

# EVOLUÇÃO DO CAMINHAR E DA MORFOLOGIA DE ROBÔS TETRAPOD

MILTON ROBERTO HEINEN\*, FERNANDO SANTOS OSÓRIO†, PAULO MARTINS ENGEL\*

\* *Universidade Federal do Rio Grande do Sul (UFRGS)*  
*Instituto de Informática – Porto Alegre, RS, Brasil*

† *Universidade do Vale do Rio dos Sinos (UNISINOS)*  
*Computação Aplicada – São Leopoldo, RS, Brasil*

Emails: mrheinen@inf.ufgrs.br, fosorio@unisinis.br, engel@inf.ufgrs.br

**Abstract**— This paper describes our experiments with autonomous robots, in which we use neural networks to generate and control stable gaits of simulated legged robots into a physically based simulation environment. In our approach, the gait is accomplished using an Elman network, and the synaptic weights are evolved using genetic algorithms. The proposed model also allow the evolution of robots morphology together with the control parameters. The model validation was performed by several experiments, and the results showed that it is possible to generate stable gaits using neural networks in an efficient manner.

**Keywords**— Intelligent Robots, Legged Robots, Genetic Algorithms, Artificial Neural Networks

## 1 Introdução

Os robôs móveis autônomos vem atuando em diferentes áreas, como desarmamento de bombas, exploração de ambientes hostis, e a condução de veículos de forma autônoma (Heinen et al., 2006). A maioria dos robôs móveis desenvolvidos até o momento se deslocam através de rodas, o que facilita o controle, mas impede que eles sejam capazes de se deslocarem em ambientes irregulares que possuam desníveis e degraus (Knight and Nehzow, 2002). Assim, para que um robô móvel possa se deslocar livremente em ambientes irregulares, ele precisa do mesmo mecanismo de locomoção utilizado pelos seres humanos, ou seja, ele precisa ser dotado de pernas (Bekey, 2005).

Entretanto, o desenvolvimento de robôs com pernas que consigam se deslocar livremente em ambientes irregulares é uma tarefa bastante árdua, que exige a configuração de diversos parâmetros relativos ao caminhar. A configuração manual destes parâmetros exige muitas horas de um especialista humano, e os resultados obtidos são sub-ótimos e dependentes da arquitetura específica do robô (Chernova and Veloso, 2004). Desta forma, é bastante útil realizar a configuração do caminhar de forma automática, através do uso de técnicas de aprendizado de máquina (*machine learning* – ML) (Mitchell, 1997).

Em nossos trabalhos anteriores (Heinen and Osório, 2006a), o controle do caminhar foi realizado através de um autômato finito cujos parâmetros forma evoluídos através de algoritmos genéticos (*genetic algorithms* – GA). Também foi realizado um estudo comparativo entre diferentes modelos de robôs de quatro e seis pernas (Heinen and Osório, 2006d) e entre diferentes funções de *fitness* (Heinen and Osório, 2006b). Neste artigo, o controle do caminhar é realizado através de uma rede neural (Haykin, 2001) cujos pesos sinápticos são evoluídos através de GAs. Além disto, a mor-

fologia do robô é evoluída em conjunto com os pesos da rede neural.

Este artigo está estruturado da seguinte forma: a Seção 2 descreve o uso de simulação baseada em física e a biblioteca *Open Dynamics Engine* (ODE); a Seção 3 descreve diversos trabalhos do estado da arte na área em questão; a Seção 4 descreve a área de vida artificial, que possui uma certa relação com o problema em questão; a Seção 5 descreve o modelo proposto, bem como protótipo implementado para a validação do mesmo; a Seção 6 descreve os experimentos realizados e os resultados obtidos; Por último, a Seção 7 traz as conclusões finais e as perspectivas futuras.

## 2 Simulação de robôs móveis

Quando se deseja realizar experimentos em robótica móvel, duas alternativas são possíveis (Pfeifer and Scheier, 1999): (i) realizar os experimentos diretamente em um robô real; ou (ii) realizar os experimentos utilizando um robô simulado em um ambiente virtual realista. A utilização de um robô real possui a vantagem de tornar reais os resultados obtidos, mas o uso de simulação possui as seguintes vantagens (Law and Kelton, 2000):

- Não existe o risco de se danificar o robô;
- A troca ou recarga de baterias e a manutenção do robô não são necessárias;
- O reposicionamento do robô pode ser realizado sem a intervenção humana;
- O relógio da simulação pode ser acelerado, reduzindo o tempo de aprendizado;
- Pode-se testar várias arquiteturas e modelos de robôs antes da construção física.

Por estes motivos, optou-se por realizar os experimentos utilizando robôs simulados em um ambiente virtual que implementa as leis da física através da biblioteca ODE, que é uma biblioteca de software que permite a realização de simula-

ções da dinâmica de corpos rígidos articulados com bastante realismo.

### 2.1 Simulação baseada em física

Para que uma simulação de robôs móveis seja realista, diversos elementos do mundo real precisam estar presentes no modelo de simulação, para que os corpos se comportem de forma similar à realidade. Em especial, é necessário que um robô sofra quedas se não for bem controlado ou se não estiver bem posicionado, e que colida contra os objetos do ambiente de forma realista. Para que isto ocorra, é necessário que as leis da física sejam modeladas no ambiente de simulação (gravidade, inércia, fricção e colisão) (Osório et al., 2006). Atualmente existem várias bibliotecas de software disponíveis para a implementação de simulações baseadas em física. Após o estudo de diversas possibilidades, optou-se pela utilização de uma biblioteca de código aberto e gratuita chamada *Open Dynamics Engine* (ODE)<sup>1</sup>, que permite a realização de simulações da dinâmica de corpos rígidos articulados com bastante realismo. Utilizando a ODE, é possível criar diversos corpos rígidos, e estes podem ser conectados através de juntas de vários tipos. Para a movimentação das juntas, é possível que sejam aplicadas forças diretamente aos corpos, ou podem ser utilizados os motores angulares que estão disponíveis no ambiente ODE. Um motor angular é um elemento que pode ser conectado a dois corpos articulados, e que possui uma velocidade e uma força máxima. Com o uso de juntas e motores angulares, é possível que sejam reproduzidas as diversas articulações presentes em um robô real, em seres humanos e nos animais com um alto nível de precisão (Osório et al., 2006).

## 3 Trabalhos relacionados

Na área de controle inteligente de robôs com pernas, os primeiros trabalhos datam do final dos anos 80 e início dos anos 90, como por exemplo o trabalho de Lewis (Lewis et al., 1992), que utilizou algoritmos genéticos para a evolução dos controladores de um robô de seis pernas (*hexapod*). Neste trabalho, o controlador foi evoluído em um robô cujo caminhar era inspirado no caminhar dos insetos. Através de vários estágios de evolução, seu comportamento foi sendo modificado até atingir um caminhar razoavelmente satisfatório. Bongard (Bongard and Pfeifer, 2002) evoluiu os parâmetros de uma rede neural artificial dinâmica utilizada para controlar diversos tipos de robôs simulados. Busch (Busch et al., 2002) utilizou Programação Genética para evoluir os parâmetros de controle de diversos tipos de robôs, simulados utilizando o pacote de software DynaMechs<sup>2</sup>. Jacob (Jacob et

al., 2005) utilizou aprendizado por reforço para o controle de um robô de quatro pernas (*tetrapod*) simulado através da biblioteca de software ODE. Reeve (Reeve and Hallam, 2005) utilizou algoritmos genéticos para a evolução dos parâmetros de diversos modelos de redes neurais utilizadas para o controle de diversos *tetrapods* simulados utilizando o DynaMechs.

Na maioria das abordagens descritas acima, a função de *fitness* utilizada foi a distância percorrida pelo robô durante um certo período de tempo. Embora esta função de *fitness* seja largamente utilizada, ela pode fazer com que a evolução privilegie formas de caminhar pouco estáveis em detrimento de soluções um pouco mais lentas porém muito mais estáveis (Golubovic and Hu, 2003). Em nossos estudos, além da distância percorrida pelo robô, foram utilizadas como critério de *fitness* informações sensoriais, provenientes de um giroscópio simulado, a fim de se garantir que os caminhares obtidos fossem tanto rápidos quanto estáveis (Heinen and Osório, 2006b).

## 4 Vida artificial

Vida artificial (*artificial life* - ALife) é o nome dado à disciplina que estuda a vida natural através da tentativa de recriar fenômenos biológicos em computadores ou outros meios artificiais (Langton, 1995). Complementa a abordagem analítica tradicional da biologia com uma abordagem sintética onde, ao invés de estudar os fenômenos biológicos através de ver como funcionam os organismos vivos já constituídos, cria um sistema que se comporta como um organismo vivo (Levy, 1992; Pfeifer and Scheier, 1999). As tentativas de recriar os fenômenos biológicos de maneira artificial podem resultar não só na melhor compreensão teórica dos fenômenos estudados, como também em aplicações práticas dos princípios biológicos nas áreas de robótica, medicina, nanotecnologia e engenharia.

Um dos pioneiros na área de vida artificial foi Karl Sims, que em (Sims, 1994a; Sims, 1994b) utilizou um ambiente tridimensional baseado em física para a evolução de criaturas virtuais. Nestas criaturas, o sistema de controle foi evoluído em conjunto com a morfologia, o que aumenta o espaço de busca, dificulta a evolução e exige um maior poder computacional. Segundo (Pfeifer and Scheier, 1999), para manter a complexidade em níveis aceitáveis, são necessárias restrições no modelo. Nas criaturas de Karl Sims, uma das restrições impostas é o formato dos segmentos – as criaturas são constituídas de corpos rígidos de tamanhos variados. Um modelo sem restrições dificilmente irá convergir para uma solução aceitável. Já em um modelo com muitas restrições haverá menos variabilidade nas criaturas evoluídas (Pfeifer and Scheier, 1999).

<sup>1</sup>ODE – <http://www.ode.org>

<sup>2</sup>DynaMechs – <http://dynamechs.sourceforge.net/>

## 5 Modelo proposto

Esta seção descreve o simulador LegGen<sup>3</sup>, (Heinen and Osório, 2006a), proposto e implementado para o controle do caminhar de robôs com pernas. Este simulador é composto de diversos módulos, mostrados na Figura 1.

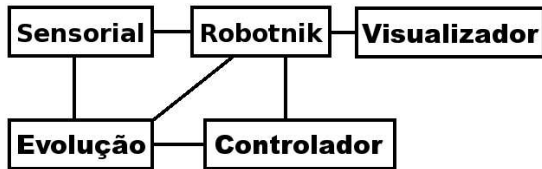


Figura 1: Módulos do LegGen

O módulo *Robotnik* é responsável pela criação do robô e do ambiente virtual através do uso da biblioteca ODE. O módulo *Evolução* é responsável pela evolução dos parâmetros do controlador através de algoritmos genéticos. O módulo *Sensorial* é responsável pela leitura de informações sensoriais durante a simulação, utilizadas no cálculo do *fitness* de cada indivíduo. O módulo *Visualizador* é responsável pela visualização dos resultados em um ambiente gráfico tridimensional. O módulo *Controlador*, implementado através de uma rede neural, é responsável pelo controle das juntas do robô durante o caminhar. As próximas seções descrevem vários elementos do modelo proposto.

### 5.1 Controle das juntas do robô

O módulo *Controlador* do sistema LegGen realiza o controle das juntas do robô através de redes neurais artificiais (*artificial neural networks* – ANN) (Haykin, 2001). Esta abordagem possui uma limitação: não é possível se obter de antemão informações locais para o cálculo do gradiente e a correção dos erros, o que limita a utilização dos algoritmos de aprendizado supervisionado tradicionais (*back-propagation* e similares) (Heinen and Osório, 2006c). Por isto, foram utilizados algoritmos genéticos para a evolução dos pesos sinápticos. A vantagem de se utilizar GAs para o ajuste dos pesos é que eles não necessitam de informações locais para a correção dos erros, ou seja, eles não necessitam de uma base de treinamento. As principais vantagens de se utilizar uma ANN no controle do caminhar são (Haykin, 2001): (i) são robustas em relação a situações novas e inesperadas; (ii) possuem um alto grau de generalização.

Como entradas da ANN, foram utilizados os ângulos atuais das juntas do robô, normalizados entre -1 ( $\alpha_{mim}$ ) e 1 ( $\alpha_{max}$ ). Na saída da ANN, são obtidos os ângulos desejados para as juntas no instante  $t+1$ , normalizados entre -1 e 1. Após alguns testes preliminares, optou-se por utilizar as redes neurais recorrentes do tipo Elman (Elman, 1990),

que são redes neurais do tipo *multi layer perceptron* (MLP) que possuem conexões de realimentação (*feedback*) na camada oculta. Estas conexões permitem que as redes de Elman aprendam a reconhecer e gerar padrões temporais. A função de ativação utilizada foi a tangente hiperbólica. O intervalo de valores possíveis para os pesos sinápticos foi limitado em  $[-1, 1]$ , que se mostrou bastante adequado para o problema em questão.

### 5.2 Evolução dos parâmetros de controle

A evolução dos parâmetros de controle (pesos sinápticos da rede neural) no sistema LegGen foi realizada através de algoritmos genéticos. Para a implementação do GA, foi selecionada a biblioteca de software GALib<sup>4</sup>, desenvolvida por Matthew Wall do Massachusetts Institute of Technology - MIT. A GALib foi selecionada por ser uma das mais completas, eficientes conhecidas bibliotecas de software para a simulação de algoritmos genéticos, e também por ser uma solução gratuita e de código aberto, baseada na linguagem C++.

No sistema desenvolvido, foi utilizado um algoritmo genético com populações sobrepostas (Goldberg, 1989), e foram adotados genomas do tipo real. O tipo de cruzamento escolhido foi o cruzamento em um ponto, e o esquema de seleção adotado foi o *stochastic remainder sampling selector*, que segundo Goldberg (Goldberg, 1989) possui um desempenho superior ao esquema da roleta (*roulette wheel*). O método de escala do *fitness* utilizado foi o *sigma truncation* (Goldberg, 1989), que permite que o *fitness* assumam valores negativos. A Tabela 1 mostra os parâmetros do algoritmo genético utilizados nas simulações.

Tabela 1: Parâmetros do GA

| Parâmetro            | Valor |
|----------------------|-------|
| Taxa de cruzamentos  | 0,80  |
| Taxa de mutação      | 0,08  |
| Tamanho da população | 350   |
| Número de gerações   | 700   |

A função de *fitness*  $F$  utilizada pelo GA é calculada através da fórmula:

$$F = \frac{D}{1 + B + a \times G} \quad (1)$$

onde  $D$  é a distância percorrida pelo robô em relação ao eixo  $x$ ,  $B$  é o índice dos *bumpers*,  $G$  é a taxa de instabilidade, e  $a$  é uma constante que serve para alterar a influência de  $G$  na função de *fitness*. Nos experimentos realizados, foi utilizado  $a = 10$ . O  $B$  é calculado através da equação:

$$B = \sum_{i=1}^P \left( \frac{n_i}{N} - \frac{1}{2} \right)^2 \quad (2)$$

<sup>3</sup>LegGen – <http://www.inf.unisinos.br/~osorio/leggen>

<sup>4</sup>GALib – <http://www.lancet.mit.edu/ga/>

onde  $P$  é o número de *endpoints* (patas),  $n_i$  é a quantidade de amostras sensoriais nas quais o *endpoint*  $i$  estava em contato com o solo, e  $N$  é o número total de leituras sensoriais realizadas. Nesta função de *fitness*, o valor de  $B$  tenderá a zero quando o robô mantiver as patas no chão por aproximadamente 50% do tempo, que é o comportamento desejado durante o caminhar. Já se o robô mantiver todas as patas no chão durante o período de simulação, o valor de  $B$  será igual a 1. O mesmo ocorrerá se o robô mantiver todas as patas no ar durante o período de simulação.

Para o cálculo da taxa de instabilidade, várias leituras de um giroscópio simulado são realizadas durante a simulação, e ao final da mesma é calculada a taxa de instabilidade do robô  $G$  (Gyro) (Golubovic and Hu, 2003):

$$G = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2 + \sum_{i=1}^N (y_i - \bar{y})^2 + \sum_{i=1}^N (z_i - \bar{z})^2}{N}} \quad (3)$$

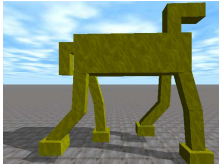
onde  $N$  é o número de amostras coletadas,  $x_i$ ,  $y_i$  e  $z_i$  são os dados coletados pelo giroscópio simulado no tempo  $i$ , e  $\bar{x}$ ,  $\bar{y}$  e  $\bar{z}$  são as médias das leituras do giroscópio, calculadas através das equações:

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N}, \quad \bar{y} = \frac{\sum_{i=1}^N y_i}{N}, \quad \bar{z} = \frac{\sum_{i=1}^N z_i}{N} \quad (4)$$

Utilizando esta função de *fitness*, o indivíduo mais bem qualificado é aquele que possui a melhor relação entre velocidade e estabilidade, de forma que as melhores soluções são aquelas mantêm o compromisso entre os dois critérios de avaliação.

### 5.3 Robô modelado

Conforme consta em sua documentação, a biblioteca ODE possui uma complexidade computacional de ordem  $O(n^2)$ , onde  $n$  é o número de corpos presentes no mundo físico simulado. Deste modo, para manter a velocidade da simulação em um nível aceitável, é preciso modelar os corpos da forma mais simples possível. Por este motivo, todos o robô simulado foi modelado com objetos simples, como retângulos e cilindros, e possui apenas as articulações necessárias para a tarefa de caminhar. Para manter o projeto do robô simples, as juntas utilizadas nos membros se movimentam apenas em torno do eixo  $z$  em relação ao robô (o mesmo movimento do nosso joelho), pois as simulações realizadas até o momento foram todas com o robô a caminhando em linha reta. No futuro, o modelo será estendido para aceitar modelos de robôs com juntas mais complexas. Inicialmente foram modelados e testados diversos tipos de robôs, até que se chegou ao modelo final, mostrado na Figura 2. A dimensões deste robô são aproximadamente as de um cachorro.



| Parte  | Dimensões |        |        |
|--------|-----------|--------|--------|
|        | $x$       | $y$    | $z$    |
| Corpo  | 45,0cm    | 15,0cm | 25,0cm |
| Coxa   | 5,0cm     | 15,0cm | 5,0cm  |
| Canela | 5,0cm     | 15,0cm | 5,0cm  |
| Pata   | 8,0cm     | 5,0cm  | 9,0cm  |

Figura 2: Robô modelado

### 5.4 Evolução da morfologia

Segundo Pfeifer (Pfeifer and Scheier, 1999), na natureza a evolução do controle (sistema nervoso) não ocorre após a morfologia do corpo estar completa. Pelo contrário, este é um processo que ocorre em conjunto ao longo da evolução. Esta estratégia é muito utilizada na área de vida artificial (Sims, 1994a; Sims, 1994b; Eggenberger, 1997).

Na seção anterior, foi descrito o modelo de robô a ser utilizado nos experimentos. Este robô foi modelado de forma empírica, inspirado em animais de quatro patas, com algumas simplificações. Ao se realizar a co-evolução da morfologia e dos parâmetros de controle, é possível que sejam descobertos novos modelos de robôs, sem equivalentes na natureza, que se mostrem mais eficientes na tarefa em questão (Pfeifer and Scheier, 1999).

Assim, o simulador LegGen original foi estendido, de forma que permitisse a evolução da morfologia em conjunto com os parâmetros de controle do robô. Para que isto fosse possível, novos genes foram incluídos no genoma original do algoritmo genético. Cada segmento do robô foi codificado utilizando três valores (dimensões em  $x$ ,  $y$  e  $z$ ).

## 6 Resultados

Esta seção descreve os experimentos realizados visando avaliar o desempenho do controlador neural: (i) utilizando o modelo de robô descrito na Seção 5.3; (ii) evoluindo a morfologia do robô em conjunto com os parâmetros de controle. A rede neural utilizada nos experimentos possui 12 entradas (uma para cada junta), 3 neurônios ocultos e 12 saídas (uma para cada motor), totalizando 96 pesos sinápticos. A camada oculta é totalmente conectada de forma recorrente entre si, seguindo a arquitetura Elman padrão. Para cada conjunto de testes, foram realizados 10 experimentos distintos, e ao final foi calculada a média e o desvio padrão dos resultados. A Tabela 2 mostra os resultados obtidos nestes experimentos, que demoraram 74,61 horas (3,11 dias).

A primeira coluna (**E**) representa o índice do experimento. As demais colunas representam, respectivamente, os resultados obtidos utilizando o modelo de robô ilustrado na Figura 2 (coluna “Controle”) e a evolução da morfologia em conjunto com os parâmetros de controle (coluna “Morfologia”). As sub-colunas representam, res-

Tabela 2: Resultados obtidos nos experimentos

| E        | Controle    |             |             | Morfologia  |             |             |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|
|          | <i>F</i>    | <i>D</i>    | <i>G</i>    | <i>F</i>    | <i>D</i>    | <i>G</i>    |
| 1        | 16,3        | 29,2        | 0,08        | 18,8        | 38,0        | 0,10        |
| 2        | 16,6        | 28,3        | 0,07        | 17,9        | 33,0        | 0,08        |
| 3        | 17,0        | 27,8        | 0,06        | 19,8        | 39,5        | 0,10        |
| 4        | 16,7        | 27,9        | 0,07        | 17,8        | 37,9        | 0,11        |
| 5        | 16,2        | 28,2        | 0,07        | 20,1        | 27,4        | 0,03        |
| 6        | 16,0        | 31,1        | 0,09        | 15,9        | 32,8        | 0,10        |
| 7        | 17,3        | 29,6        | 0,07        | 18,9        | 41,1        | 0,12        |
| 8        | 16,6        | 29,0        | 0,07        | 18,5        | 36,2        | 0,10        |
| 9        | 16,3        | 30,2        | 0,08        | 19,1        | 39,2        | 0,10        |
| 10       | 16,2        | 29,8        | 0,08        | 15,6        | 37,4        | 0,14        |
| $\mu$    | <b>16,5</b> | <b>29,1</b> | <b>0,08</b> | <b>18,2</b> | <b>36,2</b> | <b>0,10</b> |
| $\sigma$ | <b>0,4</b>  | <b>1,1</b>  | <b>0,01</b> | <b>1,5</b>  | <b>4,1</b>  | <b>0,03</b> |

pectivamente, o *fitness* (*F*), a distância percorrida pelo robô (*D*) em 30 segundos, o índice dos *bumpers* (*B*) e a taxa de instabilidade (*G*). As duas últimas linhas da tabela trazem a média ( $\mu$ ) e o desvio padrão ( $\sigma$ ) dos valores de cada coluna. A Figura 3 mostra os gráficos de *boxplot* (Law and Kelton, 2000) e do intervalo de confiança (*confidence interval* – CI) a 95%, relativos aos valores de *fitness* obtidos nos experimentos realizados.

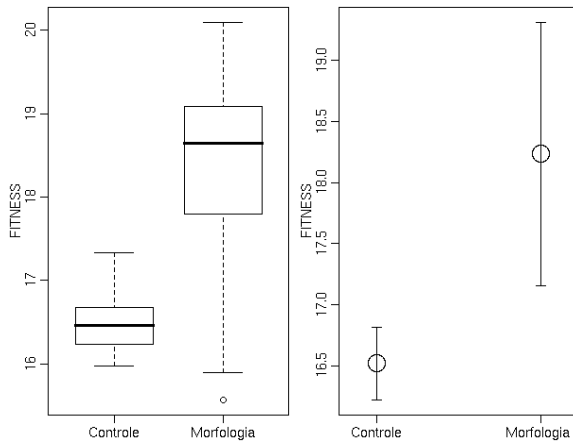


Figura 3: Gráfico de *boxplot* e CI

Percebe-se claramente que a evolução da morfologia em conjunto com os parâmetros de controle produziu melhores resultados do que a evolução dos parâmetros de controle de forma isolada. Assim, a evolução da morfologia pode ser considerada uma ferramenta bastante útil no projeto de robôs mais eficientes.

A título de exemplificação, a Figura 4 mostra um exemplo de caminhar realizado pelo robô da Figura 2, e a Figura 5 mostra o caminhar de robô cuja morfologia foi evoluída. A Figura 6 mostra as morfologias evoluídas em cada um dos experimentos da Tabela 2 (os números do canto superior esquerdo se referem ao número do experimento). Observando a Figura 6, percebe-se que o espaço de estados permite a evolução de diversas soluções

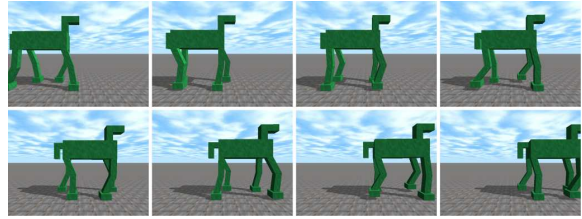


Figura 4: Caminhar do robô modelado

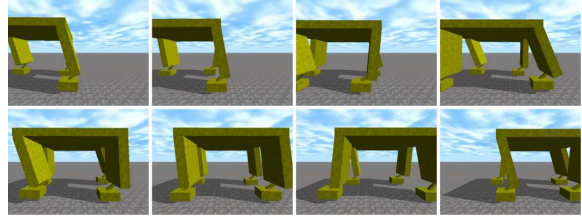


Figura 5: Robô evoluído no experimento 06

diferentes entre si, porém igualmente eficientes, a exemplo do que ocorre na natureza.

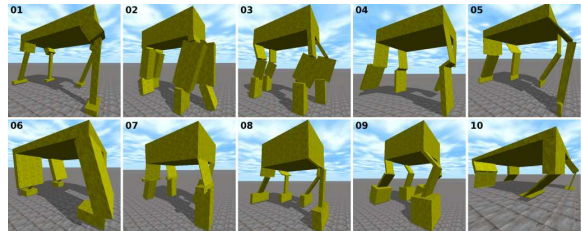


Figura 6: Morfologias evoluídas

## 7 Conclusões e perspectivas

Através dos experimentos realizados utilizando o simulador LegGen, foi possível concluir que: (i) as redes neurais artificiais são bastante eficientes para o controle das juntas do robô; (ii) o uso de algoritmos genéticos para a evolução dos pesos sinápticos surge como uma alternativa bastante atrativa de se resolver o problema da falta de informações locais para a correção dos erros e do cálculo do gradiente; (iii) a evolução da morfologia do robô em conjunto com os parâmetros de controle permite que sejam descobertos modelos de robôs mais eficientes do que os modelos projetados manualmente. As perspectivas futuras incluem tornar o caminhar possível em superfícies irregulares e a subida de escadas, bem como a construção de um robô físico conforme as especificações de um dos melhores modelos aprendidos, para assim validar o modelo em condições reais.

## Referências

Bekey, G. A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*, MIT Press, Cambridge, MA.

- Bongard, J. C. and Pfeifer, R. (2002). A method for isolating morphological effects on evolved behaviour, *Proc. 7th Int. Conf. Simulation of Adaptive Behaviour (SAB)*, MIT Press, Edinburgh, UK, pp. 305–311.
- Busch, J., Ziegler, J., Aue, C., Ross, A., Sawitzki, D. and Banzhaf, W. (2002). Automatic generation of control programs for walking robots using genetic programming, *Proc. 5th European Conf. Genetic Programming (EuroGP)*, Vol. 2278 of *LNCS*, Springer-Verlag, Kinsale, Ireland, pp. 258–267.
- Chernova, S. and Veloso, M. (2004). An evolutionary approach to gait learning for four-legged robots, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sendai, Japan.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3d organisms based on differential gene expression, *Proc. Fourth European Conf. Artificial Life (ECAL'97)*, MIT Press, Cambridge, MA, pp. 205–213.
- Elman, J. L. (1990). Finding structure in time, *Cognitive Science* **14**: 179–211.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
- Golubovic, D. and Hu, H. (2003). Ga-based gait generation of sony quadruped robots, *Proc. 3th IASTED Int. Conf. Artificial Intelligence and Applications (AIA)*, Benalmadena, Spain.
- Haykin, S. (2001). *Redes Neurais: Princípios e Prática*, 2 edn, Bookman, Porto Alegre, RS.
- Heinen, M. R. and Osório, F. S. (2006a). Applying genetic algorithms to control gait of physically based simulated robots, *Proc. IEEE Congr. Evolutionary Computation (CEC)*, Vancouver, Canada.
- Heinen, M. R. and Osório, F. S. (2006b). Gait control generation for physically based simulated robots using genetic algorithms, *Proc. Int. Joint Conf. 2006, 10th Ibero-American Conference on AI (IBERAMIA), 18th Brazilian Symposium on AI (SBIA)*, LNCS, Springer-Verlag, Ribeirão Preto - SP, Brazil.
- Heinen, M. R. and Osório, F. S. (2006c). Neural networks applied to gait control of physically based simulated robots, *Proc. Int. Joint Conf. 2006, 9th Brazilian Neural Networks Symposium (SBRN)*, Ribeirão Preto, SP, Brazil.
- Heinen, M. R. and Osório, F. S. (2006d). Uso de algoritmos genéticos para a configuração automática do caminhar em robôs móveis, *Anais do Encontro de Robótica Inteligente (EnRI)*, Campo grande, MS, Brazil.
- Heinen, M. R., Osório, F. S., Heinen, F. J. and Kelber, C. (2006). SEVA3D: Using artificial neural networks to autonomous vehicle parking control, *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Vancouver, Canada.
- Jacob, D., Polani, D. and Nehaniv, C. L. (2005). Legs than can walk: Embodiment-based modular reinforcement learning applied, *Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Espoo, Finland, pp. 365–372.
- Knight, R. and Nehmzow, U. (2002). Walking robots - a survey and a research proposal, *Technical Report CSM-375*, Univ. Essex, UK.
- Langton, C. (1995). *Artificial Life: An Overview*, MIT Press, Cambridge, MA.
- Law, A. M. and Kelton, D. W. (2000). *Simulation Modeling and Analysis*, McGraw-Hill, New York.
- Levy, S. (1992). *Artificial Life: A Report From the Frontier Where Computers Meet Biology*, Vintage books, New York.
- Lewis, M. A., Fagg, A. H. and Solidum, A. (1992). Genetic programming approach to the construction of a neural network for control of a walking robot, *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Nice, France, pp. 2618–2623.
- Mitchell, T. (1997). *Machine Learning*, McGraw-Hill, New York.
- Osório, F. S., Musse, S. R., Vieira, R., Heinen, M. R. and Paiva, D. C. (2006). *Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation*, Vol. 2, Springer-Verlag, Berlin, Germany, pp. 1–45.
- Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*, MIT Press, Cambridge, MA.
- Reeve, R. and Hallam, J. (2005). An analysis of neural models for walking control, *IEEE Trans. Neural Networks* **16**(3): 733–742.
- Sims, K. (1994a). Evolving 3d morphology and behavior by competition, *Artificial Life IV Proceedings*, MIT Press, Cambridge, MA, pp. 28–39.
- Sims, K. (1994b). Evolving virtual creatures, *Computer Graphics* **28**: 15–24.