

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

UMA ARQUITETURA HÍBRIDA PARA SISTEMAS COGNITIVOS E
SUA APLICAÇÃO EM PROGNÓSTICO DE FALHAS EM
TRANSFORMADORES DE POTÊNCIA

Fernando Cortez Sica

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais como requisito para a obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Prof. Frederico Gadelha Guimarães
Coorientador: Prof. Ricardo de Oliveira Duarte

Belo Horizonte – MG

Setembro de 2015

S565a

Sica, Fernando Cortez.

Uma arquitetura híbrida para sistemas cognitivos e sua aplicação em prognóstico de falhas em transformadores de potência [manuscrito] / Fernando Cortez Sica. - 2015.

118 f., enc.: il.

Orientador: Frederico Gadelha Guimarães.

Coorientador: Ricardo de Oliveira Duarte.

Tese (doutorado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Anexos: f.95-103.

Bibliografia: f.105-118.

1. Engenharia elétrica - Teses. 2. Transformadores elétricos - Teses. 3. Localização de falhas (Engenharia) - Teses. I. Guimarães, Frederico Gadelha. II. Duarte, Ricardo de Oliveira. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

TESE DE DOUTORADO Nº 213

**UMA ARQUITETURA HÍBRIDA PARA SISTEMAS COGNITIVOS E
SUA APLICAÇÃO EM PROGNÓSTICO DE FALHAS EM
TRANSFORMADORES DE POTÊNCIA**

Fernando Cortez Sica

DATA DA DEFESA: 04/09/2015

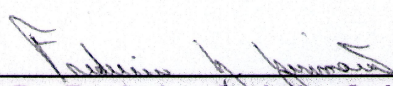
**"Uma Arquitetura Híbrida para Sistemas Cognitivos e sua
Aplicação em Prognóstico de Falhas em Transformadores de
Potência"**

Fernando Cortez Sica

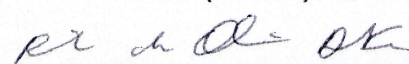
Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 04 de setembro de 2015.

Por:



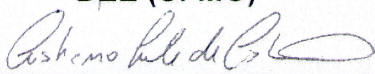
Prof. Dr. Frederico Gadelha Guimarães
DEE (UFMG) - Orientador



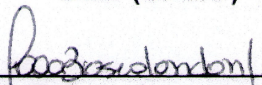
Prof. Dr. Ricardo de Oliveira Duarte
DELT (UFMG)



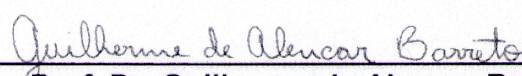
Prof. Dr. João Antônio de Vasconcelos
DEE (UFMG)



Prof. Dr. Cristiano Leite de Castro
DEE (UFMG)



Prof. Dr. João Bosco Augusto London Junior
(USP-SC)



Prof. Dr. Guilherme de Alencar Barreto
DETI (UFC)



Prof. Dr. Agnaldo José da Rocha Reis
DECAT (UFOP)

Agradecimentos

Agradeço aos meus pais e irmã pela compreensão frente ao meu distanciamento durante o período de estudos no doutoramento. Aos meus orientadores Frederico Gadelha e Ricardo Duarte por todo incentivo, direcionamentos e ensinamentos. Ao Agnaldo Reis pelo apoio e ajuda na reta final. Aos parceiros do Departamento de Computação da Universidade Federal de Ouro Preto por me apoiarem na realização do doutoramento.

Resumo

Esta tese relata a proposta de uma arquitetura para facilitar a implementação de sistemas genéricos baseados em conhecimento. A arquitetura utiliza conceitos de modelos embrionário, conexionista e evolucionário. Além dos modelos mencionados, a arquitetura também emprega o uso de modularização baseada em agentes de modo a permitir a flexibilidade e granularidade necessárias à construção de sistemas genéricos baseados em conhecimento. A abordagem conexionista emprega aprendizado em conjunto (*ensemble learning*) pelo seu potencial para se atingir maior precisão e robustez. Aprendizagem em conjunto pode integrar diferentes classificadores baseados em aprendizado de máquina, que estão disponíveis na literatura e, também, baseados em métodos estatísticos. Por sua vez, o modelo evolucionário permite representar a dinâmica relativa à abstração, criação e manipulação do conhecimento. Como maior contribuição do projeto relacionado à tese pode-se citar a implementação de um sistema capaz de gerar prognóstico de falhas em transformadores de potência.

Palavras-chave:Sistemas Cognitivos, Prognósticos de Falhas, Transformadores de Potência

Abstract

This thesis reports the proposal of an architecture to facilitate the implementation of knowledge-based systems. The architecture utilizes concepts from embryonic, connectionist and evolutionary approaches. In addition to the mentioned models, the architecture also employs the use of agent-based modularization to allow flexibility and granularity required to construct generic knowledge-based systems. The connectionist approach employs ensemble learning for its potential in achieving higher accuracy and robustness. Ensemble learning can combine together different classifiers based on machine learning that are available in the literature and statistical methods. In turn, the evolutionary model allows to represent the knowledge dynamics concerning the abstraction, creation and handling of knowledge. As a major contribution, we can cite the implementation of a system capable of generating fault prognosis in power transformers.

Keywords:Cognitive Systems, Fault Prognosis, Power Transformers

Sumário

1	Introdução	21
1.1	Conceitos Relacionados a Conhecimento	21
1.2	Motivações	24
1.3	Objetivos	25
1.4	Pontos Relevantes	25
1.5	Contribuições	28
1.6	Estruturação da Tese	29
2	Modelos Computacionais para Sistemas Inteligentes e para a Representação do Conhecimento	31
2.1	Modelos de Sistemas Cognitivos	32
2.2	Exemplos de Arquiteturas de Sistemas Inteligentes	37
2.3	Representação do Conhecimento	43
2.3.1	Lógica Matemática	43
2.3.2	Regras de Produção	44
2.3.3	<i>Frames</i>	44
2.3.4	Grafos Conceituais	45
2.3.5	Ontologias	46
2.3.6	Redes Bayesianas	47
2.4	Sistemas Evolucionários	48
2.5	Modelagem e Implementação	51
2.5.1	Sistema Sensório (<i>Sensory System</i>)	53

2.5.2	Sistema de Armazenamento, Codificação e Processamento (<i>SCP System</i>)	55
2.5.3	Sistema de Atuação (<i>Actuation System</i>)	57
2.5.4	Sistema de Comunicação (<i>Communication System</i>)	57
2.6	Implementação em Software	57
2.7	Conclusões	58
3	Prognósticos	61
3.1	Conceitos de Prognósticos	61
3.2	Classificação dos Sistemas de Prognósticos	62
3.3	Aplicações dos Sistemas de Prognósticos	66
3.4	Precisão e Acurácia dos Prognósticos	67
3.5	Conclusões	68
4	Prova de Conceito: Aplicação em Prognóstico de Falhas em Transformadores de Potência	69
4.1	Implementação do Sistema de Prognósticos de Falhas de Transformadores de Potência	73
4.1.1	Modelo de Regressão para a Variação dos Gases	77
4.1.2	Modelo Conexcionista Utilizado	78
4.1.3	Modelo evolucionário utilizado	79
4.2	Resultados	83
4.2.1	Dados experimentais	83
4.2.2	Configuração dos Experimentos	84
4.2.3	Resultados	84
4.3	Conclusões	88
5	Conclusões e Trabalhos Futuros	91
A	Aspectos da Implementação do Sistema de Prognósticos de Falhas de Transformadores de Potência em Hardware	95
A.1	Computação Reconfigurável	95

A.2	<i>System-on-Chip</i> (SoC)	99
A.3	Proposta de Implementação em Hardware	100
A.4	Conclusões	102
Referências Bibliográficas		105

Lista de Figuras

1-1	Abstrações das categorizações de Peirce no âmbito de sistemas computacionais	28
2-1	Representação genérica para um processo cognitivo	31
2-2	Modelo computacional de sistema cognitivo extraído de (Minsky, 1961)	32
2-3	Modelo Cognitivo proposto por (Atkinson & Shiffrin, 1971)	33
2-4	Modelo proposto por (Baddeley & Hitch, 1974)	33
2-5	Arquitetura proposta por (Laberge & Brown, 1989)	34
2-6	Camadas de competência da arquitetura proposta por Brooks - extraída de (Brooks, 1999)	36
2-7	Arquitetura proposta por Albus - extraída de (Albus <i>et al.</i> , 1989) . .	36
2-8	Arquitetura do sistema cognitivo apresentado por (Franklin, 1997) . .	38
2-9	Arquitetura sugerida por (Yoon <i>et al.</i> , 2000)	39
2-10	Estruturação em camadas do sistema cognitivo proposto em (Isla <i>et al.</i> , 2001)	40
2-11	Sistema cognitivo levando-se em conta o fator emocional proposto em (Imbert & de Antonio, 2005)	41
2-12	Arquitetura do IVA (<i>Intelligent Virtual Agent</i>) proposta em (Arnellos <i>et al.</i> , 2008)	41
2-13	Exemplificação do uso de <i>frames</i>	45
2-14	Arquitetura básica do sistema modelado	52
2-15	Estruturação funcional de cada agente	53
2-16	Estruturação funcional do sistema sensório	54

2-17	Estruturação do <i>Symbolic Layer</i>	56
2-18	Representação UML das classes implementadas da arquitetura proposta	58
3-1	Índice de saúde do componente em função do tempo (Figura extraída de (Xiongzi <i>et al.</i> , 2011))	62
3-2	Classificação dos sistemas de prognósticos (Figura extraída de (Okoh <i>et al.</i> , 2014))	64
3-3	Diferenciação entre as métricas Precisão e Acurácia (Figura extraída de (Uckun <i>et al.</i> , 2008))	67
4-1	Método do triângulo de Duval	71
4-2	Etapas para a geração dos prognósticos	75
4-3	Instanciação dos módulos da arquitetura proposta na implementação do sistema de prognósticos	76
4-4	Variações das concentrações dos gases em função do tempo	77
4-5	<i>Box plot</i> do RUL encontrados nas 120 execuções independentes.	85
4-6	Histograma e curva distribuição gama do RUL para o Caso de Teste 1 frente às 120 execuções independentes.	86
4-7	Histograma e curva de distribuição gama do RUL para o Caso de Teste 2 frente às 120 execuções independentes.	86
4-8	Acurácia do sistema de prognósticos em relação ao tempo transcorrido entre o estado inicial e o estado de falha.	88
A-1	Comparação entre processadores, <i>hardware</i> reconfigurável e <i>ASICs</i> em termos de flexibilidade e performance	96
A-2	Esquemas de Reconfiguração	98
A-3	Proposta de implementação em hardware	100
A-4	<i>Layout</i> da implementação em hardware	101

Lista de Tabelas

2.1	Métodos das classes (<i>threads</i>) criadas	59
4.1	Falhas descritas pelo triângulo de Duval (DTM)	72
4.2	Funcionalidades incorporadas pelos módulos da arquitetura	76
4.3	Casos de testes	83
4.4	Ambiente de Teste	84
4.5	Matriz de Confusão	87
A.1	Funcionalidades dos componentes do SoC proposto	101
A.2	Pinagem da proposta de implementação em hardware	102

Capítulo 1

Introdução

1.1 Conceitos Relacionados a Conhecimento

Inicialmente, para discorrer sobre o tema de sistemas baseados em conhecimento, deve-se discorrer sobre o que vem a ser conhecimento no tangente à sua conceituação e formas de representação e de categorização. Os conceitos aqui apresentados serão utilizados no decorrer do texto através de relacionamentos e inspirações entre o viés das ciências biológicas e das ciências humanas e o viés computacional.

A representação de conhecimento assim como suas formas de aquisição e seus assuntos correlatos, como aqueles relacionados aos mecanismos cognitivos, sempre se fizeram presentes no cotidiano de diversos setores da sociedade, inclusive a científica. O conhecimento pode ser tratado sob diversas ópticas, desde a sua representação (mais especificamente a sua modelagem), passando pelos mecanismos de transmissão e propagação até, por último, seus mecanismos cognitivos e de processamento. Estes tratamentos permitem que o conhecimento possa melhor se agregar às novas realidades.

Segundo o Dicionário de Ontopsicologia, de Antônio Maneghetti¹:

- Etimologia

¹Antônio Meneghetti, Dicionário de Ontopsicologia 2 ed. rev. Recanto Maestro: Ontopsicologica Editrice, 2008. ISBN 978-85-88381-41-4

- *cognoscere* (latim) = vir a saber
- *cum* (latim) + *vous* (grego) + *scit actionem* (latim) = quando ou como a mente sabe a ação. Saber a ação
- Significado
 - Conhecimento ontológico: quando a mente reflete a ação do real ou do ser; reflexão e ato coincidem de forma a consentir-lhe a reversibilidade.
 - Conhecimento opinativo: quando o sujeito crê na reversibilidade entre imagem e ato ou objeto.

Diante das definições acima colocadas, pode-se mencionar que, sob o viés computacional, um sistema que manipula conhecimento deve ser capaz de representá-lo e inferir em relação ao significado dentro do contexto do processamento. Deve, também, ser capaz de reagir sobre estímulos externos, podendo modificar as estruturas de conhecimento previamente obtidas.

A manipulação de conhecimento não é privilégio somente da área computacional e também não é privilégio da sociedade contemporânea. Assuntos acerca de como o conhecimento é abstraído, transformado e transmitido fazem parte, desde tempos remotos, do rol das discussões e investigações em diversas esferas da sociedade, a qual se inclui, também, a sociedade científica, como mencionado em (Ghazanfar & Turesson, 2008).

Diversas áreas tentam desvendar e explicar o funcionamento do cérebro e suas estruturas mentais para representar e manipular conhecimento. Como exemplo destas áreas, tem-se: a neurociência, psicologia cognitiva, linguística e antropologia. Inúmeros, também, são os trabalhos dentro das áreas da informática e da engenharia que foram realizados na tentativa de construir um cérebro artificial, atribuindo-lhe ou não tratamento de emoções, como por exemplo os trabalhos descritos em (Nadel *et al.*, 2006; Dubal *et al.*, 2010).

Tratando-se de modelagem de sistemas mentais artificiais, deve-se incorporar três linhas de estudos extraídas das observações e estudos realizados sobre os sistemas biológicos naturais: filogenética, ontogenética e epigenética (Sanchez *et al.*, 1997).

De forma sucinta, pode-se descrever que a filogenética relaciona-se à evolução genética natural dentro dos indivíduos e espécies. A ontogenética volta-se ao processo desenvolvimentista de um indivíduo multicelular. Por último, a epigenética inclui o processo de aprendizagem de um indivíduo durante o seu ciclo de vida (Upequi *et al.*, 2007; Canham & Tyrrell, 2003a; Canham & Tyrrell, 2003b).

Dadas as noções básicas, é possível, então, realizar um mapeamento direto entre a filogenética, ontogenética e epigenética frente à embriogenética² e a evolutividade dos sistemas biológicos artificiais. Esta correlação pode ser explicada pelo fato de que a embriogenética pode ser entendida como a divisão e diferenciação celulares e demais propriedades e ações aplicadas sobre a representação de um conjunto de estruturas lógicas computacionais (Zhang *et al.*, 2004; Thoma *et al.*, 2004). Por outro lado, o fator evolucionário garante a especialização de indivíduos (os quais podem ser componentes de *hardware*) através de geração de novas populações pela aplicação de operadores genéticos. A geração de nova população é norteadada pelos estímulos ambientais externos incididos sobre os indivíduos da população corrente (G. Husi *et al.*, 2010).

Biologicamente falando, um indivíduo (ou organismo³ pluricelular), é um conjunto de sistemas de órgãos que, por sua vez, são formados por tecidos e células especializadas.

Baseando-se, então, na concepção biológica de um indivíduo, aparecem dois conceitos importantes para a modelagem e implementação de sistemas computacionais baseados em sistemas biológicos: a interação entre os componentes de cada nível hierárquico que compõem um indivíduo (interação célula-célula; interação órgão-órgão) e a especialização das células para atender às diversas funcionalidades a elas cabíveis.

Os conceitos de interação e especialização estão presentes nos sistemas de hardware baseados em sistemas biológicos, estes também conhecidos através do termo *Evolvable Hardware* (*EHW*), instituído em (Higuchi, 1993). Segundo o autor, a diferença básica consiste no fato de que um *EHW* é baseado na evolução e interação de organismos

²Embriogenética refere-se ao processo de formação e desenvolvimento de um embrião. Compreende as fases de fertilização do óvulo, divisões mitóticas e formação do embrião.

³do grego *organismós* significando conjunto

simples e autônomos conectados a apenas organismos vizinhos. Em resumo, pode-se citar os principais pontos defendidos:

- A interação entre um vasto número de unidades independentes e autônomas determinam o comportamento total do circuito: cada unidade (podendo ser referenciada como agente) é responsável por processar uma parte do objetivo do processamento;
- adaptação ao ambiente: tratando-se *EHW*, o sistema deve ser capaz de se adaptar frente às variações do ambiente no qual encontra-se inserido. Tais adaptações implicam nos resultados de inferência sobre o conhecimento incorporado;
- auto-replicação: para permitir uma maior eficiência computacional, as unidades poderão ser auto-replicadas de modo que o processamento possa ser distribuído pelas novas unidades;
- auto-reparação: em caso de falhas de uma unidade, poderá haver a sua auto-reparação ou correção de modo a permitir a continuidade ou a reiniciação de seu processamento.

1.2 Motivações

Como principais motivações para o desenvolvimento deste trabalho, pode-se enumerar as seguintes:

- A ausência de arquiteturas para a construção de sistemas genéricos baseados em conhecimento;
- a possibilidade de proporcionar uma maior aproximação entre os conceitos idealizados de conhecimento aos sistemas computacionais;
- a possibilidade de integrar técnicas evolucionárias e conexionistas para a manipulação do conhecimento;

- a ausência de ferramentas para a geração de prognósticos de falhas em transformadores de potência;
- a possibilidade de ampliar o uso da arquitetura do sistema para novas áreas em ampla expansão, como por exemplo, sistemas de gerenciamento de saúde de sistemas.

1.3 Objetivos

Os conceitos previamente introduzidos vêm ao encontro do propósito desta tese: desenvolvimento de uma arquitetura genérica dotada de mecanismos evolucionários e embriônicos para que possam ser implementados sistemas baseados em conhecimento. Para tanto, são utilizados serviços e facilidades baseados em paradigmas embriônicos e evolucionários em um cenário de sistemas multiagentes. Como o modelo adotado é genérico, pode ser utilizado em diversas aplicações que requeiram a manipulação de conhecimento para tomada de decisões, como por exemplo, sistemas de controle, mineração de dados, análise biomédica e sistemas de diagnóstico e de telemetria. Estas aplicações demandam sistemas com a habilidade de aprender, de forma contínua, novas regras e estruturas inerentes ao conhecimento sob manipulação.

Pelo fato da arquitetura aqui proposta permitir a implementação de aplicações genéricas, emprega-se, ainda, uma integração de representação simbólica e conexionista de conhecimento para facilitar a representação e manipulação de conhecimento em seus diversos aspectos e campos de atuação.

Para validar a arquitetura, o projeto teve, como segundo objetivo, a implementação de um sistema de geração de prognósticos em transformadores de potência como prova de conceito.

1.4 Pontos Relevantes

Antes de pormenorizar as contribuições desta Tese, pode-se ressaltar alguns pontos relevantes concretizados durante o desenvolvimento da arquitetura proposta e da

aplicação implementada como prova de conceito:

- Uso, de forma integrada, de mecanismos embriônicos e evolucionários;
- adaptação e modelagem do uso do conceito de agentes inteligentes computacionais autônomos. A utilização de agentes inteligentes permitiu uma melhor abstração das estruturas inerentes ao conhecimento assim como os seus inter e intra-relacionamentos para permitir, desta forma, a representação e abstração de conhecimento;
- modelagem do conhecimento a partir da estruturação e agrupamentos de símbolos básicos. A partir desta representação, permite-se uma melhor manipulação sob o aspecto computacional uma vez que podem ser utilizados os conceitos de heranças, especializações e agrupamentos;
- modelagem de informações temporais de modo que uma sequência de ativação de símbolos pode representar eventos a serem tratados pelo sistema.

Sob o ponto de vista de modelagem, para permitir a utilização dos conceitos de sistemas embriônicos, sistemas evolucionários e agentes inteligentes, foram desenvolvidos:

- Mecanismos de representação de conhecimento;
- classes de especialização de estruturas que representarão os símbolos básicos e isolados do conhecimento;
- mecanismos de estímulo-resposta diante de sinais externos atuando sobre o conhecimento armazenado;
- mecanismos baseados em computação natural para representar a auto-adaptação e evolução das estruturas criadas;
- mecanismos de mapeamento do conhecimento propriamente dito.

Uma vez havendo a necessidade de representar conhecimento de forma a cobrir uma ampla gama de aplicações, convém, também, abordar de forma sucinta, alguns conceitos de representação de conhecimento através do co-relacionamento e inter-relacionamento de símbolos (ou signos).

Um símbolo atuando de forma isolada pode não representar um conhecimento no sentido amplo da palavra. Para tanto, pode-se inferir sobre as teorias de Peirce⁴ no que se trata das ações dos signos e suas categorizações. As categorizações são decorrentes das ideias levantadas por Kant⁵ em sua obra intitulada "Crítica da Razão Pura". Segundo Kant, para abstração de um conhecimento ou objeto, incidem, de maneira incisiva, a *Intuição* (abstração imediata após o contato com o objeto, ainda sem organização coerente), a *Sensibilidade* (recebimento da representação segundo como o objeto afeta um indivíduo) e o *Pensamento* (fenômeno que organiza, de forma coerente, o conjunto de sensações).

Derivado dos conceitos kantianos, Peirce elaborou três categorizações para a organização dos símbolos na geração do conhecimento: *Primeiridade*, *Secundidade* e *Terceiridade*. Segundo (Salatie, 2006; Morato, 2012), pode-se conceituar as categorizações como:

- *Primeiridade*: fenômenos sem representação complexa, a abstração do objeto não constitui-se em um signo propriamente dito e, sim, sensações vagas, formando os quali-signos;
- *Secundidade*: sensação provocada pela interação entre o estímulo externo e a consciência. Relaciona-se com a sensibilidade de Kant;
- *Terceiridade*: formação do pensamento através de signos, organização complexa da abstração do objeto.

Para um melhor detalhamento, a Figura 1-1 traduz as categorizações de Peirce para a óptica dos sistemas computacionais.

⁴Charles Sanders Peirce (1839-1914), filósofo dos Estados Unidos que estudou, dentre outros itens, a “ação dos signos”.

⁵Immanuel Kant (1724-1804), pensador alemão que influenciou a teoria semiótica.

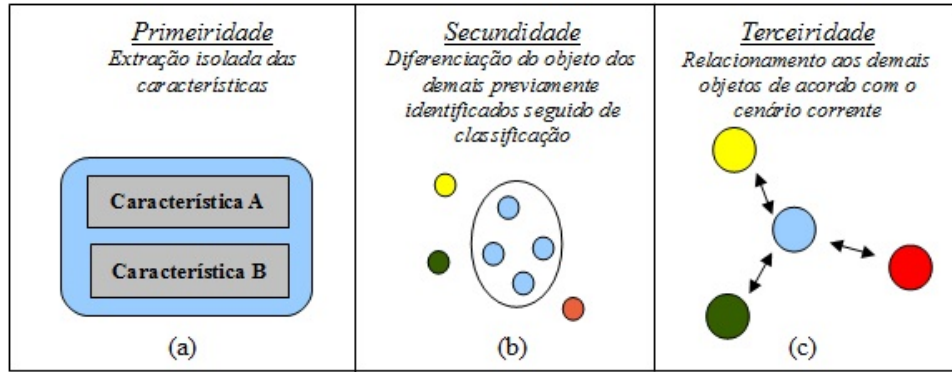


Figura 1-1: Abstrações das categorizações de Peirce no âmbito de sistemas computacionais

Na Figura 1-1 tem-se, em (a), o sensoriamento coletando um objeto (por exemplo, uma imagem) e identificando suas características básicas. Neste ponto, ainda não se tem a real percepção do objeto, ou seja, apenas se conhece suas características como formato e cor. A partir das características básicas tem-se, em (b), a categorização do objeto, identificando-o e classificando-o de acordo com regras e mecanismos do sistema. Neste ponto, o objeto já passa a ser um *signo*. Um signo isolado não tem muita valia em sistemas baseados em conhecimento. Um signo deve ser conectado a outros de modo a obter um raciocínio ou um estado corrente de conhecimento. Esta etapa final é representada em (c).

Sendo assim, a partir das categorizações de Peirce para que os conhecimentos possam ser abstraídos e manipulados, surge a necessidade de incorporar, no sistema computacional, estruturas e mecanismos necessários para a representação do conhecimento e para os processos cognitivos.

1.5 Contribuições

Em decorrência de todos os pontos até aqui abordados, pode-se mencionar que este trabalho contribuiu cientificamente pelo fato de propor modelos e mecanismos suficientes e autônomos para o desenvolvimento de aplicações cognitivas utilizando-se sistemas embrionários e evolucionários. Em visão mais ampla, podem ser citadas como contribuições iniciais:

- contextualização da aplicação de sistemas evolucionários e embriônicos no âmbito de sistemas baseados em conhecimento;
- aplicação de técnicas de *software* conexionistas, evolucionária e embriônica para o mapeamento e manipulação de conhecimento.

Como principal contribuição desta tese, pode-se mencionar o desenvolvimento de um novo sistema capaz de realizar prognósticos de falhas em transformadores de potência. Esta nova abordagem, baseada em um sistema híbrido que integra mecanismos conexionistas e evolucionários, é capaz de realizar prognósticos possíveis falhas que, porventura, incidirão sobre os transformadores de potência. Tais prognósticos podem auxiliar segmentos como o representado pelas concessionárias de energia elétrica (CEE), no tocante ao refinamento de seus planejamentos de manutenção e trocas de equipamentos. Isso proporciona, conseqüentemente, a melhoria da qualidade de fornecimento da energia elétrica e diminuição dos custos para a concessionária e para o consumidor final.

1.6 Estruturação da Tese

Para poder discorrer sobre os temas abrangidos neste projeto, este trabalho apresenta, no capítulo 2, conceitos sobre sistemas inteligentes e representação de conhecimento para que seja apresentada, no mesmo capítulo, a proposta de uma arquitetura genérica baseada em modelo que integra as abordagens conexionista e evolucionária que permitirá a implementação de sistemas baseados em conhecimento para atender a diversas áreas de atuação.

Uma vez que o sistema implementado para validar a arquitetura modelada objetiva a geração de prognósticos de falhas em transformadores de potência, são apresentados, no Capítulo 3 conceitos relacionados à área de prognósticos de modo a propiciar um melhor entendimento sobre a área de falhas de transformadores. O Capítulo 4 aborda a implementação do sistema de geração de prognósticos desenvolvido como prova de conceito.

Por fim, o Capítulo 5 objetiva a menção das conclusões do referido projeto no tocante à modelagem da arquitetura assim como a sua utilização na implementação do sistema de prognóstico em transformadores de potência. O Capítulo 5 menciona, ainda, os trabalhos futuros a serem idealizados a partir do conhecimento obtido durante o desenvolvimento desta tese.

No Anexo A, encontra-se a descrição da proposta para a implementação, em hardware, do sistema desenvolvido apresentado no Capítulo 4.

Capítulo 2

Modelos Computacionais para Sistemas Inteligentes e para a Representação do Conhecimento

Ao se falar sobre modelos computacionais cognitivos (ou inteligentes), um dos primeiros passos consiste na descrição e modelagem do conhecimento e, conseqüentemente, em sua forma de armazenamento. Sendo assim, o objetivo deste capítulo consiste em delinear aspectos importantes para a modelagem não somente de sistemas cognitivos mas, também, a forma de estruturação e armazenamento das informações que representam o conhecimento a ser manipulado.

Inicialmente, é interessante mencionar que o processo cognitivo apresenta um comportamento dinâmico. Sua dinamicidade referencia-se à própria interação com o ambiente: coletando informações, codificando-as, armazenando-as e processando-as para que, depois, possa haver a atuação sobre o próprio ambiente (Figura 2-1).

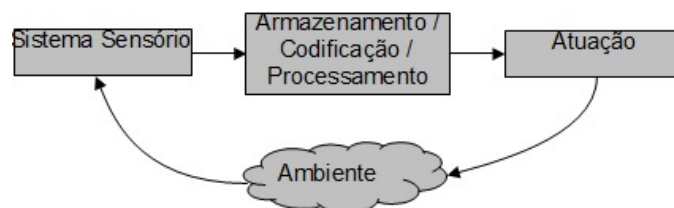


Figura 2-1: Representação genérica para um processo cognitivo

2.1 Modelos de Sistemas Cognitivos

Para iniciar a presente seção, não se pode omitir um trabalho realizado por um dos idealizadores da Inteligência Artificial: Marvin Minsky. O artigo (Minsky, 1961) explicita, dentre de vários pontos importantes ao campo da Inteligência Artificial, o modelo apresentado na Figura 2-2.

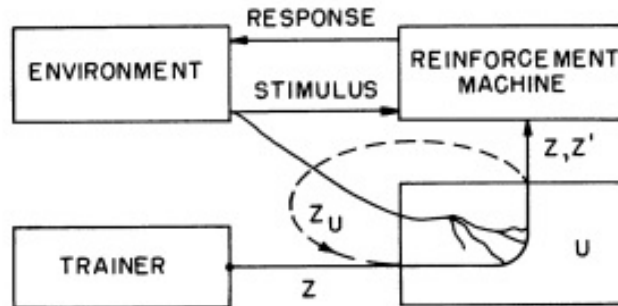


Figura 2-2: Modelo computacional de sistema cognitivo extraído de (Minsky, 1961)

No modelo da Figura 2-2, nota-se a presença do módulo U que é um dispositivo que auxilia na aprendizagem pelo fato de co-relacionar os sinais dos estímulos externos e os sinais de reforço. Segundo Minsky, o dispositivo U atua no processo de aprendizagem como um condicionador pavloviano. O termo pavloviano é associado ao processo comportamental (behaviorismo) onde a modificação de comportamento é baseado no binômio estímulo-resposta.

Um outro trabalho altamente relevante em relação ao pioneirismo é descrito em (Atkinson & Shiffrin, 1971). Nele, retrata-se uma das primeiras pesquisas inerentes à modelagem computacional de sistemas de cognitivos e de memória. No referido trabalho, é descrito um sistema de memória dotado de hierarquização para o armazenamento das informações, caracterizadas como informações sensoriais, de curta e longa duração conforme ilustrado na Figura 2-3.

Ainda em relação à Figura 2-3, os registradores permitem capturar e armazenar informações advindas do meio externo. Por sua vez, o controle de processos embute os mecanismos necessários para as estratégias de recuperação de informações, decisões e codificação de informações.

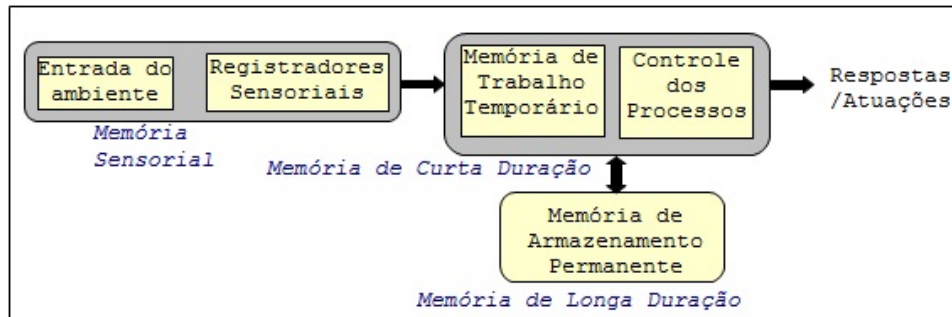


Figura 2-3: Modelo Cognitivo proposto por (Atkinson & Shiffrin, 1971)

Algumas adaptações e variações foram apresentadas ou incorporadas a esta concepção inicial, como as descritas em (Baddeley & Hitch, 1974). Estes autores propuseram um modelo de memória operacional contendo um módulo centralizador (denominado como executivo central) auxiliado por dois subsistemas para o armazenamento temporário no contexto fonológico ou visual (conforme ilustra a Figura 2-4).

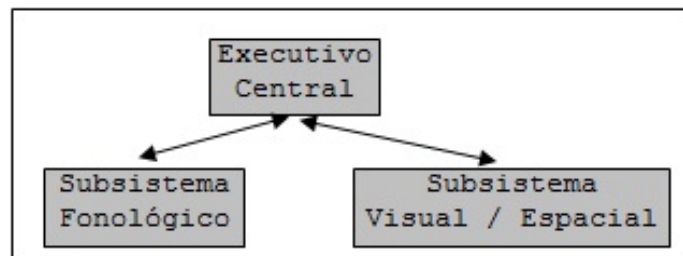


Figura 2-4: Modelo proposto por (Baddeley & Hitch, 1974)

Neste modelo, não há uma hierarquia de propagação de informações entre as memórias de curta e longa duração. Sendo assim, a informação poderia propagar-se diretamente para a memória de longa duração independentemente de seu tempo de permanência na memória de curta duração.

Outros modelos foram propostos como fator de oposição ou de complementação ao modelo de memória. Dentre estes, pode-se citar os modelos onde atributos baseados em atenção influenciam a absorção, armazenamento e manipulação de informações do conhecimento.

Sistemas dotados de filtros são capazes de delimitar, atenuar ou amplificar os sinais a serem inferidos e manipulados pelo sistema cognitivo. Sobre os filtros são aplicados atributos acerca de seu funcionamento em decorrência dos aspectos emocionais,

bagagem de conhecimento prévio e importância dos estímulos naquele momento de tempo.

O modelo proposto por (Laberge & Brown, 1989) explicita a função do filtro tomando-se como exemplo a identificação de formas. Neste trabalho, cuja arquitetura é exibida na Figura 2-5, explica a função do filtro como elemento que atuará sobre os dados fornecidos pelos registradores de entrada (com informações obtidas do meio externo) e norteado pelos critérios apresentados pelo analisador. A função do analisador consiste em abstrair as condições atuais de relevância das informações a serem, efetivamente, processadas.

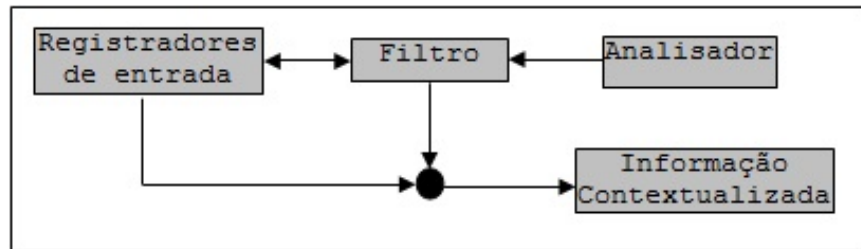


Figura 2-5: Arquitetura proposta por (Laberge & Brown, 1989)

Independente do modelo, nota-se uma categorização de memória podendo esta estar presente ao nível dos registradores de entrada, memória de trabalho (ou de curta duração) e memória de longa duração que terá, efetivamente, o conhecimento aprendido.

O conhecimento aprendido é impactado pelo conhecimento prévio, grau de atenção ou motivação do indivíduo frente aos estímulos externos e grau de importância da própria informação.

A ideia de filtro também é utilizada nas concepções atuais de modelagem em engenharia, como por exemplo, o modelo proposto por (Kogler Junior & Inojosa Filho, 2010). Neste trabalho, baseado na utilização de autômatos adaptativos, os filtros selecionam informações sensoriais relevantes para que sejam gerados símbolos a serem incorporados ao autômato corrente. Ainda, segundo o autor, os símbolos introduzidos geram autômatos capazes de reconhecê-los (denominados como percepção) e passarão a ser elementos de entrada do sistema cognitivo. Por sua vez, o sistema

cognitivo incorporará a percepção ao seu autômato principal, incidindo e alterando o comportamento dos agentes cognitivos presentes no sistema.

Além dos modelos citados previamente, não se deve deixar de mencionar o modelo proposto por Brooks (Brooks, 1986). Trata-se na arquitetura intitulada de *arquitetura de subsunção*, baseada no modelo reativo e criada para ser utilizada como mecanismo de controle de robôs autônomos. A principal motivação de um modelo reativo consiste na agilidade de resposta do sistema frente aos eventos externos em função da não utilização de um modelo de representação interna do mundo. Segundo o próprio Brooks (Brooks, 1991), o mundo é a melhor forma de representação dele próprio.

Uma arquitetura reativa faz uso de mapeamentos entre a informação sensorial e a ação ou resposta (comportamento reativo). Além do mapeamento, há a definição de da definição da coordenação dos eventos associados aos pares estímulos-respostas. Especificamente na arquitetura de subsunção, a coordenação é feita de forma competitiva com supressão e inibição, ou seja, dentro de um conjunto de possíveis reações, apenas uma ação é executada de acordo com prioridades estabelecidas.

Na arquitetura de subsunção, os comportamentos reativos são organizados em módulos interconectados formando, assim, uma rede organizada em camadas de competências. As camadas superiores, representando tarefas associadas ao cumprimento de um objetivo propriamente dito, possuem prioridades sobre as camadas de níveis inferiores, as quais abstraem comportamentos frente às ações básicas. Devido ao comportamento competitivo, uma camada superior poderá efetuar a supressão do comportamento da camada inferior, deixando-o ainda ativo para futuras decisões, ou poderá efetuar a inibição, desabilitando por completo o comportamento da camada inferior.

A Figura 2-6 ilustra melhor a ideia da utilização das camadas de competência.

Nota-se, na Figura 2-6, que os comportamentos reativos foram classificados em camadas de competências. As camadas superiores integram os comportamentos mais próximos aos objetivos globais do robô e tais objetivos são auxiliados pelos comportamentos de ordem mais básicas descritos nas camadas inferiores.

Tratando-se em organização em camadas de competência, pode-se citar, por fim,

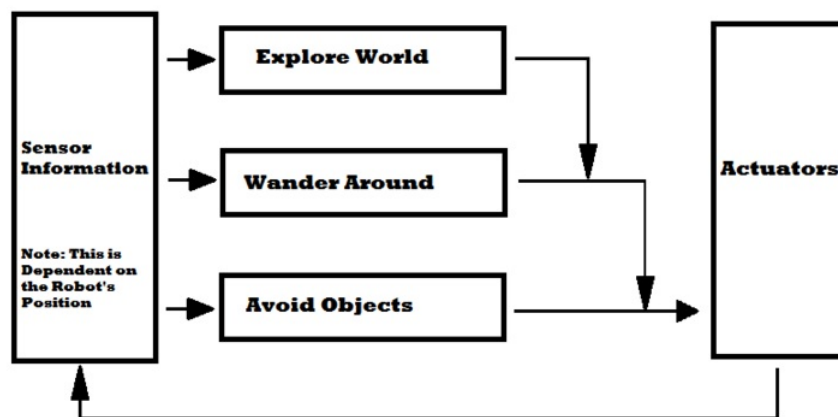


Figura 2-6: Camadas de competência da arquitetura proposta por Brooks - extraída de (Brooks, 1999)

a arquitetura intitulada como *NASREM* desenvolvida por Albus (Albus *et al.*, 1989). Nessa arquitetura, assim como a arquitetura de Brooks, as camadas superiores são relacionadas ao escopo de mais alto nível de abstração (relacionada, por exemplo, ao objetivo global do processamento). Por sua vez, as camadas inferiores estão intimamente ligadas aos dispositivos sensores e atuadores (Figura 2-7).

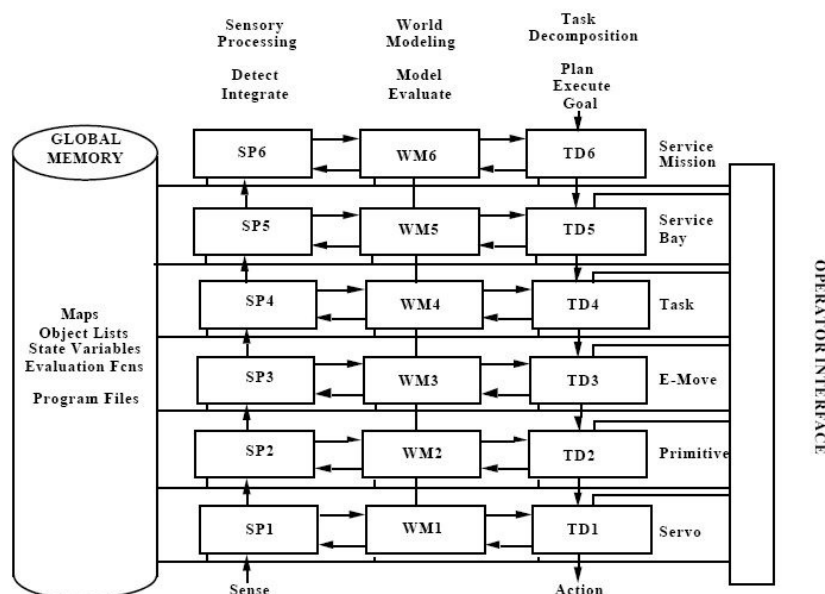


Figura 2-7: Arquitetura proposta por Albus - extraída de (Albus *et al.*, 1989)

Na Figura 2-7, nota-se que cada camada é dividida em três visões: processamento

sensorio, modelo do mundo e decomposição da tarefa. Essas três visões são capazes de, a partir das entradas sensoriais, realizar a decomposição e abstrações necessárias de acordo com os modelos que integram a visão intermediária para que, finalmente, possa ser realizada a decomposição das tarefas e seus respectivos planejamento e execução. Cada abstração inferida pela camada de representação do mundo é feita uma avaliação de riscos da ação sobre os estados observados e os possíveis e hipotéticos estados passíveis de serem assumidos pela execução das ações.

2.2 Exemplos de Arquiteturas de Sistemas Inteligentes

Vários são os projetos que objetivam modelar e implementar sistemas inteligentes. Para uma melhor visão e entendimento das características básicas associadas aos sistemas inteligentes, são apresentados cinco projetos distintos, elencando-os em ordem cronológica crescente.

De acordo com (Franklin, 1997), são citados vários elementos necessários para se modelar e implementar um sistema inteligente baseado na concepção de multiagentes:

- Um sistema computacional pode ter algoritmos inteligentes, porém, deve-se incorporar, para denotar um sistema realmente inteligente, as noções de agentes autônomos os quais interagem com o ambiente;
- sistemas inteligentes devem ir além da incorporação de elementos inerentes aos agentes autônomos, mecanismos de sensoriamento, atuação e de seleção de comportamento assim como representação das metas e objetivos;
- sistemas inteligentes devem ser capazes de evoluir em função das próprias atitudes e metas tomadas;
- é necessária a abstração do processo cognitivo natural para que as teorias criadas a partir destes estudos possam ser utilizadas e aperfeiçoadas quando na implementação de sistemas inteligentes;

- objetos podem ser instanciados nas memórias de trabalho dos agentes sendo que, os seus significados são baseados na própria percepção dos agentes acerca dos estímulos externos combinados com o conhecimento previamente aprendido;
- os agentes devem manipular várias instâncias de ontologias de modo representar os vários aspectos envolvidos em seu ciclo de vida, metas e objetivos. Cada agente deve construir e manipular sua própria ontologia.

Visando atender a todos os pontos anteriormente explicitados, o autor propõe a arquitetura exibida na Figura 2-8.

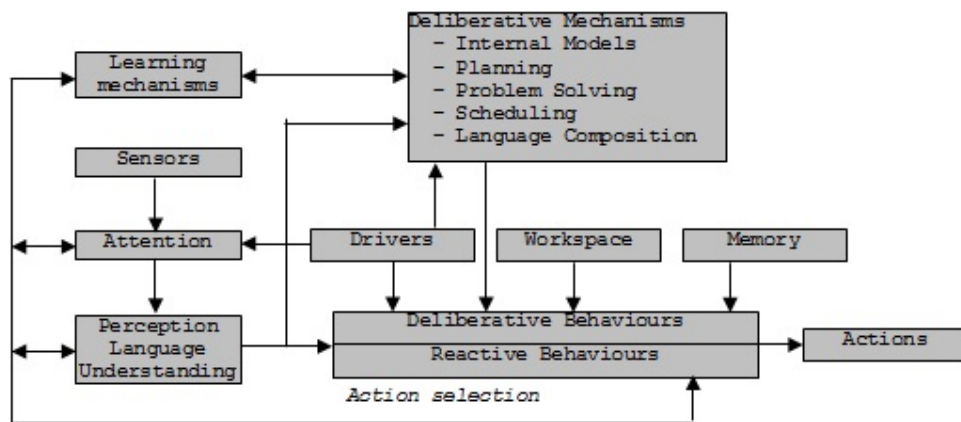


Figura 2-8: Arquitetura do sistema cognitivo apresentado por (Franklin, 1997)

De acordo com a Figura 2-8, é possível observar elementos essenciais para caracterização de um sistema inteligente:

- Mecanismos de aprendizagem (*Learning Mechanisms*): o conhecimento armazenado (ou aprendido) é atualizado ou modificado em função dos mecanismos deliberativos associados com o próprio comportamento do sistema, nível de atenção e inferências das percepções externas;
- atenção (*Attention*): denota o filtro para coletar, do ambiente, elementos essenciais para a tomada de ações em função das metas e do próprio conhecimento prévio armazenado;
- metas (*Drivers*): representa o conjunto de metas e objetivos modelado para o sistema inteligente em questão;

- sistema de memória: na proposta, foram criados dois tipos de memória denotados como de curto prazo (*workspace*) e memória de longo prazo (referenciada, na figura, como *memory*). Esta divisão faz alusão ao próprio funcionamento do cérebro biológico;
- comportamentos: representados por comportamentos reativos (*reactive behaviours*) e comportamentos deliberativos (*deliberative behaviours*). O comportamento deliberativo é associado a, por exemplo, planejamento, metas e modelos internos. Por sua vez, o reativo é regido apenas por um conjunto de regras para atuação instantânea a certos estímulos externos.

Por sua vez, (Yoon *et al.*, 2000) apresenta um modelo mais simplificado em relação ao apresentado por (Franklin, 1997) mas que, porém, apresenta as mesmas funcionalidades básicas inerentes aos sistemas cognitivos (Figura 2-9): filtragem dos estímulos externos (*Perception System*) atuando em função das objetivos preestabelecidos (*Motivation System*); o sistema cognitivo propriamente dito (*Behaviour System*) o qual recebe informações dos sistema de percepção e do sistema de motivação (podendo modificar as metas contidas neste); e o sistema motor, atuando diretamente no ambiente em de acordo com as ações evocadas pelo *Behaviour System* ou pelo *Perception System* – tais ações são análogas, respectivamente, aos comportamentos deliberativo e reativo modelados em (Franklin, 1997).

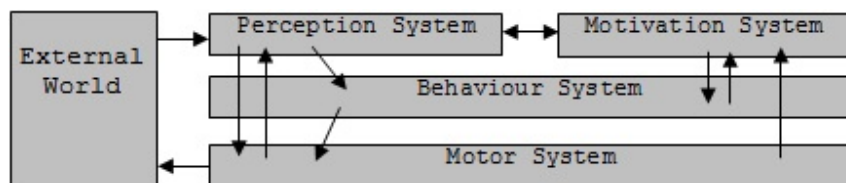


Figura 2-9: Arquitetura sugerida por (Yoon *et al.*, 2000)

O modelo apresentado na Figura 2-9 equipa uma criatura artificial (cão *Sydney K9*) onde o processo de aprendizagem é realizado por meio de ação-recompensa e atos de imitação – mesma técnica utilizada para o treinamento de animais de estimação. A interação com o K9 é feita por intermédio da voz – pelo fato da criatura artificial não entender a linguagem e, sim, apenas palavras de comando, os sons captados e

recebidos pelo K9 são convertidos em coeficientes cepstral e armazenados em sua memória auditiva (memória de curta duração) para que se possa, internamente, realizar um casamento de padrões para a identificação da referida palavra de comando.

A ideia de modelagem de criaturas artificiais também é sugerida em (Isla *et al.*, 2001). A principal característica mencionada pelos autores consiste na estruturação da arquitetura em camadas. É factível a cada grupo funcional acessar uma memória interna (concebida na forma de *blackboard*) através de barramentos compartilhados (Figura 2-10). Desta forma, obtém-se uma maior flexibilidade frente às diversas tecnologias e mecanismos que poderão ser usadas na implementação da criatura artificial.

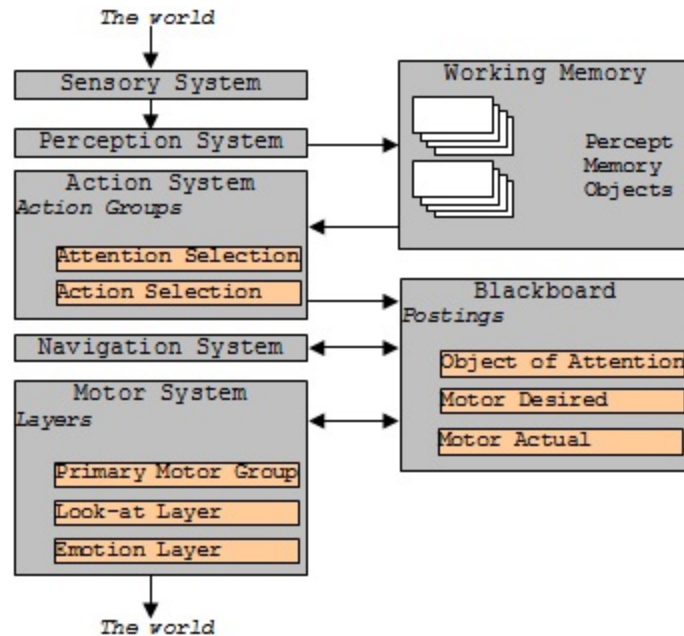


Figura 2-10: Estruturação em camadas do sistema cognitivo proposto em (Isla *et al.*, 2001)

Em (Imbert & de Antonio, 2005) é apresentada uma modelagem para uma arquitetura cognitiva genérica para emular personagens virtuais onde suas ações são impactadas pelo seu estado emocional. Conforme ilustrado na Figura 2-11, os autores propõem uma estruturação em camadas onde a atuação sobre o meio pode ser provocada por razões sociais, deliberativas ou reativas.

Igualmente às demais arquiteturas previamente apresentadas, a camada de interpretação (*Interpreter*) é responsável pela filtragem das informações coletadas pelos

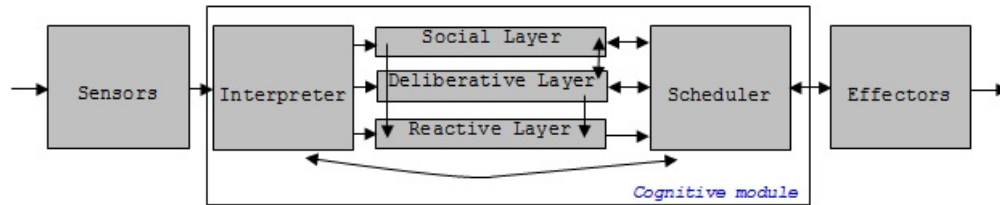


Figura 2-11: Sistema cognitivo levando-se em conta o fator emocional proposto em (Imbert & de Antonio, 2005)

módulo sensório a fim de entregar, à camada seguinte, somente as informações relevantes.

Por último, o trabalho referenciado em (Arnellos *et al.*, 2008) apresenta uma arquitetura de um *IVA* (*Intelligent Virtual Agent*) – outra nomenclatura dada aos sistemas cognitivos baseados em multiagentes. Para tanto, os autores utilizam teorias semióticas, como por exemplo, as de Peirce (citado no Capítulo 1). Neste foco, as informações armazenadas e processadas são estruturadas em signos. A Figura 2-12 apresenta a arquitetura do sistema cognitivo sugerido pelos autores.

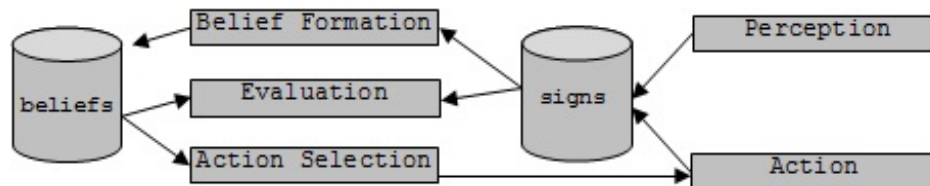


Figura 2-12: Arquitetura do *IVA* (*Intelligent Virtual Agent*) proposta em (Arnellos *et al.*, 2008)

Na arquitetura representada na Figura 2-12, os eventos de percepção (*perception*) e de ação (*action*) produzem signos que podem ser avaliados para a formação de crenças (*belief*). Define-se crenças como sendo os estados e interações esperados a serem assumidos pelo agente. Durante o evento de ação, realiza-se uma realimentação para uma possível reestruturação dos signos, adequando-os às novas realidades impostas pelo ambiente. Toda ação é resultante do processo de seleção (*Action Selection*) o qual consiste no módulo deliberativo.

Para um melhor entendimento em relação ao modelo baseado em signos, o autor estrutura o conhecimento por intermédio de uma hierarquia de signos e seus inter-

relacionamentos. Por exemplo, os signos de nível 0 (zero) compreendem, por exemplo: tempo, identificação de um objeto, posição, orientação e ação. Os signos de nível 1 (um) são gerados a partir da percepção e ações do próprio agente, relacionando os signos de nível 0, como por exemplo:

- Percepção do objeto: são os próprios valores instanciados em seu nível 0;
- relação funcional (semântica): relaciona que o objeto **o**, estava localizado, no tempo **t**, na posição **p** e apresentava uma orientação **r**;
- estado: associa um estado **s** com um tempo **t**;
- ação: denota uma ação **a** executada pelo agente no tempo **t**.

Estes níveis são combinados de modo a representar o real mapeamento do agente e do ambiente a cada instante de tempo.

Por outro lado, a formação das crenças, as quais atuam no módulo deliberativo, pode ser regida por quatro conjuntos de heurísticas, enumerados a seguir:

- Fluentes: dois signos que referenciam o objeto e apresentam signos de tempo sucessivos são unidos para formar um fluente. A semântica do signo fluente corresponde à alteração dos valores $\{v_1, v_2, \dots, v_K\}$ no signo temporal t para $\{v'_1, v'_2, \dots, v'_K\}$ em $(t+1)$;
- efeito das ações: um signo denotando propriedades do agente **o** no tempo t , um signo **a** de ação efetuada no tempo t e um signo com os valores do mesmo agente **o** no momento $(t+1)$ podem ser unidos para formar um novo signo que descreve os efeitos da ação sobre o agente **o** no tempo t de modo que assuma um novo estado no tempo $(t+1)$;
- efeito de eventos: similarmente ao efeito das ações, signos de eventos podem correlacionar objetos nos tempos t e $(t+1)$;
- estados desejados: Um signo de status associado ao tempo t pode ser associado a outros signos de mesma temporalidade. Esta heurística permite avaliar as

interferências do ambiente sobre o estado do agente, impactando, portando, na condução das ações para alcançar as metas previamente estabelecidas.

2.3 Representação do Conhecimento

Inicialmente, é conveniente mencionar que, no caso de sistemas inteligentes, as informações a serem armazenadas, para posterior processamento, representam informações de estado do ambiente e do próprio sistema, regras, metas, heurísticas, relacionamentos, ações, eventos e diversas outras para representar o mundo real e sua interação com o sistema computacional.

A seguir, são apresentadas algumas formas clássicas de representação de informações e, também, algumas que serão úteis ao desenvolvimento deste trabalho. É factível, ainda, explorar de forma mais ampla, a manipulação de signos em seus níveis de abstração (conforme já citado na seção anterior) como forma, também, de representação de conhecimento.

As mais conhecidas técnicas de representação baseiam-se em lógica matemática (cálculo de predicados), regras de produção, frames, grafos conceituais, ontologias, modelos conexionistas, lógica fuzzy e redes bayesianas, podendo haver combinações entre as técnicas.

2.3.1 Lógica Matemática

A lógica matemática (ou cálculo proposicional) consiste no uso de proposições e relações lógicas entre proposições. Aos conjuntos de proposições e relações lógicas dá-se o nome de expressões. Os sistemas inteligentes manipulam a chamada lógica de primeira ordem, a qual consiste de uma estrutura de representação mais complexa em relação ao cálculo proposicional puro.

Como exemplo de elementos da lógica de primeira ordem pode-se citar:

- Classes: identificam um tipo de entidade;
- relações: denotam o relacionamento entre dois objetos ou instâncias;

- propriedades: conjunto de atributos permissíveis aos objetos;
- funções: ações/predicados sobre os objetos;
- variáveis: instâncias das classes;
- quantificadores: quantificador universal, quantificador existencial;
- relações lógicas / operadores: E, OU, NÃO, implicação, equivalência.

2.3.2 Regras de Produção

O uso de regras de produção é extremamente simples e o seu emprego pode ser mais facilmente encontrado em sistemas especialistas. No geral, resumidamente, uma regra de produção consiste na utilização de testes lógicos que, caso sejam validados como verdade, implica no acionamento de uma ação.

Geralmente, as regras são confeccionadas no formato:

Se <condição> Então <consequente> Faça <ações>

Onde:

- Condição: verificação lógica dos predicados;
- conclusões: alteração dos estados dos objetos, armazenamento na memória de curta duração;
- ações: eventos a serem executados caso a condição seja satisfeita.

2.3.3 *Frames*

A concepção da utilização de *frames* foi introduzida por Marvin Minsky (Minsky, 1974) onde as informações são agrupadas de acordo com os atributos que descrevem o ambiente. Os *frames* são organizados de forma a se obter uma hierarquia de informações tal como o conceito de hierarquia na metodologia computacional de orientação-a-objetos.

Para um melhor entendimento, a Figura 2-13, extraída de (Fensel, 2003) ilustra o emprego de frames.

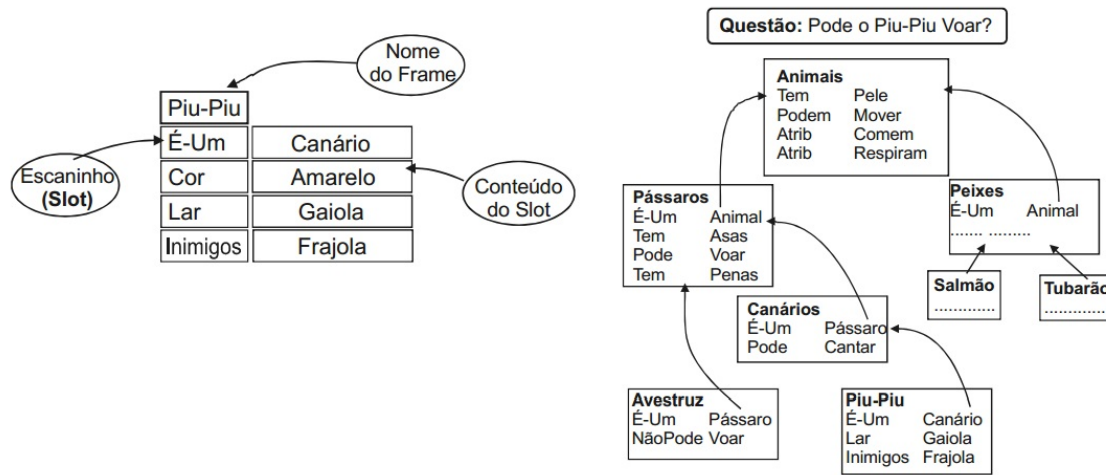


Figura 2-13: Exemplificação do uso de *frames*

Na Figura 2-13, observa-se, no lado esquerdo, um *frame* para caracterizar o personagem de desenho infantil “Piu-Piu”. Os slots representam os atributos associados ao personagem tendo suas respectivas instâncias.

Por sua vez, o lado direito da Figura 2-13 retrata um processo de busca de modo a resolver um questionamento. A avaliação dos *frames* segue uma hierarquia de modo a melhor representar os elementos do ambiente.

2.3.4 Grafos Conceituais

Grafos conceituais (também conhecidos como mapas conceituais) são baseados na teoria cognitiva de Ausubel (Ausubel *et al.*, 1981) em que, por meio de uma representação gráfica (grafo), consegue-se abstrair relações e implicações de significados. Por este motivo, os mapas conceituais relacionam-se à chamada aprendizagem significativa.

Como pontos relevantes da utilização de mapas conceituais encontram-se:

- Abstrair a estrutura de significados assim como identificar aqueles relevantes ao estado atual do sistema;

- reconhecer os significados preexistentes, construindo, quando necessário, novos conhecimentos no transcurso do tempo;
- permitir uma organização hierárquica do conhecimento;
- fazer o mapeamento de relações explícitas entre objetos do conhecimento acumulado.

2.3.5 Ontologias

Ontologia consegue descrever, formalmente, estruturas, conceitos e dados sob os aspectos inerentes a um determinado domínio. Genericamente, uma ontologia é composta por:

- Conjunto de conceitos (abstratos ou concretos; elementares ou compostos);
- hierarquia entre esses conceitos;
- conjunto de relacionamentos entre esses conceitos;
- descrição exata do conhecimento;
- conjunto de funções;
- conjunto de axiomas;
- conjunto de instâncias.

Para propiciar uma utilização mais abrangente, as ontologias podem ser classificadas em ontologias de alto nível, domínio, tarefas e aplicação:

- Alto nível: responsável pela descrição de conceitos genéricos;
- domínio: especializam os conceitos de alto nível adaptando-os e especializando-os a um determinado domínio;
- tarefas: descrevem o vocabulário inerente às tarefas de um domínio de aplicação;
- aplicação: especificam conceitos especializados relacionando-os a aplicações pontuais.

2.3.6 Redes Bayesianas

De acordo com (Pearl, 1988), uma rede bayesiana consiste de um grafo direcionado acíclico capaz de representar conhecimento dotado de incertezas. No grafo, os nós representam as variáveis relevantes ou, no caso deste *framework*, representam itens do conhecimento manipulado. Os arcos denotam relacionamentos causais probabilísticos entre as variáveis.

Para representar objetos do mundo real, cada nó contém uma tabela de probabilidades, onde:

- A tabela dos nós do topo do grafo descreve a probabilidade de ocorrência de cada estado passível de ser assumido pelo nó;
- a tabela em cada nó-filho reflete a combinação da probabilidade de cada estado do nó com as probabilidades dos estados de seus nós-pai.

Uma das principais dificuldades dos sistemas baseados em conhecimento consiste na modelagem do próprio conhecimento. Esta dificuldade também é inerente à modelagem das redes bayesianas. Em muitas situações, tais como relativas à modelagem de cérebros artificiais, não se conhece, previamente, toda a estrutura do conhecimento a ser manipulado. Devido à esta falta da visão global inicial, é necessária a adoção de mecanismos de aprendizagem da rede bayesiana. Mecanismos de aprendizagem, geralmente, NP-difícil (Chickering *et al.*, 2004), permitem a indução da rede a partir de um conjunto de observações prévias. Ainda em relação ao fato de que o problema de aprendizagem da rede bayesiana ser NP-difícil, muitos autores, tais como os idealizadores dos trabalhos descritos em (De Stefano *et al.*, 2010; dos Santos *et al.*, 2010; Khanteymoori *et al.*, 2011), têm proposto mecanismos evolucionários de aprendizagem.

2.4 Sistemas Evolucionários

Computação evolucionária baseia-se nos princípios de Darwin⁶ e da genética para a resolução de problemas. São fundamentalmente mecanismos de otimização por meio de processos iterativos. A cada iteração, uma busca paralela pode ser realizada selecionando um conjunto de soluções parciais (indivíduos). Os indivíduos são avaliados em função do objetivo e são submetidos a um processo de evolução para gerar um novo conjunto de indivíduos.

Neste processo, cada elemento do processo natural é mapeado através de uma estrutura ou uma funcionalidade computacional:

- Indivíduos podem ter os seus cromossomos representados por palavras binárias, vetores e matrizes;
- cada indivíduo possui uma aptidão para a sobrevivência no contexto da seleção natural. O grau de aptidão é gerado como resultado de uma função de avaliação do indivíduo em um referido contexto;
- mecanismos naturais de evolução (seleção natural, reprodução, cruzamento e mutação) são realizados através de funções computacionais que incidem sobre as estruturas (palavras binárias, vetores e matrizes).

Segundo (Lopes & Takahashi, 2011), pode-se dividir a computação evolucionária em cinco grupos principais:

- Algoritmo Genético;
- programação Genética;
- estratégia evolutiva;
- programação evolucionária;
- evolução diferencial.

⁶Charles Darwin (1809-1882), naturalista inglês autor da Teoria da Evolução por Seleção Natural.

Em 1994 foi introduzida, na computação evolucionária, mais um grupo representado pela evolução diferencial (Storn & Price, 1995). Evolução diferencial é um método computacional de otimização, facilmente paralelizável, para funções espaciais contínuas.

Associadas a estes grupos, encontram-se diversas linhas de pesquisa, tais como: Vida Artificial (*Artificial Life*), Sistemas Classificatórios (*Classifier Systems*), Computação e Bio-informática baseadas em DNA (*DNA-based Computing and Bio-informatics*), Robótica Evolucionária (*Evolutionary Robotics*) e *Hardware* Evolucionário (*Evolvable Hardware*).

A seguir, são descritas, sumariamente, as linhas de pesquisa exemplificadas dentro do contexto de computação evolucionária.

- *Artificial Life*: fundamenta-se na abstração do comportamento e da evolução de sistemas (organismos) naturais através de modelagens computacionais, robóticas e bioquímicas. Segundo (Egbert *et al.*, 2011), diversos estudos estão sendo realizados no campo do comportamento adaptativo através da modelagem de agentes artificiais. Estes estudos envolvem projetos, por exemplo, modelos de sistemas cognitivos;
- *Classifier Systems*: representa uma aplicação especializada de algoritmo genético de modo a torná-lo mais adaptativo dentro do contexto de um ambiente dinâmico. Um exemplo clássico do uso de sistemas classificatórios consiste na implementação de veículos autônomos como em (Stafylopatis & Blekas, 1998).
- *DNA-based Computing and Bio-informatics*: essa concepção foi criada com o objetivo de resolver problemas computacionais denominados como NP-Completo. Para tanto, baseia-se nas estruturas e operações inerentes ao DNA biológico. Em (Maojo *et al.*, 2010) pode-se ter um paralelo entre o *DNA-computing* e a nanoinformática;
- *Evolutionary Robotics*: conforme descrito (Delarboulas *et al.*, 2010), a robótica evolucionária visa projetar robôs autônomos, capazes de auto-adaptações sob

demanda em função de variações no ambiente no qual o robô se encontra. Outros projetos podem também ser citados como os descritos em (Raja & Raviraj, 2012), (Ritter *et al.*, 2003) e (Oates *et al.*, 2010).

Como mencionado, os mecanismos evolucionários são derivados das observações dos próprios processos naturais de evolução. Desta forma, muitos fatores contribuem para o sucesso de sua aplicação em diversas aplicações baseadas em conhecimento. Dentro os principais pontos, pode-se destacar (Jong, 2006):

- Populações de indivíduos: os indivíduos concorrem por recursos limitados. Neste caso, os recursos representam a solução do sistema para a qual, o indivíduo deve convergir.
- Aptidão: denota o grau de adaptação do indivíduo ao meio. Essa adaptação influencia diretamente na habilidade de sobrevivência do indivíduo assim como a sua capacidade de reproduzir-se. Neste caso, permanecem as soluções parciais que melhor se enquadram na classe de soluções ótimas do problema.
- Dinamicidade das populações: os ciclos de nascimento e de morte de indivíduos diferenciando-se em suas características acarretam em alterações das características de uma população. Aplicando-se esse ponto aos sistemas computacionais, representa o grau de convergência de toda uma população em direção ao ponto ótimo da solução do problema.
- Variabilidade e hereditariedade: os novos indivíduos, apesar de diferentes em relação aos seus progenitores mantêm muitas de suas características. Fator esse importante para caminhar no conjunto de soluções em direção ao ponto ótimo da função.

Em linhas gerais, de acordo com (Eiben & Smith, 2003), um algoritmo típico de uma estratégia evolucionária segue os passos descritos a seguir:

1. Inicia população com soluções (indivíduos) candidatos à solução (objetivo do problema) de forma aleatória.

2. Avalia cada indivíduo atribuindo, a cada indivíduo, um grau de adaptação (*fitness*).
3. Seleciona os indivíduos mais adaptados em função dos melhores valores de *fitness*.
4. Recombina pares de indivíduos.
5. Aplica mutações sobre os indivíduos resultantes
6. Avalia os novos indivíduos gerados
7. Seleciona os indivíduos para a formação da nova geração
8. Volta ao passo 3 enquanto o critério de parada não for satisfeito

2.5 Modelagem e Implementação

Como descrito anteriormente, a modelagem de um sistema inteligente, em linhas gerais, integra quatro componentes básicos:

- Sistema sensório;
- sistema de armazenamento, codificação e processamento;
- sistema de atuação;
- sistema de comunicação.

Levando-se em conta a generalidade de aplicações que podem ser construídas a partir da concepção de sistemas inteligentes, foi realizada uma modelagem genérica de uma arquitetura para que, a partir dela, possam ser construídos sistemas inteligentes em diversas áreas de aplicação.

Em termos gerais, a arquitetura proposta para o desenvolvimento de sistemas cognitivos usa a ideia de agentes inteligentes para permitir o agrupamento de informações e de classes de processamento. Este agrupamento permite uma melhor

representação e organização do conhecimento incorporado e manipulado pelos agentes além de proporcionar uma maior escalabilidade. Outra razão para o agrupamento das informações é permitir uma separação dos agentes em termo de classes funcionais levando-se em conta o tipo de conhecimento a ser representado e o tipo da ação requerida pela aplicação implementada sobre essa arquitetura. Dessa forma, consegue uma alta granularidade necessária para a implementação de aplicações mais complexas tais como *health monitoring systems* como descrito em (Julka *et al.*, 2011). Por conclusão, sendo a arquitetura proposta um sistema multiagente, permite-se a utilização simultânea de técnicas diferentes no tangente à manipulação de conhecimento, como por exemplo, utilização de redes bayesianas, modelos fuzzy, modelos estatísticos, conexionistas e evolucionários. O sistema é flexível o suficiente em termos de escalabilidade para incluir novos métodos e abordagens híbridas para tarefas complexas.

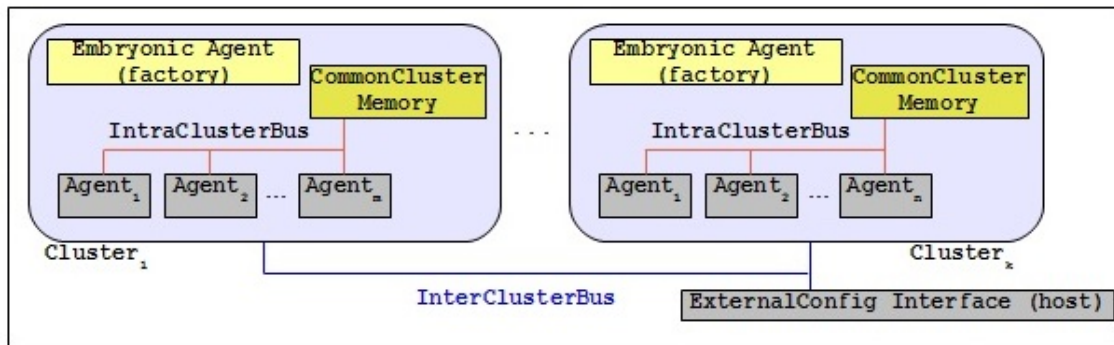


Figura 2-14: Arquitetura básica do sistema modelado

A Figura 2-14 apresenta uma visão macroscópica do sistema com a instanciação de múltiplos *clusters*. Cada *cluster* tem três básicos e permanentes componentes: (*Embryonic Agent*, *CommonCluster Memory* e *IntraClusterBus*) e suas respectivas instanciações internas de agentes.

O *Embryonic Agent* gerencia a criação ou remoção das instâncias dos agentes. Os mecanismos exportados pelo *Embryonic Agent* permitem a auto-adaptação do sistema multiagente devido ao dinamismo apresentado pelo ambiente sob monitoramento e atuação. O *CommonCluster Memory* é uma área comum para o armazenamento de informações de longo prazo (*long term memory*, ou seja, informações tais como as de configurações, caracterizações dos conhecimentos incorporados e codificações relacio-

nadas ao processo de atuação. O *IntraClusterBus* e o *InterClusterBus* são elementos comunicacionais representados por estruturas de armazenamento do tipo *mailbox* no caso da implementação do sistema em *software* ou baseados no padrão *Wishbone* (Sharma & Kumar, 2012) no caso da implementação do sistema em *hardware*.

Ainda em relação à Figura 2-14, é possível notar a presença das instâncias dos agentes internamente aos *clusters*. Cada agente, operando de forma autônoma e cooperativa, é responsável por:

- Aquisição de informações sensórias (relativo ao *sensory system*);
- codificação, armazenamento e processamento das informações;
- atuação sobre o ambiente.

Para acondicionar tais funcionalidades, cada agente possui a estruturação funcional ilustrada na Figura 2-15.

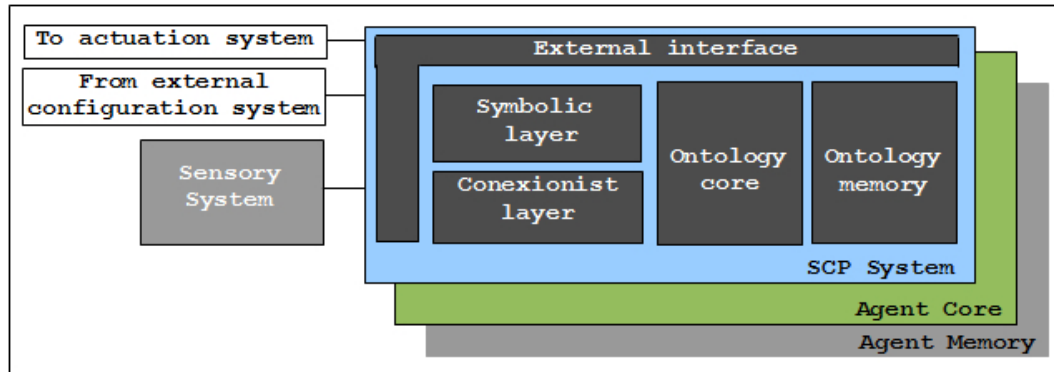


Figura 2-15: Estruturação funcional de cada agente

A Figura 2-15 apresenta os elementos básicos que compõem cada agente: *Sistema Sensório (Sensory System)*; *Sistema de Armazenamento, Codificação e Processamento (SCP System)*; *Núcleo do Agente (Agent Core)* e *Memória do Agente (Agent Memory)*, que serão descritos nas seções a seguir.

2.5.1 Sistema Sensório (*Sensory System*)

O *sensory system* objetiva a coleta dos estímulos (informações) provenientes do ambiente para que possa alimentar o *SCP System* (Subseção 2.5.2). A principal função

desta unidade consiste na adequação das informações coletadas para que possa ser manipulada pelas outras unidades funcionais do sistema e facilitar a atuação sobre o ambiente externo. Esses dois tipos de processamento representam as ações deliberativa e reativa, respectivamente.

O sistema sensório deliberativo é responsável por interfacear o sistema com o ambiente para que possa capturar, por exemplo, imagens, sons ou outras formas de onda (*stream devices*). Dependendo do tipo de dispositivo usado para a captura das informações externas, o objetivo do sistema sensório deliberativo consiste em decodificar o estímulo externo em mapeamentos conexionistas. Tal mapeamento conexionista pode ser a base do sistema cognitivo.

A estruturação do *Sensory System* é apresentado na figura 2-16 onde pode-se notar a integração de mecanismos para a captura tanto de sinais do tipo *stream* quanto dados amostrados a partir de sensores ou transdutores.

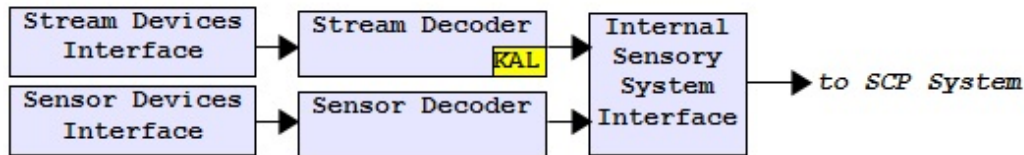


Figura 2-16: Estruturação funcional do sistema sensório

Ainda em relação à Figura 2-16, o decodificador de fluxo (*stream decoder*) visa capturar fluxos de informações (*stream*), tais como os fluxos de vídeo e de som ou qualquer outro sinal contínuo, dividindo-os em quadros (*frames*) de tal forma que eles possam ser avaliados e manipulados pelo *SCP System*. Os modelos de decodificação usados pelo *Stream Decoder* são armazenados em um repositório chamado de *Camada de Abstração de Conhecimento* (*Knowledge Abstraction Layer*) (denotado, na referida figura, por *KAL*). Por outro lado, o *Sensor Device Interface* visa o recebimento de informações provenientes de sensores e transdutores. As informações recebidas (*stream* e *sensor*) são direcionadas ao *SCP System* por intermédio da *Internal Sensory System Interface*.

2.5.2 Sistema de Armazenamento, Codificação e Processamento (*SCP System*)

De forma a permitir uma maior modularidade e reuso de estruturas mentais que representam o conhecimento, o *SCP System* é baseado em uma topologia híbrida, adotando-se, neste caso, os modelos conexionista e simbólico. O modelo híbrido foi adotado pelo fato de melhor abstrair as categorizações de Peirce para a composição e representação de conhecimento. Sendo assim, tem-se:

- *Primeiridade*: modelo conexionista onde cada estrutura conexionista denota um objeto do mundo real *quali-signos*.
- *Secundidade*: mapeamento entre o modelo conexionista e o conhecimento armazenado. Neste nível, são usadas ontologias para denotar a estruturação e composição dos signos.
- *Terceiridade*: aplicação de regras e mecanismos de alto nível para a formação do pensamento e do raciocínio lógicos, relacionando instâncias da Primeiridade e da Secundidade.

Os elementos internos do *SCP* formam a base do sistema baseado em conhecimento implementado e podem ser replicados em *clusters* para melhor representação da informação (classes de conhecimento) e para prover maior dinamismo ao processamento. Cada *cluster* pode evocar funções de criação e de remoção de agentes ou de outros *clusters* exportadas pelo *Embryonic Agent Layer (factory)*, conforme mostrado na Figura 2-14.

Ao *SCP System* estão atribuídas as funcionalidades deliberativa e reativa do sistema. Entende-se como funcionalidade reativa como a capacidade de atuar, prontamente, no ambiente externo após a inferência de uma informação apresentada. Por sua vez, a funcionalidade deliberativa relaciona-se à capacidade de efetuar modificações no conhecimento aprendido ou raciocinar a partir de uma informação apresentada na entrada do sistema (Arnellos *et al.*, 2008).

Para um melhor detalhamento da modelagem do sistema, os componentes internos do *SCP System* são sumarizados a seguir:

- Camada Conexionista (*Connectionist Layer*): a camada connexionista está associada com a funcionalidade reativa e fornecer subsídios para a funcionalidade deliberativa. Os mapas connexionistas consistirão na base do sistema cognitivo - neste caso, a informação formará, de acordo com Peirce (Peirce, 1958), a *Primeiridade*, isto é, a informação isolada relativa ao objeto, caracterizando, portanto, o *quali-signo* e não formando, ainda, o conhecimento propriamente dito.
- Camada Simbólica (*Symbolic Layer*): A função do *Symbolic Layer* consiste na formação dos signos e do próprio raciocínio do sistema baseado em conhecimento. O *Symbolic Layer* deve atuar de acordo com as ontologias armazenadas nas bases do sistema. Além disso, é responsável pela implementação dos mecanismos evolucionários os quais proporcionam a dinamicidade cognitiva. De forma a possibilitar a abstração das categorizações de Peirce, o *Symbolic Layer* é dividido em duas subcamadas conforme ilustra a Figura 2-17.

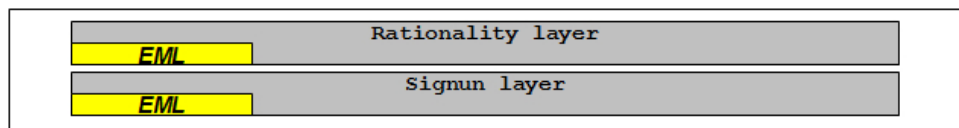


Figura 2-17: Estruturação do *Symbolic Layer*

Na Figura 2-17, nota-se a presença da camada de mecanismos evolucionários (*Evolutionary Mechanisms Layer* - *EML*) que, como anteriormente mencionado, provêm mecanismos de reconfiguração de modo a propiciar a geração de signos e adequar o raciocínio lógico às novas demandas ambientais.

- Núcleo de Ontologia (*Ontology Core*): A funcionalidade básica do *Ontology Core* é executar um *parser XML*, o qual realiza operações de busca e de manipulação sobre estruturas *XML* (neste *framework*, *XML* denotam as próprias ontologias). As ontologias provêm, neste sistema, a representação do conhecimento sob mani-

pulação, ou seja, podem denotar as regras gramaticais, estruturação do modelo conexionista ou a especificação inicial de uma rede bayesiana (Pearl, 1988).

- Memória de Ontologia (*Ontology Memory*): O módulo *Ontology Memory* tem a função de armazenar ontologias manipuladas pelos agentes instanciados no *cluster*.
- Núcleo do Agente (*Agent Core*): O *Agent Core* tem a função é realizar as operações inerentes aos agentes.
- Memória do Agente (*Agent Memory*): Diferentemente do *Ontology Memory*, o *Agent Memory* objetiva o armazenamento de código e atributos dos agentes, por exemplo, seus estados.

2.5.3 Sistema de Atuação (*Actuation System*)

O *Actuation System* representa a saída do sistema onde o tipo dos dados fornecidos é determinada pela própria aplicação.

2.5.4 Sistema de Comunicação (*Communication System*)

Além da necessidade de modelar os mecanismos inerentes à representação de conhecimento e de cognição, é também necessária a construção de uma infraestrutura de comunicação entre os agentes e entre os *clusters*. Voltando à Figura 2-14, estes dois níveis de comunicação são representados pelos barramentos denominados como *IntraClusterBus* e *InterClusterBus*. Para facilitar o protótipo inicial deste sistema, optou-se usar modelo baseado em *buffers* compartilhados do tipo *mailbox*.

2.6 Implementação em Software

Para a implementação em software, foram modelados e implementados classes e métodos básicos e suficientes para que sejam construídas aplicações genéricas baseadas em conhecimento, ou seja, as classes e métodos modelados independem da aplicação

construída. A aplicação propriamente dita deverá ser implementada adicionando-se o seu código ao *Agent Core*. O referido código, além de conter as funcionalidades da aplicação desenvolvida, será também responsável pela ativação dos métodos exportados pelas demais classes que compõem a arquitetura.

Para tanto, uma vez que a arquitetura é formada por módulos funcionais (agentes), a implementação foi baseada em *threads*. A vantagem de se utilizar *threads* consiste na fácil escalabilidade do sistema por intermédio, simplesmente, da incorporação e ativação de novas *threads* ou pela instanciação dinâmica gerenciada pelo agente embriônico (*Embryonic Agent*). A utilização das *threads* interfaceando apenas com memórias do tipo *mailbox* proporciona um fraco acoplamento do sistema. O fraco acoplamento, aliado à escalabilidade, tem como consequência a facilidade de implementação de sistemas baseados em conhecimento de forma genérica.

A Figura 2-18 explicita as *threads* implementadas de acordo com as funcionalidades da arquitetura previamente descritas na Seção 2.5.

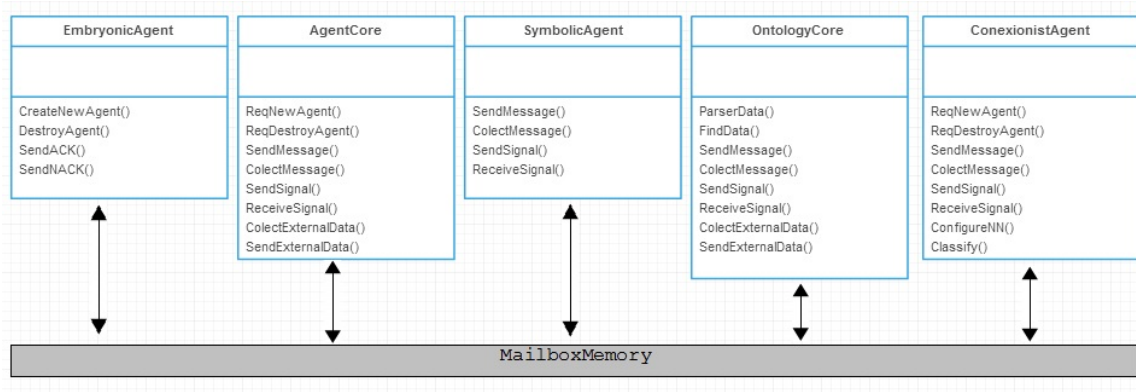


Figura 2-18: Representação UML das classes implementadas da arquitetura proposta

As descrições dos métodos pertencentes às classes da Figura 2-18 são realizadas na Tabela 2.1.

2.7 Conclusões

Realizando-se um paralelo com alguns pontos abordados no decorrer do Capítulo 1, pode-se notar uma íntima relação das arquiteturas cognitivas com os conceitos

Tabela 2.1: Métodos das classes (*threads*) criadas

Método	Descrição
CreateNewAgent()	Responsável pela instanciação de novas <i>threads</i> pelo <i>EmbryonicAgent</i> . A requisição para a criação é feita pela <i>AgentCore</i> .
DestroyAgent()	Responsável pela remoção de <i>threads</i> ativas pelo <i>EmbryonicAgent</i> . A requisição para a remoção é feita pela <i>AgentCore</i> .
SendACK()	Envia retorno de sucesso da criação ou remoção das <i>threads</i> .
SendNACK()	Envia retorno de falha da criação ou remoção das <i>threads</i> .
ReqNewAgent()	Método para a solicitação criação de novas <i>threads</i> que correspondem às funcionalidades básicas do agente ou, por exemplo, aos novos mecanismos do <i>ensemble learning</i> .
ReqDestroyAgent()	Método para solicitação de criação ou remoção de <i>threads</i> .
SendMessage()	Envia mensagem por intermédio do <i>Mailbox Memory</i> .
ColectMessage()	Coleta mensagem a partir do <i>Mailbox Memory</i> .
SendSignal()	Sincronização de envio ou recebimento das mensagens.
ReceiveSignal()	Sincronização de envio ou recebimento das mensagens.
ColectExternalData	Coleta dados a partir da interface externa da arquitetura.
SendExternalData()	Envia dados para a interface externa da arquitetura.
ParserData()	<i>Parser</i> XML.
FindData()	Procura dados na estrutura XML.

kantianos os quais categorizam os signos na geração do conhecimento em três níveis: Primeiridade, Secundidade e Terceiridade.

Como já explicitado, a Terceiridade é formada pelos objetos correlacionados que, no caso dos modelos cognitivos, consiste no armazenamento de informações coletadas do meio externo após a incidência de regras e filtros – que, neste caso, pode-se utilizar as informações previamente armazenadas no sistema (definição de Kant da consciência).

Este capítulo apresentou, também, a proposta de uma arquitetura para a implementação de sistemas baseados em conhecimento. Devido ao fato da adoção de camadas funcionais integrando processamento conexionista e simbólico, além da utilização da concepção de sistemas multiagentes e de clusterização, esta arquitetura é generalista, ou seja, permite a implementação de uma diversa gama de sistemas baseados em conhecimento. A ideia de utilizar as categorizações de Peirce auxiliaram na abstração da formação de conhecimento e, conseqüentemente, auxiliaram na criação da estruturação funcional do sistema.

Como qualquer sistema baseado em conhecimento, pode-se notar semelhanças frente aos modelos apresentados no decorrer desse capítulo. Tais semelhanças podem ser traduzidas na presença de elementos reativos e deliberativos além dos sistema de memória (o qual engloba a memória de trabalho). A diferença básica da arquitetura apresentada neste trabalho consiste na utilização de componentes que permitem a flexibilização frente às aplicações que poderão ser implementadas. Como elementos de flexibilização podem ser citados: utilização de nós funcionais de processamento em uma arquitetura multiagente, manipulação de ontologias no formato XML para garantir a facilidade na configuração do sistema e das informações a serem manipuladas, utilização do modelo híbrido para a manipulação do conhecimento.

Para a validação da arquitetura proposta, foi implementado um sistema de geração de prognóstico de falhas em transformadores de potência, que será descrito na Seção 4.1.

Capítulo 3

Prognósticos

3.1 Conceitos de Prognósticos

Prognóstico é uma disciplina de engenharia que envolve a previsão do momento no qual um sistema ou componente deixará de funcionar como pretendido além de, se possível, classificar o tipo de falha que ocorrerá (Vachtsevanos *et al.*, 2006; Liao, 2009; Srivastava & Han, 2011; Mosallam *et al.*, 2014). O tempo previsto para a ocorrência da falha é denominado vida útil remanescente (*Remaining Useful Life* - RUL). RUL é um conceito importante em termos de contingência do sistema e, também, pode ser perfeitamente associado ao Gerenciamento da Saúde do Sistema (*System Health Management* - SHM). Autores como (Xiongzi *et al.*, 2011) são mais categóricos na definição do RUL, retratando-o como sendo a duração entre o tempo atual e o final da vida útil do componente (Figura 3-1).

Na Figura 3-1, pode-se perceber claramente a correlação entre o RUL e o índice da saúde do sistema explicitando os pontos da curva que denotam o normal funcionamento do sistema, funcionamento que requer cuidado (demandando, por exemplo, uma manutenção preventiva) e o ponto de falha do sistema (exigindo-se uma reparação ou substituição de componentes).

Entrando especificamente no assunto de prognósticos, pode-se falar que a disciplina de prognósticos é baseada em quatro pontos fundamentais (Uckun *et al.*, 2008):

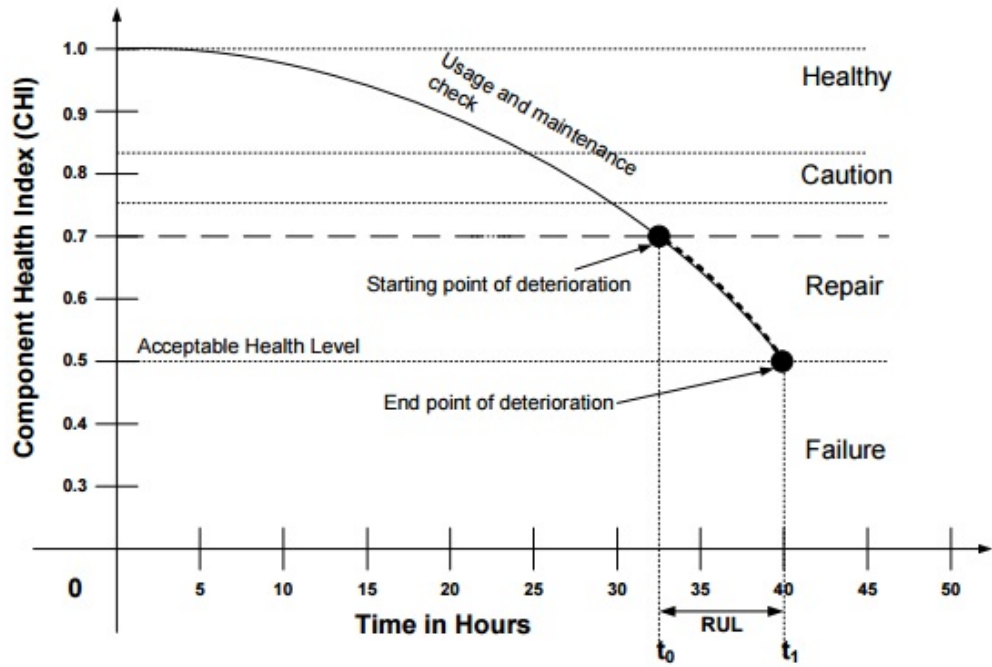


Figura 3-1: Índice de saúde do componente em função do tempo (Figura extraída de (Xiongzi *et al.*, 2011))

- O envelhecimento dos sistemas e componentes do sistema se dá através da ação do próprio uso, tempo de uso ou condições ambientais;
- pode-se associar o envelhecimento e a frequência de falhas a um processo monotônico relativo à degradação física e química do componente ou sistema;
- os sinais de envelhecimento são detectáveis em momentos anteriores à falha fatal do componente;
- é possível fazer a correlação entre os sinais de envelhecimento com um modelo de degradação do componente. Desta forma, pode-se estimar o seu RUL.

3.2 Classificação dos Sistemas de Prognósticos

Sistemas de prognósticos podem ser classificados em sistemas orientados a dados (Bolander *et al.*, 2010) e sistemas baseados em modelos (Schwabacher & Goebel,

2007; Tran *et al.*, 2009). Em ambos os casos, o método requer algum conhecimento inicial do domínio do problema sobre o modelo do sistema ou componente. Pode, ainda, requerer conhecimento na forma de informações sobre os casos e condições de ocorrências anteriores de falhas. O trabalho feito por (Muller *et al.*, 2008) adiciona um terceiro tipo: modelo baseado em experiência.

Os sistemas de prognósticos orientados a dados geralmente empregam técnicas de reconhecimento de padrões e aprendizagem de máquina para detectar alterações nos estados do sistema sob monitoramento ou mudanças nas variáveis monitoradas. Eles são recomendados quando não há compreensão adequada em relação à física por trás do funcionamento do sistema ou em relação aos princípios que levam o sistema às condições de falha.

Sistemas de prognósticos baseados em modelos, requerem a modelagem matemática do componente sob monitoramento. A geração do prognóstico é realizada em função da coerência entre os valores coletados dos componentes e as saídas do modelo matemático.

Por sua vez, modelos baseados em experiências utilizam modelos estatísticos baseados em experiências passadas. Devido ao fato de utilizarem apenas uma coletânea de dados estatísticos, os sistemas de prognósticos baseados nesta abordagem são de fácil implementação porém apresentam precisão inferior aos outros dois modelos.

Um trabalho mais recente, descrito em (Okoh *et al.*, 2014), sugere uma outra classificação dos sistemas conforme mostrada na Figura 3-2.

Em relação à figura 3-2, o autor faz uma diferenciação entre metodologia e técnicas utilizadas:

1. Metodologias:

- Baseada em Modelos: Esse modelo é utilizado nas técnicas que utilizam estatística ou inteligência computacional e consiste no uso de informações derivadas de configurações, históricos de utilização do componente ou do sistema. Em (Gasvik *et al.*, 2014), implementa-se uma decomposição do tipo *wavelet packet* e *Hidden Markov Models* (HMMs) para realizar o prognóstico do RUL.

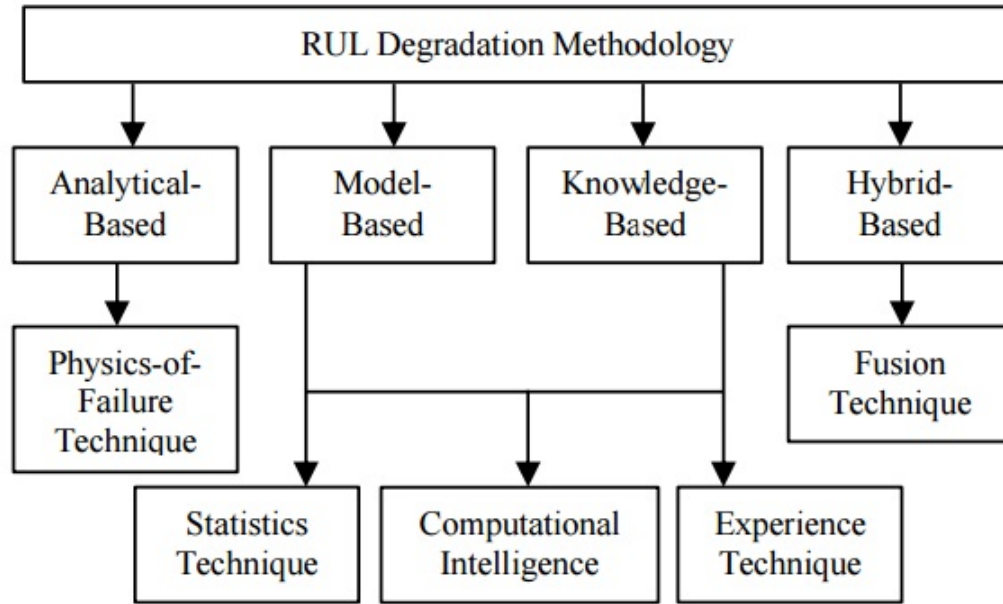


Figura 3-2: Classificação dos sistemas de prognósticos (Figura extraída de (Okoh *et al.*, 2014))

- Metodologia analítica: A metodologia analítica está associada à técnica de Física das Falhas e baseia-se na utilização de modelos físicos (Bolander *et al.*, 2010). Em (Coppe *et al.*, 2012), os autores propuseram o uso de um simples modelo de crescimento de rachadura para prever o RUL de um sistema submetido à fadiga. Eventos tais como falhas de rachaduras por fadiga, desgastes e corrosão dos componentes são descritas por leis matemáticas para que se possa realizar a estimativa do RUL (Medjaher *et al.*, 2012).

- Baseada em Conhecimento: O referido modelo objetiva o uso das técnicas por experiência e pelo uso da inteligência computacional. A abordagem baseada em conhecimento relaciona-se com a manipulação de informações recolhidas de especialistas e pela interpretação das regras associadas ao sistema ou componente (Chen *et al.*, 2012).

- Híbrida: Consiste no emprego de diversas metodologias e técnicas. O modelo híbrido usa dados paramétricos e não paramétricos para a geração dos prognósticos de RUL (Medjaher *et al.*, 2012).

2. Técnicas:

- Estatística: A técnica estatística faz referência ao emprego de técnicas estatísticas tais como *auto regressive moving average* (ARMA) e suavização exponencial para a geração de prognósticos (Cheng & Pecht, 2007) de modo que se possa reconhecer o comportamento dinâmico dos componentes (Tran *et al.*, 2012). Em (You *et al.*, 2010), os autores utilizaram *Proportional Hazard Model* (PHM) aplicando-o no campo de sistemas biomédicos.

- Por experiência: o conhecimento explícito e o tácito são obtido pelas observações feitas por especialistas. Tais observações permitem a geração de regras de modo a representar as condições que remetem às falhas (Si *et al.*, 2011).

- Usando inteligência computacional: A inteligência computacional emprega o uso de, por exemplo, redes neurais, sistemas *fuzzy*, *support vector machine* e redes bayesianas. Nessa técnica, as informações coletadas do meio, mesmo que escassas, possibilitam a formação de um conjunto de classificadores e estimadores para que os prognósticos possam ser gerados (Xiongzi *et al.*, 2011; Yang *et al.*, 2012).

- Física das Falhas (Physics-of-Failure - PoF): Essa técnica é permissível com o uso de dados paramétricos e abrange mecanismos como: *Continuum Damage Mechanisms*, *Linear Damage Rules*, *Non-linear Damage Curves* e *Two Stage Linearization*, *Crack Growth Concept* e *Energy-based Damage* (Fatemi & Yang, 1998).

- Fusão: A técnica de fusão consiste na agregação de diversos conjuntos de dados relativos ao componente. A classificação dos dados para a fusão pode ser auxiliada por algum método *fuzzy*. Sobre os dados fundidos, pode-se aplicar, por exemplo, análise do tipo *Principal Component Analysis* (PCA) (Maio *et al.*, 2011; Cheng & Pecht, 2009; Wei *et al.*, 2011) a fim de reduzir a dimensionalidade do vetor de atributos.

3.3 Aplicações dos Sistemas de Prognósticos

As aplicações industriais de sistemas de prognósticos são diversas. Como exemplos, pode-se citar as relacionadas aos processos fabris, indústrias automotivas, indústrias aeroespaciais e de companhias de geração e distribuição de energia. Nas companhias de geração e distribuição de energia, aplicações comerciais de SHM incluem máquinas equipadas com rotor (Heng *et al.*, 2009) e turbinas eólicas (Lau *et al.*, 2012). Aplicações de energia renováveis, como turbinas eólicas, também podem se beneficiar de sistemas SHM baseados na tecnologia de geração de prognósticos. Abordagens baseadas em métodos estatísticos e aprendizagem de máquina podem ser encontrados em (Kusiak & Li, 2011; Kusiak & Verma, 2011).

Em (Julka *et al.*, 2011), os autores propõem um conjunto de algoritmos de prognóstico orientadas a dados para a previsão robusta de RUL através da soma ponderada das saídas de cinco métodos diferentes: uma abordagem em *Similarity-Based Interpolation* (SBI) com *Relevance Vector Machine* (RVM) como técnica de regressão, SBI com *Support Vector Machine* (SVM), SBI com ajuste quadrático com mínimos quadrados, regressão linear bayesiana ajuste quadrático com mínimos quadrados e, por fim, uma abordagem baseada em rede neural recorrente (RNN).

Especificamente sobre o prognóstico falha em transformadores de potência, é possível encontrar alguns poucos recentes estudos na literatura, como os descritos em (Catterson, 2014; Zarei *et al.*, 2014). Em (Catterson, 2014), o autor utiliza *Bayesian Particle Filtering* para estimar o RUL do transformador considerando o grau de polimerização do material isolante e as medições de temperatura em pontos específicos do transformador (*hotspots*). Em (Zarei *et al.*, 2014), um sistema de inferência baseado em uma rede adaptativa neuro-fuzzy (*Adaptive Network-based Fuzzy Inference System* - ANFIS) é utilizada para inferir mudanças nas concentrações de gases dissolvidos. Uma ANFIS é definido para cada gás de tal forma que específicas regras fuzzy sejam aplicadas para a dinâmica de cada gás, considerando também o tempo de operação (idade) do transformador de potência.

3.4 Precisão e Acurácia dos Prognósticos

Para finalizar o presente capítulo, é conveniente sumarizar os termos precisão e acurácia (Saxena *et al.*, 2008). Salienta-se que o presente capítulo define o termo *acurácia* no contexto de prognósticos. A definição relativa à *machine learning* é realizada e utilizada no Capítulo 4. A Figura 3-3, extraída de (Uckun *et al.*, 2008), fornece uma ideia inicial sobre a diferença entre as métricas *precisão* e *acurácia*.

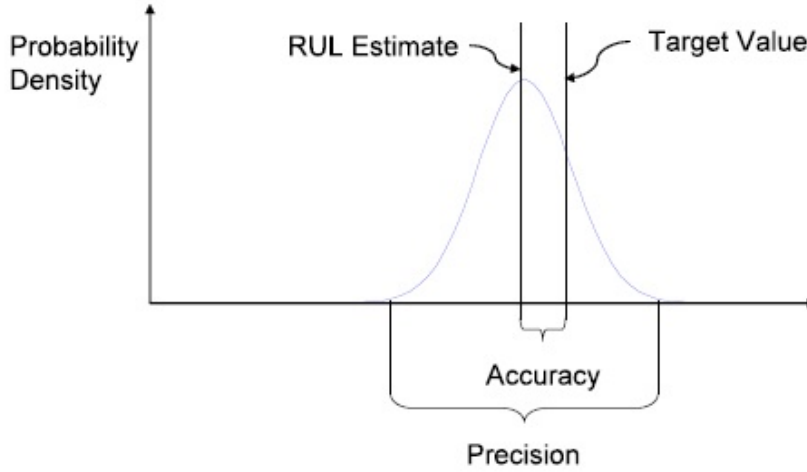


Figura 3-3: Diferenciação entre as métricas Precisão e Acurácia (Figura extraída de (Uckun *et al.*, 2008))

Como pode-se notar na Figura 3-3, a acurácia está relacionada com a diferença entre a estimativa do RUL e o valor real do tempo de falha:

$$\Delta_i = \mathbf{r}_*(i) - \mathbf{r}(i) \quad (3.1)$$

Onde $\mathbf{r}_*(i)$ representa o tempo real do RUL e $\mathbf{r}(i)$ relaciona-se com o tempo estimado em relação à falha de índice temporal i .

Por sua vez, a precisão pode ser obtida pelo desvio padrão das acurácias:

$$\mathbf{S}(i) = \sqrt{\frac{\sum_{l=1}^N (\Delta^l(i) - M)^2}{N - 1}} \quad (3.2)$$

Onde $\mathbf{M}$ representa a média dos erros das amostras.

3.5 Conclusões

Este capítulo apresentou conceitos inerentes à disciplina de prognósticos. Os conceitos apresentados neste capítulo serão utilizados no Capítulo 4 onde será tratada a aplicação desenvolvida para validar a arquitetura proposta neste projeto (descrita no Capítulo 2): um sistema para geração de prognósticos de falhas em transformadores de potência.

Capítulo 4

Prova de Conceito: Aplicação em Prognóstico de Falhas em Transformadores de Potência

Transformadores de potência são considerados elementos-chave para as concessionárias de energia elétrica (CEE). Quando esses equipamentos apresentam falhas, residências, indústrias e hospitais, por exemplo, são obrigados a lidar com impactos negativos decorrentes da falta de energia elétrica. Além disso, uma interrupção não programada no fornecimento de energia pode ser traduzida em pesadas multas para a CEE. Por isso, são necessárias ferramentas para diagnóstico, detecção de falhas e geração de prognósticos em relação às possíveis falhas. No contexto de transformadores de potência, vários estudos são notáveis em relação aos aspectos de proteção, monitoramento e diagnóstico como por exemplo os desenvolvidos por (Georgilakis, 2009; Singh & Bandyopadhyay, 2010; Georgilakis, 2011; Georgilakis & Kagiannas, 2014; Zhan *et al.*, 2015).

Por muitos anos, tarefas de manutenção preventiva em transformadores de potência consistiam de atividades de inspeção, testes e ações em intervalos periódicos de tempo - geralmente sugeridos pelos fabricantes ou determinados pela experiência prática. Também era comum a aplicação de testes e procedimentos de rotina, tais como: medição de perdas dielétricas, resistência de isolamento e resistência do enrolamento;

análise físico-química e análise cromatográfica do óleo isolante; monitoramento manual ou automático de temperatura (Duval, 2002). Essas análises possibilitavam aos operadores verificar se um determinado transformador estava operando normalmente ou se havia evidências de falhas térmicas e/ou elétricas, por exemplo. Estes tipos de falhas são decorrências de desgaste natural, ações ambientais ou sobrecargas, entre outras causas. Trabalhos de referência nesta área podem ser encontrados em (Duval, 2002; IEEE, 2008; IEC, 2008).

Entre os vários métodos de detecção de falhas, pode-se citar os baseados na medição das concentrações de gases dissolvidos no óleo isolante do transformador. Este procedimento é conhecido como detecção de falhas via Análise de Gases Dissolvidos (*Dissolved Gas Analysis* - DGA) (Duval, 2002). Normalmente, DGA pode ser realizada em dois modos: *off-line* e *on-line*. No modo *off-line*, o transformador de potência tem de ser desligado do sistema de alimentação e uma amostra de óleo é recolhida e levada ao laboratório onde será analisada através de uma técnica de cromatografia gasosa. No entanto, no modo *online*, o transformador de potência é mantido ligado ao sistema de energia e a DGA é realizada *in loco* em intervalos pré determinados de tempo (por exemplo, a cada duas horas). Neste caso, é utilizado, por exemplo, uma unidade compacta de cromatografia de gás em circuito fechado, que estará montada sobre ou perto do transformador monitorado. Técnicas como a análise óptica e cromatografia (Wan *et al.*, 2014; Arakelian, 2004; Goldstein *et al.*, 1979), sistemas eletro-químicos (Depari *et al.*, 2004; Zylka, 2005) e espectroscopia de absorção baseada em infra-vermelho (Dong *et al.*, 2012) também podem ser empregadas em DGA.

Um método para a classificação de falhas em transformadores de potência chamado *Método do Triângulo de Duval* (*Duval's Triangle Method* - DTM) é introduzido em (Duval, 2002) e descrito na IEEE C57.104 (IEEE, 2008) e na IEC 60599 (IEC, 2008) (Figura 4-1). DTM baseia-se nos índices de concentração de três gases dissolvidos no óleo isolante do transformador de potência: Etileno (C_2H_4), metano (CH_4) e acetileno (C_2H_2). A DGA por DTM envolve as percentagens das razões dos gases mencionados de acordo com as equações (1)-(3), apresentadas em uma forma gráfica. Os três lados

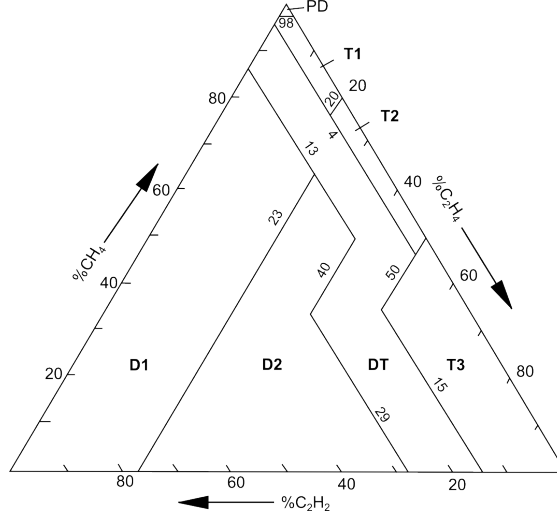


Figura 4-1: Método do triângulo de Duval

do triângulo, em coordenadas cartesianas X, Y e Z representam a proporção relativa de CH_4 , C_2H_4 e C_2H_2 , de 0-100 % para cada gás. Os cruzamentos de todas as três relações dos gases indicam o tipo de falha apresentado pelo transformador. Todos os tipos de falhas no triângulo de Duval podem ser encontrado na Tabela 4.1.

Nas Equações (1)-(3), x representa a concentração de CH_4 em ppm, y é a concentração de C_2H_4 em ppm e, por último, z denota a concentração de C_2H_2 em ppm.

$$CH_4(\%) = 100 \frac{x}{(x + y + z)} \quad (4.1)$$

$$C_2H_4(\%) = 100 \frac{y}{(x + y + z)} \quad (4.2)$$

$$C_2H_2(\%) = 100 \frac{z}{(x + y + z)} \quad (4.3)$$

Por consequência da não existência de modelos matemáticos eficientes para descrever a relação entre a taxa de alteração das concentrações dos gases e a possível convergência ao estado de falhas dos transformadores de potência e por consequência, também, que o processo de coleta de dados históricos é uma prática comum hoje em dia, vários trabalhos estão sendo desenvolvidos no que tange à detecção de falhas dos transformadores de potência. Dentre os quais, podem ser destacados: classificador

Tabela 4.1: Falhas descritas pelo triângulo de Duval (DTM)

Nomenclatura	Tipo de Falha
PD	Descarga parcial
D1	Descarga de baixa energia
D2	Descarga de alta energia
DT	Mistura de falha térmica e elétrica
T1	Falha térmica com $T < 300$ °C
T2	Falha térmica com $300 < T < 700$ °C
T3	Falha térmica com $T > 700$ °C

de padrões baseado em *Support Vector Machines* (Liao *et al.*, 2010; Fei & Sun, 2008; Cho *et al.*, 2006), abordagens usando a lógica Fuzzy (Huang & Sun, 2013), modelos neuro-fuzzy (Rajabimendi & Dadios, 2012; Naresh *et al.*, 2008), *Wavelets* (Pan *et al.*, 2008; Chen *et al.*, 2009), redes neurais (Seifeddine *et al.*, 2012; Castanheira *et al.*, 2011a), metodologias estocásticas baseadas em rede de Petri (Georgilakis *et al.*, 2006), classificador probabilístico baseado em otimização por enxame de partículas (Tang *et al.*, 2008), Árvores de Decisão (Zhao & Su, 2008; Lai & Blackburn, 2008; Castanheira *et al.*, 2011b) e de programação genética (Shintemirov *et al.*, 2009). Mais recentemente, o próprio Duval e sua equipe publicaram um novo método de detecção de falhas baseado na aplicação de uma combinação de razão gás (Kim *et al.*, 2013).

Os trabalhos previamente mencionados apresentam, pelo menos, dois pontos em comum:

- A grande maioria dos trabalhos dentro desta área refere-se à DGA realizada no modo *offline*.
- Os trabalhos focam exclusivamente em diagnósticos de falhas em transformadores de potência. Prognósticos de falhas não são o foco principal desses trabalhos por se tratar de uma estratégia relativamente nova. Projetos descritos em (Ozgur-Unluakin & Bilgiç, 2006; Brescia *et al.*, 2009; Zhou *et al.*, 2013; Catterson, 2014; Zarei *et al.*, 2014) são alguns poucos trabalhos encontrados nessa área particular de prognósticos de falhas em transformadores de potência.

4.1 Implementação do Sistema de Prognósticos de Falhas de Transformadores de Potência

Em função da lacuna apresentada pela falta de sistemas de prognósticos de transformadores de potência e, também, em função de validar a arquitetura modelada, é apresentado um novo sistema computacional capaz de realizar prognóstico de falhas em transformadores de potência (Sica *et al.*, 2015). A abordagem deste sistema de prognóstico de falhas (*Fault Prognosis System* - FPS) é baseada em um sistema cognitivo híbrido formado por mecanismos conexionistas e evolutivos. Os testes foram baseados na utilização de bases de dados reais de cromatografia de transformadores de potência encontradas na literatura (Morais & Rolim, 2006). Ao invés de informar somente o estado atual do transformador, classificando-o, por exemplo, em seus estados de *Condição Normal* (NC), *Falha Elétrica* (EF), *Falha térmica* (TF), essa ferramenta de prognóstico de falha estimará, a partir de uma determinada leitura da concentração dos gases dissolvidos, a suscetibilidade do transformador sob análise apresentar, por exemplo, uma falha térmica em um certo período de tempo e como a falha poderá evoluir ao longo do tempo. Essa ferramenta é de interesse da CEE, pois pode tornar-se uma ferramenta essencial para a manutenção preventiva dos transformadores (*Transformer Predictive Maintenance* - TPM). TPM pode ser visto como uma estratégia de manutenção que proporciona a detecção precoce de possíveis falhas do transformador e é capaz de reconhecer condições que conduzem a defeitos. Idealmente, TPM deve proporcionar a diminuição da periodicidade das manutenções corretivas ou reativas.

Apenas para ressaltar, pode-se dizer que as principais contribuições inerentes à implementação desse sistema são:

- Proposição de um sistema inteligente *online* de prognóstico de falhas para transformadores de energia que é capaz de prever o momento e o tipo de falha. O sistema proposto pode ajudar o planejamento das manutenções preventivas, oferecendo, portanto, uma solução de baixo custo e altamente flexível para o prognóstico de falhas. Além disso, o sistema de prognóstico aqui proposto ma-

nipula as concentrações de gases simultaneamente, onde todas as concentrações são utilizadas, simultaneamente, para o fornecimento dos prognósticos.

- Validação da arquitetura genérica para a implementação de sistemas baseados em conhecimento.

Esta seção tratará os pontos específicos da implementação da arquitetura apresentada no Capítulo 2 inerentes ao sistema de prognósticos de falhas de transformadores de potência.

Inicialmente, para melhor ilustrar a implementação do sistema de prognóstico utilizando a arquitetura proposta, a Figura 4-2 mostra as interações dos subsistemas e suas funcionalidades. Os números na figura indicam a sequência de ações durante a utilização do sistema, como se segue:

1. O *Ontology Core & Memory* provê ao *Connectionist Layer*, as configurações da rede neural assim como os pesos sinápticos obtidos em decorrência da fase de treinamento da rede. É importante ressaltar que o treinamento da rede ocorre de forma *offline* por intermédio da utilização de exemplos obtidos da própria companhia ou bases de dados de transformadores disponíveis na comunidade acadêmica ou de centros de pesquisa. Estes dados são utilizados para treinar a rede neural na tarefa de reconhecimento de condições normais e de falha a partir dos valores de concentração de gás.
2. O *Ontology Core & Memory* provê ao *Symbolic Layer* o modelo de regressão das variações das concentrações dos gases em função do tempo. O modelo de regressão pode ser construído a partir de análises efetuadas na literatura especializada ou a partir de observações construídas pela empresa de energia. Assim, o uso deste sistema não se torna inviável no caso de companhias que não possuem dados históricos. Além disso, outros modelos de previsão de variações de gases podem ser facilmente adaptados ao sistema. O modelo de regressão utilizado para a implementação do sistema de prognóstico é explicado na Seção 4.1.1.

3. As concentrações amostradas dos gases são enviadas aos *Sensory System* através do *Sensor Device Interface* que recebe as informações de uma unidade de cromatografia a qual poderá atuar de forma *online*. A partir do recebimento das concentrações, dá-se início ao processamento de modo a propiciar a geração dos prognósticos. No caso específico desta implementação, foram tratadas as concentrações dos seguintes gases: H_2 (dihidrogênio), C_2H_4 (etileno), C_2H_2 (acetileno), CH_4 (metano), C_2H_6 (etano).
4. As informações acerca das concentrações dos gases, após a decodificação pelo *Sensory System*, são passadas ao *Symbolic Layer*, fazendo-se iniciar o algoritmo evolucionário descrito na Seção 4.1.3.
5. Para cada iteração do algoritmo evolucionário descrito na Seção 4.1.3, a *Connectionist Layer* é acionada para realizar a análise e classificação dos descendentes gerados pelo modelo evolucionário utilizado pelo referido algoritmo. Para cada iteração, ainda, as concentrações dos gases são modificadas pela utilização do modelo de regressão alimentado ao *Symbolic Layer* pelo *Ontology Core & Memory*.
6. Após a conclusão do algoritmo evolucionário, o resultado do prognóstico é gerado.

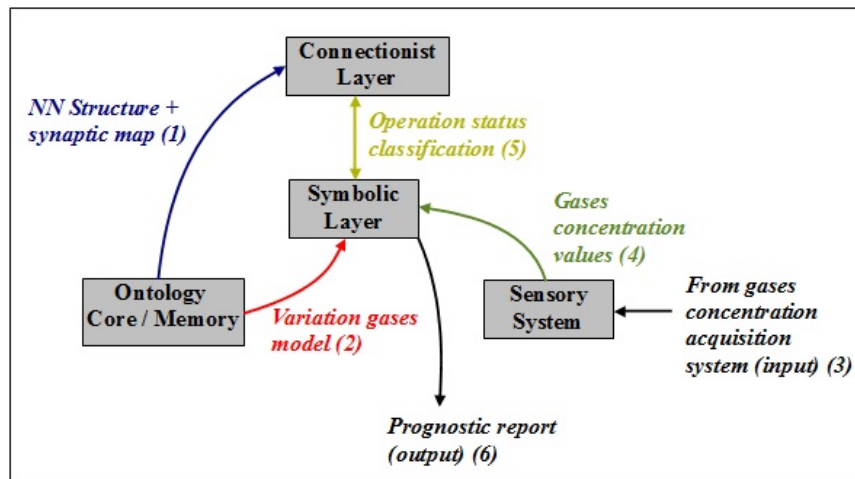


Figura 4-2: Etapas para a geração dos prognósticos

Para se ter uma melhor noção da implementação do sistema de prognóstico sobre a arquitetura proposta descrita na Seção 2.5, após as etapas definidas na Figura 4-2, os módulos funcionais da arquitetura ficam instanciados de acordo o explicitado na Figura 4-3.

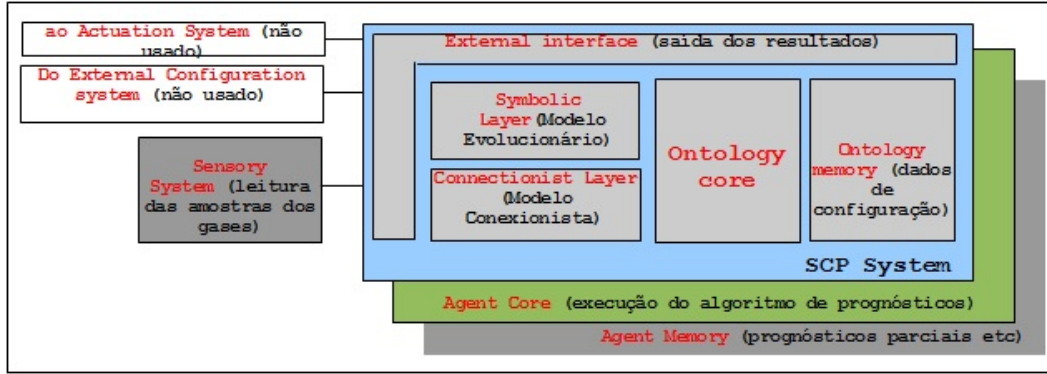


Figura 4-3: Instanciamento dos módulos da arquitetura proposta na implementação do sistema de prognósticos

A Tabela 4.2 sumariza as funcionalidades incorporadas por cada módulo presente da arquitetura instanciada da Figura 4-3.

Tabela 4.2: Funcionalidades incorporadas pelos módulos da aquitetura

Módulo	Funcionalidade
Symbolic Layer	Executa o algoritmo evolucionário descrito na Seção 4.1.3.
Connectionist Layer	Responsável por abrigar os mecanismos inerentes ao <i>ensemble learning</i> e, consequentemente, as GRNN conforme modelo descrito na Seção 4.1.2.
Ontology Core	Apenas realiza o interfaceamento com a memória da ontologia para a captura e transferências dos valores de configuração do sistema.
Ontology Memory	Armazena as informações de configurações das GRNN obtidas a partir da etapa de treinamento - etapa realizada por um sistema à parte.
Agent Core	Contém o código para a geração dos prognósticos, gerenciando a ativação dos mecanismo evolucionário, coleta das amostras, escrita dos resultados nos arquivos de saída dos prognósticos.
Agent Memory	Memória utilizada pelo <i>Agent Core</i> para a realização do prognóstico.
External Interface	Responsável pela escrita dos resultados referentes aos prognósticos gerados.
Sensory System	Coleta das amostras dos gases para que, a partir das mesmas, seja iniciada a geração do prognóstico.

4.1.1 Modelo de Regressão para a Variação dos Gases

Para que seja possível a geração do prognóstico, o sistema implementado, em seu módulo evolucionário, conta com um modelo de regressão para a variação dos gases. O modelo, construído a partir de estudos realizados em (Sloat *et al.*, 1967), objetiva fornecer as novas possíveis concentrações dos gases em função do tempo. As curvas obtidas pelo processo de regressão são mostradas na Figura 4-4.

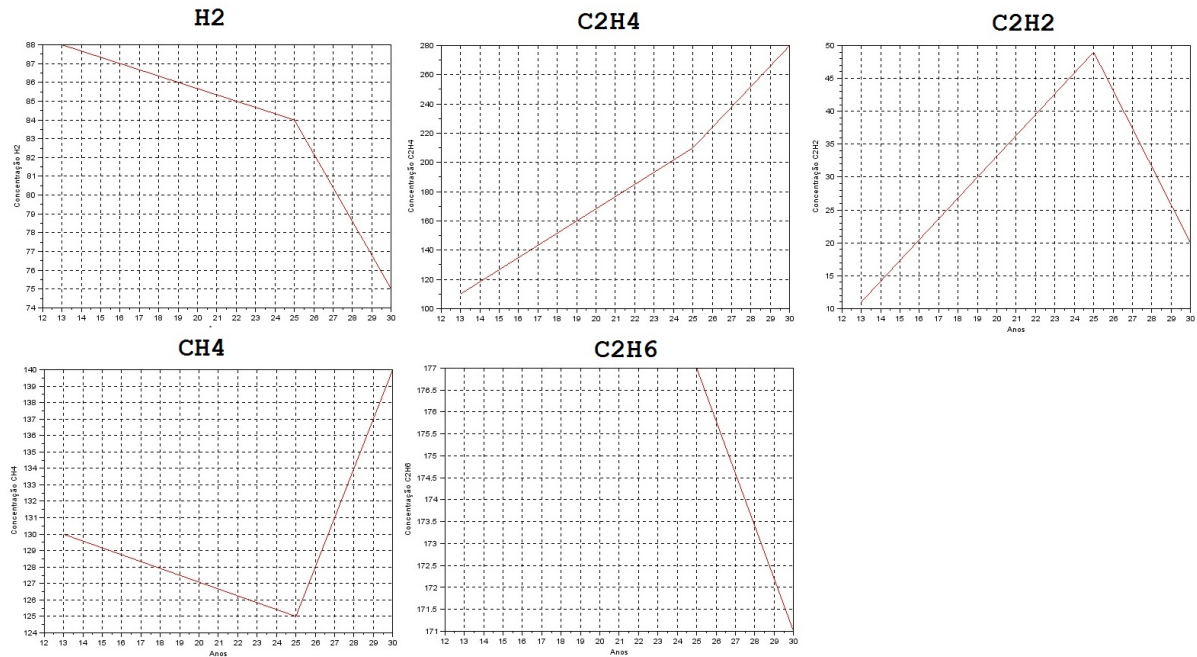


Figura 4-4: Variações das concentrações dos gases em função do tempo

Devido à ausência de informações acerca da idade do transformador e de seu regime de trabalho (por exemplo, operação sob *stress*), o mecanismo evolucionário, para a geração de uma população, aplica equações derivadas das curvas exibidas na Figura 4-4 em pontos aleatórios. Desta forma, devido à característica estocástica, optou-se, por suavizar as curvas que denotam as variações dos gases para que atenuie o impacto de se escolher pontos muito distintos às reais condições e idade do transformador.

4.1.2 Modelo Conexionista Utilizado

Como descrito anteriormente, a *Connectionist Layer* integra a camada denominada *Storage, Coding and Processing System (SCP)*. Apenas lembrando que a camada conexionista é responsável pela funcionalidade reativa do sistema e, também, serve de base para o processamento deliberativo, ou seja, a referida camada, além de possibilitar a geração de prognósticos, atua, também, como um sistema de detecção instantânea de falhas.

Para a implementação da *Connectionist Layer*, foi utilizado um modelo baseado em *ensemble learning*. Em tal modelo, cada estado de funcionamento do transformador de potência (*Condição Normal* (NC), *Falha Elétrica* (EF), *Falha Térmica* (TF)) é abstraída por um grupo específico de redes neurais. Na versão atual de implementação do sistema, foi implementado o modelo *General Regression Neural Network* (GRNN), originalmente proposto em (Specht, 1991).

A GRNN pertence à classe de redes neurais probabilísticas. Sua principal vantagem é que exige menos amostras para a etapa de treinamento em relação a, por exemplo, uma rede neural do tipo *back-propagation*. A previsão da GRNN para uma dada entrada $\mathbf{x}$ é dada por:

$$g(\mathbf{x}) = \frac{\sum_{i=1}^n y_i \exp \left(-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2} \right)}{\sum_{i=1}^n \exp \left(-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2} \right)} \quad (4.4)$$

No modelo GRNN, $\mathbf{x}$ representa uma entrada, as amostras para o treinamento da rede são representados por $\{\mathbf{x}_i; y_i\}$. Quando a entrada tiver o seu valor próximo ao valor da amostra $\mathbf{x}_j$, o peso da predição y_j é próximo a 1 (um) e a contribuição das outras amostras de treinamento torna-se menor. Com este modelo, é possível prever o comportamento dos sistemas com poucas amostras e interpolar a curva entre as amostras de treinamento. Desta forma, o parâmetro de diferença σ é calculado a cada iteração do método (Specht, 1991).

Os métodos baseados em *ensemble learning* empregam um número finito de diferentes classificadores para que, em conjunto, consigam alcançar um melhor desem-

penho preditivo. O modelo denominado como *bootstrap aggregation* foi adotado para a classificação em conjunto (Breiman, 1996). Nesta abordagem, cada classificador envolvido na votação apresenta um peso igual. Dessa forma, a saída do sistema é a média aritmética simples da saída de cada classificador. Na implementação do sistema de prognósticos desenvolvido, há um conjunto de N redes neurais treinadas para reconhecer falha elétrica, um outro conjunto de N redes neurais treinadas para reconhecer falha térmica e outro conjunto, também de N redes neurais para a operação normal. A união destes três conjuntos formam a camada connexionista. Sendo assim, tem-se:

$$E^{NC}(\mathbf{x}) = \frac{1}{N} \sum_{s=1}^N g_s^{NC}(\mathbf{x}) \quad (4.5)$$

$$E^{TF}(\mathbf{x}) = \frac{1}{N} \sum_{s=1}^N g_s^{TF}(\mathbf{x}) \quad (4.6)$$

$$E^{EF}(\mathbf{x}) = \frac{1}{N} \sum_{s=1}^N g_s^{EF}(\mathbf{x}) \quad (4.7)$$

Onde:

- E^{NC} representa a saída do conjunto associado à operação normal.
- E^{TF} representa a saída do conjunto associado à falha térmica.
- E^{EF} representa a saída do conjunto associado à falha elétrica.

No sistema desenvolvido para geração de prognósticos falhas em transformadores de potência, considerando a coleta por meio de análise de gases dissolvidos (DGA), a entrada $\mathbf{x}$ é composta pelas concentrações dos gases H_2 , C_2H_4 , C_2H_2 , CH_4 , C_2H_6 e é atribuído o valor 1 à saída correspondente à classe associada à amostra (NC, TF ou EF) e atribuído o valor 0 (zero) às demais classes.

4.1.3 Modelo evolucionário utilizado

Também como já descrito, a camada denominada como *Symbolic Layer* implementa um mecanismo evolucionário para a geração de candidatos que possam convergir para

as situações de falha. Tal convergência possibilita à camada connexionista a detecção de falhas em candidatos gerados a cada iteração. Cada iteração, por sua vez, está associada a um intervalo de tempo. O algoritmo evolutivo é baseado em estratégias de evolução natural (Back, 1996). Vale ressaltar que a geração de prognósticos é um processo dinâmico onde previsões mudam com o tempo devido à incorporação de novos dados de observação. A integração dos modelos connexionista e evolucionário é a base do sistema de prognóstico desenvolvido.

A população inicial do mecanismo evolucionário é gerada a partir da amostra atual das concentrações de gás coletadas pelo *Sensory System* e, sobre essa amostra, é aplicado, para a geração de cada indivíduo, um modelo de regressão da variação dos gases ao longo do tempo. O modelo de regressão da dinâmica dessas concentrações foi desenvolvido a partir de estudos realizados por (Sloat *et al.*, 1967). O modelo de regressão tenta prever a próxima configuração de gases com base na atual. Portanto, dada a amostra atual de concentrações de gás, uma população de μ indivíduos é gerada pela aplicação do modelo de regressão, proporcionando, então, possíveis cenários futuros para as referidas concentrações. O algoritmo é detalhado a seguir:

1. Dada a amostra atual das concentrações dos gases, obtidas por algum método de DGA, gera-se uma população inicial $\{\mathbf{p}_{t,k}\}$, $t = 1, \dots, T_{\max}$, $k = 1, \dots, \mu$, com μ indivíduos gerados pela aplicação do modelo de regressão das variações das concentrações dos gases em função do tempo.
2. Classificam-se os μ indivíduos $\{\mathbf{p}_{t,k}\}$ por intermédio das redes neurais localizadas na camada connexionista em termo de operação normal, falha térmica ou falha elétrica. O valor de *fitness* dos indivíduos corresponde ao máximo valor entre cada conjunto de rede neural. Os valores externados pelos conjuntos de redes neurais estão dentro do intervalo de zero a um. A classificação é determinada pelo maior valor de saída:

$$\phi(\mathbf{p}_{t,k}) = \max [E^{NC}(\mathbf{p}_{t,k}), E^{TF}(\mathbf{p}_{t,k}), E^{EF}(\mathbf{p}_{t,k})]$$

3. Gera-se λ descendentes $\{\mathbf{o}_{t,j}\}$, $j = 1, \dots, \lambda$ selecionando os indivíduos através de torneio binário e aplicando-se recombinação e mutação gaussiana aos seus

valores. Os melhores indivíduos têm maior probabilidade de serem selecionados para a reprodução, portanto, produzirem descendentes que melhor convirjam para estados de falhas.

4. Classificam-se os λ indivíduos, $\{\mathbf{o}_{t,j}\}$, por intermédio das redes neurais localizadas na camada connexionista em termo de operação normal, falha térmica ou falha elétrica. O valor de *fitness* dos indivíduos corresponde ao máximo valor entre cada conjunto de rede neural. Os valores externados pelos conjuntos de redes neurais estão dentro do intervalo de zero a um, sendo que valores mais próximos ao valor um correspondem à maior aproximação para uma condição específica (condição normal, condição de falha elétrica ou condição de falha térmica).
5. Aplica-se a seleção $ES(\mu + \lambda)$, ou seja, seleciona-se os μ melhores indivíduos baseando-se nos valores de *fitness*. Esses indivíduos serão os pais da população da próxima iteração.
6. Aplica-se o modelo de regressão das variações das concentrações dos gases em função do tempo sobre os μ indivíduos pais de modo a realizar a predição do possível próximo estado.
7. Retorna-se ao passo 2 até o método convergir.

Nota-se, pelo algoritmo apresentado, que a abordagem evolucionária, aliada ao modelo de regressão de alteração das concentrações dos gases, proporciona a evolução da população a fim de permitir avaliações frente aos novos cenários criados em manifestação ao avanço do tempo. Portanto, o modelo adotado por esse sistema de prognósticos é capaz de selecionar e reproduzir esses cenários futuros que são susceptíveis de levar a uma condição de reconhecimento de falha. Sendo assim, o prognóstico é feito ao longo do processo evolucionário. O método converge quando toda a população é classificada como uma falha ou quando o número máximo de iterações é alcançado.

Cada iteração do algoritmo evolucionário corresponde a um ano. Essa correspondência temporal permite a correlação entre o modelo computacional desenvolvido e o próprio prognóstico do transformador de potência. Quando toda a população é classificada como uma falha, o método pára e o número de iterações executadas corresponde ao número de anos que uma falha (elétrica ou térmica) está prevista a ocorrer (em outras palavras, representa o próprio tempo remanescente de vida - *Remaining Useful Life* ou RUL). Caso contrário, o método pára ao atingir o número máximo de iterações (T_{max}) sem ter toda a população classificada como falha. No último caso, então, o prognóstico corresponde a um determinado estado com base na aptidão média da população para cada conjunto. Essa aptidão média denota a perspectiva da proporção de indivíduos da população que podem ser classificados como falha ou funcionamento normal:

$$r^{NC}(t) = \frac{1}{\mu} \sum_{k=1}^{\mu} E^{NC}(\mathbf{p}_{t,k}) \quad (4.8)$$

$$r^{TF}(t) = \frac{1}{\mu} \sum_{k=1}^{\mu} E^{TF}(\mathbf{p}_{t,k}) \quad (4.9)$$

$$r^{EF}(t) = \frac{1}{\mu} \sum_{k=1}^{\mu} E^{EF}(\mathbf{p}_{t,k}) \quad (4.10)$$

O prognóstico é considerado como condição normal (*Normal condition* - NC) dentro do intervalo de tempo t' (em anos) caso corresponda à condição:

$$(r^{NC}(t') > r^{TF}(t')) \wedge (r^{NC}(t') > r^{EF}(t')) \wedge (r^{NC}(t') > 0.5)$$

O prognóstico é considerado como falha térmica (*Thermal Fault* - TF) dentro do intervalo de tempo t' (em anos) caso corresponda à condição:

$$(r^{TF}(t') > r^{NC}(t')) \wedge (r^{TF}(t') > r^{EF}(t')) \wedge (r^{TF}(t') > 0.5)$$

O prognóstico é considerado como falha elétrica (*Electrical Fault* - EF) dentro do intervalo de tempo t' (em anos) caso corresponda à condição:

$$(r^{EF}(t') > r^{NC}(t')) \wedge (r^{EF}(t') > r^{TF}(t')) \wedge (r^{EF}(t') > 0.5)$$

As expressões que denotam os prognósticos das condições NC, TF ou EF podem ser sumariamente explicadas como sendo a condição que apresentar o maior número de indivíduos r que caracterizam uma determinada condição desde que o número destes indivíduos represente pelo menos 50% da população no tempo t' .

4.2 Resultados

4.2.1 Dados experimentais

As redes neurais integrantes da camada conexionista foram treinadas utilizando-se a base de dados IEC TC 10 a partir do trabalho de Duval conforme citado em (Duval & de Pablo, 2001). Essa base consiste de 73 instâncias de falhas elétricas, 34 instâncias de falhas térmicas e 48 de operação normal.

A obtenção de dados para testar e validar o sistema proposto de prognósticos não é uma tarefa fácil pois as CEEs não costumam rotular os dados de medição de transformadores de potência com a data em que foi coletado. No entanto, foram utilizadas informações disponíveis a partir de (Moraes & Rolim, 2006) com dados fornecidos por uma concessionária de energia elétrica que realiza o acompanhamento de seus transformadores através da leitura e registro das concentrações dos gases assim como as respectivas datas de realização das amostragens. Dois casos de teste são sumarizados na Tabela 4.3. No primeiro caso de teste, através da análise dos dados, existe uma falha térmica depois de 4 anos e 5 meses (4,42 anos). No segundo caso, há uma falha elétrica depois de 6 anos e 5 meses (6,42 anos).

Tabela 4.3: Casos de testes

caso	H ₂	CH ₄	C ₂ H ₄	C ₂ H ₆	C ₂ H ₂	op	data
1	29	110	53	290	10	N	02/03/1983
	26	51	52	530	5	N	17/08/1984
	64	180	65	770	5	N	10/10/1986
	620	2200	2000	1100	13	T	05/08/1987
2	29	02	16	1	0.4	N	11/01/1983
	27	5	12	2	0.4	N	13/02/1984
	18	3	80	0.4	0.4	E	08/06/1989

op=operação; N=normal; T=falha Térmica; E=falha Elétrica

4.2.2 Configuração dos Experimentos

O sistema de prognósticos foi implementado em um computador do tipo *PC-based* usando compilador GNU C++ . A Tabela 4.4 detalha as características do ambiente usado para a implementação e testes.

Tabela 4.4: Ambiente de Teste

Item	Descrição
Processador	Intel Core i3-380M
Memória	3GB DDR3
Frequência de Clock	2.53 GHz
Sistema Operacional	CygWin emulado sobre Windows 7
Compilador C	GNU C++ vs 4.5.3
Análise matemática	Scilab 5.4.1

4.2.3 Resultados

O sistema de prognósticos foi aplicado aos casos de teste sumarizados na tabela 4.3. Para tanto, foi utilizada a primeira amostragem de cada caso de teste para a geração dos prognósticos. Com base na amostra inicial, o sistema de prognóstico é iniciado. A camada simbólica gera uma população cujos indivíduos são classificados, pela camada conexionista, como operação normal, falha térmica ou falha elétrica. Após o processo evolucionário convergir, obtém-se o tipo de falha e a previsão de quando se espera que ocorra. Uma vez que o método é estocástico, esse processo foi repetido por 120 vezes para cada caso gerando-se valores, em anos, diferentes para a vida útil remanescente (RUL). Os parâmetros utilizados no experimento foram: $\mu = 10$, $\lambda = 100$, $N = 5$, $T_{max} = 15$. Esses parâmetros foram escolhidos pois, dentre outras combinações testadas, apresentaram os melhores índices de acurácia. Optou-se pelo tempo $T_{max} = 15$ pois representa uma medida próxima ao dobro do tempo visualizado nos casos tomados para teste. Os resultados estão sumarizados na Figura 4-5, que mostra o *boxplot* do RUL para todas as execuções independentes realizadas em cada caso de teste.

Como pode-se notar, o sistema converge corretamente para uma estimativa que é, em média, muito perto do tempo real de ocorrência da falha. O RUL previsto para o

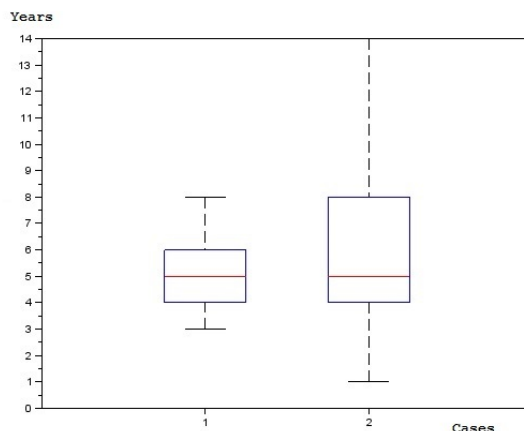


Figura 4-5: *Box plot* do RUL encontrados nas 120 execuções independentes.

caso 1 é $4,75 \pm 1,18$, enquanto o RUL previsto para o caso 2 é $6,125 \pm 2,88$.

As Figuras 4-6 e 4-7 exibem os histogramas e as curvas de distribuição estatística gama dos resultados para cada caso. Os histogramas representam as distribuições empíricas dos tempos (anos) gerados como prognósticos. Nota-se, a partir das Figuras 4-6 e 4-7, que o ponto de máximo em cada histograma denota o tempo gerado como prognóstico, ou seja, 4.75 para o primeiro caso de teste e 6.125 para o segundo caso. Os parâmetros $(\alpha; \beta)$ utilizados para a distribuição gama para o caso 1 e para o caso 2 foram $(9.794; 0.443)$ e $(6.164; 0.862)$, respectivamente.

Por fim, foi realizado um experimento adicional no qual o estado inicial é selecionado aleatoriamente a partir de uma das entradas na Tabela 4.3. Em seguida, o método é aplicado com base neste estado inicial, atingindo assim um prognóstico. O RUL obtido é comparado com o tempo real para a falha e o erro é calculado. Após 120 execuções, foram obtidos os seguintes resultados: 20% das execuções correspondem ao tempo exato de falha, 33,33% das execuções correspondem a um erro igual a um ano, 31,67% das execuções correspondem a um erro igual a dois anos e, por último, 15% das execuções correspondem a um erro igual ou superior a três anos. Estes resultados sugerem que o método aqui proposto e desenvolvido é confiável para esta aplicação.

Além disso, o custo computacional é baixo. Uma execução independente do sis-

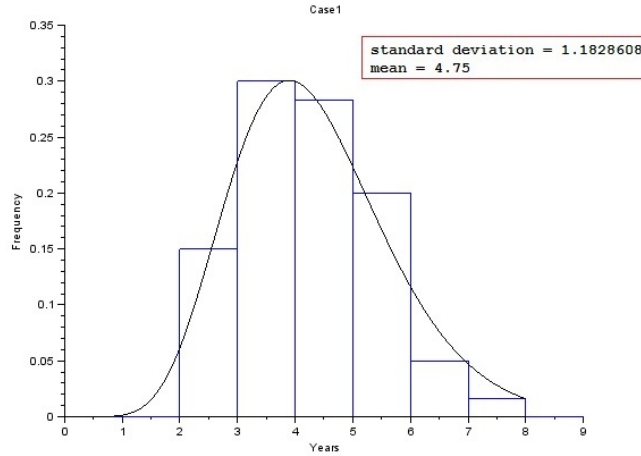


Figura 4-6: Histograma e curva distribuição gama do RUL para o Caso de Teste 1 frente às 120 execuções independentes.

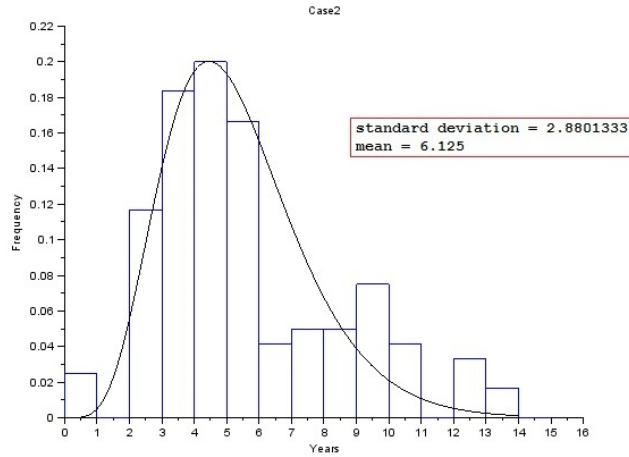


Figura 4-7: Histograma e curva de distribuição gama do RUL para o Caso de Teste 2 frente às 120 execuções independentes.

tema proposto para a geração de prognóstico de falhas leva, em média, 6 ms para ser completado utilizando-se o ambiente de teste descrito na Tabela 4.4. Portanto, o resultado torna-se disponível assim que uma amostra de concentração de gases é fornecida. Claro que o tempo de execução depende dos parâmetros do algoritmo evolucionário μ , λ e T_{max} e número de redes neurais utilizadas em cada conjunto do *ensemble learning*.

Para dar uma idéia melhor sobre a precisão do método, a Tabela 4.5 apresenta a

matriz de confusão para os casos Caso 1 e Caso 2. Considera-se que a margem de erro aceitável para o prognóstico seja de um ano ($\delta = 1$).

Tabela 4.5: Matriz de Confusão

Case	TP	FP	FN	TN
Case 1	50.00%	2.50%	30.83%	16.67%
Case 2	53.33%	45.00%	1.67%	0.00%

A partir da Tabela 4.5, considerando $\delta = 1$ and $\Delta = 5$, RT representando o tempo real da falha e PT denotando o tempo obtido pelo sistema de prognóstico, tem-se:

- True Positive (TP): quando $RT - \delta \leq PT \leq RT + \delta$, a predição está dentro da margem de erro esperada.
- False Positive (FP): quando $PT < RT - \delta$, a falha real ocorreu após o tempo estimado pelo sistema de prognósticos.
- False Negative (FN): quando $RT + \delta \leq PT \leq RT + \Delta$, a falha real ocorreu antes do tempo estimado pelo sistema de prognósticos.
- True Negative (TN): quando $PT > RT + \Delta$, o falha real ocorreu muito antes do que o tempo estimado pelo sistema de prognósticos.

FP de ações de manutenção preditiva enquanto FN provavelmente levará à necessidade de manutenção corretiva ou reativa uma vez que a falha não foi prevista corretamente. Neste sentido, FP são menos críticos do que FN. Sendo assim, como a maioria dos prognósticos gerados é classificada como RT ou PT , o sistema atende plenamente às expectativas uma vez que as falhas supostamente ocorreriam após a indicação do tempo gerado pelo sistema de prognóstico.

A partir da matriz de confusão descrita pela Tabela 4.5, pode-se estabelecer a acurácia, no contexto de aprendizagem de máquina, obtida através da relação:

$$ACC = \frac{TP + TN}{P + N}.$$

A Figura 4-8 mostra a acurácia em relação à proximidade do tempo de falha. Pode-se concluir, portanto, através das observações das experiências realizadas, que, quando

o estado inicial for mais perto do momento real de ocorrência da falha, o sistema proposto é mais confiável.



Figura 4-8: Acurácia do sistema de prognósticos em relação ao tempo transcorrido entre o estado inicial e o estado de falha.

Por fim, pela medição das acurácias em função do tempo apresentada na Figura 4-8, pode-se mencionar que, por exemplo, a partir de 3 anos antes da ocorrência da falha, o sistema apresenta uma acurácia de 100%. Essa faixa de tempo é suficiente para uma efetiva programação das equipes de manutenção preventiva de modo a realizar as devidas intervenções no sistema elétrico a ponto de atenuar transtornos decorrentes das falhas dos transformadores de potência.

4.3 Conclusões

Esse capítulo apresentou os conceitos relacionados a falhas de transformadores de potência assim como os aspectos de implementação de um sistema para a geração de prognósticos de suas possíveis falhas. Para tanto, foi utilizada a arquitetura proposta na Seção 2.5.

No *ensemble learning*, foram utilizadas apenas GRNN utilizando-se conjuntos distintos de dados de treinamento para facilitar a implementação da versão inicial do

sistema de prognóstico e teste do agregador dos resultados de cada GRNN. Futuramente, serão testados outros classificadores para atuarem em conjunto com as GRNN de modo a se tentar alcançar resultados mais precisos.

Como pode ser notado nos resultados apresentados na Seção 4.2.3, o sistema apresentou resultados altamente satisfatórios na geração dos prognósticos de falhas em transformadores de potência pois o índice de acurácia é de 100% para intervalos de tempo de aproximadamente 3 anos (período satisfatório para o planejamento das manutenções preventivas). Para a geração dos prognósticos de falhas, o isolamento ou não do transformador frente ao sistema elétrico é determinado pelo equipamento de DGA. Cabe ao sistema aqui implementado apenas coletar as informações exportadas pelo equipamento de DGA para realizar a geração dos prognósticos.

O sistema implementado poderá ser migrado para a implementação usando as abordagens inerentes aos sistemas embarcados (Capítulo A) para que possa ser acondicionado, de forma permanente, no próprio transformador. Desta forma, a saúde do sistema poderá ser continuamente monitorada.

Capítulo 5

Conclusões e Trabalhos Futuros

Devido ao fato da adoção de camadas funcionais integrando processamento conexionista e simbólico, além da utilização da concepção de sistemas multiagentes e de *clusterização*, a arquitetura apresentada é generalista, ou seja, permite-se a implementação de uma diversa gama de sistemas baseados em conhecimento. A especificidade da generalização é associada na adoção da abstração de se utilizar as categorizações de Peirce.

A motivação de se utilizar técnica evolucionária em conjunto com a conexionista reside no fato de poder representar a própria adaptação do raciocínio lógico frente ao ambiente no qual o sistema estiver inserido. Dessa forma, a arquitetura torna-se mais flexível para diversas situações, ou seja, o modelo evolucionário não fica restrito às aplicações de otimização.

No caso específico da aplicação usada para validar a arquitetura proposta, pode-se fazer as seguintes observações:

- Os resultados apresentados mostram que o sistema foi capaz de prever corretamente não só o tipo de falha mas, também, o momento esperado da respectiva falha. Esse tipo de previsão é muito útil para Gestão de Saúde do Sistema. Esta é uma contribuição importante desse trabalho uma vez que vai um passo além da classificação de falhas em transformadores de potência.
- Uma das vantagens do método apresentado consiste na não necessidade do co-

nhecimento sobre a idade do transformador de potência e o seu histórico de uso. O sistema implementado é capaz de realizar um prognóstico somente pelas informações coletadas pelo DGA. O prognóstico poderá ser gerado, prontamente, tão logo ocorra a coleta das concentrações dos gases pelo dispositivo DGA. Nesse caso, a classificação como *online* ou *offline* é remetido ao DGA.

- Em relação aos custos para a implantação do sistema proposto, eles poderão ser estimados como mínimos uma vez que a metodologia proposta pode ser implementado como um sistema embutido. A parte mais cara está relacionada com o equipamento de DGA.
- O sistema para a geração de prognósticos de falhas pode ser estendido para outras aplicações dentro da área que engloba os sistemas elétricos de potência.
- Não foi possível uma real comparação dos resultados obtidos pelo sistema aqui implementado com outros trabalhos devido à inexistência de trabalhos correlatos. Os escassos trabalhos encontrados sobre o tema atuam de forma diferente sobre os dados coletados (por exemplo, fazendo apenas uma projeção dos valores de concentrações dos gases de forma separada) ou necessitam dos dados históricos ou da idade do transformador de potência.

Como trabalho futuro, pretende-se melhorar a precisão da camada conexionista testando outros métodos de aprendizagem de máquina aplicada na ideia de conjuntos (*ensemble*). Outro importante prosseguimento deste trabalho é a finalização da implementação em hardware. Com a implementação em hardware, o sistema poderá ser acoplado diretamente nos transformadores de potência.

Além da aplicação descrita neste trabalho, aspira-se utilizar a arquitetura em outras aplicações. Os planos futuros para a utilização deste trabalho incluem:

- Dimensionamento de estruturas da fuselagem de aviões. Neste ponto, será aplicado ao projeto de desenvolvimento de modelos aviônicos em curso na Universidade Federal de Ouro Preto.

- Smart Grid: consiste na integração de sistemas de monitoramento e controle e sistemas de comunicação com os sistemas elétricos de potência. O ponto principal consiste na análise de informações coletadas ao longo da linha de transmissão de modo a se obter, como resultado do processamento, a manutenção de qualidade dos serviços de energia elétrica de forma automatizada.
- Manutenção de saúde de sistemas em sistemas aviônicos: a constante avaliação atual do sistema, gerando-se diagnósticos e prognósticos de falhas, propicia o aumento da disponibilidade do sistema através de uma melhor programação de manutenções preventivas e corretivas. Assim, consegue-se, também, a redução do custo decorrentes de manutenções desnecessárias e não programadas.

No caso específico da manutenção de saúde de sistemas em aeronaves comerciais, os sinais obtidos a partir dos sistemas e componentes da aeronaves incidirão diretamente sobre os módulos deliberativos e reativos da arquitetura proposta. Os sinais serão também modelados como elementos emocionais que interferirão no nível de sensibilidade de aprendizagem e de reação. Como emoções, poderão ser utilizados, também, o nível de fadiga dos componentes, nível de ruídos (interferência) incididos sobre os dispositivos. Cada subsistema da aeronave será representado por um *cluster*.

Apêndice A

Aspectos da Implementação do Sistema de Prognósticos de Falhas de Transformadores de Potência em Hardware

A partir da implementação do sistema de prognósticos de falhas de transformadores de potência e a sua respectiva validação pelos experimentos realizados citados no Capítulo 4, foi estimada, também, a sua implementação em hardware. Sendo assim, o objetivo deste apêndice consiste em descrever aspectos inerentes à implementação em hardware.

A.1 Computação Reconfigurável

Denomina-se computação reconfigurável o sistema computacional dotado da capacidade de modificação, sob demanda, de suas estruturas internas. Tal reconfiguração é extremamente útil nos casos onde se deseja alterar a funcionalidade do sistema para atender às novas demandas, atualização em função de novas versões ou para corrigir possíveis erros de projeto e não encontrados durante as fases de verificação e testes.

Com os sistemas reconfiguráveis, pode-se combinar características como a alta

performance dos sistemas baseados em *hardware* (*ASICs* – *Application Specific Integrated Circuits*) com a flexibilidade dos sistemas baseados em *software* (por exemplo, os processadores de uso geral).

A Figura A-1 mostra um gráfico comparativo entre os sistemas baseados em processadores de uso genérico, *ASICs* e *hardware* reconfigurável tomando-se como métrica a flexibilidade e a performance dos sistemas apresentados.

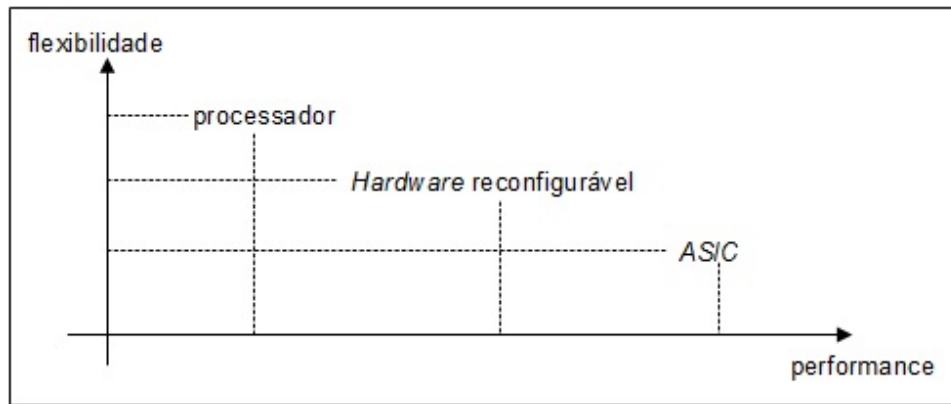


Figura A-1: Comparação entre processadores, *hardware* reconfigurável e *ASICs* em termos de flexibilidade e performance

Atualmente, sistemas reconfiguráveis envolvem o emprego de *FPGAs* (*Field Programmable Gate Arrays*). As *FPGAs* geralmente vêm interligadas a sistemas de memória, *ASICs* específicos, *DSPs* (*Digital Signal Processors*) ou microprocessadores de uso geral. Inclusive, para se diminuir o atraso na propagação dos sinais entre os elementos de *software* e de *hardware* e o *FPGA*, os sistemas computacionais estão sendo construídos dentro de um único *chip*, envolvendo, para tal, núcleo dos dispositivos (*IP cores* - *Intellectual Property*) a serem utilizados em conjunto com a *FPGA*.

Em relação à porção da *FPGA* a ser reconfigurada, alguns modelos permitem que essa possa ser realizada de forma seletiva. Tal seleção consiste na modificação de partes selecionadas do dispositivo sem afetar toda a lógica previamente descarregada. Como consequência direta da reconfiguração parcial, o tempo para a reconfiguração é menor em relação à reconfiguração total uma vez que serão modificadas apenas partes do circuito.

Quanto ao momento de reconfiguração de uma *FPGA*, pode-se classificar em reconfiguração estática ou dinâmica, havendo, para tal, a necessidade de paralisar ou não o processamento do dispositivo, respectivamente.

Na reconfiguração dinâmica (também denominada como *on-the-fly*, *on-line* ou *run-time*), pode-se utilizar memória externa para armazenar os possíveis mapas de reconfiguração que serão utilizados. Os mapas serão transferidos ao dispositivo de acordo com a necessidade. Uma outra possibilidade consiste em trabalhar com múltiplos contextos internamente a um dispositivo. Neste caso, existem vários planos (contextos) de memórias internamente ao dispositivo, onde apenas um contexto encontra-se ativo em um determinado momento. Sendo assim, a reconfiguração consiste na ativação e desativação de contextos, necessitando para tal apenas um único pulso de clock. A vantagem de se ter uma latência de reconfiguração altamente reduzida implica no custo adicional de área de silício demandada pelos dispositivos que admitem vários contextos de reconfiguração.

A Figura A-2 ilustra, de forma genérica, a utilização de contextos únicos e múltiplos além de tornar visível a reconfiguração parcial (Compton & Hauck, 2002).

As características dos dispositivos apresentadas anteriormente permitem explorar dois tipos de mecanismos de reconfiguração existentes: a reconfiguração voltada ao tempo de compilação (*compile-time reconfiguration* - *CTR*) e a reconfiguração voltada ao tempo de execução (*run-time reconfiguration* - *RTR*) (Hutchings & Wirthlin, 1995).

A *CTR* está estreitamente relacionada à reconfiguração estática, onde a reconfiguração dar-se-á apenas na transição das aplicações, ou seja, todos os recursos utilizados são previamente mapeados pelo compilador e recarregados no dispositivo apenas no momento de carga da aplicação. Apresenta as seguintes características principais (Rincon & Teres, 1997):

- A configuração do dispositivo (*download do bitstream* que representa o mapeamento das estruturas a serem utilizadas) é realizada apenas uma vez no período que corresponde ao *set-up* da aplicação. No restante do tempo, a configuração permanecerá inalterada;

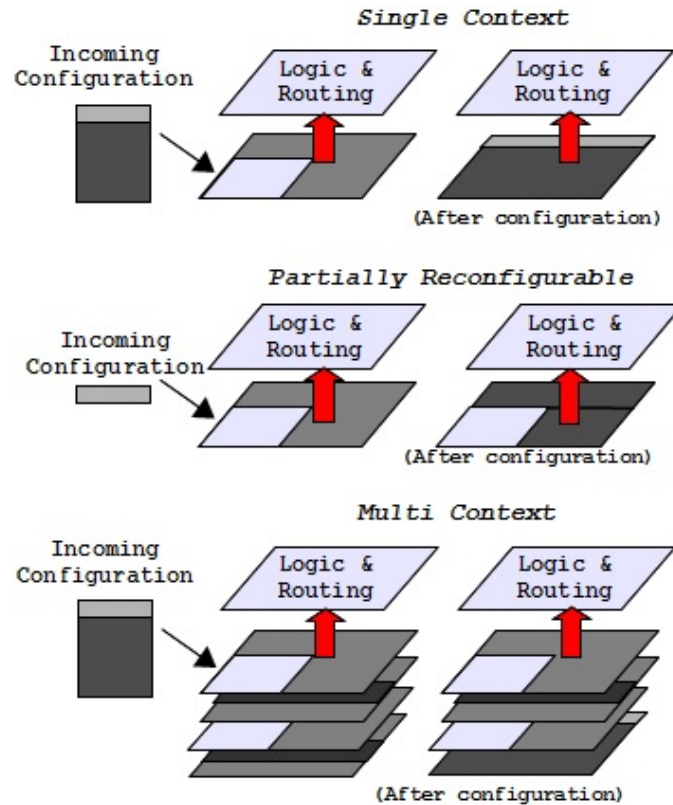


Figura A-2: Esquemas de Reconfiguração

- a execução no dispositivo deverá ser paralisada para a nova configuração inerente à nova aplicação a ser executada;
- devido à simplicidade do *CTR* frente ao *RTR*, todo dispositivo reprogramável é apto a trabalhar com tal filosofia de reconfiguração.

Por outro lado, no *RTR*, uma única aplicação pode demandar reconfigurações durante a sua execução. Sendo assim, *RTR* está relacionada aos dispositivos que permitem uma reconfiguração dinâmica.

Várias aplicações são modeladas e implementadas utilizando-se o paradigma de sistemas reconfiguráveis. Dentre os quais pode-se destacar: co-processadores, dispositivos para manipulação de criptografia e compressão de dados, processamento de imagens, equipamentos para redes de computadores, dispositivos para redes de sensores e sistemas robóticos adaptativos e evolucionários.

A.2 *System-on-Chip* (SoC)

Grandes avanços têm-se verificado no campo da utilização de sistemas embutidos, providos ou não de uma integração de hardware/software. Tais avanços têm possibilitado a aplicação dos mesmos nos mais variados setores da sociedade, desde uso doméstico até aplicações científicas, industriais e militares. Um desses avanços é representado pelos sistemas denominados como *System-on-Chip* (SoC).

De forma pontual, um SoC pode ser caracterizado como sendo um circuito integrado que integra todos os componentes do sistema computacional desenvolvido. No referido circuito integrado, poderão estar presentes circuitos analógicos, digitais e, também, códigos de software gravados em suas estruturas físicas.

Um crescimento maior dos SoC pôde ser verificado nos últimos anos favorecido pela necessidade de aplicações que integrassem, em um único chip, diversos componentes de software e de hardware dedicados. Um outro fator que favoreceu a expansão da área representada pelos SoC reside no avanço da tecnologia eletrônica, por exemplo, na tecnologia de fabricação baseada em *CMOS* (Complementary Metal-Oxide Semiconductor) permitindo-se, dessa forma, uma maior integração dos componentes a um baixo custo.

Porém o referido avanço trouxe como consequência uma maior complexidade dos sistemas e, portanto, necessidade de técnicas mais apuradas e ferramentas mais precisas e abrangentes para a verificação dos designs modelados. Além de envolver a necessidade de melhor definição de interfaces, criação de padrões de interconexão de módulos e implementação de módulos passíveis de serem utilizados em vários projetos, reaproveitando-se o seu código.

As últimas consequências citadas acima fizeram com que um novo segmento surgisse no mercado: o representado pelas indústrias responsáveis pelo desenvolvimento de *Intellectual Property Cores* (IPs), os quais representam módulos reutilizáveis de SoCs. Os IPs cores podem ser classificados como *hard*, *firm* e *soft cores*.

Hard IP cores são distribuídos de forma fechada e geralmente estão associados a altas complexidade e eficiência computacionais. Como exemplo, podem ser cita-

dos microprocessadores e *phase-locked loops*. Os cores conhecidos como *firm IP* são disponibilizados por intermédio de um *netlist* previamente sintetizado, porém sem informações de *layout*. Por último, os *soft IP cores* são distribuídos no modo *RTL HDL* e a sua síntese e definição do *layout* ocorrem totalmente por responsabilidade do usuário.

No âmbito global, vários aspectos têm que ser levados em conta em relação ao desenvolvimento de um SoC. Dentre os quais podem ser citados: escalabilidade da performance, consumo de energia, ruídos injetados no circuito (principalmente em SoCs que operam em alta frequência), integração com outros cores e custo.

A.3 Proposta de Implementação em Hardware

Levando-se em conta a flexibilidade da arquitetura proposta neste projeto para a construção de sistemas genéricos baseados em conhecimento, optou-se, em primeira instância, por utilizar Hard IP SoCs (incluindo processadores de propósito geral) sobre hardware reconfigurável (FPGA). A ideia de se utilizar processadores de propósito geral dá margem à possibilidade de implementação de diversos outros sistemas baseados em conhecimento além do implementado neste projeto. A Figura A-3 ilustra a concepção inicial de estruturação do hardware.

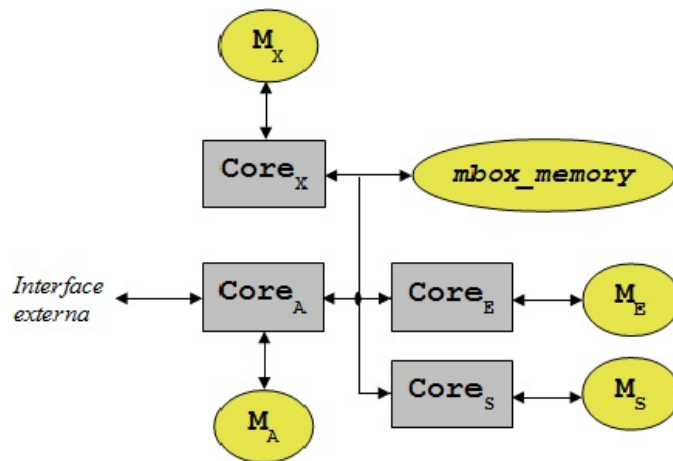


Figura A-3: Proposta de implementação em hardware

Devido baixo consumo de área e pela facilidade em obter material e aplicativos

para o desenvolvimento do código, optou-se por utilizar o processador AltOr32 (Alt, n.d.). Trata-se de um processador de 32 bits baseado no OpenRisc (Ope, n.d.).

As funcionalidades dos componentes da Figura A-3 são descritas na Tabela A.1:

Tabela A.1: Funcionalidades dos componentes do SoC proposto

Componente	Funcionalidade
Core _A	Integra: a) as funcionalidades (código) do agente representativo da aplicação principal; b) as interfaces para o mundo externo (pinos de entrada de dados, saída do sistema reativo e interface com o usuário); c) funcionalidades embriônicas
Core _X	Núcleo responsável pela manipulação das ontologias (XML)
Core _E	Corresponde às funcionalidades evolucionárias do sistema baseado em conhecimento
Core _S	Responsável pela manipulação conexionista. No caso específico da implementação do sistema de prognósticos de transformadores de potência, corresponde às implementações das redes neurais do <i>ensemble learning</i> .
M _A	Memória do agente da aplicação
M _X	Memória de ontologia
M _E	Memória de trabalho do componente evolucionário
M _S	Memória de trabalho do componente conexionista
<i>mbox_memory</i>	Memória para a troca de informações entre os módulos internos (<i>cores</i>) usando o paradigma de <i>mailbox</i> .

O *layout* da implementação em hardware é ilustrado na Figura A-4.

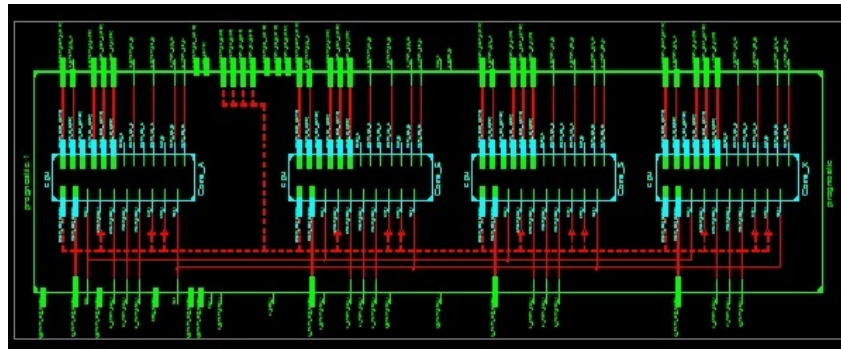


Figura A-4: *Layout* da implementação em hardware

Os principais pinos constantes da interface externa do equipamento estão relacionados na Tabela A.2. Os demais pinos que aparecem na Figura A-4 representam a interface para a interligação dos núcleos internos aos seus respectivos módulos de

memória.

Como informado anteriormente, optou-se por utilizar núcleos de processadores genéricos para permitir uma maior flexibilidade na construção das aplicações baseadas em conhecimento e, também, facilitar o processo de implementação inicial. Porém, por exemplo, o núcleo responsável pela manipulação de ontologias (parser XML) será substituído por um circuito específico destinado somente à essa função. As próprias aplicações, dependendo de suas complexidades, poderão ser convertidas em módulos do tipo ASIC (*Application Specific Integrated Circuit*).

Tabela A.2: Pinagem da proposta de implementação em hardware

Pino	direção	tamanho	Funcionalidade
clk	input	1	Sinal de clock, produzido por um oscilador externo.
rst	input	1	Sinal de reset
data_i	input	32	Entrada de dados (concentração dos gases).
e_data_i	input	1	Informa que a informação está disponível ao módulo de prognósticos.
ack_i	output	1	Sinal de <i>acknowledgement</i> informando que o dado entrado já foi consumido pelo módulo de prognósticos.
data_o	output	32	Pinos de saída de dados.
e_data_o	output	1	Informa ao módulo externo a disponibilidade de informações no pino e_data_o.
ack_o	input	1	Sinal de <i>acknowledgement</i> gerado pelo módulo externo indicando que já consumiu a informação exportada pelo módulo de prognóstico.

A.4 Conclusões

Este apêndice apresentou os conceitos e concepções iniciais inerentes à implementação, em hardware, do sistema para a geração de prognósticos em transformadores de potência. O trabalho de implementação em hardware será ainda desenvolvido para que um produto seja produzido pois o foco deste trabalho foi a modelagem da arquitetura genérica e a implementação da aplicação para a sua validação. Com base no sucesso do desenvolvimento do modelo, tornar-se-á possível, agora, a implementação

em hardware para aumentar a gama de possibilidades de utilização.

O código do agente, que representa o próprio comportamento do sistema de prognósticos, será gravado diretamente na FPGA que representa o núcleo da aplicação estabelecendo-se, assim, um SoC. Alterações nas funcionalidades dos agentes serão permissíveis por intermédio da reconfiguração da FPGA realizando-se a carga do novo código disponibilizado através do núcleo de manipulação das ontologias.

Referências Bibliográficas

- AltOr32 - Alternative Lightweight OpenRisc CPU*. <http://opencores.org/project,altor32>. Accessed: 2015-02-17.
- OpenRisc*. http://opencores.org/or1k/Main_Page. Accessed: 2015-02-17.
- Albus, J. S., McCain, H. G., & Lumia, R. 1989. *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*. Tech. rept. Accessed: 2015-11-15.
- Arakelian, V. G. 2004. The long way to the automatic chromatographic analysis of gases. *IEEE Electrical Insulation Magazine*, **20**(6), 8–25.
- Arnellos, A., Vosinakis, S., Anastasakis, G., & Darzentas, J. 2008. Autonomy in virtual agents: integrating perception and action on functionally grounded representations. *Page 51–63 of: Proc. 5th Hellenic conference on Artificial Intelligence: Theories, Models and Applications (SETN '08)*.
- Atkinson, R.C., & Shiffrin, R. M. 1971. The control of short-term memory. *Scientific American*, **225**, 82–90.
- Ausubel, D. P., Hanesian, H., & Novak, J. D. 1981. *Educational psychology*. Beltz.
- Back, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.
- Baddeley, A. D., & Hitch, G. J. 1974. Working memory. *Pages 47–89 of: Bower, G. (ed), The psychology of learning and motivation*, vol. VIII. New York: Academic Press.

- Bolander, N., Qiu, H., Eklund, N., Hindle, E., & Rosenfeld, T. 2010. Physics-based Remaining Useful Life Prediction for Aircraft Engine Bearing Prognosis. *In: Annual Conference of Prognosis and Health Management Society*.
- Breiman, L. 1996. Bagging predictors. *Machine Learning*, **24**(2), 123–140.
- Brescia, T., Bruno, S., La Scala, M., Lamonaca, S., Rotondo, G., & Stecchi, U. 2009. A fuzzy-logic approach to preventive maintenance of critical power transformers. *Pages 1–5 of: 20th International Conference on Electricity Distribution*. CIRED.
- Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, **2**(1), 14–23.
- Brooks, R. A. 1991. *Intelligence without Representation*. <http://people.csail.mit.edu/brooks/papers/representation.pdf>. Accessed: 2015-11-15.
- Brooks, R. A. 1999. *Cambrian Intelligence: The Early History of the new AI*. Bradford Book. MIT Press.
- Canham, R. O., & Tyrrell, A. M. 2003a. A hardware artificial immune system and embryonic array for fault tolerant systems. *Genetic Programming and Evolvable Machines*, **4**, 359–382.
- Canham, R. O., & Tyrrell, A. M. 2003b. A Learning Multi-Layered Hardware Artificial Immune System Implementation Upon an Embryonic Array. *Pages 174–185 of: Proc. ICES 2003, LNCS 2606*.
- Castanheira, L. G., Vasconcelos, J. A., Reis, A. J. R., Magalhães, P. H. V., & Silva, S. A. L. 2011a. Application of neural networks in the classification of incipient faults in power transformers: a study of case. *Pages 3099–3104 of: Proc. International Joint Conference on Neural Networks 2011 (IJCNN 2011)*.
- Castanheira, L. G., Vasconcelos, J. A., Reis, A. J. R., Magalhães, P. H. V., & Silva, S. A. L. 2011b. A Comparative Study of the Application of Neural Networks and Decision Trees in the Classification of Incipient Faults in Power Transformers.

Pages 1–6 of: Proc. X Brazilian Conference on Computational Intelligence (X CBIC).

Catterson, V. M. 2014. Prognostic modeling of transformer aging using Bayesian particle filtering. *Pages 413–416 of: Proc. IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP).*

Chen, C., Vachtsevanos, G., & Orchard, M. E. 2012. Machine Remaining Useful Life Prediction: An Integrated Adaptive Neuro-fuzzy and High-order Particle Filtering Approach. *Mechanical Systems and Signal Processing*, **28**(4), 597–607.

Chen, W., Pan, C., Yun, Y., & Liu, Y. 2009. Wavelet networks in power transformers diagnosis using dissolved gas analysis. *IEEE Transactions on Power Delivery*, **24**(1), 187–194.

Cheng, S., & Pecht, M. 2007. Multivariate State Estimation Technique for Remaining Useful Life Prediction of Electronics Products. *In: AAAI Fall Symposium Artificial intelligence. Prognostics*. Arlington.

Cheng, S., & Pecht, M. 2009. A Fusion Prognostics Method for Remaining Useful Life Prediction of Electronics Products. *In: IEEE International Conference on Automation Science and Engineering*.

Chickering, D. M., Heckerman, D., & Meek, C. 2004. Large-Sample Learning of Bayesian Networks is NP-Hard. *J. Mach. Learn. Res.*, **5**(Dec.), 1287–1330.

Cho, M. Y., Lee, T., Kau, S., Shieh, C., & Chou, C. 2006. Fault diagnosis of power transformers using svm/ann with clonal selection algorithm for features and kernel parameters selection. *Pages 26–30 of: Proc. International Conference on Innovative Computing, Information and Control*.

Compton, K., & Hauck, S. 2002. Reconfigurable Computing: a Survey of Systems and Software. *ACM Comput. Surv.*, **34**(2), 171–210.

- Coppe, A., Pais, M. J., Haftka, R. T., & Kim, N. H. 2012. Using a Simple Crack Growth Model in Predicting Remaining Useful Life. *Journal of Aircraft*, **6**(49), 1965–1973.
- De Stefano, C., Fontanella, F., Marrocco, C., & A. S. di Freca. 2010. A hybrid evolutionary algorithm for bayesian networks learning: an application to classifier combination. *Pages 221–230 of: Proceedings of the 2010 international conference on Applications of Evolutionary Computation - Volume Part I*. EvoApplicatons’10. Berlin, Heidelberg: Springer-Verlag.
- Delarboulas, P., Schoenauer, M., & Sebag, M. 2010. Open-Ended Evolutionary Robotics: an Information Theoretic Approach. *CoRR*, **abs/1006.4959**, 334–343.
- Depari, A., Flammini, A., Sisinni, E., De Marcellis, A., Ferri, G., & Mantenuto, P. 2004. Fast, versatile, and low-cost interface circuit for electrochemical and resistive gas sensor. *Electrical Engineering*, **14**(2), 315–323.
- Dong, L., Wright, J., Peters, B., Ferguson, B. A., Tittel, F. K., & McWhorter, S. 2012. Compact QEPAS sensor for trace methane and ammonia detection in impure hydrogen. *Applied Physics B*, **107**(2), 459–467.
- dos Santos, E. B., Hruschka Junior, E. R., Hruschka, E. R., & Ebecken, N. F. F. 2010. A Distance-Based Mutation Operator for learning Bayesian Network structures using Evolutionary Algorithms. *Pages 1–8 of: IEEE Congress on Evolutionary Computation*. IEEE.
- Dubal, S., Foucher, A., Jouvent, R., & Nadel, J. 2010. Human brain spots emotion in non humanoid robots. *Oxford Journals: Social Cognitive and Affective Neuroscience*, **6**(1), 90–97.
- Duval, M. 2002. A Review of faults detectable by gas-in-oil analysis in transformers. *IEEE Electrical Insulation Magazine*, **18**(3), 8–17.
- Duval, M., & de Pablo, A. 2001. Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases. *DEIS Feature Article*, **17**(2), 31–41.

- Egbert, M. D., Barandiaran, X. E., & Di Paolo, E. A. 2011. Behavioral Metabolution: The Adaptive and Evolutionary Potential of Metabolism-Based Chemotaxis. *Artificial Life*, **18**(1), 1–25.
- Eiben, A. E., & Smith, J. E. 2003. *Introduction to Evolutionary Computing*. SpringerVerlag.
- Fatemi, A., & Yang, L. 1998. Cumulative Fatigue Damage and Life Predictions Theory: A Survey of the State of the Art for Homogeneous Materials. *International Journal of Fatigue*, **1**(20), 9–34.
- Fei, S. Wei, & Sun, Y. 2008. Forecasting dissolved gases content in power transformer oil based on support vector machine with genetic algorithm. *Electric Power Systems Research*, **78**, 507–514.
- Fensel, D. 2003. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2 edn. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Franklin, S. 1997. Autonomous Agents as Embodied AI. *Cybernetics and Systems*, **28**(6), 499–520.
- G. Husi, Geza, Szász, C., & Chindris, V. 2010. Artificial Immune System Implementation upon Embryonic Machine for Hardware Fault-tolerant Industrial Control Applications. *Global Journal of Computer Science and Technology*, **10**(4), 60–66.
- Gasvik, K. J., Robbersmyr, K. G., Vadseth, T., & Karimi, H. R. 2014. Deformation Measurement of Circular Steel Plates Using Projected Fringes. *The International Journal of Advanced Manufacturing Technology*, **1**(70), 321–326.
- Georgilakis, P. S. 2009. *Spotlight on modern transformer design*. London, UK: Springer.
- Georgilakis, P. S. 2011. Condition monitoring and assessment of power transformers using computational intelligence. *International Journal of Electrical Power and Energy Systems*, **33**(10), 1784–1785.

- Georgilakis, P. S., & Kagiannas, A. G. 2014. A novel validated solution for lightning and surge protection of distribution transformers. *International Journal of Electrical Power and Energy Systems*, **63**, 373–381.
- Georgilakis, P. S., Katsigiannis, J. A., Valavanis, K. P., & Souflaris, A. T. 2006. A systematic stochastic petri net based methodology for transformer fault diagnosis and repair actions. *Journal of Intelligent and Robotic Systems*, **45**(2), 181–201.
- Ghazanfar, A. A., & Turesson, H. K. 2008. How robots will teach us. how the brain works. *Nature Neuroscience*, **11**(3).
- Goldstein, Y., Grover, N. B., Chang, C., Jelli, A., Andre, J., Mark, P., & Goodwin, A. 1979. A gas chromatographic air analyzer fabricated on a silicon wafer. *IEEE Transactions on Electron Devices*, **26**(12), 1880–1886.
- Heng, A., Zhang, S., Tan, A. C. C., & Mathew, J. 2009. Rotating machinery prognostics: state of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, **23**(3), 724–739.
- Higuchi, T. 1993. Evolvable hardware: A first step towards building a Darwin machine. *Pages 417–424 of: in Proc. of the 2nd Int. Conf. on Simulated Behaviour*. MIT Press.
- Huang, Y. C., & Sun, H. C. 2013. Dissolved gas analysis of mineral oil for power transformer fault diagnosis using fuzzy logic. *IEEE Transactions on Dielectrics and Electrical Insulation*, **20**, 974–981.
- Hutchings, B. L., & Wirthlin, M. J. 1995. Implementation Approaches for Reconfigurable Logic Applications. *Pages 419–428 of: Proc. International Workshop on Field-Programmable Logic and Applications*. Springer.
- IEC. 2008. *IEC mineral oil-impregnated electrical equipment in service – guide to the interpretation of dissolved and free gases analysis, IEC 60599*.

- IEEE. 2008. *IEEE guide for the interpretation of gases generated in oil-immersed transformers, IEEE Standard C57.104.*
- Imbert, R., & de Antonio, A. 2005. An emotional architecture for virtual characters. *Pages 63–72 of: Proceedings of the Third international conference on Virtual Storytelling: using virtual reality technologies for storytelling.* ICVS'05. Berlin, Heidelberg: Springer-Verlag.
- Isla, D., Burke, R., Downie, M., & Blumberg, B. 2001. A layered brain architecture for synthetic creatures. *Pages 1051–1058 of: Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2.* IJCAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Jong, K. A. 2006. *Evolutionary computation - a unified approach.*
- Julka, N., Thirunavukkarasu, A., Lendermann, P., Gan, B. P., Schirrmann, A., Fromm, H., & Wong, E. 2011. Making use of prognostics health management information for aerospace spare components logistics network optimisation. *Computers in Industry*, **62**, 613–622.
- Khanteymoori, A. R., Menhaj, M. B., & Homayounpour, M. M. 2011. Structure Learning in Bayesian Network Using Asexual Reproduction Optimization. *Etril Journal*, **33**, 39–49.
- Kim, S., Kim, S., Seo, H., Jung, J., Yang, H., & Duval, M. 2013. New methods of DGA diagnosis using IEC TC 10 and related databases part 1: application of gas-ratio combinations. *IEEE Transactions on Dielectrics and Electrical Insulation*, **20**(2), 685–690.
- Kogler Junior, J. E., & Inojosa Filho, R. 2010. First steps toward a cognitive architecture based on adaptive automata. *Advances in Modeling Adaptive and Cognitive Systems*, 38–47.
- Kusiak, A., & Li, W. 2011. The prediction and diagnosis of wind turbine faults. *Renewable Energy*, **36**(1), 16–23.

- Kusiak, A., & Verma, A. 2011. A data-driven approach for monitoring blade pitch faults in wind turbines. *IEEE Transactions on Sustainable Energy*, **2**(1), 87–96.
- Laberge, D., & Brown, V. 1989. Theory of Attentional Operations in Shape Identification. *Psychological Review*, **96**(1), 101–124.
- Lai, K. X. P., & Blackburn, T. R. B. T. 2008. Descriptive Data Mining of Partial Discharge using Decision Tree with Genetic Algorithm. *Pages 1–6 of: Proc. Australasian Universities Power Engineering Conference (AUPEC'08)*.
- Lau, B. C. P., Ma, E. W. M., & Pecht, M. 2012. Review of offshore wind turbine failures and fault prognostic methods. *Pages 1–5 of: 2012 IEEE Conference on Prognostics and System Health Management (PHM)*. IEEE.
- Liao, L. 2009. Design of a reconfigurable prognostics platform for machine Tools. *Expert Systems with Applications*, **37**, 240–252.
- Liao, R.J., Bian, J.P., Yang, L.J., Grzybowski, S., Wang, Y.Y., & Li, J. 2010. Forecasting dissolved gases content in power transformer oil based on weakening buffer operator and least square support vector machine–Markov. *IET Generation, Transmission & Distribution*, **6**(2), 142–151.
- Lopes, Heitor S., & Takahashi, Ricardo H. C. (eds). 2011. *Computação Evolucionária em Problemas de Engenharia*. 1 edn. Curitiba (PR): Omnipax. 385 p.
- Maio, F. Di, Ng, S. S., Tsui, K., & Zio, E. 2011. Bayesian Classifier for On-line Remaining Useful Life Prediction of Degrading Bearings. *In: Proceedings of the Seventh International Conference on Mathematical Methods in Reliability*.
- Maojo, V., Martin-Sanchez, F., Kulikowski, C., Rodriguez-Paton, A., & Fritts, M. 2010 (01). *Nanoinformatics and DNA-Based Computing: Catalyzing Nanomedicine*.
- Medjaher, K., Tobon-Mejia, D. A., & Zerhouni, N. 2012. Remaining Useful Life Estimation of Critical Components with Application to Bearings. *IEEE Journal of Transactions on Reliability*, **2**(61), 292–302.

- Minsky, M. 1961. Steps Toward Artificial Intelligence. *Pages 406–450 of: Computers and Thought*. McGraw-Hill.
- Minsky, M. 1974. *A Framework for Representing Knowledge*. Tech. rept. Cambridge, MA, USA.
- Morais, D. R., & Rolim, J. G. 2006. A hybrid tool for detection of incipient faults in transformers based on the dissolved gas analysis of insulating oil. *IEEE Transactions on Power Delivery*, **21**(2), 673–680.
- Morato, E. F. 2012. *Presença da Filosofia Kantiana na Teoria Semiótica de Charles Sanders Peirce*.
- Mosallam, A., Medjaher, K., & Zerhouni, N. 2014. Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 1–12.
- Muller, A., Suhner, M., & Iung, B. 2008. Formalisation of a New Prognosis Model for Supporting Proactive Maintenance Implementation on Industrial System. *Reliability Engineering & System Safety*, **2**(93), 234–253.
- Nadel, J., Simon, M., Canet, P., Soussignan, R., Blancard, P., Canamero, L., & Gaussier, P. 2006. Human response to an expressive robot. *Epigenetic Robotics*, **6**, 79–86.
- Naresh, R., Sharma, V., & Vashisth, M. 2008. An integrated neural fuzzy approach for fault diagnosis of transformers. *IEEE Transactions on Power Delivery*, **23**(4), 2107–2024.
- Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J. M., & Kendall, G. 2010. The Application of a Dendritic Cell Algorithm to a Robotic Classifier. *CoRR*, **abs/1004.5222**.
- Okoh, C., Roy, R., Mehnen, J., & Redding, L. 2014. Overview of Remaining Useful Life Prediction Techniques in Through-Life Engineering Services. *Pages 158–*

163 of: *Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems*. Elsevier.

Ozgur-Unluakin, D., & Bilgiç, Taner. 2006. Predictive maintenance using dynamic probabilistic networks. *Pages 239–246 of: Probabilistic Graphical Models'06*.

Pan, C., Chen, W., & Yun, Y. 2008. Fault diagnostic method of power transformers based on hybrid genetic algorithm evolving wavelet neural network. *IET Electric Power Applications*, 71–76.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Peirce, C. S. 1958. *Collected Papers*. Harvard University Press.

Raja, M. S., & Raviraj, P. 2012. Biologically inspired design framework for Robot in Dynamic Environments using Framsticks. *CoRR*, **abs/1201.2100**.

Rajabimendi, M., & Dadios, E. P. 2012. A hybrid algorithm based on neural-fuzzy system for interpretation of dissolved gas analysis in power transformers. *Pages 1–6 of: Proc. TENCON 2012 - 2012 IEEE Region 10 Conference*.

Rincon, F., & Teres, L. 1997. Reconfigurable Hardware Systems. *Pages 45–54 of: Proc. Int. Semiconductor Conference*.

Ritter, H., Steil, J. J., Nölker, C., Röthling, F., & McGuire, P. 2003. Neural architectures for robot intelligence. *Rev Neurosci*, **14**(1-2), 121–43.

Salatie, J. R. 2006. Peirce e Kant Sobre Categorias: Parte II – da Filosofia Transcendental à Cosmologia Evolucionária. *Cognitio-Estudios: Revista Eletrônica de Filosofia*, **3**(2), 169–175.

Sanchez, E., Mange, D., Sipper, M., Tomassini, M., Perez-Uribe, A., & Stauffer, A. 1997. Phylogeny, ontogeny and epigenesis: three sources of biological inspiration for softening hardware. *Lecture Notes in Computer Science*, **1259**, 33–54.

- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. 2008. Metrics for Evaluating Performance of Prognostic Techniques. *Pages 1–17 of: International Conference on Prognostics and Health Management - PHM 2008*. IEEE.
- Schwabacher, M., & Goebel, K. 2007. A survey of artificial intelligence for prognostics. *Pages 107–114 of: Proc. AAAI Fall Symposium*.
- Seifeddine, S., Khmais, B., & Abdelkader, C. 2012. Power transformer fault diagnosis based on dissolved gas analysis by artificial neural network. *Pages 230–236 of: Proc. First International Conference on Renewable Energies and Vehicular Technology (REVET)*.
- Sharma, M., & Kumar, D. 2012. Wishbone bus architecture - a survey and comparison. *CoRR*, **abs/1205.1860**.
- Shintemirov, A., Tang, W., & Wu, Q. H. 2009. Power transformer fault classification based on dissolved gas analysis by implementing bootstrap and genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, **39**(1), 69–79.
- Si, X., Wang, W., Hu, C., & Zhou, D. 2011. Remaining Useful Life Estimation - A Review on the Statistical Data Driven Approaches. *European Journal of Operational Research*, **1**(213), 1–14.
- Sica, F. C., Guimarães, F. G., de O. Duarte, R., & Reis, A. J. R. 2015. A Cognitive System for Fault Prognosis in Power Transformers. *Electric Power Systems Research*, 109–117.
- Singh, S., & Bandyopadhyay, M. 2010. Dissolved gas analysis technique for incipient fault diagnosis in power transformers: A bibliographic survey. *International IEEE Electrical Insulation Magazine*, **26**(6), 41–46.

- Sloat, T. K., Johnson, J. L., & Sommerman, G. M. L. 1967. Gas evolution from transformer oils under high-voltage stress. *IEEE Transactions on Power Apparatus and Systems*, **86**(3), 374–384.
- Specht, D. F. 1991. A general regression neural network. *IEEE Transactions on Neural Networks*, **2**(6), 568–576.
- Srivastava, A. N., & Han, J. 2011. *Machine learning and knowledge discovery for engineering systems health management*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press.
- Stafylopatis, A., & Blekas, K. 1998. Autonomous vehicle navigation using evolutionary reinforcement learning. *European Journal of Operational Research*, **108**(2), 306–318.
- Storn, R., & Price, K. 1995. *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*.
- Tang, W. H., Goulermas, J. Y., Wu, Q. H., Richardson, Z. J., & Fitch, J. 2008. A probabilistic classifier for transformer dissolved gas analysis with a particle swarm optimizer. *IEEE Transactions on Power Delivery*, **23**(2), 751–759.
- Thoma, Y., Tempesti, G., Sanchez, E., & Arostegui, J. M. 2004. POETic: an electronic tissue for bio-inspired cellular applications. *Biosystems*, **76**(1–3), 191 – 200.
<ce:title>Papers presented at the Fifth International Workshop on Information Processing in Cells and Tissues</ce:title>.
- Tran, V., Yang, B., & Tan, A. 2009. Multi-step Ahead Direct Prediction for Machine Condition Prognosis Using Regression Trees and Neuro-fuzzy systems.
- Tran, V. T., Pham, H. Thom, Yang, B., & Nguyen, T. Tien. 2012. Machine Performance Degradation Assessment and Remaining Useful Life Prediction Using Proportional Hazard Model and Support Vector Machine. *Mechanical Systems and Signal Processing*, 320–330.

- Uckun, S., Goebel, K., & Lucas, P. J.F. 2008. Standardizing Research Methods for Prognostics. *Pages 1–10 of: 2008 International Conference on Prognostics and Health Management*. IEEE.
- Upequi, A., Thoma, Y., Sanchez, E., Perez-Uribe, A., Malho, J. M., & Manderas, J. 2007. The Perplexus Bio-Inspired Reconfigurable Circuit. *Pages 600–605 of: Proc. 2nd NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*.
- Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. 2006. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Wiley.
- Wan, F., Zhou, Q., Zou, J., & Gu, Z. 2014. Using a sensitive optical system to analyze gases dissolved in samples extracted from transformer oil. *IEEE Electrical Insulation Magazine*, **30**(5), 15–22.
- Wei, M., Chen, M., Zhou, D., & Wang, W. 2011. Remaining Useful Life Prediction Using a Stochastic Filtering Model with Multi-Sensor Information Fusion. *In: Prognostics and System Health Management Conference (PHM-Shenzhen)*.
- Xiongzi, C., Jinsong, Y., Diyin, T., & Yingxun, W. 2011. Remaining Useful Life Prognostic Estimation for Aircraft Subsystems or Components: A Review. *In: 10th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. IEEE.
- Yang, S., Liu, C., Zhou, X., Liang, W., & Miao, Q. 2012. Investigation on Data-Driven Life Prediction Methods. *In: International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE)*. IEEE.
- Yoon, S. Y., Burke, R., Blumberg, B., & Schneider, G. 2000. Interactive Training for Synthetic Characters. *Pages 249–254 of: AAAI /IAAI 2000*.
- You, M. Y., Lin, L., Guang, M., & Jun, N. 2010. Two-zone Proportional Hazard Model for Equipment Remaining Useful Life Prediction. *Journal of Manufacturing Science and Engineering*, **4**(132).

- Zarei, J., Shasadeghi, M., & Ramezani, A. 2014. Fault prognosis in power transformers using adaptive-network-based fuzzy inference system. *Journal of Intelligent & Fuzzy Systems*, **26**, 2577–2590.
- Zhan, W., Goulart, A. E., Falahi, M., & Rondla, P. 2015. Development of a low-cost self-diagnostic module for oil-immersed forced-air cooling transformers. *IEEE Transactions on Power Delivery*, **30**(1), 129–137.
- Zhang, X., Dragffy, G., Pipe, A. G., & Zhu, Q. M. 2004. Partial-DNA supported artificial-life in an embryonic array. *Pages 203–208 of: Plaks, Toomas P. (ed), Proc. International Conference on Engineering of Reconfigurable Systems and Algorithms, ERSA 04, June 21–24, 2004, Las Vegas, Nevada, USA*. CSREA Press.
- Zhao, F., & Su, H. 2008. A Decision Tree Approach for Power Transformer Insulation Fault Diagnosis. *Pages 6882–6886 of: Proc. 7th World Congress on Intelligent Control and Automation (WCICA)*.
- Zhou, D., Wang, A., & Li, C. 2013. Data requisites for transformer statistical life-time modelling — Part I: aging-related failures. *IEEE Transactions on Power Delivery*, **28**(3), 1750–1757.
- Zylka, P. 2005. Electrochemical gas sensors can supplement chromatography based DGA. *Electrical Engineering*, **87**(3), 137–142.