

# Algoritmo de Caminhada em Malha Fechada para Robô Humanoide Darwin

Rafael Sebastião Miranda, Rodrigo da Silva Guerra

**Abstract**— *A caminhada de robôs humanoides bípedes representa um grande desafio para utilização destes robôs em ambientes não adaptados. A copa de futebol de robôs, RoboCup, tem como um de seus objetivos incentivar a pesquisa na área e a partir edição de 2015 passou a utilizar grama sintético no campo de jogo, o que dificultou a manutenção do equilíbrio dos robôs. Este trabalho descreve modificações no algoritmo de caminhada em malha aberta proposto por Marcell Missura em 2013 que visam o controle do equilíbrio em malha fechada. Para isso foram feitas duas alterações baseadas na percepção de postura: (1) controle de fase do movimento com base na inclinação lateral (roll) e (2) mudança na abordagem do sinal de referência de velocidade sagital para inclinação sagital (pitch). Estas alterações resultaram em um controle de equilíbrio mais estável, porém com deslocamento mais lento e, por vezes, oscilatório. Acredita-se que com algumas melhorias o algoritmo possa ser suficientemente estável e eficiente para utilização do mesmo em edições seguintes.*

**Index Terms**— Controle de robô e locomoção bípede

## I. INTRODUÇÃO

ROBÔS bípedes representam um grande desafio para manutenção de seu equilíbrio, pois estes possuem uma dinâmica bastante complexa para ser modelada. Tendo isto em vista, muitos trabalhos simplificam o modelo do robô utilizando analogia com o problema já amplamente estudado do pêndulo invertido [1]. A maioria dos algoritmos propostos, pressupõem um piso horizontal e liso; nestas condições é possível modelar a reação do robô ao chão e desenvolver algoritmos de caminhada em malha aberta com desempenho aceitáveis. Em 2015, porém, a principal competição mundial de futebol de robôs, a *Robo Cup*, optou por utilizar grama sintético em suas partidas. Isto fez com que os algoritmos que pressupunham solo liso tivessem seu desempenho fortemente prejudicado. Com intuito de buscar uma caminhada estável para a competição, modificou-se o algoritmo proposto por Marcell Missura [2] e implementou-se o mesmo em um robô DARwIn-OP da equipe Taura Bots, da Universidade Federal de Santa Maria.

Este trabalho tem por objetivo documentar o algoritmo de caminhada utilizado pela equipe Taura Bots na *Robo Cup* 2015, ocorrida na China. Inicialmente, será feito um resumo sobre estimação de postura, seguido da apresentação do algoritmo em malha aberta utilizado como base e, finalmente, serão apresentadas as alterações realizadas para obter um algoritmo em malha fechada.

## II. ESTIMAÇÃO DE POSTURA

A estimação de postura é uma medida essencial para permitir que técnicas de caminhada em malha fechada percebam o meio e atuem de forma adequada. Ela consiste na obtenção dos ângulos de Euler – geralmente roll, pitch e yaw que descrevem a rotação do torso em relação ao sistema referencial inercial – e da consequente matriz de rotação que descrevem a orientação do robô em relação ao plano horizontal.

Há diversas maneiras de se aferir a postura de um robô; neste trabalho implementou-se uma técnica que utiliza os dados do acelerômetro e do giroscópio associadas através de um observador de ganho fixo [3]. Outra técnica bastante empregada para melhorar a estimativa é a utilização do filtro de Kalman para a associação das medidas [4]. Este por sua vez, pode produzir resultados ainda mais precisos se parametrizado através de lógica difusa [5].

Por fatores didáticos, a explicação da técnica é apresentada em três etapas: (1) aferição utilizando medidas do acelerômetro; (2) aferição através dos dados do giroscópio; e (3) associação com observador de ganho fixo.

### A. Aferição por aceleração linear

O método utiliza-se do fato de que a aceleração gravitacional é uma força que está sempre presente e paralela ao eixo z, razão pela qual não se pode estimar yaw, e o fato de um acelerômetro imóvel produzir saída com módulo de 1g. A metodologia consiste em encontrar os valores de roll ( $\varphi$ ) e pitch ( $\theta$ ) que tornem a saída do sensor perpendicular ao plano horizontal, considerando yaw igual a zero. Para isso, inverte-se a matriz de rotação resultante e a multiplica dos dois lados igualdade, resultando na Equação (1), onde  $a_m$  é a leitura do sensor.

$$a_m = \begin{bmatrix} g s(\theta) \\ -g c(\theta)s(\varphi) \\ -g c(\theta)c(\varphi) \end{bmatrix} \quad (1)$$

A partir deste resultado, é possível utilizar as Equações (2) e (3) para estimar rapidamente os valores de roll e pitch, respectivamente. Contudo, este resultado baseia-se na hipótese de um robô estático, ou seja, vibrações e o próprio movimento de caminhada interferem na precisão do método.

$$\hat{\varphi} = \arctan\left(\frac{a_{m,y}}{a_{m,z}}\right) \quad (2)$$

$$\hat{\theta} = \arctan\left(\frac{a_{m,x} \cdot \cos(\varphi)}{a_{m,z}}\right) \quad (3)$$

### B. Aferição por velocidade angular

Giroscópios fornecem a velocidade angular do sensor no seu sistema de coordenadas. Para estabelecer uma relação entre a derivada dos ângulos de Euler e a resposta do sensor, é preciso que cada componente da derivada seja rotacionada para o sistema de coordenadas do robô, conforme Equação (4). Após trabalho matemático é possível obter a Equação (5), o que permite calcular a derivada dos ângulos parametrizados a partir da resposta do sensor.

$$\omega = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + R_{x,-\varphi} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{x,-\varphi} R_{y,-\theta} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\varphi)\tan(\theta) & \cos(\varphi)\tan(\theta) \\ 0 & \cos(\varphi) & -\text{sen}(\varphi) \\ 0 & \text{sen}(\varphi)\sec(\theta) & \cos(\varphi)\sec(\theta) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (5)$$

Conhecendo-se a derivada de roll e pitch, é possível estimar os mesmos a partir dos métodos iterativos apresentados, respectivamente, nas Equações (7) e (8). Este método apresenta resultados satisfatórios para pequenas janelas de tempo, quando utiliza-se um baixo tempo de discretização  $T$ . No entanto, devido a ruídos do sensor, os valores dos ângulos podem divergir a longo prazo.

$$\hat{\varphi}(k+1) = \hat{\varphi} + \dot{\varphi} T \quad (7)$$

$$\hat{\theta}(k+1) = \hat{\theta} + \dot{\theta} T \quad (8)$$

### C. Observador de ganho fixo

Os métodos anteriores apresentam vantagens e desvantagens opostas; enquanto o primeiro possui resposta invariável no tempo, mas é sensível às forças externas, o segundo diverge devido a ruído, mas é confiável para medidas em curtas janelas de tempo. Visando as vantagens de ambas abordagens, uma técnica bastante utilizada devido à sua simplicidade é o uso de um observador de ganho fixo, ou filtro complementar. Baseando-se no pressuposto que as medidas aferidas pelo acelerômetro são próximas a realidade, o observador consiste na união de ambas medidas de forma complementar conforme exibido nas Equações (9) e (10). O valor do ganho projetado varia conforme a confiabilidade das medições: um ganho próximo de zero indica que os giroscópios não divergem muito rápido, enquanto um ganho próximo de um indica que a medida dos acelerômetros tende a ser mais confiável.

$$\hat{\varphi} = (1-L)\hat{\varphi}_{previsto} + L\hat{\varphi}_{acelerometro} \quad (9)$$

$$\hat{\theta} = (1-L)\hat{\theta}_{previsto} + L\hat{\theta}_{acelerometro} \quad (10)$$

## III. ALGORITMO EM MALHA ABERTA

O algoritmo utilizado como base deste trabalho utiliza um controle de malha aberta que possui como entrada um vetor velocidade desejado. Este algoritmo possui estrutura modular, o que facilita a sua adaptação a diferentes sistemas. A Fig. 1 **!Erro! Fonte de referência não encontrada.** mostra um diagrama de blocos que representa o algoritmo. A entrada do algoritmo é um vetor de velocidades  $\check{V}$  com duas componentes lineares (sagital e lateral), e uma componente angular (*yaw*). A saída, por sua vez, são os ângulos das juntas de cada perna de acordo com a fase do movimento.

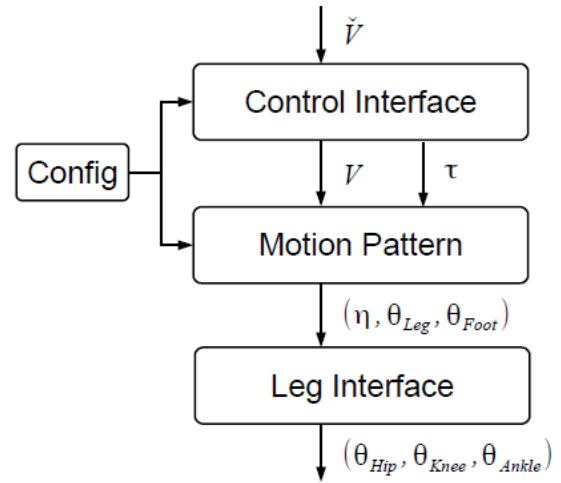


Fig. 1 Diagrama de blocos das interfaces. Fonte: [2]

O primeiro módulo tem a função inicial de adequar o vetor de entrada  $\check{V}$  de acordo com a configuração realizada. Primeiramente o vetor que contém as 3 velocidades é normalizado, utilizando norma p, de forma que o vetor possua valores reais entre -1 e 1. Devido ao fato de ser utilizado norma p, outra restrição garantida por este módulo é que não será passado valores desejados muito elevados em mais de uma direção ao mesmo tempo. Em um segundo momento, mudanças bruscas da entrada são suavizadas por uma aceleração limite, de forma que o robô não mude de comportamento subitamente e perca o equilíbrio. Além disso, esta interface controla a fase da caminhada. Esta variável possui valor entre  $-\pi$  e  $\pi$  e é passada como parâmetro para o próximo módulo.

Em um segundo bloco, as velocidades e a fase geradas anteriormente abastecem um gerador de padrão do movimento. Este gerador é a parte central do algoritmo e possui o código de movimento das pernas resultando em posições parametrizadas dos pés (Fig. 2). As 3 componentes de saída deste bloco são: (1) extensão parametrizada da perna, que é um valor entre 0 e 1 que representa o quanto a perna está estendida, sendo 0 totalmente estendida ( $\eta$ ); (2) ângulos *roll*, *pitch* e *yaw* entre tornozelo e linha do tronco ( $\theta_{Leg}$ ); e (3) ângulos *roll* e *pitch* entre o pé e o solo ( $\theta_{Foot}$ ).

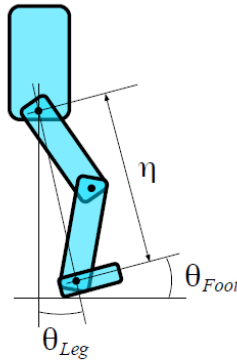


Fig. 2 Parametrização de perna robótica. Fonte: [2].

Finalmente, no bloco de interface os parâmetros gerados são convertidos para os ângulos dos motores. Este é o módulo que faz com que os demais cálculos não dependam da configuração das pernas. Em outras palavras, é possível utilizar os módulos anteriores, sem alteração, em um robô que possua pernas com diferentes configurações de juntas. Esta interface pode ser implementada utilizando equações simples com os ângulos parametrizado no módulo anterior [2]. Contudo, o método utilizado neste trabalho consiste na conversão dos parâmetros anteriores em posição e orientação desejadas para o pé em coordenadas cartesianas, conforme Equações (11), (12), (13), (14), (15), (16) e (17). Isso permite que esses valores sejam passados para o método implementado originalmente no robô para o cálculo da cinemática inversa. Vale destacar que o método adotado é bem menos eficiente que o originalmente proposto, no entanto os valores intermediários são mais facilmente interpretados e a adequação dos valores pode ser realizada de forma mais clara.

$$D = (1 - \eta) LegLength \quad (11)$$

$$F_x = D (c(\theta_{Leg}^{Yaw}) s(\theta_{Leg}^{Pitch}) c(\theta_{Leg}^{Roll}) + s(\theta_{Leg}^{Yaw}) s(\theta_{Leg}^{Roll})) \quad (12)$$

$$F_y = D (s(\theta_{Leg}^{Yaw}) s(\theta_{Leg}^{Pitch}) c(\theta_{Leg}^{Roll}) - c(\theta_{Leg}^{Yaw}) s(\theta_{Leg}^{Roll})) \quad (13)$$

$$F_z = D c(\theta_{Leg}^{Yaw}) c(\theta_{Leg}^{Roll}) \quad (14)$$

$$F_{Roll} = \theta_{Foot}^{Roll} \quad (15)$$

$$F_{Pitch} = \theta_{Foot}^{Pitch} \quad (16)$$

$$F_{Yaw} = \theta_{Leg}^{Yaw} \quad (17)$$

Para implementação deste algoritmo criou-se um módulo de movimento com base na classe original do robô, *MotionModule*. Este módulo ficou organizado em duas classes: *Omnivalk* e *MotionPattern*. A primeira consiste nos parâmetros de ajustes do algoritmo, controle da velocidade e fase, e controle das juntas das pernas. O segundo consiste somente no gerador de padrão da caminhada, o que representa como vantagem, a fácil adaptação do algoritmo a diferentes geradores de padrão.

#### IV. ALGORITMO EM MALHA FECHADA

Testando o algoritmo em malha aberta o robô manteve-se equilibrado quando em piso liso – chão do laboratório - no entanto sem resistir a pequenos impactos. Nestes testes percebeu-se que os pés do robô deslizavam muito durante a fase de apoio, o que, contra intuitivamente, auxiliava no equilíbrio. Ao testar este mesmo algoritmo na grama sintética percebeu-se que o fato do pé de apoio não ter o mesmo comportamento causava quedas. Além disso, frequentemente as quedas não ocorriam repentinamente e era possível perceber a tendência a queda devido a inclinação do robô. Devido a constante falta de equilíbrio dinâmico, confirmou-se o problema explicitado neste trabalho e a necessidade de implementar o algoritmo em malha fechada.

Com intuito de realizar a manutenção do equilíbrio de forma mais eficiente quando em piso não liso, foram realizadas duas adaptações: (1) controle de fase do movimento com base na percepção indireta do contato do pé de balanço com o solo e (2) mudança do parâmetro sagital de entrada do algoritmo.

Uma modificação que teve resultados bastante favoráveis na obtenção do equilíbrio foi a modificação da fase do movimento conforme a detecção do toque do pé no chão. A ideia implementada surgiu ao observar que, ao utilizar a caminhada em malha aberta na grama sintética, muitas vezes o robô iniciava a fase de balanço de uma perna sem que a outra tivesse tocado o chão, causando sua queda. Para corrigir este problema, colocou-se a condição de que para iniciar o movimento de balanço de uma perna, o pé oposto deveria estar em altura similar ou mais baixa. No entanto, sem um sensor no pé que permitisse a detecção de contato, precisou-se criar outra solução: fazendo uso da estimativa de postura e considerando que o plano é sempre horizontal, é possível realizar a rotação das coordenadas de posição dos pés para o plano e assim analisar suas alturas através de análise direta do eixo vertical. Esta modificação praticamente resolveu os problemas de queda lateral quando o robô é utilizado na grama sintética. Estas ainda ocorrem, mas são raras e geralmente ocorrem para o lado da perna de apoio, o que dificilmente poderia ser contornado por algoritmos mais sofisticados como o de Capture Step.

Apesar dos bons resultados com o acréscimo citado, quedas no plano sagital ainda eram frequentes. Contudo, observou-se que as quedas sagitais eram precedidas por uma inclinação no ângulo Pitch. Buscando uma solução de rápida implementação para a competição, modificou-se o método de entrada do algoritmo em malha aberta de forma que, ao invés de utilizar uma referência de velocidade, uma referência de ângulo Pitch é passada. Esta ideia baseia-se no entendimento de que, desde que o robô mantenha o equilíbrio, a inclinação no plano sagital define a direção deste movimento. A modificação proposta consiste na aplicação de um controle proporcional integral (PI) para o ângulo Pitch onde a variável de atuação é a velocidade sagital de entrada do gerador de padrão, conforme mostrado em Fig. 3.

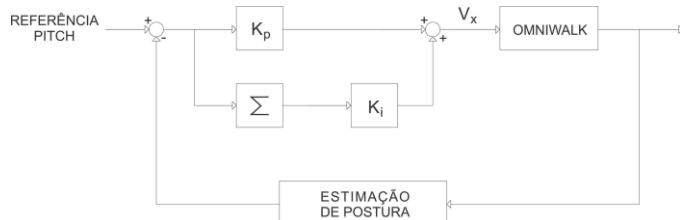


Fig. 3 - Controle de equilíbrio utilizando controlador PI e gerador de padrão

A utilização destas duas modificações resume-se na utilização da estimação de postura como forma de fechar a malha do gerador de padrão de caminhada de forma simples.

## V. CONCLUSÃO

Os resultados obtidos com esta aplicação resultaram em satisfatória manutenção do equilíbrio, no entanto a caminhada mostrou-se oscilatória no eixo sagital e conseqüentemente lenta do ponto de vista final.

Apesar das dificuldades encontradas ao longo do trabalho, a complexidade da área de caminhada e o fato de ser uma das primeiras implementações de algoritmos em malha fechada em robô *kidsize* tornam os resultados obtidos satisfatórios. O algoritmo final deste trabalho é facilmente modificado e totalmente adaptado ao código original da Robotis, o que poderá servir como base para trabalhos futuros. Acredita-se que algoritmos de caminhada de *capture point* sejam o próximo passo na busca por melhores resultados. No entanto, sua implementação representa um maior desafio em robôs do porte do DARwIn; crença esta reforçada pelo fato de não haver ainda trabalhos relatando implementações na categoria.

## REFERÊNCIAS

- [1] PATHAK, K., FRANCH, J., AGRAWAL S.K. **Velocity and Position Control of a Wheeled Inverted Pendulum by Partial Feedback Linearization**. IEEE TRANSACTIONS ON ROBOTICS, VOL. 21, NO. 3, JUNE 2005.
- [2] MISSURA, M.; BEHNKE S. **Self-stable omnidirectional walking with compliant joints**: 8<sup>th</sup> workshop on Humanoid Soccer Robots, Atlanta, Estados Unidos, 2013
- [3] CH ROBOTICS. **Fundamentals of attitude estimation**: Disponível em <<http://www.chrobotics.com/library/attitude-estimation>> Acesso em: 25 Jun. 2015.
- [4] KALMAN, R. E. **A new Approach to Linear Filtering and Prediction Problems**. Journal of Basic Engineering, 82 (Series D): p 35-45,1960.
- [5] KANG C. W.; PARK C. G. **Attitude estimation with accelerometers and gyros using fuzzy logic to tuned Kalman filter**: European Control Conference, Budapeste, Hungria, 2009