Instituto Tecnológico de Aeronáutica - ITA
CT-213 - Inteligência Artificial aplicada à Robótica Móvel
Aluno: Ulisses Lopes da Silva

Relatório do Laboratório 11 - Aprendizado por Reforço Livre de Modelo

---

1 Breve Explicação em Alto Nível da Implementação

Neste laboratório, foram implementados dois algoritmos clássicos de Aprendizado por Reforço Livre de Modelo: **SARSA** e **Q-Learning**. Ambos os algoritmos utilizam uma estrutura tabular para representar os valores de ação (Q-values) em cada estado, atualizando esses valores de acordo com a experiência adquirida durante a interação do agente com o ambiente.

Ambos os algoritmos foram validados em um ambiente simplificado de corredor unidimensional, onde o objetivo é alcançar a célula mais à direita. Os resultados mostraram que os dois algoritmos conseguiram aprender uma política que leva o agente ao estado objetivo, sendo possível observar a convergência da tabela Q e a política gulosa correspondente. As diferenças de comportamento entre SARSA e Q-Learning tornam-se mais evidentes em ambientes mais complexos ou com ruído, mas já são perceptíveis nos caminhos de aprendizado obtidos durante os testes.

1.1 SARSA

O algoritmo **SARSA** (State-Action-Reward-State-Action) segue uma abordagem *on-policy*, ou seja, ele aprende com a mesma política que está sendo executada. A política utilizada foi do tipo ε-greedy, que permite um equilíbrio entre exploração (ações aleatórias) e exploração (ações com maior valor estimado). A função de atualização considera a ação efetivamente executada no próximo estado, o que significa que a política de aprendizado é influenciada diretamente pelas decisões tomadas, mesmo que estas não sejam as ótimas. Isso torna o aprendizado mais conservador, mas também mais robusto em ambientes estocásticos.

1.2 Q-Learning

Por sua vez, o **Q-Learning** é um algoritmo *off-policy*, pois aprende com base em uma política ótima estimada (greedy), independentemente da política de comportamento. Ele atualiza os valores de ação utilizando a melhor ação possível no próximo estado, ainda que essa ação não tenha sido realmente executada. Isso tende a acelerar o processo de aprendizado em ambientes determinísticos como o usado neste laboratório, pois o algoritmo converge mais rapidamente para uma política ótima. No entanto, pode ser menos estável em ambientes com incerteza.

2 Figuras Comprovando Funcionamento do Código

2.1 SARSA

2.1.1 Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

$$\text{Q-table}_{\text{SARSA}} = \begin{bmatrix} -9.33 & -8.35 & -10.77 \\ -10.51 & -9.52 & -11.29 \\ -11.13 & -10.28 & -11.15 \\ -11.70 & -11.32 & -12.07 \\ -12.32 & -12.26 & -12.29 \\ -11.70 & -12.08 & -11.47 \\ -11.09 & -11.42 & -10.62 \\ -10.36 & -11.44 & -9.30 \\ -9.34 & -10.35 & -8.49 \\ -7.19 & -8.36 & -8.46 \end{bmatrix}$$

Política aprendida: `[L, L, L, L, L, R, R, R, R, S]`

A **tabela do SARSA** mostra valores mais negativos em geral, refletindo uma política mais conservadora. Como um algoritmo *on-policy*, SARSA aprende com base nas ações realmente executadas, o que pode incluir decisões subótimas durante a exploração (via política ε-greedy). Isso leva a uma propagação mais lenta das boas recompensas e a uma penalização mais acentuada em estados intermediários.

2.1.2 Convergência do Retorno

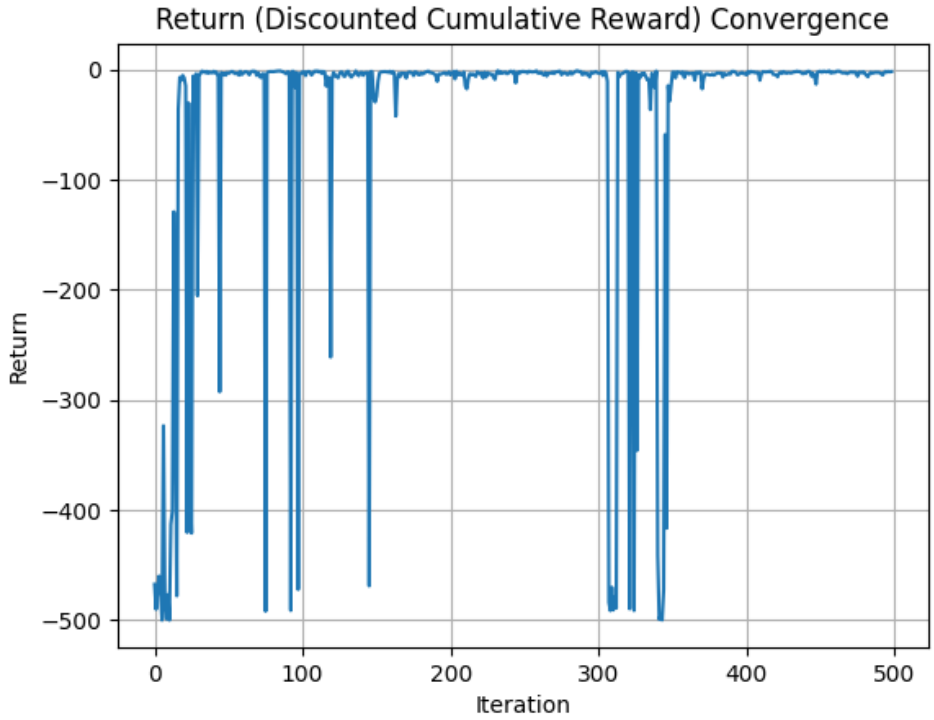

Fig. 1: Evolução do retorno (recompensa acumulada descontada) ao longo dos episódios de treinamento com SARSA.

2.1.3 Tabela Q e Política Determinística que Seria Obtida Através de *Greedy*(Q)

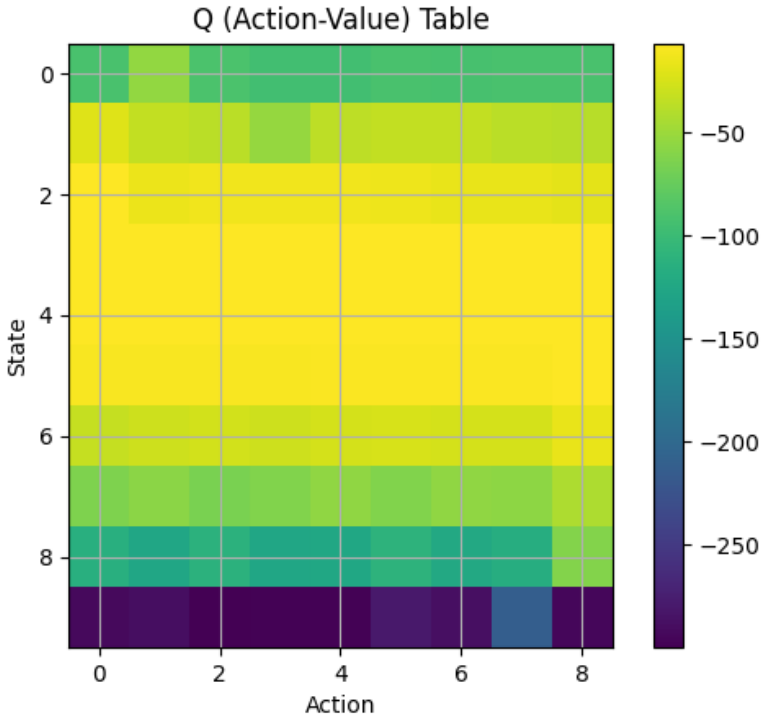

Fig. 2: Tabela de valores de ação (Q-table) aprendida pelo algoritmo SARSA.

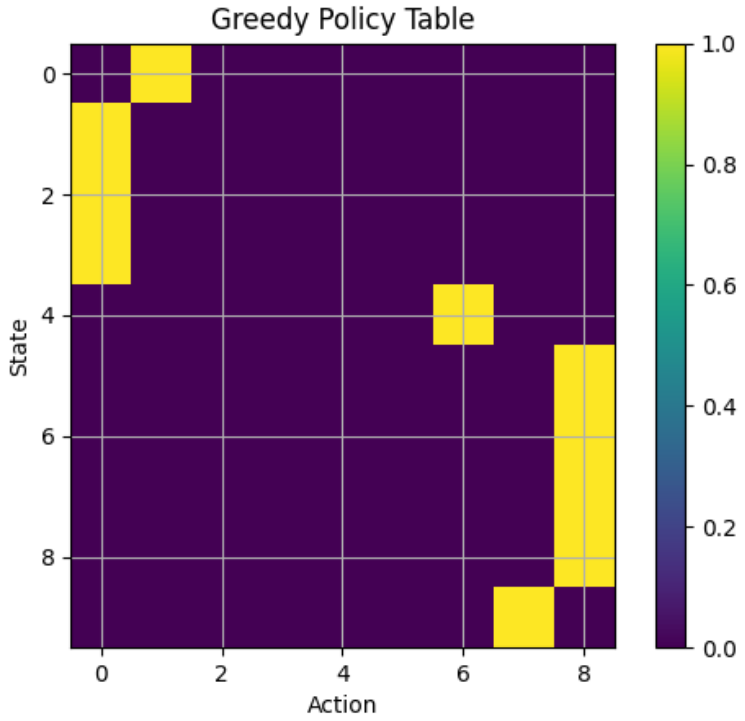

Fig. 3: Política gulosa derivada da Q-table após o treinamento com SARSA.

2.1.4 Melhor Trajetória Obtida Durante o Aprendizado

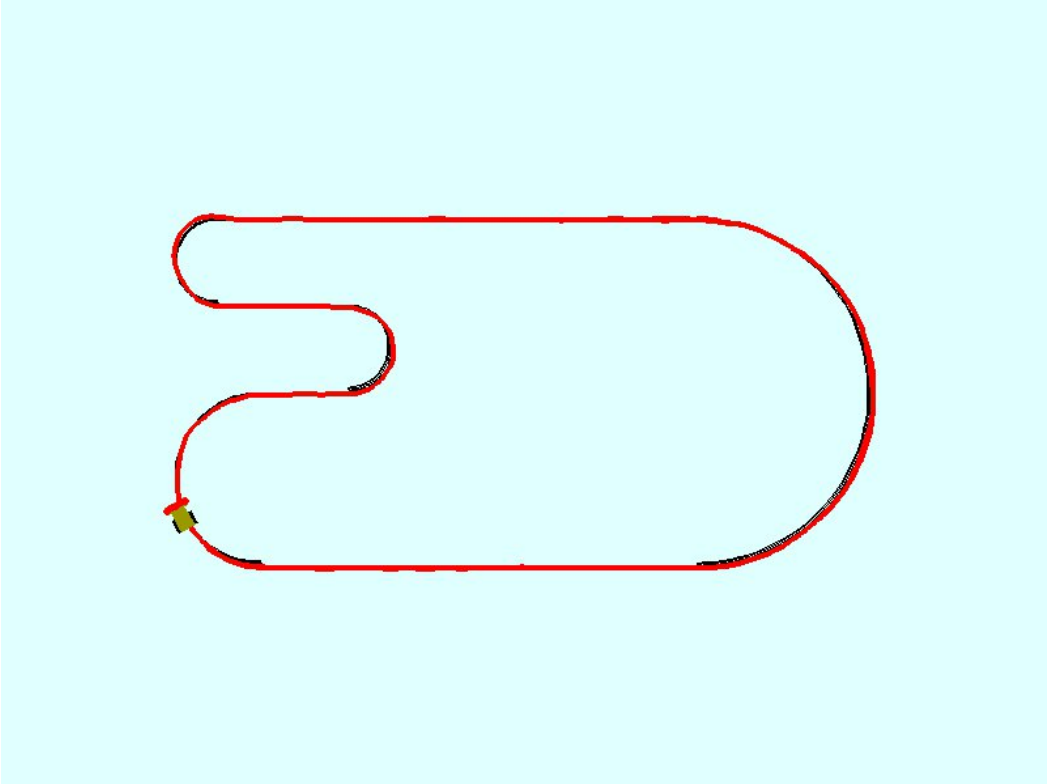

Fig. 4: Trajetória do robô seguidor de linha após treinamento com SARSA no circuito complexo.

2.2 Q-Learning

2.2.1 Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

$$\text{Q-table}_{\text{Q-Learning}} = \begin{bmatrix} -1.99 & -1.00 & -2.97 \\ -2.97 & -1.99 & -3.93 \\ -3.45 & -2.97 & -4.44 \\ -4.28 & -3.94 & -4.33 \\ -5.12 & -4.89 & -4.89 \\ -4.22 & -4.65 & -3.94 \\ -3.65 & -4.30 & -2.97 \\ -2.96 & -3.93 & -1.99 \\ -1.99 & -2.97 & -1.00 \\ 0.00 & -0.99 & -0.99 \end{bmatrix}$$

Política aprendida: `[L, L, L, L, L, R, R, R, R, S]`

A **tabela do Q-Learning** apresenta valores mais suaves e moderados, com maior proximidade entre os estados vizinhos, o que é característico de sua natureza *off-policy*, onde a atualização considera a melhor ação no estado seguinte, independentemente da ação efetivamente tomada. Isso tende a promover uma convergência mais rápida e precisa para a política ótima.

2.2.2 Convergência do Retorno

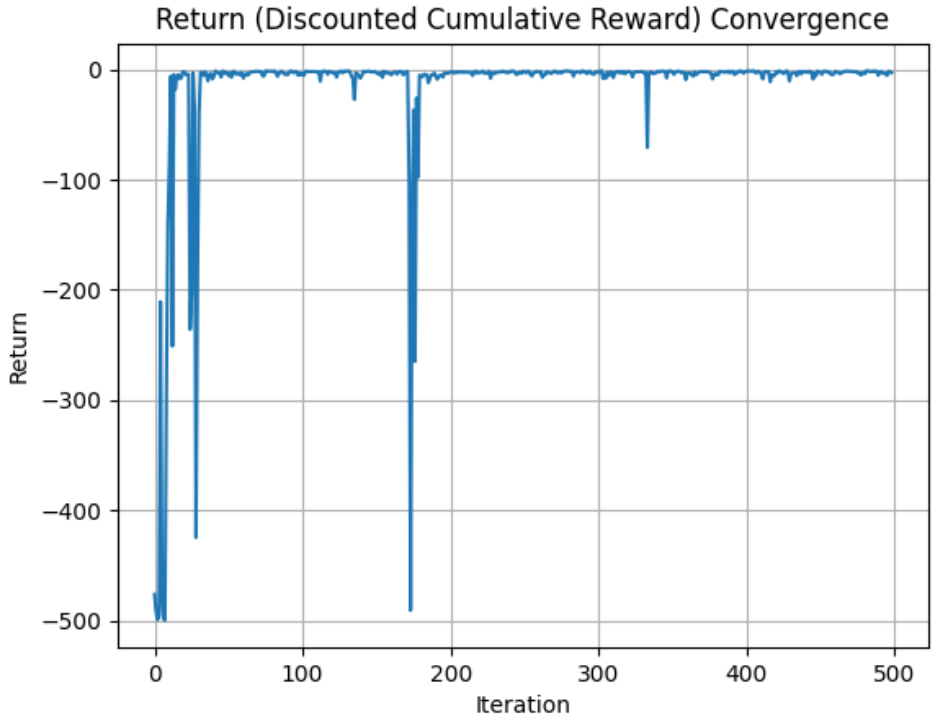

Fig. 5: Evolução do retorno (recompensa acumulada descontada) ao longo dos episódios de treinamento com Q-Learning.

2.2.3 Tabela Q e Política Determinística que Seria Obtida Através de *Greedy*(Q)

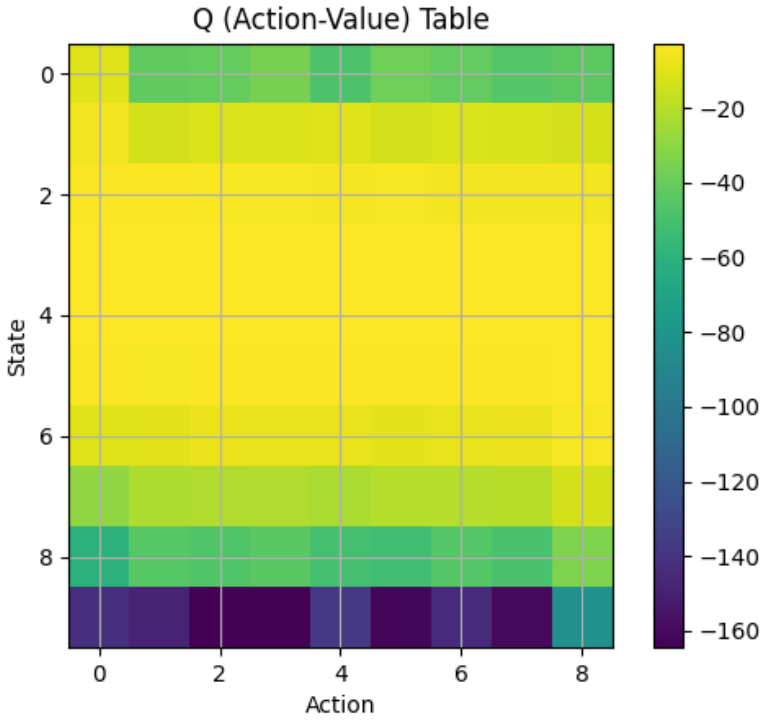

Fig. 6: Tabela de valores de ação (Q-table) aprendida pelo algoritmo Q-Learning.

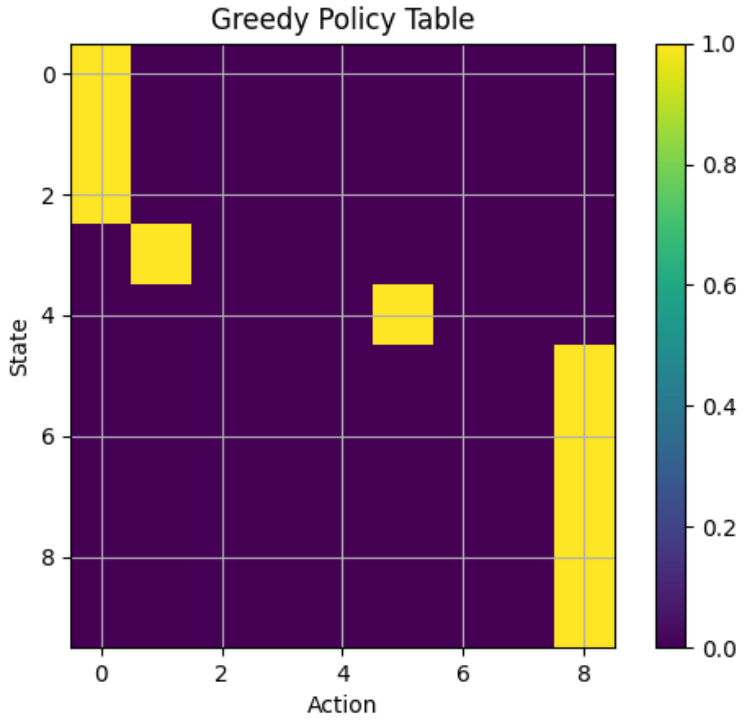

Fig. 7: Política gulosa derivada da Q-table ao final do treinamento com Q-Learning.

2.2.4 Melhor Trajetória Obtida Durante o Aprendizado

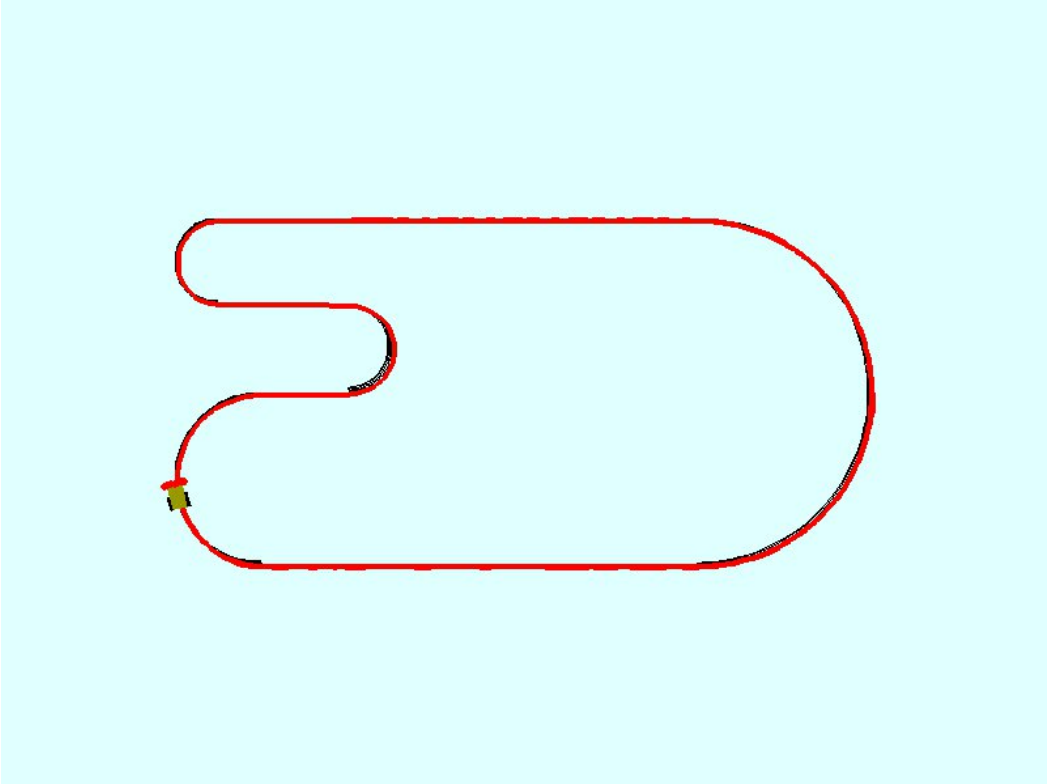

Fig. 8: Trajetória do robô após o treinamento com Q-Learning. Observa-se alinhamento preciso à linha.

3 Discussão dos Resultados

Análise dos Resultados com o Algoritmo SARSA

O treinamento do robô seguidor de linha com o algoritmo SARSA apresentou os seguintes resultados:

- **Trajetória Obtida:** A Fig. 4 mostra que o robô foi capaz de percorrer grande parte da pista complexa após o treinamento. Apesar de pequenas oscilações e desvios em curvas mais fechadas, a política aprendida foi suficiente para permitir a navegação ao longo do trajeto. Isso demonstra que o SARSA conseguiu aprender uma política útil mesmo em um ambiente com dinâmica não trivial e atraso no controle.
- **Tabela de Ações-Q (Q-table):** A Fig. 2 apresenta a matriz de valores Q. Observa-se que os estados associados a grandes erros (normalmente nas linhas inferiores) apresentam valores mais negativos, o que está de acordo com a penalização severa nesses estados. Já os valores menos negativos aparecem próximos ao centro da matriz, indicando que o robô foi mais bem recompensado ao adotar ações que corrigem suavemente o erro.

- **Política Gulosa Aprendida:** A Fig. 3 mostra a política determinística derivada da Q-table. Pode-se notar que o robô aprendeu a escolher ações mais extremas nos estados com maior erro (ações nas bordas da matriz), enquanto em estados centrais, onde o erro é pequeno, a política sugere ações mais suaves. Isso condiz com a expectativa de um controlador eficiente que tenta manter o robô centrado sobre a linha.

- **Histórico de Retorno (Convergência):** O gráfico da Fig. 1 mostra a evolução do retorno por episódio durante o treinamento. Observa-se um padrão típico de aprendizado com ε-greedy: valores iniciais muito negativos, seguidos por picos esporádicos positivos e, gradualmente, uma estabilização próxima de zero. Os picos negativos representam episódios de exploração com falhas, enquanto os episódios com retorno elevado indicam que o robô conseguiu completar a pista com sucesso ou quase isso, tendência esta que é coerente com o funcionamento apropriado desse algoritmo.

Análise dos Resultados com o Algoritmo Q-Learning

O treinamento do robô seguidor de linha com o algoritmo Q-Learning apresentou os seguintes resultados::

- **Trajetória Obtida:** A Fig. 8 mostra que o robô foi capaz de percorrer toda a pista de forma estável e precisa. A trajetória é bem ajustada à linha preta, com correções suaves, o que demonstra a eficácia da política aprendida para manter o robô centrado no percurso.

- **Tabela de Ações-Q (Q-table):** A Fig. 6 apresenta a matriz de valores Q aprendida. Nota-se uma organização mais clara dos valores ao longo dos estados, com valores menos negativos nas regiões centrais (estados com menor erro de seguimento) e valores mais penalizados nas extremidades, o que é esperado para um algoritmo que busca minimizar o erro de rastreamento de linha.

- **Política Gulosa Aprendida:** A política gulosa derivada da Q-table é apresentada na Fig. 7. Ela mostra que o agente aprendeu a tomar ações bem definidas para cada estado, com uso predominante de ações centrais (ações de correção leve) para estados centrais, e ações mais extremas nos estados de maior desvio. Isso caracteriza uma política refinada e estável.

- **Histórico de Retorno (Convergência):** O gráfico da Fig. 5 mostra a evolução do retorno por episódio. Após uma fase inicial de aprendizagem com valores bastante negativos (exploração), o algoritmo convergiu rapidamente para altos retornos, estabilizando-se próximo de zero. Isso indica que o robô passou a completar a pista frequentemente, recebendo poucas penalizações.

De forma geral, os resultados mostram que o algoritmo Q-Learning, por ser off-policy e usar a política ótima na atualização dos valores, é capaz de aprender de forma mais agressiva e eficiente que o SARSA. A convergência foi mais rápida e a política obtida demonstra maior precisão, o que resultou em trajetórias mais suaves e consistentes.