

Um Modelo para Otimização de Caminhada em Robôs Bípedes usando Pêndulo Invertido e Aprendizado por Reforço

Wesley S. Silva , Josemar Rodrigues de Souza , Ivanoé J. Rodowanski, Marco A. C. Simões

¹Centro de Pesquisa em Arquitetura de Computadores, Sistemas Inteligentes e Robótica (ACSO)
Universidade do Estado da Bahia (UNEB) – Salvador – BA – Brazil

teambahiar@gmail.com

Abstract. *This work focuses on developing an inverted pendulum prototype with reinforcement learning and a complete training environment using the BahiaRT-GYM platform. In this environment, the agent's trunk inclination during walking was used as a practical case to allow adjustments for achieving stable and fluid locomotion. The inclination was a useful example to demonstrate the environment's capability to support and optimize practical training. The results show a 26% performance increase with a 27.8% faster speedup of the reinforcement learning-trained model over the inverted pendulum approach. Both outperform the original cart table-based model.*

Resumo. *Este trabalho foca no desenvolvimento de um protótipo de Pêndulo Invertido (PI) com aprendizagem por reforço, acompanhado de um ambiente completo de treinamento utilizando a plataforma BahiaRT-GYM. Com o ambiente, utilizou-se a questão da inclinação do tronco do agente durante a caminhada, permitindo que seja realizado um treino com ajustes que proporcionem uma locomoção estável e fluida. A inclinação foi empregada como um caso prático para demonstrar a capacidade do ambiente de suportar e otimizar treinamentos eficazes. Os resultados mostram um aumento de desempenho de 26% com uma velocidade 27,8% maior do modelo treinado por aprendizagem por reforço em relação ao PI. ambos superam o modelo original baseado no cart table.*

1. Introdução

A simulação 3D de robôs desempenha um papel crucial no desenvolvimento de modelos de caminhada realistas, aproximando-se do comportamento de robôs físicos. Dentro desse contexto, a RoboCup Soccer Simulation 3D, uma competição onde times de robôs autônomos disputam partidas de futebol em um ambiente virtual, oferece um desafio estimulante para o desenvolvimento de algoritmos de controle de locomoção [Kitano et al. 1998]. Otimizar a caminhada de robôs bípedes é essencial para melhorar a sua agilidade, eficiência energética e estabilidade em diferentes terrenos e situações.

Abordagens tradicionais, como o modelo Cart Table [Shafii 2015], apresentam limitações, como a trajetória vertical fixa do centro de massa (CoM), que restringe a fluidez dos movimentos. Os movimentos menos fluidos devem-se à necessidade do robô flexionar os joelhos para manter a altura do CoM fixa. Para superar essas limitações, este trabalho apresenta um modelo que utiliza o Pêndulo Invertido (PI), uma abordagem que modela a caminhada humana de forma mais realista, considerando a variação da altura do

CoM [Shafii 2015]. A fim de refinar ainda mais o controle da caminhada, empregamos técnicas de aprendizado por reforço, que permitem ao robô aprender a ajustar seus movimentos de forma autônoma, buscando maximizar uma função de recompensa que leva em conta a estabilidade e a eficiência da locomoção.

O aprendizado por reforço é uma técnica de aprendizado de máquina que permite a um agente aprender a interagir com um ambiente, recebendo recompensas por ações que levam a resultados desejáveis. O agente explora estratégias para maximizar suas recompensas ao longo do tempo, equilibrando exploração e levantamento de conhecimento adquirido [Sutton and Barto 2018]. Um exemplo do uso de aprendizagem por reforço é apresentado em [Liu et al. 2021], onde foi implementado um controle de equilíbrio para um agente parado, sujeito a forças externas. O agente aprendeu, a ajustar as melhores variáveis em conjunto com o Zero Moment Point (ZMP), garantindo que o objetivo de manter o equilíbrio fosse alcançado.

O robô NAO da Softbank Robotics ¹ foi escolhido como plataforma para este trabalho, devido à sua ampla utilização na RoboCup 3D e à disponibilidade de modelos simulados.

O presente trabalho, esta inserido no contexto da RoboCup Soccer Simulation 3D com o time Bahia Robotics Team (BahiaRT), uma iniciativa científica do Centro de Pesquisa em Arquitetura de Computadores, Sistemas Inteligentes e Robótica (ACSO) na Universidade do Estado da Bahia (UNEB), que participa regularmente das competições da RoboCup desde 2007, visa desenvolver um modelo para otimização da caminhada de robôs bípedes em um ambiente tridimensional.

Este texto está organizado da seguinte maneira: na Seção 2, apresentamos a anatomia do robô NAO e a importância de suas juntas para o controle da caminhada. A Seção 3 descreve o modelo de PI e como ele é utilizado para melhorar a eficiência e estabilidade da locomoção do robô. Na Seção 4, detalhamos a aplicação do algoritmo de Proximal Policy Optimization (PPO) para otimizar a caminhada por meio de aprendizado por reforço. Em seguida, são apresentados os resultados obtidos nos experimentos. Por fim, na última seção, são destacadas as conclusões e perspectivas futuras desta pesquisa.

2. Anatomia do Robô NAO

O entendimento da anatomia do robô NAO, especificamente de suas juntas e atuadores, é fundamental para o modelo de Pêndulo Invertido (PI) e os métodos de aprendizado por reforço utilizados neste trabalho. A quantidade e o tipo de juntas influenciam diretamente a liberdade de movimentos do robô e, consequentemente, a sua capacidade de realizar uma caminhada estável e eficiente. Para controlar as juntas, o controlador do robô envia para o simulador uma mensagem no formato (*nome, velocidade*), especificando a velocidade a ser aplicada a cada junta.

O NAO possui 22 juntas, classificadas em quatro categorias de acordo com seu eixo de rotação em relação ao tronco (Figura 1 e Tabela 1): *Yaw* (rotação em torno do eixo Z), *Roll* (rotação em torno do eixo Y), *Pitch* (rotação em torno do eixo X) e *Yaw-Pitch* (rotação em torno dos eixos X e Z) [Joschka and Asada 2008]. A compreensão desses movimentos é crucial para a modelagem da cinemática do robô, que é utilizada

¹<https://us.softbankrobotics.com/nao>

tanto no cálculo da trajetória do PI quanto na definição do espaço de ações do agente de aprendizado por reforço. Por exemplo, as juntas do quadril, que permitem movimentos de *pitch*, são essenciais para o controle da inclinação do tronco, um dos principais focos deste trabalho. O conhecimento preciso da amplitude e das restrições de movimento de cada junta permite definir limites realistas para o espaço de ações do agente, contribuindo para a eficácia do treinamento.

Tabela 1. Tipos de juntas e suas funções

Tipo	Descrição
Yaw (Verde)	Rotaciona em torno do eixo Z
Roll (Laranja)	Rotaciona em torno do eixo Y
Pitch (Amarelo)	Rotaciona em torno do eixo X
YawPitch (Azul)	Rotaciona em torno dos eixos X e Z

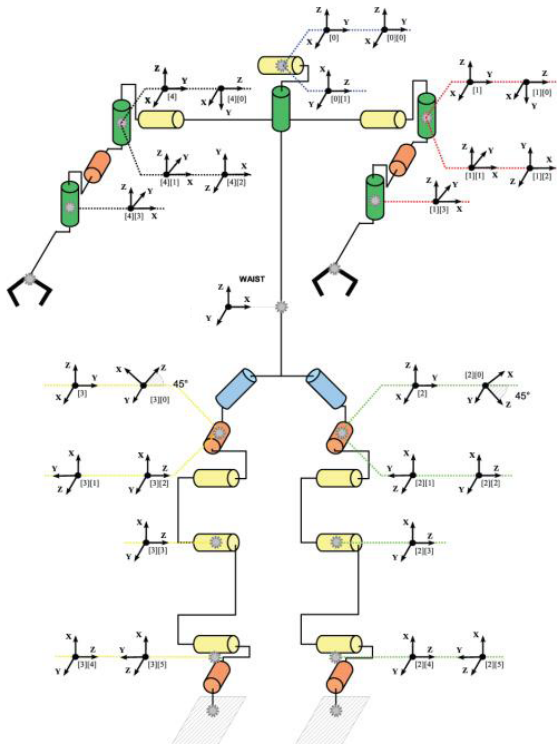


Figura 1. Esquema de Juntas do Robô NAO. As cores indicam os tipos de juntas, conforme a Tabela 1 [Joschka and Asada 2008].

3. Modelo do Pendulo Invertido

Um modelo de caminhada estabelece uma fórmula para otimizar a locomoção do robô, sendo desenvolvido em dois planos: o coronal (YZ) e o sagital (XZ). Essa abordagem é aplicada tanto no modelo do Cart Table quanto no PI, garantindo uma orientação consistente para a locomoção eficiente do robô [Shafii 2015].

Para que o agente tenha uma caminhada estável é preciso que os passos que ele vai executar sejam planejados, para que o modelo de caminhada possa trabalhar em cima

dessa trajetória. Foram desenvolvidos vários modelos de controle da caminhada, um dos mais populares é o ZMP que demonstrou ter uma grande eficácia como, por exemplo, a construção de um método utilizando algoritmos genéticos e ZMP, construído por Yilmaz em [Yilmaz et al. 2020].

Para construir o PI foi utilizado como base o código do BahiaRT com sua implementação do Cart Table, já que vários elementos poderiam ser aproveitados pelo protótipo construído, como o planejamento dos passos e o cálculo do ZMP. Além disso o cálculo do CoM vertical de altura variável e CoM horizontal também foram vitais para o funcionamento do modelo [Shafii 2015].

O cálculo da trajetória do CoM vertical é modelado como um movimento periódico, por isso a trajetória vertical será representada por uma função de base de Fourier. Como a equação de altura de quadril é par ou seja ela permanece inalterada sob reflexão no eixo y , foram utilizados apenas os termos de cosseno no gerador da trajetória. O cálculo desse gerador é dado pela equação (1), onde o parâmetro L é igual ao período do passo e os parâmetros β_0 , β_1 e β_2 vão ponderar a influência do período de passo L na geração da trajetória. s [Shafii 2015]. Os valores ótimos obtidos estão apresentados na tabela 2.

$$F(t) = \beta_0 + \beta_1 \cos(2t\pi/L) + \beta_2 \cos(4t\pi/L) \quad (1)$$

Tabela 2. Valores Treinados por Aprendizagem por Reforço

Sigla	Valores
β_0	0,187
β_1	-0,0025
β_2	0,0008

O valor do CoM Vertical impacta diretamente nos valores do CoM Horizontal. Foi utilizada uma aproximação numérica e o PI para gerar um CoM horizontal de uma corrida baseada em ZMP [Wang et al. 2018]. Nessa aproximação numérica, para gerar uma CoM horizontal, primeiro, a posição e aceleração do CoM é discretizada em pequenas fatias de tempo Δt

$$x(i\Delta t) - > x(i) \quad (2)$$

Depois temos um sistema tridiagonal para as equações diferenciais, do PI, escrito como

$$Px = A_i * x(i-1) + B_i * x(i) + C_i * x(i+1) \quad (3)$$

Onde temos que A_i , B_i e C_i têm suas próprias funções que representam seus valores. Perceba que A_i e C_i têm a mesma equação.

$$A_i = C_i = -1/\Delta t^2 * (z(i\Delta t)/g + z(i\Delta t)) \quad (4)$$

$$Bi = 1 - 2Ai \quad (5)$$

Para gerar a trajetória do CoM, o sistema linear é obtido utilizando o sistema tridiagonal da Figura 2.

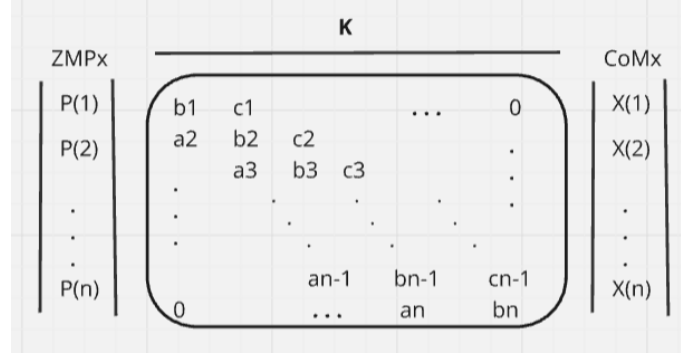


Figura 2. Sistema Tridiagonal utilizado para geração da trajetória do CoM Horizontal[Shafii 2015]

Para resolver este sistema tridiagonal, o algoritmo de Thomas [JUSTO et al. 2020] foi utilizado. A solução pode ser obtida em $O(n)$ operações, onde $n = Ts/DeltaT$, Ts é o tempo total que o CoM é calculado.

Com base no resultado desse cálculo, foi possível gerar uma trajetória de CoM, assim encaixando todas as peças da caminhada de forma integrada. Primeiramente, a posição do pé de apoio é determinada por meio do planejamento do passo. Posteriormente, a trajetória do ZMP é planejada dentro dos limites do polígono de apoio. O CoM vertical e horizontal são calculados com base no modelo de PI. Por fim, os ângulos das juntas das pernas são calculados usando a cinemática inversa, método utilizado pelo BahiaRT para mandar a velocidade para as juntas, com base na trajetória do pé de balanço, posição horizontal e vertical do CoM, e posição do pé de apoio [KOFINAS 2012].

4. Otimização com Aprendizagem por Reforço

Para o treinamento do protótipo, utilizamos o ambiente BahiaRT-Gym na versão v1.0.6 com Python na versão 3.8 e o Stable Baselines 3 [Simões et al. 2022], uma plataforma personalizada baseada no framework OpenAI-Gym [Schulman et al. 2017], que permite a implementação de diversos algoritmos de aprendizado por reforço. O treinamento foi executado no supercomputador OGUN-HPC, hospedado e operado nas instalações do Centro de Supercomputação para Inovação Industrial do SENAI CIMATEC. Escolhemos o algoritmo Proximal Policy Optimization (PPO) [Schulman et al. 2017] por sua eficiência em lidar com espaços de ação contínuos, como o necessário para o controle das juntas do robô. A implementação do PPO foi realizada utilizando a biblioteca Stable Baselines3, com os seguintes hiperparâmetros: taxa de aprendizado de 0.00025, tamanho do batch de 64 e 10 épocas por iteração, um total time steps de 20000, gamma de 0.99, GAE lambda 0.95, max-grad-norm com 0.5 e um n-steps de 500. Inicialmente, utilizamos os valores padrão sugeridos pela biblioteca para essa arquitetura. Posteriormente, realizamos ajustes empíricos com base na observação do desempenho do agente durante os testes,

buscando melhorar a qualidade do aprendizado. Diversos trabalhos recentes demonstram o sucesso do PPO para o aprendizado e otimização de movimentos em robôs humanoides [Kasaei et al. 2021, Abreu et al. 2023].

Para validar o ambiente de treinamento, focamos em aprimorar o mecanismo de compensação da inclinação do robô durante a caminhada. O objetivo era solucionar o problema de quedas causadas por ajustes abruptos nas juntas do quadril. O novo ambiente de aprendizagem aplica pequenos incrementos nos ângulos das juntas de forma dinâmica, permitindo que o robô aprenda a corrigir sua postura por meio de feedback contínuo da função de recompensa.

O espaço de observação do agente inclui informações relevantes para o controle da sua postura e a navegação até o alvo (Tabela 3). A inclinação do quadril é o ângulo de *pitch* do tronco do robô em relação à linha vertical, calculado com base na orientação de suas juntas e utilizado para corrigir o equilíbrio, o valor é medido em graus entre -20 e 20 (inclinação máxima para frente e para trás). O tipo de agente refere-se a um dos cinco modelos de robô oferecido pelo simulador, que influencia a dinâmica da caminhada. A ação anterior informa ao agente a sua última ação, permitindo um controle mais suave dos movimentos. Agente caído é uma variável booleana que indica se o robô caiu. Quantidade de *SelfCollisions* representa o número de colisões entre as partes do corpo do robô, penalizando movimentos inadequados. O tempo do episódio é o tempo decorrido desde o início do episódio em segundos (considerando o tempo máximo de 50 segundos). A posição relativa do alvo indica o ângulo da direção do alvo em relação ao robô, em radianos, normalizada entre $-\pi$ e π . A distância do alvo é a distância euclidiana entre o robô e o alvo, em metros.

Tabela 3. Espaço de Observação

ORIGEM	DADOS
Agente	Inclinação do Quadril do Agente (Durante a caminhada)
Agente	Tipo do Agente
Ambiente Gym	Action Anterior
Ambiente Gym	Agente Caído (True ou False)
Ambiente Gym	Quantidade de SelfCollisions
Valores Calculados	Tempo do Episódio
Valores Calculados	Posição relativa do Alvo (A frente do Alvo)
Valores Calculados	Distância do Alvo

O espaço de ação consiste em ajustes nos ângulos das juntas do quadril, aplicados a cada passo. Os valores de ajuste são contínuos, limitados entre -20 e 20 graus, representando a variação no ângulo de pitch do quadril.

No início do episódio, o agente é posicionado a 5 metros do alvo e quando o agente chega próximo da bola, a uma distancia de 0, 5 metros o episodio acaba. Ou seja o objetivo principal é chegar até a bola com a melhor recompensa possível.

A função de recompensa R (Equação (6)) foi projetada para incentivar o robô a caminhar de forma estável e eficiente até o alvo. O primeiro termo da equação recompensa a proximidade do alvo, sendo inversamente proporcional à distância e ao tempo decorrido.

A constante 3 ajusta o peso desse termo na recompensa total. O segundo termo penaliza quedas e colisões, sendo inversamente proporcional ao tempo do episódio. A constante 4 ajusta o peso desse termo. O terceiro termo penaliza inclinações excessivas do tronco, sendo proporcional à magnitude da inclinação e à ocorrência de quedas. A descrição de cada variável que compõe a função de recompensa pode ser vista na tabela 4.

$$R = \frac{PesoDeProximidade \times 3}{DistanceToTarget \times TempEpisodio} - \frac{ContQ + ContSF}{TempEpisodio \times 4} - somaPesoPitchQuedas \quad (6)$$

Tabela 4. Elementos da Função de Recompensa

SIGLAS	DESCRIÇÃO
TempEpisodio	Tempo do episódio, valor máximo de 50 segundos por episódio
PesoDeProximidade	Aumenta conforme o agente se aproxima do alvo
somaPesoPitchQuedas	Penalidade baseada na inclinação do tronco e quedas
DistanceToTarget	Distância do agente ao alvo
ContQ	Número de quedas desde o início do episódio
ContSF	Penalidades por SelfCollision no episódio

5. Resultados

Para avaliar o desempenho do modelo de PI, realizamos dois experimentos. No primeiro, comparamos o PI sem treinamento com o modelo Cart Table em um ambiente de simulação simplificado. O ambiente consiste em um campo plano de 20m x 30m, sem obstáculos. A bola é reposicionada aleatoriamente dentro do campo a cada vez que o agente a alcança. A bola nem sempre estará dentro campo de visão do agente. Quando o agente não vê a bola ele receberá esta informação por audição enviada por um agente posicionado na extremidade do campo no centro do gol defensivo, onde tem a visão do campo inteiro. Este ambiente simplificado foi escolhido para isolar o impacto do modelo de caminhada na desempenho do agente, eliminando a influência de outros fatores, como navegação e estratégia. Entretanto, o reposicionamento aleatório da bola exige que o agente execute trajetórias não-retilíneas para ajustar sua orientação em relação ao alvo. O agente executa 20.000 passos, divididos em episódios de 90 segundos. Um passo corresponde a um ciclo de interação entre o agente e o ambiente, no qual o agente executa uma ação, recebendo em seguida uma observação e uma recompensa. Em cada episódio, coletamos a velocidade média do agente, o número de vezes que ele alcança a bola e o número de quedas. O score de desempenho é calculado conforme a Equação 7. A velocidade média e o número de vezes que o agente alcançou a bola são somados e divididos pelo número de quedas para encontrar o score final do episódio.

$$Score = (velocidadeMedia + contaChegadaBola) / contQuedas \quad (7)$$

Os resultados desse primeiro experimento podem ser observados na tabela 5, é importante destacar que os agentes durante a obtenção foram configurados a rodar utilizando sua velocidade máxima. Os resultados apontam que o PI se comporta melhor que o *cart table* neste cenário de testes.

Tabela 5. Comparação entre PI (sem treinamento) e Cart Table

Variáveis	PI(Sem treino)	Cart Table
Score final	7.35	0.71
Velocidade média	1.76 m/s	1.46 m/s
Total de vezes que alcançou a bola	117	63
Total de quedas	24	125

O segundo teste do primeiro experimento utilizou o MagmaChallenge Benchmark [magmaOffenburg 2023] selecionando o desafio *Run Challenge*. Neste desafio, os robôs precisam correr e linha reta até o alvo. O *benchmarmk* calcula um score baseado na velocidade média e no total tempo em que o robô manteve os dois pés simultaneamente em fase de voo (sem tocar o chão)².

Tabela 6. Resultados MagmaChallenge Cart Table e PI (Sem treino)

Sigla	Cart Table	PI(Sem treino)
Score	0.733	0.745
Velocidade	0.57	0.593
Tempo com os dois pés em fase de voo	0.153	0.153

Os resultados do MagmaChallenge (Tabela 6) mostram que o PI sem treinamento obteve um score ligeiramente superior ao Cart Table (0.745 vs. 0.733). A pequena diferença no score é atribuída principalmente à maior velocidade do PI (0.593 vs. 0.57). O fato de o tempo com os dois pés em fase de voo ser o mesmo para ambos os modelos (0.153) sugere que o PI apesar da maior velocidade, não compromete a estabilidade da caminhada. No entanto, é importante notar que o MagmaChallenge não considera as quedas, o que pode mascarar diferenças na robustez dos modelos.

No segundo experimento, treinamos o modelo PI utilizando o algoritmo PPO, conforme descrito na Seção 4. A Figura 3 mostra a evolução da recompensa média durante o treinamento. O gráfico apresenta uma tendência de crescimento, indicando que o agente está aprendendo a otimizar a sua caminhada. No entanto, a convergência ainda não foi alcançada, sugerindo que um treinamento mais longo poderia resultar em um desempenho ainda melhor. Os picos e vales no gráfico podem ser atribuídos à estocasticidade do ambiente e do algoritmo PPO.

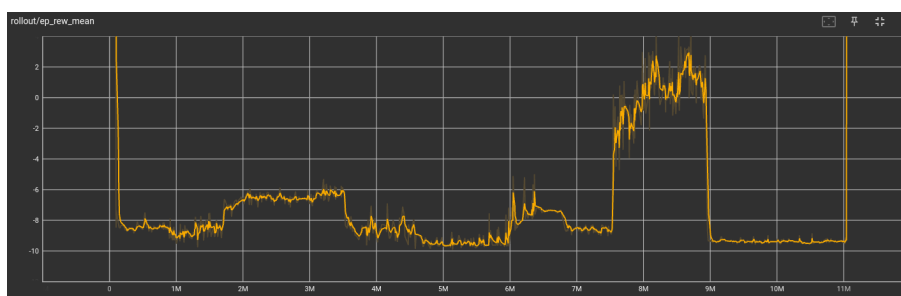


Figura 3. Gráfico da média da recompensa durante o treinamento.

²<https://github.com/magmaOffenburg/magmaChallenge>

Foi observado que o agente está aprendendo com base na função de recompensa, evidenciando que o ambiente desenvolvido é funcional e capaz de permitir que o agente aprenda e gere resultados. Os resultados da Tabela 7 mostram que o PI treinado atingiu uma velocidade média significativamente maior (2.25 m/s) em comparação com o PI sem treinamento (1.76 m/s) e o Cart Table (1.46 m/s). Apesar do ligeiro aumento no número de quedas, o PI treinado alcançou a bola com mais frequência (141 vs. 117), resultando em um score final superior (9.27 vs. 7.35). Este resultado demonstra a eficácia do modelo proposto para otimizar a caminhada de robôs bípedes.

Tabela 7. Comparação entre PI (Com e Sem treinamento) e Cart Table

Variáveis	PI(Sem treino)	Cart Table	PI(Treinado)
Score final	7.35	0.71	9.27
Velocidade média	1.76 m/s	1.46 m/s	2.25 m/s
Total de chegadas na bola	117	63	141
Total de quedas	24	125	25

6. Conclusão

Neste trabalho, foi desenvolvido um modelo de PI que posteriormente foi otimizado com aprendizagem por reforço, implementado na plataforma BahiaRT-GYM para treinamento. Por meio de um ambiente de treinamento projetado, exploramos a inclinação do tronco do agente durante a locomoção. Além disso, um ambiente foi criado para a coleta de métricas dos modelos de caminhada.

Os resultados obtidos demonstraram uma melhoria no desempenho do agente com o modelo treinado em um cenário onde a velocidade máxima é utilizada. Embora o agente tenha registrado uma queda a mais em comparação ao modelo não treinado, sua maior velocidade resultou em um maior número de chegadas à bola, culminando em um score final superior. Isso ressalta a eficácia do treinamento e a capacidade do ambiente de suportar e otimizar o aprendizado. No entanto, é importante ressaltar que o treinamento foi realizado em um ambiente de simulação simplificado. O desempenho do modelo em um ambiente mais complexo, com obstáculos e interação com outros agentes, ainda precisa ser avaliada. Trabalhos futuros devem investigar a generalização do modelo para diferentes cenários, além de explorar outras arquiteturas de redes neurais e funções de recompensa.

Agradecimentos

Os autores agradecem ao Centro de Supercomputação para Inovação Industrial do SENAI CIMATEC, pela cooperação, fornecimento e operação das instalações computacionais.

Referências

- Abreu, M., Reis, L. P., and Lau, N. (2023). Designing a Skilled Soccer Team for RoboCup: Exploring Skill-Set-Primitives through Reinforcement Learning. Codebase release at <https://github.com/m-abr/FCPCodebase>.
- Joschka, B. and Asada, M. (2008). Simspark—concepts and application in the robocup 3d soccer simulation league. *Autonomous Robots*, 174:181.

- JUSTO, D., SAUTER, E., AZEVEDO, F., GUIDI, L., and KONZEN, P. (2020). *Cálculo Numérico: um livro colaborativo—versão Scilab*. UFRGS.
- Kasaei, M., Abreu, M., Lau, N., Pereira, A., and Reis, L. P. (2021). Robust biped locomotion using deep reinforcement learning on top of an analytical control approach. *Robotics and Autonomous Systems*, 146:103900.
- Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawai, E., and Matsubara, H. (1998). Robocup: A challenge problem for ai and robotics. *RoboCup-97: Robot soccer world cup I*, H. Kitano, Org.
- KOFINAS, N. (2012). *Forward and inverse kinematics for the NAO humanoid robot*. PhD thesis, Technical University of Crete.
- Liu, C., Gao, J., Tian, D., Zhang, X., Liu, H., and Meng, L. (2021). A disturbance rejection control method based on deep reinforcement learning for a biped robot. *Applied Sciences*.
- magmaOffenburg (2023). Magma challenge documentation.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms.
- Shafii, N. (2015). *Development of an optimized omnidirectional walk engine for humanoid robots*. PhD thesis, Universidade do Porto (Portugal).
- Simões, M. A., Mascarenhas, G., Fonseca, R., dos Santos, V. M., Mascarenhas, F., and Nogueira, T. (2022). Bahiart setplays collecting toolkit and bahiart gym. *Software Impacts*, 14:100401.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, Massachusetts, second edition edition.
- Wang, M., Wang, R., Zhao, J., and Sun, P. (2018). An optimized algorithm based on energy efficiency for gait planning of humanoid robots. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 5612–5617. IEEE.
- Yilmaz, S., Gokasan, M., and Bogosyan, S. (2020). Oscillation preventing closed-loop controllers via genetic algorithm for biped walking on flat and inclined surfaces. *International Journal Of Advanced Computer Science And Applications*, 11(5).