

Controle Inteligente do Caminhar de Robôs Móveis Simulados

Milton Roberto Heinen¹ e Fernando Santos Osório²

¹ UFRGS, Instituto de Informática, CEP 91501-970, Porto Alegre, RS, Brasil

² USP, ICMC, CEP 13560-970, São Carlos, SP, Brasil
mrheinen@inf.ufrgs.br, fosorio@icmc.usp.br

Resumo Este artigo descreve o simulador LegGen, que realiza a configuração automática do caminhar em robôs com pernas simulados com a biblioteca *Open Dynamics Engine* (ODE). Este simulador utiliza redes neurais artificiais do tipo Elman para o controle das juntas dos robôs e algoritmos genéticos para a evolução dos pesos sinápticos das mesmas. Além disto, o simulador permite a evolução da morfologia do robô em conjunto com os parâmetros de controle, o que torna possível a seleção de modelos de robôs bastante eficientes. o modelo proposto foi validado através da realização de diversos experimentos estatisticamente válidos, que demonstraram que o modelo proposto é eficiente tanto no controle das juntas do robô quanto na evolução da morfologia.

Student level: MSc (Mestrado)

Date of Conclusion: 20/02/2007

URL: http://bdt.d.unisinos.br/tde_busca/arquivo.php?codArquivo=380

1 Introdução

Os robôs móveis autônomos tem atraído a atenção de um grande número de pesquisadores, devido ao desafio que este novo domínio de pesquisas propõe: dotar sistemas de uma capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos [1]. Atualmente os robôs móveis atuam em diferentes áreas, como desarmamento de bombas, exploração de ambientes hostis, e a condução de veículos de forma semi-autônoma. A maioria dos robôs móveis desenvolvidos até o momento se deslocam através do uso de rodas, o que facilita bastante o controle, mas impede que eles sejam capazes de se deslocarem em ambientes irregulares que possuam desníveis e degraus [2]. Assim, para que um robô móvel possa se deslocar em ambientes irregulares, ele precisaria ser dotado do mesmo mecanismo de locomoção utilizado pelos seres humanos e a um grande número de seres vivos, ou seja, ele precisaria de pernas ou articulações [3,4]. Mas 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 [5].

Desta forma, seria bastante útil realizar a configuração do caminhar de forma automática, através da utilização de técnicas de aprendizado de máquina [6].

O objetivo deste artigo é descrever o simulador LegGen [7,8,9,10], que é um simulador capaz de realizar a configuração do caminhar de robôs com pernas de forma automática através do uso de técnicas de aprendizado de máquina. Neste simulador, o controle das juntas do robô é realizado através de redes neurais artificiais (*artificial neural networks* – ANN) [11] cujos pesos sinápticos são evoluídos através de algoritmos genéticos (*genetic algorithms* – GA). Além disto, a morfologia do robô é evoluída em conjunto com os pesos da rede neural. As principais contribuições deste trabalho são: (i) o modelo de controle do caminhar proposto; (ii) a elaboração de uma função de *fitness* mais eficiente para a tarefa em questão; (iii) a realização de diversos experimentos visando determinar os melhores modelos de robôs e as formas de controle mais eficientes; (iv) a evolução da morfologia do robô; e (v) o desenvolvimento do simulador LegGen.

Este artigo está estruturado da seguinte forma: A Seção 2 descreve diversos trabalhos do estado da arte da área em questão; A Seção 3 descreve o uso de simulação baseada em física e a biblioteca ODE; A Seção 4 descreve o modelo proposto e o protótipo implementado; A Seção 5 descreve os experimentos realizados e os resultados obtidos; Por último, a Seção 6 traz as conclusões finais.

2 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 o trabalho de Lewis [12], 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 [13] evoluiu os parâmetros de uma rede neural artificial dinâmica utilizada para controlar diversos tipos de robôs simulados. Busch [14] utilizou programação genética para evoluir os parâmetros de controle de diversos tipos de robôs simulados. Jacob [15] utilizou aprendizado por reforço para o controle de um robô de quatro pernas (*tetrapod*) simulado através da biblioteca de software ODE. Reeve [16] 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 [17]. Em nossos estudos, além da distância percorrida pelo robô, foram utilizadas como critério de *fitness* informações sensoriais, provenientes de um giroscópio e *bumpers* simulados, a fim de se garantir que os caminhares obtidos fossem rápidos e estáveis [18,19].

3 Simulação de robôs móveis

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) [20,21,22]. 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)³, 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. 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 [23,20].

4 Modelo proposto

Esta seção descreve o simulador LegGen [1], proposto e implementado para a validação de diferentes estratégias de controle do caminhar de robôs com pernas. Ele foi implementado utilizando a linguagem de programação C++ e a biblioteca de software ODE, descrita anteriormente. Este simulador recebe como entrada dois arquivos, um descrevendo o formato e as dimensões do robô e o outro descrevendo os parâmetros de simulação. O funcionamento do simulador LegGen ocorre da seguinte forma: inicialmente o arquivo que descreve o robô é lido, e o robô é criado no ambiente virtual de acordo com as especificações presentes neste arquivo. Em seguida, os parâmetros do simulador LegGen são lidos, e o GA é inicializado e executado até que seja atingido o número de gerações desejado. A avaliação dos indivíduos ocorre da seguinte forma:

- O robô é colocado na orientação e na posição inicial do ambiente virtual;
- O cromossomo é lido, e a partir dele são configurados os pesos da rede neural;
- A simulação física é realizada por um tempo determinado (30 segundos);
- Durante a simulação física, são coletadas diversas informações sensoriais através de um giroscópio e *bumpers* simulados;
- O *fitness* é calculado e retornado para o GA.

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

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

³ ODE – <http://www.ode.org>

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 *bumpers* B é calculado através da equação:

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

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 B será igual a 1. O mesmo ocorrerá se o robô mantiver todas as patas no ar durante a 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) [17]:

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

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:

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

A Equação 1 foi definida, configurada e validada através da realização de diversos experimentos prévios [1], que comprovaram a sua superioridade em relação a outras funções de *fitness* na tarefa em questão. Utilizando esta função de *fitness*, o indivíduo mais bem qualificado é aquele que possui a melhor relação entre velocidade e estabilidade, pois: (i) um robô instável terá um baixo *fitness*, pois valores elevados de G funcionam como uma espécie de termo de “punição”; (ii) um robô que se deslocar muito pouco também terá um baixo *fitness* independente dos valores de B e G . Assim, os indivíduos mais bem avaliados serão aqueles que conseguirem manter um compromisso entre D , B e G , pois assim conseguirão se deslocar da forma rápida e estável.

4.1 Controle das juntas do robô

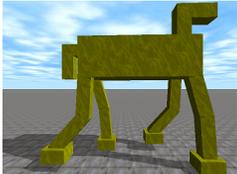
No simulador LegGen, o controle das juntas do robô é realizado através de redes neurais artificiais (*artificial neural networks* – ANN) [11]. Esta abordagem possui uma limitação: não é possível se obter de antemão informações locais para o cálculo do gradiente e a correção dos erros, o que limita a utilização dos algoritmos de aprendizado supervisionado tradicionais (*backpropagation* e similares) [24]. Por isto, foram utilizados GAs para a evolução dos pesos sinápticos.

As principais vantagens de se utilizar uma ANN são [11]: (i) são robustas em relação a situações novas e inesperadas; (ii) possuem alto grau de generalização.

Como entradas da ANN, foram utilizados os ângulos atuais das juntas do robô (12 entradas), normalizados entre -1 (α_{mim}) e 1 (α_{max}). Na saída da ANN, são obtidos os ângulos desejados para as juntas no instante $t + 1$ (12 saídas), normalizados entre -1 e 1. Após alguns testes preliminares, optou-se por utilizar as redes neurais recorrentes do tipo Elman [25], 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, e a camada intermediária possui 3 neurônios ocultos totalmente conectados entre si de forma recorrente, seguindo assim a arquitetura Elman padrão. O intervalo de valores possíveis para os pesos sinápticos foi limitado em $[-1, 1]$, que se mostrou bastante adequado para o problema.

4.2 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 os robôs simulados foram modelados com objetos simples, como retângulos e cilindros, e eles possuem 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 caminhar em linha reta. Inicialmente foram modelados e testados diversos tipos de robôs, até que se chegou ao modelo mostrado na Figura 1. As dimensões deste robô são similares 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 1. Modelo de robô utilizado em algumas simulações

4.3 Evolução da morfologia

Segundo Pfeifer [4], 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. 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 [4,26,27]. 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 cromossomo original do algoritmo genético. Cada segmento do robô foi codificado utilizando três valores (dimensões em x , y e z).

5 Resultados

Esta seção descreve diversos experimentos realizados utilizando o protótipo do modelo proposto. A Tabela 1 mostra os resultados obtidos nestes experimentos. Os valores entre a 2^a e a 5^a coluna (Somente controle) são relativos aos experimentos realizados evoluindo somente os parâmetros de controle, e os valores entre a 6^a e a 9^a coluna (Morfologia e controle) são relativos aos experimentos realizados evoluindo a morfologia em conjunto com os parâmetros de controle. Os gráficos da Figura 2 mostram a média e o intervalo de confiança (*Confidence Interval* – CI) a 95% dos valores do *fitness* (Figura 2(a)), da distância percorrida (Figura 2(b)) e da taxa de instabilidade (Figura 2(c)).

Tabela 1. Resultados obtidos nos experimentos realizados

	Somente controle				Morfologia e controle			
E	<i>F</i>	<i>D</i>	<i>B</i>	<i>G</i>	<i>F</i>	<i>D</i>	<i>B</i>	<i>G</i>
1	16.265	29.189	0.0019	0.0793	18.802	38.028	0.0105	0.1012
2	16.635	28.306	0.0070	0.0695	17.903	32.959	0.0940	0.0747
3	16.991	27.850	0.0085	0.0631	19.839	39.525	0.0057	0.0987
4	16.678	27.915	0.0017	0.0672	17.801	37.858	0.0017	0.1125
5	16.157	28.201	0.0076	0.0738	20.093	27.409	0.0572	0.0307
6	15.965	31.126	0.0218	0.0928	15.902	32.802	0.0091	0.1054
7	17.335	29.629	0.0048	0.0704	18.869	41.132	0.0134	0.1167
8	16.654	29.038	0.0060	0.0738	18.498	36.218	0.0074	0.0951
9	16.289	30.151	0.0022	0.0849	19.078	39.162	0.0023	0.1050
10	16.227	29.808	0.0039	0.0833	15.572	37.397	0.0057	0.1396
μ	16.520	29.121	0.0065	0.0758	18.235	36.249	0.0207	0.0979
σ	0.420	1.075	0.0059	0.0091	1.504	4.096	0.0304	0.0288

Percebe-se que nos resultados em *morfologia e controle* possuem um desvio padrão bem maior do que os resultados de *somente controle*. Isto se ocorre porque o espaço de estados em *morfologia e controle* é bem maior, o que permite que sejam evoluídas diversas soluções para o mesmo problema. Em termos da distância percorrida, se percebe claramente que a evolução da morfologia em conjunto com os parâmetros de controle produziu melhores resultados. Já em termos de estabilidade, a evolução dos parâmetros de controle de forma isolada é mais eficiente. Assim, é possível afirmar que a evolução da morfologia é uma ferramenta

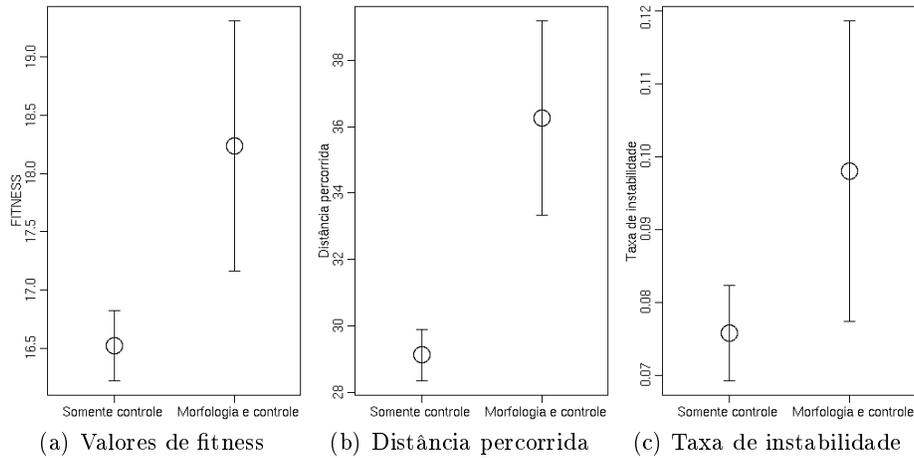


Figura 2. Gráficos que mostram a média e intervalo de confiança a 95%

útil no projeto de robôs com pernas, embora nem todas as morfologias evoluídas sejam eficientes em termos de estabilidade. A Figura 3(a) mostra as morfologias que evoluíram ao final das 700 gerações, para cada experimento controlado pela tabela de ângulos. Os números no canto superior esquerdo se referem ao experimento no qual a morfologia do robô foi evoluída. A Figura 4(a) mostra a caminhada de um dos modelos de robô evoluídos, e a Figura 4(b) mostra um exemplo de caminhar evoluído para o robô padrão⁴.

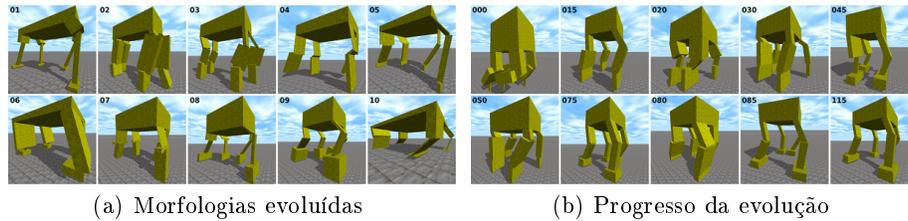


Figura 3. Exemplos de morfologias evoluídas ao final dos experimentos

A Figura 3(b) mostra as alterações da morfologia de um robô durante a evolução. Os números no canto superior esquerdo se referem a geração na qual cada modelo de robô se tornou dominante. Percebe-se que a morfologia vai aos poucos sendo melhorada, até que se chegue a um modelo de robô satisfatório. Cabe ressaltar que o espaço de estados permite surjam diversas soluções eficientes, a exemplo do que ocorre na natureza.

⁴ Vídeos de demonstração disponíveis em <http://www.inf.ufrgs.br/~mrheinen/leggen>

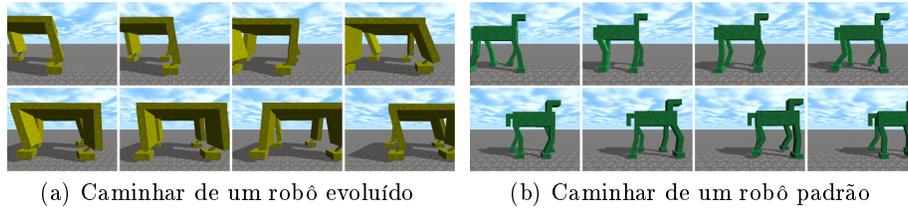


Figura 4. Exemplos de formas de caminhar evoluídas

Além destes resultados, em [1] são descritos diversos experimentos realizados visando: (i) descobrir a função de *fitness* mais adequada para o problema em questão; (ii) determinar o robô mais eficiente dentre diversos modelos de quatro e seis pernas; e (iii) avaliar diversas estratégias de controle propostas (uma autômato finito, funções cíclicas e redes neurais artificiais). Por questões de espaço não é possível detalhar todos estes experimentos aqui, mas a título de exemplificação a Figura 5 mostra dois gráficos que relacionam a distância percorrida D e a taxa de instabilidade G . Estes gráficos deixam claro que existe uma forte relação entre D e G , ou seja, as soluções mais velozes são as mais instáveis e vice-versa. Outro exemplo interessante é mostrado na Figura 6, na qual os robôs evoluídos foram capazes de transpor desníveis e escadas.

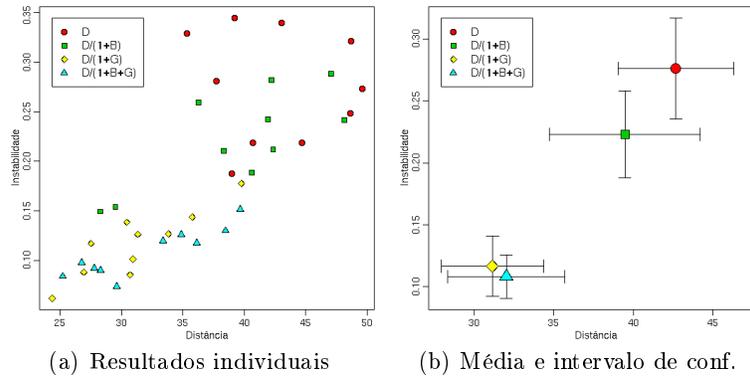


Figura 5. Relação entre a distância percorrida D e a taxa de instabilidade G

6 Conclusões e perspectivas

O objetivo deste artigo foi descrever o simulador LegGen, que é um simulador desenvolvido para realizar a configuração automática do caminhar de robôs móveis com pernas. Neste simulador, a configuração do caminhar é realizada utilizando algoritmos genéticos, que evoluem os parâmetros do caminhar (pesos sinápticos

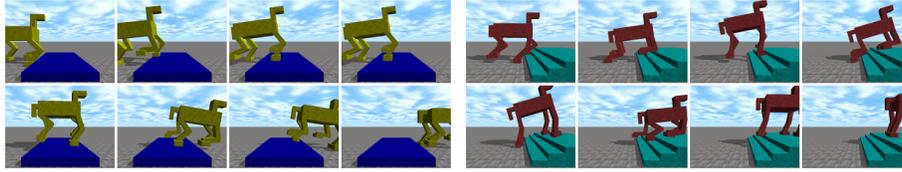


Figura 6. Robôs treinados para transpor desníveis e escadas

de uma rede neural) através de robôs simulados em um ambiente virtual realista. Além disso, o protótipo do modelo proposto permite a evolução da morfologia do robô em conjunto com os parâmetros de controle. Os resultados obtidos demonstram que: (i) as redes neurais são bastante eficientes no 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 o cálculo do gradiente; (iii) a evolução da morfologia permite que sejam descobertos modelos de robôs bastante eficientes.

Agradecimentos

Agradecemos ao apoio da CAPES e do CNPq a este trabalho.

Referências

1. Heinen, M.R.: Controle inteligente do caminhar de robôs móveis simulados. Dissertação de mestrado, Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS, Brasil (2007) Data de Conclusão: 20/02/2007. URL: http://bdt.d.unisinos.br/tde_busca/arquivo.php?codArquivo=380.
2. Knight, R., Nehmzow, U.: Walking robots - a survey and a research proposal. Technical Report CSM-375, Univ. Essex, Essex, UK (2002)
3. Bekey, G.A.: Autonomous Robots: From Biological Inspiration to Implementation and Control. MIT Press, Cambridge, MA (2005)
4. Pfeifer, R., Scheier, C.: Understanding Intelligence. MIT Press, Cambridge, MA (1999)
5. Chernova, S., Veloso, M.: An evolutionary approach to gait learning for four-legged robots. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), Sendai, Japan (2004)
6. Mitchell, T.: Machine Learning. McGrall-Hill, New York (1997)
7. Heinen, M.R., Osório, F.S.: Applying genetic algorithms to control gait of physically based simulated robots. In: Proc. IEEE Congr. Evolutionary Computation (CEC), Vancouver, Canada (2006)
8. Heinen, M.R., Osório, F.S.: Applying genetic algorithms to control gait of simulated robots. In: Proc. IEEE Electronics, Robotics and Automotive Mechanics Conf. (CERMA) 2007, Cuernavaca, Morelos, Mexico (2007) 500–505
9. Heinen, M.R., Osório, F.S.: Gait control generation for physically based simulated robots using genetic algorithms. In: Proc. Int. Joint Conf. 2006, 10th Ibero-American Conference on AI (IBERAMIA), 18th Brazilian Symposium on AI (SBIA). LNCS, Ribeirão Preto - SP, Brazil, Springer-Verlag (2006)

10. Heinen, M.R., Osório, F.S.: Controle inteligente do caminhar de robôs móveis utilizando algoritmos genéticos e redes neurais artificiais. In: Anais do VI Encontro Nacional de Inteligência Artificial (ENIA), Rio de Janeiro, Brazil (2007) 952–961
11. Haykin, S.: *Redes Neurais: Princípios e Prática*. 2nd edn. Bookman, Porto Alegre, RS, Brazil (2001)
12. Lewis, M.A., Fagg, A.H., Solidum, A.: Genetic programming approach to the construction of a neural network for control of a walking robot. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Nice, France (1992) 2618–2623
13. Bongard, J.C., Pfeifer, R.: A method for isolating morphological effects on evolved behaviour. In: Proc. 7th Int. Conf. Simulation of Adaptive Behaviour (SAB), Edinburgh, UK, MIT Press (2002) 305–311
14. Busch, J., Ziegler, J., Aue, C., Ross, A., Sawitzki, D., Banzhaf, W.: Automatic generation of control programs for walking robots using genetic programming. In: Proc. 5th European Conf. Genetic Programming (EuroGP). Volume 2278 of LNCS., Kinsale, Ireland, Springer-Verlag (2002) 258–267
15. Jacob, D., Polani, D., Nehaniv, C.L.: Legs than can walk: Embodiment-based modular reinforcement learning applied. In: Proc. IEEE Int. Symposium on Computational Intell. in Robotics and Autom. (CIRA), Espoo, Finland (2005) 365–372
16. Reeve, R., Hallam, J.: An analysis of neural models for walking control. *IEEE Trans. Neural Networks* **16**(3) (2005) 733–742
17. Golubovic, D., Hu, H.: Ga-based gait generation of sony quadruped robots. In: Proc. 3th IASTED Int. Conf. Artificial Intelligence and Applications (AIA), Benalmadena, Spain (2003)
18. Heinen, M.R., Osório, F.S.: Evolving gait control of physically based simulated robots. *Revista de Informática Teórica e Aplicada (RITA)* **XVI**(1) (2007) 119–134
19. Heinen, M.R., Osório, F.S., Engel, P.M.: Evolução do caminhar e da morfologia de robôs quadrúpedes. In: Proc. VIII Congr. Brasileiro de Redes Neurais (CBRN), Florianópolis, SC, Brazil (2007) 6
20. Osório, F.S., Musse, S.R., Vieira, R., Heinen, M.R., Paiva, D.C. In: *Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation*. Volume 2 of *Research in Interactive Design – Proc. Virtual Concept Conf.* 2006. Springer-Verlag, Berlin, Germany (2006) 1–45
21. Heinen, M.R., Osório, F.S.: Co-evolução da morfologia e controle de robôs móveis simulados utilizando realidade virtual. In: Proc. IX Symposium on Virtual and Augmented Reality, Petrópolis, RJ, Brazil, SBC Editora (2007) 187–196
22. Heinen, M.R., Osório, F.S.: Uso de algoritmos genéticos para a configuração automática do caminhar em robôs móveis. In: *Anais do Encontro de Robótica Inteligente (EnRI)*, Campo grande, MS, Brazil (2006)
23. Heinen, M.R., Osório, F.S.: Evolução do caminhar de robôs móveis simulados utilizando algoritmos genéticos. In: *Anais do XV Seminário de Computação (Sem-inco)*, Blumenau, SC, Brazil, FURB Editora (2006) 131–142
24. Heinen, M.R., Osório, F.S.: Neural networks applied to gait control of physically based simulated robots. In: Proc. Int. Joint Conf. 2006, 9th Brazilian Neural Networks Symposium (SBRN), Ribeirão Preto, SP, Brazil (2006)
25. Braga, A.d.P., Ludermir, T.B., Carvalho, A.C.P.d.L.F.: *Redes Neurais Artificiais - Teoria e Aplicações*. LTC Editora, Rio de Janeiro, Brazil (2000)
26. Nolfi, S., Floreano, D.: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA (2000)
27. Heinen, M.R., Osório, F.S.: Inteligência artificial aplicada no controle do caminhar e na evolução da morfologia de robôs móveis simulados. In: Proc. XXXIII Conferência Latino-Americana de Informática (CLEI), San José, Costa Rica (2007)