COORDENAÇÃO DOS ATUADORES DAS PERNAS DE ROBÔS MÓVEIS USANDO APRENDIZADO POR REFORÇO: SIMULAÇÃO E IMPLEMENTAÇÃO

Jeeves Lopes dos Santos*
jeeves@ita.br

Cairo Lúcio Nascimento Júnior*
cairo@ita.br

*Laboratório de Máquinas Inteligentes - LMI
Divisão de Engenharia Eletrônica
Instituto Tecnológico de Aeronáutica - ITA
São José dos Campos, São Paulo, Brasil

ABSTRACT

Actuator Coordination for Legged Mobile Robots Using Reinforcement Learning: Simulation and Implementation

This article presents a solution to the problem of how to coordinate the actuators of a legged robot such that its frontal speed is maximized. It is assumed that the position of each leg actuator is described by a periodic function that has to be determined using a reinforcement learning technique called Learning Automata. Analysis of the robot morphology is used to group similar legs and decrease the number of actuator functions that must be determined. MATLAB/Simulink and the SimMechanics Toolbox are used to simulate the robot walking on a flat surface. The simulated robot response is evaluated by the reinforcement learning technique considering: 1) the robot frontal speed, 2) the smoothness of the robot movements, 3) the largest torque required by all actuators, and 4) the energy consumption. After the reinforcement learning algorithm converges to a solution, the actuators functions are applied to the real robot that was built using the Bioloid Comprehensive Kit, an educational robot kit manufactured by Robotis. The response of the real robot is then evaluated and compared with the simulated robot response. This article presents two case studies: a quadrupedal robot and a tripedal robot. In both cases, each leg has three actuators. The solutions obtained by the proposed methodology are presented and shown to be satisfactory.

KEYWORDS: Mobile Robotics, Walking Machines, Legged Robots, Reinforcement Learning, Learning Automata, Applied Artificial Intelligence.

RESUMO

Este artigo apresenta uma solução para o problema de coordenação dos atuadores das pernas de robôs móveis com o objetivo principal de maximizar a sua velocidade frontal. É assumido que a posição no tempo de cada atuador é descrita por uma função periódica que deve ser determinada de forma iterativa por um algoritmo de aprendizado por reforço. As pernas similares do robô são identificadas e agrupadas visando diminuir o número de funções que precisam ser determinadas. O *toolbox* SimMechanics do software MATLAB/Simulink é usado para simular o caminhar do robô em uma superfície plana. O desempenho do robô simulado é medido considerando: a) a velocidade frontal e a suavidade na locomoção do robô, e b) o máximo torque e o consumo de energia dos atuadores. As funções que foram determinadas no ambiente de simulação pelo algoritmo de reforço são então usadas nos atuadores do robô real construído usando o

Artigo submetido em 16/02/2011 (Id.: 01271)
Revisado em 18/04/2011, 04/06/2011
Aceito sob recomendação do Editor Associado Prof. Guilherme Pereira

kit de robótica educacional Bioloid Comprehensive Kit. O desempenho do robô real é então medido e comparado com o desempenho do robô simulado. Este artigo apresenta dois estudos de caso: um robô quadrúpede e um trípode. Nos dois casos os robôs possuem três atuadores por perna. As soluções obtidas pela aplicação do método proposto são apresentadas e se mostraram satisfatórias.

PALAVRAS-CHAVE: Robôs Móveis, Robôs com Pernas, Inteligência Artificial, Aprendizado por Reforço.

1 INTRODUÇÃO

A robótica móvel constitui-se como uma vertente no âmbito da robótica que almeja aumentar a versatilidade de diversos tipos de equipamentos com o advento da locomoção. Neste contexto, a utilização de rodas corresponde à configuração mais comum para os robôs que se locomovem em terra devido à sua facilidade de operação e ao seu desempenho em terrenos regulares. Porém, o uso de rodas pode se tornar inviável em terrenos acidentados. Neste tipo de ambiente, os robôs com pernas, também conhecidos como *walking machines*, constituem uma opção promissora.

Além da maior possibilidade de mobilidade em relação aos robôs com rodas, existem outras vantagens que podem ser verificadas na utilização dos robôs com pernas:

Utilização das pernas para outros fins: As pernas utilizadas na locomoção não estão necessariamente limitadas a essa aplicação. Dentre as possibilidades, esses elementos do robô podem manipular e transportar objetos como verificado em alguns seres vivos (Silva e Machado, 2007);

Maior tolerância a falhas: Como as rodas necessitam constantemente estar em contato com a superfície de locomoção, a falha de uma delas (p. ex., travamento) pode inviabilizar a locomoção do robô. Por outro lado, como os robôs com pernas podem possuir pernas redundantes, há a possibilidade dos mesmos manterem um caminhar após ter uma ou mais pernas danificadas (Spenneberg *et al.*, 2004; Yang, 2003);

Maior identificação entre homem e robô: As pernas podem proporcionar um maior grau de identificação entre o homem e o robô, facilitando a inserção desses equipamentos em seu cotidiano (Pfeifer e Scheier, 1999).

Pesquisas sobre a locomoção dos seres vivos: As pesquisas desenvolvidas com os robôs dotados de pernas podem ser utilizadas para testar idéias de como funciona o sistema de locomoção dos seres vivos (Ijspeert, 2008);

Desenvolvimento de equipamentos: Os avanços obtidos com os robôs dotados de pernas podem ser utilizados para desenvolver equipamentos para auxiliar pessoas com dificuldade de locomoção. Um exemplo desse tipo de equipamento consiste nos chamados exoesqueletos (Santos *et al.*, 2009; Siqueira *et al.*, 2008; Winter *et al.*, 2008).

Coordenar os atuadores que compõem um robô com pernas corresponde a um dos grandes desafios nessa área de pesquisa devido à complexidade da dinâmica do robô e ao número de variáveis envolvidas no seu controle.

Na literatura existem duas grandes linhas de pesquisa na busca de soluções para o problema da coordenação dos atuadores dos robôs com pernas. A primeira usa uma abordagem matemática do problema para obter o modelo dinâmico do robô e gerar as leis de controle para os atuadores, como em (Westervelt *et al.*, 2007; Mistry *et al.*, 2007; Plestan *et al.*, 2003).

A segunda linha usa alguma técnica de aprendizado de máquina (Mitchell, 1997) para realizar a busca por uma solução adequada em um espaço de possibilidades. Nessa, soluções candidatas são tipicamente testadas em um robô simulado usando algum pacote de software computacional de forma tal que os modelos cinemático e dinâmico do robô não precisam ser explicitados pelo projetista. Como exemplos, (Belter e Skrzypczynski, 2010; Heinen e Osório, 2008; Xu *et al.*, 2006) utilizam algoritmos genéticos para realizar a coordenação dos atuadores dos robôs com pernas, enquanto que, dentro do campo do aprendizado por reforço, (Holland e Snaith, 1992) utiliza uma técnica conhecida como Q-learning e (Porta, 2000) utiliza uma variação dessa técnica denominada de ρ-learning. Além do Q-learning e suas variações que são comumente encontradas na literatura, outras técnicas também são utilizadas para esse fim como o *stochastic gradient ascent* (Murao *et al.*,2001), hill-climbing algorithm (Tal, *et al.*, 2005) e model-based reinforcement learning (Morimoto *et al.*, 2004).

Os pesquisadores que utilizam técnicas de aprendizado de máquina na coordenação dos atuadores de robôs com pernas tentam minimizar o número de tentativas necessárias através da simplificação do problema. Uma alternativa se baseia numa característica observada nos animais, onde, para um estilo de locomoção intermitente, há um padrão que se repete por um longo período caracterizando assim um movimento cíclico (Alexander,

1989). Essa propriedade vem sendo utilizada por pesquisadores ao compor o comportamento dos atuadores dos seus robôs (Heinen, 2007; Still e Douglas, 2006; Kohl e Stone, 2004).

O lado negativo dessa estratégia corresponde à limitação da utilização do robô em terrenos regulares, uma vez que, para as superfícies irregulares, há a necessidade do robô se adaptar às diferentes condições do terreno onde está sendo realizada a locomoção. Como alternativa, existem as técnicas de caminhar livre (*free gait*), onde a sequencia dos movimentos realizados durante a locomoção raramente se repetem. Como exemplo, (Erden e Leblebicioglu, 2008) utiliza uma técnica onde é realizada uma escolha aleatória de um estado a partir de um subconjunto de estados estáveis que satisfazem determinadas características e (Porta e Celaya, 2004) utiliza um efeito de reação para realizar o controle das pernas do robô.

É importante salientar que, apesar de uma das maiores vantagens da utilização dos robôs com pernas ser a sua utilização em terrenos irregulares, o problema da locomoção de *walking machines* em terrenos planos e regulares ainda não foi totalmente resolvido.

Com o objetivo de viabilizar a locomoção de robôs com pernas em uma determinada direção e sentido desejados em uma superfície plana e regular, este artigo propõe uma metodologia para a coordenação das pernas de robôs utilizando a técnica de aprendizado por reforço conhecida como *Learning Automata* para buscar soluções que satisfaçam múltiplos critérios.

Neste artigo, as soluções obtidas são avaliadas considerando quatro medidas:

1. a velocidade de locomoção na direção e sentido desejados;
2. a suavidade da locomoção;
3. o consumo de energia, e
4. o máximo torque exigido pelos atuadores.

Neste trabalho, resolver o problema de coordenação das pernas do robô significa propor um conjunto de funções periódicas a serem utilizadas como referências angulares pelos atuadores localizados nas articulações das pernas do robô. A solução do problema é encontrada usando um ambiente de simulação construído com o *SimMechanics Toolbox* do programa MATLAB R2009b fornecido pela empresa MathWorks (http://www.mathworks.com). Após essa etapa, a solução encontrada no ambiente de simulação é também testada no robô real construído com o kit de robótica educacional *BIOLOID Comprehensive Kit* fabricado pela empresa Robotis (http://www.robotis.com/xe/bioloid_en).

A generalidade da solução proposta é demonstrada por dois estudos de caso onde os robôs apresentam diferentes morfologias: um robô com quatro pernas (quadrúpede) e outro com três pernas (trípode). Em ambos os casos as pernas dos robôs possuem 3 atuadores.

Neste artigo a seção 2 apresenta a composição geral das morfologias utilizadas na montagem dos robôs, a seção 3 descreve a formulação do problema abordado no artigo, a seção 4 apresenta a proposta de solução adotada, a seção 5 expõe os estudos de casos realizados e a seção 6 apresenta as conclusões e as propostas para trabalhos futuros.

2 MORFOLOGIA DOS ROBÔS

De forma simplificada, os robôs dotados de pernas são compostos por um corpo principal e pelas pernas. As pernas correspondem a um conjunto de elementos rígidos com uma ou mais articulações que podem ou não ser acionadas por atuadores (Figura 1). Um robô pode possuir desde uma perna até um grande número delas que, em relação à locomoção, têm como finalidade sustentar/equilibrar o corpo do robô e gerar o impulso necessário para o seu deslocamento. Já o corpo principal corresponde à parte do robô onde as pernas estão conectadas.

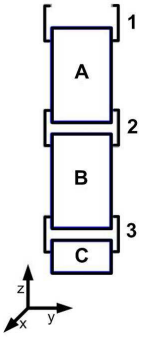

Figura 1: Exemplo de uma perna cuja morfologia possui 3 corpos rígidos (A, B e C) e 3 articulações (1, 2 e 3).

Os robôs dotados de pernas são classificados pelo número de pernas que possuem, podendo ser monópodes (uma perna), bípedes (duas pernas), trípodes (três pernas), quadrúpedes (quatro pernas), etc.

Nas morfologias utilizadas neste trabalho (quadrúpede e trípode), cada articulação de cada perna possui apenas um grau de liberdade angular que é acionado por um pequeno servomotor localizado na articulação. A velocidade e o ângulo desse servomotor são ajustados pelo seu controlador local que recebe um sinal de referência enviado por um controlador principal através de uma rede de comunicação serial cabeada tipo "*daisy-chain*". Assim sendo, a solução do problema de coordenação das pernas de um robô é obtida pela geração do sinal temporal de referência para cada articulação em cada perna.

3 FORMULAÇÃO DO PROBLEMA

Neste artigo, o sinal de referência angular utilizado pelo atuador a da perna p é caracterizado como uma função periódica no tempo $f_p^a(t)$ que é definida por um conjunto de NE pontos linearmente interpolados entre si, como ilustrado pela Figura 2.

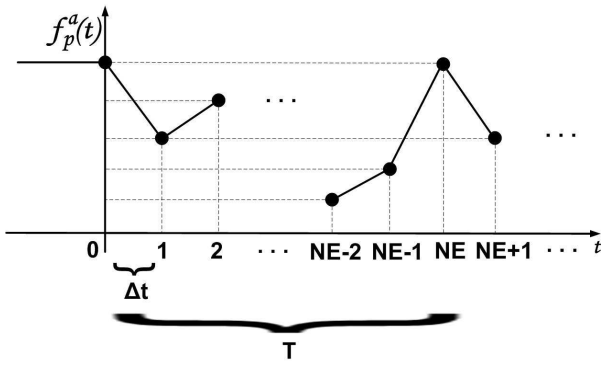

Figura 2: Representação da função aplicada no atuador a da perna p.

Dessa forma, para cada atuador existente no robô, deve-se ajustar os valores dos NE pontos que descrevem a função, juntamente com o período de tempo T.

Dado que o robô possui N atuadores, o mesmo necessitará de N funções para descrever seus movimentos. Assim, levando em consideração que cada uma das N funções é caracterizada por NE pontos e todas possuem o mesmo período T, o número total de variáveis que precisam ser determinadas (Nv) é dado pela Equação (1).

$$Nv = NE\ N\ +\ 1 \tag{1}$$

Como exemplo, para um robô quadrúpede ($Np = 4$) com três articulações por perna ($Na = 3$) e quatro pontos por função ($NE = 4$), $Nv = NE\ Np\ Na + 1 = 49$.

Visando simplificar o problema, pode-se levar em consideração as simetrias existentes no robô para minimizar o número de variáveis a serem ajustadas, como realizado em (Santos *et al.*, 2010). Para tal, as pernas que são simétricas em relação ao Centro de Massa do robô (CM) e possuem uma mesma estrutura podem ser agrupadas tal que as pernas de um mesmo grupo compartilhem as mesmas funções $f_p^a(t)$, porém com uma determinada defasagem $\phi^g(p)$ para cada perna do grupo g (uma das pernas do grupo é adotada como a perna de referência e, por definição, assume o n^o 1 no grupo e $\phi^g(1) = 0$).

Utilizando essa estratégia, considerando que todas as pernas do exemplo acima citado sejam similares, haveria a necessidade de se definir 3 funções com 4 pontos cada (uma função por atuador), 3 defasagens (1 defasagem por perna) e o período T. Assim, o número de variáveis Nv diminui de 49 para 16.

Em suma, além do período T, o algoritmo de aprendizado deve ajustar ($NE\ Na + Np - 1$) variáveis para cada grupo de pernas similares que for identificado pela análise da morfologia do robô, pois:

- a função periódica usada como função de referência da posição angular do k-ésimo atuador da perna 1 do grupo, denotada por $f_1^k(t)$, é definida pelos seus valores nos NE instantes de tempo $t = (j-1)\frac{T}{NE}$, onde $j \in \{1 : NE\}$;
- se o grupo tem mais de uma perna ($Np > 1$), então é preciso determinar a defasagem $\phi(p)$ da perna p do grupo, onde $p \in \{2 : Np\}$.

Definidas as variáveis a serem ajustadas, a busca pelos seus valores leva em consideração o desempenho do robô através da resposta obtida durante a simulação do seu caminhar. Usando como base um sistema de coordenadas inercial formado pelos eixos x, y e z onde o robô está inicialmente posicionado tal que a sua frente está alinhada no sentido positivo do eixo x, os seguintes sinais são observados na simulação:

1. $V_x(t)$, $V_y(t)$, $V_z(t)$: velocidades lineares do CM nos eixos x, y e z;
2. $W_x(t)$, $W_y(t)$, $W_z(t)$: velocidades angulares observadas no corpo principal do robô em torno dos eixos x, y e z (velocidades de rolagem, arfagem e guinada) com a origem das coordenadas localizada no CM do robô;
3. $\tau_p^a(t)$: torque no atuador a da perna p;

4. $W_p^a(t)$: velocidade angular do atuador a da perna p.

Utilizando as variáveis medidas, as seguintes matrizes amostradas são definidas:

$$V = \begin{bmatrix} V(x,1) & V(x,2) & ... & V(x,N) \\ V(y,1) & V(y,2) & ... & V(y,N) \\ V(z,1) & V(z,2) & ... & V(z,N) \end{bmatrix} \quad (2)$$

$$W = \begin{bmatrix} W(x,1) & W(x,2) & ... & W(x,N) \\ W(y,1) & W(y,2) & ... & W(y,N) \\ W(z,1) & W(z,2) & ... & W(z,N) \end{bmatrix} \quad (3)$$

onde:

- $V(i,j)$ corresponde à velocidade linear do CM no eixo x, y ou z ($i = 1,\ 2$ ou 3 respectivamente) no instante de amostragem j ($j = 1$ a N);
- $W(i,j)$ corresponde à velocidade angular do CM em torno do eixo x, y ou z ($i = 1,\ 2$ ou 3 respectivamente) no instante de amostragem j ($j = 1$ a N).

Com esses sinais obtidos, quatro índices são utilizados para avaliar a qualidade da resposta obtida:

1. Velocidade: Visando viabilizar que o robô chegue ao seu destino rapidamente, busca-se um comportamento que maximize a média da velocidade linear ($\overline{V}$) do CM do robô no sentido positivo do eixo x;
2. Suavidade da locomoção do robô: Deseja-se minimizar as variações das velocidades lineares e angulares observadas no CM do robô para evitar que, ao carregar uma carga, o seu conteúdo seja danificado.

   Para mensurar essas variações, são definidas as taxas de variações das velocidades lineares (I_V) e das velocidades angulares (I_W). A primeira é mensurada a partir das variações das velocidades lineares observadas no CM, cujo valor é obtido através da Equação (4).

$$I_V = \sqrt{\frac{\sum_{i=1}^{3}(\sum_{j=1}^{N}(V(i,j) - \overline{V(i)})^2)}{N}} \quad (4)$$

   A taxa de variação das velocidades angulares (I_W) é obtida de forma análoga à I_V. Para tal, utiliza-se a Equação (5) (Golub e Hu, 2003).

$$I_W = \sqrt{\frac{\sum_{i=1}^{3}(\sum_{j=1}^{N}(W(i,j) - \overline{W(i)})^2)}{N}} \quad (5)$$

3. Máximo torque exigido: No intuito de evitar a saturação do atuador real e permitir que robôs com atuadores menos potentes possam viabilizar o desempenho desejado, busca-se minimizar o máximo torque instantâneo aplicado pelos atuadores considerando todas as pernas (τ_{max}). Dessa maneira, implicitamente assume-se que todos os atuadores são iguais.
4. Consumo de energia: Visando maximizar o tempo de operação do robô sem a necessidade de paradas para a recarga das suas baterias, a minimização do consumo de energia é considerada. Para tal, busca-se minimizar a soma da energia cinética rotacional verificada em todos os atuadores (Ec).

   O valor de Ec é dado pela Equação 6, onde t_i e t_f são os instantes de tempo inicial e final da realização de um passo.

$$Ec = \sum_{a=1}^{Na}\left(\int_{t_i}^{t_f} \tau_p^a(t)\ W_p^a(t)\ dt\right) \quad (6)$$

Dessa forma, essas grandezas escalares compõem o vetor de desempenho $J = [\overline{V_x}\ I_V\ I_W\ \tau_{max}\ Ec]$ que quantifica o resultado obtido com a utilização de um determinado conjunto de funções $f_p^a(t)$.

4 PROPOSTA DE SOLUÇÃO

Para ajustar as variáveis que definem as funções de referência de cada atuador, este artigo utiliza uma técnica de aprendizado por reforço conhecida como *Learning Automata* (Narendra e Thathachar, 1974).

O Aprendizado por Reforço (AR) corresponde a um meio de mapear situações em ações visando maximizar um sinal de reforço numérico. Para tal, avalia-se o conhecimento acumulado pela aplicação de propostas de soluções para direcionar a busca por melhores soluções (Sutton e Barto, 1998; Thathachar e Sastry, 2002). Dessa forma, o AR caracteriza-se como um método de aprendizado com supervisão fraca, cujo supervisor apenas fornece informações de sucesso ou fracasso durante a fase de treinamento (Nascimento Jr. e Yoneyama, 2000).

As técnicas existentes em AR são compostas por quatro elementos:

***Política de Ações*:** Define a ação em um dado momento do aprendizado, podendo ser comparado na psicologia como as regras de respostas a estímulos ou associações. Na aplicação em questão, esse elemento corresponde à descrição de um determinado conjunto de funções $f_p^a(t)$ a serem testadas;

***Função Objetivo*:** Corresponde à função que avalia o desempenho da ação tomada. Esta função tem como objetivo fornecer um reforço, uma contribuição imediata que, em sistemas biológicos, pode ser comparado ao prazer e à dor;

***Função de Avaliação*:** Avalia a qualidade como um todo das possíveis ações a serem tomadas levando em consideração um longo período. Dessa forma, é feita uma avaliação mais refinada e abrangente, definindo numericamente o conhecimento obtido;

***Modelo do Sistema*:** Com a função de descrever o comportamento do sistema que se está aplicando o aprendizado por reforço, este ítem é optativo nesse tipo de aplicação.

Com esses elementos, o aprendizado por reforço utiliza a experiência com as tentativas para obter o conhecimento desejado. Para tal, o conhecimento armazenado na função de avaliação é utilizado para selecionar a próxima política de ações a serem tomadas. Após executadas, o resultado é então avaliado quantitativamente através da função objetivo que, por sua vez, define o reforço a ser utilizado na atualização da função de avaliação. Este ciclo é então repetido até que haja a convergência do conhecimento.

Nesse contexto, o *Learning Automata* (LA) é uma técnica de AR que tem como base os chamados autômatos que correspondem à modelagem matemática das Máquinas de Estados Finitos (MEF). Uma MEF é uma representação do comportamento de um sistema através de um conjunto de estados, transições e ações, tal que (Hopcroft *et al.*, 2006):

- Os estados correspondem a um conjunto mínimo de variáveis capazes de descrever o sistema em um determinado instante;
- As transições correspondem às mudanças entre estados que ocorrem regidas por condições;
- As ações são atividades que devem ser realizadas em um determinado instante (ao entrar ou sair de um estado, durante uma transição entre estados, etc.).

Nesse contexto, o LA tem como função ajustar a função de avaliação representada pelo conjunto de probabilidades associadas às possíveis transições do autômato utilizando os conceitos de AR.

Para aplicar a teoria desenvolvida em LA na pesquisa aqui apresentada, cada variável a ser ajustada é associada a um autômato com apenas um estado e um conjunto de possíveis transições discriminadas pelo projetista (Thathachar e Sastry, 2003).

4.1 Armazenamento do Conhecimento

O primeiro passo para a implementação do LA consiste em gerar a estrutura capaz de armazenar o conhecimento, em outras palavras, gerar a função de avaliação. Para tal, este artigo utiliza um vetor coluna P_T e duas matrizes P_ϕ^g e P_f^g para cada grupo de pernas similares g. Com essa representação, o vetor e cada coluna das matrizes armazena a função de avaliação de um autômato específico.

Associado às posições angulares que descrevem as funções $f_p^a(t)$, P_f^g tem dimensões NPP x NE x Na^g, onde:

- NPP é o número de possíveis posições angulares para os atuadores;
- NE é o número de pontos nas funções de referência dos atuadores, e
- Na^g é o número de atuadores em uma das pernas do grupo g.

Assim, na matriz P_f^g (Figura 3):

Linhas: cada linha corresponde a uma possível posição angular do atuador;

Colunas: cada coluna corresponde a um instante de tempo;

Profundidades: cada profundidade corresponde a um atuador.

Analisando a morfologia do robô, deve-se definir a matriz θ^g com dimensão NPP x Na^g associada à matriz P_f^g. O elemento $\theta^g(i,k)$ representa uma possível posição angular para o k-ésimo atuador da perna de referência do grupo de pernas similares g. Assim, os elementos da k-ésima coluna da matriz θ^g são definidos de forma linearmente espaçados entre as posições angulares mínima e máxima do k-ésimo atuador.

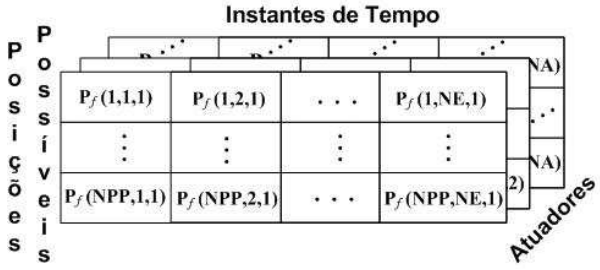

Figura 3: Organização da matriz P_f^g que armazena o conhecimento adquirido pelo aprendizado por reforço para as posições angulares dos atuadores.

Com essa representação, o valor do elemento da matriz $P_f^g(i,j,k)$ representa a estimativa de probabilidade de sucesso quando o elemento $\theta^g(i,k)$ define o j-ésimo ponto da função de referência do k-ésimo atuador da perna de referência do grupo g. Assim, a soma dos valores de uma mesma coluna da matriz P_f^g deve ser sempre 1.

Inicialmente, como ainda não foi adquirido nenhum conhecimento a respeito dos pontos que irão compor cada função, todas as probabilidades $P_f^g(i,j,k)$ assumem o valor $1/NPP$.

A matriz P_ϕ^g (Figura 4) armazena o conhecimento referente às defasagens das pernas do grupo g e tem dimensão NE x Np^g onde Np^g é o número de pernas no grupo g. Essa matriz é organizada da seguinte forma:

Linhas: cada linha corresponde a um possível valor de defasagem;

Colunas: cada coluna corresponde a uma perna do grupo de pernas similares.

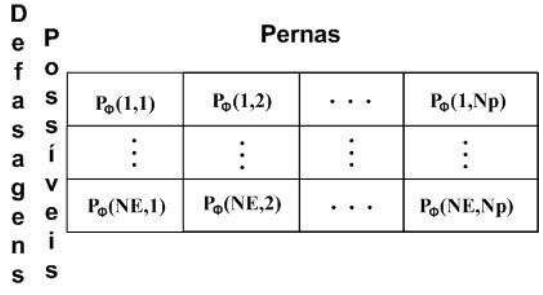

Figura 4: Organização da matriz P_ϕ^g que armazena o conhecimento adquirido pelo aprendizado por reforço para as defasagens entre as pernas de um mesmo grupo de pernas similares.

As defasagens das pernas de um mesmo grupo (elementos do vetor ϕ^g) admitem valores inteiros entre 0 e $NE-1$. Portanto, como $f_1^k(t)$ denota a função periódica do k-ésimo atuador da perna 1 (perna de referência do grupo), então a função periódica do k-ésimo atuador da j-ésima perna será dada por:

$$f_j^k(t) = f_1^k\left(t + \phi^g(j)\frac{T}{NE}\right) \tag{7}$$

Como a primeira coluna da matriz P_ϕ^g está associada à perna de referência (cuja defasagem por definição é 0), então tal coluna é definida como $P_\phi^g(:,1) = [1,0,...,0]^T$ e não é alterada pelo algoritmo de aprendizado. Os elementos das demais colunas da matriz P_ϕ^g são definidos inicialmente como $1/NE$.

Por fim, o projetista deve ainda definir os vetores V_T e P_T com o mesmo número de elementos. O vetor V_T contém os possíveis valores para o parâmetro T (período das funções de referência de todos os atuadores do robô). O Apêndice A deste artigo mostra como o projetista pode definir os elementos do vetor V_T.

O elemento $P_T(i)$ representa a estimativa da probabilidade de sucesso quando o valor $V_T(i)$ é usado como o período das funções de referência de todos os atuadores do robô.

Como no caso das colunas das matrizes P_f^g e P_ϕ^g, inicialmente $P_T(i) = 1/NPT$, onde NPT corresponde ao tamanho do vetor V_T.

4.2 Algoritmo de aprendizado

Para implementar o algoritmo de aprendizado proposto (LA) segue-se os seguintes passos a cada iteração:

1. Seleção da solução a ser testada;
2. Quantificação da qualidade da resposta obtida utilizando da representação simulada do robô;
3. Ajuste das probabilidades de sucesso e verificação da convergência do conhecimento.

4.2.1 Seleção da Solução a Ser Testada

O primeiro passo do algoritmo de aprendizado corresponde à seleção da solução a ser testada, ou seja, a definição da política de ações composta pelas funções de referências $f_p^a(t)$. Os parâmetros que caracterizam as referidas funções são selecionados aleatóriamente considerando as probabilidades registradas no vetor P_T e nas matrizes P_f^g e P_ϕ^g.

Para tal, inicialmente seleciona-se um elemento do vetor V_T considerando o vetor de probabilidades P_T. Em seguida, para cada grupo de pernas similares g:

- são selecionadas as defasagens de cada perna, onde a definição da defasagem da j-ésima perna considera as probabilidades registradas na coluna $P_{\phi}^{g}(:,j)$;
- são selecionados os NE pontos que definem as funções de referência de cada atuador da perna de referência ($p = 1$) considerando os valores da coluna $\theta^{g}(:,k)$, as probabilidades registradas nas colunas $P_{f}^{g}(:,:,k)$ e a máxima velocidade angular especificada para os atuadores reais (W_{max}) (Maiores detalhes sobre a seleção dos pontos que descrevem as funções $f_1^a(t)$ podem ser vistos no Apêndice B deste artigo).

4.2.2 Avaliação da Resposta Obtida Utilizando o Robô Simulado

O comportamento do robô usando a solução selecionada na etapa anterior é então avaliado em um ambiente de simulação através do vetor de desempenho J (apresentado na seção 3).

Neste artigo, o ambiente de simulação foi criado usando o *SimMechanics Toolbox* do MATLAB/Simulink R2009b (http://www.mathworks.com). Essa ferramenta permite descrever e simular o modelo de complexos equipamentos mecânicos através de um diagrama composto por um conjunto de blocos representando uma combinação de corpos rígidos conectados entre si por juntas translacionais e/ou rotacionais. Dessa forma, o modelo oferece uma simulação física da cinemática e da dinâmica do robô, com parâmetros e resultados que consideram gravidade, peso, colisões, etc. Como exemplo, a Figura 5 ilustra a conexão entre esses elementos de tal forma a gerar a simulação desejada[1].

4.2.3 Ajuste das Probabilidades de Sucesso e Verificação da Convergência do Conhecimento

Após o cálculo do vetor de desempenho J, o mesmo é utilizado para ajustar a função de avaliação composta pelo vetor P_T e pelas matrizes P_f e P_{ϕ}. Os detalhes desse procedimento são mostrados no Apêndice C deste artigo.

[1]Maiores informações sobre a utilização e o funcionamento do SimMechanics podem ser encontradas em http://www.mathworks.com/products/simmechanics/.

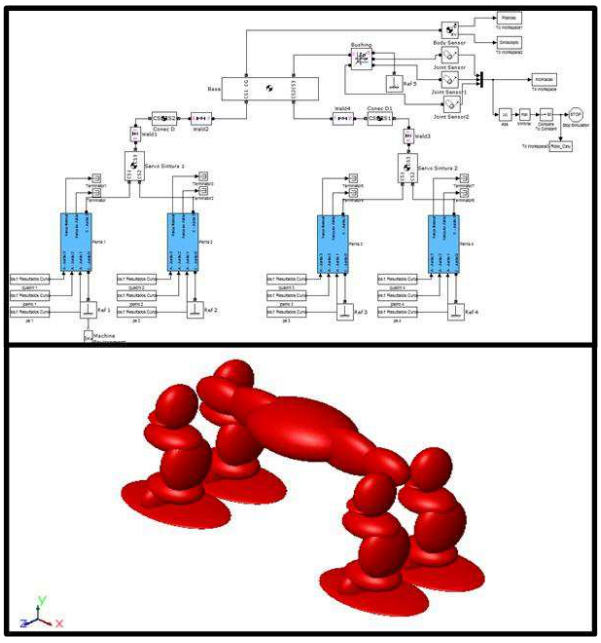

Figura 5: Modelo de simulação do robô móvel quadrúpede usado no *SimMechanics Toolbox* do MATLAB e a representação gráfica gerada pelo mesmo software.

Ao final do ajuste das probabilidades associadas aos parâmetros que definem a política de ações selecionada, o passo seguinte consiste em verificar se houve convergência do conhecimento. Neste artigo, o critério usado para identificar a convergência do conhecimento foi a presença de um elemento com valor superior a 0,95 no vetor P_T e em todas as colunas das matrizes P_f e P_{ϕ} de todos os grupos de pernas.

Não sendo verificada a convergência, uma nova iteração é realizada, caso contrário o treinamento é finalizado e o conjunto final de funções $f_p^a(t)$ é definido. Para tal, sendo Ng o número de grupos de pernas similares existentes, a solução final é identificada pelos valores dos parâmetros com maior probabilidade associada utilizando o seguinte algoritmo:

5 ESTUDOS DE CASO

Para analisar o desempenho da metodologia de coordenação aqui proposta, foram utilizadas duas morfologias de robôs móveis com pernas: um robô quadrúpede e um robô trípode (Figura 6).

Os robôs reais foram construídos utilizando o *BIOLOID Comprehensive Kit* da empresa ROBOTIS (http://www.robotis.com/xe/bioloid_en) que contém um conjunto de componentes que podem ser dis-

encontre $i^* = \{i$ que maximiza $P_T(i)\}$
$T = V_T(i^*)$
for $g = 1 \rightarrow Ng$ **do**
  for $k = 1 \rightarrow Na^g$ **do**
    for $j = 1 \rightarrow NE$ **do**
      encontre $i^* = \{i$ que maximiza $P_f^g(i, j, k)\}$
      $f_1^k \left[(j-1)\frac{T}{NE}\right] = \theta^g(i^*, k)$
    end for
  end for
  if $Np > 1$ **then**
    for $p = 1 \rightarrow Np^g$ **do**
      encontre $i^* = \{i$ que maximiza $P_\phi^g(i, p)\}$
      $f_p^k(t) = f_1^k \left[t + (i^* - 1)\frac{T}{NE}\right]$
    end for
  end if
end for

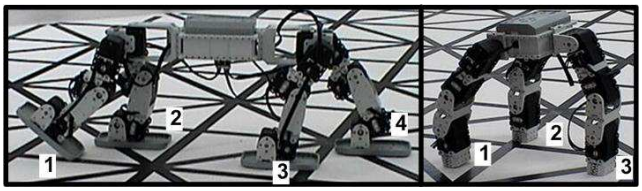

Figura 6: Robôs utilizados nos testes.

postos de diversas formas viabilizando a montagem de robôs com pernas, garras e/ou rodas. Os referidos componentes correspondem a:

1. Uma unidade de processamento microcontrolada conhecida como CM-5 que age como o coordenador central e é responsável por gerenciar os demais elementos (atuadores e sensores) através de uma rede de comunicação serial cabeada tipo "*daisy-chain*" embarcada no robô;

2. Servomotores microcontrolados que são usados como atuadores em cada junta; o microcontrolador de cada servomotor recebe a função de referência da posição angular e gera os sinais de controle para o servomotor;

3. Diversos tipos de armações para conectar os componentes, permitindo montar o robô almejado.

Em ambos os estudos de caso os processos de aprendizado utilizaram um $NPP = 20$, um $NPT = 20$ e um $NE = 4$.

5.1 Robô Móvel Quadrúpede

Com as características apresentadas na Tabela 1 e os limites das posições angulares dos atuadores do robô apresentadas na Tabela 2, todas as pernas do robô quadrúpede podem ser agrupadas em um único grupo ao verificar a similaridade do robô. Assim, o algoritmo de aprendizado deve definir os valores para 16 variáveis que são:

- o período de tempo T, usando os vetores V_T e P_T,
- os 12 pontos que formam as funções de referência dos 3 atuadores da perna de referência, com 4 pontos por função (usando as matrizes θ e P_f), e
- as 3 defasagens das outras 3 pernas (usando a matriz P_ϕ).

Tabela 1: Características do robô quadrúpede.

		Peso	Dimensões (cm)		
Np	Na	(kg)	X	Y	Z
4	3	$1,54$	30	$19,5$	11

Tabela 2: Limites das posições angulares dos 3 atuadores de cada perna do robô quadrúpede.

Atuador					
1		2		3	
Min	Max	Min	Max	Min	Max
$-42,5^o$	$57,5^o$	-90^o	0^o	-60^o	12^o

Utilizando um vetor de pesos $F = [1\ 2\ 1\ 1]$ (Maiores detalhes sobre o vetor F podem ser vistos no Apêndice C), um $T_{min} = 0,28\ s$ e um $T_{max} = 1\ s$, obteve-se o resultado cujo progresso está representado na Figura 7 através de três gráficos, onde:

- o primeiro gráfico mostra o histórico das taxas de convergência (Txc) do vetor P_T e das matrizes P_f e P_ϕ (a taxa de convergência do vetor P_T é definida pelo seu valor máximo e a taxa de convergência das matrizes P_f e P_ϕ é definida pela média dos valores máximos de todas as suas colunas);
- O segundo gráfico apresenta a média móvel com 20 iterações das velocidades $\overline{V}$ ao longo do treinamento. Para se obter a média móvel com N iterações, o seu elemento i corresponde à média dos resultados obtidos na iteração $i - N + 1$ à iteração i;

- O último gráfico mostra a porcentagem móvel de quedas detectadas com 50 iterações. A porcentagem móvel é obtida de forma análoga à média móvel, ou seja, o elemento i corresponde à porcentagem das quedas verificadas na iteração $i - N + 1$ à iteração i. Já a queda do robô é identificada quando a posição do seu CM em relação ao eixo z atinge uma altura de 0 m.

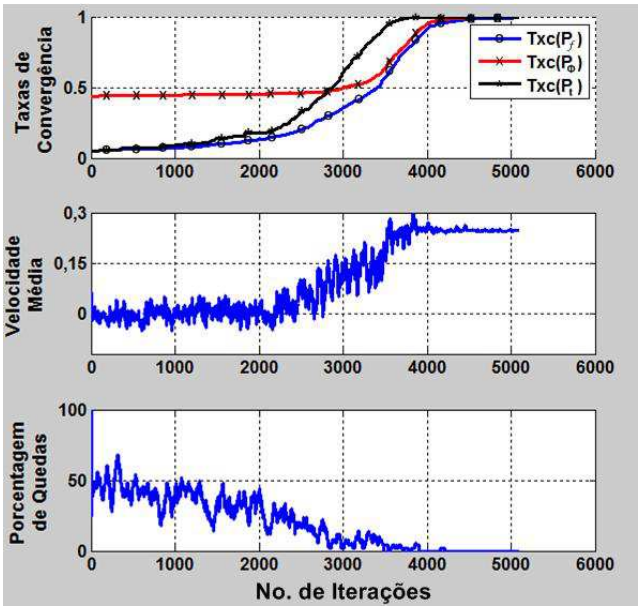

Figura 7: Progresso do aprendizado do robô quadrúpede ao longo do treinamento.

O primeiro gráfico da Figura 7 mostra que o período do passo (T) foi o primeiro parâmetro a convergir, seguido pelas defasagens (vetor ϕ) e pelos pontos que descrevem as funções de referência ($f_1^a(t)$).

Seguindo a análise do processo de aprendizado, o segundo e o terceiro gráficos confirmam o progresso verificado no anterior. Neles verifica-se que a média móvel da velocidade do robô aumenta à medida que as taxas de convergência aumentam enquanto que a porcentagem móvel de quedas diminui.

Após a convergência do processo de aprendizado, que ocorreu com 5108 iterações, o período T foi ajustado para $0,32$ s e obteve-se as funções $f_1^1(t)$, $f_1^2(t)$ e $f_1^3(t)$ apresentadas na Figura 8 para as juntas 1, 2 e 3 (respectivamente as juntas do quadril, joelho e tornozelo da perna do robô).

O processo de aprendizado também ajustou a defasagem das 4 pernas para [0 2 2 0], ou seja, as 2 pernas traseiras (pernas 1 e 2) estão defasadas 180^o entre si e as pernas

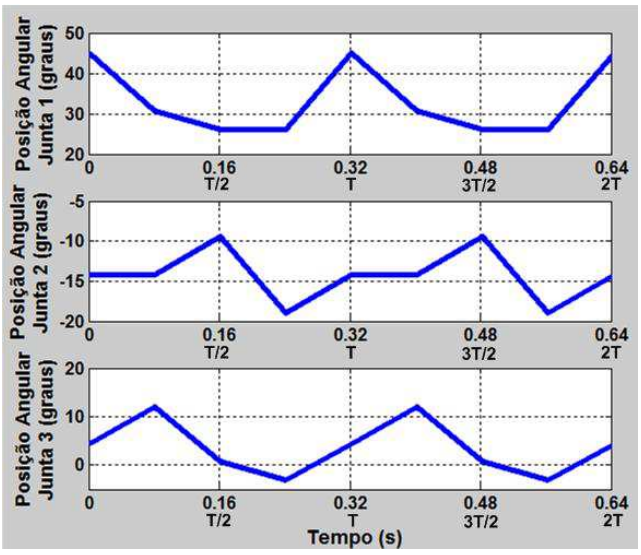

Figura 8: Funções de referência angular obtidas pelo processo de aprendizado para o robô quadrúpede.

na mesma diagonal estão em fase (pernas 1 e 4 e pernas 2 e 3). A numeração das pernas do robô quadrúpede é apresentada na Figura 6.

A Tabela 3 mostra a medida de desempenho da solução obtida pelo processo de aprendizado (componentes do vetor J) para o robô quadrúpede simulado[2].

Tabela 3: Medida de desempenho da solução obtida pelo processo de aprendizagem para o robô quadrúpede simulado.

V	I_V	I_W	τ_{max}	Ec
$27,87\frac{cm}{s}$	$0,14\frac{rad}{s}$	$0,27\frac{m}{s}$	$26,21Nm$	$0,44J$

Ao aplicar a solução obtida pelo processo de aprendizado no robô real[2] (Figura 9), obteve-se uma velocidade de $26,60$ cm/s, ou seja, cerca de 95% da velocidade obtida na simulação.

Como o robô real não possui sensores, atualmente não há como verificar o máximo torque exigido, a potência média e as taxas de variações das velocidades lineares e angulares ao executar o movimento no robô real.

5.2 Robô Móvel Trípode

Com as características apresentadas na Tabela 4, o robô trípode corresponde a uma morfologia com maior dificuldade para determinar o modo de caminhar quando

[2]O caminhar obtido para o robô quadrúpede simulado e para o robô real são mostrados nos vídeos disponíveis em:
ftp://labattmot.ele.ita.br/ele/jeeves/videos/C&A2011_4ps.wmv
ftp://labattmot.ele.ita.br/ele/jeeves/videos/C&A2011_4pr.wmv

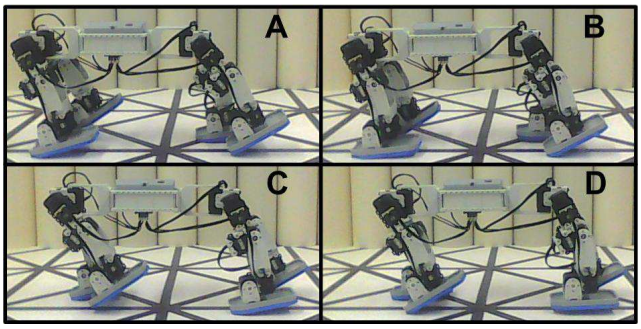

Figura 9: Comportamento verificado no quadrúpede real com as funções obtidas no aprendizado.

comparado ao robô quadrúpede. O principal fator que aumenta essa complexidade corresponde ao fato de que, para uma postura estaticamente estável, há a necessidade das três pernas estarem em contato com a superfície de suporte. Assim, quando uma perna do robô trípode é levantada do chão para executar o movimento de caminhar, a postura do robô fica instável.

Tabela 4: Características do robô trípode.

		Peso	Dimensões (cm)		
Np	Na	(kg)	X	Y	Z
3	3	$1,05$	11	$22,6$	19

Outro fator que dificulta a convergência do conhecimento é o número de variáveis a serem ajustadas. Ao analisar a similaridade do robô, obtém-se dois grupos de pernas: a) o Grupo 1 é formado pelas pernas traseiras 1 e 2, e b) o Grupo 2 é formado apenas pela perna dianteira 3 (pernas numeradas conforme Figura 6). A Tabela 5 mostra os limites das posições angulares dos 3 atuadores dos 2 grupos de pernas.

Tabela 5: Limites das posições angulares dos 3 atuadores de cada grupo de pernas.

Grupo	Atuador	Min	Max
1	3	-150^o	150^o
	2	-90^o	90^o
	1	-90^o	90^o
2	3	-12^o	150^o
	2	-90^o	90^o
	1	-90^o	90^o

Para este robô o processo de aprendizado precisa determinar 26 variáveis:

- o período T,
- a defasagem da perna 2 do Grupo 1,
- 4 pontos para a função de referência de cada um dos 3 atuadores da perna de referência do Grupo 1 (sub-total: 12 variáveis), e
- o mesmo que o item anterior para a perna de referência do Grupo 2.

Utilizando um vetor de pesos $F = [1\ 0\ 1\ 1]$, um $T_{min} = 0,76\ s$ e um $T_{max} = 1\ s$, obteve-se a evolução apresentada na Figura 10 onde é mostrada a maior dificuldade para a convergência do processo de aprendizado nesse caso quando comparado ao caso do robô quadrúpede. Nesse procedimento foram necessários 11458 iterações para que o algoritmo de aprendizado atingisse a convergência em todas as variáveis envolvidas na coordenação das pernas do robô.

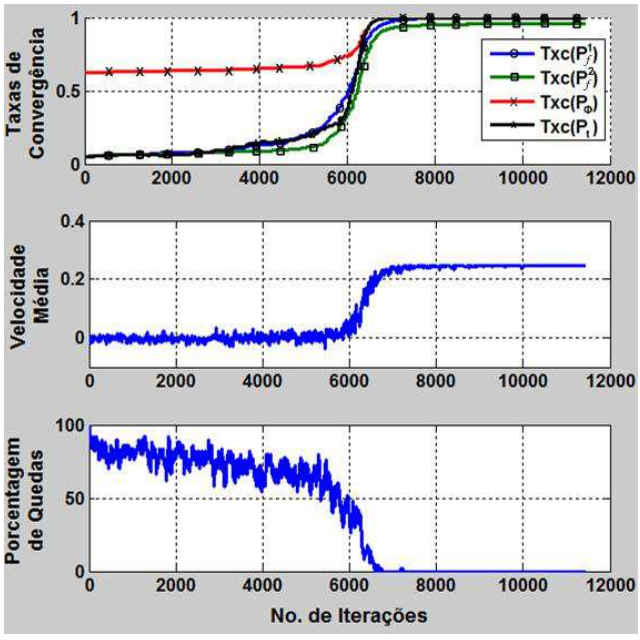

Figura 10: Evolução observada durante o processo de aprendizado para o robô trípode.

Como resultado da etapa de aprendizado, obteve-se as curvas apresentadas na Figura 11 com um período $T = 0,82\ s$, onde as juntas 1, 2 e 3 correspondem aos atuadores superior, central e inferior, respectivamente. Para o grupo de pernas 1 obteve-se uma defasagem nula entre as pernas. Com essas características, o caminhar obtido[3] para o robô apresentou o desempenho ilustrado pela Tabela 6.

[3]O caminhar obtido para o robô trípode simulado e para o robô real são mostrados nos vídeos disponíveis em:
ftp://labattmot.ele.ita.br/ele/jeeves/videos/C&A2011_3ps.wmv
ftp://labattmot.ele.ita.br/ele/jeeves/videos/C&A2011_3pr.wmv

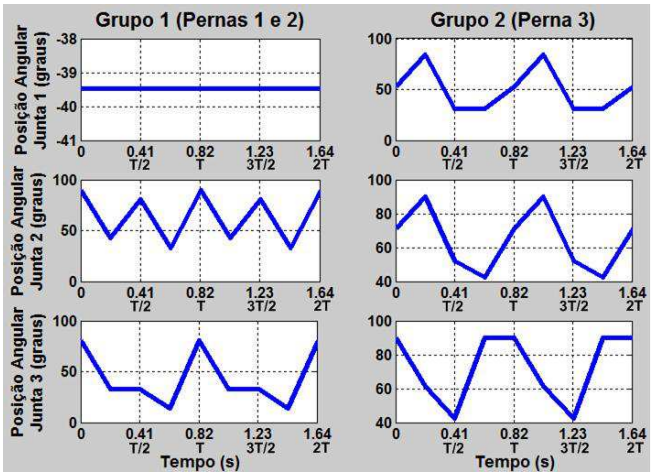

Figura 11: Funções de referência angular obtidas pelo processo de aprendizado para as juntas 1, 2 e 3 dos dois grupos de pernas do trípode.

Tabela 6: Medida de desempenho da solução obtida pelo processo de aprendizagem para o robô trípode simulado.

V	I_V	I_W	τ_{max}	Ec
$28,21\frac{cm}{s}$	$0,40\frac{rad}{s}$	$1,99\frac{m}{s}$	$52,07Nm$	$1,81J$

Quando testado no ambiente real[3] (Figura 12) o robô apresentou uma velocidade de $18,59\ cm/s$ que corresponde a cerca de 66% da velocidade atingida durante a simulação.

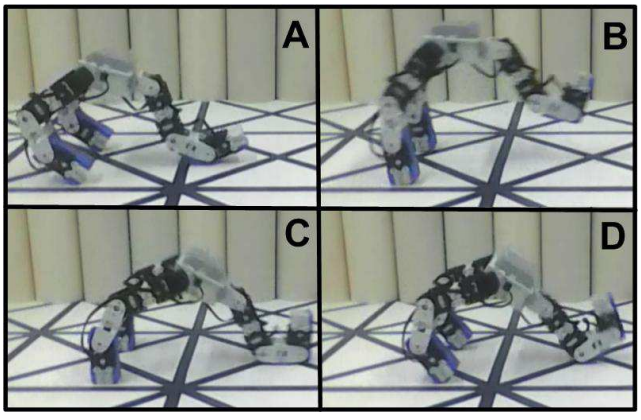

Figura 12: Comportamento verificado no Trípode real com as funções obtidas no aprendizado.

Ao comparar o caminhar simulado e o real, constata-se que no segundo ocorrem momentos onde os pés traseiros derrapam enquanto que no robô simulado isso não ocorre. Sendo assim, a diferença de velocidade verificada tem como principal fator esse efeito, indicando que há a necessidade de se aprimorar o modelo utilizado para simular o efeito das forças de atrito que incidem nos pés.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou uma metodologia capaz de gerar a coordenação dos atuadores de robôs com pernas para duas distintas morfologias de robôs, onde pode-se esperar que tal metodologia possa também ser adotada para outras morfologias.

A metodologia proposta procura:

1. maximizar a velocidade do robô na sua direção frontal;
2. maximizar a suavidade do deslocamento do robô;
3. minimizar o máximo torque e o consumo de energia dos atuadores localizados nas juntas do robô.

Para ajustar as variáveis envolvidas no deslocamento, uma técnica de aprendizado por reforço foi utilizada e a simetria entre as pernas foi levada em consideração no intuito de facilitar e, consequentemente, agilizar o aprendizado.

Nesse contexto, o projetista deve equilibrar a relação entre a velocidade de convergência e a exploração dos conjuntos de possíveis parâmetros que descrevem as funções de referência utilizadas no controle dos atuadores. Para tal, deve-se ajustar o número de pontos que descrevem as referidas funções (NE), o número de possíveis posições angulares que podem compor esses pontos (NPP) e o número de possíveis períodos (NPT).

Nos estudos de caso, a metodologia foi testada em 2 morfologias (robôs com 4 e 3 pernas) em ambiente simulado e em protótipos dos robôs reais. Apesar da diferença verificada entre os desempenhos de velocidade na simulação e no robô trípode real, os resultados obtidos mostram que a metodologia proposta atinge o resultado desejado ao viabilizar a coordenação das pernas de diferentes morfologias.

Algumas possibilidades de trabalhos futuros são:

1. Ajuste dos modelos usados na simulação que descrevem a força de reação de contato e o atrito entre os pés do robô e a superfície onde ele se locomove;
2. Inserção de mais sensores nos robôs reais de tal forma a viabilizar a extração de mais informações a

cerca da interação entre o robô e o ambiente de navegação. Dentre as opções, podem ser adicionados sensores de pressão sob os pés dos robôs, um sensor inercial com 6 graus de liberdade para medição de posição e orientação, sensor infravermelho para identificar e localizar possíveis obstáculos, dentre outros;

3. Continuação do aprendizado usando o robô real, após a obtenção da solução usando o ambiente de simulação;

4. Avaliação da influência da variação dos valores de NE, NPP e NPT no desempenho do robô e na velocidade de convergência.

AGRADECIMENTOS

Os autores agradecem o suporte financeiro concedido pela CAPES (Projeto Pró-Engenharias PE-041-2008) e pela FAPESP (Processo no. 2006/06005-0) e o apoio da Divisão de Engenharia Eletrônica do ITA ao Laboratório de Máquinas Inteligentes (LMI).

APÊNDICE A

O vetor V_T possui NPT elementos que são definidos de forma linearmente espaçada entre o T_{min} dado pela Equação A.1 e o T_{max} definido pelo projetista.

$$T_{min} = NE\frac{max(|\triangle\theta|_{i+3})}{W_{max}} \qquad \text{(A.1)}$$

onde:

- $max(|\triangle\theta_{i+3}|)$ é a variação angular máxima entre as posições representadas pelos elementos $\theta^g(i,k)$ e $\theta^g(i+3,k)$, para k variando de 1 a Na^g considerando todos os grupos de pernas;
- W_{max} é a velocidade máxima que os atuadores podem atingir.

APÊNDICE B

A seleção dos pontos da função de referência do k-ésimo atuador da perna de referência do grupo g é realizada seguindo o seguinte algoritmo:

APÊNDICE C

for $j = 1 \rightarrow NE$ **do**
  if $j = 1$ **then**
    Seleciona $f_1^k\left[(j-1)\frac{T}{NE}\right]$ utilizando $\theta^g(:,k)$ e $P_f^g(:,j,k)$
  else
    L_i = mínimo i tal que $\theta^g(i,k) \geq f_1^k\left[(j-2)\frac{T}{NE}\right] - \frac{W_{max}NE}{T}$
    L_s = máximo i tal que $\theta^g(i,k) \leq f_1^k\left[(j-2)\frac{T}{NE}\right] + \frac{W_{max}NE}{T}$
    $\theta_{aux} = \theta^g(L_i : L_s, k)$ e $P_{aux} = P_f^g(L_i : L_s, j, k)$
    if $\sum_{i=1}^{L_s-L_i} P_{aux} < 0,01$ **then**
      j=1
    else
      Normalize P_{aux}
      Seleciona $f_1^k\left[(j-1)\frac{T}{NE}\right]$ utilizando θ_{aux} e P_{aux}
    end if
  end if
end for

Após a avaliação da resposta obtida através do vetor J, o ajuste das probabilidades de sucesso associadas aos elementos que compõem a solução selecionada é realizado através da atribuição de um sinal de reforço R. Além de favorecer o desempenho almejado, R segue algumas outras características:

1. R favorece uma evolução do desempenho obtido, ou seja, o valor de R será positivo se o desempenho observado na iteração atual for superior a uma determinada média obtida pelo histórico de iterações realizadas;

2. R estimula apenas os resultados considerados "bons", em outras palavras, caso o reforço calculado (R_c) seja positivo, o mesmo é aplicado nas estimativas das probabilidades de sucesso das variáveis selecionadas, caso contrário, nenhum ajuste é realizado;

3. R_c deve estar compreendido entre os limites determinados pelo projetista viabilizando que o mesmo influencie na velocidade de convergência.

Dessa forma, R é determinado pela Expressão (C.1), onde R_G corresponde ao limite de reforço superior informado pelo projetista para R_c.

$$R = \begin{cases} R_G & \text{se } R_c \geq R_G \\ R_c & \text{se } R_G > R_c > 0 \\ 0 & \text{se } R_c \leq 0 \\ 0 & \text{se o robô cair.} \end{cases} \quad \text{(C.1)}$$

Já o valor de R_c é obtido através da Equação (C.2), onde R_P equivale ao limite de reforço inferior determinado pelo projetista e f_R representa a função de reforço.

$$R_c = f_R\left(\frac{R_G - R_P}{2}\right) + \frac{R_G + R_P}{2} \quad \text{(C.2)}$$

A função f_R é calculada a partir da Equação (C.3), onde:

1. $\overline{I_V^{Nc}}$, $\overline{I_W^{Nc}}$, $\overline{\tau_{max}^{Nc}}$ e $\overline{Ec^{Nc}}$ correspondem, respectivamente, às médias de I_V, I_W, τ_{max} e Ec nas últimas Nc iterações que não houve a queda do robô, sendo Nc determinado pelo projetista;
2. $\overline{V_+^{Nc}}$ corresponde à média das velocidades obtidas nas últimas Nc iterações onde as médias das velocidades $\overline{V_x}$ foram positivas;
3. F (Expressão (C.4)) é um vetor de pesos onde o projetista pode ajustar a influência que cada elemento do vetor J tem sobre a função f_R;
4. C_{fr} é uma constante calculada pela Equação (C.5) que é utilizada para equilibrar f_R de tal forma que seu valor tenda a 0 quando os valores do vetor J tenderem às médias das suas últimas Nc iterações.

$$f_R = \frac{1 + F_V \frac{\overline{V_x}}{V_+^{Nc}}}{1 + F_I\left(\frac{I_W}{I_W^{Nc}} + \frac{I_V}{I_V^{Nc}}\right) + F_\tau \frac{\tau_{max}}{\tau_{max}^{Nc}} + F_{Ec}\frac{Ec}{Ec^{Nc}}} - C_{fr} \quad \text{(C.3)}$$

$$F = [F_V \ F_I \ F_\tau \ F_{Ec}] \quad \text{(C.4)}$$

$$C_{fr} = \frac{1 + F(1)}{1 + F(2) + \sum_{i=2}^{4}(F(i))} \quad \text{(C.5)}$$

Determinado o reforço oriundo do comportamento gerado pelos parâmetros e defasagens selecionados, é necessário ajustar as probabilidades associadas à cada um desses elementos. Para tal, utiliza-se a Equação (C.6), onde:

- P_{k-1} corresponde a cada uma das estimativas de probabilidades de sucesso prévias das variáveis selecionadas (P_f, P_T e P_ϕ);
- P_K é a estimativa de probabilidade de sucesso após o ajuste, e
- Fcc é um fator de correção de convergência que tem por finalidade equilibrar o tempo de aprendizado entre as diferentes variáveis que compõem a solução testada.

$$P_K = P_{K-1}\left(1 + \frac{R}{100Fcc}\right) \quad \text{(C.6)}$$

O valor de Fcc é dado pela Expressão (C.7), onde:

- $Txc(P_i)$ representa a taxa de convergência da matriz de probabilidade, cujo valor é calculado como sendo a média das máximas probabilidades de cada coluna existente na matriz P_i;
- $min(Txc)$ é o valor mínimo dentre $Txc(P_f)$, $Txc(P_\phi)$ e $Txc(P_T)$.

$$Fcc = \begin{cases} 1 & se \ \frac{Txc(P_i)}{min(Txc)} \leq 1 \\ \frac{Txc(P_i)}{min(Txc)} & se \ \frac{Txc(P_i)}{min(Txc)} > 1 \end{cases} \quad \text{(C.7)}$$

Depois de ajustar as probabilidades dos parâmetros e defasagens utilizados na iteração em questão, todas as probabilidades de cada coluna são normalizadas fazendo com que a sua soma resulte em 1.

Neste artigo, em ambos os estudos de caso realizados, utiliza-se: $Nc = 20$, $R_P = -20$ e $R_G = 20$.

REFERÊNCIAS

Alexander, R. M. (1989). Optimization and gaits in the locomotion of vertebrates, *Physiological Reviews*, v. 69, n. 4, pp. 1199-1227.

Belter, D., Skrzypczynski, P. (2010). A biologically inspired approach to feasible gait learning for a hexapod robot, *Applied Mathematics and Computer Science*, v. 20, pp. 69-84.

Erden, M. S., Leblebicioglu, K. (2008). Free gait generation with reinforcement learning for a six-legged robot, *Robotics and Autonomous Systems*, v. 56, n. 3, pp. 199-212.

Golubovic, D., Hu, H. (2003). GA-based gait generation of Sony quadruped robots, 3^{th} *IASTED International Conference on Artificial Intelligence and Applications (AIA 2003)*, Benalmadena, Espanha, pp. 118-123.

Heinen, M. R. (2007). *Controle inteligente de caminhar de robôs móveis simulados*, Dissertação de Mestrado, Universidade do Vale do Rio dos Sinos, Porto Alegre, RS.

Heinen, M. R., Osório, F. S. (2008). Morphology and gait control evolution of legged robots, *IEEE Latin American Robotic Symposium (LARS 2008)*, Washington, DC, pp. 111-116.

Hopcroft, J. E., Motwani, R., Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation*, 3. ed., Addison Wesley, Hardcover.

Ijspeert, A. J. (2008). Central pattern generator for locomotion control in animals and robots: A review, *Neural Networks*, vol. 21, no. 4, pp. 642-653.

Kohl, N., Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion, *IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, pp. 2619-2624.

Mistry, M., Nakashi, J., Schaal, S. (2007). Task space control with prioritization for balance and locomotion, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, CA, USA, pp. 331-338.

Mitchell, T.M. (1997). *Machine Learning*, McGraw-Hill, New York, USA.

Morimoto, J., Cheng, G., Atkeson, C. G., Zeglin, G. (2004). A simple reinforcement learning algorithm for biped walking, *IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, pp. 3030-3035.

Murao, H., Tamaki, H., Kitamura, S. (2001). Walking pattern acquisition for quadruped robot by using modular reinforcement learning, *IEEE International Conference on Systems, Man and Cybernetics (SMC 2001)*, Tucson, AZ, USA v. 3, pp. 1402-1405.

Narendra, K. S., Thathachar, M. A. L. (1974). Learning automata - A survey, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-4, no. 4, pp. 323-334.

Nascimento Jr., C. L., Yoneyama, T. (2000). *Inteligência Artificial em Controle e Automação*, São Paulo, Editora Edgard Blücher.

Pfeifer, R., Scheier, C. (1999). *Understanding Intelligence*, MIT Press.

Plestan, F., Grizzle, J., Westervelt, E., Abba, G. (2003). Stable walking of a 7-dof biped robot, *IEEE Transactions on Robotics and Automation*, v. 19, n. 4, pp. 653-668.

Porta, J.M. (2000). Rho-learning: a robotics oriented reinforcement learning algorithm, Technical Report IRI-DT-00-03, *Institut de Robòtica i Informàtica Industrial, CSIC-UPC*. Disponível em http://www.iri.upc.edu/publications/show/520.

Porta, J. M., Celaya, E. (2004). Reactive free-gait generation to follow arbitrary trajectories with a hexapod robot, *Robotics and Autonomous Systems*, v. 47, n. 4, p. 187-201.

Santos, D., Siqueira, A. A. G. (2009). ADAMS/Matlab Co-simulation of an Exoskeleton for Lower Limbs, *International Congress of Mechanical Engineering (COBEM 2009)*, Gramado, RS.

Santos, J. L., Nascimento Jr., C. L. e Barbosa, L. F. W. (2010). Desenvolvimento de um sistema de aprendizado para o controle do caminhar de um robô utilizando aprendizado por reforço, *XVIII Congresso Brasileiro de Automática (CBA 2010)*, Bonito, MS, pp. 5024-5035.

Silva, M. F. e Machado, J. T. (2007). A historical perspective of legged robots, *Journal of Vibration and Control*, vol. 13, no. 9-10, pp. 1447-1486.

Siqueira, A. A. G., Jardim, B., Vilela, P. R., Winter, T. F. (2008). Analysis of Gait-Pattern Adaptation Algorithms Applied in an Exoskeleton for Lower Limbs, *16th Mediterranean Conference on Control and Automation*, Ajaccio, Corsica, France.

Spenneberg, D., McCullough, K., Kirchner, F. (2004). Stability of walking in a multilegged robot suffering leg loss, *IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, pp. 2159-2164.

Still, S., Douglas, R. J. (2006). Neuromorphic walking gait control, *IEEE Transactions on Neural Networks*, v. 17, pp. 496-508.

Sutton, R. S., Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, MIT Press.

Tal, D., Kallen, H., Atelier, E., Ch-Rufenach (2005). Robot and locomotion controller design optimization for a reconfigurable quadruped robot, *Universities Space Research Association / Research Institute for Advanced Computer Science at NASA Ames Research*. Disponível em http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.130.7181.

Thathachar, M. A. L., Sastry, P. S. (2003). *Networks of Learning Automata: Techniques for Online Stochastic Optimization*, Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Xu, K., Chen, X., Liu, W., Williams, M. (2006). Legged robot gait locus generation based on genetic algorithms, *International Symposium on Practical Cognitive Agents and Robots (PCAR 2006)*, New York, NY, USA, pp. 51-62.

Westervelt, E. R., Grizzle, J. W., Chevallereau, C., Choi, J. H. e Morris, B. (2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*, CRC Press.

Winter, T. F., Siqueira, A. A. G. (2008). Modelagem e Simulação de um Exoesqueleto para Membros Inferiores, *XVII Congresso Brasileiro de Automática (CBA 2008)*, Juiz de Fora, MG.

Yang, J. (2003). Fault-tolerant gait generation for locked joint failures, *IEEE International Conference on Systems, Man and Cybernetics (SMC 2003)*, Washington, DC, USA, pp. 2237-2242.