



Estratégia de Desvio de Obstáculos e Controle de Equilíbrio para Robôs Bípedes com Rodas

Gustavo Corrêa de Oliveira, Departamento de Engenharia Mecânica, EESC/USP, gc.oliveira@usp.br
Marcelo Becker, Departamento de Engenharia Mecânica, EESC/USP, becker@sc.usp.br

OBJETIVOS

Robôs bípedes com rodas formam uma classe híbrida de sistemas robóticos que combina a versatilidade da locomoção bípede com a eficiência e velocidade da locomoção por rodas. Essa configuração permite a movimentação em superfícies irregulares e, ao mesmo tempo, oferece a capacidade de superar obstáculos ou interagir com o ambiente em pé. Nos últimos anos, sensores, computadores embarcados e atuadores, aliado com estratégias de controle robustas têm possibilitado o desenvolvimento desses robôs com maior autonomia e desempenho nas tarefas. Um exemplo desse tipo de robô é o Ascento, um bípede que combina pernas com juntas e rodas motorizadas. Essas características permitem sua aplicação em tarefas como inspeção, vigilância e pesquisa em navegação autônoma (Klemm et al., 2019). Esse robô tem sido utilizado como plataforma de testes para técnicas de controle, aprendizado por reforço e estratégias de locomoção em terrenos não estruturados (Klemm et al., 2020; Chamorro et al., 2024). O presente trabalho se insere nesse contexto, propondo uma abordagem de controle para equilíbrio dinâmico e desvio de obstáculos estáticos de forma autônoma, utilizando o robô Ascento como plataforma em ambiente simulado computacionalmente.

METODOLOGIA

Neste trabalho foi obtido um modelo dinâmico simplificado do Ascento, com o intuito de manter o seu equilíbrio por meio de um controlador LQR (*Linear Quadratic Regulator*). A estratégia de desvio de obstáculos foi baseada em um sensor do tipo LiDAR (*Light Detection and Ranging*) para a detecção de objetos. O LQR foi desenvolvido no MATLAB, utilizando a função `lqr`, e validado inicialmente no Simulink. Após essa etapa, ele foi implementado na simulação com o modelo digital do robô Ascento, empregando ROS 2 Humble e o simulador Gazebo Fortress, no sistema operacional Linux Ubuntu 22.04.5 LTS em um notebook Dell Precision 3581 (Intel Core i9-13900H, 32 GB RAM, GPU Nvidia RTX 2000 Ada com 8 GB de memória dedicada).

O modelo dinâmico desenvolvido considerou o Ascento como um pêndulo invertido sobre duas rodas, conforme proposto por (Nader and Sarsri, 2023), mantendo fixas as juntas de joelhos e quadril. As equações dinâmicas, obtidas pelo método de Lagrange-Euler, foram linearizadas no ponto de equilíbrio ($\theta \approx 0$, $\dot{\theta} \approx 0$), considerando pequenos ângulos ($\sin \theta \approx \theta$, $\cos \theta \approx 1$) e simplificação de termos de acoplamento. Por fim, esse sistema foi representado no espaço de estados na Eq. 1, considerando as variáveis $x = [\dot{x}, \theta, \dot{\theta}, \varphi]^T$, que correspondem, respectivamente, à velocidade linear do robô, ao ângulo de *pitch*, à velocidade angular de *pitch* e à velocidade angular de *yaw*.

$$\dot{x} = Ax + Bu, \quad y = Cx \quad (1)$$

$$A = \begin{bmatrix} 0 & \frac{m_s d^2 g}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{m_s d g (3m_c + m_s)}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{(m_s d^2 + I_3)/R - m_s d}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} & -\frac{(m_s d^2 + I_3)/R - m_s d}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} \\ 0 & 0 \\ -\frac{(m_s d/R) - 3m_c m_s}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} & -\frac{(m_s d/R) - 3m_c m_s}{3m_c I_3 + 3m_c m_s d^2 + m_s I_3} \\ \frac{2L/R}{6m_c L^2 + m_c R^2 + 2I_2} & -\frac{2L/R}{6m_c L^2 + m_c R^2 + 2I_2} \end{bmatrix}, \quad C = I_{4 \times 4}$$

Na modelagem, foram considerados os seguintes parâmetros físicos do robô: massa do corpo m_s , massa de cada roda m_c , momento de inércia do corpo em relação ao eixo das rodas I_2 , momento de inércia em torno do eixo vertical I_3 , distância do centro de massa ao eixo das rodas d , raio das rodas R e distância entre a roda e o eixo de rotação vertical L . Os torques nas rodas esquerda e direita são u_L e u_R , respectivamente. O controlador LQR foi projetado minimizando uma função de custo ponderada pelas matrizes Q e R , na Eq. 2. O ganho ótimo K foi determinado a partir da solução da equação algébrica de Riccati, utilizando a função `lqr` do MATLAB. A lei de controle resultante é expressa por $u = -Kx$.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt, \quad Q = \text{diag}(5, 10, 10, 5), \quad R = I_{2 \times 2}, \quad K = \begin{bmatrix} 1.581 & 32.151 & 3.985 & 0.707 \\ 1.581 & 32.151 & 3.985 & -0.707 \end{bmatrix} \quad (2)$$

Para a implementação do desvio de obstáculos, foi adicionado um sensor LiDAR ao modelo digital do Ascento. Desse modo, obteve-se as distâncias entre o robô e os obstáculos ao seu redor. Com isso, foi desenvolvido em C++ um nó



responsável por publicar as velocidades $\dot{x}$ e $\dot{\varphi}$ necessárias para movimentar o robô de forma a desviar dos obstáculos à sua frente. Quando o robô atinge uma distância de 2 m em relação a um dos obstáculos, o controle de desvio assume automaticamente os comandos de velocidade ($\dot{x}$ e $\dot{\varphi}$), conduzindo o robô para contornar o obstáculo e tentar retomar a trajetória previamente seguida durante a *teleoperation* (comandos de velocidade via teclado).

RESULTADOS E CONCLUSÕES

A Fig. 1 (a) mostra o ângulo de *pitch* θ do robô, que atingiu inclinação máxima de 7° e manteve-se em pé durante toda a trajetória, demonstrando a eficácia do controlador LQR em manter o equilíbrio.

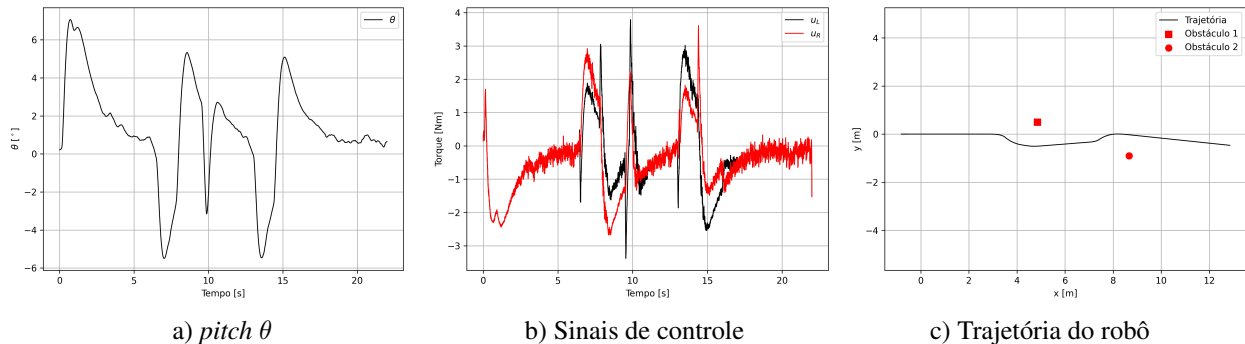


Figura 1. Resultados para a inclinação, os torques das rodas e a trajetória do robô

Os sinais de controle de cada roda são apresentados na Fig. 1 (b). A diferença de torque em certos momentos indica que o robô está realizando a rotação em seu próprio eixo para desviar dos obstáculos. Por fim, a trajetória realizada pelo Ascento para desviar dos obstáculos é apresentada na Fig. 1 (c). Os obstáculos são representados pelo quadrado e pelo círculo vermelhos. Por esse gráfico é possível notar que, após ultrapassar o segundo obstáculo, o Ascento não retorna à orientação original, desviando da trajetória prevista. Além disso, a posição inicial do robô é (0, 0), e o gráfico mostra que ele se desloca na direção negativa do eixo x , enquanto não recebe comandos de movimento para frente na direção positiva do eixo x .

A análise dos resultados indica que a malha com o controlador LQR para equilíbrio, associada ao algoritmo de desvio de obstáculos, cumpriu seu objetivo, mantendo o robô estável e em pé durante toda a execução da trajetória. Contudo, dois problemas foram observados: antes do início do comando por *teleoperation*, o robô desloca-se para trás, não permanecendo na posição quando parado; e, após ultrapassar os obstáculos, não retorna à trajetória que seguia antes do desvio. Essas questões podem ser investigadas e solucionadas em trabalhos futuros.

REFERÊNCIAS

- Chamorro, S., Klemm, V., Valls, M. I., Pal, C., and Siegwart, R. (2024). Reinforcement learning for blind stair climbing with legged and wheeled-legged robots.
- Klemm, V., Morra, A., Gulich, L., Mannhart, D., Rohr, D., Kamel, M., de Viragh, Y., and Siegwart, R. (2020). Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops. *IEEE Robotics and Automation Letters*, 5(2):3745–3752.
- Klemm, V., Morra, A., Salzmann, C., Tschopp, F., Bodie, K., Gulich, L., Kung, N., Mannhart, D., Pfister, C., Vierneisel, M., Weber, F., Deuber, R., and Siegwart, R. (2019). Ascento: A two-wheeled jumping robot. In *2019 International Conference on Robotics and Automation (ICRA)*, page 7515–7521. IEEE.
- Nader, K. and Sarsri, D. (2023). Modelling and control of a two-wheel inverted pendulum using fuzzy-pid-modified state feedback. *Journal of Robotics*, 2023:1–13.

AGRADECIMENTOS

Este trabalho foi financiado pela Petróleo Brasileiro S/A – Petrobras (nº 0050.0124520.23.9), Fundação de Apoio à Física e à Química (FAFQ) e Universidade de São Paulo (USP).

RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.