



HOKKAIDO UNIVERSITY

Title	Animated Robotの研究 : 剛体モデリングツールの開発とその応用
Author(s)	岩館, 健司; 米陀, 佳祐; 鈴木, 育男 他
Citation	精密工学会誌, 76(2), 232-237 https://doi.org/10.2493/jjspe.76.232
Issue Date	2010
Doc URL	https://hdl.handle.net/2115/50882
Type	journal article
File Information	seimitsu76020232.pdf





Animated Robot の研究*

—剛体モデリングツールの開発とその応用—

岩館健司** 米陀佳祐** 鈴木育男*** 山本雅人*** 古川正志***

A study on Animated Robot

- Development of Modeling Tool for Rigid Objects and its Application -

Kenji IWADATE, Keisuke YONEDA, Ikuo SUZUKI, Masahito YAMAMOTO and Masashi FURUKAWA

In this decade, physics animation which realizes a realistic motion of the object has attracted the attention of researchers. However, the physics animation with animate beings which have autonomous intelligence is still in the research phase. This study aims at developing a new physics modeling software to generate behaviors for virtual creatures which have complex shapes. Developed software has functions to create physics models which have complex shapes on GUI, simulate them in real time, and generate an autonomous behavior with Artificial Neural Network. This paper describes architecture of the software and several examples of motion generations.

Key words: physics modeling, artificial neural network, genetic algorithm, adaptive behavior, artificial life

1. 結 言

現在、三次元コンピュータグラフィックス技術は工学、物理学、医学、映画産業、ゲーム産業等、多岐にわたる分野において必要不可欠な技術となっており、コンピュータの発展と共に劇的な技術革新が行われてきた。近年のエンタテインメントの分野においては、製作したモデルをいかに「それらしく」動かすかというアニメーション技術への関心が高まっており、仮想の物理環境を構築し、モデルを重力や慣性力に従って挙動させる手法や、仮想の粒子を多数配置し、各々の粒子間に働く力（引力、斥力、粘性等）をシミュレートすることで流体（水、炎、煙等）の振る舞いを表現する手法¹⁾⁷⁾が提案され、実際に映画やゲーム、ソフトウェアなどに応用されている²⁾⁸⁾¹⁰⁾。

しかしながら、生物のように自らの意思によって自律的に行動する物体に対するアニメーションの自動生成手法は未だ研究段階にあり、実用段階には至っていない。現在の三次元 CG アニメーションでは、製作したモデルに対して、クリエイターが時系列に沿ってモデルのポーズ（位置、姿勢）を指定し、ポーズ間の動作を補間することでアニメーションを製作する手法や、現実世界の物体の動きをモーションキャプチャによって数値化し、この値をモデルに対して適用する手法が主流となっている。しかし、これらを用いたアニメーション製作には専門的な知識が要求され、莫大な時間と労力を要する。

この問題を解決するため、単純な演算ルールの組み合わせや、人工ニューラルネットワーク(Artificial Neural Network, ANN)の利用によって自律的に物体のモーションを生成するエージェントベースのアプローチが提案され³⁾、多くの研究者によって研究が進められている⁴⁾⁵⁾⁹⁾¹¹⁾。しかしながら、これらの既存研究は

いずれも形状に特定の制限を持つモデルに対するモーション生成を行っており、自由に製作された複雑な形状を持つモデルに対してモーションを生成する研究は行われていない。また、これらの研究では物理環境を考慮せず、重力や慣性力等の簡潔な力のみを実装することが多い。これらの原因のひとつに、シミュレーションに用いるモデルの作成と仮想物理環境におけるモーションの生成を相互的かつ円滑に行う環境が整備されていないことが挙げられる。従って、これらの基盤を整えることは今後のエージェントベースアプローチの発展に大きく貢献すると考えられる。そこで本研究では、モデルの製作、仮想物理環境におけるモデルのシミュレーション、さらに、コントローラを用いた自律的モーションの獲得までを相互的かつ円滑に実現する物理モデリングソフトウェアを開発し、これを用いたモーション生成実験を実施した。本ソフトウェアではインタラクティブ性を重視した直感的な GUI を開発し、プログラム知識を持たない人が容易にモデル製作、自律的モーションの生成を行える開発環境の実現を目指した。モーション生成の基本的戦略として、生物が行動する際に根源的に持ち合わせている幾つかの行動評価を満たすようにコントローラの最適化を行うことで問題として設定したモーションの自動生成を実現した。

以下に、本研究で導入した概念である Animated Robot、製作したソフトウェアの構成およびモーション生成実験について報告する。

2. Animated Robot - Anibot -

Anibot とは、仮想物理空間内において自律的に行動するエージェントを示す造語であり、センサ、アクチュエータ、意思決定機能を持ち合わせたソフトウェアロボットと言い換えられる。本研究では、この概念を採用し、Anibot を仮想の人工生物として導入し、この生物自身が自律的に振舞うことでアニメーションを生成するツールの作成とその応用を研究目的としている。

* 原稿受付 平成 20 年 10 月 31 日

** 学生会員 北海道大学大学院（札幌市北区北 14 条西 9 丁目）

*** 正 会 員 北海道大学大学院

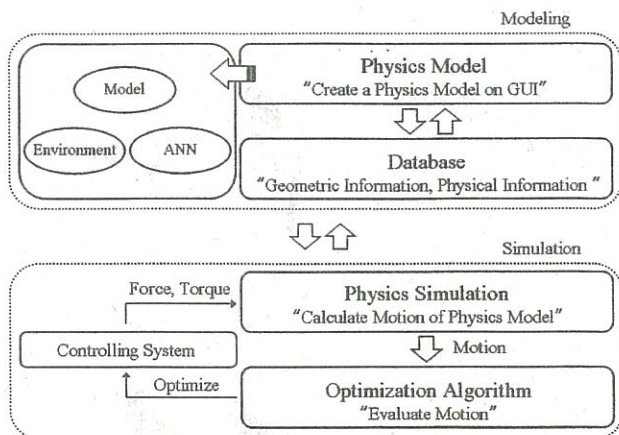


Fig.1 Software architecture

3. ソフトウェアの構成

本章では、ソフトウェアの構成および各構成要素について述べる。図1にソフトウェアの構成図を示す。

本ソフトウェア¹²⁾ではユーザが効率的にモデリング、シミュレーションを実施できる作業環境を実現するために、モデリングとシミュレーションの作業を自由に行き来し、これらの作業を相互的に進めることを可能とした。これにより、ユーザは製作した物理モデルが仮想物理環境内でどのように挙動するかを逐次確認しながらモデリングを進めることが可能である。また、本ソフトウェアで取り扱う物理モデルには、感覚器官に相当するセンサ、駆動系に相当するアクチュエータを実装することが可能であり、これらが実装された場合に、センサからの情報を入力値、アクチュエータへの制御力を出力値に持つコントローラが機能し、物理モデルはランダムに生成されたコントローラによって自律的にモーションを生成する。ここで生成されたモーションを最適化アルゴリズムにより評価し、コントローラを最適化することにより、様々なモーション生成が可能である。

以下にソフトウェアの各構成要素について詳細を述べる。

3.1 物理モデル

本研究で取り扱う物理モデルとは、幾何情報および物理情報を持ち合わせた三次元剛体モデルを示す。本ソフトウェアで実装した物理モデルは、プリミティブ、ジョイントの二つの構成要素から成っており、複数のプリミティブをジョイントで接合することによって任意形状のモデルを製作することが可能である。以下に各構成要素について記述する。

3.1.1 プリミティブ

プリミティブとは基本的な形状を持つ立体であり、図2に示す球、直方体、自由に変形可能なメッシュを実装した。各々のプリミティブは表1に示す情報を備えており、ソフトウェアのユーザは図3(a) - (d)に示すGUIによる移動、拡大縮小、回転、または(e)に示すダイアログボックスを用いた数値入力によって任意にこの値を変更可能である。また、各プリミティブにはセンサとしての属性を与えることが可能である。生物が周囲の環境情報を入手するためのセンサには様々なものが考えられるが、本ソフトウェアではその中でモーション生成の際に最も重要な光センサの属性を与えることが可能である。他のセンサについては今後実装することを検討している。光センサの属性を与えられたプリミティブは、仮想物理環境におけるシミュレーション時に環境に配置された光源からの光の強度 L を式(1)に従って感知し、コントローラに入力信号として伝達する。ここで θ_s はセンサと光源の角度、 R はセンサの視野角を示す。

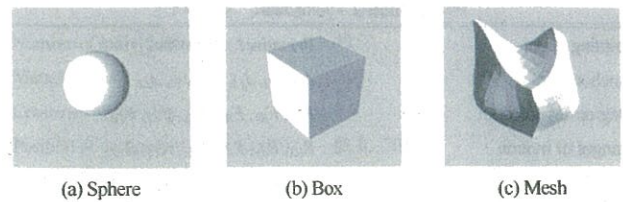
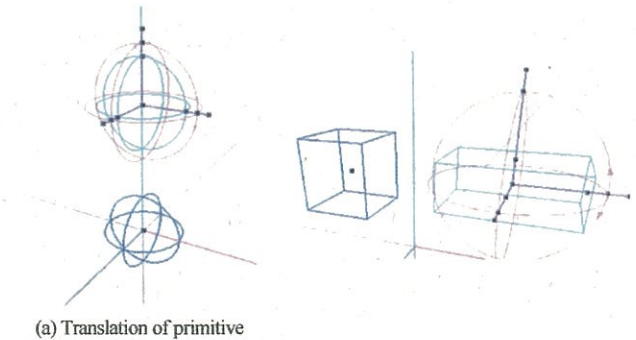
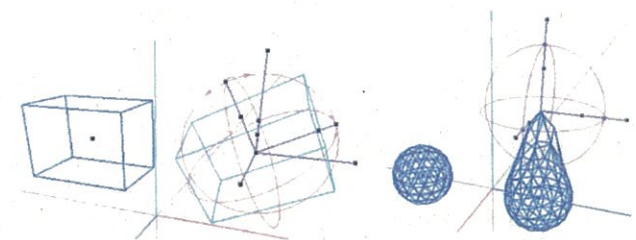


Fig.2 Primitives



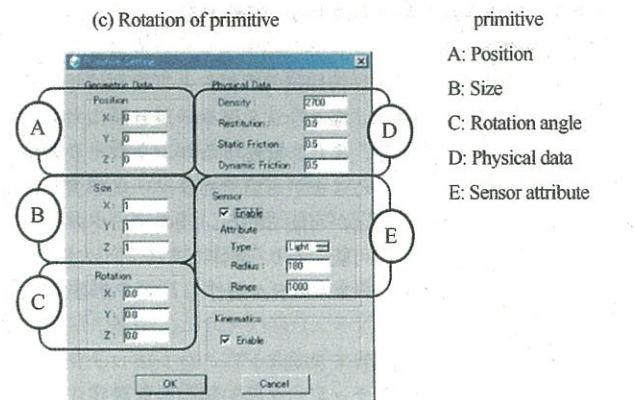
(a) Translation of primitive

(b) Transformation of primitive



(c) Rotation of primitive

(d) Transformation of mesh



(e) Dialog box

Fig.3 Software interface to edit primitives

Table 1 Primitive data

Size	D_x, D_y, D_z
Position	P_x, P_y, P_z
Rotation angle	α, β, γ
Density	ρ
Restitution coefficient	e
Static friction coefficient	μ_s
Dynamic friction coefficient	μ_d
Sensor enable	{True, False}
Angular range of the view	R

ン時に環境に配置された光源からの光の強度 L を式(1)に従って感知し、コントローラに入力信号として伝達する。ここで θ_s はセンサと光源の角度、 R はセンサの視野角を示す。

Table 2 Joint data

Jointing target	Primitive1, Primitive2
Anchor point	$J_{X1}, J_{Y1}, J_{Z1}, J_{X2}, J_{Y2}, J_{Z2}$
Degrees-of-freedom	$F_{RX}, F_{RY}, F_{RZ}, F_{TX}, F_{TY}, F_{TZ}$
Ranges of motion	$R_{RX}, R_{RY}, R_{RZ}, R_{TX}, R_{TY}, R_{TZ}$
Spring constant	k
Damper constant	C
Actuator enable	{True, False}
Spring enable	{True, False}

$$L = \begin{cases} \theta_s \leq R & : \theta_s \leq R \\ \text{otherwise} & : \text{otherwise} \end{cases} \quad (1)$$

3.1.2 ジョイント

ジョイントは、複数のプリミティブ同士を接合する役割を担う構成要素であり、本ソフトウェアでは回転、並進の合計 6 自由度、及び各自由度に対する回転、並進の許容量を任意に設定可能な 6 自由度ジョイントを実装した。各々のジョイントは表 2 に示す情報を備えており、ソフトウェアのユーザは図 4(a)に示す様に、GUI によって個々のプリミティブ ((a) - ①) 間をジョイントで接合でき ((a) - ②), (b)に示すダイアログボックスを用いた数値入力によって任意にジョイントが持つ情報を変更することが可能である。また、各ジョイントには、駆動系に相当するアクチュエータの属性を与えることが可能であり、アクチュエータの属性を与えられたジョイントは後述するコントローラによって算出された制御力によって駆動する。

3.2 データベース

3.1 で示した物理モデルの持つ幾何情報および物理情報はデータベースとして保存され、様々なシミュレーションに対して適用することが可能である。また、後述するコントローラの情報もデータベースに格納することが可能であり、一度学習したコントローラを保持することが可能である。このデータベースは、XML 形式での入出力が可能であり、製作したモデルの保存、読み込み、及びテキストベースのモデル製作も可能である。

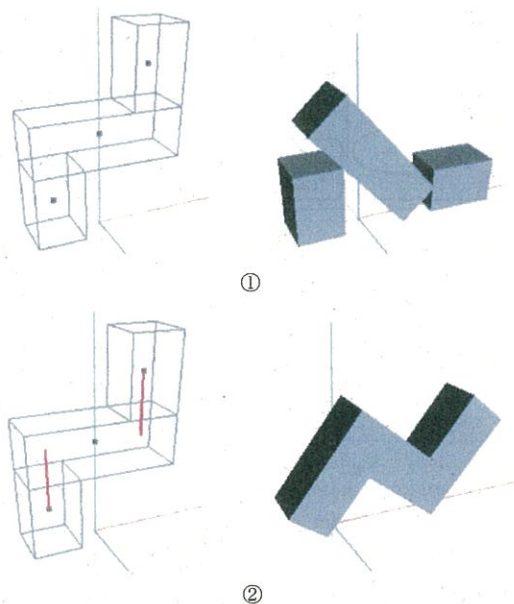
3.3 コントローラ

本ソフトウェアでは、ANN を物理モデルのコントローラとして採用し、これによりアクチュエータへの制御力を決定する。この ANN には式(1)で示したセンサからの情報、アクチュエータの角度、角速度が入力信号として与えられる。ANN は、図 5 に示す構造を持っており、中間層のニューロン数には、入力層のニューロン数及び出力層のニューロン数に従って経験的に算出されたデフォルトの値が設定される。またユーザがこのネットワーク構造、中間層のニューロン数を自由に変更することが可能となっており、単純な三層構造から、複雑なリカレント構造まで、様々なネットワーク構造を実現することが可能である。

本ソフトウェアでは各ニューロンの反応関数に sigmoid 関数を採用しており、ニューロンの入出力関係は入力信号を u_i 、出力信号を v_i 、ニューロン j 、 i 間の結合重みを w_{ji} 、温度係数を T として式(2)~(3)で与えられる。

$$v_i = \frac{1}{1 + e^{-u_i/T}} \quad (2)$$

$$u_i = \sum_j w_{ji} v_j \quad (3)$$



(a) Creation of 6DOF-Joint

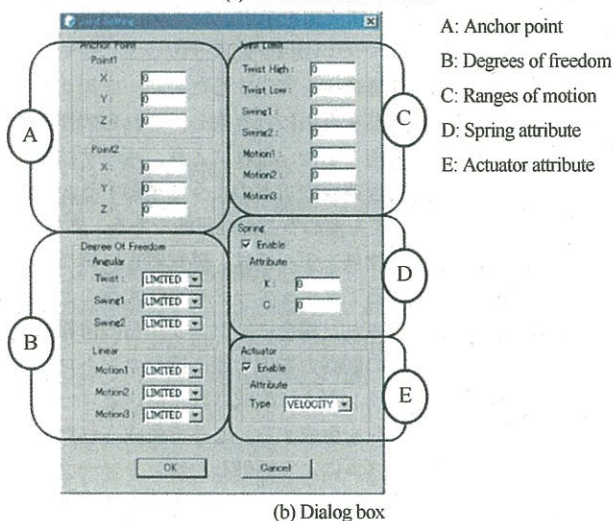


Fig.4 Software interface to edit joints

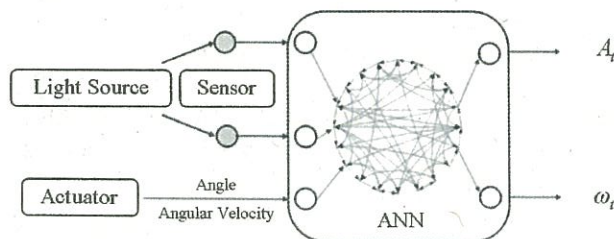


Fig.5 Structure of controller

この ANN によって出力された信号は、各アクチュエータに伝達され、物理モデルの制御力となる。アクチュエータに回転の自由度が与えられている場合にトルクが出力され、並進の自由度が与えられている場合には伸縮力が出力される。生物が歩行や遊泳を行う場合、筋肉が特定の周期運動をとることから⁹⁾¹¹⁾、本ソフトウェアでは各アクチュエータに与えられるトルク T_i に正弦関数を導入し、振幅 A_i 、角速度 ω_i 、初期位相 φ 、角度 θ_i 、時間刻み Δt として式(4)~(5)で与えた。ここで i はアクチュエータの番号を示す。

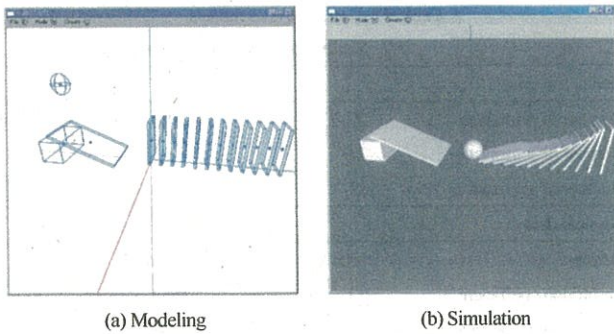


Fig.6 Modeling and simulation

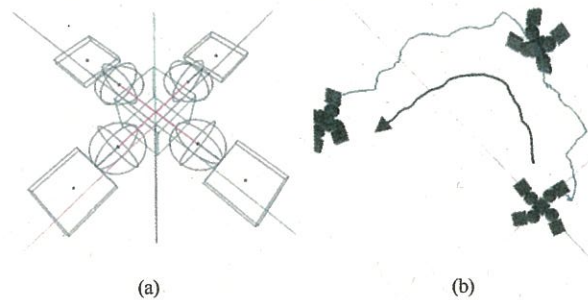


Fig.7 Physics model simulation with sensors

$$T_i = A_i \sin(\theta_i + \omega_i \Delta t + \varphi) \quad (4)$$

$$\theta_i = \theta_{i-1} + \omega_i \Delta t \quad (5)$$

3.4 仮想物理環境におけるシミュレーション

シミュレーションに用いる仮想の物理環境を構築するにあたり、本ソフトウェアでは物理エンジンを導入した。物理エンジンとは与えられた形状情報、物理情報を元に、物理法則に従って各時刻における物体の座標情報を算出するライブラリである。本ソフトウェアでは、NVIDIA 社から提供されている PhysX ライブラリ⁹⁾を使用している。PhysX は物体の衝突検出、剛体力学、弾性力学、及び流体力学に則ったシミュレーションをサポートしている。本ソフトウェアでは、図 6 で示すように製作した物理モデルに対し、物理環境におけるシミュレーションをリアルタイムに実施することが可能である。このため、ソフトウェアのユーザはモデリングとシミュレーションをインタラクティブに進めることが可能である。

また、物理モデルにセンサ、アクチュエータを与えた場合、シミュレーション開始と同時にニューラルコントローラが有効となり、モデルはランダムに生成されたニューラルコントローラに従って自律的にモーションを生成する。

図 7 にセンサ、アクチュエータを持つ簡易的なモデルによるモーション生成例を示す。図 7(a) がモデル製作時、図 7(b) はシミュレーション時のモデルの軌跡を上面から捉えた図である。このモデルは 4 本のアクチュエータを持ち、中央の直方体がセンサとなっており、ランダムに生成されたニューラルコントローラからの制御力によってアクチュエータを可動させ、仮想物理環境内で自律的にモーションを生成している。

3.5 最適化アルゴリズムによるモーションの最適化

ユーザによって製作された自由な形状を持つ物理モデルの行動を獲得させる場合、明確な教師信号を定義することは困難

Table 3 Setting value of optimization

Number of individuals	40
Mutation rate	0.05
Crossover probability	0.4
Position of destination (light source)	(0, 0, -20)
Simulation time	300

であり、教師あり学習による ANN の最適化は行えない。そこで、本ソフトウェアでは実数値遺伝的アルゴリズム(Genetic Algorithm, GA)を導入した。ANN のニューロン間の結合重み w_{ij} を遺伝子表現とし、各個体毎の結合重みを用いたコントローラにより得られた物理モデルのモーションを評価し、結合重みを最適化することで任意のモーションを獲得する。最適化に用いるパラメータを表 3 に示す。

以下に GA による ANN 最適化の手順を示し、詳細を述べる。

- (1) 各個体が持つ ANN を用いて Simulation time (300 step) の間物理モデルのシミュレーションを実施する
- (2) 得られたモーションを評価関数により評価する
- (3) 評価値に従って個体の再生、淘汰を行う
- (4) 遺伝演算(交叉、突然変異)を実施する
- (5) (1)に戻り、(1)から(4)を繰り返す

3.5.1 モーションの評価

ANN 最適化の戦略として、実際の生物が持ち合わせている習性を評価関数として導入する。生物がある目標へ移動する場合、「目標を視界に捕らえつつ、最もエネルギー効率の良い振る舞いにより最短経路で目的地へ移動する」ことが考えられる。このことから本研究では、点光源を目標位置として定め、評価関数にはシミュレーション中におけるモデルと光源の距離 D の総和および、アクチュエータを駆動させるために ANN が出力したトルクの絶対値の総和 E を最小化し、センサの入力値 L の総和を最大化する式(6)を設定した。ここで、最適化の第一の目的である光源の追跡を実現するのが D の最小化に該当するが、この値を最小化する際に、消費トルク E が増大することが考えられ、 D , E , L 間には相関関係が成り立つ。 E の項はモデルにトルクを効率的に利用したモーションを獲得させる為に設定したが、この項の影響力が大きい場合、モデルがトルク発生を抑える事を優先し、動かなくなる場合がある。上記の問題を回避するため、各評価値の評価関数へのウェイト W_1 , W_2 , W_3 を設定し、式(7)の関係を満たすような値を経験的に設定した。ここで $D_{\max}$, $L_{\max}$, $E_{\max}$ には最適化の初期世代で観測された D , L , E の最大値を定数として採用した。

$$f(D, E, L) = -W_1 D - W_2 E + W_3 L \quad (6)$$

$$W_1 D_{\max} > W_3 L_{\max} > W_2 E_{\max} \quad (7)$$

ここで W_1, W_2, W_3 は各評価のウェイトである

3.5.2 再生と淘汰

3.5.1 に示した評価関数により計算した評価値を基に再生と淘汰を実施する。再生する子孫の数 n_i を式(8)によって計算し、次世代の個体を決定する。ここで N は個体数、 f_i は個体 i の評価値、 $f_{\min}$ は全個体のうち最小の評価値を示す。

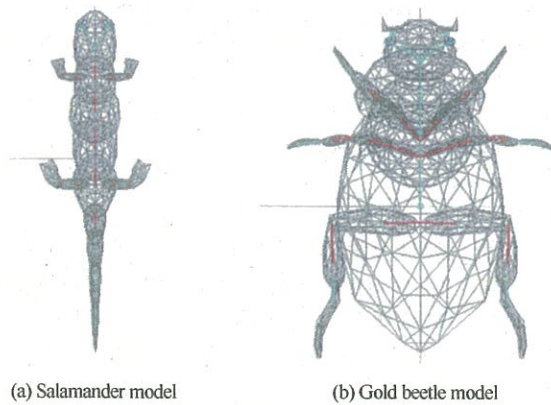


Fig.8 Physics models

$$n_i = N \frac{f_i - f_{\min}}{\sum_j (f_j - f_{\min})} + 1 \quad (8)$$

また、下位 50%の個体は、遺伝演算後にランダムに生成した新たな個体に置き換える。

3.5.3 遺伝演算

遺伝演算には突然変異、交叉を採用した。突然変異ではランダムに選択されたニューロン間の結合重み w_{ij} に対して $[-0.2, 0.2]$ のランダムな実数を加える。交叉では二つの個体をランダムに選択し、一方の個体の入出力ニューロンからひとつをランダムに選択する。選択されたニューロンから m 経路長内にあるニューロンを近傍ニューロンとして定義し、近傍ニューロン間を結合重みを交叉相手の個体の結合重みと入れ替える。

4. モーション生成実験

開発したソフトウェアを用いて複雑な形状をもつ物理モデルを製作し、モーション生成実験を実施した。以下にモーション生成実験用に製作したモデルと生成されたモーションについて記述する。

4.1 製作した物理モデル

実験用に二種類のモデルを製作した。図 8 に製作した物理モデルの概観を、表 4 に各物理モデルの情報を示す。

各モデルはメッシュプリミティブにより構成されており、目の位置に光センサの属性を付加した球プリミティブを設置した。

静摩擦係数及び動摩擦係数は、胴体と足で異なる値を採用し、モデルが主に足と地面の摩擦力を利用した歩行行動を獲得するようにモデルを設計した。各プリミティブは固定ジョイント(図 8・緑線)及び回転の 3 自由度を持つアクチュエータ(図 8・赤線)により結合し、ANN にはリカレント構造を持たない中間層を二層もつフィードフォワードネットワークを採用した。

4.2 モーション生成結果

4.1 に示したモデルを用いてモーション生成実験を実施した。100 世代の進化計算の後に各モデルが獲得したモーションの一例¹³⁾を図 9~10 に示す。(a)にストロボ撮影したモーション、(b)にモデルの重心が通った軌跡を示す。また、各モデルのアクチュエータへの制御量の時間推移を図 11~12 に示す。

4.3 考察

実験用に作成したモデルは図 9~10 に示した軌跡を辿り、光源に近づく行動を獲得した。Salamander モデルは図 11(b)に示すように、前足と後足のアクチュエータが逆の位相を持つ制御力を

Table 4 Data of physics models

	Salamander model	Gold beetle model
Geometric data		
Number of primitives	20	26
Number of sensors	2	2
Number of actuators	13	12
Physical data		
Density	1000 kg/m ³	1000 kg/m ³
Restitution coefficient	0.1	0.1
Static friction coefficient	body: 0.3 limb: 0.9	body: 0.3 limb: 0.9
Dynamic friction coefficient	body: 0.2 limb: 0.8	body: 0.2 limb: 0.8
ANN data		
Number of neurons	420	588
Type of ANN	Feed Forward	Feed Forward
Number of hidden layer	2	2

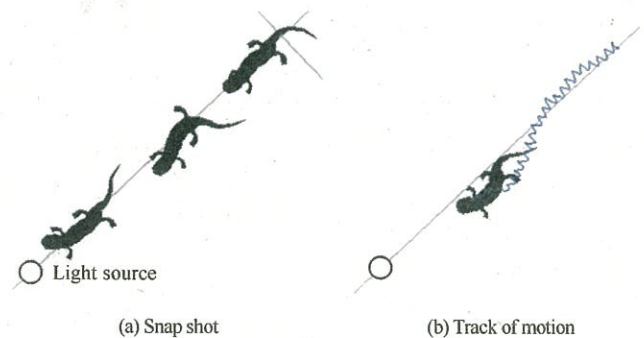


Fig.9 Motion of salamander model

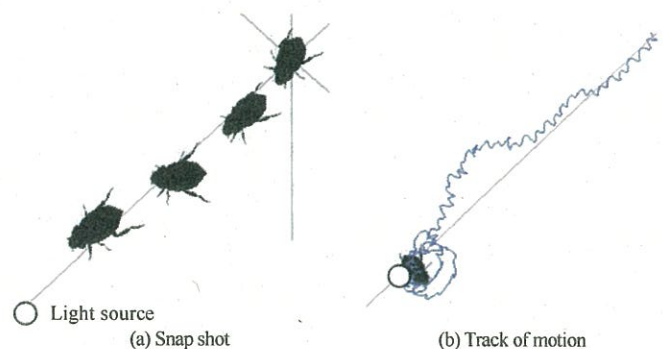


Fig.10 Motion of gold beetle model

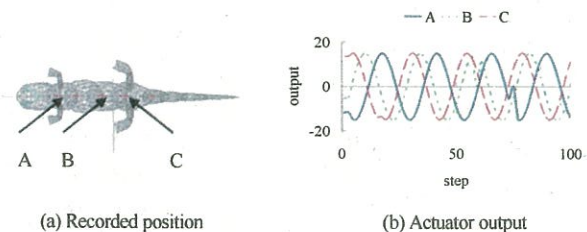


Fig.11 Actuator outputs of salamander model

獲得し、振動を体の前方から後方に伝達することで生じる体のうねりを利用して光源に向かって前進する。一方、Gold beetle モデルは図 12(b)に示すように、前足、中足、後足の対がそれぞれ左右のアクチュエータで逆の位相を持つ制御力を獲得し、6 本の足

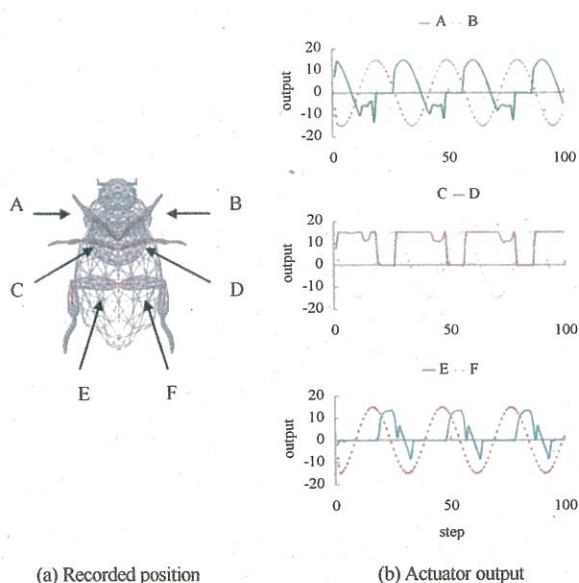


Fig.12 Actuator outputs of gold beetle model

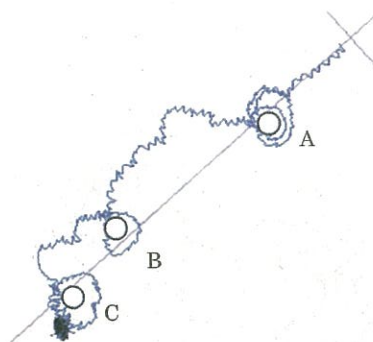


Fig.13 Pursuit behavior of gold beetle model

を動かすことで光源に向かって前進しており、光源に到達後、その周辺を周回する行動が観察された。

また、最適化された ANN を用いて、図 13 に示すように、光源の位置を A, B, C の順に動的に移動させてシミュレーションを実施した場合、モデルが夫々の位置に対して追従性を示した。

以上より、最適化アルゴリズムの評価関数に生物が持つ基本的な行動原理を導入することで各々のモデルが仮想物理環境内で自らの形状に適応したモーションを自律的に獲得可能であることが明らかとなった。獲得されたモーションは実際の生物の運動に対して出力の位相が良く一致し、アニメーションとして

用いる上で十分に使用可能であると考えられる。

このモーション獲得には、摩擦係数、反発係数等のパラメータのチューニングが必要とされるが、インタラクティブなモデリングとシミュレーションを可能とした GUI により容易にパラメータのチューニングを行うことが可能であり、高校レベルの物理の知識で十分に本ソフトウェアによるモーション生成を行うことが可能である。

5. 結 言

- (1) 形状モデリングからコントローラを用いた自律的な行動の獲得までを相互的かつ円滑に実現する物理モデリングソフトウェアを開発した。
- (2) 複雑な形状を持つ物理モデルを製作し、行動獲得実験を実施することで、ソフトウェアの有用性を検証した。
- (3) 最適化アルゴリズムの評価関数に生物が持つ基本的な行動原理を導入することで、各々のモデルが仮想物理環境内で自らの形状に適応したモーションを自律的に獲得可能であることが明らかとなった。

参 考 文 献

- 1) J. J. Monaghan: An introduction to SPH, Computer Physics Communications, 48 (1988) 88.
- 2) E. Emerfeldt: Phun, <http://www.phunland.com/wiki/Home>
- 3) K. Sims: Evolving Virtual Creatures, Computer Graphics (SIGGRAPH '94), 7 (1994) 15.
- 4) D. Terzopoulos et al.: Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World, Artificial Life, 1, 4 (1994) 327.
- 5) J. Wu et al.: Realistic Modeling of Bird Flight Animations, ACM Transactions on Graphics (SIGGRAPH 2003), 22, 3 (2003) 888.
- 6) nVidia: PhysX, http://www.nvidia.com/object/nvidia_physx.html
- 7) S. Koshizuka, A. Nobe, and Y. Oka: Numerical Analysis of Breaking Waves Using the Moving Particle Semi-Implicit Method. Numerical Methods in Fluids, 26, 7 (1998) 751.
- 8) ASSIST: http://rationale.csail.mit.edu/project_assist.shtml
- 9) A.J., Ijspeert, and J. Kodjabachian: Evolution and Development of a Central Pattern Generator for the Swimming of a Lamprey, Artificial Life 5, 3 (1999) 247.
- 10) M., Masry, and H., Lipson: A Sketch-Based Interface for Iterative Design and Analysis of 3D Objects Proceedings of Eurographics workshop on Sketch-Based Interfaces, Dublin, Ireland, (2005) 109.
- 11) L.N., Patel, A., Murray, and J., Hallam: Increased Swimming Control with Evolved Lamprey CPG Controllers, Neural Networks, IJCNN '05, 4 (2005).
- 12) Hokkaido University, Laboratory of Autonomous System Engineering: <http://junji.complex.eng.hokudai.ac.jp/researches/physics-modeling/tool/>
- 13) Hokkaido University, Laboratory of Autonomous System Engineering: <http://junji.complex.eng.hokudai.ac.jp/researches/physics-modeling/movies/>