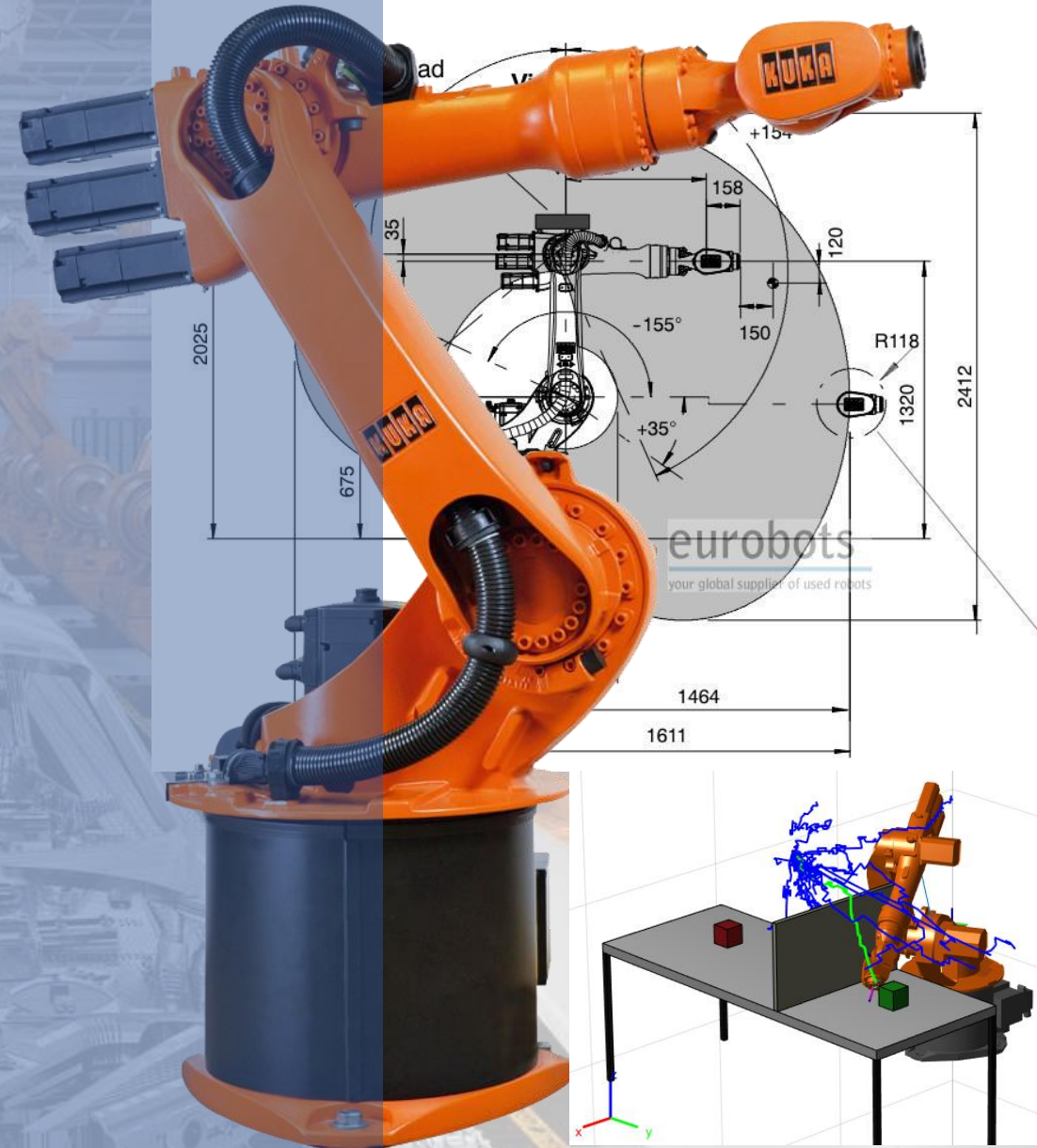


Controle de Robô Manipulador com Aprendizado por Reforço

Lucas Pereira Cotrim
Marcos Menon José

Eduardo Lobo Lustosa Cabral

São Paulo
Dezembro de 2019



Sumário

The background image shows a modern automotive manufacturing plant. In the center, a silver car chassis is positioned on a conveyor belt. On either side of the chassis, there are several large, orange robotic arms (industrial robots) used for assembly. To the right, a robotic arm is holding a tablet that displays a 3D model of a car part and some data graphs. The overall scene is brightly lit, typical of a factory environment.

1. **Introdução**
2. **Estado da Arte**
3. **Fundamentos Teóricos**
4. **Detalhamento do Projeto**
5. **Resultados**
6. **Conclusão**

Introdução

1. **Introdução**
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão

Introdução

Robôs Industriais em Operação



Fonte: Executive Summary World Robotics 2018 Industrial Robots

Introdução

Programação de Robôs Manipuladores

Método
Tradicional

Por
Demonstração

Inteligência
Artificial

Programação On-Line

- Realizada no próprio ambiente de trabalho
- Requer robô físico, tempo de ociosidade
- Maior semelhança com tarefa real

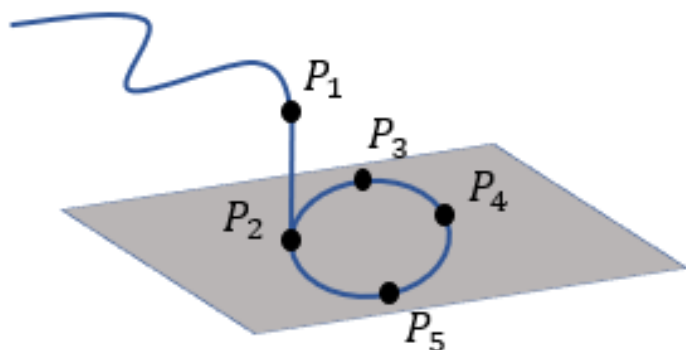
Programação Off-line

- Redução de tempo ocioso
- Teste de controladores novos
- Redução de riscos de acidentes
- Diferenças entre simulação e realidade

Introdução

Programação Tradicional

- Definição manual de pontos no espaço de trabalho
- Interpolação de trajetórias entre pontos definidos



Vantagens

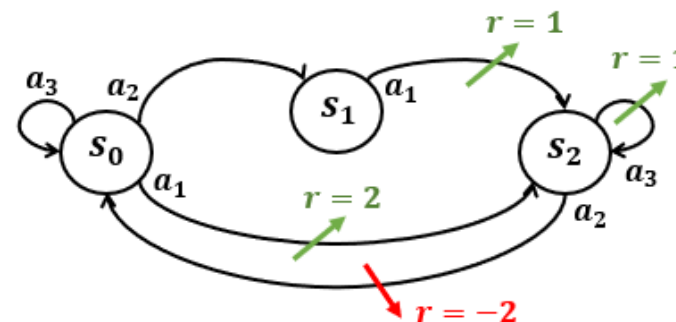
- Alta precisão
- Maior segurança

Desvantagens

- Tarefas fixas e repetitivas
- Baixa flexibilidade
- Procedimento manual

Aprendizado por Reforço

- Agente é treinado a partir de interações com ambiente
- Função recompensa representa comportamento desejado para execução de tarefa



Vantagens

- Exploração
- Aprimoramento contínuo sob treino
- Processo automatizado

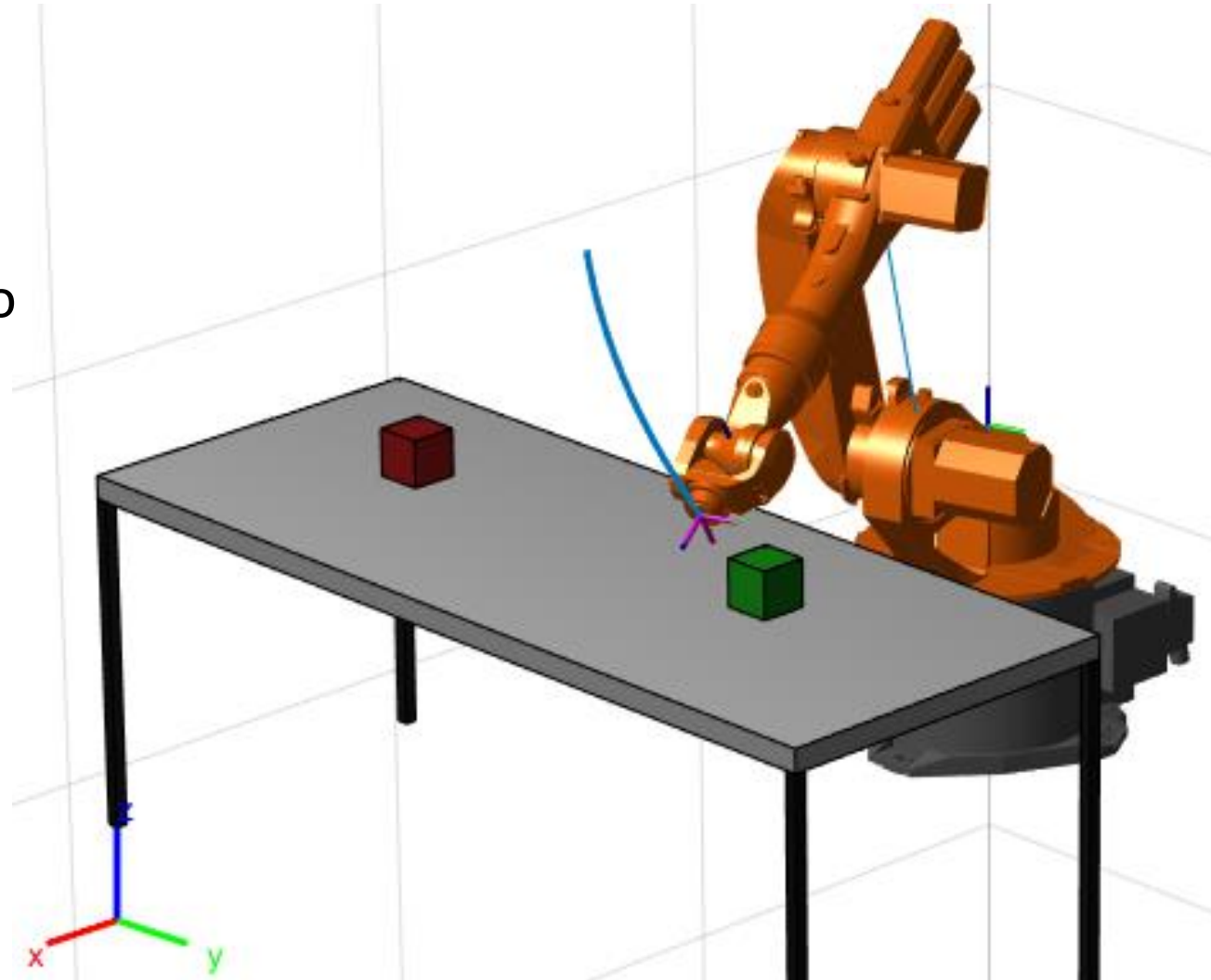
Desvantagens

- Tempo de treino
- Simulação x Realidade
- Determinação de função recompensa

Introdução

Especificação da Tarefa

- KUKA-KR16
- Tarefa de Posicionamento com Obstáculo
- Aprendizado autônomo



Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



Estado da Arte

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão

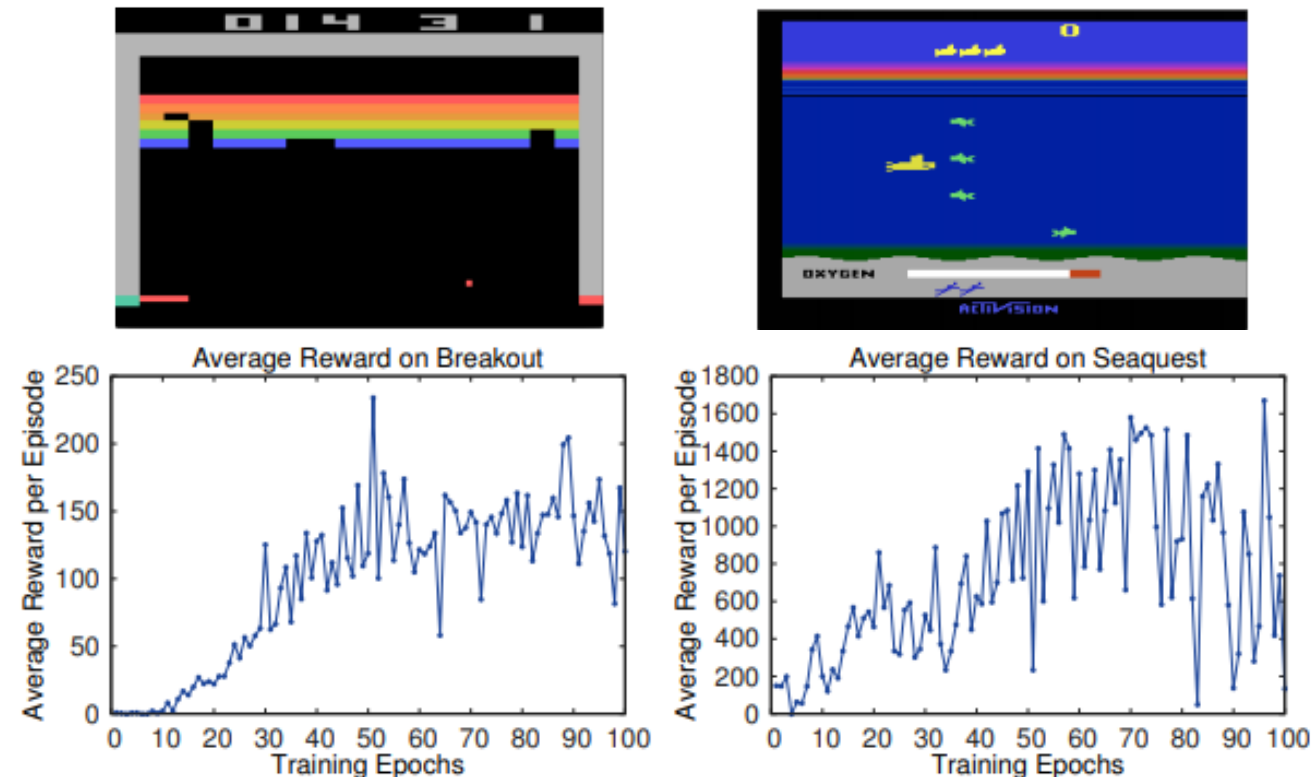


Estado da Arte

Aprendizado por Reforço

- Ambientes de simulação →
- Facilidade de obtenção de dados
- Treino automatizado
- Aplicações comuns em jogos
- Agentes treinados a partir de imagens

Figura 1: Capturas de telas e gráficos de recompensas médias ao longo do treino de agente por algoritmo DQN em simulador de Atari para jogos *breakout* (esquerda) e *seaquest* (direita) (fonte: MNIH et al, 2013)



Estado da Arte

Aprendizado por Reforço em Robótica

- Aplicações práticas limitadas →
- Transferência de aprendizado de simulação para ambiente real

- Alto custo e tempo de treino
- Requisitos de segurança

- Sucesso sob estados de dimensão reduzida, como articulações do robô (FRANCESCHETTI, 2016).
- Sucesso parcial para treino a partir de imagens (JAMES, S; JOHNS, E 2016).

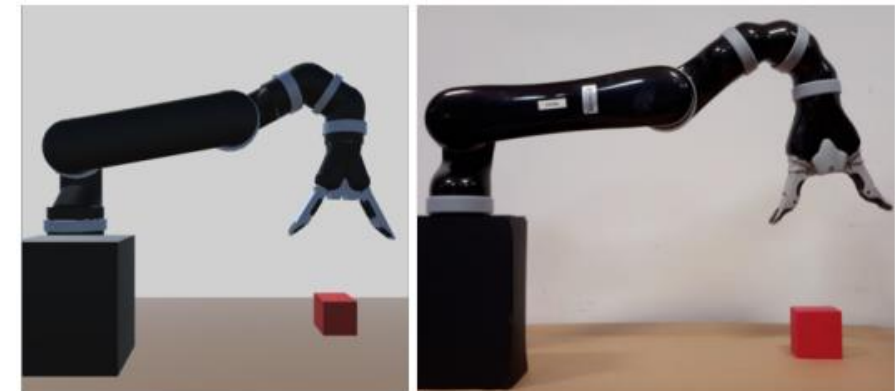
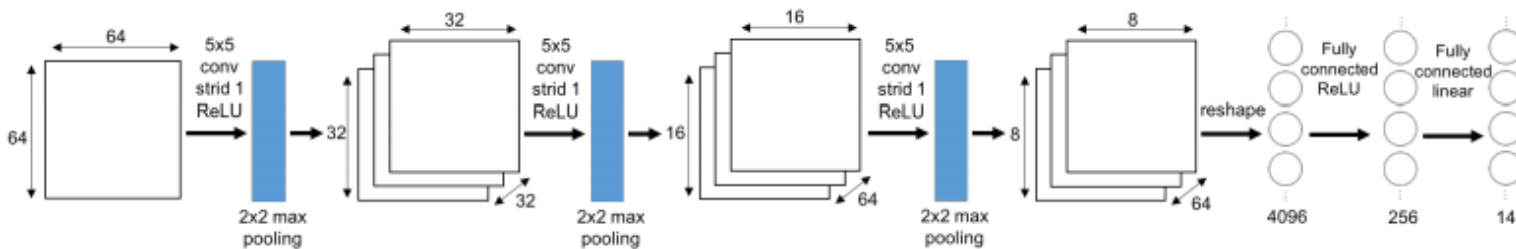


Figura 2: Estrutura de rede DQN (esquerda) e imagens de ambiente de simulação e ambiente real (direita) durante treino de agente para tarefa de segurar objeto. Sucesso parcial na transferência do aprendizado de simulação para ambiente real (fonte: JAMES, S; JOHNS, E 2016).

Estado da Arte

Aprendizado por Reforço em Robótica

- Treino com múltiplos robôs físicos



- Implementação de algoritmos assíncronos DDPG e NAF (Extensão de DQN para ações contínuas)
- Tempo de treino reduzido consideravelmente com múltiplos agentes

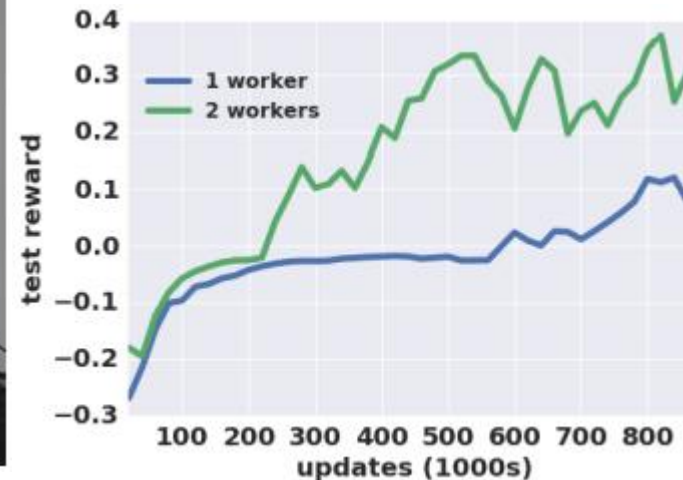


Figura 2: Ambiente de treino com múltiplos agentes (esquerda) e gráfico comparativo de curvas de aprendizado para treinos com 1 e 2 agentes (direita) (fonte: GU, S et al, 2016).

Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



Fundamentos Teóricos

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão

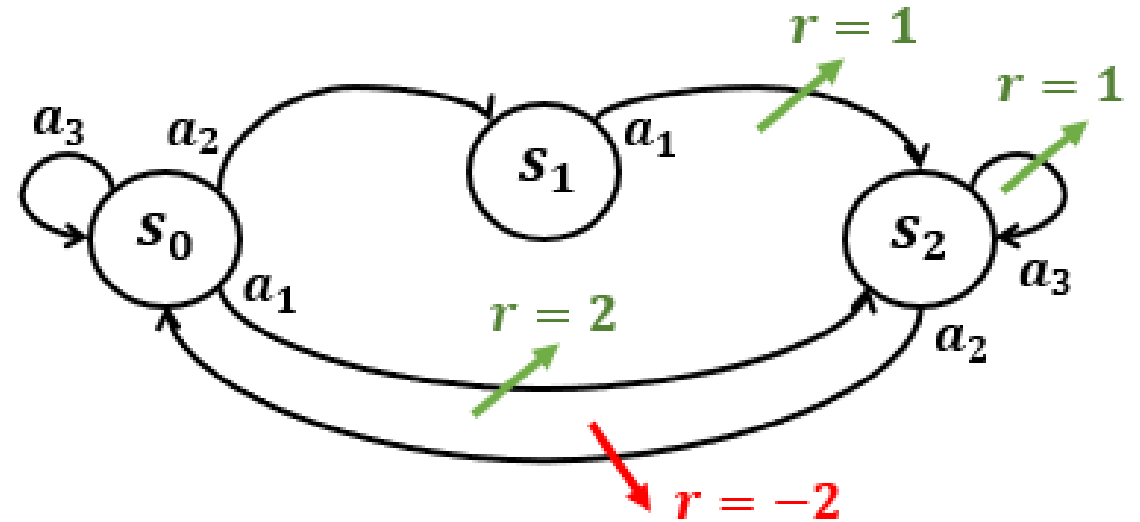


Fundamentos Teóricos

Processos de Decisão de Markov (MDPs)

- Processos estocásticos em tempo discreto
- Modelos para otimização de tomada de decisões
- Definidos como $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, onde:
 - \mathcal{S} é um conjunto de estados s .
 - \mathcal{A} é um conjunto de ações a .
 - \mathcal{P} é um conjunto de probabilidades $p(s'|s, a)$.
 - \mathcal{R} é um conjunto de recompensas $r(s, a)$.
- A solução de um MDP é uma função política $\pi: \mathcal{S} \rightarrow \mathcal{A}$ que determina a ação $a = \pi(s)$ a ser tomada em cada estado para maximizar o valor esperado de recompensas cumulativas

$E_{a_t \sim \pi(s_t)} [\sum_{t=0}^T \gamma^t r(s_t, a_t)]$, onde $\gamma \in (0,1)$ é um fator de desconto.



Fundamentos Teóricos

- Obtenção de uma política de ações ótima π^* .
- Avaliar valores de estados e ações segundo a política atual π .

Função Valor dos Estados

$$V_{\pi}(s) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right] = E_{\pi}[G_t | s_t = s]$$

Função Valor dos Pares Estado-Ação

$$Q_{\pi}(s, a) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right] = E_{\pi}[G_t | s_t = s, a_t = a]$$

- Para simplificar a notação introduz-se o conceito de retorno cumulativo a partir do instante t :

$$\begin{aligned} G_t &= \sum_{k=0}^{\infty} \gamma^k r_{t+k} \\ &= r_t + \gamma r_{t+1} + \dots \end{aligned}$$

Fundamentos Teóricos

- A política de ações π pode ser não determinística, de modo que $\pi(a|s)$ é a probabilidade de o agente tomar a ação a no estado s .
- É possível expandir as funções $V_\pi(s)$ e $Q_\pi(s, a)$ iterativamente para obter a equação de Bellman:

Equação de Bellman

$$\begin{aligned} Q_\pi(s, a) &= E_\pi[G_t | s_t = s, a_t = a] \\ &= E_\pi[r_t + \gamma G_{t+1} | s_t = s, a_t = a] \\ &= \sum_{s' \in S} p(s' | s, a) \left[r(s, a) + \gamma \sum_{a' \in A} \pi(a' | s') Q_\pi(s', a') \right] \end{aligned} \quad (1)$$

- Para políticas e sistemas determinísticos, temos: $Q_\pi(s, a) = r(s, a) + \gamma \max_{a' \in A} Q_\pi(s', a')$

Fundamentos Teóricos

Classes de Algoritmos

Iteração sobre Política de Ações

- Parametrização da função política de ações $\pi_{\theta}(a|s)$
- Aumento da performance $J(\theta)$

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta)$$

- Onde $J(\theta) = V_{\pi_{\theta}}(s_0)$ é uma função que mede a performance da política atual $\pi_{\theta}(a|s)$

Iteração sobre Função Valor

- Parametrização da função valor do par estado-ação $Q_{\theta}(s, a)$
- Diminuição do custo $L(\theta)$

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} L(\theta)$$

- Onde $L(\theta) = \frac{1}{2} (Q_{\theta}(s, a) - y)^2$, com $y = \begin{cases} r(s, a), & \text{se } s' \text{ é terminal} \\ r(s, a) + \gamma \max_{a' \in A} Q_{\theta}(s', a'), & \text{caso contrário} \end{cases}$
- A função $Q_{\theta}(s, a)$ é atualizada para satisfazer a Equação de Bellman

Fundamentos Teóricos

Classes de Algoritmos

Iteração sobre Política de Ações

- Algoritmo REINFORCE episódico:
- São simuladas N episódios a cada época do treino e estima-se o gradiente da performance como

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{T-1} G_t \frac{\nabla \pi_{\theta}(\mathbf{s}_t, \mathbf{a}_t)}{\pi_{\theta}(\mathbf{s}_t, \mathbf{a}_t)}$$

Iteração sobre Função Valor

- Algoritmo DQN:
- São simuladas N trajetórias e armazena-se transições $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$ em um *buffer* B
- São amostradas N_{batch} transições aleatoriamente de B para atualização da função $Q_{\theta}(\mathbf{s}, \mathbf{a})$

Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



Detalhamento do Projeto

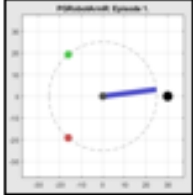
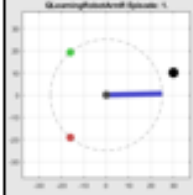
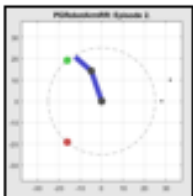
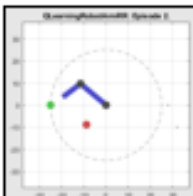
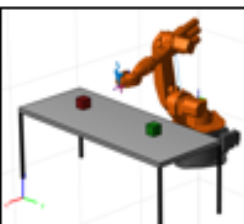
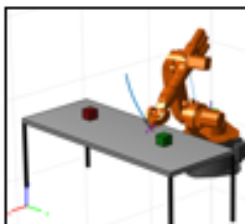
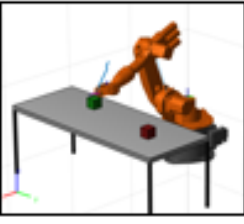
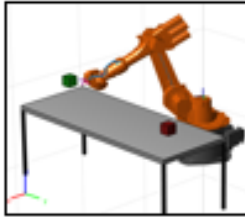
1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. **Detalhamento do Projeto**
5. Resultados
6. Conclusão



Detalhamento do Projeto

Projetos de Teste

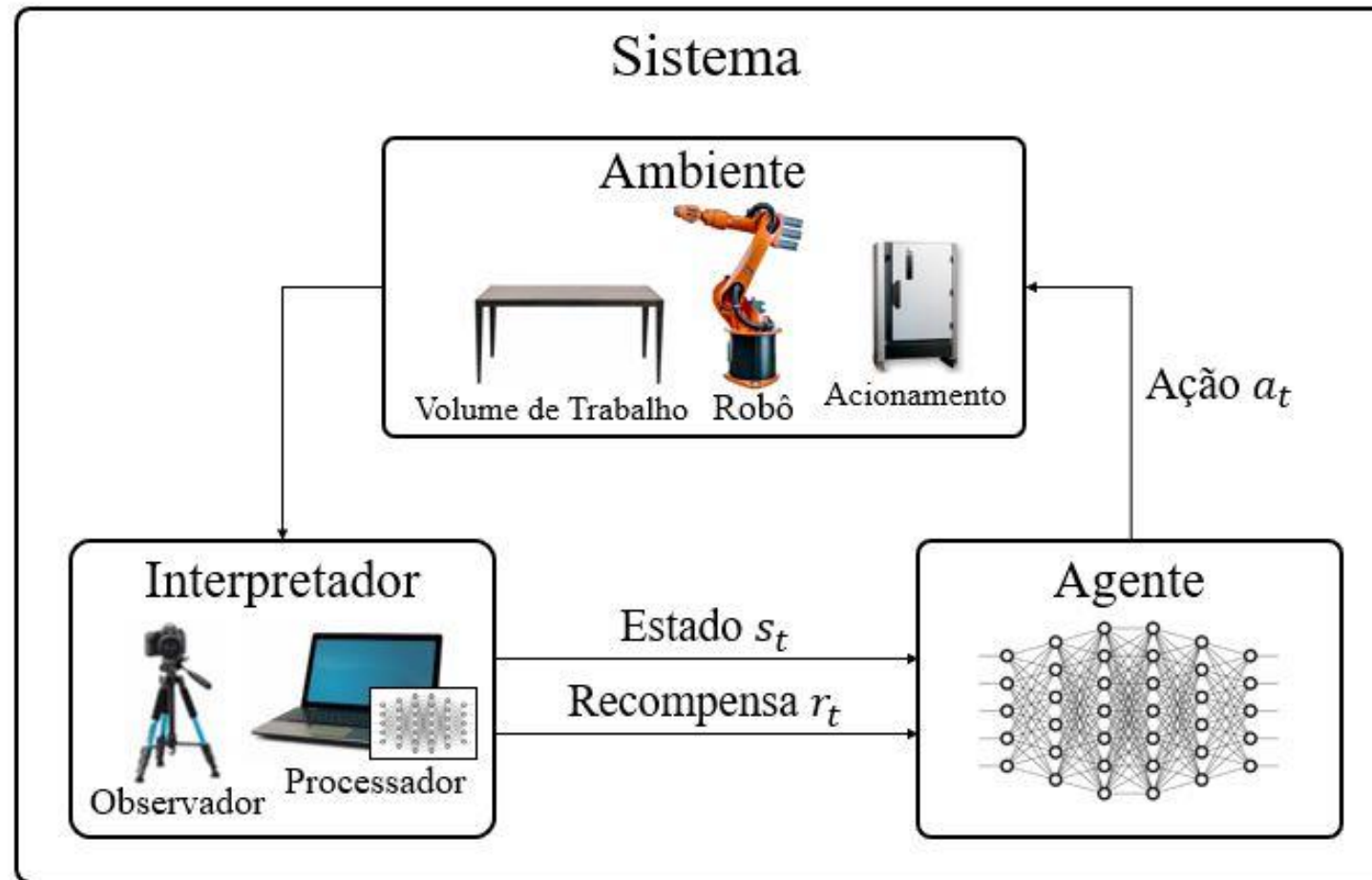
- Comparação de algoritmos e determinação de hiper-parâmetros
- Simplificações:
 - Redução do número de graus de liberdade
 - Redução da dimensão dos estados
 - Redução da dimensão das ações
 - Configuração fixa de obstáculo e posição de destino

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

Detalhamento do Projeto

Projeto Final

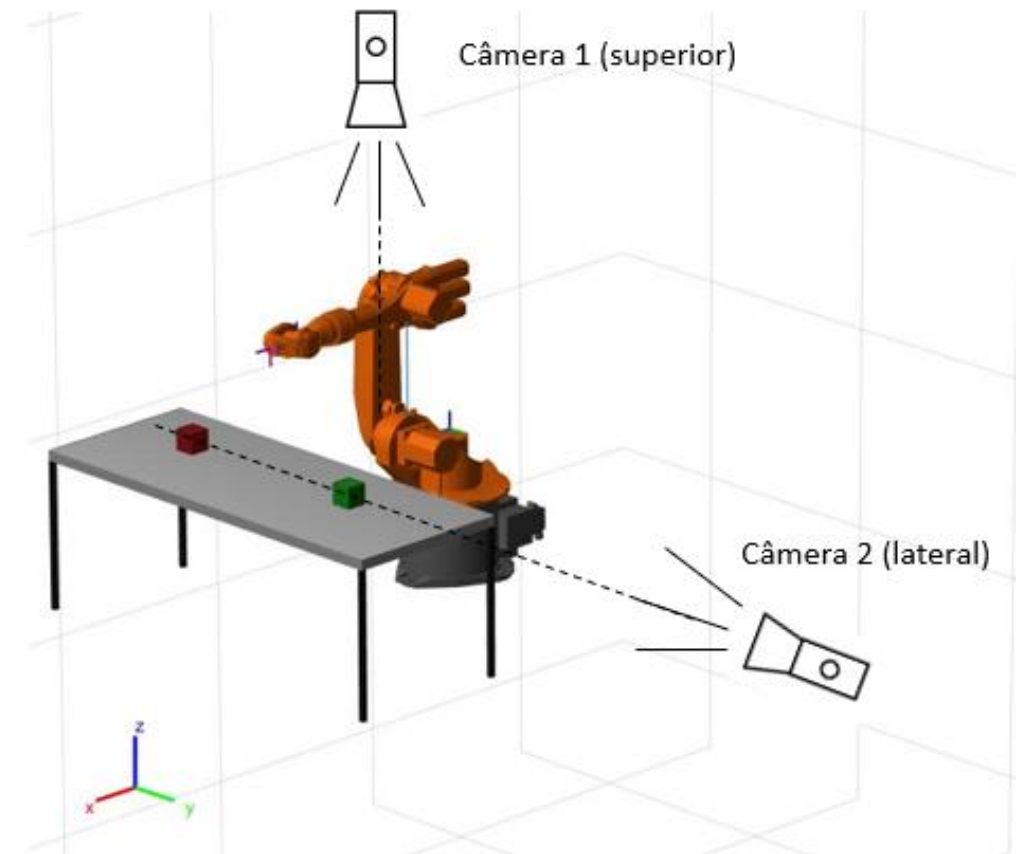
Arquitetura de Controle



Detalhamento do Projeto

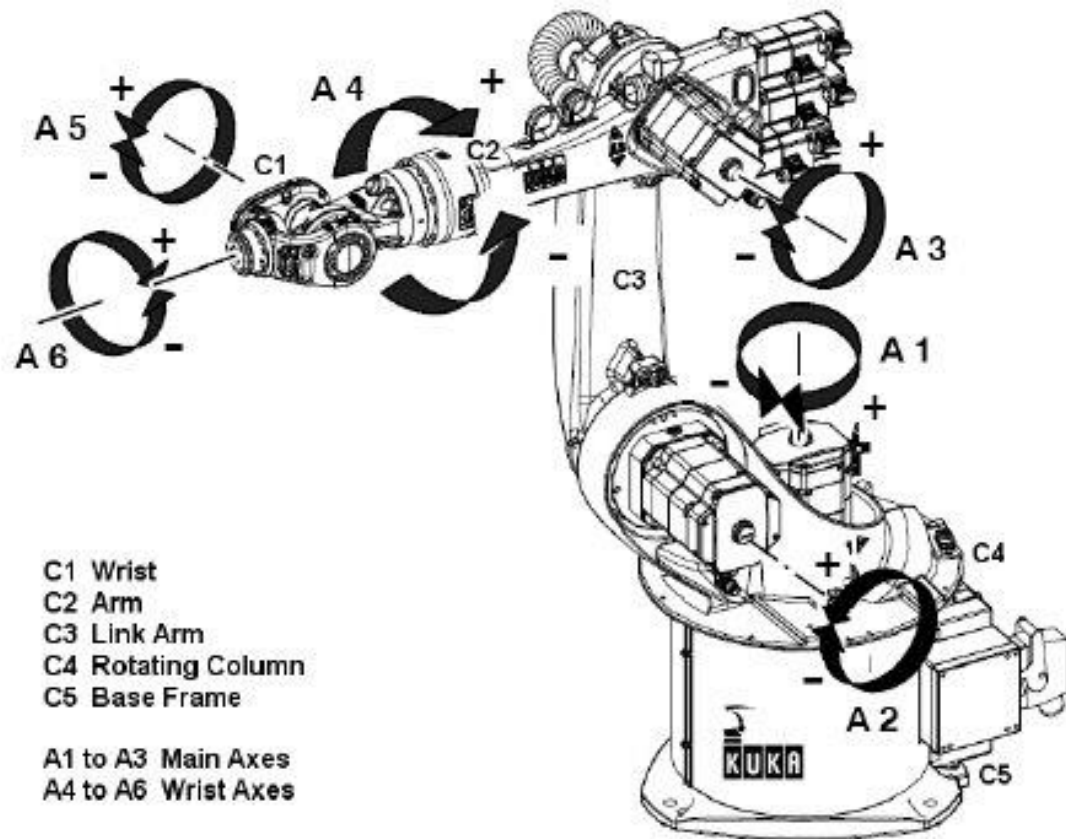
Espaço de Estados

- Cada estado s é um conjunto de duas imagens RGB de tamanho $24 \times 24 \times 3$ (dimensão 3456)
- Configuração de câmeras para visualização lateral e superior



Detalhamento do Projeto

Espaço de Ações



$$A = \begin{matrix} & \begin{matrix} a_{A_1} & a_{A_2} & a_{A_3} & a_{A_4} & a_{A_5} \end{matrix} \\ \begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 0 \\ 1 & 1 & 1 & -1 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 0 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \end{matrix}$$

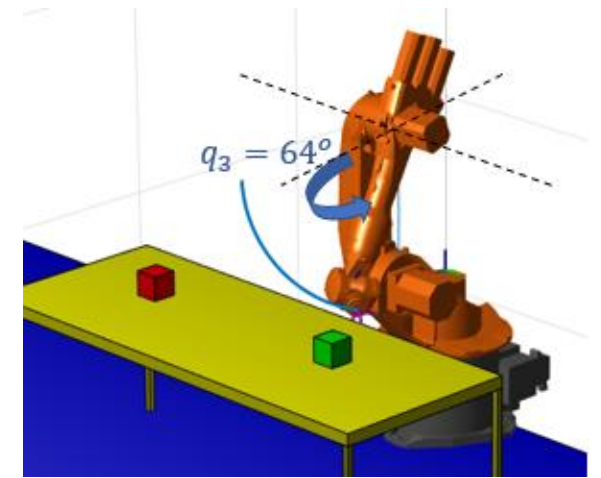
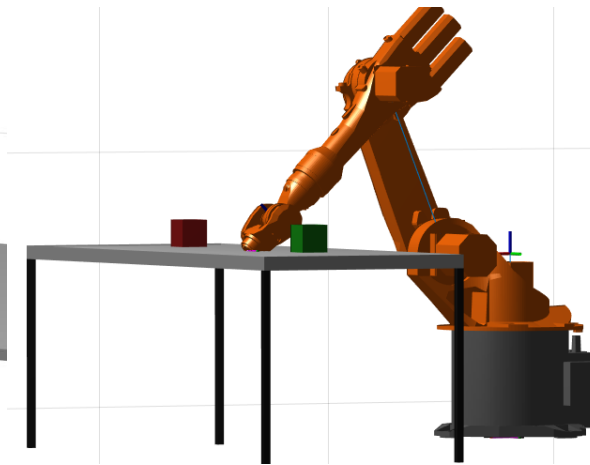
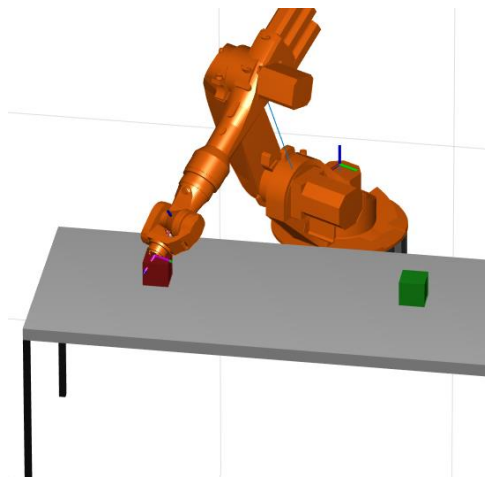
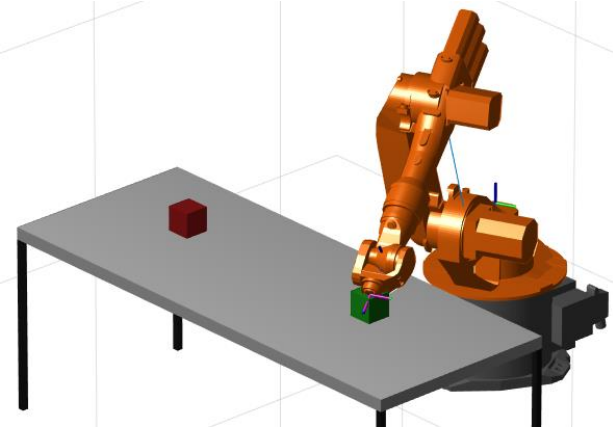
243 ações discretas

Figura 3: Diagrama de Articulações de robô KUKA-KR16 (fonte: KUKA ROBOTICS: Kuka KR6, KR16, KR16 L6, KR 16S specification, 2003)

Detalhamento do Projeto

CrITÉRIOS de Parada

- Posição Desejada Alcançada
- Colisão com Obstáculo ou Mesa
- Limite de Curso de Articulação



- Simulação de trajetória é encerrada se
 - Um dos três critérios de parada é satisfeito
 - Número máximo T de transições é alcançado
- O agente recebe um bônus ou uma penalidade em função do critério de parada

Detalhamento do Projeto

Funções de Recompensa

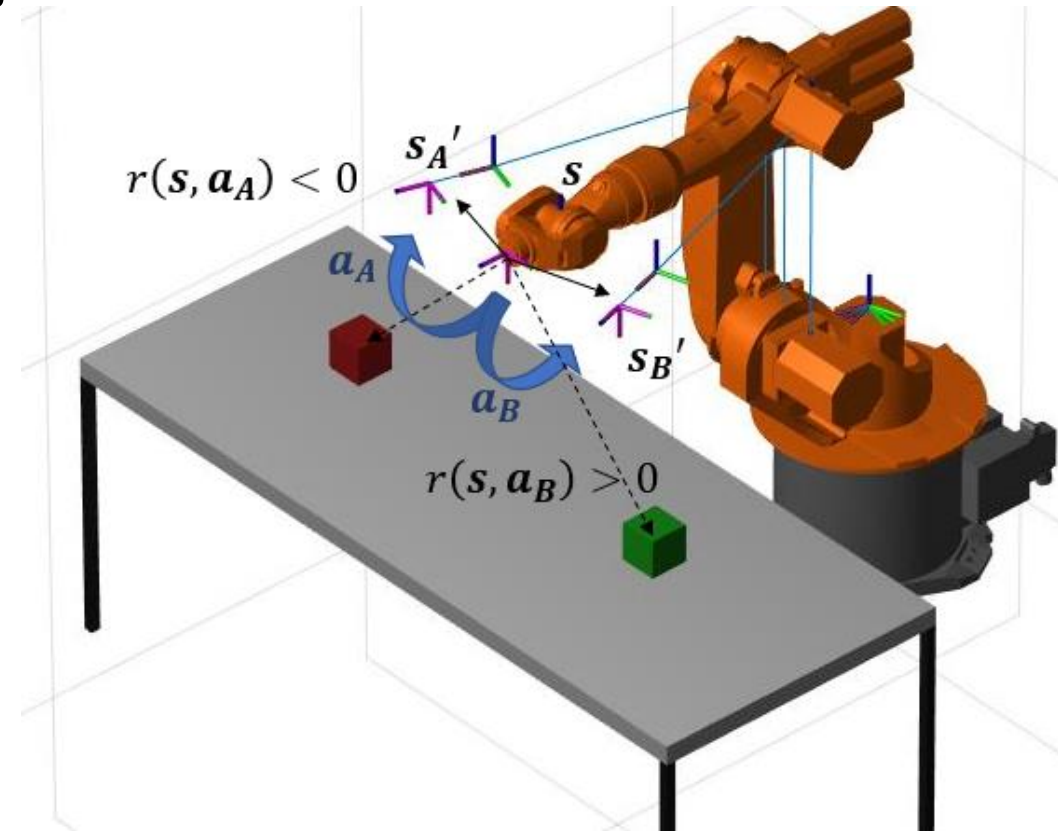
- Comparação entre diferentes funções de recompensa em projetos de teste
- Determinação de função mais adequada para projeto final

$r_1(s, a)$: Distâncias Absolutas

$r_2(s, a)$: Aproximação ou Distanciamento

$r_3(s, a)$: Projeção de Vetor Deslocamento

$$r_3(s, a) = [k_s r_{setpoint} + k_o r_{obstacle}] + B_{goal} + P_{joint\ boundary} + P_{collision}$$



Detalhamento do Projeto

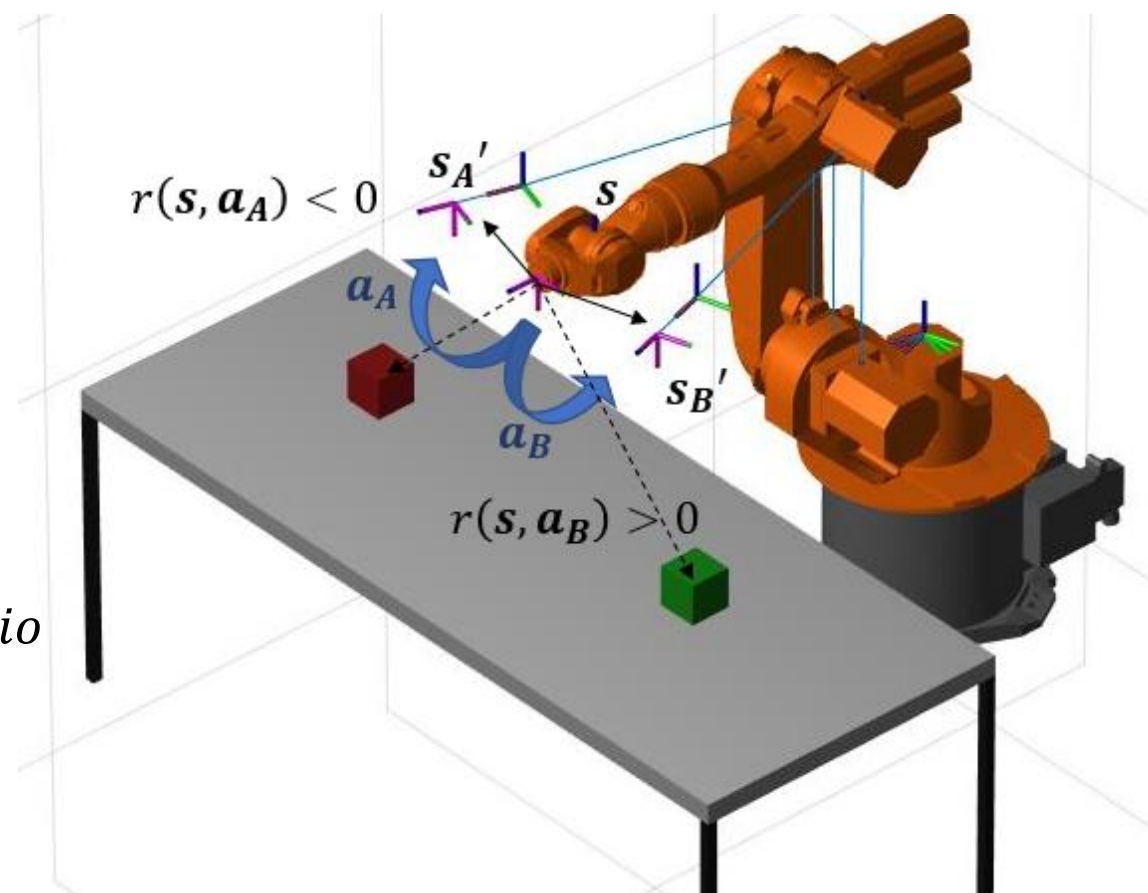
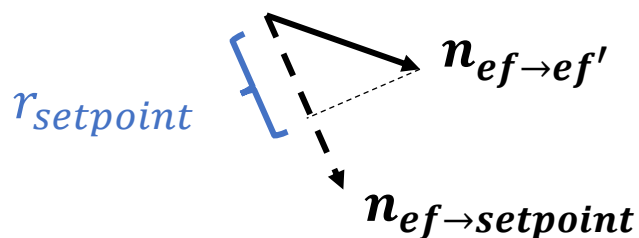
Função de Recompensa Escolhida: $r_3(s, a)$

Posição de destino e de obstáculo

$$r_3(s, a) = \underbrace{[k_s r_{\text{setpoint}} + k_o r_{\text{obstacle}}]}_{\text{Posição de destino e de obstáculo}} + \underbrace{B_{\text{goal}}}_{\text{Bônus}} + \underbrace{P_{\text{joint boundary}} + P_{\text{collision}}}_{\text{Penalidades}}$$

$$r_{\text{setpoint}}(s, a) = \mathbf{n}_{ef \rightarrow ef'} \cdot \mathbf{n}_{ef \rightarrow \text{setpoint}}$$

$$r_{\text{obstacle}}(s, a) = \begin{cases} 0, & \text{se } \|\mathbf{p}_{ef'} - \mathbf{p}_{\text{obstacle}}\| > r_{\text{infl}} \\ -(\mathbf{n}_{ef \rightarrow ef'} \cdot \mathbf{n}_{ef \rightarrow \text{obstacle}}), & \text{caso contrário} \end{cases}$$



Detalhamento do Projeto

Algoritmos Implementados

- REINFORCE episódico
- Q-Learning (tabela)
- **DQN**

A partir dos resultados parciais obtidos optou-se pelo DQN

REINFORCE

Algoritmo 1: REINFORCE Episódico

- Inicializa Robô, *setpoint*, *obstáculo*, estado inicial \mathbf{s}_0 e espaço de ações \mathcal{A} ;
- Inicializa Hiperparâmetros (bônus e penalidades, tamanho da rede, número de *timesteps*, trajetórias e épocas, fator de desconto γ e taxa de aprendizado α);
- Inicializa estruturas de armazenamento de épocas e políticas de ações;
- Inicializa política de ações parametrizada π_{θ_0} aleatória e armazena em PolicyBuffer(1);

Gera N trajetórias $\{\tau_n\}_{n=1}^N$ a partir de política de ações π_{θ_0} , onde

$$\tau_n = \mathbf{S}_0, \mathbf{A}_0, R_0, \dots, \mathbf{S}_{T-1}, \mathbf{A}_{T-1}, R_T;$$

Calcula retornos $\{G_t\}_{t=0}^{T-1}$ e armazena em cada trajetória;

Armazena $\{\tau\}_{n=1}^N$ em EpochBuffer(1);

for $ep \leftarrow 2$ to *MaxEpoch* do

Aplica Gradiente Ascendente sobre π_{ep-1} para obter π_{ep} :

$$\theta_{ep} \leftarrow \theta_{ep-1} + \alpha \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{T-1} G_t \frac{\nabla \pi_{\theta}(\mathbf{s}_t, \mathbf{a}_t)}{\pi_{\theta}(\mathbf{s}_t, \mathbf{a}_t)};$$

Armazena π_{ep} em PolicyBuffer(ep);

Gera N trajetórias $\{\tau_n\}_{n=1}^N$ a partir de política de ações $\pi_{\theta_{ep}}$, onde

$$\tau_n = \mathbf{S}_0, \mathbf{A}_0, R_0, \dots, \mathbf{S}_{T-1}, \mathbf{A}_{T-1}, R_T;$$

Calcula retornos $\{G_t\}_{t=0}^{T-1}$ e armazena em cada trajetória;

Armazena $\{\tau\}_{n=1}^N$ em EpochBuffer(ep);

Mostra performance média da época atual;

Mostra melhor trajetória da época atual;

end

DQN

Algoritmo 2: DQN

- Inicializa Robô, *setpoint*, *obstáculo*, estado inicial \mathbf{s}_0 e espaço de ações \mathcal{A} ;
- Inicializa Hiperparâmetros (bônus e penalidades, tamanho da rede e das imagens, número de *timesteps*, épocas e transições no *Buffer*, fator de desconto γ , taxa de aprendizado α) e ϵ ;
- Inicializa estruturas de armazenamento de épocas e redes DQN;
- Inicializa rede DQN parametrizada Q_{θ_0} aleatória e armazena em DQNBuffer(1);

for $ep \leftarrow 1$ to *MaxEpoch* do

inicializa estado: $\mathbf{s} \leftarrow \mathbf{s}_0$;

Preenche *Buffer* de Experiência com N transições segundo política ϵ -Greedy e rede atual $Q_{\theta_{ep}}$;

Amostra *mini-batch* aleatório de tamanho N_{batch} ;

for $i \leftarrow 1$ to N_{batch} do

Ler i -ésima transição: $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}', bool_{term})$;

if $bool_{term} == true$ then

$y = r$;

else

$y = r + \gamma \max_{\mathbf{a}' \in \mathcal{A}} Q_{\theta_{ep}}(\mathbf{s}', \mathbf{a}')$;

end

Armazenar saída prevista $q = Q(\mathbf{s}, \mathbf{a})$ e alvo y ;

end

Aplica Gradiente Descendente para minimizar função custo dada por

$\mathcal{L}(\theta) = \frac{1}{2} (Q_{\theta}(\mathbf{s}, \mathbf{a}) - y)^2$, ou seja:

$$\theta_{ep+1} \leftarrow \theta_{ep} - \alpha \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \nabla_{\theta} \frac{1}{2} (Q_{\theta}(\mathbf{s}, \mathbf{a}) - y)^2;$$

Armazena rede $Q_{\theta_{ep+1}}$ em DQNBuffer(ep+1);

Limpa *Buffer* de transições;

Mostra trajetória *greedy* atual;

Mostra valor médio de recompensas imediatas;

end

Detalhamento do Projeto

Algoritmo DQN

Inicialização de variáveis

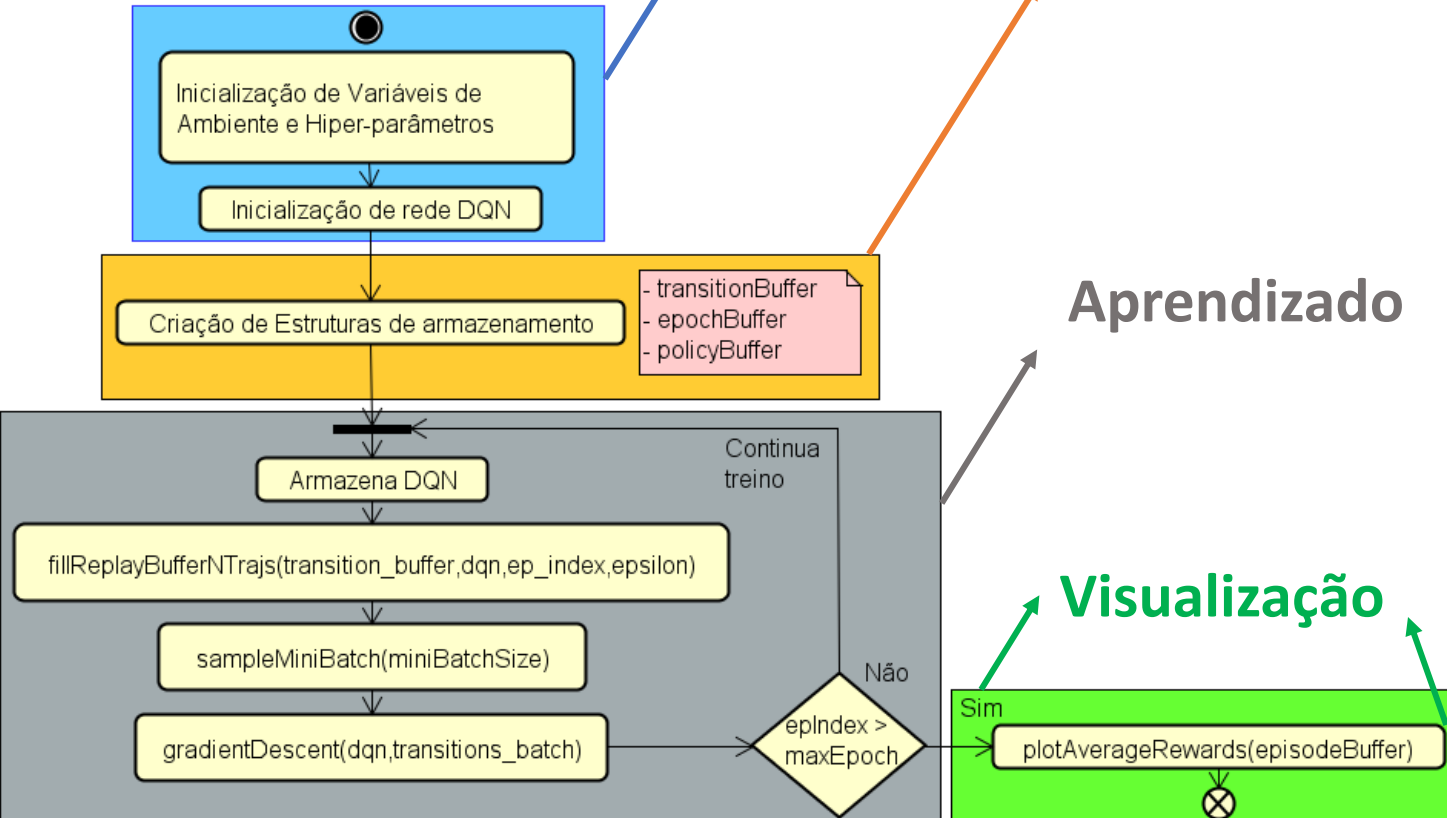
Estruturas de Armazenamento

Aprendizado

Visualização

Algoritmo 2: DQN

```
- Inicializa Robô, setpoint, obstáculo, estado inicial  $s_0$  e espaço de ações  $\mathcal{A}$ ;  
- Inicializa Hiperparâmetros (bônus e penalidades, tamanho da rede e das imagens, número de timesteps, épocas e transições no Buffer, fator de desconto  $\gamma$ , taxa de aprendizado  $\alpha$ ) e  $\epsilon$ ;  
- Inicializa estruturas de armazenamento de épocas e redes DQN;  
- Inicializa rede DQN parametrizada  $Q_{\theta_0}$  aleatória e armazena em DQNBuffer(1);  
for  $ep \leftarrow 1$  to  $MaxEpoch$  do  
  inicializa estado:  $s \leftarrow s_0$ ;  
  Preenche Buffer de Experiência com N transições segundo política  $\epsilon$ -Greedy e rede atual  $Q_{\theta_{ep}}$ ;  
  Amostra mini-batch aleatório de tamanho  $N_{batch}$ ;  
  for  $i \leftarrow 1$  to  $N_{batch}$  do  
    Ler  $i$ -ésima transição:  $(s, a, r, s', bool_{term})$ ;  
    if  $bool_{term} == true$  then  
       $y = r$ ;  
    else  
       $y = r + \gamma \max_{a' \in \mathcal{A}} Q_{\theta_{ep}}(s', a')$ ;  
    end  
    Armazenar saída prevista  $q = Q(s, a)$  e alvo  $y$ ;  
  end  
  Aplica Gradiente Descendente para minimizar função custo dada por  
   $\mathcal{L}(\theta) = \frac{1}{2} (Q_{\theta}(s, a) - y)^2$ , ou seja:  
   $\theta_{ep+1} \leftarrow \theta_{ep} - \alpha \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \nabla_{\theta} \frac{1}{2} (Q_{\theta}(s, a) - y)^2$ ;  
  Armazena rede  $Q_{\theta_{ep+1}}$  em DQNBuffer( $ep+1$ );  
  Limpa Buffer de transições;  
  Mostra trajetória greedy atual;  
  Mostra valor médio de recompensas imediatas;  
end
```

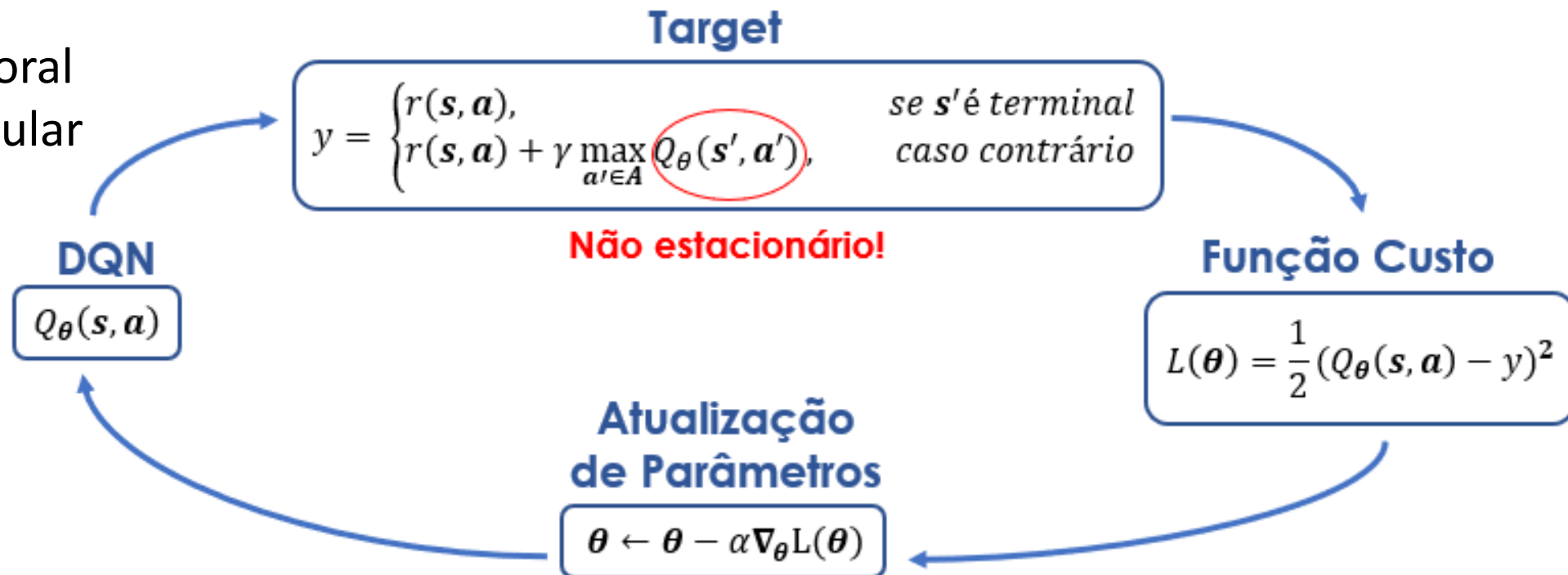


Detalhamento do Projeto

Principais Problemas

Alvo Não Estacionário

- Correlação temporal
- Treinamento Circular

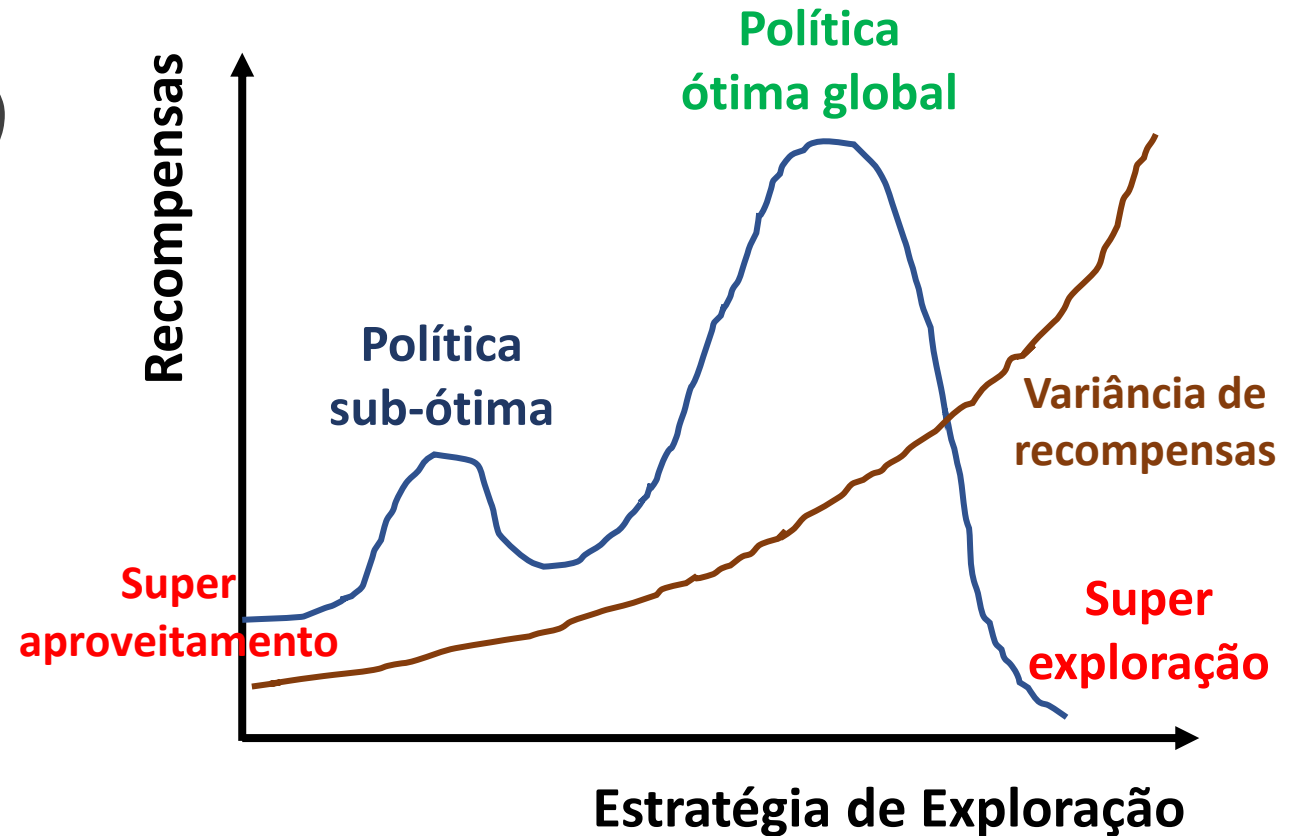


Detalhamento do Projeto

Principais Problemas

Explorar x Aproveitar (*Exploration vs Exploitation trade-off*)

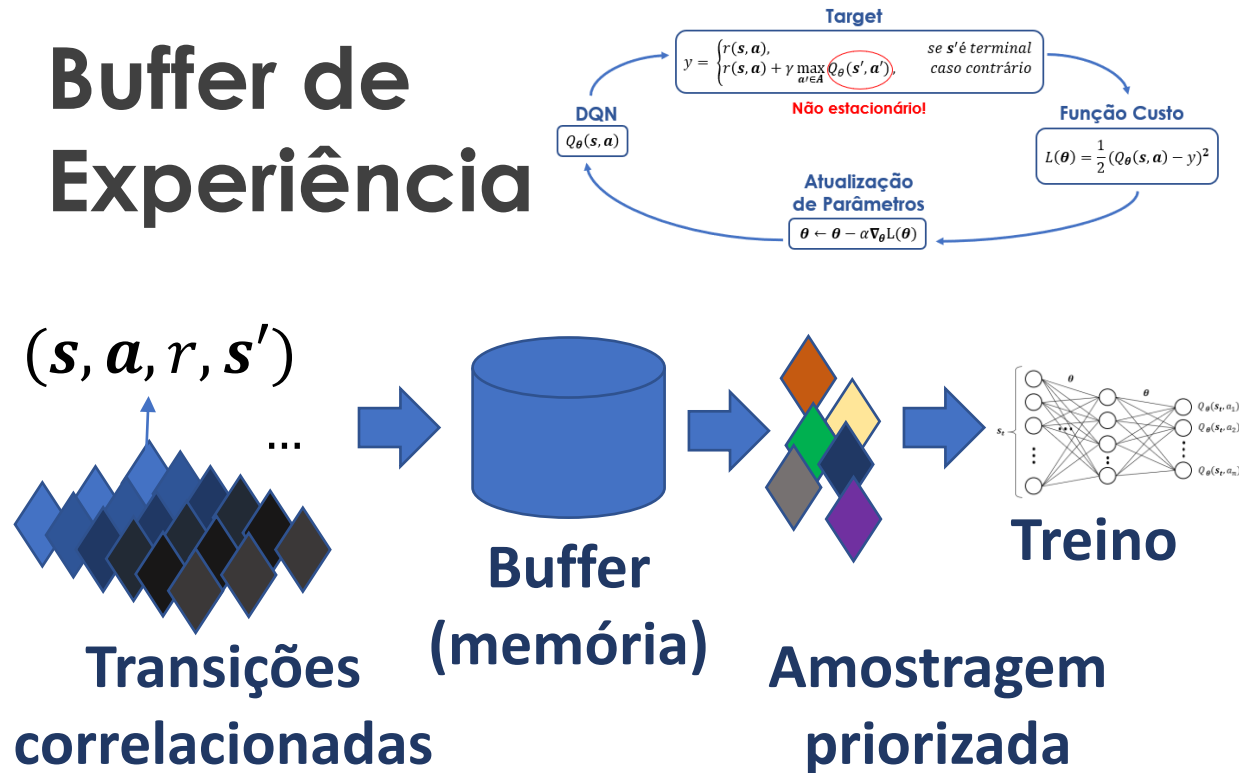
- Convergência para políticas sub-ótimas



Detalhamento do Projeto

Técnicas de Melhoria de Convergência

Buffer de Experiência



Política ϵ – Greedy

ϵ : Probabilidade de tomar ação aleatória

$$a_t = \begin{cases} \operatorname{argmax}_{a' \in A} Q_\pi(s_t, a'), & \text{se } \operatorname{rand}(0,1) > \epsilon \\ \text{ação } a \in A \text{ aleatória,} & \text{caso contrário} \end{cases}$$



Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão











Resultados

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



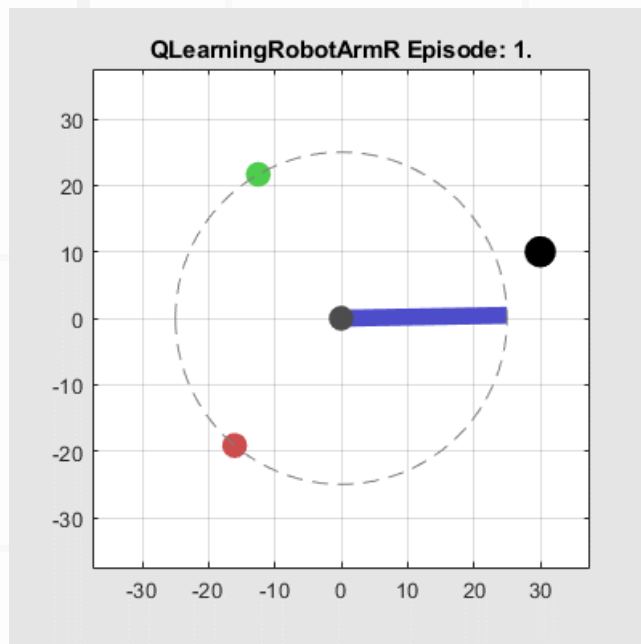
Resultados Parciais

1 Grau de Liberdade (R)

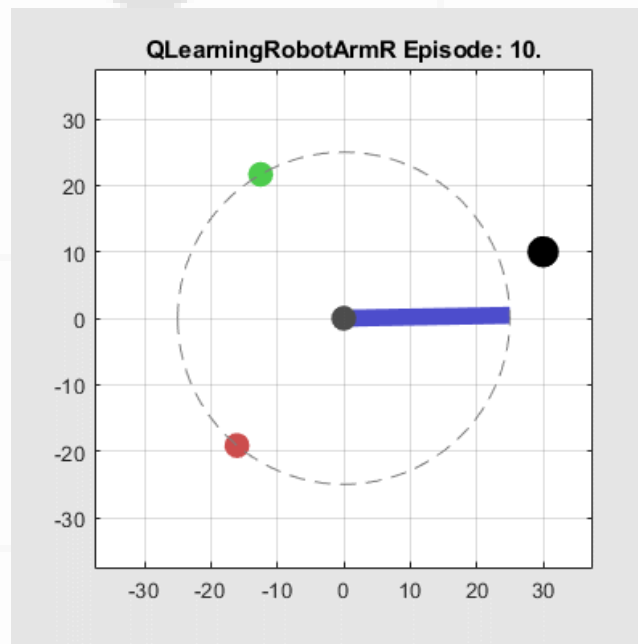
Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
	DRL	DQN
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

Q-Learning: Trajetórias ao longo do treino com r_2

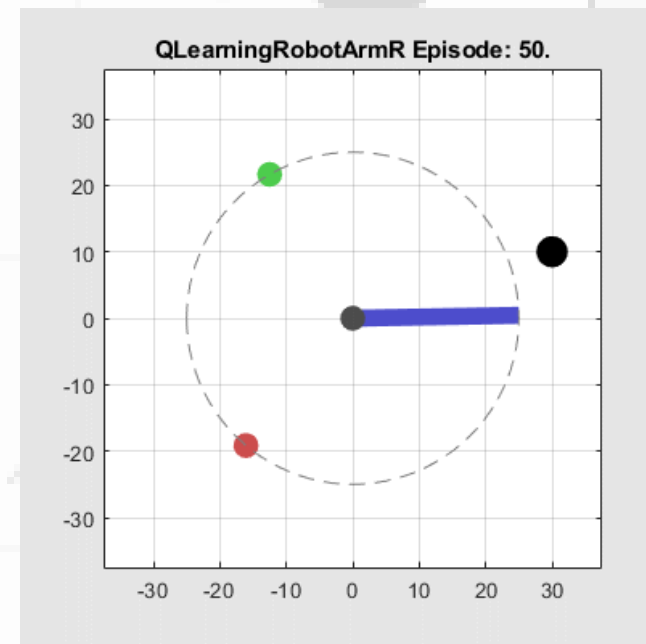
Época 1



Época 10











Época 50



Resultados Parciais

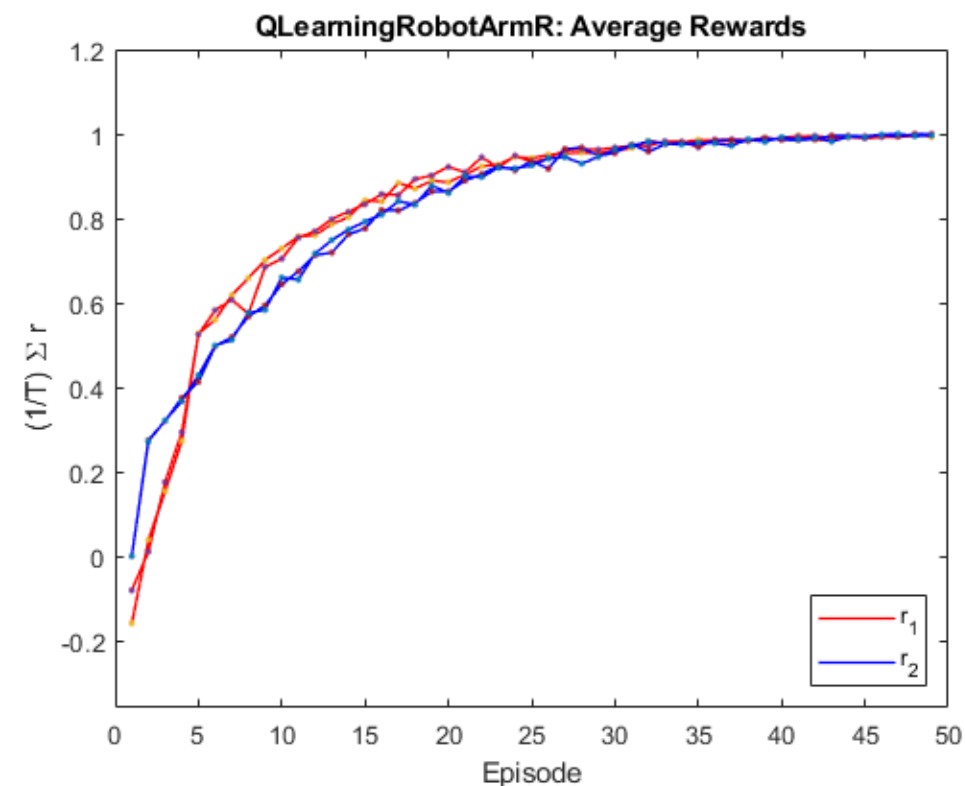
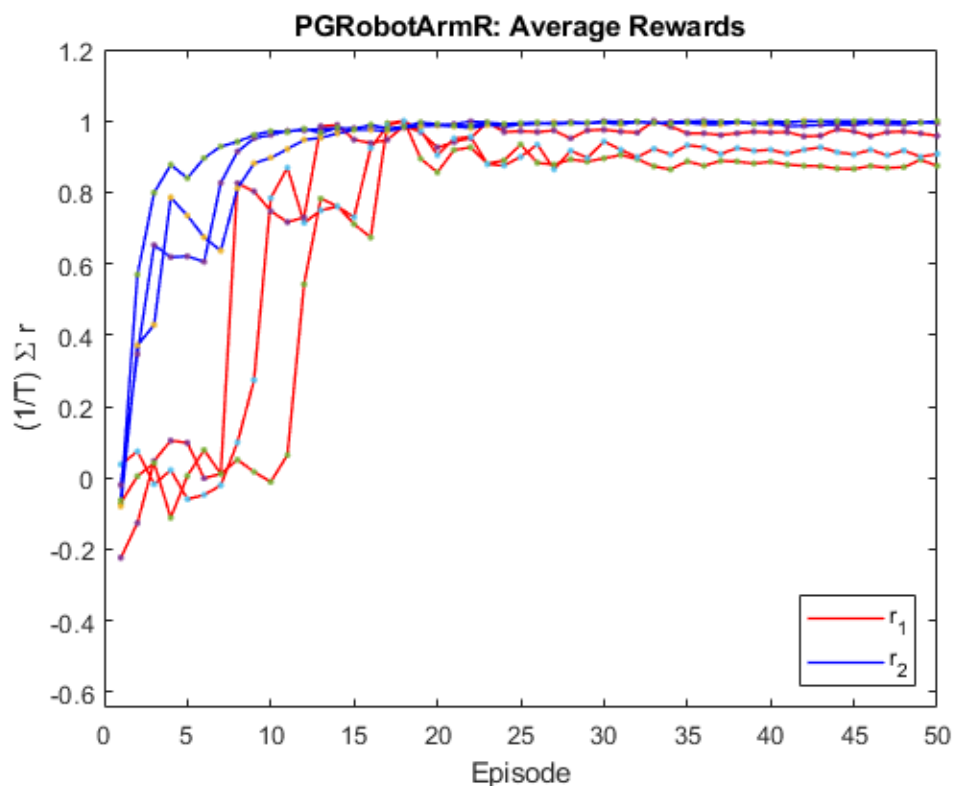
1 Grau de Liberdade (R)

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

Comparação entre funções de recompensa r_1 e r_2









REINFORCE

DQN



Resultados Parciais

2 Graus de Liberdade (RR)

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
	DRL	DCN
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

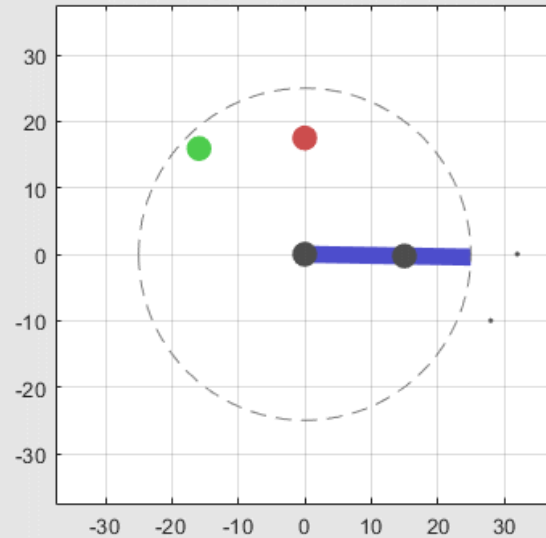
Q-Learning: Trajetórias ao longo do treino com r_3

Época 1

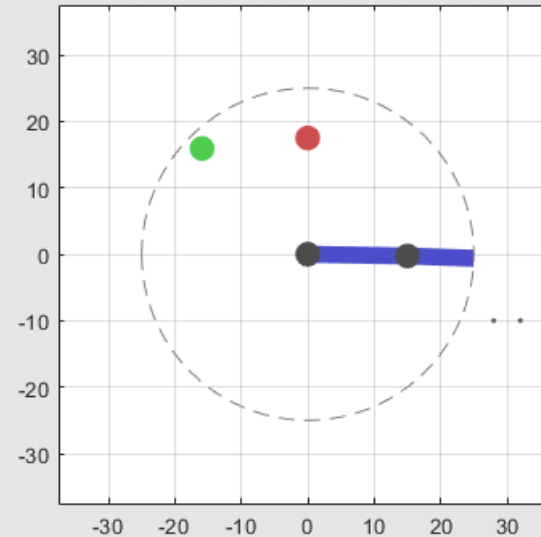
Época 10

Época 50

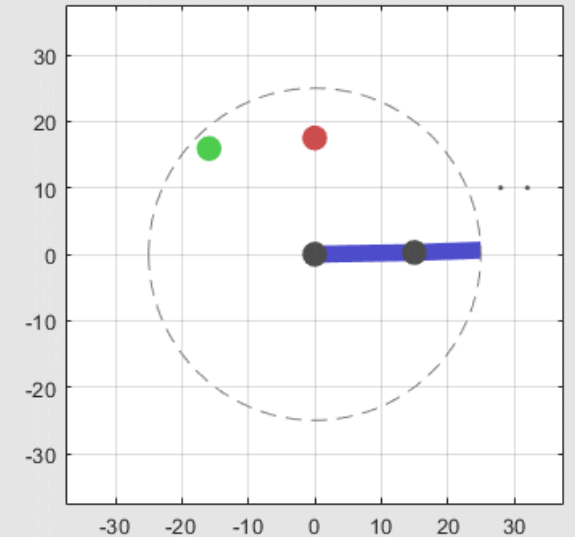
QLearningRobotAmRR: Episode 1.



QLearningRobotAmRR: Episode 10.



QLearningRobotAmRR: Episode 50.



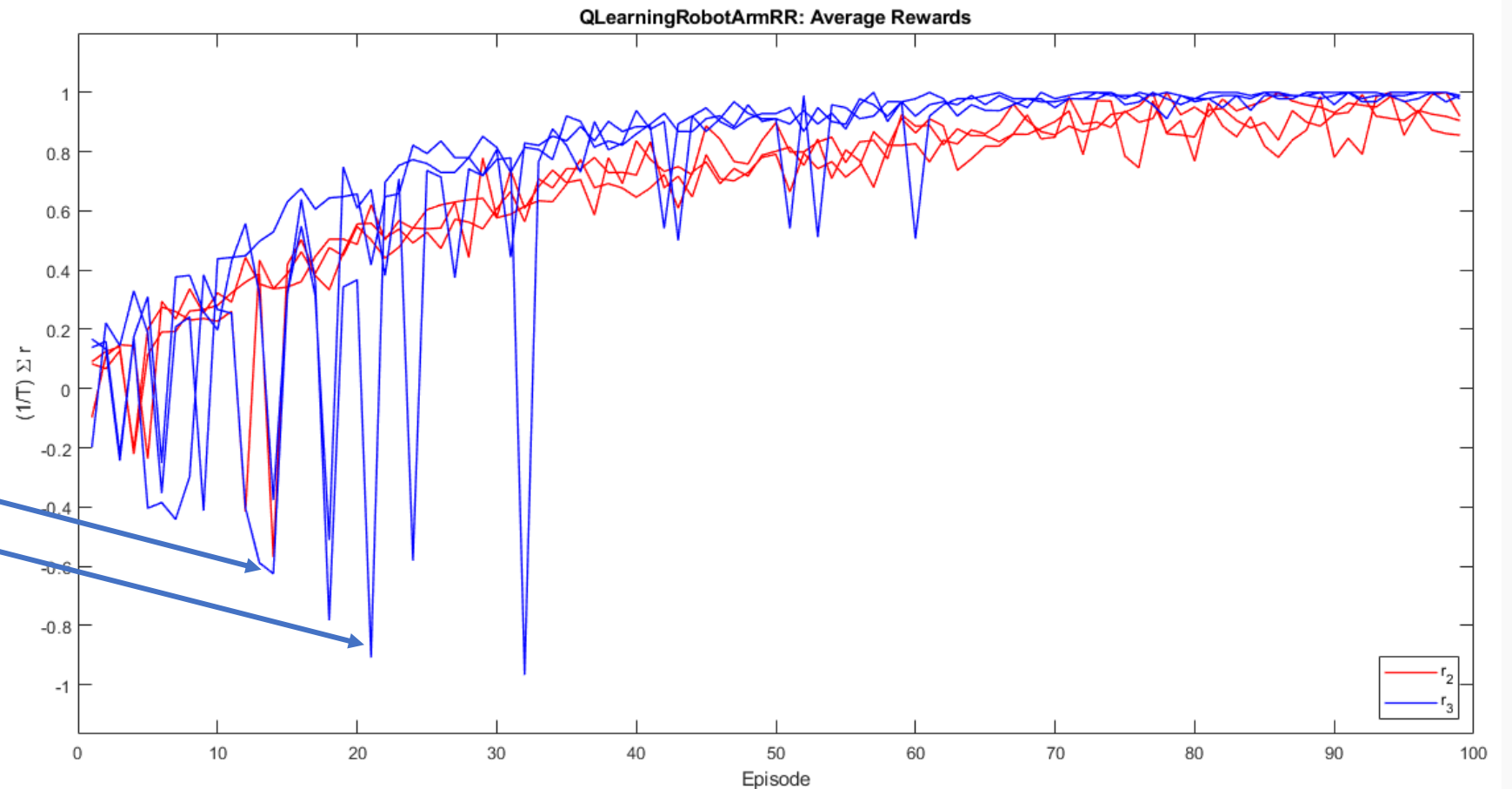
Resultados Parciais

2 Graus de Liberdade (RR)

Q-Learning: Comparação entre funções de recompensa r_2 e r_3

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		









Colisões com obstáculo

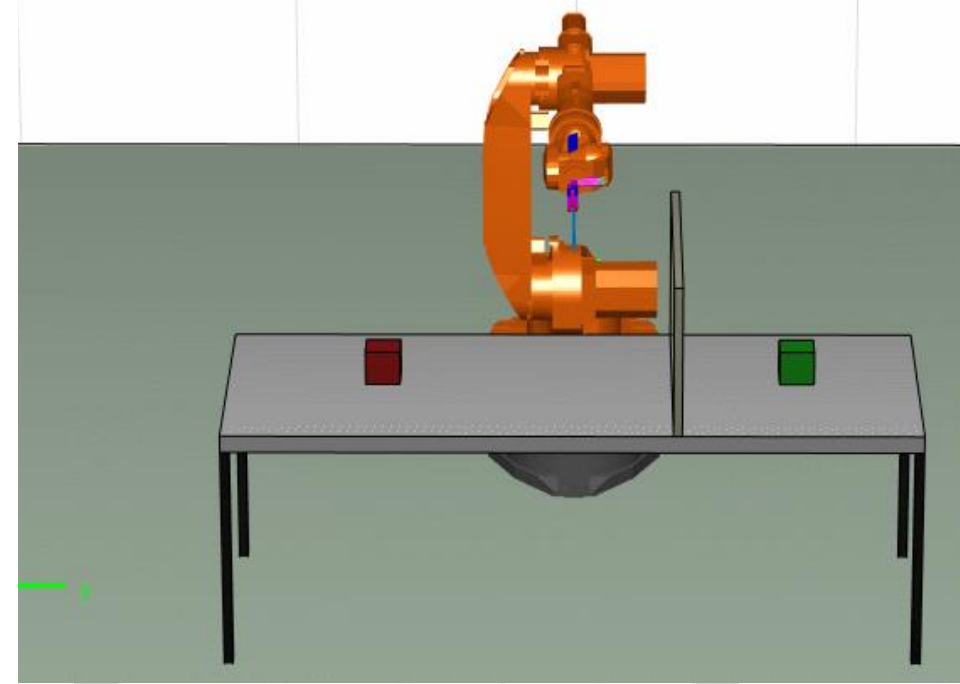
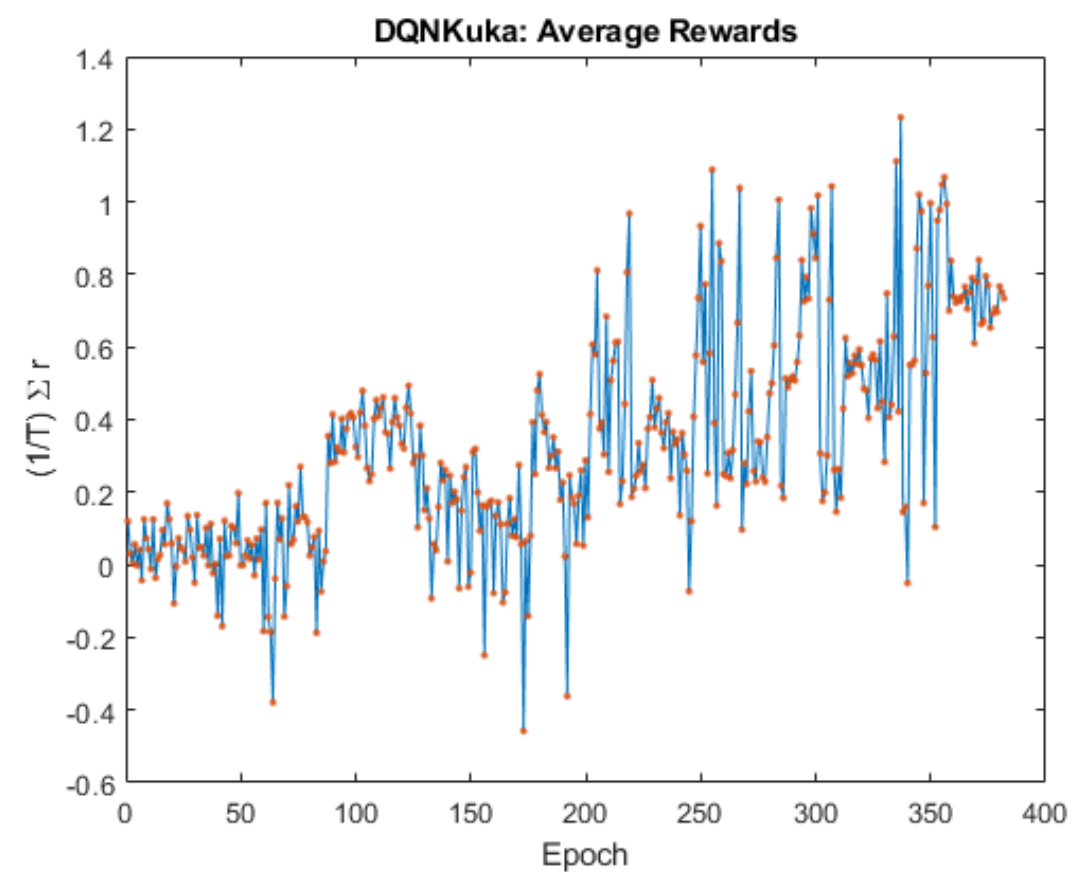


Resultados Parciais

6 Graus de Liberdade (6R)

DQN: Obstáculo Desconhecido

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		











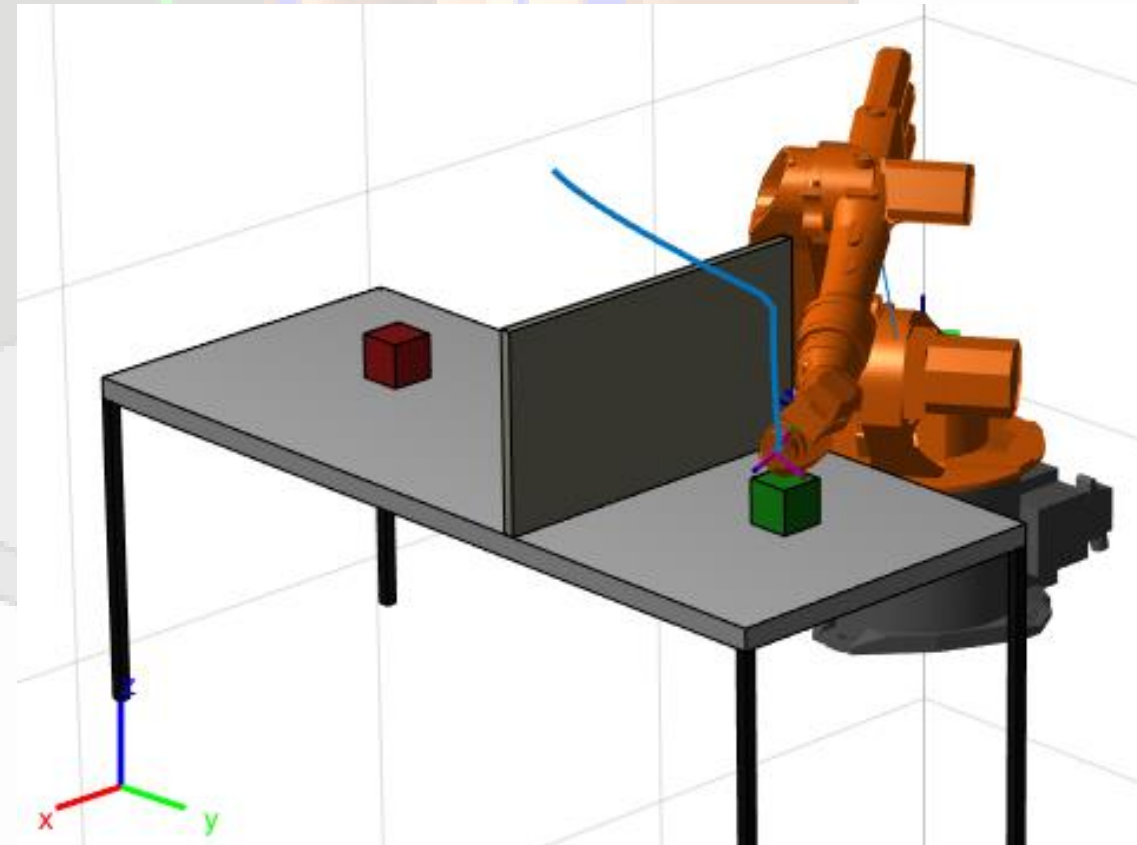
Resultados Parciais

6 Graus de Liberdade (6R)

DQN: Obstáculo Desconhecido









Trajetoória obtida por agente treinado

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
	DRL	DQN
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		



Resultados Parciais

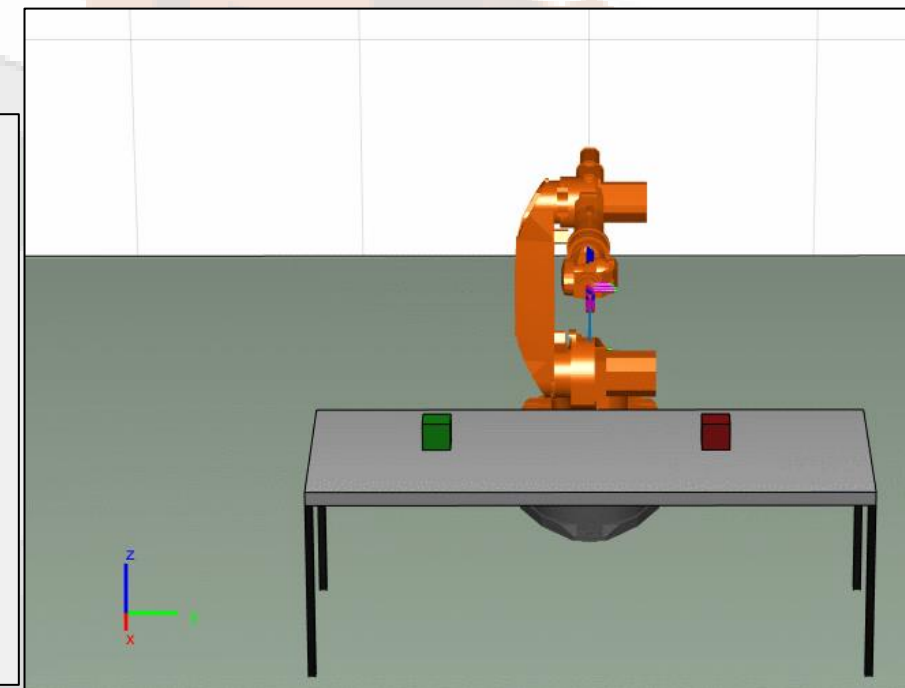
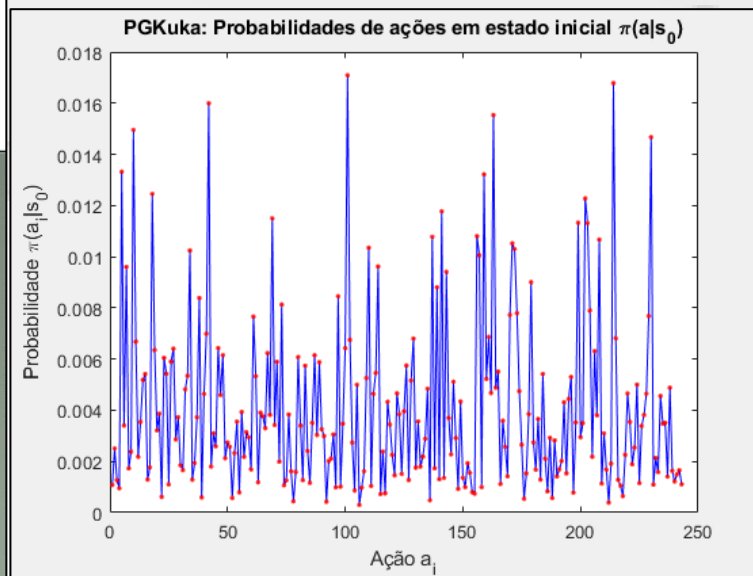
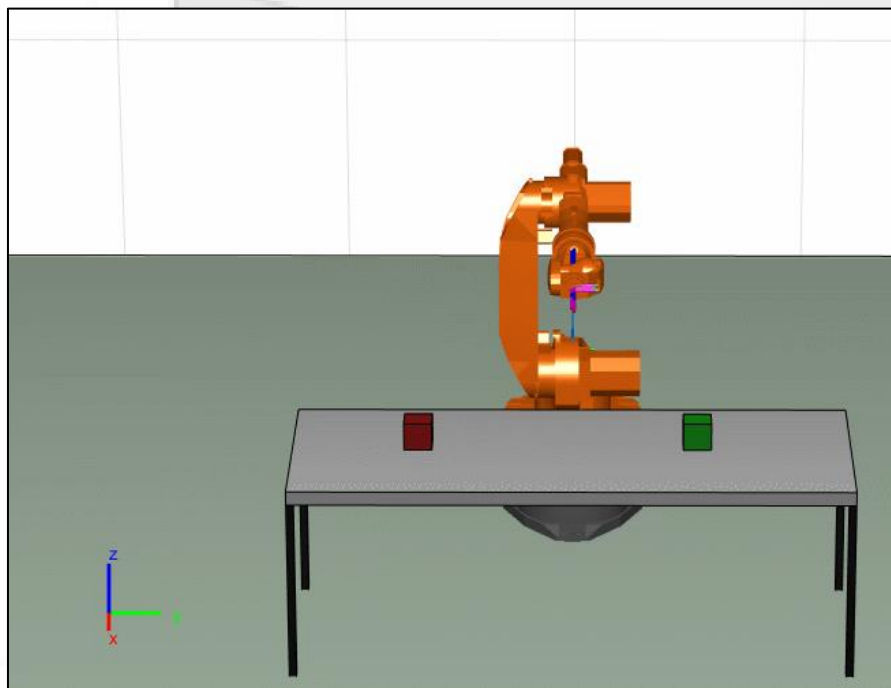
6 Graus de Liberdade (6R)

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

REINFORCE: Retreinamento de Agente

1º treino

2º treino



Resultados Parciais

6 Graus de Liberdade (6R)

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

REINFORCE: Retreinamento de Agente

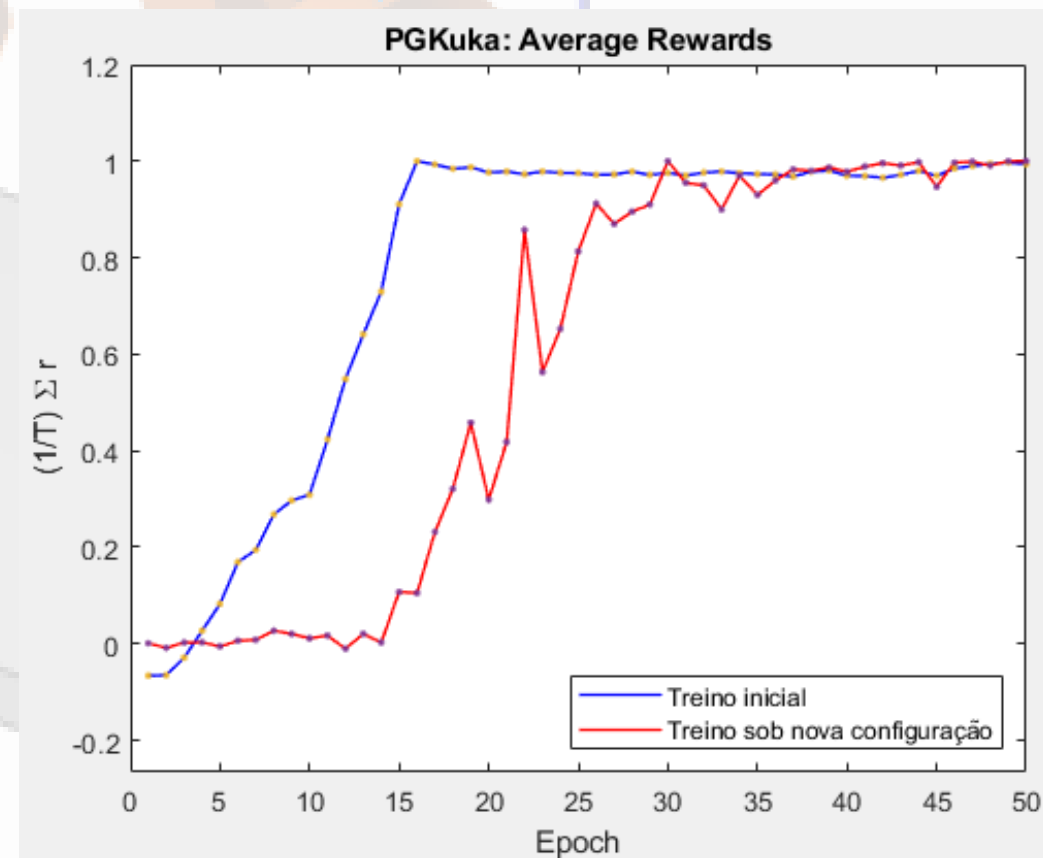
Treino Inicial

- Política inicial π_{θ_0} aleatória
- Convergência mais rápida

Retreinamento

- Política inicial π_{θ_0} sub-ótima
- Convergência mais lenta
- Perda de aprendizado anterior

Solução: Treinar simultaneamente para diversas configurações

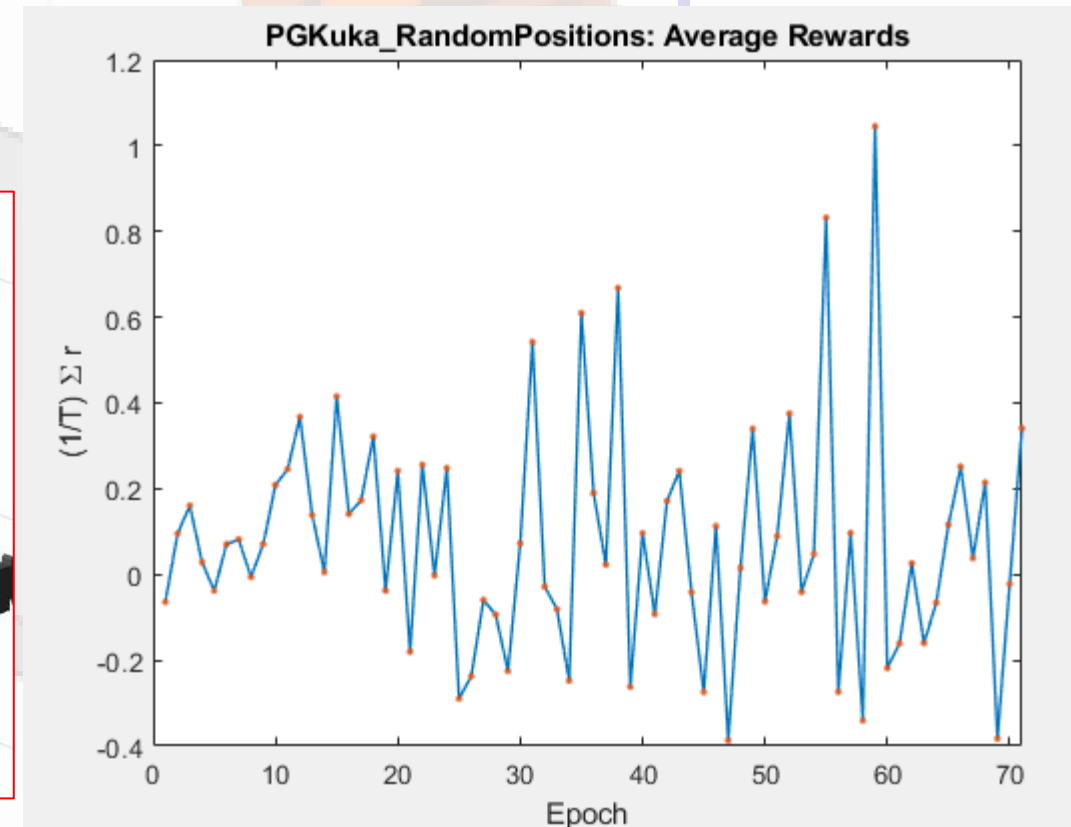
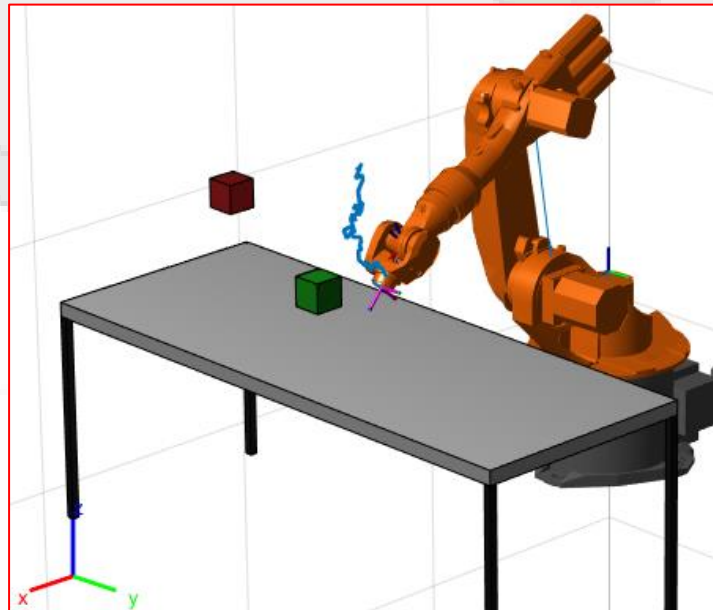
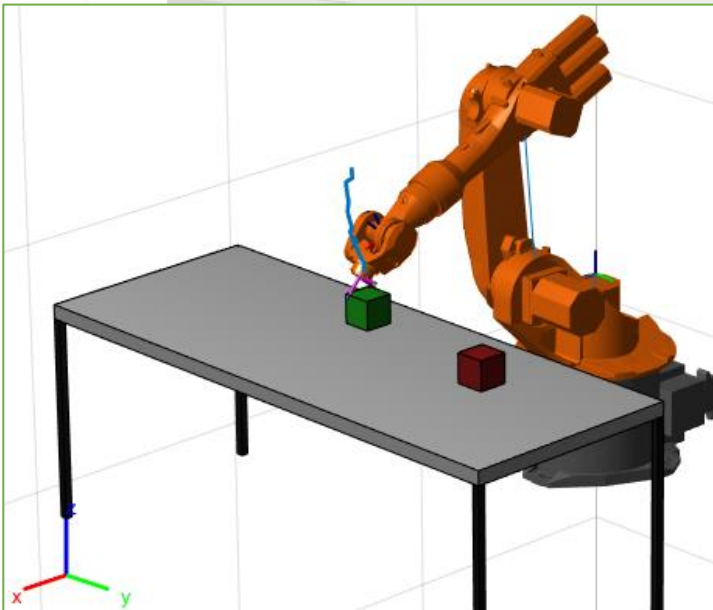


Resultados Parciais

6 Graus de Liberdade (6R) e Configurações Genéricas

REINFORCE: Não foi observada convergência









Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		

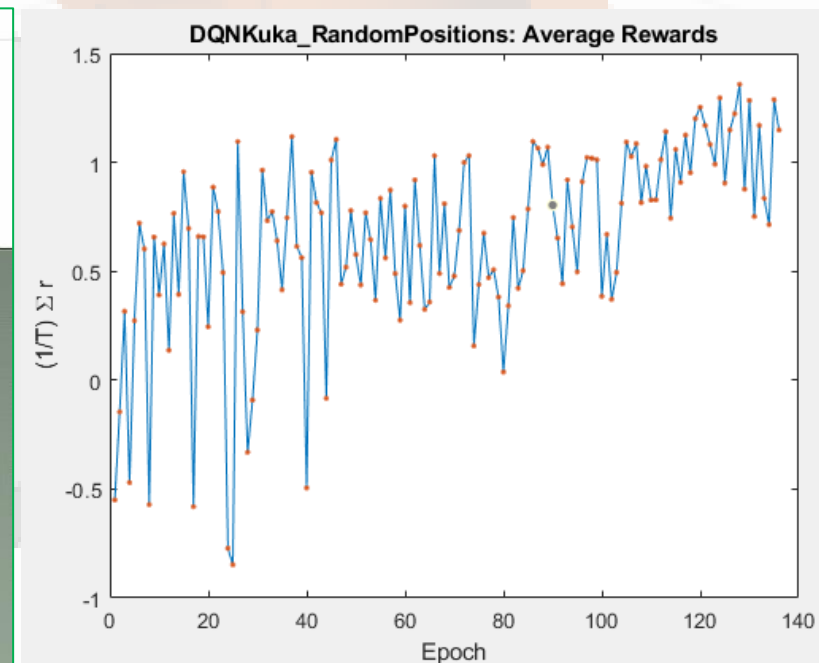
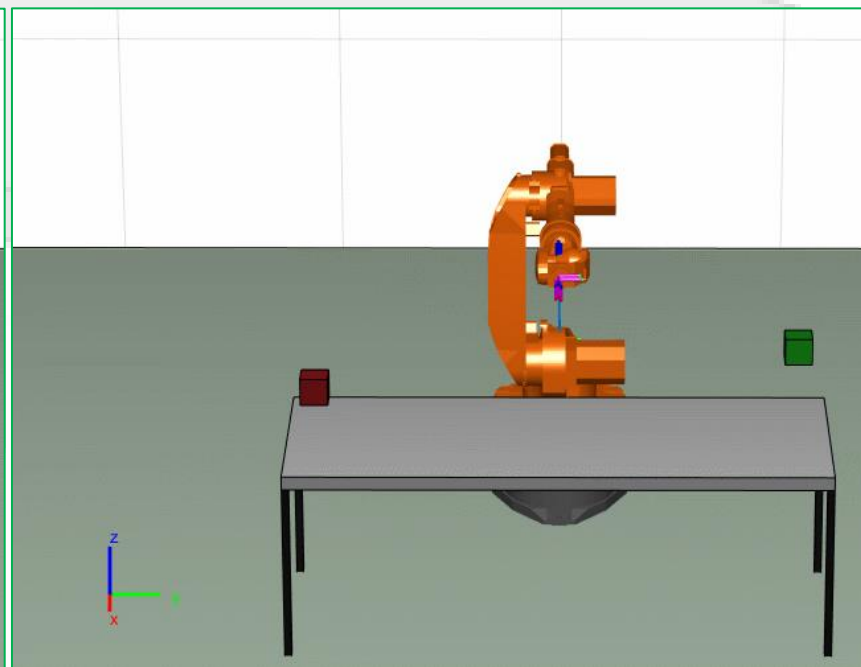
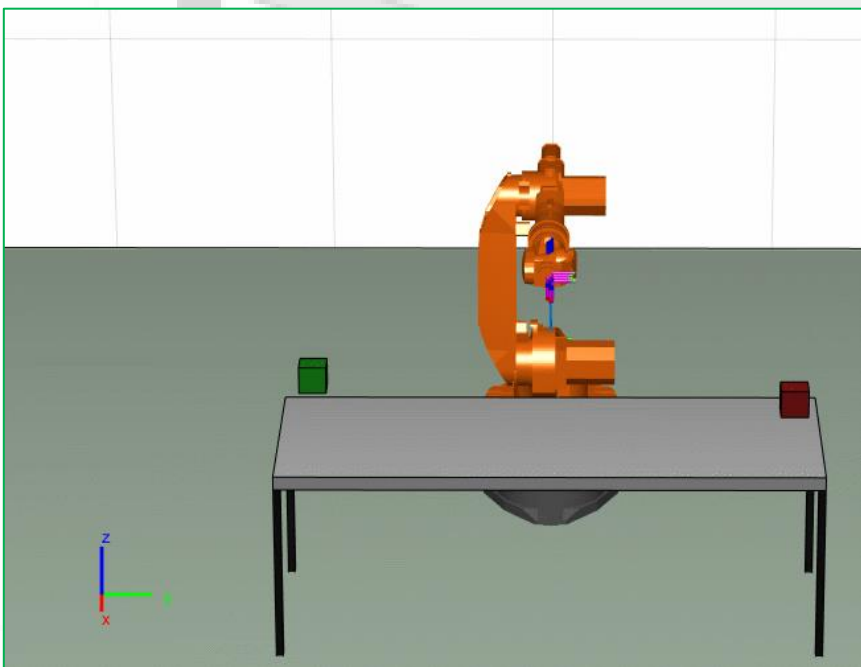


Resultados Parciais

6 Graus de Liberdade (6R) e Configurações Genéricas

DQN: Convergência para configurações específicas

Projetos de Teste	REINFORCE	Q-Learning
1 Grau de Liberdade (R)		
2 Graus de Liberdade (RR)		
6 Graus de Liberdade (6R), configuração fixa		
6 Graus de Liberdade (6R), configurações aleatórias		



Resultados: Parâmetros da Solução Final

Algoritmo
Escolhido: DQN

Função de
Recompensa: r_3

Estrutura da rede
 $Q_\theta(s, a)$

Hiper-parâmetros:
 $\gamma, \epsilon, \alpha, MiniBatchSize, B_{goal},$
 $P_{collision}, k_s, k_o, \dots$

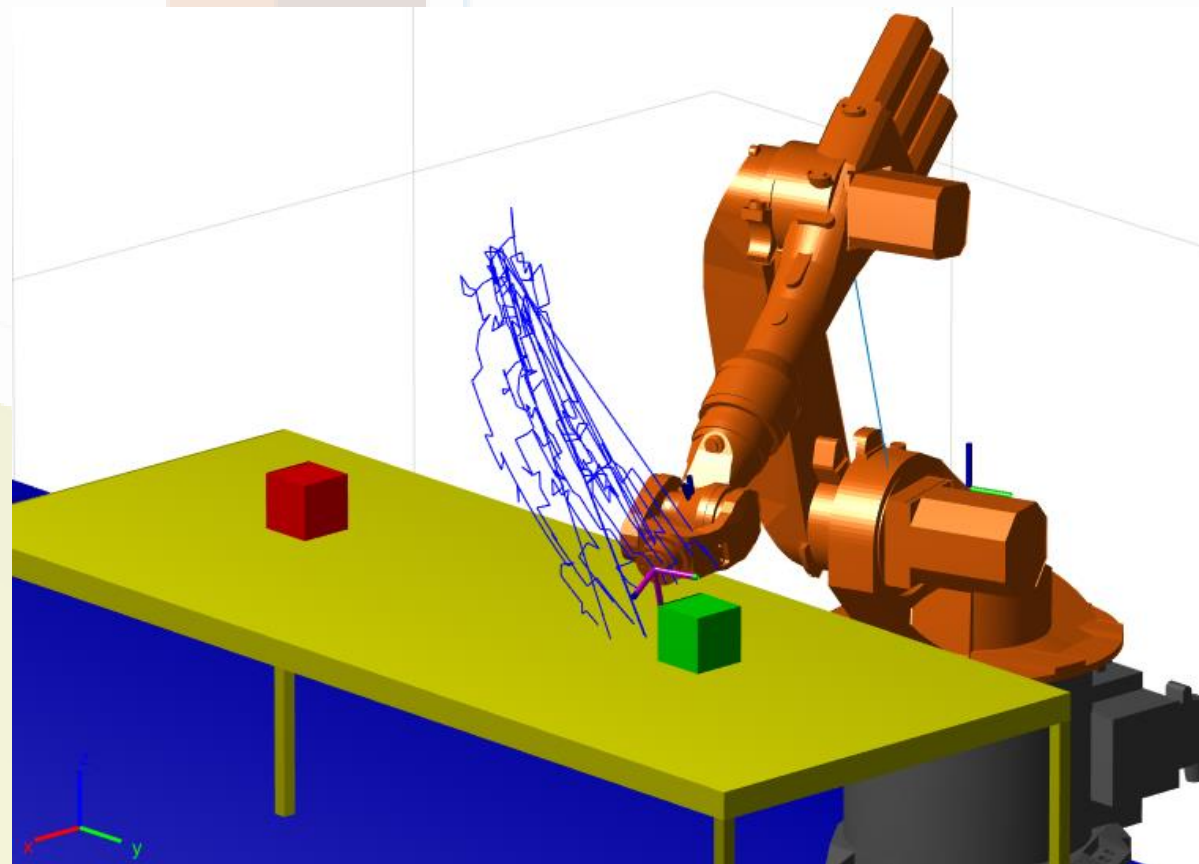
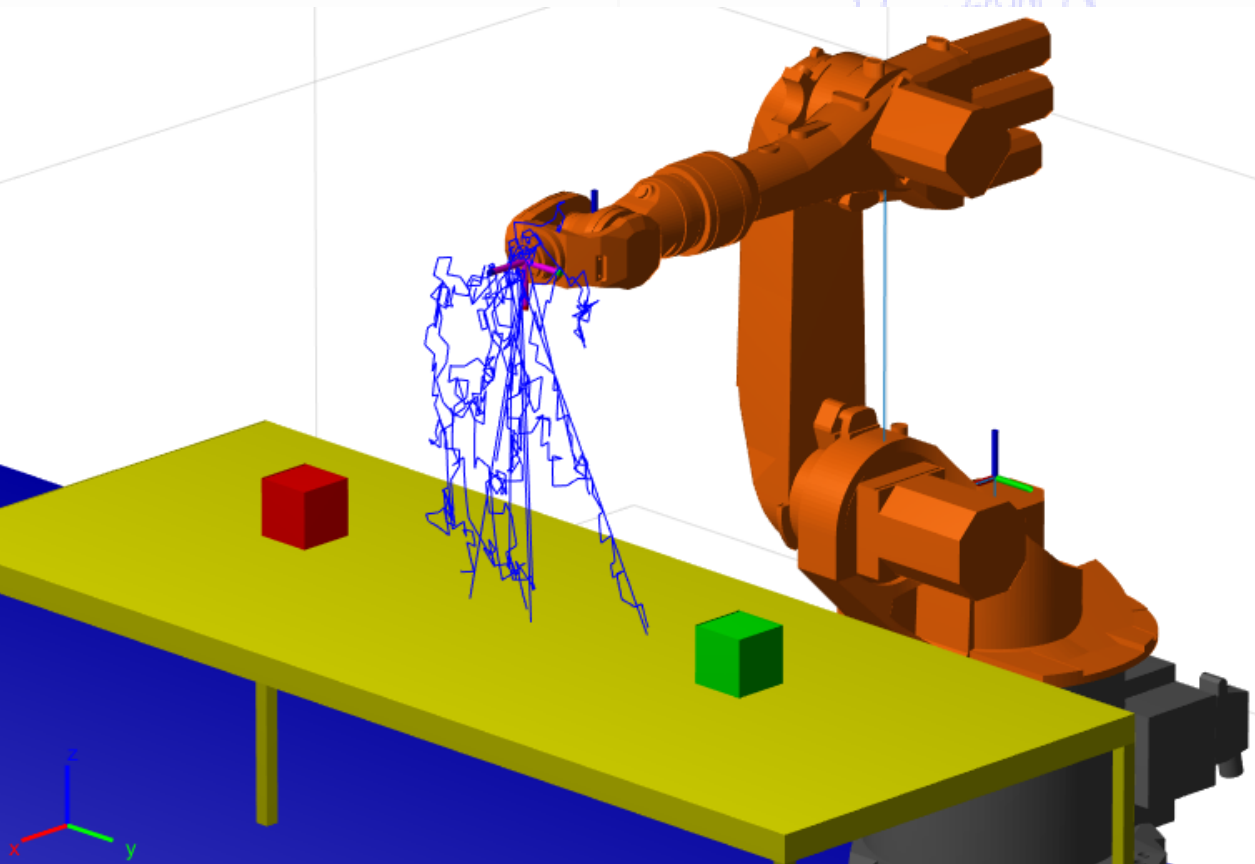
Parâmetro	Descrição	Valor
$P_{setpoint}$	Posição de destino (m)	(1.05, 0.45, 0.75)
$P_{obstacle}$	Posição de obstáculo (m)	(1.05, -0.55, 0.75)
$table_{length}$	Comprimento da mesa (m)	2
$table_{width}$	Largura da mesa (m)	0.8
$table_{height}$	Altura da mesa (m)	0.7
$\Delta\theta$	Mínima variação angular	1°
α	Taxa de Aprendizado	0.005
B_{goal}	Bônus de destino	20
$P_{collision}$	Penalidade de colisão	-20
$P_{joint\ boundary}$	Penalidade de fim de curso	-10
r_{infl}	Raio de influência do obstáculo (m)	0.50
$MaxEpoch$	Número máximo de épocas de treino	300
N_{trajs}	Número de trajetórias por época	10
$MiniBatchSize$	Número de transições amostradas para treino	200
T	Número máximo de transições por trajetória	70
γ	Fator de desconto	0.3
$dim(s)$	Dimensão de cada estado s	3456
$dim(a)$	Dimensão de cada ação a	5
$size(\mathcal{A})$	Tamanho do espaço de ações	243
$(n_{in}, n_{h1}, n_{h2}, n_{out})$	Dimensões de camadas da rede $\pi_\theta(a s)$	(3456, 400, 300, 243)
k_s	Fator multiplicativo de r_3	2
k_o	Fator multiplicativo de r_3	1

Resultados

Exploração ao longo do treino

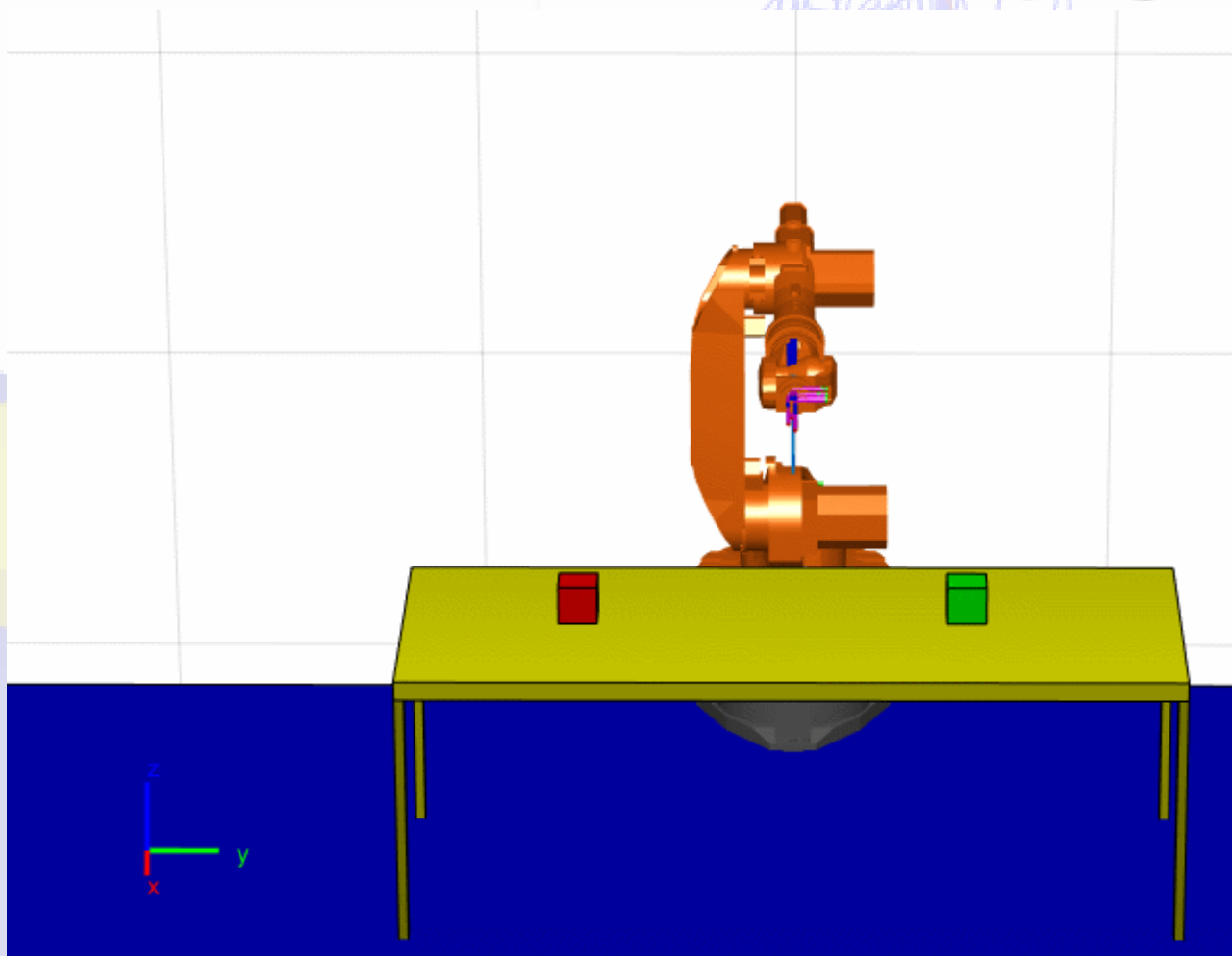
Época 1

Época 30

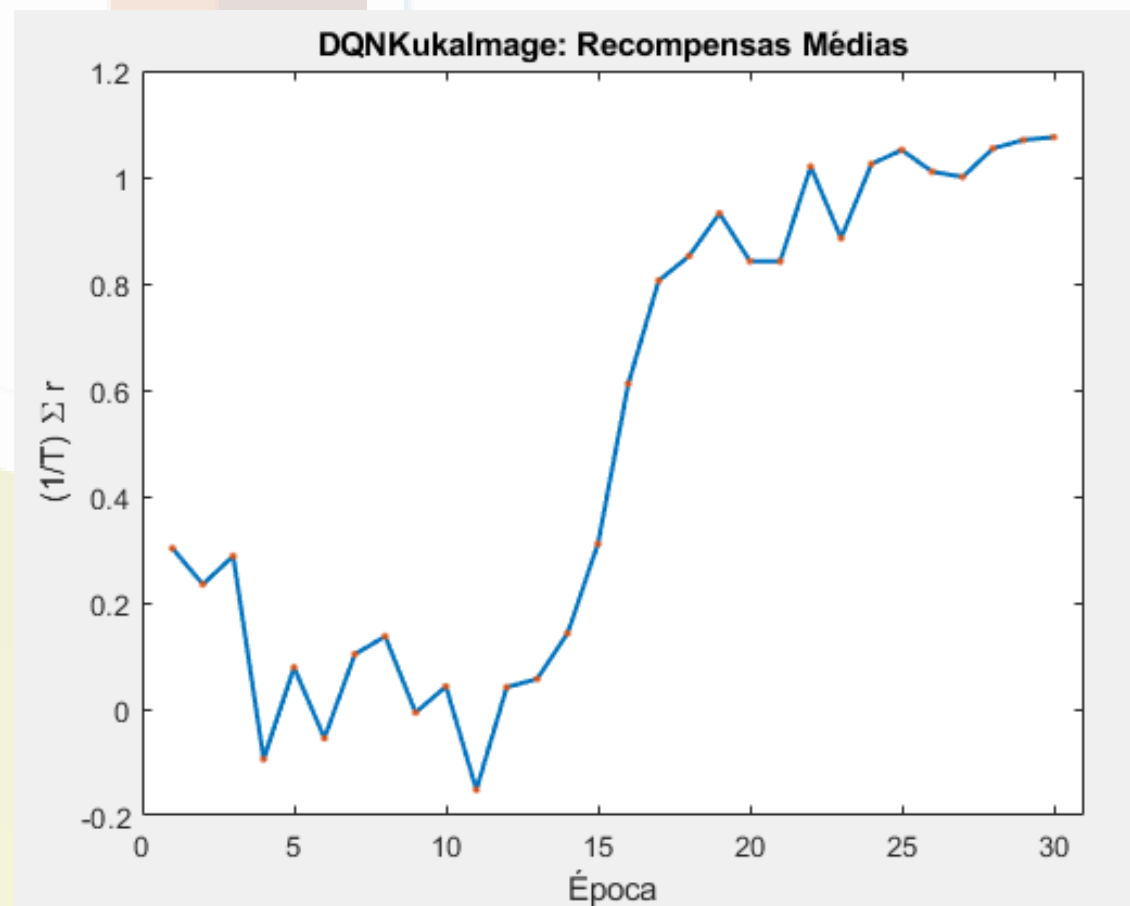


Resultados

Trajectoria Obtida



Curva de Aprendizizado



Resultados

Tempos de Treino


Projeto	REINFORCE	DQN
1 Grau de Liberdade (R)	6,8 min	6,9 min
2 Graus de Liberdade (RR)	11,7 min	7,4 min
6 Graus de Liberdade (Configuração Fixa)	30 h	16 h
6 Graus de Liberdade (Configuração Aleatória)	----	25 h
Projeto Final (Imagem)	----	70 h

Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



Conclusão

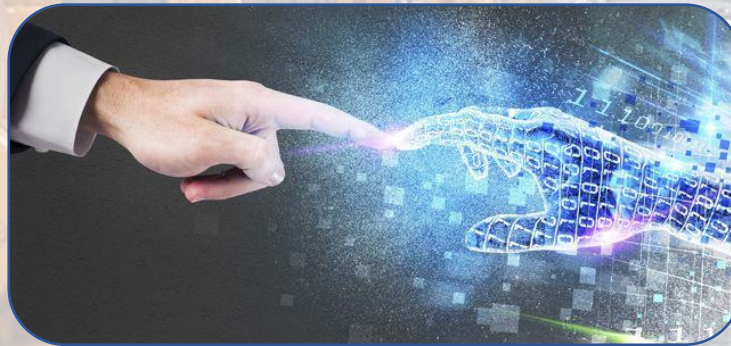
1. Introdução
 2. Estado da Arte
 3. Fundamentos Teóricos
 4. Detalhamento do Projeto
 5. Resultados
 6. Conclusão
- 

Conclusão

Principais Desafios de RL na Robótica



**Eficiência
Amostrai**



**Transferência de
Aprendizado**



**Especificação
de Função
Recompensa**



Segurança

Sumário

1. Introdução
 2. Estado da Arte
 3. Fundamentos Teóricos
 4. Detalhamento do Projeto
 5. Resultados
 6. Conclusão
- Trabalhos Futuros

Sumário

1. Introdução
2. Estado da Arte
3. Fundamentos Teóricos
4. Detalhamento do Projeto
5. Resultados
6. Conclusão



Trabalhos Futuros

Trabalhos Futuros

Aprimoramento do Algoritmo

Actor-Critic

- Parametrização e aproximação de funções

$\pi_{\theta}(s|a)$ e $Q_{\theta}(s, a)$
ator **crítico**



- Função Vantagem: $A(s, a) = Q(s, a) - V(s)$

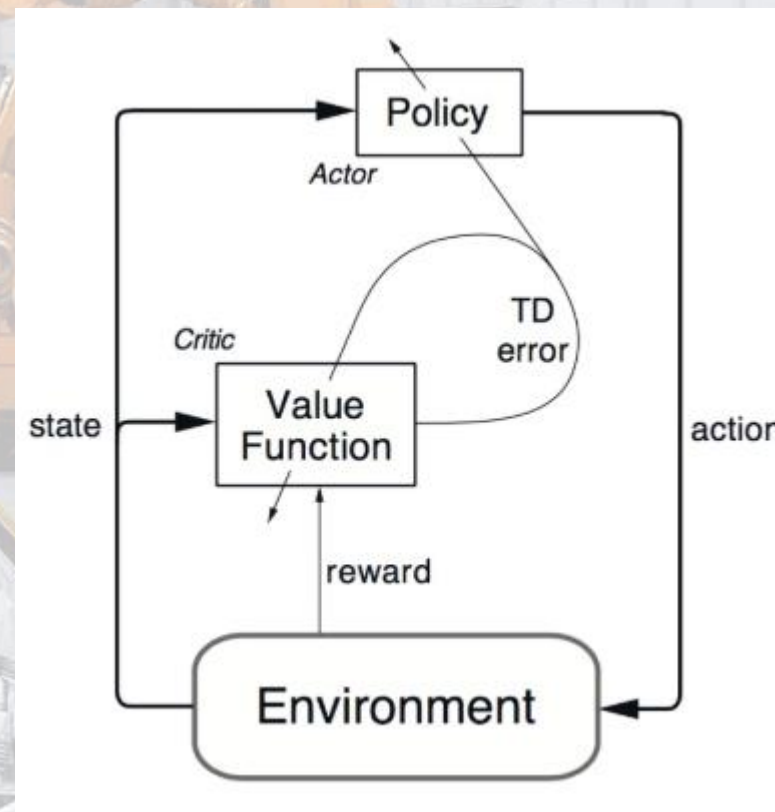
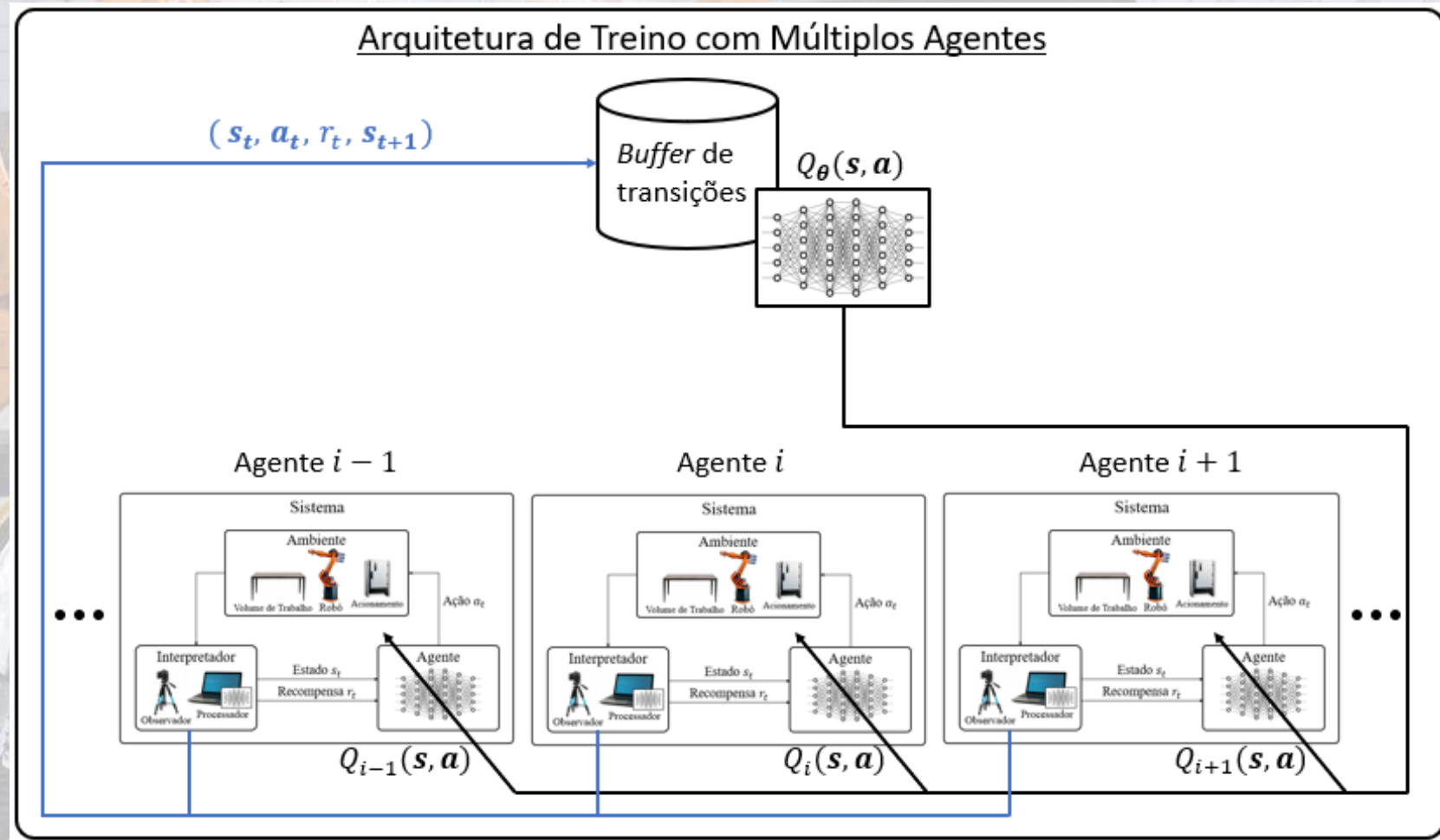


Diagrama Esquemático de algoritmo Actor-Critic. Fonte: SUTTON, R, S; BARTO, A, G, 2017

Trabalhos Futuros

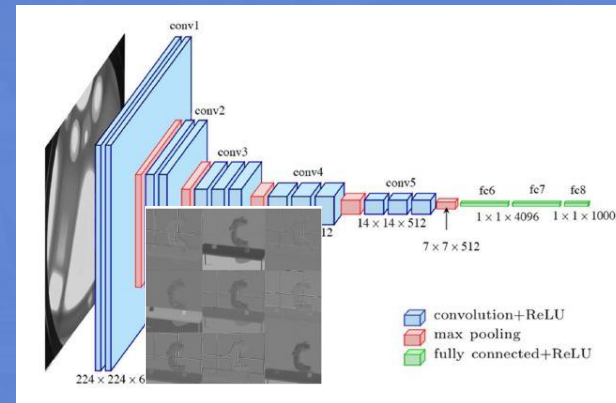
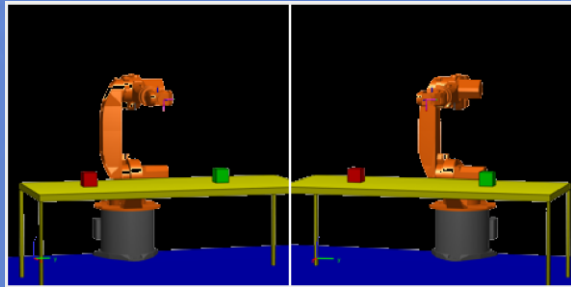
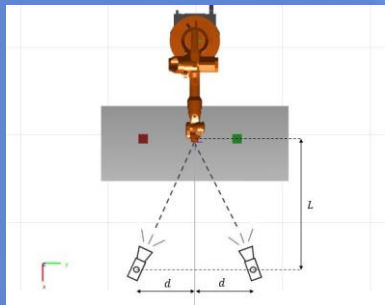
Arquitetura Multi-Agente

- Melhor exploração dos espaços de Estados e Ações
- Redução do tempo de treino
- Computação distribuída



Trabalhos Futuros

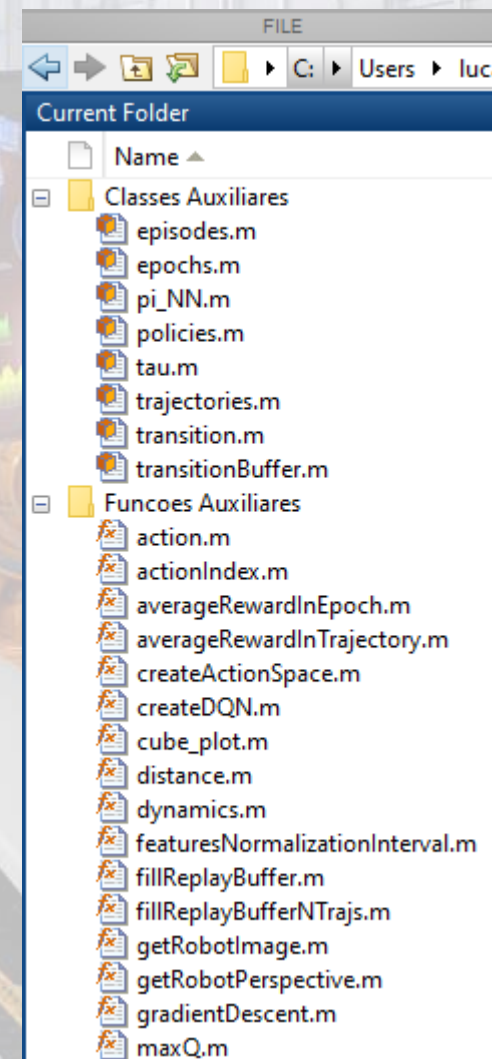
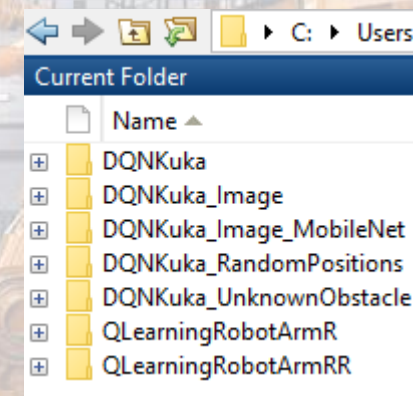
Implementação de rede convolucional



Conclusão e Trabalhos Futuros

Contribuições Deste Projeto

- Ambiente de implementação, teste e visualização de algoritmos de RL em robótica
- Análise comparativa de algoritmos REINFORCE e DQN
- *Open Source:*
https://github.com/MMenonJ/Controle_Robo_Manipulador



Referências Bibliográficas

FRANCESCHETTI, A et al. **Robotic Arm Control and Task Training through Deep Reinforcement Learning**, Intelligent Autonomous Systems Lab, University of Padova, 2017.

JAMES, S; JOHNS, E. **3D Simulation for Robot Arm Control with Deep Q-Learning**, Imperial College London, UK, 2016.

MNIH, V et al. **Playing Atari With Deep Reinforcement Learning**, Google DeepMind, 2013

SUTTON, R. S.; BARTO A. G. **Reinforcement Learning: An Introduction**. 2nd Edition. The MIT Press, 2017.

ROS-Industrial, Github, 2016. Disponível em < https://github.com/ros-industrial/kuka_experimental >. Acesso em 15 de Maio de 2019

Agradecimentos

Muito Obrigado!

