



Fábio Jessen Werneck de Almeida Martins

**Métodos de Aceleração de Aprendizado
Aplicado ao Modelo Neuro-Fuzzy Hierárquico
Politree com Aprendizado por Reforço**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da PUC-Rio como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientadora: Prof.^a Marley Vellasco

Co-orientadora: Prof.^a Karla Figueiredo

Rio de Janeiro,
Julho de 2010



Fábio Jessen Werneck de Almeida Martins

**Métodos de Aceleração de Aprendizado
Aplicado ao Modelo Neuro-Fuzzy Hierarquico
Politree com Aprendizado por Reforço**

Dissertação de Mestrado apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Profa. Marley Maria Bernardes Rebuszi Vellasco
Orientadora

Departamento de Engenharia Elétrica - PUC-Rio

Profa. Karla Tereza Figueiredo Leite
Co-Orientadora

Departamento de Engenharia Elétrica - PUC-Rio

Prof. Ricardo Tanscheit
Departamento de Engenharia Elétrica - PUC-Rio

Prof. Flávio Joaquim de Souza
UERJ

**Prof. Reinaldo Augusto da Costa
Bianchi**
Centro Universitário da FEI

**Prof. Adriano Joaquim de Oliveira
Cruz**
NCE

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico - PUC-Rio

Rio de Janeiro, 12 de julho de 2010

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e da orientadora.

Fábio Jessen Werneck de Almeida Martins

Graduou-se em Engenharia Mecânica na Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2007. Graduou-se em Engenharia de Controle e Automação na PUC-Rio.

Ficha Catalográfica

Martins, Fábio Jessen Werneck de Almeida

Métodos de aceleração de aprendizagem aplicado ao modelo neuro-fuzzy hierárquico politree com aprendizado por reforço / Fábio Jessen Werneck de Almeida Martins orientadora: Marley Vellasco ; co-orientadora: Karla Figueiredo. – 2010.

157 f. : il. ; 30 cm

Dissertação (mestrado)—Pontifícia Universidade Católica do Rio de Janeiro, 2010.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Agente inteligente. 3. Neuro-fuzzy. 4. Aprendizado por reforço. 5. Aprendizado automático. 6. Modelo hierárquico politree. I. Vellasco, Marley. II. Figueiredo, Karla. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Para meus pais, Manuel e Magali,
pelo apoio e confiança.

Agradecimentos

Às minhas orientadoras Professora Marley Vellasco e Professora Karla Figueiredo pelo ensinamentos, estímulo e parceria para a realização deste trabalho.

Ao CNPq, Faperj e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus pais, pela educação, atenção e carinho.

À minha madrinha, com seus 94 anos de otimismo, por suas palavras de apoio.

Ao meu irmão, pela companhia de todas as horas.

Aos meus amigos e amigas por todo apoio, paciência e compreensão.

Aos meus colegas da PUC-Rio.

Aos professores que participaram da Comissão examinadora.

A todos os professores e funcionários da faculdade que de uma forma ou de outra me ajudaram.

Resumo

Martins, Fábio Jessen Werneck de Almeida; Vellasco, Marley (Orientadora); Figueiredo, Karla (Orientadora). **Métodos de Aceleração de Aprendizado Aplicado ao Modelo Neuro-Fuzzy Hierárquico Politree com Aprendizado por Reforço**. Rio de Janeiro, 2010. 157p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Neste trabalho foram desenvolvidos e avaliados métodos com o objetivo de melhorar e acelerar o processo de aprendizado do modelo de *Reinforcement Learning* Neuro-Fuzzy Hierárquico Politree (RL-NFHP). Este modelo pode ser utilizado para dotar um agente de inteligência através de processo de Aprendizado por Reforço (*Reinforcement Learning*). O modelo RL-NFHP apresenta as seguintes características: aprendizado automático da estrutura do modelo; auto-ajuste dos parâmetros associados à estrutura; capacidade de aprendizado da ação a ser adotada quando o agente está em um determinado estado do ambiente; possibilidade de lidar com um número maior de entradas do que os sistemas neuro-fuzzy tradicionais; e geração de regras linguísticas com hierarquia. Com intenção de melhorar e acelerar o processo de aprendizado do modelo foram implementadas seis políticas de seleção, sendo uma delas uma inovação deste trabalho (*Q-DC-roulette*); implementado o método *early stopping* para determinação automática do fim do treinamento; desenvolvido o *eligibility trace* cumulativo; criado um método de poda da estrutura, para eliminação de células desnecessárias; além da reescrita do código computacional original. O modelo RL-NFHP modificado foi avaliado em três aplicações: o *benchmark* Carro na Montanha simulado, conhecido na área de agentes autônomos; uma simulação robótica baseada no robô Khepera; e uma num robô real NXT. Os testes efetuados demonstram que este modelo modificado se ajustou bem a problemas de sistemas de controle e robótica, apresentando boa generalização. Comparado o modelo RL-NFHP modificado com o original, houve aceleração do aprendizado e obtenção de menores modelos treinados.

Palavras-chave

Agente Inteligente; Neuro-Fuzzy; Aprendizado por Reforço; Aprendizado Automático; Modelo Hierárquico Politree

Abstract

Martins, Fábio Jessen Werneck de Almeida; Vellasco, Marley (Advisor); Figueiredo, Karla (Advisor). **Methods for Acceleration of Learning Process of Reinforcement Learning Neuro-Fuzzy Hierarchical Politree Model**. Rio de Janeiro, 2010. 157p. MSc. Dissertation - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

In this work, methods were developed and evaluated in order to improve and accelerate the learning process of Reinforcement Learning Neuro-Fuzzy Hierarchical Politree Model (RL-NFHP). This model is employed to provide an agent with intelligence, making it autonomous, due to the capacity of ratiocinate (infer actions) and learning, acquired knowledge through interaction with the environment by Reinforcement Learning process. The RL-NFHP model has the following features: automatic learning of structure of the model; self-adjustment of parameters associated with its structure, ability to learn the action to be taken when the agent is in a particular state of the environment; ability to handle a larger number of inputs than the traditional neuro-fuzzy systems; and generation of rules with linguistic interpretable hierarchy. With the aim to improve and accelerate the learning process of the model, six selection action policies were developed, one of them an innovation of this work (*Q-DC-roulette*); implemented the *early stopping* method for automatically determining the end of the training; developed a cumulative *eligibility trace*; created a method of pruning the structure, for removing unnecessary cells; in addition to rewriting the original computer code. The modified RL-NFHP model was evaluated in three applications: the simulated benchmark Car-Mountain problem, well known in the area of autonomous agents; a simulated application in robotics based on the Khepera robot; and an application in a real robot. The experiments show that this modified model fits well the problems of control systems and robotics, with a good generalization. Compared the modified RL-NFHP model with the original one, there was acceleration of learning process and smaller structures of the model trained.

Keywords

Intelligent Agent; Neuro-Fuzzy Systems; Learning; Reinforcement Learning; Automatic Learning; Hierarchical Politree Model

Sumário

| | |
|------------------------------------------------------------------------------------|----|
| 1. INTRODUÇÃO | 17 |
| 1.1. Motivação | 17 |
| 1.2. Objetivo | 20 |
| 1.3. Descrição do Trabalho e Contribuições | 20 |
| 1.4. Estrutura do Trabalho | 22 |
| 2. MODELO DE REINFORCEMENTE LEARNING NEURO-FUZZY HIERÁRQUICO POLITREE (RL-NFHP) | 23 |
| 2.1. Introdução | 23 |
| 2.2. Particionamento Politree | 25 |
| 2.3. Célula Básica RL-Neuro-Fuzzy Hierárquico Politree | 26 |
| 2.4. Arquitetura RL-NFHP | 30 |
| 2.4.1. Antecedentes das Regras do Modelo RL-NFHP | 34 |
| 2.4.2. Consequentes das Regras do Modelo RL-NFHP | 35 |
| 2.5. Algoritmo de Aprendizado | 37 |
| 2.5.1. Criação do Controle RL-NFHP | 37 |
| 2.5.2. Início da Simulação | 39 |
| 2.5.3. Seleção das Ações | 39 |
| 2.5.4. Cálculo da Saída | 40 |
| 2.5.5. Cálculo do Retorno e Retropropagação | 44 |
| 2.5.6. Atualização da Função Valor Q | 45 |
| 2.5.7. Particionamento | 50 |
| 2.5.8. Fim do Episódio | 56 |
| 2.5.9. Fim do Treinamento | 56 |
| 2.6. Teste e Uso do Modelo RL-NFHP | 57 |
| 3. MÉTODOS DE ACELERAÇÃO DO MODELO RL-NFHP | 58 |
| 3.1. Políticas de seleção de ação | 58 |
| 3.1.1. Política ϵ -greedy | 58 |
| 3.1.2. Política Q -roulette | 59 |
| 3.1.3. Política DC -roulette | 60 |
| 3.1.3. Política $Q+DC$ -roulette | 61 |

| | |
|----------------------------------------------------------------------|-----|
| 3.1.5. Política <i>Q-DC-roulette</i> | 61 |
| 3.2. <i>Early stopping</i> | 64 |
| 3.3. <i>Elegibility trace</i> cumulativo | 64 |
| 3.4. Poda da estrutura RL-NFHP | 66 |
| 3.5. Reescrita do algoritmo | 66 |
| 4. ESTUDO DE CASOS | 70 |
| 4.1. Carro na montanha (Mountain-Car Problem) | 70 |
| 4.1.1. Experimento 1: Política de escolha das ações | 73 |
| 4.1.2. Experimento 2: Variação da probabilidade de escolha aleatória | 83 |
| 4.1.3. Experimento 3: Função de crescimento | 86 |
| 4.1.4. Experimento 4: <i>Elegibility Trace</i> Cumulativo | 90 |
| 4.1.5. Experimento 5: Alfa de atualização de $Q(s,a)$ | 94 |
| 4.1.6. Experimento 6: Função de Atualização | 98 |
| 4.1.7. Experimento 7: Comparação entre modelos | 103 |
| 4.2. Aplicação simulada em robótica (Khepera) | 113 |
| 4.2.1. Experimento 1: Sem obstáculo | 117 |
| 4.2.2. Experimento 2: Poda da estrutura | 121 |
| 4.2.3. Experimento 3: Com obstáculo | 123 |
| 4.3. Aplicação real em robótica (Lego MindStorms NXT) | 131 |
| 5. CONCLUSÕES | 140 |
| 5.1. Conclusões | 140 |
| 5.2. Sugestões para Trabalhos Futuros | 143 |
| 6. REFERÊNCIAS BIBLIOGRÁFICAS | 145 |
| APÊNDICE: APRENDIZADO POR REFORÇO | 149 |

Lista de Figuras

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 2.1: (a) Exemplo de particionamento Politree para espaços de dimensão 2 e (b) árvore representativa deste particionamento. | 25 |
| Figura 2.2: Célula Reinforcement Learning Neuro-Fuzzy Quadtree (Politree com $n=2$). | 27 |
| Figura 2.3: Exemplo de perfil das funções de pertinência da célula RL-NFP. | 27 |
| Figura 2.4: Célula RL-NFP representada sob o formato de rede neuro-fuzzy. | 28 |
| Figura 2.5: Divisão em quadrantes realizada pelas FPs alto e baixo com entrada incidindo sobre o quadrante 4. | 29 |
| Figura 2.6: (a) Exemplo de arquitetura RL-NFHP e (b) Representação genérica em árvore do modelo RL-NFHP com n entradas e três células (círculos), onde cada célula possui $m = 2^n$ consequentes (quadrados). | 31 |
| Figura 2.7: Esquema do processo de aprendizado do agente | 34 |
| Figura 2.8: Interior da célula RL-NFHP com duas entradas (Quadtree). | 35 |
| Figura 2.9: Fluxograma de treinamento. | 38 |
| Figura 2.10: (a) Exemplo de arquitetura RL-NFHP com 2 entradas e (b) sua representação em árvore. Esta estrutura possui duas células (círculos), onde cada célula tem 4 consequentes (quadrados). | 41 |
| Figura 2.11: Exemplo de arquitetura RL-NFHP de 2 entradas, mostrando os graus de pertinência dos conjuntos fuzzy das duas células que compõe a estrutura. | 41 |
| Figura 2.12: Exemplo de arquitetura RL-NFHP de 2 entradas, mostrando os graus de pertinência dos conjuntos fuzzy das duas células que compõe a estrutura. | 43 |
| Figura 2.13: (a) Exemplo de arquitetura RL-NFHP com 2 entradas e (b) sua representação em árvore mostrando a retropropagação do retorno do ambiente. Esta estrutura possui duas células (círculos), onde cada célula tem 4 consequentes (quadrados). | 45 |
| Figura 2.14: Caso exemplo de atualização da função de valor Q_{t+1} relativa à eq. 2.17. | 48 |
| Figura 2.15: Particionamento da célula RL-NFP ₀ . | 50 |
| Figura 2.16: Função de crescimento. | 53 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figura 3.1: Probabilidade de seleção das ações da tabela 3.1 de acordo com as políticas: (a) Q-roulette; (b) Q-roulette Giggs-Boltzman; (c) DC-roulette e (d) Q-DC-roulette | 63 |
| Figura 3.2: Hierarquia entre as classes do modelo RL-NFHP | 68 |
| Figura 4.1: Benchmark Carro na Montanha | 71 |
| Figura 4.2: Tela de simulação do Carro na Montanha | 72 |
| Figura 4.3: Particionamento do estado no modelo RL-NFHP modificado de 23 células. | 109 |
| Figura 4.4: Árvore de células do modelo RL-NFHP modificado de 23 células. | 109 |
| Figura 4.5: Ambiente de simulação do robô. | 113 |
| Figura 4.6: Detecção de obstáculo. | 114 |
| Figura 4.7: Identificação dos ângulos de interesse. | 115 |
| Figura 4.8: Ambiente de simulação do robô Khepera sem obstáculo. | 117 |
| Figura 4.9: Ambiente de simulação do robô Khepera com obstáculo. | 124 |
| Figura 4.10: Caminho percorrido após aprendizado para posições e ângulos iniciais de 1 a 12. | 126 |
| Figura 4.11: Particionamento do estado no modelo RL-NFHP modificado de 238 células. | 129 |
| Figura 4.12: Cenário real mostrando: (a) o robô; (b) o obstáculo central; (c) a posição final desejada ou objetivo; (d) as paredes laterais. | 131 |
| Figura 4.13: Robô Lego MindStorms NXT montado para avaliação prática do modelo RL-NFHP, constituído por: (a) brick NXT; (b) 3 sensores ultrassônicos; (c) 2 servo-motores. | 132 |
| Figura 4.14: Detalhe da estrutura do TriBot do robô. | 133 |
| Figura 4.15: Caminho percorrido pelo robô real para posições e ângulos iniciais de 1 a 12. | 136 |
| Figura 4.16: Caminho percorrido pelo robô real para posições e ângulos iniciais de 1 a 5. | 137 |

Lista de Gráficos

| | |
|------------------------------------------------------------------------------------------------|----|
| Gráfico 4.1: Aproveitamento do treinamento. | 76 |
| Gráfico 4.2: Histograma da média de ciclos | 76 |
| Gráfico 4.3: Histograma das épocas de treinamento usando <i>Q-roulette</i> | 77 |
| Gráfico 4.4: Histograma das épocas de treinamento usando DC-roulette. | 78 |
| Gráfico 4.5: Histograma das épocas de treinamento usando Q+DC-roulette. | 78 |
| Gráfico 4.6: Histograma das épocas de treinamento usando Q-DC-roulette. | 79 |
| Gráfico 4.7: Histograma das épocas de treinamento usando ϵ -greedy. | 79 |
| Gráfico 4.8: Histograma do número de células do modelo usando Q-roulette. | 80 |
| Gráfico 4.9: Histograma do número de células do modelo usando DC-roulette. | 81 |
| Gráfico 4.10: Histograma do número de células do modelo usando Q+DC-roulette. | 81 |
| Gráfico 4.11: Histograma do número de células do modelo usando Q-DC-roulette. | 82 |
| Gráfico 4.12: Histograma do número de células do modelo usando ϵ -greedy. | 82 |
| Gráfico 4.13: Histograma do número máximo de células ativas por episódio usando Q-DC-roulette. | 83 |
| Gráfico 4.14: Histograma das épocas de treinamento usando DC-roulette. | 85 |
| Gráfico 4.15: Histograma das épocas de treinamento usando Q-DC-roulette. | 85 |
| Gráfico 4.16: Histograma das épocas de treinamento usando ϵ -greedy. | 86 |
| Gráfico 4.17: Histograma das épocas de treinamento usando DC-roulette. | 88 |
| Gráfico 4.18: Histograma das épocas de treinamento usando Q-DC-roulette. | 88 |
| Gráfico 4.19: Histograma das épocas de treinamento usando ϵ -greedy. | 89 |

| | |
|------------------------------------------------------------------------------------|-----|
| Gráfico 4.20: Histograma do número de células do modelo usando DC-roulette. | 89 |
| Gráfico 4.21: Histograma do número de células do modelo usando Q-DC-roulette. | 90 |
| Gráfico 4.22: Histograma do número de células do modelo usando ϵ -greedy. | 90 |
| Gráfico 4.23: Histograma das épocas de treinamento usando DC-roulette. | 92 |
| Gráfico 4.24: Histograma das épocas de treinamento usando Q-DC-roulette. | 92 |
| Gráfico 4.25: Histograma das épocas de treinamento usando ϵ -greedy. | 93 |
| Gráfico 4.26: Histograma do número de células do modelo usando DC-roulette. | 94 |
| Gráfico 4.27: Histograma do número de células do modelo usando Q-DC-roulette. | 94 |
| Gráfico 4.28: Histograma do número de células do modelo usando ϵ -greedy. | 95 |
| Gráfico 4.29: Histograma das épocas de treinamento usando DC-roulette. | 96 |
| Gráfico 4.30: Histograma das épocas de treinamento usando Q-DC-roulette. | 96 |
| Gráfico 4.31: Histograma das épocas de treinamento usando ϵ -greedy. | 97 |
| Gráfico 4.32: Histograma do número de células do modelo usando DC-roulette. | 98 |
| Gráfico 4.33: Histograma do número de células do modelo usando Q-DC-roulette. | 98 |
| Gráfico 4.34: Histograma do número de células do modelo usando ϵ -greedy | 98 |
| Gráfico 4.35: Histograma das épocas de treinamento usando DC-roulette. | 100 |
| Gráfico 4.36: Histograma das épocas de treinamento usando Q-DC-roulette. | 100 |
| Gráfico 4.37: Histograma das épocas de treinamento usando ϵ -greedy. | 101 |
| Gráfico 4.38: Histograma do número de células do modelo usando DC-Roulette. | 102 |

| | |
|----------------------------------------------------------------------------------------------------------|-----|
| Gráfico 4.39: Histograma do número de células do modelo usando Q-DC-Roulette. | 102 |
| Gráfico 4.40: Histograma do número de células do modelo usando ϵ -greedy. | 103 |
| Gráfico 4.41: Posição e velocidade do carro ao longo do episódio iniciando no vale. | 108 |
| Gráfico 4.42: Posição e velocidade do carro ao longo do episódio iniciando junto à parede. | 109 |
| Gráfico 4.43: Histograma das épocas de treinamento. | 120 |
| Gráfico 4.44: Histograma do número de células do modelo. | 120 |
| Gráfico 4.45: Histograma do número máximo de células ativas por episódio durante a fase de testes. | 121 |
| Gráfico 4.45: Histograma das épocas de treinamento usando Q-DC-roulette com e sem o método de poda. | 123 |
| Gráfico 4.46: Histograma do número de células do modelo usando Q-DC-roulette com e sem o método de poda. | 123 |
| Gráfico 4.47: Distância e ângulo do robô durante um episódio para a condição inicial 8. | 129 |
| Gráfico 4.48: Distância e sensores do robô durante um episódio para a condição inicial 8. | 129 |

Lista de Tabelas

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela 3.1: Valores de $Q(s,a_i)$ e a quantidade de visitas $C(s,a_i)$ para determinada polipartição em um estado s . | 62 |
| Tabela 3.2: Arquivos do modelo RL-NFHP. | 67 |
| Tabela 4.1: Comparação entre as políticas no modelo RL-NFHP. | 82 |
| Tabela 4.2: Comparação entre as taxas de incremento da porcentagem aleatória de 5% e 40% para as diferentes políticas no modelo RL-NFHP. | 86 |
| Tabela 4.3: Comparação entre valores de n da função de crescimento para as diferentes políticas no modelo RL-NFHP. | 90 |
| Tabela 4.4: Comparação entre valores do parâmetro α usado na atualização de $Q(s,a)$ para as diferentes políticas no modelo RL-NFHP. | 98 |
| Tabela 4.5: Comparação entre as formas de atualização de $Q(s,a)$, com e sem a equação de punição, para as diferentes políticas no modelo RL-NFHP. | 102 |
| Tabela 4.6: Comparação entre modelos para o benchmark Carro na Montanha. | 105 |
| Tabela 4.7: Comparação entre modelos RL-NFHP para o benchmark Carro na Montanha. | 107 |
| Tabela 4.8: Comparação entre as políticas de seleção no modelo RL-NFHP. | 120 |
| Tabela 4.9: Comparação entre o uso ou não do método de poda da estrutura para a política Q-DC-roulette no modelo RL-NFHP. | 123 |

Lista de Siglas

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| ANFIS: <i>Adaptive-Network-Based Fuzzy Inference System</i> | 18 |
| DC-roulette: política de seleção baseada em visita | 60 |
| ϵ -greedy: política de seleção através da maior função valor Q | 58 |
| ELF: <i>Evolutionary Logic Fuzzy</i> | 19 |
| FACL: <i>Fuzzy-Actor-Critic-Learning</i> | 19 |
| FP: função de pertinência | 27 |
| FQL: <i>Fuzzy-Q-Learning</i> | 19 |
| FSOM: <i>Fuzzy Self-Organing Map</i> | 18 |
| JAFIMA: <i>Java Application Framework for Intelligent and Mobile Agents</i> | 144 |
| LF: Lógica Fuzzy | 17 |
| NEFCLASS: <i>Neuro-Fuzzy Classification</i> | 18 |
| NEFCON: <i>Neuro-Fuzzy Controller</i> | 19 |
| Q+DC-roulette: política de seleção baseada ora na função valor Q ora na visita | 61 |
| Q-DC-roulette: política de seleção baseada na função valor Q mesclada à visita | 61 |
| Q-Learning: algoritmo de RL | 156 |
| Q-roulette: política de seleção baseada na função valor Q | 59 |
| RL: <i>Reinforcement Learning</i> (Aprendizado por Reforço) | 149 |
| RL-NFHB: <i>Reinforcement Learning Neuro-Fuzzy Hierarchical Binary Space Partitioning</i> | 23 |
| RL-NFHP: <i>Reinforcement Learning Neuro-Fuzzy Hierarchical Politree</i> (<i>Reinforcement Learning – Neuro-Fuzzy Hierárquico Politree</i>) | 23 |
| RL-NFHP ⁺⁺ : RL-NFHP utilizando os métodos de aceleração | 106 |
| RL-NFP: RL-Neuro-Fuzzy Politree | 25 |
| RN: Redes Neurais | 17 |
| SARSA: algoritmo de RL <i>State-Action-Reward-State-Action</i> | 155 |
| SIF: Sistema de Inferência Fuzzy | 17 |