

유전 알고리즘을 사용한 뱀형 로봇의 이동 생성 및 부분모듈 선택 분석

Generation of Locomotion for Snake-like Robot using Genetic Algorithm and Analysis for Selections of Partial Modules

안인석* · 장재영** · 서기성**+

Ihnseok Ahn*, Jaeyoung Jang** and Kisung Seo**+

* 위덕대학교 에너지전기공학부

** 서경대학교 전자공학과

요 약

뱀형 모듈라 로봇은 모듈을 일련의 형태로 연결하여 구성된 것으로, 다양한 환경에 대해서 강인성을 가지고 있고, 모듈 일부의 고장에도 이동할 수 있는 장점을 가진다. 그러나 이동 제어 방법이 어렵고, 아직까지 효율적이고 다양한 이동법의 개발이 미비한 편이다. 본 연구에서는 뱀형 로봇의 이동제어를 위하여 GA(Genetic Algorithm)기반의 위상생성 방식과 임의의 궤적 생성방식을 비교하고, 이를 확장하여 일부 모듈만의 선택에 따른 영향을 분석하기 위해서 GA를 통한 모듈 선택 실험을 수행하였다. KMC사의 뱀형 로봇을 대상으로 먼저 webots 시뮬레이터 상에서 모델링 및 시뮬레이션 환경을 구축하고, 위의 GA 기반 이동 생성 실험들을 수행하였다.

키워드 : 뱀형 로봇, 자동 이동 생성, 유전 알고리즘, 모듈라 로봇, 모듈 선택

Abstract

Modular snake-like robots, which consist of series of modules, are robust for failure and have flexible locomotions for environment. However, they are difficult to control and few efficient and various locomotions are introduced yet. In this paper, GA based phase generation and trajectory generation approaches are implemented and compared for locomotion of snake-like robots and extended for analysis for selections of partial modules. In addition, modeling and simulation environments are implemented in Webots simulator and above GA based experiments for locomotion are executed for KMC snake-like robot.

Key Words : Automatic Locomotion Generation, Genetic Algorithm, Snake-like Robot, Modular Robot, Module Selection

1. 서 론

진화로봇 연구(Evolutionary Robotics)는 지능적이고 자율적인 로봇 시스템을 자동적으로 구현하는 방법을 개발하는 것이다. 로봇의 컨트롤러, 몸체 또는 컨트롤러와 몸체를 동시에 진화시키는 것을 목적으로 한다[1,2]. 모듈라 로봇이란 독립적인 단위 구성 요소인 각 모듈을 원하는 기능을 수행할 수 있도록 특정한 형태로 연결하여 만든 로봇으로서 [1-3], 진화로봇의 개념에 가장 근접하게 부합되는 특징을 가지고 있다.

모듈라 뱀형 로봇은 모듈을 일렬로 연결하여 구성된 로봇으로서, 기존의 바퀴가 달린 로봇이나 보행 로봇에 비하여 다음과 같은 장점이 있다. 일부 모듈이 고장 난 상태나 일부 모듈만을 사용하는 경우에서도 동작이 가능하고, 계단

등의 장애물이 있는 지형 혹은 불안정한 환경에서도 유연하게 동작할 수 있다. 그러나 기존의 로봇들에 비하여 이동속도가 느리고 제어가 어렵다는 단점이 있다[4-6].

기존의 모듈라 뱀형 로봇의 이동 제어에 사용된 방식으로는 삼각함수로 표현된 식을 모듈마다 주기가 다르게 적용하여 관절의 움직임을 제어하는 방식이 있다[7,8]. 그리고 CPG(Central Pattern Generator)를 이용해 걸음걸이를 생성하는 방식도 있다[5]. 그러나 이런 방법들은 생물학적인 운동 형태를 모방하여, 안정적인 움직임을 생성하는데 중점을 두어서, 걸음걸이의 최적화가 되지 않았고, 속도와 같은 주요 성능 지표의 개선이 필요한 상황이다.

최근 모듈라 로봇의 이동 제어에 지능적 기법을 사용한 연구가 수행되어 왔고[4-6,8,9], 상당한 성과를 보여 주고 있다. 그러나 현재 대부분의 연구들은 가상의 모듈만을 사용하고 있는 등, 실제 로봇에의 적용 및 모듈라 로봇의 다양한 특성을 활용한 이동 방식 연구는 미흡한 편이다.

본 논문에서는 GA 기반의 위상생성 방식과 임의궤적 생성방식을 비교하고, 뱀형 로봇의 전체 모듈대신 일부 모듈을 사용할 경우에 대한 모듈 선택 문제와 이동 성능과의 관계를 분석한다. KMC사의 실제의 뱀형 로봇을 대상으로, Webots[11]을 사용한 시뮬레이션과 실제 실험을 수행하였다.

접수일자 : 2009년 7월 9일

완료일자 : 2009년 9월 29일

+ 교신저자

본 논문은 2007년 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음. (KRF-2007-314-D00176)

2. 모듈라 뱀형 로봇의 이동제어

2.1 모듈라 뱀형 로봇의 이동 방법

모듈라 뱀형 로봇의 이동 방법으로는 크게 수직적인 관절 움직임을 가지는 방법과 수평적인 관절 움직임을 가지는 방법, 수직과 수평 모두 움직일 수 있는 방법이 있다. 수평적인 관절의 움직임을 가지는 방법으로는 몸체의 측면 방향으로 전진하는 사이드와인더 방식[6,8], S모양으로 움직이는 측면 파동 방법(Serpentine)[5,6] 등이 있다. 수직적인 관절의 움직임을 가지는 방법은 일반적인 애벌레의 움직임과 같은 Caterpillar 방식, 꼬리부분으로 몸체를 밀어 이동하는 Inching 방식[6] 등이 있다. 그 외에 수평 수직 복합적인 움직임의 방법으로는 Helix Theory 방법[9]이나 Screw 방법[6] 등이 있으며, 이들 방법들은 모듈이나 관절의 구조와 연결방식에 적합하게 선택된다.

대부분의 모듈라형 뱀 로봇에서 수평관절은 수직관절의 보조로서 회전을 하는데 사용되고 있으며, 이동을 위한 경우는 모듈 밑에 바퀴가 있거나, 수면 위를 이동하는 특수한 환경에서만 사용되고 있다. 따라서 본 연구에서도 수직관절의 조합에 의한 이동을 제어 대상으로 한다.

기존에 가장 널리 쓰이는 주기적 제어방식은 식 (1)과 같이 각 모듈마다 사인파에 일정한 위상차를 입력한다.

$$\sin(t+n\phi) \quad (n=0,1,2\dots) \quad (1)$$

3. 진화연산을 기반으로 한 이동제어

GA(Genetic Algorithm)[10]는 진화연산의 한 종류로서, 비트열이나 실수형태의 염색체를 이용한다. GA는 고정된 크기의 염색체로 구성되며, 주로 수치최적화에 적합하다.

3.1 GA 기반 위상생성(GAPs)

이 방식은 식 (2)에 나타난바와 같이, 사인 함수의 위상(phase, ϕ)만을 가변 파라미터로 하여 움직임을 결정한다. 즉, 기존의 일반적인 제어방식은 고정된 위상 값을 순차적으로 일정하게 증가시켜 사용하는 데 비해서, 이 방식은 모든 모듈이 최대와 최소 범위 내에서 임의의 위상을 가질 수 있다. 개체는 7개의 모듈에 대해서 각 1개의 파라미터를 가지고 총 7개의 염색체를 사용한다.

$$\sin(t+\phi) \quad (2) \quad (-6.28 \leq \phi \leq 6.28)$$

3.2 GA 기반 임의계적생성

각 모듈에 대한 관절 값을 제어하기 위해서, 식 (3)과 같이 주어진 사인함수에 대해서, 진폭(A)과 주기(ω), 위상(ϕ)의 3가지 파라미터를 모두 결정하고, 이에 따라 생성된 식을 통하여 움직임을 생성한다.

$$A\sin(\omega t+\phi) \quad (3) \quad (-1.57 \leq A \leq 1.57, -1.5 \leq \omega \leq 1.5, -6.28 \leq \phi \leq 6.28)$$

또한, 유전해 표현에서 한 모듈은 진폭(A), 주기(ω), 위상(ϕ)의 3개의 필드로 구성되며, 한 개체는 7개의 모듈을 가지므로 총 21개의 필드를 포함한다.

기본적인 유전해에 7개의 모듈 중 어느 모듈을 사용할지를 나타내는 선택 필드를 추가하여, 3.3절의 부분 모듈 선

택에 의한 이동 생성에도 사용될 수 있다.(그림 1)

Modules	1 st Module	2 nd Module	3 rd Module	4 th Module	5 th Module	6 th Module	7 th Module
Selection	Used	Used	Used	Used	Used	Used	Used
Locomotion Parameter	Amp	Amp	Amp	Amp	Amp	Amp	Amp
	Period	Period	Period	Period	Period	Period	Period
	Phase	Phase	Phase	Phase	Phase	Phase	Phase

그림 1. GA 유전자(Chromosome)의 구성
Fig. 1. Construction of Chromosome

3.3 GA 기반 부분 모듈 선택에 의한 이동 생성

뱀형 모듈라 로봇의 주요 특징 중 하나는 일부 모듈이 고장이 나서 사용을 못하더라도 이동이 가능한 점이다. 따라서, 이러한 상황에서 정상 모듈만을 가지고 이동 방법을 생성하는 것은 매우 중요한 문제이다. 또한, 모든 모듈이 정상적인 상태에서 최소한의 또는 주어진 수의 모듈만을 사용하여 이동을 하는 문제도 매우 실용적이고 흥미로운 문제이다.

본 연구에서는 주어진 모듈수에 맞추어 효율적인 모듈의 선택 및 선택된 모듈의 궤적을 GA로 생성하고자 한다. 전체 7개의 모듈을 다 사용하는 대신에, 4, 5, 6 개만을 사용할 경우에 대한 부분 모듈의 선택에 따른 영향을 분석한다. 사용된 유전해 표현은 그림 1 과 같으며, selection 필드에서 선택된 모듈은 1로 그렇지 않은 모듈은 0로 표시된다.

4. 뱀형 로봇의 모델링

4.1 KMC 뱀형 로봇

본 논문의 실험에서 사용된 로봇은 KMC Robotics사의 뱀형 로봇 KMC-KOBI이다[12].

로봇은 10개의 RC 서보모터로 이루어진 관절과 전원, 8개의 프레임과 CPU인 Atmega8535로 구성되어 있다. 로봇의 무게는 약 11 킬로그램(11.15 Kg)이고, 총 길이는 1 미터이다. 관절은 3 개의 수평 관절과 7 개의 수직 관절로 이루어져 있다.

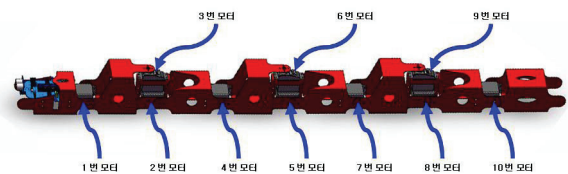


그림 2. 뱀형 로봇의 모습
Fig. 2. The shape of snake-like robot

4.2 로봇의 모델링 과정과 수치 설정

KMC 뱀형 로봇을 시뮬레이션하기 위해서 기본 설계도에서 주어진 정보를 바탕으로 모델링을 수행하였다. 머리의 프레임과 PCB1를 포함한 부분부터 시작하여 꼬리 프레임까지 차례대로 관절이 상위 관절의 움직임에 의해 영향을 받

도록 하위 관절이 상위 관절에 귀속되는 형태로 구성된다.

구체적인 모델링 과정은 다음과 같다. 그래픽 프로그램 Solid Works를 이용하여 iqs 확장자로 구성된 KMC 로봇의 설계도면을 표준 VRML2.0의 포맷의 wrl 확장자 파일로 변환한다. Webots 시뮬레이터의 Import 기능을 사용하여 표준 VRML2.0 포맷으로 변환된 로봇의 설계도를 Webots의 world에 불러온다. 로봇을 프레임별로 분해하고 각 프레임의 조합으로 기본적인 머리, 꼬리, 몸통의 모듈 형태를 구성한다.(그림 3) 이렇게 구성된 각 모듈을 서보와 연결부를 이용하여 연결하고 모듈들을 결합시켜 로봇을 구성한다.(그림 4)

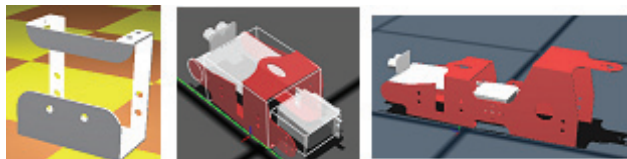


그림 3. 뱀형 로봇의 부분 모델링 과정

Fig. 3. Process of partial modeling of snake-like robot

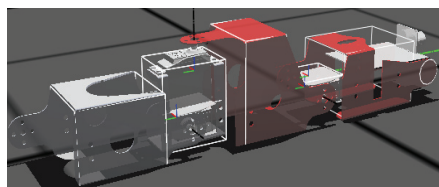


그림 4. 머리와 몸통 모듈의 결합 과정

Fig. 4. Connection process of head module and body module

모든 구성부인 프레임과 서보에 충돌과 물리현상을 적용하기 위해서 실제적인 Bounding Object를 구현한다. 그리고, 개체들의 크기, 좌표 등의 기하 정보와 무게, 마찰 및 토크 등의 물리 수치를 설정한다.(그림 5) 최종적으로, 완성된 모델에 대해서 각 모듈에 대한 서보의 목표값을 대입하여 로봇의 움직임이 정상적인지를 확인한다.(그림 6)

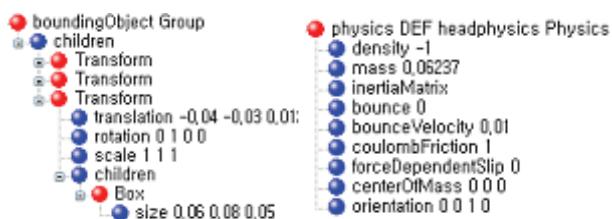


그림 5. 기하 및 물리 정보의 설정

Fig. 5. Parameters of geometry and physics

5. 컴퓨터 시뮬레이션

5.1 Webots 시뮬레이션 환경

진화연산에 의한 이동방식을 생성하기 위해서는 반복적인 시뮬레이션 수행이 필요하다.

시뮬레이션 환경은 Cyberbotics사의 Webots[11]을 사용하였다. Webots는 모바일 로봇에 대한 모델링, 프로그래밍, 그리고 시뮬레이션 기능을 제공하는 모바일 로봇 시뮬레이

션 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.

시뮬레이션에 사용된 물리 수치는 다음과 같이 설정하였다. 프레임과 모터는 실제 수치를 적용하였고, 모듈의 탄성, 마찰, 미끄러짐은 실제와 최대한 유사한 수치로 설정하였다.

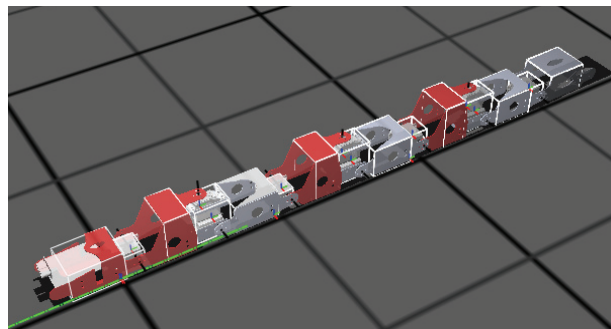


그림 6. 시뮬레이션상의 뱀형 로봇의 모습

Fig. 6. The shape of snake-like robot in simulation

5.2 적합도와 파라미터

이동 제어의 성능을 측정하는 척도는 이동 거리를 기본으로 하였고, 직진성을 우선하기 위해서 좌우로 벗어난 거리(z)를 벌칙으로 두었다. 실험에 사용된 적합도 함수는 다음 식 (4)와 같이 구성되었다.

$$fitness = (0.9 \times x - 0.4 \times |z|) \quad (4)$$

GA 프로그램으로는 GAttr과 GApS 모두에 SGA[10]가 사용되었다. 실험에 이용된 파라미터는 다음과 같다.

Number of generations : 50
 Population sizes : 100
 Crossover : 0.9
 Mutation : 0.1
 Selection : Roulette Wheel

5.3 실험 결과

KMC 뱀로봇에 대한 시뮬레이션 과정은 다음과 같다. Webots 시뮬레이션 코드는 로봇 모델에 대한 컨트롤러와 GA 진화연산 부분으로 구성된다.

GA 연산을 통해 생성된 각 개체들은 모듈의 진폭, 주기, 위상 등에 관한 파라미터들을 포함하며, 이를 시뮬레이션 환경의 로봇의 컨트롤러에 전달하여 ODE 기반의 동작실험을 수행한다. 시뮬레이션 종료 후 얻은 로봇의 이동거리 정보로 적합도를 계산하고, 로봇을 원래 위치로 초기화 시킨다. 모든 개체에 대해서 위의 시뮬레이션 과정을 반복하고, 다음 세대로 넘어간다.

적합도와 속도는 대략적으로 비례하나 적합도가 큰 것은 직진 이동 거리가 긴 것을 의미한다. 실험은 뱀 로봇의 수직 이동 방식을 이용하였고, 성능지표는 정의된 적합도와 이동 속도(cm/s)로 측정하였다.

5.3.1 전체 모듈 사용

실험 먼저 위상제어방식(GApS)과 임의의제어(GAttr)

기법으로 전체 모듈을 사용하는 일반적인 이동움직임을 생성하였다. 생성된 궤적에 대해 한 사이클당 40개의 보간점을 사용하였다. 각각 10회 이상씩을 반복 실험하여, 구해진 해의 시뮬레이션 재연(replay)중 쓰러지거나 제자리걸음을 하는 등의 문제가 있는 해를 제외하고 상위 5회의 결과를 평균하였다.

진화연산에서 구한 해는 시뮬레이터에서 재연(replay)시마다 100% 동일한 환경이 아니므로 약간씩 차이가 난다. 따라서 각 실험 마다 보다 정확한 결과를 얻기 위해서 구해진 해를 5회 반복 재연하여 얻어진 평균값을 한 회의 실험 수치로 사용하였다.

표 1. 제어방법별 적합도 및 속도값 비교

Table 1. Comparison of each methods for fitness and velocity

제어방식	GAps	GAtr
적합도	2.358	4.743
속도	7.861	15.81
속도표준편차	0.264	0.418

표 1에 나타난 시뮬레이션 결과를 보면, 적합도와 속도 모두에서 임의궤적 생성방식(GAtr)이 위상생성(GAps) 방식보다 우수한 결과를 보였다.

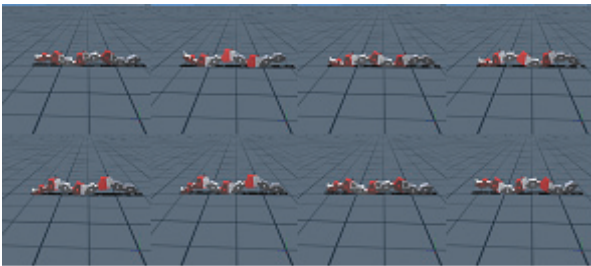


그림 7. KMC 기본 제어방식의 이동움직임
Fig. 7. Locomotion of the basic control of KMC robot

그림 7은 KMC 뱀형 로봇에 내장되어 있는 기본적인 제어방식에 의한 이동 움직임을 나타내고, 그림 8에는 위상 파라미터만을 진화연산으로 탐색한 GAps 방법에 대한 이동 움직임이 나와있다. 그림 9에는 GAtr로 전체 3가지 파라미터를 탐색한 방식에 의해서 얻어진 이동 움직임이 나와있다. 이동 움직임을 분석해보면, KMC의 기본 움직임은 일반적인 애벌레의 움직임과 비슷한 모습을 보이며 움직임의 최대 높이가 낮은 것을 볼 수 있다.

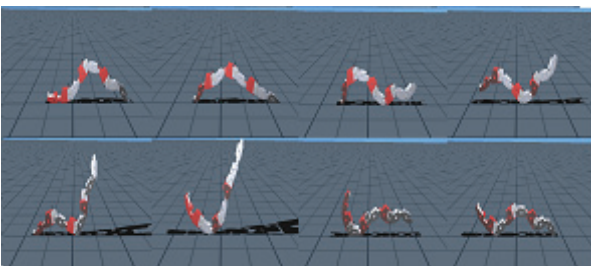


그림 8. GAps 방식의 이동움직임(최우수개체, 7 모듈)
Fig. 8. Locomotion of GAps method (the best individual, 7 Modules)

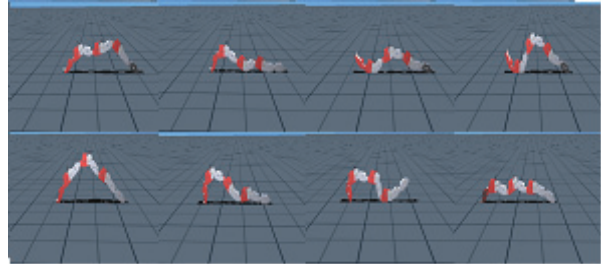


그림 9. GAtr 방식의 이동움직임(최우수개체, 7 모듈)
Fig. 9. Locomotion of GAtr method (the best individual, 7 Modules)

진화 연산을 사용한 두 가지의 움직임을 살펴보면 모든 모듈을 사용하여 움직이고, 로봇의 움직임이 사인파의 형태와 비슷하게 이동하는 움직임을 나타낸다. 하지만, GAps 방식은 좀 더 굴러가는 듯한 움직임을 보이고 GAtr 방식은 머리와 중간 모듈을 작용점으로 하여 기어가는 움직임을 보인다. 이는 임의궤적생성 방식이 좀 더 다양한 해 공간을 탐색하여 정형적인 움직임을 벗어난 해를 발견했음을 나타낸다고 볼 수 있다.

5.3.2 부분 모듈 사용

두 번째로 부분 모듈만을 사용한 실험이 다음과 같이 수행되었다. 모듈의 선택 및 각 모듈의 임의 궤적 생성을 위해서 역시 진화연산 GA가 적용되었다. 4-6개의 부분 모듈 사용에 대한 실험 결과가 표 2에 나와 있다. 각 모듈수 별 결과를 파악하기 쉽게 전체 7개의 모듈을 사용한 결과를 함께 넣었다.

표 2의 결과를 보면, 일반적으로 많은 모듈을 사용한 개체들이 더욱 멀리까지 이동함을 나타낸다. 하지만, 효율 면에서는 모듈을 6개만 사용하는 것이 전체 모듈 7개를 사용하는 것보다 5% 정도 효율이 좋은 결과를 나타낸다. 그림 10-12에는 4-6개의 모듈을 선택했을 경우, 각 모듈수당 탐색된 최우수 개체의 이동 움직임을 보여준다. 그림 10은 4개의 모듈의 경우로서, 이동시 2-5번째의 모듈만을 사용하였고 로봇이 나타내는 사인파의 움직임이 매우 작고 느리다. 5개의 모듈의 경우에는(그림 11) 1-5번째의 모듈을 사용하였고, 움직임이 4모듈만을 사용한 경우에(그림 10) 비해 매우 커진 것을 확인할 수 있다. 움직임에 있어서는 로봇 전체의 모듈이 사인파에 유사한 형태를 보이며 이동한다. 6모듈을 사용한 경우가 그림 12에 나와 있으며, 4모듈과 5모듈의 중간정도에 해당하는 움직임을 나타내고 있다.

표 2. 사용 모듈수별 적합도 및 속도 비교

Table 2. Comparison among different modules for fitness and velocity

사용 모듈수	4모듈	5모듈	6모듈	7모듈
평균 적합도	2.243	3.067	4.291	4.743
평균 속도	7.478	10.223	14.305	15.81
모듈당 적합도	0.56	0.613	0.715	0.677
모듈당 속도	1.869	2.194	2.384	2.258
모듈당 속도표준편차	0.184	0.4	0.252	0.418

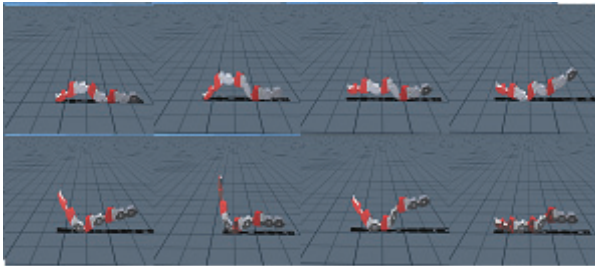


그림 10. 부분 모듈 이동 움직임(GAttr 방식, 4개 모듈)
Fig. 10. Locomotion of partial modules(GAttr method, 4 Modules)

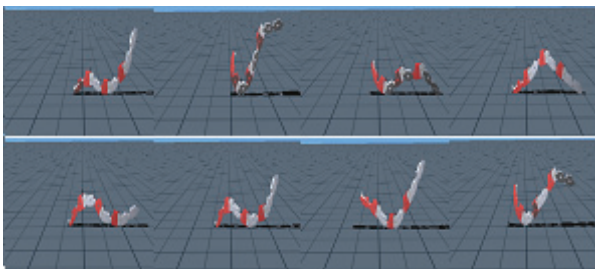


그림 11. 부분 모듈 이동 움직임(GAttr 방식, 5개 모듈)
Fig. 11. Locomotion of partial modules(GAttr method, 5 Modules)

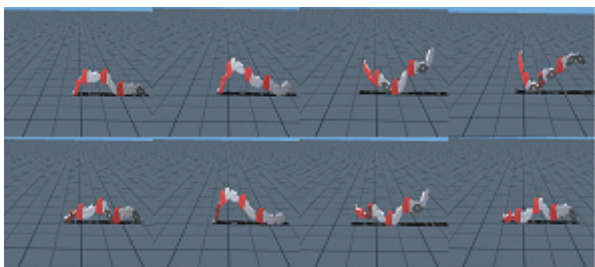


그림 12. 부분 모듈 이동 움직임(GAttr 방식, 6개 모듈)
Fig. 12. Locomotion of partial modules(GAttr method, 6 Modules)

표 3. 선택된 모듈 패턴 (적합도 순)
Table 3. Patterns of selected modules (descending order by fitness)

	1	2	3	4	5	6	7	fit
4모듈	x	o	o	o	o	x	x	2.462
	x	o	o	o	o	x	x	2.4
	o	o	o	o	x	x	x	2.201
	o	o	o	o	x	x	x	2.139
5모듈	x	o	o	o	x	x	o	2.016
	o	o	o	o	o	x	x	3.431
	o	o	o	o	o	x	x	3.324
	o	o	o	o	o	x	x	3.292
6모듈	o	o	o	o	x	o	x	3.242
	o	o	o	x	o	o	x	2.579
	o	o	o	o	o	o	x	4.585
	o	o	o	o	o	o	x	4.543
7모듈	o	o	o	o	o	o	x	4.163
	o	o	o	o	o	o	x	4.112
	o	o	o	o	o	o	x	4.055
	o	o	o	o	o	x	o	4.055

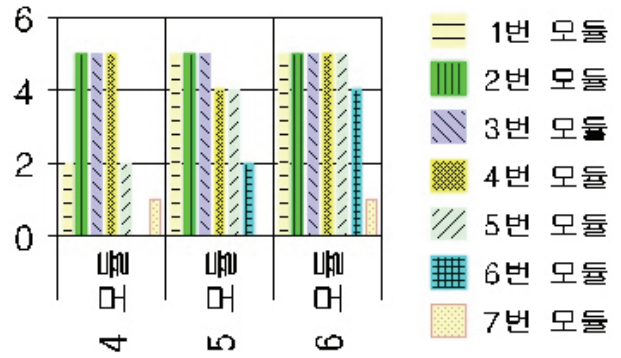


그림 13. 부분 모듈 사용 분포.
Fig. 13. Frequencies of each modules by selection of partial modules.

표 3에는 각 경우에 사용된 모듈 패턴을 보여주고, 그림 13은 반복실험동안 모듈이 사용된 빈도를 나타낸다. 선택적인 모듈 사용에 있어서 높은 적합도를 가진 개체들은 모두 연속된 모듈을 사용한 결과를 보였고, 2, 3, 4번 모듈들은 거의 모든 실험에서 사용되는 결과를 나타냈다.

6. 실제 로봇에 대한 적용

실제 로봇에 대한 적용에서 이동결과가 시뮬레이션과는 차이를 보였는데, 이는 사용된 마찰력, 탄성 등의 수치가 실제와 다른 것을 사료된다. 기본적으로 시뮬레이션에서 얻어진 해의 관절값을 따라서 움직이는 모습을 보였지만, 시뮬레이션에 비하여 속도면에서 느리고, 움직임의 높이면에서도 낮은 움직임을 보였다. 그림 14에는 가장 우수한 GAttr 방식을 통해서 얻어진 해를(그림 9) KMC 뱀형 로봇에 실제 적용한 결과의 움직임이 나타나 있다.



그림 14. GAttr 방식을 사용한 실제 KMC 로봇의 움직임.
Fig. 14. Locomotion of Real KMC Robot using GAttr method

7. 결 론

본 논문에서는 KMC의 뱀형 로봇 KOBi를 모델링하고, 시뮬레이션을 통해 진화연산을 사용한 두 가지 이동 제어 방식을 비교하였다. 또한, 전체 모듈중 일부 모듈만을 사용한 경우에 대해서 모듈 선택의 영향을 비교 분석하였다.

성능면에서, 평균적으로 GAttr에 의한 생성해의 적합도가 GAs보다 2배(200%) 좋았고, GAs도 기본적인 KMC 로봇의 기본 움직임에 비하여 약 2배(230%)정도 좋은 적합도를 나타냈다. 움직임에서는 진화연산을 사용한 제어방식에

서 모두 사인파와 유사한 형태의 움직임은 보였지만, GAtr를 사용한 개체가 GAp를 사용한 개체보다 좀 더 다양하고 평균적으로 큰 움직임을 보였다.

일부 모듈만을 선택 사용하는 문제에서는, 모든 경우에서 대부분이(약87%) 인접한 모듈을 사용하는 개체들이 해로 선택되었다. 뱀형 로봇의 움직임에 있어서, 많은 모듈을 사용할수록 먼 거리를 이동하는 모습을 나타냈지만, 효율면에서는 6개의 모듈만을 사용한 경우가 모듈당 이동거리에서 제일 좋은 결과를 나타냈다. 또한, 4개의 모듈을 사용한 경우는 다른 경우에 비하여 낮은 효율을 보였다. 이는 뱀형 로봇이 이동하는데 있어서 최소한 5개의 모듈을 사용하는 것이 직진이동에 적합한 것으로 생각된다.

향후, 본 논문에서 분석된 결과를 기반으로 하여 실제 KMC 로봇에 대한 실제 실험의 확장과 시뮬레이션과 실제 실험과의 차이를 줄이는 기법, 이동에 대한 다양한 적합도 함수, 개선된 탐색 기법 및 새로운 제어 방법, 그리고 수직과 수평관절을 모두 사용하는 뱀형 로봇의 이동연구등이 필요하다.

참고 문헌

[1] G. S. Hornby, H. Lipson, J. B. Pollack, "Generative Representations for the Automated Design of Modular Physical Robots", *IEEE Trans. on Robotics and Automation*, vol. 19, no. 4, pp. 703-719, 2003.

[2] A. L. Nelson, G. J. Barlow, L. Doitsidis, "Fitness functions in Evolutionary Robotics: A Survey and Analysis", *Robotics and Autonomous Systems*, 57, pp 345-370, 2009

[3] V. Zykov, E. Mytilinaios, M. Desnoyer, H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics", *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 308-319, 2007.

[4] 장재영, 현수환, 서기성, "모듈라 뱀형 로봇의 이동 제어에 대한 진화연산 기법 비교", *제어·로봇·시스템학회 논문지*, 제 15권 제6호 pp. 604-611, 2009. 6

[5] A. Crespi, A. J. Ijspeert, "AmphiBot II : An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator", *In Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, pp. 19-27, Brussels, Belgium, September 2006.

[6] K. J. Dowling, *Limbless Locomotion: Learning to Crawl with a Snake Robot*, in his Ph.D Thesis, Robotics Institute, Carnegie Mellon University 1997.

[7] Mark W. Sherman, "Sine-Wave Locomotion in a Robotic Snake Model Form and Programming", *In Proceedings of AAI Mobile Robot Competition: Papers from the AAI Workshop*, pp. 34-40, American Association for Artificial Intelligence, Menlo Park, California, 2002.

[8] I. Tanev, "Genetic Programming Incorporation Biased Mutation for Evolution and Adaptation of Snakebot", *Genetic Programming and Evolvable Machines*, vol. 8, no. 1, pp. 39-59, 2007.

[9] I. Tanev, T. Ray, K. Shimohara, "On the

Analogy in the Emergent Properties of Evolved Locomotion Gaits of Simulated Snakebot", *Mobile Robots Toward New Applications*, ch. 19, pp. 559-578, Germany, December 2006.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[11] L. Hohl, R. Tellez, O. Michel, A. J. Ijspeert, "Aibo and Webots: Simulation, wireless remote control and controller transfer", *Robotics and Autonomous systems*, vol. 54, pp. 472-485, 2006.

[12] *KMC 코비 매뉴얼*, <http://www.kmicrobot.com/kr>

저자 소개

안인석(Ihn-Seok Ahn)



1987년 : 연세대학교 전기공학과 공학사
 1989년 : 동 대학원 전기공학과 공학석사
 1997년 : 동 대학원 전기공학과 공학박사
 1989년~1999년 : 삼성전자(주) 선임연구원
 1999년~현재 : 위덕대학교 에너지전기공학부 부교수

관심분야 : 로봇틱스, 지능제어, 유전알고리즘
 E-mail : isahn@uu.ac.kr

장재영(Jaeyoung Jang)



2003~현재 : 서경대학교 전자공학과 학사 과정

관심분야 : 지능로봇, 진화연산
 E-mail : geruld@naver.com

서기성(Kisung Seo)



1986년 : 연세대학교 전기공학과 공학사
 1988년 : 연세대학교 전기공학과 공학석사
 1993년 : 연세대학교 전기공학과 공학박사
 1993~1998년 : 서경대학교 산업공학과 조교수

1999~2003년 : Michigan State University, Genetic Algorithms Research and Applications Group, Research Associate
 2003~현재 : 서경대학교 전자공학과 부교수

관심분야 : 지능로봇, 진화 연산(GA, GP, ES), 진화 디자인
 E-mail : ksseo@skuniv.ac.kr