

# Um algoritmo de aprendizado iterativo baseado no ciclo de marcha humana para a locomoção de um robô humanoide

Fábio Suim Chagas<sup>a</sup>, Luis David Peregrino de Farias<sup>b</sup>, Matheus Bozza<sup>b</sup>, Paulo Fernando Ferreira Rosa<sup>a</sup>

<sup>a</sup>Instituto Militar de Engenharia (IME),

Praça General Tibúrcio, 80, 22290-270,

Praia Vermelha, Rio de Janeiro, RJ, Brasil.

<sup>b</sup>Centro Tecnológico do Exército (CTEx)

Av. das Américas - Guaratiba, Rio de Janeiro - RJ, 23032-050

\*fsuim@yahoo.com.br

**RESUMO:** Este artigo apresenta um algoritmo de aprendizado iterativo que prioriza a estabilidade estática em vez da estabilidade dinâmica, com a finalidade de diminuir a complexidade da marcha de um robô humanoide. Inicialmente, há um conjunto de posições almejadas que o algoritmo deve atingir durante o ciclo de marcha. Um processo de tentativa e erro conduz o método de aprendizado. Cada tentativa de movimento do robô fica disponível em uma lista de ações. Sempre que o robô alcança um objetivo, o algoritmo armazena a sequência de movimentos em uma memória. Se há uma falha, a lista de ações fornece a posição anterior à queda – e o processo recomeça a partir deste ponto. Com o objetivo de testar o algoritmo, foi desenvolvido um simulador usando o Matlab Simulink, juntamente com a biblioteca de forças de contato do Simscape Multibody. Os dados da simulação são apresentados através de gráficos que descrevem o comportamento das articulações durante o processo de aprendizado para um ciclo de marcha.

**PALAVRAS-CHAVE:** Robô Humanoide. Algoritmo de Aprendizagem. Cinemática. Simulador Humanoide.

**ABSTRACT:** This paper shows an iterative learning algorithm that prioritizes static over dynamic stability to decrease gait complexity for an articulated humanoid robot. Initially, we have an array of goal positions, which the algorithm must achieve during the gait cycle. A trial and error process leads to the learning approach. Each robot motion attempt is available on an action list. At the moment the robot achieves a goal, the algorithm stores the sequence of movements in memory. When a failure happens, the action list provides the position just before the fall - and the process starts from that point onwards. In order to test the algorithm, we developed a simulator using Matlab Simulink, together with the Simscape Multibody contact forces library. We present the simulation data through graphs that describe the behavior of the joints during the learning process of a gait cycle.

**KEYWORDS:** Humanoid robot. Learning Algorithm. Kinematic. Humanoid Simulator.

## 1. Introdução

Desde que o primeiro robô humanoide [1] foi apresentado pelo laboratório do professor Kato, na Universidade de Waseda nos anos 1970, o número de pesquisas relacionadas ao assunto cresce a cada dia. Nos dias de hoje, plataformas cada vez mais elaboradas e amigáveis estão sendo desenvolvidas em uma escala bem acima daquela década [2]-[5]. Isto pode ser explicado pela evolução e queda nos preços dos circuitos integrados (CIs) e outros componentes como os atuadores, além do fato do uso amplamente difundido da impressão 3D a um baixo custo.

Com relação ao uso de humanoides em aplicações militares, eles poderiam ser úteis em tarefas de resgate e desarmamento de bombas [6]-[7]. A locomoção similar aos seres humanos permitiria a passagem por terrenos irregulares como os rochosos ou montanhosos, substituindo robôs dotados de rodas ou esteiras. Em ambientes urbanos, poderiam se locomover por ruelas, subir escadas, escalar edifícios durante uma perseguição, agir contra um oponente de forma não letal, visto que não necessita preservar sua própria vida, como no caso de um soldado humano.

Um fator que ainda apresenta desafios no que diz respeito aos humanoides é a locomoção. Andar

com equilíbrio, ritmo e eficiência energética não é uma tarefa fácil [8]-[9]. Por mais de três décadas, esse assunto atraiu a atenção de pesquisadores das universidades e dos institutos mais prestigiados ao redor do mundo [10]-[12].

Até o momento, não há um modelo matemático completo que represente a marcha humana, mas formas alternativas de resolução do problema da locomoção são abordadas por modelos e algoritmos heurísticos, tais como: o pêndulo linear invertido [13] e o pêndulo linear invertido 3D [14]; os quais são de baixa ordem, e adicionalmente, quando combinadas com outras técnicas como o Zero Moment Point (ZMP) [15]-[16] melhoram substancialmente a solução do problema. Apesar da evolução nessa área ao longo dos anos, a dinâmica da marcha humana ainda demanda desenvolvimento. Locomoção com sincronismo, equilíbrio e economia de energia é ainda um campo a ser explorado.

O principal objetivo deste artigo é apresentar um algoritmo de aprendizagem para o problema do caminhar em robôs bípedes baseado na marcha humana. A principal vantagem está no fato da solução ser iterativa o que simplifica a obtenção das equações cinemáticas da marcha.

O artigo está organizado da seguinte forma: a Seção 2 apresenta uma revisão de literatura; a Seção 3 mostra o plano de fundo teórico; a Seção 4 mostra o algoritmo proposto; a Seção 5 mostra os experimentos e resultados; e finalmente, a Seção 6 apresenta a conclusão e os próximos passos.

## 2. Revisão de literatura

No estudo da locomoção robótica, a marcha pode ser classificada como estaticamente e dinamicamente estável. No caminhar estático, a projeção do centro de massa do robô está na região do pé: se o robô se movimenta lentamente para garantir isso, o caminhar é sempre estável. O problema é o alto consumo de energia e perda dos movimentos dinâmicos [17][18].

No caminhar dinâmico, o centro de massa do robô assume a posição fora do ponto de suporte e não é, portanto, dinamicamente estável. McGeer [19]

mostrou que certos sistemas mecânicos possuem uma tendência natural para caminhar e chamou isso de caminhada passiva dinâmica. Nesta caminhada, o dispositivo usa as forças gravitacionais para se mover usando uma pequena inclinação [21], [22]. Como resultado, possui as vantagens da eficiência energética e de dinamismo do movimento. Em contrapartida, ela se restringe ao plano sagital e não obterá sucesso sobre um piso desnivelado.

Outro fator preponderante na marcha dinamicamente passiva é a alta não-linearidade do modelo, o que dificulta o projeto dos robôs caminhantes. A observação biológica dos seres vivos, juntamente com ajustes por experimentação, auxilia a modelagem da tarefa de caminhar, como mostrado em [22]-[23]. Devido a não-linearidade da marcha e a dificuldade de obter um modelo, aproximações por algoritmos heurísticos são desejáveis na solução do problema.

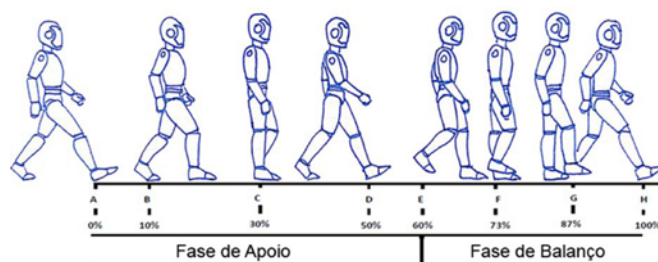


Fig. 1 - Ciclo de marcha (Adaptado de Magee [27]).

Nesse cenário, o aprendizado por reforço emerge como uma opção para resolver o problema. Pesquisas, como as feitas por Yamaguchi *et al* [24], investigaram o algoritmo Q-Learning para o controle de um robô 50 GDL. Eles compararam uma abordagem por aprendizado por reforço hierárquico e uma função aproximada. Como resultado, eles obtiveram um aprendizado apropriado em um intervalo de 20 a 30 episódios. Kulk e Welsh [25] propuseram um algoritmo baseado em funções que se opõem e em funções redundantes de adequação, focando em três componentes principais: velocidade, eficiência e estabilidade. O algoritmo demonstrou uma melhora na velocidade e eficiência para realizar o ciclo de marcha aprendido.

Outros fatores preponderantes são os simuladores. O desenvolvimento do aprendizado por reforço baseado no algoritmo de locomoção humanoide exige uma gama de experimentações e os simuladores aceleram o resultado. Para muitas situações, o algoritmo é incompatível com o hardware. Em Oliveira *et al* [26], há uma tentativa de criar uma função para transferir rotinas de um ambiente simulado para o robô real. Infelizmente, algumas funções simuladas não funcionam no sistema real e ainda precisam de melhorias.

Neste artigo, é proposto um algoritmo iterativo de inteligência de máquina, o qual não requer o uso de um modelo matemático de ordem superior. Foi adotada a estratégia de aprendizagem observacional. Um conjunto de pontos sobre o corpo humano, formado por posições angulares capturadas durante a marcha, compõe uma matriz de objetivos que o robô deve alcançar.

### 3. Tópico tutorial

Esta sessão descreve brevemente o ciclo da marcha humana e o comportamento da mesma nos planos sagital, frontal e transversal.

#### 3.1 O ciclo de marcha

O ciclo de marcha é uma sequência de posições que o corpo toma para se mover de um ponto a outro. Em um caminhar normal, isso ocorre suave e ciclicamente. O começo do processo ocorre quando o calcanhar da perna que está à frente toca o chão. A sequência termina quando o mesmo calcanhar, que começou o ciclo, toca o solo novamente.

O ciclo ocorre em duas fases, sete períodos e oito eventos. Duas fases que dividem o ciclo de marcha são a fase de apoio (ou suporte) e a fase de oscilação (ou balanço), sendo que a fase de oscilação é mais rápida que as anteriores. A fase de suporte inclui os eventos de contato inicial e calcanhar livre, além das fases de apoio médio e de apoio final. O contato inicial oposto acontece em duas fases de transição. O evento dedos dos pés livres, pés adjacentes, tibia vertical, e próximo contato inicial estão na fase de balanço. Pré-balanço, balanço inicial, balanço médio e balanço final são os períodos da fase de balanço, conforme a **figura 1**.

Inicialmente, o movimento começa com o contato do

calcanhar com o solo (**figura 1 A**). Na sequência, o pé fica estável no solo para receber a carga de resposta do peso e corresponde a do ciclo (**figura 1 B**). Quando o calcanhar oposto ao que começou o movimento está fora do solo, atinge-se do ciclo e no evento de calcanhar livre (**figura 1 C**). A perna oposta se move para frente, e o calcanhar oposto toca o chão, do ciclo de marcha é alcançado. O evento correspondente é o contato inicial oposto (**figura 1 D**). Daqui em diante, é iniciada a fase de pré-balanço, quando o calcanhar da perna que iniciou o ciclo está fora do solo é atingido do ciclo (**figura 1 E**). Quando os pés estão adjacentes o ciclo, encontra-se em (**figura 1 F**) e quando a tibia está vertical (**figura 1 G**) e o movimento desacelera. O que determina o fim do ciclo é o início de um novo contato do calcanhar da perna que iniciou o movimento, (**figura 1 H**). Vale a pena mencionar que a fase de balanço é mais rápida que a fase de apoio, e dura de todo o ciclo.

### 4. Algoritmo de aprendizado

Durante o ciclo de marcha as articulações assumem várias posições angulares e isso pode ser utilizado para construir a matriz de objetivo do algoritmo. As posições acessadas por captura são: os tornozelos direito e esquerdo, o quadril no início do fêmur, direito e esquerdo; o centro de massa e os ombros direito e esquerdo. A **figura 2** ilustra as articulações a serem analisadas, onde as letras de a, b, c, d, e, f, g, h e i indicam a posição dos sensores no corpo.

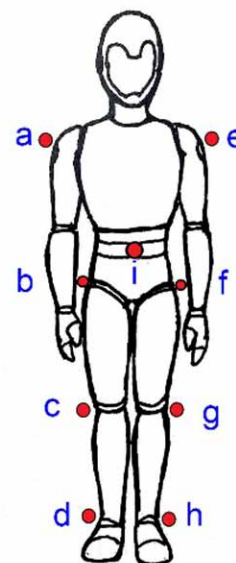


Fig. 2 - Distribuição dos sensores no corpo humano.

**Tab. 1** – Descritores analisados durante o ciclo de marcha.

	Tornozelo		Joelho		Quadril		CG	Ombro	
	Esquerdo	Direito	Esquerdo	Direito	Esquerdo	Direito		Esquerdo	Direito
Evento A	$al(\theta_x, \theta_y, \theta_z)$	$ar(\theta_x, \theta_y, \theta_z)$	$kl(\theta_x, \theta_y, \theta_z)$	$kr(\theta_x, \theta_y, \theta_z)$	$hl(\theta_x, \theta_y, \theta_z)$	$hr(\theta_x, \theta_y, \theta_z)$	$c(\theta_x, \theta_y, \theta_z)$	$sl(\theta_x, \theta_y, \theta_z)$	$sr(\theta_x, \theta_y, \theta_z)$
Evento B									
Evento C	.	.	.	.	.	.	.	.	.
Evento D	.	.	.	.	.	.	.	.	.
Evento E	.	.	.	.	.	.	.	.	.
Evento F									
Evento G									
Evento H	$al(\theta_x, \theta_y, \theta_z)$	$ar(\theta_x, \theta_y, \theta_z)$	$kl(\theta_x, \theta_y, \theta_z)$	$kr(\theta_x, \theta_y, \theta_z)$	$hl(\theta_x, \theta_y, \theta_z)$	$hr(\theta_x, \theta_y, \theta_z)$	$c(\theta_x, \theta_y, \theta_z)$	$sl(\theta_x, \theta_y, \theta_z)$	$sr(\theta_x, \theta_y, \theta_z)$

A **tabela 1** apresenta os descritores analisados durante o ciclo de marcha. A sequência é dividida em oito eventos que vão de A até H. Em cada um dos quais é possível obter os valores dos ângulos  $\theta_x$ ,  $\theta_y$  e  $\theta_z$  para cada uma das articulações como mostrado na **figura 1**.

A nomenclatura  $al$  é para o tornozelo esquerdo,  $ar$  é para o tornozelo direito,  $kl$  é para o joelho esquerdo,  $kr$  é para o joelho direito,  $c$  é para o centro de massa (CG),  $sl$  ombro esquerdo e  $sr$  ombro direito. A terna  $(\theta_x, \theta_y, \theta_z)$  representa a variação angular nos eixos  $x$ ,  $y$  e  $z$ . O conjunto nomeado como  $q$  representa as articulações  $al$ ,  $ar$ ,  $kl$ ,  $kr$ ,  $c$ ,  $hl$ ,  $hr$ ,  $sl$ ,  $sr$ .

Seja  $[\psi_i]_{t_i}$  o conjunto de todas as combinações angulares de todas as articulações  $q$ , mostrado em .

$$[\psi_i]_{t_i} = \{q\}_{t_i} \quad (1)$$

Durante o ciclo de caminhada obteve-se uma matriz  $\xi$  (2), na qual as colunas representam as variações angulares das articulações no instante. As linhas representam os eventos, períodos e fases do ciclo de caminhada.

$$\xi = \{[\psi_i]_{t_1}, [\psi_i]_{t_2}, \dots, [\psi_{n-1}]_{t_{n-1}}, [\psi_n]_{t_n}\} \quad (2)$$

Durante um evento, é possível obter tantos conjuntos  $\xi$  quanto o sistema de captura é capaz de processar, fazendo o número de conjuntos  $\xi$  estar diretamente associado ao poder computacional

disponível.

É importante perceber que nem todos os estados alcançados são válidos. Portanto, é necessário encontrar um subconjunto de posições dentro de  $\xi$  que leva às posições de equilíbrio. Nesse caso, busque por um subconjunto como descrito na **equação 3**.

$$\exists \Lambda \in \xi, \Lambda = (\lambda_1, \lambda_2, \dots, \lambda_{n-1}, \lambda_n) \quad (3)$$

onde,  $\Lambda$  é o subconjunto de estados estáveis de  $\xi$  e esses estados são representados por  $\lambda$ .

As posições  $\lambda_i$  são responsáveis por suprir os valores da matriz de objetivo  $G$  do algoritmo . Seja  $n(G)$  o número de linhas da matriz de objetivos e  $n(\Lambda)$  o número de estados válidos ou posições de equilíbrio que o corpo assume durante o ciclo de marcha, onde  $n(G) \leq n(\Lambda)$ .

#### 4.1 Fluxograma do algoritmo de aprendizado

A **figura 3** mostra o fluxograma do algoritmo proposto. Inicialmente, é definido o tamanho da passada e a discretização angular para atingir o objetivo. O bloco 1 mostra a matriz objetivo  $G$  com as posições desejadas.  $G$  é composta pela seleção de algumas posições das fases do ciclo de marcha.

No Bloco II, a função recebe o primeiro objetivo  $e$ , a partir desse ponto, o sistema cria uma lista de ações. Depois, o programa checa se o valor coincide com o último valor de . Se o teste falhar, o programa executa o próximo passo.

---

 Algoritmo 1: Pseudocódigo para o algoritmo de aprendizagem
 

---

 1 Aprendizagem-por-iteração ( $q$ ;  $G$ ;  $inc$ ;  $deg$ )

**Dados:**
**Entradas:** Número de ciclos, tamanho do passo,  
 $q$ : Conjunto de juntas;  $G$ : Matriz de objetivos;  $inc$ : tempo gasto nas interações;  $deg$ : verifica se  $q$  alcançou objetivo;

**Saída:**  $M$  é a memória do robô.

2 Inicialização;

 3 **repita**

 4 Pega o objetivo ( $Get\_Goal(G(i))$ );

5 Coloca a informação na Lista de Ações;

 6 **Enquanto**  $l < deg$  **Faça**

 7 **Para**  $i = 2$  até o número de juntas do robô **Faça**

 8 **Se**  $q(i) < G(i)$  **Então**

 9  $q$  é decrementado por (5);

 10  $checkforfall()$ ;

 11 **Se**  $q(i) > G(i)$  **Então**

 12  $q$  é incrementado por (6);

 13  $checkforfall()$ ;

 14 **Senão**

 15  $M = q$ ;

 16 **Fim**

 17 **Fim**

 18 **Até**  $G$  seja terminal;
 

---

**Procedimento**  $checkforfall()$ 


---

**Saída:** Lista de Ações atualizada

 1 **Se** Condição de Equilíbrio Falhou **Então**

 2  $q \leftarrow q(i-1)$ ;

 3  $M \leftarrow q$ ;

4 Termina a Aplicação;

 5 **Senão**

6 Atualiza a lista de Ações;

 7 **Fim**


---

O valor  $deg$  é o número de vezes que o algoritmo executa o teste para verificar que o valor de  $q$  alcançou o valor alvo. Se o valor de  $q(i)$  excede o valor de  $G(i)$ , isto é, o valor calculado excede o valor alvo, o valor de  $q(i)$  é decrementado obedecendo à regra vista em (5). Por outro lado, se o valor de  $q(i) < G(i)$ , então o valor de  $q(i)$  é incrementado pela regra dada por (6). A função aleatória  $k$  garante que os valores de  $q$  não sejam repetidos no evento de uma queda. Ela pode assumir o valor 0 ou 1.

$inc$  (4) é um conjunto contendo o tempo que cada articulação  $q(i)$  vai permanecer durante sua atualização.  $\sigma_i$ , em uma situação real, é a frequência de operação do atuador na articulação do robô.

$$inc = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{n(G)}\} \quad (4)$$

$$q(i) = q - \left[ \left( \frac{|G - inc|}{deg} \right) * k \right] \quad (5)$$

$$q(i) = q + \left[ \left( \frac{|G - inc|}{deg} \right) * k \right] \quad (6)$$

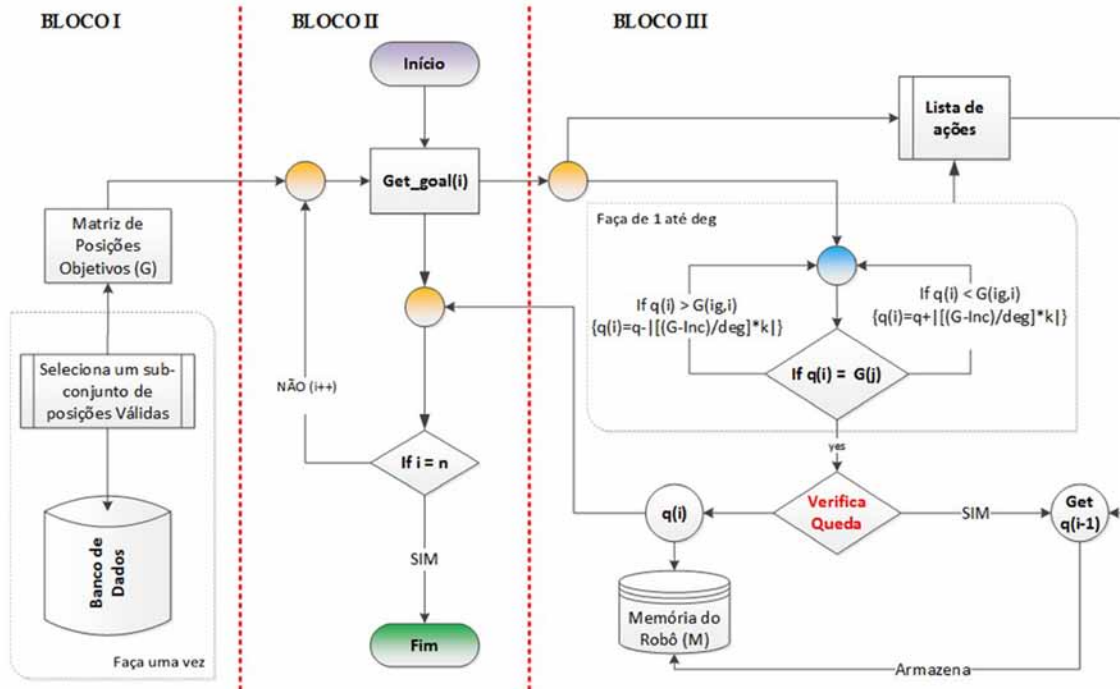


Fig. 3 - Fluxograma do algoritmo de aprendizado.

Durante esse processo, o algoritmo checa se o robô cai. Se o alvo é atingido, a matriz  $M$  armazena os valores de  $q$  e a função *Get\_goal* recebe o próximo objetivo e repete o processo.

Se o objetivo não for alcançado, o algoritmo recebe o valor de  $q(i - 1)$  da lista de ações e o armazena na matriz  $M$ . Então ele tenta alcançar o objetivo a partir do ponto de queda, repetidamente até alcançar sucesso. Após atingir o objetivo, a matriz  $M$  é atualizada. Quando o objetivo final é alcançado, o processo para e o aprendizado para um ciclo de caminhada está completo.

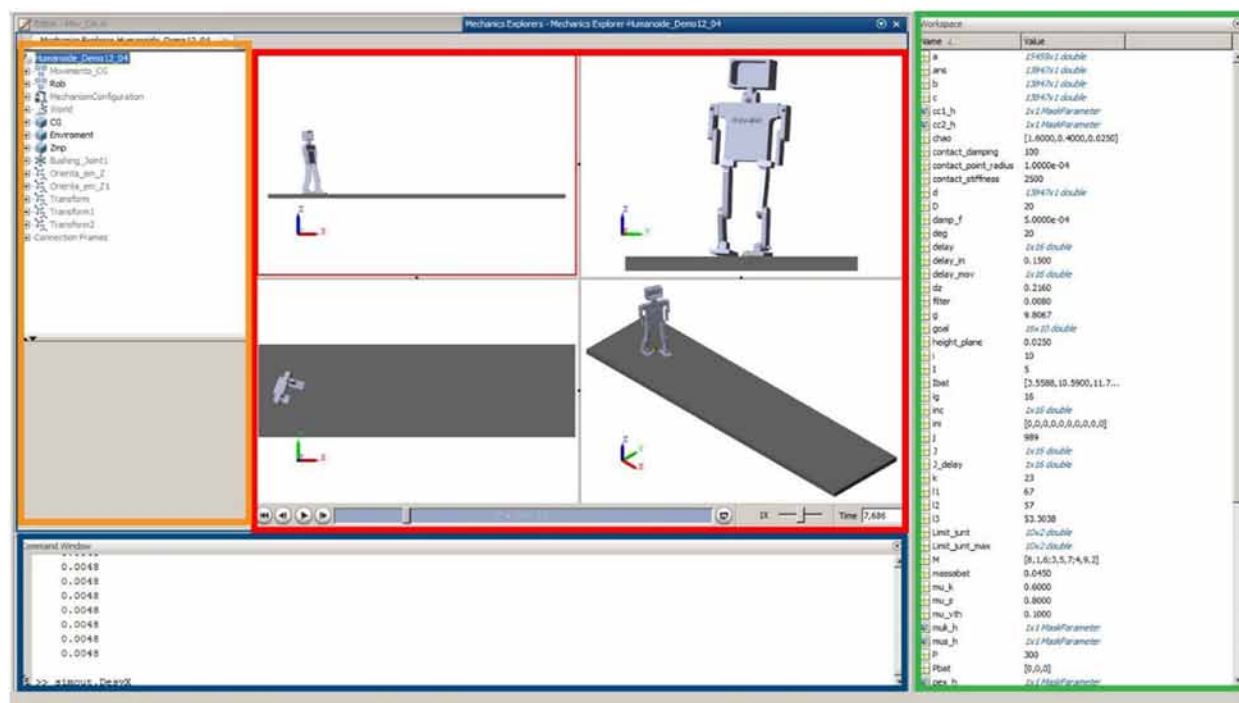
## 5. Experimentos e resultados

Com o objetivo de testar o algoritmo, fez-se necessário o desenvolvimento de um Dispositivo de Captura de Dados de Movimento Humano (DCMH) para montar a matriz objetivo e um simulador. O DCMH consiste de um conjunto de nove sensores inerciais de seis eixos (MPU-6050), conectados a uma placa Raspberry pi 3.

Foram medidos os ângulos dos tornozelos, esquerdo e direito no plano sagital (lateral) e frontal. Os ângulos dos joelhos, esquerdo e direito. Os ângulos, esquerdo e direito, dos quadris nos planos sagital, frontal e Centro de Gravidade. A **figura 4** mostra a configuração feita para o experimento com o propósito de coletar dados durante o ciclo de marcha.



**Fig. 4** – Esquerda marcadores da posição dos sensores; direita dispositivo montado no corpo humano.



**Fig. 5** - A interface gráfica do simulador: O retângulo laranja indica os objetos do robô. O retângulo vermelho mostra as vistas Lateral, Frontal, Superior e Isométrica. O retângulo verde disponibiliza as informações variáveis após a execução do simulador e o retângulo azul - mostra a janela de comandos.

## 5.1 Simulador

A **figura 5** mostra o simulador desenvolvido usando o *Simulink* na versão 2018 do *MatLab* e a biblioteca de forças de contato do *Simscape Multibody*. Para modelar o robô, foi usado o *SolidWorks* 2017. A simulação exigiu um processador Intel(R) Xeon(R) CPU E7- 4870, com *clock* de 2.4 GHz, 32 núcleos, 32 GB de RAM, e uma placa de vídeo Cirrus Logic GD 5446. A **tabela 2** mostra a disposição das juntas do robô.

**Tab. 2** – Disposição das juntas no corpo do robô

Membro	Graus de liberdade	Movimento
Pé direito	2	Flexão/dordiflexão, inversão/eversão.
Pé esquerdo	2	Flexão/dordiflexão, inversão/eversão.
Joelho direito	1	Flexão, extensão.
Joelho esquerdo	1	Flexão, extensão.
Quadril direito	2	Flexão, rotação.
Quadril esquerdo	2	Flexão, rotação.
Ombro direito	2	Flexão e rotação externa.
Ombro esquerdo	2	Flexão e rotação externa.
Cotovelo direito	1	Flexão do antebraço e extensão
Cotovelo esquerdo	1	Flexão do antebraço e extensão
Pescoço esquerdo	1	Rotação esquerda e direita.

Para a simulação, foram analisados somente os ângulos dos membros inferiores. Os dados foram colhidos de um modelo humano e adaptados para gerar a matriz objetivo para a caminhada do robô. A **tabela 3** mostra a matriz objetivo. Ao todo são 16 objetivos a serem cumpridos ao longo de um passo. Os valores negativos para o pé significam uma inversão, já os positivos uma eversão; para o tornozelo, valores negativos representam uma flexão plantar e os positivos uma dorsiflexão. A flexão do joelho é sinalizada por um valor negativo, já a extensão por um valor positivo; para a coxa valores negativos significam flexão e os positivos uma extensão. Valores positivos e negativos para o quadril representam uma rotação interna e externa respectivamente.

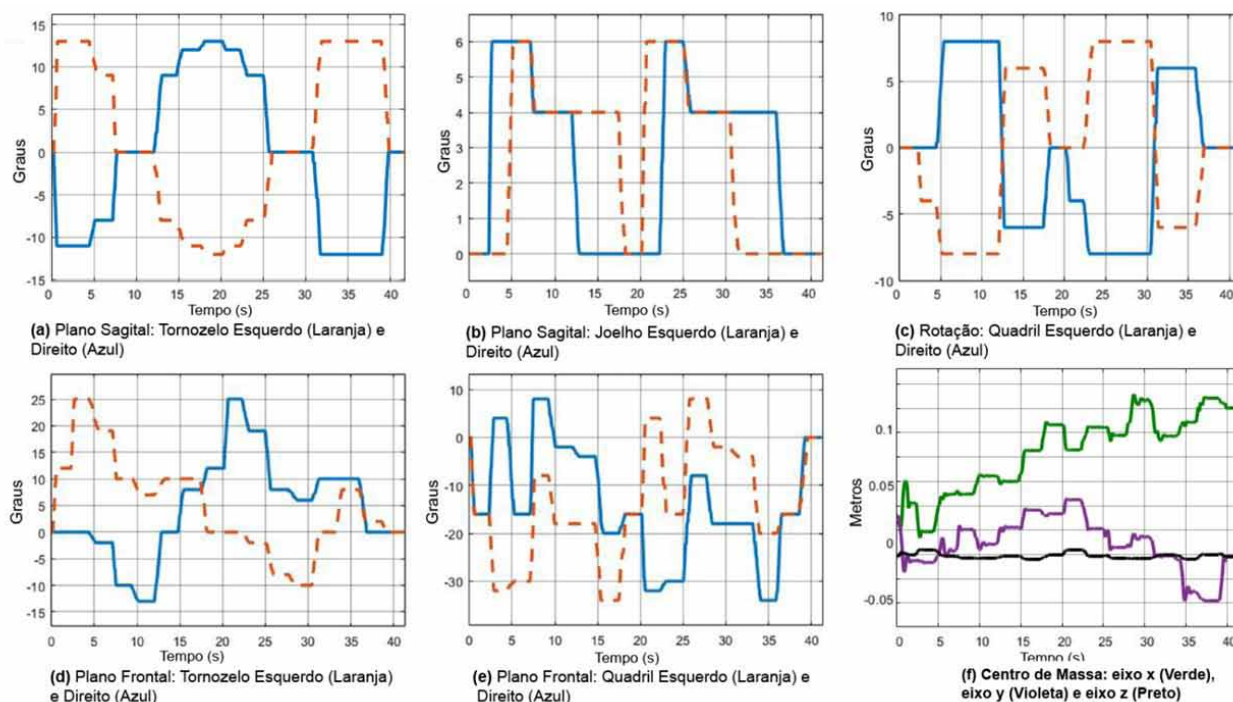
**Tab. 3** – Matriz de objetivos – entrada de dados para o algoritmo

Eventos	Lado direito					Lado esquerdo				
	Pé	Tornozelo	Joelho	Coxa	Quadril	Pé	Tornozelo	Joelho	Coxa	Quadril
1	-11	0	0	-16	0	13	12	0	-16	0
2	-11	0	6	4	0	2	25	0	-32	-4
3	-8	-2	6	-16	8	9	19	6	-30	-8
4	0	-10	4	8	8	0	10	4	-8	-8
5	0	-13	4	-2	8	0	7	4	-18	-8
6	13	0	0	-4	-6	-12	10	4	-18	6
7	13	8	0	-20	-6	-12	10	4	-34	6
8	13	12	0	-16	0	-12	0	0	-16	0
9	2	25	0	-32	-4	-11	0	6	4	0
10	9	19	6	-30	-8	-8	-2	6	-16	8
11	0	8	4	-8	-8	0	-8	4	8	8
12	0	6	4	-18	-8	0	-10	4	-2	8
13	-12	10	4	-18	6	13	0	0	-4	-6
14	-12	10	4	-34	6	13	8	0	-20	-6
15	-12	0	0	-16	0	13	2	0	-16	0
16	0	0	0	0	0	0	0	0	0	0

## 5.2 Resultados

A **figura 6** mostra o resultado obtido durante o treinamento do robô para um ciclo de marcha, no qual foram selecionados 16 objetivos. Os eixos representam a informação do tempo de viagem e a variação angular das articulações, respectivamente. As linhas laranja e azul representam as articulações nos lados direito e esquerdo do humanoide. As linhas verde, violeta e preta indicam a variação do centro de massa nos eixos.

Inicialmente o robô está com as duas pernas paralelas e inicia o movimento com a perna direita à frente. As linhas azuis e laranjas representam os membros inferiores do lado direito e esquerdo respectivamente. A **figura 6 (a)** mostra o comportamento dos tornozelos no plano sagital. No início do movimento, ambos os tornozelos sofrem uma variação, devido ao ajuste de todo o corpo do robô. Quando ambos estão paralelos eles estão na posição 0°. A **figura 6 (b)** mostra os movimentos de inversão e eversão sofridas durante a caminhada.



**Fig. 6** - Representação do comportamento das articulações durante o processo de aprendizado.

A **figura 6 (c)** mostra o comportamento do quadril no plano sagital. A amplitude máxima atingida é de  $8^\circ$  para ambos os lados, o comportamento dos lados direito e esquerdo são espelhados em quase todo ciclo. Já a **figura 6 (d)** mostra a análise do quadril no plano frontal, inicialmente as duas curvas variam de  $0^\circ$  a  $-15^\circ$  isto indica o momento de ajuste que o robô fez no início do ciclo de caminhada; no decorrer do percurso as curvas oscilam de forma espelhada, e no final da caminhada retornam a posição zero.

A **figura 6 (e)** mostra a análise dos joelhos. Inicialmente, eles estão paralelos na posição zero, à amplitude máxima atingida são 6 graus, as curvas apresentam um comportamento simétrico. O segmento de curva que o joelho direito faz no intervalo de 4 a 15 segundos é idêntico ao movimento do joelho esquerdo no intervalo de 20 a 35 segundos.

A **figura 6 (f)** diz respeito ao comportamento do Centro de Massa do robô nos eixos x, y e z que são representados pelas linhas verde, violeta e preta

respectivamente. As variações sofridas pelo centro de massa no eixo z o robô são sutis. No eixo y as variações máximas são de aproximadamente 0,04 metros para ambos os lados, isso acontece devido à necessidade de compensar o movimento de rotação do quadril. No eixo x o gráfico apresenta um desvio de 0,15 m do objetivo final que era caminhar em linha reta.

## 6. Conclusão

Neste artigo foi apresentada uma solução para o problema da caminhada baseada em um método iterativo. O algoritmo se inspira no andar humano, em que determinados pontos das posições de equilíbrio estático do ciclo de marcha, norteiam a construção de uma matriz de objetivos, os quais o robô deve atingir.

Inicialmente, desenvolveu-se um dispositivo de captura de movimentos do corpo humano (DCMH) através de sensores de inerciais de medidas, esses sensores foram dispostos na parte inferior dos membros para capturar as informações dos ângulos



articulares dos tornozelos, joelhos e quadril. Inspirados nas curvas obtidas com o DCMH, foi possível estimar posições para preencher uma matriz de objetivos, onde as colunas da matriz representam os valores angulares das juntas e as linhas a posição no ciclo de marcha.

O algoritmo segue a premissa de que há pontos que levam o robô a uma posição de equilíbrio durante o ciclo de marcha. Esses pontos são dispostos na matriz de objetivos. O algoritmo parte da posição inicial onde o robô está na posição ereta e com os pés paralelos; o valor da posição de referência das juntas é zero. A seguir, uma função pega a primeira posição da matriz e através de um processo incremental tenta chegar na posição alvo. Durante o processo pode haver quedas; caso aconteça, o robô armazena a última posição de sucesso e reinicia o processo a partir do ponto de queda.

Um simulador desenvolvido no Matlab possibilitou a execução da simulação computacional. O algoritmo convergiu para um ciclo de marcha após 6 tentativas utilizando uma matriz de dezesseis objetivos.

A principal vantagem do algoritmo é a de que não foi necessário resolver nenhuma equação diferencial,

o que simplifica a modelagem matemática do problema. A desvantagem é que a solução apresentada pertence ao conjunto das soluções numéricas, e estas estão sujeitas à singularidades, que resultam em uma falha na obtenção de uma solução, mesmo que exista. Além disso, a solução iterativa não é adequada para a execução em tempo real; o número de iterações é limitado e, portanto, elas podem não convergir. A análise desses fatores sugere uma mudança na tratativa do problema. A direção da pesquisa segue para uma solução analítica, que será abordada em trabalhos futuros.

## Agradecimentos

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001. O presente trabalho utilizou a infraestrutura computacional do Laboratório de Computação de Alto Desempenho para Defesa Cibernética, financiado com recursos dos Projetos Estratégicos do Exército Brasileiro. Os autores agradecem João Carlos José dos Santos e o Cel Ney Bruno pelo seu suporte com os experimentos.

## Referências bibliográficas

- [1] B. Siciliano and O. Khatib, Springer Handbook of Robotics, 2nd ed. Springer Publishing Company, Incorporated, 2016.
- [2] D. Gouaillier et al., "Mechatronic design of NAO humanoid," 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 769-774, doi: 10.1109/ROBOT.2009.5152516.
- [3] K. Kaneko et al., «Design of prototype humanoid robotics platform for HRP,» IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 2431-2436 vol.3, doi: 10.1109/IRDS.2002.1041632.
- [4] Ill-Woo Park, Jung-Yup Kim, Jungho Lee and Jun-Ho Oh, «Mechanical design of humanoid robot platform KHR-3 (KAIST Humanoid Robot 3: HUBO),» 5th IEEE-RAS International Conference on Humanoid Robots, 2005., 2005, pp. 321-326, doi: 10.1109/ICHR.2005.1573587.
- [5] J. Chestnutt, P. Michel, J. Kuffner and Takeo Kanade, «Locomotion among dynamic obstacles for the honda ASIMO,» 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2572-2573, doi: 10.1109/IROS.2007.4399431.
- [6] J. Lim et al., «Robotic software system for the disaster circumstances: System of team KAIST in the DARPA Robotics Challenge Finals,» 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 1161-1166, doi: 10.1109/HUMANOIDS.2015.7363509.
- [7] L. Pineda, T. Takahashi, H. Jung, S. Zilberstein and R. Grupen, «Continual planning for search and rescue robots,» 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 243-248, doi: 10.1109/HUMANOIDS.2015.7363542.
- [8] N. Banerjee et al., «Human-supervised control of the ATLAS humanoid robot for traversing doors,» 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 722-729, doi: 10.1109/

HUMANOIDS.2015.7363442.

- [9] H. Jeong, O. Sim, H. Bae, K. Lee, J. Oh and J. Oh, «Biped walking stabilization based on foot placement control using capture point feedback,» 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 5263-5269, doi: 10.1109/IROS.2017.8206418.
- [10] Y. Minami Shiguematsu, M. Brandao, K. Hashimoto and A. Takanishi, «Effects of Biped Humanoid Robot Walking Gaits on Sparse Visual Odometry Algorithms,» 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), 2018, pp. 160-165, doi: 10.1109/HUMANOIDS.2018.8625015.
- [11] S. Kajita et al., «Position-Based Lateral Balance Control for Knee-Stretched Biped Robot,» 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), 2019, pp. 17-24, doi: 10.1109/Humanoids43949.2019.9035077.
- [12] G. Nava et al., «Modeling and control of humanoid robots in dynamic environments: ICub balancing on a seesaw,» 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), 2017, pp. 263-270, doi: 10.1109/HUMANOIDS.2017.8246884.
- [13] S. Kajita and K. Tani, «Study of dynamic biped locomotion on rugged terrain-theory and basic experiment,» in *Advanced Robotics, 1991. Robots in Unstructured Environments*, 91 ICAR., Fifth International Conference on. IEEE, 1991, pp. 741-746.
- [14] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, «The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation,» in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, 2001, pp. 239–246 vol.1.
- [15] M. Vukobratović and B. Borovac, «Zero-moment point – thirty five years of its life,» *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2005. [Online]. Available: <http://www.cs.cmu.edu/~cga/legs/vukobratovic.pdf>
- [16] S. Kajita et al., «Biped Gait Control Based on Spatially Quantized Dynamics,» 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 2018, pp. 75-81, doi: 10.1109/HUMANOIDS.2018.8624942.
- [17] A. S. Baskoro and M. G. Priyono, «Design of humanoid robot stable walking using inverse kinematics and zero moment point,» 2016 International Electronics Symposium (IES), 2016, pp. 335-339, doi: 10.1109/ELECSYM.2016.7861027.
- [18] T. K. Hooi and M. N. Mahyuddin, «A study of walking gait stability and gait efficiency of a cost-effective small humanoid bipedal robot: Analysis, simulation and implementation,» 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 2017, pp. 125-129, doi: 10.1109/IRIS.2017.8250109.
- [19] T. McGeer, «Passive dynamic walking,» *the international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990.
- [20] T. Aoyama, K. Sekiyama, Z. Lu, T. Kobayashi, Y. Hasegawa, and T. Fukuda, «Stability enhancement of 3-d biped walking based on passive dynamic autonomous control,» in 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Nov 2012, pp. 443–448.
- [21] H. Suzuki, J. H. Lee and S. Okamoto, «Development of semi-passive biped walking robot embedded with CPG-based locomotion control,» 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, 2017, pp. 75-78, doi: 10.1109/URAI.2017.7992889.
- [22] K. Hashimoto, Y. Takezaki, H. Motohashi, T. Otani, T. Kishi, H. o. Lim, and A. Takanishi, «Biped walking stabilization based on gait analysis,» in 2012 IEEE International Conference on Robotics and Automation, May 2012, pp. 154–159.
- [23] C. Piazza et al., «Toward an adaptive foot for natural walking,» 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 1204-1210, doi: 10.1109/HUMANOIDS.2016.7803423.
- [24] A. Yamaguchi, S. Hyon, and T. Ogasawara, «Reinforcement learning for balancer embedded humanoid locomotion,» in 2010 10th IEEE-RAS International Conference on Humanoid Robots, Dec 2010, pp. 308–313.
- [25] J. Kulk and J. S. Welsh, «Using redundant fitness functions to improve optimisers for humanoid robot walking,» in 2011 11th IEEE-RAS International Conference on Humanoid Robots, Oct 2011, pp. 312-317.
- [26] M. Oliveira, S. Doncieux, J. Mouret, and C. P. Santos, «Optimization of humanoid walking controller: Crossing the reality gap,» in 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Oct 2013, pp. 106-111.
- [27] D. Magee, *Orthopedic Physical Assessment*, 6th ed., ser. *Orthopedic Physical Assessment*. Saunders Elsevier, 2013, no. v. 1.