA(genetic algorithm) to generate gait automatically and shown significant improvement over previous results. Most of current GA based approaches used pre-selected parameters, but it is difficult to select the appropriate parameters for the optimization of gait. To overcome these problems, we proposed an efficient approach which optimizes joint angle trajectories using genetic programming. Our GP based method has obtained much better results than GA based approaches for experiments of Sony AIBO ERS-7 in Webots environment.

**Keywords** : automatic gait generation, quadruped robot, joint space trajectory, genetic programming

#### I. 서론

보행로봇은 바퀴 로봇보다 장애지형이나 평평하지 않은 지형도 이동할 수 있어 높은 이동성을 가지는 장점이 있으며, 4족 보행 로봇의 경우, 현재 SONY 사의 Aibo 로봇처럼 점차 각광을 받고 있는 엔터테인먼트 분야에서도 널리 사용되고 있다[1-4].

4족 보행로봇의 걸음새(gait) 생성 문제는 로봇의 보행 계획을 결정하는 핵심적인 부분으로, 발의 궤적, 스텝 수, 초기자세 등의 수많은 파라미터를 동시에 설계해야 하는 복잡한 문제이다. 걸음새는 각 다리의 주기적인 움직임을 얻기 위한 실수 파라미터 벡터의 집합으로 정의된다. 관련된 걸음새 파라미터들 중에서 필요한 파라미터들이 사전에 또는 동적으로 선택되어야 하고, 선택된 걸음새 파라미터의 실수 값이 결정되어야 한다[1,2].

지금까지 상당수의 걸음새 연구는 사람에 의해서 반복적인 시뮬레이션을 통해 수작업으로 구해지는 방식 또는 수학적 탐색방법이었으며[3], 최근 들어 GA(Genetic Algorithm)를 이용한 자동적인 생성방법들이 시도되고 있다[1,2,4-7]. 이러한 접근법은 걸음새를 자동적으로 생성해줄 수 있고 GA의 성능에 따라 최적에 가까운 값을 탐색할 수 있는 장점이 존재한다.

현재의 GA등을 이용한 자동 걸음새 생성 기법들은 필요한 주요 파라미터들을 사전에 정해놓고, 이들 값을 탐색하는 파라미터 최적화 문제로만 접근하고 있는 실정이다. 그러나 걸음새에 관련된 수많은 파라미터를 다 포함시킬 수 없고, 사전에 선택된 파라미터만이 걸음새 생성을 위해

서 충분한 조건인지 판단하기 어려우며, 사용된 최소한의 파라미터도 10개 이상을 사용하고 있어 이들의 최적 값을 구하는 것은 매우 어려운 문제이다.

참고적으로 최근 대부분의 걸음새 연구에서 사용되고 있는 파라미터는 직교좌표계에서의 발끝의 자취(locus)와 초기자세, 그리고 한 사이클 당 보간수 등으로 구성되어 있다 [1-3,6,7].

발끝의 자취의 경우 형태에 따라 직사각형, 일반 사각형, 사다리꼴, 원호 등을 사용하고 있으며, 각각에 대해 2차원, 또는 3차원의 구성이 가능하다.

초기자세는 앞쪽 높이, 뒤쪽 높이, 앞발의 측면 편차, 앞발의 정면 편차, 뒷발의 측면 편차, 뒷발의 정면 편차, 머리의 수직각도 및 수평각도 등 접근법에 따라 서로간의 선택되는 파라미터 종류와 정의가 약간씩 차이가 나며, 선택되는 수도 많게는 20개 이상에서 적게는 10개 정도로 차이가 있다. 사이클 당 보간 수는 반복되는 발의 움직임 궤적에서 한 주기에 대한 제어 명령 실행의 샘플링 수를 의미한다.

현재의 GA 접근법은 위에서 언급한 3가지 종류의 모든 파라미터들을 유전자(chromosome)로 표현해야 하기 때문에 최적화해야 될 파라미터 수도 많을 뿐 아니라, 3가지 종류의 파라미터의 특성이 서로 다르므로 연산 시간 및 최적화 도달에 큰 어려움을 가지고 있다.

그러므로 이러한 직교좌표계 기반의 GA 접근법의 문제점 및 최적화로의 어려움을 원천적으로 극복하기 위하여, 본 연구에서는 GP(Genetic Programming)[9,10]를 이용한 관절(joint) 좌표계 중심의 진화 접근법을 제안한다. 직교좌표계에서 사용되는 수많은 파라미터 대신에 각 관절에 대한 궤적만을 구함으로써 걸음새를 생성할 수 있는 방법이다. GP는 해의 탐색공간이 구조적으로 제한되지 않는(open-ended) 설계 특성을 만족시킬 수 있다.

\* 책임저자(Corresponding Author)

논문접수 : 2007. 9. 27., 채택확정 : 2008. 3. 16.

서기성, 현수환 : 서경대학교 전자공학과

(ksseo@skuniv.ac.kr/xjavalov@shyun.com)

GP 기반의 4족 보행로봇의 걸음새 자동 생성 기법의 구현을 위하여, 관절공간에서의 걸음새(gait) 생성용 GP 함수 및 터미널 구현하고, 보행속도와 안정성을 최적화하는 걸음새의 자동 생성 방법을 구현한다. 제안된 기법의 검증을 위하여 SONY Aibo 4족 보행 로봇에 대해서 ODE 기반의 물리적 특성을 포함한 정교한 시뮬레이션이 가능한 Webots [10]을 이용하여 실험을 수행하였다.

II. 4족 보행로봇 걸음새 제어

1. 4족 보행 로봇 모델

걸음새 생성을 위한 4족 보행로봇 모델은 SONY Aibo ERS-7[11]을 사용하였다(그림 1). ERS-7은 1세대인 ERS-110과 2세대인 ERS-210을 거친 3세대 모델로 로봇축구 및 엔터테인먼트용 로봇으로 잘 알려져 있다[1]. 그림 2에는 머리 및 각 관절의 움직임 범위가 나와 있다. 그림 2에 표시된 수치들의 단위는 degree이다.

2. 4족 보행 걸음새 생성

기존의 주요 걸음새 자동 생성방법은 다음과 같다. UNSW 팀은 Aibo ERS-7의 걸음새 생성을 위해서 Powell's minimization 기법을 사용하였다[3]. 표 1과 같이 17-8개의

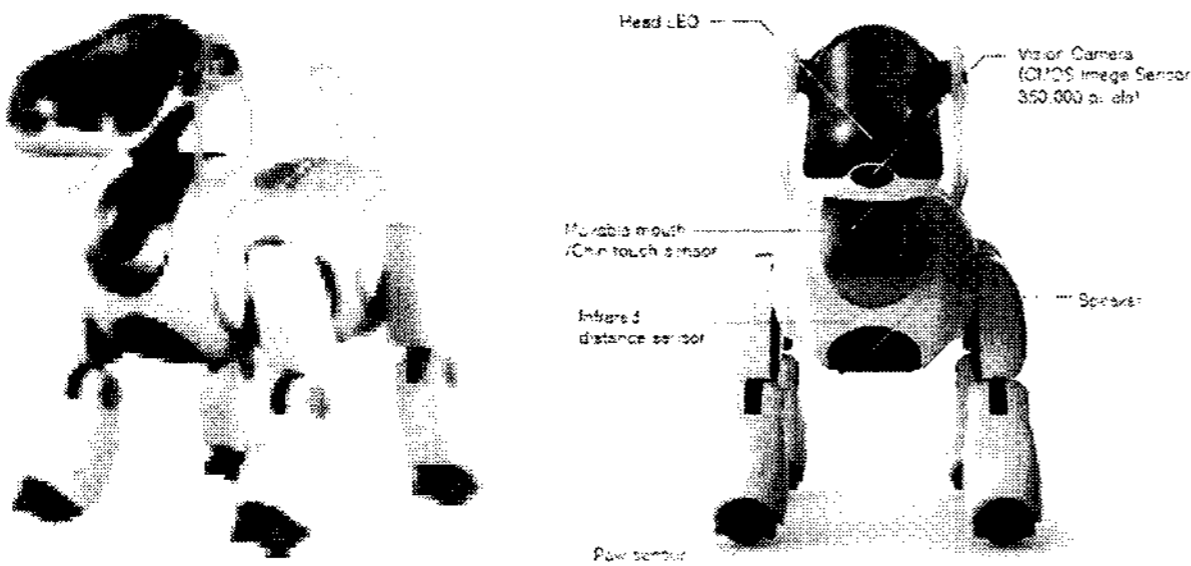


그림 1. SONY Aibo ERS-7.  
Fig. 1. SONY Aibo ERS-7.

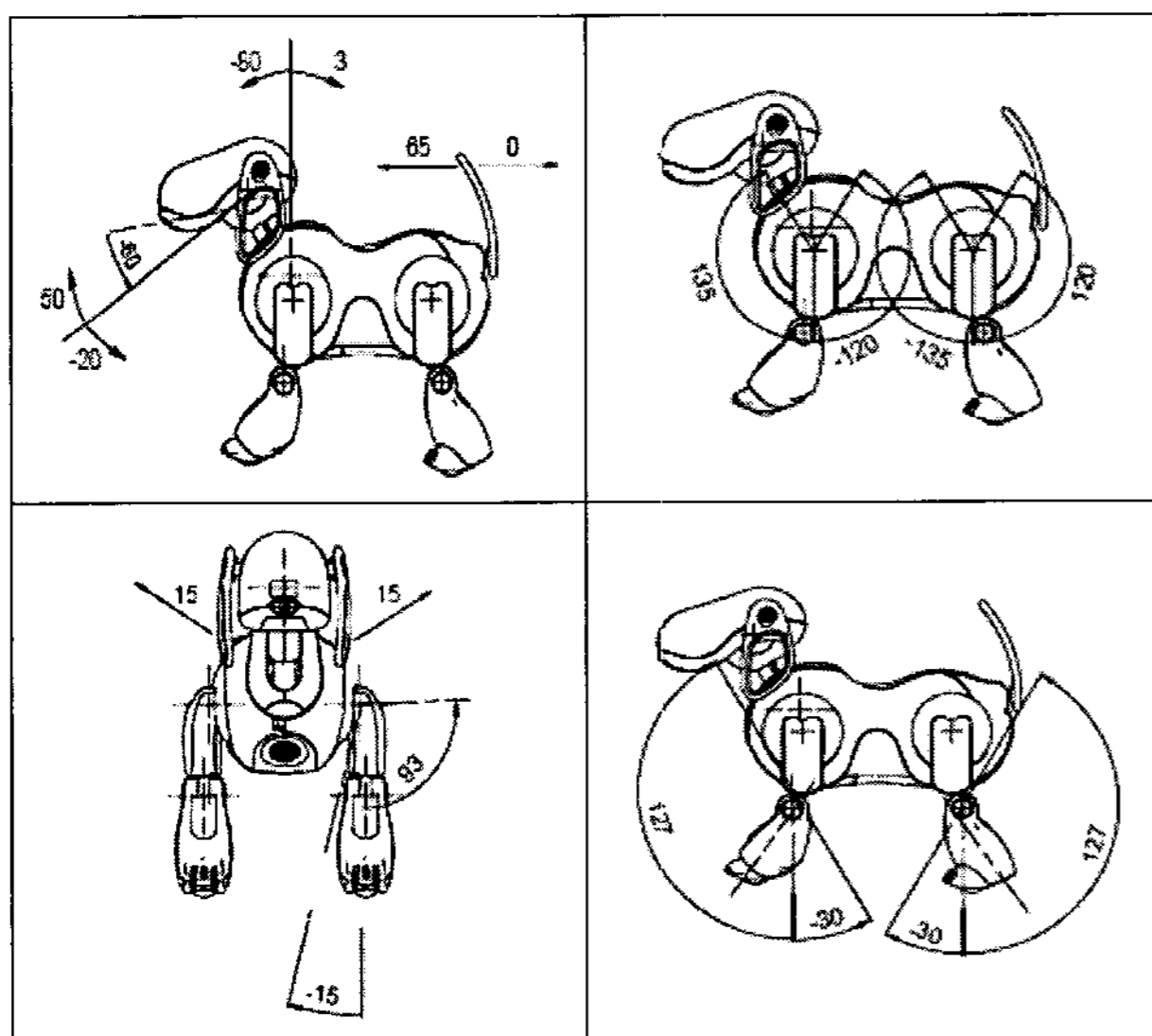


그림 2. SONY Aibo ERS-7의 관절 공간.  
Fig. 2. Joint space for SONY Aibo ERS-7.

걸음새 파라미터가 사용되었으며, 미리 결정된 사각형의 자취로부터 위의 분석적 기법을 통해 그림 3의 사다리꼴 자취를 구하였다.

Hornby[1]는 4족 보행로봇 모델로는 Aibo ERS-110을 대상으로 하여, 표 2와 같이 20개 이상의 걸음새 파라미터를 정의하여 사용하였다. 앞의 UNSW 팀과는 달리 발끝의 특정 자취를 사용하지 않았고, 직교좌표 공간에서 발끝의 위치 제어 대신에 스윙과 계인 방식을 이용하여 주기적인 걸음새를 생성하였다.

또 다른 접근법으로, Mericli[7]는 사각형이나 사다리꼴의

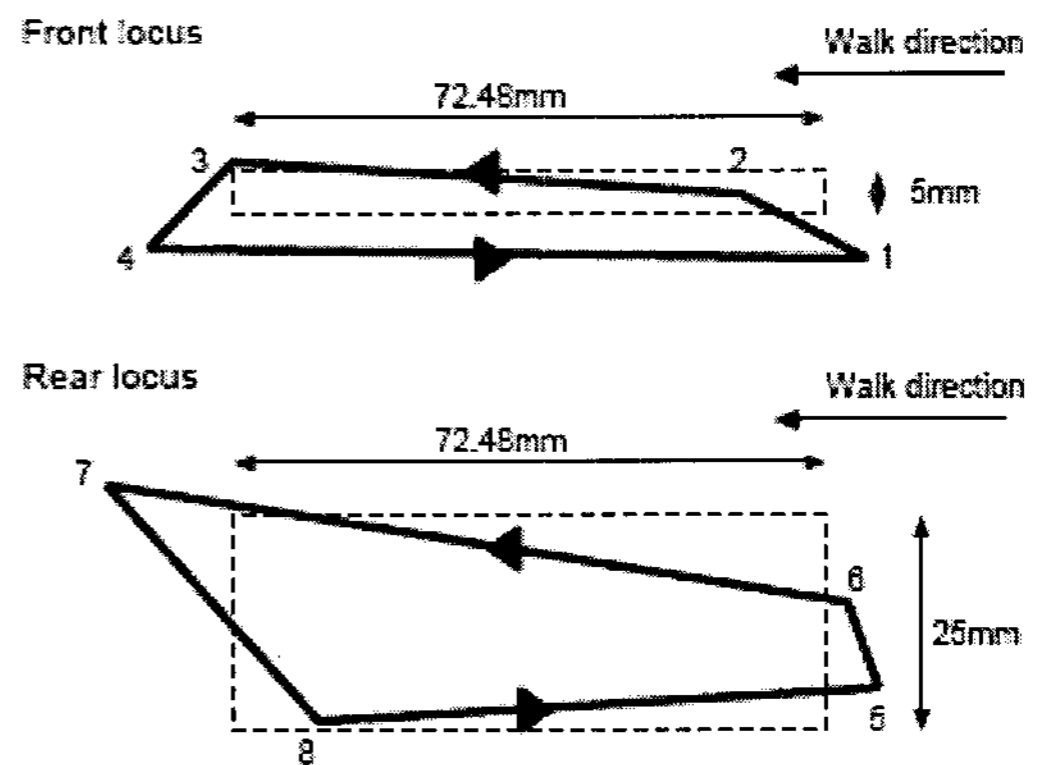


그림 3. UNSW 팀의 걸음새 자취.  
Fig. 3. Gait locus of UNSW team.

표 1. UNSW팀의 로봇 걸음새 파라미터.

Table 1. Gait parameters of UNSW team.

parameter	unit	initial range
walkType	n.a.	0
Forward	cm.	-8.0 - 8.0
Left	cm.	-4.0 - 4.0
turnCCW	degrees.	-30 - 30
PG	n.a.	65 - 70
hF	mm.	default 73.
hB	mm.	default 97.
hdF	mm.	default 16. must be > 0
hdB	mm.	default 19. must be > 0
ff0	mm.	default 33.
fs0	mm.	default 20.
bf0	mm.	default -35.
bs0	mm.	default 20.
hType	n.a.	0=no action, 1-relative, 2-absolute
tilt	degrees.	Up +ve, down -ve
pan	degrees.	Left +ve, right -ve.
mouth	degrees.	0=closed. -ve=down.

표 2. Hornby의 접근법의 걸음새 파라미터.

Table2. Gait parameters of Hornby's method.

parameter	unit	initial range
body center x	mm.	105 - 125
body center z	mm.	-10 - 10
body pitch	degrees	-10 - 10
posture center x	mm.	0 - 20
all legs y	mm.	-5 - 15
front legs z	mm.	10 - 30
rear legs z	mm.	-5 - 15
step length	n.a.	60 - 100
swing height front	mm.	25 - 45
swing height rear	mm.	25 - 45
swing time	ms.	460 - 540
swing mult.	n.a.	3 - 5
switch time	ms.	500 - 900
ampl. body x	mm.	-10 - 10
ampl. body y	mm.	-25 - -5
ampl. body z	mm.	-20 - 0
ampl. yaw	degrees	-10 - 10
ampl. pitch	degrees	-10 - 10
ampl. roll	degrees	-5 - 15
L-R	n.a.	0.25 - 0.5
F-H	n.a.	0.5 - 0.75

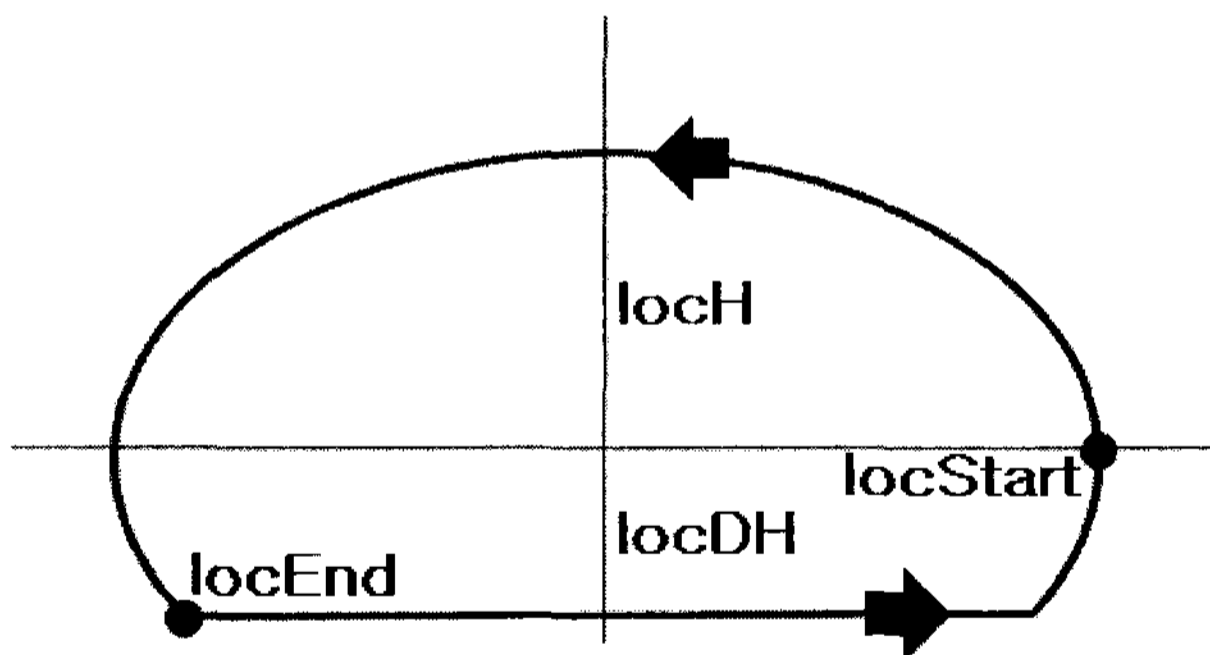


그림 4. 타원형의 걸음새 자취.

Fig. 4. Elliptical gait locus.

표 3. 기존 주요 연구 결과 요약.

Table3. Summary of previous approaches.

	발끝자취	속도
Hornby[1]	없음	9.97cm/s
Golubovic[4]	직사각형	언급 없음
Merikli[7]	타원형	26.99(±5)cm/s
rUNSWift[3]	사다리꼴	27cm/s
	기울어진 타원	43cm/s
German team[6]	3차원 다각형	45.1cm/s
Dong[2]	3차 스플라인	47cm/s

발끝 자취 대신에 그림 4와 같은 타원형을 선택하였고, 10여개의 걸음새 파라미터를 GA 유전자로 표현하여 최적화 시도를 하였다.

기존의 Aibo 걸음새 연구는 빠른 속도를 얻기 위한 측면에 집중되어 왔으며, 주요 우수 연구결과는 표 3에 나와 있다.

### III. GP 기반 관절 좌표계에서의 걸음새 제어

#### 1. 직교좌표공간에서의 걸음새 계획의 문제점

기존의 대부분의 연구가 사용하는 직교좌표계에서 발끝 자취를 제어하는 접근법은 먼저 어떠한 모양의 발끝 자취를 선택해야 하는지의 어려움이 있고, 자취가 선택되면 해당된 모양에 대해서만 관련 파라미터를 최적화해야 하므로 전체적인 최적화하는 거리가 있다.

또한 걸음새에 관련된 수많은 파라미터 중 최적화 대상 파라미터 선택의 어려움이 존재한다. 즉, 연구자마다 파라미터의 정의가 일정치 않고, 관련된 모든 파라미터를 다 포함시킬 수 없으며, 최소한의 파라미터도 10개 이상을 사용하고 있어 이들의 최적 값을 동시에 구하는 것은 매우 어려운 문제이다.

이 접근법은 발끝의 움직임이 특정 경로를 따라서 이동해야 하는 경우에는 장점이 있다. 그러나 Aibo 같은 형태의 4족 보행로봇은 평지 또는 굴곡이 심하지 않는 지형을 걷는 것이 주목적이고, 발끝의 자취가 특정 경로를 따를 필요가 적은 경우에는 위에서 언급한 파라미터 선택 및 수많은 파라미터에 대한 최적화의 어려움이 존재한다.

#### 2. 관절 공간(joint space) 걸음새 계획

이와는 달리 본 연구에서 제안된 관절 공간에서의 걸음새 계획법은 앞발과 뒷발에 대해서 각기 3개의 관절에 대한 궤적을 구하는 문제로 변환될 수 있다. 또한 시간에 따른 관절의 궤적을 하나의 수학 함수의 조합으로 구성함으로써 최대 6개 독립적인 변수의 최적화로 문제 공간을 축소할 수 있는 큰 장점이 존재한다. 이 방법은 발끝의 특정 경로를 계획할 필요가 별로 없는 Aibo와 같은 모델에 매우 유용하다.

아래의 그림 5에서 보는 것과 같이 시간에 따른 관절의 궤적(trajjectory)을 생성함으로써 각각의 관절의 움직임의 조

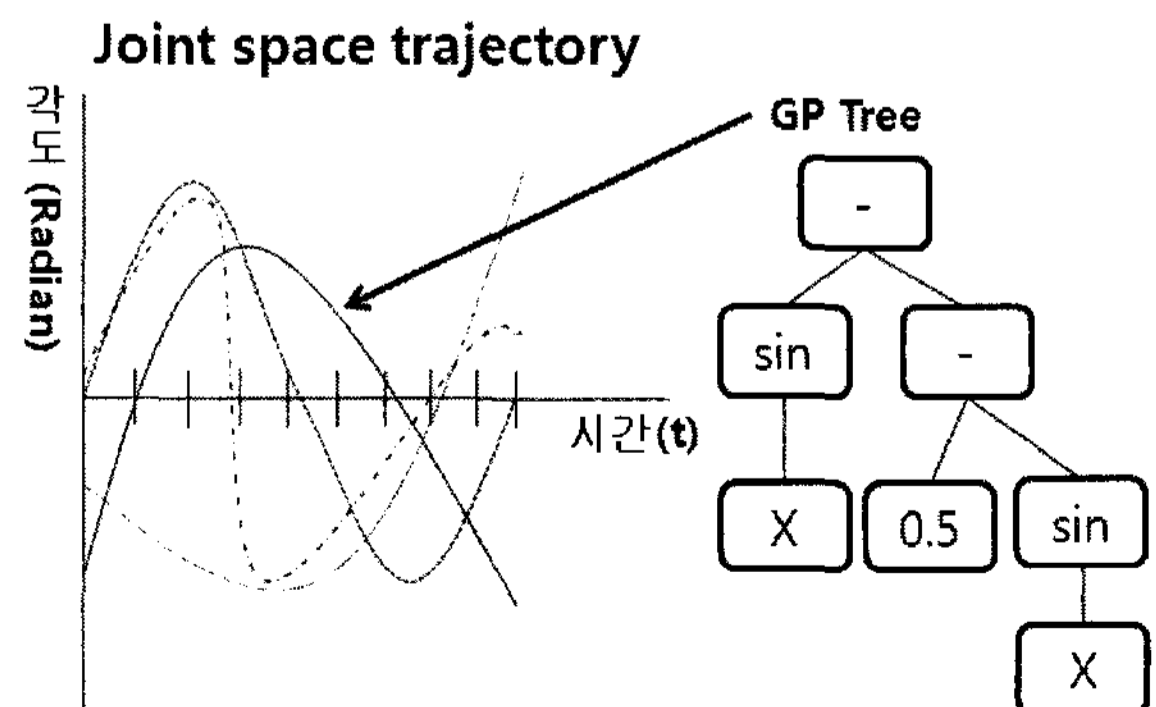


그림 5. 관절의 궤적 그래프 및 GP 트리 표현.

Fig. 5. Representation of trajectory of joints and GP tree.

합만으로도 기존의 직교좌표공간에서 쓰이는 걸음새와 유사한 결과를 얻을 수 있다.

그리고 시간에 대한 관절 궤적이 임의의 곡선 그래프로 표현되므로 유전자 표현이 간단해지는 장점이 있다. 특히, GP(Genetic Programming)[8,9] 트리로 제한되지 않는 곡선식을 생성할 수 있으므로 적은 연산으로 효율적이고 개선된 걸음새 생성이 가능하다.

더욱이 관절좌표계 접근법에서는 Aibo 로봇의 구조상 shoulder와 knee의 관절만이 직진보행에 관계하기 때문에 단지 앞뒤 2개씩 총 4개의 변수에 대해서만 시간에 따른 궤적을 결정하면 된다.

3. GP(Genetic Programming)

유전 프로그래밍(GP)[8,9]은 큰 부류에서 GA와 함께 진화 연산(evolutionary computation)에 속하지만, GA와는 몇 가지 다른 특징이 있다. 가장 큰 다른 점은 GA가 개체를 표현할 때 스트링을 사용하는데 비해 GP는 트리를 사용한다. 이때 트리에서의 각 노드가 하나의 함수를 나타내며, 각 개체는 함수의 집합, 즉 컴퓨터 프로그램으로 구성된다. 다시 말해 GP는 각 개체가 직접 해를 표현하는 것이 아니라, 해를 생성해 주는 하나의 프로그램을 표현한다. 또 다른 차이점은 대부분의 GA가 고정크기의 염색체를 사용하는데 비해 GP는 가변크기의 염색체를 사용한다.

Aibo 로봇의 걸음새 생성을 위해서 사용된 GP 함수와 터미널은 다음과 같다.

$$F = \{ \sin, \cos, \exp, +, -, *, / \},$$

$$T = \{ X, \text{erc} \}$$

본 연구에서는 하나의 해를 단일 트리로 구성하는 일반적인 GP 접근법과는 달리, 멀티-트리로 해를 구성하는 방법을 사용하여 진화의 효율성을 꾀하였다. 그림 6에는 위의 GP 함수와 터미널들의 조합으로 생성될 수 있는 임의의 GP 트리 예가 나와 있다.

4. GP 기반 관절공간 걸음새 생성법

GP를 통해 3개의 관절(현재 연구에서는 2개의 관절만 사용, flap 관절은 설계상 직진 걸음새에는 불필요함)에 대한 각각의 궤적 그래프에 대한 수식이 구해지고, 각 수식에 대한 시간의 변화 값을 각 관절에 대한 입력으로 사용된다.

제안된 GP 기반 관절공간 걸음새 생성법의 단계는 다음과 같다.

- 1) 제어할 로봇의 관절 선택

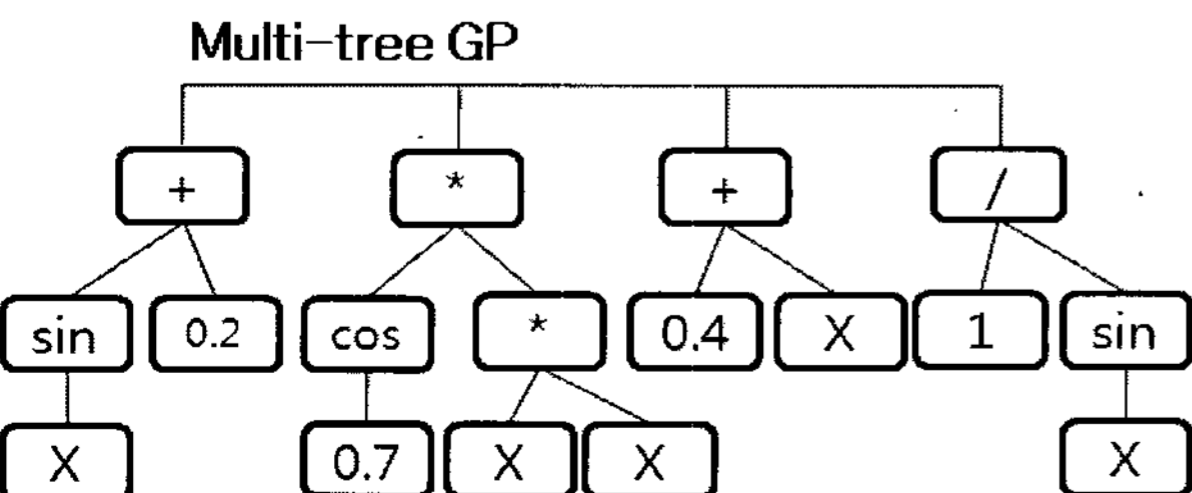


그림 6. Multi-tree GP.  
Fig. 6. Multi-tree GP.

- 2) 로봇의 걸음새 종류 선택(crawl, trot, pace 등)
- 3) 걸음새 생성용 GP 함수와 터미널 정의
- 4) 초기 트리 개체를 생성
- 5) 조건이 만족될 때까지 GP의 실행
- 6) 탐색된 GP 트리를 해석하여 관절 공간 궤적 생성
- 7) 각 관절에 대한 궤적 값들을 로봇(또는 시뮬레이터에) 입력

IV. 실험 및 결과

1. 시뮬레이션 환경

시뮬레이션 환경은 Cyberbotics 사의 Webots[10]을 사용하였다. Webots은 모바일 로봇에 대한 모델링, 프로그래밍, 그리고 시뮬레이션 기능을 제공하는 모바일 로봇 시뮬레이션 s/w 이다. 주요 기능은 다음과 같다. 각종 센서와 액츄에이터에 대한 라이브러리 제공, 정확한 physics 시뮬레이션을 위한 ODE(Open Dynamics Engine) 라이브러리 제공, 그리고 실제 모바일 로봇에 컨트롤러를 트랜스퍼 할 수 있는 기능을 가지고 있다.

Core 2 Duo 2.13 GHz PC에서 시뮬레이션을 수행하였으며, 평균 수행시간은 보간 스텝이 40일 경우에는 약 1시간 50분, 보간 스텝이 60일 경우에는 약 2시간 40분, 그리고 보간 스텝이 80일 경우에는 약 3시간 40분이 걸렸다.

2. 적합도 함수 및 GP 파라미터

걸음새 생성을 위한 적합도 함수는 직진 거리(x) 에다 측면(z) 의 벗어남을 벌칙으로 주어 구성하였다(그림 7).

$$fitness = ((0.8 * x) - (0.4 * z))^2 \tag{1}$$

여기서, 직진으로 이동한 거리(x) = +x  
양 옆으로 흔들린 폭(z) = (+z) ~ (-z)

사용된 GP 진화연산 제어 파라미터는 다음과 같다.

- Number of generations: 100
- Population sizes: 20× 5 (Multi-pop)
- Migration : Ring Migration
- Initial population: half\_and\_half
- Initial depth: 1-3
- Max depth: 12
- Selection: Tournament (size=7)
- Crossover: 0.9
- Mutation: 0.1

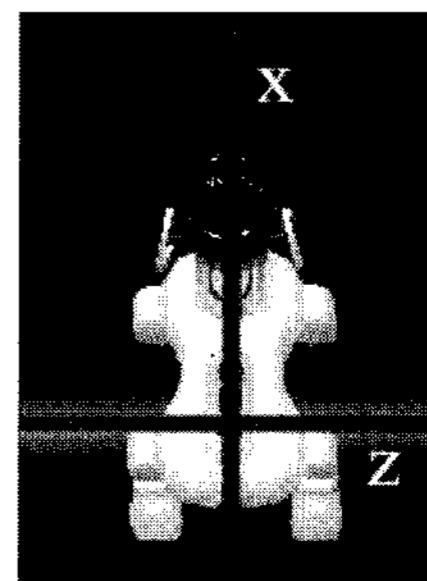


그림 7. Fitness 함수.  
Fig. 7. Fitness function.

3. 실험 결과

3.1 사이클 당 보간횟수(step) 변화

사이클 당 보간횟수(step)를 각기 40, 60, 80으로 변화시키면서 각 5회씩 반복 실험하였다. 여기서 side limit는 로봇의 좌우측 움직임 폭의 제한이며, 단위는 cm이다. 그리고 즉, 적합도가 높을수록 단위 시간당 직진성분의 이동거리가 많음을 나타낸다. 참고적으로 이동속도도 같이 구하였다.

아래 표 4에 나타난 결과를 보면 속도 면에서 step 수치 40을 이용한 실험이 가장 뛰어난 성능을 보여줌을 알 수 있다. 이 결과는 기존의 연구결과인 UNSW팀[3]의 경우에도 40~50 step 시에 가장 빠른 속도를 보여 준다고 언급 되었으며, 실제 우리의 실험에서도 이를 확인할 수 있었다.

일반적으로 step이 작을수록 더 빠른 속도를 내는데 유리하며, step 수치가 커지면 속도보다는 더 안정적인 움직임을 내는데 유리하다고 알려져 있다. 참고로 40 step 이하의 실험에서는 동작이 불안정함을 보였다.

3.2 GP 함수 구성 변경

표 4의 실험 최적화 결과를 토대로 step 부분에 있어서는 40 step이 가장 빠른 속도를 보여주기때 step은 고정을 시키고 새롭게 GP 트리를 구성하는데 있어서, 근사식 구성에 많이 쓰이는 exp 함수를 추가 시켰다.

그러나 exp 함수가 추가 시 로봇의 움직임이 직진하지 않고 불안정한 형태가 나타난 것을 볼 수 있었다. 이는 지수적인 증가 형태를 보이게 되면 상당히 급격한 움직임을 나타내기 때문에 로봇의 속도는 빠르게 변하지만 더 불안정한 움직임을 보여주는 것으로 생각된다(표 5).

표 4. 사이클당 보간횟수 변화 결과.

Table4. Results by variation of step numbers per cycle.

	Side Limit 10(SL10)			Side Limit 15(SL15)		
	No	Fitness	cm/s	No	Fitness	cm/s
40 Step	1	1.483	44.8	1	0.861	37.89
	2	1.110	39.5	2	1.661	50.7
	3	2.036	50.2	3	1.391	43.35
	4	1.275	39.7	4	1.297	46.01
	5	2.165	51.9	5	2.592	65.23
	avg	1.614	45.22	avg	1.560	48.64
60 Step	1	1.174	26.04	1	2.017	37.5
	2	1.821	29.68	2	2.547	44.01
	3	2.240	40.1	3	3.486	40.62
	4	3.014	41.6	4	3.720	50.7
	5	2.460	45.05	5	3.013	41.6
	avg	2.142	36.5	avg	2.957	42.89
80 Step	1	4.141	38.4	1	2.920	33.57
	2	0.738	16.6	2	3.051	27.92
	3	2.120	31.45	3	2.619	31.64
	4	3.105	30.85	4	5.334	48.04
	5	3.753	33.2	5	4.436	43.06
	avg	2.771	30.1	avg	3.672	36.85

3.3 Pop 크기 변화

속도와 안정성 면에서 모두 좋은 성능을 보여주는 40 step의 side limit 15에 대해 개체 집단의 크기를 변화시켜 실험을 했다. 일부 데이터를 제외하고는 개체집단의 크기가 증가함에 따라 더 우수한 결과를 보여주고 있다(표 6).

가장 좋은 성능을 보여준 경우는 의외로 가장 적은 개체 수를 이용한 20×5 Popsizes의 5번 데이터이지만, 이것은 진화연산이 방향성을 가진 확률적 탐색법이므로 흔치 않은 경우가 나온 것으로 볼 수 있다.

5회 평균값을 기준으로 했을 때 100×5의 개체집단을 사용한 실험이 평균 속도 59.04(cm/s) 로 가장 우수한 성능을 보여주고 있다.

표 5. exp 함수 추가시의 결과.

Table5. Results by adding exp function.

	Side Limit 10(SL10)			Side Limit 15(SL15)		
	No	Fitness	cm/s	No	Fitness	cm/s
40 Step	1	1.3724	44.31	1	1.3781	47.22
	2	1.1068	37.22	2	1.3992	47.1
	3	2.3463	61.8	3	2.4259	45.31
	4	1.5373	50.4	4	1.9569	55.62
	5	1.7288	53.51	5	2.1151	46.69
	avg	1.618	49.45	avg	1.855	48.39

표 6. 개체 집단크기 별 성능 결과.

Table6. Results by various pop sizes.

	Side Limit 15(SL15)		
	No	Fitness	cm/s
20×5 Popsizes	1	0.8605	37.89
	2	1.6606	50.7
	3	1.3914	43.35
	4	1.2974	46.01
	5	2.5923	65.23
	avg	1.560	48.64
50×5 Popsizes	1	2.1507	50.7
	2	2.8370	31.2
	3	2.0402	57.2
	4	2.1039	52.04
	5	2.5806	61.7
	avg	2.342	50.57
100×5 Popsizes	1	2.8794	56.91
	2	1.8150	54.92
	3	2.6361	63.28
	4	2.0913	57.77
	5	2.5422	62.34
	avg	2.393	59.04

3.4 관절 궤적 데이터 및 로봇의 움직임

그림 8-그림 16에 4가지 대표적인 걸음새에 대한 관절 궤적 및 로봇의 걸음새 동작을 나타내었다. 서로 다르게 생성된 관절 궤적의 그래프를 볼 수 있으며, 걸음새 동작은 8개의 프레임으로 구분하여 구성하였다. 그림 8의 관절 궤적 그래프에서 4개의 곡선은 각각 다음을 의미한다.

FJ1: Front Shoulder, FJ2: Front Knee, BJ1: Back Shoulder, BJ2: Back Knee.

- (1) 40 Step, 20×5 Popsize, SL15. exp 함수, 4번째
- (2) 40 Step, 50×5 Popsize, SL15, 5번째
- (3) 40 Step, 100×5 Popsize, SL15, 3번째
- (4) 40 Step, 20×5 Popsize, SL15, 5번째(Best)

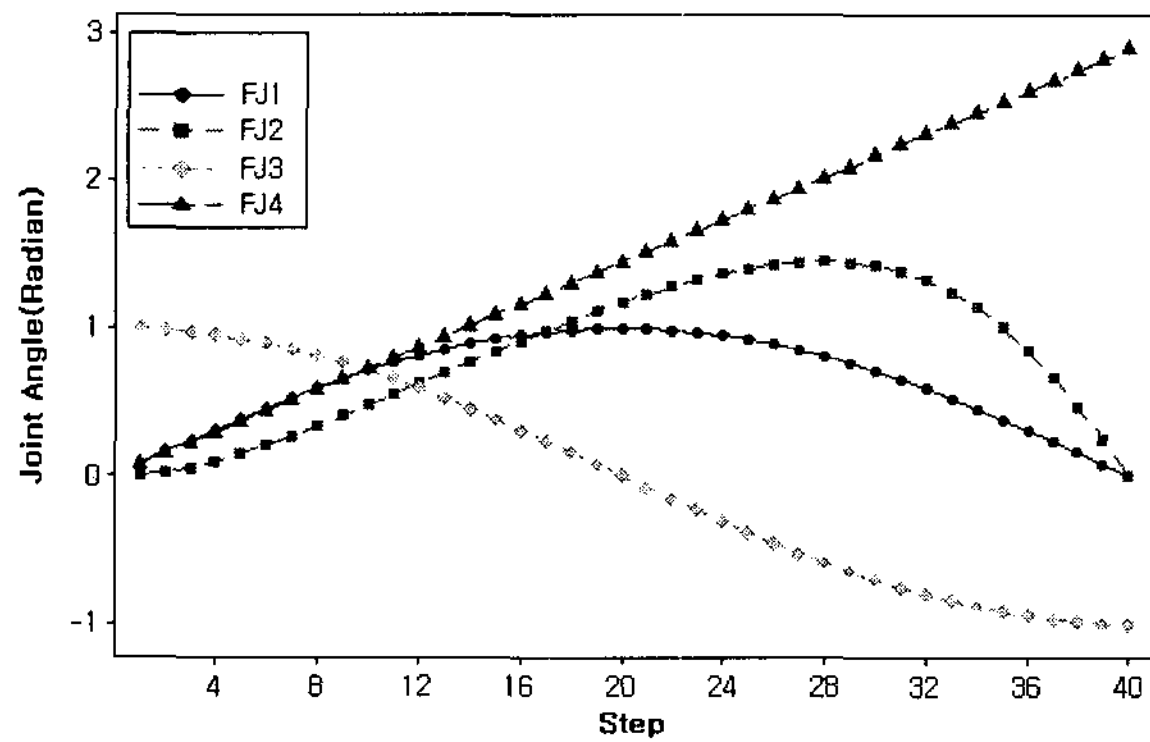


그림 8. 관절 궤적 그래프(1).

Fig. 8. Joint trajectory graph(1).



그림 9. 걸음새 동작(1).

Fig. 9. Gait motion(1).

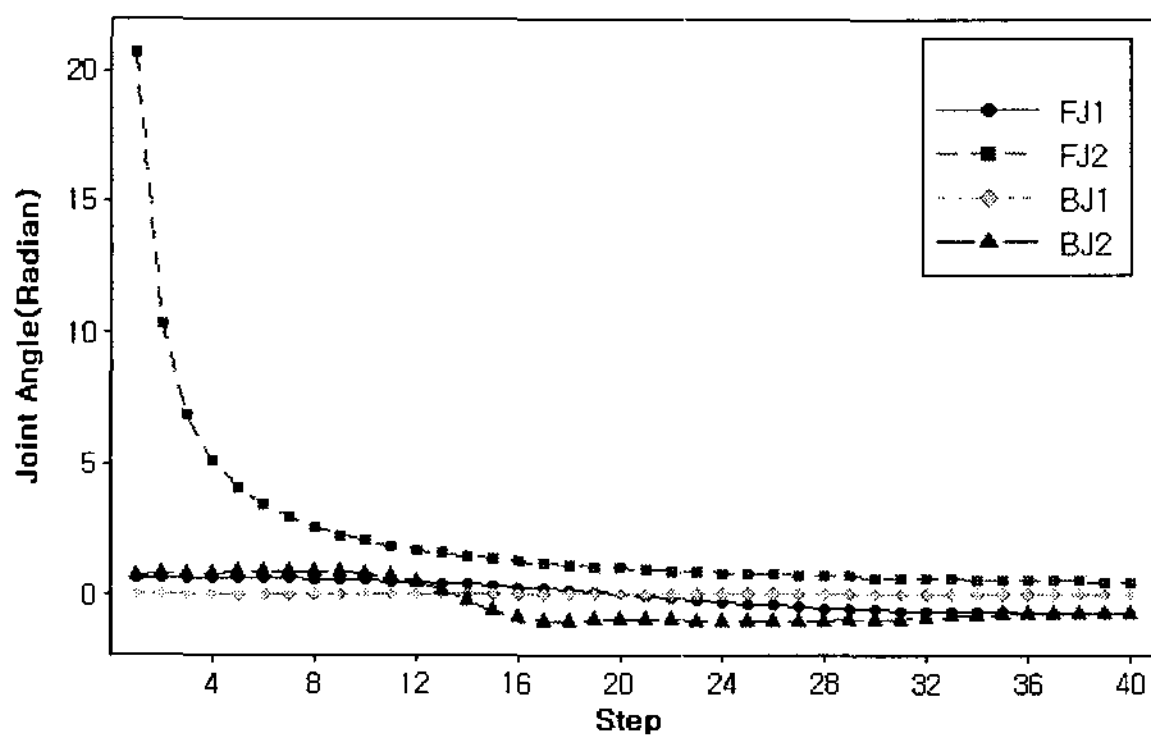


그림 10. 관절 궤적 그래프(2).

Fig. 10. Joint trajectory graph(2).

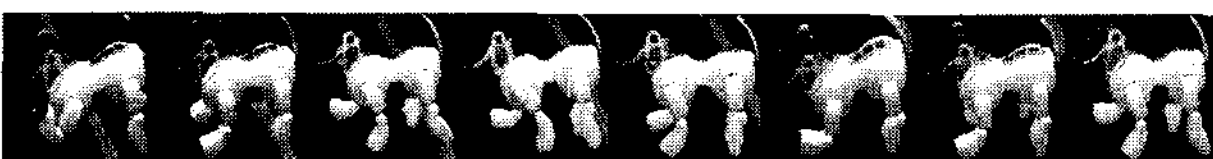


그림 11. 걸음새 동작(2).

Fig. 11. Gait motion(2).

걸음새 동작(1)에서는 로봇이 자세를 약간 낮추면서 걷는 동작을 볼 수 있고, 걸음새 동작(2)에서는 비교적 서있는 자세처럼 높은 자세를 유지하고 있다. 특히하게, 걸음새

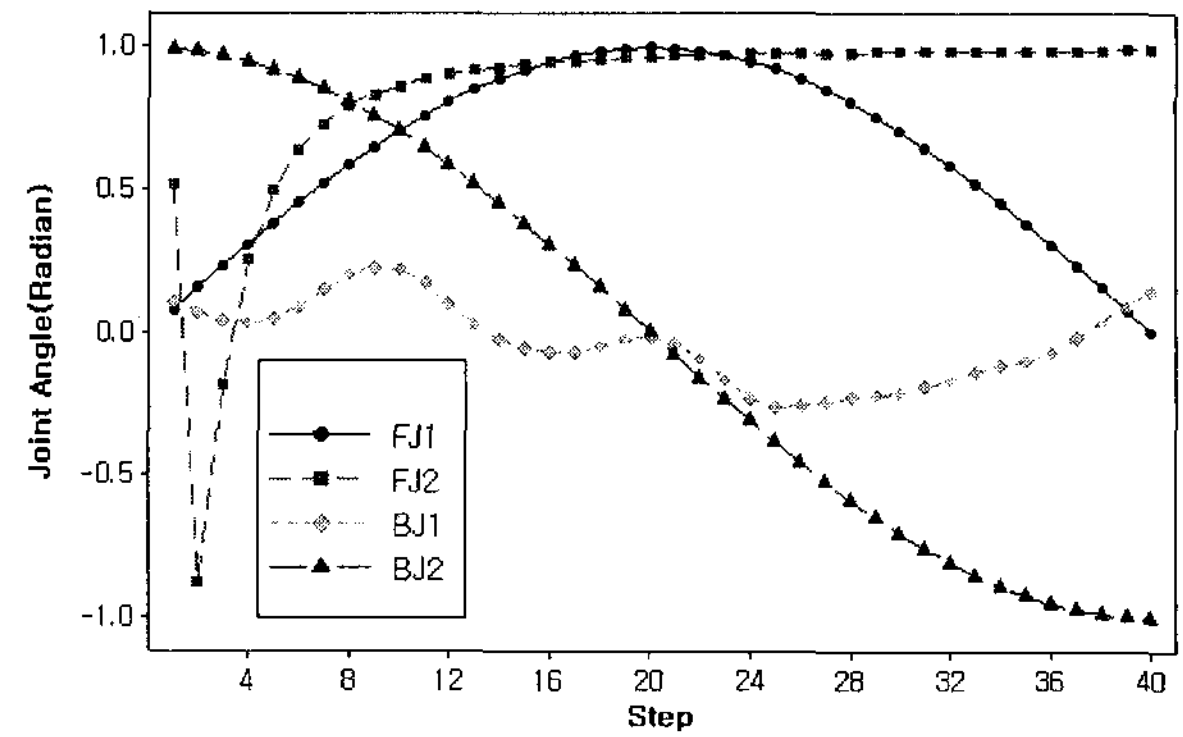


그림 12. 관절 궤적 그래프(3).

Fig. 12. Joint trajectory graph(3).



그림 13. 걸음새 동작(3).

Fig. 13. Gait motion(3).

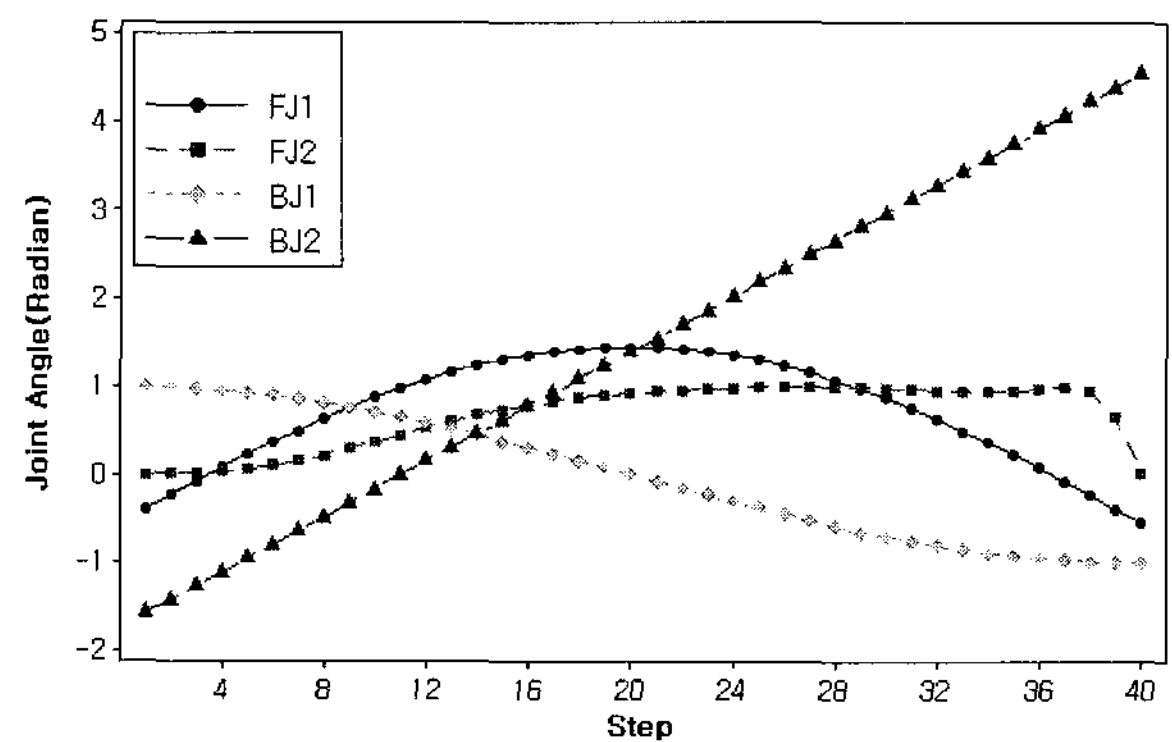


그림 14. 관절 궤적 그래프(4).

Fig. 14. Joint trajectory graph(4).



그림 15. 걸음새 동작(4).

Fig. 15. Gait motion(4).



그림 16. 걸음새 동작(4)의 확대.

Fig. 16. Enlargement of gait motion(4).

동작(3)에서는 자세를 아주 낮추어 앞발을 안으로 모으고 기어가는 자세를 취하고 있다. 이 자세는 기존의 연구결과들에서도[3,5,7] 공통적으로 나타나고 있다. 걸음새 동작(4)는 걸음새 동작(1)과 비교적 유사한 형태를 보이고 있다.

Aibo 모델에 대한 걸음새 동작 기존 연구 결과가 다른 로봇 모델 및 수행환경, 그리고 시뮬레이션 프로그램의 오차 등에 따라 약간씩 차이가 존재하여 엄밀한 비교가 어렵지만, 기존 연구 중 가장 좋은 결과의[2] 최대 속도가 47cm/sec임을 감안하면 본연구의 관절좌표계 기반 GP 최적화 기법이 매우 우수한 성능을 보여주는 것을 알 수 있다.

## V. 결론

기존의 GA등을 이용한 4족 보행로봇의 자동 걸음새 생성 기법들은 직교좌표 공간에서 사전에 제한된 주요 파라미터들에 한정된 최적화 문제로만 접근하고 있으며, 제한적으로 사용된 최소한의 파라미터도 10-20개 이상을 사용하고 있어 이들의 최적 값을 구하는 것은 현실적으로 매우 어려운 문제이다.

본 연구에서는 특정 경로계획이 필요하지 않은 Aibo의 걸음새 생성에 대해서, 위와 같은 GA 기법의 단점을 극복하기 위하여 직교좌표계에서의 걸음새 자취 대신 관절 공간에서의 관절 궤적을 GP(Genetic Programming)로 자동 생성하는 기법을 제안하였다. 이를 통해, 시간에 따른 관절의 궤적을 하나의 수학적 함수의 조합으로 구성함으로써 최대 6개 독립적인 변수의 최적화로 문제 공간을 대폭 축소할 수 있는 큰 장점이 존재한다.

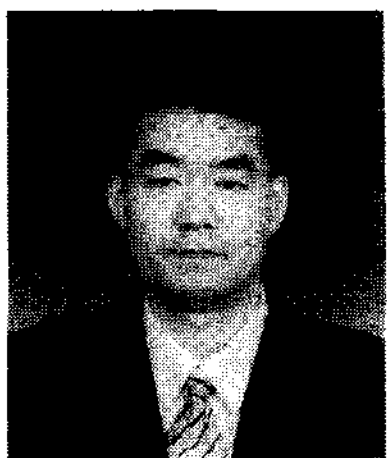
4족 보행로봇의 걸음새 자동 생성 기법의 구현을 위하여, GP 기반의 관절공간에서의 걸음새(gait) 생성용 GP 함수 및 터미널 구현하고, 보행속도와 안정성을 최적화하는 걸음새의 자동 생성 방법을 구현하였다. 제안된 기법의 검증은 위하여 SONY Aibo 4족 보행 로봇에 대해서 Webots [10]을 이용한 ODE 기반의 시뮬레이션 실험을 수행하였으며, 기존 연구보다 우수한 성능을 나타내었다.

## 참고문헌

- [1] G. S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita, "Autonomous evolution of dynamic gaits with

two quadruped robots," *IEEE Trans. Robotics*, vol. 21, no. 3, pp. 402-410, 2005.

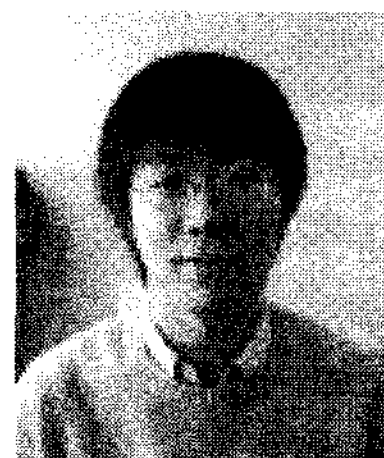
- [2] H. Dong, M. Zhao, J. Zhang, Z. Shi, and N. Zhang, "Gait planning of quadruped robot based on third-order spline interpolation" in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems Beijing*, China, 9-15, October 2006.
- [3] Z. D. Wang, J. Wong, T. Tam, B. Leung, M. S. Kim, J. Brooks, A. Chang, and N. V. Huben, The 2002 rUNSWift Team Report, 2002.
- [4] D. Golubovic and H. Hu, "Parameter optimisation of an evolutionary algorithm for on-line gait generation of quadruped robots," in *Proceedings of IEEE International Conference on Industrial Technology - ICIT'03*, Maribor, Slovenia, Dec. 2003.
- [5] S. Chernova and M. Veloso, "An evolutionary approach to gait learning for four-legged robots," in *Proceedings of IROS'04*, Sendai, Japan, September 2004.
- [6] T. Röfer, T. Laue, and D. Thomas. *Particle-Filter-Based Self-Localization Using Landmarks and Directed Lines*. In *RoboCup 2005: Robot Soccer World Cup IX, Lecture Notes in Artificial Intelligence*, Springer, 2005.
- [7] T. Mericli, H. L. Akın, C. Mericli, K. Kaplan, B. Celik, The Cerbus'05 Team Report.
- [8] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, 1992.
- [9] J. R. Koza, F. H. Bennett, D. Andre, M. A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann Publishers, 1999.
- [10] L. Hohl, R. Tellez, O. Michel, and A. J. Ijspeert, "Aibo and webots: simulation, wireless remote control and controller transfer," *Robotics and Autonomous Systems*, 54(2006), pp. 472-485, 2006.
- [11] Sony Corporation, Model information for ERS-7, OPEN-R SDK, 2004.



서기성

1986년 연세대학교 전기공학과 공학사. 1988년 연세대학교 전기공학과 공학석사. 1993년 연세대학교 전기공학과 공학박사. 1993년~1998년 서경대학교 산업공학과 조교수. 1999년~2003년 Michigan State University, Genetic

Algorithms Research and Applications Group, Research Associate. 2002년~2003년 Michigan State University, Electrical & Computer Engineering, Visiting Assistant Professor. 2003년~2004년 서경대학교 전자공학과 조교수. 2004년~현재 서경대학교 전자공학과 부교수. 관심분야는 GA, GP, 진화 디자인, 지능로봇.



현수환

2003년~현재 서경대학교 전자공학과 학사과정.