

뱀형 로봇의 모델링과 진화연산 기반 이동 생성

서기성*, 장재영*, 안인석**

*서경대학교 전자공학과, **위덕대학교 에너지전기공학부

Modeling of Snake-like Robot and Evolutionary Computation based Generation of Locomotion

Kisung Seo*, Jaeyoung Jang*, In-Seok Ahn**

*Dept. of Electronic Engineering, Seokyeong University, **Div. of Energy & Electrical Engineering, Uiduk University

Abstract - 뱀형 모듈라 로봇은 다양한 환경에 대해서 강인성을 가지고 있고, 모듈 일부의 고장에도 이동할수 있는 장점을 가진다. 그러나 이동 제어 방법이 어렵고, 아직까지 효율적인 이동법의 개발이 미비한 편이다. 본 연구에서는 뱀형 로봇의 이동제어를 위하여 GA(Genetic Algorithm)기반의 위상제어 방식과 GP(Genetic Programming)를 사용한 임의의 관절계적 생성 방식을 비교한다. KMC사의 뱀형 로봇을 대상으로 먼저 webots 시뮬레이터 상에서 모델링 및 시뮬레이션을 수행하여 위의 방법들을 비교하였다.

1. 서론

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]. 모듈라 로봇은 독립적인 단위 구성 요소인 각 모듈을 원하는 기능을 수행할 수 있도록 특정한 형태로 연결하며 만든 로봇으로서[1,2], 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 고장난 상태에서도 동작이 가능하고, 계단 등의 장애물이 있는 지형 혹은 불안정한 환경에서도 유연하게 동작할 수 있다. 그러나 기존의 로봇들에 비하여 이동속도가 느리고 제어가 어렵다는 단점이 있다[3,4].

기존의 모듈라 뱀형 로봇의 이동 제어에 사용된 방식으로는 삼각함수로 표현된 식을 모듈마다 주기가 다르게 적용하여 관절의 움직임을 제어하는 방식이 있다[5]. 그리고 CPG(Central Pattern Generator)를 이용해 걸음걸이를 생성하는 방식도 있다[3]. 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 진화연산 기법을[5,6] 사용한 연구가 수행되어 왔고, 성과를 보여 주고 있다. 그러나 현재 대부분의 연구들은 가상의 모듈을 사용하거나[4,7] 다양한 진화연산의 비교는 미흡한 편이다.

본 논문에서는 GA 기반의 위상제어 방식, GP 기반의 관절계적 생성 방식을 비교하고자 한다. 대상 로봇은 KMC사의 실제의 뱀형 로봇을 사용하였고, Webots[8]를 사용한 모델링과 시뮬레이션을 수행하였다.

2. 모듈라 뱀형 로봇의 이동

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[4,9], S모양으로 움직이는 측면 파동 방법(Serpentine)[3,9] 등이 있다. 수직적인 관절의 움직임을 가지는 방법은 일반적인 애벌레의 움직임과 같은 Caterpillar 방식, 꼬리부분으로 몸체를 밀어 이동하는 Inching 방식[9] 등이 있다. 그 외에 수평 수직 복합적인 움직임의 방법으로는 Helix Theory 방법[7]이나 Screw 방법[9] 등이 있다.

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 1과 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)[5]와 GP(Genetic Programming)[6]는 진화연산의 한 종류로서, 유전자(chromosome)를 표현하는 방식에 서로 차이가 있다. GA는 비트열이나 실수형태의 염색체를 이용하고, GP는 트리구조의 염색체를 이용한다. 이 차이로 인해 GA는 고정된 크기의 염색체로 구성되며, 주로 수치최적화에 적합하다. 이에 비해 GP는 트리구조를 사용하기 때문에 염색체의 크기가 가변이며, 구조적인 설계문제나 임의의 차수로 구성되는 복잡한 함수를 탐색하는 것에 적합하다.

3.1 GA 기반 위상제어(GAps)

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

$$\sin(t + \phi) \quad (2)$$

$$(-6.28 \leq \phi \leq 6.28)$$

3.2 GP 기반 관절계적 생성

GA 기반 방법이 고정된 구조의 사인함수에 파라미터만을 변화시켜 이동 제어식을 탐색하는데 비하여, GP 방식에서는 각 모듈이 움직이는 관절 계적이 임의의 함수로 구성된다. 즉, 각 관절에 대한 계적 함수가 GP에 의해서 생성, 진화되는 트리를 통해 탐색된다[11].

각 관절의 독립적인 제어를 위해서, 일반적인 GP의 단일트리 방식이 아닌 멀티트리 방식을 이용함으로써 더 효율적인 연산을 피하였다[11]. GP 트리의 구성에 이용된 함수와 터미널은 다음과 같다.

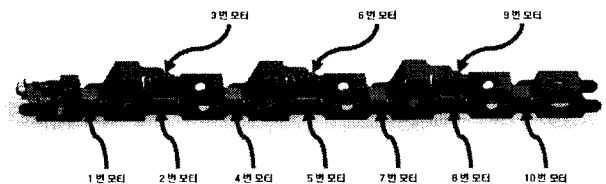
$$F = \{ \sin, \cos, +, -, *, /, \exp, \log \}$$

$$T = \{ X, ERC \}$$

4. 뱀형 로봇의 모델링

4.1 KMC 뱀형 로봇

본 논문의 실험에서 사용된 로봇은 KMC Robotics사의 뱀형 로봇 KMC-KOBI이다[10]. 로봇은 10개의 RC 서보모터로 이루어진 관절과 전원, 8개의 프레임과 CPU인 Atmega8535로 구성되어 있다. 로봇의 무게는 약 11킬로그램(11.1513 Kg)이고, 총 길이는 1미터이다. 관절은 3개의 수평 관절과 7개의 수직 관절로 이루어져 있다.

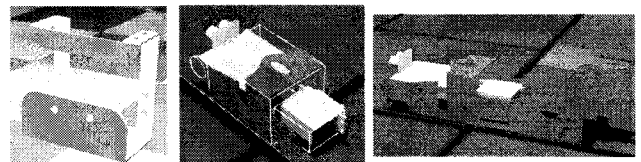


<그림 1> 뱀형 로봇의 모습.

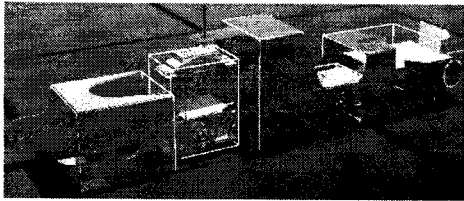
4.2 로봇 모델링

Webots를 이용하여 KMC-KOBI 로봇을 모델링하였으며, 그 과정은 다음과 같은 절차를 가진다. 먼저, Solid works 프로그램을 사용하여, igs 형식의 설계도를 VRML2.0 포맷의 wrl 형식의 파일로 변환한다. Webots의 import 기능을 이용하여 이를 불러들인다. 전체의 설계도를 각 프레임 별로 나누어 모듈의 형태를 구성한 뒤(그림 2) 각 프레임에 서보 및 연결부를 포함시켜 각 모듈을 구성한 다음 이들을 결합하였다(그림 3).

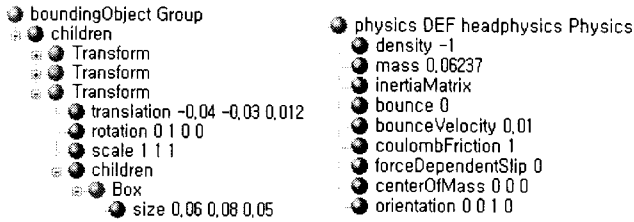
그리고, 각 모듈에 대한 크기, 좌표등의 기하 정보 및 무게, 마찰, 토크등의 물리 정보 수치를 설정한다(그림 4). 최종적으로, 완성된 모델에 대해서 각 모듈에 대한 서보에 목표값을 대입하여 움직임을 정상인가를 확인하였다.



<그림 2> 뱀형 로봇 모듈의 부분 모델링



<그림 3> 개별 모듈의 결합



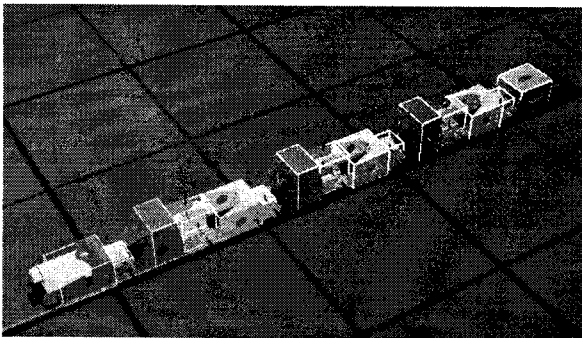
<그림 4> 기하 및 물리정보의 설정

5. 컴퓨터 시뮬레이션

5.1 Webots 시뮬레이션 환경

진화연산에 의한 이동방식을 생성하기 위해서는 반복적인 시뮬레이션 수행이 필요하다. 시뮬레이션 환경은 Cyberbotics사의 Webots[8]을 사용하였다. Webots은 다양한 로봇에 대한 모델링, 프로그래밍, 그리고 시뮬레이션 기능을 제공하는 로봇 시뮬레이션 S/W 이다. 각종 센서와 액츄에이터에 대한 라이브러리 제공, 정확한 Physics 시뮬레이션을 위한 ODE(Open Dynamics Engine) 라이브러리 제공, 그리고 실제 로봇에 컨트롤러를 트랜스퍼할 수 있는 기능을 가지고 있다.

그림 5에 Webots 상에서의 KMC-KOBI 로봇의 모습이 나와 있다. 시뮬레이션에 사용된 물리 수치는 다음과 같이 설정하였다. 프레임과 모터는 실제 사양에 주어진 수치를 적용하였고, 모듈의 탄성, 마찰, 미끄러짐은 실제와 최대한 유사한 수치를 사용하였다.



<그림 5> 시뮬레이션상의 뱀형 로봇의 모습.

5.2 적합도와 진화연산 파라미터

이동 제어의 성능을 측정하는 척도는 이동 거리를 기본으로 하였고, 직진성을 우선하기 위해서 벗어난 거리(z)를 벌칙으로 두었다. 실험에 사용된 적합도 함수는 다음 식 (3)와 같이 구성되었다.

$$fitness = (0.9 \times x - 0.4 \times |z|) \quad (3)$$

GA 프로그램으로는 SGA[5]가 사용되었고, GP 엔진으로 lil-gp[12]가 사용되었다. GA와 GP에 공통으로 이용된 파라미터는 다음과 같다.

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

GP의 초기 트리 생성에 대한 파라미터는 다음과 같다.

Initial population : half_and_half
 Initial depth : 1-3
 Max depth : 12

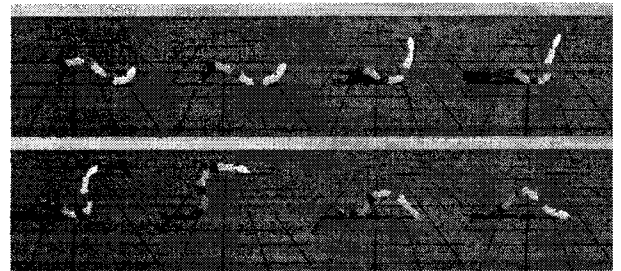
5.3 실험 결과

각각의 생성된 함수에 대해 GA, GP 모두 한 사이클당 40개의 보간점을 이용하였다. 각각 5회씩 반복 실험하였으며, 적합도가 클수록 단위시간당 이동 거리가 긴 것을 의미한다. 실험은 뱀 로봇의 수직 이동 방식을 이용하였고, 성능지표는 정의된 적합도와 속도(cm/s)로 측정하였다. 표 1에 나타난 시뮬레이션 결과를 보면, GP에 의한 임의의 관절 제작성 방식이 위상제어 방식인 Gaps 보다 적합도와 속도에서 더 우수함을 나타낸다.

<표 1> 적합도 및 속도값 비교

	Gaps	GP
적합도	1.75	2.5
속도	28.03	39.1

그림 6과 그림 7에 두가지 방법에 의해 얻어진 이동 움직임의 모습이 나와 있다. 이동 움직임을 분석해보면, Gaps의 경우 모든 관절을 사용하여 움직이며, 로봇의 움직임도 사인파와 비슷한 형태로 이동하는 형태를 보인다. GP의 경우는 머리와 꼬리의 프레임은 작용점으로 하여 몇 개의 관절만을 주로 사용하여 이동을 하는 모습을 보인다. 이는 정형화된 곡선의 움직임을 벗어난 결과로 탐색공간이 열린 GP 기법의 특징을 보여주고 있다.



<그림 6> Gaps 제어를 사용한 로봇의 움직임



<그림 7> GP 제어를 사용한 로봇의 움직임

6. 결론

본 논문에서는 KMC의 뱀형로봇 KOBI를 모델링하고, 시뮬레이션을 통해 진화연산을 사용한 이동 제어 방식을 생성하였다. 평균적으로 GP에 의한 생성해가 GA보다 적합도와 속도 면에서 우수하였으나, 움직임들이 정형화된 사인 곡선의 움직임을 벗어난 결과를 보였다.

추후, 실제 로봇에의 실험 및 다양한 적합도 함수와 개선된 탐색 기법 등에 대한 연구가 필요하다고 사료된다.

감사의 글 : 본 논문은 2007년 정부재원(교육인적자원부 학술연구 조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음. (KRF-2007-314-D00176)

[참 고 문 헌]

- [1] A. L. Nelson, G. J. Barlow, L. Doitsidis, "Fitness functions in Evolutionary Robotics: A Survey and Analysis", Robotics and Autonomous Systems, 57(2009) pp 345-370
- [2] 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.
- [3] 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.
- [4] I. Tanev, "Genetic Programming Incorporation Biased Mutation

- ng and Evolvable Machines, vol. 8, no. 1, pp. 39-59, 2007.
- [5] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
- [6] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, 1992.
- [7] I. Tanev, T. Ray, K. Shimohara, "On the Analogy in the Emergent Properties of Evolved Locomotion Gaits of Simulated Snake bot", Mobile Robots Toward New Applications, ch. 19, pp. 559-578, Germany, December 2006.
- [8] 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.
- [9] K. J. Dowling, Limbless Locomotion: Learning to Crawl with a Snake Robot, in his Ph.D Thesis, Robotics Institute, Carnegie Mellon University 1997.
- [10] (주)케이엠씨 로보틱스, <http://www.kmrobot.com/kr>
- [11] 서기성, 현수환, "관절 공간에서의 GP 기반 진화기법을 이용한 4족 보행로봇의 걸음새 자동생성", 제어·로봇·시스템학회 논문지, 제 14권 6호, pp. 573-579, 2008. 6.
- [12] D. Zongker, B. Punch, Lil-GP User's Manual, Michigan State University, July 1995.