

경로 제어가 가능한 가상생명체를 위한 2단계 진화 알고리즘

(Two-Stage Evolutionary Algorithm for Path-Controllable Virtual Creatures)

심윤식[†] 김창현^{**}

(Yoon-Sik Shim) (Chang-Hun Kim)

요약 본 논문은 사용자의 제어가 가능한 3차원 물리 기반 가상생명체를 생성하는 2단계 진화 시스템을 제안한다. 기존의 방법은 가상생명체의 형상과 기동, 그리고 목표지점추적(target-following)과 같은 상위 레벨의 행위를 한꺼번에 하나의 진화 시스템으로 생성해냄으로 인하여 진화 단계에서의 사용자의 개입을 허용하지 않았다. 본 논문은 하나로 묶여있던 시스템을 다루기 용이한 두 개의 서브시스템으로 분리함으로써 사용자의 개입을 허용한다. 첫 번째 단계로 가상생명체의 몸체와 직진 기동을 위한 하위 레벨 모터 컨트롤러가 진화 알고리즘(evolutionary algorithm)으로 동시에 생성된다. 두 번째 단계에는 생성된 기본 생명체 위에 주어진 경로를 따라가기 위한 상위 레벨 컨트롤러가 인공 신경망을 사용하여 탑재된다. 경로제어(path-following)를 위한 신경망의 연결 가중치는 유전자 알고리즘(genetic algorithm)을 사용하여 최적화되며 한번 진화된 신경망 컨트롤러는 어떠한 임의의 경로도 잘 따라감을 보여준다. 이로써 사용자는 모든 진화과정이 끝나지 않고도 중간단계에서 기호에 맞는 생명체를 끌라내거나 버릴 수 있으며, 동일한 기본 생명체 위에 또 다른 형태의 상위레벨 행위를 생성하는 것도 가능해진다. 본 논문은 이러한 2단계 알고리즘과 함께 직진기동을 위한 새로운 분절 삼각 함수(piecewise sinusoidal) 컨트롤러를 제안하고 마개 실린더(capped-cylinder)를 기본 요소로 하는 가상생명체에 대한 효율적인 실시간 수증역학 모델링 기법도 함께 소개한다.

키워드 : 컴퓨터 애니메이션, 물리기반 시뮬레이션, 인공생명, body-brain 공진화, 진화 알고리즘, 인공 신경망

Abstract We present a two-step evolution system that produces controllable virtual creatures in physically simulated 3D environment. Previous evolutionary methods for virtual creatures did not allow any user intervention during evolution process, because they generated a creature's shape, locomotion, and high-level behaviors such as target-following and obstacle avoidance simultaneously by one-time evolution process. In this work, we divide a single system into manageable two sub-systems, and this more likely allows user interaction. In the first stage, a body structure and low-level motor controllers of a creature for straight movement are generated by an evolutionary algorithm. Next, a high-level control to follow a given path is achieved by a neural network. The connection weights of the neural network are optimized by a genetic algorithm. The evolved controller could follow any given path fairly well. Moreover, users can choose or abort creatures according to their taste before the entire evolution process is finished. This paper also presents a new sinusoidal controller and a simplified hydrodynamics model for a capped-cylinder, which is the basic body primitive of a creature.

Key words : computer animation, physically-based simulation, artificial life, body-brain coevolution, evolutionary algorithm, neural network

· 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음
(KRF-2003-041-D20461)

† 학생회원 : 고려대학교 컴퓨터학과
necromax@korea.ac.kr

** 종신회원 : 고려대학교 컴퓨터학과 교수
chkim@korea.ac.kr

논문접수 : 2005년 6월 13일

심사완료 : 2005년 9월 20일

1. 서론

컴퓨터의 계산능력이 점점 강력해지면서, 가상세계의 풍부함을 제한하는 요소는 점차 컴퓨터의 성능보다는 사람의 디자인 능력으로 옮겨가고 있다. 컴퓨터 그래픽스에 있어서 가상생명체는 특수효과와 배경캐릭터로서

점차적으로 중요한 역할을 맡고 있다. 이러한 가상생명체의 인공적 진화는 잠재적으로 그것들의 형태와 행위를 지정해주어야 하는 어렵고 시간 소모적인 일을 다소 완화시켜준다. 인공생명의 기술 중 하나인 인공진화는 복잡한 가상의 객체들을 '세부요소의 자세한 이해 없이도' 생성할 수 있도록 하여 준다. 그들 중의 하나의 형태로 제시된 진화 알고리즘(EA)은 자연진화로부터 영감을 받은 기술이며, 가상세계의 캐릭터뿐만 아니라 실제 로봇의 움직임을 생성하는 데 있어서도 뛰어난 가능성을 보여왔다[1,2].

진화는 초기 디자인의 영향을 초월하는 복잡성을 생성할 수 있도록 해주는 강력한 도구이다. '공진화(Co-Evolution)'란 둘 이상의 종이나 개체군 사이에서 상호적으로 유발되는 진화론적인 변화라고 정의될 수 있다. 따라서 body-brain 공진화란 진화적 변화가 한 생명체의 뇌와 신체 사이에서 작용한다고 볼 수 있다. 물고기의 꼬리의 움직임 패턴을 제어하는 뇌의 부분과 유전형인 형상은 서로 공진화를 했다고 볼 수 있다. 지금까지 인공생명분야에서는 생명 형태를 진화 시키는 데 초점을 맞춘 많은 연구들이 있어왔다. Karl Sims [3,4]는 3차원 가상생명체의 형태와 행위 두 가지를 동시에 진화 시키는 흥미로운 접근 방법을 개척하였다. 그가 만들어낸 "블러키즈(blockies)" 생명체들은 다양한 형태와 아름답고 유기적인 움직임을 보여주었다. 이후로 Sexual Swimmers[5], Framstick[6], Aesthetic creatures[7], GOLEM[8] 등 최근의 많은 연구들이 그들 자신의 방식대로 수준에서 헤엄치거나 땅위에서 기어다니는 다양한 막대모양의 생명체들의 형태와 움직임을 보여주었다.

비록 진화 알고리즘이 흥미롭고 유기적인 모습과 움직임을 보여주는 에이전트들을 만들 수 있도록 하여 주지만, 진화된 가상생명체들의 내부 컨트롤러의 메커니즘을 이해한다는 것은 사실상 불가능하다. 이는 인공생명의 연구목적에 위한 가상생명체의 매우 복잡한 컨트롤러 모델링에서 기인하는데 대부분 임의로 연결된 임의의 개수의 뉴런을 사용한 컨트롤러(augmented neural network)를 사용하기 때문이다. 이러한 컨트롤러는 인공진화의 연구에는 적합하지만 컴퓨터 그래픽스 분야로 도입하기에는 그 복잡함과 모호함으로 인하여 많은 어려움이 있다. 특히 그래픽스에서 꼭 필요한 경로제어와 같은 상위레벨의 유저 컨트롤을 적용하기가 매우 어려워진다. 게다가 진화 알고리즘의 탐색공간이 매우 크고, 원하는 기능에 대한 내부적인 구별이 모호하기 때문에 사용자의 입장에서 보기 좋은 형상과 효율적이고 제어 가능한 움직임의 두 가지를 동시에 얻는다는 것은 사실상 불가능하다. 풍부한 가상 환경을 위한 가상생명

체를 진화 시키기 위해서는 필요한 모든 정보를 유전형에 함축시켜야 하며 단일 구조의 유전형 자료구조와 규칙적인 생명체 변환 규칙을 가져야 한다. 그러면서도 충분한 개체의 다양성을 보장해야 하며 동시에 후보 해가 존재하는 탐색공간의 불필요한 증식을 피해야 한다. 이러한 문제점을 보장하기 위하여 사용자가 부분적으로 유전형을 직접 손으로 디자인하여 접근하거나[6], 진화 과정에 있어서 매 세대마다 사람의 심미적인 기준을 적용하여 직접 선택과정에 관여하는 등의 방법[7]이 제안되었으나 이러한 방법들은 사용자의 노력을 너무 많이 요구하게 되므로 사실상 자동적인 가상생명체 생성 시스템이라고 하기 어렵다. 다른 방법으로, 개별적인 행위 모듈(직진, 좌로 돌기, 우로 돌기 등)을 각각 하나씩 생성한 후, 이들을 상황에 따라 적절히 블렌딩하여 전체적인 상위레벨의 행위를 조합해 내는 방법[9]이 제시되었으나 이 또한 개별적인 행위를 모두 생성시켜줘야 하는 복잡함이 있으며 가지고 있는 행동모듈에 없는 행위를 자연적으로 창발시키지 못한다는 제한이 있다.

이러한 견지에서 본 연구에서는 위의 조건을 만족하면서도 단순하고 구현이 쉬우며 사용자의 개입을 좀 더 편하게 하기 위하여 단일의 진화 시스템을 두 개의 다루기 용이한 시스템으로 나눈 새로운 진화 프레임워크를 소개한다. 본 논문은 가상 생명체의 효율적인 움직임을 위한 직관적인 모터 컨트롤러 모델을 제안하고, 경로 제어를 위한 효율적인 신경망 진화 시스템을 기술한다. 추가로, 빠른 시뮬레이션을 위한 단순화된 수중역학 모델을 소개한다.

2장에서는 전체적인 시스템 개요를 살펴보고, 3, 4장에서는 형태적 대칭성과 함께 더욱 복잡한 컨트롤 네트워크를 가지면서도 유사성을 잃지 않도록 하기 위한 유전형 구조와 그에 따른 유전형-생명체 변환 시스템을 기술한다. 5장에서는 역학 시뮬레이션, 6, 7장에서는 두 단계의 진화 프로세스를 설명 한 후 마지막으로 결론 및 논의로 글을 마치도록 하겠다.

2. 시스템 개요

전체 시스템은 두 단계로 나누어진다. 그림 1에서 볼 수 있는 바와 같이, 첫 번째 단계로서, 가상생명체 진화 시스템이 생명체의 몸체와 직진기동을 위한 하위레벨의 모터 컨트롤러를 생성한다(그림 1, 점선 박스). 기본적인 생명체가 생성된 후에, 신경망이 하위레벨모터 컨트롤러에 연결되며, 주어진 길에 대하여 신경망의 연결가중치가 최적화된다.

신경망을 위한 개체군으로는 연결 가중치로 된 벡터가 쓰여진다. 매 시간 스텝마다 생명체의 질량중심의 위치와 속도, 그리고 목표 지점이 입력되며 목표지점의 계

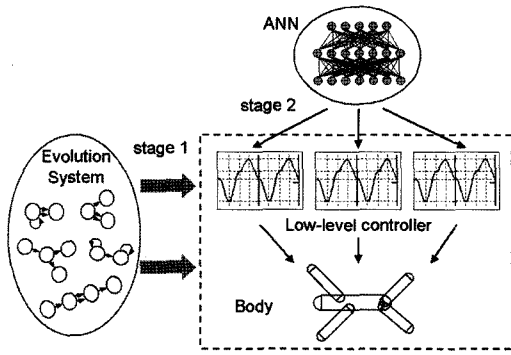


그림 1 시스템 개요도

산은 생명체의 속도와 주어진 경로의 함수로 실시간으로 계산된다. 네트워크로부터의 출력은 생명체의 모터 컨트롤러로 주어지며 각 관절의 움직임을 변화시키고 따라서 진행 방향을 바꾸게 된다.

이러한 개념은 직관적으로 이해할 수 있는데, 실제 척추동물과 마찬가지로 가상생명체 시스템도 몸체와 소뇌의 모터뉴런(low-level motor controller), 그리고 그것을 상위에서 제어하는 대뇌(neural network)의 세가지 부분과 비교하여 이해할 수 있다.

3. 유전형의 표현과 생명체의 발현

유전형 인코딩은 Sim[3]와 같은 방향그래프(그림 2)로 이루어진다. 그러나 내부의 임의의 신경망 기반 컨트롤러는 포함되지 않는다. Sims의 구조와 또 하나의 다른 점은 박스모양의 몸체 대신 마개실린더(capped-cylinder)를 사용한다는 것이다. 실린더 형태는 수중역학 계산시에 박스보다 더 수월하고 간단하게 사용되어질 수 있기 때문이다. 각 실린더는 부모와 자식 실린더를 연결하는 조인트를 가지고 있다. 실린더에는 각각 실

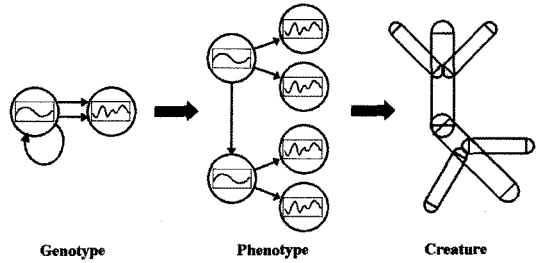


그림 2 유전형 인코딩. 유전형(Genotype)의 정보를 표현 규칙에 따라 각각의 노드와 연결을 복사하여 표현형(Phenotype)을 만든다. 만들어진 표현형의 정보를 사용하여 다시 실제적인 물리적 몸체 구조를 만들게 된다.

린더의 길이 방향으로 시작지점과 끝 지점이 명시되어 있다.

시작점은 0, 끝점은 1로서 지정되는데 방향그래프의 연결선 정보 안에 두 개의 실린더가 이 위치를 기반으로 어떻게 연결되어야 하는지에 대한 정보를 담고 있다. 그래프 연결선의 파라미터인 결합위치는 자식 실린더의 시작지점이 부모 실린더의 어디에 붙어야 하는가를 나타내며 0에서 1사이의 값을 갖는다.

부모 실린더에 대한 자식 실린더의 오리엔테이션은 역시 그래프 연결선의 정보인 θ, ϕ, δ 로 나타내어진다. 두 실린더를 연결하는 과정은 그림 3에 나타나 있다. 연결된 실린더쌍의 초기 각도는 각각 다관절 시뮬레이션에서의 0도로 설정되며 시뮬레이션시에 시간에 따라 변하게 된다. 그래프 연결선에는 이외에 자식 실린더의 길이와 반지름 방향으로의 확대 또는 축소값을 나타내는 변수들이 포함되어 임의의 크기를 갖도록 한다.

4. 모터 컨트롤러

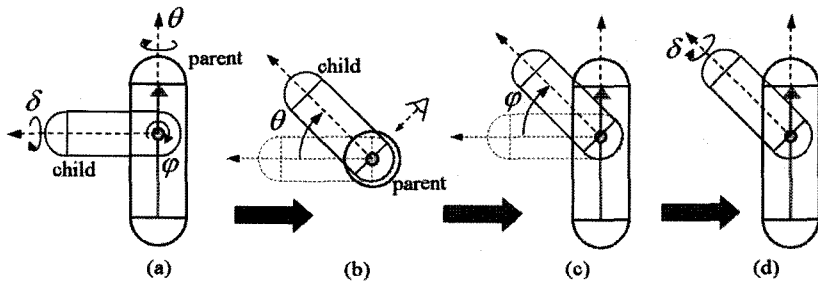


그림 3 실린더 쌍의 결합 과정. (a) 자식 실린더가 부모 실린더의 결합위치로 지정된 지점에 놓여진다. (b) 자식 실린더는 부모 실린더의 로컬 z-축을 중심으로 θ 만큼 회전한다. (c) 자식 실린더는 부모 실린더의 로컬 y-축을 중심으로 ϕ 만큼 회전한다. (d) 자식 실린더는 자신의 로컬 z-축을 중심으로 δ 만큼 회전한다.

모터 컨트롤러는 분절 삼각 함수로 이루어지며 Grzeszczuk과 Terzopoulos의 스플라인 기반 컨트롤러[9]와 유사한 형태를 가진다. 컨트롤러의 출력은 해당 모터의 목표 각속도를 나타낸다. 각 모터는 이 출력을 받아 자신의 최대로 낼 수 있는 토크 한도 내에서 이 속도에 이르기 위해 모멘트를 가하게 된다. 컨트롤러의 파라미터로는 임의의 개수의 컨트롤 포인트와 컨트롤러의 주기가 있다. 각각의 컨트롤 포인트는 -1에서 1 사이의 값을 가지며 이 값은 각 조인트의 지정된 가동범위 각도에 의하여 스케일링된다. 하나의 모터 컨트롤러 함수는 다음과 같이 표기될 수 있다.

$$U(t) = \frac{p_n + p_{n+1}}{2} + \frac{p_n - p_{n+1}}{2} \cos\left(\frac{Nt}{D} - n\right)\pi$$

$$n = 0, 1, 2, \dots, N-1 \quad p_N = p_0 \quad 0 \leq t \leq D$$

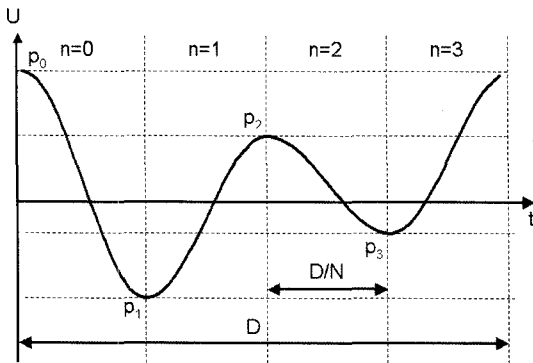


그림 4 모터 컨트롤러

N은 컨트롤 포인트의 개수를 나타내고, D는 한 주기이다. 컨트롤 포인트의 개수와 각 포인트의 값, 그리고 컨트롤러 주기는 모두 유전형 안에 인코딩된다. 컨트롤 함수는 각 컨트롤 포인트 사이를 코사인 함수로 부드럽게 보간함으로써 만들어진다. 맨 마지막의 컨트롤 포인트는 전체 컨트롤 함수가 주기적인 형태를 갖게 하기 위하여 다시 첫 번째의 컨트롤 포인트와 연결된다. 이러한 삼각함수 형태의 컨트롤 함수를 사용하는 것에는 두 가지 이점이 있다. 첫째로 대부분의 실제 관절을 가진 생물들의 움직임은 이러한 삼각 함수의 조합으로 표현되어질 수 있다. 둘째로 삼각 함수는 무한한 미분 연속성을 가지므로 예기치 않은 모터 토크의 불안정한 진동(fluctuation)을 최소화 할 수 있다. 삼각 함수로 연결된 분절 삼각함수 컨트롤러는 이산적인 시간스텝 당 모터 각속도의 변화를 최소화 시켜준다.

5. 물리 시뮬레이션

물리기반 가상생명체의 움직임은 다관절 강제 역학을

로 계산되었다. 본 논문에서는 Russell Smith [10]의 Open Dynamics Engine (ODE)을 사용하였다. ODE는 오픈소스 다관절 시뮬레이션 엔진으로서 수치적분을 위하여 1차 준 암묵적(semi-implicit)[11] 방법을 사용하고 있는데, 각각의 링크를 관절의 위치에 묶어주는 구속력에 대해서는 암묵적(implicit) 방법이, 외부 힘에 대한 움직임에 대해서는 명시적(explicit) 방법이 혼합되어 사용되고 있다. 구속력의 계산은 전체적인 알고리즘의 속도 비교로 볼 때 중간정도의 속도로 볼 수 있는 Lagrange multiplier[12] 방법을 사용하며 대략 $O(n^3)$ (n은 링크의 개수)의 계산량이 소요된다.

각 실린더에 가해지는 유체의 힘은 Weichert와 Haumann[13]의 컴퓨터 애니메이션을 위한 단순화된 유체역학을 기반으로 모델링되었다. 기존의 방법처럼 다면체의 모든 면에 대하여 유체의 반력을 계산하는 대신, 본 연구에서 쓰인 매우 전형적이고 규칙적인 기하체(capped-cylinder)인 점을 이용하여 저항력과 저항토크의 두 부분으로 나누어 보다 간결해진 물리식으로 유도하여 사용하였다. 실린더로 향하는 유체의 유속은 단순히 실린더의 질량중심의 속도를 뒤집어서 얻어질 수 있다. 유체의 저항력은 다음과 같이 기술되어질 수 있다.

$$F = \rho(2RL \sin \theta + \pi R^2)vv$$

R과 L은 각각 실린더의 반지름과, 마개(cap)부분을 뺀 길이를 나타낸다. ρ 는 물의 밀도이며 섭씨 4도를 기준으로 한 1000 kg/m^3 의 값을 갖는다[14]. 그림 5에서 실린더에 가해지는 저항의 계산에 대한 설명이 나타나고 있다.

저항토크의 계산에서, 계산의 편리함을 위하여 마개실린더는 기본적인 실린더와 두 개의 반구 부분으로 나누어져 계산되었다. 기본 실린더의 저항 토크의 계산은 실린더를 z축 방향으로 잘게 나눈 후, 나누어진 단위 슬라이스가 받는 힘으로부터 생기는 전체 실린더의 질량중심에 대한 토크를 모든 슬라이스에 대해 적분함으로써 쉽게 얻어질 수 있다. 두 개의 반구로부터의 토크는 간단히 반구의 사영된 면(반원모양)에 대한 저항력으로 계산할 수 있다. 그림 6에서 볼 수 있듯이 각각의 로컬 축에 대한 전체 실린더의 회전 저항은 다음과 같은 식으로 나타내어진다.

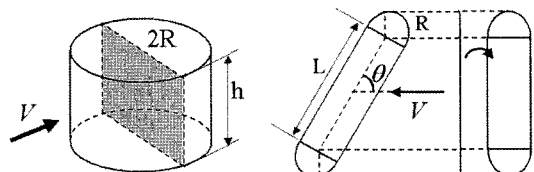


그림 5 실린더에 가해지는 linear force

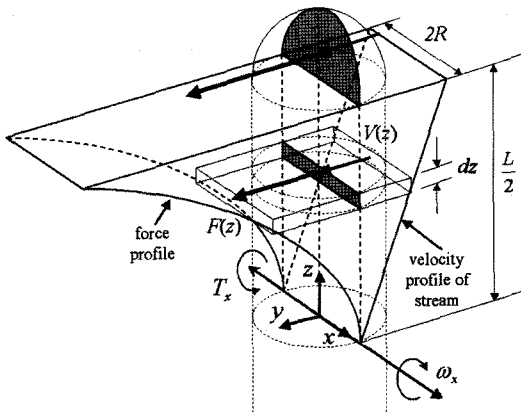


그림 6 실린더의 저항토크의 계산

$$\mathbf{T}_x^c = -4\rho\omega_x^2 R \int_0^L z^3 dz \cdot \mathbf{x}$$

$$\mathbf{T}_y^c = -4\rho\omega_y^2 R \int_0^L z^3 dz \cdot \mathbf{y}$$

$$\mathbf{T}_z^c = -2\rho\omega_z^2 (L+2R) \int_0^R r^3 dr \cdot \mathbf{z}$$

ω 는 실린더의 각 로컬 축에 대한 각속도이며, x, y, z 는 각 로컬 축 방향으로의 단위벡터이다. 양끝의 반구 부분으로부터의 저항토크는 다음과 같다.

$$\mathbf{T}_x^s = -\rho\pi R^2 \omega_x^2 \left(\frac{4R}{3\pi} + \frac{L}{2} \right)^2 \cdot \mathbf{x}$$

$$\mathbf{T}_y^s = -\rho\pi R^2 \omega_y^2 \left(\frac{4R}{3\pi} + \frac{L}{2} \right)^2 \cdot \mathbf{y}$$

$$\mathbf{T}_z^s = -\rho\pi R^2 \omega_z^2 \left(\frac{4R}{3\pi} \right)^2 \cdot \mathbf{z}$$

항 $4R/3$ 은 반원(반구의 단면)의 중심으로부터 반원의 질량중심까지의 거리를 나타낸다. 위에서 기술된 토크와 각속도들은 실린더의 각 로컬 축에 대하여 측정된 뒤 총합이 실린더의 질량중심에 가해지게 된다.

$$\mathbf{T}_{total} = (\mathbf{T}_x^c + \mathbf{T}_x^s) + (\mathbf{T}_y^c + \mathbf{T}_y^s) + (\mathbf{T}_z^c + \mathbf{T}_z^s)$$

6. 가상생명체 진화 시스템

직진 기동을 위한 가상생명체의 생성에 있어서 생명체는 정확히 직선으로 움직일 필요는 없다. 차후에 신경망 컨트롤러에 의하여 주어진 경로를 따라가도록 다시 훈련되어질 것이기 때문이다. 정확한 직선운동을 원한다면 상위레벨 컨트롤러가 생성된 후 직선의 경로를 제공하면 된다. 기본적인 움직임을 위한 적합도는 우선적으로 매 시뮬레이션 스텝마다 측정된 생명체 몸체의 질량중심의 속도로 지정되었다. 추가로 너무 직선이지 않고 제

자리에서 작은 원을 그리며 도는 등의 움직임으로 국소 최적화가 되지 않도록 적절한 가중치로 직진운동에 대한 보상을 주었다. 이는 시뮬레이션이 끝났을 때 생명체의 질량중심이 처음 시작했던 위치로부터 얼마나 멀리 떨어져 있는지를 측정함으로써 계산되었다. 적합도 식은 다음과 같이 표현된다.

$$f = k_a \sum_i \|\mathbf{p}_i - \mathbf{p}_{i-1}\| + k_b \|\mathbf{p}_{end} - \mathbf{p}_{start}\|$$

T 는 전체 시뮬레이션 시간을 나타내고, k 는 각 항(속도와 직진성)의 가중치를 나타낸다.

처음부터 열등한 개체를 초기에 제거하여 시뮬레이션 시간의 단축을 위한 몇 개의 조치가 취해졌다. 첫번째로 시뮬레이션 기간동안 시뮬레이션 불안정으로 인한 blow-up을 발생시키는 개체는 0점의 적합도를 준 후 시뮬레이션을 중단시켰다. 또, 전체 시뮬레이션 기간의 4분의 1동안의 적합도를 측정한 후, 바로 전 세대의 최저 적합도의 5분의 1 미만인 개체의 시뮬레이션도 또한 취소되었다.

한 세대의 모든 개체에 대해 적합도가 특정된 후에는 일정한 규칙에 따라 유전형에 대해 변형과 교배를 가하였다. 사용된 연산은 mutating, crossover, grafting[3] 의세가지이며 Sims의 방법과 동일하게 진행되었다. 만일 하나의 외부노드가 연산중에 삭제되었다면 그 노드를 가리키고 있던 외부노드 안의 뉴런들의 커넥션은 모두 끊어지게 되며 새로운 연결된 외부노드의 뉴런들과 랜덤하게 재연결된다. 돌연변이는 외부노드와 외부커넥션의 파라미터와 뉴런의 파라미터, 그리고 뉴런의 연결 구조에 적용되었으며 앞에서 정해진 값의 범위 내에서 행하여졌다.

7. 신경망의 결함

1단계 진화가 끝난 기본적인 움직임을 갖는 생명체의 상위 레벨로서 경로제어가 신경망을 사용하여 구현되었다. 신경망은 각 관절의 모터 컨트롤러의 신호의 형태를 조작하여 전체적인 움직임의 변화를 만들어낼 수 있다. 경로제어를 위한 훈련 샘플을 가지고 있지 않으며 신경망의 출력과 생명체의 최종적인 움직임 사이의 구배정보도 가지고 있지 않기 때문에 전통적인 신경망 훈련 기법인 오류 역전파(back propagation)등의 방법을 쓸 수 없다. 따라서 신경망의 연결강도는 유전자 알고리즘으로 최적화를 시키게 된다.

7.1 신경망 파라미터

각 관절모터마다 증폭(scaling)과 상 왜곡(phase-warp)의 두 개의 변조 형태가 사용되었다. 이 모든 변조는 컨트롤러의 주기 안에서 이루어지며 주기의 변화는 모든 관절 컨트롤러에 대하여 동일하게 적용되었다.

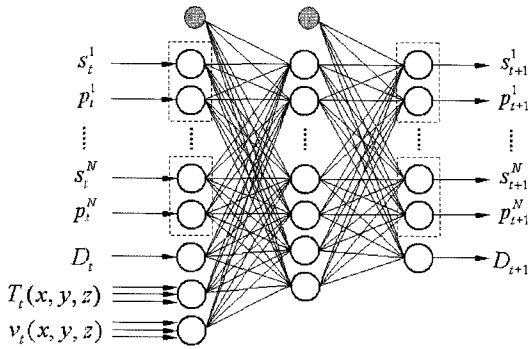


그림 7 신경망 컨트롤러 모델. s, p는 각각 하나의 컨트롤러에 대한 스케일링과 시간왜곡을 조절하는 신호이며, T는 목표지점 벡터, v는 가상생명체의 진행속도 벡터이다.

이는 실제 생명체의 움직임에 비추어 볼 때 타당한 것으로서 반복적인 관절의 움직임으로 기동을 할 때는 각 관절들 사이는 불규칙한 순서로 움직이는 듯 보이지만 전체적으로 보면 모든 관절들이 하나의 주기를 가지는 운동을 보여주기 때문이다. 증폭 신호는 단순히 모터 컨트롤러에 곱해져서 모터신호의 진폭을 변화시킨다. 상 왜곡 신호는 시간을 왜곡 시키는 time warping으로서 Wu와 Popović[15]의 방법과 마찬가지로 시간방향으로 모터신호를 왜곡시킨다. 그림 9는 이러한 신호변조에 대해 나타내고 있다.

신경망 노드에 쓰인 활성화함수는 -1에서 1 사이의 값을 가지는 bipolar sigmoid 함수를 사용하였다. 모터 컨트롤러의 한 주기가 시작될 때마다 새로운 증폭(scaling), 상 왜곡(phasewarp), 지속 시간(duration)값들이 연속적인 신경망의 출력으로부터 샘플링되어 각각의 모

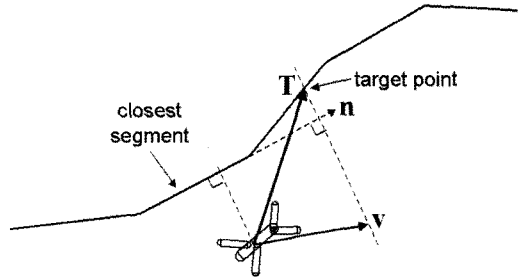


그림 8 경로상의 목표지점의 계산. 매시간마다 분절 직선으로 이루어져 있는 경로상의 목표지점을 계산하기 위하여, 가상생명체의 질량중심과 최단 거리에 위치해 있는 부분직선(closest segment)과 그 때의 생명체의 속도벡터를 이용하여 다음 시간에 진행하여야 할 목표지점(target point)을 계산하게 된다. 속도벡터 v의 끝점을 지나면서 최단 거리에 있는 부분직선 방향벡터 n을 법선으로 가지는 평면과 만나는 경로상의 점 T를 구할 수 있다.

터 컨트롤러의 입력으로 주어지고, 그 주기가 끝날 때까지의 관절들의 움직임을 결정하게 된다. 바로 전 스텝의 출력은 다른 외부 입력(속도, 목표 지점의 위치)과 함께 다시 신경망의 입력으로 피드백된다.

외부 입력인 속도(v)와 목표지점(T)을 나타내는 벡터들은 생명체의 특정 몸체부분에 대한 로컬 좌표계에 대하여 변환된 후 사용되는데 생명체의 형태가 임의적이기 때문에 어떤 부분이 팔다리고 어떤 것이 몸통인지의 구별이 없다. 하지만 속도와 목표 지점 등의 정보에 대한 기준을 필요로 하므로 어떤 특정 부분의 실린더를 “몸통”이라고 결정해야만 한다. 이들의 구분을 정의하기

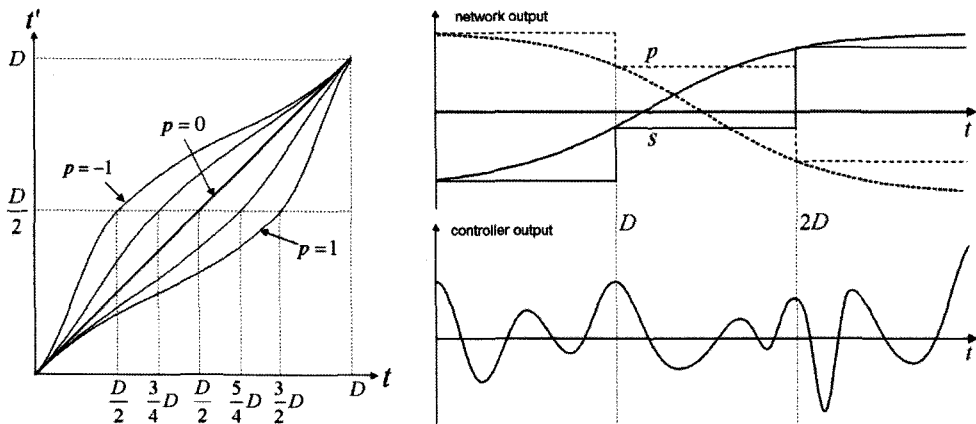


그림 9 시간왜곡(Time-warping) 함수(좌)와 신경망 신호에 의한 모터 컨트롤러의 변화(우). 신경망의 출력은 컨트롤러의 한 주기가 끝날 때 마다 샘플링되어 다음 주기의 움직임을 결정한다.

위하여 본 논문에서는 시뮬레이션을 짧은 시간 돌려본 후 가장 평균적으로 생명체의 질량중심에 가장 가까운 부분을 "몸통"으로 정의하였다.

생명체가 따라가게 될 경로는 사용자에게 의하여 분절 선형 곡선(piecewise linear curve)으로 주어진다. 매 시뮬레이션 스텝마다 생명체의 속도의 함수로 경로상의 목표지점이 얻어진다. 시간이 흐름에 따라 변하는 목표 지점을 얻어내는 방법은 Rival[16]의 heading-based 접근법과 흡사한 방법을 사용하였다. 주 아이디어는 자동차의 속도가 더 빠를수록 운전자는 더 먼 지점을 바라본다는 원리를 모델링한 것이다. 그림 8에서 이러한 경로 인지 과정을 묘사하고 있다.

7.2 주어진 경로에 대한 신경망 진화

유전자 알고리즘을 위한 염색체의 길이는 (H+1)×(4N+9)가 되며 H는 은닉층 뉴런 개수, N은 모든 관절의 총 자유도의 개수이다. 염색체는 연결강도와 바이어스들을 포함한다. 본 논문에서는 은닉뉴런의 개수로 모든 자유도 개수의 1에서 2배의 개수를 사용하였다. Path-following을 위한 적합도 함수는 다음과 같다.

$$f = \int \{k_1 d_{min}(t) - k_2 v_t(t) - k_3 a_T(t)\} dt$$

T는 시뮬레이션된 전체 시간이고, d_{min} 은 경로로부터의 생명체까지의 최단 거리이며 v_t 는 d_{min} 지점에서의 경로방향으로의 생명체의 속도이다. a_T 는 현재 목표지점 방향으로의 가속도를 나타낸다. $k > 0$ 는 각 항의 가중치이다. 적합도가 낮은 값을 가지는 생명체가 더 우수하다고 정의된다.

적합도 측정이 끝날 때 마다 개체군은 세 개의 그룹으로 나누어진 후 비정상적인 형질을 보이는 하위 두 집단에 대하여 강한 페널티를 주었다. 이러한 하위 두 그룹의 정의로는 첫번째로 물리 시뮬레이션이 불안정하여 폭발

(blow-up)할 때(수치오류, blow-up failure), 그리고 두 번째로는 생명체의 위치가 주어진 경로와 너무 많이 멀어졌을 때이다(경로 오류, heading failure). 이러한 오류 중 어느것에도 해당되지 않는 '정상적'인 개체는 세 집단 중 맨 첫번째로 분류된다. 수치오류 집단은 경로오류보다 더 비정상적인 집단으로 분류되었다. 이러한 두 비정상적 집단에 대해서는 확인되는 즉시 시뮬레이션이 중단되고 다음의 식에 의하여 적합도가 매겨진다.

$$f_{worse} = P + \frac{1}{t_a} \int_0^{t_a} S(t) dt$$

$$f_{worst} = Q + \frac{1}{t_a} \int_0^{t_a} S(t) dt$$

P와 Q는 각각 각 그룹을 확실히 구별할 정도로 큰 값을 가지며, t_a 는 시뮬레이션이 중단된 시간을 나타낸다. 두 번째 항은 시뮬레이션된 기간동안의 평균 적합도(S(t)는 매시간마다 측정된 순간 적합도이다)를 나타낸다.

이러한 검사는 오직 어떤 시간간격 동안만 행해졌는데($0 \leq t \leq t_c$), 편의상 검사 시간이라 부른다. 이는 만약, 이러한 오류검사를 전체 시뮬레이션 시간동안 수행한다면 첫 번째 그룹(정상적 그룹)에 살아남을 개체는 거의 없을 것이기 때문이다. 아마도 수치에러는 드물겠지만 거의 모든 개체들이 한 번 이상 경로오류에 걸리게 될 것이며 잠재적으로 우수한 형질을 가진 개체들에 대한 시뮬레이션을 충분히 유지하지 못하게 된다. 검사 시간은 전체 시뮬레이션 시간의 일부분 () 동안으로 초기화되며, 개체군의 평균 적합도가 높아짐에 따라 다음과 같이 점차적으로 그 시간범위를 늘리게 된다.

$$t_c^{g+1} = T \left\{ (1-\alpha) \frac{N^g}{P} + \alpha \right\}$$

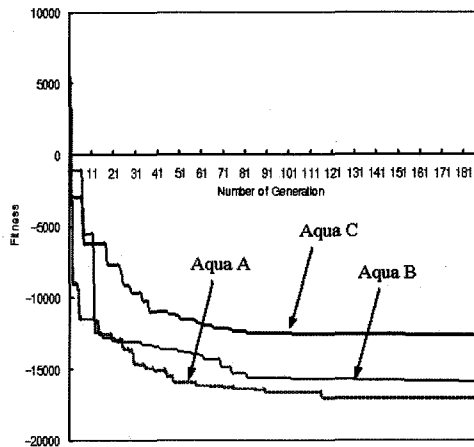
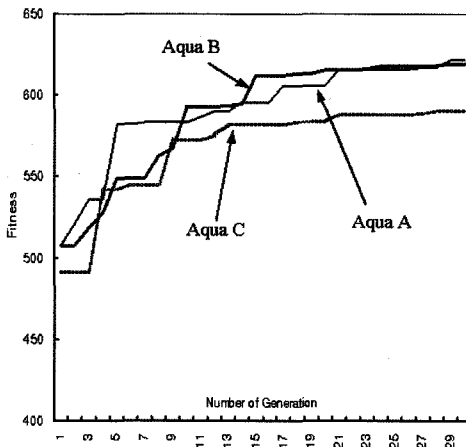


그림 10 기본생명체 진화과정(좌)과 신경망 진화과정(우)의 적합도 그래프

T 는 전체 시뮬레이션 기간이고, g 는 세대 수를 나타낸다. N 은 첫번째 그룹에 들어간 생존개체의 수, 그리고 P 는 개체군의 크기를 나타낸다.

8. 결과

진화 시스템 설정은 다음과 같다. 개체군의 수는 대략 300~500이며 50~300세대에 걸쳐 진행되었다. 각 세대마다 상위 20%의 우수한 개체가 다음세대로 변화 없이 복사되었고 남은 80%는 크기 2의 토너먼트 선택(elite 선택 확률 90%)으로 선정된 부모로부터 40%의 재생산, 30%의 교배(crossover), 30%의 접붙이기(grafting)로 채워졌다. 교배연산을 위한 두번째 부모는 상위 20% 들로부터 임의로 선택되었다. 다음 세대가 완전히 채워진 후 상위 20%를 제외한 나머지 개체들에 대하여 돌연변이가 적용되었다.

신경망 연결 가중치 진화 단계에서는 개체군의 수는 200~300, 100~200 세대동안 실행되었다. 시뮬레이션 기간은 1000~2000 스텝이었으며 생명체의 기동 속도에 반비례하여 조절되었다. 유전자 알고리즘의 연산들은 생명체 생성시와 마찬가지로 설정되었으며 단, 교배연산에 있어서는 4점 교배까지를 허용하였다.

전체적인 시스템 수행 시간은 기본 생명체 진화에 1~2 시간, 신경망 진화에 2~4시간이 소요되었으며 플랫폼은 인텔 펜티엄-4, 2.53GHz PC를 사용하였다. 기존의 수십개의 프로세서를 장착한 CM-4 Connection Machine 에서 4시간 가량이 걸린 것을 볼 때 매우 시간이 단축되었음을 알 수 있다. 이는 매우 간결한 모터 컨트롤러의 모델링과 주기적인 움직임에 특화된 디자인으로 인하여 진화과정의 탐색공간이 매우 축소되었기 때문이라고 할 수 있다.

경로를 따라가는 데 있어서 생명체는 한 개의 적절한 훈련 경로를 사용했음에도 불구하고 주어진 어떤 경로도 성공적으로 따라가는 모습을 보였다. 이는 수중환경이 지닌 높은 유체의 저항력과 부력의 작용으로 인한 기동의 안정성이 높기 때문이다. 그러나 몇몇 지역 최적화가 관찰되었는데, 매우 느린 속도로 경로 가까이 계속

표 1 시뮬레이션 파라미터

파라미터	값 또는 범위
링크의 개수	3~10
실린더 길이	0.2~1.0 m
실린더 반지름	0.1~0.4 m
θ	0.0~rad
ϕ	0.0~rad
δ	0.0~rad
실린더 밀도	1000.0 kg/m ³
물의 밀도	1000.0 kg/m ³
k_a, k_b	0.5, 1.0
k_1, k_2, k_3	0.1, 2.0, 1.0
모터 컨트롤러 주기	0.3~1.2 sec
컨트롤 포인트 개수	2~6

붙어있지만 하거나 훈련경로의 변화가 다소 적을 때 경로오류를 범하지 않는 범위를 유지하면서 계속 직진만을 수행하는 경우가 다소 나타났다. 이는 더욱 세밀한 적합도 함수의 도입으로 해결할 수 있을 것이다. 표 1은 진화과정과 물리 시뮬레이션에 쓰인 파라미터들을 보여 준다.

9. 결론 및 향후연구

본 논문은 효율적인 분절 삼각함수 컨트롤러와 신경망을 활용한 사용자 제어와 재활용이 가능한 가상생명체를 생성하는 시스템에 대하여 기술하였다. 가상생명체들은 간단하고 사용하기 편리한 모터 컨트롤러를 사용하여 쉽게 진화되었으며 신경망 컨트롤러를 사용한 경로찾기 행위는 더 빠르고 수월하게 이루어질 수 있었다. 가상생명체의 기본적 기동과 상위 컨트롤을 분리함으로써 사용자는 진화된 가상생명체들을 훨씬 쉽게 제어할 수 있게 되었다.

하지만 본 논문의 시스템 디자인에서, 기존의 매우 생물학적으로 그럴듯해보이는 물뱀 같은 부드럽게 물결치는 듯한 움직임을 진화해내기 힘들었는데, 이는 모든 관절들이 서로 강제적으로 동기화가 되도록 설계됨으로 인하여 링크의 한 부분으로부터 다음 부분으로 움직임이 전파되는 움직임이 나오기 힘들기 때문이다. 이러한 움직임을 위해서는 기존의 뉴런기반의 모터컨트롤러가 알맞겠지만 본 논문의 분절 삼각함수 컨트롤러로도 각 컨트롤러의 상 이동(phase shift)을 위한 변수를 유전형 안에 인코딩함으로써 이루어질 수 있을 것으로 본다.

본 시스템은 또한 육상환경의 가상생명체에도 쉽게 적용되어질 수 있을 것이라 예상된다. 육상의 기어다니는 기동은 수중에서와 달리 실제적으로 평면상의 움직임에 가깝기 때문에 더 적은 신경망의 뉴런 개수를 가지고도 효율적으로 경로찾기등의 행위를 생성해낼 수

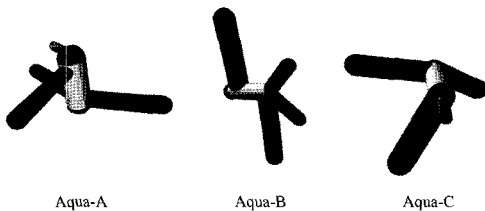


그림 11 진화된 기본 생명체들. 하얀 부분이 “몸통”을 나타낸다.



그림 12 진화된 가상생명체들 : Aqua-A, Aqua-B, Aqua-C

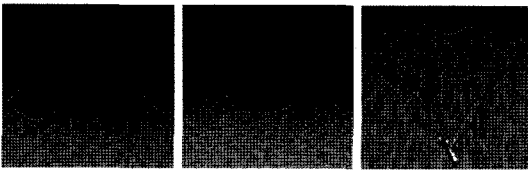
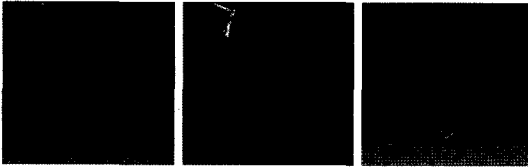


그림 13 Aqua-A, Aqua-C, Aqua-C for following S-curve, spiral, heart shape

있을 것이다. 한단계 더 나아가서 이러한 시스템을 활용하여 무리짓기 행동, 장애물 회피 등의 다른 상위레벨 행위도 진화시킬 수 있을 것이다. 또는 많은 수의 가상 생명체들이 서로 상호 작용하는 총체적인 가상환경의 구축도 가능하다. 컴퓨터의 계산 능력이 점점 강력해짐에 따라 이러한 연구는 가상환경에 무한한 풍부함과 다양성을 자동으로 생성할 수 있게 해 줄 것이다.

참 고 문 헌

- [1] Ijspeert, A.J., Hallam, J. and Willshaw, D. (1998) From lampreys to salamanders: evolving neural controllers for swimming and walking, From Animals to Animats, Proceedings of the Fifth International Conference on the Simulation of Adaptive Behavior(SAB98), R. Pfeifer, B. Blumberg, J.-A. Meyer and S.W. Wilson (eds), MIT Press, pp. 390-399.
- [2] Reeve, R. and Hallam, J. (2005) An Analysis of Neural Models for Walking Control. *IEEE Transactions on Neural Networks*, vol. 16, no. 3.
- [3] Sims, K. 1994. Evolving Virtual Creatures, *ACM Computer Graphics (SIGGRAPH '94)*, pp. 15-22.
- [4] Sims, K. 1994. Evolving 3D Morphology and Behavior by Competition. R. Brooks, & P. Maes, (eds.) *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 28-39, MIT Press.
- [5] Ventrella, J. 1998. Attractiveness vs. Efficiency: How Mate Preference Affects Locomotion in the Evolution of Artificial Swimming Organisms. Adami, C. et al. (eds.) *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*. pp. 178-186: MIT Press.
- [6] Komosinski, M and Ulatowski, S. 1999. Framsticks: Towards a Simulation of a Nature-Like World, Creatures and Evolution. Floreano, D. et al. (eds.) *Advances in Artificial Life: Proceedings of the Fifth European Conference on Artificial Life*, pp. 261-265: Springer Verlag.
- [7] Ray, T. S. 2000. Aesthetically Evolved Virtual Pets. Maley, C.C. & Boudreau, E. (eds.) *Artificial Life VII Workshop Proceedings*. pp. 158-161.
- [8] Lipson, H. and Pollack, J. B. 2000. Automatic Design and Manufacture of Robotic Lifeforms. *Nature* 406, pp. 974-978.
- [9] Grzeszczuk, R., and Terzopoulos, D. Automated learning of muscleactuated locomotion through control abstraction. In Proceedings of SIGGRAPH 95 (Los Angeles, Aug. 6.11). ACM Press, New York, 1995, pp. 63.70.
- [10] Smith, R. 1998. Intelligent Motion Control with an Artificial Cerebellum. *PhD Thesis, Dept of Electrical and Electronic Engineering, University of Auckland, New Zealand*. (Available online including ODE engine at <http://opende.sourceforge.net>)
- [11] Anitescu, M. and Potra, F. A. 1997. Formulating rigid multi-body dynamics with contact and friction as solvable linear complementarity problems, *Nonlinear Dynamics* 14, 231.247.
- [12] Stewart, D. E. and Trinkle, J. C. 1996. An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction, *International J. Numerical Methods in Engineering* 39, 2673-2691.
- [13] Wejchert, J. and Haumann, D. 1991. Animation aerodynamics. *SIGGRAPH 91 Proceedings*. volume 25, number 4.
- [14] Fox, R. W. and McDonald, A. T. 1976. Introduction to fluid mechanics. fifth edition, Wiley.
- [15] Wu, X. C. and Popović, Z. 2003. Realistic Modeling of Bird Flight Animations. in *Proceeding of SIGGRAPH '03*, pp. 888-895.
- [16] Rivals, I., Personnaz, L., Dreyfus, G., and Canas, D. 1993. Real-time Control of an Autonomous Vehicle : A Neural Network Approach to the Path Following Problem. in *5th International Conference*

on Neural Networks and their Applications
(*NeuroNimes93*).



심 윤 식

2004년 고려대학교 컴퓨터학과 대학원
(석사). 관심분야는 컴퓨터 그래픽스

김 창 현

정보과학회논문지 : 시스템 및 이론
제 32 권 제 6 호 참조