

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

MÁRCIO MENDONÇA

UMA CONTRIBUIÇÃO AO DESENVOLVIMENTO DE SISTEMAS
INTELIGENTES UTILIZANDO REDES COGNITIVAS DINÂMICAS

TESE

CURITIBA

2011

MÁRCIO MENDONÇA

**UMA CONTRIBUIÇÃO AO DESENVOLVIMENTO DE SISTEMAS
INTELIGENTES UTILIZANDO REDES COGNITIVAS DINÂMICAS**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de “Doutor em Ciências” – Área de Concentração: Engenharia de Automação e Sistemas

Orientadora: Profa. Dra. Lúcia Valéria Ramos de Arruda

CURITIBA

2011

AGRADECIMENTOS

Agradeço a Universidade Tecnológica Federal do Paraná, Campus de Cornélio Procopio, em especial ao Professor Mestre Devanil Antonio Francisco, então gerente de pós graduação, atualmente diretor do Campus Cornélio Procopio pela oportunidade de afastamento para capacitação e amizade.

Agradeço a minha orientadora Profa. Dra. Lúcia Valéria Ramos de Arruda pelo apoio, contribuições e condução das atividades.

A Deus, aos meus amigos do LASCA e da UTFPR de Cornélio, em especial ao Bruno, Bassetto, Castoldi, Montezuma, Marco Antônio, que me deram incentivo na reta final.

A todos da minha família que motivaram e ajudaram superar dificuldades. Ao meu pai que desde muito cedo me deu lições de eletrônica; e ao meu sobrinho Júnior Candido Mendonça, pelo seu sorriso sincero, de uma criança de quase 4 anos. Alias, desde o seu nascimento, me deu outra visão da vida.

Finalmente, dedico esta Tese a memória dos meus três irmãos, Marcos, Maria Aparecida e Maria Cristina.

RESUMO

MENDONÇA, Márcio. Uma contribuição ao desenvolvimento de sistemas inteligentes utilizando redes cognitivas dinâmicas. 2011. 177 f. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Este trabalho propõe o uso de Redes Cognitivas Dinâmicas, em inglês *Dynamics Cognitive Networks* (DCN), uma evolução dos mapas cognitivos e mapas cognitivos *fuzzy*, como uma ferramenta para o desenvolvimento de sistemas inteligentes. Sistemas inteligentes têm como objetivo utilizar o conhecimento empírico de especialistas através da aquisição, tratamento, modelagem e validação de conhecimento na forma de dados e informações, a fim de modelar sistemas complexos. Uma breve introdução sobre as diversas ferramentas para construção de sistemas inteligentes é apresentada, dando uma ênfase especial aos mapas cognitivos *fuzzy*, técnica a partir da qual evoluíram as redes cognitivas dinâmicas. Uma revisão da literatura lista vários trabalhos que utilizam mapas cognitivos em diversas áreas de conhecimento, em especial, trabalhos relacionados com as áreas de conhecimento abordadas (controle de processo e navegação robótica) e trabalhos que abordam evoluções dos *Fuzzy Cognitive Maps*, em especial, as Redes Cognitivas Dinâmicas. Quanto à capacidade de adaptação e ajuste do conhecimento empírico, propostas de adaptação e aprendizado, por exemplo, o uso de Redes Neurais Artificiais com mapeamento reverso e técnicas *soft computing* (algoritmos genéticos, enxame de partículas) são discutidos na otimização do FCM e/ou DCN. Tópicos de controle adaptativo e inteligente são necessários para o entendimento e a apresentação da arquitetura de um controlador inteligente adaptativa em um processo de fermentação. Uma comparação com um controlador PID clássico e alguns benefícios também são observadas, principalmente relacionadas com a facilidade de construção e aplicação do modelo na área de controle. Finalmente, uma proposta de arquitetura chamada DCN (evolução FCM) utiliza uma base conhecimento qualitativa com dois níveis de tomadas de decisões, através de bases de regras e relações de causa e efeito é validada através da aplicação de conhecimento em ambas as áreas mencionadas. De modo específico, essa abordagem inteligente utiliza conhecimento qualitativo, em um controle supervisor de um processo de fermentação alcoólica e tomadas de decisões (ações de controle) em navegação robótica. O conhecimento qualitativo ou “estrutura de crenças” é ajustado através de dados de operação e/ou observação do sistema e especificamente na aplicação de navegação autônoma é também sintonizado dinamicamente através da interação com o ambiente.

Palavras-chave: Mapas Cognitivos *Fuzzy*, Redes Cognitivas Dinâmicas, Sistemas Inteligentes, Navegação Robótica, Sistemas Supervisórios e Controle de Processos.

ABSTRACT

MENDONÇA, Márcio. A Contribution to Intelligent System Modeling based on Dynamic Cognitive Networks. 2011. 177 f. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Centro Federal de Educação Tecnológica do Paraná. Curitiba, 2011.

This work proposes the use of Dynamic Cognitive Networks (DCN), an evolution of Fuzzy Cognitive Maps (FCM) as an intelligent tool to model human knowledge, with examples of applications in process control and robotic navigation. Intelligent systems are designed to use empirical knowledge of experts through the acquisition, processing, modeling and validation in the form of data and information. The introduction of the fundamentals of the tool and a review of work in various areas of knowledge are presented. In particular, work on proposals for development of architectures based on cognitive maps, is special Dynamic Cognitive Maps, and, work related to the areas of expertise addressed (process control and robotic navigation) using Fuzzy Cognitive Maps and / or intelligent systems related to Fuzzy logic, for example. Regarding the ability of adjustment and adaptation of empirical knowledge, proposals for adjustment and learning, for example, the use of Artificial Neural Networks with reverse mapping and soft computing techniques (genetic algorithms, particle swarm) are discussed in the optimization of FCM and / or DCN. Topics for adaptive and intelligent control are necessary to the understanding and presentation of the architecture of an intelligent adaptive controller in a process of fermentation. A comparison with a classical PID controller and some benefits are also observed, mainly related to ease of construction and application of the model in the control area. Finally, a proposed architecture called DCN (evolution FCM) uses a qualitative knowledge base with two levels of decision-making, through rule bases and relations of cause and effect is validated through the application of knowledge in both areas mentioned. Specifically, a new intelligent systems uses knowledge-based qualitative in a supervisory control of a process of fermentation and making decisions (control actions) in robotic navigation. The qualitative knowledge or “belief structure” is adjusted by operating and / or observation of the system and specifically the application of autonomous navigation is also dynamically tuned by interaction with the environment.

Keyword: Fuzzy Cognitive Maps, Dynamic Cognitive Networks, Intelligent Systems, Robotic Navigation, Supervisory Systems and Control Process.

LISTA DE FIGURAS

FIGURA 1 – PROCESSO INICIAL PARA A AQUISIÇÃO DE CONHECIMENTO ESTRUTURADO.....	14
FIGURA 2 – HIERARQUIA DE UM SISTEMA DE CONTROLE INTELIGENTE.....	16
FIGURA 3 – MODELO DO CONTROLADOR LÓGICO <i>FUZZY</i>	31
FIGURA 4 – ARQUITETURA DE UMA REDE NEURAL.....	34
FIGURA 5 – ARQUITETURA DE SUBSUNÇÃO.....	40
FIGURA 6 – ARQUITETURA SUBSUNÇÃO DE 3 CAMADAS.....	41
FIGURA 7 – MODELO PADRÃO DE APRENDIZAGEM POR REFORÇO	42
FIGURA 8 – EXEMPLO HÍBRIDO-SEQUENCIAL	43
FIGURA 9 – EXEMPLO DE UM FCM (GRAFO)	46
FIGURA 10 – FUNÇÕES PERTINÊNCIA <i>FUZZY</i> PARA AQUISIÇÃO RELACIONES CAUSAIS.....	63
FIGURA 11 – EXEMPLO DE UM FCM DISTRIBUÍDO	64
FIGURA 12 – PROCESSO QUÍMICO COM MISTURADOR.....	69
FIGURA 13 – FCM PARA O CONTROLE DO MISTURADOR	70
FIGURA 14 – RNA PARA ADAPTAÇÃO DO FCM (FIGURA 13).....	72
FIGURA 15 – INTERVALOS DE CONFIANÇA E VALORES CALCULADOS PARA OS PESOS PELOS DIFERENTES MÉTODOS	78
FIGURA 16 – ARQUITETURA DCN	88
FIGURA 17 – DCN - ESTRATÉGIA DE FUTEBOL ROBÔS.....	91
FIGURA 18 – PROCESSO DE FERMENTAÇÃO ALCÓOLICA.....	96
FIGURA 19 – COMPORTAMENTO DINÂMICO DAS VARIÁVEIS DE ESTADO DO PROCESSO DE FERMENTAÇÃO	97
FIGURA 20 – ARQUITETURA BASEADA EM DCN PARA CONTROLE SUPERVISÓRIO.....	98
FIGURA 21 – DCN _{NL} /FCM PARA OPERAÇÃO EM MODO NORMAL.....	99
FIGURA 22 – VARIAÇÃO VOLUME, VOLUME E <i>SET-POINT</i>	103
FIGURA 23 – SUPERFÍCIE <i>FUZZY</i> DA RELAÇÃO WF_1	104
FIGURA 24 – DCN _C PARA OPERAÇÃO EM MODO CRÍTICO	107
FIGURA 25 – SUPERVISOR BASEADO EM DCN	108
FIGURA 26 – ARQUITETURA DO SUPERVISOR	108
FIGURA 27 – CONTROLE REGULATÓRIO (PID): EXPERIMENTO 1.....	112
FIGURA 28 – CONTROLE SUPERVISÓRIO (DCN _N + PID):	112
FIGURA 29 – CONTROLE SUPERVISÓRIO (DCN _N + PID):	113
FIGURA 30 – CONTROLE SUPERVISÓRIO (DCN-PID):	116
FIGURA 31 – CONTROLE SUPERVISÓRIO (DCN-PID):	116
FIGURA 32 – CONTROLE REGULATÓRIO (PID): EXPERIMENTO 3.....	118
FIGURA 33 – CONTROLE SUPERVISÓRIO (DCN-PID):	119
FIGURA 34 – CONTROLE SUPERVISÓRIO (DCN-PID):	119
FIGURA 35 – DCN (ARQUITETURA DO CONTROLADOR).....	127
FIGURA 36 – DCN INICIAL.....	130
FIGURA 38 – MODELO PADRÃO DE APRENDIZAGEM POR REFORÇO ..	133

FIGURA 39 – PROPOSTA DA DCN	137
FIGURA 40 – HIERARQUIA DO NAVEGADOR BASEADO EM DCN	138
FIGURA 41 – SUPERFÍCIE DO SISTEMA <i>FUZZY</i> DE NAVEGAÇÃO	140
FIGURA 42 – SUPERFÍCIE DO SISTEMA <i>FUZZY</i>	140
FIGURA 43 – DCN, CENÁRIO COM OBJETO DINÂMICO	143
FIGURA 44 – <i>FUZZY</i> CENÁRIO COM OBJETO DINÂMICO	143
FIGURA 45 – DCN EM CENÁRIO DINÂMICO	144
FIGURA 46 – DCN EM CENÁRIO DINÂMICO	144
FIGURA 47 – <i>FUZZY</i> EM CENÁRIO DINÂMICO	145
FIGURA 48 – <i>FUZZY</i> EM CENÁRIO DINÂMICO COM	145
FIGURA 49 – DCN OBSTÁCULO INESPERADO MAIS RÁPIDO	147
FIGURA 50 – <i>FUZZY</i> OBSTÁCULO INESPERADO MAIS RÁPIDO	147
FIGURA 51 – DCN OBSTÁCULO INESPERADO MAIS RÁPIDO	147
FIGURA 52 – SENSORES E VARIAÇÕES DOS PESOS	148
FIGURA 53 – DCN CENÁRIO DRÁSTICO.	148
FIGURA 54 – SENSORES E VARIAÇÕES DOS PESOS DAS RELAÇÕES CAUSAIS	149
FIGURA 55 – <i>FUZZY</i> CENÁRIO DRÁSTICO	149
FIGURA 56 – CENÁRIO DA DCN COM REFINAMENTO	151
FIGURA 57 – CENÁRIO E TRAJETÓRIA DA DCN	151
FIGURA 58 – SENSORES E VARIAÇÕES DOS PESOS	152
FIGURA 59 – CENÁRIO E TRAJETÓRIA DA DCN COM REFINAMENTO ...	153
FIGURA 60 – CENÁRIO E TRAJETÓRIA DA DCN	153
FIGURA 61 – SENSORES E VARIAÇÕES DOS PESOS	154
FIGURA 62 – EXPERIMENTO 1: CENÁRIO REAL,	155
FIGURA 63 – EXPERIMENTO 1: CENÁRIO REAL,	155
FIGURA 64 – EXPERIMENTO 1: CENÁRIO REAL,	155
FIGURA 65 – EXPERIMENTO 2: CENÁRIO REAL,	156
FIGURA 66 – EXPERIMENTO 2: CENÁRIO REAL,	156
FIGURA 67 – EXPERIMENTO 2: CENÁRIO REAL,	156

LISTA DE QUADROS

QUADRO 1 – CONSTRUÇÃO FCM CLÁSSICO.....	64
QUADRO 2 – RESULTADO DO MODELO BASEADO EM FCM COM OS DIFERENTES MÉTODOS OTIMIZAÇÃO.....	77
QUADRO 3 – CONSTRUÇÃO DA DCN.....	87
QUADRO 4 – RELAÇÃO CAUSAL DO DCN ₁	102
QUADRO 5 – EVOLUÇÃO DA DCN _N PARA UMA VARIAÇÃO DE <i>SET- POINT</i>	105
QUADRO 6 – CONDIÇÕES INICIAIS.....	110
QUADRO 7 – ÍNDICES DE DESEMPENHO PARA O EXPERIMENTO1.....	111
QUADRO 8 – ÍNDICES DE DESEMPENHO PARA O EXPERIMENTO2.....	115
QUADRO 9 – DESCRIÇÃO DAS RELAÇÕES DA DCN DE NAVEGAÇÃO..	129

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmos Genéticos
ANFIS	<i>Adaptive Network-Based Fuzzy Inference System</i>
CM	<i>Cognitive Maps</i>
FCM	<i>Fuzzy Cognitive Maps</i>
PID	Proporcional Integral Derivativo
RNA	Redes Neurais Artificiais
PSO	<i>Particle Swarm Optimization</i>
Rb-FCM	<i>Rule Base – Fuzzy Cognitive Map</i>
ED-FCM	<i>Event Drive – Fuzzy Cognitive Map</i>
DCN	<i>Dynamic Cognitive Network</i>

SUMÁRIO

1 INTRODUÇÃO – CONCEITOS E FUNDAMENTOS.....	12
1.1 INTRODUÇÃO - FUNDAMENTOS E ESTRUTURA	12
1.2 AQUISIÇÃO DE CONHECIMENTO	12
1.2.1 Sistemas Inteligentes	14
1.3 CONTROLE INTELIGENTE	15
1.3.1 Abordagens Para Construção de Sistemas Inteligentes	18
1.4 MOTIVAÇÃO.....	19
1.5 OBJETIVO.....	20
1.6 CONTRIBUIÇÕES	22
1.7 ESTRUTURA DO TRABALHO.....	24
2 FUNDAMENTOS DE CONTROLE INTELIGENTE.....	27
2.1 INTRODUÇÃO	27
2.2 SISTEMAS BASEADOS EM CONHECIMENTO.....	27
2.2.1 Regras Como Representação de Conhecimento	28
2.3 METODOLOGIAS DA INTELIGÊNCIA COMPUTACIONAL	29
2.3.1 Sistemas Fuzzy	30
2.3.2 Redes Neurais Artificiais	33
2.3.3 Computação Evolutiva.....	35
2.3.4 Agentes Inteligentes	37
2.3.5. Arquiteturas de Agentes	38
2.3.6 Aprendizagem Por Reforço	41
2.3.7 Sistemas Híbridos Neuro-Fuzzy	44
2.4. MAPAS COGNITIVOS <i>FUZZY</i>	45
2.5 REDES COGNITIVAS DINÂMICAS	49
2.6 REVISÃO DA LITERATURA (FCM E DCN)	51
3 DESENVOLVIMENTO E MÉTODOS DE OTIMIZAÇÃO DE MAPAS COGNITIVOS FUZZY APLICADO EM CONTROLE DE PROCESSOS.....	60
3.1 INTRODUÇÃO	60
3.1.1 MAPAS COGNITIVOS FUZZY EM CONTROLE DE PROCESSOS	60
3.2 CONSTRUÇÃO DO FCM.....	61
3.2.1 FCM Sintonizado Por Redes Neurais.....	66
3.2.2 FCM Sintonizado Por AG	66
3.2.3 FCM Sintonizado Por PSO	67
3.3 CASO DE ESTUDO: CONTROLE DE NÍVEL	67
3.3.1 Descrição do Processo Estudado	68
3.4 MÉTODOS EMPREGADOS NA SINTONIA DO FCM.....	72
3.4.1 Sintonia do FCM Empregando RNA.....	72
3.4.2 Sintonia do FCM utilizando AG	74
3.4.3 Sintonia do FCM Utilizando PSO	76
3.4.4 Comparação dos Resultados	76
3.5 DESVANTAGENS NA UTILIZAÇÃO DE FCM	78
3.6 CONCLUSÃO DO CAPÍTULO.....	80
4 NOVO MÉTODO PARA MODELAGEM DE SISTEMAS DINAMICOS BASEADO EM DCN	81

4.1 INTRODUÇÃO	81
4.1.1 Redes dinâmicas cognitivas	81
4.2 MODELAGEM DE SISTEMAS DINÂMICOS ATRAVÉS DE DCNS	83
4.3 MODELO DE UMA REDE COGNITIVA DINÂMICA	84
4.4 ALGORITMO PARA CONSTRUÇÃO DE UMA DCN	86
4.5 DCN APLICADA NA ESTRATÉGIA DE TIME DE FUTEBOL DE ROBÔS	90
4.6 CONCLUSÃO DO CAPÍTULO	94
5 MODELO DE UM SISTEMA SUPERVISÓRIO BASEADO EM DCN	95
5.1 INTRODUÇÃO DO CAPÍTULO	95
5.1.1. PROCESSO DE FERMENTAÇÃO ALCÓOLICA	95
5.2 CONTROLE SUPERVISÓRIO COM DCN	97
5.3 RESULTADOS DE SIMULAÇÃO	109
5.4 CONCLUSÃO DO CAPÍTULO	120
6 DCN EM NAVEGAÇÃO ROBÓTICA	122
6.1 INTRODUÇÃO	122
6.1.1 Navegação Robótica	122
6.1.2 Proposta de DCN Para Navegação Robótica	124
6.1.3 Construção da DCN	127
6.1.4 O Modelo DCN Adaptado Por Reforço	132
6.1.5 Expansão da DCN	136
6.2 CONTROLADOR BASEADO EM LÓGICA FUZZY	139
6.3 RESULTADOS DAS SIMULAÇÕES	141
6.4 SIMULAÇÕES COM AQUISIÇÃO DE ALVOS E DESVIO DE OBSTÁCULOS	150
6.5 EXPERIMENTO EM CENÁRIO REAL	154
6.6 CONCLUSÃO DO CAPÍTULO	157
7 CONCLUSÕES	159
7.1 INTRODUÇÃO	159
7.2 VANTAGENS E DESVANTAGENS NO USO DE SISTEMAS INTELIGENTES BASEADOS EM DCN	159
7.3 CONTRIBUIÇÕES	160
7.4 FUTUROS TRABALHOS	163
7.5 CONCLUSÃO FINAL	164
REFERÊNCIAS	165

1 INTRODUÇÃO – CONCEITOS E FUNDAMENTOS

1.1 INTRODUÇÃO - FUNDAMENTOS E ESTRUTURA

Este capítulo apresenta uma introdução sobre métodos de aquisição e representação de conhecimento, mais comumente utilizados no desenvolvimento de sistemas inteligentes. Esta apresentação contextualiza a proposta desta Tese para a modelagem de sistemas complexos e apresenta os objetivos, a motivação e as contribuições esperadas no desenvolvimento do trabalho. A estrutura para o desenvolvimento do trabalho é apresentada por meio de um breve resumo dos demais capítulos.

1.2 AQUISIÇÃO DE CONHECIMENTO

A aquisição de conhecimento pode ser definida como o processo de compreender e organizar o conhecimento, e é a atividade inicial da engenharia de conhecimento. Por conseguinte, esse procedimento é a primeira etapa para a modelagem da ferramenta proposta nessa Tese.

O processo de aquisição de conhecimento objetiva a construção de um modelo com comportamento mais próximo possível do sistema físico real. É fato que tais modelos podem envolver variáveis, parâmetros ou fatores de difícil obtenção experimental ou que sejam coletados desordenadamente e/ou extraídos da literatura; ou ainda adequados a problemas não característicos daquelas classes de problemas matemáticos para os quais não se tem como obter a solução exata (objeto de estudo da área de matemática). De um modo geral, algumas das dificuldades encontradas na aquisição de conhecimento são: como reconhecer os processos/sistemas; identificar as variáveis e parâmetros importantes; classificar o tipo de problema físico; identificar a família de modelos matemáticos que pode ser associado; selecionar o método e/ou a ferramenta para a busca e análise do modelo; construir o algoritmo associado; desenvolver ou usar pacotes de computação científica; interpretar resultados; analisar as possíveis fontes de erros; e, se necessário, retomar algumas das etapas anteriores? Neste

contexto, em geral, torna-se necessário simplificar o modelo matemático ou procurar métodos de resolução que gerem soluções aceitáveis.

Sistemas computacionais inteligentes pode ser um método alternativo para o processo de aquisição de conhecimento. O principal método utilizado nessa Tese foi através da observação e análise de estímulo e resposta no comportamento do sistema ou processo. Deste modo, o conhecimento inicialmente modelado é qualitativo e é representado de forma estruturada, na qual identifica as variáveis, as causas e efeitos de forma qualitativa utilizando termos linguísticos. O processo de aquisição é apresentado de forma resumida na figura 1.

Para ilustrar esse processo de aquisição inicial de conhecimento. Pode-se citar o exemplo de aplicação do capítulo 6. Nesse exemplo, o modelo cognitivo executa tomadas de decisões como virar à esquerda quando surge um obstáculo à direita. Nesse caso, duas variáveis (conceitos, no modelo cognitivo) são detectadas, uma entrada por meio do sensor esquerdo e uma saída que modela a ação de virar à direita. Uma relação do tipo causa efeito se surgir obstáculo virar e finalmente pela observação do comportamento dinâmico. Deste modo, um termo linguístico deve ser definido de acordo com a intensidade do efeito da causa, por exemplo, fraco, médio ou forte. Outras formas de representação de conhecimento são também empregadas na ferramenta cognitiva propostas nessa Tese, entretanto, são aplicadas para adaptar, ajustar, sintonizar e incluir a ocorrência de eventos no modelo inicial.

Deste modo, é possível realizar a coleta adequada de informações, identificação e análise de dados importantes (variáveis, parâmetros, interações, entre outras). Ressalta-se que, o sucesso das etapas seguintes na construção de um modelo depende do sucesso das anteriores. Finalmente, é de fundamental importância, efetuar a simulação ou análise do processo em estudo, objetivando o seu planejamento, controle ou implementação melhorada.

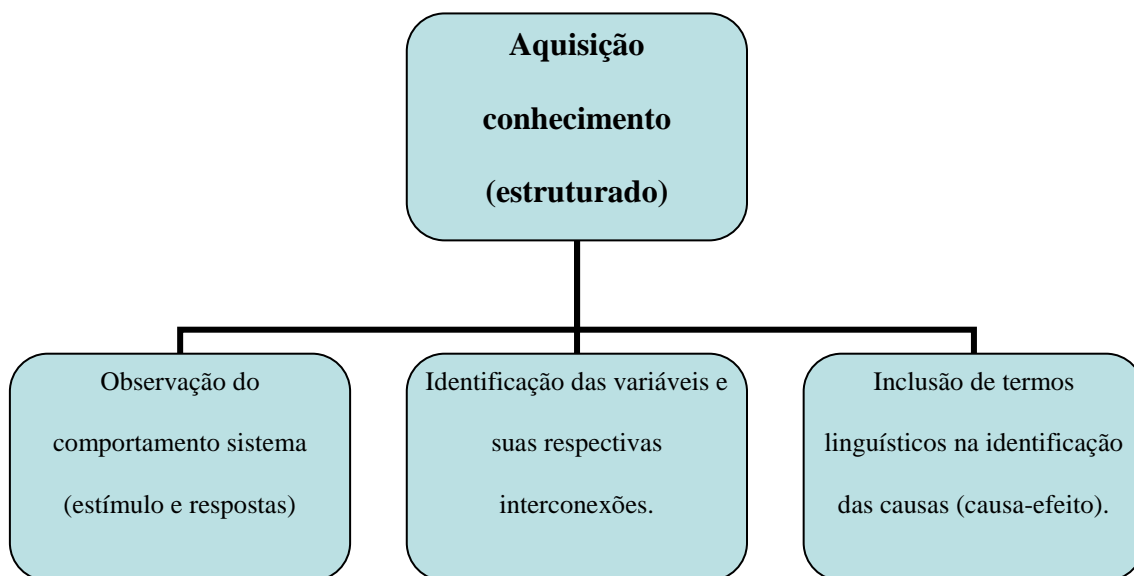


Figura 1- Processo inicial para a aquisição de conhecimento estruturado.
Fonte: Autoria própria.

Estas são questões abordadas nos próximos capítulos as quais são inerentes ao processo de construção e desenvolvimento de sistemas computacionais inteligentes que é o tema de interesse desta Tese. Em especial, as etapas de desenvolvimento e aplicações de um sistema computacional inteligente baseado em mapas cognitivos, conhecimento heurístico e modelos mentais serão discutidos em detalhes.

1.2.1 Sistemas Inteligentes

Seres humanos são capazes de lidar com processos bastante complexos, baseados em informações imprecisas e/ou aproximadas. A estratégia adotada é também de natureza imprecisa e geralmente possível de ser expressa em termos linguísticos. Deste modo, através de conceitos da lógica *fuzzy*, é possível modelar esse tipo de informação (ZADEH, 1992). Entretanto, informações também podem ser adquiridas a partir dos dados de operação e/ou, funcionamento de um sistema. Este tipo de manipulação pode ser realizado mediante técnicas de identificação e modelagem (PASSINO; YOURKOVICH, 1997). Entre estas técnicas as redes neurais se destacam pela capacidade de processar dados massivos de uma maneira paralela (HAYKIN,

2000). Por outro lado, o mapa cognitivo *fuzzy*, do inglês *Fuzzy Cognitive Map* (FCM), é uma ferramenta para a modelagem do conhecimento humano, obtido por meio de termos linguísticos, inerentes aos sistemas *fuzzy*, mas com uma estrutura semelhante à das Redes Neurais Artificiais (RNA), a qual facilita o tratamento de dados e possui capacidade de treinamento e adaptação. Neste contexto, um FCM pode ser considerado um modelo híbrido (fusão de duas ou mais abordagens em sistemas inteligentes) com um forte grau de iteração, no qual não se pode identificar e separar as estruturas das áreas de origem, somente a semântica de construção.

Além das vantagens e características herdadas das técnicas primárias, FCM foi originalmente proposto como uma ferramenta para construção de modelos e/ou mapas cognitivos em diversas áreas de conhecimento, o que sugere que a técnica confere uma facilidade na abstração de informações necessária à modelagem de sistemas complexos, devido à sua semelhança de construção com o raciocínio humano. Deste modo, os mapas cognitivos *fuzzy* agregam vantagens de aquisição, tratamento e capacidade de adaptação a partir de dados e informações do sistema a ser modelado, com uma capacidade inteligente de tomada de decisão devido à sua natureza heurística. Estas características os tornam apropriados, a priori, para desenvolvimento de sistemas inteligentes de controle e automação.

1.3 CONTROLE INTELIGENTE

A área de controle tem expandido de forma significativa o conceito de automação e ampliado a implantação do controle automático na prática, em especial na área industrial. No entanto, as técnicas envolvidas geralmente dependem da existência de um modelo matemático para a planta e para o controlador, além de requerer formas estruturadas de representação e raciocínio lógico (ASTRÖM; WITTENMARK, 1995). Uma alternativa para se controlar sistemas complexos, em que não existem modelos disponíveis sobre o processo está no uso de sistemas inteligentes. Para estes sistemas não são necessários conhecimentos intrínsecos do modelo matemático, mas sim um conhecimento heurístico do comportamento e funcionamento do sistema físico. A classe de controladores que usa técnicas oriundas da inteligência computacional, tais como

sistemas *fuzzy*, redes neurais, algoritmos evolutivos entre outras para manipular, representar e implementar conhecimento heurístico são conhecidos como controladores inteligentes (FABRO; ARRUDA, 2003).

Passino e Yourkovich (1997) apresentam uma proposta com três níveis de hierarquia para um sistema de controle inteligente autônomo: nível de execução, nível de coordenação e nível de gerenciamento e organização, como mostra a figura 2.

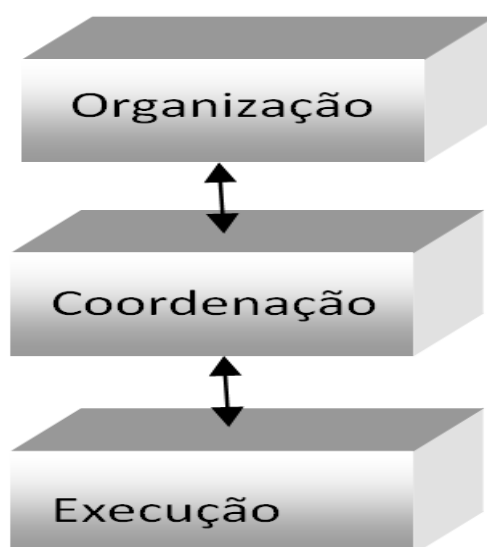


Figura 2- Hierarquia de um Sistema de Controle Inteligente
Fonte: Adaptado (PASSINO E YOURKOVICH, 1997).

O nível de execução é responsável pelas tarefas de aquisição e processamento de sinais e dados e também pelos algoritmos de controle. Estas tarefas são realizadas na planta pelos sensores, controladores lógicos e atuadores do sistema entre outros. O nível de coordenação é responsável pelas tarefas de sintonia, gerenciamento de tarefas, supervisão, e adaptação dos algoritmos de controle, além de detecção de falhas, e tomada de decisão simbólica envolvendo o funcionamento geral da planta. Por exemplo, uma das mais importantes tarefas deste nível, considerada parte da supervisão, é a geração de *set-points* para os controladores nos níveis inferiores. Finalmente, o nível de gerenciamento, o mais alto de acordo com a hierarquia sugerida, é o nível de organização em que se realizam as tarefas de análise de desempenho, determinação dos

objetivos do sistema de controle, tomada de decisão e adaptação que se refletem no nível de coordenação.

Os sistemas inteligentes e/ou controladores inteligentes podem ser utilizados com sucesso em várias funções nos três níveis da hierarquia acima descrita. Por exemplo, controladores *fuzzy* e *neuro-fuzzy* podem ser utilizados no nível de execução como controladores adaptativos, redes neurais podem ser usada para detecção de faltas no nível de coordenação e os algoritmos genéticos são utilizados na tarefa de otimização de desempenho e planejamento de tarefas.

Nesta Tese, os sistemas inteligentes serão utilizados principalmente para a tarefa de controle adaptativo/reativo no nível mais baixo da hierarquia, e como tomado de decisão no nível de coordenação/planejamento.

Ainda segundo Passino e Yourkovich (1997), um controlador inteligente seja no chão de fábrica seja no nível de supervisão, deve possuir um considerável grau de autonomia sendo capaz de:

1. Detectar alterações na planta ou no ambiente;
2. Tomar decisões mesmo na presença de informações incompletas ou conflitantes, provenientes de múltiplos sensores;
3. Gerar modelos ou representações qualitativas;
4. Gerar ações apropriadas para modificar o desempenho do sistema frente a eventos inesperados; e reconfigurar o sistema na presença de falhas internas;
5. Integrar várias funcionalidades tais que diversas estratégias de controle e identificação de plantas, adaptação, incorporação de perturbações e incertezas nos modelos.

Como resultado, as seguintes características são necessárias aos sistemas inteligentes de controle e automação (FIGUEIREDO et al, 2007):

- Aprendizado automático;
- Auto-reconfiguração;
- Mecanismos de inferência;
- Elaboração e uso de planejamentos;
- Mecanismos de tomada de decisão;

- Habilidade para extrair as informações mais relevantes de bases de dados não-rotuladas, não-estruturadas e ruidosas.

O desenvolvimento e a implementação computacional deste tipo de sistemas devem ser feitos através de uma metodologia que possibilite uma capacidade de decisão através da aquisição de conhecimento empírico humano, uma capacidade de adaptação e ou aprendizado a partir de dados numéricos, a fim de atender diferentes necessidades e/ou situações de controle, de acordo com a dinâmica do processo e do ambiente na qual o sistema está inserido. Além disso, esse sistema deve ser capaz de fazer inferências a partir de uma base de conhecimentos em que regras heurísticas de tomadas de decisão possam ser facilmente incorporadas. Assim, um sistema inteligente construído a partir de dados e informações heurísticas possibilita a integração de diferentes processos e comportamentos de modo flexível e intuitivo, objetivando melhorar o desempenho do sistema modelado como um todo e auxiliar na tomada de decisão de controle, manutenção, detecção de falhas entre outras funcionalidades inerentes a um sistema real de automação e controle.

1.3.1 Abordagens Para Construção de Sistemas Inteligentes

As duas abordagens descritas a seguir auxiliam na construção de modelos cognitivos úteis ao desenvolvimento de sistemas inteligentes de controle e automação: o raciocínio dedutivo e o raciocínio indutivo.

O primeiro permite a construção de um modelo inicial através de conhecimento empírico humano, enquanto que o raciocínio indutivo permite ajustes e adaptação do modelo previamente construído através da observação do desempenho dinâmico.

O raciocínio dedutivo é o processo que as pessoas utilizam para inferir conclusões baseadas em informações já conhecidas. Operadores humanos podem, por exemplo, realizar inferências com informações incompletas, como por exemplo, controlar processos industriais e plantas com características não-lineares, comportamento dinâmico (STACH, et al, 2005b), através de experiência e inferência de relações entre as variáveis do processo. No controle inteligente, este tipo de conhecimento pode ser capturado, por exemplo, em um controlador *fuzzy*, possibilitando

a implementação de um controlador automático pode em algumas situações específicas obter tomadas de decisões similares ao de um operador humano (PASSINO; YOURKOVICH, 1997).

A indução, por outro lado, é a forma de inferência lógica que permite obter conclusões genéricas sobre um conjunto particular por meio de exemplos. É caracterizada como o raciocínio que se origina em um conceito específico e o generaliza, ou seja, raciocina da parte para o todo. Na indução, um conceito é aprendido efetuando-se inferência indutiva sobre os exemplos apresentados. Portanto, as hipóteses geradas através da inferência indutiva podem ou não preservar a verdade. Mesmo assim, a inferência indutiva é um dos principais métodos utilizados para derivar conhecimento novo e prever eventos futuros. Apesar de a indução ser o recurso mais utilizado pelo cérebro humano para derivar conhecimento novo, deve ser utilizado com cautela, pois se o número de exemplos for insuficiente, ou se os exemplos não forem bem escolhidos, as hipóteses obtidas podem ser de pouco valor. O aprendizado indutivo é efetuado a partir de raciocínio sobre exemplos fornecidos por um processo externo ao sistema de aprendizado. Deste modo, o aprendizado indutivo pode ser dividido em supervisionado e não-supervisionado.

O aprendizado indutivo pode ser utilizado no projeto de controladores inteligentes do tipo *fuzzy* ou *neuro-fuzzy*, onde seria possível o aprendizado e generalização através de exemplos particulares provenientes da observação do comportamento do processo numa situação dinâmica, ou variante no tempo. Vantagens significativas podem ser obtidas de sistemas que podem aprender com a experiência de tal forma que quando uma situação é encontrada repetidamente, esses sistemas saberão como gerenciar o problema.

1.4 MOTIVAÇÃO

O uso de mapas cognitivos (*Cognitive Maps* - CM), mapas cognitivos *Fuzzy* (*Fuzzy Cognitive Maps* - FCM), e em especial de redes cognitivas dinâmicas (*Dynamic Cognitive Networks* - DCN), proposta dessa Tese para o desenvolvimento de sistemas inteligentes baseados em conhecimento heurístico, permitem a construção de modelos

de um sistema através das duas formas de raciocínio citadas anteriormente. Com essas ferramentas é possível por um lado codificar o conhecimento heurístico sobre o sistema, e por outro incluir funcionalidades computacionais, como por exemplo, capacidade de efetuar ajustes nos parâmetros do modelo a partir dos dados quantitativos do sistema, treinamento e inferência. Os modelos assim construídos refletem as características cognitivas inerentes aos modelos mentais humanos, que os possibilitam controlar sistemas complexos. Assim, a motivação dessa Tese está no desenvolvimento de métodos que possibilitem a criação de modelos estruturados baseados em mapas cognitivos, em especial em DCNs, para a construção de sistemas inteligentes, objetivando realizar tarefas de automação, controle e supervisão de sistemas dinâmicos. Como por exemplo: processos industriais não-lineares que são controlados por operadores e robótica móvel que requer autonomia e capacidade de adaptação devido a mudanças no ambiente.

Pretende-se desenvolver estruturas inteligentes baseadas em DCN que facilitem a construção de sistemas computacionais inteligentes com capacidade de raciocínio para tomadas de decisões dinâmicas por meio de inferências baseadas em conhecimento modelado de forma estruturada. Através de mapas cognitivos, a aquisição do conhecimento é relativamente simples e intuitiva, uma vez que um mapa cognitivo é uma representação simbólica baseada em grafos do conhecimento empírico sobre a natureza e o funcionamento do sistema que se deseja modelar.

Ao longo desta Tese, os FCM e posteriormente as DCN serão considerados como técnicas híbridas do tipo *neuro-fuzzy* já que se baseiam em raciocínio indutivo (abstração de informações imprecisas, aproximadas e ambíguas) de forma similar às técnicas *fuzzy*, e, apresentam capacidade de aprendizado e adaptação a partir de dados históricos do sistema de modo semelhante às redes neurais artificiais.

1.5 OBJETIVO

O objetivo principal desta Tese de doutorado é desenvolver uma metodologia baseada em técnicas de inteligência computacional e princípios de engenharia cognitiva, capaz de modelar um sistema complexo de tomada de decisão em que informações

qualitativas oriundas do conhecimento humano e informações extraídas a partir de observação, comportamento, e principalmente de dados de operação desses sistemas possam ser igualmente consideradas (SCHREIBER; AKKERMANS, 2000). Essa metodologia deve ser abrangente, permitindo a modelagem de diferentes tipos de sistemas complexos. No entanto, a aplicação alvo deste projeto de doutorado é na modelagem dos sistemas inteligentes de automação e controle, principalmente aqueles que requerem a tomada de decisões em face de eventos dinâmicos.

A metodologia proposta será baseada em redes dinâmicas cognitivas que serão construídas em duas camadas (etapas): uma dedicada à modelagem do comportamento dinâmico do sistema e outra relacionada à tomada de decisão e planejamento em face dos objetivos e do desempenho desejado para o modelo.

A metodologia será validada em duas aplicações distintas: a supervisão de processos industriais e em um sistema de navegação autônoma para robótica móvel.

Na primeira aplicação, propõe-se mais especificamente uma nova arquitetura para a supervisão e controle qualitativo de processos industriais, baseada em redes cognitivas dinâmicas. O modelo DCN proposto obedece à hierarquia descrita na figura 2 e realiza várias tarefas dos níveis de execução e coordenação. O ajuste / sintonia do modelo é obtido através de algoritmos de aprendizado baseado em métodos evolutivos (algoritmos genéticos ou nuvem de partículas). O desenvolvimento da arquitetura proposta englobará de um modo geral as seguintes etapas, as quais são comuns no desenvolvimento de sistemas inteligentes: aquisição de conhecimento heurístico sobre a planta e / ou processo, tratamento da informação, modelagem do processo utilizando técnicas cognitivas, análise matemática e raciocínio qualitativo, e refinamento do modelo através de algoritmos de aprendizado a partir de dados históricos e / ou de simulação do processo. A etapa de refinamento poderá ser feita *off-line*, *on-line* ou uma combinação das duas. Finalmente, a etapa de validação que objetiva credenciar o uso do sistema computacional inteligente proposto é feita através da comparação de desempenho com outros sistemas de supervisão.

Deste modo, o modelo baseado em DCN a ser desenvolvido emprega conhecimento humano e dados numéricos do processo e/ou sistema na construção de uma base de conhecimentos capaz de fazer inferências e tomar decisões. O sistema

inteligente assim desenvolvido pode ser aplicado no controle e supervisão de processos industriais que apresentem não-linearidades, incertezas e forte interação entre variáveis, ressaltando-se a capacidade de adaptação desses sistemas, em especial na presença de eventos indesejáveis. Esta capacidade de adaptação dinâmica é uma característica necessária a todos os sistemas de controle e supervisão de processos complexos.

A segunda aplicação aborda o desenvolvimento de um sistema de navegação autônoma, com tomadas de decisões para desvio de obstáculos coleta e alvos (funcionalidade de um sistema de navegação robótica) sintonizados dinamicamente, a ser utilizado em robótica móvel. Desenvolve-se uma arquitetura híbrida (reativa e deliberativa) baseada em DCN, que é capaz de guiar um robô através de um ambiente desconhecido de um ponto inicial até um ponto final, realizando tarefas específicas ao longo da trajetória. O modelo cognitivo é desenvolvido em duas camadas, uma responsável por decisões de baixo nível tais como, por exemplo, vire para a direita se houver um objeto à esquerda, e outra de mais alto nível que determina a realização de tarefas pré-agendadas, tais como manter uma tendência (direção) de trajetória ou virar bruscamente caso apareça um obstáculo inesperado ao longo da trajetória. Diferente do supervisor citado acima, o ajuste / sintonia do modelo para navegação robótica é feito através de uma técnica de aprendizado por reforço, nesse caso um algoritmo *Q-learning* modificado, devido à necessidade de interação entre o robô e o ambiente de navegação, a qual é característica desse tipo de aplicação. As demais etapas de aquisição de conhecimento heurístico sobre o robô e o ambiente de navegação, tratamento da informação, modelagem do processo utilizando técnicas cognitivas, análise matemática e raciocínio qualitativo, e por fim validação do navegador é realizada de forma semelhante ao desenvolvimento do supervisor.

1.6 CONTRIBUIÇÕES

Espera-se com este trabalho, contribuir para a área de sistemas inteligentes com o desenvolvimento de uma nova metodologia baseada em redes cognitivas dinâmicas que facilite a modelagem de sistemas complexos. A metodologia proposta pode ser considerada uma evolução dos mapas cognitivos *fuzzy*, é inovadora e explora as

facilidades de representação de conhecimento por meio de mapas cognitivos, aliado a uma capacidade de adaptação obtida por meios de algoritmos de aprendizado.

A primeira evolução de mapas cognitivos, que são modelos do tipo causa-efeito, para construção de sistemas baseadas em conhecimento em presença de incertezas foi o FCM. Entretanto, esses mapas apresentam limitações ao tentar modelar o comportamento de sistemas dinâmicos, principalmente os aspectos temporais e o conceito de realimentação, os quais são comuns aos sistemas de controle e automação. Em seguida, surgiram as redes cognitivas dinâmicas que diferem dos FCMs não apenas na duração do efeito provocado por uma determinada causa, mas também no modo como a causa gera o efeito. As DCNs têm a capacidade de modelar e gerenciar comportamentos não-lineares dependentes do tempo, muitas vezes em tempo real, porém não existe na literatura uma solução fechada e ainda há um amplo espaço para modificações e atualizações dessa técnica, uma vez que nem todas as relações existentes nos sistemas de automação e controle são do tipo causa-efeito.

Neste contexto, a presente Tese contribui com um novo tipo de DCN, na qual os conceitos e as relações causais são dinamicamente modificados a partir da ocorrência de eventos. A nova DCN comporta diferentes tipos de relações, tais como relações do tipo *fuzzy*, declarações condicionais e do tipo seleção, além das relações de causa-efeito comuns aos mapas cognitivos. Também são incluídos no novo modelo de DCN, novos tipos de conceitos (nível e fator), de acordo com o tipo de variável a ser representada. Com isso é possível modelar a maior parte das relações e comportamentos existentes em um sistema dinâmico não-linear. Para a adaptação e otimização *on-line* das relações causais da DCN são utilizadas técnicas baseadas em aprendizado por reforços e otimização *off-line* métodos evolucionários, tais como algoritmo genético e nuvem de partículas.

Em resumo, a principal contribuição deste trabalho está no desenvolvimento de uma metodologia baseada em redes dinâmicas cognitivas para o desenvolvimento de sistemas inteligentes dinâmicos a partir de conhecimentos heurísticos. Também do ponto de vista do desenvolvimento de novas técnicas computacionais, esta Tese contribui com a utilização de novos métodos que permitem a otimização, adaptação e treinamento de mapas cognitivos, *fuzzy* ou não, e de suas extensões, as redes cognitivas

dinâmicas. Por fim uma terceira contribuição consiste numa proposta de arquitetura para sistemas inteligentes em duas camadas, a qual será validada tanto para a supervisão e controle de processos como no desenvolvimento de um sistema de navegação autônomo para robôs móveis.

Esta arquitetura não é restrita a apenas as duas aplicações apresentadas, ela pode ser aplicada na construção de qualquer tipo de sistema em que se precise modelar a tomada de decisão *on-line* a partir de eventos dinâmicos. Como será demonstrada, a arquitetura baseada em DCN constitui uma ferramenta flexível e robusta, capaz de modelar a imprecisão e incertezas dos sistemas dinâmicos de um modo formal, através de uma modelagem matemática baseada em grafos. Uma das principais vantagens da abordagem proposta é que aquisição e a representação do conhecimento são facilitadas pelo uso de mapas cognitivos, além de gerarem códigos simples de software, capazes de serem facilmente embarcados em hardware dedicado.

1.7 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma:

O capítulo 2 apresenta e fundamenta os Mapas Cognitivos *Fuzzy* (FCM) e por extensão as Redes Cognitivas Dinâmicas (DCN) como ferramentas para modelagem qualitativa, abordando características de construção e vantagens da ferramenta, e discute técnicas de sistemas inteligentes que estão envolvidas nas atividades de modelagem, aquisição de conhecimento, otimização e adaptação dos modelos inicialmente propostos.

O capítulo 3 exemplifica uma aplicação que utiliza o FCM e discute basicamente as etapas iniciais na construção de um sistema inteligente empregando tal técnica. Um exemplo de controle clássico é utilizado para demonstrar as necessidades de ajuste e/ou otimização de um modelo construído a partir de conhecimento humano. Esse capítulo contribui com uma comparação entre as metaheurísticas conhecidas na literatura para treinamento de FCM, e uma proposta de otimização empregando redes neurais com um mapeamento reverso.

Esse capítulo também apresenta uma proposta de controle supervisório empregando FCM, no seu estado da arte, sintonizado dinamicamente por uma Rede Neural Artificial. Conclui com resultados, algumas vantagens, dificuldades, e, principalmente, a necessidade de um nível de hierarquia mais alto para aplicações do FCM em controle de processos, em especial, para tomadas de decisões e inferências em sistemas dinâmicos.

Partes do capítulo 3 foram publicadas no **IFAC-SAFEPROCESS - 3rd International Conference on Safety and Security Engineering**, Barcelona, 2009, e no **IX Simpósio Brasileiro de Automação Inteligente – SBAI**, Brasília, 2009.

O capítulo 4 propõe uma arquitetura baseada em Rede Dinâmica Cognitiva a qual é construída como um FCM direcionado a eventos. Esta caracterização das DCNs como um FCM dinâmico, cujos relacionamentos causais se modificam na presença de eventos, é uma proposta original da presente Tese. De fato, neste trabalho considera-se uma DCN como um ED-FCM (*Event Driven-Fuzzy Cognitive Maps*). Esse capítulo também endereça algumas das dificuldades encontradas no desenvolvimento de modelos baseados em mapas cognitivos. Essas dificuldades foram detectadas nos estudos iniciais e primeiros modelos desenvolvidos para esta Tese. A maior parte desses problemas é também descrita na literatura, e são relatados neste capítulo a fim de ilustrar a evolução da técnica, permitindo uma melhor contextualização da metodologia proposta. Por fim, conclui-se com um algoritmo descrevendo os passos ou etapas necessários a construção das redes cognitivas dinâmicas (DCN) e um exemplo de aplicação para estratégia de times de futebol de robôs.

O capítulo 5 estende a arquitetura proposta no capítulo 4, propondo uma nova metodologia para construção de sistemas inteligentes, e desenvolve um sistema supervisório baseado nesta metodologia, o qual corresponde a uma rede cognitiva dinâmica. Algumas funcionalidades implementadas do sistema supervisório empregam conhecimento qualitativo no gerenciamento dinâmico das mudanças de *set-point* para os controladores PID no nível regulatório da planta. O sistema supervisório desenvolvido é aplicado no monitoramento de um processo de fermentação, o qual funciona em modo de operação normal e com falhas. Este capítulo é objeto de um artigo (versão final já

aceita e entregue) que será brevemente publicado na revista **Controle & Automação**, da Sociedade Brasileira de Automática.

O capítulo 6 apresenta o desenvolvimento de um sistema de navegação robótica baseado em DCN, desde a aquisição de dados até a validação através de simulações e com um robô móvel real. Partes desse capítulo foram publicadas no **XVIII Congresso Brasileiro de Automática** (CBA2010) e outra parte está em período de avaliação (segunda revisão) pela revista **Applied Intelligence** (Springer-Verlag).

O capítulo 7 apresenta e discute os resultados teóricos e práticos obtidos, resalta as principais contribuições da Tese e conclui com sugestões para continuação dos trabalhos.

2 FUNDAMENTOS DE CONTROLE INTELIGENTE

2.1 INTRODUÇÃO

Este capítulo apresenta alguns paradigmas e conceitos necessários ao desenvolvimento de sistemas inteligentes. Apresenta também, de forma resumida, as principais áreas de conhecimento envolvidas na modelagem através de redes cognitivas dinâmicas, em especial, sistemas baseados em conhecimento, mapas cognitivos, grafos, sistemas *fuzzy*, redes neurais artificiais e sistemas híbridos, com ênfase nos sistemas *neuro-fuzzy*.

2.2 SISTEMAS BASEADOS EM CONHECIMENTO

Os Sistemas Baseados em Conhecimento (SBC) têm sido desenvolvidos em vários domínios de aplicação para auxiliar o homem na resolução de problemas que envolvem tarefas de conhecimento intensivo, tais como: controle, planejamento, diagnóstico, conselho, configuração, projeto, avaliação, decisão, etc. (SCHREIBER; AKKERMANS, 2000). O núcleo de um SBC é formado pelo conhecimento especialista do domínio da aplicação e por estratégias de raciocínio. Essas estratégias compõem os chamados métodos de resolução de problema (*problem-solving methods – PSM*). Um PSM é uma especificação do processo de raciocínio de um SBC, independente do domínio da aplicação, e tem um papel importante no ciclo de vida de desenvolvimento de SBCs. Os PSM podem ser usados como modelos que guiam o processo de aquisição de conhecimento e modelagem do problema, possibilitando o desenvolvimento de aplicações robustas e manuteníveis através da reutilização de seus elementos.

Os fatores a seguir são algumas das características de um SBC, em que o conhecimento específico é um requisito indispensável na resolução de problemas complexos:

- Existem problemas em que os seres humanos têm melhor desempenho do que as máquinas.

- Para que a máquina resolva este tipo de problema, ela deve ter o conhecimento que o ser humano tem para resolvê-los.
- Os computadores são melhores que os seres humanos em relação à velocidade e consistência com que executam certas tarefas.
- Para que os computadores desempenhem tarefas para as quais o ser humano é melhor, devem possuir conhecimento especializado comparável ao que um especialista humano possui.

A seguir é apresentada uma breve introdução sobre metodologias de inteligência computacional que permeiam a implementação de sistemas baseados em conhecimento, com especial atenção aos métodos empregados nessa Tese: os Mapas Cognitivos *Fuzzy* e as Redes Cognitivas Dinâmicas.

2.2.1 Regras Como Representação de Conhecimento

Uma das maneiras de representar conhecimento é pelo uso de regras que expressem o que deve acontecer ou o que realmente acontece diante de certas condições. Regras são geralmente expressas na forma de declarações condicionais do tipo SE-ENTÃO, tal como: SE *A* ENTÃO *B*.

Sistemas baseados em regras são sistemas computacionais que usam regras para fornecerem diagnósticos ou fornecer recomendações ou uma linha de ação em uma situação particular, ou até mesmo para solucionar um problema específico (COPPIN, 2010). Um sistema baseado em regras contém os seguintes componentes:

1. Uma base de regras (também chamada base de conhecimento);
2. Uma base de dados e/ou fatos;
3. Um interpretador ou mecanismo de inferência.

Deste modo, a base de conhecimento é formada por um conjunto de regras que representam o conhecimento modelado. A base de dados representa as entradas do sistema que são usadas para obter conclusões ou provocar ações. O interpretador, ou mecanismo de inferência, é a parte do sistema que controla o processo de obtenção de conclusões.

Outro modo de se representar conhecimento de forma estruturada é através de mapas mentais ou cognitivos, onde são modeladas relações do tipo causa e efeito entre conceitos (variáveis) de um sistema. Esses mapas mentais são representados matematicamente através de grafos. Um exemplo desse tipo de modelo são os mapas cognitivos *fuzzy*, e as redes cognitivas dinâmicas. Os mapas cognitivos permitem uma interpretação mais próxima ao raciocínio humano e realizam inferências ou tomadas de decisões através de operações matriciais que “evoluem” um grafo desde o seu estado inicial até um ponto de equilíbrio ou um ciclo limite (STYLIOS; GROUMPOS, 1998).

2.3 METODOLOGIAS DA INTELIGÊNCIA COMPUTACIONAL

O termo “inteligência computacional” foi proposto por Zadeh (1968) para denominar um conjunto de métodos que visam explorar a tolerância às falhas, a imprecisão e a incerteza dos sistemas reais, proporcionando robustez e solução de baixo custo para aplicação de controle, reconhecimento de padrões, aproximação de funções, entre outras, em diversas áreas de conhecimento. De acordo com a literatura, as principais metodologias de inteligência computacional são:

- Sistemas *fuzzy* os quais utilizam modelos relacionais, sistemas baseados em regras do tipo Mamdani e Takagi-Sugeno-Kang,
- Raciocínio probabilístico (teoria de Dempster-Shafer)
- Redes Bayesianas;
- Redes neurais artificiais;
- Computação evolutiva (algoritmos genéticos, programação evolutiva, programação genética e estratégia evolutiva);
- Agentes Inteligentes
- Sistemas híbridos que são uma combinação de uma ou mais técnicas citadas acima.

Algumas partes, definições e conceitos desses métodos serão utilizados ao longo deste trabalho e por isso serão brevemente descritos a seguir.

2.3.1 Sistemas *Fuzzy*

A teoria de conjuntos *fuzzy* tem sido empregada com sucesso para exprimir conhecimento impreciso e resolver problemas em muitas áreas de conhecimento, em especial quando o modelo matemático é difícil de ser obtido ou é incompleto. Deste modo, a possibilidade de descrição linguística do modelo possibilita o aproveitamento do conhecimento heurístico dos operadores e facilita o desenvolvimento de soluções para diferentes áreas de conhecimento, em especial para a área de controle de processos (PASSINO; YOURKOVICH, 1997), (ZHANG, LIU, 2006). As leis de controle *fuzzy* podem ser definidas através de regras linguísticas em uma base de regras. Usando os conceitos *fuzzy*, as regras podem ser processadas através de um formalismo matemático baseado na teoria *fuzzy*. Para se modelar um sistema utilizando teoria *Fuzzy*, é necessário além da definição da base de regras, determinarem o universo de discurso em que são definidas as variáveis do problema, identificar as partições *fuzzy* desse universo associadas a cada variável e as regras semânticas e sintáticas que definem essas variáveis, bem como o mecanismo de inferência que instancia a base de regras.

A estrutura de universo de discurso, o método de fuzzificação, o formato da base de regras, a máquina de inferência e o método de defuzzificação propostos por King e Mamdani (1977) formam um dos mais utilizados sistemas de inferência *fuzzy* utilizados na literatura. Quando bem aplicado, este sistema de inferência pode contribuir para simplificação e aumento da velocidade de processamento e robustez do controlador, possibilitando decisões rápidas e coerentes num ambiente de incertezas, em situações críticas como, por exemplo, situações de falhas ou conflitos (KEMPOWSKY; SUBIAS; AGUILAR-MARTIN, 2006). A figura 3 ilustra em blocos, o controlador lógico *fuzzy* (*Fuzzy Logic Controller* – FLC) proposto por King e Mamdani (1977):

De acordo com a figura 3, e, segundo Figueiredo et al. (1993), as etapas de projeto para construção de um controlador lógico *Fuzzy* consiste em:

1. Definir os universos de discurso das variáveis linguísticas do sistema através da observação do comportamento do sistema e/ou processo, como por exemplo, observando o erro e a variação do erro.

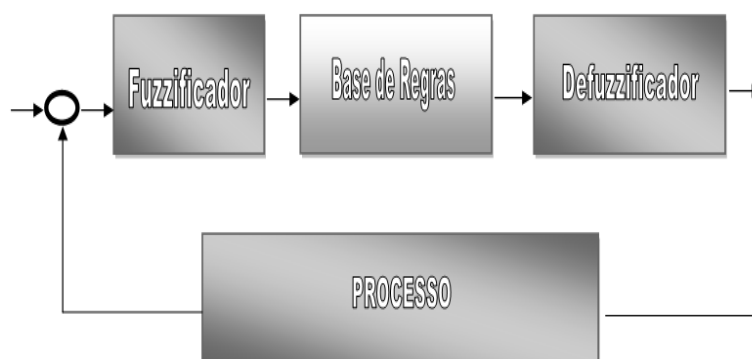


Figura 3 – Modelo do Controlador Lógico Fuzzy
Fonte: Adaptado de King e Mamdani (1977)

2. Definir o número de termos primários e graus de pertinência dos conjuntos fuzzy que representam cada termo;
3. Determinar as regras que formam o algoritmo de controle que compõem o mecanismo de inferência ou motor de inferência; o mecanismo de inferência emula as tomadas de decisões dos especialistas interpretando e aplicando o conhecimento modelado (PASSINO; YOURKOVIC, 1997).
4. Definir os parâmetros de projeto, como método de inferência, método de fuzzificação e defuzzificação e atuação do controlador.

O propósito da utilização de um controlador FLC (*Fuzzy Logic Controller*) em um determinado processo é capturar o conhecimento de um operador e convertê-lo em um modelo para tomadas de decisões de controle, por exemplo. (COELHO; ALMEIDA; COELHO, 2003). Esta conversão em controladores do tipo FLC utiliza variáveis linguísticas e regras. Além disso, estes controladores permitem modelar a incerteza dos processos, possibilitando desenvolver sistemas mais robustos, que podem ser utilizados em processos complexos (PASSINO; YOURKOVICH, 1997). Um controle do tipo FLC para operar no chão de fábrica deve ter as seguintes funcionalidades:

- Controlar e operar automaticamente processos complexos não-lineares multivariáveis com desempenho pelo menos equivalente aos operadores.

- Respeitar as especificações e restrições operacionais.
- Ser robusto e operar em tempo real (PADILHA, 2001).

O controle *fuzzy* tem se mostrado também uma alternativa viável ao controle clássico para processos com parâmetros variantes no tempo, não-lineares e com informações imprecisas. Para esses casos, a aplicação de um controlador *Fuzzy* frequentemente oferece um desempenho igual ou superior aos controladores adaptativos, do tipo preditivo ou por modelo de referência (CALLAI et al., 2007). Entretanto, uma das desvantagens do uso de controladores *fuzzy*, é o problema da dimensionalidade o qual gera uma explosão na quantidade de regras à medida que o sistema se torna cada vez mais complexo. O número de funções de pertinência e regras de inferências aumenta com a complexidade do modelo devido à natureza física do sistema e também com a necessidade de tomadas de decisão e inferências necessárias para um bom desempenho do sistema de controle.

Para o controle supervisório, os sistemas baseados em lógica *fuzzy* podem fornecer uma solução eficiente através do desenvolvimento de controladores com base na experiência de especialistas e/ou operadores em vez de utilizar modelos matemáticos complexos. Uma das tarefas do supervisor é a de, por exemplo, gerar *set-points* para os controladores nos níveis inferiores da planta ou ainda gerenciar estratégias de recuperação da campanha em presença de perturbações. Essas atividades podem ser implementada através de supervisores *fuzzy* (ARRIAGA-DE-VALLE; DIECK-ASSAD, 2006).

Nessa Tese um sistema de navegação robótico baseado em FLC será usado no capítulo 6 como referência para a comparação de desempenho do navegador baseado em DCN. Também é proposta a utilização de regras *fuzzy* para cálculo dinâmico dos valores de pesos (arcos) das redes dinâmicas cognitivas apresentadas nos capítulos 5 e 6. De fato, técnicas *fuzzy* são um dos métodos possíveis de serem utilizados para atualização *on-line* da arquitetura baseada em DCN proposta no capítulo 4 desta Tese.

2.3.2 Redes Neurais Artificiais

Redes Neurais Artificiais são técnicas computacionais inspiradas na estrutura neural de organismos inteligentes, e que adquirem conhecimento através da experiência. Uma rede neural é uma estrutura de processamento de informação distribuída paralelamente na forma de um grafo direcionado, com algumas restrições e definições próprias. Os nós deste grafo são chamados elementos de processamento, e suas arestas são conexões, que funcionam como caminhos de condução instantânea de sinais em uma única direção, de forma que seus elementos de processamento podem receber qualquer número de conexões de entrada. Estas estruturas podem possuir memória local, e também possuir qualquer número de conexões de saída desde que os sinais nestas conexões sejam os mesmos. Portanto, estes elementos têm na verdade uma única conexão de saída, que pode dividir-se em cópias para formar múltiplas conexões, sendo que todos carregam o mesmo sinal.

De forma geral, a operação de uma rede se resume em:

1. Sinais são apresentados à entrada;
2. Cada sinal é multiplicado por um peso que indica sua influência na saída da unidade;
3. É feita a soma ponderada dos sinais que produz um nível de atividade;
4. Se este nível excede um limite (*threshold*) a unidade produz uma saída;

A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a generalização necessária para executar adequadamente o processo desejado a partir de outro conjunto de dados. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais. A figura 4 mostra a arquitetura clássica de uma rede multicamada, formada pelos neurônios das camadas de entrada, intermediária e de saída interconectados.

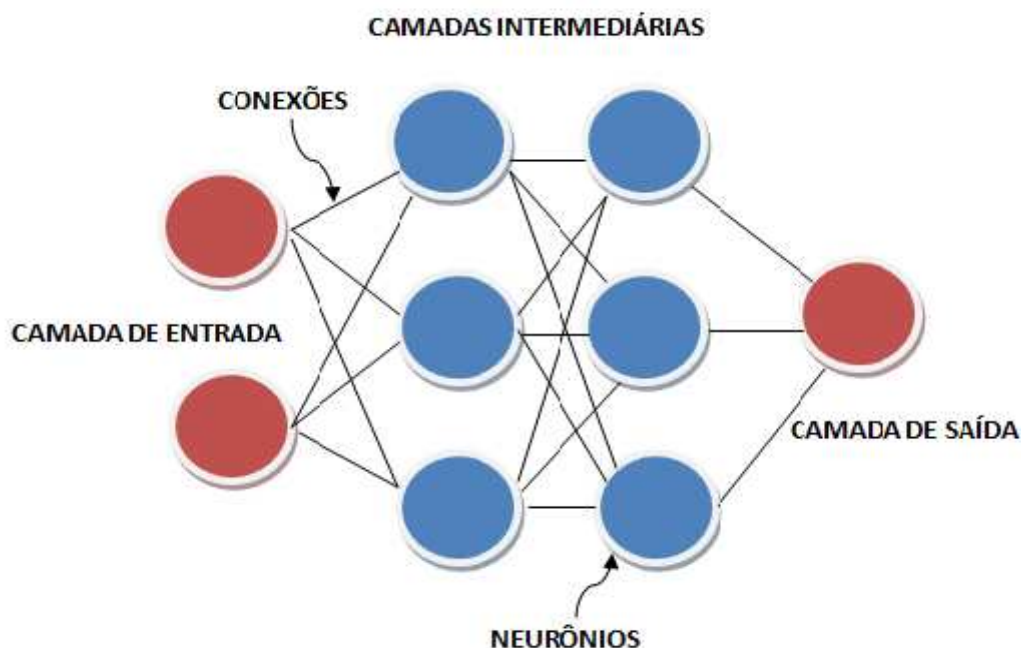


Figura 4 – Arquitetura de uma Rede Neural
Fonte: Adaptado de Haykin (2000)

Outra importante característica, também inspirada no cérebro, é a capacidade de auto-organização, ou plasticidade, ou seja, através de um processo de aprendizado, é possível alterar-se os padrões de interconexão entre seus elementos. As redes neurais do tipo *perceptron* multicamadas são aproximadores universais e assim são frequentemente empregadas em modelagem de sistemas não lineares (HAYKIN, 2000). Na área de controle de processos, arquiteturas baseadas em redes neurais artificiais têm sido usadas em controle adaptativo por modelo inverso, controle preditivo, controle por modelo de referência neural ou controle adaptativo neural, entre outros (TAI; RYACIOTAKI-BOUSSALIS; HOLLAWAY, 1991).

Nessa Tese, propõe-se o uso de RNA como método de adaptação de um FCM, conforme será apresentado no capítulo 3. Uma rede multicamada é treinada para aprender o mapeamento reverso de um FCM e em seguida é usada para adaptar os valores das relações causais do modelo para se atingir dois *set-points* desejados em um controle de nível e de massa de uma mistura líquida.

2.3.3 Computação Evolutiva

Na natureza, os indivíduos mais bem adaptados à competição por recursos, sobrevivem. A adaptação às variações do ambiente é imprescindível para a sobrevivência dos indivíduos e das espécies. Com base na evolução biológica, John Holland publicou o livro “*Adaptation in Natural and Artificial Systems*” (HOLLAND, 1975), hoje considerado como a referência básica sobre os algoritmos genéticos (AGs). Desde então, esses algoritmos têm sido aplicados com sucesso em diversos problemas do mundo real de busca e otimização (STACH, 2006). Os AGs operam com uma busca estocástica, onde o espaço de busca é percorrido baseando-se em transições probabilísticas (GOLDBERG, 1989). Existem quatro diferenças básicas entre os AGs e os métodos tradicionais de busca e otimização. De um modo geral, os algoritmos genéticos possuem as seguintes características (GOLDBERG, 1989).

- Empregam uma codificação do conjunto de parâmetros e não os próprios parâmetros;
- Operam em uma população e não em um único ponto;
- Utilizam informações do valor da função custo e não informações sobre derivadas da função de otimização;
- Utilizam regras de transição probabilísticas e não determinísticas.

Os AGs são compostos por uma população de indivíduos e um conjunto de operadores sobre a população. De acordo com as teorias evolucionárias, segundo as quais foi desenvolvido o AG, os elementos mais bem adaptados ao seu meio têm maior probabilidade de sobreviver e de se reproduzir, transmitindo o seu material genético para as novas gerações. Um AG é composto basicamente por quatro etapas:

- Geração de uma população inicial;
- Avaliação de cada um dos elementos da população;
- Seleção dos melhores elementos da população;
- Manipulação genética, através dos operadores de cruzamento e mutação, de forma a criar uma nova população.

Após a realização destas etapas, um *loop* é realizado retornando ao passo 2, enquanto um determinado critério de parada não for atingido. Ressalta-se que a

aplicação de Algoritmos Genéticos requer conhecimento prévio do problema abordado (STACH, 2006, 2006b), e, que, determinar os parâmetros iniciais como função de avaliação, número de indivíduos na população inicial, porcentagens de operadores, nem sempre é trivial.

Deste modo, ajustes empíricos, técnicas de refinamento, experiência no uso da ferramenta, agregados ao conhecimento do problema conduzem a resultados satisfatórios. Outro fator relevante está na criação de diferentes tipos de indivíduos, e na utilização de operadores que garantam que as sucessivas gerações sempre contenham soluções factíveis dentro do espaço de busca do problema.

O maior problema da utilização dos métodos que empregam heurísticas de busca, como é o caso de algoritmos genéticos, é que não existe nenhuma garantia da obtenção de um ponto de mínimo (ou máximo) global. Assim um algoritmo de busca, seja ele um AG ou outra metaheurística, pode convergir para um ponto de ótimo local da função de avaliação, que não necessariamente corresponde à solução ótima do problema.

Outra técnica de otimização inspirada na natureza que está relacionada à computação evolutiva é a inteligência coletiva. Nos métodos baseados em inteligência coletiva, os diversos membros de um determinado grupo contribuem com as suas próprias experiências para o grupo, tornando-o mais forte perante os outros, ou seja, as capacidades do grupo são superiores às capacidades que os membros podem atingir individualmente. Um dos conceitos de inteligência coletiva é a propriedade de os sistemas compostos de elementos não inteligentes exibirem comportamentos coletivos inteligentes, conforme apresentado em Kennedy e Eberhart (1995). Os mais conhecidos métodos baseados em inteligência coletiva são as Colônias de Formiga e a otimização por nuvem de partículas (*Particle Swarm Optimization*), (CLERCAND; KENNEDY, 2001).

No desenvolvimento do capítulo 3 a otimização dos pesos de um FCM será realizada através de AG e PSO. Os resultados obtidos com estes algoritmos serão comparados com os resultados obtidos com um FCM sintonizado por RNA.

2.3.4 Agentes Inteligentes

Agentes são sistemas computacionais que habitam um ambiente complexo e dinâmico, percebendo-o por meio de sensores e atuando autonomamente sobre este ambiente através de atuadores, a fim de realizar uma série de metas e tarefas para as quais foram projetados (MAES, 1995). De um modo geral, os agentes inteligentes executam continuamente três funções: percebem as condições dinâmicas do ambiente, agem alterando as condições do ambiente e raciocinam de modo a interpretar percepções, resolver problemas, fazer inferências e determinar ações (GUDWIN, 2002).

De fato, agentes são entidades autônomas que são especialistas na execução de uma determinada tarefa, tendo a capacidade de perceber o ambiente em que atuam tomar decisões sobre informações obtidas deste ambiente e executar alguma tarefa (ação) como resultado desta interação com o ambiente.

Atualmente, os agentes autônomos vêm despertando crescente interesse junto à comunidade científica. Esse interesse está fortemente relacionado às técnicas utilizadas para investigar essa classe de sistemas, onde se faz uma integração do agente ao ambiente. Essa integração decorre do fato do agente autônomo atuar sobre si próprio, o que caracteriza sua autonomia. De um modo geral, um agente é uma entidade capaz de realizar alguma tarefa para auxiliar um usuário humano. Um agente não-autônomo, em contraposição, deverá ser estático ou sofrer uma atuação de algum agente externo (COPPIN, 2010).

Algumas propriedades encontradas em certas classes de sistemas autônomos permitem caracterizá-los como sistemas inteligentes. Por exemplo:

- Capacidade de adquirir conhecimento pela interação com o ambiente;
- Capacidade de adaptar seu comportamento com base em suas informações sensoriais, utilizando técnicas de adaptação e/ou aprendizado;
- Capacidade de encontrar soluções factíveis (e, muitas vezes, ótimas), que atendam simultaneamente a múltiplos objetivos (possivelmente conflitantes);

- Capacidade para operação em condições adversas tais como: ausência de um conjunto de informações completo para um planejamento prévio de seu comportamento, imprevisibilidade na interação com o ambiente (possivelmente de topologia desconhecida), ruído nos sensores e atuadores;

Este conjunto de propriedades, que caracterizam um agente autônomo como inteligente, cria restrições importantes em seu processo de programação e controle, as quais vêm sendo objeto de pesquisa na literatura (COLOMBETTI; DORIGO; BORGHI, 1996). Dado um sistema autônomo pertencente a um ambiente complexo, imprevisível e possivelmente dotado de múltiplos sistemas autônomos, o que se busca é superar a impossibilidade prática de se programar e controlar este sistema autônomo utilizando técnicas de inteligência artificial convencional (BROOKS, 1990). Também pode-se citar que: sistemas de navegação autônomos devem ser capazes de definir uma sequência de ações a serem tomadas por robôs móveis (FIGUEIREDO, 1999), dotados de um conjunto limitado de sensores, quando expostos a um ambiente externo supostamente desconhecido e tendo que atender simultaneamente a um elenco de objetivos previamente especificados. De modo resumido, um agente autônomo tem capacidade de agir e tomar decisões, independente de um programador ou do usuário agente (COPPIN, 2010).

Uma das contribuições dessa Tese é o desenvolvimento de uma arquitetura baseada em DCN, a qual modela os diversos comportamentos e a capacidade de tomada de decisões para agentes autônomos. Deste modo o próximo item fundamenta agentes autônomos e arquiteturas aplicadas à navegação robótica, em especial a arquitetura de subsunção de Brooks (1986).

2.3.5. Arquiteturas de Agentes

Um tópico importante relacionado à tecnologia de agentes diz respeito às diferentes arquiteturas que podem ser idealizadas para a sua implementação. Entende-se por uma arquitetura de agentes como o conjunto de especificações e técnicas utilizadas

para a definição funcional dos agentes. Nesta seção, descreveremos alguns dos conceitos relacionados a arquiteturas de agentes, em especial as arquiteturas reativas (que seja capaz de reagir aos eventos que ocorrem no ambiente sem envolver raciocínios complexos). E deliberativas (que contém um modelo simbólico do mundo desenvolve plano e tome decisões).

Muitos pesquisadores sugerem que nem um enfoque completamente deliberativo, nem um completamente reativo são apropriados para a construção de agentes autônomos (GUDWIN, 2002). Discute-se então, o caso dos sistemas híbridos, abordagem que associam ambos os enfoques. Um enfoque óbvio é construir um agente com dois (ou mais) subsistemas, um deles deliberativo e outro reativo. Com frequência o componente reativo é colocado com algum tipo de precedência sobre o componente deliberativo (FRACASSO; COSTA, 2005). Desta forma, ele pode oferecer uma resposta rápida a eventos importantes do ambiente. Este tipo de estrutura leva naturalmente à idéia de uma arquitetura em camadas. Nesta arquitetura, os subsistemas de controle de agentes são organizados dentro de uma hierarquia, na qual as camadas mais altas lidam com as informações em níveis crescentes de abstração. Assim, por exemplo, as camadas mais baixas poderiam mapear diretamente os dados dos sensores para as saídas dos atuadores sem nenhum tratamento prévio, enquanto as camadas superiores fazem o tratamento das tarefas de longo prazo. O principal problema nesta arquitetura é que o tipo de estrutura de controle deverá ser embutido nos subsistemas de agentes para manipular as interações entre as diferentes camadas.

Do ponto de vista de tomada de decisão e inferência a arquitetura de controle empregada em navegação robótica inteligente é a de **Subsunção**, proposta por Brooks (1986). As tomadas de decisões e inferências neste tipo de arquitetura funcionam em processamento paralelo e é capaz de tomar decisões relacionadas a comportamentos humanos como “raciocinar sobre o comportamento de objetos” e “explorar” o ambiente. A figura 5 mostra a arquitetura para agentes inteligentes proposta por Brooks (1986). Ressalta-se que existem outras propostas conhecidas na literatura (GUDWIN, 2002), entretanto, optou-se por se utilizar a arquitetura de Subsunção, essa arquitetura foi originalmente idealizada para a utilização em robôs moveis autônomos capazes de

realizar o processamento de informações complexas em tempo real (CHONG, TAN, NG, 2007). Algumas características são citadas:

- Utiliza uma hierarquia que prioriza funções de acordo com a heurística de operação do processo.
- É uma arquitetura baseada na ação-situação. As decisões dos agentes são realizadas através de um conjunto de comportamentos direcionados a tarefas a cumprir.
- Tem inspiração na neurobiologia, em especial, na sinergia que existe entre a sensação e a atuação de insetos.

A arquitetura é utilizada para tomadas de decisões em que há um mapeamento direto da situação a ação, é disposta em camadas hierárquicas, e foi projetada para robôs cujo mecanismo de controle não é centralizado.

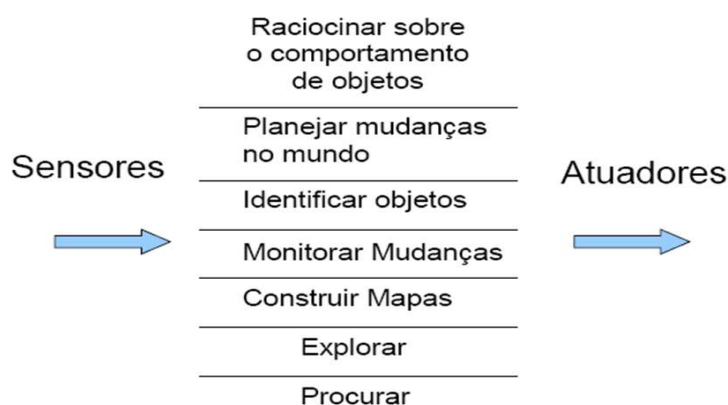


Figura 5 - Arquitetura de Subsunção.
Fonte: Adaptado de Brooks (1986)

Deste modo, cada camada é capaz de inibir o comportamento das camadas abaixo dela. A figura 6 mostra um exemplo de funcionalidades com três camadas.

Os veículos auto-guiados que utilizam esta arquitetura possuem grandes expectativas quanto às possibilidades de aplicação prática, inclusive industriais. Muito embora seres vivos com capacidade motora não encontrem grandes dificuldades em gerar trajetórias eficazes e tomar decisões nos mais variados ambientes, dotar sistemas artificiais com tais habilidades não é uma tarefa elementar (COLOMBETTI; DORIGO; BORGHI, 1996).



Figura 6 - Arquitetura Subsunção de 3 Camadas
Fonte: Adaptado de Coppin (2010)

Nesta Tese a arquitetura de subsunção em camadas será utilizada no capítulo 6 para o desenvolvimento de um sistema de navegação autônomo.

2.3.6 Aprendizagem Por Reforço

Segundo Haykin (2000), há três paradigmas de aprendizado: supervisionado, não-supervisionado e aprendizado por reforço.

Em um modelo de aprendizagem por reforço, um agente aprende a otimizar sua interação com o ambiente por tentativa e erro, como mostrado na figura 7. O ambiente pode ser parcialmente ou completamente observável e, através de sensores (em aplicações robóticas), o agente identifica os estados “ S ” atual do sistema. Em geral, o agente pode modificar o ambiente, utilizando assim seus atuadores para efetuar ações “ a ” e mudar de estado. Para cada ação tomada, o agente pode receber uma recompensa “ r ”, que será utilizada para adaptar o seu comportamento. Assim, uma função de reforço $r(S(t), a(t))$ é um mapeamento de estados/ações para recompensas, num instante de tempo t . À medida que se dá à interação com o ambiente, o agente aprende a tomar

decisões em determinadas situações que o levam a maximizar a soma de prêmios recebidos (SUTTON; BARTO, 1998).

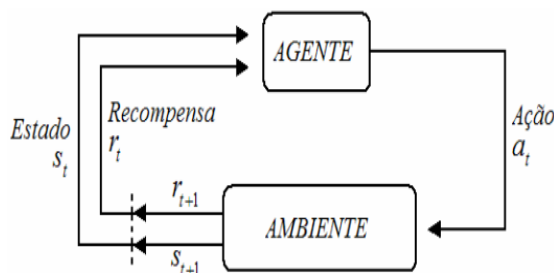


Figura 7 - Modelo Padrão de Aprendizagem Por Reforço
Fonte: adaptado de Sutton e Barto (1998).

O algoritmo *Q-learning* é uma técnica comumente utilizada em aprendizagem por reforço, em especial na área de navegação robótica (HOFFMANN; SILVA 2005). A vantagem do *Q-learning* é a de realizar apenas um mapeamento do par estados/ações para valores Q , substituindo a função de valores e a função de transição de estados. Esse mapeamento é conhecido como função- Q , a qual define um novo valor- Q para cada ação possível a partir de um dado estado (SUTTON; BARTO, 1998), de acordo com a equação (1).

$$Q(s, a, t+1) = Q(s, a, t) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a, t) \right] \quad (1)$$

Essa equação será explicada e adaptada para sintonia (ajuste fino) dinâmica das relações causais de uma DCN. Neste contexto, essa técnica de aprendizado por reforço baseado é utilizada nesta Tese (capítulo 6) para ampliar a capacidade de adaptação do robô (agente) móvel através da interação entre o robô e o ambiente no sistema de navegação autônoma.

2.3.7 Sistemas Híbridos

O conceito de sistema híbrido inteligente é bastante amplo e pode englobar diferentes tipos de abordagens para aquisição de conhecimentos. De uma maneira mais

geral, pode-se dizer que, todo o sistema que integre dois ou mais métodos diferentes para a solução de um problema é um sistema híbrido.

Há várias formas de se definir sistemas híbridos. A mais simples entende os sistemas híbridos como sendo aqueles que utilizam mais de uma técnica inteligente, beneficiando-se em consequência da combinação das vantagens de cada uma delas. Deste modo, dois ou mais sistemas híbridos podem ser combinados de diferentes formas (SOUZA, 1999).

Híbrido sequencial

Este modelo representa o grau mais fraco de integração possível entre as duas técnicas e consiste de dois subsistemas ligados em série. Como exemplo, podemos citar um modelo *neuro-Fuzzy* em que um sistema *Fuzzy* fornece entradas para uma rede neural como mostra a figura 8.

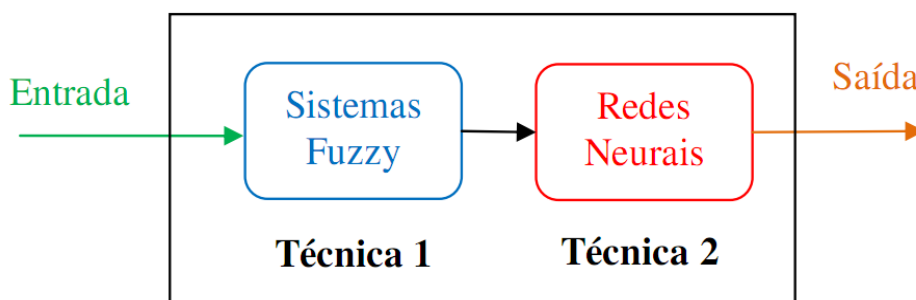


Figura 8 – Exemplo híbrido-sequencial
 Fonte: Adaptado de Coppin (2010)

Híbrido auxiliar

Neste modelo, um subsistema é auxiliado por outro subsistema inteligente. O grau de integração é maior que no caso anterior. Exemplo: uma rede neural que invoca um algoritmo genético para a otimização de seus pesos ou de sua estrutura.

Híbrido incorporado

Representa o grau de integração mais elevado, não havendo separação visível entre os subsistemas. Pode-se dizer que o primeiro sistema contém o segundo ou vice-versa. Um bom exemplo é o sistema *neuro-fuzzy ANFIS* (*Adaptive Network-Based Fuzzy Inference System*), no qual a inferência *fuzzy* é implementada segundo uma

estrutura de rede neural. Entre os modelos híbridos mais populares estão os sistemas, neuro-genéticos, neural-estatísticos, *fuzzy*-genéticos e alguns sistemas *neuro-fuzzy*.

2.3.7 Sistemas Híbridos Neuro-*Fuzzy*

Os sistemas híbridos mais utilizados em controle de processos são os sistemas *neuro-fuzzy* que combinam a capacidade de aprendizado das redes neurais artificiais com o poder de interpretação linguística dos sistemas de inferência *fuzzy*. A maior parte destes sistemas é do tipo híbrido incorporado em que a associação de neurônios com lógica *fuzzy* permite aliar as características dos dois métodos: a capacidade de aprendizado das RNA e a flexibilidade e interpretabilidade dos sistemas *Fuzzy* (KOSKO, 1992; SOUZA, 1999).

O interesse por esses sistemas deve-se principalmente aos seguintes fatores:

- Aplicabilidade dos algoritmos de aprendizado desenvolvidos para redes neurais;
- Possibilidade de promover a integração de conhecimentos (implícito e explícito);
- Possibilidade de extração de conhecimento, sob o formato de regras *Fuzzy*, a partir de um conjunto de dados.

Várias propostas são conhecidas na literatura para desenvolvimento de modelos *neuro-fuzzy*: *ANFIS* (JANG, 1993), *FuzzyARTMAP*, *FuzzyCOPE*, *NEFCONNEFCLASS*, *NEFPROX*, e *FUNEGEN* (LIN; LEE, 1996).

Dos sistemas citados acima, o *ANFIS* (*Adaptive Network-Based Fuzzy Inference System*), em especial, tem sido aplicado para controle de processos industriais (SHAW; SIMÕES, 1999). No entanto, outro tipo de sistema híbrido *neuro-fuzzy* que começa a ter aplicabilidade em diversas áreas de conhecimento, como será ilustrado pelo levantamento bibliográfico apresentado a seguir, são os mapas cognitivos *fuzzy* e suas extensões as redes dinâmicas cognitivas.

2.4. MAPAS COGNITIVOS *FUZZY*

Os mapas cognitivos foram inicialmente propostos por Axelrod (1976) para representar palavras, idéias, tarefas ou outros itens ligados a um conceito central e dispostos radialmente em volta deste conceito. São diagramas que representam conexões entre porções de informação sobre um tema ou tarefa. Os elementos são arranjados intuitivamente de acordo com a importância dos conceitos. Eles são organizados em grupos, ramificações ou áreas.

Axelrod (1976) desenvolveu também um tratamento matemático por meio de operações com matrizes e baseado na teoria de grafos, para seus mapas cognitivos, nos quais crenças ou afirmações a respeito de um domínio de conhecimento limitado são expressas por meio de palavras ou expressões linguísticas, interligadas por relações simples de causa e efeito (causa/não-causa).

Estes mapas podem ser considerados um modelo matemático da “estrutura de crenças” de uma pessoa ou grupo, que permitem inferir ou predizer as consequências que esta organização de idéias causa no universo representado. Este modelo matemático foi adaptado para inclusão de incertezas através da lógica *Fuzzy* por Kosko (1986), gerando os mapas cognitivos *Fuzzy*. À semelhança dos mapas cognitivos originais, os FCM são grafos direcionais (dígrafos), em que os valores numéricos são variáveis ou conjuntos *fuzzy*. Os “nós” destes grafos são conceitos linguísticos, representados por conjuntos *fuzzy* e cada “nó” é associado a outros através de conexões. A cada uma dessas conexões é associado um peso numérico, que representa uma variável *Fuzzy* relacionada com o nível de causalidade entre os conceitos, como mostra a figura 9.

Esses mapas conceituais, *Fuzzy* ou clássicos, representam o conhecimento individual ou de um grupo de especialistas, através dos valores de seus conceitos e dos pesos das suas relações. O conhecimento explícito é representado através das atribuições de valores a conceitos e relações causais, e o conhecimento implícito é representado através da estrutura do mapa. No entanto, permanece a dificuldade de transformar o conhecimento em modelos de comportamento humano. Na construção dos mapas conceituais esta dificuldade se reflete principalmente na definição da semântica e do relacionamento entre os conceitos (PACHECO, 2005). Além desta

difficuldade de construção, a aquisição de dados deve ser bastante criteriosa, no sentido de se transformar dados em informação consistente para o mapa cognitivo.

Os mapas cognitivos *Fuzzy* usam uma representação simbólica pra descrever e modelar um sistema, o qual utiliza conceitos pra ilustrar diferentes comportamentos dinâmicos integrando a experiência de especialistas na construção do modelo. Como citado anteriormente, os FCM são grafos direcionados acíclicos em que os “nós” são conceitos linguísticos, representados por conjuntos *Fuzzy*, e cada “nó” é associado com outros através de “arcos”. A cada um destes “arcos” é associado um peso numérico, que representa uma variável *Fuzzy* relacionada com o nível de causalidade entre os conceitos. Cada conceito representa uma característica do sistema, em geral: eventos, ações, metas, valores e tendências do sistema modelado. Além disso, cada conceito é caracterizado por um número que representa seu valor e é o resultado da normalização, em um intervalo $[-1,1]$, do real valor da variável do sistema para qual este conceito foi modelado. Este intervalo representa a incerteza na relação causal que leva de um nó a outro.

De um modo geral pode definir que o FCM é uma metodologia de causalidade baseada no conhecimento para modelar sistemas complexos de decisão, originado a partir da combinação de lógica *fuzzy* e redes neurais (Kosko, 1986). Um FCM descreve o comportamento de um sistema desconhecido em termos de conceitos, cada conceito representa uma entidade, um estado, uma variável, ou uma característica do sistema (Kosko, 1992).

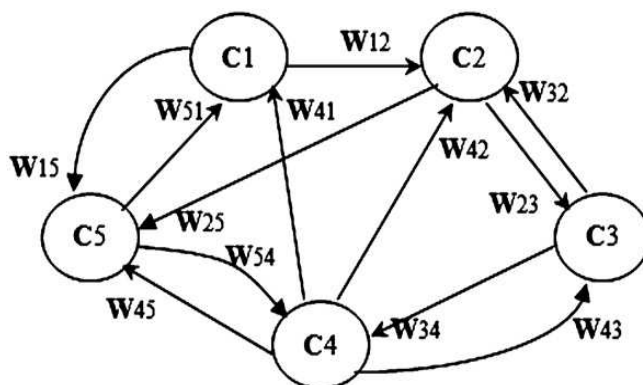


Figura 9 – Exemplo de um FCM (grafo)
Fonte: Stylios; Georgopoulos; Groumpos, (1997).

Um exemplo de FCM é mostrado na figura 9, o grafo da figura identifica as respectivas conexões causais e os conceitos do mapa.

Os conceitos de um mapa cognitivo (C_1 a C_5) podem ser atualizados através da iteração com os outros conceitos por meio das interconexões (relações causais $w_{i,j}$) e com o seu próprio valor. Isto é dado pela matriz (2) com os pesos das relações causais, e são representadas pelo peso da somatória.

$$w(i, j) = \begin{bmatrix} 0 & w_{12} & 0 & 0 & w_{15} \\ 0 & 0 & w_{23} & 0 & w_{25} \\ 0 & w_{32} & 0 & w_{34} & 0 \\ w_{41} & 0 & w_{43} & 0 & w_{45} \\ w_{51} & 0 & 0 & w_{54} & 0 \end{bmatrix} \quad (2)$$

Os valores dos conceitos vão evoluindo após várias iterações, como mostra a equação (3) até normalmente estabilizarem-se num ponto fixo ou num ciclo limite (STYLIOS; GROUMPOS, 1998).

$$A_i = f\left(\sum_{\substack{j=1 \\ j \neq i}}^n (A_j \times W_{ji})\right) + A_i^{\text{anterior}} \quad (3)$$

Onde j é o contador das iterações, n é o número de nós no grafo, W_{ji} é o peso do arco que conecta o conceito C_j ao conceito C_i , A_i (A_i^{anterior}) é o valor do conceito C_i na iteração atual (anterior) e a função f (equação 4) é uma função do tipo sigmóide:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (4)$$

Em alguns casos, o FCM pode não estabilizar e oscilar, ou até mesmo apresentar um comportamento caótico (STYLIOS; GEORGOPOULOS; GROUMPOS, 1999). Geralmente, para sistemas bem comportados, observa-se que após um número finito de iterações, os valores dos conceitos atingem um ponto de equilíbrio fixo ou um ciclo limite. Existem algumas variações da função de aprendizado utilizada (GLYKAS, 2010) e outros métodos de aprendizagem como será citado no capítulo 3.

Esta interatividade entre conceitos permite a construção de mapas de forma modular, em que pequenos mapas são criados para representar partes do problema e depois interconectados entre si. Esses pequenos mapas são testados de forma independente e só depois estas partes são conectadas para compor um mapa maior. A interatividade entre mapas cognitivos é certamente uma vantagem na construção de modelos de um sistema mais complexo, devido à possibilidade de se construir módulos diferentes de forma independente e conectá-los, *a posteriori* (MIAO et al, 2001).

Como demonstrado em Bose e Liang (1996); Silva, Spatti e Flauzino (2010) uma rede neural com conexões diretas (*feedforward*) é equivalente a um grafo direto acíclico, e assim os FCM também podem ser considerados uma rede neural. Neste contexto, os problemas relativos à construção de RNA, tais como a capacidade de representação, aprendizado, generalização, complexidade da rede, auto-organização entre outros estão presentes na construção dos FCMs.

Na interpretação de um FCM como uma rede neural, um nó ou conceito corresponde a um neurônio cuja ativação pode ser binária (*on – off*) ou contínua (0% a 100%) e as relações causais são regras *Fuzzy* ponderadas, como as sinapses que interligam os neurônios em uma RNA. Assim, os arcos ou relações causais entre conceitos podem ser obtidos via treinamento, a partir dos dados históricos do sistema. Do mesmo modo que é possível o ajuste *on-line* das sinapses em uma rede neural através de algoritmos de adaptação, a dinâmica temporal dos relacionamentos entre conceitos pode ser incorporada ao FCM, tornando-o adaptativo. Com isto obtém-se um método para a atualização constante das interconexões do grafo, somando, diminuindo, acrescentando ou apagando alguns conceitos e / ou relações.

De acordo com as definições de sistemas híbridos apresentadas nas seções anteriores, um mapa cognitivo *fuzzy* é um exemplo de sistema híbrido incorporado. Entretanto, apesar de alguns autores o considerarem como uma nova proposta de sistemas *neuro-fuzzy* (STYLIOS, GEORGOPOULOS; GROUMPOS, 1997) a maior parte dos trabalhos da literatura consideram o FCM como uma evolução dos mapas cognitivos (MIAO et al, 2001, 2010), (STYLIOS et al, 1998, 1999, 1999b), (DICKERSON; KOSKO, 1994, 1996) e, portanto de um grafo, em que foram agregadas características semânticas dos sistemas *Fuzzy* e propriedades de plasticidade e

estabilidade das redes neurais (STYLIOS et al., 2008).

Esta capacidade de auto-organização, ou, plasticidade também contribui para flexibilidade no desenvolvimento e aplicações de modelos computacionais inteligentes através de FCM. No entanto, apesar da semelhança estrutural com as redes neurais, os FCM são inicialmente construídos a partir de conhecimento empírico em que cada conceito tem um significado físico e cada conexão representa uma relação de causa-efeito entre os mesmos. Os valores iniciais dos conceitos e relações causais também têm correspondência com o comportamento físico do sistema modelado. Esta característica é inexistente nas redes neurais e tem sido um forte argumento contra a interpretabilidade deste tipo de modelo (HAYKIN, 2000).

Em resumo, FCM é uma opção de modelagem para representar sistemas complexos que não estão bem definidos, combinando as potencialidades das abordagens cognitivas dos sistemas *Fuzzy* às vantagens de modelagem oriundas das redes neurais, devido à sua estrutura em grafos, capacidade de aprendizagem, entre outras.

Esta Tese explora essas potencialidades dos FCM a fim de desenvolver uma nova metodologia para modelagem de sistemas dinâmicos complexos.

2.5 REDES COGNITIVAS DINÂMICAS

Como será ilustrado na próxima seção, diferentes variações de arquiteturas FCM são encontradas na literatura, principalmente com o objetivo de aumentar a capacidade de inferência desses mapas quando aplicados na modelagem de sistemas dinâmicos. Neste contexto, as redes cognitivas dinâmicas (DCNs) surgem como uma evolução dos mapas cognitivos (CM e FCM) que oferecem possibilidades mais amplas na gestão de estruturas causais e na modelagem de sistemas que apresentam não linearidades fortes e fenômenos temporais acentuados (KOULOURIOTIS et al., 2005).

Nas Redes Cognitivas Dinâmicas, diferentemente dos CMs e FCM em que os valores dos pesos associados aos arcos são fixos, os valores das relações causais variam no tempo de acordo com alguma função, permitindo a construção de modelos cognitivos dinâmicos naturalmente adaptativos.

De um modo geral, os FCMs na sua formatação clássica não consideram a condição temporal da relação causal, ou seja, não consideram que ao longo do tempo uma mesma relação pode apresentar diferentes graus de intensidade. Neste contexto, as DCNs foram introduzidas por Miao et al. (2001) como uma versão estendida dos CMs e FCMs, a fim de eliminar as deficiências na representação temporal das relações de causa-efeito e assim flexibilizar a aplicação dos mapas cognitivos.

As DCNs são mapas cognitivos, *fuzzy* ou não, que consideram os seguintes aspectos de uma relação causal:

1. A força da relação de causalidade;
2. A força da causa;
3. O grau de efeito.

Deste modo, as relações de causalidade em DCNs diferem não apenas na duração do efeito provocado por uma determinada causa, mas também sobre como a causa gera o efeito. As DCNs têm a capacidade de modelar e gerenciar comportamentos não-lineares dependentes do tempo, muitas vezes em tempo real, porém também não existe uma solução fechada e ainda há um amplo espaço para modificações e atualizações dessa ferramenta (KOULOURIOTIS, et al., 2005).

Esta Tese propõe um novo tipo de DCN que pode ser facilmente aplicada na modelagem de sistemas dinâmicos complexos. A metodologia proposta se desenvolve em duas camadas, cada uma associada a um tipo de conhecimento (comportamento) que se deseja modelar. A camada de baixo nível pode ser considerada um FCM ou mesmo uma DCN em que as relações de causa-efeito são representadas, na segunda camada um conhecimento mais elaborado para a tomada de decisão a partir de um planejamento a priori, mas em face de situações imprevistas, é modelado. Nesta camada, os pesos da DCN são adaptados dinamicamente a partir da ocorrência de eventos.

A fim de embasar os desenvolvimentos propostos nesta Tese e inseri-lá no estado da arte de modelagem de sistemas inteligentes baseados em mapas cognitivos, uma revisão da literatura é apresentada a seguir.

2.6 REVISÃO DA LITERATURA (FCM E DCN)

Desde o trabalho pioneiro de Kosko (1986), que estendeu os mapas cognitivos de Axelrod (1976) para a inclusão da lógica *Fuzzy*, várias aplicações de FCM são reportadas na literatura em diversas áreas do conhecimento. Em especial citam-se as aplicações em vida artificial (DICKERSON; KOSKO, 1994, 1996), sistemas sociais (TABER, 1994), tomada de decisão em rodovias de acesso rápido (PERUSICH, 1996), modelagem e tomada de decisão em ambientes corporativos e comércio eletrônico (LEE; LEE, 2003), detecção de pontos em imagens geradas por sistema de câmeras estéreo (PAJARES; DE LA CRUZ, 2006), tomada de decisão na área médica (PAPAGEORGIOU; STYLIOU; GROUMPOS, 2007) entre vários outros. Observa-se que, estes trabalhos também apresentam variações semânticas da proposta original de FCM, adaptando a estrutura do FCM de acordo com o problema tratado.

No entanto, apenas os trabalhos na área de controle e automação e/ou trabalhos que apresentem algoritmos de adaptação tem relevância para os objetivos desta Tese. Os trabalhos citados a seguir são exemplos de FCM em aplicações industriais, em especial, controle de processos, tomada de decisão e navegação robótica, tópicos relacionados ao desenvolvimento deste trabalho. Outros trabalhos citados também são importantes por empregar diferentes combinações de aprendizado, ajuste, tratamento e aquisição de informação e/ou, mudanças de estrutura do FCM clássico.

Uma das primeiras aplicações de FCMs em automação foi proposta por Gotoh et al. (1989) para a modelagem e controle supervisão de plantas industriais.

O trabalho de Ismael, Hussien e McLaren (1994) emprega um FCM para ajustar os ganhos proporcional, integral e derivativo de controladores PID de acordo com o valor de erro de regime do processo, obtido a partir de regras *Fuzzy* do tipo SE-ENTÃO definidas pelos operadores.

Também Pelaez e Bowles (1996) desenvolveram sistemas de análise e detecção de falhas empregando FCMs tendo como objetivo apoiar a tomada de decisão na supervisão e controle de processos industriais

Stylis, Georgopoulos e Groumpos (1997) utilizam uma abordagem *neuro-Fuzzy* baseada em FCM para controle distribuído de processos de uma planta industrial.

Os mesmos autores utilizam um FCM para controle de níveis e temperaturas de tanques interligados em (STYLIOS; GEORGOPOULOS; GROUMPOS, 1997). Nesse trabalho os autores apresentam ainda vários artigos que empregam uma combinação de conhecimento representativo e modelos causais (FCM) na supervisão de controladores convencionais em processos industriais (STYLIOS; GROUMPOS, 1999a, 1999b e 2000). Na mesma linha se encontra o trabalho de Sadiq, Kleiner e Rajani (2006) que usam um FCM para apoiar a tomada de decisão em controle supervísório e para previsão de detecção de falhas em um sistema de distribuição de água.

Stylios e Groumpos (2000) apresentam um sistema de controle supervísório em dois níveis aplicado a um processo de manufatura. O FCM de baixo nível modela o funcionamento do processo e características qualitativas do produto final, com vistas à detecção de falhas. O segundo FCM utiliza técnicas de agrupamento na construção de diferentes funcionalidades no nível supervísório. Para ambos FCM, regras de tomada de decisão *Fuzzy* do tipo SE-ENTÃO são empregadas nas relações causais do mapa e uma interface dedicada permite a comunicação e troca de dados entre os dois níveis. A mesma arquitetura é utilizada para modelar e descrever o comportamento dinâmico de um trocador de calor em Stylios e Groumpos (2004) e para controle de nível de um sistema de tanques acoplados em Papageorgiou, Stylios e Groumpos (2003).

Como uma evolução da proposta hierárquica os mesmo autores apresentam em Papageorgiou, Stylios e Groumpos (2006) um modelo em árvore (*Decision Tree-FCM*) para construção de um sistema de tomada de decisão na área médica. Nesta arquitetura, uma grande quantidade de dados de entrada quantitativa induz a decisão da árvore e, os dados qualitativos são usados para a construção do modelo do FCM, através do conhecimento dos especialistas. Regras de inferências do tipo **SE-ENTÃO** são usadas para ajustar a direção e os valores dos pesos das relações causais do mapa. Finalizando, o FCM modelado e treinado pelo algoritmo NHL (*Nonlinear Hebbian Learning*) (HAYKIN, 2000).

O controle hierárquico de um sistema de radioterapia, que calcula e decide a dosagem de radiação em pacientes com câncer é proposto em Papageorgiou, Stylios e Groumpos (2007). O FCM de baixo nível é responsável pelo cálculo de três variáveis de saídas (fatores e seletores), as quais devem ser analisadas no nível superior. O FCM de

alto nível atua como um sistema de tomada de decisão. As diferentes opções de ajuste entre os parâmetros dos fatores e seletores são feitos através de regras de decisão simples do tipo **SE-ENTÃO**. Observa-se que as inferências alteram os valores dos conceitos e não das relações causais previamente ajustadas de acordo com o conhecimento empírico do processo. O FCM do nível superior ajusta a dosagem final e a dosagem planejada da radiação aplicada dentro de limites máximos e mínimos de protocolos médicos já pré-estabelecidos. A interface entre os dois níveis de controle também é feita através de regras *Fuzzy* em que combinações de alterações em conceitos do FCM do nível inferior alteram conceitos no FCM do nível superior e vice-versa. Uma extensão deste trabalho é apresentada em Stylios e colaboradores (2008).

Outro exemplo de estruturas para aumentar o desempenho e/ou capacidade de tomada de decisões em FCM pode ser encontrada em Tsadiras e Konstantinos (1997). Neste trabalho, uma arquitetura baseada em redes neurais recorrentes, denominada CFFCM (*Certainty Neuron Fuzzy Cognitive Maps*) é desenvolvida, tendo como principal diferencial a utilização de uma função de transferência com duas variáveis e a introdução do domínio do tempo na estrutura FCM. De acordo com os autores, o FCM com a nova função de ativação permite níveis de ativação em qualquer decimal dentro do intervalo $[-1,1]$, incrementando a capacidade de representação do modelo. Resultados obtidos através de simulações demonstram a capacidade de inferência e o comportamento da CFFCM comparado com a versão clássica do FCM.

Outra proposta de arquitetura FCM é apresentada em Liu e Satur (1999) denominada *Contextual Fuzzy Cognitive Maps* (CFCM). O modelo CFCM emprega dados de um sistema de informações geográfico e conhecimento qualitativo de especialistas. Essa arquitetura emprega relações modeladas por funções *Fuzzy* no mecanismo de inferência para auxílio em tomada de decisões em um sistema de predição geográfica.

O modelo de FCM apresentado em Carvalho (2000) utiliza regras de inferência *Fuzzy* entre as relações causais. Esta arquitetura é denominada RB-FCM (*Rule based-FCM*). De acordo com o autor, somente as relações causais entre os conceitos restringem a aplicação de FCM em sistemas dinâmicos. O uso de regras de inferência (**SE-ENTÃO**) entre as relações permite a tomada de decisões, alterando conceitos para

diferentes pontos de operação ou para conectar vários FCM em diferentes níveis hierárquicos. Além disso, um autômato celular é utilizado para transição entre diferentes tipos de regras *Fuzzy* em diferentes cenários de um sistema dinâmico.

O trabalho de Aguilar (2004) apresenta uma nova arquitetura cognitiva denominada *Dynamic Random Fuzzy Cognitive Map* ou DR-FCM que propõe mudanças dinâmicas nas relações causais inspirados nas redes neurais aleatórias. Nessa arquitetura, funções densidades de probabilidade são inseridas no lugar de um simples valor numérico para determinar o valor das relações causais. Essas relações são atualizadas de acordo com condições específicas das variações no ambiente. O mecanismo dinâmico de atualização é através de algoritmos de aprendizagem por reforço. Para validação, é proposta uma aplicação de realidade virtual (modelo de soldados em combate), que permite avaliar o desenvolvimento, a capacidade de inferência e a estabilidade da arquitetura DR-FCM.

Jones, Roy e Corbett (2004) apresentam uma variação da proposta original de Kosko (1986), em que um FCM é acrescido de capacidade de inferência através da inclusão de um sistema ANFIS na modelagem das relações causais. O FICM (*Fuzzy Inference Cognitive Maps*) emprega dois tipos de conceitos diferentes. Os conceitos e relações causais comumente utilizados na construção de FCMs e acrescenta conceitos auxiliares para a tomada de decisão.

Uma arquitetura híbrida FCM e Algoritmos Genéticos Multiobjetivos denominados *Multiobjective Evolutionary Cognitive Maps* (MOECM) é apresentada em Mateou e Andreou (2008) e modela um cenário de política internacional. O algoritmo genético é usado para conferir capacidade de adaptação das relações causais do mapa em diferentes cenários.

Recentemente, Chun-Mei (2008) apresenta e discute a evolução de topologias FCM para a construção de modelos qualitativos, desenvolvendo uma proposta de controle multivariado com uso de FCM.

Um FCM é utilizado em Neves Júnior, Arruda e Mendonça (2009) para gerar *set-points* no nível supervisorio para o mesmo processo de fermentação utilizado nesta Tese. Nesse trabalho, os autores utilizam um FCM para modelar o processo de fermentação e um algoritmo genético para alterar as relações causais entre os conceitos

do mapa a fim de garantir o funcionamento adequado do processo nos diversos pontos de operação desejados. O desempenho da arquitetura proposta é comparado com duas outras arquiteturas, em que a tarefa de gerar *set-points* para o nível regulatório é implementada via controlador *Fuzzy* ou via controle preditivo baseado em DMC (*Dynamic Matrix Control*).

Também trabalhos que utilizam mapas cognitivos na área de robótica podem ser encontrados na literatura. Entre eles, cita-se (MIN et al, 2006), que emprega uma versão FCM probabilística na tomada de decisão de uma equipe de futebol robótico. Estas ações estão relacionadas com o comportamento da equipe, como por exemplo, chutar a bola na presença de adversários, entre outras. Essa arquitetura FCM tem uma função de probabilidade para atualizar os conceitos do mapa. Uma contribuição deste trabalho está na inclusão de tempo e memória nas tomadas de decisões, além da função de probabilidade.

Um mapa cognitivo *fuzzy* (FCM) é utilizado por Pipe (2000) para guiar um robô autônomo. A proposta desse FCM foi concebida a partir de conhecimento a priori de especialistas e, posteriormente, é refinada por um algoritmo genético. Apesar da utilização de uma trajetória conhecida, as ações são necessárias devido a erros e incertezas inerentes ao deslocamento do robô, como derrapagens, erros de leitura dos sensores, entre outros. O trabalho de Yeap, Wong e Schmidt (2008) também propõem o uso de FCM em robótica móvel para tomada de decisões dinâmicas necessárias durante tempo de navegação de um robô móvel. O artigo também apresenta uma revisão de outros trabalhos relacionados com o uso de FCM e navegação inteligente em robótica.

Algumas variações dos FCMs são relatadas na literatura como extensão ou evolução dos mapas cognitivos. Por exemplo, em Hagiwara (1992) um mapa cognitivo *Fuzzy* estendido, *Extend Fuzzy Cognitive Maps* (EFCM) é obtido pela introdução de um elemento de atraso e outras características não-lineares no FCM clássico. Infelizmente, o EFCM é uma extensão do FCM que simplesmente inclui funções de valores binários para os pesos, introduzindo pouca melhoria no FCM em termos de desempenho e capacidade de inferência. O trabalho de Park e Kim (1995) também introduzem tempos de atrasos nas relações causais, e propõe o FTCM, (*Fuzzy Time Cognitive Maps*). Os

autores usam o FTCM para modelar a gestão de cidades, e justificam a necessidade da inclusão de atrasos no processo de inferência devido a não ocorrência de todas as relações causais no mesmo ciclo. Por exemplo, a migração de pessoas para a cidade causa aumento no número de habitantes, uma maior população causa modernização. Entretanto, em um sistema real essas causas normalmente não ocorrem simultaneamente.

Outra ferramenta baseada em mapas cognitivos encontrada na literatura é a *Fuzzy-SDG (signed direct graph)*. Esta topologia utiliza o conhecimento de especialistas na forma estruturada, e é abstraído através do desenvolvimento de um mapa cognitivo. Os conceitos dão origem a variáveis *fuzzy* e as relações causais originam uma base de regras do tipo antecedente – consequente. O trabalho de Ghazanfari, Alizadeh e Jafari (2010) desenvolve um sistema dinâmico de compras para exemplificar a proposta. Esse sistema inteligente modela o tempo através do disparo de regras e inclui um fator de probabilidade para cada regra modelada. A probabilidade objetiva atenuar fatores como: incertezas, imprecisões, ambiguidade, entre outros, inerentes ao processo de aquisição de conhecimento indutivo.

O trabalho de Huang e Wang (1999) apresenta uma FCN (*Fuzzy Causal Network*) como uma proposta de evolução do *Fuzzy-SDG*. Essa ferramenta também emprega o conhecimento dos especialistas para o desenvolvimento de um mapa cognitivo, ou seja, atribuem conceitos como variáveis do problema e modelam as relações causais como correlação entre variáveis. Entretanto, os autores citam a dificuldade na identificação, e principalmente dificuldades na quantificação das relações causais. Como solução, os autores propõem uma divisão da FCN em sub-redes menores e aplicam treinamento supervisionado *backpropagation* para obtenção da magnitude das relações causais modeladas. Para exemplificar a ferramenta, os autores empregam a FCN em controle de processos, num sistema de tratamento de águas residuais.

Em Zhang, Liu e Zhou (2006), é apresentado outro tipo de FCN em que o mecanismo de inferências se dá através da criação e ativação dinâmica de subconjuntos dos conceitos modelados. Esses conjuntos e subconjuntos são contidos em intervalos mínimos e máximos das variáveis modeladas (conceitos) e, de acordo com regras de funcionamento, são ativados considerando também sua evolução temporal. Os autores

consideram a ferramenta desenvolvida ideal para representação de sistemas dinâmicos. Os autores utilizam um sistema de tomada de decisões para gerenciamento de uma universidade baseada em políticas internas e conhecimento de especialistas de diferentes departamentos, como exemplo de aplicação da ferramenta.

Como uma combinação de FCMs e FCNs surge em seguida às redes cognitivas dinâmicas DCNs que pretendem combinar as propriedades do FCMs aos sistemas de inferência causal das FCNs. A DCN descrita em Miao et al.(2001) é uma das primeiras propostas de DCN e utilizam funções de transferência no domínio de Laplace para representar o comportamento dinâmico das relações causais de um mapa cognitivo. Assim cada conceito de uma DCN tem seu valor calculado a partir de uma função de transferência ou de um modelo em espaço de estado. Estruturalmente uma DCNs é mais flexível que um FCM. De um modo geral, uma DCN pode ser tão simples quanto um mapa cognitivo (CM), um FCM, ou tão complexo como um sistema dinâmico não-linear. Para demonstrar o potencial da ferramenta, um sistema de estratégia e tomada de decisão financeira para gerenciamento de plantações é construído e resultados simulados são apresentados e comparados com um FCM.

O trabalho de Koulouriotis et al. (2005) apresenta uma proposta de rede cognitiva dinâmica (DCN) para aproximação de séries temporais aplicada em sistemas financeiros. Segundo os autores, os FCMs têm limitações para modelagem de sistemas complexos principalmente devido a não-linearidades e fenômenos temporais. A arquitetura proposta desse trabalho é semelhante à proposta em (MIAO et al, 2001) na medida em que é obtida a partir de FCM. Entretanto, a DCN proposta tem mais semelhança com redes neurais uma vez que usa o algoritmo de *Levenberg-Marquadt* para treinamento e/ou sintonia da rede cognitiva. Os autores validam e concluem a proposta através de comparativos entre DCN e RNAs.

Uma nova proposta de mapa cognitivo dinâmico é apresentada em Kottas; Boutalllis; Christodoulou (2007) e denominado rede cognitiva *Fuzzy (Fuzzy Cognitive Network - FCN)*. Essas FCNs são semelhantes às DCN e utilizam um mecanismo de atualização da matriz W baseado no algoritmo do gradiente (regra delta) a partir da realimentação do erro medido entre o estado atual e o estado desejado da rede. Os autores aplicam a FCN assim projetada no controle de nível de um modelo simplificado

de hidroelétrica. Como as DCNs em Koulouriotis et al. (2005), essas FCNs estão mais próximas de uma rede neural do que de um mapa cognitivo, propriamente dito.

A fim de estabelecer um estado da arte, Miao et al. (2010) apresentam a transformação de CM em FCM, e de FCM para DCN. Esses autores utilizam o uma DCN simplificada (*Simplified Dynamic Cognitive Network*) aplicada ao problema da ameaça terrorista, o qual foi utilizado originalmente por Kosko (1986) para exemplificar a evolução de CM, FCM até a sDCN. A sDCN usa conceitos e relações causais fuzzificadas (com a aquisição de conhecimento semelhante ao FCM) para cálculo dos valores numéricos das relações e conceitos. Entretanto, a sDCN é capaz de modelar conhecimento não monotônico, ou seja, tomar conclusões diferentes conforme ocorrem mudanças no sistema modelado. No modelo de ameaça terrorista é possível viajar ou adiar a viagem de acordo com mudanças no nível de ameaça terrorista e/ou a necessidade de realização de negócios. Para executar esse exemplo de inferência é criado um conjunto de valores para as relações e os conceitos inicialmente modelados. A transição entre os estados dos valores das relações e dos conceitos são realizados por funções de tomadas de decisões conforme necessidade do sistema modelado. Em seguida os autores mostram a DCN com relações causais expressas através de funções de transferência de Laplace como uma evolução das sDCNs.

Do exposto acima, existem diferentes arquiteturas que combinam algoritmos tais como RNA, regras de decisão *Fuzzy*, algoritmos evolutivos e populacionais, funções de probabilidade entre outros ao FCM na busca de um sistema autônomo e inteligente principalmente para tomada de decisão. No entanto, não existe ainda uma solução fechada e definitiva sobre uma estrutura eficiente, robusta e flexível para este tipo de sistema. Num primeiro momento, essa Tese propõe a utilização de metaheurísticas tais que algoritmos genéticos na otimização das relações causais dos FCM, e, em especial o uso de técnicas de aprendizagem por reforço como uma proposta de adaptação dinâmica, semelhante às apresentadas em Koulouriotis, Diakoulakis e Emiris (2001) e Petalas, Parsopoulos e Vrahatis (2009). Nestas condições, os FCMs são considerados como uma máquina de estados, que pode evoluir para diferentes pontos de operação, porém a troca de estados é discretizada pelos ciclos de atualização do algoritmo. Propõe-se então uma arquitetura híbrida em que as relações causais de um

FCMs são modificadas a partir da ocorrência de eventos no ambiente em que o sistema modelado está inserido. Os novos valores das relações causais são então obtidos a partir da execução de uma base de regras ou de um sistema *Fuzzy*.

De um modo geral, todos os trabalhos citados contribuíram no desenvolvimento, porém podem-se citar cinco trabalhos que tiveram maior influência na arquitetura DCN nessa Tese.

O trabalho (STYLIOS; GROUMPOS, 1999) apresentou a necessidade de ajuste no FCM por meio de dados de operação (conhecimento quantitativo) após aquisição do conhecimento qualitativo dos especialistas. Esse trabalho inspirou o capítulo 3 e também contribuiu para verificar as dificuldades de se construir modelos dinâmicos utilizando um FCM clássico (KOSKO, 1986).

O trabalho de Papageorgiou e Stylios (Papageorgiou, Stylios, 2003) inspirou a construção da estrutura dividida em dois níveis de conhecimento e a inclusão de conceitos de seleção e fator para tomadas de decisões. Os trabalhos de Carvalho, em especial (Carvalho, 2001) contribuíram na inclusão de conceitos de variação, inclusão de relações do tipo *Fuzzy* e no tratamento dinâmico da informação.

Finalmente os trabalhos de Miao e colaboradores (Miao et, al.,2001, 2010) contribuíram na inclusão de relações matemáticas temporais, na construção de inferências não monotônicas, e na conceituação da DCN como uma evolução dos mapas cognitivos *Fuzzy*. Neste contexto, uma proposta de uma arquitetura cognitiva inicialmente modelada de forma estruturada, baseada em CM e principalmente em FCM; é então expandida pela inclusão de novos tipos de conceitos e relações baseados em regras do tipo antecedente-consequente. Que não apenas aquelas do tipo causa-efeito estendendo o escopo da aplicação dos FCMs e propondo um novo tipo de DCN. Deste modo, esta arquitetura DCN é a principal contribuição desta Tese.

3 DESENVOLVIMENTO E MÉTODOS DE OTIMIZAÇÃO DE MAPAS COGNITIVOS FUZZY APLICADO EM CONTROLE DE PROCESSOS

3.1 INTRODUÇÃO

Este capítulo descreve a aplicação de FCM em um problema simples retirado de Stylios e Groumpos (1998) e retomado em Papageorgiou, et al (2005), a fim de demonstrar o uso, os requisitos e as limitações da técnica no desenvolvimento de um sistema de controle inteligente baseado em mapas cognitivos. Um comparativo entre alguns métodos de otimização e adaptação é apresentado para se demonstrar e analisar a necessidade de refinamento do FCM obtido apenas a partir de conhecimento especialista.

3.1.1 MAPAS COGNITIVOS FUZZY EM CONTROLE DE PROCESSOS

De acordo com as premissas apresentadas no capítulo 2, os FCMs têm sido usados na literatura como uma técnica inteligente para construção de modelos em diversas áreas de conhecimento (CHEN; HUANG, 1995; MIAO; LIU 1999; CHUN-MEI, 2008). Este fato sugere que a técnica é versátil na abstração de informações, facilitando a modelagem do conhecimento devido à sua semelhança de construção com o raciocínio humano, devido à possibilidade de abstração de incertezas através de termos linguísticos e devido inclusão de causas e efeitos entre conceitos (variáveis, ações, entre outros) de um determinado modelo cognitivo (Kosko, 1986).

Sendo uma técnica *neuro-fuzzy*, os FCMs são capazes de modelar sistemas complexos com boa precisão. De fato os mapas cognitivos em geral são grafos ponderados e direcionados (dígrafo) com realimentação, formados por nós e arcos. Os nós do grafo são utilizados para modelar conceitos descrevendo o comportamento do sistema. Estes conceitos são conectados entre si por arcos ponderados que representam as relações de causalidade existentes entre eles. Os mapas cognitivos *Fuzzy*, por sua vez, são uma classe de mapa cognitivo em que os valores dos conceitos e relações causais são números *fuzzy* (KOSKO, 1986). Como um grafo ponderado e direcionado, um FCM pode também ser considerado um autômato ou máquina de estado finito a qual

pode ser gerada a partir de conhecimento especialista e calibrada com dados do sistema (MIAO; LIU, 1999). Nesse sentido, Miao e Liu (1999) demonstram a convergência das trajetórias de inferência de um FCM e propõem teoremas que mostram que para um grande FCM, estas trajetórias podem ser decompostas, gerando uma hierarquia de FCM menores que são mais facilmente construídos. Com isto, eles demonstram que os FCM são uma técnica eficiente para a análise e modelagem de sistemas de grande escala existentes no mundo real.

O crescente interesse pela aplicação de sistemas inteligentes no controle automático de processos industriais complexos é devido às dificuldades significativas em razão de não-linearidades em regime dinâmico e, em alguns casos, baixa qualidade na aquisição de medidas disponíveis, níveis de ruído e sensoriamento fortemente distribuído. Em geral, nesses sistemas, controlam-se apenas variáveis secundárias, as quais podem ser medidas e modificadas (temperatura, pressão, fluxo, entre outros), deixando a cargo do operador a tomada de decisão que influencia a quantidade e a qualidade final e/ou parcial do processo. Com isto, o maior fator de paradas de produção é a falha humana, devido a erros inerentes à cognição humana, como por exemplo, erro de interpretação dos alarmes e respostas do sistema, sobrecarga de informação, que ocasionam erros em tomadas de decisão, entre outros.

O uso de sistemas inteligentes baseados em mapas mentais, como por exemplo, FCM e DCN apresentam vantagens na modelagem e simulação de sistemas complexos, devido a fatores como à flexibilidade da representação do conhecimento através de grandezas *fuzzy*, agregada à facilidade de abstração do conhecimento presente nos mapas cognitivos e a possibilidade de construção de grandes mapas a partir da união de pequenos mapas que representam porções do conhecimento.

3.2 CONSTRUÇÃO DO FCM

Uma das principais dificuldades na construção de mapas cognitivos é o processo de aquisição de conhecimento humano. Essa dificuldade se deve a vários fatores, entre eles, inicialmente a divergência de opinião dos especialistas na construção de um modelo único para um sistema. Quais as variáveis relevantes? Como estas se

relacionam? As respostas a estas perguntas são exemplos de pontos divergentes entre especialistas. Isso se deve à natureza do raciocínio indutivo usado pelos seres humanos para formar suas opiniões. Com isso, existem dificuldades até mesmo em se criar uma tabela de correspondência para um tipo de relação causal estimada como forte. Por exemplo, mesmo que haja um consenso entre especialistas quanto à intensidade da relação (forte) entre conceitos, poderão existir divergências na sua quantificação para valores normalizados, como por exemplo, 0,75 ou 0,80.

Assim, a consistência e a confiabilidade de um mapa cognitivo construído através da agregação de conhecimento qualitativo de vários especialistas podem ficar comprometidas, devido a incertezas na aquisição de conhecimento (geralmente feita através de entrevistas), divergência de opiniões dos especialistas ou até mesmo da dificuldade de quantificar o conhecimento empírico. Estas dificuldades podem ser contornadas através do uso de sistemas *fuzzy*, gerando os chamados mapas cognitivos *fuzzy* (KOSKO, 1986).

Para construir um FCM, inicialmente os especialistas apresentam os fatores relevantes, ou variáveis físicas descrevendo a planta que determinam os conceitos do mapa. Depois, os especialistas são individualmente perguntados sobre as relações causais entre os conceitos definidos e usam notação linguística para representá-las. Dois fatores são importantes, se as relações entre os conceitos são positiva, negativa ou neutra e qual a intensidade destas relações. Após essa etapa, termos do tipo forte, fraco, moderado, muito forte e muito fraco são utilizados para representar a intensidade dessa relação.

A figura 10 representa uma ilustração de níveis de pertinência para um determinado peso (relação causal) atribuído pelos especialistas aos termos linguísticos associados aos conceitos. Neste exemplo, o eixo x está discretizado com passo de 0,2, o que sugere cinco termos linguísticos (muito fraco, fraco, médio, forte e muito forte) para as causalidades positivas e negativas. O peso dos especialistas, devido a sua experiência e / ou seu grau de influência no desenvolvimento varia de 0 a 1 é o eixo y e os valores numéricos das variáveis linguísticas são discretizados no eixo x variando de -1 a 1. Em seguida, os vários conjuntos *Fuzzy* são agregados e o mapa final (valor do peso das relações causais) é obtido através de um processo de defuzzificação como, por

exemplo, o método do centro de área (LIN; LEE, 1996).

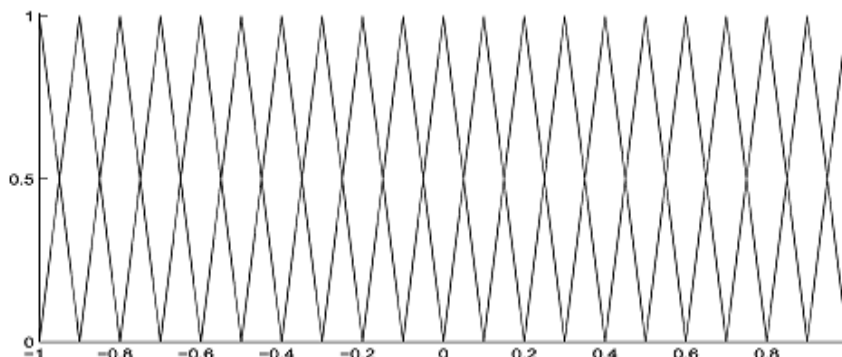


Figura 10 – Funções Pertinência *Fuzzy* Para Aquisição Relações Causais
Fonte: Adaptado de Papageorgiou et al (2005)

Apesar do uso de variáveis linguísticas e de sistemas de inferência *fuzzy* podem auxiliar na ponderação de crenças na construção dos mapas. Além disso, os sistemas de inferência *fuzzy* podem auxiliar no tratamento da incerteza, fator inerente a aquisição de conhecimento qualitativo, por exemplo, um especialista ainda pode ter dúvidas se uma relação causal é forte ou muito forte, comprometendo assim a consistência do FCM obtido. Uma etapa de sintonia ou aprendizagem baseada em dados históricos do sistema torna-se necessária para refinamento, ajuste e/ou otimização do modelo inicialmente proposto a fim de se obter a resposta desejada.

Outra característica relevante no desenvolvimento de FCMs está na capacidade de acoplamento e modularidade entre mapas que podem ser desenvolvidos por diferentes pesquisadores e/ou diferentes fases de desenvolvimento de um projeto. A figura 11 (STYLIOS, et al, 2007) mostra um exemplo de um desenvolvimento de um modelo cognitivo que acopla vários FCMs/DCNs.

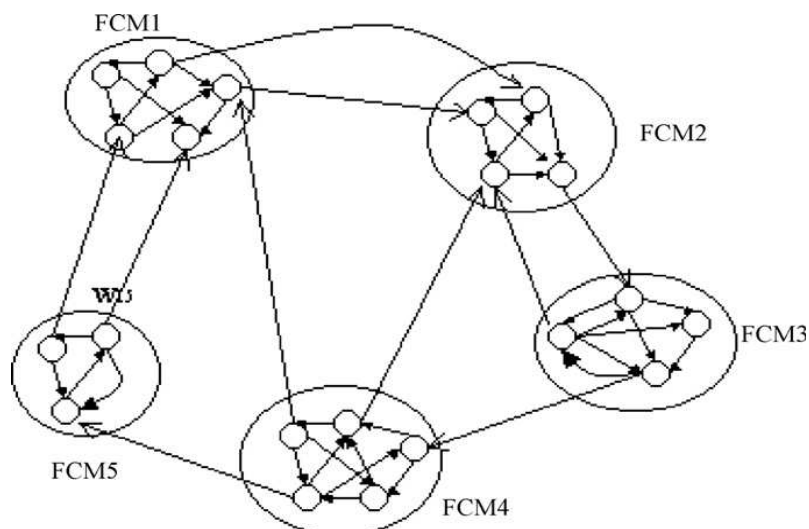


Figura 11 – Exemplo de um FCM distribuído
Fonte: Stylios, et al, (2007)

A partir da proposta original de Kosko (KOSKO, 1986) e dos diversos trabalhos subsequentes que foram publicados, pode-se resumir a construção de um FCM através da realização dos cinco passos listados no quadro 1.

Passo 1 – Identificação dos conceitos e das suas interconexões ou relações determinando a natureza (positiva, negativa, neutra) das relações causais entre conceitos.

Passo 2 – Aquisição de dados iniciais, através de ponderação de opinião de especialistas e ou análise do sistema de equações, quando se conhece o modelo matemático.

Passo 3 – Apresentação dos dados referentes à opinião dos diversos especialistas a um sistema lógico *fuzzy* que tem como saída os valores dos pesos do FCM.

Passo 4 – Tratamento da informação, adaptação e ou otimização do FCM inicialmente proposto, ajustando suas respostas às saídas desejadas.

Passo 5 – Validação do FCM ajustado nas condições de operação do sistema ou processo modelado.

Quadro 1 – Construção de um FCM clássico.
Fonte: Autoria própria.

Os passos 1 a 3 estão relacionados ao conhecimento dos especialistas e modeladores. No entanto, o passo 4 refere-se à utilização de dados históricos e/ou simulados no ajuste de um modelo geral ao sistema específico que ele deve representar. Este passo relaciona-se ao aprendizado em redes neurais, ou seja, ao ajuste (otimização) de parâmetros de uma arquitetura a fim de adequá-la ao sistema a ser modelado. Uma avaliação dos métodos de sintonia e/ou aprendizado de um FCM é apresentado no trabalho de Alizadeh e colaboradores (2007). Segundo os autores existem quatro categorias importantes para extração do conhecimento através do histórico de dados.

1. Associação de regras
2. Algoritmos baseados em correlação (classificação)
3. Algoritmo baseado no aprendizado de Hebb (*soft computing*)
4. Metaheurísticas de otimização, algoritmos genéticos, inteligência coletiva, entre outros (*soft computing*)

Atualmente a maioria dos trabalhos emprega metaheurísticas de otimização (GLYKAS, 2010). Por exemplo, Alizadeh e colaboradores (2007) usam Busca Tabu e Algoritmos genéticos. Stach, e colaboradores (2005) utilizam uma variação do algoritmo genético denominada RCGA (*real codec-G.A.*). O trabalho de Ghazanfari e colaboradores (2007) emprega e compara os métodos, *simulated annealing* e algoritmos genéticos na otimização de um FCM clássico. O trabalho de Baykasoglu, Durmusoglu e Kaplanoglu (2010) emprega uma variação do algoritmo genético denominado *Extended Great Deluge Algorithm* na otimização de um FCM abordando problemas clássicos, como por exemplo, o processo de controle de nível e massa de um tanque.

Os trabalhos citados empregam uma função contínua para o aprendizado do FCM semelhante à equação (4) apresentada no capítulo 2.

Para analisar a viabilidade do passo 4 do algoritmo do quadro 1 e levantar os principais pontos de investigação atual dos modelos baseados em FCM, aplica-se três técnicas distintas para a sintonia de FCMs. Esta avaliação será baseada em um exemplo retirado da literatura e que constitui um dos primeiros exemplos da aplicação de FCM em controle de processo (STYLIOS; GROUMPOS, 1999). Os métodos utilizados para a

sintonia do FCM são as redes neurais, os algoritmos genéticos e o algoritmo de nuvens de partículas.

3.2.1 FCM Sintonizado Por Redes Neurais

O uso de redes neurais artificiais como um algoritmo para ajuste dos mapas cognitivos é uma proposta dessa Tese. O método busca o refinamento e ajuste dos valores das relações causais determinados pelos especialistas durante a construção do mapa, a partir de dados de operação do sistema. Ou seja, a RNA utiliza os dados de operação do FCM inicialmente sugerido pelos especialistas para mapear ajustes nas relações causais obtidas somente pelo conhecimento qualitativo. Assim, a função da RNA é de identificar valores ou faixa de valores de relações causais que contemplem a saída ou as saídas desejadas para o modelo do sistema. Isto é obtido através de um mapeamento reverso em que os valores dos conceitos, a resposta desejada do modelo FCM, são utilizados como entrada da rede neural, e os valores das relações causais (pesos do FCM), são considerados a saída da rede da neural. Após a etapa de treinamento, a RNA é capaz de identificar novos conjuntos de valores para as relações causais que levam o FCM a obter respostas mais próximas as desejadas.

3.2.2 FCM Sintonizado Por AG

O FCM sintonizado por algoritmos genéticos é uma proposta já conhecida na literatura, tendo sido utilizada em (KOULOURIOTIS, DIAKOULAKIS E EMIRIS, 2001) e (STACH, et al., 2005), entre outros. Na implementação do AG, um cromossomo contém todas as relações causais do FCM proposto, e deste modo, através de operações de cruzamento e mutação é possível encontrar a melhor solução (conjunto de pesos) dentro das faixas de intervalo das relações causais propostas pelos especialistas. Na implementação do AG, uma população inicial é gerada aleatoriamente dentro do intervalo de discurso dos conceitos propostos pelos especialistas. À medida que o método evolui através da utilização dos operadores genéticos de reprodução, mutação e seleção, os valores das relações causais vão sendo otimizados a partir da avaliação de uma função de adequabilidade (GOLDBERG, 1989). A flexibilidade de

implementação e a não necessidade de um período de treinamento, como nas RNA. Porém é necessário um período de tempo para convergência do método, o qual foi menor comparado com o tempo de treinamento da RNA, nos experimentos realizados. Essas características citadas são relevantes na aplicação dessa metaheurística para a adaptação de FCM.

3.2.3 FCM Sintonizado Por PSO

A sintonia dos pesos de um FCM proposta por Shi e Eberhart (1998) e também utilizada por Papageorgiou e colaboradores (2005) e por Parsopoulos e colaboradores (2005) utiliza a técnica de otimização estocástica baseada em nuvem de partículas (*Particle Swarm Optimization - PSO*). Especificamente, o PSO é um algoritmo evolutivo inspirado no comportamento social de grupos de aves migratórias. Inicialmente sem nenhuma orientação prévia, essas partículas se aglomeram até que uma delas consiga encontrar uma melhor posição, do ponto de vista do objetivo escolhido, o que acaba por atrair as partículas que estiverem mais próximas. Desta forma desenvolve-se o que se chama de inteligência social, em que um indivíduo aprende com o acerto do outro. Como o AG e outros algoritmos evolutivos, o PSO é um algoritmo que procura soluções em um espaço de busca, explorando populações de indivíduos. A aplicação dessa técnica para a sintonia de FCMs corresponde à busca das conexões causais do FCM que contemplem as saídas especificadas pelo problema. A técnica consiste em se minimizar uma função objetivo, similar à função de adequabilidade dos algoritmos genéticos, representativa da qualidade do modelo descrito.

3.3 Caso de estudo: Controle de nível

Essa seção descreve uma aplicação de FCM em um controle de processo simplificado com o objetivo de evidenciar as dificuldades da aplicação do algoritmo do quadro 1 na construção de controladores inteligentes baseados em FCM. Outro objetivo dessa seção é demonstrar a necessidade de adaptação e/ou otimização do FCM, devido à divergência de opinião dos especialistas na construção do modelo, à presença de

incertezas e dificuldades de aquisição de conhecimento, entre outros fatores. Mesmo que todas essas dificuldades de projeto pudessem ser contornadas, ainda assim seria necessária a adaptação do modelo por FCM, já que o controlador precisa se ajustar aos diferentes pontos de operação do processo. A adaptação do FCM será feita inicialmente via redes neurais conforme explicado no item anterior, e em seguida via algoritmos genéticos. Os resultados obtidos serão comparados com resultados da literatura (PAPAGEORGIOU et al, 2005) que emprega a técnica de PSO na adaptação do FCM modelado para o mesmo problema.

3.3.1 Descrição do Processo Estudado

O controle de nível descrito a seguir é um estudo de caso bastante utilizado na literatura (STYLIOS; GROUMPOS, 1999), (BAYKASOGLU, DURMUSOGLU, KAPLANOGLU, 2010) e (GLYKAS, 2010) para teste de controladores. Este caso de estudo foi selecionado para ilustrar a necessidade de refinamento de um modelo baseado em FCM construído exclusivamente com conhecimento especialista, e para validar a utilização de RNA no aprendizado de um FCM. O processo apresentado na figura 12 consiste em um tanque com duas válvulas de entrada para diferentes líquidos, um misturador, uma válvula de saída para retirada do líquido produzido pela mistura e um medidor de massa específica que mede a qualidade do líquido produzido. As válvulas (V_1) e (V_2) inserem dois diferentes líquidos no tanque. Durante a reação dos dois líquidos, um novo líquido caracterizado por seu valor de massa específica é produzido. A válvula (V_3) esvazia o tanque, quando esse líquido atinge o nível especificado.

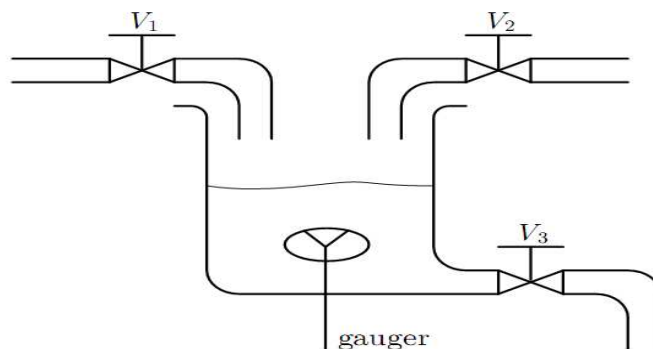


Figura 12 – Processo Químico com Misturador

Fonte: modificada de (STYLIOS; GROUMPOS, 1999)

Apesar de ser relativamente simples, este processo é do tipo TITO (*two inputs – two outputs*) com acoplamento entre as variáveis. Para estabelecer o controle de qualidade em regime do líquido produzido, um medidor colocado dentro do tanque pesa (mede a massa específica) o líquido produzido. Quando o valor da variável de medição G (massa líquido) atinge a faixa de valores entre o máximo e mínimo $[G_{min}, G_{max}]$ especificado, a mistura líquida desejada está pronta. A retirada do líquido só é possível quando o volume (V) estiver em uma faixa especificada entre os valores $[V_{min}$ e $V_{máx}]$. O controle consiste então em manter essas duas variáveis em suas faixas de operação.

$$V_{min} < V < V_{max}; \quad (5)$$

$$G_{min} < G < G_{max} \quad (6)$$

De acordo com Papageorgiou et al (2005), os especialistas definiram uma lista de cinco conceitos relacionados às principais grandezas físicas envolvidas no processo:

Conceito 1 – Quantidade de líquido (volume) no tanque, que depende do estado operacional das válvulas V_1 , V_2 e V_3 .

Conceito 2 – Estado da válvula 1 (fechada, aberta ou parcialmente aberta).

Conceito 3 – Estado da válvula 2 (fechada, aberta ou parcialmente aberta).

Conceito 4 – Estado da válvula 3 (fechada, aberta ou parcialmente aberta).

Conceito 5 – Valor medido pelo sensor G para a massa específica do líquido.

Além desses conceitos, os especialistas estabeleceram um domínio para as variáveis, resultando numa faixa de valores para as relações causais dadas pelas equações 7 a 14.

$$-0,50 < W_{12} < 0,30; \quad (7)$$

$$-0,40 < W_{13} < 0,20; \quad (8)$$

$$0,20 < W_{15} < 0,40; \quad (9)$$

$$0,30 < W_{21} < 0,40; \quad (10)$$

$$0,40 < W_{31} < 0,50; \quad (11)$$

$$-1,0 < W_{41} < 0,80; \quad (12)$$

$$0,50 < W_{52} < 0,70; \quad (13)$$

$$0,20 < W_{54} < 0,40; \quad (14)$$

O FCM assim construído é mostrado na figura 13, após a sua estabilização, obteve-se a seguinte matriz de pesos e os seguintes valores para os conceitos.

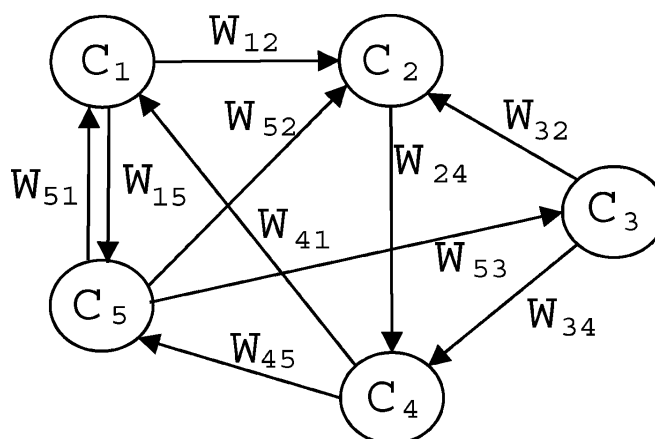


Figura 13 – FCM para o Controle do Misturador
Fonte: Papageorgiou et al (2005).

$$W^{inicial} = \begin{pmatrix} 0,00 & -0,40 & -0,25 & 0,00 & 0,30 \\ 0,36 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,45 & 0,00 & 0,00 & 0,00 & 0,00 \\ -0,90 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,60 & 0,00 & 0,30 & 0,00 \end{pmatrix} \quad (15)$$

$$A^{inicial} = [0,1 \quad 0,45 \quad 0,39 \quad 0,04 \quad 0,01] \quad (16)$$

No exemplo simulado, os limites das equações (5) e (6) são reajustados para os valores abaixo, correspondendo a um determinado ponto de operação desejado.

$$0,68 < C_1 < 0,70; \quad (17)$$

$$0,78 < C_5 < 0,85; \quad (18)$$

A figura 13 mostra os conceitos e as conexões das relações causais modeladas pelos especialistas. E, como mostrado na matriz (15), os valores escolhidos para W apesar de estarem na faixa inicialmente proposta pelos especialistas, equações (7) a (14), não contemplam o ponto de operação desejado, uma vez que os valores de C_1 e C_5 obtidos não estão dentro da faixa de operação desejada dados pelas equações (17) e (18). Deste modo, o FCM precisa ter suas relações ajustadas a fim de que as variáveis controladas representadas pelos conceitos C_1 e C_2 do FCM, possam atingir os valores em regime estabelecidos.

Neste ponto, duas questões se impõem:

1. É possível encontrar uma matriz de W dentro do intervalo considerado nas equações (7) a (14), tal que o FCM atinja um estágio de equilíbrio em que os conceitos C_1 e C_5 obtidos estejam dentro da faixa de operação desejada, (17) e (18)?
2. Em caso negativo, quais das restrições (7) a (14) devem ser relaxadas para assegurar o desempenho desejado?

A resposta a essas questões está no desenvolvimento de métodos de

aprendizado e/ou sintonia de FCMs. Três métodos serão apresentados e seus resultados são avaliados no desenvolvimento desse capítulo.

3.4 Métodos empregados na sintonia do FCM

Os três métodos propostos para realizar a sintonia do FCM emprega aprendizado de uma rede neural artificial e duas metaheurísticas de otimização, inspirados na computação evolutiva e inteligência distribuída.

3.4.1 Sintonia do FCM Empregando RNA

Para o ajuste do mapa, uma RNA como a da figura 14 pode ser treinada com os conceitos e relações causais do FCM para realizar o ajuste das relações causais do FCM.

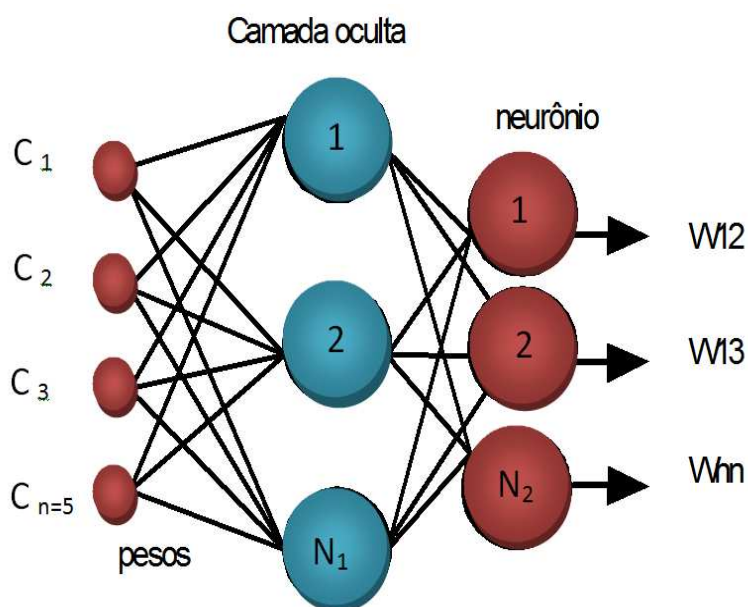


Figura 14 – RNA para Adaptação do FCM (Figura 13)
Fonte: autoria própria.

A rede neural artificial é do tipo multicamada com uma camada intermediária contendo 120 neurônios, determinados empiricamente. Os dados aplicados na entrada

da rede são as respostas do FCM, que foi inicialmente ajustado para as faixas de pesos das relações causal propostas pelos especialistas, equações (7) a (14). O treinamento consiste em discretizar uma faixa de valores inicialmente proposta em uma região de treinamento com 200 pontos diferentes.

Após o treinamento, os valores dos conceitos são apresentados à entrada da rede, observando que os mesmos têm que estar dentro da região de treinamento. Assim, a RNA generaliza e estima novos valores de pesos de conexões, os quais sugerem uma nova região de trabalho, permitindo através de poucas iterações do problema ajustar novos valores das conexões do FCM pra se atingir a região desejada.

O método de otimização empregado no treinamento da rede é o método de *Levenberg Marquadt*, devido à melhor precisão de resposta e menor número de épocas necessário para o treinamento, quando comparado com outros métodos de treinamento, como, por exemplo, o método do gradiente descendente (resultados observados em simulações da RNA para este estudo de caso). Deste modo, após o treinamento, que de fato é um mapeamento reverso do FCM, a saída é um vetor com estimativas para os pesos das conexões causais mais próximos da região desejada estimado pela rede na primeira iteração são apresentados na matriz (19).

$$W(1) = \begin{bmatrix} 0 & -0,3085 & -0,2303 & 0 & 0,4196 \\ 0,4098 & 0 & 0 & 0 & 0 \\ 0,5009 & 0 & 0 & 0 & 0 \\ -0,7901 & 0 & 0 & 0 & 0 \\ 0 & 0,6916 & 0 & 0,4119 & 0 \end{bmatrix} \quad (19)$$

Ao receber os valores de conceitos calculados pelo FCM na primeira iteração, a rede neural calcula novos valores de pesos, sugerindo mudanças na faixa inicialmente proposta pelos especialistas, especialmente para os valores de W_{13} , W_{15} e W_{54} . Assim, o conjunto de pesos a ser aplicados no FCM na segunda iteração vale:

$$W(2) = \begin{bmatrix} 0 & -0,3176 & -0,2465 & 0 & 0,4657 \\ 0,4441 & 0 & 0 & 0 & 0 \\ 0,4813 & 0 & 0 & 0 & 0 \\ -0,7692 & 0 & 0 & 0 & 0 \\ 0 & 0,7123 & 0 & 0,3896 & 0 \end{bmatrix} \quad (20)$$

Como resposta, os conceitos C_1 e C_5 assumem respectivamente na segunda iteração os valores 0,6664 e 0,7336, mais próximos da região desejada, validando a evolução das mudanças nas relações de causalidade. Novos valores para a matriz W são calculados pela RNA, a partir dos novos valores dos conceitos.

A resposta do FCM na terceira iteração calcula $C_1=0,6752$ e $C_5=0,7420$, os quais estão mais próximos dos valores desejados. Repetindo-se o procedimento de evolução dos conjuntos soluções propostos pela rede, o FCM se estabiliza nos valores de conceitos pretendidos, que são obtidos com os valores das relações causais propostos pela RNA, cujo aprendizado reverso é ilustrada na figura 14. Resumindo, após a terceira iteração já se encontra uma faixa de valores das causalidades que contempla a solução. Os valores resultantes (arredondados com duas casas decimais), para estas faixas de operação são:

$$W(\infty) = \begin{bmatrix} 0 & -0,29 & -0,28 & 0 & 0,68 \\ 0,51 & 0 & 0 & 0 & 0 \\ 0,50 & 0 & 0 & 0 & 0 \\ -0,77 & 0 & 0 & 0 & 0 \\ 0 & 0,79 & 0 & 0,43 & 0 \end{bmatrix} \quad (21)$$

3.4.2 Sintonia do FCM utilizando AG

Para sintonizar o FCM através de AG, utiliza-se como função de adequabilidade (*fitness*) o erro entre a saída desejada e os valores atuais do FCM para os conceitos C_1 e C_5 . O cromossomo proposto é formado pela lista ordenada dos pesos das relações causais W [W_{12} , W_{13} , W_{15} , W_{21} , W_{31} , W_{41} , W_{52} e W_{54}] do modelo FCM. Valores de máximo e mínimo do espaço de busca são previamente definidos a partir de valores propostos pelos especialistas, equações (7) a (14). A probabilidade de aplicação

dos operadores de cruzamento foi fixada em 95%, e do operador de mutação em 5%. Não se utilizou parâmetro de clonagem nem técnicas elitistas. O tamanho da população é de 20 indivíduos. Esses parâmetros foram obtidos de modo empírico.

O algoritmo implementado é o AG clássico como proposto por Goldberg, (1989). No entanto a cada geração, são efetuadas mudanças dos valores máximos e mínimos inicialmente sugeridos pelos especialistas para os pesos das relações causais. Essa mudança do espaço de busca da população é implementada através do deslocamento das faixas de ajustes de acordo com a proximidade dos valores das relações causais aos valores limites inicialmente propostos.

A evolução da primeira geração calcula o seguinte conjunto de pesos:

$$W(1) = \begin{bmatrix} 0 & -0,30 & -0,40 & 0 & 0,50 \\ 0,30 & 0 & 0 & 0 & 0 \\ 0,50 & 0 & 0 & 0 & 0 \\ -0,80 & 0 & 0 & 0 & 0 \\ 0 & 0,50 & 0 & 0,20 & 0 \end{bmatrix} \quad (22)$$

Estes valores de relação causal levam aos seguintes valores para os conceitos $C_1 = 0,6421$ e $C_5 = 0,7436$ com um valor de função de adequabilidade $fval = 25,7$. Como o problema modelado pelo AG é de minimização do erro entre o valor desejado para os conceitos dentro da faixa proposta pelos especialistas e a resposta do FCM para os valores de pesos da geração corrente, o valor de $fval$ deve tender a zero, indicando que o método está a cada geração mais próximo de uma possível solução.

O espaço de busca é definido pelo limite máximo e mínimo das relações causais, utilizando as faixas de valores inicialmente propostas pelos especialistas, equações (06) a (13). Porém, a cada iteração do algoritmo, ou seja, a cada geração, um novo espaço de busca é definido tomando como ponto médio o valor atual de cada relação causal. Após a evolução dos conjuntos soluções propostos pelo algoritmo genético, a saída desejada do FCM, dada na coluna 2 do quadro 2, é atingida para os valores de pesos dados na figura 14. Observa-se que o espaço de busca é sempre alterado a cada geração do algoritmo e que, o valor de saída está na faixa de valores, dados pelas inequações (17) e (18), o que facilita a obtenção de uma resposta e contribui na diminuição do número de gerações do algoritmo.

3.4.3 Sintonia do FCM Utilizando PSO

O PSO utilizado neste estudo de caso é semelhante ao PSO clássico, apresentado no trabalho de Serapião (2009). O algoritmo é baseado na posição e velocidade em direção do indivíduo (partícula) e do grupo. A função de custo é a diferença entre as variáveis desejadas e os respectivos conceitos de saída do FCM. Os parâmetros iniciais R1 e R2 são parâmetros aleatórios uniformemente distribuídos entre [0,1]; já os fatores de confiança são $c1 = c2 = 2$ (valores default). A inércia da partícula (w), é inicialmente em torno de 1, com um declínio gradual para 0.

No trabalho de (PAPAGEORGIOU et al, 2005) a evolução do método de PSO teve seu espaço de busca alterado (ampliado e transladado), ou seja, as restrições iniciais devido à faixa de valores das relações causais não permite a convergência do FCM para as duas faixas de saídas desejadas. Este comportamento também foi confirmado pelos resultados obtidos pela RNA e pelo AG como mostra a figura 15.

3.4.4 Comparação dos Resultados

Os três métodos evoluíram a partir das mesmas condições iniciais, ou seja, a primeira opinião inicialmente proposta pelos especialistas, para os conceitos. O modelo tem inicialmente a seguinte saída

$$C_i = [0,6723 \quad 0,7417 \quad 0,6188 \quad 0,6997 \quad 0,7311] \quad (23)$$

Observa-se que a região desejada descritas nas inequações (17) e (18) não é contemplada por estes valores, deste modo, é necessário um refinamento nas relações causais do mapa. Assim, após algumas iterações, os métodos avaliados auxiliam o FCM a atingir um novo ponto de equilíbrio e conseqüentemente uma nova região desejada de operação do sistema. O quadro 2 apresenta os resultados dos métodos de treinamento baseados em RNA, PSO e AG.

Os resultados dos três métodos, valores finais dos conceitos do FCM,

determinam praticamente os mesmos valores dos conceitos para a massa específica e para o nível do tanque, dentro da faixa de operação desejada previamente definida pela estratégia de controle. Estes valores similares são atingidos pelos três métodos, apesar desses métodos calcularem valores diferentes para os conceitos e conjuntos de pesos durante suas aplicações.

Conceitos	RNA	AG	PSO
C_1	0,6914	0,7008	0,6805
C_2	0,7645	0,7645	0,7798
C_3	0,6003	0,6124	0,6176
C_4	0,7466	0,7375	0,6816
C_5	0,7768	0,7999	0,7967

Quadro 2- Resultado do Modelo Baseado em FCM Com os Diferentes Métodos Otimização.
Fonte: Autoria própria

As faixas das relações causais propostas inicialmente pelos especialistas para uma possível sintonia do FCM é representada a área verde da figura. A solução encontrada para os diferentes métodos são indicadas por seus respectivos símbolos como mostra a figura 15. De acordo com os resultados encontrados, somente os pesos das relações causais W_{15} , W_{52} , ficaram significativamente fora da faixa inicialmente proposta pelos especialistas. A solução obtida com a RNA manteve o conjunto solução mais próximo da solução proposta inicialmente. A solução encontrada pelo algoritmo genético teve resultados diferentes das faixas dos valores encontrados pela RNA e do PSO.

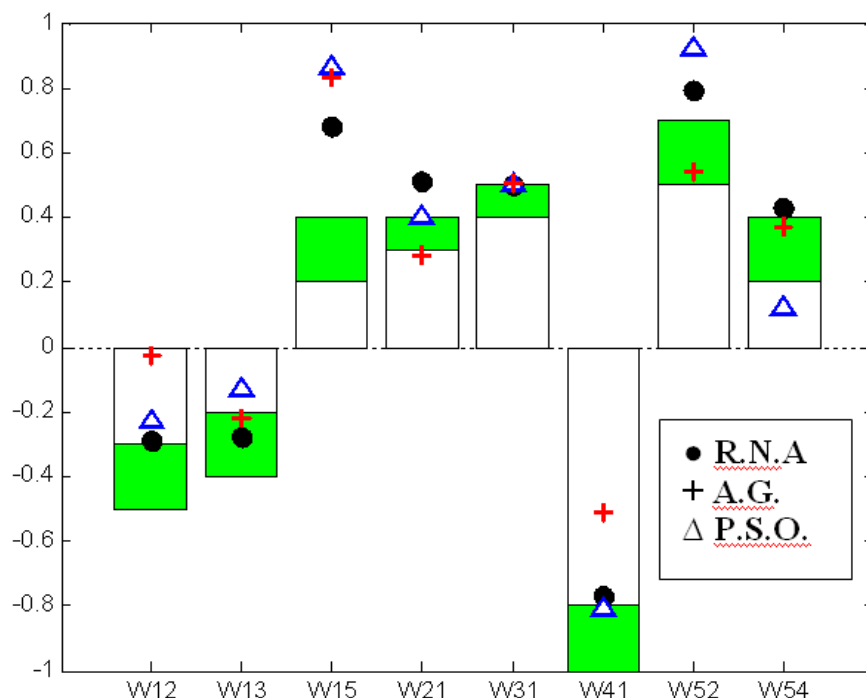


Figura 15 – Intervalos de confiança e valores calculados para os pesos pelos diferentes métodos
Fonte: Autoria própria.

A diferença de resultados se deve principalmente por o problema de controle ter umas soluções dentro de uma faixa de valores que possibilita a existência diferentes conjuntos de soluções factíveis e não apenas uma solução ótima. Observa-se que a rede neural utiliza aprendizagem reversa a partir de dados de treinamento igualmente espaçados em uma grade. Já os métodos evolucionários aplicam um artifício de deslocamento e redução do espaço de busca. A natureza estocástica dos três métodos também determina a convergência para diferentes pontos, que poderia ser contornada através de um estudo estatístico mais abrangente.

3.5 Desvantagens na utilização de FCM

Uma vantagem da modelagem através de *Fuzzy Cognitive Maps* está na possibilidade de ajustar suas conexões causais e, se necessário, os valores iniciais dos conceitos do modelo, permitindo um refinamento específico do FCM. Esse processo de

refinamento pode ser através da interação direta do especialista, ou através de algoritmos de aprendizado ou uma combinação de ambas, como mostrado no caso de estudo acima apresentado. Por exemplo, se um especialista fornecer um novo conjunto de variáveis para simulação devido a uma mudança de ponto de operação em uma planta, esta pode ser facilmente implementável mudando as relações causais do mapa construído.

No entanto, a principal dificuldade da modelagem através de FCM utilizando o algoritmo do quadro 1, está na dificuldade de se modelar o efeito do tempo nas relações causais. No modelo clássico de FCM proposto por (KOSKO, 1986) todas as causalidades ocorrem de modo simultâneo, o que restringe a utilização do modelo obtido em sistemas dinâmicos.

Outra desvantagem dos modelos baseados em FCM, também já abordado na revisão bibliográfica do capítulo 2, está na dificuldade de se modelar sistemas dinâmicos que comportem não-linearidades acentuadas ou ainda relações causais não-montônicas. Outra dificuldade está na necessidade de adaptação do modelo em presença de eventos que não foram inicialmente modelados pelos especialistas. De um modo geral, estes procedimentos são semelhantes às técnicas de poda e crescimento de redes neurais.

De um modo geral, a maior restrição ao uso de FCM como ferramenta de modelagem de sistemas dinâmicos mais frequente na literatura é que os FCMs apesar de simples não são capazes de modelar os diversos tipos de relação existentes em sistemas dinâmicos, principalmente aquelas relacionadas à realimentação e ao comportamento temporal das variáveis. De fato, um FCM construído segundo a proposta original de Axelrod (1976) e Kosko (1986) somente representa relações causais monotônicas e simétricas.

Para contornar esses problemas, essa Tese propõe o uso de DCNs capazes de modelar a causa, o efeito e principalmente o tempo das relações do modelo cognitivo.

3.6 CONCLUSÃO DO CAPÍTULO

De acordo com os resultados obtidos, e, principalmente as dificuldades apresentadas, o próximo capítulo propõe um novo método para aquisição, estruturação e desenvolvimento de modelos inteligentes de sistemas dinâmicos baseado em mapas cognitivos *fuzzy*. Este novo método utiliza um tipo de mapa cognitivo conhecido na literatura como Rede Cognitiva Dinâmica.

A nova arquitetura baseada em DCN para modelagem de sistemas dinâmicos agrega além da aquisição de conhecimento na sua forma estruturada, do refinamento e sintonia através de algoritmos de treinamento, um tratamento a eventos que permite calcular dinamicamente os valores dos pesos da DCN. Além disto, novos tipos de relações, conceitos e base de regras serão incluídos neste modelo cognitivo, a fim de contemplar todos os aspectos inerentes aos sistemas dinâmicos.

4 NOVO MÉTODO PARA MODELAGEM DE SISTEMAS DINÂMICOS BASEADO EM DCN

4.1 INTRODUÇÃO

Este capítulo apresenta um novo método baseado em redes dinâmicas cognitivas DCN para a modelagem de sistemas dinâmicos. A DCN é obtida a partir de um FCM ao qual são atribuídas novas potencialidades. Inicialmente, para dar maior flexibilidade aos FCMs na modelagem de sistemas complexos, novos tipos de conceitos e relações são introduzidos. Também para contornar as dificuldades do FCM em modelar aspectos temporais e comportamentos dinâmicos e melhorar a sua capacidade de inferência, propõe-se uma estratégia para conferir adaptabilidade aos modelos, notadamente os valores das relações (arcos do mapa cognitivo) ao longo do tempo, a partir da ocorrência de eventos. Um mapa cognitivo cujos valores de arcos são variantes no tempo forma uma Rede Cognitiva Dinâmica.

4.1.1 Redes dinâmicas cognitivas

Mapas cognitivos (CMs), mapas cognitivos *Fuzzy* (FCMs) e redes cognitivas dinâmicas (DCNs) são métodos usados para modelagem da cognição humana e para facilitar o desenvolvimento de máquinas de inferências (CARVALHO, 2010). FCMs estendem CMs, e as DCNs estendem os FCMs. A construção de modelos cognitivos apresenta dificuldades na aquisição do conhecimento, devido à possibilidade de erros humanos, à presença de incertezas, à existência de informação incompleta, entre outros fatores.

Além das dificuldades de aquisição de conhecimento citadas, existem ainda dificuldades relativas à complexidade/tamanho dos sistemas reais que se deseja modelar. De acordo com revisão da literatura de FCM, FCN, DCN apresentada no capítulo 2, e conforme ressaltado no capítulo 3, a maior dificuldade para construção de modelos cognitivos, em especial dos mapas cognitivos do tipo FCM, está no tratamento dos aspectos temporais. Em geral, o processo de inferência de um FCM pressupõe que o

efeito é consequência da ocorrência simultânea de todas as suas causas. Por exemplo, em um sistema de gerenciamento de população de uma cidade ou um estado, pode-se considerar que a ocorrência de doenças e/ou epidemias tem uma influência negativa no processo de imigração. Deste modo, empregando um FCM na sua forma clássica em que valores de arco são fixos, não é possível modelar uma sequência temporal de relações causais, ou seja, todas as relações incluídas no FCM ocorrem de modo simultâneo. Deste modo não se pode “aguardar” a evolução de uma doença ou epidemia para depois se considerar sua influência negativa no processo de imigração.

Outro exemplo clássico citado por Miao et al (2010) é um sistema de gerenciamento de turismo. A taxa cambial pode ter impacto positivo ou negativo em um modelo econômico, e de modo mais específico, este tipo de situação pode ter impacto mais forte ou mais fraco em diferentes épocas do ano.

Em resumo, a inferência realizada por este modelo de mapa pressupõe que todos os eventos ou acontecimentos ocorrem de maneira simultânea e não se pode construir inferências simultâneas. Além disso, não é possível construir modelos que sejam capazes de construir inferências não monotônicas. Ou seja, após a construção de um modelo cognitivo baseado em FCM, não se consegue realizar uma inferência que tenha uma conclusão divergente das anteriores. Entretanto, em um sistema real a ocorrência simultânea de todos os eventos e/ou causas são situações específicas, e em alguns casos é necessário realizar inferências não monotônicas.

A fim de contornar essas dificuldades, apresenta-se, como proposta desta Tese, um novo método para desenvolvimento de sistemas inteligentes baseados em mapas cognitivos com capacidade de raciocínio não monotônicos e capacidade de adaptação dinâmica à variação do ambiente. Este novo método, denominado Rede Dinâmica Cognitiva conserva as facilidades de abstração de conhecimento através de mapeamentos causa-efeito, e são aptos a modelar sistemas dinâmicos complexos, não-lineares e variantes no tempo. Com isto, a abrangência da técnica se expande permitindo aplicações em diversas áreas de conhecimento, como, por exemplo: futebol de robôs, navegação robótica, controle de processos, mundo virtual, entre outras.

Após a apresentação das etapas de construção do modelo cognitivo dinâmico, um exemplo de aplicação para tomadas de decisões de um time de futebol de robôs é apresentado de forma qualitativa.

Essa proposta de DCN será validada nos próximos capítulos, na construção de controladores para um sistema supervisório. Em especial, para a geração de *Set-points* e funcionalidades de operação de um processo de fermentação alcoólica.

Após a validação da aplicação da DCN em controle de processos, a metodologia proposta será utilizada em navegação robótica apresentando uma arquitetura em que a DCN é sintonizada através de algoritmos de aprendizagem por reforço para melhor adaptação das tomadas de decisões de acordo com mudanças no ambiente em que o robô será inserido.

4.2 Modelagem de sistemas dinâmicos através de DCNs

A arquitetura para construção de modelos cognitivos proposta nesta Tese é baseada em redes cognitivas dinâmicas. Sendo um tipo de mapa cognitivo, as DCNs são obtidas a partir de conhecimento qualitativo de especialistas, que descrevem seus conhecimentos através de relações de causa e efeito. Para a descrição de um sistema complexo através de um mapa causal, baseado em DCNs, é necessário a definição de 3 elementos fundamentais:

1. A causa;
2. A relação de causalidade;
3. O efeito.

De acordo com a literatura, os FCMs melhoraram os CMs, descrevendo a força das relações causais por meio de números *fuzzy*. As DCNs por sua vez, melhoram os FCMs permitindo que cada conceito selecione sua relação de causalidade com os outros conceitos de acordo com as exigências do sistema. Essas relações assumem valores que podem ser retirados de um conjunto binário, de um conjunto *fuzzy*, ou de uma função contínua ou trivalente. A DCN proposta nessa Tese extrapola esse comportamento na medida em que introduz procedimentos para determinar como as causas produzem efeitos de forma sazonal, ou seja, através do disparo de regras que identificam a

ocorrência simultânea de vários fatores (causa) ou a partir de funções matemáticas em que o tempo é uma variável independente. Também o efeito da causa pode ser variante no tempo. Deste modo, as DCNs aqui propostas podem ser consideradas modelos causais com habilidades dinâmicas em que as relações de causalidade e os efeitos são variáveis temporais. De modo simplificado, a ativação e duração das regras inseridas na DCN₂ orientadas a eventos gerenciam a dinâmica do modelo.

A DCN proposta se comporta de acordo com a complexidade do problema abordado. Para as decisões de mais baixo nível, em que se admitem apenas relações de causa-efeito positiva, negativa e neutra entre conceitos, a DCN se comporta como um mapa cognitivo simples. Para decisões com vários fatores implicados, a DCN se comporta como um FCM, em que as relações causais e conceitos são números *fuzzy*, porém considera-se que todas as causalidades ocorrem de modo simultâneo. Para os sistemas dinâmicos mais complexos, a DCN é capaz de alterar dinamicamente os valores de suas relações conceitos, além de inserir e ou eliminar novos conceitos e relações quando necessário, constituindo-se assim em um modelo sofisticado de um sistema dinâmico não-linear multivariável.

4.3 MODELO DE UMA REDE COGNITIVA DINÂMICA

A DCN proposta nessa Tese é uma evolução dos mapas cognitivos *fuzzy* e incorpora novos tipos de relações e conceitos que as tornam mais aptas à modelagem de sistemas dinâmicos complexos. Para tratamento de incertezas, não-linearidades e inclusão de um caráter temporal nas relações entre conceitos utiliza-se uma relação causal cujo valor é variante no tempo. O valor desta relação pode ser obtido a partir de uma função trivalente, ou mesmo de uma função dinâmica no domínio de Laplace como proposto por Miao et al (2001), ou como resultado da execução de uma base de regras ou mesmo de um sistema *fuzzy*. Este tipo de relação, denominado relação de seleção, também confere às DCN a capacidade de raciocínio não monotônico.

De um modo geral, a DCN proposta nesta Tese possui os seguintes tipos de conceitos e relações:

Conceitos de entrada e saída de dados. Conceitos que fazem a interface com os dispositivos de entrada e saída. Esses conceitos têm suas entradas e saídas normalizadas.

Conceito de nível: Este conceito é utilizado quando se pode utilizar o valor absoluto de uma variável de um processo, e auxilia na representação da evolução dinâmica de um modelo físico, por exemplo.

Conceito de variação: Esse conceito representa a equação da diferença em primeira ordem de uma determinada grandeza em um intervalo de tempo conhecido. Deste modo, esse conceito pode, por exemplo, junto com conceito de nível auxiliar na construção de uma base de regras, para tomadas de decisões dinâmicas em um processo considerando o valor de uma variável e sua variação em um determinado contexto.

Conceito de decisão ou relação: Esses conceitos são utilizados quando se necessita mudar e/ou atualizar conceitos (variáveis) em uma DCN porem não ocorre interação com outros conceitos. Esses conceitos são inseridos na DCN através de relações de seleção.

Conceito de memória: A utilização deste tipo de conceito na DCN insere no modelo cognitivo uma capacidade de decisão que considera situações passadas do sistema. Esse conceito pode ser dividido em dois tipos: memória temporal será utilizada no controlador do robô móvel. E memória de dados, como por exemplo, armazenar os dados de campanha na aplicação do fermentador.

Conceito de fator: insere uma operação de multiplicação normalmente ponderada no conceito que está conectada na saída da DCN. Esse conceito é ativado por regras do tipo declaração condicional, por exemplo, as relações de seleção ou relações de fator.

Relação causal pura: são as relações que estabelecem a causa - efeito entre os conceitos de um mapa cognitivo como proposto por Axerold (1976) e Kosko (1986). As relações causais são calculadas através de uma matriz W constante.

Relação de fator: insere regra(s) para uma operação ou mais operações de multiplicação realizadas pelos conceitos de fator.

Relação causal variante no tempo: são as relações que estabelecem uma causa - efeito entre os conceitos de um mapa cognitivo, dirigida por uma função do

tempo como proposto por Miao et al (2001, 2010). Essas relações causais são calculadas através de uma matriz W em função do tempo.

Relação *Fuzzy*: é uma relação que permite a construção de tomadas de decisões do tipo não monotônica, e caracterizada quando se tem a influência direta de dois conceitos em um terceiro. A matriz W resultante também é variante no tempo de acordo com o resultado da execução de um sistema *fuzzy*.

Relação de seleção: essa relação permite representar a ocorrência de eventos e identificar quais são as relações de causa-efeito válidas em um dado instante de tempo. Esta relação é ativada através da execução de uma ou mais regras do tipo SE-ENTÃO, que tem seu disparo e duração de acordo com o fenômeno modelado. Estas relações são naturalmente variantes no tempo, desde que o valor da matriz W é resultante de um processo de inferência baseada em regras. Essa relação permite tomada de decisões em níveis mais altos, ou seja, permitem tomadas de decisões estratégicas orientadas a eventos.

4.4 ALGORITMO PARA CONSTRUÇÃO DE UMA DCN

A construção de uma DCN (quadro 3) segue inicialmente os mesmos passos utilizados no desenvolvimento de mapas cognitivos e mapas cognitivos *Fuzzy*, e que foi apresentado na quadro 1 do capítulo 3. Entretanto, para a construção de tomadas de decisões estratégicas, devem-se acrescentar mais etapas ou passos no desenvolvimento da DCN.

A utilização do algoritmo acima para modelagem de sistemas dinâmicos pode sofrer pequenas alterações, dependendo do tipo de problema que se quer modelar (controle supervisão, robótica, mundo virtual, tomada de decisão, etc.). Por exemplo, pode não ser necessário o uso de relações *fuzzy* ou de conceitos do tipo variação. Ou ainda, pode não ser necessária uma etapa de treinamento ou de aprendizado. Também os algoritmos utilizados em cada uma das etapas podem ser diferentes. No entanto, a construção de um modelo cognitivo baseado em DCN segue sempre a mesma sequência de etapas mostrada no algoritmo de um FCM clássico.

-
- Etapa 1:** Identificação dos conceitos e suas funções (entrada, saída, fator e seleção), as suas interligações, e / ou seleção das relações entre conceitos, determinando sua natureza causal (positiva, negativa, neutra) e seus tipos (*fuzzy*, seleção, fator, causal).
- Etapa 2:** Aquisição de dados inicial, através da opinião ponderada de especialistas e / ou análise de um modelo matemático, ou análise de dados e/ou comportamento dinâmico do sistema.
- Etapa 3:** Apresentação de dados sobre as opiniões de vários peritos para um sistema de inferência *fuzzy* que tem como resultado os valores de pesos da DCN.
- Etapa 4:** A construção da base de regras *fuzzy* que tem como inferência os valores de pesos da DCN para as relações *fuzzy*, caso esse tipo de relação seja implementada.
- Etapa 5:** Tratamento da informação, adaptação e otimização da DCN, ajustando as suas respostas para a saída desejada. Caso necessário, são empregados algoritmos de aprendizagem por reforço para refinamento dinâmico do modelo ou qualquer outro algoritmo de treinamento (AG, PSO, hebbiano, etc.).
- Etapa 6:** Construção da base de regras dos conceitos e relações de seleção, e, caso seja implementada, construção das regras de aprendizado por reforço.
- Etapa 7:** Validação do modelo DCN que é testado em condições de operação do sistema modelado.
-

Quadro 3. Construção de uma DCN

Fonte: Autoria própria.

Um dos passos mais importantes na qualidade do modelo obtido está na realização dos passos 1 e 2 do algoritmo acima. De fato, a abstração do conhecimento necessário para a realização dos passos não é uma simples tarefa e se alguma informação relevante ao comportamento do sistema não for modelada, certamente prejudicará as etapas seguintes na construção da DCN.

De um modo geral: as etapas 1, 2 e 3 são semelhantes ao desenvolvimento de um FCM clássico (DCNi) (Identificação, aquisição e números *fuzzy*). A etapa 4 (relação *fuzzy*) é específica de uma DCNi, porém modela causa e efeito de modo contínuo e é opcional. A etapa 5 - adaptação (FCM, DCNi ou DCN₁). A Etapa 6 - inclusão base de regras (nível estratégico) DCN_f ou DCN₂. E, finalmente, a etapa 7, etapa de validação do modelo, pode ser usada a cada nível de construção da DCN. Ou seja, inicialmente deve-se validar a DCNi (causa-efeito), depois valida DCN_f com a inclusão das ações de planejamento (base de regras, eventos) e A.R. (opcional). Observando-se que: O sucesso das etapas seguintes também depende do sucesso das etapas antecedentes. Deste modo, a validação de cada nível DCN (nível 1 e nível 2) atenua essa dificuldade. Além disso, a DCN nível 2 diferentes funcionalidades podem ser desenvolvidas (modularidade), neste contexto, pode-se validá-las separadamente.

Para facilitar a construção da rede dinâmica utilizando o algoritmo proposto, esta Tese também propõe uma arquitetura para a modelagem do conhecimento baseado em DCN, a qual é mostrada na figura 16.

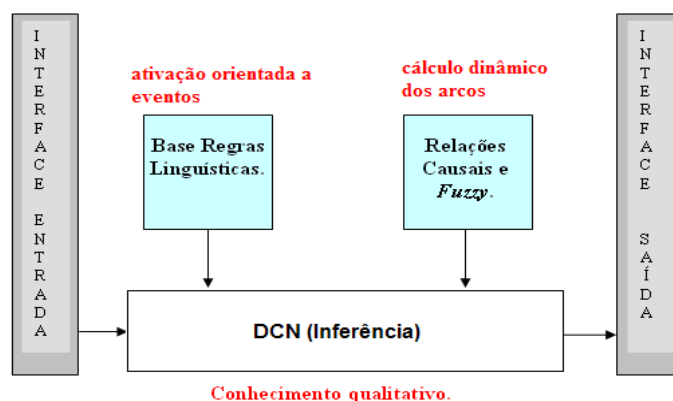


Figura 16 - Arquitetura DCN
Fonte: autoria própria

A arquitetura DCN, como mostra a figura 16, de um modo geral apresenta dois blocos extremos que representam as interfaces de entrada e saída, um bloco inferior que representa o motor de inferência e/ou tomada de decisão. Os dois blocos superiores representam a base de regras e relações para tratamento do conhecimento na forma dinâmica e de eventos. Essa representação em blocos auxilia no entendimento do funcionamento da arquitetura como um todo. Deste modo, as aplicações que empreguem DCN terão representações gráficas semelhantes, basicamente com

diferenças nas variáveis de entrada e saída. Na prática, esses blocos darão origem a conceitos e relações, como por exemplo, os blocos de interfaceamento com o ambiente em que o modelo será inserido, são feitos através de conceitos de entrada e saída da DCN. O bloco Relações causais e *fuzzy* representa as tomadas de decisões contínuas e representam as suas respectivas relações e os conceitos conectados, o bloco base de regras linguísticas representa as outras relações, em especial, as relações e conceitos de seleção e fator com suas respectivas regras que modelam as ações orientadas pela ocorrência de eventos.

De um modo geral, a arquitetura é baseada numa hierarquia de conhecimento com duas camadas ou níveis. No primeiro nível, o conhecimento é modelado e formalizado através de conceitos e relações causa-efeito, de forma semelhante à construção de um FCM na sua proposta original. No entanto, ainda no primeiro nível, podem ser necessários ajustes através de treinamento e/ou aprendizado por reforço para adequar o modelo inicial aos dados históricos do problema, tal qual foi apresentado no capítulo 3. Também neste nível é possível que as relações causais variantes no tempo sejam modeladas através de funções e/ou através de um sistema *fuzzy* relacional.

No segundo nível, modela-se um conhecimento heurístico mais ligado à análise avaliativa do ambiente em que está inserido o modelo, uma avaliação se os objetivos e/ou o desempenho do modelo estão sendo atingidos é realizada culminando na tomada de decisão. Em resumo, modela-se um conhecimento relativo ao planejamento estratégico da aplicação. Este conhecimento é modelado através de base de regras *fuzzy* ou não, as quais são disparadas quando o sistema se encontra em algum estado especial. O resultado da execução dessas regras gera valores de peso para as relações seja na mesma camada seja na camada inferior. Os conceitos inseridos no mapa neste nível em geral são do tipo fator, e tem por função ponderar os diversos conceitos envolvidos na tomada de decisão de acordo com o estado atual da DCN.

Esta proposta de arquitetura explora a capacidade de aquisição de conhecimento estruturado, com diferentes fontes de conhecimento, representadas pelos diferentes blocos. Essas fontes podem ou não ter a mesma origem, o que permite a construção de tomadas de decisões de baixo nível (relações causais) e estratégicas por um especialista ou um grupo de especialistas trabalhando de modo simultâneo ou não.

Neste contexto, essa arquitetura pode ser considerada um “*White board*” (COPPIN, 2010).

4.5 DCN APLICADA NA ESTRATÉGIA DE TIME DE FUTEBOL DE ROBÔS

O algoritmo proposto e a arquitetura de representação de conhecimento proposta na seção anterior serão validados num exemplo simples de futebol de robôs. De acordo com a literatura, o desenvolvimento de estratégias de times de futebol de robôs é um problema dinâmico de tomada de decisão fortemente influenciado por mudanças no ambiente, sendo por isso um exemplo de aplicação bastante utilizado para validação de sistemas inteligentes (COSTA; PEGORARO, 2000).

Diversos módulos compõem o sistema de um time de futebol robôs, os principais são: visão, estratégia, controle. Os módulos de visão e controle são responsáveis pela aquisição de dados e ações de controle do sistema. O módulo de estratégia define, para cada robô, uma posição alvo - posição a ser atingida no campo - e respectivas velocidades e orientação associadas para que estes dados possam ser traduzidos em termos de velocidades e sentido de rotação dos motores que comandam as rodas dos robôs. A definição da posição alvo depende do estado do jogo e dos comportamentos associados a cada robô.

O principal desafio do futebol de robôs é analisar o ambiente em tempo real, e tomar as decisões corretas para executar as ações necessárias ao objetivo do jogo, como por exemplo, assumir o controle da bola durante o jogo.

A estratégia empregada para construção de uma DCN para o futebol de robôs, desenvolvida aqui, é baseada no trabalho de Costa e Pegoraro (2000) que modela o problema através de uma máquina de estados capaz de tomar as decisões do time.

De fato, como uma DCN é um grafo, também pode ser considerada uma máquina de estado e a tradução da estratégia proposta por Costa e Pegoraro (2000) em uma DCN é facilmente realizada.

A estratégia a ser modelada é desenvolvida para dois jogadores de linha e um goleiro e é implementada através de ações simples tais como ir a determinado ponto do campo ou seguir a bola. A estratégia global é formada por uma composição dessas

estratégias mais simples com base na situação do jogo. As funções de cada jogador variam de acordo com a sua posição em campo. O goleiro deve permanecer na área do gol e tem como objetivo evitar que a bola passe da linha que limita o gol se locomovendo lateralmente. O jogador de linha pode funcionar como zagueiro se estiver posicionado no seu campo defensivo ou como atacante se estiver no campo adversário. O zagueiro tem a função de evitar que algum jogador adversário aproxime-se do gol com a bola sob controle. Já o atacante tem a função de recuperar a bola e conduzi-la até o gol do adversário, tomando decisões de qual ângulo e com que intensidade chutará a bola em direção ao gol. Por simplicidade de funcionalidades, quando um jogador ataca o outro defende.

De acordo com a estratégia apresentada, a heurística de controle para um mesmo jogador que pode atuar como atacante ou como zagueiro corresponde passo a passo à seleção de uma das ações listadas a seguir:

- Chutar em direção ao gol;
- Caminhar junto com a bola (bola com o jogador);
- Ir para a posição de ataque;
- Ir para a posição de defesa;
- Ir ao encontro da bola.

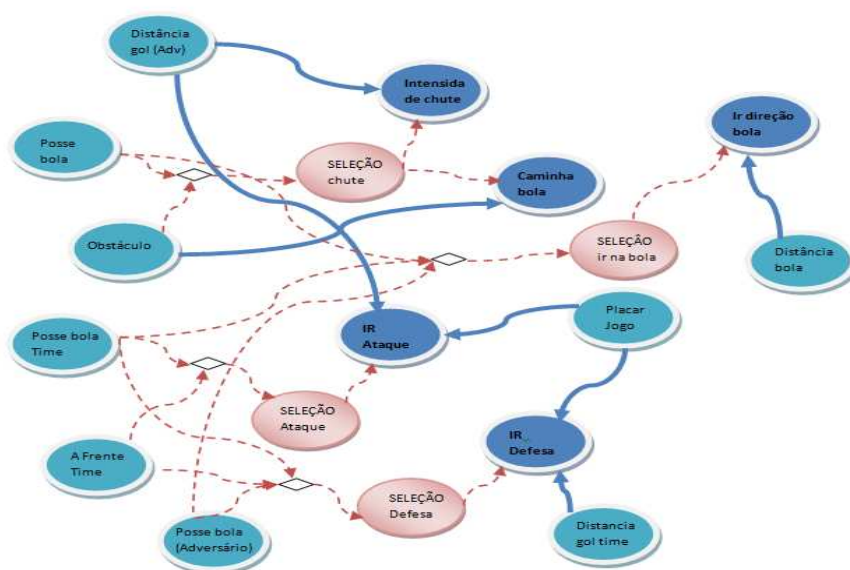


Figura 17 - DCN - Estratégia de Futebol Robôs
Fonte: Autoria Própria

A figura 17 mostra a DCN que modela a estratégia de um jogador de linha. Nessa figura os conceitos em azul claro são as variáveis de entrada, os conceitos em azul escuro representam as saídas que são as tomadas de decisões e os conceitos em vermelho claro representam conceitos de seleção. Observa-se que esses conceitos de seleção poderiam também ser tratados como conceitos de fator binário, onde o fator assume valores (0 ou 1). Os arcos (azul contínuo) representam as relações causais. Os arcos (vermelho tracejado) representam as relações de seleção, e indicam os antecedentes (variáveis ou conceitos de entrada), e os seus consequentes (variáveis ou conceitos de saída). De um modo geral, a arquitetura proposta pode ser dividida em dois níveis. No nível mais baixo (nível 1, azul), as relações causais representam ações que estão sempre ocorrendo, por exemplo, a influencia da variável (conceito) placar do jogo que tem função de incentivar o ataque ou defesa do jogador em qualquer circunstância. O quanto o placar do jogo influencia na tática de ataque ou defesa do time será decidido pelo especialista e será o peso da relação causal, que neste caso poderia ser “fraca”. Além do peso deve se definir se a relação tem uma influencia “positiva”, por exemplo, na ação ir para defesa e “negativa” para a ação ir ao ataque.

No nível mais alto (nível 2, vermelho), a estratégia é mais elaborada e ocorre em situações específicas (orientada a eventos) e são modeladas por uma base de regras e das relações de seleção, nesse caso específico da estratégia de futebol, nos capítulos 5 e 6 outras relações e conceitos também são utilizados para essa mesma função. Como o próprio nome sugere, essas relações selecionam qual a ação deverá ser executada e, deste modo caracteriza a ocorrência de eventos, lembrando que raciocinar por eventos é uma maneira alternativa de representar o tempo (COPPIN, 2010). A tomada de ações é executada de acordo com uma base de regras modeladas por implicações linguísticas. Por exemplo, a ação de chutar a bola é definida pela seguinte regra:

Se jogador tem “posse bola” e “não tem obstáculo” **então** “chuta”

A intensidade do chute será o valor do conceito que é calculado de acordo com a distância ao do gol adversário e o peso da relação causal, que neste caso deve ser “positiva” e “forte”. Porém, caso o jogador tenha a posse de bola e tenha um obstáculo à frente, a seguinte regra será acionada.

Se jogador tem “posse bola” e “tem obstáculo” **então** “caminha com a bola”

A velocidade com que o jogador caminha com a bola dominada será determinada de maneira análoga por outro conjunto de regras. Essas regras consideram a distância do outro jogador (obstáculo), e a respectiva intensidade da relação causal. Outro exemplo para melhor ilustrar a lógica da DCN é a tomada de decisão de ir ao encontro da bola. Nesse caso, o conceito de seleção usa a seguinte regra:

Se jogador não tem “posse bola” e “jogador do mesmo time não tem posse bola” e “jogador adversário não tem posse bola” **então** “ir em direção da bola”

A velocidade com que o jogador vai ao encontro da bola será o valor do conceito “ir direção bola”, calculado de acordo com o valor do conceito de entrada “distância do jogador” e da respectiva relação causal que deve ser “positiva”, “média” ou “pouco forte”. De maneira análoga, considerando as regras, as relações causais e os valores atribuídos aos respectivos conceitos, são realizadas as outras ações modeladas na DCN, como intensidade do chute, por exemplo.

E, finalmente, como foi citado anteriormente, para ser executada uma ação, por exemplo, a ação de chute, deverá ser calculada a direção “ângulo” do chute ao gol do adversário. Deste modo, essa ação é executada no instante que ocorre o respectivo evento, com a direção calculada (de acordo com a geometria e posição do jogador) e a intensidade do chute (conceito de saída da DCN).

Deste modo, a DCN da figura 17 representa um modelo de estratégia de futebol de robôs. Como o objetivo deste exemplo é apenas de apresentar as diferenças entre as relações causais e não causais, principalmente a capacidade de se modelar eventos (tempo) da ferramenta não serão apresentados resultados reais ou simulados de uma partida de futebol. Deste modo, as implicações linguísticas e as relações causais foram apresentadas somente de modo qualitativo. Para uma aplicação real os valores dos conceitos e das relações causais deverão ser quantificados, como por exemplo, atribuir um valor para o conceito “distancia ao gol adversário” de acordo com a geometria e a posição no campo.

4.6 CONCLUSÃO DO CAPÍTULO

O exemplo qualitativo apresentado neste capítulo ilustra bem as diferenças entre relações causais e não causais modeladas, bem como os diferentes tipos de conceitos propostos para desenvolvimento de modelos baseados em DCN.

O exemplo também mostra a capacidade de inferência da DCN em combinar duas técnicas de representação de conhecimento de forma estruturada, seja através de grafos, seja através de base de regras.

A partir da arquitetura proposta e do algoritmo de modelagem desenvolvido pode-se considerar que a DCN evolui de um FCM, caso se modele inicialmente os conceitos e as relações causais e depois se refine a construção com a ocorrência de eventos, que caracterizam as ações da estratégia e planejamento adotados. A ocorrência desses eventos tem duas características: o disparo e o tempo de duração desse evento.

Os capítulos 5 e 6 formalizam e utilizam a metodologia para a construção de sistemas inteligentes distintos. Esses capítulos descrevem o modelo inteligente desenvolvido desde a aquisição de conhecimento, incluindo justificativas de inclusão de novos tipos de conceitos e relações, até os resultados obtidos, passando por todos os passos de construção e validação de uma DCN, citados no quadro 2.

5 MODELO DE UM SISTEMA SUPERVISÓRIO BASEADO EM DCN

5.1 INTRODUÇÃO DO CAPÍTULO

Este capítulo apresenta o desenvolvimento de um sistema supervisório para um processo de fermentação alcoólica. Esse sistema supervisório emprega uma DCN e consequentemente dois níveis de tomadas de decisões na campanha do processo. No primeiro nível os *set-points* são enviados para controladores PID objetivando ações de otimização, ou melhor, desempenho no processo. No segundo nível, as tomadas de decisões em nível estratégico ou de planejamento são objetivadas para garantir, por exemplo, a ocorrência da reação química.

O sistema tem dois níveis de controle, correspondendo às duas camadas da DCN e é orientada a eventos, característica desejável para um sistema supervisório. Conclusões sobre a aplicação e desenvolvimento da DCN, juntamente com os resultados dos comparativos finalizam o capítulo. Um comparativo de desempenho com controladores PID, nas mesmas condições de operação, valida a proposta.

5.1.1. PROCESSO DE FERMENTAÇÃO ALCOÓLICA

O processo de fermentação utilizado nas simulações deste artigo foi estudado inicialmente por Maher (1995) e tem servido como *test-bed* para validar arquiteturas de controle com diferentes enfoques (MELO JUNIOR et al., 2002; FABRO; ARRUDA, 2003; NEVES JUNIOR; ARRUDA; MENDONÇA, 2009). Um diagrama representando o processo é apresentado na figura 18. Existem duas válvulas, *Fin* e *Fout*, que controlam, respectivamente, o fluxo de substrato inserido no tanque e o fluxo de produto (fermentado) retirado. Estas válvulas são controladas por dois controladores independentes do tipo Proporcional – Integral - Derivativo (PID). O processo apresenta quatro variáveis de estado: as concentrações de substrato (s), de biomassa (c), de produto (P) e o volume (V) do tanque onde ocorre a fermentação. As concentrações são dadas em gramas/litro (g/l), e o volume em litros (l). As equações diferenciais

descrevendo o comportamento dinâmico do processo são dadas nas equações (24) a (27).

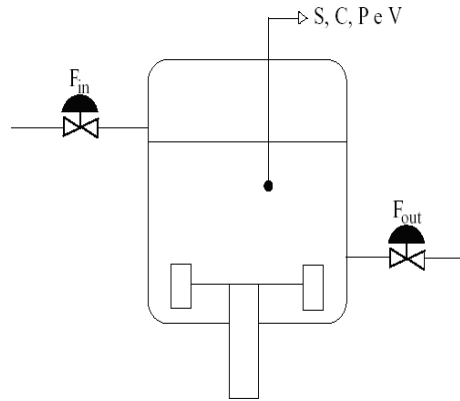


Figura 18 – Processo de Fermentação Alcoólica

Fonte: (MELO JUNIOR et al., 2002)

$$\frac{dS}{dt} = -\frac{1}{Y_{C/S}} \mu C + \frac{F_{in}}{V} S_a - \frac{F_{out}}{V} S, \quad (24)$$

$$\frac{dC}{dt} = \mu C - \frac{F_{out}}{V} S, \quad (25)$$

$$\frac{dP}{dt} = -\frac{Y_{P/S}}{Y_{C/S}} \mu C - \frac{F_{out}}{V} P, \quad (26)$$

$$\frac{dV}{dt} = F_{in} - F_{out}. \quad (27)$$

Onde S_a é a concentração de substrato da alimentação, $Y_{P/S}$ e $Y_{C/S}$ são as constantes de conversão de biomassa e produto respectivamente. A variável μ representa a função de crescimento da biomassa durante o processo de fermentação e é dada por:

$$\mu = \mu_0 \frac{S}{K_s + S} \left(1 - \frac{P}{P_m} \right) \quad (28)$$

O valor inicial μ_0 corresponde à máxima taxa de crescimento, K_s é a constante de *Michaelis-Menton*, e P_m é o coeficiente de inibição do produto durante a reação. Nesta Tese, as equações (24) a (28) são utilizadas para simular o processo de fermentação e são resolvidas através do método de Runge-Kutta de 4ª ordem.

Este processo é não-linear e apresenta um comportamento de fase não mínima e tempo de acomodação grande, como mostra a figura 19. Além disso, por medida de segurança quando o volume do reservatório excede 3,5 litros, a válvula *Fin* é completamente fechada e se ao contrário o volume cair abaixo do mínimo (1,5l), a válvula de saída *Fout* é completamente fechada. Estas duas ações são automáticas e funcionam como perturbações durante a evolução do processo.

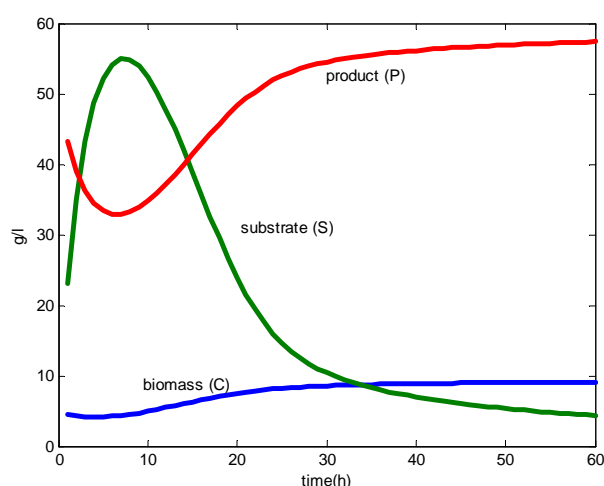


Figura 19 – Comportamento Dinâmico das Variáveis de Estado do Processo de Fermentação
Fonte: MELO JUNIOR et al., 2002

5.2 CONTROLE SUPERVISÓRIO COM DCN

A utilização de sistemas hierárquicos para o controle supervisório é amplamente aceita na literatura (PASSINO; YOURKOVICH, 1998). Uma arquitetura hierárquica baseada em FCM é proposta em Stylios e Groumpos (2004) para supervisão de processos complexos. Esta arquitetura, adaptada para DCN é mostrada na figura 20, e possui um supervisor no nível superior composto por várias DCNs interconectadas,

cada uma delas realizando uma tarefa específica do controle supervísório, tais como detecção de falhas, monitoramento, planejamento, tomada de decisão entre outros.

De acordo com esta arquitetura, desenvolve-se neste capítulo um modelo baseado em DCN para supervisionar a operação do processo de fermentação durante uma campanha com objetivos fixos, substituindo o operador do sistema por um supervisor. O objetivo é minimizar os efeitos de transição de *set-points*, determinados pelas mudanças de ponto de operação, sobre as variáveis do processo e dispositivos (válvulas), além de melhorar o desempenho da reação, ao mesmo tempo em que satisfaz restrições operacionais.

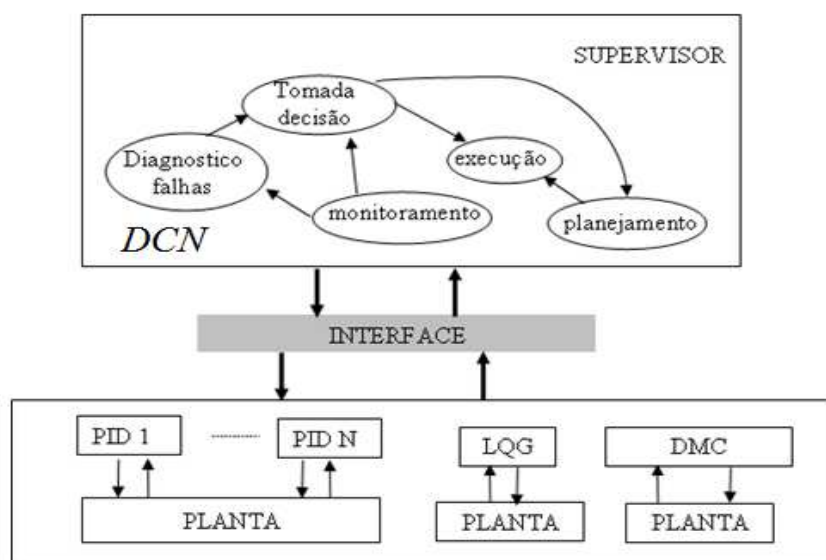


Figura 20 – Arquitetura baseada em DCN para controle supervísório
Fonte: Adaptado de STYLIOS e GROUMPOS (2004)

A construção do modelo baseado em DCN segue os passos do quadro 3 apresentado no capítulo 4 e, como explicado, é realizada em duas camadas, as quais podem ser consideradas duas DCNs distintas. A primeira, relativa à camada de nível mais baixo, corresponde ao modo de operação normal do processo e outra, relativa à camada de supervisão propriamente dita, é responsável pela supervisão do processo em modo de operação crítico. Estas redes foram desenvolvidas, testadas e validadas

separadamente e em seguidas unidas para formar uma DCN maior que realiza a tarefa de supervisão do processo.

As duas DCNs foram construídas a partir da observação do comportamento do processo em diferentes situações e das relações entre as suas variáveis físicas (Substrato, Biomassa, Produto e o Volume) e os dispositivos de controle (Válvulas de entrada *Fin* e saída *Fout*), após ações de controle. Para isto, além de se analisar as equações diferenciais (24) a (28), que descrevem a reação química de fermentação, procedeu-se a uma análise de causa e efeito entre as variáveis físicas (biomassa, produto, substrato e volume) do processo para diferentes valores de *set-points* dos PIDs que acionam as válvulas *Fin* e *Fout*, caracterizando os diversos pontos de operação do fermentador.

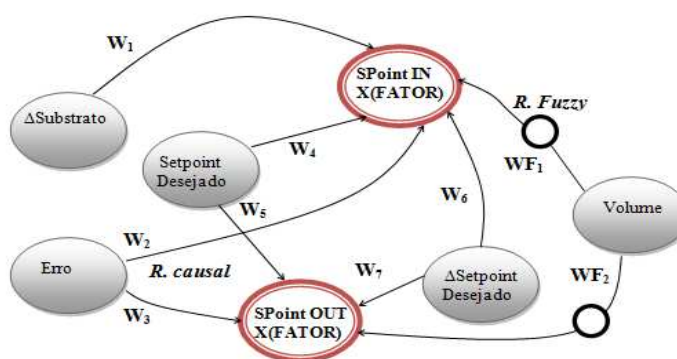


Figura 21 – DCN_n/FCM para Operação em Modo Normal
Fonte: autoria própria

Como resultado observou-se o seguinte comportamento para o processo:

- A concentração de produto no reator, variável controlada do processo, é afetada por variações no *set-point* do PID que controla a válvula de entrada. Este comportamento é modelado através de uma relação causal representada por um arco (W_2) na figura 21. DCN
- A concentração de produto no reator, variável controlada, é afetada por variações no *set-point* do PID que controla a válvula de saída. (W_3)
- O valor do *set-point* desejado do processo, que caracteriza o ponto de operação da reação, determina o valor de *set-point* efetivamente aplicado ao PID que controla a válvula de entrada. (W_4)

- O valor do *set-point* desejado do processo, que caracteriza o ponto de operação da reação, determina o valor do *set-point* efetivamente aplicado ao PID da válvula de saída. (W₅)
- A variação de *set-point* desejado (mudança de ponto de operação) afeta o valor do *set-point* aplicado à válvula de entrada. (W₆)
- Variação do *s* *set-point* desejado (mudança de ponto de operação) afeta o valor do *set-point* aplicado à válvula de saída. (W₇)
- Uma variação acentuada na concentração (superior a 3g/l) de substrato no reator determina uma variação no *set-point* do PID que controla a válvula de entrada, a fim de garantir o equilíbrio da reação de fermentação. (W₁)
- Esta variação da concentração de substrato não influencia o *set-point* do PID da válvula de saída, pois a inserção de substrato é feita por Fin, e válvula Fout está relacionada à retirada de produto.
- O volume no tanque é afetado pelo valor do *set-point* da válvula Fin, no entanto esta relação é assimétrica e possui uma restrição quanto ao volume máximo, sendo modelada como uma relação do tipo Fuzzy, como será explicada a seguir. (WF₁)
- O volume no tanque é afetado pelo valor do *set-point* da válvula Fout, no entanto esta relação também é assimétrica e possui uma restrição quanto ao volume mínimo, sendo modelada como uma relação do tipo Fuzzy, como será explicado a seguir. (WF₂)

A partir deste comportamento, é construída a primeira DCN, que é semelhante a um FCM, para a operação do fermentador, mostrado na figura 21. Esta DCN corresponde à operação do processo em modo normal e será denominada DCNn. Como pode ser visto na figura 21, a DCNn corresponde a um grafo acíclico simples, representando o modo como um operador ajusta os *set-points* efetivamente aplicado aos PIDs do processo. O objetivo do supervisor é minimizar os efeitos de transição de *set-points*, determinado pelas mudanças de ponto de operação, e melhorar o desempenho da reação, ao mesmo tempo em que satisfaz as restrições de operação, tal como manter o volume do produto dentro de seus limites.

A obtenção de um grafo acíclico ou mapa cognitivo trivial está de acordo com Miao e colaboradores (2001), que atestam que na construção por etapas de grandes mapas cognitivos sempre aparecem mapas menores geralmente acíclicos, que correspondem a relações de causa-efeito unidirecional bem definidas.

Na DCN da figura 21, as variáveis físicas: produto, volume e o *set-point* desejado são conceitos de entrada do tipo nível e tem seus valores normalizados no intervalo [0, 1] a partir de medidas do processo. As variáveis “variação de substrato” e “variação de *set-point* desejado” são conceitos de entrada do tipo variação, e as variáveis *set-point* F_{in} e *set-point* F_{out} são variáveis de saída do tipo decisão, e assumem valores no intervalo [0, 1]. O valor de *set-point* efetivamente aplicado na entrada do PID corresponde ao valor desta decisão multiplicado pelo *set-point* desejado.

As relações w_2 a w_7 são relações causais cujos valores são obtidos a partir de dados simulados. Já a relação w_1 é uma relação causal expressa por uma função trivalente, determinada por restrições de equilíbrio da reação química, e dada por:

$$w_1 = \begin{cases} w_1, & \text{se } \Delta\text{substrato} > 3 \\ 0, & \text{se } -3 < \Delta\text{substrato} < 3 \\ -w_1, & \text{se } \Delta\text{substrato} < -3 \end{cases} \quad (29)$$

Onde $\Delta\text{substrato}$ corresponde à variação de substrato ocorrida entre dois instantes de tempo consecutivos.

A matriz de conexão da DCNn inicial obtida a partir de dados de simulação é dada no quadro 4.

Conexão	Valor
W_1	-0,1000
W_2	0,2500
W_3	0,2500
W_4 (<i>Set-point</i> =35)	-0,8845
W_4 (<i>Set-point</i> =40)	-0,5070
W_4 (<i>Set-point</i> =45)	-0,2230
W_5	0,2500
W_6	-0,5000
W_7	-0,1000

Quadro 4 - Relação Causal do DCN₁
Fonte: autoria própria

A intensidade da relação modelada por W_4 depende do ponto de operação do processo (3 valores de *set-point* para as campanhas utilizadas) e está associado ao comportamento da reação química de acordo com a sequência de *set-points* que formam a campanha. Estes valores foram ajustados por algoritmo genético de modo semelhante ao procedimento realizado no capítulo 3 para o reator. Mais detalhes deste procedimento e uma avaliação do método podem ser encontrados em (NEVES JUNIOR; ARRUDA; MENDONÇA, 2009).

As relações WF_1 e WF_2 são relações *Fuzzy* cujos valores são obtidos a partir de uma base de regras *Fuzzy*. A criação de uma relação *Fuzzy* torna-se necessária sempre que a natureza das variáveis observadas é não monotônica, isto é, as variáveis não são crescentes ou decrescentes ou estritamente crescentes e decrescentes. Esta situação ocorre com o volume em presença de variações negativas ou positivas de *set-points*,

para ambas as válvulas. Para ilustrar tal comportamento, a figura 22 mostra o volume e sua variação quando ocorrem mudanças no *set-point* do PID associado à válvula *Fin*. Observa-se que as variações de volume são mais acentuadas quando ocorre maior variação no *set-point*. De acordo com esse comportamento, foi criado um sistema *Fuzzy* do tipo Mandami (PASSINO; YOURKOVICH, 1999) que tem como entrada os valores de volume e variação de volume e como saída o valor da relação *Fuzzy* WF_1 que altera o conceito do *set-point* efetivamente aplicado na válvula *Fin*.

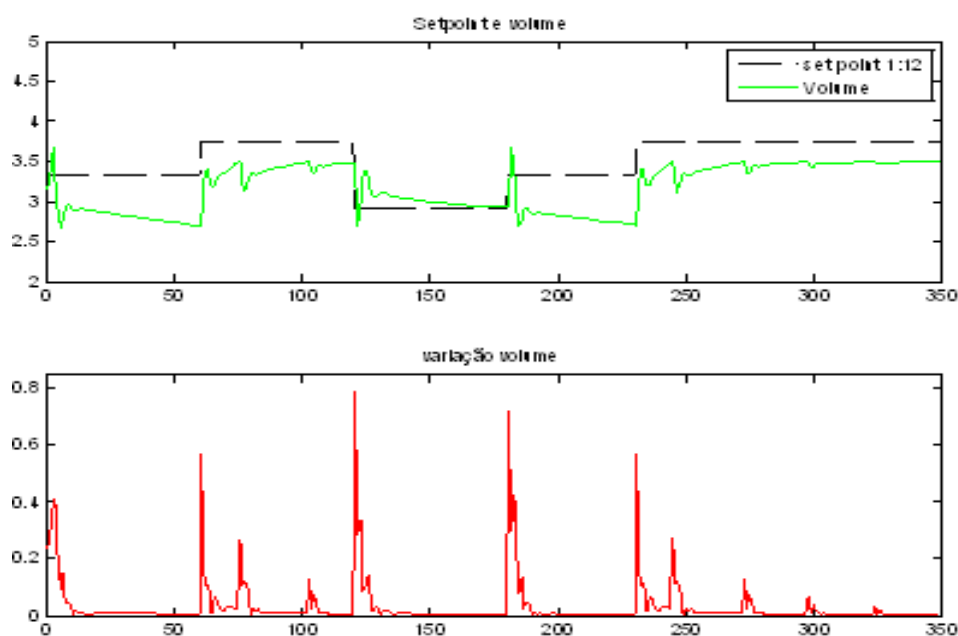


Figura 22 – Variação Volume, Volume e *Set-point*

Fonte: autoria própria

As seguintes regras são utilizadas para calcular a relação *Fuzzy* WF_1 :

1. Se volume é baixo e variação do volume é média então WF_1 é alta.
2. Se volume é baixo e variação do volume é alta então WF_1 é alta.
3. Se volume é baixo e variação do volume é baixo então WF_1 é igual a 1.
4. Se volume é alto e existe variação do volume então WF_1 é baixa
5. Se volume é médio existe variação do volume então WF_1 é média.

A figura 23 mostra a superfície *Fuzzy* associada à base de regras desenvolvida que descreve o efeito não-linear (WF_1) causado pelas variáveis de entrada volume e variação de volume sobre o *set-point* aplicado ao PID da válvula F_{in} .

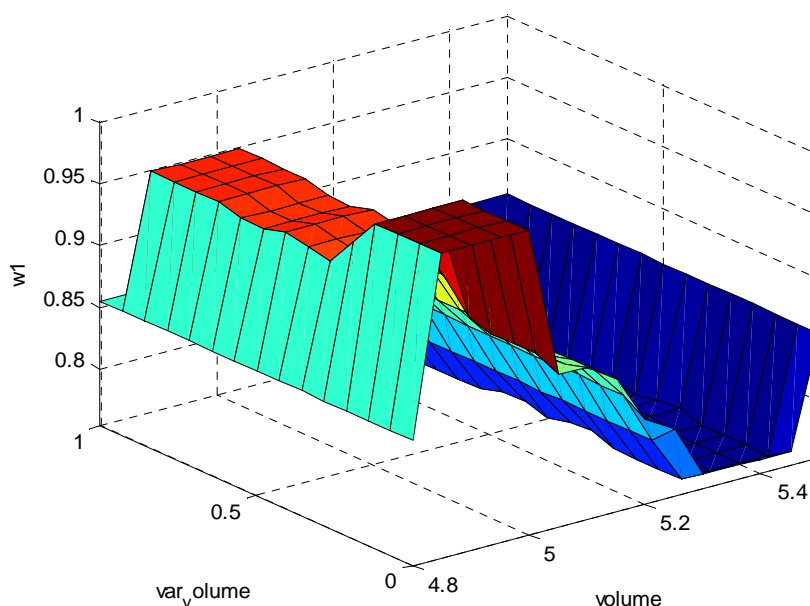


Figura 23 – Superfície *Fuzzy* da Relação WF_1
Fonte: autoria própria

De modo semelhante, uma relação *Fuzzy* foi criada para relacionar os conceitos volume e *set-point* da válvula F_{out} através da relação dada por WF_2 .

A DCNn representada na figura 21 foi testada para diferentes valores de *set-point* no intervalo 10 a 50g/l e em todos eles estabilizou em no máximo 2 iterações. Por exemplo, para um valor de *set-point* de 40g/l, o quadro 5 mostra a evolução dos conceitos obtidos até a estabilização. Este comportamento era esperado, pois de acordo com os teoremas demonstrados em (MIAO; LIU, 1999), um FCM ou uma DCN que não contém ciclos (FCM/DCN trivial) atinge um estado estático em no máximo L iterações, onde L é o tamanho de sua maior trajetória.

Conceitos	Valor inicial	Primeira iteração	Segunda iteração	Terceira iteração
<i>Set-point</i> in	0,35	0,420	0,420	0,420
<i>Set-point</i> out	0,35	0,421	0,429	0,429

Quadro 5 - Evolução da DCNn para uma variação de *Set-point* de 35 Para 40g/L
Fonte: Autoria Própria.

No entanto, foi constatado que em algumas campanhas de longa duração, a concentração de substrato e/ou de biomassa no interior do reator pode cair a níveis que não permitam a realização do processo de fermentação. Neste caso, a operação torna-se crítica e é necessário recuperar os componentes do processo através de uma variação de *set-point*, independente dos valores determinados para a campanha. Para isto foi desenvolvido um novo modelo que se interliga à DCN da figura 21, para a determinação dos *set-points* enviados a ambos os controladores PID. Este novo modelo corresponde a uma DCN para operação em modo crítico e será denominado DCNc.

O novo modelo cognitivo tem por função monitorar condições críticas no processo e assim possui uma inferência orientada a eventos. A DCNc construída utiliza um outro tipo relação, denominada relação de seleção, que é modelada através de regras do tipo **SE-ENTÃO** (implicação linguística) que ativam um conceito cujo valor é retirado de um conjunto de valores discretizados. Neste caso, as seguintes restrições devem ser modeladas:

1. A quantidade de biomassa não deve superar uma concentração máxima de 10 g/l.
2. Como explicado anteriormente, o volume no reator não pode exceder 3l, nem cair abaixo de 1,5l.
3. A quantidade de substrato não pode cair abaixo de uma concentração mínima de 1 g/l.
4. Os valores de *set-point* para concentração de produto correspondem a uma lista de valores de concentração discretizados no intervalo [10, 50] g/l descrevendo os pontos de operação do processo.

O modelo construído para operação crítica é mostrado na figura 24. Três novos conceitos associados às variáveis do processo: “seleção substrato”, “seleção biomassa” e “seleção volume” são inseridos para modelar o processo em operação crítica. Esses três conceitos são do tipo seleção e influenciam o valor do *set-point* a ser aplicado no processo (conceito “*Setpoint* Processo IN” na figura 24). O número de regras associadas aos conceitos de seleção irá depender das condições modeladas. A relação de seleção de volume emprega duas regras que testam os limites máximos e mínimos para o volume, enquanto que as relações de seleção de substrato e biomassa empregam somente uma. Essas relações e conceitos são resultantes do conhecimento empírico sobre a operação do processo em modo crítico. Neste contexto, essa DCN representa o modo como um operador ajusta os *set-points* efetivamente aplicados aos controladores PIDs do processo, quando é necessário recuperar os componentes da reação química. O objetivo do supervisor é minimizar os efeitos de transição de *set-points*, determinado pelas mudanças de ponto de operação, e melhorar o desempenho da reação, ao mesmo tempo em que satisfaz as restrições de operação para manter o volume do produto dentro de seus limites.

De um modo geral, o conceito de seleção representa a ação que executa uma mudança no conceito relacionado e a relação de seleção representa o mecanismo que dispara a ativação do conceito. Para a DCNc da figura 24, as restrições 1 a 4 citadas acima são expressas através das seguintes regras:

- Se o volume é maior que um limiar máximo, e o *set-point* da variável de entrada “Set point processo IN” é menor que o máximo então incrementar o valor atual do *set-point* desejado na campanha (“Set point processo IN”). Isto é, seleciona-se como *set-point* de entrada o próximo valor da lista de pontos de operação do processo via conceito “seleção volume” (**WS₂**).
- Se o volume é menor que o mínimo e *set-point* da variável de entrada “Set point Processo IN” é menor que o mínimo então decrementar valor atual de *set-point* via conceito “seleção volume” (**WS₂**).
- Se a concentração de substrato é menor do que mínimo e o *set-point* atual é menor que o máximo então incrementar o valor de *set-point* atual do

conceito de seleção “Set point Processo IN”, via conceito “seleção substrato” (WS_1).

- Se a concentração de biomassa é maior que o limite máximo e *set-point* atual é igual ao máximo então decrementar o valor de *set-point* atual do conceito de seleção “Set point Processo IN”, via conceito “seleção biomassa” (WS_3).

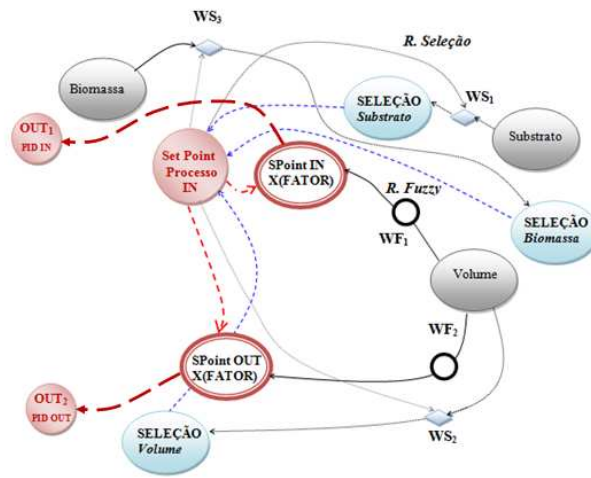


Figura 24 – DCNc Para Operação em Modo Crítico
Fonte: autoria própria

A conexão entre os dois modelos cognitivos se dá através das duas variáveis “**SPoint IN**” e “**SPoint OUT**” as quais se conectam às variáveis de interface (OUT_1 e OUT_2) que enviam valores para o fermentador. Também o conceito “volume” é o mesmo para ambas DCNs. O valor das variáveis OUT_1 e OUT_2 é resultado do produto entre os conceitos de saída dos dois modelos, equações (30) e (31).

$$out_1 = (SPoint IN) \times [(WF1) \times (Setpoint Processo In)] \quad (30)$$

$$out_2 = (SPoint OUT) \times [(WF2) \times (Setpoint Processo In)] \quad (31)$$

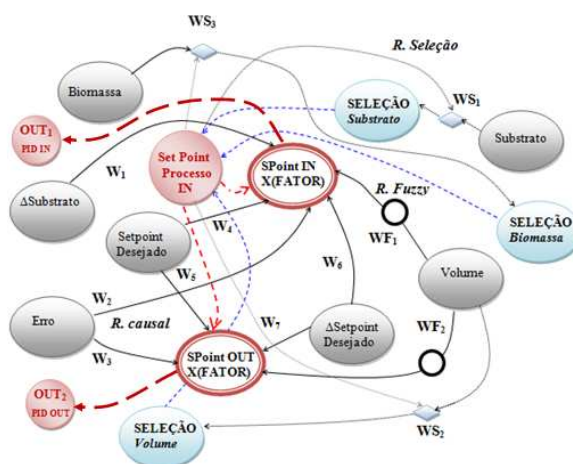


Figura 25 - Supervisor Baseado em DCN
Fonte: autoria própria

A DCN final assim construída (composta pela superposição da DCNn + DCNc) corresponde à arquitetura de controle supervisorio mostrada na figura 25, que é semelhante à arquitetura base proposta no capítulo 4. Nesta arquitetura coexistem dois tipos de decisões: as decisões resultantes da análise de relações causa-efeito (relações causais e *Fuzzy*) e decisões orientadas a eventos, resultantes da análise de condições específicas de operação do processo, modeladas através de uma base de regras (DCNc).

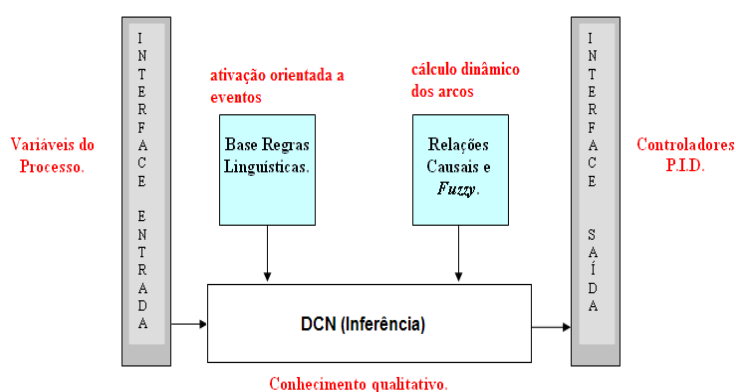


Figura 26 - Arquitetura do Supervisor
Fonte: autoria própria

As DCNs apresentadas nas figuras 21, 24 e 25 são utilizadas na próxima seção para controlar o processo de fermentação.

5.3 RESULTADOS DE SIMULAÇÃO

Três conjuntos de testes foram executados para avaliação de desempenho e robustez dessa arquitetura. Em todos eles foi simulada uma campanha que tem por objetivo levar o fermentador de um estado inicial caracterizado por uma concentração de aproximadamente 25 g/l de produto até um estado final em que a concentração de produto é de 45g/l, passando por 3 pontos de operação intermediários caracterizados por valores de *set-point* de 30, 35 e 40g/l.

No primeiro experimento, o processo é considerado operando em modo normal e o desempenho do controle regulatório é comparado ao controle supervísório realizado através da DCNn da figura 21. Convém ressaltar que no controle regulatório, os dois controladores PID atuam a partir de um *set-point* único e o incremento de controle calculado é somado ao valor atual da válvula *Fin* e decrementado do valor atual da válvula *Fout*. No controle supervísório, a operação das válvulas é independente, o incremento de controle é calculado a partir de dois conjuntos de *set-points* distintos, um para cada PID do nível regulatório, gerados pela DCN. Além disso, quando uma das válvulas é completamente fechada, a outra também será por medida de segurança.

No segundo experimento, a mesma campanha é realizada, e conforme será explicado a seguir, a reação atinge um estado crítico após um determinado instante de tempo, sendo necessária uma operação mais criteriosa do processo. Neste caso, avalia-se o desempenho do controle supervísório realizado através da DCN da figura 25, o qual incorpora os dois modos de funcionamento (em operação normal e crítica).

No último experimento, a mesma campanha é realizada, no entanto uma falha é simulada, forçando a válvula *Fin* a permanecer temporariamente fechada durante certo instante de tempo, o que leva a uma perturbação não controlada do processo. O desempenho do controle regulatório é novamente comparado com o controle supervísório com a DCN dada na figura 25, a fim de testar o modo de operação crítica quando uma causa externa (fechamento de válvulas) leva a um desequilíbrio da reação. Em todos os experimentos as condições iniciais são dadas no quadro 6.

Variáveis		Constantes	
S	4.5 g/l	S_a	100 g/l
C	5 g/l	P_m	100
P	50 g/l	K_s	10
V	2 l	μ_0	0,31
		$Y_{P/S}$	0,07
		$Y_{C/S}$	0,44

Quadro 6 – Condições Iniciais
Fonte: Autoria Própria

No experimento 1, a fim de validar a arquitetura proposta, os seguintes critérios de desempenho são calculados para cada arquitetura: índice ITAE (*Integral Time of Absolute Value Error*) e as variâncias dos sinais de controle e saída do processo. O ITAE foi escolhido por ser um indicador de desempenho largamente aceito na comunidade de controle para avaliação de malhas PID. A variância observada no sinal de saída $y(t)$ do processo mede a robustez do sistema em malha fechada na presença de perturbações e a variância do sinal de controle $u(t)$ é um indicador da suavidade das ações de controle efetivamente utilizadas (ASTRÖM; WITTENMARK, 1995). Ressalta-se que: quanto mais baixos forem os valores obtidos pela variância e o ITAE, melhor será o desempenho do controlador.

As figuras 27, 28 e 29 mostram os resultados das arquiteturas de controle obtidas com o primeiro conjunto de *set-points*. A figura 27 traz os resultados para o controle regulatório apenas com o PID, o *set-point* aplicado a ambos os controladores é o mesmo. No entanto, como explicado anteriormente, o incremento de controle calculado a partir do erro é adicionado à válvula **Fin** e decrementado da válvula **Fout**. As saídas dos PIDs estão mostradas, bem como os valores das variáveis de estado concentração, substrato, biomassa e volume. Os resultados para o controle supervisório

são dados nas figuras 28 e 29. A figura 28 traz os diferentes *set-points* aplicados aos PIDs e as respostas destes controladores, a figura 29 mostra a evolução das variáveis de estado do processo. Os índices de desempenho deste experimento são dados no quadro 7.

Critério	PID	DCN-PID
ITAE	1,0729	1,0242
$\sigma^2_{y \text{ (produto)}}$	14,2712	14,5266
$\sigma^2_{\text{(erro)}}$	1,2104e-04	9,2023e-05
σ^2_{Fin}	0,0156	0,0177
σ^2_{Fout}	0,0133	0,0132

Quadro 7 - Índices de Desempenho Para o Experimento 1
Fonte: Autoria Própria

Para a análise desses resultados, convém ressaltar que todos os valores só foram calculados até o instante $t = 285h$. Isto porque a partir deste instante para ambas as arquiteturas de controle, a reação atinge um valor limite, nesse modelo, com volume igual a 3,5l, determinando o fechamento da válvula de entrada e em consequência a válvula de saída também é fechada. De fato, a partir deste instante a biomassa atinge sua concentração máxima de 7g/l, o substrato é inteiramente consumido e não acontece mais a reação química, ou seja, o processo pára. Esta situação pode ser comprovada através da análise do comportamento das variáveis citadas nas figuras 28 e 29.

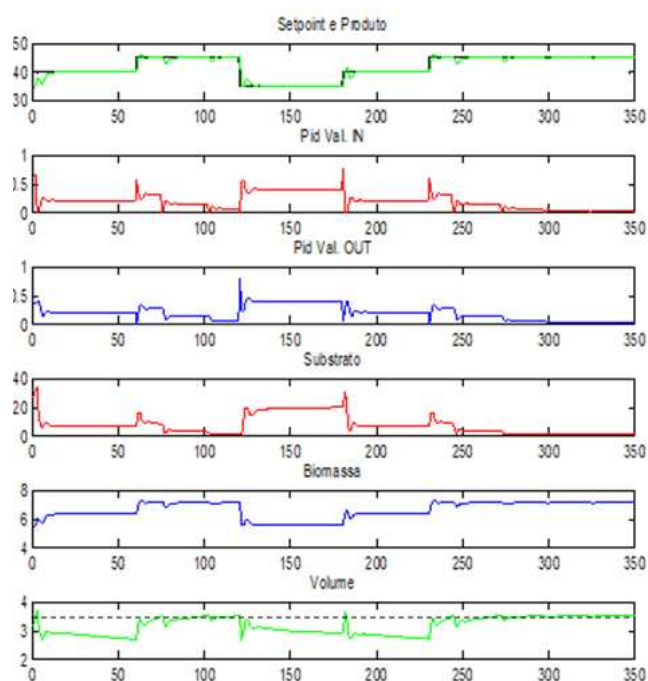


Figura 27 – Controle Regulatório (PID): Experimento 1
Fonte: autoria própria

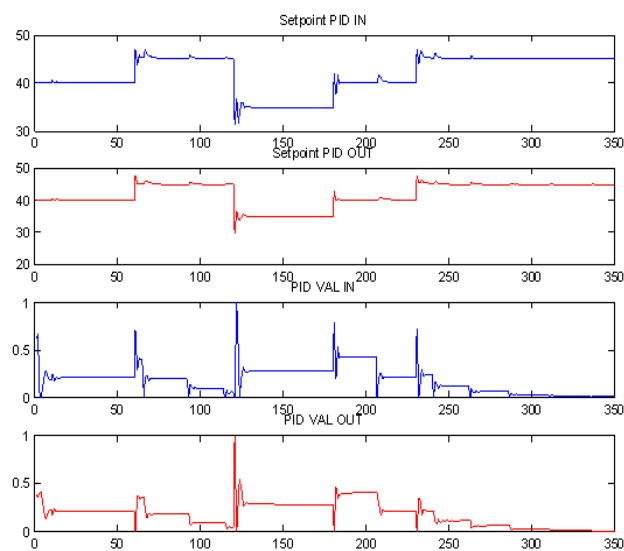
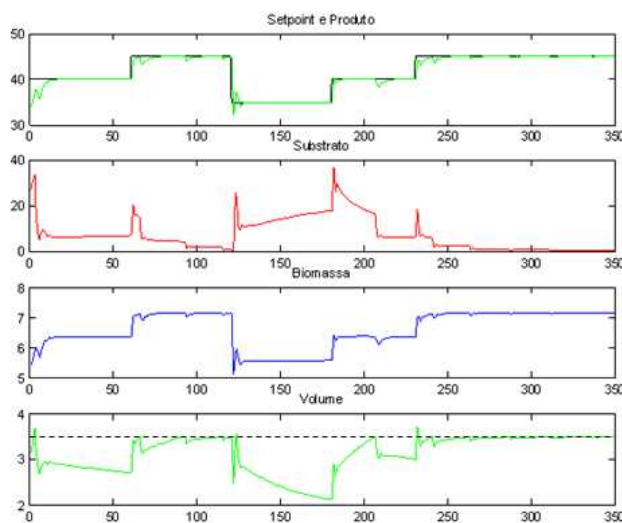


Figura 28 – Controle Supervisório (DCNn+ PID):
Experimento 1 - Sinais de Controle
Fonte: autoria própria



**Figura 29 – Controle Supervisório (DCNn+ PID):
Experimento 1 - Variáveis do Processo
Fonte: autoria própria**

Analisando os resultados mostrados no quadro 7, nota-se que a métrica ITAE indicou um desempenho de aproximadamente 5% menor para o controle supervisorio (DCNn+PID) em relação ao controle regulatório apenas, confirmado pelos valores de variância do erro. No que se refere ao sinal de saída associada à válvula F_{out} , os valores de variância (coluna σ^2_{Fout}) são equivalentes. O controle efetivo do processo é feito sobre a válvula F_{in} que regula a entrada de substrato no reator e que é influenciado fortemente pelas variações de *set-points*. No instante $t = 120h$ há uma transição de *set-point* de 45g/l para 35g/l. Neste caso para o controle regulatório o erro é calculado e as válvulas são abertas o suficiente para corrigir esse erro.

Já o controle supervisorio considera que além do erro há uma variação de substrato em instantes anteriores. Em consequência, o valor de W_I é invertido (relação trivalente dada pela equação 29), diminuindo mais ainda o valor do *set-point* desejado e determinando uma abertura completa da válvula F_{in} . Este fato é responsável pelas diferenças observadas nos valores de variância da válvula F_{in} (coluna σ^2_{Fin}) e da variável de saída, gerando uma variação do sinal de controle na válvula F_{in} 13,5% maior para o controle supervisorio. No entanto o impacto disto na variância de saída do controle supervisorio é de apenas (σ^2_y – produto) 1,8% em relação ao controle

regulatório. As variações positivas de *set-point* são tratadas igualmente por ambas as arquiteturas.

Em resumo, para o modo de operação normal com diferentes variações de *set-point*, o comportamento do controle supervisório é equivalente ao controle regulatório, indicando que quando não há ocorrência de eventos críticos, se a malha de controle regulatório está operando satisfatoriamente a inclusão de um nível supervisório não interfere nesta operação. O supervisor cumpre assim o seu papel, que é equivalente ao do operador humano cujo comportamento determina uma não intervenção no processo quando este opera normalmente.

Durante a simulação da campanha, como explicado anteriormente, uma situação crítica para o processo ocorre a partir do instante $t = 285h$, determinando o fechamento do reator (as válvulas *Fin* e *Fout* são completamente fechadas) e o término da reação (ver figura 29). Para analisar esta situação, um segundo experimento, exatamente igual ao primeiro, é realizada. A diferença é que neste experimento o controle supervisório incorpora o modo de operação crítico, sendo assim implementado pelo DCN da figura 25.

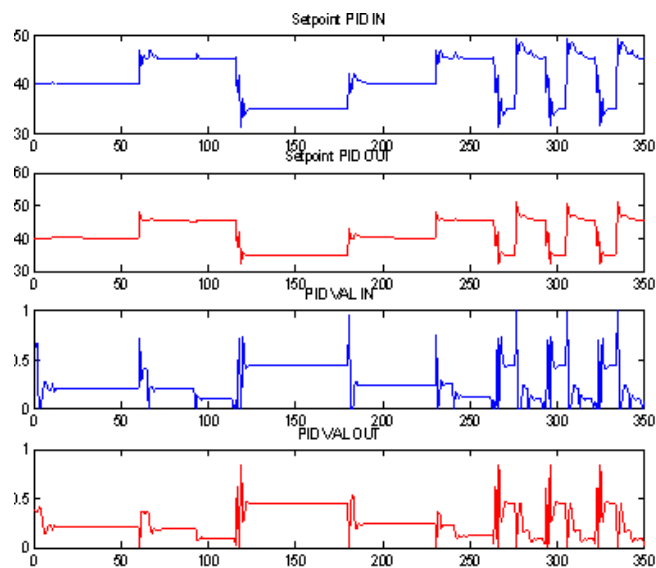
As figuras 32 e 33 mostram os resultados obtidos com controle supervisório realizado pela DCN. A partir de uma avaliação das variáveis do processo, um novo conjunto de *set-points* diferentes do experimento 1, é gerado pela heurística modelada pela base de regras. Por exemplo, um pouco antes do instante em que ocorre a transição de *set-point* de 45g/l para 35g/l ($t = 115h$ até $t = 120h$), o supervisor agora comandado pela DCN da figura 25, gera um *set-point* intermediário de 40g/l, a fim de recuperar a quantidade de substrato que está caindo. Com isso, quando ocorre a transição de *set-point* em $t = 120h$, os erros considerados nos PIDs são menores e os sinais de controle calculados também são menores se comparados com os do experimento 1 (compare os gráficos das figuras 30 e 32 em $t = 120h$). A fim de comprovar, a atuação do supervisor DCN, os índices de desempenho são recalculados até o instante $t = 250h$, para as três arquiteturas de controle: regulatório (PID), supervisório com a DCN da figura 25(DCNn-PID). Estes valores são dados no quadro 8 e mostram que o supervisor DCN-PID apresenta um desempenho ligeiramente superior às outras duas arquiteturas.

Critério	PID	DCNn-PID	DCN-PID
ITAE	1,0192	1,0038	0,9771
σ^2_y (produto)	13,5921	13,6985	13,4003
$\sigma^2_{\text{(erro)}}$	1,3708e-004	1,2696e-004	1,0728e-004
σ^2_{Fin}	0,0154	0,0169	0,0220
σ^2_{Fout}	0,0127	0,0118	0,0169

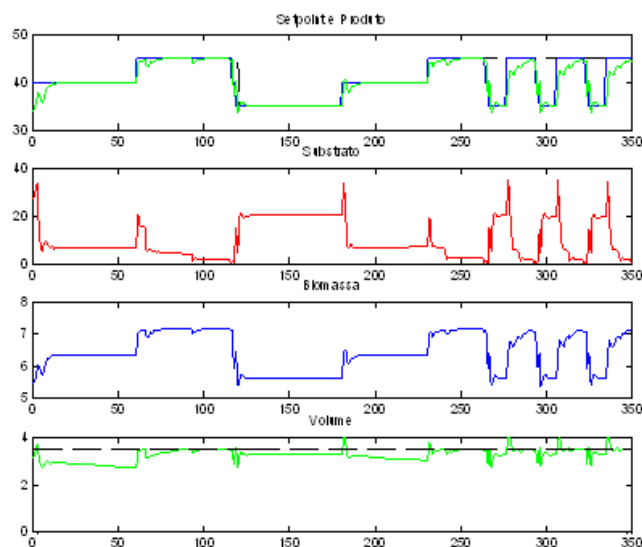
Quadro 8 - Índices de Desempenho Para o Experimento 2

Fonte: Autoria Própria.

Do mesmo modo, analisando ainda as figuras 30 e 31, a situação crítica, que no experimento 1 ocorre a partir de $t = 285h$, é antecipada (a concentração mínima de substrato considerada pela DCN não pode cair abaixo de 0,5 g/l), e o supervisor determina logo no instante $t = 264h$ (ver figura 30) uma diminuição do *set-point* de 45 g/l para 40g/l e logo em seguida para 35g/l. Este *set-point* permanece até o instante $t = 275h$, quando o supervisor tenta recuperar o *set-point* desejado e incrementa o *set-point* aplicado para 40 g/l e em seguida para 45 g/l. Esta situação se repetirá indefinidamente, a menos que o *set-point* da campanha seja expressamente diminuído a fim de retirar a reação de seu estado crítico, caracterizado por concentração de $P = 45g/l$, concentração de substrato $S = 0.5g/l$, concentração de biomassa $C = 7g/l$ e volume $V = 3,5l$ (valores aproximados). Na figura 31 duas linhas de *set-points* são inseridas, a linha preta é o *set-point* de campanha; a linha azul é o novo *set-point* modificado pela ação da DCN.



**Figura 30 – Controle Supervisório (DCN-PID):
Experimento 2 - Sinais Controle
Fonte: autoria própria**



**Figura 31– Controle Supervisório (DCN-PID):
Experimento 2 - Variáveis do Processo
Fonte: autoria própria**

Como conclusão do experimento 2, tem-se que a combinação dos dois FCMs, um para cada modo de operação, normal e crítico, em um único supervisor melhora o comportamento do processo em ambas as situações, validando a arquitetura proposta.

O terceiro experimento analisa o comportamento do supervisor DCN quando ocorre uma falha no processo. Neste caso, simulou-se um fechamento temporário (travamento) da válvula F_{in} durante os instantes $t = 195h$ até $t = 205h$, cortando a entrada de substrato no reator. Novamente, compara-se o desempenho do controle regulatório, mostrado na figura 32, com o desempenho do controle supervísório feito pelo DCN da figura 25 é ilustrado nas figuras 33 e 34. Nestas figuras é mostrado apenas o comportamento das variáveis do processo a partir do instante $t=150h$. Analisando-se os resultados gráficos na figura 32, observa-se que a reação química para logo após o fechamento da válvula F_{in} , com controle regulatório apenas. A válvula F_{out} também é fechada, a concentração de produto cai, mas é recuperada tão logo a válvula seja liberada. No entanto a retirada de produto do reator é mínima, as válvulas estão praticamente fechadas, (instantes $t = 210h$ até $t = 230h$) até que ocorra a mudança de *set-point* em $t = 230h$. A partir deste ponto, o controle regulatório, tenta manter a concentração de produto, com uma retirada pequena (valores de sinal de controle F_{in} e F_{out} pequenos) até que novamente uma situação crítica é atingida ($t = 280h$) e as válvulas são fechadas, parando a reação.

Com o controle supervísório, o fechamento da válvula F_{in} leva a uma diminuição do *set-point* de 40g/l para 35g/l enquanto durar o travamento da válvula, em seguida este *set-point* é restaurado, e a retirada de produto não é prejudicada (figura 32). Quando ocorre a mudança de *set-point* em $t = 230h$ os efeitos do fechamento da válvula F_{in} já não são mais sentidos, e como no experimento 2, o processo antecipa o estado crítico da reação e tenta recuperá-la como acontece no experimento 2.

Como resultado do experimento 3, conclui-se que apesar do supervisor não ter sido especificamente projetado para funcionar durante uma falha, este é capaz de amenizar a influência da falha sobre o desempenho do processo, validando mais uma vez a função da arquitetura proposta. Uma das vantagens do uso da arquitetura proposta está no fato de que novas funções tais como alarme de situação crítica para a reação e detecção de falhas como as que ocorrem nos experimentos mostrados podem ser facilmente incorporadas ao supervisor, através da criação de novas DCNs. Estes novos mapas são conectados entre si, através de conceitos comuns, configurando-se num supervisor completo como sugere a figura 20.

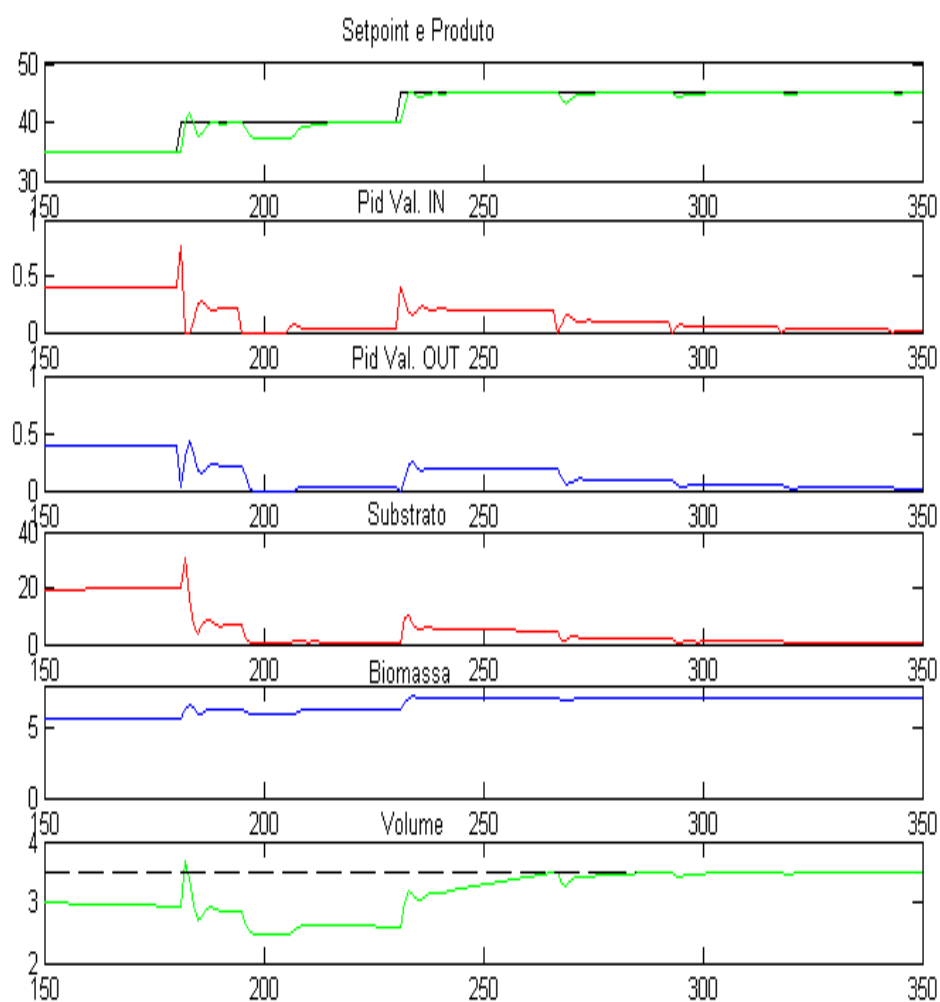


Figura 32 – Controle Regulatório (PID): Experimento 3
Fonte: autoria própria

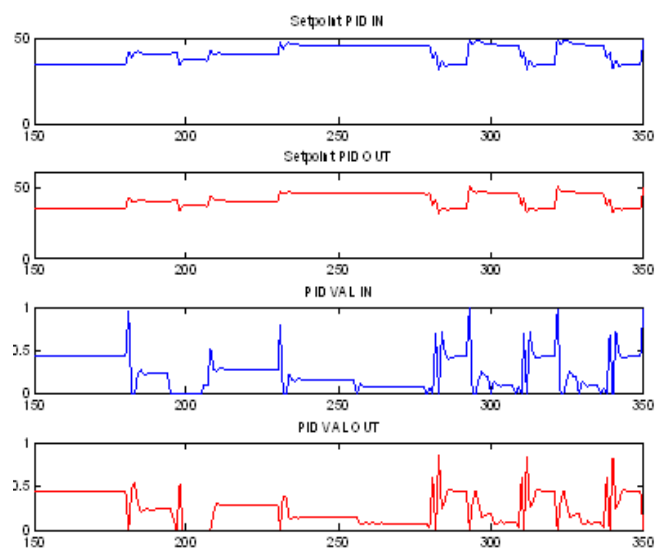


Figura 33 – Controle Supervisório (DCN-PID):
Experimento 3 - Sinais Controle
Fonte: autoria própria

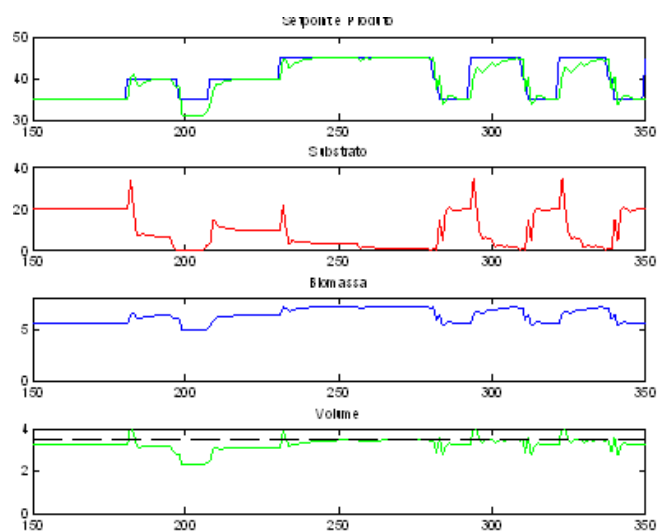


Figura 34 – Controle Supervisório (DCN-PID):
Experimentos 3 - Variáveis do Processo
Fonte: autoria própria

5.4 CONCLUSÃO DO CAPÍTULO

Este capítulo desenvolveu um sistema supervisorio baseado em redes cognitivas dinâmicas, baseado no modelo proposto no capítulo 4, para o controle de um processo de fermentação. O supervisor representa uma abstração do conhecimento especialista sobre o comportamento do processo em operação normal e com restrições. A construção do mapa cognitivo foi obtida através de análise de resultados de simulação, utilizando exclusivamente conhecimento qualitativo, não sendo necessária a utilização de um modelo matemático do processo. O supervisor assim obtido é bastante simples, de fácil implementação e de fácil compreensão, e apresentou um bom desempenho nas situações simuladas.

Apesar de implementar apenas a ação de calcular *set-points* para o controle regulatório em modo de operação normal e crítico, o supervisor proposto pode ser facilmente expandido, para incorporar outras tarefas supervisorias tais que detecção de falhas, escalonamento de ganho, inferência, entre outras. Para isto, basta que sejam construídos novos modelos baseados em mapas do tipo DCN que sejam associados a estas tarefas e conectados entre si, constituindo um supervisor completo como mostrado na figura 20.

Uma das restrições à utilização de modelos baseados em mapas cognitivos, como por exemplo, os FCMs, na representação de sistemas dinâmicos mais frequentes na literatura é que os FCMs apesar de simples não são capazes de modelar os diversos tipos de relação existentes em sistemas dinâmicos, principalmente aqueles relacionados à realimentação e ao comportamento temporal das variáveis. De fato, um mapa cognitivo ou um FCM construído segundo a proposta original de Axerold (1976) e/ou Kosko (1986) somente representa relações causais monotônicas e simétricas. Para contornar este problema, é proposta nesta Tese a utilização de DCNs que incorporam diferentes tipos de conceito (nível, variação, decisão e seleção) e de relações (causal, *Fuzzy*, implicação linguística) capazes de modelar todos os relacionamentos existentes entre as variáveis de um sistema dinâmico. Este novo tipo de mapa cognitivo, denominado rede cognitiva dinâmica, estende a abordagem proposta em Miao et al.

(2001) na medida em que utiliza novas formas de cálculo dinâmico dos arcos de um mapa cognitivo.

Esta nova ferramenta foi inserida em uma arquitetura em que partes do mapa cognitivo são acionadas continuamente e outras apenas a partir da ocorrência de eventos. Com isso foi possível emular as ações de um operador humano para a correta supervisão de uma malha de controle de processos. O sistema resultante “supervisor + processo” pode ser considerado um sistema híbrido em que é possível monitorar e calcular variáveis contínuas, ao mesmo tempo em que determinadas ações são realizadas a partir da ocorrência de eventos discretos.

Desta forma, a DCN apresentada é um sistema computacional inteligente com capacidade de inferência que abstrai e modela o conhecimento humano de forma estruturada por meio de relações de causa-efeito e/ou baseado em regras. Através dela, é possível modelar de forma similar comportamentos dinâmicos contínuos e/ou orientado a eventos. Além disso, é possível refinar a DCN (versão final do modelo cognitivo), conferindo-lhe capacidade de generalização a partir de um conjunto de dados, através do uso técnicas de aprendizado baseado em metaheurísticas tais que algoritmos evolucionários, *simulated annealing*, enxames de partículas entre outras, ou ainda técnicas de aprendizado por reforço.

6 DCN EM NAVEGAÇÃO ROBÓTICA

6.1 INTRODUÇÃO

Este capítulo desenvolve um sistema de navegação autônomo a partir do modelo baseado em DCN proposto no capítulo 4, sendo o alvo da aplicação à robótica móvel.

O objetivo básico deste capítulo é desenvolver novos mecanismos de aprendizagem junto a um sistema de navegação autônomo inteligente baseado em conhecimento heurístico, inspirado na capacidade de tomada de decisões em ambientes desconhecidos. Deste modo, para executar tarefas de navegação em ambientes dinâmicos, os robôs necessitam reagir rapidamente a mudanças, tornando-se adequado o uso de uma arquitetura reativa no sistema de controle. No entanto, o robô também deve realizar tarefas de acordo com um planejamento prévio e tomar decisões a fim de realizar o planejado, mesmo em face de situações imprevistas.

Estas habilidades serão introduzidas no navegador através da representação dos diversos conceitos e relações de uma rede cognitiva dinâmica.

6.1.1 Navegação Robótica

A aplicação de robôs em diversos tipos de tarefas vem aumentando, fruto de pesquisas, estudos e avanços tecnológicos. A pesquisa na área de navegação autônoma se encontra em um estágio de ascensão e tem se mostrado um importante suporte no auxílio das investigações e avaliações de sistemas inteligentes. A complexidade envolvida nas tarefas de geração de trajetórias eficientes é reconhecidamente alta (FRACASSO; COSTA, 2005). Em muitos casos exige-se que o sistema autônomo seja capaz de aprender uma estratégia de navegação através da interação com o ambiente, como por exemplo, no trabalho de (CALVO; ROMERO, 2006).

Existe um interesse crescente no desenvolvimento de robôs autônomos e/ou veículos, principalmente por causa da grande diversidade de tarefas que podem ser realizadas por eles, especialmente aquelas que põem em perigo a saúde humana e / ou o

ambiente, (ASAMI, 1994; SCHRAFF, 1994). Como exemplo, o trabalho de Mandow e colaboradores (1996) descreve um robô móvel autônomo para utilização na agricultura, a fim de substituir o trabalho humano, em atividades inóspitas como a pulverização de inseticidas.

Na robótica móvel, basicamente dois paradigmas guiam o desenvolvimento dos diversos tipos de arquitetura: o reativo e o deliberativo (MURPHY, 2000). O paradigma reativo busca emular a reação imediata dos animais aos estímulos do ambiente. As arquiteturas reativas são geralmente utilizadas como uma camada de nível inferior na navegação de robôs móveis com a vantagem de responderem em tempo real, por mapearem diretamente a leitura dos sensores em ações do robô (ARKIN, 1999). Já o paradigma deliberativo intercala o processo da tomada de decisão, desde a percepção até a ação, com uma etapa de planejamento. Deste modo, a arquitetura deliberativa geralmente demanda grande tempo de processamento, dificultando a atuação do robô em tempo real. Soluções híbridas são muito exploradas atualmente, definindo arquiteturas híbridas (SELVATICI, 2005; GUDWIN, 2002), que combinam os paradigmas reativos e deliberativos.

De um modo geral, o problema de controle de robôs móveis é composto por dois subproblemas principais:

1. Navegação que corresponde à determinação da posição do robô móvel e sua orientação em um determinado momento.
2. Guiagem que se refere ao controle do caminho a ser seguida pelo robô móvel.

Outra área de interesse em robótica móvel é o aprendizado em robôs, e consiste essencialmente em fazer com o que o robô execute tarefas sem a necessidade de programá-los explicitamente. Deste modo, a programação de robôs é uma tarefa desafiadora, por muitas razões, pois não basta apenas conhecer o funcionamento dos sensores e motores, além do modelo cinemático do robô, mas também deverá se conhecer um modelo do ambiente em que robô será inserido, o que nem sempre é possível. Ainda que o ambiente seja conhecido, este pode sofrer mudança de condições devido ao tempo e/ou intempéries. Por exemplo, a mudança da aderência do piso resultante de envelhecimento ou de chuva pode mudar a resposta dinâmica do robô.

Outra dificuldade refere-se à decomposição do problema de navegação em uma sucessão de tarefas de baixo nível, tais como: andar em linha reta, virar à esquerda, entre outras similares.

Vários trabalhos na literatura desenvolvem aplicações de aprendizado por reforço na área de robótica e navegação. Nos trabalhos de Gaskett; Fletcher e Zelinsky (2000); Smart e Kaelbling (2001); Zhu e Levinson (2001) é feito o uso de informações provenientes de sensores de visão computacional, incorporados a algoritmos de aprendizado por reforço para controlar a navegação de robôs móveis. Aranibar e Alsina (2004) aplicam o algoritmo *Q-learning* no planejamento de rotas de robôs autônomos. Bhanu e colaboradores (2001) apresentam uma implementação física do ambiente de labirintos, descritos no trabalho de Sutton e Barto (1998). Yen e Hickey (2002) propõem uma arquitetura de aprendizagem de reforço, que incorpora um mecanismo de esquecimento, permitindo assim exploração, e um aprendizado baseado em características do ambiente, para mapeamento dos estados.

Faria e Romero (2004) propõem uma modificação no algoritmo de aprendizagem por reforço *R-learning*, através da inclusão de informações *fuzzy*, provenientes dos sensores, no cálculo de atualização dos valores-R. O uso de redes neurais e RL pode ser encontrado em Macek, Petrovic e Peric (2002) para tratar o problema de representação estrutural de estados contínuos, em aplicação de desvio de obstáculos em robôs móveis.

Nesta Tese, um algoritmo *Q-learning* ajustado através de regras do tipo antecedente consequente será empregado para ajustar as relações causais de uma DCN. Esse algoritmo de aprendizagem por reforço deverá sintonizar as relações causa e efeito das ações de baixo nível (DCN inicial) de acordo com possíveis mudanças no ambiente.

6.1.2 Proposta de DCN Para Navegação Robótica

O problema de navegação autônoma de robôs consiste em desenvolver mecanismos de tomada de decisão para um ou mais robôs móveis em um ambiente arbitrário no qual devem se realizar determinadas tarefas de forma autônoma. Em outras palavras, sistemas de navegação autônoma devem ser capazes de guiar robôs sem

nenhuma externa e definir ações para estes, como atingir alvos, desviar de obstáculos, explorar ambientes, entre outras (Calvo, 2007).

A proposta DCN é executar funcionalidades, em especial: desvio de obstáculos e coleta de alvos em ambientes simulados e reais (somente desvio de obstáculo com robôs reais). Essas funcionalidades objetivam avaliar a capacidade de tomadas de decisões e adaptação dinâmica da rede cognitiva. A complexidade dos ambientes é aumentada com acréscimo de mais obstáculos, em especial com o acréscimo de obstáculos móveis e obstáculos que surgem de modo inesperado no cenário (obstáculos surpresa), outra dificuldade que aumenta a complexidade de tomadas de decisões será o gerenciamento das ações do sistema de navegação.

Originalmente em navegação autônoma, são modeladas as ações de uma mudança de direção e velocidade para desvio de obstáculos (CALVO; ROMERO, 2006). Esses obstáculos podem ser estáticos ou dinâmicos, esses últimos caracterizam as mudanças repentinas no ambiente. Assim, o sistema de navegação autônoma proposto deve ser capaz de tomar decisões dinâmicas, ou seja, mover-se através do ambiente e às vezes até necessário alterar a trajetória, como resultado de um evento.

Ações como a decisão de virar à esquerda ou à direita quando os sensores acusam os obstáculos e acelerar quando há um caminho livre são ações de controle sempre válidas em todas as circunstâncias. Desta forma, este tipo de ação é modelado como uma relação de causalidade de modo semelhante ao de um FCM clássico.

No entanto, existem situações específicas, tais fatos como: a necessidade de manter uma tendência de movimento, principalmente nas curvas, quando o veículo está virando à esquerda e os sensores acusam um novo obstáculo na mesma direção. Devido à inércia e restrições físicas, tais como ruído nos sensores, o robô móvel não pode mudar de direção bruscamente. Este tipo de manobra deve ser cuidadosamente executado. Neste contexto, algumas situações específicas também devem ser modeladas no mapa por relações de causalidade e conceitos, mas elas são válidas apenas como resultado de uma tomada de decisão, causada por eventos em curso. Para implementar essa estratégia, a metodologia proposta no capítulo 4, baseada nas redes cognitivas dinâmicas DCN incluindo novos tipos de relações e conceitos será utilizada para construir o navegador.

A DCN proposta para navegação de robôs é estruturada em duas camadas. A camada mais baixa está relacionada com ações diretas, vire à esquerda, vire à direita e acelerar. Esta camada é modelada como um FCM clássico por meio apenas de relações causais, de entrada e saída. O modelo cognitivo na camada de baixo nível tem um comportamento reativo com tomadas de decisões com base em medições de sensores. Na camada superior, a ocorrência de mudanças dinâmicas (eventos) é representada e as reações do robô e as decisões são modeladas por meio de regras do tipo “declaração condicional”. Esta camada emprega uma estratégia deliberativa, onde as metas planejadas estão embutidas nas regras.

A Arquitetura de controle inteligente para o sistema de navegação é mostrada na figura 35. A interface de entrada tem a funcionalidade ler as medições dos sensores, que são inversamente proporcionais à distância dos obstáculos. A DCN é o motor de inferência que reúne os dados de entrada e o conhecimento (valores de pesos W) para tomar uma decisão sobre o movimento do robô / veículo. Uma sintonia dinâmica das relações causais do nível mais baixo é feita através de algoritmos de aprendizagem por reforço, correspondendo a etapa 5 do algoritmo do quadro 2. A base de regra por outro lado representa o conhecimento heurístico para tomar uma decisão na presença de eventos em conflito, ao mesmo tempo em que considera o planejamento original estabelecido para o sistema de navegação.

Na estrutura da DCN apresentada na figura 35 é possível distinguir exatamente as duas camadas. A camada 1 (reativa) modelada pelas relações causais e a camada 2 (deliberativa) modelada por uma base de regras. Essas duas camadas coexistem e podem tomar decisões ao mesmo tempo. Entretanto, com o aumento da complexidade do problema, pode ser necessária a utilização de precedência (hierarquia) entre as camadas para a correta execução das ações do controlador.



Figura 35 - DCN (Arquitetura do Controlador Para um Sistema de Navegação Robótica)
Fonte: autoria própria

A seção a seguir apresenta cada uma das etapas de desenvolvimento da DCN, com destaque para o desenvolvimento do algoritmo de aprendizado por reforço existente na camada reativa. O bloco memória representa nesse caso, a memória temporal, ou seja, a ação anterior executada pelo controlador.

6.1.3 Construção da DCN

O desenvolvimento da DCN parte da observação do problema (capítulo 1), onde são identificados inicialmente 3 características relacionadas com a descrição do ambiente (presença de obstáculos à esquerda, a direita ou em frente da trajetória seguida pelo robô) e 3 saídas descrevendo os movimentos do robô: virar à esquerda, virar à direita e seguir em frente. As três entradas assumem valores medidos pelos três sensores localizados à esquerda, direita e frente do móvel. Estes conceitos são conectados por arcos que representam as ações de aceleração (positiva) e de frenagem (negativa). Três decisões são originalmente modeladas, se os sensores acusam um obstáculo à esquerda, o veículo deve virar para o lado direito, e por outro lado se os sensores da direita acusam um obstáculo no lado direito, o veículo vira para lado esquerdo. A decisão de

mudar de direção implica em uma desaceleração suave do robô. A terceira decisão está relacionada a um ambiente livre de obstáculos. Neste caso, a unidade móvel segue uma linha reta acelerando suavemente.

A construção da DCN pode ser dividida em dois passos: no primeiro passo: a DCN inicial modela apenas relações do tipo causal, as quais representam a camada reativa do robô. As funções de baixo nível são então modeladas pela DCN da seguinte forma: os conceitos de entrada são SL, SR e SF que correspondem respectivamente às leituras dos sensores laterais (esquerdo e direito) e do sensor frontal e os conceitos de saída são **Outleft**, **OutRigth** e **OutFront**, que representam os movimentos a esquerda, a direita e em frente. As relações causais entre estes conceitos expressas em termos de efeito e intensidade estão descritas no quadro 9.

O quadro 9 descreve e detalha as relações causais do modelo cognitivo, essas relações são obtidas pela observação da dinâmica do robô móvel (ajuste *on-line*) no simulador em diferentes cenários. A figura 36 ilustra o mapa cognitivo representativo destas relações. Os valores dos conceitos são as leituras dos sensores correspondentes. Como um número *Fuzzy*, esses valores são normalizados no intervalo $[0, 1]$. As relações entre esses conceitos são modelados pelos pesos w_1 até w_9 . Vale à pena observar na figura 36, que os conceitos O.L. (-1) e O.R. (-1) são os valores dos conceitos no estado anterior, portanto são conceitos do tipo memória conforme definidos no capítulo 4. Esta representação é equivalente a inserir valores negativos (-1) nas posições correspondentes à diagonal da matriz de pesos W . Observa-se que nesta aplicação o conceito de memória tem efeito contínuo em toda a trajetória do robô móvel, ou seja, nesse caso é um conceito de memória temporal. Entretanto, é possível ocorrer à inclusão desse conceito para uma tomada de decisão na camada deliberativa da DCN devido à ocorrência de um evento.

Relação Causal	Descrição	Efeito	Intensidade
W_1	Conecta Sensor Right (S.R.) na saída Out Left (O.L.)	Positivo	Forte
W_2	Conecta Sensor Front (S.F.) na saída Out Left (O.L.)	Positivo	Médio
W_3	Conecta Sensor Front (S.F.) na saída Out Front (O.F.)	Positivo	Forte
W_4	Conecta Sensor Front (S.F.) na saída Out Right (O.R.)	Positivo	Médio
W_5	Conecta Sensor Left (S.L) to Out Right (O.R.)	Positivo	Forte
W_6	Conecta conceito saída Out Left (O.L) na saída Out Right (O.R.)	Negativo	Muito fraco
W_7	Conecta conceito saída Out Right (O.R.) na saída Out Left (O.L.)	Negativo	Muito fraco
W_8	Conecta Sensor Right (S.R.) na saída Out Right (O.R.)	Negativo	Muito fraco
W_9	Conecta Sensor Right (S.R.) na saída Out Right (O.R.)	Negativo	Muito Fraco

Quadro 9 – Descrição das relações da DCN navegação.

Fonte: Autoria própria.

O uso deste tipo de conceito permite o robô lembrar-se das medidas tomadas anteriormente para virar à esquerda ou à direita, suavizando um efeito de “zig-zag” no seu movimento. Como resultado, o robô móvel consegue manter uma tendência de movimento.

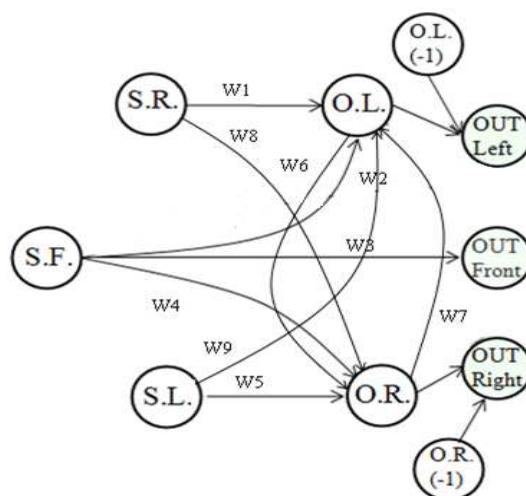


Figura 36 - DCN Inicial
Fonte: autoria própria

O sistema reativo modelado pela DCN na figura 36 tem sempre o mesmo comportamento independente de variações no ambiente e/ou no robô (desgaste de engrenagens e variação de aderências das rodas). Para contornar esses problemas de variações do robô e/ou do ambiente (como por exemplo, a presença de um obstáculo inesperado durante a trajetória), um ajuste dinâmico dos valores das relações causais (w) pode ser realizado através de um algoritmo de aprendizado por reforço. Esta etapa não corresponde a um treinamento propriamente dito, mas antes a um refinamento das relações causais dinâmicas dentro de uma faixa previamente estabelecida pelos especialistas e de acordo com regras heurísticas. A introdução do algoritmo RL na DCN da figura 36 será apresentada na seção a seguir, após a descrição da camada deliberativa.

A fim de modelar a tomada de decisão em face de eventos não modelados (imprevistos), mas considerando um planejamento a priori, são introduzidos 3 novos conceitos para a DCN, associados a uma "intensidade" do movimento (aceleração ou frenagem) em cada direção. Estes conceitos são do tipo fator e ponderam as decisões em cada direção, fator da esquerda, fator da direita e fator de frente. A DCN com a camada deliberativa incorporada é mostrada na figura 37. Os conceitos do tipo fator têm seus valores alterados de acordo com a condição atual do movimento do veículo e a ocorrência de eventos. Estes eventos são modelados pelos pesos " ws " na figura 37, que

são obtidos pela aplicação de regras (implicações linguísticas) do tipo SE-ENTÃO. Estas regras representam algumas decisões, como se o robô está virando à direita porque o sensor da esquerda detectou um obstáculo e de repente o sensor direito também detecta um obstáculo, então o fator de saída à direita é pequeno (ws_3). O valor padrão de um conceito fator é 1. Se nenhuma regra é disparada para uma determinada relação, então o valor do peso associado ao conceito fator é nulo. Finalmente as saídas da DCN são: o produto entre os conceitos fator e a saída da DCN de baixo nível (**Outleft**, **OutRight** e **OutFront**).

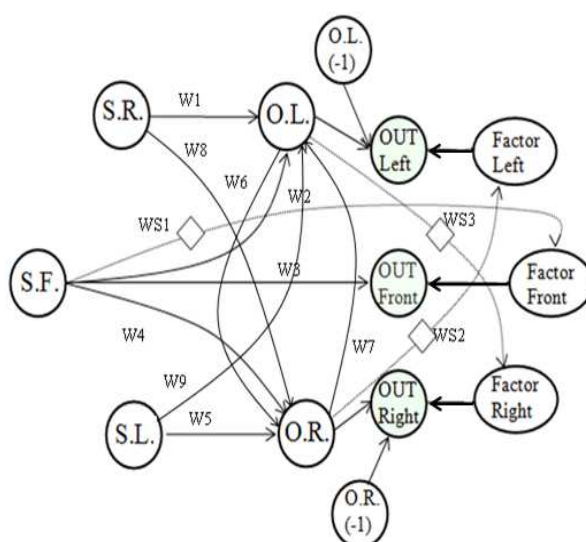


Figura 37 - Proposta da DCN
Fonte: autoria própria

O sistema de navegação autônomo modelado pela DCN da figura 37 confere ao robô / veículo o seguinte comportamento:

- Os robôs são autônomos e movem-se em ambiente desconhecidos desde o ponto de origem até um ponto final dentro de uma região estimada.
- Caso um obstáculo seja detectado por sensores na frente, à esquerda e / ou à direita do móvel, o robô deve tomar uma decisão sobre o novo rumo a seguir.
- A posição de navegação padrão é em linha reta com velocidade

constante, ou seja, os movimentos laterais são usados apenas como resultado de detecção de obstáculos.

- Quando o robô móvel está em movimento e os sensores não identificam qualquer obstáculo, este acelera suavemente e em seguida permanece em uma velocidade constante.
- A tendência de movimento corresponde a uma média entre os valores de movimento em curso e os valores no instante imediatamente anterior, o que impede qualquer alteração acentuada da direção de navegação do robô.
- Quando o robô móvel está virando à esquerda e o sensor oposto também detecta um obstáculo, a tendência de movimento é mantida, porém o robô começa a ponderar suavemente a presença de obstáculos e toma decisões considerando os movimentos anteriores.

6.1.4 O Modelo DCN Adaptado Por Reforço

A aprendizagem por reforço para o ajuste dinâmico dos pesos das relações causais da DCN é uma proposta desta Tese. Trabalhos da literatura apontam a necessidade de refinamento e ajuste (*off-line*) na opinião dos especialistas empregada na construção de um FCM (GLYKAS, 2010) como discutido no capítulo 2 e ilustrado no capítulo 3. No entanto, devido à natureza dinâmica da navegação robótica, esse ajuste deve ser feito de maneira *on-line*, o que descarta a utilização de algoritmos metaheurísticos tais como AG e PSO que são pesados computacionalmente. Em face disso, o uso de algoritmos de aprendizagem por reforço aparece como alternativa para conferir adaptação ao modelo.

Esses algoritmos permitem ajustes que possibilitam uma melhor adaptação através de um refinamento do controlador de navegação para eventuais mudanças no ambiente e/ou na dinâmica do robô (por exemplo, desgaste de peças). De um modo geral, o aprendizado por reforço possibilita o aprendizado de um agente diretamente a partir da sua interação com o ambiente. O modelo padrão de um sistema de aprendizado por

reforço já foi apresentado no capítulo 2 e será retomado aqui. Este modelo foi ilustrado na figura 7 a qual é copiada na figura 38 para facilitar a leitura. A partir desta figura, a cada instante de tempo t (dita iteração), o robô (agente), através dos seus sensores, percebe que está em um estado s_t (referenciado como o seu estado atual). De acordo com a política que está seguindo, ele escolhe uma ação a ser executada pelos seus atuadores. Esta ação provoca uma transição de estados, isto é, o agente sai do estado s_t e passa para um estado s_{t+1} . Em função desta mudança, o ambiente responde com uma medida de reforço que pode ser uma recompensa, se a ação tomada foi boa, ou uma punição, caso a ação tomada tenha sido ruim. Esta medida de reforço, ao longo do tempo, provocará uma correção da política, definindo um processo de aprendizado dinâmico e, em alguns casos, iterativo.

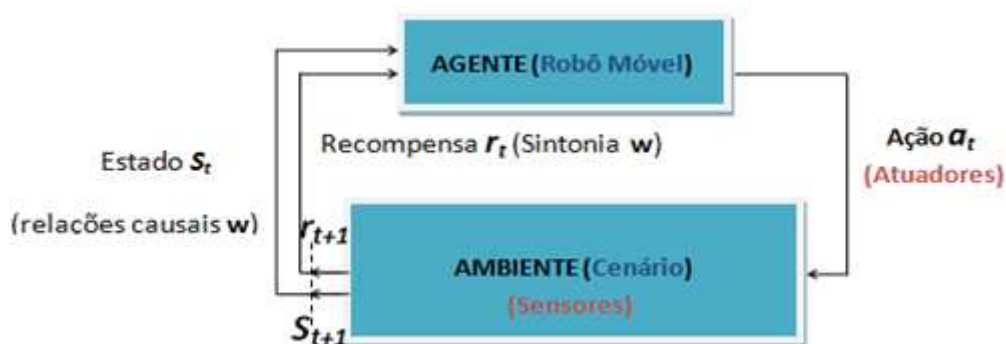


Figura 38 - Modelo Padrão de Aprendizagem Por Reforço
 Fonte: adaptado de Sutton e Barto (1998).

Para a aplicação em navegação robótica, uma política é empregada no mapeamento de estados em ações, isto é, determina-se qual ação a_t deve ser executada quando o robô móvel (agente) estiver no estado s_t , definindo assim, o comportamento do agente ao longo do tempo. Esse algoritmo usa a experiência de cada transição de estado para atualizar um elemento de uma tabela que armazena os passos de evolução do algoritmo. Esta tabela, denotada por Q , tem uma entrada, $Q(s, a)$, para cada par de estados S e de ação a . Após a transição $s_t; s_{t+1}$ (figura 38), esse algoritmo executa a atualização e recebe uma recompensa r_{t+1} , o gerenciamento da aplicação das penalidades será feita por uma política (base de regras, na proposta da DCN) e de alguns parâmetros, como por exemplo, α é um parâmetro que determina o tamanho do

passo positivo, como mostra a equação (32). Em condições adequadas, esse algoritmo converge quando se estabelece uma política que aplica bônus e punições de acordo com as mudanças do ambiente (SUTTON; BARTO, 1998).

Este tipo de aprendizado difere do aprendizado supervisionado, como aquele encontrado nas redes neurais e utilizado basicamente no perceptron de Rosenblatt (1963, apud HAYKIN, 2000) e na Adaline de Widrow e Hoff (1960, apud HAYKIN, 2000). Os métodos com aprendizado supervisionado são baseados na correção de erros, e para isso requerem um conjunto de treinamento constituído de pares de vetores de entrada e saída, enquanto no aprendizado por reforço, o conhecimento adquirido na interação com o ambiente determina uma obtenção ou perda de recompensa, levando o agente a evoluir para um estado ótimo, ou um estado melhor adaptado a um determinado contexto. O objetivo do sistema de aprendizado por reforço é encontrar uma política ou estratégia que maximiza o retorno \mathbf{R} fornecido pelo ambiente.

Algoritmo de aprendizagem por reforço empregado no desenvolvimento do sistema de navegação robótica com DCN é semelhante ao *Q-learning*, em que as ações de recompensa e punição implementadas no algoritmo, são obtidas a partir de regras heurísticas que descrevem o ambiente. O objetivo é o refinamento das relações causais a fim de proporcionar uma melhor adaptação do modelo proposto para possíveis mudanças na dinâmica do robô e/ou ambiente, como por exemplo, um possível desgaste de peças e/ou mudança de aderência no piso para o caso de mudanças na dinâmica do robô. Ou o aparecimento de um obstáculo surpresa e/ou uma situação drástica provocada por um obstáculo dinâmico para o caso de mudanças no ambiente (situações que serão aplicadas em cenários de experimentos que serão realizados pelo simulador). Deste modo, não existe um período de treinamento, e, sim uma auto-sintonia das relações causais dentro de uma faixa previamente estabelecida pelos especialistas, de acordo com regras heurísticas.

O conhecimento modelado pelas regras é baseado nas respostas dos sensores, de acordo com a proposta do sistema de navegação autônomo. Como por exemplo, se a entrada do sensor direito estiver muito intensa então o peso da relação causal que conecta este sensor de entrada aos conceitos de saída deve ser aumentado. O mesmo

acontece com os sensores de entrada a esquerda e frontal. Assim, é possível estabelecer ajustes dinâmicos nas relações causais de acordo com as medições dos sensores.

No entanto, a adaptação da rede cognitiva deve ser cuidadosamente elaborada para evitar a ocorrência de mudanças muito acentuadas nos valores dos pesos, para não desconfigurar a estrutura e/ou regras do modelo inicialmente proposto. A inclusão de intervalos com valores de mínimo e máximo para as relações causais foi a solução encontrada.

Esses limites objetivam a sintonia (ajuste fino) para um ou no máximo dois intervalos discretizados na construção do modelo. Se essa análise fosse feita por termos linguísticos, essa sintonia do modelo A.R. busca um melhor ponto de operação dentro do intervalo discretizado como forte ou muito forte, por exemplo. De maneira análoga, comparado com lógica *Fuzzy*, essa sintonia seria semelhante a um ajuste fino nos vértices de um triângulo de uma função de pertinência, por exemplo.

Os valores máximos e mínimos são impostos pelos especialistas, como feito no capítulo 3. Por exemplo, o valor da relação causal dos sensores laterais de entrada deve variar no intervalo de 0.6 a 1, valores empíricos, porém determinados de acordo com a proposta de sintonia (ajuste fino) apresentado. Os estados dos valores dos pesos das relações causais desejados são adaptados de acordo com as regras heurísticas listadas a seguir:

- SE a intensidade do sensor frontal (SF) é maior do que “*threshold médio*” ENTÃO W_{lim} aplicado no ajuste de w_3 é o valor máximo W_{Fmax} .
- SE a intensidade do sensor frontal (SF) menor do que “*threshold mínimo*” ENTÃO W_{lim} aplicado no ajuste de w_3 é o valor mínimo W_{Fmin} .
- SE a intensidade do sensor direito é maior do que “*threshold médio*” ENTÃO W_{lim} aplicado no ajuste de w_1 é o valor máximo valor W_{Rmax} .
- SE a intensidade do sensor direito é menor do que “*threshold mínimo*” ENTÃO W_{lim} aplicado no ajuste de w_1 é o valor mínimo W_{Rmin} .
- SE a intensidade do sensor esquerdo é maior do que “*threshold médio*”

ENTÃO W_{lim} aplicado no ajuste de w_5 é o valor máximo valor **WL_{max}** .

- **SE** a intensidade do sensor esquerdo é menor do que “*threshold mínimo*” **ENTÃO W_{lim}** aplicado no ajuste de w_5 é o valor mínimo **WL_{min}** .

As relações causais citadas nas regras são respectivamente as regras W_1 , W_3 e W_5 que são os pesos das relações causais responsáveis pelas tomadas de decisão de baixo nível, virar à direita, à esquerda e acelerar da DCN.

As regras listadas determinam a política de mudança de estado através de recompensa e punição, termo r da equação (32) que descreve o algoritmo *Q-learning*. A aplicação da equação (32) para cálculo dos pesos leva a:

$$w_i(k) \leftarrow w_i(k-1) + \alpha \times [r + \gamma \times W_{lim} - w_i(k-1)] \quad (32)$$

Onde α é o fator de aprendizagem, γ é o fator de desconto, r é o valor da recompensa ou punição e W_{lim} é o máximo valor do estado desejado para o peso da relação causal, neste caso determinado de acordo com as regras listadas. Por exemplo, se a regra 2 é disparada para a sintonia de W_3 , o valor máximo utilizado para W_{lim} será W_{fmin} , que corresponde a uma punição. A equação (32) aplica diretamente a experiência da interação do agente com o ambiente, e tem como característica a possibilidade de mudar a política em tempo real. Entretanto, os valores de limites de bônus (W_{fmax}) e punições (W_{fmin}) são delimitados dentro dos intervalos definidos pelos especialistas.

6.1.5 Expansão da DCN

A fim de ilustrar a facilidade com que se pode adicionar novas tarefas no sistema de navegação autônoma baseado em DCN e assim expandir as potencialidades do modelo, será acrescentado ao sistema da figura 39 a funcionalidade de encontrar alvos durante a exploração do ambiente. Com isso, o sistema apresentará capacidade de

desviar de obstáculos e encontrar alvos quando percorre uma trajetória em direção a um ponto final pré-estabelecido. Nesse desenvolvimento, os alvos têm suas posições previamente conhecidas no cenário. Entretanto, existem outras maneiras para a localização dos alvos, que poderia ser obtida através de fontes de luz captada por sensores óticos, por exemplo.

Para execução dessa função foram incluídos dois conceitos e duas relações de seleção no modelo da figura 37, o que gera a DCN da figura 39. Os conceitos **DX L** e **DX R** são as distâncias da posição da lateral esquerda e direita do robô móvel, respectivamente em relação às coordenadas dos alvos. Essas distâncias são normalizadas para que os valores assumidos estejam no intervalo de [0 a 1]. As relações de seleção **WS₄** e **WS₅** atuam de forma binária (ligado e desligado). Ou seja, caso o robô móvel esteja navegando a uma distância menor do que 10 e não tenha nenhuma leitura nos sensores o peso das conexões é forte, numericamente igual a 0,8, caso contrário zero.

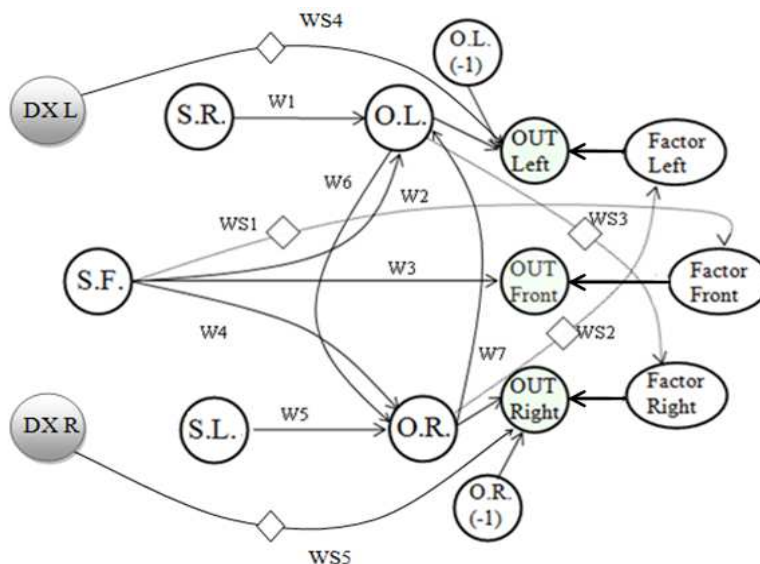


Figura 39 - Proposta da DCN
Fonte: autoria própria

A inclusão da tarefa de recolher alvos pode entrar em conflito com os demais objetivos do robô, em especial com a tarefa de andar em linha reta para o ponto final da trajetória. A fim de resolver os conflitos, uma hierarquia de operação semelhante àquela

da arquitetura de Subsunção de Brooks será adotada. Com isso, as ações de baixo nível serão organizadas em níveis hierárquicos como mostrado na figura 40 e quando uma tarefa de mais alto nível está em operação, às ações das camadas de níveis inferiores serão desabilitadas total ou parcialmente. As prioridades adotadas foram:

- Prioridade 1: Andar em linha reta quando não há nenhum obstáculo e/ou nenhum alvo.
- Prioridade 2: Navegar em direção ao alvo próximo da trajetória do móvel a uma distância menor do que dez centímetros.
- Prioridade 3: Desviar de obstáculos que possam aparecer nas leituras dos sensores.



Figura 40. Hierarquia do Navegador baseado em DCN
Fonte: autoria própria

A arquitetura inteligente final para o sistema de navegação é a mesma mostrada na figura 35. A interface de entrada lê os valores medidos pelos sensores, os quais são inversamente proporcionais à distância dos obstáculos. A DCN representa a máquina de inferência que deve tomar decisões sobre a movimentação do robô baseado nos valores de pesos w e ws . Os valores de iniciais dos pesos (w e ws) foram heurísticamente ajustados (etapa 6 do algoritmo do quadro 2) a partir da observação do comportamento do robô durante várias simulações. A base de regras representa o conhecimento heurístico para a tomada de decisão em presença de eventos e tarefas conflitantes, mas considerando os objetivos planejados. Esta base de regras representa a estratégia deliberativa. O bloco de relações causais representa o conhecimento estruturado sobre

navegação e corresponde ao comportamento reativo. Assim, cada bloco representa uma fonte de conhecimento/informação que pode operar simultaneamente, mas obedecendo as prioridades de tarefas. Por fim, de acordo com o resultado da inferência que considera as decisões de ambos os blocos, as ações de controle são enviadas aos atuadores do robô por meio da interface de saída.

Esta arquitetura será validada a seguir em alguns experimentos simulados e em um robô didático.

6.2 CONTROLADOR BASEADO EM LÓGICA FUZZY

Um sistema de navegação autônoma utilizando lógica *fuzzy* foi implementado e simulado para fornecer um padrão de comparação para o navegador baseado em DCN. Esse navegador foi construído na medida do possível com o mesmo conhecimento utilizado para construir a DCN. Deste modo, é possível comparar e avaliar não só o desempenho de ambos os sistemas, como também as facilidades e dificuldades que cada técnica apresenta para aquisição, formalização e utilização do conhecimento dos especialistas na construção do modelo.

O sistema *fuzzy* desenvolvido é similar ao do trabalho de Harisha e colaboradores (2008), que utiliza um conjunto de regras com 8 variáveis de entrada como o ângulo de posição e distância do obstáculo para calcular a velocidade desejada móvel. O sistema *fuzzy* é implementado é do tipo Mamdani com 3 entradas, 3 saídas e emprega 23 regras para descrever a heurística de navegação. As entradas são os sensores à direita, à esquerda e frontal, as saídas são as ações virar à direita, à esquerda e acelerar. Estas regras são as mesmas utilizadas no navegador DCN. Alguns exemplos dessas regras são:

1. **Se** o sensor direito é forte **então** vire à esquerda forte.
2. **Se** o sensor direito é fraco **então** vire à esquerda fraco.
3. **Se** o sensor direito e frontal são muito forte, **então** acelerar pouco e virar à esquerda muito forte.

4. **Se** o sensor direito, esquerdo, frontal são fraco **então** acelerar muito.
5. **Se** o sensor direito está muito próximo e o sensor frontal distante **então** vire muito a esquerda.

A figura 41 mostra a superfície *fuzzy* das saídas esquerda e direita do sistema navegação.

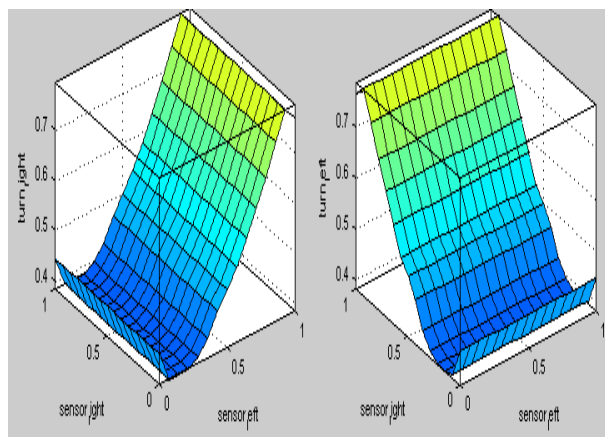


Figura 41 - Superfície do Sistema *Fuzzy* de Navegação (saída esquerda e direita)
Fonte: autoria própria

A figura 42 mostra a superfície *Fuzzy* gerada através das entradas dos dois sensores laterais e a resposta de aceleração.

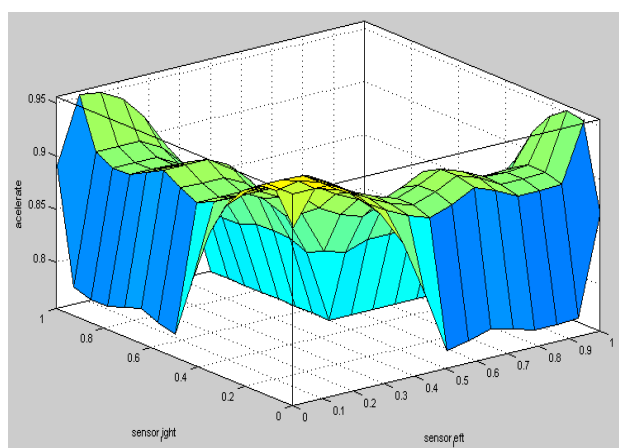


Figura 42 - Superfície do Sistema *Fuzzy* de Navegação (Aceleração)
Fonte: autoria própria

A estratégia empregada utilizou 23 regras que foram criadas para modelar a mesma heurística de navegação empregada na DCN. O modelo *Fuzzy* utilizou o mesmo número de entradas e saídas (3 entradas dos sensores e 3 saídas, virar à direita, à esquerda e acelerar). A base de regras utilizou regras com entrada de um sensor e/ou dois sensores, como por exemplo, sensor esquerdo e sensor frontal ou somente sensor direito. A figura 42 mostra o resultado do efeito da base de regras da saída aceleração relacionada às entradas dos sensores direito e esquerdo. O número 23 foi encontrado através da análise de superposição de regras e do comportamento dinâmico do sistema.

6.3 RESULTADOS DAS SIMULAÇÕES

Um ambiente de simulação com animação 2-D foi também desenvolvido para testar e validar os dois sistemas de navegação propostos. No ambiente de simulação, o rastro com três cores ou tons mais intensos simboliza o robô móvel. Já os sinais de “+” mais claros simbolizam os obstáculos estáticos o sinal “*” tom mais forte representa obstáculo dinâmico e finalmente o “▲” representa os alvos. Nas simulações com obstáculos dinâmicos uma trilha ou rastro também é usada para mostrar a trajetória. Caso seja reproduzida em um ambiente real, as unidades de medidas poderiam ser utilizadas em metros.

As equações de cinemática simulam o comportamento dinâmico do robô e foram inspiradas em Malhotra e Sarkar (2005). Na verdade, o robô simulado corresponde a uma plataforma móvel com dois micromotores que acionam as rodas, e três sensores, um frontal, e dois em cada lado semelhante ao robô educacional em SPARKFUN *Electronics* (2009). Os sensores são de ultrassom, assim, a percepção de barreira ou obstáculo só existe dentro de uma zona de resposta do sensor. Além disso, a intensidade da medida do sensor é inversamente proporcional à distância do objeto.

Este ambiente de simulação serviu inicialmente para a aquisição de conhecimentos através da observação de dados de entrada e saída, e observação do comportamento do robô em várias situações. Posteriormente, três experimentos foram realizados para validar a DCN do sistema de navegação. No primeiro e segundo experimentos, dois diferentes cenários com obstáculos estáticos e dinâmicos foram

simulados. O segundo experimento é semelhante ao primeiro, porém, um obstáculo dinâmico é aleatoriamente inserido no ambiente, durante a navegação do robô. No segundo experimento, o obstáculo com coordenadas (4, 100) aparece como elemento surpresa na trajetória do móvel. Portanto, após a metade da trajetória, este objeto entra no cenário e o exige do sistema de navegação uma decisão drástica. O terceiro experimento é similar aos dois anteriores. Entretanto, é caracterizado pela inclusão do algoritmo de aprendizado por reforço na sintonia das relações causais da DCN.

Na proposta de navegação autônoma em todos os experimentos propostos, os robôs móveis são guiados somente pelas respostas dos sensores, ou seja, não tem registro da posição e orientação durante a trajetória. Além disso, os controladores *Fuzzy* e DCN desses robôs não têm conhecimento prévio do cenário (a exceção dos experimentos com coleta de alvos para a DCN). Para o caso específico da DCN todas as ações de controle nas duas camadas (reativa e deliberativa) e a sintonia são orientadas pelas respostas dos sensores, de acordo com o conhecimento e a estratégia inserida no modelo.

Os resultados são apresentados nas figuras 43 em diante. Nestas figuras, o gráfico mostra o cenário no plano (XY) com o ponto inicial (10,0) e ponto final (0,160) da trajetória do robô. O gráfico também mostra a trajetória dinâmica seguida pelo robô. As falhas evidentes na trajetória representam o aumento de velocidade, quando os sensores não vêem um obstáculo e o robô acelera. Em todos os experimentos, considerou-se que o robô atinge o ponto de destino se a sua posição final estiver em um intervalo horizontal $[-8, 8]$ em torno do ponto final desejado.

Em todos os cenários simulados, existe uma situação crítica em torno da posição $y = 140$. Neste caso, o robô deve tomar a decisão de mover-se em frente, passar entre os dois obstáculos e imediatamente virar à esquerda para evitar uma barreira frontal e em seguida atingir o ponto de destino. Ao analisar os resultados, como por exemplo, nas trajetórias mostradas nas figuras 43 e 44, para respectivamente o navegador DCN e o navegador *fuzzy*, nota-se que o robô tomou as decisões corretas, para ambos os modelos.

A estratégia adotada para a tendência de movimento do robô é permanecer em linha reta até o ponto final, quando está em movimento livre, ou seja, quando um

obstáculo aparece na posição à esquerda, por exemplo, o robô toma uma decisão de virar suavemente para o lado oposto, como mostram as figuras 43 e 44.

No primeiro experimento, aparece um obstáculo dinâmico inesperado no cenário. A presença de obstáculos dinâmicos aumenta a dificuldade da estratégia empregada pelo controlador (HUI, MAHENDAR, PRATIHAR, 2006). Este evento ocorre aproximadamente quando o robô está passando pelas coordenadas (6, 88), como mostra a tomada de decisão correta para evitar uma colisão, mas também mantém uma tendência de movimento de seguir uma linha reta. A figura 43 mostra o rastro do robô móvel guiado pelo controlador DCN, a figura 44 mostra o desempenho do controlador *Fuzzy*.

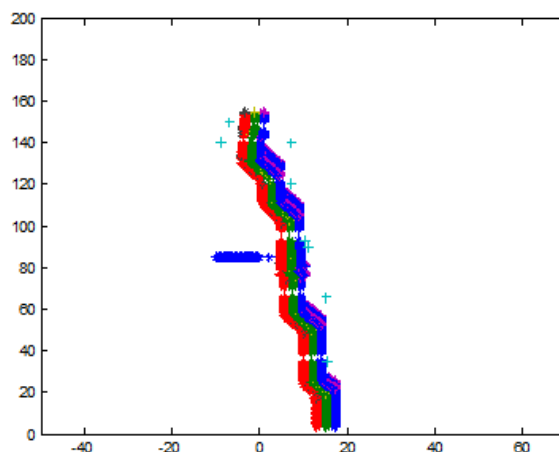


Figura 43 - DCN, Cenário Com Objeto Dinâmico
Fonte: autoria própria

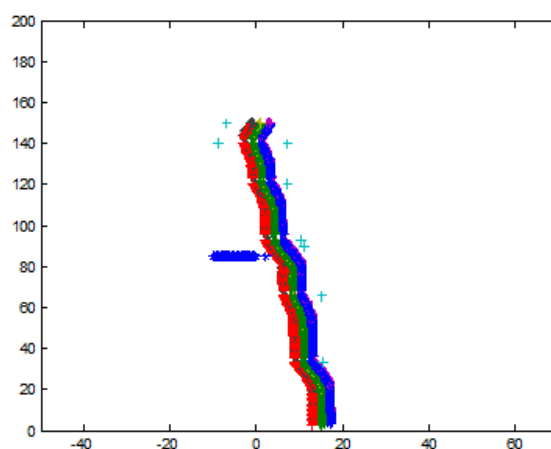


Figura 44 - Fuzzy Cenário Com Objeto Dinâmico
Fonte: autoria própria

No segundo experimento um cenário semelhante ao primeiro é apresentado, e um obstáculo surpresa surge no cenário na posição ($x = 7$, $y = 96$) aproximadamente e obriga o robô móvel a uma tomada de decisão rápida e drástica. Para o navegador DCN, a figura 45 ressalta a posição e o instante do obstáculo surpresa no cenário e a figura 46 mostra toda a trajetória do móvel

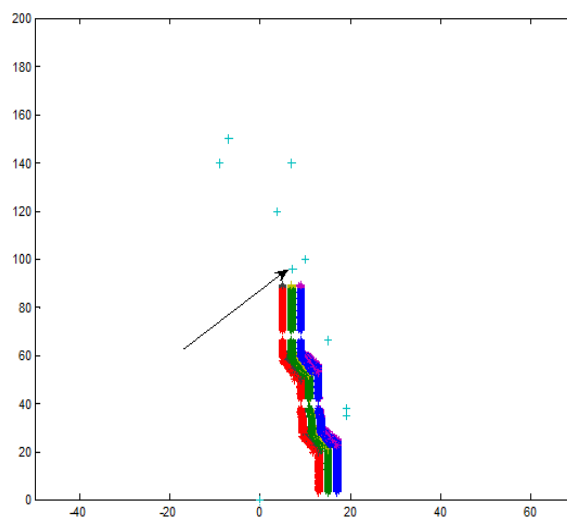


Figura 45 - DCN em Cenário Dinâmico
Antes do Obstáculo Surpresa
Fonte: autoria própria

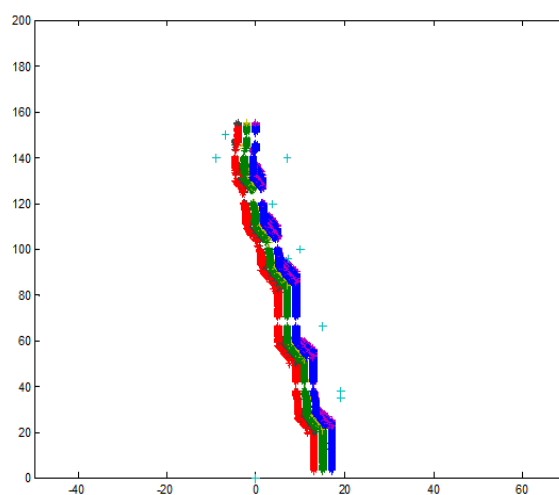
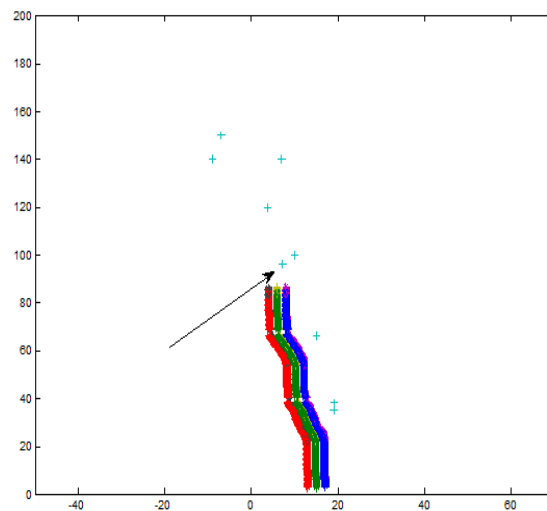
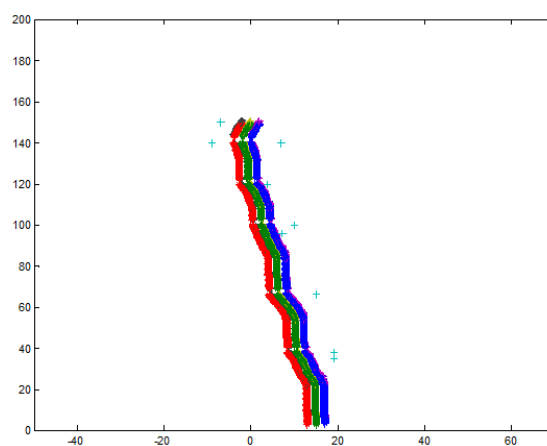


Figura 46 - DCN em Cenário Dinâmico
Com Obstáculo Surpresa
Fonte: autoria própria



**Figura 47 - *Fuzzy* em Cenário Dinâmico
Antes do Obstáculo Surpresa
Fonte: autoria própria**



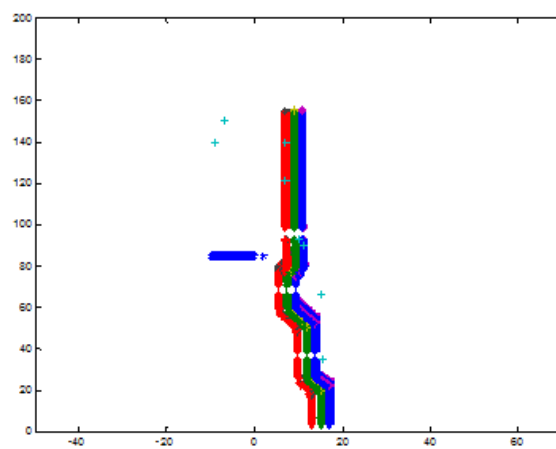
**Figura 48 - *Fuzzy* em Cenário Dinâmico com
Obstáculo Surpresa
Fonte: autoria própria**

Já as figuras 47 e 48 mostram o desempenho do controlador *Fuzzy*. Nesse experimento os dois controladores obtiveram sucesso em suas manobras e concluíram a trajetória na região desejada sem nenhuma colisão.

O experimento 3 apresenta os resultados obtidos em cenários semelhantes aos anteriores, assim todas as situações de tomadas de decisões anteriormente observadas continuam valendo. Entretanto, a velocidade do experimento com obstáculo dinâmico foi aumentada em aproximadamente 20%, e, por conseguinte, gerou dificuldade de manobras para os sistemas de navegação.

A figura 49 mostra os resultados da DCN sem algoritmo de refinamento por reforço, e, de acordo com as tomadas de decisões que levaram a uma colisão, aproximadamente na posição ($x=13$, $y=95$), o que mostra a necessidade de uma melhoria no controle devido ao erro de tomada de decisão, a qual é introduzida pelo uso do algoritmo de aprendizado por reforço.

A figura 50 mostra o sucesso do navegador *fuzzy* que conseguiu corretamente evitar do objeto surpresa que surge à sua esquerda e na sequência desviar-se dos obstáculos à direita. Com a introdução do algoritmo de aprendizagem por reforço, o navegador DCN também consegue realizar a trajetória com sucesso, como mostra a figura 51. Além disso, a rede cognitiva dinâmica tem ligeira vantagem na velocidade de tráfego. A figura 52 mostra a evolução dinâmica dos ajustes dos pesos das relações causais de acordo com as variações do ambiente captada pelos sensores. Observa-se um comportamento suave no ajuste dos pesos das relações causais apesar das variações bruscas dos sensores. Nesta mesma figura (figura 52), a linha verde contínua mostra a variação da relação causal dentro de uma faixa definida pelo especialista [0,35 a 0,65] para a relação causal do sensor frontal e [0,6 a 1] para as relações causais dos sensores laterais. A primeira caixa mostra a variação da relação causal (W_3) e a resposta do respectivo sensor. De modo análogo a caixa 2 e 3 apresentam a variação das relações causais W_1 e W_5 e a resposta dos seus respectivos sensores durante a trajetória em todo o cenário.



**Figura 49 - DCN Obstáculo Inesperado Mais Rápido
(Sem Refinamento)**

Fonte: autoria própria.

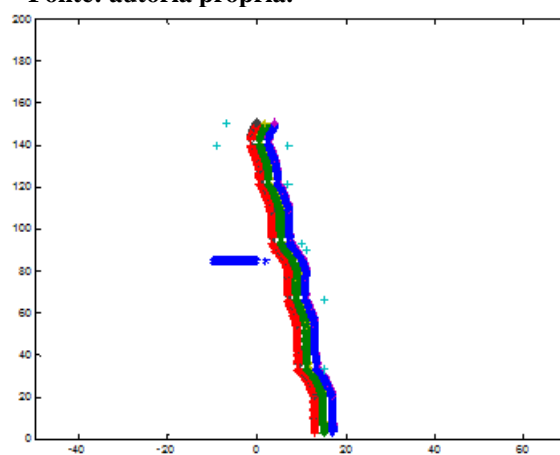
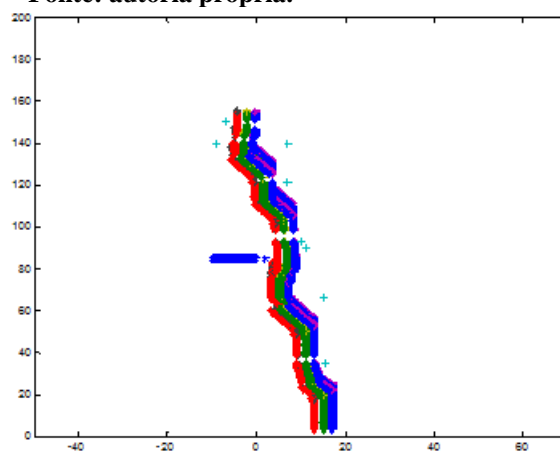


Figura 50 - Fuzzy Obstáculo Inesperado Mais Rápido

Fonte: autoria própria.



**Figura 51 - DCN Obstáculo Inesperado Mais Rápido
(Com refinamento)**

Fonte: autoria própria

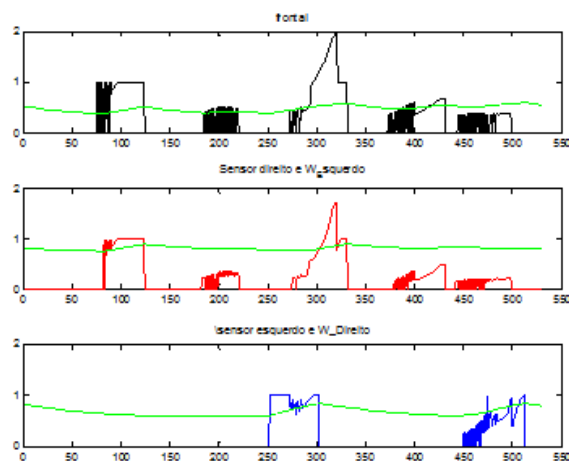


Figura 52 - Sensores e Variações dos Pesos das Relações Causais
Fonte: autoria própria

O próximo experimento objetiva avaliar os dois controladores em um cenário mais difícil, onde o robô móvel tem a mesma velocidade do anterior, porém a posição final do obstáculo móvel, que cruza completamente a trajetória do robô (posição final é em $y=101$), provoca uma situação que exige manobras drásticas e uma tomada de decisão rápida. O navegador baseado em DCN consegue desviar o robô do obstáculo dinâmico e completa a trajetória sem colisões (figura 53).

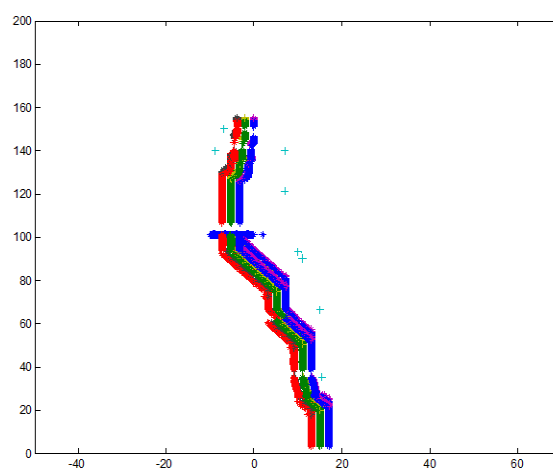


Figura 53 – DCN cenário drástico.
Fonte: autoria própria

A figura 54 mostra os valores das relações causais calculadas com o algoritmo de aprendizado por reforço. O navegador *fuzzy* também consegue desviar do obstáculo dinâmico, porém colide em um obstáculo estático logo após a manobra. A figura 55 mostra a trajetória do móvel com o navegador *fuzzy* em que a flecha indica o ponto de colisão.

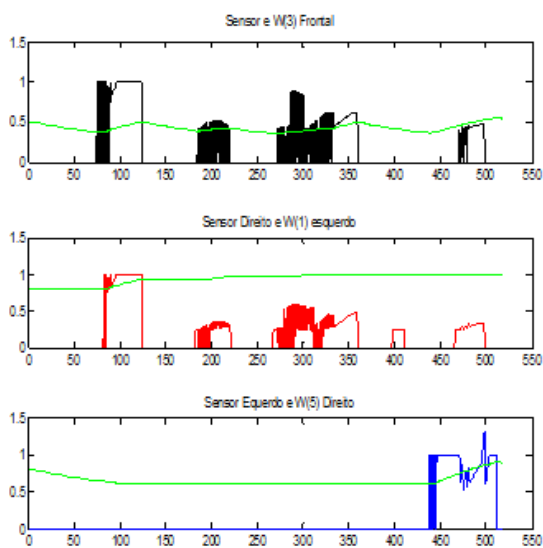


Figura 54 - Sensores e Variações dos Pesos das Relações Causais
Fonte: autoria própria

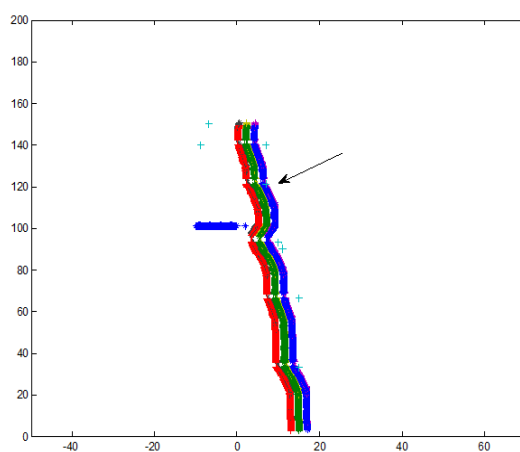


Figura 55 – Fuzzy cenário drástico
(colisão em obstáculo).
Fonte: autoria própria

Comparando as trajetórias do robô nas figuras 53 e 55, observa-se que o navegador DCN manobra para o lado esquerdo quando detecta o obstáculo, enquanto o navegador *fuzzy* manobra para o lado direito, sendo difícil evitar a colisão. Essa diferença nas tomadas de decisão dos dois navegadores ocorre porque apesar seguirem as mesmas heurísticas, os dois sistemas não são exatamente iguais, e, deste modo, em uma situação de dificuldade limite as ações podem ser diferentes, como mostra esse exemplo.

6.4 SIMULAÇÕES COM AQUISIÇÃO DE ALVOS E DESVIO DE OBSTÁCULOS

Essas simulações pretendem demonstrar a funcionalidade de aquisição de alvos, com posição previamente conhecidos, durante a trajetória do robô. De acordo com a arquitetura de controle da figura 39, a tarefa principal do robô é se esquivar de objetos indesejáveis (obstáculos) e recuperar os alvos quando estiver trafegando livre e a uma distância pré-definida desses alvos. Inicialmente são apresentados resultados para um cenário apenas com obstáculos estáticos, em seguida o mesmo cenário é utilizado com o aparecimento de um obstáculo surpresa. Os resultados para o cenário estático são apresentados nas figuras 56 e 57 que mostram respectivamente o cenário e a trajetória do robô móvel, com a presença de um obstáculo móvel. Como pode ser visto na figura 57, o robô executa as ações planejadas corretamente, ou seja, o móvel trafega explorando o ambiente em linha reta caso não tenha nenhum alvo ou obstáculo para coletar e/ou desviar. A figura 58 mostra o comportamento dos sensores e da variação das relações causais provocados pelas ações do algoritmo de aprendizado por reforço, implementado na camada reativa

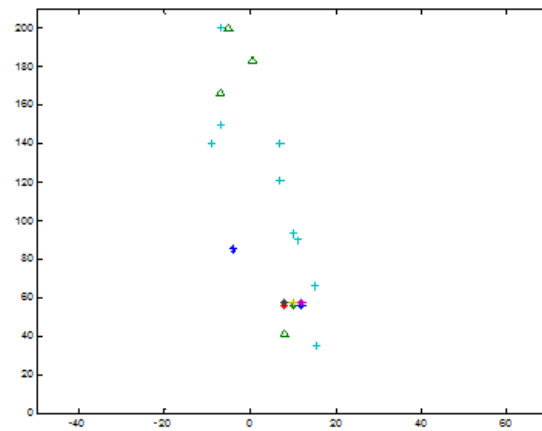


Figura 56 - Cenário da DCN com refinamento e capacidade de coleta de alvos
Fonte: autoria própria

. As regiões de partida e chegada são as mesmas dos experimentos anteriores.

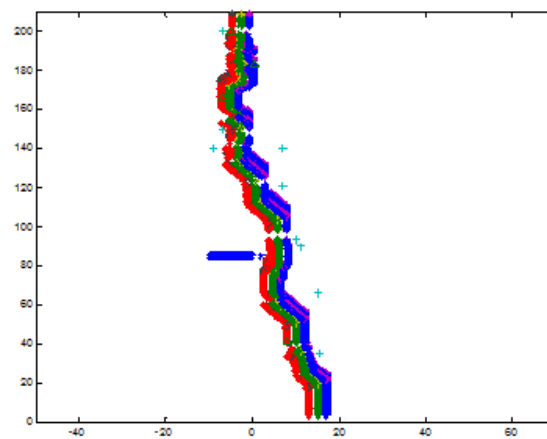


Figura 57 - Cenário e trajetória da DCN com refinamento e capacidade de coleta de alvos
Fonte: autoria própria

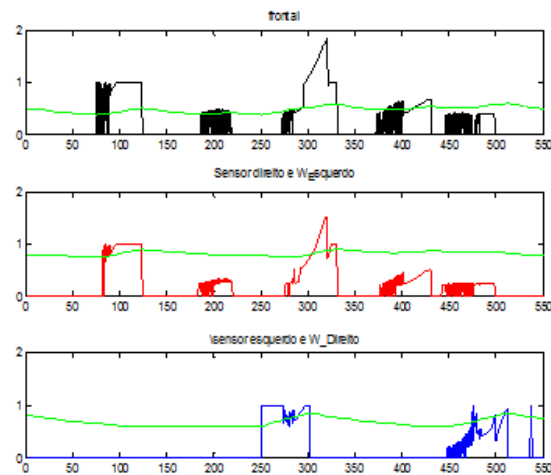


Figura 58 - Sensores e Variações dos Pesos das Relações Causais
Fonte: autoria própria

No experimento a seguir aparece um obstáculo surpresa na posição $x = 7,25$ e $y = 96$, em instantes imediatamente antes da passagem do móvel por essa posição. O cenário para esse experimento é mostrado na figura 59. A trajetória seguida pelo robô e o ajuste dinâmico das relações causais são ilustrados respectivamente nas figuras 59 e 60. Da figura 59, observa-se que o primeiro alvo (posicionado no início da trajetória, a esquerda do robô móvel) não é coletado, isto acontece devido à prioridade das ações. Nesse exemplo, o robô toma a decisão de desviar-se do obstáculo e evitar uma colisão, antes de colher o alvo. Essa decisão é possível devido à prioridade de tarefas estabelecidas pela camada deliberativa, na qual a estratégia de exploração adotada prioriza evitar colisões do que coletar alvos

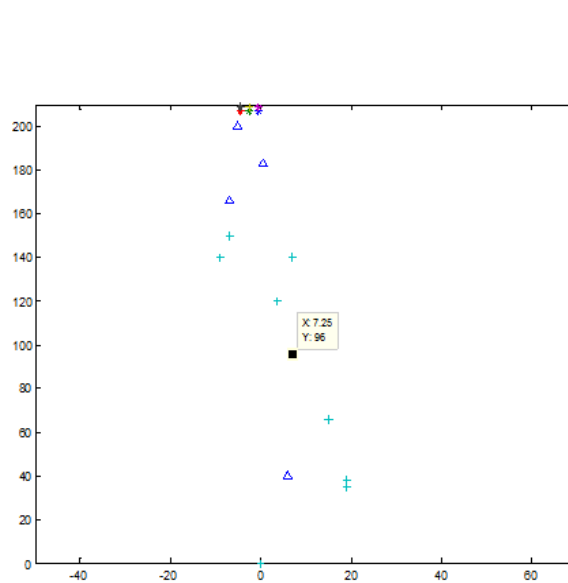


Figura 59 - Cenário e trajetória da DCN com refinamento e capacidade de coleta de alvos
Fonte: autoria própria

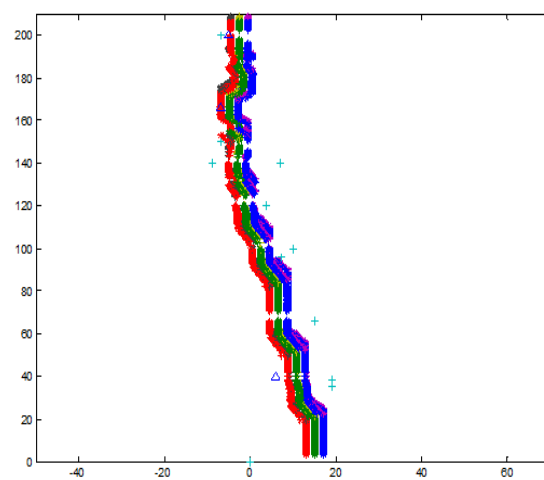


Figura 60 - Cenário e Trajetória da DCN com Refinamento e Capacidade de Coleta de Alvos
Fonte: autoria própria

. A figura 61 mostra a adaptação das relações causais da DCN através da leitura dos sensores.

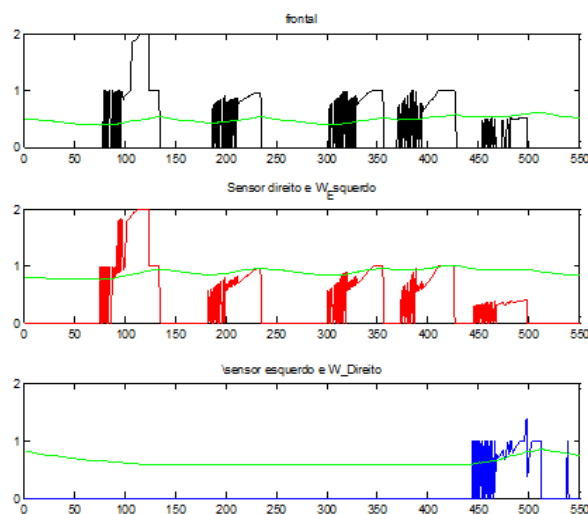


Figura 61 - Sensores e Variações dos Pesos das Relações Causais
Fonte: autoria própria

6.5 EXPERIMENTOS EM CENÁRIO REAL

Para a validação final do navegador DCN, dois experimentos (com cenários diferentes) foram realizados com um robô didático. O navegador DCN da figura 36, contendo apenas a camada reativa, foi codificado em linguagem **C** e implementado em um Netbook Intel Atom N270, 1GB de memória. A plataforma robótica utilizada é um Dr Robot X-80 com sensores do tipo ultrassom dispostos nas duas laterais e na frente. As saídas da DCN sofreram uma pequena adaptação devido à necessidade de acionamento dos motores de passo para cada uma das rodas, direita e esquerda. A saída direita do navegador acelera a roda esquerda, e, de maneira inversa, a saída esquerda acelera a roda direita. Já a saída “aceleração” aciona as duas rodas realizando o controle de velocidade quando o robô anda em linha reta.

Dois cenários foram construídos para avaliar a capacidade de o robô desviar-se de obstáculos, e, deste modo, avaliar as tomadas de decisão do controlador em situações reais.

As figuras 62, 63 e 64 mostram o cenário e três momentos de tomadas de decisão do experimento 1 com cenário real.



**Figura 62- Experimento 1: Cenário Real,
Próximo ao Início da Trajetória
Fonte: autoria própria**



**Figura 63 – Experimento 1: Cenário Real,
Próximo ao Meio da Trajetória
Fonte: autoria própria**



**Figura 64 – Experimento 1: Cenário Real,
Próximo do Final da Trajetória
Fonte: autoria própria**

As figuras 65, 66 e 67 mostram o desempenho do robô no segundo experimento com um cenário real diferente do anterior (obstáculos colocados em pontos diferentes).



Figura 65 – Experimento 2: Cenário Real, Próximo ao Início da Trajetória
Fonte: autoria própria



Figura 66 – Experimento 2: Cenário Real, Próximo ao Meio da Trajetória
Fonte: autoria própria



Figura 67 – Experimento 2: Cenário Real, Próximo do Final da Trajetória
Fonte: autoria própria

Em ambos os experimentos com cenários reais, as tomadas de decisões do navegador foram satisfatórias, ou seja, o robô tomou decisões que realizaram ações para mudança de direção e velocidade. Deste modo, foi possível contornar os obstáculos e chegar a uma região final do cenário. De forma semelhante às simulações, o robô também não tinha nenhum conhecimento prévio do cenário.

6.6 CONCLUSÃO DO CAPÍTULO

Este capítulo desenvolveu um sistema autônomo de navegação baseado em mapas cognitivos, especificamente em uma DCN. Esta arquitetura utiliza novos tipos de relações e conceitos diferentes de um FCM clássico, que permitem modelar a habilidade humana de tomar decisões na presença de eventos imprevistos. Além disso, nessa arquitetura, o conhecimento humano pode ser representado através de mapas cognitivos e também por regras que são acionadas quando uma situação crítica ocorre.

Um processo de refinamento do conhecimento, inspirado em algoritmos de aprendizado por reforço, aumentou a capacidade de adaptação do sistema de navegação. Esse fato pode ser observado no experimento com a presença de obstáculo móvel com maior velocidade. Esta abordagem é uma contribuição deste trabalho para a área de sistemas inteligentes.

De acordo com os resultados apresentados neste trabalho, pode-se concluir que a proposta de arquitetura DCN constitui uma ferramenta flexível e robusta para o desenvolvimento de sistemas de navegação que devem operar em ambientes desconhecidos e imprevisíveis. Uma das principais vantagens da abordagem proposta é que a aquisição de conhecimento e sua representação são simplificadas pelo uso de modelos cognitivos.

Quando comparado com um navegador *fuzzy*, o número de regras utilizado foi bem menor para a descrição da mesma heurística. Os resultados dos dois sistemas foram semelhantes, com uma vantagem mínima para o navegador DCN, que consegue alinhar de um modo mais suave a sua trajetória após o aparecimento dos obstáculos surpresa. Este resultado é devido à sua capacidade de se adaptar a variações não previstas do ambiente.

Nos experimentos com coleta de alvos, o sistema inteligente baseado em DCN executou tomadas de decisões multitarefa (coleta alvos, desvio de obstáculo), funcionando de modo semelhante à arquitetura do Brooks (1986).

De um modo geral, a arquitetura desenvolvida para exploração de ambientes apresenta algumas das propriedades encontradas em certas classes de sistemas autônomos:

- Capacidade de adquirir conhecimento (sintonia dinâmica do modelo cognitivo) por meio de interação com o ambiente;
- Capacidade de adaptar seu comportamento com base em suas informações sensoriais, utilizando técnicas de adaptação e/ou aprendizado; constatado nos experimentos que empregaram aprendizagem por reforço.
- Capacidade de encontrar soluções factíveis que atendam simultaneamente a múltiplos objetivos (possivelmente conflitantes); o robô móvel conseguiu organizar suas ações de modo que consegue coletar alvos, evitar obstáculos e navegar em linha reta de acordo com a “percepção” dos sensores.
- Capacidade de operar em condições adversas caracterizadas pela ausência de informações consistentes, imprevisibilidade do ambiente, ruído nos sensores e atuadores, entre outras, como sugerem os resultados obtidos pelo experimento com o robô real. Além disso, apresentaram portabilidade, os códigos implementados em linguagens diferentes.

7 CONCLUSÕES

7.1 INTRODUÇÃO

Neste capítulo são discutidas as principais vantagens e dificuldades encontradas na modelagem de sistemas dinâmicos utilizando a DCN proposta nesta Tese. Como conclusões são apresentadas as principais contribuições para o desenvolvimento de sistemas inteligentes obtidas com a metodologia proposta. Também são discutidos trabalhos futuros para a expansão da metodologia e para a sua aplicação em sistemas supervisórios, controle adaptativo, sistemas autônomos de navegação, estratégia de times de futebol de robôs, sistemas virtuais entre outros.

7.2 VANTAGENS E DESVANTAGENS NO USO DE SISTEMAS INTELIGENTES BASEADOS EM DCN

De modo geral, resumem-se algumas vantagens da utilização de DCNs para modelar processos complexos:

- Modularidade, a qual permite a construção de grandes modelos a partir da conexão de modelos menores.
- Conectividade, a qual reflete a capacidade de agrupamento de diferentes mapas.
- Não é necessária a existência de um modelo matemático do sistema a ser modelado.
- Capacidade de modelagem através de conhecimento empírico e prático de especialistas.
- Capacidade de treinamento e aprendizado da DCN, a partir de dados históricos do sistema, a qual possibilita minimizar ou suprimir dificuldades de aquisição de conhecimentos especialista no desenvolvimento do modelo. Este treinamento pode ser realizado de modo *on-line* ou *off-line*.

- Flexibilidade na documentação do conhecimento empírico do sistema através de um agrupamento intuitivo de causas e efeitos, na qual todas ou as principais variáveis do processo são relacionadas.
- Existência de um tratamento matemático para os modelos baseado na teoria de grafos.
- Capacidade de tomada de decisão através da inserção de base de regras na camada superior da arquitetura proposta.
- Capacidade de refinamento em tempo real de acordo com mudanças nas variáveis de ambiente, através da incorporação de um algoritmo de aprendizado por reforço.

No entanto, encontraram-se duas dificuldades na modelagem por mapas cognitivos: A primeira dificuldade está na abstração do conhecimento através de mapas que relacionam as variáveis do problema e as causas entre elas. Deste modo, após a familiarização com este tipo de modelo, as relações e os conceitos passam a ser de fácil leitura.

Outra desvantagem está na necessidade de desenvolvimento de uma interface de entrada e saída para se condicionar os sinais dentro das faixas de operação do mapa e a correspondência entre os valores atingidos e as variáveis controladas. Apesar de existir uma correspondência entre os conceitos e as variáveis reais, garantindo uma boa interpretabilidade do modelo (semelhante aos dos sistemas *fuzzy*), os valores assumidos por estes conceitos não tem nenhuma relação com seus valores reais, como acontece com os valores dos neurônios nas redes neurais.

Uma terceira desvantagem reside no fato de que, como as outras técnicas inteligentes tais como sistemas *fuzzy* e RNA entre outros, as redes dinâmicas cognitivas são heurísticas, e não se pode garantir uma solução ótima.

7.3 CONTRIBUIÇÕES

Essa Tese desenvolveu uma metodologia para modelagem de sistemas inteligentes baseada em redes dinâmicas cognitivas, que são um tipo de mapa cognitivo,

com aplicações em diferentes áreas de conhecimento. Os mapas cognitivos são representações mentais em que o conhecimento é modelado e organizado de uma forma estruturada e incremental. Esse conhecimento inicialmente modelado pode ainda ser otimizado, adaptado e /ou ajustado através de dados empregando metaheurísticas, como por exemplo, algoritmos genéticos, técnica de inteligência coletiva ou até mesmo por um processo de aprendizagem reversa empregando RNA.

Durante o desenvolvimento da Tese, estudos iniciais para construção de modelos cognitivos determinaram a inclusão nos mapas cognitivos de novos tipos de conceitos e relações que permitiram abstrair conhecimento qualitativo na forma de eventos e aumentou a capacidade de inferência de um mapa cognitivo. Como foi observado nos resultados de navegação robótica, ainda é possível fazer um ajuste e/ou refinamento dinâmico em um modelo inicialmente proposto empregando algoritmos de aprendizagem por reforço. Deste modo, a arquitetura cognitiva proposta baseada em DCN é um modelo inteligente híbrido que pode ser utilizado para a construção de controladores e/ou máquinas de inferência em diferentes áreas de conhecimento.

De fato, a metodologia baseada em DCN proposta é uma ferramenta computacional híbrida apropriada para a criação de modelos a partir do conhecimento explícito qualitativo originário de especialistas humanos e em seguida adaptado e sintonizado a partir de conhecimento implícito, embutido em um conjunto de dados obtidos *on-line* através de sensores ou de um banco de dados histórico.

Os resultados simulados apresentados neste trabalho credenciam a metodologia baseada em DCN proposta para a modelagem de sistemas dinâmicos. Diferentes propostas de técnicas computacionais inteligentes foram utilizadas na otimização, aprendizado, adaptação e tomadas de decisão do modelo inteligente de acordo com a aplicação modelada.

Em especial, um sistema supervisor foi desenvolvido para um processo de fermentação. Apesar de apenas a tarefa de geração de *set-point* para o nível regulatório da planta ter sido apresentada, outras funções pertinentes aos sistemas supervisórios tais como detecção de falhas, monitoração, diagnósticos entre outras podem ser facilmente incluídas no supervisor. Cada uma delas corresponderia a uma DCN que seria conectada às outras através de conceitos comuns.

Também um sistema de navegação autônoma híbrido foi desenvolvido. O sistema combina uma camada reativa e uma camada deliberativa numa arquitetura de subsunção de Brooks. Este navegador foi validado através de simulação e em experimento com um robô real.

De um modo geral, espera-se ter contribuído com o desenvolvimento de uma metodologia para a modelagem de sistemas inteligentes baseado em DCN com as seguintes características:

- Flexibilidade para agrupar diferentes modelos cognitivos, permitindo a inclusão de novas funcionalidades em sistemas já existentes, característica herdada dos FCM (KOSKO E DICKERSON, 1994, 1996) como na coleta de alvos do navegador robótico e na inclusão de funções de geração de *set-point* no fermentador.
- Combinação de várias técnicas inteligentes para refinamento e/ou adaptação do modelo baseada em dados (treinamento) e/ou interação com o ambiente (aprendizado por reforço).
- Um sistema construído em duas etapas que coexistem, em um nível básico, com decisões mais simples (tomadas de decisões de baixo nível) e em um nível estratégico para a ocorrência de eventos, como mostrou o desenvolvimento da DCN dos capítulos 5 e 6.
- De acordo com a construção da DCN citada acima, pode ser observada a portabilidade de modelos FCM. Essa característica é devido à primeira etapa, na qual se modela as relações de causa e efeito e os conceitos, de modo semelhante à construção de um FCM clássico. Essa identificação inicial pode ser observada nos trabalhos de Miao e colaboradores (2001, 2010).
- Um sistema inteligente fácil de embarcar, como no experimento com robô real do capítulo 6.
- Uma ferramenta de documentação do conhecimento, ou seja, a DCN apresenta uma representação que após a elaboração do modelo final é mais fácil de ser interpretada por especialistas que conhecem o modelo físico.

7.4 FUTUROS TRABALHOS

Os seguintes tópicos norteiam novas aplicações da metodologia baseada redes cognitivas dinâmicas:

- Implementar novas funções no sistema supervisor do processo de fermentação, como por exemplo, implementar funções de segurança e detecção de falhas.
- Explorar a estratégia de time de futebol de robôs em simuladores e times de robôs reais objetivando a facilidade de integração, e a adaptação dinâmica da DCN apresentada.
- Implementar novas funções no navegador que darão flexibilidade e capacidade de inferência para realização de outras tarefas e funcionalidades como por exemplo, gerenciamento de energia, interação com outros agentes autônomos em um mesmo ambiente.
- Implementar a DCN (utilizada nas simulações) em um robô móvel real, em especial, a DCN2 com cenários mais complexos, como por exemplo, incluir obstáculos surpresa e/ou obstáculos dinâmicos.
- Novos testes com novos cenários e/ou diferentes modelos de robôs com mais funcionalidades empregando DCN também poderão fazer parte de futuras aplicações da ferramenta computacional inteligente apresentada nessa Tese. Como por exemplo, a inclusão de câmeras nos robôs.
- Investigar a possibilidade de se empregar estimativas bayesianas para ajuste e/ou adaptação das relações e conceitos de fator e seleção que modelam a ocorrência de eventos em arquiteturas DCN.
- Utilizar a capacidade de agrupamento da DCN, em uma planta de grande porte, objetivando o desenvolvimento de um sistema do tipo wide-control, capaz de controlar e, ou, supervisionar diferentes processos dentro de uma mesma planta. Esse sistema seria construído modularmente, com uma DCN dedicada a cada processo e uma DCN maior que realizaria a coordenação de todos os mapas interligados.
- Desenvolver e simular novas arquiteturas de controle inteligente

empregando DCN como modelo de inferência e tomada de decisões, agregado a uma base de conhecimentos, obtidos através de conhecimento heurístico humano.

- Desenvolver novos algoritmos de aprendizagem por reforço, e / ou construtivos na criação ou exclusão dinâmica de conceitos e de relações para aplicação diferentes áreas de conhecimento, como em navegação móvel. Por exemplo, promover maior adaptação do modelo durante a execução da trajetória do robô móvel de acordo com mudanças no ambiente, cinemática e/ou dinâmica do robô móvel.

7.5 CONCLUSÃO FINAL

Essa Tese apresentou uma contribuição para o desenvolvimento de sistemas inteligente através de uma metodologia baseada em redes dinâmica cognitivas. A arquitetura desenvolvida permite a tomada de decisões dinâmicas através do mapeamento de duas camadas distintas que modelam diferentes tipos de conhecimento. Essas camadas cooperam e contribuem para o sucesso da aplicação, podendo se necessário, utilizar métodos de treinamento (RNA, GA, PSO) para calibração e sintonia *off-line* do modelo, ou ainda algoritmos de aprendizagem por reforço para aumentar a capacidade de adaptação dinâmica do modelo.

REFERÊNCIAS

AGUILAR, J. Dynamic random *Fuzzy* cognitive maps. **Computación y Sistemas**, v. 7, n. 4, p. 260-270, 2004.

ALIZADEH, S., GHAZANFARI, M., JAFARI, M., HOOSHMAND, S.: Learning FCM by tabu search. *International Journal of Computer Science* 3, 142–149, 2007.

ARANIBAR, D. B.; ALSINA, P. J. Reinforcement learning-based path planning for autonomous robots. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, p. 10, 24. Salvador, 2004.

ARKIN, R. C. **Behavior-based robotics**. Cambridge: MIT Press, 1999.
ARRIAGA-DE-VALLE, E.; DIECK-ASSAD, G. Modeling and simulation of a *Fuzzy* supervisory controller for an Industrial boiler. **Simulation**, v. 82, n. 12, p. 841-850, Dec. 2006.

ASAMI, S. Robots in Japan: present and future. **IEEE Robotics and Automation Magazine**, v. 1, n. 2, p. 22-26, 1994.

ASTRÖM, K. J.; WITTENMARK, B. **Adaptive control**. 2nd. Boston: Addison-Wesley Longman Publishing, 1995.

ASTUDILLO, L. et al. Intelligent control of an autonomous mobile robot using type-2 *Fuzzy* logic. **Engineering Letters**, v. 13, n. 2, 2006.

AXELROD, R. **Structure of decision**: the cognitive maps of political elites. New Jersey: Princeton University Press, 1976.

BAYKASOGLU, A. DURMUSOGLU Z.U.D.; KAPLANOGLU V. **Training Fuzzy Cognitive Maps via Extended Great Deluge Algorithm with applications**. Computers and Industry. Elsevier, 2010.

BHANU, B. LEANG P; COWDEN C.; LIN Y. AND PATTERSON M. . Real-time robot learning. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001. p. 491-498.

BOSE, N. K.; LIANG, P. **Neural network fundamentals with graphs, algorithms, and applications**. New York: McGraw-Hill, 1996.

BROOKS, R. A. A robust layered control system for a mobile robot. **IEEE Journal of Robotics and Automation**, v. 2, n. 1, p. 14-23, Mar. 1986.

BROOKS, R. A. Elephants don't play chess. **Robotics and Autonomous Systems**, v. 6, p. 3-15, 1990.

CALLAI, T. C.; COELHO L.S.; COELHO A.A.R.. Controle nebuloso adaptativo por modelo de referência: projeto e aplicação em sistemas não lineares. **SBA Controle & Automação**, Campinas, v.18, n. 4, p. 479-489, Oct./Dec. 2007.

CALVO, R.; ROMERO, R. A. F A Hierarchical self-organizing controller for navigation of mobile robots. In: IEEE WORLD CONGRESS COMPUTATIONAL INTELLIGENCE, 2006, Vancouver. **Proceedings of International Joint Conference on Neural Networks**.

CALVO, Rodrigo. **Arquitetura híbrida inteligente para navegação autônomo de robôs**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional). IMC-USP, 2007.

CARVALHO, J. P.; TOMÉ, J. A. Rule based Fuzzy cognitive maps-qualitative systems dynamics. In: PROCEEDINGS 19th INTERNATIONAL CONFERENCE OF THE NORTH AMERICA. FUZZY INFORMATION FUZZY PROCESSING SOCIETY, 2000, p. 407-411.

CARVALHO, J. P. B. **Mapas cognitivos baseados em regras difusas: modelação e Simulação da Dinâmica de Sistemas Qualitativos**. 2001. 361 f. Tese (Doutorado em Engenharia Electrotécnica e de Computadores) – Instituto Superior Técnico, Universidade Técnica de Lisboa, 2001. Disponível em: <http://comp.ist.utl.pt/uke/papers/TeseRB-FCM.pdf>. Acesso em: 12 nov. 2010.

CHEN, M.; HUANG, Y. Guard heuristic by dynamic *Fuzzy* reasoning model for Chinese chess. In Proceedings of the 3rd international Symposium on Uncertainty Modeling and Analysis (March 17 - 20, 1995). ISUMA. IEEE Computer Society, Washington, DC, 1995. p. 530-533.

CHONG H.-Q., TAN A.-H., and NG. G.-W. Integrated cognitive architectures: a survey. **Artif. Intell. Rev.** 28, 2, 103-130, (August 2007). DOI=10.1007/s10462-009-9094-9 <http://dx.doi.org/10.1007/s10462-009-9094-9>

CHUN-MEI, L. Using Fuzzy cognitive map for system control. **WSEAS Transactions on Systems**, v. 7, n. 12, p. 1504-1515, Dec. 2008.

CLERCAND, M.; KENNEDY, J. The particle swarm The particle swarm explosion, stability, and convergence in a multidimensional complex space. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 1, p. 58-73, 2001.

COELHO, L. S.; ALMEIDA, O. M.; COELHO, A. A. R. Projeto e estudo de caso da implementação de um sistema de controle nebuloso. **Sba Controle & Automação**, Campinas, v. 14, n. 1, p. 20-29, jan./mar. 2003.

COLOMBETTI, M.; DORIGO, M.; BORGHI, G. Behavior analysis and training: a methodology for behavior engineering. **IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics**, v. 26, n. 3, p. 365-380, Jun. 1996.

COPPIN, B. **Inteligência artificial**. Rio de Janeiro: Livros Técnicos e Científicos, 2010.

COSTA, A. H. R.; PEGORARO, R. Construindo robôs autônomos para partidas de futebol: o time guaraná. **SBA Controle & Automação**, Campinas, v. 11, n. 3, p. 141-149, set./dez. 2000.

DICKERSON, J. A., KOSKO, B. Virtual worlds as fuzzy cognitive maps. **Presence**, v. 3, n. 2, p. 173-189, 1994.

DICKERSON, J. A., KOSKO, B. Virtual worlds as fuzzy dynamical systems. In: SHEU, B. (Ed.) **Technology for multimedia**, New York: IEEE Press, 1996.

FABRO, J. A.; ARRUDA, L. V. R. *Fuzzy-neuro predictive control, tuned by genetic algorithms, applied to a fermentation process*. IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL HOUSTON, Texas, October 5-8, 2003.

FARIA, G.; ROMERO, R. A. F. Estratégia para futebol de robôs baseada em campos potenciais. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 24., 2004, Salvador, p. 10.

FIGUEIREDO, K. VELLASCO, M.; PACHECO, M.; SOUZA, F. Modelo Neuro-Fuzzy hierárquico politree com aprendizado por reforço para agentes inteligentes. **SBA Controle & Automação**, Campinas, v. 18, n. 2, p. 234-250, abr./jun. 2007.

FIGUEIREDO, L. C; ALMEIDA; P.E.M., BRAGA A.R.; JOTA, F.G.; ARAÚJO, E.O. Ambiente integrado para análise e desenvolvimento de controladores difusos. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 1. p. 291-299. , 1993, Rio Claro.

FIGUEIREDO, M. **Navegação autônoma de robôs**. Novo Hamburgo: VII Escola de Informática da SBC - Regional Sul, 1999. p. 74 - 106.

FRACASSO, P. T., COSTA, A. H. R. Navegação reativa de robôs móveis utilizando lógica nebulosa com regras ponderadas. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 7.; IEEE LATIN-AMERICAN ROBOTICS SYMPOSIUM, 2., 2005, São Luís. **Anais...** São Luís, 2005. Disponível em: <http://www.lti.pcs.usp.br/publicacoes/FracassoCosta05-SBAI.pdf> Acesso em: 12 nov. 2010.

GASKETT, C., FLETCHER, L., ZELINSKY, A. Reinforcement learning for a vision based mobile robot. In: PROCEEDINGS OF THE IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2000.

GHAZANFARI, M., ALIZADEH, S., FATHIAN, M., KOULOURIOTIS, D.E.: Comparing simulated annealing and genetic algorithm in learning fcm. *Applied Mathematics and Computation* 192(1), 56–68, 2007.
GHAZANFARI, M.; ALIZADEH, S.; JAFARI, M. Using Fuzzy expert system for solving *Fuzzy* system dynamics. In: PROCEEDING OF THE FIRST EURO ASIA-ICT CONFERENCE (ADVANCES IN INFORMATION AND COMMUNICATION TECHNOLOGY) Shiraz, Iran, October 2002. Disponível em: <http://ce.sharif.edu/~shafazand/WORK-english/pdf/349-710/paper4.pdf> Acesso em: 12 nov. 2010.

GLYKAS, Michael. **Fuzzy Cognitive Maps** Advances in Theory, Methodologies, Tools and Applications. Greece Springer: 2010.

GOLDBERG, D. E. **Genetic algorithms in search optimization and machine learning**. Mass: Addison-Wesley, 1989.

GOTOH, K. et al. Application of *Fuzzy* cognitive maps to supporting for plant control. PROCEEDINGS OF THE SICE JOINT SYMPOSIUM OF 15TH SYST. SYMPOSIUM AND 10TH KNOWLEDGE ENGINEERING SYMPOSIUM, 1989. p. 99–104.

GUDWIN, Ricardo Ribeiro, **Semiônica: Uma Proposta de Contribuição à Semiótica Computacional**, 2002, Tese (obtenção do Título de Professor Livre-Docente) Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2002.

HAGIWARA, M. Extended *Fuzzy* cognitive maps. In: PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEM, New York, 1992. p. 795–801.

HARISHA S.K., RAMKANTH KUMAR P., KRISHNA M., and SHARMA S. C. Fuzzy Logic Reasoning to Control Mobile Robot on Pre-defined Strip Path. PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 32 AUGUST 2008 ISSN 2070-3740. 2008.

HAYKIN, S. **Redes neurais, princípios e prática**, 2. ed. São Paulo: Bookman, 2000.

HOFFMANN, L. T.; SILVA, J. D. S. Modelagem de um agente móvel de aprendizagem para vagueio em ambientes inexplorados. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 25. 2005, São Leopoldo, p. 882-891.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor: University of Michigan Press, 1975.

HUANG, Y. C., WANG, X. Z. Application of *Fuzzy* causal networks to waste water treatment plants. **Chemical Engineering Science**, v. 54, n. 13/14, p. 2731-2738, Jul. 1999. Disponível em: <http://www.sciencedirect.com/science/article/B6TFK3WNMYW2W/2/7b1153d946e1289f1e617f4b57ccb27f>. Acesso em: 12 nov. 2010.

HUI N.B., MAHENDAR V., PRATIHAR D.K., Time-optimal, collision-free navigation of a car-like mobile robot using a neuro-fuzzy approach, *Fuzzy Sets and Systems*, 157(16), pp. 2171-2204, 2006.

ISMAEL, A.; HUSSIEN, B.; MCLAREN, R. W. *Fuzzy* neural network implementation of self tuning PID Control. IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, Columbus, 1994.

JANG, J. R. ANFIS: Adaptive-Network-Based *Fuzzy* Inference Systems. **IEEE Transactions on Systems, Man and Cybernetics**, v. 23, n. 3, p. 665-685, May/Jun. 1993.

JONES, P. M.; ROY, R.; CORBETT, J. Modeling Physical Systems using *Fuzzy* Inference Cognitive Maps. IEEE ANNUAL MEETING OF THE *FUZZY* INFORMATION, 2004. PROCESSING NAFIPS '04.

KEMPOWSKY, T., SUBIAS, A.; AGUILAR-MARTIN, J. Process situation assessment: from a *Fuzzy* partition to a finite state machine. **Engineering Applications of Artificial Intelligence**, v. 19, n. 5, p. 461–477 2006.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORK. Perth, Australia, 1995. p. 1942-1948.

KING, P. J.; MAMDANI, E. H., The application of *Fuzzy* control systems to industrial process. **Automática**, v. 13, n. 3, p. 235-242, May 1977.

KOSKO, B. *Fuzzy* cognitive maps. **International Journal Man-Machine Studies**, v. 24, n. 1, p.65-75, 1986.

KOSKO, B. **Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence**. New York: Prentice Hall, 1992.

KOTTAS, T. L.; BOUTALLIS, Y. S.; CHRISTODOULOU, M. A. *Fuzzy* cognitive network: a general framework. **Intelligent Decision Technologies**. v. 1, n.4, p. 183-196, 2007.

KOULOURIOTIS, D. E.; DIAKOULAKIS I.E.; EMIRIS D.M.; ZOPOUNIDIS C.D.. Development of dynamic cognitive networks as complex systems approximators: validation in financial time series. **Applied Soft Computing**, v. 5, n. 2, p. 157-179. Jan. 2005.

KOULOURIOTIS, D.E., DIAKOULAKIS, I. E.; EMIRIS, D. M **Learning Fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior**. In: IEEE Congr. On Evolutionary Computation (CEC2001), p. 364-371, 2001.

LEE, C.; TENG, C. Tuning of PID controllers for unstable processes based on gain and phase margin specifications: a *Fuzzy* neural approach. **Fuzzy Sets and Systems**, v. 128, n. 1, p. 95-106, May 2002.

LEE, K. C.; LEE, S. A cognitive map simulation approach to adjusting the design factors of the electronic commerce web sites. **Expert Systems with Applications**, v. 24, n. 1, p. 1-11, Jan. 2003.

LIN, C.; LEE, C. S. **Neural Fuzzy Systems: a neuro-Fuzzy synergism to intelligent systems**. New York: Prentice-Hall, 1996.

LIU, Z. Q.; SATUR, R. Contextual *Fuzzy* cognitive map for decision support in geographic information systems. **IEEE Transactions and Fuzzy Systems**, v. 7, p. 495–507, 1999.

MACEK, K.; PETROVIC, I.; PERIC, N. A reinforcement learning approach to obstacle avoidance of mobile robots. In: INTERNATIONAL WORKSHOP ON ADVANCED MOTION CONTROL, 2002. P. 462-466.

MAES, P. Artificial life meets entertainment: life like autonomous agents. **Communications of the ACM**, v. 38, n. 11, p. 108-114, Nov. 1995. Disponível em: <http://xml3.nada.kth.se/kurser/kth/2D1381/ArtificialLifeMaes.pdf> Acesso em: 12 nv. 2010.

MAHER, M. **Modélisation et élaboration d'algorithmes d'estimation et de commande**: application à um bioprocédé. 1995. Tese - Université Paul Sabatier, LAAS/CNRS, Toulouse, 1995.

MALHOTRA, R.; SARKAR, A. Development of a *Fuzzy* logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment. PROCEEDINGS OF THE 2005 IEEE/ASME INTERNATIONAL Monterey, 24-28 July, 2005.

MANDOW, A. et al. The autonomous mobile robot AURORA for greenhouse operation. **IEEE Robotics and Automation Magazine**, v. 3, n. 4, p. 18-28, 1996.

MATEOU, N. H.; ANDREOU, A. S. A framework for developing intelligent decision support systems using evolutionary *Fuzzy* cognitive maps. **Journal of Intelligent and Fuzzy Systems**, v. 19, n. 2, p. 151-170, 2008.

MELO JUNIOR, W. S. et al. Controle preditivo multivariável baseado em modelo adaptativo aplicado a um fermentador alcoólico. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 14., 2002, Natal. **Anais...**Natal, 2002. p. 2816-2821.

MENDONÇA, M.; ARRUDA, L. V. R.; NEVES JUNIOR, F. Mapas cognitivos aplicados ao controle qualitativo de processos. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 9., 2009, Brasília.

MIAO, Y., LIU Z-Q, SIEW, C. K., and MIAO C. Y. Dynamical cognitive network: an extension of *Fuzzy* cognitive map. **IEEE Transactions on Fuzzy Systems**, v. 9, n. 5, p. 760-770, 2001.

MIAO, Y. MIAO C. Y. TAO X., SHEN Z. LIU Z. Transformation of cognitive maps. **IEEE Transactions on Fuzzy Systems**, v. 18, n. 1, p. 114-124, Feb. 2010.

MIAO, Y.; LIU, Z. Q. On causal inference in *Fuzzy* cognitive maps. **IEEE Transactions on Fuzzy Systems**, v. 8, n. 1, p. 107-119, 1999.

MIN, H. Q. et al. Probability *Fuzzy* cognitive map for decision-making in soccer robotics. PROCEEDINGS OF THE IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON INTELLIGENT AGENT TECHNOLOGY (IAT'06) 0-7695-2748-5/06, 2006.

MURPHY, R. **Introduction to AI robotics**. Cambridge: MIT Press, 2000.

NEVES JUNIOR, F.; ARRUDA, L. V. R.; MENDONÇA, M. A Combined FCM-GA approach to supervise industrial process. IFAC-SAFEPROCESS - 3rd INTERNATIONAL CONFERENCE ON SAFETY AND SECURITY ENGINEERING, Barcelona, 2009.

PACHECO, R. L. **Avaliação Formativa Continuada do Processo Educativo em Engenharia Usando Mapas Fuzzy**. 2005. 297 f. Tese (Doutorado em Engenharia Elétrica) - Universidade Federal de Santa Catarina, Florianópolis. 2005.

PADILHA, Paulo Cesar Carneiro. **Desenvolvimento de uma metodologia de sintonia de controladores “Fuzzy” utilizando redes neurais**. Aplicações em processos petroquímicos, 2001, Dissertação Mestrado. Instituto Militar de Engenharia, Rio de Janeiro 2001.

PAJARES, G.; DE LA CRUZ, J. M. *Fuzzy cognitive maps for stereovision matching* **Pattern Recognition**, v. 39, n. 11, p. 2101-2114, Nov. 2006.

PAPAGEORGIOU, E., STYLIOU, C.; GROUMPOS, P. A combined *Fuzzy* cognitive map and decision trees model for medical decision making. **Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society**, v. 1, p. 6117-6120, 2006.

PAPAGEORGIOU, E., STYLIOU, C.; GROUMPOS, P. Novel for supporting medical decision making of different data types based on *Fuzzy* Cognitive Map Framework. PROCEEDINGS OF THE 29TH ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE EMBS CITÉ INTERNATIONALE, Lyon, France August 23-26, 2007

PAPAGEORGIOU, E.; STYLIOU, C.; GROUMPOS, P. An Integrated Two-Level Hierarchical System for Decision Making in Radiation Therapy Based on *Fuzzy* Cognitive Maps. **IEEE Transactions on Biomedical Engineering**, v. 50, n. 12, p. 1326-1339, 2003.

PAPAGEORGIOU, E.I., PARASOPOULOS, K.E., STYLIOU, C.S., GROUMPOS, P.P., VRAHATIS, M.N. Fuzzy cognitive maps learning using particle swarm optimization. *Journal of Intelligent Information Systems* 25, 95–121, 2005.

PARK, K. S.; KIM, S. H. Fuzzy cognitive maps considering time relationships. **International Journal of Human-Computer Studies**, v. 42, n. 2, p. 157-168, Feb. 1995.

PARSOPOULOS, K. E. PAPAGEORGIOU E.I.; GROUMPOS P.P.; VRAHATIS M.N.. A first study of *Fuzzy* cognitive maps learning using particle swarm optimization. **Journal of International Information Systems**, v. 25, n. 1, p. 95-121, 2005.

PASSINO, M. K.; YOURKOVICH, S. **Fuzzy control**. Menlo Park: Addison-Wesley, 1997.

PELAEZ, C. E.; BOWLES, J. B. Using *Fuzzy* cognitive maps as a system model for failure models and effects analysis. **Information Sciences: Informatics and Computer Science Intelligent Systems Applications**, v. 88, n. 1/4, p. 177–199, Jan. 1996.

PERUSICH, K. Fuzzy cognitive maps for policy analysis. IEEE PURDUE UNIVERSITY SOUTH BEND, IN 46634 USA.1996.

PETALAS, Y. G.; PARSOPOULOS, K. E.; VRAHATIS, M. N. Improving *Fuzzy* cognitive maps learning through memetic particle swarm optimization, **Soft Computing**, v. 13, n.1, p. 77–94, 2009.

PIPE, A. G. An architecture for building “potential field” cognitive maps in mobile robot navigation. **Adaptive Behavior**, v. 8, n. 2, p. 173-203, 2000.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. Englewood Cliffs: Prentice Hall, 1995.

SADIQ, R.; KLEINER, Y.; RAJANI, B. Interpreting *Fuzzy* Cognitive Maps (FCMs) using *Fuzzy* measures to evaluate water quality failures in distribution networks. JOINT INTERNATIONAL CONFERENCE ON COMPUTATION IN CIVIL AND BUILDING ENGINEERING, Montreal, QC, June 14-16, 2006. p. 1-10.

SCHRAFF, R. D. Mechatronics and robotics for service applications. **IEEE Robotics and Automation Magazine**, v. 1, n. 4, p. 31-35, Dec. 1994.

SCHREIBER, G. T.; AKKERMANS, H. **Knowledge engineering and management: the commonKADS methodology**. Cambridge: MIT Press, 2000.

SELVATICI, A. H. P. **AAREACT**: uma arquitetura comportamental adaptativa para robôs móveis que integra visão, sonares e odometria. 2005. 101 f. Dissertação (Mestrado em Engenharia Elétrica) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

SERAPIÃO, A. B. S. Fundamentos de Otimização por Inteligência de Enxames: uma visão geral. Revista Controle & Automação/Vol.20 no. 3/Julho Agosto e Setembro 2009.

SHAW, I. S.; Simões, M. G. **Controle e Modelagem Fuzzy**. 1ª Edição Ed. Edgard Blucher LTDA, 1999.

SHI, Y., EBERHART, R.: A modified particle swarm optimizer. In: **IEEE International Conference on Evolutionary Computation Proceedings**, 1998. IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais: para engenharia e ciências aplicadas - curso prático. São Paulo: Artiber, 2010.

SIRAJ, A.; BRIDGES, S., VAUGHN, R. *Fuzzy cognitive maps for decision support in an intelligent intrusion detection system*. IFSA WORLD CONGRESS and 20th NAFIPS INTERNATIONAL CONFERENCE, 2001. Joint 9th.

SMART, W. D.; KAEHLING, L. P. Reinforcement learning for robot control. In: **MOBILE ROBOTS XVI PROCEEDINGS**, Boston, USA, Oct.28-Nov.2, 2001. **Proceedings of SPIE**, v. 4573, p.187-.194.

SOUZA, F. J. **Modelos neuro-fuzzy hierárquicos**. 1999. Tese (Doutorado), Departamento de Engenharia Elétrica – Pontifícia Universidade Católica, Rio de Janeiro, 1999.

SPARKFUNElectronics. Disponível em:

http://www.sparkfun.com/commerce/product_info.php?products_id=9211 Acesso em: 25 set. 2009.

STACH, W. Parallel Genetic Learning of the *Fuzzy Cognitive Maps*. Final Report for IEEE-CIS. Walter Karplus Summer Research Grant. 2006.

STACH, W. Kurgan L.; Pedrycz W; Reformat M. Genetic learning of *Fuzzy cognitive maps*, **Fuzzy Sets and Systems**, v. 153, n. 3, p. 371–401, Aug. 2005.

STACH, W.; Kurgan, L.; Pedrycz, W.; Reformat, M. Evolutionary Development of Fuzzy Cognitive Maps. **Fuzzy Systems, 2005. FUZZ '05**. The 14th IEEE International Conference on, vol., no., pp.619-624, 25-25 May 2005b.
doi: 10.1109/FUZZY.2005.1452465

STYLIOS, C. D., GEORGEPOULOS, V.C. MALANDRAKI, G.A., CHOULIARA S. Fuzzy cognitive map architectures for medical decision support systems. **Applied Soft Computing**, v. 8, n. 3, p. 1243-1251, Jun. 2008.

STYLIOS, C. D.; GEORGOPOULOS, V. C.; GROUMPOS, P. P Introducing the theory of *Fuzzy cognitive maps* in distributed systems. PROCEEDINGS OF THE 12TH IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, Istanbul, Turkey, 1997, pp. 55–60.

STYLIOS, C. D.; GEORGOPOULOS, V. C.; GROUMPOS, P. P. The use of Fuzzy cognitive maps in modeling systems. In: 5th IEEE MEDITERRANEAN CONFERENCE ON CONTROL AND SYSTEMS, PAPHOS, Cyprus, 21-23 July 1997.

STYLIOS, C. D.; GROUMPOS, P. P. A Soft computing approach for modeling the supervisor of manufacturing systems. **Journal of Intelligent and Robotic Systems**, v. 26, n. 3-4, p. 389-403, Nov. 1999b.

STYLIOS, C. D.; GROUMPOS, P. P. Fuzzy cognitive maps in modeling supervisory control systems. **Journal of Intelligent & Fuzzy Systems**, v. 8, n. 2, p. 83–98, 2000.

STYLIOS, C. D.; GROUMPOS, P. P. Fuzzy cognitive maps: a model for intelligent supervisory control systems. **Computers in Industry**, v. 39, n. 3, p. 229-238, 1999.

STYLIOS, C. D.; GROUMPOS, P. P. The challenge of modeling supervisory systems using fuzzy cognitive maps. **Journal of Intelligent Manufacturing**, 9, 339-345, 1998.
STYLIOS, C. D.; GROUMPOS, P. P: Modeling complex systems using fuzzy cognitive maps. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 34(1): 155-162, 2004

SUTTON, R.; BARTO, A. **Reinforcement learning**: an introduction. Cambridge: MIT Press, 1998.

TABER, R. *Fuzzy cognitive maps model social systems*. **AI Expert**, v. 9, p. 18-23, 1994.

TAI, P.; RYACIOTAKI-BOUSSALIS, H. A.; HOLLAWAY, D. Neural network implementation to control systems: a survey of algorithms and techniques. IEEE 1991.

TSADIRAS, A.; KONSTANTINOS, G. M. Cognitive mapping and certainty neuron *Fuzzy cognitive maps*. **Information Science**, v. 101, n. 1-2, p. 109-130, Sep. 1997.

YEAP, W. K.; WONG, C. K.; SCHMIDT, J. Using a mobile robot to test a theory of cognitive mapping. **Robotics and Cognitive Approaches to Spatial Mapping**, v. 38, p. 281-295, 2008.

YEN, G.; HICKEY, T. Reinforcement learning algorithms for robotic navigation in dynamic environments. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2002. v. 2, p. 1444-1449.

ZADEH, L.A. **Fuzzy algorithms**, Info. & Ctl. Vol. 12, pp. 94-102, 1968.

ZADEH, L. A. **An introduction to Fuzzy logic applications in intelligent systems**. Boston: Kluwer Academic Publisher, 1992.

ZHANG, J. Y.; LIU, Z. Q.; ZHOU, S. Dynamic domination in *Fuzzy causal network*. **IEEE Transactions on Fuzzy Systems**, v. 14, n. 1, p. 42-57, Feb. 2006.

ZHANG, H.; LIU, D. **Fuzzy Modeling and Fuzzy Control**. Control Engineering Birkhäuser Boston, 2006.

ZHU, W.; LEVINSON, S. Vision-based reinforcement learning for robot navigation. In INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, Washington DC., 2001.