15º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2024

Robô Seguidor de Linha com Aprendizagem por Reforço

Nathaly dos Santos Mendes[1], Victor Cesar de Mecê Prando[2], Ricardo Pires[3]

[1]Graduanda em Engenharia Mecânica, campus São Paulo, IFSP, bolsista do CNPq - nathaly.santos@aluno.ifsp.edu.br
[2]Graduando em Engenharia Eletrônica, campus São Paulo, IFSP - m.prando@aluno.ifsp.edu.br
[3]Docente no campus São Paulo, IFSP - ricardo_pires@ifsp.edu.br
Rua Pedro Vicente, 625 - CEP 01109-010 - São Paulo (SP) - Brasil

Área de conhecimento (Tabela CNPq): 3.05.05.04-6 Robotização

RESUMO: A Inteligência Artificial está em rápido desenvolvimento, bem como a Robótica e suas aplicações. A Aprendizagem por Reforço é uma abordagem computacional de aprendizagem, na qual um agente busca receber um máximo de recompensas enquanto interage com um ambiente complexo e incerto. O Q-learning é uma implementação dessa abordagem. Nele, um sistema aprende, com a experiência, qual é a melhor ação a ser realizada a partir de cada estado do sistema. O objetivo deste trabalho foi o projeto e a obtenção do protótipo de um robô seguidor de linha o qual, nessa tarefa, executasse o algoritmo Q-learning. Verificou-se experimentalmente a viabilidade de implementá-lo em um robô que foi montado com material de uso comum em ambientes didáticos na área de Robótica e afins, como uma placa Arduino e sensores de infravermelho. Ao ser colocado numa pista apropriada, o robô aprendeu a seguir uma linha nela demarcada, a partir da experiência que ele próprio adquiriu experimentando ações em variados estados. O envolvimento de estudantes com essas técnicas, usando exemplos básicos como o deste trabalho, serve como uma etapa inicial no aprofundamento de seus conhecimentos nessas áreas com aplicações em expansão.

PALAVRAS-CHAVE: Inteligência Artificial; Aprendizagem por Reforço; Robótica; seguidor de linha; Q-learning

Line Follower Robot with Reinforcement Learning

ABSTRACT: Artificial Intelligence is rapidly developing, as well as Robotics and its applications. Reinforcement Learning is a computational approach to learning in which an agent seeks to receive a maximum of rewards while interacting with a complex and uncertain environment. Q-learning is an implementation of this approach. In it, a system learns, through experience, what is the best action to take based on each state of the system. The objective of this work was to design and obtain the prototype of a line-following robot which, in this task, would execute the Q-learning algorithm. The feasibility of implementing it in a robot was experimentally verified, which was assembled with material commonly used in teaching environments in the area of Robotics and the like, such as an Arduino board and infrared sensors. When placed on an appropriate track, the robot learned to follow a line marked on it, based on the experience it acquired by experimenting with actions in different states. Involving

students with these techniques, using basic examples like the one in this work, serves as an initial step in deepening their knowledge in these areas with expanding applications.

KEYWORDS: Artificial intelligence; Reinforcement Learning; Robotics; line follower; Q-learning

INTRODUÇÃO

Segundo Smith (2024), a Robótica consiste no projeto, na construção e no uso de máquinas chamadas robôs, para realizar tarefas tradicionalmente realizadas por humanos. O termo Inteligência Artificial (IA) pode se referir à habilidade de um sistema em interpretar dados externos e, a partir deles, aprender a atingir certos objetivos, ou à ideia de que um tal sistema é aquele que busca imitar a inteligência humana (MIKALEF; GUPTA, 2021). Tem havido uma convergência da Robótica com a IA (HELFRICH, 2022).

Um ramo da IA é a Aprendizagem por Reforço, definida como sendo uma abordagem computacional, na qual um agente busca receber um máximo de recompensas enquanto interage com um ambiente complexo e incerto (SUTTON; BARTO, 2018). Ela é útil quando não se tem um modelo acurado do agente ou do ambiente ou quando os seus parâmetros variem com o tempo, já que ela permite que o agente busque atuar de forma ótima, usando o que aprende enquanto atua.

Um algoritmo de Aprendizagem por Reforço é o Q-Learning (WATKINS, 1989). Ele pressupõe que o agente percorre uma trajetória de estados discretos. A partir de cada estado, deve haver um número finito de ações possíveis. Caberá ao agente aprender, com a experiência, qual é a melhor ação a ser realizada a partir de cada estado. A cada par estado s e ação a, é associado um número, $Q(s, a)$, aprendido com a experiência, o qual é uma medida da qualidade de se executar a ação a no estado s. A tabela $Q(s, a)$ é atualizada e consultada ao longo da atuação do agente.

O objetivo deste trabalho foi o projeto e a obtenção do protótipo de um robô seguidor de linha o qual, nessa tarefa, executasse o algoritmo Q-learning. Desejou-se avaliar a viabilidade de implementá-lo em um robô que só usasse material de uso comum em ambientes didáticos.

FUNDAMENTAÇÃO TEÓRICA

O Q-Learning pressupõe que o agente percorre uma trajetória de estados discretos. A partir de cada estado, deve haver um número finito de ações possíveis. Caberá ao agente aprender, com a experiência, qual é a melhor ação a ser realizada a partir de cada estado. A cada par estado e ação (s, a) é associado um número, $Q(s, a)$, o qual é uma medida da qualidade da decisão de se executar a ação a no estado s. Na execução do algoritmo, valores para $Q(s, a)$ para todos os possíveis pares (s, a) podem ser iniciados arbitrariamente. Enquanto o agente explora estados e ações, os valores para $Q(s, a)$ são ajustados por meio de recompensas. Valores positivos para a recompensa são atribuídos a pares (s, a) que levam o agente a melhores estados e valores negativos atribuídos a pares (s, a) que o levam a piores estados. No caso de um robô seguidor de linha, uma recompensa positiva pode ser atribuída a um par (s, a) que faça o robô avançar mantendo-se bem centralizado sobre a linha. Uma recompensa negativa deve ser atribuída a um evento de saída da linha. Então, o agente é colocado em funcionamento, para aprender com a experiência. A partir de cada estado, ele pode experimentar ações, medir as consequências delas e ajustar os valores de $Q(s, a)$. Quanto mais o agente experimentar os possíveis estados e ações, mais experiência ele ganhará no desempenho de sua função. Enquanto aprende, o agente pode, a cada estado, favorecer a adoção de ações que já estejam com bons valores

de $Q(s, a)$ ajustados em episódios anteriores. Mas, é aconselhável que ele continue experimentando, com uma certa probabilidade, ações provisoriamente consideradas não ótimas, para garantir que ele faça uma boa exploração das possibilidades e que ele se ajuste a um ambiente em possível mudança. Deve ser adotada, portanto, uma política de escolha da ação em função dos valores atuais de $Q(s, a)$. Cada vez que um par (s, a) é experimentado, o valor de $Q(s, a)$ é atualizado. O novo valor é calculado como uma combinação do valor anterior, do valor da recompensa atual e do máximo valor de $Q(s, a)$ a partir do novo estado, ou seja, de quanto o novo estado é considerado promissor (WATKINS, 1989). Na Robótica, em várias situações, a Aprendizagem por Reforço é conveniente, por possibilitar que um bom algoritmo de controle seja obtido, sem que seja necessário um conhecimento acurado do modelo físico do robô. Ela também possibilita que o comportamento do robô se adapte a mudanças em seus próprios parâmetros físicos e a mudanças nas características do ambiente, enquanto o robô estiver em funcionamento.

Chang et al. (2022) publicaram um trabalho sobre robôs seguidores de linha para uso em hospitais, com o sistema comandado por uma placa Arduino (ARDUINO, 2024). Ele detecta a linha usando sensores de infravermelho. Usa a Lógica Fuzzy (KELLER; LIU; FOGEL, 2016) de forma a requerer que o algoritmo de controle seja completamente ajustado antes de o robô ser colocado em uso. Uma vantagem de se usar a Aprendizagem por Reforço em lugar da Lógica Fuzzy é que aquela possibilita que o robô aprenda enquanto estiver em uso, tendo seu comportamento ajustado de forma realista e contínua aos seus próprios parâmetros físicos e às características do ambiente.

A finalidade do robô apresentado por Ribeiro et al. (2019) foi a de aprender, de forma autônoma, a seguir um labirinto formado por caminhos e paredes, a serem detectadas por sensores de infravermelho. Foi usado o Q-learning. Os estados foram as combinações de valores lidos pelos sensores. As possíveis ações consistiram em combinações de valores discretos para a velocidade com direções para a continuidade do movimento. Os valores usados para a recompensa eram negativos para os casos de se superar um tempo predeterminado e para colisões com as paredes e positivos para prosseguimento bem-sucedido. Na execução de simulações computacionais, foi feita extensa exploração de ajustes dos hiperparâmetros que determinam, no cálculo do novo valor de $Q(s, a)$, os pesos atribuídos ao seu valor anterior, ao valor da recompensa e ao máximo valor de $Q(s, a)$ a partir do novo estado. Também foram experimentadas várias formas de partição, em valores discretos, das faixas de valores possíveis contínuas para a velocidade e para a direção a serem adotadas como a próxima ação. Outro fator explorado naquele trabalho foram as possibilidades para a política de tomada de decisão em cada estado. Foram experimentadas desde a escolha aleatória da próxima ação, o que permite uma boa exploração do espaço de estados, até a atribuição de probabilidade muito maior do que as demais para a ação considerada, até o momento, a mais promissora no estado atual. Esta última política faz com que a exploração tenda a ficar mais restrita, porém com boa probabilidade de se chegar rapidamente a um bom desempenho. Foram comparados entre si os ajustes feitos nos hiperparâmetros quanto aos resultados obtidos, sendo que alguns dos ajustes tornaram o robô capaz de aprender a percorrer o labirinto.

MATERIAIS E MÉTODOS

Dos trabalhos citados na seção anterior, foram adotados neste trabalho: o uso de uma placa Arduino (modelo Uno, por estar disponível aos alunos no campus) para o controle do robô, o uso de sensores de infravermelho para a detecção da linha (preta sobre fundo branco) e o uso do algoritmo Q-learning, executado no Arduino. O robô foi montado usando-se componentes comuns de kits didáticos de

robótica. Em sua parte dianteira, foram colocados três sensores de infravermelho, lado a lado, e duas rodas, cada uma acoplada a um motor de corrente contínua. Sob a parte traseira, havia uma roda passiva centralizada. A fonte de energia foi um conjunto de pilhas recarregáveis. O protótipo do robô é visto na Figura 1, com ele colocado na pista na Figura 2.

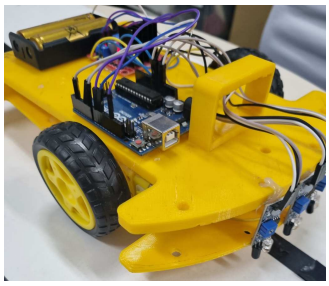

Figura 1: O robô.

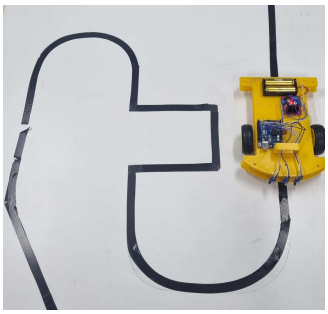

Figura 2: Na pista.

Cada motor foi controlado usando-se modulação por largura de pulso, por meio de um *driver* de potência. Foram usadas apenas duas velocidades angulares no controle de cada motor: uma "baixa" e uma "alta", ambas no mesmo sentido, cada uma correspondendo a um valor de ciclo de trabalho na modulação por largura de pulso (17% e 34%, respectivamente), ajustadas empiricamente.

No uso do Q-learning, foi adotado que o estado s era definido pelos valores de cinco variáveis: os valores de velocidade de cada motor (0 = baixa e 1 = alta) e os valores de leituras de cada sensor (0 = fundo branco e 1 = linha preta). As ações a possíveis eram quatro: (0,0), (0,1), (1,0) e (1,1), interpretadas como os valores de velocidades impostos ao motor esquerdo e ao direito. Portanto, havia $2^5 = 32$ estados e $2^2 = 4$ ações.

A tabela Q foi iniciada com zeros. A cada passo da execução do algoritmo, ela era atualizada por meio da expressão (1) (WATKINS, 1989):

$$Q^{\text{novo}}(s_t, a_t) \leftarrow (1-\alpha) \cdot Q^{\text{anterior}}(s_t, a_t) + \alpha \cdot \left(r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a)\right) \quad (1)$$

na qual s_t é o estado no instante t e indexa uma linha da tabela Q, a_t é a ação executada no instante t e indexa uma coluna da tabela Q, s_{t+1} é o estado no instante $t+1$, atingido em decorrência da ação a_t, $Q^{\text{anterior}}(s_t, a_t)$ é o valor que estava em $Q(s_t, a_t)$ no instante t, antes da execução da ação a_t, $Q^{\text{novo}}(s_t, a_t)$ é o valor atualizado em $Q(s_t, a_t)$, após a execução da ação a_t, α é a taxa de aprendizado, r_{t+1} é o valor da recompensa, conhecido após a realização da ação a_t, γ é o chamado *valor de desconto*, $\max_a Q(st+1, a)$ é o maior valor disponível na tabela Q na linha do estado s_{t+1} atingido após a execução da ação a_t. Neste trabalho, α se iniciava com o valor 1,0 e caía gradualmente até 0,5. γ começava com 0,5 e subia até 0,8, seguindo ideias dadas por Watkins (1989). Periodicamente, a cada 33 passos de execução, uma ação era sorteada, para permitir continuar a exploração de possibilidades. Nos demais 32 passos, era executada a melhor ação segundo a tabela Q daquele momento. Chegou-se a esse valor empiricamente, experimentando-se os valores: 11 (ciclos com 10 ações escolhidas e uma sorteada), 22 e 33. O primeiro desses valores para o qual o robô aprendeu a seguir a linha e passou a errar a trajetória raramente após o aprendizado foi 33.

A Tabela 1 contém os valores de recompensa r_{t+1} (entre -10 e 10) que foram usados, em função do estado s_{t+1} atingido em decorrência da execução de cada ação a_t. Como exemplo, sendo atingido o estado 11, a recompensa tem o valor 10, porque esse é um estado no qual o robô está centralizado sobre a linha e segue em frente. No outro extremo, está o estado 5, no qual apenas o sensor direito

sente a linha, mas o motor direito gira mais rapidamente do que o esquerdo, aumentando o desalinhamento, caso em que a recompensa vale -10. Nos estados de 0 a 3, o robô está fora da linha. Nesses casos, considerou-se que é melhor ele andar devagar em trajetória curva, tentando reencontrar a linha (recompensa igual a -5), do que andar em linha reta devagar (recompensa igual a -6) ou em linha reta rapidamente (recompensa igual a -8). O resto da tabela também foi preenchido subjetivamente, usando-se considerações desse tipo.

Tabela 1: Tabela de valores de recompensa. Colunas: ES: estado, SE: sensor da esquerda, SC: sensor do centro, SD: sensor da direita, ME: motor da esquerda, MD: motor da direita, R: recompensa

ES	SE	SC	SD	ME	MD	R	ES	SE	SC	SD	ME	MD	R
0	0	0	0	0	0	-6	16	1	0	0	0	0	-8
1	0	0	0	0	1	-5	17	1	0	0	0	1	8
2	0	0	0	1	0	-5	18	1	0	0	1	0	-10
3	0	0	0	1	1	-8	19	1	0	0	1	1	-8
4	0	0	1	0	0	-8	20	1	0	1	0	0	0
5	0	0	1	0	1	-10	21	1	0	1	0	1	2
6	0	0	1	1	0	8	22	1	0	1	1	0	2
7	0	0	1	1	1	-8	23	1	0	1	1	1	-5
8	0	1	0	0	0	9	24	1	1	0	0	0	0
9	0	1	0	0	1	-5	25	1	1	0	0	1	-2
10	0	1	0	1	0	-5	26	1	1	0	1	0	-8
11	0	1	0	1	1	10	27	1	1	0	1	1	-5
12	0	1	1	0	0	0	28	1	1	1	0	0	-8
13	0	1	1	0	1	-8	29	1	1	1	0	1	3
14	0	1	1	1	0	-2	30	1	1	1	1	0	3
15	0	1	1	1	1	-5	31	1	1	1	1	1	-10

Para avaliação, o robô foi colocado manualmente dezenas de vezes sobre uma plataforma lisa, com fundo branco e linha preta com curvas, formada por fita isolante (Figura 2). Periodicamente, o programa era reiniciado, com a tabela Q preenchida com zeros, para se verificar novamente o robô aprendendo desde o início.

RESULTADOS E DISCUSSÃO

Nas dezenas de vezes em que foi reiniciado e colocado na pista, o robô iniciava seus movimentos ficando relativamente pouco tempo sobre a linha, saindo dela frequentemente. Com o passar do tempo, os eventos de saída da linha ficavam cada vez mais raros. Ele seguia continuamente trechos cada vez mais longos dela, apresentando, após cerca de dez minutos, o comportamento aprendido esperado. Resta ainda se encontrar uma forma adequada de se apresentar estes resultados quantitativamente, por exemplo, apresentando-se o número de ocorrências de saída da linha por minuto ao longo do tempo.

Uma limitação do algoritmo Q-learning é a de ele só poder lidar com números finitos de estados e de ações. O acréscimo ao algoritmo de mais velocidades permitidas, por exemplo, causa explosão no número de estados e de ações, com aumento na memória requerida para o programa e tempo muito maior para a aprendizagem pelo agente, já que a exploração do espaço de estados e das possíveis ações em cada um deles fica muito extensa. Formas de estender a aprendizagem por reforço a espaços contínuos de estados são apresentadas por François-Lavet et al. (2018). Mas, elas requerem programas mais complexos e com requisitos de hardware superiores ao que é proporcionado por material de uso mais didático e acessível, como o Arduino Uno. Uma outra mudança de abordagem a ser explorada seria, como em Dogru et al. (2022), adotar-se um algoritmo de controle do tipo proporcional integral derivativo, com a ação a ser aprendida passando a ser o ajuste das constantes relacionadas a esses fatores, ao invés de se buscar ajustar diretamente as velocidades dos motores. Também deverá ser

mais explorada a escolha da proporção entre as ações escolhidas na tabela Q e as ações sorteadas.

CONCLUSÕES

Neste trabalho, verificou-se a viabilidade de se implementar um algoritmo de aprendizado por reforço num robô seguidor de linha simples, didático, controlado por uma placa Arduino Uno. Essa placa pôde comportar o algoritmo implementado na linguagem do Arduino, usando uma tabela Q contendo 32 estados e 4 ações possíveis por estado. A placa Arduino Uno foi suficiente, como controladora, para que o robô aprendesse em poucos minutos a seguir a trajetória definida pela linha e mantivesse o comportamento desejado após o aprendizado.

CONTRIBUIÇÕES DOS AUTORES

Todos os autores contribuíram com a elaboração do trabalho e aprovaram a versão submetida.

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela bolsa de iniciação tecnológica concedida a N. S. Mendes.

REFERÊNCIAS

ARDUINO. *What is Arduino?* 2024. Disponível em: https://www.arduino.cc/. Acesso em 14/08/2024.

CHANG, K.-C. et al. Arduino line follower using fuzzy logic control. In: SPRINGER. *The 8th International Conference on Advanced Machine Learning and Technologies and Applications (AMLTA2022)*. [S.l.], 2022. p. 200–210.

DOGRU, O. et al. Reinforcement learning approach to autonomous pid tuning. *Computers & Chemical Engineering*, Elsevier, v. 161, p. 107760, 2022.

FRANÇOIS-LAVET, V. et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., v. 11, n. 3-4, p. 219–354, 2018.

HELFRICH, T. *Forbes - Why Robotics and Artificial Intelligence are the Future Of Mankind*. 2022. Disponível em: https://www.forbes.com/sites/forbestechcouncil/2022/05/31/why-robotics-and-artificial-intelligence-are-the-future-of-mankind/?sh=3e7745391689. Acesso em 14/08/2024.

KELLER, J. M.; LIU, D.; FOGEL, D. B. *Fundamentals of computational intelligence: neural networks, fuzzy systems, and evolutionary computation*. [S.l.]: John Wiley & Sons, 2016.

MIKALEF, P.; GUPTA, M. Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Information & Management*, Elsevier, v. 58, n. 3, p. 103434, 2021.

RIBEIRO, T. et al. Q-learning for autonomous mobile robot obstacle avoidance. In: *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. [S.l.: s.n.], 2019. p. 1–7.

SMITH, R. J. *Encyclopaedia Britannica, Inc. - Robotics*. 2024. Disponível em: https://www.britannica.com/technology/robotics. Acesso em 14/08/2024.

SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.l.]: MIT press, 2018.

WATKINS, C. J. C. H. Learning from delayed rewards. King's College, Cambridge United Kingdom, 1989. Ph.D. Thesis.