

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

METODOLOGIAS DA INTELIGÊNCIA COMPUTACIONAL
EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS:
ABORDAGEM NEBULOSA, EVOLUTIVA E NEURAL

Leandro dos Santos Coelho

Florianópolis, fevereiro de 1997

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
ÁREA DE CONCENTRAÇÃO: SISTEMAS DE CONHECIMENTO

METODOLOGIAS DA INTELIGÊNCIA COMPUTACIONAL
EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS:
ABORDAGEM NEBULOSA, EVOLUTIVA E NEURAL

por
Leandro dos Santos Coelho

Orientador: Prof. Antonio Augusto Rodrigues Coelho, Dr. Sc.

Dissertação apresentada ao Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Santa
Catarina, como parte dos requisitos exigidos para
obtenção do Título de Mestre em Ciência da Computação.


Florianópolis, fevereiro de 1997

METODOLOGIAS DA INTELIGÊNCIA COMPUTACIONAL EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS: ABORDAGEM NEBULOSA, EVOLUTIVA E NEURAL

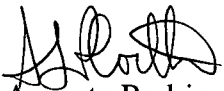
Leandro dos Santos Coelho

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, na área de concentração de Sistemas de Conhecimento, e aprovada em sua forma final pelo Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

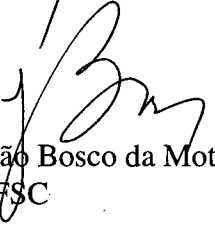

Prof. Antonio Augusto Rodrigues Coelho, Dr. Sc.
Orientador, EEL, UFSC

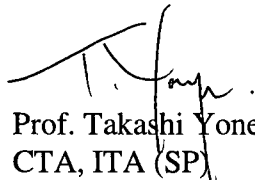

Prof. Murilo Silva de Camargo, Dr. Eng.
Coordenador do Curso, INE, UFSC

Banca Examinadora:


Prof. Antonio Augusto Rodrigues Coelho, Dr. Sc.
Presidente, EEL, UFSC


Prof. Rogério Cid Bastos, Dr. Eng.
INE, UFSC


Prof. João Bosco da Mota Alves, Dr. Sc.
INE, UFSC


Prof. Takashi Yoneyama, Dr. Sc.
CTA, ITA (SP)

Agradecimentos

Ao orientador Antonio Augusto Rodrigues Coelho pela dedicação e competência demonstradas na orientação deste trabalho, além do incentivo e amizade ao longo de nossa convivência.

A todos colegas, professores e funcionários da UFSC, que consolidei laços de amizade e de companheirismo.

A minha família pelo apoio.

Em especial à Viviana Cocco Mariani pelo amor e carinho, além do incessante estímulo, compreensão e paciência que tornaram possível esta caminhada.

Ao CNPq, por conceder-me a bolsa de estudos.

Aos membros da banca examinadora pelas contribuições ao aprimoramento da dissertação.

Resumo

A necessidade de projetistas e engenheiros de trabalhar com sistemas de controle cada vez mais complexos tem motivado a realização de pesquisas que viabilizam a utilização de identificação e algoritmos de controle em sistemas não-lineares, que apresentam incertezas e com características de atraso de transporte variantes no tempo. Os métodos clássicos apresentam limitações quando aplicados a sistemas com estas características.

Diversas abordagens não-convencionais em identificação e controle, tais como controle auto-ajustável, preditivo e inteligente tem sido propostas na literatura. Este trabalho trata da descrição, projeto e implementação de metodologias da inteligência computacional (ou artificial) em identificação e controle de processos, em escala laboratorial, com propósito ao tratamento adequado das restrições e dos requisitos exigidos no projeto de sistemas. As metodologias de inteligência computacional utilizadas são: lógica nebulosa, computação evolucionária (algoritmos genéticos e estratégias de evolução), redes neurais artificiais (“feedforward” e parcialmente recorrente de Elman) e “simulated annealing”.

O projeto de sistemas híbridos inteligentes, composto de algoritmos genéticos híbrido com “simulated annealing”, controle nebuloso com ajuste de parâmetros via computação evolucionária e algoritmos combinando redes neurais artificiais com o paradigma da computação evolucionária também são abordados.

Neste trabalho avalia-se os algoritmos de identificação dos mínimos quadrados recursivo e variável instrumental, e as estratégias de controle do tipo PID e linear invariante no tempo. Os algoritmos são desenvolvidos em linguagem de programação C e linguagem orientada a objetos C++ para plataforma IBM-PC.

Testes experimentais e análise do desempenho dos algoritmos em identificação de processos são realizados na identificação de um processo de temperatura, um processo de nível mono-tanque e, um processo não-linear que representa um turbo-gerador.

Os experimentos em controle abordam um processo não-linear de nível, composto de dois tanques acoplados e o sistema dinâmico não-linear vinculado a manobrar um caminhão para estacioná-lo em uma garagem (“truck backer-upper controller”).

Abstract

The need of designers and engineers in working with increasingly complex control systems have motivated the realization of researches that support the utilization of identification and control algorithms in non-linear systems, which present uncertainties and with time-varying deadtime features. The classical methods present limitations when applied to systems with these characteristics.

Some non-conventional approaches in identification and control, such as self-tuning, predictive and intelligent control have been proposed in the literature. This work deals with the description, design and implementation of the computational intelligence (or artificial) methodologies in process identification and control using processes, in laboratorial scale, with the purpose to adequately deal with the restrictions and requirements in the system design. The following computational intelligence methodologies are utilized: fuzzy logic, evolutionary computation (genetic algorithms and evolution strategies), artificial neural networks (feedforward networks and partially recurrent Elman networks), and simulated annealing.

The intelligent hybrid systems composed by genetic algorithms with simulated annealing, fuzzy control with parameters adjustment through evolutionary computation, and algorithms combining artificial neural networks with the evolutionary computation paradigm are present, too.

This work evaluates the identification algorithms of the least squares recursive and instrumental variable, and the PID and linear time-invariant control algorithms. The algorithms are developed in C programming language and C++ object-oriented programming for IBM-PC platform.

Experimental tests and performance analysis of the algorithms in process identification are carried out in the temperature process, mono-tank level process, and a non-linear process representing a turbo generator.

The control experiments address a non-linear level process, composed by coupled twin-tanks and the dynamic system for steering a trailer truck while backing up to a loading dock (“truck backer-upper controller”).

Índice

	pág.
Resumo	v
Abstract	vi
Índice	vii
Lista de figuras	ix
Lista de tabelas	xiii
Simbologia	xiv
1 Introdução	01
1.1 Motivação e relevância do trabalho	01
1.2 Objetivo	03
1.3 Organização da dissertação	03
2 Controle clássico e controle avançado	05
2.1 Introdução	05
2.2 Técnicas de identificação	08
2.2.1 Mínimos quadrados recursivo	09
2.2.2 Algoritmo da variável instrumental	12
2.3 Controle clássico	13
2.3.1 Controle PID	13
2.3.2 Controle linear invariante no tempo	14
2.4 Metodologias da inteligência computacional em identificação e controle	15
2.5 Critérios de erro	17
2.6 Conclusão	18
3 Paradigmas da computação evolucionária	20
3.1 Introdução	20
3.2 Algoritmos genéticos	22
3.2.1 Representação do cromossomo	24
3.2.2 Operadores genéticos e mecanismos de seleção	25
3.2.3 Definição dos parâmetros e função de adequabilidade	26
3.3 Algoritmos genético híbrido com “simulated annealing”	27
3.4 Estratégias de evolução	30
3.4.1 Estratégias de evolução multimembros	32
3.4.2 Características de algoritmos genéticos e estratégias de evolução	32
3.5 Conclusão	33
4 Metodologias de análise e projeto de controle lógico nebuloso	34
4.1 Introdução	34
4.2 Conjuntos nebulosos e a lógica nebulosa	35
4.2.1 Conceitos e definições básicas	36
4.2.2 Operações e propriedades dos conjuntos nebulosos	37
4.3 Controladores lógico nebuloso	37
4.3.1 Parâmetros de projeto	39
4.3.2 Controlador nebuloso com configuração PD ou PI	41
4.3.3 Controlador nebuloso com configuração PID	41
4.3.4 Implementação de sistema híbrido de projeto com computação evolucionária	43
4.4 Conclusão	44

5	Modelagem e controle via redes neurais artificiais	45
5.1	Introdução	45
5.2	Topologias	46
5.3	Rede neural “feedforward” com treinamento “backpropagation”	46
5.4	Redes neurais recorrentes	50
5.4.1	Rede neural parcialmente recorrente de Elman	51
5.4.1.1	Rede neural de Elman modificada	53
5.5	Identificação de processos via redes neurais	54
5.6	Controle neural de processos	55
5.6.1	Projeto de controle PID neural	56
5.7	Combinação de redes neurais e computação evolucionária	57
5.8	Conclusão	58
6	Aplicações em identificação de processos	59
6.1	Introdução	59
6.2	Descrição dos Processos	59
6.2.1	Processo de nível mono-tanque	59
6.2.2	Processo térmico	60
6.2.3	Turbo-gerador	61
6.3	Identificação de processos	61
6.3.1	Identificação do processo de nível mono-tanque e térmico	62
6.3.1.1	Mínimos quadrados recursivo e variável instrumental	62
6.3.1.2	Computação evolucionária	66
6.3.1.2.1	Identificação do processo de nível	66
6.3.1.2.2	Identificação do processo térmico	70
6.3.1.3	Redes neurais artificiais	74
6.3.2	Identificação do turbo-gerador	78
6.3.2.1	Mínimos quadrados recursivo e variável instrumental	81
6.3.2.2	Computação evolucionária	84
6.3.2.3	Redes neurais artificiais	87
6.4	Conclusão	91
7	Aplicações em controle de processos	93
7.1	Introdução	93
7.2	Descrição dos Processos	93
7.2.1	Sistema de regulação de nível com dois tanques acoplados	93
7.2.2	Sistema dinâmico não-linear para a manobra de um caminhão até a garagem	94
7.3	Controle de processos	96
7.3.1	Controle do sistema de regulação de nível com dois tanques acoplados	97
7.3.2	Controle de um caminhão para estacionamento em uma garagem	106
7.4	Conclusão	128
8	Conclusões gerais e perspectivas para futuros trabalhos	129
8.1	Identificação de processos	129
8.2	Controle de processos	130
8.3	Pesquisa futura e perspectivas na área de controle inteligente	132
	Referências bibliográficas	134

Lista de figuras

	pág.
2.1 - Diagrama básico dos componentes de um sistema de controle	6
2.2 - Controle digital de processos via computador	7
2.3 - Procedimento para identificação de sistemas	8
2.4 - Interação de metodologias inteligentes em controle de processos	16
3.1 - Pseudocódigo básico de um AE	22
3.2 - Ciclo básico dos dados nos AGs	23
3.3 - Fluxo básico dos dados no algoritmo SA	29
3.4 - O efeito de mutações simples (esquerda) e mutações correlatas (direita)	31
4.1 - Estrutura e fluxo dos dados do controlador lógico nebuloso	39
4.2 - Parâmetros de projeto de um controlador lógico nebuloso	40
4.3 - Configuração de CLN tipo (PD+I)	43
5.1 - Rede neural multicamadas “feedforward” com treinamento BP	47
5.2 - Pseudocódigo do algoritmo heurístico para treinamento de RN	49
5.3 - Rede neural parcialmente recorrente de Elman	52
5.4 - Rede neural parcialmente recorrente de Elman modificada	54
5.5 - Controlador PID com configuração neural	57
6.1 - Diagrama de configuração do processo de nível	60
6.2 - Diagrama de configuração do processo térmico	61
6.3 - Estrutura hierárquica para aplicação da VI com módulo de configuração genético	63
6.4 - Estrutura dos cromossomos utilizados nos AGs para configuração da VI	64
6.5 - Comparação do critério ISE nos 10 experimentos (média) do processo de nível	68
6.6 - Resposta real e estimada do processo de nível em malha aberta via CE	69
6.7 - Resíduo do processo de nível em malha aberta via CE	69
6.8 - Resposta real e estimada do processo térmico em malha aberta via CE	73
6.9 - Resíduo do processo térmico em malha aberta via CE	73
6.10- Resposta real e estimada do processo de nível em malha aberta via RNs	75
6.11 - Resíduo do processo de nível em malha aberta via RNs	76
6.12 - Curva de aprendizado do processo de nível via RNs	76
6.13 - Resposta real e estimada do processo térmico em malha aberta via RNs	76
6.14 - Resíduo do processo térmico em malha aberta via RNs	77

6.15 - Curva de aprendizado do processo térmico via RNs	77
6.16 - Sinal de entrada aplicado ao turbo-gerador em malha aberta (experimento 1)	79
6.17 - Sinal de entrada (experimento 2) aplicado ao turbo-gerador em malha aberta	80
6.18 - Resposta real e estimada do turbo-gerador em malha aberta através da VICAG (experimento 1)	82
6.19 - Resíduo do turbo-gerador em malha aberta através da VICAG (experimento 1)	83
6.20 - Resposta real e estimada do turbo-gerador em malha aberta através da VICAG (experimento 2)	84
6.21 - Resíduo do turbo-gerador em malha aberta através da VICAG (experimento 2)	84
6.22 - Resposta real e estimada do turbo-gerador em malha aberta via CE (experimento 1)	85
6.23 - Resíduo do turbo-gerador em malha aberta através via CE (experimento 1)	86
6.24 - Resposta real e estimada do turbo-gerador em malha aberta via CE (experimento 2)	86
6.25 - Resíduo do turbo-gerador em malha aberta através via CE (experimento 2)	87
6.26 - Resposta real e estimada do turbo-gerador em malha aberta via RN (experimento 1)	89
6.27 - Resíduo do turbo-gerador em malha aberta via RN (experimento 1)	89
6.28 - Curva de aprendizado do turbo-gerador via RN (experimento 1)	90
6.29 - Resposta real e estimada do turbo-gerador em malha aberta via RN (experimento 2)	90
6.30 - Resíduo do turbo-gerador em malha aberta via RNs (experimento 2)	91
6.31 - Curva de aprendizado do turbo-gerador via RNs (experimento 2)	91
7.1 - Sistema de regulação de nível não-linear composto de dois tanques acoplados	94
7.2 - Descrição do sistema caminhão com carroceria	95
7.3 - Configuração do controlador PID com sintonia por CE	98
7.4 - Comparação da função custo nos 10 experimentos (média) para sintonização do PID	98
7.5 - Saída e controle para mudanças de referência do PID-CE	101
7.6 - Saída e controle para mudanças de referência do PID-CLN	101
7.7 - Saída e controle para mudanças de referência do PID-RN	101
7.8- Saída e controle para mudanças de referência do CLIT-CE	102
7.9 - Saída e controle para mudanças de referência do RN-CE	102
7.10 - Saída e controle para mudanças de referência e distúrbios de carga do PID-CE	103
7.11 - Saída e controle para mudanças de referência e distúrbios de carga do PID-CLN	103
7.12 - Saída e controle para mudanças de referência e distúrbios de carga do PID-RN	103
7.13 - Saída e controle para mudanças de referência e distúrbios de carga do CLIT-CE	104

7.14 - Saída e controle para mudanças de referência e distúrbios de carga do RN-CE	104
7.15 - Configuração de controladores PID em paralelo com sintonia por CE	107
7.16 - Resultados do controle do caminhão com duas entradas (x,y) via PID-CE (experimento 1)	110
7.17 - Resultados do controle do caminhão com duas entradas (x,y) via CLN (experimento 1)	111
7.18 - Resultados do controle do caminhão com duas entradas (x,y) via RN-CE (experimento 1)	111
7.19 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via PID-CE (experimento 1)	113
7.20 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via CLN (experimento 1)	114
7.21 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via RN-CE (experimento 1)	114
7.22 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via PID-CE (experimento 1)	116
7.23 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via CLN (experimento 1)	117
7.24 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via RN-CE (experimento 1)	117
7.25 - Resultados do controle do caminhão com duas entradas (x,y) via PID-CE (experimento 2)	120
7.26 - Resultados do controle do caminhão com duas entradas (x,y) via CLN (experimento 2)	121
7.27 - Resultados do controle do caminhão com duas entradas (x,y) via RN-CE (experimento 2)	121
7.28 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via PID-CE (experimento 2)	123
7.29 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via CLN (experimento 2)	124
7.30 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via RN-CE (experimento 2)	124

7.31 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via PID-CE (experimento 2)	126
7.32 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via CLN (experimento 2)	127
7.33 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via RN-CE (experimento 2)	127

Lista de tabelas

	pág.
3.1 - Diferenças entre algoritmos genéticos e estratégias de evolução	33
4.1 - Propriedades e operações da lógica nebulosa	37
6.1 - Identificação do processo de nível e térmico pela VI e MQR	65
6.2 - Parâmetros usados nas metodologias de CE	66
6.3 - Identificação do processo de nível por técnicas de computação evolucionária	67
6.4 - Identificação do processo térmico por técnicas de computação evolucionária	71
6.5 - Identificação do processo térmico com estimação do atraso de transporte via CE	72
6.6 - Identificação do processo de nível via RNs	75
6.7 - Identificação do processo térmico via RNs	75
6.8 - Termos NARX obtidos pelo algoritmo VICAG (experimento 1)	82
6.9 - Termos NARX obtidos pelo algoritmo VICAG (experimento 2)	83
6.10 - Identificação do turbo-gerador por computação evolucionária	85
6.11 - Identificação do turbo-gerador via RNs	88
7.1 - Paradigmas da CE utilizados na sintonia de controle inteligente	98
7.2 - Base de regras estipulada para o controlador PID-CLN	99
7.3 - Resultados da fase de sintonia no controle do processo de nível duplo-acoplado	100
7.4 - Resultados do controle do processo de nível duplo-acoplado	105
7.5 - Comparação entre as técnicas de controle inteligentes	106
7.6 - Resultados do controle do caminhão com duas entradas (x,y) (experimento 1).	109
7.7 - Resultados do controle do caminhão com três entradas (x,y,θs) (experimento 1)	112
7.8 - Resultados do controle do caminhão com quatro entradas (x,y,θs,θc) (experimento 1)	115
7.9 - Resultados do controle do caminhão com duas entradas (x,y) (experimento 2).	119
7.10 - Resultados do controle do caminhão com três entradas (x,y,θs) (experimento 2)	122
7.11 - Resultados do controle do caminhão com quatro entradas (x,y,θs,θc) (experimento 2)	125

Simbologia

Abreviaturas

<i>sigla</i>	<i>significado</i>
AAH	algoritmo adaptativo heurístico
AIC	“Akaike information criterion”
AG	algoritmo genético
ARMAX	“AutoRegressive Moving Average model structure with eXogenous inputs”
BAM	“Bidirectional Associative Memory”
BJ	“Non-linear Box-Jenkins model structure”
BP	algoritmo da retropropagação do erro (“error Back-Propagation”)
CC	corrente contínua
CE	computação evolucionária (ou evolutiva)
CLIT	controle linear invariante no tempo
CLN	controlador lógico nebuloso
DARPA	“Defense Advanced Research Projects Agency”
DNA	ácido desoxirribonucléico
EP	elemento de processamento
ES	estratégia de evolução
IC	inteligência computacional (ou artificial)
ISE	“Integral Square-Error”
ITAE	“Integral-of-Time-multiplied Absolute-Error”
ITSC	“Integral-of-Time-multiplied Square-Control”
ITSE	“Integral-of-Time-multiplied Square-Error”
LN	lógica nebulosa
MLP	“multilayer perceptron”
MQR	mínimos quadrados recursivo
NARMAX	“Non-linear AutoRegressive Moving Average model structure with eXogenous inputs”
NARX	“Non-linear AutoRegressive model structure eXogenous inputs”
NFIR	“Non-linear Finite Impulse Response model”
NOE	“Non-linear Output Error model”
PD	proporcional mais derivativo
PI	proporcional mais integral
PID	proporcional mais integral mais derivativo
PRBS	“Pseudo-Random Binary Sequence”
RPE	“Relative Prediction-Error”
RN	rede neural artificial
RNPREE	rede neural parcialmente recorrente de Elman
RNPREMOD	rede neural parcialmente recorrente de Elman modificada
RNRE	rede neural recorrente
SA	“simulated annealing”
SSE	“Sum of the Square-Error”
VI	variável instrumental
VICAG	variável instrumental com configuração genética

Capítulo 1

Introdução

1.1 Motivação e relevância do trabalho

A união da ciência da computação e as potencialidades da inteligência computacional têm um papel relevante no contexto do desenvolvimento de “software” e utilização de recursos computacionais vinculados a área de automação industrial.

Esta combinação constitui-se de uma necessidade no contexto de pesquisa, implementação, adaptabilidade e integração de metodologias convencionais, - preditivas, adaptativas e inteligentes, via controle digital de processos por computador, tratando as incertezas e atendendo as necessidades exigidas em projetos de algoritmos em controle de processos.

As necessidades de projetistas e engenheiros, de trabalhar com sistemas de controle cada vez mais complexos, tem viabilizado a identificação e o controle de sistemas não-lineares pela utilização da inteligência computacional, a fim de satisfazer a requisitos em projeto e em implementação na área de automação industrial, onde os algoritmos convencionais apresentam dificuldades na modelagem e controle (Fukuda and Shibata, 1992; RayChaudhuri *et al.*, 1996).

A progressiva implementação destas técnicas, a partir dos anos 70, vem de encontro também a crescente necessidade de eficiência das plantas industriais, aumento da qualidade dos produtos, diminuição dos impactos nocivos ao meio ambiente, diminuição de perdas e outros fatores vinculados ao aprimoramento e tratamento de deficiências de métodos clássicos em sistemas de controle.

O desenvolvimento da inteligência computacional (IC), por pesquisadores das mais diversas áreas do conhecimento, tem levado a promissores resultados quanto a aplicabilidade das técnicas inteligentes, não só no meio científico e tecnológico, mas também econômico, comercial, financeiro, político, social e médico. As aplicações industriais, militares e em produtos comerciais, empregando técnicas bem sucedidas como computação evolucionária, lógica nebulosa e redes neurais artificiais, têm levado a crescentes investimentos em pesquisa na área, principalmente por grandes centros, órgãos governamentais e privados dos Estados Unidos,

Japão e Europa (Coelho e Coelho, 1996g, EvoNews, 1996; Katayama *et al.*, 1993; Narendra, 1996).

O paradigma computação evolucionária tem suas principais áreas de aplicação em problemas de otimização, projeto, roteamento e escalonamento, sendo particularmente significativa em aplicações na indústria aeroespacial, automotiva, telecomunicações e engenharia química. Existem muitos exemplos da colaboração entre a academia e a indústria, um deles é a adoção de metodologias híbridas abrangendo diversos algoritmos que compõem o paradigma da IC, obtendo-se assim poderosas ferramentas para implementações, aumentando a robustez e a flexibilidade de algoritmos em automação industrial (EvoNews, 1996; Soucek and The IRIS Group, 1992; Tanomaru, 1995).

As aplicações da lógica nebulosa tem focalizado, com sucesso, problemas de decisão, planejamento, predição, identificação de processos não-lineares, controle adaptativo nebuloso (direto e indireto), projetos de sistemas auto-ajustáveis e controle lógico nebuloso adotado para a configuração de controladores PD, PI ou PID (Hirota and Sugeno, 1995).

A lógica nebulosa tem sido, das metodologias da IC, a mais utilizada, principalmente por empresas japonesas, que combinam os progressos da IC aliados a robótica e a eletrônica no aprimoramento do meio industrial e, conseqüentemente, introduzindo no mercado uma grande variedade de bens de consumo (eletrodomésticos e componentes automotivos) com emprego maciço de lógica nebulosa.

As redes neurais artificiais, devido as características de aprendizado, adaptabilidade, paralelismo inerente e tolerância a falhas, têm se consolidado com uma promissora ferramenta para reconhecimento de padrões, predição, identificação, controle inteligente e adaptativo.

As aplicações das redes neurais artificiais abrangem diversas metodologias em identificação de processos, combinadas muitas vezes a eficientes métodos numéricos e algoritmos de otimização, e em controle de processos ressalta-se a importância dos desenvolvimentos na área de controle combinando-se redes neurais artificiais com algoritmos de controle adaptativo e outros paradigmas da IC (Narendra and Parthasarathy, 1990; Narendra, 1996; Simpson, 1995; Sjöberg, 1995).

Em síntese, as metodologias da IC tem sido alvo de intensas pesquisas gerando nos últimos anos inúmeras aplicações, entre estas, ambientes de desenvolvimento, “hardware” com projeto inteligente e diversas patentes de projetos ligadas ao aprimoramento do meio industrial e uso comercial dos resultados obtidos (Katayama *et al.* 1993; Watkin and Rau, 1996).

As metodologias da IC, descritas e analisadas neste trabalho, em identificação e controle de processos são os paradigmas da computação evolucionária, o algoritmo “simulated annealing”, a lógica nebulosa, as redes neurais artificiais e alguns sistemas híbridos combinando algoritmos inteligentes.

1.2 Objetivo

Esta dissertação objetiva a descrição, projeto e implementação da inteligência computacional no contexto de identificação e controle digital de processos. A proposta da dissertação é a utilização de cinco metodologias:

- i) computação evolucionária (algoritmos genéticos e estratégias de evolução);
- ii) “simulated annealing”;
- iii) lógica nebulosa;
- iv) redes neurais artificiais (“feedforward” e parcialmente recorrente de Elman modificada); e
- v) sistemas híbridos inteligentes.

O projeto de sistemas híbridos inteligentes é composto de algoritmos genéticos híbrido com “simulated annealing”, controle nebuloso com ajuste de parâmetros via computação evolucionária e algoritmos combinando redes neurais artificiais com o paradigma da computação evolucionária.

Neste trabalho avalia-se também os algoritmos convencionais de identificação dos mínimos quadrados recursivo e variável instrumental, e o algoritmo de controle PID. Testes experimentais e análise do desempenho dos algoritmos em identificação de processos são realizados na identificação de:

- i) processo de temperatura;
- ii) processo de nível mono-tanque; e
- iii) modelo matemático de um turbo-gerador.

Os experimentos em controle abordaram um processo não-linear de nível, composto de dois tanques acoplados, e o sistema dinâmico não-linear de manobra para estacionar um caminhão em uma garagem (“truck backer-upper”).

1.3 Organização da dissertação

A dissertação está estruturada como segue:

No *capítulo 2* apresenta-se as noções da terminologia utilizada em controle de processos e uma visão geral de alguns métodos clássicos e avançados em identificação e controle de processos. Os métodos de identificação apresentados são os mínimos quadrados recursivo e o algoritmo da variável instrumental, enquanto o método convencional de controle introduzido é o controle PID e controle linear invariante no tempo.

Aspectos relativos a metodologias da inteligência computacional em identificação e controle de processos são apresentados, servindo como introdução aos *capítulos 3, 4 e 5*, que referem-se as metodologias inteligentes utilizadas.

O *capítulo 3* descreve alguns paradigmas da computação evolucionária, tais como: algoritmos genéticos, algoritmos genéticos híbridos com “simulated annealing” e estratégias de evolução. O tratamento e utilização destas técnicas com ferramentas em identificação e sintonia de controladores é descrita.

No *capítulo 4* são abordadas definições básicas e as características vinculadas a análise e o projeto da aplicação de controle lógico nebuloso tipo PD e PID aliado a otimização através de computação evolucionária.

O *capítulo 5* trata a abordagem via redes neurais artificiais em identificação e controle de processos. As topologias “feedforward” e recorrentes são abordadas, além da possibilidade de hibridização com o paradigma computação evolucionária.

A descrição dos processos monovariáveis de: i) temperatura; ii) nível mono-tanque; e iii) turbo-gerador utilizados nos ensaios de simulação são tratados no *capítulo 6*. Neste mesmo capítulo são também apresentados resultados experimentais e análise do desempenho decorrente da aplicação de técnicas inteligentes em identificação dos processos descritos.

O *capítulo 7* apresenta a descrição dos processos não-linear de nível, composto de dois tanques acoplados, e o de manobra para estacionar um caminhão em uma garagem (“truck backer-upper”). Os resultados e análise das simulações da aplicação das metodologias da inteligência computacional em controle dos processos descritos são tratadas.

O *capítulo 8* apresenta conclusões gerais da dissertação e perspectivas para aprimoramento do presente trabalho para futuros trabalhos na área de controle inteligente.

Capítulo 2

Controle clássico e controle avançado

2.1 Introdução

Os primórdios das pesquisas e desenvolvimento na área de automação industrial remanam a antes de Cristo. A partir desta época o trabalho muscular humano começou progressivamente a ser substituído pelo trabalho das máquinas. As conseqüências sociais, econômicas e políticas deste fato são profundas e incontestáveis, principalmente a partir da metade do século XX, com eclosão do emprego das linhas de montagem no meio industrial (Mayr, 1970). O surgimento de um aparato abrangente em sistemas de controle, aliado ao progresso e as pesquisas em instrumentação, eletrônica, informática, robótica, planejamento e gerenciamento de produção, são aspectos engajados ao desenvolvimento tecnológico.

O aprimoramento contínuo das técnicas de identificação e controle é necessário no tratamento de processos industriais, cada vez mais complexos, proporcionando uma utilização adequada da tecnologia, com o aproveitamento do potencial dado pela interligação e velocidade de processamento de computadores. Os investimentos nos diversos pólos industriais têm sido significativos em técnicas que proporcionam um aumento de produtividade, flexibilidade e robustez em vários setores da indústria.

Este capítulo visa a apresentação de noções básicas da terminologia utilizada em automação industrial, além de abordar algumas metodologias em identificação e controle de processos. Uma breve visão da teoria de controle é dada a seguir, tratando o significado e os conceitos da terminologia básica de cada componente que descreve um sistema de controle. Um diagrama dos componentes básicos é ilustrado na figura 2.1.

Em decorrência deste fato, a primeira pergunta que usualmente surge, de um neófito na área, é: *O que é controle de processos?* Existem várias definições na literatura, contudo menciona-se em poucas palavras que a área de controle de processos tem por diretriz a manutenção do comportamento de processos físicos dentro de uma região operacional eficiente, estável e segura, atendendo a requisitos e restrições de projeto, obtendo uma melhor relação custo e benefício em operação.

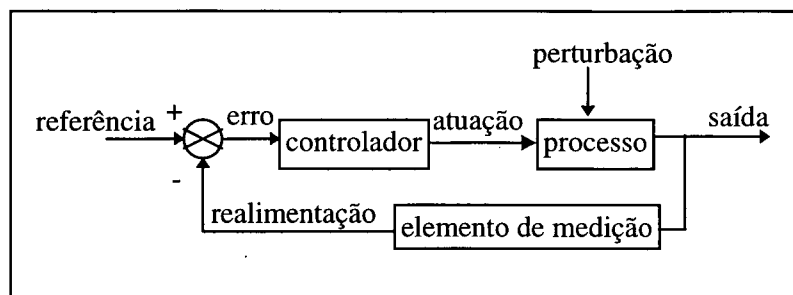


Figura 2.1 - Diagrama básico dos componentes de um sistema de controle.

O *processo* constitui-se de qualquer operação a ser controlada, enquanto o *controlador* é o elemento responsável pela análise do sinal de erro e aplicação do sinal de atuação necessário para corrigir ou minimizar, de maneira apropriada, as especificações (por exemplo: instabilidade, tempo de estabilização, sobrelevação, esforço de controle) quanto ao sistema, de acordo com uma estratégia (ação, lei) de controle. Uma das formas, para realização do controle de um processo, é por meio das informações a respeito da saída do processo controlado, através da realimentação da saída visando a determinação de uma eficiente lei de controle para atuar no processo. O tipo de controlador, selecionado para qualquer aplicação em particular, depende dos requisitos e restrições impostas pelo projetista do sistema de controle.

A entrada de *referência* (“set-point”) é a saída desejada ou estipulada pelo projetista para o processo. O *erro* constitui-se no valor dado pela diferença entre a referência e a saída do processo. O *elemento de medição ou sensor* constitui-se de um dispositivo que converte a variável de saída em outra variável, tal como um deslocamento, uma pressão ou uma voltagem, que pode ser usada para comparar a saída e a entrada de referência.

A *perturbação* (distúrbio) é um sinal que tende a afetar adversamente o valor da saída de um sistema. As perturbações mencionadas no decorrer do trabalho são: *mudança de referência*, que constitui-se no ajuste intencional do ponto de atuação de um controlador em particular, e *perturbação de carga*, que consiste em alguma mudança ou variação à variável controlada (saída), refletindo-se na necessidade do processo em atender as demandas operacionais impostas pelo projeto da malha de controle (Ogata, 1990).

Os sistemas de controle podem ser divididos em *sistemas de malha fechada* e *sistemas de malha aberta*. Nos *sistemas de controle de malha fechada* (ou realimentados) o sinal do erro atuante, que denota a diferença entre o sinal de entrada e o sinal realimentado, é introduzido no controlador de maneira a reduzir o erro e aproximar a saída do sistema a entrada de referência, enquanto no *sistema de controle de malha aberta*, a saída não tem nenhum efeito sobre a ação de

controle, ou seja, a saída não é medida nem realimentada para comparação com a entrada (Ogata, 1990).

Os processos que apresentam apenas um sinal de entrada e um sinal de saída são denominados *processos monovariáveis*, enquanto os processos que apresentam diversos sinais de entrada e/ou diversos sinais de saída são denominados de *processos multivariáveis*.

O controle de processos que utiliza-se do computador para atuação é denominado de *controle digital*, onde emprega-se um controle de tempo discreto dado por equações a diferenças (equações discretas no tempo discreto) em vez de equações diferenciais (equações contínuas no tempo), conforme apresentado na figura 2.2.

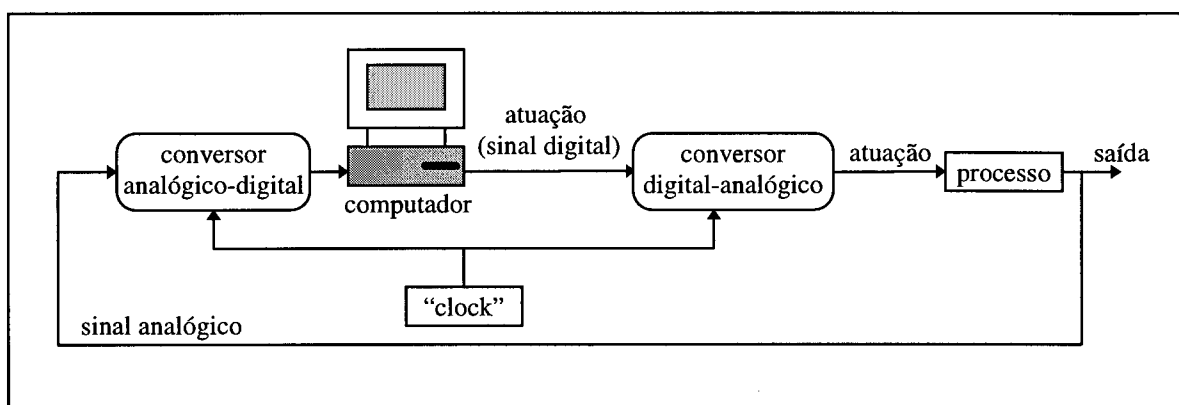


Figura 2.2 - Controle digital de processos via computador.

Na figura 2.2 são introduzidos dois novos termos. O *atuador* constitui-se de um mecanismo que utiliza sinais pneumáticos, hidráulicos ou eletrônicos. Outro termo é o *conversor* que constitui-se de um dispositivo que recebe um sinal elétrico de entrada em uma forma e o converte para outra. Um *conversor analógico-digital (A/D)* aceita uma voltagem analógica e produz uma palavra digital representando o valor da voltagem. Estes dispositivos são usados nas entradas de controladores digitais.

Um *conversor digital-analógico (D/A)* produz uma tensão ou corrente analógica em resposta a uma entrada digital, sendo estes utilizados entre o controlador e o elemento de controle (atuador).

2.2 Técnicas de identificação

A idéia da identificação de sistemas é permitir a construção do modelo matemático de um sistema dinâmico, baseado em dados coletados pelo ajuste de parâmetros do modelo, até que a saída do coincida “tão bem”, quanto possível, com as saídas medidas (Ljung, 1995). Os procedimentos envolvidos na elaboração de modelos matemáticos são classificados em:

- i) *análise físico-matemática*: consiste num importante procedimento quando um modelo deve ser construído sem a obtenção de medidas no sistema, e
- ii) *identificação experimental*: quando se disponibiliza as medidas do processo a ser controlado. O procedimento para identificação experimental de processos consiste das etapas básicas, conforme mostrado na figura 2.3.

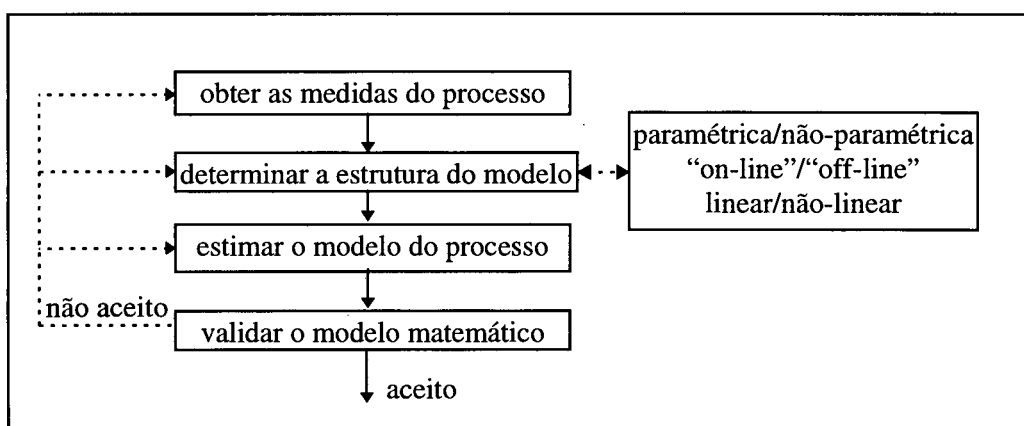


Figura 2.3 - Procedimento para identificação de sistemas.

A realização do experimento proporciona a *obtenção das medidas do processo* (dados experimentais). Assim após escolhido o período de amostragem (T_s), coleta-se um conjunto de dados (visando-se a obtenção de um modelo discretizado), sendo

$$Z^N = \{ [u(t),y(t)] \mid t = 1,\dots,N \} \tag{2.1}$$

onde $u(t)$ é o conjunto de entradas do sistema, dado pelo sinal de controle, $y(t)$ representa os sinais de saída medidos e “ t ” especifica os instantes de amostragem múltiplos de T_s ($T_s, 2T_s, 3T_s, \dots$)

Na fase seguinte, a *determinação da estrutura do modelo*, adequada ao sistema observado, é realizada a partir de um conjunto de modelos candidatos, baseado em algum conhecimento a priori sobre o sistema a ser identificado e as perturbações, estes necessários a

escolha do conjunto de regressores para aplicação da técnica de identificação adotada, seja esta do tipo paramétrica ou não-paramétrica, linear ou não-linear.

A *estimação do modelo* consiste na seleção de modelos para uma classe com estrutura especificada. No caso da estimação paramétrica objetiva-se aproximar os parâmetros desconhecidos do modelo. A partir de Z^N , a *validação do modelo matemático* é efetuada pela comparação do modelo obtido com as medidas em um conjunto de dados que não foi utilizado na estimação do modelo matemático (Ljung, 1987; Ljung, 1995).

A teoria e a prática, relativos a identificação de sistemas, tem recebido grande atenção nas últimas duas décadas e a literatura apresenta uma grande diversidade de técnicas para modelagem de processos industriais. Os métodos clássicos de identificação de processos, tais como resposta ao degrau, análise espectral, resposta em frequência, não fazem parte do escopo de abrangência deste trabalho, cabendo mencionar que geralmente estes apresentam inadequação e imprecisão à aplicação em estruturas de controle preditivas e adaptativas.

Os métodos de identificação tratados são os mínimos quadrados recursivo e o algoritmo da variável instrumental, consistindo de algoritmos robustos na estruturação de técnicas de controle comumente utilizados em identificação “on-line” ou em tempo real e, em estimação de parâmetros para aplicações de controle adaptativo.

2.2.1 Mínimos quadrados recursivo

Em 1809, o alemão Carl Friedrich Gauss expôs num livro com o título “Theoria Motus Corporium Coelestium”, a descoberta do princípio dos mínimos quadrados, com aplicação em astronomia. A partir deste marco, a literatura apresenta diferentes técnicas à estimação de parâmetros desenvolvidas ao longo do tempo, com suas vantagens, desvantagens e particular área de aplicação. A seguir, é tratada a técnica na sua forma recursiva, o mínimos quadrados recursivo (MQR). No MQR, as medidas de entrada e saída do processo são atualizados a cada período de amostragem, permitindo que o modelo do sistema seja atualizado, sendo esta abordagem usualmente implementada e combinada a controladores adaptativos e “self-tuning”.

O modelo matemático para estimação dos parâmetros de processos monovariáveis, é dado por uma relação básica entre a entrada $u(t)$ e a saída $y(t)$, que é exposto por uma equação a diferenças do tipo

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + v(t) \quad (2.2)$$

onde

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na} \quad (2.3)$$

$$B(z^{-1}) = b_1 + b_2z^{-1} + b_3z^{-2} + \dots + b_{nb}z^{-nb} \quad (2.4)$$

e sendo $y(t)$ e $u(t)$ as variáveis de saída e entrada, respectivamente. O $v(t)$ é uma seqüência aleatória normal $N(0,1)$, com média zero e variância unitária. O objetivo é estimar os polinômios $A(z^{-1})$ e $B(z^{-1})$. O atraso de transporte “ d ”, usualmente é dado como conhecido.

A partir do modelo matemático descrito pela equação (2.2), define-se os vetores de parâmetros θ e o vetor de medidas ϕ de acordo com o algoritmo exposto por (Seborg, 1986) que considera o modelo ARMAX (“AutoRegressive Moving Average model structure with eXogenous inputs”) e a ordem do polinômio $C(z^{-1}) = 1$, onde

$$\theta^T = [a_1, \dots, a_{na}, b_1, \dots, b_{nb}] \quad (2.5)$$

$$\phi^T(t) = [-y(t-1), \dots, -y(t-na), u(t-d), \dots, u(t-nb-d)] \quad (2.6)$$

$$y(t) = \phi^T(t)\theta + \varepsilon(t) \quad (2.7)$$

$$\hat{y}(t) = \phi^T(t)\hat{\theta}(t) \quad (2.8)$$

onde $\varepsilon(t)$ representa o ruído o qual admite-se ser estatisticamente independente da entrada e da saída. As estimativas com o algoritmo MQR podem ser calculadas por:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{y}(t)] \quad (2.9)$$

$$K(t) = P(t-1)\phi^T(t) / [1 + \phi^T(t)P(t-1)\phi(t)] \quad (2.10)$$

$$P(t) = [I_p - K(t)\phi^T(t)] / P(t-1) \quad (2.11)$$

$$\text{trp } P(t) = \sum_{i=1}^{n_a+n_b} p(i, i) \quad (2.12)$$

onde $\hat{y}(t)$ é o valor estimado para a saída $y(t)$ baseado no vetor de parâmetros estimado $\hat{\theta}(t)$, $K(t)$ é o ganho do estimador, $P(t)$ é a matriz de covariância, I_p é a matriz identidade, $p = (n_a + n_b)$, $\text{trp } P(t)$ é o traço da matriz de covariância $P(t)$. O princípio do MQR estabelece que a mais apropriada estimativa de θ (num universo de N medidas) é a que minimiza o quadrado do erro $e(t) = \hat{y}(t) - y(t)$, sendo

$$J(N) = \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{y}(t)]^2 \quad (2.13)$$

onde N é o número de amostras do processo. Diversos ajustes, para tratar complexidades em processos, no algoritmo do MQR são propostos na literatura. Contudo emprega-se, neste trabalho, o fator de esquecimento λ e a reinicialização da matriz de covariância $P(t)$, para adequação do algoritmo ao tratamento de processos variantes no tempo, evitando que o ganho do estimador torne-se demasiadamente “pequeno” (Salgado *et al.*, 1988, Wellstead and Zarrop, 1991). O algoritmo dos MQR com fator de esquecimento leva a obter-se um estimador de parâmetros que minimiza a função custo J_1 , cujo domínio é $\mathcal{R}^{n_a+n_b}$ e o contradomínio é \mathcal{R} , dada por:

$$J_1(N) = \frac{1}{N} \sum_{t=1}^N \lambda^{N-t} [y(t) - \hat{y}(t)]^2 \quad (2.14)$$

O parâmetro λ é responsável pela ponderação às medidas mais recentemente obtidas em detrimento das medidas anteriores e usualmente é definido como um escalar ($0 < \lambda \leq 1$). Outro dispositivo que pode ser incorporado ao algoritmo dos MQR é a reinicialização da matriz de covariância $P(t)$, utilizado como forma de impedir a diminuição dos elementos da $P(t)$, através da adição de uma matriz diagonal constante positiva definida R . A matriz é adicionada quando o $\text{trp } P(t) < \text{trp } P(t)_{\text{mínimo}}$, sendo o valor do $\text{trp } P(t)$ definido pelo projetista.

Assim, com a utilização do fator de esquecimento λ e reinicialização de $P(t)$, o algoritmo básico dos MQR, para o cálculo das estimativas, passa a ser implementado pelas seguintes equações

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{y}(t)] \quad (2.15)$$

$$K(t) = P(t-1)\phi^T(t) / [\lambda + \phi^T(t)P(t-1)\phi(t)] \quad (2.16)$$

$$P(t) = \lambda^{-1} \{P(t-1) - [P(t-1)\phi(t)\phi^T(t)P(t-1) / (\lambda + \phi^T(t)P(t-1)\phi(t))]\} \quad (2.17)$$

$$P(t) = P(t) + R, \quad \text{se } \text{trp } P(t) < \text{trp } P(t)_{\text{mínimo}} \quad (2.18)$$

2.2.2 Algoritmo da variável instrumental

O algoritmo da variável instrumental (VI) tem desempenho adequado para uma vasta gama de modelos de processos, que apresentam ruído. A técnica da VI tem seus fundamentos na teoria de estimação da estatística clássica, onde a VI constitui-se de um método para problemas de estimação estrutural de modelos paramétricos. O algoritmo foi introduzido, em 1941 por Reiersol (Montiel *et al.*, 1987).

O modelo matemático para estimação dos parâmetros em processos monovariáveis é dado por uma relação básica entre a entrada $u(t)$ e a saída $y(t)$, que é exposto por uma equação a diferenças linear, similar ao algoritmo dos MQR dado pelas equações (2.2) a (2.11), exceto as equações do ganho do estimador (equação (2.9)) e a matriz de covariância (equação (2.10)) apresentam alterações, isto é, à introdução do termo da variável instrumental $Z(t)$ conforme apresentado a seguir

$$Z(t) = [-y(t-d_{vi}-1), \dots, -y(t-d_{vi}-n_a), u(t-1), \dots, u(t-n_b-1)] \quad (2.19)$$

$$K(t) = P(t-1)Z(t) / \{(\lambda + \phi^T(t-1)P(t-1)Z(t))\} \quad (2.20)$$

$$P(t) = [P(t-1) + P(t-1)Z(t)\phi^T(t-1)P(t-1) / \{\lambda + \phi^T(t-1)P(t-1)Z(t)\}] / \lambda \quad (2.21)$$

Apesar dos diversos estudos para seleção de $Z(t)$ (aspectos relativos a melhor inicialização dos valores da montagem de $Z(t)$, a reinicialização da matriz $P(t)$ e os valores do atraso (d_{vi}) a ser atribuído a $Z(t)$) não existe um algoritmo determinístico de seleção e existe a

necessidade de estudos empíricos e um conhecimento particular de cada processo onde está sendo aplicada a técnica de estimação da VI (Ljung, 1987; Wellstead and Zarrop, 1991).

2.3 Controle clássico

A maioria dos sistemas de controle, utilizados no meio industrial, baseia-se em mecanismos clássicos de controle, empregando controladores “on-off” e controladores proporcional mais integral mais derivativo (PID) analógico e digital. A seguir são abordadas características e um breve equacionamento de um controlador PID e um controlador linear invariante.

2.3.1 Controle PID

O controlador PID é o mais popular e vastamente utilizado no meio industrial, ocupando uma grande fatia da operacionalização de processos industriais com cerca 90% das malhas industriais, devido a sua simplicidade de implementação e baixo custo (Åström and Hägglund, 1996). O controlador PID contínuo é dado pela equação clássica

$$u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right\} \quad (2.22)$$

onde $e(t)$ é o sinal de erro, K_p é o ganho proporcional, T_i é o tempo integral e T_d é o tempo derivativo. A ação (controle) integral tem por objetivo eliminar o erro em regime permanente (e_{ss}) da saída em relação a referência, enquanto que a ação derivativa fornece uma ação antecipatória para reduzir a sobrelevação (“overshoot”) da variável controlada. Cabe contudo mencionar que ação derivativa como exposta na equação clássica do PID (2.22) não é realizável na prática, havendo a necessidade da aplicação de um filtro passa-baixa, conforme tratado por Åström and Hägglund (1996).

De acordo com um período de amostragem (T_s), a equação (2.22) pode ser discretizada resultando em uma equação a diferenças para aplicação de controle PID digital, onde a parte derivativa é substituída pela diferença de primeira ordem e a parte integral pela aproximação retangular. Assim, a equação do controlador PID digital é dada pela equação a diferenças:

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (2.23)$$

onde

$$q_0 = K_p \left(1 + \frac{T_s}{2T_i} + \frac{T_d}{T_s} \right) \quad (2.24)$$

$$q_1 = -K_p \left(1 + \frac{2T_d}{T_s} - \frac{T_s}{2T_i} \right) \quad (2.25)$$

$$q_2 = K_p \frac{T_d}{T_s} \quad (2.26)$$

A sintonia adequada dos parâmetros K_p , T_i e T_d , de acordo com um período de amostragem (T_s) do PID, proporciona um apropriado comportamento dinâmico ao processo controlado. As técnicas tradicionais de sintonia baseiam-se em métodos como domínio da frequência, heurísticos de Ziegler-Nichols, lugar geométrico das raízes e tentativa e erro, como refinamentos para satisfazer as condições de projeto. No presente trabalho adota-se a sintonia do PID por computação evolucionária como metodologia alternativa às técnicas tradicionais (Coelho e Coelho, 1996b; Hwang and Thompson, 1993).

2.3.2 Controle linear invariante no tempo

A utilização do controlador linear invariante no tempo (CLIT), no presente trabalho, está vinculada a um CLIT de 3ª ordem, onde a sintonia dos parâmetros do controlador é efetuada via computação evolucionária. O controlador de 3ª ordem está representado por

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^3 + b_2 s^2 + b_1 s + b_0} \quad (2.27)$$

onde $U(s)$ é a transformada de Laplace da saída do controlador $u(t)$, e $E(s)$ é a transformada do erro de entrada $e(t)$ (Li *et al.*, 1995b).

A função de transferência (2.27) utilizada no projeto do controlador é discretizada de acordo com a aproximação trapezoidal:

$$z = e^{sT} \cong \frac{e^{sT/2}}{e^{-sT/2}} \cong \frac{1 + sT/2}{1 - sT/2} \cong \frac{2 + sT}{2 - sT} \quad (2.28)$$

de onde conclui-se que a função de transferência pulsada pode ser obtida a partir da função de transferência fazendo-se

$$s \cong \frac{2(z-1)}{T_s(z+1)} \quad (2.29)$$

onde z é o operador deslocamento. Os parâmetros a serem determinados por computação evolucionária são $a_i \in \mathfrak{R} \forall i \in \{0, 1, 2, 3\}$ e $b_i \in \mathfrak{R} \forall i \in \{0, 1, 2\}$. O objetivo do projeto é a seleção de um conjunto apropriado de valores para estes coeficientes, de forma a minimizar-se um índice de desempenho, este vinculado a função de adequabilidade das metodologias de computação evolucionária abordadas no capítulo 3.

2.4 Metodologias da inteligência computacional em identificação e controle

Existem diversos pesquisadores que propõem sua própria definição para *o que é inteligência computacional ou artificial*. Aqui menciona-se uma delas, que diz que “inteligência computacional ou artificial é o ramo da ciência da computação que trata dos equipamentos e técnicas que permitem aos computadores emular alguns aspectos da inteligência, como reconhecimento de fala, a dedução, a inferência, a resposta criativa, a capacidade de aprender com a experiência e de fazer inferências racionais a partir de informações incompletas” (Ximenes, 1993). Contudo outros especialistas afirmam que um computador (ou qualquer máquina) pode ser considerada inteligente se satisfizer o teste de Turing (RayChaudhuri *et al.*, 1996).

A utilização de métodos da IC para projeto e implementação de sistemas de controle automático tem sido extensamente descrita como *controle inteligente*. Uma breve definição da área denominada *controle inteligente* é a que constitui-se de uma disciplina em que algoritmos de

controle são desenvolvidos para emular certas características de sistemas biológicos “inteligentes” (Åström, 1991; Passino, 1996).

Esta área emergente abrange um conjunto de paradigmas bastante amplo e multidisciplinar, onde o desenvolvimento da IC por pesquisadores das mais diversas áreas do conhecimento, tem levado a promissores resultados quanto a abrangente aplicabilidade das técnicas inteligentes, não apenas no meio científico (Mendel, 1995; Katayama *et al.*, 1993).

As aplicações industriais, militares, na medicina e em produtos comerciais que empregam com sucesso técnicas como lógica nebulosa, computação evolucionária, “simulated annealing”, redes neurais artificiais, redes neurais nebulosas, sistemas especialistas e sistemas híbridos inteligentes, tem conduzido a diversas pesquisas na área de controle de processos (Levine, 1996; RayChaudhuri *et al.*, 1996).

A figura 2.4 ilustra a interação entre as técnicas convencionais da teoria de controle e inteligentes na área de automação industrial. Nos próximos capítulos apresenta-se a teoria e tópicos relacionados a implementação e utilização dos paradigmas da computação evolucionária, da lógica nebulosa e as redes neurais artificiais, direcionados a aplicação em identificação e tratamento de problema de sintonia e configuração de controladores na área de controle de processos.

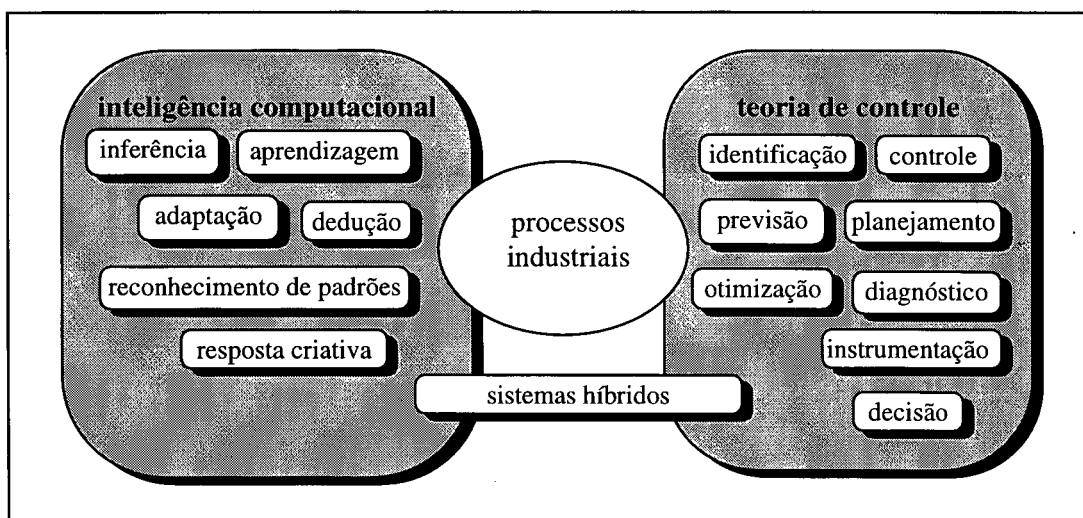


Figura 2.4 - Interação de metodologias inteligentes em controle de processos.

2.5 Critérios de erro

O projeto de sistemas de controle pode ter como objetivo a determinação de uma regra necessária a decisão da ação de controle adequada, sujeita a certas limitações, que minimiza alguma medida de um desvio de comportamento ideal. Esta medida é normalmente provida por um critério de otimização, ou índice de desempenho.

O *índice de desempenho* é uma função cujo valor indica quão adequado o desempenho atual do sistema se compara com o desejado. O índice de desempenho é importante, pois determina, em grande parte, a natureza do controle resultante. O engenheiro de controle (ou operador) determina o índice baseado nos requisitos do problema, deste modo, o engenheiro influencia a natureza do sistema resultante, observando as características gerais consideradas desejáveis, entre elas: confiabilidade, seletividade e aplicabilidade do índice.

Quanto a identificação dos processos, a modelagem dos modelos discretos estimados baseia-se na comparação entre a resposta real do processo “y” e a resposta estimada \hat{y} . A estimação é adequada se os critérios de erro adotados na estimação estão em valores admissíveis, pela análise do projetista. Os critérios de erro em identificação são os seguintes:

i) soma do erro quadrático (SSE - “Sum of the Square-Error”)

$$\text{SSE} = \frac{\sum_{k=1}^N [y(k) - \hat{y}(k)]^2}{N} \quad (2.30)$$

ii) erro de previsão relativo (RPE - “Relative Prediction-Error”)

$$\text{RPE} = \frac{\sum_{k=1}^N [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^N [y(k)]^2} \quad (2.31)$$

iii) integral do erro quadrático (ISE - “Integral Square-Error”)

$$\text{ISE} = \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (2.32)$$

onde $y(k)$ é a saída desejada, $\hat{y}(k)$ é a saída estimada e N é o número de amostras coletadas do experimento. Quanto aos critérios de erro utilizados na análise do desempenho dos algoritmos de controle têm-se os seguintes:

iv) integral do erro quadrático multiplicado pelo tempo

$$ITSE = \sum_{k=1}^N k[y(k) - \hat{y}(k)]^2 \quad (2.33)$$

v) soma do critério de erro ITSE (“Integral Time Squared Error”) e a soma da variação ponderada do sinal de controle, sendo:

$$J(u) = \sum_{k=1}^N f(e(k), u(k)) = \sum_{k=1}^N k(e(k))^2 + w\Delta u(k)^2 \quad (2.34)$$

2.6 Conclusão

Neste capítulo apresentou-se um breve panorama da terminologia e descrições utilizadas em controle de processos. Os componentes básicos de um sistema de controle são enfocados, tais como processo, controlador, entrada de referência, sistemas em malha fechada e aberta, conversores, controle digital, entre outros tópicos.

A seguir, abordou-se os algoritmos de identificação e controle de processos. Na seção de identificação foram tratadas, primeiramente, as fases que compõem a identificação experimental de processos, como sendo:

- i) a obtenção das medidas;
- ii) determinação da estrutura;
- iii) estimação do modelo, e
- iv) validação do modelo estimado.

Uma abordagem básica dos algoritmos de identificação dos mínimos quadrados recursivo e variável instrumental foram apresentados, onde destacou-se as equações que regem os algoritmos, relevantes no contexto de controle adaptativo.

Em controle de processos, a metodologia convencional de controle PID foi abordada, sendo descritos detalhes inerentes a cada um dos três parâmetros que compõem a mais utilizada

técnica de controle no meio industrial, além das vantagens e limitações destas metodologias. Outro controle apresentado foi o controle linear invariante no tempo e suas equações.

Algumas metodologias no contexto dos paradigmas e da aplicação da inteligência computacional em automação industrial também foram tratadas, servindo como introdução aos próximos capítulos sobre os paradigmas computação evolucionária, “simulated annealing”, lógica nebulosa e redes neurais artificiais.

Capítulo 3

Paradigmas da computação evolucionária

3.1 Introdução

A “sabedoria” da natureza manifestada através dos princípios de evolução natural, genética e seleção natural tem inspirado e provado ser um poderoso mecanismo para o aparecimento e aperfeiçoamento de seres vivos no nosso planeta por meio da execução de buscas aleatórias no espaço de possíveis seqüências de DNA (ácido desoxirribonucléico).

O conhecimento corrente sobre as características da evolução natural biológica, levou alguns pesquisadores a tentar utilizar os procedimentos básicos da evolução natural, como embasamento à implementação de algoritmos computacionais de busca estocástica e métodos de otimização em extensos espaços de busca.

O paradigma computacional da computação evolucionária ou evolutiva (CE) imita um modelo rudimentar e simplificado da natureza como um procedimento adaptativo de busca e otimização, que possibilite implementações computacionais. A CE sugere um mecanismo em que uma população $P(t)$ de indivíduos (soluções) visa melhorar, em média, a sua adequabilidade em relação ao ambiente, ou seja, o seu desempenho geral com respeito a um dado problema (Goldberg, 1989; Srinivas and Patnaik, 1994).

Os métodos de busca e otimização são classificados em métodos baseados em cálculo (diretos e indiretos), procura aleatória e enumerativas. Neste trabalho são tratados os métodos de procura aleatória. Os métodos de busca aleatória orientada (“guided random search”) baseiam-se em técnicas enumerativas, contudo utilizam-se de informações adicionais para guiar a busca. Estes métodos são genéricos em seu escopo, sendo capazes de resolver problemas complexos, e suas duas subclasses principais são os paradigmas de “simulated annealing” e da CE (Filho *et al.*, 1994).

O algoritmo “simulated annealing” proposto entre outros pelos pesquisadores S. Kirkpatrick, C. D. Gelatt e M.P. Vecchi (1983), usa um procedimento computacional que simula a evolução termodinâmica de moléculas para procura de um estado final com um mínimo de energia, de acordo com a velocidade imposta por um cronograma de “têmpera”. O algoritmo

genético com ajuste fino por “simulated annealing”, proposto no presente trabalho, constitui-se de uma melhora no desempenho do algoritmo genético frente a buscas locais.

Atualmente, a CE constitui-se numa alternativa às técnicas convencionais em busca e otimização. As aplicações industriais da CE em processos industriais é uma realidade na Europa e Estados Unidos, resolvendo problemas de otimização, projeto, roteamento e escalonamento. Quanto a aplicações na academia tem-se relevantes metodologias de projeto de sistemas de controle, otimização multiobjetivo, controle robusto, identificação de sistemas e integração de sistemas (Chipperfield and Fleming, 1996; Linkens and Nyongesa; 1995).

Quanto a exemplos da aplicação da CE no meio industrial existem sistemas de projeto de bulbos de lâmpadas e turbinas de avião (General Electric); ambiente de simulação de aviões (Beranek & Newmann); propulsão de navios (Naval Surface Weapons Center); roteamento de redes de comunicação (CNET, Cap Volmac e Hewlett Packard Laboratories); escalonamento e controle (Daimler-Benz e Rolls-Roice), entre outras aplicações (Coelho e Coelho, 1996g, EvoNews,1996; Lipton and Rosenof, 1996; Soucek and The IRIS Group, 1992).

A CE engloba um número crescente de metodologias, das quais as mais importantes são (Bäck and Schwefel, 1993; Fogel, 1994; Goldberg, 1989; Tanomaru, 1995):

- *algoritmos genéticos*: desenvolvidos principalmente por A.S. Fraser, H.J. Bremermann, J. Reed e J. Holland entre a década de 50 e 70, com refinamentos posteriores por K. De Jong, J. Grefenstette e D. Goldberg;
- *estratégias de evolução*: desenvolvidas na Alemanha, por I. Rechenberg e H.P. Schwefel;
- *programação evolucionária*: desenvolvidas por L.J. Fogel, A.J. Owens e M. J. Walsh, nos Estados Unidos, na década de 60, refinada recentemente por D.B. Fogel;
- *programação genética*: tratadas por pesquisadores como J.R. Koza, J.P. Rice, P.J. Angeline e K.E. Kinneary; e
- *sistemas classificadores*: abordados na literatura por G.E. Goldberg, S.W. Wilson e J.H. Holland.

O paradigma CE atualmente tende a interagir de modo a dar origem aos denominados algoritmos evolutivos ou evolucionários (AEs). Na figura 3.1 é apresentado um pseudocódigo básico de um AE. A seguir, são descritas e analisadas as técnicas algoritmos genéticos, algoritmos genéticos híbridos com “simulated annealing” e estratégias de evolução, sendo estes paradigmas da IC parte do escopo deste trabalho.

```
geração ← 0
inicialização (P(t))
avaliação da aptidão da população (P(t))
enquanto o critério de parada para P(t) não é atingido
{
    geração ← geração + 1
    P(t) = seleção (P(t-1))
    recombinação (P(t))
    mutação (P(t))
    avaliação da aptidão da população(P(t))
}
```

Figura 3.1 - Pseudocódigo básico de um AE.

3.2 Algoritmos genéticos

Os algoritmos genéticos (AGs) são algoritmos probabilísticos que manipulam um espaço de busca, utilizando operações randômicas na determinação de potenciais soluções, baseadas em mecanismos de seleção natural de Charles Darwin e genética de Gregor Mendel. Os AGs são poderosas ferramentas que provêm uma alternativa às tradicionais técnicas de otimização na efetivação de buscas globais em funções complexas e irregulares.

Contudo, apesar dos esforços de diversos pesquisadores como Uesaka (1995) não existe ainda uma prova formal de convergência dos AGs, além do fato de que o algoritmo não garante a obtenção de uma solução ótima. Apesar destas constatações os AGs têm sido aplicados nos mais diversos problemas que tratam espaços de busca multidimensionais e multimodais (Srinivas *et al.*, 1994; Michalewicz, 1992; Goldberg, 1989).

A seguir, são mencionadas algumas definições necessárias ao entendimento do procedimento de evolução das técnicas de CE (Ramos e Barrone, 1994), tais como:

- *fenótipo*: é o conjunto de características fornecidas pelo genótipo.
- *genótipo*: consiste de um arranjo dos genes.
- *indivíduo*: exemplar de uma espécie, que interage com o meio ambiente.
- *espécie*: grupo de populações com fenótipos semelhantes.
- *população*: é o conjunto de indivíduos da mesma espécie.
- *cromossomo*: componente genético responsável pelo fenótipo do indivíduo.
- *genes*: constituem-se da unidade básica do cromossomo, que define, de acordo com seu valor (alelo) e posição (“locus”), uma característica do cromossomo.
- *reprodução*: multiplicação de novos indivíduos a partir de outros da mesma espécie.

Entre as características básicas que diferenciam os AGs dos métodos tradicionais de otimização, estão que os AGs:

- trabalham com uma codificação de um conjunto de parâmetros, não com eles próprios (quando utiliza-se a representação binária dos cromossomos),
- buscam através de uma população de pontos, não a partir de um ponto isolado,
- usam somente a informação de uma função objetivo (adequabilidade); e
- usam regras de transição probabilísticas, não regras determinísticas (Goldberg, 1989).

A figura 3.2 ilustra o ciclo básico dos dados nos AGs, onde o projeto baseia-se nos seguintes passos:

- i) inicializar uma população de cromossomos;
- ii) avaliar o grau de adaptação de cada cromossomo da população, ou seja, o cálculo da função de adequabilidade;
- iii) selecionar os indivíduos mais aptos de acordo com a estratégia de seleção;
- iv) aplicar os operadores genéticos de recombinação (“crossover”) e mutação;
- v) gerar uma nova população, e
- vi) repetir os passos (ii) a (v) até que uma condição de parada seja satisfeita.

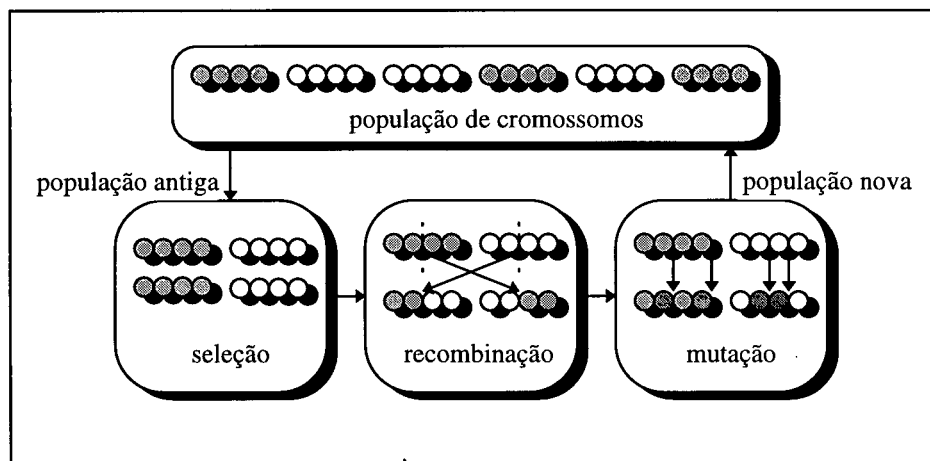


Figura 3.2 - Ciclo básico dos dados nos AGs.

A utilização de AGs requerem a determinação dos seguintes componentes fundamentais:

- representar geneticamente (cromossomos) as parâmetros do problema a serem otimizados;
- definir os operadores genéticos eficazes na reprodução de novos cromossomos, os adequados mecanismos de seleção e uma função de adequabilidade (função de avaliação, função objetivo, adaptabilidade, índice de desempenho ou função custo) dos cromossomos; e

- os parâmetros utilizados nos AGs, tais como: tamanho da população, tamanho da estrutura dos cromossomos, probabilidade de uso dos operadores genéticos, geração da população inicial, critério de término, entre outros.

3.2.1 Representação do cromossomo e fundamentos dos operadores genéticos

A representação do cromossomo é necessária para descrever cada indivíduo da população, sendo este o componente genético responsável pelo fenótipo do indivíduo. O esquema de representação define como o problema é estruturado no AG e determina os operadores genéticos que são utilizados.

Cada indivíduo ou cromossomo é constituído de uma seqüência de genes de um certo alfabeto (binário, inteiros, ponto flutuante, símbolos). No projeto original de John Holland, o alfabeto foi limitado a cadeias de dígitos binários (0 e 1), onde cada “string” representa uma solução potencial para o problema a ser resolvido (representação binária). A teoria tratada por Holland menciona que o alfabeto binário aplicado a AGs tem o processo de seleção, recombinação e mutação, associados a “building blocks” e ao teorema de “Schema” (Goldberg, 1989).

Os “*building blocks*” representam as estruturas genéticas mais bem adaptadas de uma população. Entretanto, somente o procedimento de recombinação pode não evitar a perda de “building blocks” com características promissoras frente a outras estruturas genéticas que possam aprisionar o mecanismo de busca em mínimos (ou máximos) locais.

O *operador de recombinação* não é capaz de explorar regiões do espaço de busca que, eventualmente, não estejam presentes na estrutura genética de uma população. Assim, surge a necessidade da aplicação de outro operador genético, o *operador de mutação*, que complementa o operador de recombinação, auxiliando o algoritmo de busca a escapar das armadilhas criadas pelos mínimos ou (máximos) locais.

O *teorema de “Schema”* afirma que cadeias de tamanho fixo, criadas a partir de alfabetos de baixa cardinalidade, consistem em um meio efetivo de busca através do espaço, onde se encontra a solução de um problema (Goldberg, 1989; Filho *et al.*, 1994).

Desde então, o problema de representação tem estado sujeito a constante investigação e, uma das representações mais usadas para um indivíduo, na otimização de funções envolvem genes de um alfabeto de números de ponto flutuante (representação real), com definição dos valores de limite superior e inferior, estipulados pelo usuário.

O pesquisador Michalewicz tem feito extensivos experimentos comparando AGs com representação por ponto flutuante e binária e, mostra que os AGs com representação por ponto flutuante são mais precisos, consistentes e eficientes em termos de tempo de CPU, além de facilmente serem implementados para tratamento de problemas com restrições, o que justifica a sua escolha no lugar da representação binária (Janikow and Michalewicz, 1991; Michalewicz, 1992; Fogel, 1994).

3.2.2 Operadores genéticos e mecanismos de seleção

O mecanismo de seleção em AGs emula os procedimentos de reprodução e seleção natural. A literatura menciona diversos operadores de seleção, tais como: seleção “ranking”, “ranking” geométrico normalizado, elitista, “tournament”, “steady-state”, entre outros. As estratégias de seleção implementadas são a seleção convencional por roleta e a elitista “breeder”.

Na *seleção por roleta*, cada indivíduo pertencente à população do ciclo de geração corrente, possui uma fatia da roleta proporcional à sua aptidão de sobrevivência. Os indivíduos com maior adequabilidade (aptidão) possuem uma fatia maior na roleta e, assim, tem maior probabilidade de contribuir com um número maior de valores nos indivíduos selecionados, para a realização das operações de recombinação e mutação (Tanomaru, 1995).

O mecanismo de *seleção “breeder”*, constitui-se de um mecanismo de seleção elitista, que consiste na seleção, em cada geração, dos $\beta\%$ melhores elementos na corrente população de n elementos (onde β é o coeficiente de truncamento) e tem valores típicos entre 10% e 50%. Dentro dos $\beta\%$ melhores elementos é salvo o elemento mais apto para próxima geração e são selecionados (de 1 a $n-1$) dois elementos para passarem por operações de recombinação e mutação completando a população (n) (De Falco *et al.*, 1996).

Os operadores evolucionários abordados, são os de mutação não-uniforme e recombinação aritmética, ambos propostos por Z. Michalewicz (Janikow and Michalewicz, 1991; Michalewicz, 1992). A seguir, apresenta-se uma descrição destes operadores:

i) *mutação não-uniforme*: operador responsável pelas capacidades de ajuste fino do sistema. Se $\vec{x}'_t = (x_1, \dots, x_k, \dots, x_q)$ é um cromossomo e o elemento x_k é selecionado para mutação, o resultado é um vetor $\vec{x}'_{t+1} = (x_1, \dots, x'_k, \dots, x_q)$ onde:

$$x'_k = \begin{cases} x_k + \Delta(t, x_{k(\text{direita})} - x_k), & \text{se } z = 0, \\ x_k + \Delta(t, x_k - x_{k(\text{esquerda})}), & \text{se } z = 1 \end{cases} \quad (3.1)$$

o “z” é um número randômico (0 ou 1), $\Delta(t,y)$ é a função que retorna um valor no intervalo $[0,y]$ tal que a probabilidade de $\Delta(t,y)$ inicia em zero e é incrementada de acordo com o número de gerações (t), tal que

$$\Delta(t,y) = y \cdot r \cdot (1-t/T)^b \quad (3.2)$$

onde “r” é um número randômico do intervalo $[0;1]$, “T” é o número máximo de gerações e “b” é um parâmetro escolhido pelo usuário, que determina o grau de dependência com o número de iterações. Esta propriedade leva o operador a efetuar uma busca uniforme dentro do espaço inicial quando “t” é pequeno e, mais localmente, nas gerações posteriores.

ii) *recombinação aritmética*: é definida como combinação linear de dois vetores: se \bar{x}_1 e \bar{x}_2 são recombinados, os cromossomos-filhos resultantes são:

$$\begin{cases} \bar{x}'_1 = \alpha \cdot \bar{x}_1 + (1 - \alpha) \cdot \bar{x}_2 \\ \bar{x}'_2 = \alpha \cdot \bar{x}_2 + (1 - \alpha) \cdot \bar{x}_1 \end{cases} \quad (3.3)$$

Este operador pode usar um parâmetro α , que é uma constante (recombinação aritmética uniforme) ou uma variável que depende do tamanho da população (recombinação aritmética não-uniforme).

3.2.3 Definição dos parâmetros e função de adequabilidade

A definição dos parâmetros intrínsecos aos AGs são estipulados heurísticamente, tais como tamanho da população, tamanho da estrutura dos cromossomos, probabilidade de uso dos operadores genéticos, criação da população inicial, critério de término, entre outros. Não existe uma regra determinística adequada para estipular-se os parâmetros do tamanho de população e probabilidade de aplicação dos operadores genéticos.

A literatura reporta que o AG, com representação binária, na grande maioria das aplicações é utilizado com um tamanho de população entre 30 e 200, probabilidade de recombinação entre 0.5 e 1.0 e, probabilidade de mutação entre 0.001 e 0.05 (Srinikas and Patnaik, 1994). Quanto as funções de adequabilidade, estas são independentes do AG.

Neste trabalho, diversas funções de adequabilidade são empregadas, abordam as funções custo (J) expostas no capítulo 2, sendo que a função da adequabilidade $f(J)$ de um determinado indivíduo é dada por:

$$f(J) = \frac{k_e}{(1+J)} \quad (3.4)$$

onde k_e é o fator de escala. Se o fator de escala $k_e=1$ e $J=0$ tem-se uma suposta condição ótima da função de adequabilidade $f(J)=1.0$, ou seja, de 100%.

3.3 Algoritmo genético híbrido com “simulated annealing”

O algoritmo “simulated annealing” (SA) ou têmpera simulada é uma variação de algoritmos de subida de encosta onde o objetivo é a minimização de uma função objetivo (nível de energia) e baseia-se em uma analogia com a mecânica estatística de materiais com resfriamento, onde substâncias físicas como os metais são elevadas a altos níveis de energia e posteriormente são gradualmente resfriadas até alcançar um estado mínimo de energia.

Sob outras condições, menos cuidadosas de resfriamento, o material se cristalizaria com uma energia “localmente mínima”, o que freqüentemente se traduz em imperfeições estruturais. A esse processo cuidadoso de resfriamento denomina-se “annealing”.

A idéia do SA origina-se numa combinação dessas observações sobre a física dos materiais, com procedimento computacional tendo por finalidade a simulação do comportamento de uma coleção de átomos (variáveis do problema) em condições de temperatura fixa (Kirkpatrick *et al.*, 1983).

Essa simulação a uma temperatura fixa “T” consiste em uma seqüência de passos. A cada passo, é dado um pequeno deslocamento a um dos átomos e é computada a variação ΔE , que a energia do sistema sofre com aquele deslocamento.

Se $\Delta E \leq 0$, o deslocamento é incorporado ao estado do sistema, que é então utilizado para o passo seguinte, contudo se $\Delta E \geq 0$, a aceitação ou não da transição para um estado maior de energia tem uma probabilidade dada pela função

$$p = e^{-\Delta E/kT} \quad (3.5)$$

onde “k” é a constante da distribuição de Boltzmann, “T” é a temperatura e ΔE é a mudança no nível de energia, com a finalidade de simplificar a implementação do SA pode-se ignorar a constante “k” e reescrever a probabilidade de aceitação de uma configuração de energia mais alta como

$$p = e^{-\Delta E/T} \quad (3.6)$$

A figura 3.3 apresenta um procedimento genérico para aplicação do SA, onde a escolha de parâmetros dependem do problema particular em questão, isto é: a escolha da solução inicial x_0 e a escolha do conjunto $S(x) \subseteq D$ de soluções. Entre as soluções, uma delas será selecionada para o teste de aceitação quando a simulação se encontra no ponto x .

Outras escolhas também devem ser levadas em consideração como a definição das temperaturas inicial T_0 e final $T_{\text{mínima}}$, a escolha de equilíbrio a uma dada temperatura, e a definição da função $r(k)$ que determina a redução gradual de temperatura em função de um fator de “annealing” α no intervalo $[0;1]$, que determina a velocidade com que o sistema é resfriado (cronograma de têmpera) onde adota-se a seguinte equação:

$$r(k) = \alpha^{k+1} T_k, \text{ para todo } k \geq 0 \quad (3.7)$$

```

inicializar os parâmetros de temperatura  $T_{\text{inicial}}$  e  $T_{\text{mínima}}$ , e fator de “annealing”  $\alpha$ 
gerar randomicamente um estado inicial  $x_0$  como estado corrente
atribuir iteração  $k = 0$ ;  $x_k = x_0$ ;  $T_k = T_{\text{inicial}}$ ;
fazer enquanto ( $T_k \geq T_{\text{mínima}}$ )
{
  fazer enquanto “o sistema não está em equilíbrio na temperatura  $T_k$ ”
  {
    selecionar aleatoriamente um ponto  $x'$ ;
     $\Delta E = f(x') - f(x)$ ;
    if  $\Delta E \leq 0$  então
       $x_k = x'$ 
    senão
      se  $\text{mínimo}(1, e^{-\Delta E/T}) > (\text{número aleatório entre } [0;1])$  então  $x_k = x'$ ;
  }
   $T_{k+1} = r(k) = \alpha^{k+1} T_k$ ;    $x_{k+1} = x_k$ ;    $k = k + 1$ ;
}
o estado corrente  $x_k$  é a solução.

```

Figura 3.3 - Fluxo básico dos dados no algoritmo SA.

O SA tem encontrado aplicações principalmente em problemas de otimização combinatória, sistemas híbridos com outros métodos numéricos de otimização e técnicas de IC. Neste trabalho visa-se focar a aplicação de SA conjuntamente a AGs. Os AGs se caracterizam por serem robustos em aplicações complexas e seus mecanismos efetuam eficientemente uma procura global.

Entretanto, para buscas locais geralmente não apresentam um comportamento adequado. A convergência do AG pode tornar-se mais rápida e precisa, assim implementa-se um algoritmo genético ajuste fino por “simulated annealing” (AGSA). O processo de “annealing” é aplicado ao AG, a cada geração, após as operações de recombinação e mutação, para sintonização fina dos valores dos cromossomos do AG.

O SA é aplicado a cada geração após as operações de recombinação e mutação, através de uma pequena perturbação no valor dos cromossomos. A sintonia fina dos valores da metade da população dos AG gera a outra metade da população em cada geração, obtendo-se assim o mesmo número de avaliações da função de adequabilidade que os outros algoritmos da CE. Um baixo fator de temperatura T e um rápido fator de “annealing” α são utilizados no AGSA (Li *et al.*, 1995; Ghoshray *et al.*, 1995).

3.4 Estratégias de evolução

As estratégias de evolução (EEs) foram desenvolvidas para resolução de problemas técnicos de otimização em engenharia e, atualmente, constituem-se de importantes algoritmos computacionais em problemas de otimização de parâmetros. As EEs são técnicas que enfatizam a ligação comportamental entre as populações geradas (indivíduos-pais e indivíduos-filhos), mais que a ligação genética.

A primeira EEs a ser desenvolvida foi a EE-(1+1), proposta em 1964, na Technical University of Berlin, por I. Rechenberg e H.-P. Schwefel, como uma técnica para otimização experimental. A EE-(1+1) é um tipo de técnica de gradiente estocástico que apresenta similaridades com o SA, é baseada em uma população consistindo de um indivíduo pai e um indivíduo descendente (filho), criado pela adição de um número randômico normalmente distribuído para cada componente do vetor.

Neste algoritmo, o indivíduo mais apto, de ambos, sobrevive e serve como antecedente (pai) na seguinte geração. O desvio padrão σ das mutações é atualizado de acordo com as mutações bem sucedidas (regra de $\mu/\lambda=1/5$ de I. Rechenberg), onde se o filho é avaliado como mais apto que o pai, ele o substitui (Coelho e Coelho, 1996a; Fogel, 1994; Soucek and The IRIS Group, 1992).

Em contraste com os AGs, as soluções são diretamente representadas em vetores de valores reais $\vec{x} \in \mathcal{R}^n$ e os indivíduos $\vec{a} = (\vec{x}, \vec{\sigma}, \theta)$ consistem não somente do vetor \vec{x} mas também incorporam um vetor adicional $\vec{\sigma} \in \mathcal{R}_+^n$ de desvio padrão positivo σ_i e um conjunto de ângulos de inclinação θ , que influencia as covariâncias das mutações.

Os parâmetros usuais das EEs são utilizados pelo operador mutação, para modificar os correspondentes objetos variáveis x_i ($i \in \{1, \dots, n\}$). A mutação opera em cada objeto variável x_i pela adição de números randômicos normalmente distribuídos com esperança zero e variância σ_i^2 (indicada pela notação $N(0, \sigma_i^2)$). Um novo vetor solução $\vec{a}' = (\vec{x}', \vec{\sigma}', \theta')$ pode ser criado como:

$$\vec{\sigma}'_i = \vec{\sigma}_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_j(0,1)) \quad (3.8)$$

$$\theta'_j = \theta_j \cdot \exp(N_j(0, \Delta\theta)); \quad j \in \{1, \dots, w\} \quad (3.9)$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0, A); \quad i \in \{1, \dots, n\} \quad (3.10)$$

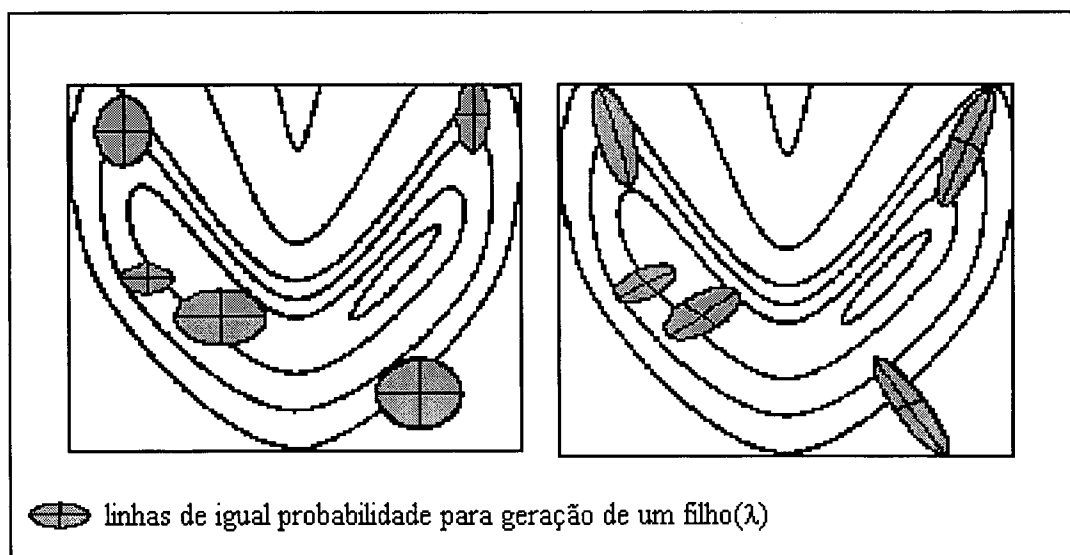
onde a mutação de $\bar{\sigma}_i$ é baseada em um fator de busca global $\tau' \cdot N(0,1)$ e um fator local $\tau \cdot N_i(0,1)$. O termo $N(0,A)$ é um vetor normalmente distribuído z com expectativa zero e densidade de probabilidade

$$p(z) = \sqrt{\frac{\det A}{(2\pi)^n}} \exp\left(-\frac{1}{2} z^T A z\right) \quad (3.11)$$

Os elementos da diagonal da matriz de covariância A^{-1} são variâncias independentes para componentes x_i do vetor de decisão \bar{x} , enquanto os elementos de fora da diagonal representam as covariâncias $c_{i,j}$ das mudanças.

Os estudiosos da técnica Bäck e Schwefel (1993) restringiram as áreas de densidade de probabilidade para hiperelipsóides n -dimensionais, que é realizado por um conjunto de ângulos de inclinação $\theta' \in \mathfrak{R}$, $w = n / 2 \cdot (n - 1)$ para o eixo central do hiperelipsóide.

Os desvios padrões $\bar{\sigma}$ servem como um tipo de média do tamanho de passo ao longo destes eixos. Na figura 3.4 é mostrado o efeito de mutações simples ($\theta = \theta' = 0$) e mutações correlatas.



Fonte: Hoffmeister and Back (1991).

Figura 3.4 - O efeito de mutações simples (esquerda) e mutações correlatas (direita).

Os operadores de recombinação são similares aos implementados em representação por ponto flutuante nos AGs. Entre as opções tem-se a recombinação discreta, a intermediária (local

e global) ou mesmo a não realização da operação de recombinação (Bäck and Schwefel, 1993; Coelho and Coelho, 1996c; Rudolph, 1992).

A literatura de CE apresenta várias técnicas de EEs multimembros (ou multimembrada), entre elas, contudo avaliam-se as seguintes estruturas mencionadas a seguir.

3.4.1 Estratégias de evolução multimembros

A técnica EE-(1+1) vem sendo progressivamente generalizada por variantes do número de ancestrais (pais) $\mu > 1$ e número de descendentes (filhos) por geração $\lambda > 1$. As EEs com vários membros (multimembros) tem o embasamento biológico relacionado a características de *poligenia* (característica do fenótipo de um indivíduo é influenciada por diversos genes) e *pleiotropia* (capacidade de um gene influenciar diversas características dos fenótipos).

As EEs com multimembros são divididas em EE-($\mu+\lambda$) que sugerem a μ pais produzirem λ filhos, após os μ pais e os λ filhos competem pela sobrevivência, ao contrário da EE-(μ,λ), onde os λ filhos competem para sobreviver e os pais são completamente substituídos a cada geração (Coelho e Coelho, 1996a, 1996b; Soucek and The IRIS Group, 1992).

A literatura menciona outros tipos de EE multimembros tais como EE-(1, λ), EE-($\mu+\lambda$), EE-(μ,λ) e EE contemporâneas (μ,κ,λ,ρ). Neste trabalho são tratadas as EE-($\mu+\lambda$) e EE-(μ,λ). Além da implementação da técnica EE-(1+ λ) que sugere apenas que o pai produza os λ descendentes. As duas variantes destas EEs empregadas utilizam o operador de mutação com:

- i) *controle do tamanho de passo individual*: que possibilita o tratamento de problemas inadequadamente escalonados com relação a uma maior sensibilidade dos parâmetros inerentes ao operador de mutação, e
- ii) *variações correlatas e esquema de adaptação dos parâmetros*: onde o algoritmo é complexo, robusto e adequado para efetivação de variações correlatas a nível de fenótipo dos indivíduos (Bäck and Hoffmeister, 1991; Rudolph, 1992).

3.4.2 Características de algoritmos genéticos e estratégias de evolução

Os AGs e as EEs apresentam similaridades no processo de otimização de parâmetros. Algumas características diferem de uma técnica para outra e as principais diferenças estão resumidas na tabela 3.1 (Coelho e Coelho, 1996a; Bäck and Schwefel, 1993).

Tabela 3.1 - Diferenças entre algoritmos genéticos e estratégias de evolução.

características	algoritmos genéticos	estratégias de evolução
<i>representação</i>	binária (nível de genótipo dos indivíduos) ou real*	real (nível de fenótipo dos indivíduos)
<i>auto-adaptação</i>	nenhuma	desvios padrão e covariâncias
<i>função de adequabilidade</i>	não há conhecimento sobre as propriedades da função objetivo	conhecimento da dimensão da função objetivo (número de variáveis)
<i>seleção</i>	probabilística, preservativa	determinística, extintiva
<i>recombinação</i>	operador principal	importante para auto-adaptação (diferentes variantes)
<i>mutação</i>	operador secundário	operador principal

* representação real: utiliza-se dos operadores evolucionários propostos por Michalewicz (1992).

3.5 Conclusão

Este capítulo apresenta os paradigmas de CE através da descrição e análise das metodologias algoritmos genéticos, algoritmos genéticos híbridos com “simulated annealing” e estratégias de evolução, sendo estes paradigmas da IC parte do escopo deste trabalho.

Quanto aos algoritmos genéticos foram apresentadas características, formas de representação do cromossomo, exposição de operadores genéticos e mecanismos de seleção por roleta e “breeder”. O tratamento e tomada de decisão quanto a definição dos parâmetros adequados e função de adequabilidade também foram discutidos.

A seguir, trata-se da implementação de um algoritmo genético híbrido com “simulated annealing” enfocando a adequação do algoritmo “simulated annealing” para ajuste fino nos valores dos algoritmos genéticos. Após apresenta-se a metodologia de EE-(1+1) e EE multimembros, apresentando-se uma tabela que sintetiza as diferenças desta metodologia para os algoritmos genéticos.

Capítulo 4

Metodologias de análise e projeto de controle lógico nebuloso

4.1 Introdução

A idéia básica da lógica nebulosa é o desenvolvimento de uma estrutura apta no tratamento de informações imprecisas ou incompletas. A lógica nebulosa foi primeiramente aplicada com sucesso em sistemas de controle por E.H. Mamdani, em 1974. Contudo, a evolução da área denominada *controle lógico nebuloso* teve inicialmente controvérsias com a comunidade de engenharia de controle, mas atualmente a lógica nebulosa é uma metodologia da IC com abrangente espectro de aplicações, principalmente por empresas japonesas, em controladores lógico nebulosos (Hirota and Sugeno, 1995). ✓

As aplicações da lógica nebulosa tem abrangido com sucesso problemas nas áreas de decisão, planejamento, previsão, identificação e controle de processos não-lineares. As aplicações da LN, e mais especificamente de controle lógico nebuloso, são amplamente enumeradas na literatura, a citar o controle do metrô de Sendai (Hitachi), controle de elevadores (Fujitec, Hitachi, Mitsubishi, Toshiba), auto-foco de câmera de vídeo (Canon, Panasonic, Sanyo & Fisher), reatores nucleares (Hitachi), subsistemas automotivos (Honda, Nissan, Subaru), ar condicionado (Mitsubishi), produção de celulose (Siemens), aeronaves (Rockwell Corp.) e diversas outras (Bonissone, 1994; Coelho e Coelho, 1996g; Hirota and Sugeno, 1995; Munakata and Jani, 1994; Thomas and Armstrong-Hélouvry, 1995).

Entre as vantagens dos controladores lógico nebulosos tem-se a facilidade de dividir o espaço de estado em regiões, possibilidade da integração de conhecimento qualitativo, não necessitam de modelo matemático e são aproximadores universais de qualquer função, com uma dada precisão arbitrária em um intervalo, tal como algumas arquiteturas de redes neurais artificiais (Buckley and Hayashi, 1993a; 1993b).

4.2 Conjuntos nebulosos e a lógica nebulosa

A teoria clássica dos conjuntos é bem conhecida e caracteriza-se por estabelecer a pertinência $\mu_A(x)$ do elemento x em um conjunto A , que é subconjunto do universo X , definido por

$$\mu_A(x) = \begin{cases} 1 \Leftrightarrow x \in A \\ 0 \Leftrightarrow x \notin A \end{cases} \quad (4.1)$$

Isto significa que um elemento x é um membro do conjunto A ($\mu_A(x)=1$) ou não ($\mu_A(x)=0$). Em 1965, Lotfi A. Zadeh introduziu uma nova abordagem para tratamento e solução de diversos problemas do mundo real, que caracterizam-se por aspectos relativos a sua natureza vaga, imprecisa, de contornos indefinidos e, por isso mesmo, extremamente complexos, assim nascia a área denominada de *lógica nebulosa* ("fuzzy logic").

Os fundamentos da lógica nebulosa (LN) levam em consideração o princípio de admitir incerteza e imprecisão na solução de problemas e permitir que se trabalhe com descrições simplificadas da realidade, possibilitando assim que a modelagem de sistemas complexos possam ser efetuadas e conclusões, possivelmente corretas, serem extraídas.

A definição dos conceitos da denominada LN, a partir da extensão da teoria clássica de conjuntos, costuma ser apresentada como um simples trabalho de associação entre conceitos da teoria de conjuntos e conceitos de lógica. Logo, associa-se o conceito de pertinência ao de valor-verdade de uma proposição lógica, o de união, interseção e complemento de conjuntos às operações *ou*, *e* e *não*, respectivamente; a inferência é vista como uma relação entre conjuntos.

Estas associações estão plenamente justificadas no campo da lógica clássica, porque é conhecido o isomorfismo entre álgebra de Boole e a estrutura do cálculo proposicional clássico. No campo da teoria dos conjuntos nebulosos, as inúmeras maneiras de estender as operações clássicas trazem dificuldades quanto à maneira mais adequada e rigorosa de apresentar os fundamentos da lógica nebulosa.

A justificativa básica da aplicação da LN, a determinada classe de problemas, pode ser expressa através do *princípio da incompatibilidade*, determinado por Zadeh. Em essência, o princípio afirma que: "Quando a complexidade de um sistema aumenta, nossa capacidade de fazer afirmações precisas é ao mesmo tempo significativas sobre ele diminui até um limiar no qual a precisão e a relevância tornam-se características mutuamente exclusivas" (Mendel, 1996).

A LN é uma metodologia que possibilita o tratamento de conhecimento incerto e impreciso e, além disso, os conjuntos nebulosos não obedecem as leis da contradição e do meio excluído da teoria dos conjuntos clássicos (“crisp sets”).

O conjunto nebuloso A em X é um conjunto de pares ordenados associados com uma função característica $\mu_A(x): X \rightarrow [0;1]$, onde $\mu_A(x)$ representa o grau de pertinência, atribuindo-se a medida do grau de similaridade de um elemento no universo de discurso X para o subconjunto nebuloso A. A $\mu_A(x)$ indica quanto um elemento pertence a um dado conjunto, assumindo valores em que a pertinência tem graus no intervalo fechado $[0;1]$, ou seja, $\mu_A(x) \in [0;1]$. Importantes classes de curvas para as funções de pertinência são: a triangular, a trapezoidal e a gaussiana (Mendel, 1996; Yager and Filev, 1994).

Um conjunto nebuloso A é denotado por:

$$A = \sum_{i=1}^n \mu_A(x_i) / x_i = \mu_A(x_1) / x_1 + \mu_A(x_2) / x_2 + \dots + \mu_A(x_n) / x_n \quad (4.2)$$

4.2.1 Conceitos e definições básicas

A seguir são tratados alguns conceitos e definições dos elementos que compõem os conjuntos nebulosos, relevantes para determinação dos fundamentos necessários ao desenvolvimento do trabalho. Os conceitos abordados aqui constituem-se de um breve resumo da LN. Um tratamento mais completo desta área de pesquisa pode ser encontrado em Kosko (1992), Mendel (1996) e Yager and Filev (1994).

Uma *variável nebulosa* é uma variável cujos valores são “labels” de conjuntos nebulosos. Por exemplo, a temperatura de um dado processo pode ser uma variável nebulosa assumindo valores *pequena, média, alta*, etc. Estes valores são descritos por intermédio de conjuntos nebulosos, generalizando, os valores de uma variável podem ser sentenças em uma linguagem especificada, neste caso, a variável é uma *variável lingüística*.

Uma *variável lingüística* é uma variável cujos valores não são números, mas palavras ou seqüências de determinada linguagem, caracterizada por uma quintupla dada por um nome (“label”), um conjunto de termos lingüísticos T (por exemplo: baixo, alto, médio), um universo de discurso (X), uma regra semântica (M) e uma regra sintática (G). Para ilustrar, os valores da variável temperatura poderiam ser expressos valores nebulosos como: *alta, não alta, bastante alta, não muito alta, alta mas não muito alta*.

Os valores *nebulosos* são sentenças formadas a partir do “label” *alta*, da negação *não*, dos conectivos *e* e *mas*, e dos qualificativos (“hedges”) *muito* e *bastante*. A variável temperatura, neste contexto, é uma variável lingüística (Tanscheit, 1992).

4.2.2 Operações e propriedades da lógica nebulosa

As operações e propriedades da lógica nebulosa são apresentadas na tabela 4.1. Assume-se que A, B e C são subconjuntos de X (Yager and Filev, 1994).

Tabela 4.1 - Propriedades e operações da lógica nebulosa.

operações		
(a)	união	$C = A \vee B$
(b)	intersecção	$C = A \wedge B$
(c)	complemento relativo	$C = A - B = \text{Max}[0, A-B]$
(d)	complemento (negação)	$A' = X - A = A' = 1 - A$
propriedades / axiomas		
(a)	leis de comutatividade	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
(b)	leis de idempotência	$A \vee A = A$ $A \wedge A = A$
(c)	leis de associatividade	$A \vee (B \vee C) = (A \vee B) \vee C =$ $(A \vee B) \vee C$ $A \wedge (B \wedge C) = (A \wedge B) \wedge C =$ $(A \wedge B) \wedge C$
(d)	leis de distributividade	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
(e)	leis de absorção	$(A \vee B) \wedge B = B$ $(A \wedge B) \vee B = B$
(h)	leis de Morgan	$(A \vee B)' = A' \wedge B'$ $(A \wedge B)' = A' \vee B'$
(g)	lei de involução	$(A')' = A$
convenções		
'		operador de complemento
\vee		operador de máximo (conectivo ou)
\wedge		operador de mínimo (conectivo e)

Além do breve sumário das operações e propriedades básicas apresentadas na tabela 4.1, a LN abrange tópicos relativos a operações de norma, co-norma, conjunção, disjunção, implicação, regra composicional de inferência e relações nebulosas, tratados estes em detalhes por Mendel (1996) e Yager and Filev (1994).

4.3 Controlador lógico nebuloso

A aplicação da LN pode ser classificada em dois níveis:

i) *análise de decisão baseada em LN*: visa a substituição de um especialista humano por sistemas de tomada de decisões baseado em LN. Estes com relevante aplicação em otimização,

“clusterização”, sistemas especialistas, sistemas classificadores, restauração e busca em banco de dados, tomada de decisão em modelos financeiros e econômicos, reconhecimento de padrões, processamento de imagens, implementação de “shells”, linguagens e ambientes de programação, entre outras técnicas que utilizam-se dos benefícios da LN. Engajadas neste sentido estão empresas, tais como American Neurologix, Apronix, Byte Craft, Hitachi American, Hyperlogic, Inform Software, Motorola, Togai Infralogic, entre outras (Munakata and Jani, 1994; Thomas and Armstrong-Hélouvry, 1995);

ii) *controle lógico nebuloso*: Este nível de aplicações é tratado a seguir. Empresas com pesquisas e produtos no mercado tem-se a General Electric, Honeywell, Microelectronics, Motorola, Omron Electronics, SGS-Thompson, Yokogawa, e outras (Bonissone, 1994).

Os controladores baseados em lógica “fuzzy” ou nebulosos (CLNs) usam o paradigma da teoria dos conjuntos nebulosos aliando a simplicidade e a eficiência da LN, para tratar principalmente processos não-lineares complexos. Os CLNs podem ser divididos em três categorias:

- i) CLNs, em que a inferência nebulosa e as regras são empregadas sem o uso de modelos matemáticos precisos; ✓
- ii) CLNs, em que a estrutura é projetada baseada em lógica nebulosa com modelos matemáticos precisos; e ✓
- iii) controladores combinando lógica nebulosa e técnicas convencionais em controle, usadas para auto-sintonia de parâmetros de controle, por exemplo, parâmetros de controle PID.

O projeto dos CLNs apresentam um fluxo de dados que passam por uma fase de nebulização, análise e execução de regras e desnebulização. A fase de *nebulização* (“fuzzyfication”) consiste em atribuir ou calcular um valor, que represente um grau de pertinência pré-definida de entrada, mapeados em um ou mais grupos qualitativos.

A *análise e execução de regras* (expressa por uma coleção de regras de produção SE <condição> ENTÃO <conclusão>) são responsáveis pela avaliação das regras de produção, quando uma regra é ativada, é efetuado um procedimento de cálculo baseado nos valores dos antecedentes e, então é atribuída a saída da regra. A essência e a identidade do controlador nebuloso reside nas regras de produção. ✓

A fase de *desnebulização* (“defuzzyfication”) visa a obtenção de um número (valor “crisp”) que melhor decifre a inferência às regras composicionais, resolvendo adequadamente conflitos entre as ações (Coelho e Coelho, 1995, 1996f; Tanscheit, 1992). ✓

A estrutura e o fluxo dos dados em um CLN é apresentado na figura 4.1. Os métodos de desnebulização mais utilizados são os métodos do critério máximo, média dos máximos, altura, altura modificada e centro de área (centróide), sendo que este último é utilizado no trabalho. O

método do centro da área dita que a saída “crisp”, da fase de desnebulização do CLN, é expressa através da seguinte forma:

$$w_o = \frac{\sum_{i=1}^n A(w_i) \times (w_i)}{\sum_{i=1}^n A(w_i)} \quad (4.3)$$

onde “ w_o ” é o valor da saída desnebulizada, “ n ” é o número de conjuntos nebulosos da variável de saída, $A(w_i)$ corresponde a área do conjunto nebuloso “ i ”, ponderada pelo grau de pertinência, e o w_i é o ponto médio (centróide) do conjunto suporte “ i ” (Kröhling, 1994).

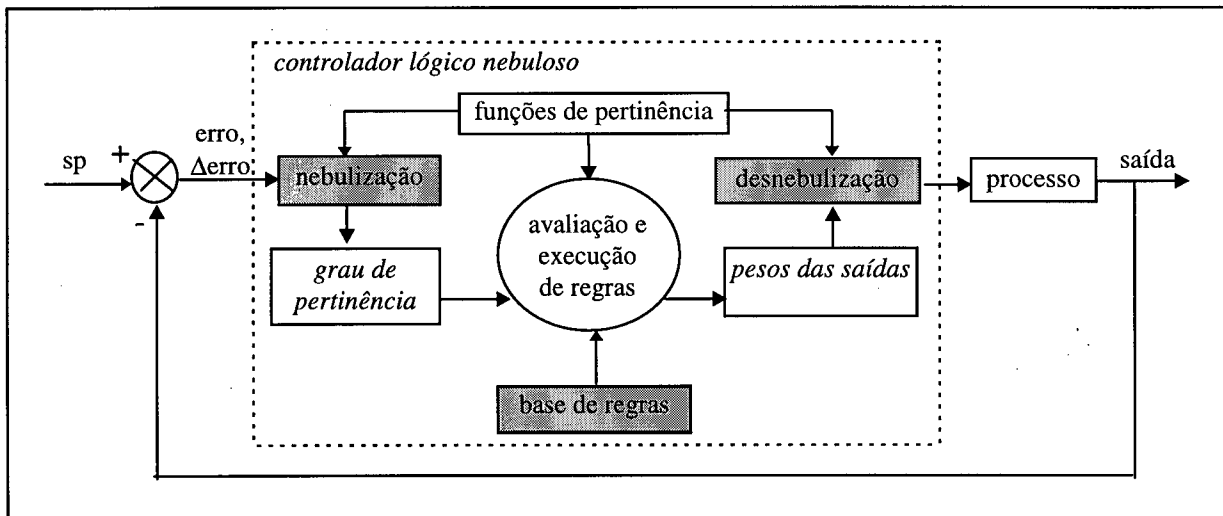


Figura 4.1 - Estrutura e fluxo dos dados do controlador lógico nebuloso.

4.3.1 Parâmetros de projeto

Os principais parâmetros a serem analisados no projeto de um CLN enquadram-se nos tópicos de estratégias de nebulização e desnebulização, base de dados e de regras e a utilização da lógica de decisão (Lee, 1990), conforme ilustrado na figura 4.2.

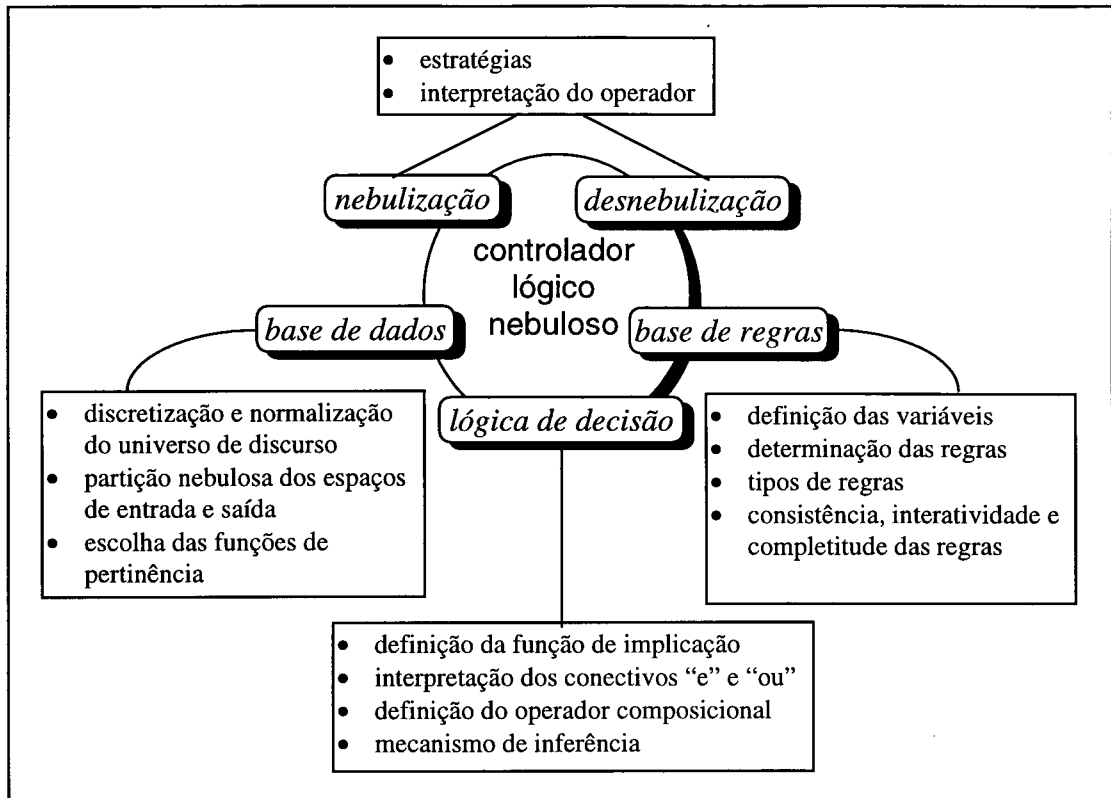


Figura 4.2 - Parâmetros de projeto de um controlador lógico nebuloso.

O projeto de CLNs é uma tarefa árdua, devido a quantidade de parâmetros e a conseqüente complexidade envolvida na determinação de parâmetros adequados a um projeto eficiente e robusto.

A literatura menciona ferramentas e procedimentos no auxílio ao projetista, contudo é comum dizer-se que, no projeto de CLN, o conhecimento de operadores humanos experientes pode ser aproveitado (Fathi-Torbaghan and Hildebrand, 1994; Hirota and Sugeno, 1995; Yager and Filev, 1994). Deve-se notar, no entanto, que este é apenas o ponto de partida. ✓

O ajuste das funções de pertinência, dos conjuntos nebulosos envolvidos, a escolha dos operadores e demais parâmetros de análise e projeto do sistema de controle costumam levar a um longo trabalho de tentativa e erro. A transferência das estratégias de decisão de especialistas e operadores humanos para estes sistemas não é tarefa fácil, mesmo com a flexibilidade apresentada por regras nebulosas.

Esta dificuldade, aliada ao fato de serem recentes os resultados teóricos quanto à estabilidade e robustez destes controladores, faz com que alguns especialistas tenham restrições aos sistemas nebulosos de controle, apesar dos bons resultados obtidos (Romariz, 1996).

4.3.2 Controlador nebuloso com configuração PD ou PI

Os CLNs são projetados, usualmente, para utilizarem as variáveis de controle dada pelo *erro* (e), gerado a partir da diferença entre o sinal de referência e a saída do processo, e a *variação do erro* (Δe), decorrente de

$$\Delta e(k) = e(k) - e(k-1) \quad (4.4)$$

A variável de saída é o sinal de controle ou variação do sinal de controle (forma incremental). Os universos considerados para as variáveis nebulosas são discretos, finitos e normalizados. Assim, nota-se a necessidade da utilização de fatores de escala para fazer corresponder os valores normalizados aos valores reais medidos de erro e variação do erro (Tanscheit, 1992).

Diversos tipos de estruturas de CLNs tem sido avaliadas. O controle PD que gera entrada de controle (u) a partir do erro e da variação do erro, é denominado de *controle tipo posicional*. Outra estratégia de controle é o PI ocasionador de uma entrada de controle incremental (Δe) do erro e da variação do erro, e sendo denominado de *controle tipo velocidade*.

O terceiro tipo é o controle PID responsável pela geração da entrada de controle incremental do erro, variação do erro e erro de aceleração ($\Delta^2 e$). O CLN tipo PI é mais utilizado em aplicações industriais que o CLN tipo PD porque é uma tarefa mais difícil projetar controle PD para atuar de maneira a remover o erro em regime do processo.

O controle tipo PI apresenta, como limitação, o desempenho pouco promissor em resposta transiente para processos de alta ordem, devido a operação de integração interna (Åström and Hägglund, 1996; Li and Ng, 1995; Linkens and Nyongesa, 1995).

Assim, para o tratamento destas dificuldades o CLN tipo PID é uma alternativa indicada, contudo este impõe ao projetista dificuldades inerentes a um projeto de engenharia complexa (Li and Gatland, 1995).

4.3.3 Controlador nebuloso com configuração PID

A implementação usual para o CLN tem configuração PD ou PI, porém a importância de incluir a parte integral do erro é proporcionar maior robustez e melhorar a resposta em regime

permanente do CLN do tipo PD. O projeto em controle PID nebuloso necessita de três entradas, o que pode expandir a base de regras de acordo com os requisitos do projeto e necessidade de desempenho do controlador (projeto de difícil estruturação) (Kwok *et al.*, 1990).

A utilização de algumas aproximações no erro de aceleração reduz as dificuldades, deste modo diversas propostas em controle CLN tipo PID tem sido tratadas na literatura para configuração alternativa e simplificação no projeto de CLN completo em três dimensões, com estrutura PID. A configuração em controle PID nebuloso, adotadas, são reajustes das equações que regem a forma padrão do PID convencional, para adequarem-se a implementações com estruturas alternativas (Coelho e Coelho, 1996f; Kwok *et al.*, 1990; Li and Ng, 1995).

Neste trabalho, opta-se pela determinação, a priori, da base de regras, utilização de cinco termos lingüísticos (NG- Negativo Grande, NP - Negativo Pequeno, ZE - Zero, PP - Positivo Pequeno, PG - Positivo Grande) para as funções de pertinência e o algoritmo de otimização das funções de pertinência é efetuado via CE, cujo esquema de otimização é tratado no capítulo 7. O primeiro projeto adotado é a técnica da implementação direta, onde a ação de controle $u_{PID}(k)$ é originada de um controlador PID convencional e reescrita como:

$$u_{PID}(k) = u_{PD}(k) + \phi \cdot \Delta t \cdot u_I(k) \quad (4.5)$$

$$u_{PD}(k) = K_p e(k) + K_D \Delta e(k) \quad (4.6)$$

$$u_I(k) = u_I(k-1) + u_P(k) \quad (4.7)$$

$$u_P(k) = K_p e(k) \quad (4.8)$$

onde “u” é o sinal de controle e ϕ atua como um parâmetro de seleção estrutural, que aciona ou não a ação integral, de acordo com a natureza da planta (Li and Ng, 1995). A configuração e fluxo dos dados na metodologia com ação PD nebulosa mais a ação I nebulosa (CLN-PD+I) é apresentada na figura 4.3.

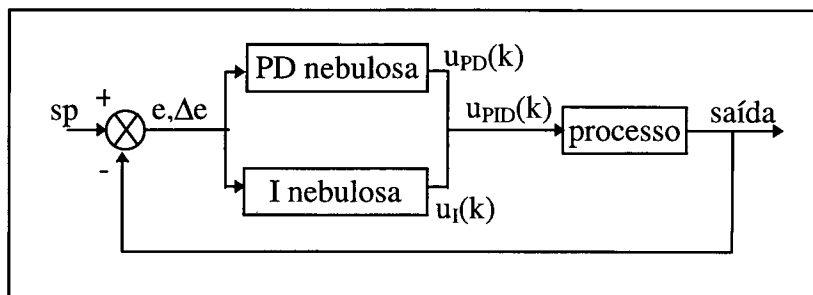


Figura 4.3 - Configuração de CLN tipo (PD+I).

O segundo projeto constitui-se de uma simplificação do primeiro projeto com uma redução da base de regras, já que a parte integral provém da base de regras do controle PD, dado pela seguinte equação:

$$u_{PID}(k) = u_{PD}(k) + \varphi \cdot \Delta t \cdot \sum_{i=0}^k u_{PD}(i) \quad (4.9)$$

Similarmente, ao CLN tipo PD+I quando $\varphi=0$, este CLN tipo PID pode também ser chaveado para CLN tipo PD, pela atribuição do parâmetro $\varphi=0$.

4.3.4 Implementação de sistema híbrido de projeto com computação evolucionária

O projeto de CLN é equivalente a um problema de otimização multimodal e multidimensional e, apresenta dificuldades na especificação de um adequado conjunto de funções de pertinência, o que é tradicionalmente realizado por um procedimento heurístico decorrente do estudo da dinâmica em malha fechada do processo, para atender os requisitos de qualidade e precisão do projeto do CLN.

A LN e a CE têm diferentes vantagens. A LN propicia a representação do conhecimento de diferentes formas em ambientes imprecisos, permitindo modelar as interações e relações entre variáveis do sistema e tornar métodos de inferência mais robustos e flexíveis. A CE, no caso AGs e EEs, aliados a SA, envolvem características de aprendizado e busca global e local.

O projeto e sintonia de CLNs são confiados, a escolha e conhecimento heurístico do projetista, o que abre a perspectiva da utilização de técnicas de CE, o que tem sido proposto na literatura, principalmente para processos monovariáveis, porém o projeto pode ser estendido para

processos multivariáveis. Os enfoques mais freqüentes são para otimização de fatores de escala, funções de pertinência das variáveis nebulosas e base de regras lingüísticas.

A vantagem da utilização de CE é que facilita o trabalho do especialista em controle para configuração, estruturação e sintonização do CLN, pois é possível modelar uma função arbitrária automaticamente sem o uso de especialistas. Contudo a relevante desvantagem da CE é que devido a complexidade computacional dos seus algoritmos, inviabiliza aplicação destas técnicas em otimização “on-line”.

Existem circunstâncias onde a CE, mais especificamente AG incremental (um indivíduo avaliado por geração), pode ser aplicado “on-line” e em tempo real, como mencionado por Linkens and Nyongesa (1995), que incluem:

- i) o modelo do processo é avaliado e o período de tempo é longo o suficiente para permitir o processamento e convergência do AG;
- ii) o AG pode ser usado, com pequeno ou sem conhecimento sobre o processo, para aprender a controlar o processo em tempo real, e
- iii) o AG é usado para adaptação “on-line” para um “bem projetado” controlador, porém para aplicação das otimizações propostas existe a necessidade de processamento rápido da informação, pois a convergência dos AGs é lenta, o que exige a aplicação de sistemas paralelos rápidos.

Como desvantagens deve-se mencionar o aumento da complexidade de implementação do algoritmo, a não correspondência dos conjuntos originais determinado pelo especialista, o que implica também na diferente caracterização dos termos lingüísticos antes e depois da otimização por CE (Fathi-Torbaghan and Hildebrand, 1994; Li and Ng, 1995).

4.4 Conclusão

Neste capítulo apresentou-se uma breve introdução dos conjuntos nebulosos e da lógica nebulosa, onde discutiu-se os conceitos, operações e propriedades inerentes a lógica nebulosa. Aplicações da LN e CLN no meio industrial também foram tratados ressaltando a relevância destas metodologias no contexto industrial e vida cotidiana.

As diferentes estruturas, combinando controladores lógico nebulosos do tipo PD, PI e PID, foram descritas ressaltando-se os parâmetros de projeto, vantagens e desvantagens. A implementação de um sistema híbrido de projeto com otimização, por meio de paradigmas da CE, também foi tratado, que apesar da complexidade computacional demandada constitui-se em alternativa para projeto heurístico e por tentativa e erro pelo projetista do sistema de controle.

Capítulo 5

Modelagem e controle via redes neurais artificiais

5.1 Introdução

As redes neurais artificiais, ou simplesmente redes neurais (RNs), constituem-se de paradigmas da computação, que são estruturados para emular em escala bastante simplificada, o processo biológico de aprendizado humano, com a intenção de simular a forma como o cérebro aprende, recorda e processa informações.

As RNs são um sistema composto de elementos de processamento simples, operando em paralelo, cuja função é determinada pela estrutura da rede, intensidade das conexões, e o processamento executado à computação de elementos ou nós (neurônios).

Os computadores convencionais (arquitetura de John von-Neumann) e o cérebro humano resolvem tarefas de um modo totalmente diferente. Os computadores superam o cérebro humano, quando realizam cálculos matemáticos precisos, contudo o cérebro humano tem a habilidade de reconhecer padrões (visual, tátil ou acústico) e fazer inferências de modo a associá-los, o que excede em muito a capacidade dos computadores.

É desejável então encontrar um modo de como estas vantagens podem ser simuladas em computadores o que permitiria ao computador resolver tarefas que inicialmente tinha desempenho inadequado ou deficiente.

A idéia das RNs é a utilização de computadores digitais em implementações motivadas pela simulação de habilidades de aprendizado, processamento e generalização do cérebro humano, mesmo que rudimentarmente, pois o cérebro tem um sistema neural estimado da ordem de 10 bilhões de neurônios biológicos no córtex humano, conectados num sistema altamente complexo de cerca de 60 trilhões de sinapses ou conexões, que é até hoje de conhecimento limitado (Haykin, 1994).

As RNs são projetadas como um sistema interconectado de elementos de processamento (neurônios), cada um dos quais com um número limitado de entradas e saídas. Entre as características que melhor explanam o potencial das RNs sobressaem-se a adaptabilidade, paralelismo inerente, tratamento de não-linearidades, multidisciplinaridade e tolerância a falhas.

Como não são baseadas em modelo algumas topologias de RNs são aproximadores

universais de qualquer função com uma dada precisão arbitrária dentro de um intervalo, o que proporciona uma promissora ferramenta para identificação, controle inteligente e adaptativo.

As RNs podem “aprender” através de dados de treinamento e, são na realidade uma forma de aproximar funções matemáticas. Atualmente, o termo “neurocontrol” (controle neural) é utilizado na literatura de sistemas de controle e, refere-se a métodos de controle que empregam RNs (RayChaudhuri *et al.*, 1996).

As RNs ou métodos “conexionistas” têm emergido como ferramentas muito poderosas para projeto de sistemas inteligentes, principalmente em identificação (Narendra and Parthasarathy, 1990; Sjöberg *et al.*, 1994; Sjöberg, 1995) e controle de processos industriais complexos e não-lineares (Fukuda and Shibata, 1992; Narendra, 1996).

As relevantes aplicações das RNs no meio industrial tem sido alvo de pesquisa, entre elas, controle de fornos de arco (Neural Applications), aplicações automotivas (Ford, General Motors), diagnóstico de motores elétricos (Siemens), área militar e aeroespacial (U.S. Naval Air Warfare Center, NASA), controle de processos químicos (Pavilion Tech., Estman Kodak) e produção de aço (Fujitsu, Nippon Stell) (Coelho e Coelho, 1996g; Widrow *et al.*, 1994).

5.2 Topologias

As RNs podem ser classificadas em RNs “feedforward” e recorrentes. Na literatura, diversas estruturas de RNs tem sido mencionadas, para aplicação na área de identificação e controle de processos, as RNs de Hopfield, de função base (radial, “wavelet” e polinomial) e híbridas com técnicas de controle adaptativo têm polarizado a maioria das aplicações.

Apesar das vantagens das RNs, sua implementação apresenta limitações por não poder justificar suas respostas (“black-box”), alto custo computacional, baixa velocidade de aprendizado para aplicações “on-line” e, heurística para determinação da arquitetura e parâmetros ligados ao aprendizado.

5.3 Rede neural “feedforward” com treinamento “backpropagation”

O “multilayer perceptron” (MLP) representa uma generalização da RN “perceptron” de camada única. O MLP tem sido aplicado com sucesso para resolver problemas de mapeamento, por meio de aprendizado supervisionado, utilizando o algoritmo da retropropagação do erro ou “error Back-Propagation” (BP).

O algoritmo gradiente BP desenvolvido por P.J. Werbos, em 1974, foi popularizado por pesquisadores como D.E. Rumelhart, J.L. McClelland, D.B. Parker e Y. le Cun, no meio da década de 80. O BP consiste basicamente de duas fases através das diferentes camadas da RN: fase “forward” e fase “backward”, sintetizados pelas seguintes etapas:

- i) inicializar aleatoriamente os pesos de todas as conexões da RN (por exemplo, entre -1 e 1);
- ii) aplicar o conjunto de treinamento (dados do processo) à RN;
- iii) propagar as saídas dos neurônios de cada camada da RN para os neurônios da próxima camada passando pela função de ativação dos neurônios (passo “forward”);
- iv) especificar a saída desejada e calcular todos os erros de todas as camadas;
- v) corrigir os pesos pelos “gradientes locais” da RN (passo “backward”);
- vi) repetir os passos (iv) e (v) enquanto a função do erro quadrático da saída não encontrar o valor de tolerância do erro aceitável imposta pelo usuário (critério de parada do treinamento);
- vii) utilizar “recall” e conjunto(s) de teste para analisar o desempenho do treinamento da RN.

Após a conclusão do treinamento, a RN atua como uma RN “feedforward”, pois os pesos das conexões permanecem constantes, exceto no caso da necessidade de um novo treinamento da RN com um novo conjunto de dados. A figura 5.1 ilustra um exemplo de uma RN multicamadas “feedforward” com treinamento BP.

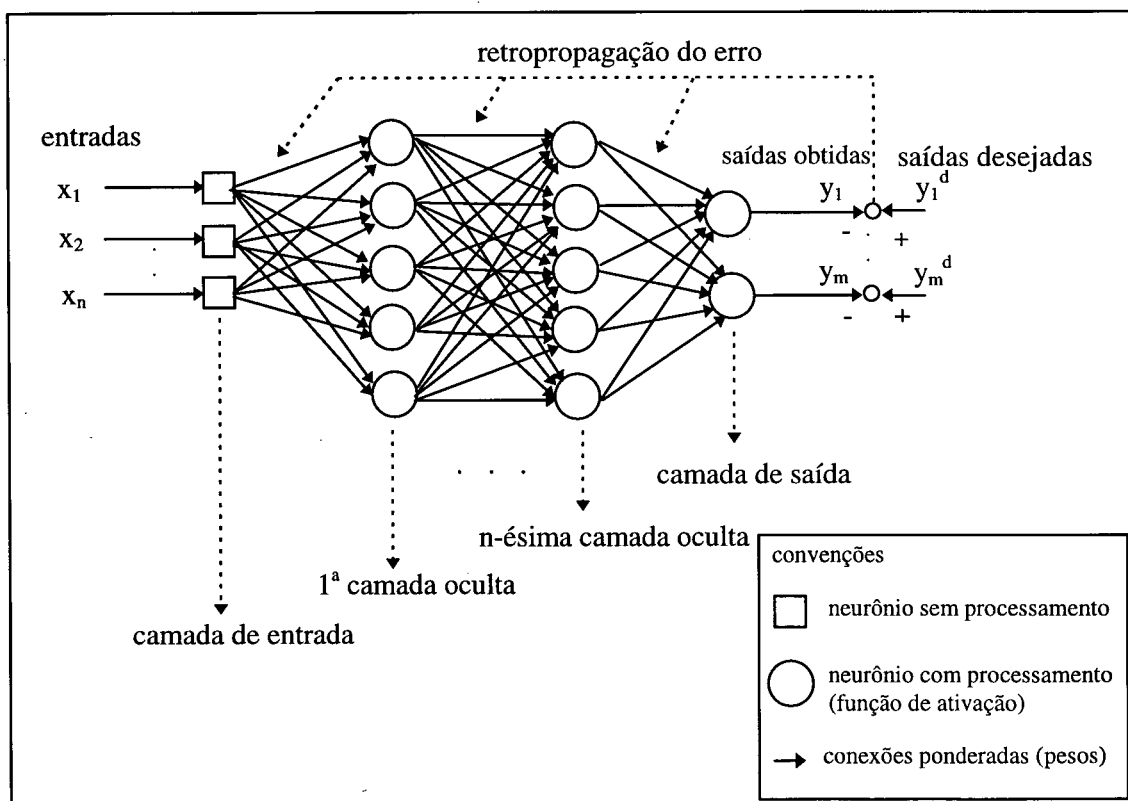


Figura 5.1- Rede neural multicamadas “feedforward” com treinamento BP.

As equações que regem o algoritmo de treinamento BP, através da regra delta generalizada, são apresentadas a seguir (Fukuda and Shibata, 1992; Haykin, 1994).

$$E_p = \frac{1}{2} \sum_{i=1}^n (t_{p,j} - o_{p,j})^2 \quad (5.1)$$

onde $t_{p,j}$ é a entrada desejada para o j -ésimo componente do padrão de saída para o padrão “p”, $o_{p,j}$ é o j -ésimo elemento do padrão de saída atual, produzido pela apresentação do padrão de entrada “p”, “n” é o número de padrões contidos no conjunto de treinamento apresentado à RN.

A equação (5.1) avalia o erro dos padrões “p” de entrada e saída, onde E_p é a função custo para o padrão p. O erro total de treinamento, E, é calculado por

$$E = \sum_{i=1}^n E_p \quad (5.2)$$

A definição da soma ponderada da saída da camada anterior é obtido por

$$s_{p,j} = \sum_{i=1}^n w_{j,i} o_{p,i} \quad (5.3)$$

onde $w_{j,i}$ é o peso da conexão da j -ésima para a i -ésima camada da RN. A saída,

$$o_{p,j} = f_j(s_{p,j}) \quad (5.4)$$

utiliza-se da função tangente hiperbólica como função de ativação dos neurônios da(s) camada(s) oculta(s) e da camada de saída da RN,

$$f(x) = \tanh(x) = \frac{1 - e^{-ax}}{1 + e^{-ax}} \quad (-1 < f(x) < 1) \quad (5.5)$$

Após diversas manipulações matemáticas, que não são apresentadas obtém-se a equação da generalização da regra delta, definida por

$$\Delta w_{j,i} = -\eta \frac{\partial E_p}{\partial w_{j,i}} \quad (5.6)$$

onde $\Delta w_{j,i}$ é a variação a ser realizada no peso (w) da conexão da i -ésima para a j -ésima camada da RN e η é a constante que determina o coeficiente de aprendizado da RN. As limitações do treinamento BP estão relacionadas a baixa velocidade de aprendizado, convergência para mínimos locais e a heurística para determinação da arquitetura e coeficiente de aprendizado.

A literatura menciona diversas heurísticas para o tratamento das limitações do BP, tais como:

- uso de funções de ativação eficientes para os neurônios;
- emprego de termo de momento;
- adição de ruído nos pesos da RN;
- utilização de métodos numéricos eficientes (gradiente conjugado, MQR, filtro de Kalman, quasi-Newton e outros);
- estatística na análise e determinação do conjunto de treinamento da RN; e
- aplicação de CE (ajuste da inclinação das funções de ativação, determinação dos pesos iniciais da RN, otimização da topologia e conexões, evolução do coeficiente de aprendizado) (Belew *et al.*, 1990; Coelho e Coelho, 1995; Haykin, 1994).

Neste trabalho, o aspecto do aumento da velocidade do treinamento BP é realizado através do *algoritmo adaptativo heurístico* (AAH) para a variação dinâmica do coeficiente de aprendizado e momento (Anguita *et al.*, 1993). O AAH tem a regra de atualização dos pesos da RN regida pelas equações (5.7) a (5.9), mencionadas a seguir. A variação do valor dos pesos $\Delta w_{i,j}$ é dada por

$$\Delta w_{i,j}(t) = -\eta(t) \sum_{p=1}^N \frac{\partial E_p}{\partial w_{i,j}} + \alpha(t) \Delta w_{i,j}(t-1) \quad (5.7)$$

onde o pseudocódigo do AAH para o treinamento da RN é explicitado na figura 5.2.

i) inicializar $\eta(0) = \eta_0$ e $\alpha(0) = \alpha_0$;
 ii) se $E_{total} < E_{total}(t-1)$ então
 $\eta(t) = \phi \cdot \eta(t-1)$ e $\alpha(t) = \alpha_0$;
 se $E_{total} < (1+\epsilon)E_{total}(t-1)$ então
 $\eta(t) = \phi \cdot \eta(t-1)$ e $\alpha(t) = \alpha_0$;
 se $E_{total} > E_{total}(t-1)$ então
 descarta-se o último passo de aprendizado e
 $\eta(t) = \beta \cdot \eta(t-1)$ e $\alpha(t) = 0$;

Figura 5.2 - Pseudocódigo do algoritmo adaptativo heurístico para treinamento de RN.

Os fatores de aceleração ϕ e desaceleração β são regidos por

$$\phi(t) = 1 + \frac{K_a}{K_a + \eta(t-1)} \quad (5.8)$$

$$\beta(t) = 1 + \frac{K_d}{K_d + \eta(t-1)} \quad (5.9)$$

onde w_{ij} são os pesos das conexões da RN, t é o contador do número de ciclos de aprendizado, ϵ é um fator de ajuste de “sensibilidade”, η é o coeficiente de aprendizado, α é o fator de momento, E_{total} é o erro total do mapeamento efetuado pela RN, K_a é a constante de aceleração e K_d é a constante de desaceleração (Anguita *et al.*, 1993).

5.4 Redes neurais recorrentes

Nas redes “feedforward”, os elementos de processamento das redes “feedforward” são conectados de tal maneira que todos os sinais propagam-se em apenas uma direção das unidades de entrada da rede para as unidades de saída. As redes neurais recorrentes (RNRE) diferem das redes “feedforward”, pois nas RNRE existem conexões “feedforward” e conexões “feedback” ao longo dos sinais que podem ter propagação em direções opostas. Exemplos de RNRE são as RNs de Hopfield e “Bidirectional Associative Memory” (BAM).

As redes “feedforward” não tem memória dinâmica, e o método “tapped-delay-line” é usualmente adotado visando a RN em representar sistemas dinâmicos. Contudo, este método apresenta desvantagens, quanto a velocidade de aprendizado, devido ao largo número de unidades na camada de entrada.

Esta desvantagem é ocasionada, se a ordem do sistema a ser identificado é desconhecida, isto tem de ser sobrestimado e por conseqüência um elevado número de unidades na entrada no “buffer” deve ser dado para tratar sistemas de ordem elevada. Isto usualmente leva a uma RN com estrutura de grandes dimensões e baixa velocidade de aprendizado.

Um grande número de unidades, na camada de entrada, também faz o identificador altamente susceptível a ruídos externos. Uma outra desvantagem é que a estrutura de treinamento é diferente da estrutura do conjunto “recall” e freqüentemente as RNs são treinadas de maneira

adequada com a conjunto de dados de treinamento, mas têm performance deficiente com conjuntos de “recall” (padrões de teste, objetivando observar quão genérica é a RN). Por esta razão, não é uma atividade fácil obter-se simuladores independentes de um sistema dinâmico (Pham and Liu, 1992).

Devido a sua estrutura, as RNRE não sofrem das desvantagens da redes “feedforward”. As RNRE podem ser classificadas como totalmente ou parcialmente recorrentes. As RNRE totalmente recorrentes possuem conexões “feedforward” e “feedback”, isto é, possuem todo tipo de conexão recorrente e todas ajustáveis.

Em redes parcialmente recorrentes, a estrutura principal da RN é não-recorrente. As conexões “feedforward” são treináveis. As conexões realimentadas são formadas através de um conjunto de *unidades de contexto* não treináveis.

As unidades de contexto memorizam alguns estados passados das unidades ocultas, e somente as saídas da RN dependem de um agregado de estados anteriores e da entrada atual da RN. Isto é o porque destas redes parcialmente recorrentes possuírem as características de uma memória dinâmica (Haykin, 1994; Zuben e Netto, 1994).

Entre as RNRE, a rede neural de Elman (Elman, 1990) é um dos tipos mais simples que podem ser utilizados com o algoritmo BP. A rede é originalmente projetada para aplicações de processamento de linguagem, contudo pode ser aplicada na identificação de sistemas dinâmicos, cujo procedimento é descrito a seguir.

5.4.1 Rede neural parcialmente recorrente de Elman

O diagrama de blocos de uma rede neural parcialmente recorrente de Elman (RNPREE) é mostrado na figura 5.3. Nota-se que em adição aos neurônios ou elementos de processamento (EPs) da camada de entrada, camada oculta e camada de saída, a RNPREE tem também unidades de contexto, como é o caso de redes parcialmente recorrentes em geral.

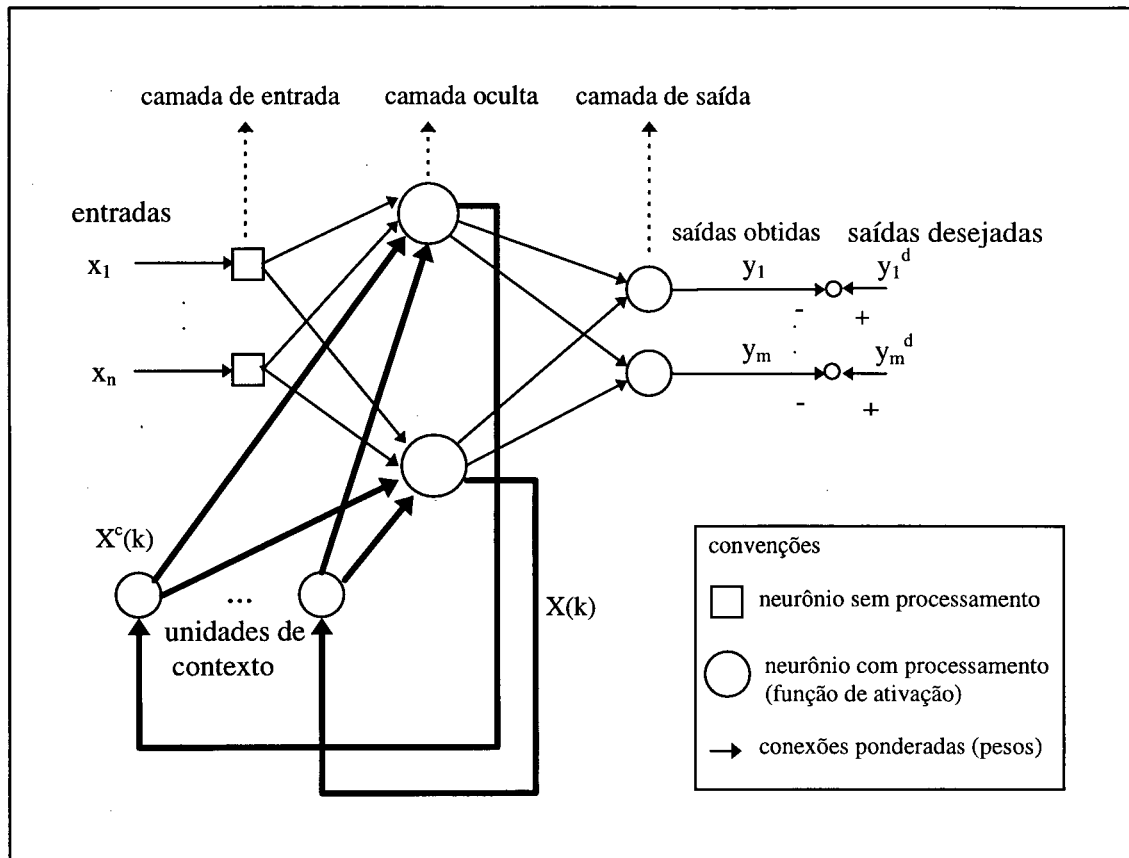


Figura 5.3 - Rede neural parcialmente recorrente de Elman.

As unidades de entrada e saída interagem com o ambiente externo, enquanto que as unidades oculta e unidade de contexto não interagem. Os EPs da camada de entrada são somente unidades de “buffer” (armazenamento temporário dos dados) que passam o sinal sem processá-los. Os EPs da camada de saída são lineares, que servem para a soma dos sinais passados para eles. Enquanto os EPs da camada oculta podem ter funções de ativação lineares ou não-lineares.

As unidades de contexto são usadas somente para memorizar as prévias ativações dos EPs da camada oculta e podem ser considerados para a função de armazenar atrasos de tempo de um passo. As conexões “feedforward” são modificáveis e as conexões recorrentes são fixas. Devido ao fato que as conexões recorrentes serem fixas, as RNPREE são somente parcialmente recorrentes (Elman, 1990).

Para um específico tempo “k”, as ativações prévias dos EPs da camada oculta (tempo “k-1”) e entradas correntes (no tempo “k”) são utilizadas como entradas para a RNPREE. Nesta fase a RNPREE é uma RN “feedforward”. A regra de aprendizado BP é utilizada para o treinamento da RN.

Após este passo de treinamento, as ativações dos EPs, da camada oculta no tempo “k”, são enviadas para trás (“back”), através das interconexões, para as unidades de contexto e salvas para o próximo passo de aprendizado (tempo “k+1”). As funções de ativação dos EPs da camada oculta podem ser lineares ou não-lineares, contudo neste trabalho utiliza-se a tangente hiperbólica, conforme equação (5.5).

5.4.1.1 Rede neural de Elman modificada

As RNPREE com EPs constituídos de funções de ativação lineares têm sido utilizadas para identificar sistemas lineares. Através de simulações computacionais, nota-se que a RNPREE pode identificar sistemas lineares de primeira ordem facilmente, mas apresenta dificuldades para identificar sistemas lineares de ordem mais elevada que 1ª. ordem.

Coefficientes de aprendizado elevado causam oscilações e igualmente instabilidades no processo de treinamento. Para obter-se desempenho mais adequado introduz-se “auto-conexões” nas unidades de contexto da RNPREE, para dar a RNPREE unidades de inércia (“units inertias”), e deste modo melhorar a capacidade de memorização da RNPREE tanto para utilização de funções de ativação lineares quanto não-lineares, obtendo-se assim uma RNPREE modificada (RNPREMOD) (Pham and Liu, 1992).

A modificação é efetuada na RNPREE básica, exposta na figura 5.2. A saída da *j*-ésima unidade de contexto da RNPREE é dada por:

$$x_k^c(k) = \alpha x_j^c(k-1) + x_j(k-1) \quad (5.10)$$

onde $x_j^c(k)$ e $x_j(k)$ são as saídas da *j*-ésima unidade de contexto e *j*-ésimo EP da camada oculta e α é o ganho da realimentação das auto-conexões. O valor de α adotado é o mesmo para todas auto-conexões e não é modificado no algoritmo de treinamento. Enquanto na RNPREMOD a equação (5.10) é transformada em

$$x_j^c = x_j(k-1) + \alpha x_j(k-2) + \alpha^2 x_j(k-3) + \dots + \alpha^n x_j(k-n-1) \quad (5.11)$$

A equação 5.11 mostra que a saída das unidades de contexto é uma integração da saída da camada oculta que é conectada a ela. O valor de α é entre $[0;1]$, sendo que o valor mais próximo de 1 habilita a unidade de contexto para agregar mais saídas passadas.

Desde que a ordem do sistema dinâmico é relatada ao número das saídas passadas que a saída presente depende. Isto leva a possibilidade da introdução de auto-alimentações nas unidades de contexto visando-se aumentar a capacidade da RNPREE em modelar sistemas dinâmicos de ordem elevada. Um diagrama esquemático da RNPREMOD é apresentado na figura 5.4.

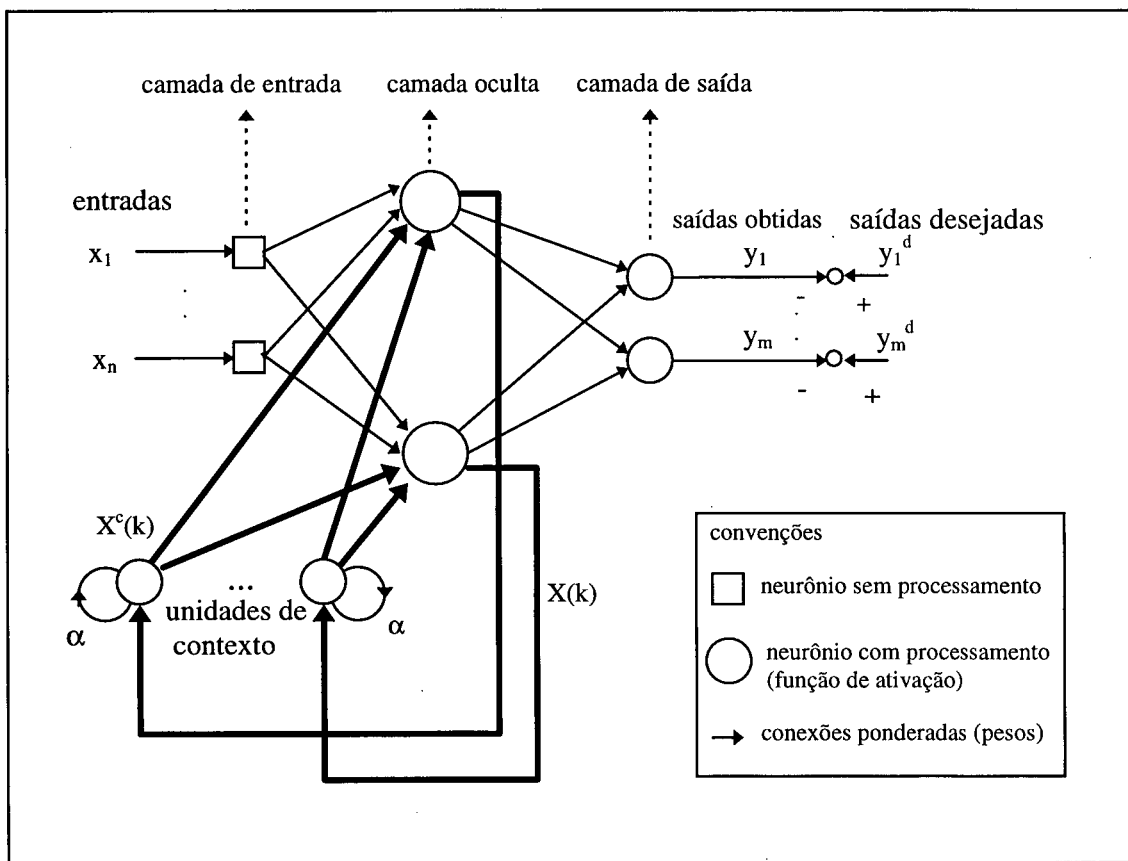


Figura 5.4 - Rede neural parcialmente recorrente de Elman modificada.

5.5 Identificação de processos via redes neurais

Apesar de diversas aplicações, em identificação de sistemas lineares, reportadas na literatura (utilização de funções de ativação lineares nos EPs). Em identificação é importante o papel das RNs em modelos “black-box” (identificação estrutural) de sistemas dinâmicos não-

lineares, conforme tratado por Sjöberg (1995) e Narendra and Parthasarathy (1990). As RNs são aplicadas em diferentes modelos, tais como:

- RN-NFIR (“Non-linear Finite Impulse Response model”): usa somente entradas medidas passadas $u(t-k)$ como regressores;
- RN-NARX (“Non-linear AutoRegressive model structure eXogenous inputs”): usa $u(t-k)$ e saídas medidas passadas $y(t-k)$ como regressores, também denominado de modelo série-paralelo;
- RN-NOE (“Non-linear Output Error model”): usa $u(t-k)$ e saídas preditas passadas $\hat{y}_u(t - k|\theta)$, também denominado de modelo paralelo;
- RN-NARMAX (“Non-linear AutoRegressive Moving Average model structure with eXogenous inputs”): usa $u(t-k)$, $y(t-k)$ e $\hat{y}_u(t - k|\theta)$;
- RN-BJ (“Non-linear Box-Jenkins model structure”): usa todos os quatro tipos de regressores mencionados.
- modelos não-lineares de espaço de estados: utiliza os componentes passados das saídas “virtualmente”.

No presente trabalho, utiliza-se a RN “feedforward” e RNPREMOM com treinamento BP, descrita na seção 5.3 e 5.4, anteriormente, para modelagem de processos, onde as entradas são compostas por parâmetros do polinômio $A(z^{-1})$ e $B(z^{-1})$, enquanto a saída é dada por $y(k)$, conforme é apresentado no capítulo 6 corresponde as aplicações de algoritmos de IC em identificação de processos.

5.6 Controle neural de processos

A área denominada “*neurocontrol*” é um subconjunto de ambas as áreas de pesquisa tanto RNs quanto a teoria de controle. Os trabalhos existentes podem ser reduzidos em algumas estratégias fundamentais de projeto, apesar de existirem diversas variantes na literatura, tais como controle “self-learning”. As estratégias são (Haykin, 1994; Levine, 1996; Narendra, 1996):

- *controle supervisionado*: estratégia de construir uma RN que imite um sistema de controle pré-existente, de modo análogo a sistemas especialistas;

- *controle inverso direto*: as RNs que podem seguir a trajetória especificada pelo usuário (utilizando um modelo de sistema inverso) ou sistema de nível hierárquico superior;
- *controle adaptativo neural (direto ou indireto)*: faz o que o controle adaptativo convencional faz, usa RNs para aumento da robustez do sistema e tratamento de não-linearidades;
- “*backpropagating utility*” e “*adaptive critics*”: maximizar a performance ou minimizar uma função custo no tempo. Ex: “reinforcement learning”.

5.6.1 Projeto de controle PID neural

Diversas são as propostas e variantes de controle PID combinado a RNs na literatura (Omatu and Akhyar, 1993; Pereira *et al.*, 1994). O algoritmo proposto para configuração do controle PID neural (PID-RN) é um esquema de aprendizado especializado (Psaltis *et al.*, 1988; Saerens and Soquet, 1991; Zhao, 1994), baseado em uma aproximação do Jacobiano do processo a ser controlado.

Uma simples aproximação pela avaliação do erro, entre a saída desejada e a saída obtida, deve ser retropropagada permitindo que a RN aprenda a mapear e controlar o processo através de um esquema de aprendizado especializado, conforme ilustração na figura 3.9. A estrutura de treinamento do controlador PID com configuração neural é apresentada na figura 5.5.

O objetivo do controle PID-RN, neste trabalho, é adotar uma forma de aprendizado especializado em método de controle direto (os parâmetros do controlador são diretamente ajustados para reduzir o erro da saída) proposto por Saerens and Soquet (1991) e aplicado aqui em um controle PID-RN, baseado em “conhecimento qualitativo” da planta.

Em poucas palavras, o conhecimento básico qualitativo, a priori, do processo a ser controlado constitui-se da direção que o parâmetro de controle $u(k)$ deve ser modificado para produzir um incremento na saída $y(k+1)$ da planta, pois desta forma possibilita-se o aprendizado do PID-RN via BP, pois determina-se a direção a aproximação do gradiente que levará ao decréscimo do erro da saída do processo (Saerens and Soquet, 1991).

A forma de configuração do controle “u” dado pelo controlador PID-RN obedece a uma forma alternativa às equações que regem o controlador PID convencional, dada por:

$$\Delta u(k) = K_P[e(k)-e(k-1)] + K_I e(k) + K_D[e(k)-2e(k-1)+e(k-2)] \quad (5.12)$$

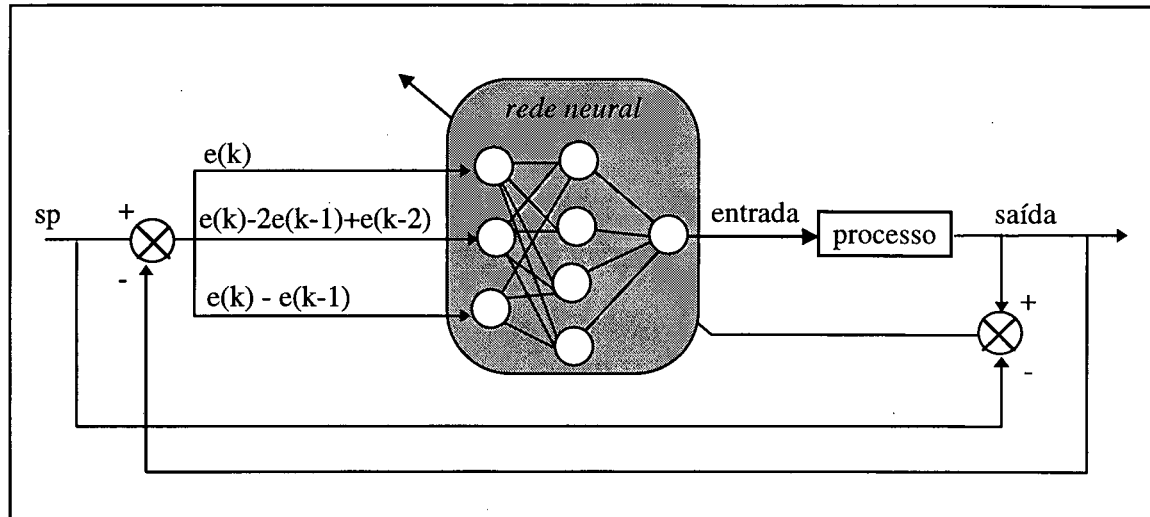


Figura 5.5 - Controlador PID com configuração neural.

A técnica para configuração do PID-RN emprega fatores de escala nas entradas e saída da RN. A arquitetura da RN é multicamadas e atua em funções de ativação dos neurônios, dadas por tangentes hiperbólicas. O treinamento é BP realizado conjuntamente pelo AAH descrito anteriormente neste capítulo.

5.7 Combinação de redes neurais e computação evolucionária

Diversas técnicas de otimização, baseadas em gradiente descendente, tem sido utilizadas para treinamento de RNs multicamadas. Contudo, há problemas quanto a velocidade de convergência e a presença de mínimos locais. Técnicas alternativas como CE podem ser aplicados as RNs de três modos:

- i) otimização das conexões dos pesos;
- ii) otimização da topologia da RN; e
- iii) otimização das regras de aprendizado (coeficiente de aprendizado, termo de momento e função de ativação) (Belew *et al.*, 1990).

A CE é eficiente para busca global, mas é relativamente lenta para sintonia fina local, o que parece natural para integração com BP ou outro método numérico para busca local.

As vantagens mencionadas pela literatura da aplicação de CE interagindo com RNs são: aplicação em problemas simples, como OU-exclusivo e problema dos dois espirais (“benchmarks”); e aplicar um algoritmo híbrido de um método gradiente local em RNs de dimensões elevadas.

As vantagens da aplicação de CE são tratadas no capítulo 3, contudo deve-se mencionar algumas desvantagens da CE, tais como:

- não existe um estudo matemático formal de maneira a garantir que as metodologias de CE garantem a convergência para o ótimo global e tem dificuldade à sintonia fina;
- RNs com dimensões elevadas levam longo tempo para convergir;
- apresentar procedimentos de busca global, usualmente compreendem uma alta complexidade computacional;
- aumento da complexidade do algoritmo da RN.

As aplicações em controle de processos de RNs combinadas a CE são abordadas em problemas de controle, tais como problemas do pêndulo invertido, controle de satélite, robótica, entre outras (Dracopoulos and Jones, 1995; Whitley, 1991).

5.8 Conclusão

Este capítulo apresentou um panorama de relevantes aplicações de RNs em automação industrial, mostrando um grande espectro das possibilidades da sua utilização em identificação e controle de processos industriais.

As topologias de RNs abrangendo RNs “feedforward” e RNs recorrentes são também abordadas. As RNs multicamadas “feedforward” com treinamento BP e parcialmente recorrente de Elman são tratadas e servem de base para os algoritmos utilizados nos capítulos 6 e 7 sobre a aplicações dos paradigmas da IC.

As estruturas reportadas na literatura em identificação (não-linear) e controle de processos são apresentadas. Após o algoritmo de treinamento supervisionado de RNs multicamadas foi apresentado, além de heurísticas para o aumento da velocidade de convergência deste frente a mínimos locais, tais como a variação dinâmica dos coeficientes de aprendizado e momento.

A possibilidade da combinação de RNs e CE que incluem otimização: das conexões dos pesos; da topologia da RN; e das regras de aprendizado, são brevemente tratadas.

Capítulo 6

Aplicações em identificação de processos

6.1 Introdução

Neste capítulo apresenta-se os resultados do desempenho da aplicação de metodologias da IC denominadas computação evolucionária, incluindo-se neste caso algoritmos genéticos, algoritmos genéticos híbridos com “simulated annealing” e estratégias de evolução em identificação de processos.

Outro paradigma de IC que também é abordado são as redes neurais artificiais, através das topologias “feedforward” e parcialmente recorrente de Elman. As técnicas convencionais dos controle mínimos quadrados recursivo e variável instrumental com configuração genética são utilizadas.

A descrição e as principais características dos processos de temperatura, nível mono-tanque e turbo-gerador, utilizados na avaliação dos algoritmos de identificação inteligentes e convencionais são tratadas brevemente.

A seguir, são apresentados os resultados experimentais e comentários que servem de base ao aprimoramento das metodologias aplicadas, visando a obtenção de algoritmos robustos e alternativos a técnicas convencionais em automação industrial.

6.2 Descrição dos processos

Os processos avaliados são os processo de nível e térmico, implementados e em utilização no UFSC/EEL/LCMI (Coelho, 1995). O processo de nível caracteriza-se por apresentar não-linearidades e o processo térmico tratando-se de um processo lento.

O processo simulado não-linear de regulação de nível, composto de dois tanques acoplados tem implementação e análise frente a distúrbios de carga e a mudanças de “setpoint”.

6.2.1 Processo de nível mono-tanque

O processo de nível consiste em um tanque de vidro retangular de dimensões 29 cm x 12 cm x 28 cm e capacidade de 10 litros. Na base do processo existe um reservatório com

capacidade para armazenar 20 litros do fluido. A água é bombeada para o tanque com o auxílio de uma bomba elétrica de corrente contínua (CC).

O nível desejado é medido por um potenciômetro ligado a uma bóia e conectado no topo do tanque. Uma variação na resistência do sensor de nível é convertida em um sinal analógico via ponte de Wheatstone com ajuste de zero.

Um amplificador operacional implementado na configuração ganho, é utilizado para calibrar o nível desejado para um intervalo de 0 até 5 volts (faixa de 5 cm até 18 cm). A figura 6.1 ilustra o diagrama de configuração e os componentes do processo de nível.

A planta é submetida a uma entrada e, através de uma placa de aquisição de dados, conectada ao computador, são coletadas 1500 amostras do processo, para um período de amostragem de 1 s.

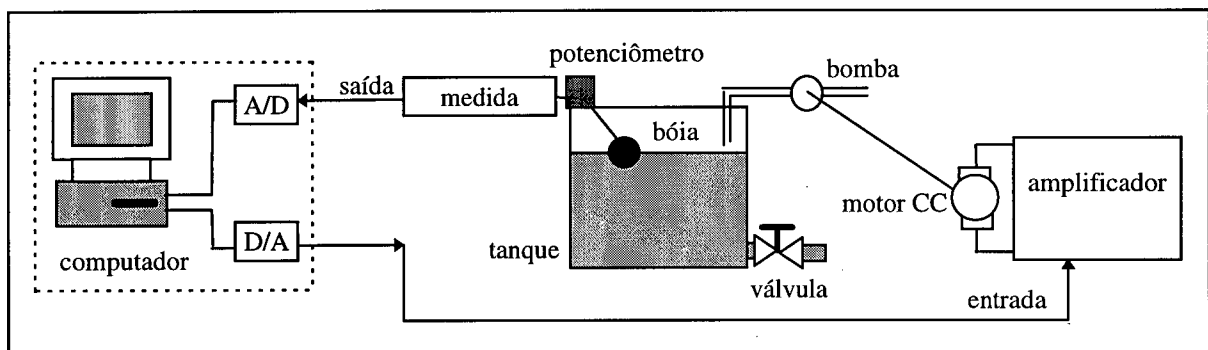


Figura 6.1 - Diagrama de configuração do processo de nível.

6.2.2 Processo térmico

O processo térmico é um tanque retangular metálico de dimensões 15 cm x 13 cm x 23 cm, com capacidade para 4.5 litros. Os diversos componentes do sistema de temperatura são: um sensor de temperatura do tipo PT-100 conectado a um circuito de transmissão para escalonamento (0 a 5 V) e amplificação do sinal do sensor (faixa de 25°C até 90°C) e, uma resistência elétrica industrial de 750 W, conforme é apresentado na figura 6.2. O processo é submetido a uma entrada e são coletadas 300 amostras do processo para um período de amostragem de 5 s.

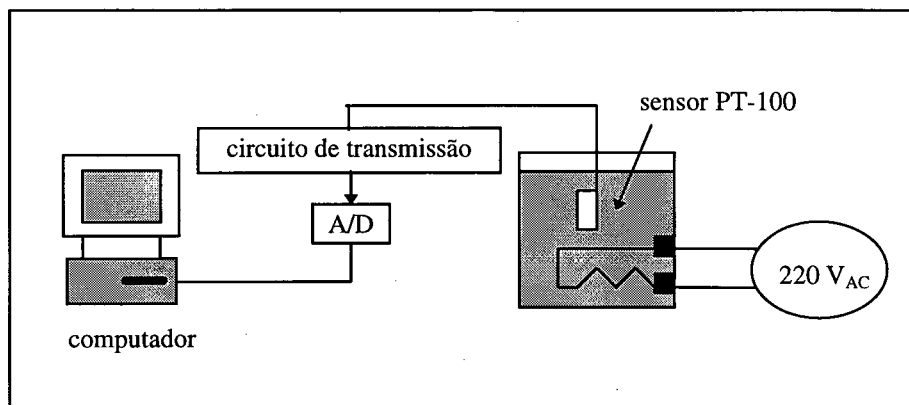


Figura 6.2 - Diagrama de configuração do processo térmico.

6.2.3 Turbo-gerador

A planta do turbo-gerador utilizada nas simulações é representada pela seguinte equação a diferenças não-linear:

$$y(k) = 0.0888 u(k-1) + 0.9909 u(k-2) + 0.3633 y(k-1) - 0.0467 u(k-2)^2 + \quad (6.1)$$

$$0.6193 u(k-1)y(k-1) + 0.0242 u(k-3)^2 - 0.2500 y(k-1)^2 -$$

$$0.8098 u(k-3)y(k-1) + 0.3974 y(k-1) y(k-2)$$

que foram obtidos de um processo real de laboratório para identificação não-linear (Pickhardt and Unbehauen, 1994).

6.3 Identificação de processos

A análise da aplicação de metodologias convencionais e inteligentes, nos processos mencionados é tratada a seguir. Um dos cálculos adotados ao longo dos experimentos é o resíduo, necessário ao traçado de gráficos para visualização adequada dos resultados obtidos. O resíduo (r) da resposta real em relação a resposta estimada é dado por, conforme a equação

$$r = \sum_{k=1}^N [y(k) - \hat{y}(k)] \quad (6.2)$$

6.3.1 Identificação do processo de nível mono-tanque e térmico

A identificação paramétrica dos processos, pelos algoritmos da CE (AGs, AGSA e EEs), MQR e VI, é efetuada obedecendo-se as equações (2.2) a (2.4), descritas no capítulo 2. A identificação do modelo matemático dos processos de nível e térmico por estas técnicas visa a obtenção de um modelo linear (linearização).

Na estimação dos processos de nível mono-tanque e térmico aborda-se os parâmetros do processo de ordem reduzida, tanto para 1^a quanto para 2^a ordem, em outras palavras, $\{n_a \in \mathbb{N} / n_a=(1,2,3)\}$ e $\{n_b \in \mathbb{N} / n_b=(1,2,3)\}$, visando a obtenção de um modelo linearizado (MQR, VI, CE) em resposta a uma entrada degrau.

Nas implementações dos algoritmos para identificação do processo de nível e térmico, o atraso de transporte “d” é dado como conhecido de valor unitário, exceto em alguns experimentos com CE vinculados ao processo térmico. O projeto é realizado observando-se que o critério adotado na minimização do erro é o ISE, contudo é apresentado o erro com o critério RPE e SSE.

As RNs, ao contrário, foram implementadas à identificação estrutural do modelo do processo, onde o conhecimento dos processos está armazenado sob forma de ponderações das conexões da RN. Em outras palavras, a RNs (utilização de funções de ativação não-lineares) armazena o modelo do processo sob a forma de uma estrutura “black-box” pela minimização dada pela regra delta generalizada, aplicada ao algoritmo BP, tanto para a RN “feedforward” quanto para a RNPREMOD descrita no capítulo 5.

6.3.1.1 Mínimos quadrados recursivo e variável instrumental

Diferentes estudos de como selecionar os parâmetros da VI tem sido propostos na literatura (Ljung, 1987; Wellstead and Zarrop, 1991). Quanto aos aspectos relativos a montagem da VI, o melhor valor a ser utilizado na reinicialização da matriz de covariância e o atraso de transporte a ser atribuído a variável instrumental requerem, na implementação do algoritmo VI convencional, um estudo empírico e um conhecimento particular para cada processo onde está sendo aplicada a técnica de estimação VI.

Neste trabalho, apresenta-se resultados preliminares da aplicação de um algoritmo de identificação com um módulo de configuração baseado em AG para ajuste e seleção de parâmetros à inicialização do algoritmo VI (VICAG).

A estrutura conforme apresentada na figura 6.3 é implementada visando avaliar o AG na determinação de um adequado conjunto de parâmetros para inicialização de estimação da VI de modo a torná-lo robusto em aplicações práticas.

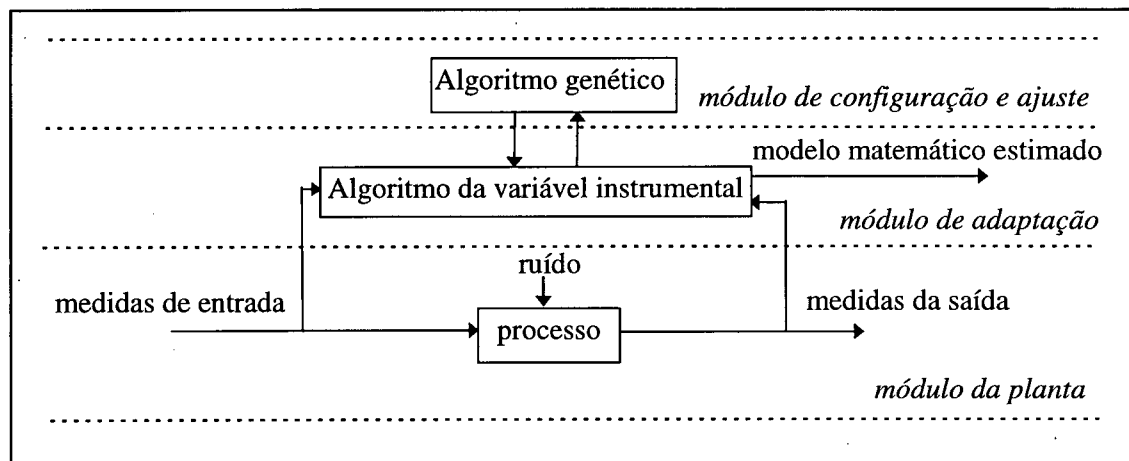


Figura 6.3 - Estrutura hierárquica para aplicação da VI com módulo de configuração genético.

O AG tem o cromossomo definido por três variáveis: o tipo de montagem da variável instrumental $Z(t)$, o atraso de transporte (d_{vi}) a ser atribuído a variável instrumental e a reinicialização da matriz de covariância $P(t)$. O algoritmo VI estima os parâmetros dos processos utilizando um valor de reinicialização da matriz de covariância $P(t)=100I_p$, onde $p=(n_a+n_b)$.

O fator de esquecimento λ utilizado varia exponencialmente com o tempo de 0.95 a 1. O espaço de busca das soluções leva em conta os intervalos de busca, de acordo com a representação do cromossomo da figura 6.4, onde c_1 , c_2 e c_3 representam os intervalos de variação dos fenótipos da montagem da variável instrumental $Z(t)$, atraso de $Z(t)$ e reinicialização da matriz de covariância $P(t)$, respectivamente.

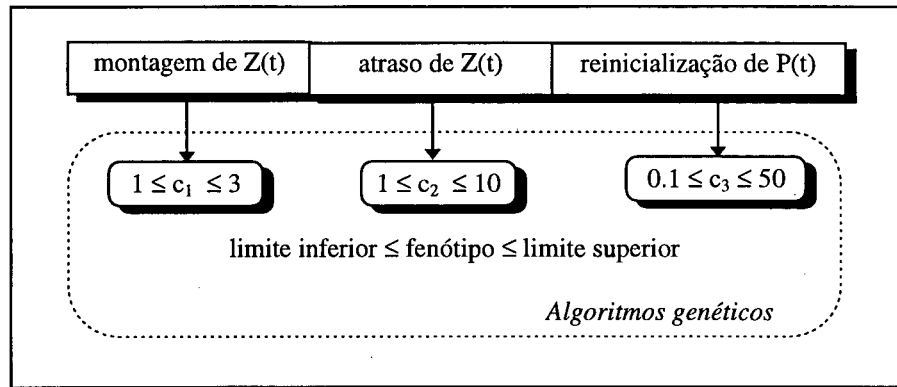


Figura 6.4 - Estrutura dos cromossomos utilizados nos AGs para configuração da VI.

Na montagem da variável instrumental se $c_1=1$, então utiliza-se a equação (6.3). Se $c_1=2$, a variável instrumental é implementada pela equação (6.4), entretanto se $c_1=3$ a variável instrumental é descrita pela equação (6.5), conforme as equações a seguir:

$$Z(t) = [u(t-d_{vi}), \dots, u(t-d_{vi}-nb)] \quad (6.3)$$

$$Z(t) = [-y(t-d_{vi}), \dots, -y(t-d_{vi}-na)] \quad (6.4)$$

$$Z(t) = [-y(t-d_{vi}), \dots, -y(t-d_{vi}-na), u(t-d_{vi}), \dots, u(t-d_{vi}-nb)] \quad (6.5)$$

O atraso de $Z(t)$, $c_2=d_{vi}$, tem seu intervalo de atuação entre $[1;10]$. A reinicialização da matriz de covariância $P(t)$ é implementada com c_3 em valores no intervalo $[0.1;50]$ e, é efetuada quando o traço da matriz de covariância $P(t)$ é menor que 0.5.

O AG tem critério de parada de 2000 iterações, população de 30 indivíduos, probabilidade recombinação de 0.8, probabilidade de mutação de 0.1 e a seleção é por roleta. Os resultados obtidos com a VI com módulo de configuração genético e o algoritmo dos MQR com projeto convencional (heurístico) para identificação do processo de nível e térmico são apresentados na tabela 6.1.

Os parâmetros estipulados para o MQR são inicialização de $P(t) = 100I_p$, reinicialização de $P(t)$ com valor 5, e fator de esquecimento com crescimento exponencial entre 0.95 e 1 (Johansson, 1993; Montiel *et al.*, 1993; Salgado *et al.*, 1988; Seborg *et al.*, 1986).

Tabela 6.1 - Identificação do processo de nível e térmico pela VI e MQR.

Processo de nível								
MQR								
na	nb	a_1	a_2	b_1	b_2	ISE	SSE	RPE
1	1	-0.9978	-	0.0021	-	25.4321	1.69×10^{-2}	9.71×10^{-4}
1	2	-0.9958	-	0.2033	-0.2082	15.0152	1.00×10^{-2}	5.73×10^{-4}
2	1	-0.9912	-0.0021	0.1055	-	16.2377	1.08×10^{-2}	6.20×10^{-4}
2	2	-1.0004	-0.0390	0.0040	-0.0042	18.3002	1.22×10^{-2}	6.99×10^{-4}
VICAG								
na	nb	a_1	a_2	b_1	b_2	ISE	SSE	RPE
*1	1	-0.9962	-	0.0037	-	20.9970	1.40×10^{-2}	8.02×10^{-4}
**1	2	-0.9959	-	0.5048	-0.5149	14.8762	9.91×10^{-3}	5.68×10^{-4}
***2	1	-1.0019	-0.0061	0.6041	-	16.5217	1.10×10^{-2}	6.31×10^{-4}
****2	2	-1.0097	-0.0140	0.5021	0.5019	16.7630	1.12×10^{-2}	6.40×10^{-2}
<i>ajuste genético para inicialização da VI:</i>								
* montagem de Z(t): equação (6.4); atraso de Z(t) = 3 e reinicialização de P(t) = 36.19								
** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 48.39								
*** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 17.42								
**** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 14.20								
Processo térmico								
MQR								
na	nb	a_1	a_2	b_1	b_2	ISE	SSE	RPE
1	1	-0.9916	-	0.0081	-	20.7491	5.18×10^{-2}	4.12×10^{-3}
1	2	-0.9945	-	-0.2158	0.2222	15.1186	3.77×10^{-2}	2.99×10^{-3}
2	1	-0.9776	-0.1168	0.0064	-	15.4283	3.86×10^{-2}	3.06×10^{-3}
2	2	-1.0150	-0.0206	-0.0158	0.0221	13.2865	3.32×10^{-2}	2.63×10^{-3}
VICAG								
na	nb	a_1	a_2	b_1	b_2	ISE	SSE	RPE
*1	1	-0.9917	-	0.0079	-	20.5767	5.14×10^{-2}	4.08×10^{-3}
**1	2	-0.9936	-	0.4135	0.4136	9.7429	2.44×10^{-2}	1.93×10^{-3}
***2	1	-0.9955	0.0011	0.1165	-	11.2717	2.82×10^{-2}	2.24×10^{-3}
****2	2	-1.01526	0.02084	0.2648	0.0015	12.4447	3.11×10^{-2}	2.47×10^{-3}
<i>ajuste genético para inicialização da VI:</i>								
* montagem de Z(t): equação (6.4); atraso de Z(t) = 1 e reinicialização de P(t) = 0.67								
** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 46.96								
*** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 42.74								
**** montagem de Z(t): equação (6.5); atraso de Z(t) = 1 e reinicialização de P(t) = 1.02								

De acordo com os resultados expostos na tabela 6.1, nota-se que a VI com módulo de configuração genético apresenta resultados mais adequados na identificação do processo de nível e térmico que o MQR com projeto heurístico dos parâmetros, contudo demanda uma alta complexidade computacional, sendo indicado para identificação “off-line”.

O motivo é que os AGs trabalham por busca aleatória e precisam de todas as medidas do processo a ser identificado, ao contrário do MQR apresentado que é mais indicado para projeto em malha fechada, em controle adaptativo indireto e direto.

6.3.1.2 Computação evolucionária

As metodologias de CE aplicadas a identificação são: AGs, AGSA e EEs. A tabela 6.2 apresenta os parâmetros das metodologias de CE utilizadas à identificação experimental do processo de nível e térmico, descritos nas seções 6.2.1 e 6.2.2, respectivamente.

Tabela 6.2 - Parâmetros usados nas metodologias de CE.

n ^o .	técnica (população)	σ	seleção	recombinação	mutação
1	AG(30)	-	roleta	*p _r = 0.8	#p _m =0.1
2	AGSA(30)	-	roleta	*p _r = 0.8	#p _m =0.1
3	AGSA(30)	-	“breeder”	*p _r = 0.8	#p _m =0.1
4	EE-(1+1)	1	-	-	sim
5	EE-(5+25)	1	-	não	sim
6	EE-(5+25)	1	-	**	sim
7	EE-(1,29)***	1	-	-	correlata
8	EE-(1+29)****	1	-	-	correlata
convenções					
#	mutação não-uniforme com b=2				
*	recombinação aritmética				
**	recombinação discreta (x) e aritmética intermediária (σ)				
***	controle do tamanho de passo individual				
****	variações correlatas e esquema de adaptação dos parâmetros				

6.3.1.2.1 Identificação do processo de nível

Na tabela 6.3 são apresentados os resultados obtidos para estimação do processo de nível. Nas metodologias de CE adotadas são realizados 10 experimentos e o espaço de busca dos parâmetros é [-2;2]. O critério de parada adotado é de 2000 iterações, neste caso, número de avaliações da função de adequabilidade, contudo o projetista pode adotar outro critério de parada que leve em conta a tolerância do erro.

Uma comparação do critério de erro ISE nos 10 experimentos (média), do melhor indivíduo de cada geração, é apresentado na figura 6.5. A resposta real do processo em malha aberta e a identificação mais adequado do processo via CE é mostrada na figura 6.6.

O resíduo (r) da resposta real em relação a resposta estimada é apresentado na figura 6.7. A seguir, uma análise dos resultados obtidos com as metodologias de CE é tratada, apresentando-se os critérios de erro obtidos e parâmetros estimados.

Tabela 6.3 - Identificação do processo nível por técnicas de computação evolucionária.

n.º	na	nb	a_1	a_2	b_1	b_2	ISE	SSE	RPE
1	1	1	-0.9958	-	0.0039	-	15.2700	1.02×10^{-2}	5.83×10^{-4}
2	1	1	-0.9958	-	0.0039	-	15.3544	1.02×10^{-2}	5.86×10^{-4}
3	1	1	-0.9958	-	0.0040	-	15.1523	1.01×10^{-2}	5.78×10^{-4}
4	1	1	-0.9956	-	0.0041	-	16.8942	1.12×10^{-2}	6.45×10^{-4}
5	1	1	-0.9958	-	0.0040	-	15.1523	1.01×10^{-2}	5.78×10^{-4}
6	1	1	-0.9958	-	0.0040	-	15.1523	1.01×10^{-2}	5.78×10^{-4}
7	1	1	-0.9958	-	0.0040	-	15.1523	1.01×10^{-2}	5.78×10^{-4}
8	1	1	-0.9958	-	0.0040	-	15.1523	1.01×10^{-2}	5.78×10^{-4}
<hr/>									
1	1	2	-0.9957	-	0.0531	-0.0490	15.5212	1.03×10^{-2}	5.93×10^{-4}
2	1	2	-0.9959	-	0.0882	-0.0843	14.2665	9.51×10^{-3}	5.45×10^{-4}
3	1	2	-0.9957	-	-0.3115	0.3155	14.3508	9.57×10^{-3}	5.48×10^{-4}
4	1	2	-0.9960	-	0.8562	-0.8523	23.4102	1.56×10^{-2}	8.94×10^{-4}
5	1	2	-0.9958	-	0.8305	-0.8266	14.1431	9.42×10^{-3}	5.40×10^{-4}
6	1	2	-0.9958	-	-0.1732	0.1771	14.1528	9.44×10^{-3}	5.41×10^{-4}
7	1	2	-0.9958	-	0.5878	-0.5839	14.3694	9.58×10^{-3}	5.49×10^{-4}
8	1	2	-0.9958	-	0.5878	-0.5839	14.3694	9.58×10^{-3}	5.49×10^{-4}
<hr/>									
1	2	1	-0.0418	-0.9489	0.0081	-	94.1770	6.28×10^{-2}	3.60×10^{-3}
2	2	1	-0.1227	-0.9793	0.0080	-	15.0203	1.00×10^{-2}	5.74×10^{-4}
3	2	1	-0.2833	-0.7095	0.0067	-	15.0753	1.01×10^{-2}	5.76×10^{-4}
4	2	1	-0.0361	-0.9561	0.0075	-	23.0083	1.53×10^{-2}	8.79×10^{-4}
5	2	1	-0.1777	-0.8146	0.0072	-	14.9133	9.94×10^{-3}	5.70×10^{-4}
6	2	1	-0.5536	-0.4403	0.0057	-	14.9903	9.99×10^{-3}	5.73×10^{-4}
7	2	1	-0.2217	-0.7709	0.0071	-	15.3394	1.02×10^{-2}	5.86×10^{-4}
8	2	1	-0.2216	-0.7709	0.0071	-	15.3250	1.02×10^{-2}	5.85×10^{-4}
<hr/>									
1	2	2	-0.0358	-0.9511	-0.3718	0.3830	189.800	1.26×10^{-1}	7.24×10^{-3}
2	2	2	-0.0219	-0.9700	-0.1285	0.1363	14.4840	9.66×10^{-3}	5.53×10^{-4}
3	2	2	-0.9153	-0.0801	0.5456	-0.5413	14.9236	9.95×10^{-3}	9.95×10^{-3}
4	2	2	-0.5019	-0.4919	-0.2013	0.2071	16.0045	1.07×10^{-2}	6.11×10^{-4}
5	2	2	-0.2521	-0.7406	0.5442	-0.5373	14.3826	9.59×10^{-3}	5.49×10^{-4}
6	2	2	-0.2323	-0.7577	-0.1302	0.1394	73.2236	4.88×10^{-2}	2.79×10^{-3}
7	2	2	-0.6201	-0.3741	-0.1138	0.1193	15.3356	1.02×10^{-2}	5.85×10^{-4}
8	2	2	-0.3338	-0.6593	-0.3443	0.3509	15.3089	1.02×10^{-2}	5.84×10^{-4}

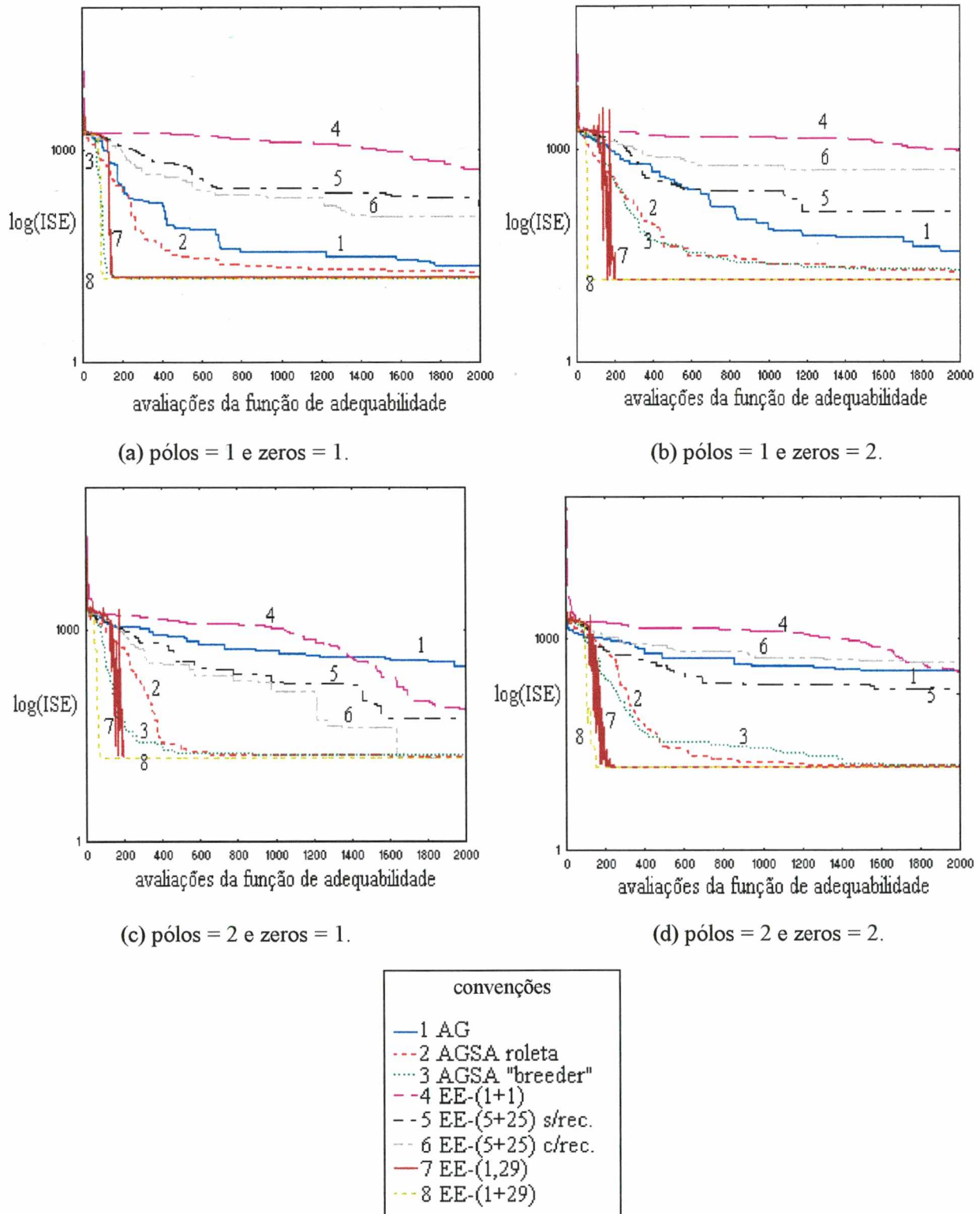


Figura 6.5 - Comparação do critério ISE nos 10 experimentos (média) do processo de nível.

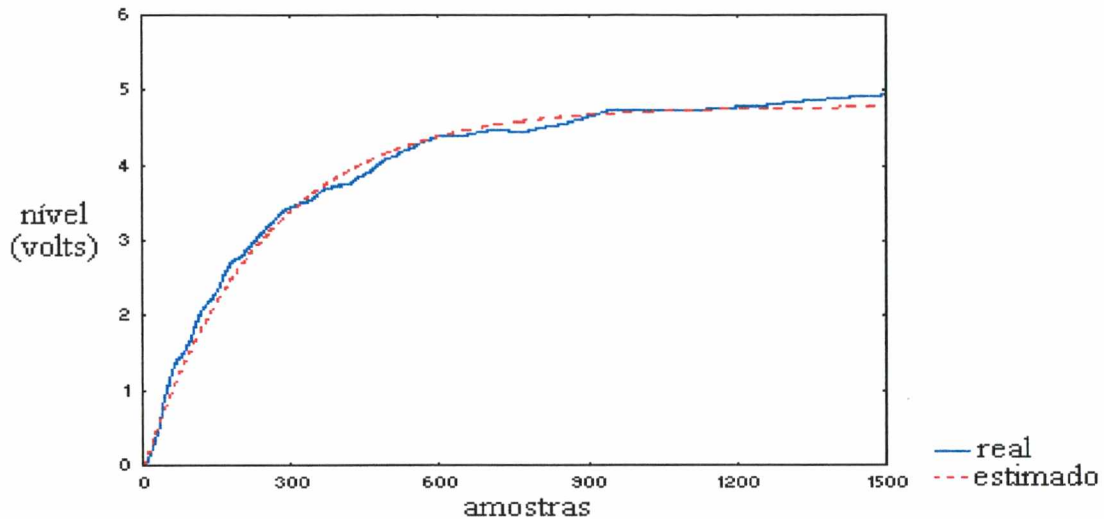


Figura 6.6 - Resposta real e estimada do processo de nível em malha aberta via CE.

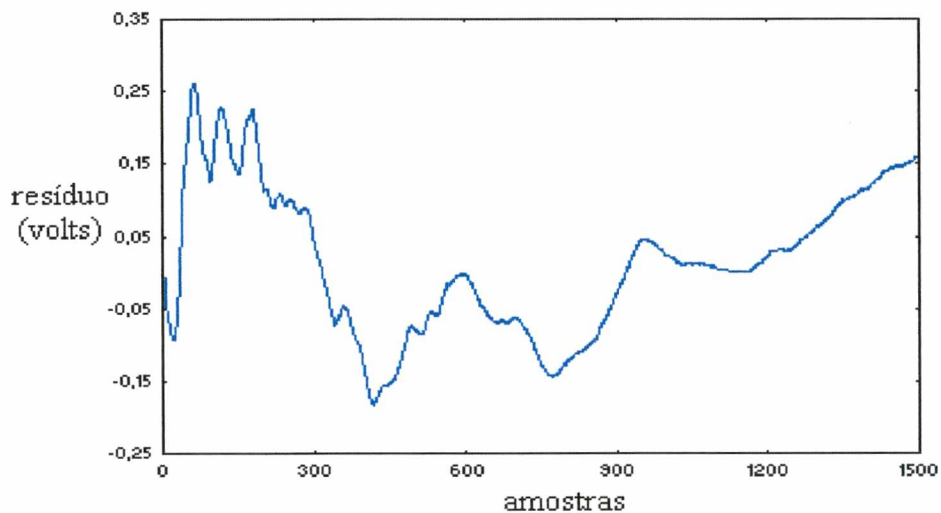


Figura 6.7 - Resíduo do processo de nível em malha aberta via CE.

O tempo de CPU (em microcomputador 486 DX4 de 100 Mhz com 8 MB de memória RAM) despendido em cada experimento de 2000 iterações para a identificação do processo de nível pelas metodologias da CE com numeração, conforme tabela 6.2, de 1, 4, 5 e 6 é entre [2min 19s] e [2min 50s], das metodologias 2,3 e 7 é entre [3min 28s] e [3min 53s] e a metodologia 8 é a de maior tempo de CPU com o tempo de otimização entre [7min 29s] e [8min 12s].

Contudo, é necessário mencionar que existe a possibilidade da adaptação, em trabalhos futuros, dos algoritmos desenvolvidos em linguagens C e C++ para o ambiente do software MATLAB da MathWorks Inc., para adequada análise do número de operações de ponto flutuante efetuadas e conseqüente quantificação da complexidade computacional de cada metodologia aplicada em identificação e controle de processos.

A partir dos resultados experimentais da modelagem de 1ª e 2ª ordem para o processo de nível com as técnicas de CE obteve-se como melhor modelagem para um pólo ($a_1=-0.9958$) e dois zeros ($b_1=0.8305$, $b_2=-0.8266$), pela técnica EE-(5+25) com utilização de um desvio padrão e operador de recombinação, como abordado na tabela 6.3. Comparando-se estes resultados com os coeficientes obtidos pelos algoritmos MQR e VICAG, nota-se uma diferenciação acentuada nos valores dos polinômios de $B(z^{-1})$ obtidos, conforme apresentado na tabela 6.1. É necessário enfatizar, contudo, que os paradigmas da CE “guiam-se” na busca dos parâmetros apenas pelo valor da função de adequabilidade e não em gradiente.

Os valores obtidos pela minimização do ISE na identificação dos processos tem valores próximos, porém quanto a velocidade de convergência, analisando-se a figura 6.5 (critério de erro ISE em escala logarítmica), nota-se que a técnica EE-(1+29) com variações correlatas e esquema de adaptação dos parâmetros é o algoritmo que converge mais rápido, entre 50 a 150 avaliações da função de adequabilidade, seguido pelo algoritmo EE-(1,29) e AGSA.

6.3.1.2.2 Identificação do processo térmico

Na tabela 6.4 são apresentados os resultados da estimação de parâmetros para o processo térmico. De maneira análoga ao procedimento aplicado ao processo de nível, nas metodologias de CE adotadas são realizados 10 experimentos e o espaço de busca dos parâmetros é $[-2;2]$.

Devido ao fato do processo térmico tratar-se de um processo lento e apresentar atraso de transporte, opta-se por realizar também experimentos para estimação do atraso de transporte do processos através de CE, para espaço de busca entre $[-1;1]$ e $[-2;2]$.

A codificação do atraso de transporte é estipulada para um dígito que pode variar no intervalo $[0;9]$. Os resultados são apresentados na tabela 6.5, enquanto que figura 6.8 é apresentada a estimação mais adequada do modelo do processo térmico (a) *sem* e, (b) *com* estimação do atraso de transporte. A seguir, é apresentada uma análise dos resultados obtidos com as técnicas AGs, AGSA e EEs.

Tabela 6.4 - Identificação do processo térmico por técnicas de computação evolucionária.

n ^o .	na	nb	a ₁	a ₂	b ₁	b ₂	ISE	SSE	RPE
1	1	1	-0.9957	-	0.0053	-	9.2633	2.32x10 ⁻²	1.84x10 ⁻³
2	1	1	-0.9959	-	0.0052	-	9.2374	2.31x10 ⁻²	1.83x10 ⁻³
3	1	1	-0.9958	-	0.0052	-	9.0960	2.27x10⁻²	1.80x10⁻³
4	1	1	-0.9959	-	0.0052	-	9.4590	2.36x10 ⁻²	1.88x10 ⁻³
5	1	1	-0.9958	-	0.0052	-	9.0960	2.27x10⁻²	1.80x10⁻³
6	1	1	-0.9958	-	0.0052	-	9.0960	2.27x10⁻²	1.80x10⁻³
7	1	1	-0.9958	-	0.0052	-	9.0960	2.27x10⁻²	1.80x10⁻³
8	1	1	-0.9958	-	0.0052	-	9.0960	2.27x10⁻²	1.80x10⁻³
<hr/>									
1	1	2	-0.9958	-	0.1616	-0.1564	10.5768	2.64x10 ⁻²	2.10x10 ⁻³
2	1	2	-0.9959	-	0.2318	-0.2267	10.4907	2.62x10 ⁻²	2.08x10 ⁻³
3	1	2	-0.9960	-	0.0334	-0.0283	10.4907	2.62x10 ⁻²	2.08x10 ⁻³
4	1	2	-0.9959	-	0.7176	-0.7124	11.5616	2.89x10 ⁻²	2.29x10 ⁻³
5	1	2	-0.9960	-	-0.9201	0.9253	10.4763	2.62x10 ⁻²	2.08x10 ⁻³
6	1	2	-0.9960	-	-0.2936	0.2988	10.4825	2.62x10 ⁻²	2.08x10 ⁻³
7	1	2	-0.9960	-	0.7069	-0.7017	10.6725	2.67x10 ⁻²	2.12x10 ⁻³
8	1	2	-0.9960	-	0.7069	-0.7017	10.6725	2.67x10 ⁻²	2.12x10 ⁻³
<hr/>									
1	2	1	-0.9932	-0.0136	-0.0068	-	35.9040	8.98x10 ⁻²	7.12x10 ⁻³
2	2	1	-0.6661	-0.3270	0.0087	-	9.3876	2.35x10 ⁻²	1.86x10 ⁻³
3	2	1	-0.9992	0.0034	0.0052	-	9.1216	2.28x10 ⁻²	1.81x10 ⁻³
4	2	1	-0.6279	-0.3665	0.0070	-	9.4564	2.36x10 ⁻²	1.88x10 ⁻³
5	2	1	-0.9523	-0.0432	0.0054	-	9.1253	2.28x10 ⁻²	1.81x10 ⁻³
6	2	1	-0.2160	-0.7766	0.0093	-	9.4055	2.35x10 ⁻²	1.87x10 ⁻³
7	2	1	-0.6352	-0.3511	0.0053	-	9.4812	2.37x10 ⁻²	1.88x10 ⁻³
8	2	1	-0.9993	0.00351	0.0052	-	9.2938	2.32x10 ⁻²	1.84x10 ⁻³
<hr/>									
1	2	2	-0.9877	-0.0082	-0.6287	0.6369	33.6011	8.40x10 ⁻²	6.67x10 ⁻³
2	2	2	-0.2477	-0.7452	-0.2486	0.2576	9.9900	2.50x10 ⁻²	1.98x10 ⁻³
3	2	2	-0.6467	-0.3476	-0.7036	0.7106	9.8986	2.47x10 ⁻²	1.96x10 ⁻³
4	2	2	-0.2002	-0.7930	0.1116	-0.1026	10.3318	2.58x10 ⁻²	2.05x10 ⁻³
5	2	2	-0.9947	-0.0012	0.4856	-0.4804	9.6777	2.42x10 ⁻²	1.92x10 ⁻³
6	2	2	-0.4536	-0.5401	0.4083	-0.4003	9.9292	2.48x10 ⁻²	1.97x10 ⁻³
7	2	2	-0.1863	-0.8061	0.3185	-0.3089	11.7350	2.93x10 ⁻²	2.33x10 ⁻³
8	2	2	-0.6670	-0.3218	0.0391	-0.0322	10.0459	2.51x10 ⁻²	1.99x10 ⁻³

A partir dos resultados experimentais da modelagem de 1^a e 2^a ordem para o processo térmico com as técnicas de CE obteve-se como melhor modelagem para um pólo ($a_1=-0.9958$) e um zero ($b_1=0.0052$) através de cinco técnicas (3,5,6,7 e 8) da CE. A convergência das metodologias de CE não é apresentada pois é bastante similar aos resultados obtidos com a identificação do processo de nível, expostos na figura 6.5.

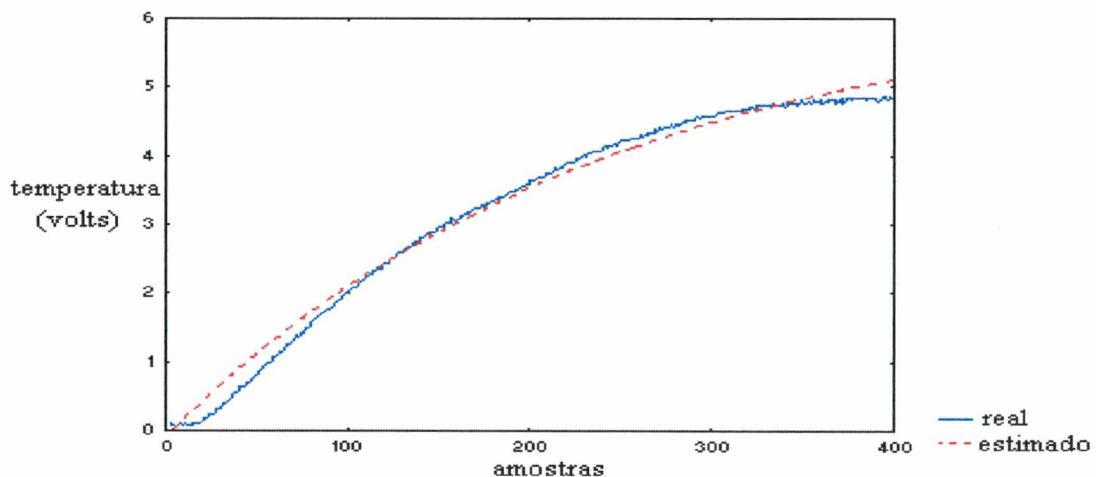
Tabela 6.5 - Identificação do processo térmico com estimação do atraso do transporte via CE.

<i>intervalo de busca dos polinômios $A(z^{-1})$ e $B(z^{-1})$ de $[-2;2]$</i>										
n ^o .	pólos	zeros	a_1	a_2	b_1	b_2	d	ISE	SSE	RPE
1	2	2	-0.1732	-0.8230	-1.861	1.8701	9	114.202	2.86×10^{-1}	2.27×10^{-2}
2	1	1	-0.9949	-	0.0059	-	3	3.9704	9.92×10^{-3}	7.88×10^{-4}
3	1	1	-0.9952	-	0.0058	-	7	3.9533	9.88×10^{-3}	7.84×10^{-4}
4	1	1	-0.9956	-	0.0052	-	2	9.0182	2.25×10^{-2}	1.79×10^{-3}
5	1	2	-0.9959	-	0.1623	-0.1599	2	10.4556	2.61×10^{-2}	2.07×10^{-3}
6	1	1	-0.9950	-	0.1501	-0.1456	3	12.0432	3.01×10^{-2}	2.39×10^{-3}
7	1	1	-0.9950	-	0.0059	-	2	3.9561	9.89×10^{-3}	7.85×10^{-4}
8	1	1	-0.9950	-	0.0058	-	2	3.9200	9.80×10^{-3}	7.78×10^{-4}

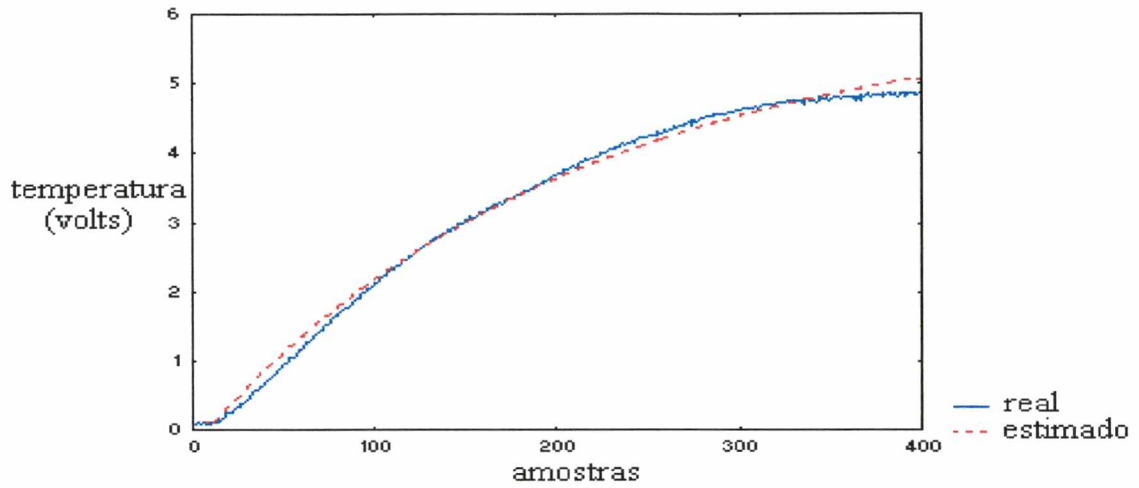
<i>intervalo de busca dos polinômios $A(z^{-1})$ e $B(z^{-1})$ de $[-1;1]$</i>										
n ^o .	pólos	zeros	a_1	a_2	b_1	b_2	d	ISE	SSE	RPE
1	1	1	-0.9952	-	0.0058	-	4	4.0822	1.02×10^{-2}	8.10×10^{-4}
2	1	1	-0.9951	-	0.0058	-	7	3.9433	9.86×10^{-3}	7.82×10^{-4}
3	1	1	-0.9950	-	0.0059	-	6	3.9543	9.89×10^{-3}	7.85×10^{-4}
4	1	1	-0.9950	-	0.0059	-	6	3.9543	9.89×10^{-3}	7.85×10^{-4}
5	1	1	-0.9951	-	0.0058	-	5	3.9406	9.85×10^{-3}	7.81×10^{-4}
6	1	1	-0.9950	-	0.0059	-	4	3.9632	9.91×10^{-3}	7.86×10^{-4}
7	1	1	-0.9950	-	0.0058	-	4	3.9560	9.89×10^{-3}	7.85×10^{-4}
8	1	1	-0.9950	-	0.0058	-	2	3.9200	9.80×10^{-3}	7.78×10^{-4}

Na análise da tabela 6.5 observa-se que a estimativa de parâmetros mais adequada é a realizada pela metodologia EE-(1+29) com variações correlatas e esquema de adaptação dos parâmetros, em estimação dos parâmetros do processo térmico, com um pólo ($a_1=-0.9950$) e um zero ($b_1=0.0058$) e atraso de transporte estimado em $d=2$.

O resultado é utilizado em traçado dos gráficos de resposta real e estimada do processo térmico em malha aberta, conforme apresenta a figura 6.8. O resíduo (equação 6.5) da resposta real em relação a resposta estimada é apresentado na figura 6.9.

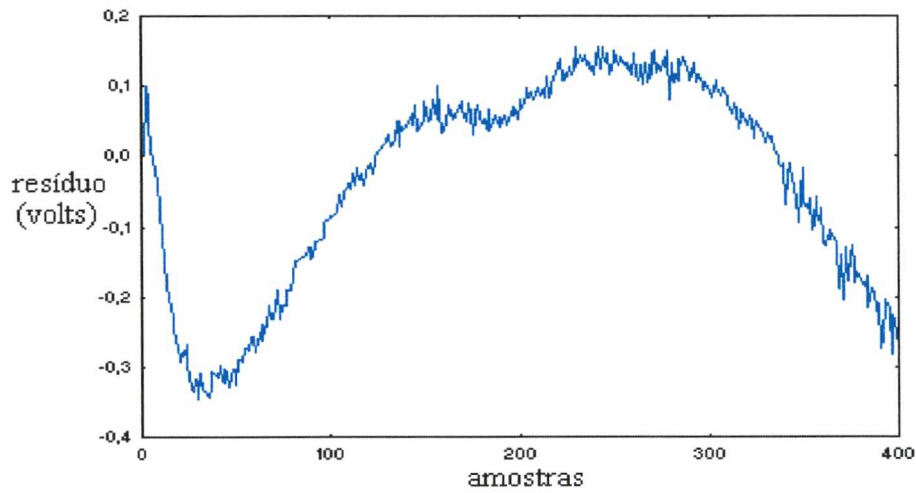


(a) sem estimação do atraso de transporte

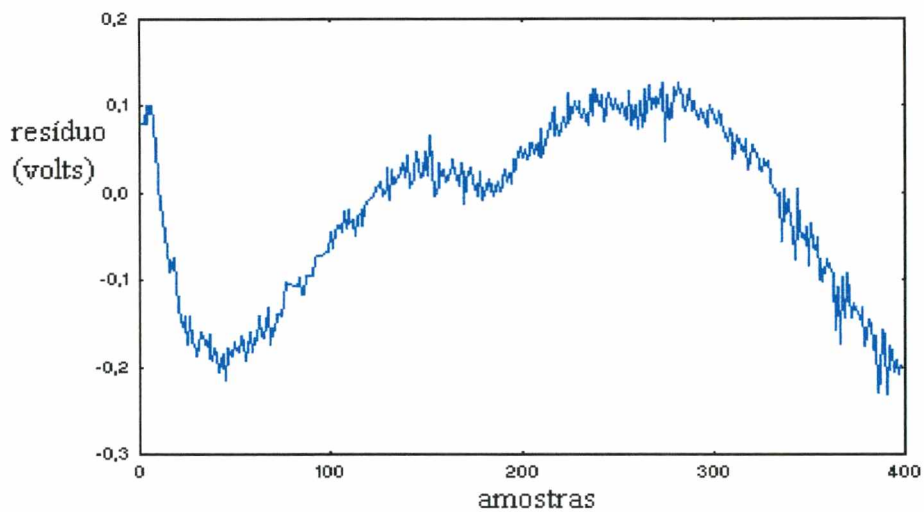


(b) com estimação do atraso de transporte

Figura 6.8 - Resposta real e estimada do processo térmico em malha aberta via CE.



(a) sem estimação do atraso de transporte



(b) com estimação do atraso de transporte

Figura 6.9 - Resíduo do processo térmico em malha aberta via CE.

O tempo de CPU (em microcomputador 486 DX4 de 100 Mhz com 8 MB de memória RAM) despendido em cada experimento de 2000 iterações para a identificação do processo térmico pelas metodologias da CE com numeração, conforme tabela 6.2, de 1, 4, 5 e 6 é entre [2min 04s] e [2min 20s], das metodologias 2,3 e 7 é entre [2min 40s] e [2min 52s] e a metodologia 8 é a de maior tempo de CPU com o tempo de otimização entre [3min 23s] e [3min 46s].

6.3.1.3 Redes neurais artificiais

O algoritmo aplicado na estimação do modelo “black-box” do processo de nível e térmico é realizado por uma rede multicamadas com treinamento BP “feedforward” e RNPREMOMOD.

A aceleração do aprendizado das RNs é obtido pela substituição da *inicialização dos pesos* da RN aleatoriamente pela aplicação de AGs. Os resultados obtidos são apresentados na tabela 6.6 e 6.7.

Na identificação dos processos pela RN é estipulado como critério de parada 2000 iterações, tanto para o processo de nível mono-tanque quanto para o processo térmico. O coeficiente de aprendizado e momento são setados para 0.1, enquanto que o parâmetro α que habilita as unidades de contexto em agregar mais (ou menos) os valores das saídas passadas do processo na estrutura da RNPREMOMOD é ajustado para 0.6.

A resposta real e melhor resposta estimada do processo de nível e térmico são apresentadas nas figuras 6.10 e 6.13, respectivamente. O resíduo (equação (6.2)) da resposta real em relação a resposta estimada dois processos são apresentadas na figura 6.11 e 6.14.

As figuras 6.12 e 6.15 apresentam as curvas de aprendizado do processo de nível e térmico via RNs.

Tabela 6.6 - Identificação do processo de nível via RNs.

<i>Rede neural "feedforward"</i>								
neurônios	entradas				saída	índice de desempenho		
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE
2-20-1	X		X		X	0.8185	5.45×10^{-4}	3.13×10^{-5}
3-20-1	X		X	X	X	1.4904	9.93×10^{-4}	5.69×10^{-5}
3-20-1	X	X	X		X	0.5726	3.82×10^{-4}	2.19×10^{-5}
4-20-1	X	X	X	X	X	1.4037	9.36×10^{-4}	5.36×10^{-6}

<i>Rede neural parcialmente recorrente de Elman modificada</i>								
neurônios	entradas				saída	índice de desempenho		
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE
2-20-1	X		X		X	0.2602	1.73×10^{-4}	9.94×10^{-6}

Tabela 6.7 - Identificação do processo térmico via RNs.

<i>Rede neural "feedforward"</i>								
neurônios	entradas				saída	índice de desempenho		
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE
2-20-1	X		X		X	1.9172	4.79×10^{-3}	3.80×10^{-4}
3-20-1	X		X	X	X	2.0421	5.10×10^{-3}	4.05×10^{-4}
3-20-1	X	X	X		X	2.6618	6.65×10^{-3}	1.02×10^{-4}
4-20-1	X	X	X	X	X	2.8553	7.13×10^{-3}	1.09×10^{-4}

<i>Rede neural parcialmente recorrente de Elman modificada</i>								
neurônios	entradas				saída	índice de desempenho		
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE
2-20-1	X		X		X	1.5428	3.86×10^{-3}	3.06×10^{-4}

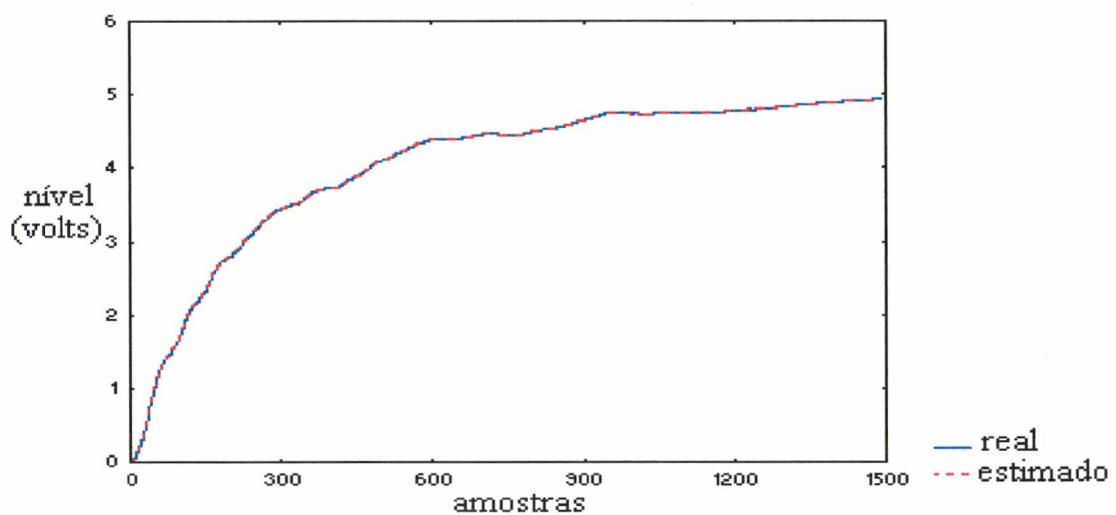


Figura 6.10 - Resposta real e estimada do processo de nível em malha aberta via RNs.

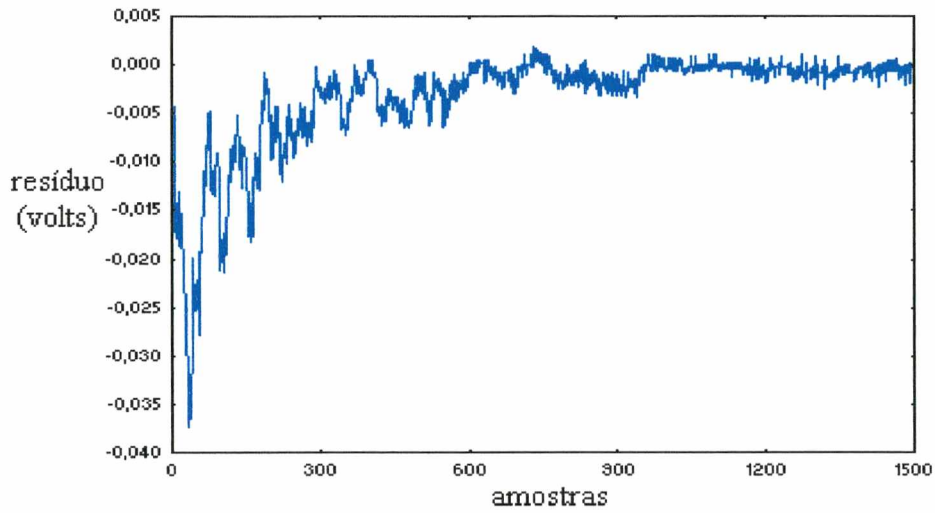


Figura 6.11 - Resíduo do processo de nível em malha aberta via RNs.

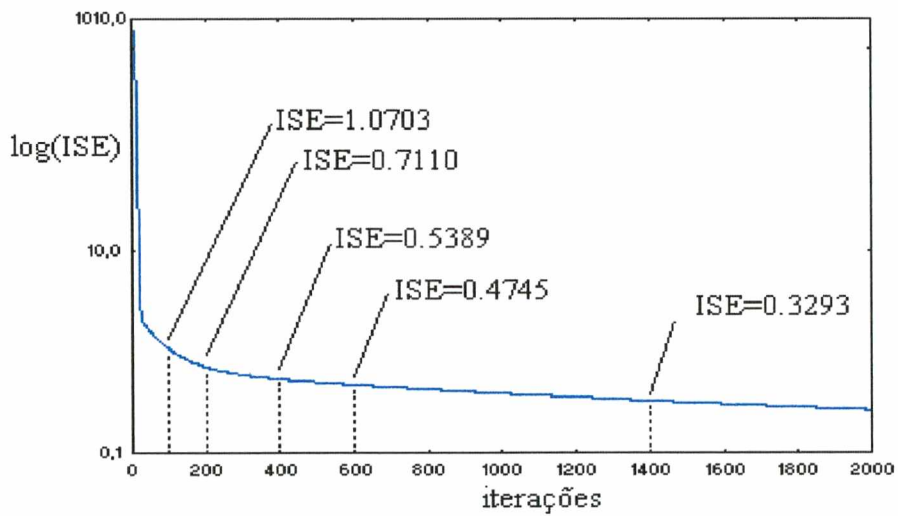


Figura 6.12 - Curva de aprendizado do processo de nível via RNs.

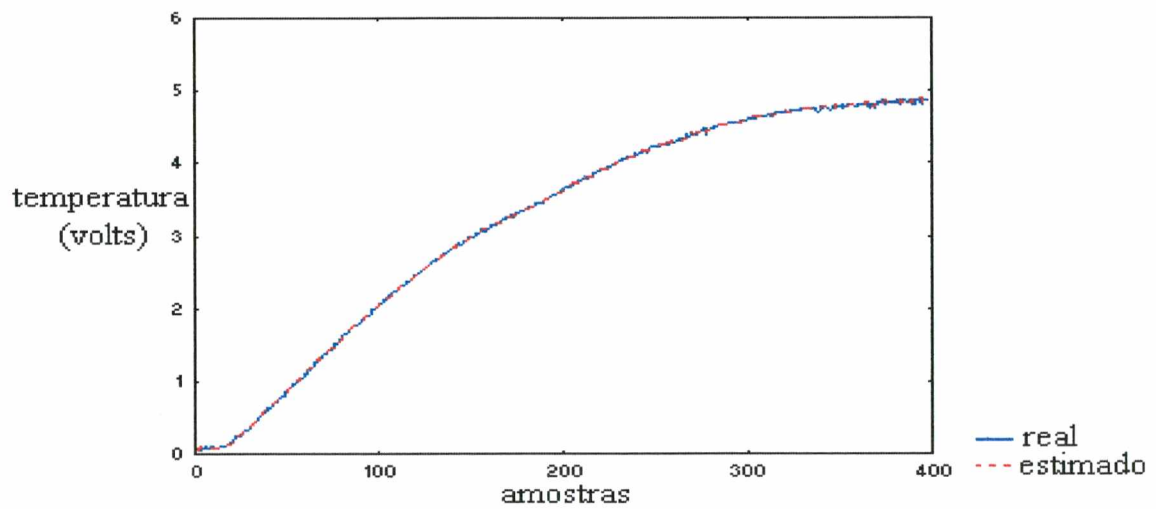


Figura 6.13 - Resposta real e estimada do processo térmico em malha aberta via RNs.

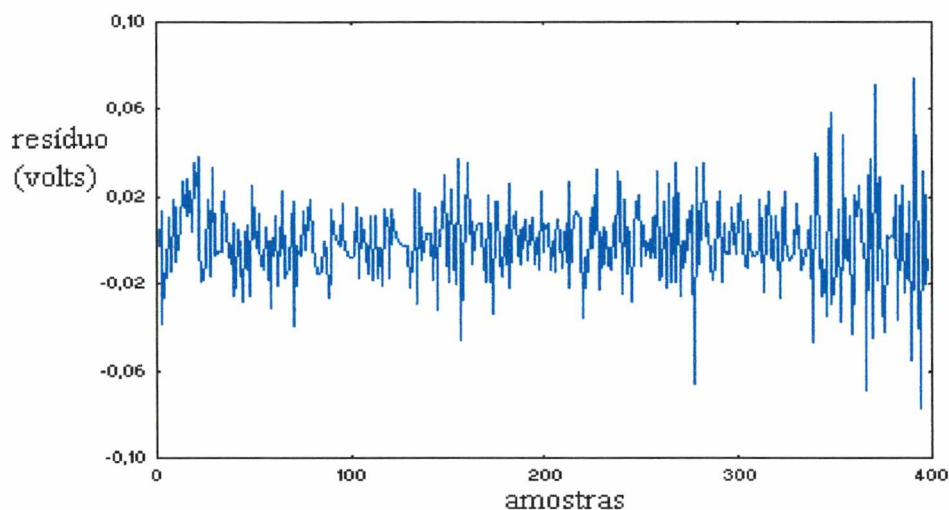


Figura 6.14 - Resíduo do processo térmico em malha aberta via RNs.

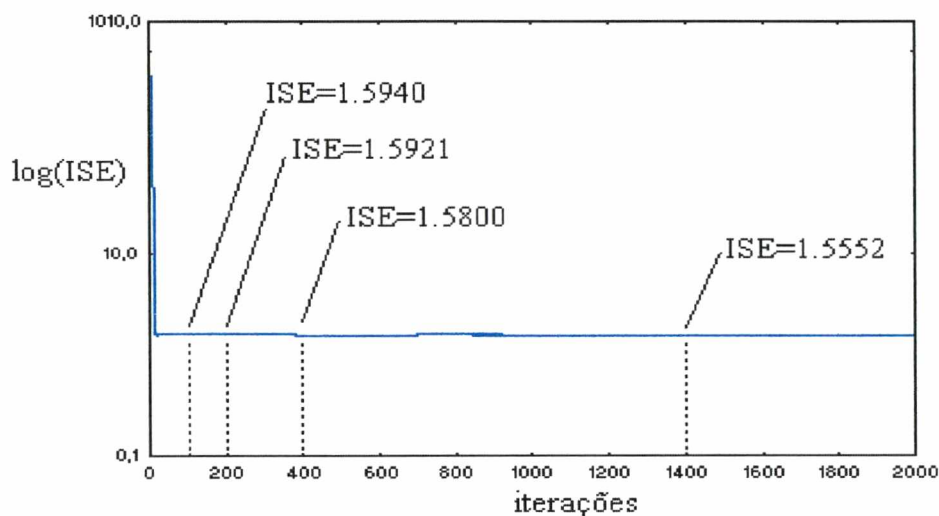


Figura 6.15 - Curva de aprendizado do processo térmico via RNs.

A partir da análise dos resultados apresentados na tabela 6.4 nota-se que as RNs modelaram adequadamente os processos de nível e térmico tanto para primeira quanto para segunda ordem, onde os erros para o número de iterações estipulado são mínimos.

O tempo de CPU (em microcomputador 486 DX4 de 100 Mhz com 8 MB de memória RAM) para a identificação do processo de nível pelas RN é entre [2h 12min 04s] e [2h 20min 20s], enquanto a identificação do processo térmico demandou entre [35min 39s] e [42min 16s].

As RNs apresentam menores erros em relação as metodologias de CE e algoritmos dos MQR e VI, contudo possuem uma alta complexidade computacional e demandam um tempo de CPU maior que as outras técnicas.

Dentre as vantagens da RNs menciona-se não haver necessidade de modelo matemático da planta para a modelagem, além do que as RNs se comportam bem para modelagem tanto de processos lineares quanto não-lineares. No caso dos processos analisados, as RNs aproximam quase todas as não-linearidades dos processos de nível e térmico, conforme apresentado nas figuras 6.11 e 6.13.

6.3.2 Identificação do turbo-gerador

A identificação do turbo-gerador utiliza-se da equação (6.1) que descreve um processo real (turbo-gerador) de laboratório para identificação não-linear (Pickhardt and Unbehauen, 1994). As simulações visaram dois experimentos, isto é:

Experimento 1:

A estimação do modelo matemático do processo é realizada utilizando-se um sinal de entrada degrau unitário, durante as amostras 0 a 100. Entre as amostras 100 e 300, é aplicado a entrada do processo um sinal PRBS₁ (sinal degrau unitário mais $\pm 5\%$ de um PRBS (“Pseudo-Random Binary Sequence”) de -1 ou +1). Uma seqüência aleatória normalmente distribuída (ruído branco) com média zero e variância um, entre as amostras 300 e 500. O sinal de entrada (degrau + PRBS + ruído branco) aplicado ao turbo-gerador em malha aberta é apresentado na figura 6.16.

Os algoritmos de identificação utilizam-se das primeiras 500 amostras para identificar o turbo-gerador. Após a fase de estimação, os modelos matemáticos são validados frente a outras 300 amostras (entre 500 e 800) através de uma entrada PRBS₂ (sinal degrau unitário mais $\pm 10\%$ de um PRBS de -1 ou +1), utilizando-se de semente de números aleatórios diferente do PRBS₁.

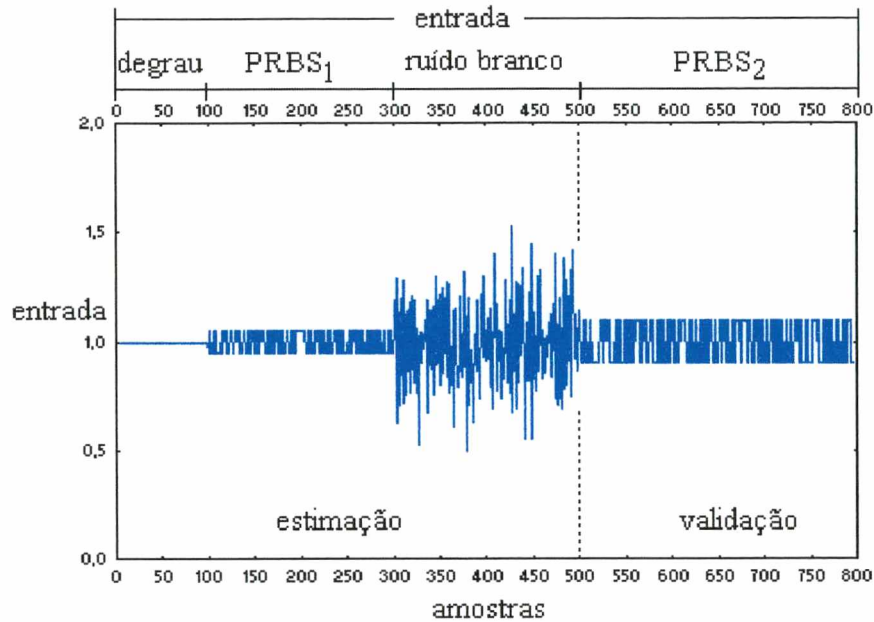


Figura 6.16 - Sinal de entrada aplicado ao turbo-gerador em malha aberta (experimento 1).

Experimento 2:

A estimação do modelo matemático do processo é realizada utilizando-se um sinal de entrada $PRBS_1$ (-1 ou 1), durante as amostras 0 a 300, e uma seqüência aleatória normalmente distribuída (ruído branco) com média zero e variância um, entre as amostras 300 e 500. O sinal de entrada ($PRBS +$ ruído branco) aplicado ao turbo-gerador em malha aberta é apresentado na figura 6.17.

Os algoritmos de identificação utilizam-se das primeiras 500 amostras para identificar o turbo-gerador. Após a fase de estimação, os modelos matemáticos são validados frente a outras 300 amostras (entre 500 e 800) através de uma entrada $PRBS_2$ (-1 ou +1), utilizando-se de semente de números aleatórios diferente do $PRBS_1$.

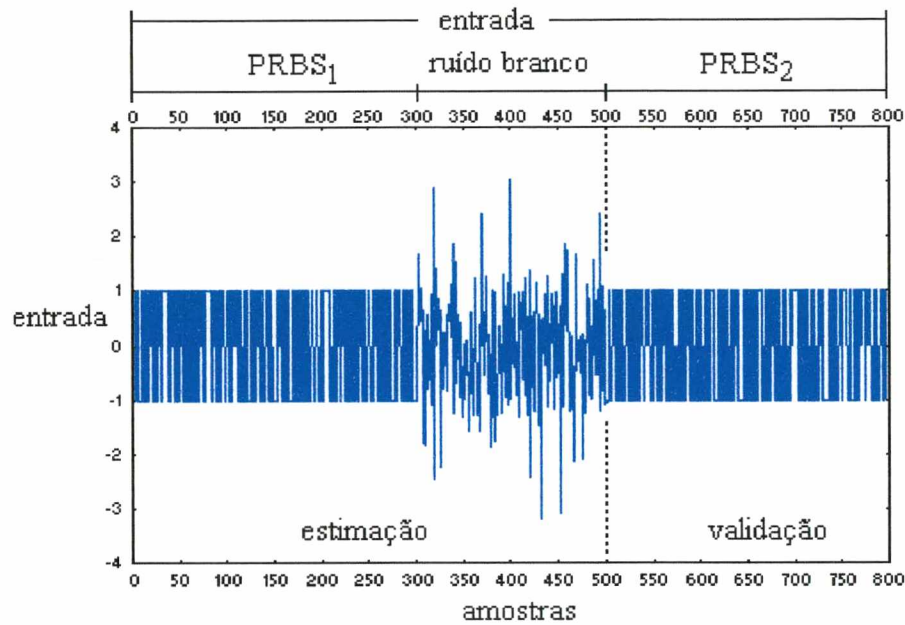


Figura 6.17 - Sinal de entrada (experimento 2) aplicado ao turbo-gerador em malha aberta.

A aplicação dos algoritmos MQR, VI com configuração via AGs e CE objetiva a estimação dos parâmetros de um modelo matemático NARX para o turbo-gerador.

Os modelos NARX são regidos por equação diferenças não-lineares estruturadas por parâmetros lineares. Estes modelos são função da entradas e saídas passadas, conforme equações (2.3) e (2.4), contudo apresentam termos não-lineares. A estrutura típica de um modelo NARX pode ser abordada pela seguinte equação:

$$y(t) = \sum_{i=1}^{nb} b_i u(t - d - i + 1) - \sum_{i=1}^{na} a_i y(t - i) + cte + \tag{6.6}$$

(termos não - lineares)

$$+ \sum_{i=1}^{na} \sum_{j=1}^{nb} \sum_{k=1}^{pep-1} \sum_{m=1}^{pep-1} c_{ijkm} u(t - d - i + 1)^k y(t - j)^m \Big|_{\text{maximo } k + m = 1}$$

$$+ \sum_{i=1}^{nb} \sum_{j=1}^{nb} \sum_{k=1}^{pep-1} \sum_{m=1}^{pep-1} d_{ijkm} u(t - d - i + 1)^k u(t - d - j + 1)^m \Big|_{\text{maximo } k + m = 1}$$

$$+ \sum_{i=1}^{na} \sum_{j=1}^{na} \sum_{k=1}^{pep-1} \sum_{m=1}^{pep-1} e_{ijkm} y(t - i)^k y(t - j)^m \Big|_{\text{maximo } k + m = 1}$$

O número máximo de termos (n_t) no modelo NARX é dado por:

$$n_t = M + 1 \quad (6.7)$$

onde

$$M = \sum_{i=1}^p n_i \quad (6.8)$$

e

$$n_i = \{n_{i-1} (na + nb + i - 1)\} / i, \text{ onde } n_0=1 \quad (6.9)$$

onde “nb” obedece a equação (2.3), “na” é dado de acordo com a equação (2.4). O termo “pep” é responsável pela potência de expansão polinomial. O modelo NARX utilizado na identificação do turbo-gerador é estipulado em nb=3, na = 2 e pep=3, o que totaliza $n_t = 56$ termos (Chen *et al.*, 1990; Li and Jeon, 1993).

A utilização do algoritmo da VICAG visa identificar o processo com 15 termos no máximo permitidos do modelo matemático apresentado na equação (6.6). Neste caso, optou-se por AG com representação binária, onde se o cromossomo tem valor “1” significa a presença do termo e “0” significa ausência do termo. O número de gerações adotado é de 2000 gerações (Li and Jeon, 1993).

Se algum cromossomo do AG passar mais que 15 termos para o algoritmo da VI a função de adequabilidade sofre uma penalização, consistindo de um acréscimo no valor desta de uma constante ρ . Enquanto os algoritmos dos MQR e paradigmas da CE visam a estimação dos 56 termos do modelo NARX para identificação do turbo-gerador.

6.3.2.1 Mínimos quadrados recursivo e variável instrumental

Nos experimentos 1 e 2 são utilizados o algoritmo MQR e VICAG. No experimento 1, o resultado mais adequado para identificação do turbo-gerador é obtido com o algoritmo VICAG com os seguintes dados: montagem de $Z(t)$ pela equação (6.6), atraso de $Z(t) = 1$, reinicialização de $P(t) = 23.95$ e índice de desempenho $ISE = 7.43$ ($SSE=1.49 \times 10^{-2}$; $RPE=5.11 \times 10^{-3}$).

As figuras 6.18 e 6.19 apresentam a resposta real e estimada do turbo-gerador em malha aberta através da VICAG e resíduo para o experimento 1, respectivamente. A tabela 6.8

apresenta os 15 termos relevantes do modelo NARX obtidos pelo algoritmo VICAG na identificação mais adequada efetuada do turbo gerador no experimento 1.

Tabela 6.8 - Termos NARX obtidos pelo algoritmo VICAG (experimento 1).

termo	valor	termo	valor	termo	valor
b_1	0.1797	c_{2111}	0.4976	d_{1221}	0.3071
b_2	-0.1683	c_{2112}	-0.1149	e_{1112}	-0.1167
b_3	0.3368	c_{1121}	0.2737	e_{2112}	-0.2179
a_1	0.2728	d_{2111}	-0.1890	e_{1221}	0.2241
a_2	-0.2476	d_{3211}	0.2595	cte	0.1958

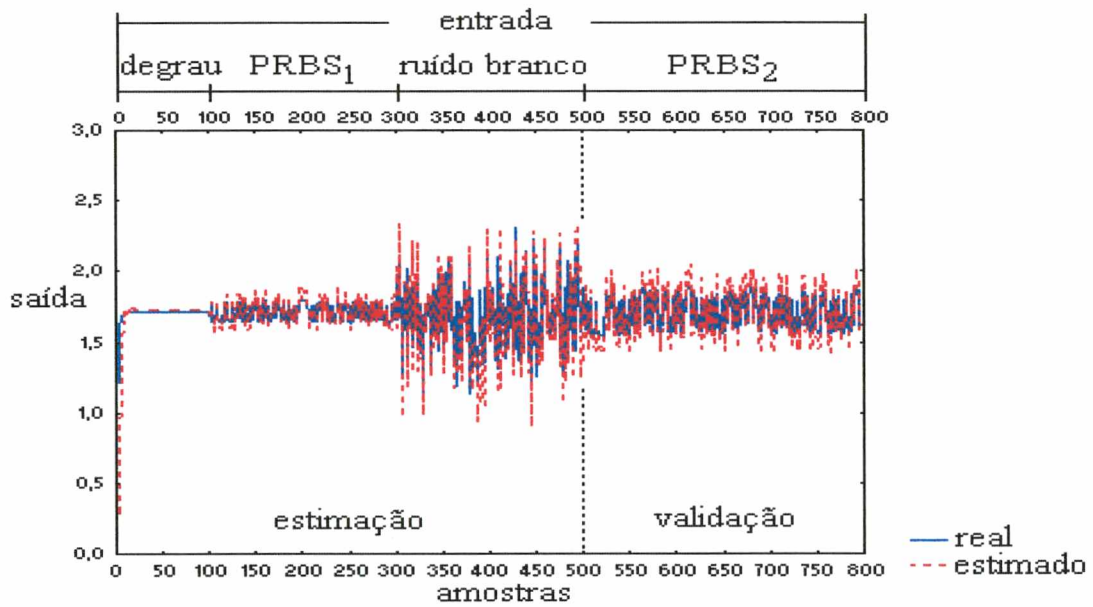


Figura 6.18 - Resposta real e estimada do turbo-gerador em malha aberta através da VICAG (experimento 1).

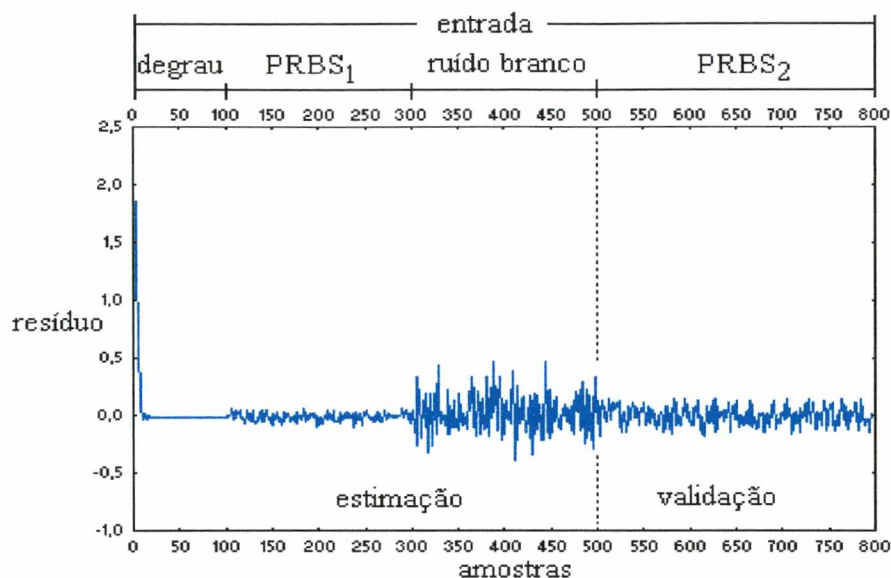


Figura 6.19 - Resíduo do turbo-gerador em malha aberta através da VICAG (experimento 1).

No experimento 2, os algoritmos MQR e VICAG apresentaram como resultado mais adequados, o MQR apresentando índice de desempenho $ISE=195.87$ ($SSE=3.92 \times 10^{-1}$; $RPE=3.27 \times 10^{-1}$) e o algoritmo VICAG com $ISE=133.01$ ($SSE=2.66 \times 10^{-1}$; $RPE=2.22 \times 10^{-1}$). Resultados mais adequados podem ser obtidos empregando-se os algoritmos MQR e VICAG com todos os termos do modelo NARX (56 termos), contudo estes experimentos não são alvo desde trabalho.

As figuras 6.20 e 6.21 apresentam a resposta real e estimada do turbo-gerador em malha aberta através da VICAG e resíduo para o experimento 1, respectivamente.

A tabela 6.9 apresenta os 15 termos relevantes do modelo NARX obtidos pelo algoritmo VICAG na identificação mais adequada efetuada do turbo gerador no experimento 2.

Tabela 6.9 - Termos NARX obtidos pelo algoritmo VICAG (experimento 2).

termo	valor	termo	valor	termo	valor
b_1	0.1667	c_{1311}	-0.1667	d_{3112}	0.1209
b_2	0.0958	c_{2112}	0.1232	d_{3212}	-0.1444
b_3	0.0223	c_{1121}	0.0127	d_{3221}	0.1514
a_1	-0.1341	d_{1111}	0.0824	e_{1221}	0.0960
a_2	0.0929	d_{3311}	-0.0919	cte	0.0953

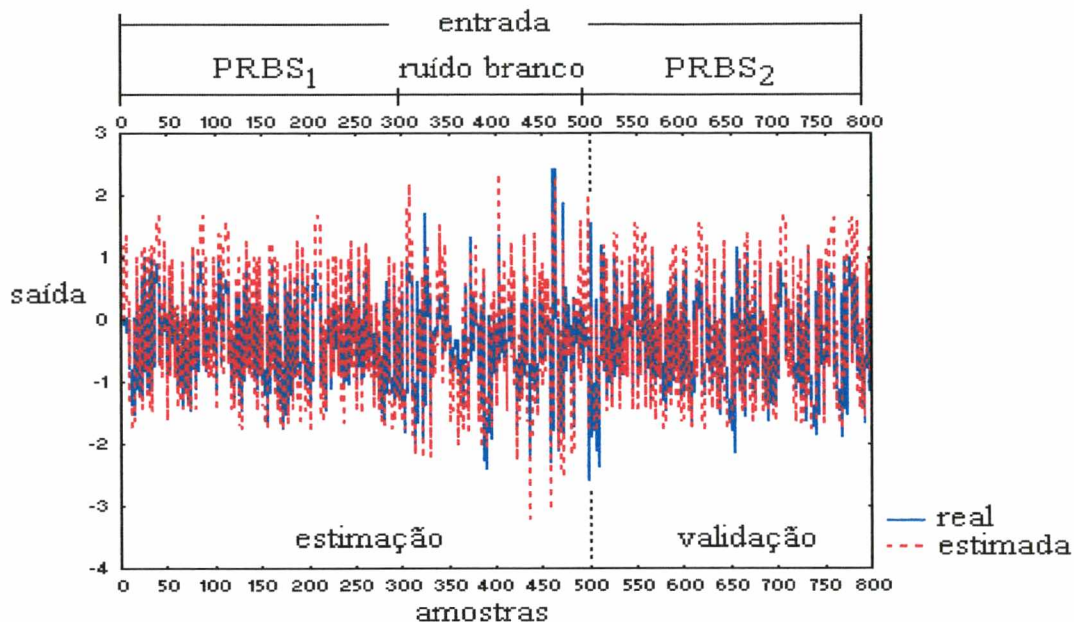


Figura 6.20 - Resposta real e estimada do turbo-gerador em malha aberta através da VICAG (experimento 2).

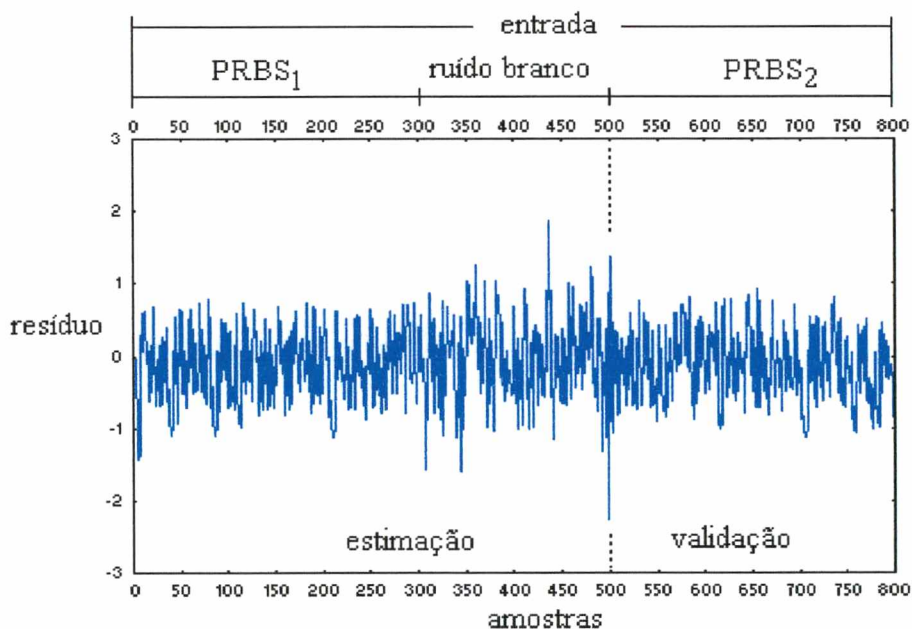


Figura 6.21 - Resíduo do turbo-gerador em malha aberta através da VICAG (experimento 2).

6.3.2.2 Computação evolucionária

As metodologias de CE utilizados na identificação do turbo-gerador são enumerados na tabela 6.2 (mesmos parâmetros de configuração utilizados no processo térmico em AG, AGSA e

EE). Os resultados mais adequados dos 10 experimentos realizados com cada método são explicitados na tabela 6.10.

Tabela 6.10 - Identificação do turbo-gerador por computação evolucionária.

n ^o .	<i>Experimento 1</i>			<i>Experimento 2</i>		
	ISE	SSE	RPE	ISE	SSE	RPE
1	7.4516	1.49x10 ⁻²	5.12x10 ⁻³	112.9320	2.25x10 ⁻¹	1.88x10 ⁻¹
2	6.0692	1.21x10 ⁻²	4.17x10 ⁻³	124.1148	2.48x10 ⁻¹	2.07x10 ⁻¹
3	1.7198	3.40x10⁻³	1.18x10⁻³	110.2344	2.20x10⁻¹	1.84x10⁻¹
4	6.3582	1.27x10 ⁻²	4.37x10 ⁻³	179.1330	3.58x10 ⁻¹	2.99x10 ⁻¹
5	6.4391	1.29x10 ⁻²	4.36x10 ⁻³	132.9201	2.66x10 ⁻¹	2.22x10 ⁻¹
6	6.1666	1.23x10 ⁻²	4.23x10 ⁻³	121.0155	2.42x10 ⁻¹	2.02x10 ⁻¹
7	4.0118	8.00x10 ⁻³	2.76x10 ⁻³	119.1611	2.38x10 ⁻¹	1.99x10 ⁻¹
8	3.0200	6.04x10 ⁻³	2.08x10 ⁻³	113.3532	2.27x10 ⁻¹	1.89x10 ⁻¹

As figuras 6.22 e 6.23 apresentam a resposta real e estimada do turbo-gerador em malha aberta e resíduo obtidos pelo algoritmo AGSA com seleção “breeder” para o experimento 1, enquanto as figuras 6.24 e 6.25 apresentam de forma análoga os resultados para o experimento 2.

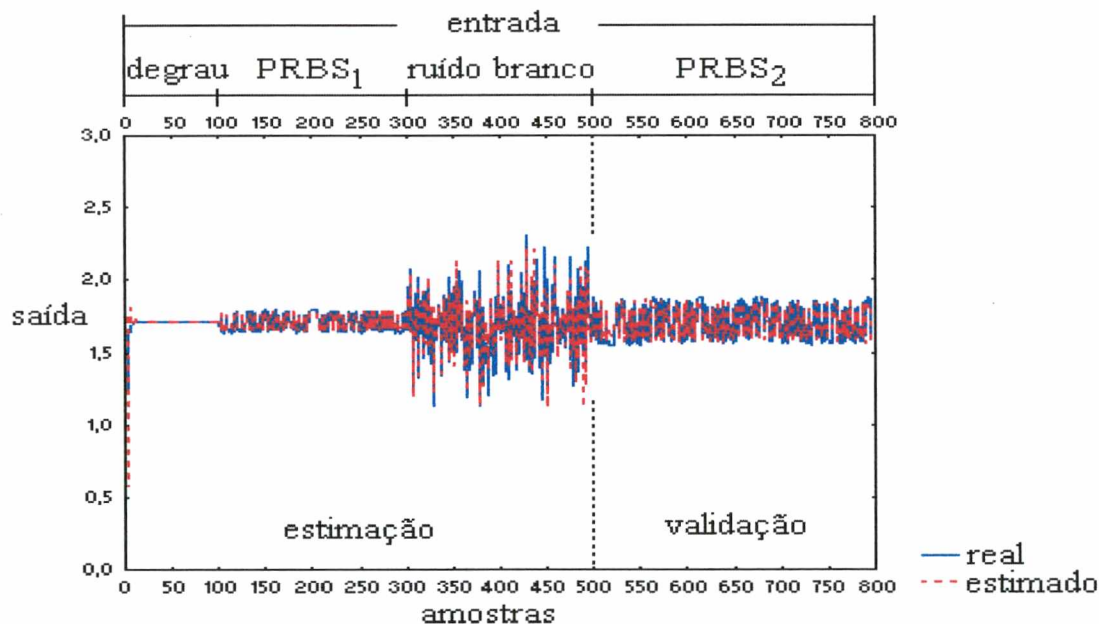


Figura 6.22 - Resposta real e estimada do turbo-gerador em malha aberta via CE (experimento 1).

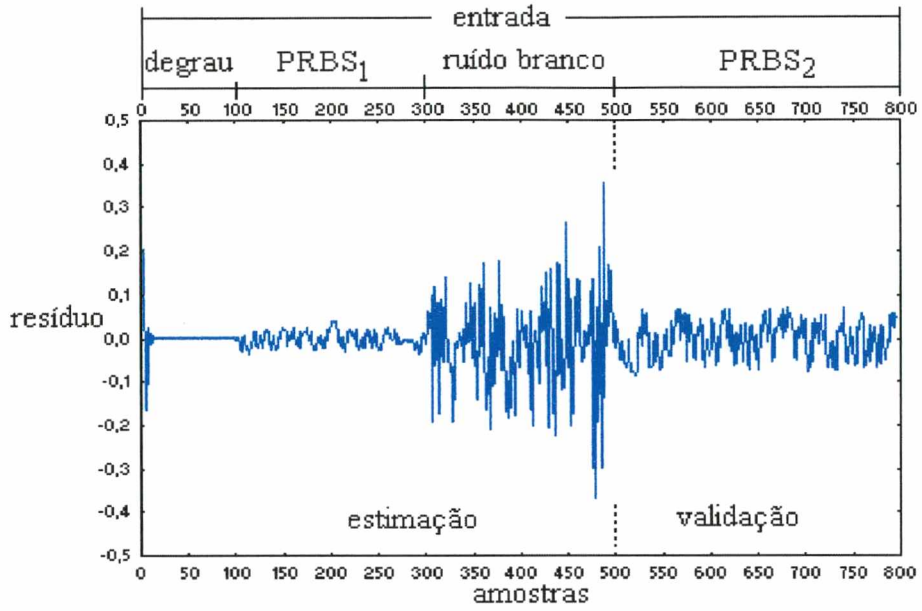


Figura 6.23 - Resíduo do turbo-gerador em malha aberta via CE (experimento 1).

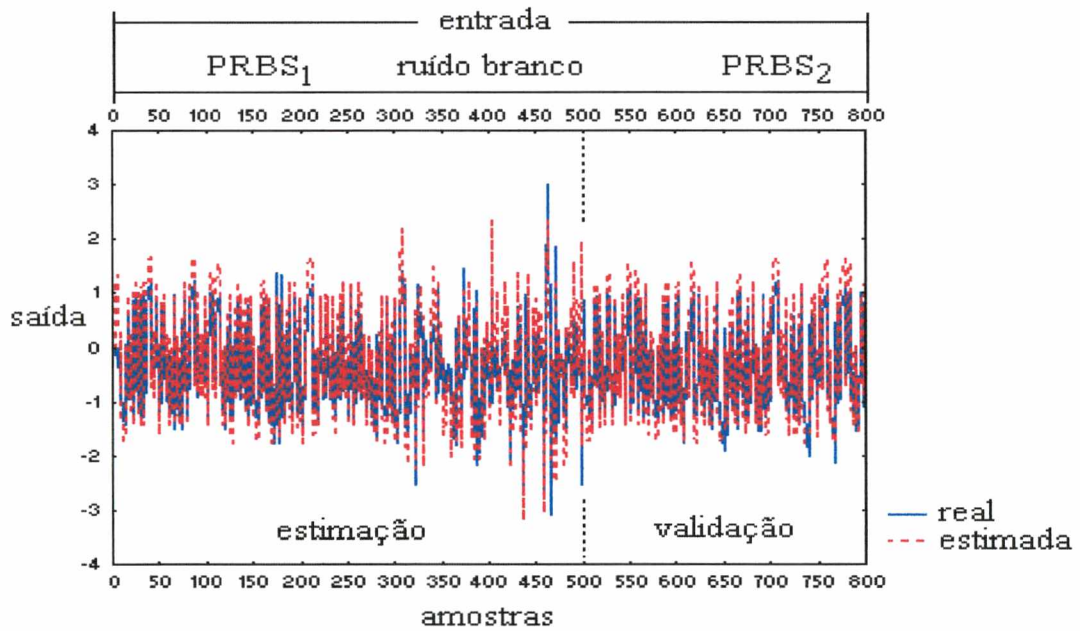


Figura 6.24 - Resposta real e estimada do turbo-gerador em malha aberta via CE (experimento 2).

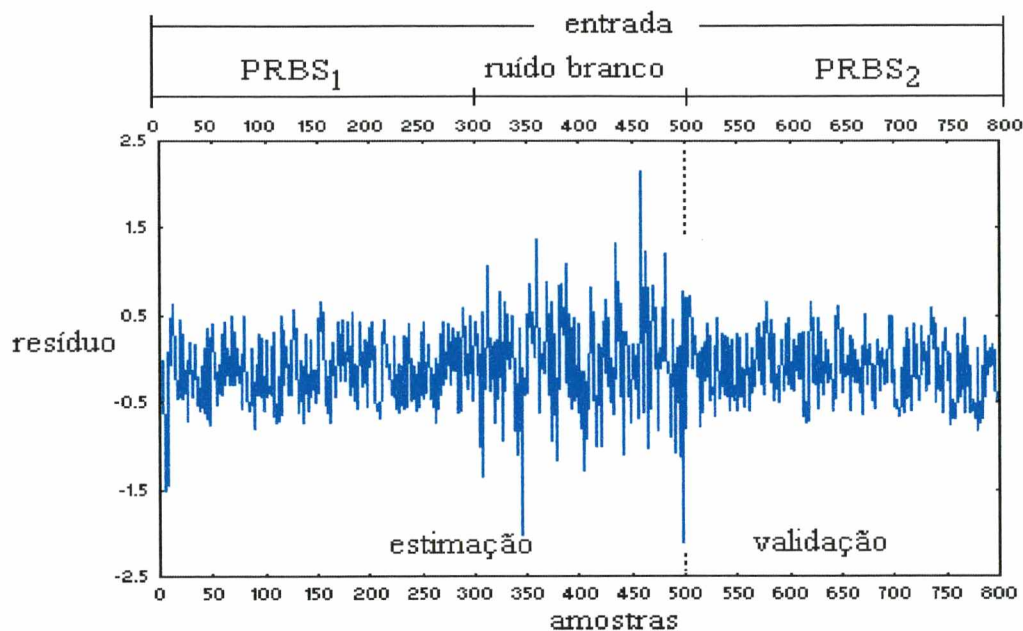


Figura 6.25 - Resíduo do turbo-gerador em malha aberta via CE (experimento 2).

6.3.2.3 Redes neurais artificiais

O algoritmo aplicado na estimação do modelo “black-box” do turbo-gerador é realizado por uma RN “feedforward” multicamadas com treinamento BP e RNPREMOD.

Na identificação dos processos pela RN é estipulado como critério de parada 2000 iterações. O coeficiente de aprendizado e momento são setados para 0.1, enquanto que o parâmetro α da RNPREMOD é ajustado para 0.6. A tabela 6.11 resume os resultados obtidos com a RN “feedforward” e RNPREMOD.

Tabela 6.11 - Identificação do turbo-gerador via RNs.

Experimento 1									
<i>Rede neural "feedforward"</i>									
neurônios	entradas				saída	índice de desempenho			
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE	
2-20-1	X		X		X	3.4045	6.81×10^{-3}	2.34×10^{-3}	
3-20-1	X		X	X	X	4.6512	9.30×10^{-3}	3.19×10^{-3}	
3-20-1	X	X	X		X	4.0925	8.19×10^{-3}	2.81×10^{-3}	
4-20-1	X	X	X	X	X	6.0013	1.20×10^{-2}	4.12×10^{-3}	
<i>Rede neural parcialmente recorrente de Elman modificada</i>									
neurônios	entradas				saída	índice de desempenho			
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE	
2-20-1	X		X		X	2.5921	5.18×10^{-3}	1.78×10^{-3}	
Experimento 2									
<i>Rede neural "feedforward"</i>									
neurônios	entradas				saída	índice de desempenho			
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE	
2-20-1	X		X		X	12.5427	2.51×10^{-2}	2.09×10^{-2}	
3-20-1	X		X	X	X	11.3033	2.26×10^{-2}	1.88×10^{-2}	
3-20-1	X	X	X		X	9.6991	1.94×10^{-2}	1.62×10^{-3}	
4-20-1	X	X	X	X	X	16.4032	3.28×10^{-2}	2.73×10^{-2}	
<i>Rede neural parcialmente recorrente de Elman modificada</i>									
neurônios	entradas				saída	índice de desempenho			
	$y(k-1)$	$y(k-2)$	$u(k-1)$	$u(k-2)$	$y(k)$	ISE	SSE	RPE	
2-20-1	X		X		X	9.6668	1.93×10^{-2}	1.61×10^{-2}	

A resposta real e melhor resposta estimada do turbo-gerador são apresentadas nas figuras 6.26 (experimento 1) e 6.28 (experimento 2). O resíduo da resposta real em relação a resposta estimada é apresentada na figura 6.27 (experimento 1) e 6.30 (experimento 2).

A curva de aprendizado do turbo-gerador via RN é mostrada na figura 6.28 (experimento 1) e 6.31 (experimento 2).

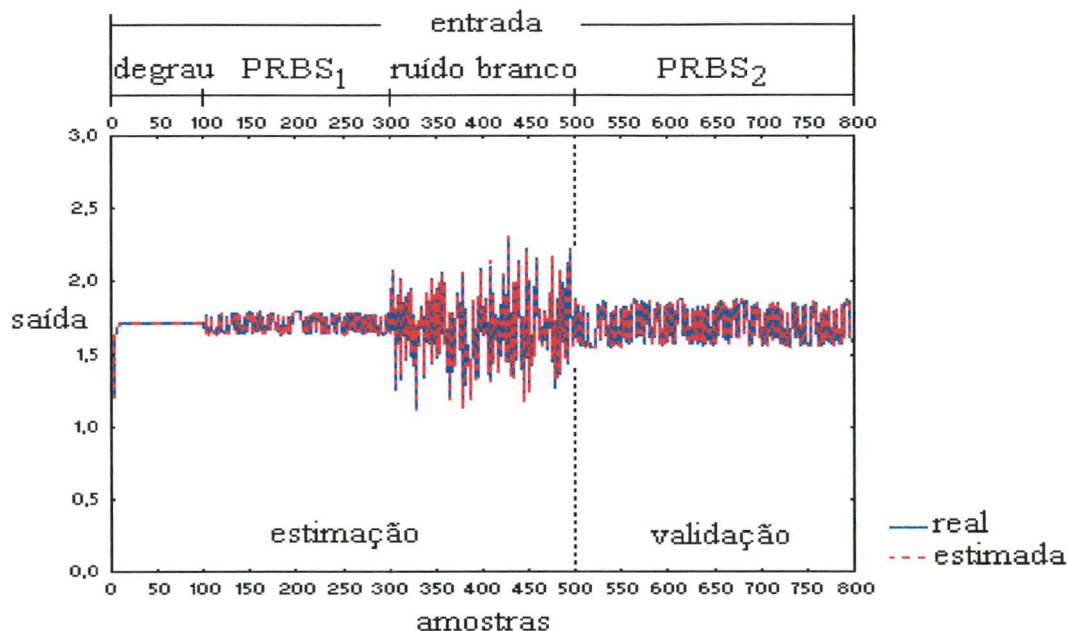


Figura 6.26 - Resposta real e estimada do turbo-gerador em malha aberta via RN (experimento 1).

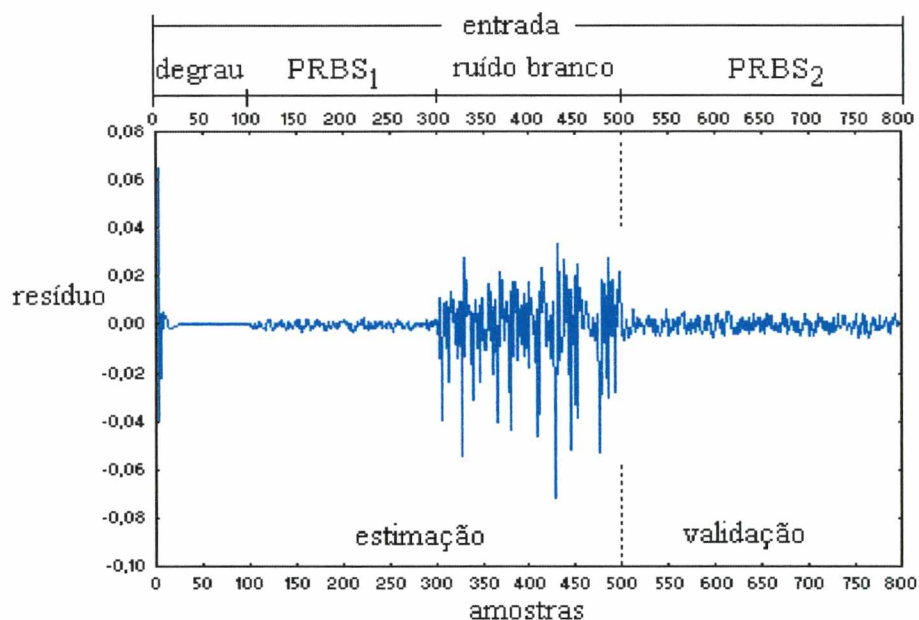


Figura 6.27 - Resíduo do turbo-gerador em malha aberta via RN (experimento 1).

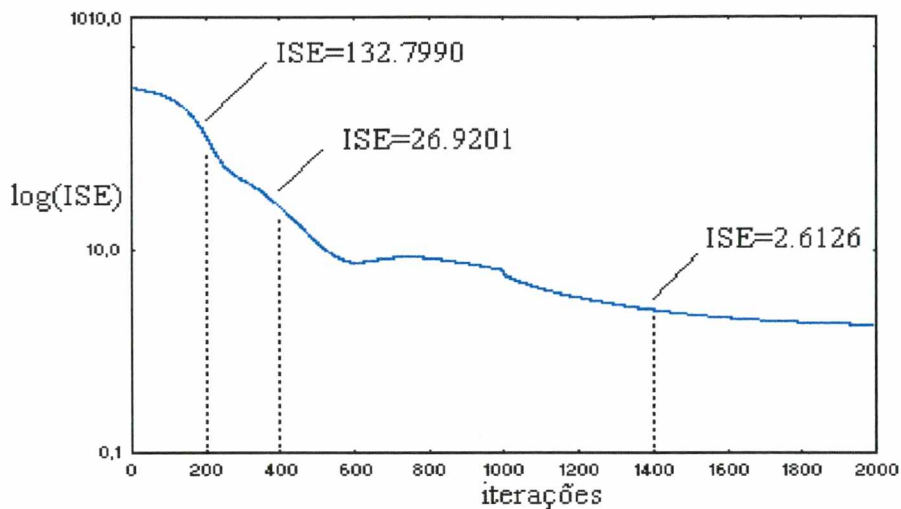


Figura 6.28 - Curva de aprendizado do turbo-gerador via RN (experimento 1).

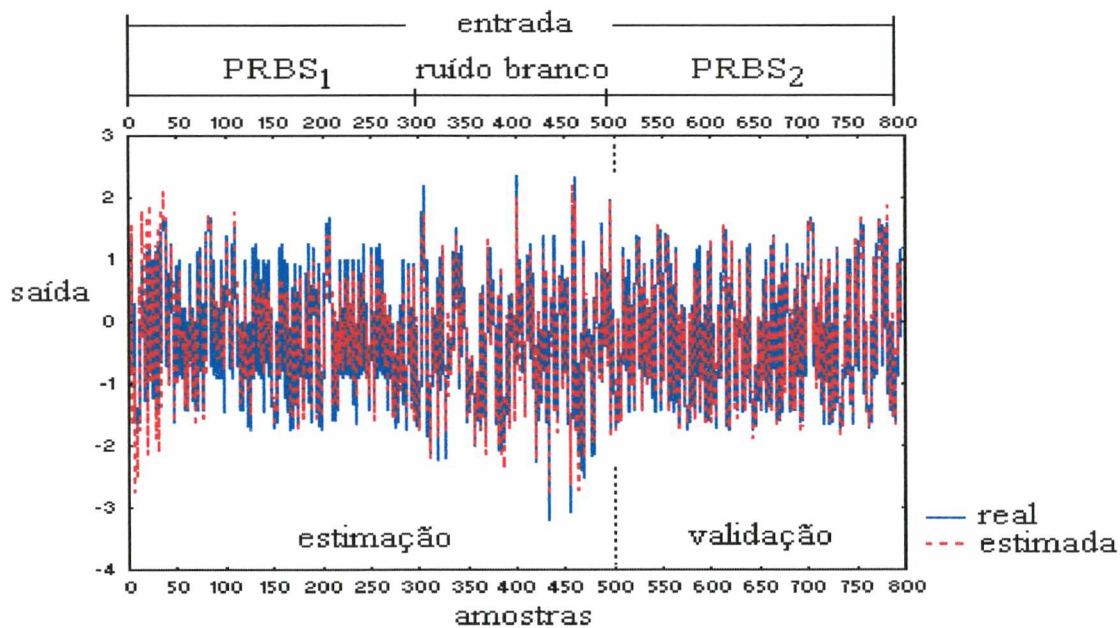


Figura 6.29 - Resposta real e estimada do turbo-gerador em malha aberta via RN (experimento 2).

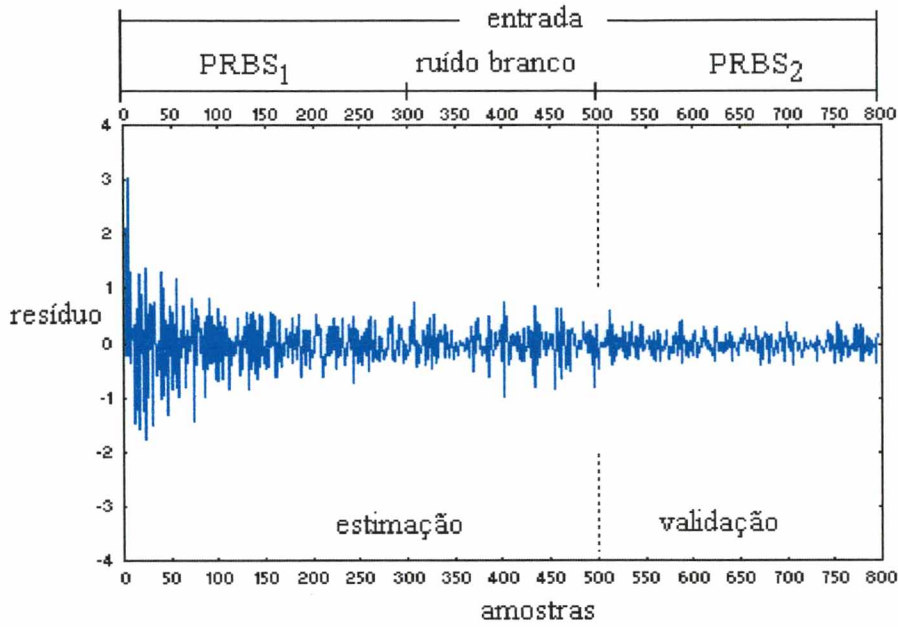


Figura 6.30 - Resíduo do turbo-gerador em malha aberta via RNs (experimento 2).

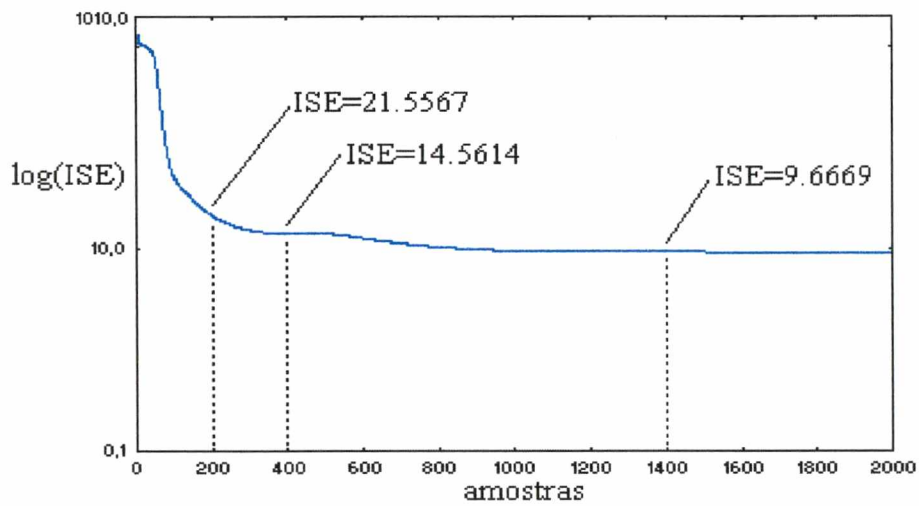


Figura 6.31 - Curva de aprendizado do turbo-gerador via RNs (experimento 2).

6.4 Conclusão

Este capítulo apresentou a descrição dos processos utilizados nos experimentos em identificação de processos. Os processos avaliados foram: processo térmico, processo de nível mono-tanque e modelo simulado de um turbo-gerador.

Os processos utilizados em identificação experimental foram o processo de nível mono-tanque e térmico, e os algoritmos aplicados os MQR e VICAG, além das metodologias

inteligentes da CE (algoritmos genéticos, híbrido de algoritmos genéticos com “simulated annealing” e estratégias de evolução) e RNs “feedforward” e RNPREMOD.

A fase de validação do modelo do turbo-gerador (experimento 1 e 2) com sinais de PRBS e ruído branco, mostra apenas uma faceta dos métodos de validar modelos matemáticos entre uma vasta gama de procedimentos que utilizam critério como AIC (“Akaike information criterion”), análise de correlação e métodos estatísticos (Chen *et al.*, 1990). A obtenção de vários modelos (turbo-gerador), com diferentes pontos de operação e sinais de entrada, possibilita a utilização destes modelos na configuração de controladores multimodelos inteligentes (Al-Akhras *et al.*, 1996).

A modelagem mais adequada considerando-se o índice de desempenho ISE foi obtida com as RNs “feedforward” e RNPREMOD. Obteve-se resultados promissores, contudo deve-se ressaltar a necessidade de análise mais aprofundada de critérios de complexidade algorítmica e convergência das metodologias aplicadas em relação a métodos convencionais de controle e de acordo com as necessidades do projetista, se precisa-se de identificação “on-line” ou “off-line” e se a metodologia escolhida é a mais adequada para tratamento das dificuldades decorrentes das complexidades do processo a ser modelado.

Capítulo 7

Aplicações em controle de processos

7.1 Introdução

Neste capítulo apresenta-se os resultados de simulação e desempenho da aplicação de metodologias da IC e algumas técnicas convencionais em controle de processos. As metodologias utilizadas são CE (AG, AGSA e EE), controlador lógico nebuloso, redes neurais artificiais (“feedforward” e parcialmente recorrente de Elman) e sistemas híbridos de metodologias da IC.

A descrição e as principais características dos processos não-linear de nível duplo-tanque e do sistema dinâmico não-linear de manobrar um caminhão para estacioná-lo em uma garagem, utilizados na avaliação dos algoritmos de controle são abordados.

Os experimentos realizados no processo de nível duplo-tanque, composto de dois tanques acoplados, têm implementação e análise frente a distúrbios de carga e a mudanças de “set-point”.

Quanto ao sistema dinâmico caminhão, objetiva-se estacionar um caminhão utilizando-se apenas a marcha ré, em uma posição pré-determinada (garagem), visando a minimização de uma função custo (equação 7.14), com ênfase à **precisão** em estacionar do caminhão na garagem no experimento 1, enquanto no experimento 2, enfatiza-se a **rapidez** em estacionar o caminhão.

7.2 Descrição dos processos

7.2.1 Sistema de regulação de nível com dois tanques acoplados

O sistema de regulação de nível não-linear é formado por dois tanques acoplados e está ilustrado na figura 7.1 (Coelho e Coelho, 1996f; Li *et al.*, 1995). As equações que caracterizam a dinâmica em malha aberta são

$$A_1 \dot{h}_1 = u - a_1 c_1 \sqrt{2g(h_1 - h_2)} \quad (7.1)$$

$$A_2 \dot{h}_2 = a_1 c_1 \sqrt{2g(h_1 - h_2)} - a_2 c_2 \sqrt{2g(h_2 - h_0)} \quad (7.2)$$

$$y = h_2(t - \tau) \quad (7.3)$$

onde a área dos tanques é $A_1=A_2=97 \text{ cm}^2$; a área dos orifícios é $a_1=0.396 \text{ cm}^2$ e $a_2=0.395 \text{ cm}^2$; as constantes de descarga são $c_1=0.53$ e $c_2=0.63$; a altura dos orifícios é $h_0=3 \text{ cm}$, e a aceleração da gravidade é $g=981 \text{ cm/s}^2$. O atraso de transporte “ τ ” inserido no sistema é de 6 unidades de tempo.

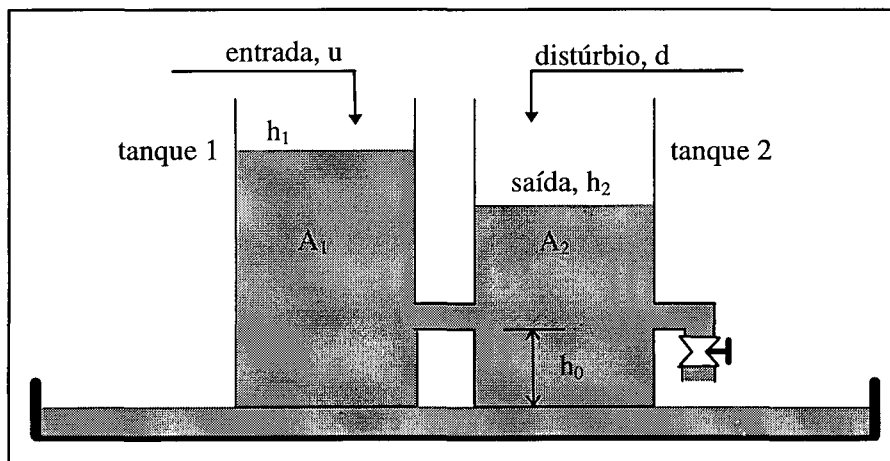


Figura 7.1 - Sistema de regulação de nível não-linear composto de dois tanques acoplados.

7.2.2 Sistema dinâmico não-linear para a manobra de um caminhão até a garagem

A figura 7.2 descreve o sistema caminhão (com cabine e carroceria), onde o objetivo da ação de controle “ u ” (dada por um ângulo) é estacionar o caminhão na posição $x=0$, $y=0$ e $\theta_s=0$, dada uma condição inicial de (x,y,θ_s, θ_c) , podendo-se apenas utilizar a marcha ré, observando-se que a cada instante de tempo o caminhão desloca-se 3 metros, dado pelo parâmetro “ r ”.

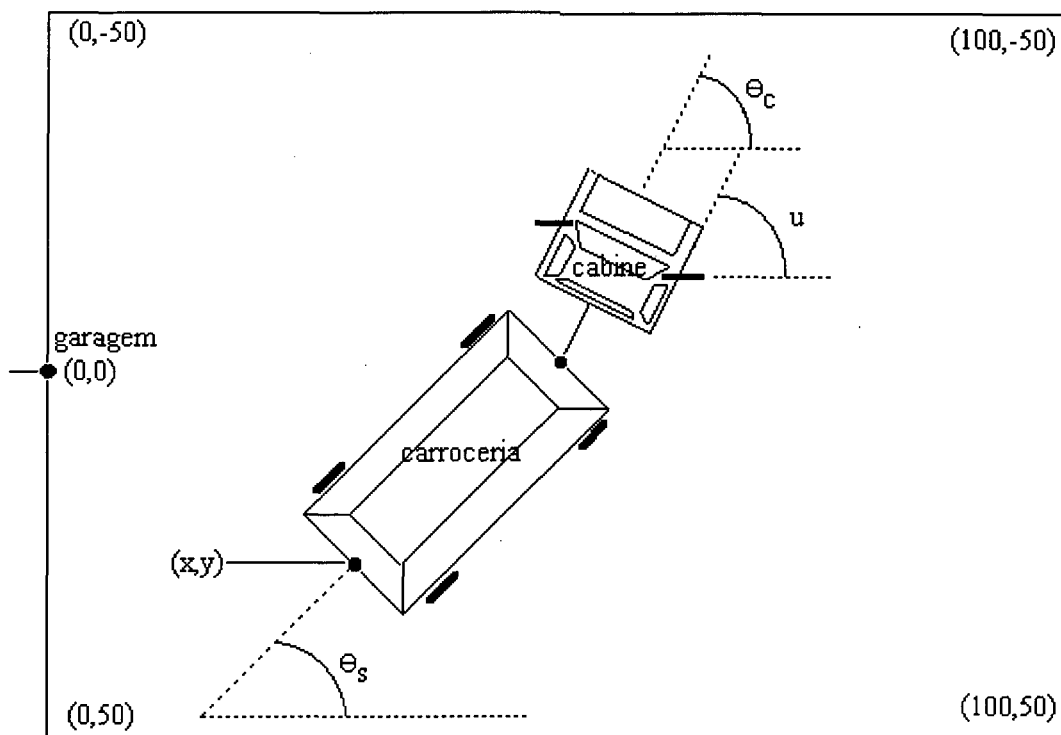


Figura 7.2 - Descrição do sistema caminhão com carroceria.

Os estados que regem o sistema do caminhão são definidos como segue (Nguyen and Widrow, 1990; Schoenauer and Ronald, 1994):

- x,y : coordenadas do centro da traseira do caminhão;
- θ_s : ângulo da carroceria com o eixo x ;
- θ_c : ângulo da cabine com o eixo x ;
- u : ângulo de direção das rodas da frente em relação do eixo da cabine.

As restrições impostas são: $x > 0$ (a garagem está em $x = 0$), o ângulo entre a cabine e a carroceria $|\theta_s - \theta_c| \leq 1.5707$ rad (ou $|\theta_s - \theta_c| \leq 90^\circ$) e o limite de direção das rodas da frente do caminhão é de -1.2217 rad $\leq u \leq 1.2217$ rad (ou $-70^\circ \leq u \leq 70^\circ$).

As equações que regem o movimento do caminhão são:

$$A = r \cdot \cos(u) \quad (7.4)$$

$$B = A \cdot \cos(\theta_c(t) - \theta_s(t)) \quad (7.5)$$

$$x(t+1) = x(t) - B \cdot \cos(\theta_s(t)) \quad (7.6)$$

$$y(t+1) = y(t) - B \cdot \text{sen}(\theta_s(t)) \quad (7.7)$$

$$\theta_s(t+1) = \theta_s(t) - \arcsen((A \cdot \text{sen}(\theta_c(t) - \theta_s(t))) / L_s) \quad (7.8)$$

$$\theta_c(t+1) = \theta_c(t) + \arcsen(r \cdot \text{sen}(u) / (L_s + L_c)) \quad (7.9)$$

Observa-se que após a aplicação da equação (7.9), o valor do ângulo $\theta_c(t+1)$ é então ajustado para obedecer a restrição $|\theta_s - \theta_c| \leq 1.5707$ rad. Os parâmetros estipulados para o caso estudado são (Schoenauer and Ronald, 1994):

- a distância que movem-se as rodas da frente a cada passo de tempo é de $r = 3$ metros;
- o comprimento do reboque até a articulação é de $L_s = 14$ metros; e
- o comprimento da cabine da articulação ao eixo da frente é de $L_c = 6$ metros.

7.3 Controle de processos

As principais características e avaliação do desempenho dos algoritmos que utilizam-se da IC para o tratamento do sistema de regulação de nível com dois tanques acoplados e o sistema de estacionamento de um caminhão em uma garagem são tratados a seguir. Para melhor atender e auxiliar a análise do desempenho dos algoritmos da IC frente aos processos estudados utiliza-se as equações da variância do erro (ve) e variância do controle (vu), calculadas por:

$$ve = \frac{1}{N} \sum_{i=1}^N [\text{referência}(t) - y(t)]^2 \quad (7.10)$$

$$vu = \frac{1}{N} \sum_{i=1}^N [u(t) - u_{\text{média}}]^2 \quad (7.11)$$

onde N é o número da amostras e $u_{\text{média}}$ é a média dos valores da ação de controle.

7.3.1 Controle do sistema de regulação de nível com dois tanques acoplados

Os experimentos em controle de processos realizados no sistema não-linear de regulação de nível visam o projeto e a configuração dos algoritmos de controle pelas seguintes metodologias em controle inteligente:

- CE (algoritmo genético, algoritmo genético híbrido com “simulated annealing” e estratégias de evolução) em sintonia convencional do controle PID (PID-CE);
- CLN com otimização das funções de pertinência por estratégias de evolução (PID-CLN);
- RN com variação dinâmica dos coeficientes de aprendizado (PID-RN).
- controlador neural com treinamento por CE (RN-CE); e
- controlador linear invariante no tempo de 3ª. ordem com sintonia por CE (CLIT-CE).

Os testes experimentais são conduzidos para analisar a habilidade das técnicas em controle do processo, sendo o processo sujeito a mudanças de referência, na *fase de sintonia* dos controladores e sujeito a distúrbios de carga, na *fase de testes* de robustez dos algoritmos de controle. A função custo $J(u)$ a ser minimizada é a equação (2.34), com “w” estipulado em 0.1 (Coelho e Coelho, 1996f).

Os ensaios realizados no sistema não-linear de regulação de nível visam manter o nível de líquido do tanque 2 durante 600 iterações, em cada uma das três referências propostas de 9 cm, 12 cm e 5 cm, respectivamente, com um mínimo de sobre-elevação e erro em regime nulo.

Quanto ao projeto dos controladores com configuração inteligente, cabe enfatizar, que os controladores são configurados e otimizados apenas para as mudanças de referência, *não prevendo distúrbios de carga*.

O sinal do controle de entrada do líquido no tanque 1 é restringido a faixa de $[0;33.3 \text{ cm}^3/\text{s}]$ e obtém-se a saída da planta calculada, usando-se o método de Runge-Kutta de 4ª. ordem (Coelho e Coelho, 1996f; Li *et al.*, 1995).

A figura 7.3 apresenta a configuração do controlador PID com sintonia por CE. Na tabela 7.1 são mencionados os algoritmos de CE utilizados em sintonia convencional do controle PID.

A convergência das técnicas de CE pode ser visualizada pela figura 7.4, apresentando a comparação da função custo $J(u)$ nos 10 experimentos (média) na sintonização do PID, com um total de 20 gerações (iteraões) e a otimização dos três parâmetros (K_p , T_i e T_d) limitada a faixa de $[0;100]$.

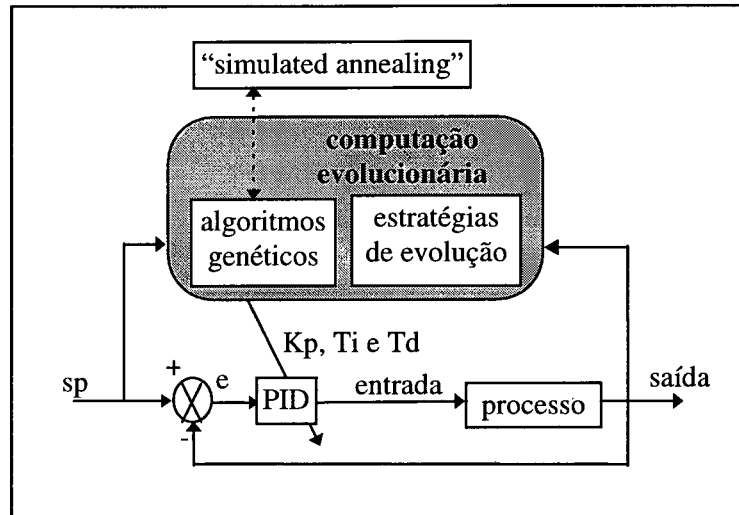


Figura 7.3 - Configuração do controlador PID com sintonia por CE.

Tabela 7.1 - Paradigmas da CE utilizados na sintonia de controle inteligente.

nº.	Técnica (população)	σ	seleção	recombinação	mutação
1	AG(30)	-	roleta	* $p_r = 0.8$	# $p_m = 0.1$
2	AGSA(30)	-	roleta	* $p_r = 0.8$	# $p_m = 0.1$
3	AGSA(30)	-	“breeder”	* $p_r = 0.8$	# $p_m = 0.1$
4	EE-(1+29)***	-	-	-	correlata
5	EE-(1+29)****	-	-	-	correlata
6	EE-(5,25)	1	-	**	não-uniforme

convenções

- # mutação não-uniforme com $b=2$
- * recombinação aritmética
- ** recombinação discreta (x) e aritmética intermediária (σ)
- *** controle do tamanho de passo individual
- **** variações correlatas e esquema de adaptação dos parâmetros

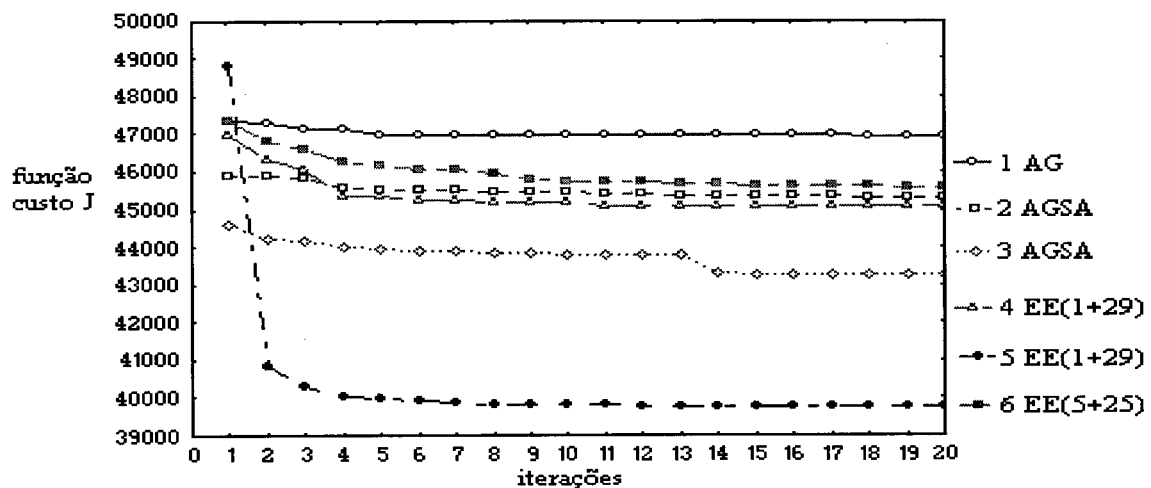


Figura 7.4 - Comparação da função custo nos 10 experimentos (média) para sintonização do PID via CE.

O melhor resultado obtido pelo PID-CE, analisando-se AG, AGSA e EE, é dado pela metodologia EE-(1+29) com variações correlatas e esquema de adaptação de parâmetros. Os parâmetros de sintonia obtidos são $K_p=16.68$, $T_i=0.32$ e $T_d=0.02$, e $J(u)=127104$.

O PID-RN teve implementação com uma arquitetura de 3 neurônios na camada de entrada, 1 camada oculta com 7 neurônios e 1 neurônio na camada de saída do PID-RN.

O parâmetros de aprendizado e momento são estipulados inicialmente em 1×10^{-4} , e os parâmetros do AAH são $\phi=1.05$, $\beta=0.5$ e $\epsilon=0.03$. O PID-RN é sintonizado em 400 iterações, obtendo-se $J(u)=130085$, contudo deve ser mencionado que a função custo no caso específico deste controlador não serve para correção dos pesos do PID-RN, mas sim, o erro da saída do processo em relação a referência desejada.

O projeto PID-CLN utilizado é o da implementação direta, descrito no capítulo 4 (item 4.3.3). O controlador PID-CLN tem a otimização dos parâmetros das funções de pertinência efetuada em 400 iterações. Os parâmetros otimizados com uma população de 30 indivíduos é realizado por EE-(1+29) com mutações correlatas e esquema de adaptação de parâmetros, sendo que obteve-se o valor de $J(u)=132679$. A base de regras estipulada para o controlador PID-CLN é apresentada na tabela 7.2 (Boverie *et al.*, 1991; Coelho e Coelho,1996d, Lee, 1990; Li and Gatland, 1995).

Tabela 7.2 - Base de regras estipulada para o controlador PID-CLN.

parte proporcional e derivativa					
<i>f.p.Δe</i>					
<i>e \ Δe</i>	<i>NG</i>	<i>NP</i>	<i>ZE</i>	<i>PP</i>	<i>PG</i>
<i>NG</i>	NG	NG	NP	ZE	ZE
<i>NP</i>	NG	NP	NP	ZE	PP
<i>ZE</i>	NP	NP	ZE	PP	PG
<i>PP</i>	ZE	ZE	PP	PP	PG
<i>PG</i>	ZE	PP	PG	PG	PG

parte integral (proporcional ao erro)					
<i>e</i>	<i>NG</i>	<i>NP</i>	<i>ZE</i>	<i>PP</i>	<i>PG</i>
<i>f.p.e</i>	PG	PP	ZE	NP	NG

convenções	
<i>f.p.e</i>	funções de pertinência do erro
<i>f.p.Δe</i>	funções de pertinência da variação do erro
NG	Negativo Grande
NP	Negativo Pequeno
ZE	Zero
PP	Positivo Pequeno
PG	Positivo Grande

Quanto aos algoritmos CLIT-CE e RN-CE obteve-se como resultado mais adequado para $J(u)=131948$ e $J(u)=136470$, respectivamente. O resumo dos resultados obtidos é esboçado na tabela 7.3.

Os resultados da fase de sintonia dos controladores inteligentes abordados são empregados no teste dos controladores sujeitos a mudanças de referência e distúrbios de carga, onde obtém-se uma adequada noção do desempenho e robustez das técnicas para o projeto de controladores inteligentes.

Tabela 7.3 - Resultados da fase de sintonia no controle do processo de nível duplo-acoplado.

resultados																	
<i>PID-CE</i>																	
n°	K _p	T _i	T _d	ts ₁	tp ₁	o ₁ %	Δu ₁	ts ₂	tp ₂	o ₂ %	Δu ₂	ts ₃	tp ₃	o ₃ %	Δu ₃	vu	J(u)
7	16.68	0.32	0.02	150	162	0.52	114.4	309	323	0.12	4.17	152	164	0.77	109.4	85.4	127104
<i>PID-CLN</i>																	
n°				ts ₁	tp ₁	o ₁ %	Δu ₁	ts ₂	tp ₂	o ₂ %	Δu ₂	ts ₃	tp ₃	o ₃ %	Δu ₃	vu	J(u)
7				154	190	0.39	110.9	317	355	0.31	4.79	117	145	0.51	104.4	81.7	132679
<i>PID-RN</i>																	
	n ₀	n ₁	n ₂	ts ₁	tp ₁	o ₁ %	Δu ₁	ts ₂	tp ₂	o ₂ %	Δu ₂	ts ₃	tp ₃	o ₃ %	Δu ₃	vu	J(u)
	3	3	1	148	171	0.98	111.8	310	334	0.29	4.50	101	116	1.28	102.1	102.1	130085
<i>CLIT-CE</i>																	
n°				ts ₁	tp ₁	o ₁ %	Δu ₁	ts ₂	tp ₂	o ₂ %	Δu ₂	ts ₃	tp ₃	o ₃ %	Δu ₃	vu	J(u)
3				189	200	0.38	117.1	358	370	0.07	5.17	168	173	0.08	102.3	84.2	131948
<i>RN-CE (implementado com o RNPREMOD)</i>																	
n°	n ₀	n ₁	n ₂	ts ₁	tp ₁	o ₁ %	Δu ₁	ts ₂	tp ₂	o ₂ %	Δu ₂	ts ₃	tp ₃	o ₃ %	Δu ₃	vu	J(u)
7	3	3	1	147	157	0.20	113	312	317	0.04	4.98	98	102	0.94	136.0	86.0	136470
convencões																	
n°	metodologia da CE utilizada na otimização (conforme a tabela 7.1)																
n _i , {i ∈ ℕ / i = (1,2,3)}	i=0 → número de neurônios da camada de entrada i=1 → número de neurônios da camada oculta i=2 → número de neurônios da camada de saída																
ts _k	tempo de subida																
tp _k	tempo de pico																
o _k %	porcentagem de sobre-elevação (“overshoot”)																
Δu _k	variação do sinal de controle																
k ∈ ℕ / k = (1,2,3)}	k=1 (0 a 600 amostras); k=2 (600 a 1200 amostras); k=3 (1200 a 1800 amostras)																
vu	variância do sinal de controle u																
J(u)	valor da função custo																

As figuras de 7.5 a 7.9 ilustram a resposta em malha fechada dos controladores PID inteligentes, PID-CE, PID-CLN, PID-RN, CLIT-CE e RN-CE ao final da *fase de sintonia*.

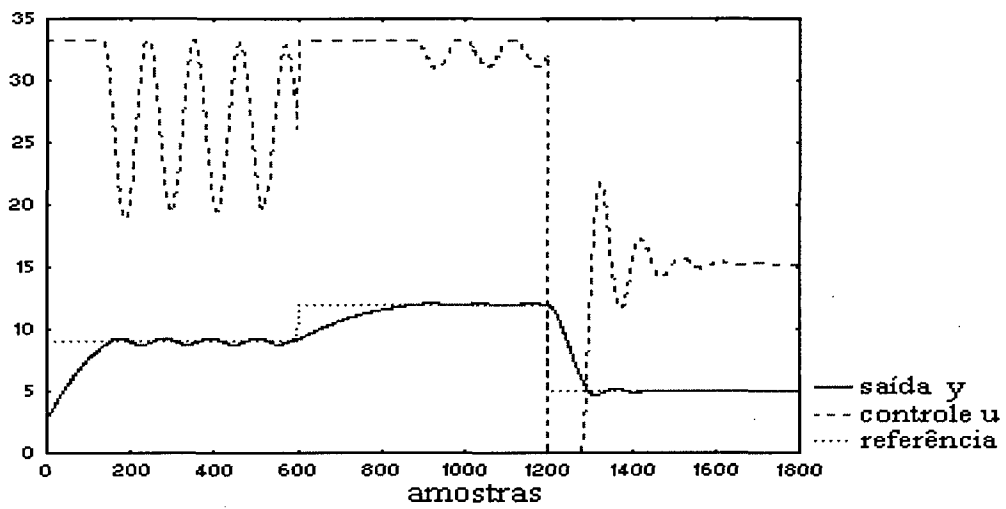


Figura 7.5 - Saída e controle para mudanças de referência do PID-CE.

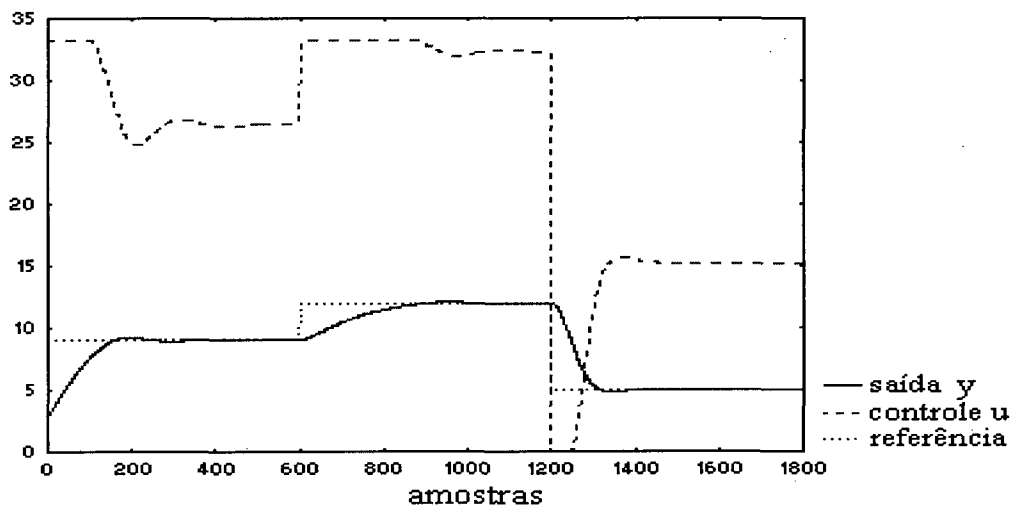


Figura 7.6 - Saída e controle para mudanças de referência do PID-CLN.

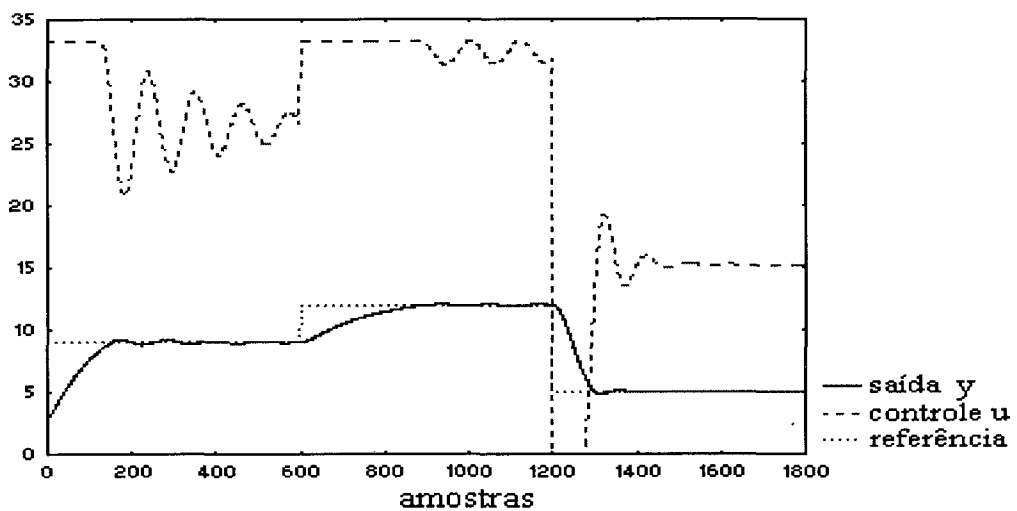


Figura 7.7 - Saída e controle para mudanças de referência do PID-RN.

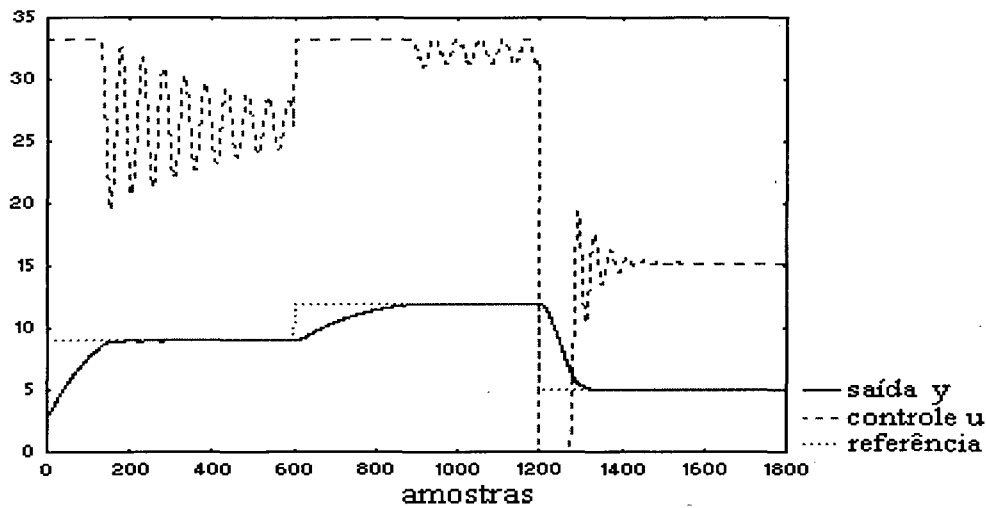


Figura 7.8 - Saída e controle para mudanças de referência do CLIT-CE.

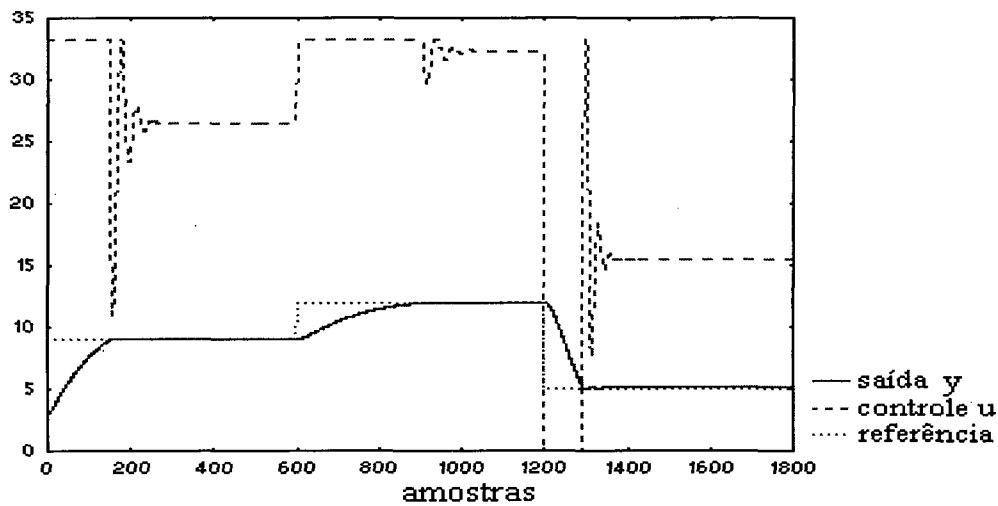


Figura 7.9 - Saída e controle para mudanças de referência do RN-CE.

Os distúrbios de carga em cerca de 25% do controle máximo ($8.325 \text{ cm}^3/\text{s}$) são aplicados no tanque 2, nos instantes 300 a 600, 900 a 1200 e 1500 a 1800, após o projeto dos controladores (fase de sintonia), para análise da robustez, sensibilidade e atuação dos controladores inteligentes frente a situações imprevistas no projeto do controlador.

As figuras de 7.10 a 7.14 ilustram a resposta em malha fechada dos controladores PID inteligentes, PID-CE, PID-CLN, PID-RN, CLIT-CE e RN-CE sujeitos a mudanças de referência e distúrbios de carga (*fase de testes*).

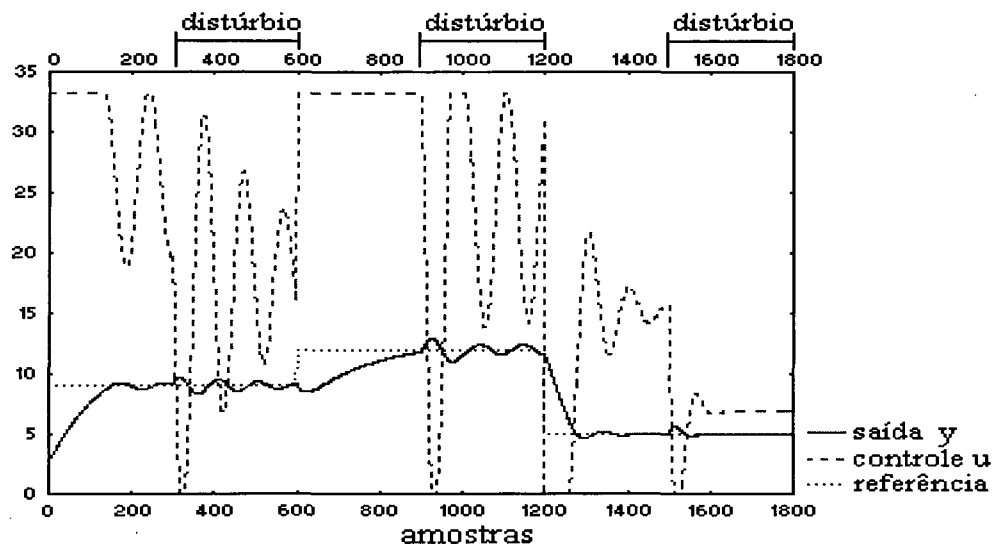


Figura 7.10 - Saída e controle para mudanças de referência e distúrbios de carga do PID-CE.

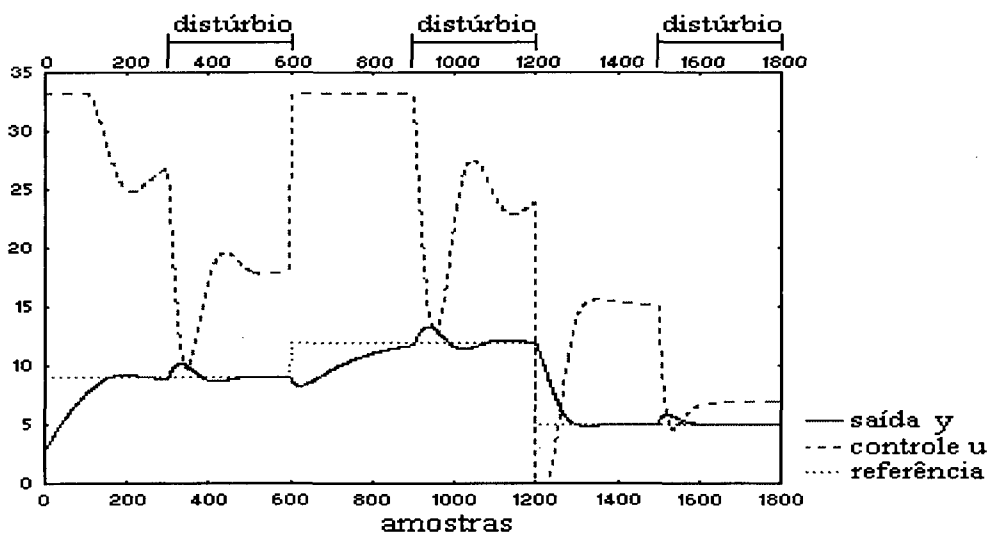


Figura 7.11 - Saída e controle para mudanças de referência e distúrbios de carga do PID-CLN.

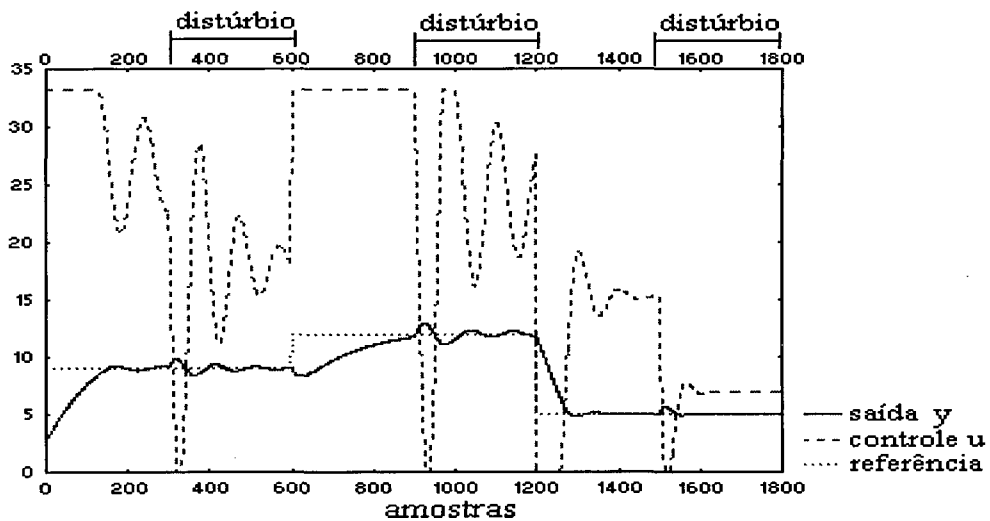


Figura 7.12 - Saída e controle para mudanças de referência e distúrbios de carga do PID-RN.

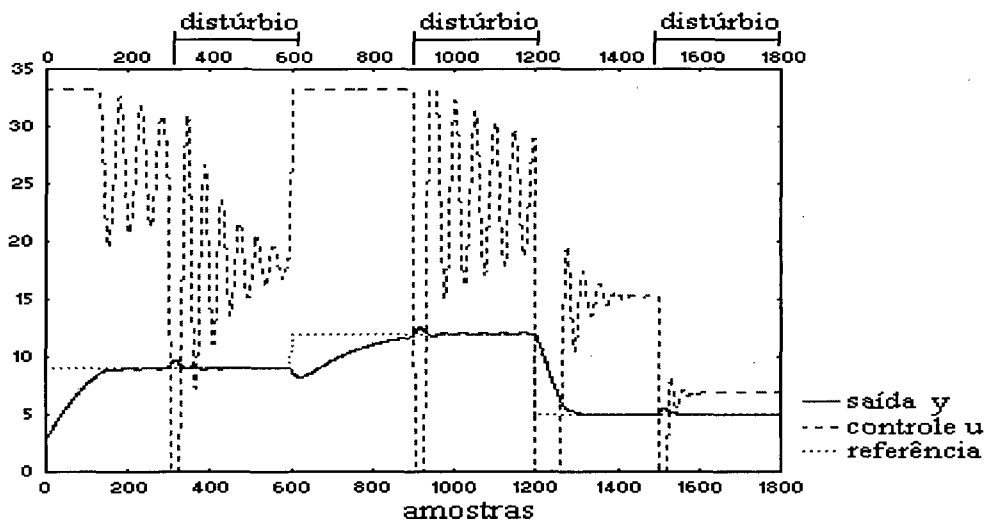


Figura 7.13 - Saída e controle para mudanças de referência e distúrbios de carga do CLIT-CE.

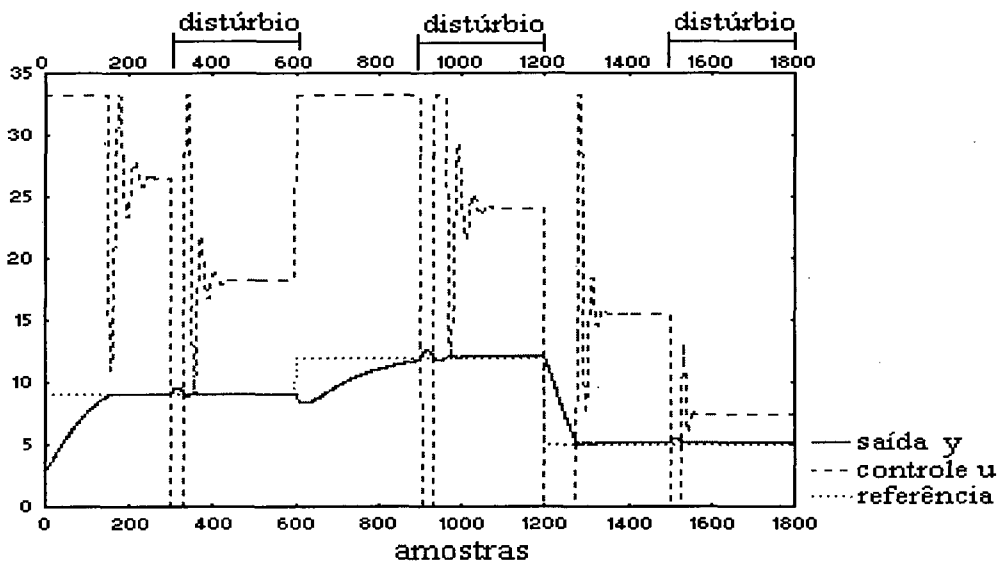


Figura 7.14 - Saída e controle para mudanças de referência e distúrbios de carga do RN-CE.

Nota-se pela resposta a mudanças de referência dos controladores inteligentes apresentadas nas figuras 7.5 a 7.9 que os controladores utilizados são adequados para tratamento de mudanças de referência, contudo cabe mencionar que a resposta destes controladores, em aplicações práticas, pode ser aprimorada evitando-se a saturação da ação de controle, alterando-se a equação (2.34) de $J(u)$ de maneira a acrescentar-se algum termo de penalização a ação de controle.

Pela tabela 7.4, apresentada a seguir, nota-se que os controladores inteligentes estudados comportam-se com desempenho adequado frente ao efeito das mudanças na dinâmica da planta

com um acréscimo na variância da ação de controle para o tratamento desta situação, pois são sintonizados apenas para tratar mudanças de referência.

Tabela 7.4 - Resultados do controle do processo de nível duplo-acoplado.

algoritmo	fase de sintonia		fase de testes	
	vu	J(u)	vu	J(u)
PID-CE	85.4	127104	128.4	200150
PID-CLN	81.7	132679	99.85	219996
PID-RN	84.8	130085	119.4	193443
CLIT-CE	84.2	131948	118.5	195688
RN-CE	86.0	136470	118.9	230454

A partir dos resultados experimentais, verifica-se que os controladores inteligentes apresentados são adequados para o tratamento do processo de nível contido cada um deles tem suas peculiaridades e características que devem ser avaliadas na escolha do controlador mais apropriado no contexto de controle de processos. Outros experimentos que podem ser realizados, futuramente, são a configuração dos controladores inteligentes adotados, neste trabalho, prevendo tanto para mudanças de referência quanto para distúrbios de carga, em vez de duas fases, uma de sintonia e outra de testes, conforme adotado.

O controle PID-CE é o que apresenta a minimização mais adequada de $J(u)$ no projeto do controlador, contudo pelos resultados apresentados no teste de resposta em malha fechada (conforme apresentado nas figuras 7.5 e 7.10 e tabela 7.4) nota-se que o controle PID-CE mostra acentuada flutuação na saída do processo, o que pode tornar o controlador inadequado para plantas que exijam requisitos de precisão no projeto do controlador e erro mínimo na saída do tanque 2.

O controle PID-CLN (ver figura 7.6 e 7.11) apresenta performance próxima ao controle PID-RN (ver figura 7.7 e 7.12), contudo o PID-RN apesar de apresentar função custo com valor maior que o PID-CE (fase de sintonia), seu desempenho teve resposta em malha fechada (fase de testes) com minimização mais apropriada de $J(u)$ que os outros controladores analisados, acompanhando adequadamente a saída do processo. Quanto ao controle CLIT-CE (conforme as figuras 7.8 e 7.13) e RN-CE (de acordo com as figuras 7.9 e 7.14) apresentaram $J(u)$ com valores significativos, contudo suas respostas para mudanças de referência e a distúrbios de carga podem ser aprimoradas, em trabalhos futuros, com a adoção de configurações de projeto com maior

número de parâmetros a serem otimizados e incremento no número de iterações de otimização por CE (tabela 7.3).

A tabela 7.5 aborda o sumário da comparação entre as técnicas de configuração de controle inteligente utilizadas no controle do processo de regulação não-linear de nível (Coelho e Coelho, 1996e, 1996f).

Tabela 7.5 - Comparação entre as técnicas de controle inteligentes.

Critério	PID-CE	PID-RN	PID-CLN	CLIT-CE	RN-CE
<i>estrutura do controlador</i>	linear	não-linear	não-linear	linear	não-linear
<i>sintonia dos parâmetros</i>	K_p, T_i, T_d	neurônios	funções de pertinência / regras	polinômios A e B	neurônios e da CE
<i>modelo matemático</i>	do controlador	não-necessário	não-necessário	do controlador	não-necessário

7.3.2 Controle de um caminhão para estacionamento em uma garagem

No sistema de controle de um caminhão são realizados dois tipos de experimentos para análise do desempenho dos controladores e a observação de relevância (ou não) das informações sobre x, y, θ_s e θ_c . Os experimentos são para três casos: duas entradas (x, y), três (x, y, θ_s) e quatro entradas (x, y, θ_s, θ_c) para os controladores de modo a efetivar-se a posteriori uma análise dos resultados obtidos.

Dados importantes para esta análise são o comprimento da trajetória do caminhão ($traj$), o erro de trajetória (e_{traj}) e a função custo adotada (J_c) a ser minimizada, esta proposta por Schoenauer and Ronald (1994). As equações que regem este dados são as seguintes:

$$traj = \sum_{k=1}^N \sqrt{(x(k) - x(k-1))^2 + (y(k) - y(k-1))^2} \quad (7.12)$$

$$erro_{traj} = \frac{traj}{\text{distância da posicao inicial a posicao desejada}(0, \theta, 0)} \quad (7.13)$$

$$J_c = x^2 + y^2 + \text{mínimo}(\theta_s^2, (\theta_s - 2\pi)^2, (\theta_s + 2\pi)^2) + \gamma * np \quad (7.14)$$

onde x e y são dados em metros (m) e, θ_s é avaliado em radianos (rad). O parâmetro “np” representa o número de passos (iterações) necessários para estacionar o caminhão o mais próximo possível de (0,0,0) e γ é um parâmetro adotado para assegurar um compromisso adequado entre a precisão da trajetória e o seu comprimento.

Os controladores inteligentes utilizados neste sistema dinâmico são: o PID-CE implementado em paralelo de controladores PID, implementado desta forma para tratar a sintonia de 6, 9 e 12 parâmetros, de acordo com a figura 7.15. Um importante detalhe a ser mencionado que para implementar este algoritmo PID necessita-se reescalonar a ação de controle (u) que é dada entre $[-1.2217; 1.2217]$ rad, para uma faixa de valores positivos mantendo-se assim coerência com a teoria de controle PID. O espaço de busca dos parâmetros do PID-CE é estipulado na faixa $[0;1]$.

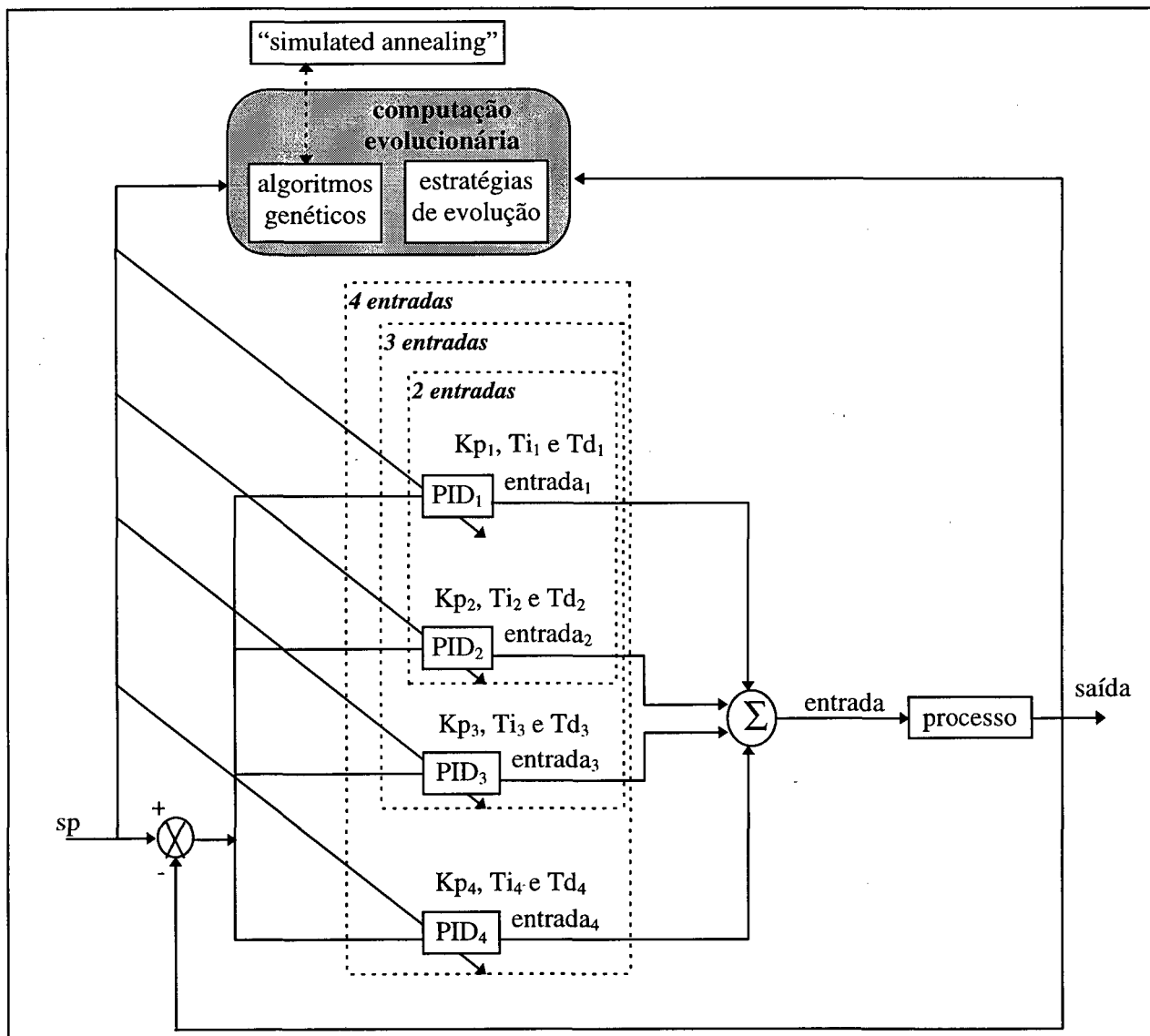


Figura 7.15 - Configuração de controladores PID em paralelo com sintonia por CE.

Outro controlador adotado é o controle CLN com determinação da base de regras conforme apresentada por Kosko (1992), contudo o ajuste das funções de pertinência é efetuado pelo paradigma da CE denominado EE com mutações correlatas e esquema de adaptação de parâmetros. O controle RN-CE tem a mesma arquitetura dos apresentados para o controle do processo de nível, contudo neste caso adota-se a topologia RNPREMOT sendo as entradas dadas pela equação do PID-RN (equação 5.12).

Tanto para o controle PID-CE quanto para o RN-CE são realizados experimentos com as metodologias de CE apresentadas na tabela 7.1. A seguir são tratados os experimentos realizados para os três casos estudados para 2,3 e 4 entradas para os controladores PID-CE, CLN e RN-CE.

São realizados dois experimentos diversos, no primeiro experimento, o valor de γ é ponderado com valor de 0.005, visando-se a utilização da função custo (equação 7.14) com objetivo de impor uma ponderação maior vinculada a **precisão** no estacionar do caminhão na garagem. Enquanto no segundo experimento o γ adotado é de 0.1, objetivando-se assim ênfase a **rapidez**, isto é, menor número de iterações para alcançar o objetivo de estacionar o caminhão na garagem (0,0,0) ou o mais próximo dela.

- **Experimento 1 ($\gamma = 0.005$):**

Caso 1: Duas entradas (x,y)

A tabela 7.6 apresenta os resultados do controle do caminhão utilizando-se de duas entradas (x,y) para os controladores inteligentes utilizados.

Tabela 7.6 - Resultados do controle do caminhão com duas entradas (x,y) (experimento 1).

dados									
número de entradas		2 → (x,y)							
condição inicial		x=20 m, y=10 m, $\theta_s=-2$ rad, $\theta_c=-2$ rad							
objetivo		(x,y, θ_s) = (0,0,0)							
distância mínima da condição inicial até a garagem		22.36							
número máximo de iterações permitido		300							
número de gerações nas metodologias da CE		500							
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	nº	parâmetro	ve	vu	traj (m)	erro _{traj}	valor	iteração	J _c
PID-CE	5	Kp ₁ =0.50 Ti ₁ =0.82 Td ₁ =0.64 Kp ₂ =0.67 Ti ₂ =1.00 Td ₂ =0.59	4.27	0.63	35.08	1.57	x = 0.293 y = 11.167 $\theta_s = 0.320$	np=57	125.18
CLN	5		9.97	1.35	36.40	1.63	x = 1.244 y = 1.893 $\theta_s = 0.572$	np=300	6.958
RN-CE	3	n ₀ = 2 n ₁ = 7 n ₂ = 1	15.89	0.53	45.86	2.05	x = 0.067 y = 1.676 $\theta_s = 0.895$	np=250	4.864
convenções									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2)}						k = 1 → controle PID ₁ da coordenada x k = 2 → controle PID ₂ da coordenada y			
valor						valores mais adequados obtidos de x (em m), y (em m), θ_s (em rad) e θ_c (em rad)			

A partir dos resultados mais adequados obtidos (tabela 7.6) são traçados os gráficos inerentes ao controle PID-CE (figura 7.16), CLN (figura 7.17) e RN-CE (figura 7.18).

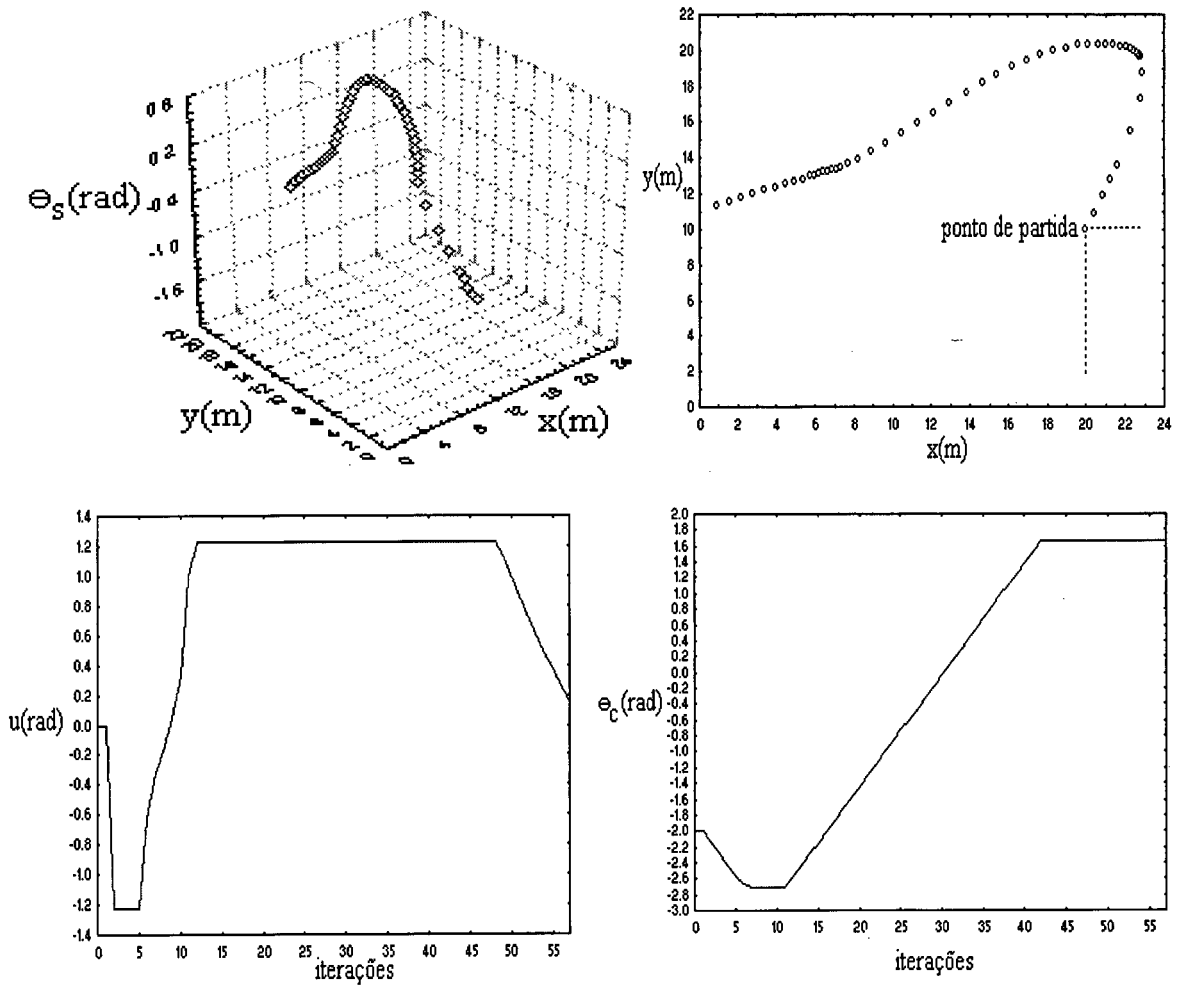
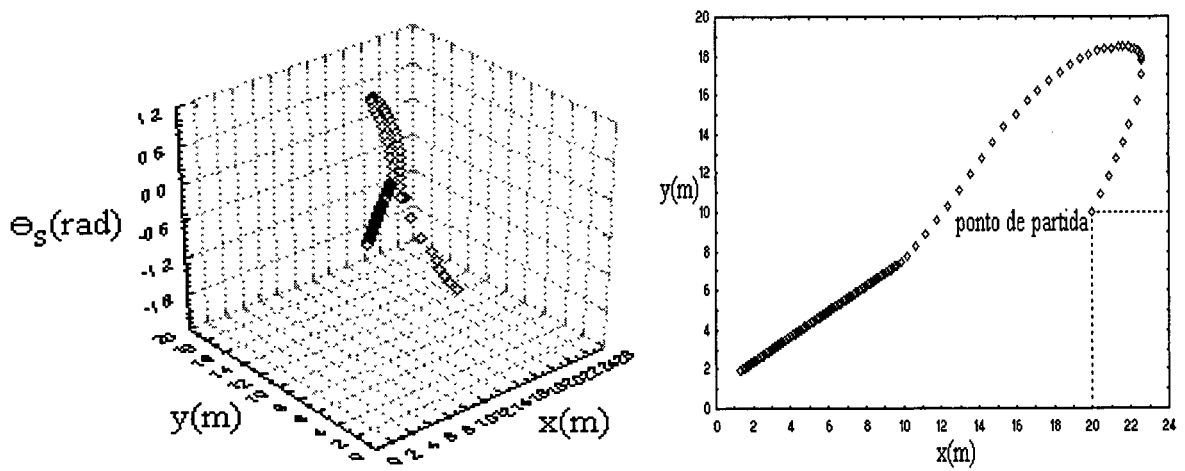


Figura 7.16 - Resultados do controle do caminhão com duas entradas (x,y) via PID-CE (experimento 1).



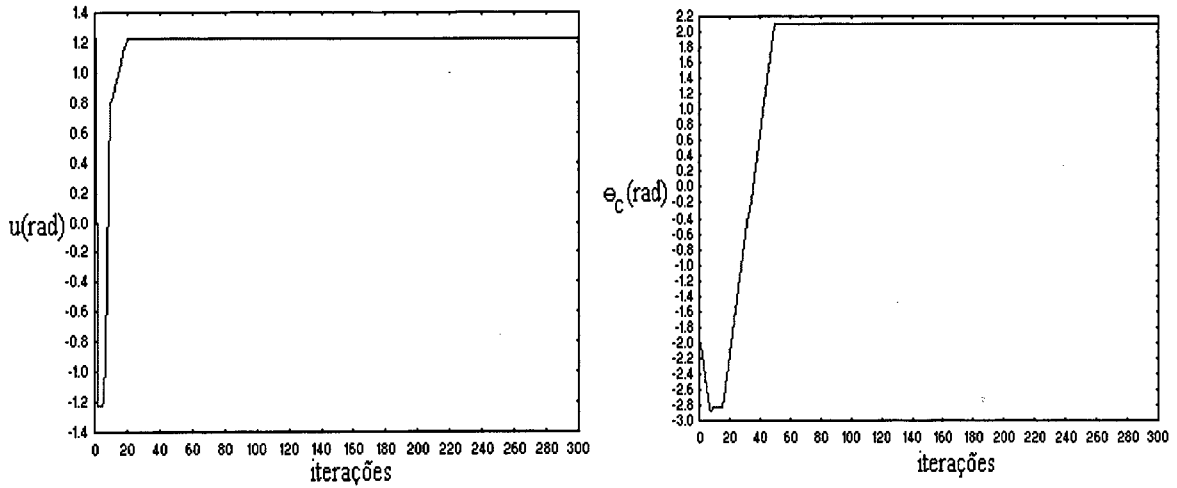


Figura 7.17 - Resultados do controle do caminhão com duas entradas (x,y) via CLN (experimento 1).

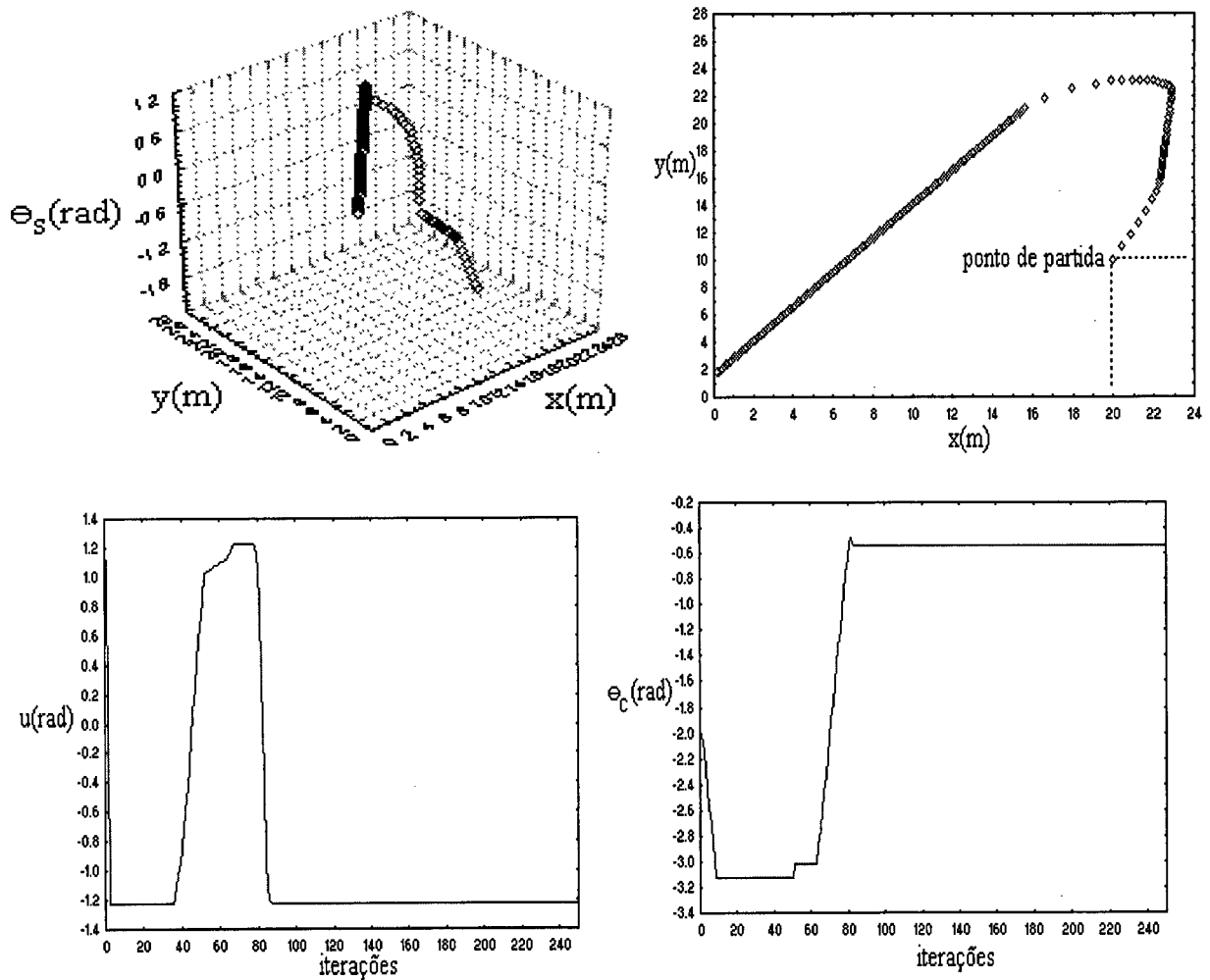


Figura 7.18 - Resultados do controle do caminhão com duas entradas (x,y) via RN-CE (experimento 1).

Caso 2: Três entradas (x,y,θs)

A tabela 7.7 apresenta os resultados do controle do caminhão utilizando-se de três entradas (x,y,θs) para os controladores inteligentes utilizados. A partir dos resultados mais adequados obtidos (tabela 7.7) são traçados os gráficos inerentes ao controle PID-CE (figura 7.19), CLN (figura 7.20) e RN-CE (figura 7.21).

Tabela 7.7 - Resultados do controle do caminhão com três entradas (x,y,θs) (experimento 1).

dados									
número de entradas					3 → (x,y,θs)				
condição inicial					x=20 m, y=10 m, θs=-2 rad, θc=-2 rad				
objetivo					(x,y,θs) = (0,0,0)				
distância mínima da condição inicial até a garagem					22.36				
número máximo de iterações permitido					300				
número de gerações das metodologias da CE					500				
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	n°	parâmetro	ve	vu	traj.(m)	erro _{traj}	valor	iteração	J _c
PID-CE	5	Kp ₁ =0.50 Ti ₁ =0.92 Td ₁ =0.80 Kp ₂ =0.77 Ti ₂ =1.00 Td ₂ =1.00 Kp ₃ =0.90 Ti ₃ =0.92 Td ₃ =0.85	8.66	2.87	48.61	2.17	x = 0.195 y = -0.855 θs = 0.919	np=142	2.324
CLN	5		5.97	0.14	37.73	1.68	x = 0.078 y = -0.965 θs = 0.837	np=129	2.283
RN-CE	5	n ₀ = 3 n ₁ = 7 n ₂ = 1	5.97	0.14	37.62	1.68	x = 0.073 y = -0.958 θs = 0.837	np=129	2.269
convenções									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2,3)}						i = 1 → controle PID ₁ da coordenada x i = 2 → controle PID ₂ da coordenada y i = 3 → controle PID ₃ da coordenada θs			
valor						valores mais adequados obtidos de x (em m), y (em m), θs (em rad) e θc (em rad)			

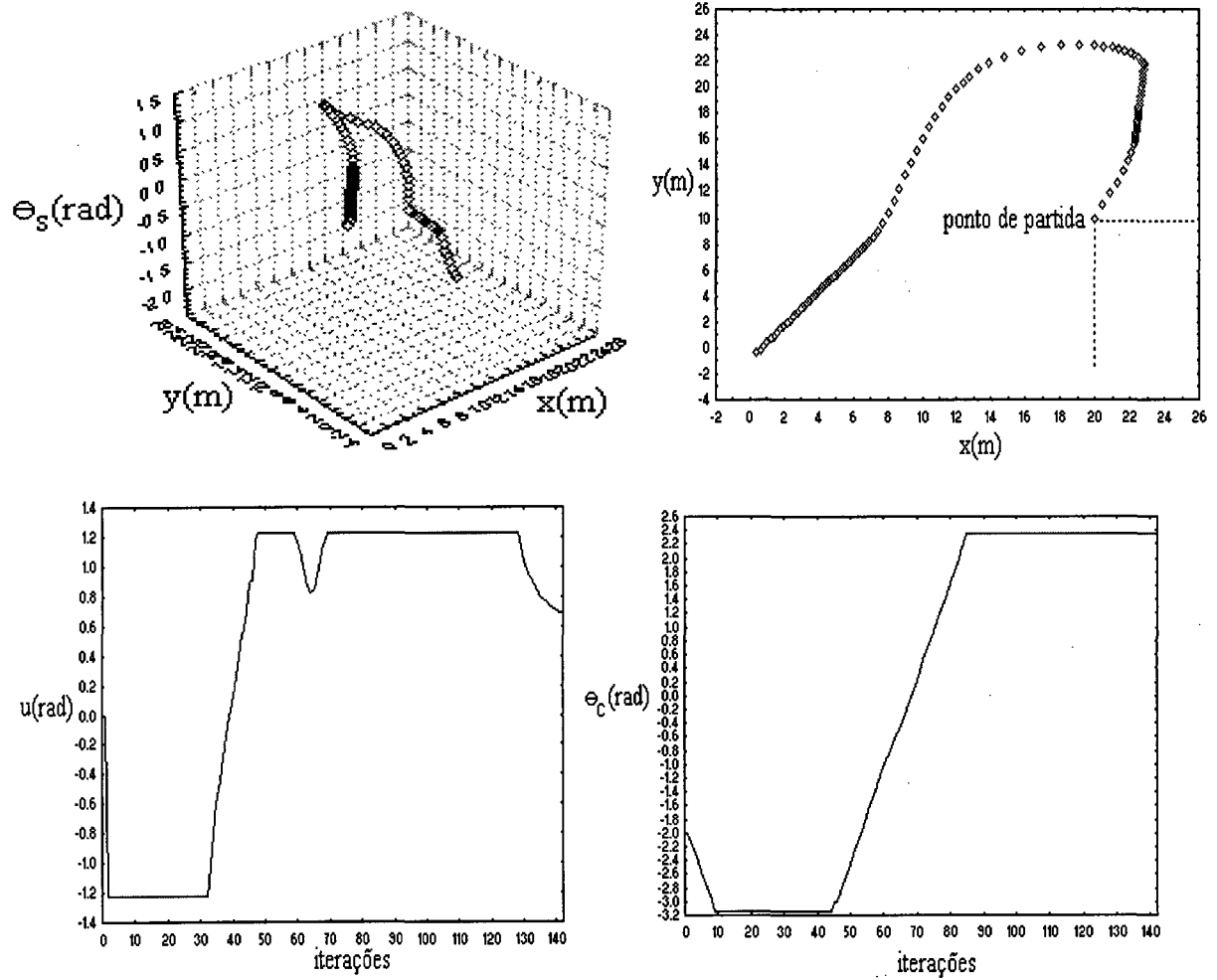
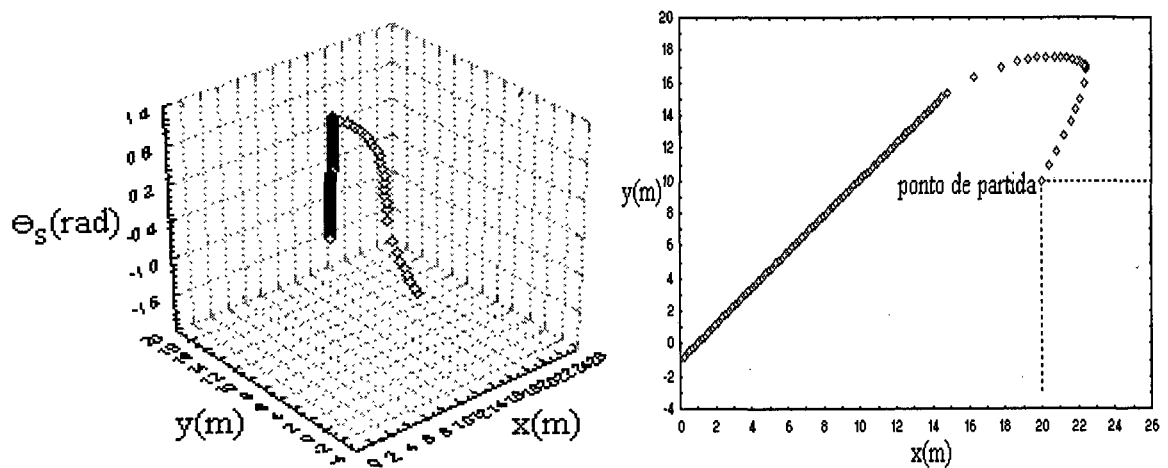


Figura 7.19 - Resultados do controle do caminhão com três entradas (x, y, θ_s) via PID-CE (experimento 1).



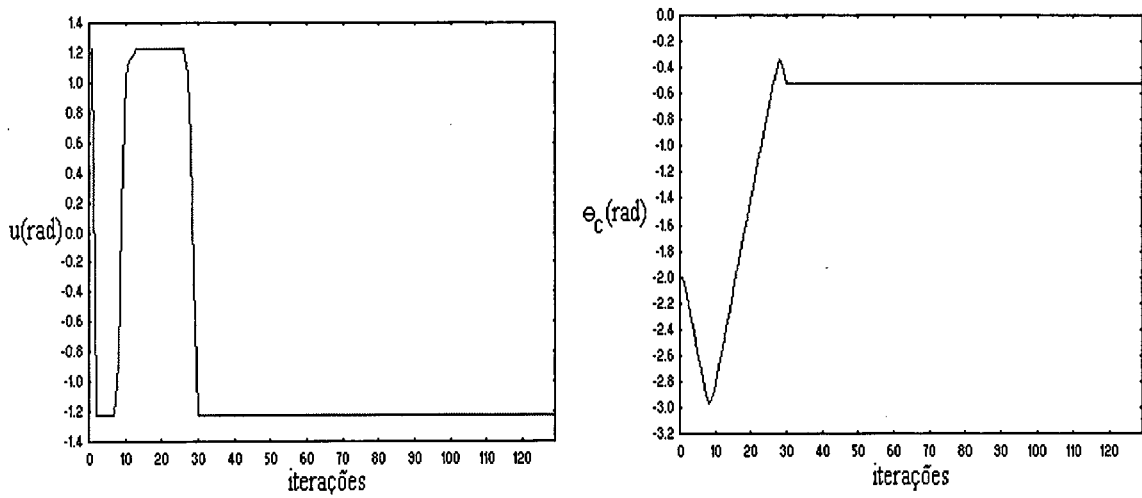


Figura 7.20 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via CLN (experimento 1).

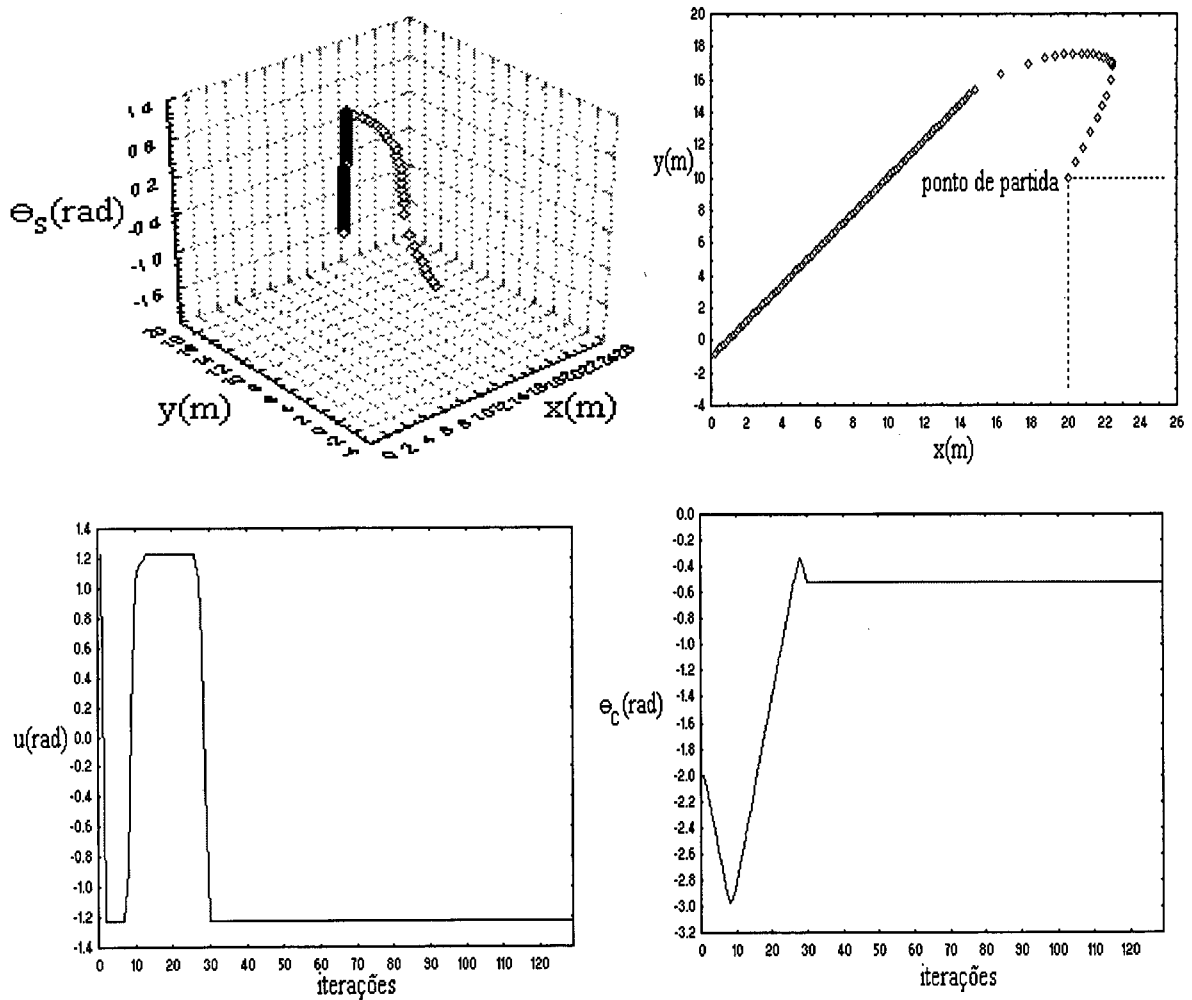


Figura 7.21 - Resultados do controle do caminhão com três entradas (x,y,θ_s) via RN-CE (experimento 1).

Caso 3: Quatro entradas (x,y,θs,θc)

A tabela 7.8 apresenta os resultados do controle do caminhão utilizando-se de quatro entradas (x,y,θs,θc) para os controladores inteligentes utilizados. A partir dos resultados mais adequados obtidos (tabela 7.8) são traçados os gráficos inerentes ao controle PID-CE (figura 7.22), CLN (figura 7.23) e RN-CE (figura 7.24).

Tabela 7.8 - Resultados do controle do caminhão com quatro entradas (x,y,θs,θc) (experimento 1).

dados									
número de entradas					4 → (x,y,θs,θc)				
condição inicial					x=20 m, y=10 m, θs=-2 rad, θc=-2 rad				
objetivo					(x,y,θs) = (0,0,0)				
distância mínima da condição inicial até a garagem					22.36				
número máximo de iterações permitido					300				
número de gerações das metodologias da CE					500				
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	nº	parâmetro	ve	vu	traj.(m)	erro _{traj}	valor	iteração	J _c
PID-CE	3	Kp ₁ =0.50 Ti ₁ =0.99 Td ₁ =0.72 Kp ₂ =0.79 Ti ₂ =0.99 Td ₂ =0.96 Kp ₃ =0.57 Ti ₃ =0.62 Td ₃ =0.09 Kp ₄ =0.60 Ti ₄ =0.56 Td ₄ =0.12	6.59	0.10	39.74	1.78	x = 0.021 y = -0.082 θs = 0.875	np=142	1.483
CLN	5		11.73	0.24	43.19	1.93	x = 0.100 y = 0.631 θs = 0.848	np=195	2.102
RN-CE	3	n ₀ = 4 n ₁ = 7 n ₂ = 1	6.99	0.07	43.58	1.95	x = 0.029 y = 0.691 θs = 0.531	np=97	1.245
convenções									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2,3,4)}						i = 1 → controle PID ₁ da coordenada x i = 2 → controle PID ₂ da coordenada y i = 3 → controle PID ₃ da coordenada θs i = 4 → controle PID ₄ da coordenada θc			
valor						valores mais adequados obtidos de x (em m), y (em m), θs (em rad) e θc (em rad)			

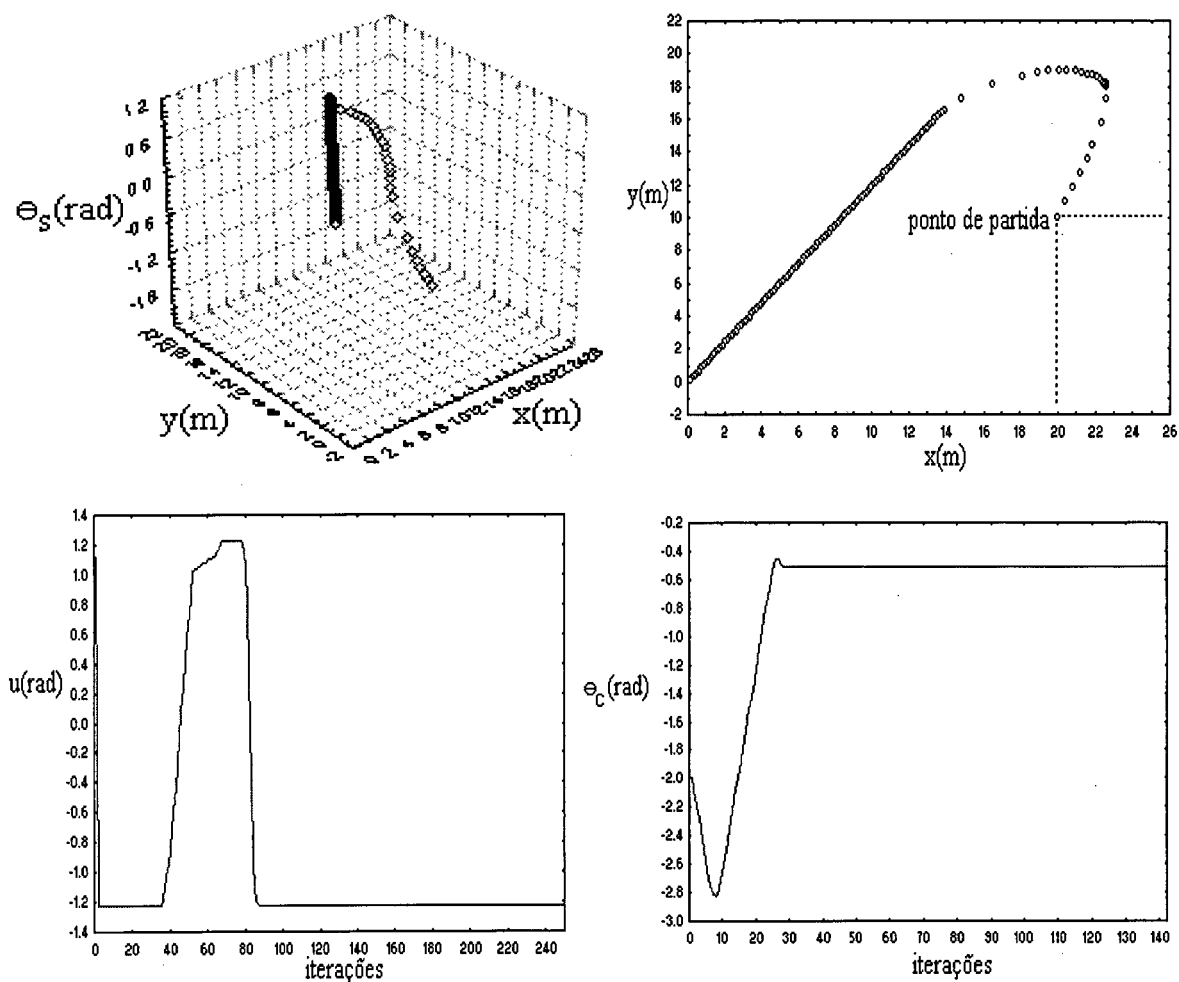
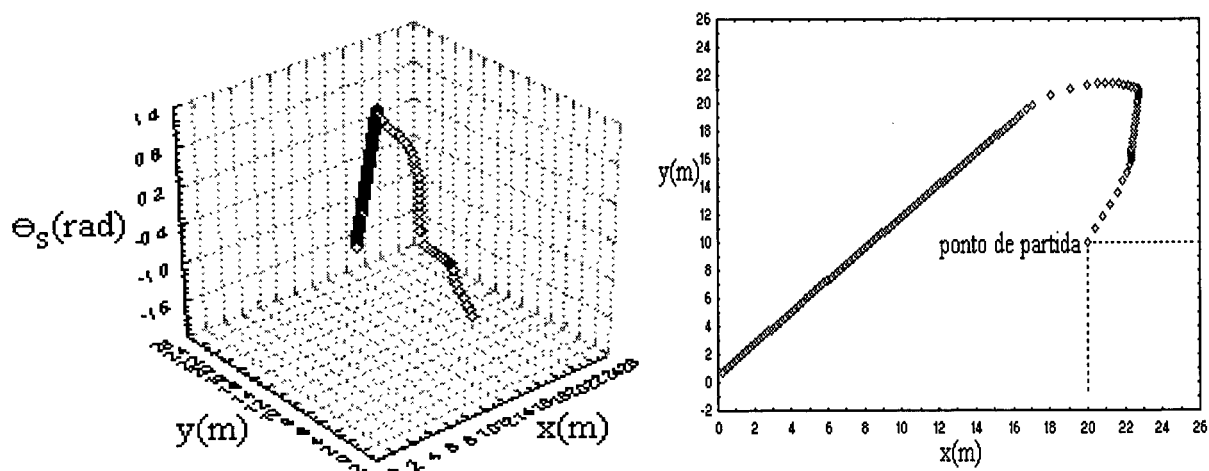


Figura 7.22 - Resultados do controle do caminhão com quatro entradas (x, y, θ_s, θ_c) via PID-CE (experimento 1).



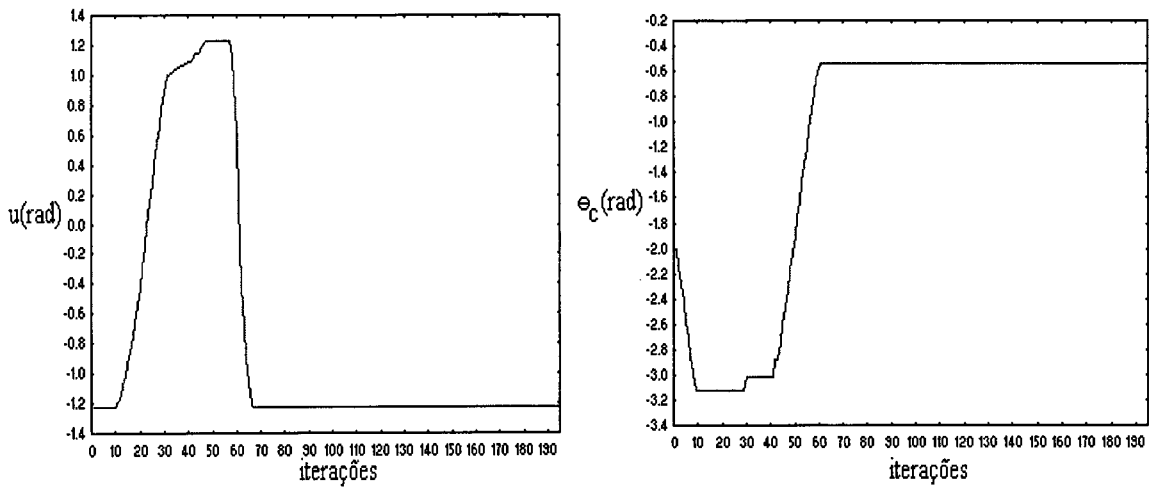


Figura 7.23 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via CLN (experimento 1).

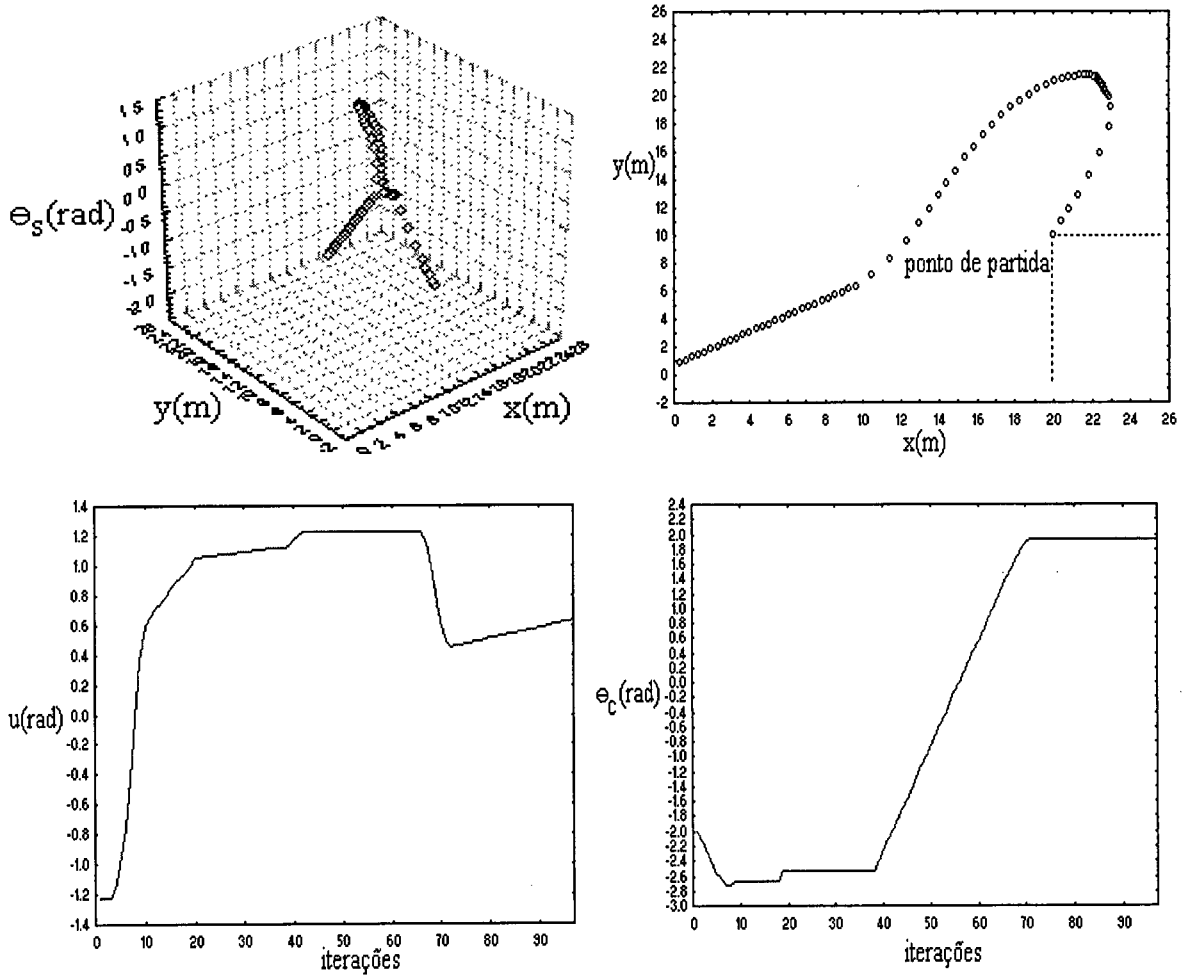


Figura 7.24 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via RN-CE (experimento 1).

A análise dos resultados obtidos com o experimento 1, para os três casos (2,3 e 4 entradas) estão ressaltados na tabelas 7.6 a 7.8. Nota-se que exceto para o caso de duas entradas onde o controle PID-CE obteve resultados pouco promissores, para os outros casos (3 e 4 entradas) os resultados apresentados pelos controladores inteligentes PID-CE, CLN e PID-RN são bastante próximos quanto a avaliação de J_c .

Todavia é relevante mencionar que o controlador mais adequado dos estudados para o experimento 1 é o RN-CE com otimização dos pesos via EE-(1,29) com variações correlatas e esquema de adaptação de parâmetros, com a obtenção de função custo $J_c=1.245$, onde $x=0.029$ m, $y=0.691$ m e $\theta_s=0.531$ rad, em 97 iterações, para o caso de quatro entradas (x,y,θ_s,θ_c).

- **Experimento 2 ($\gamma = 0.1$):**

Caso 1: Duas entradas (x,y)

A tabela 7.9 apresenta os resultados do controle do caminhão utilizando-se de duas entradas (x,y) para os controladores inteligentes utilizados.

Tabela 7.9 - Resultados do controle do caminhão com duas entradas (x,y) (experimento 2).

dados									
número de entradas					2 → (x,y)				
condição inicial					x=20 m, y=10 m, $\theta_s=-2$ rad, $\theta_c=-2$ rad				
objetivo					$(x,y,\theta_s) = (0,0,0)$				
distância mínima da condição inicial até a garagem					22.36				
número máximo de iterações permitido					300				
número de gerações nas metodologias da CE					500				
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	n°	parâmetro	ve	vu	traj. (m)	erro _{traj.}	valor	iteração	J _c
PID-CE	3	Kp ₁ =0.50 Ti ₁ =0.32 Td ₁ =0.24 Kp ₂ =0.65 Ti ₂ =1.00 Td ₂ =0.62	9.83	0.08	32.44	1.45	x = 0.042 y = 12.063 $\theta_s = 0.535$	np=159	161.70
CLN	5		6.07	0.11	41.49	1.86	x = 0.319 y = -2.919 $\theta_s = 0.942$	np=127	22.210
RN-CE	3	n ₀ = 2 n ₁ = 7 n ₂ = 1	5.53	0.07	35.41	1.58	x = 0.066 y = 2.754 $\theta_s = 0.775$	np=110	19.189
convenções									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2)}					k = 1 → controle PID ₁ da coordenada x k = 2 → controle PID ₂ da coordenada y				
valor					valores mais adequados obtidos de x (em m), y (em m), θ_s (em rad) e θ_c (em rad)				

A partir dos resultados mais adequados obtidos (tabela 7.9) são traçados os gráficos inerentes ao controle PID-CE (figura 7.25), CLN (figura 7.26) e RN-CE (figura 7.27).

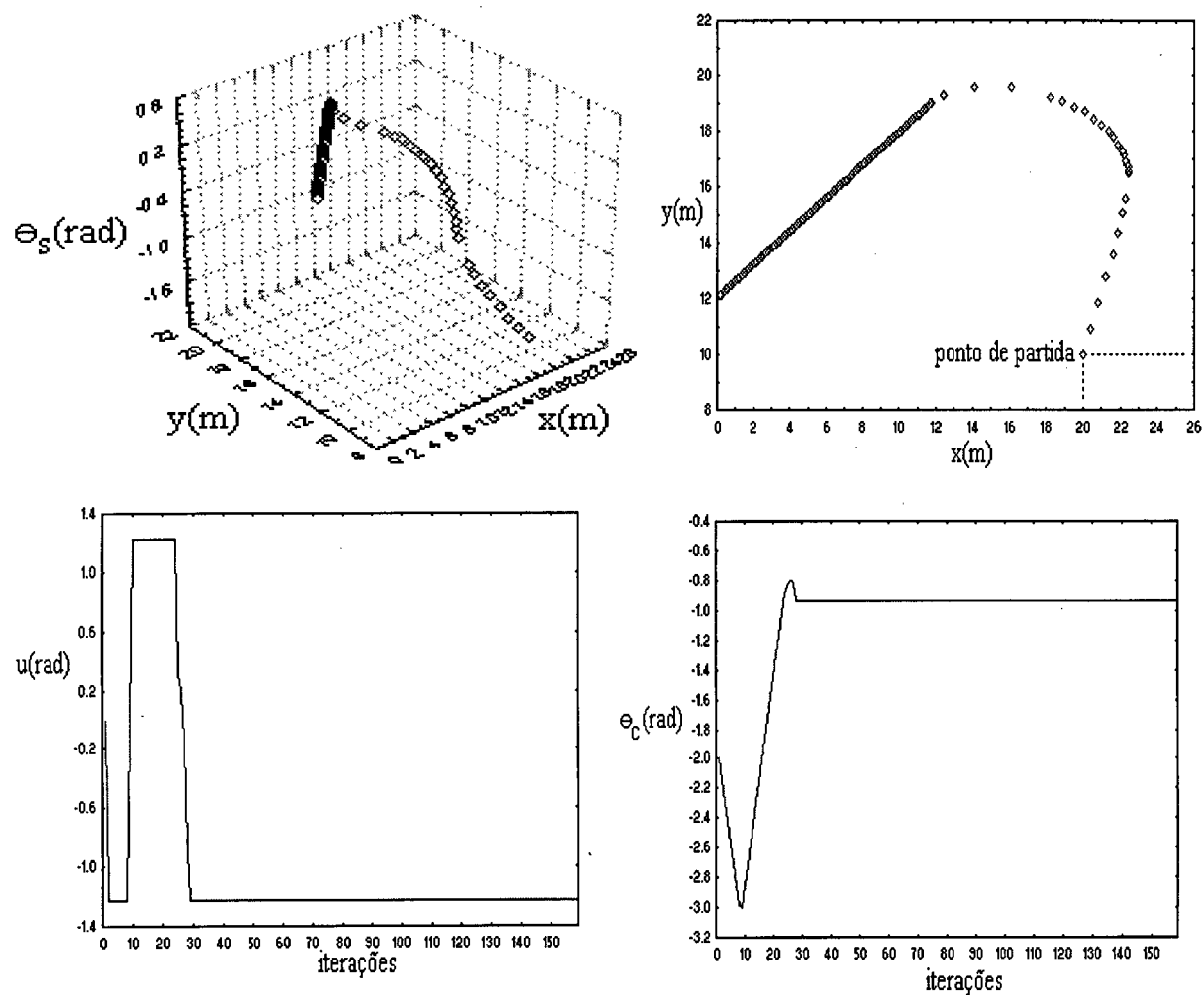
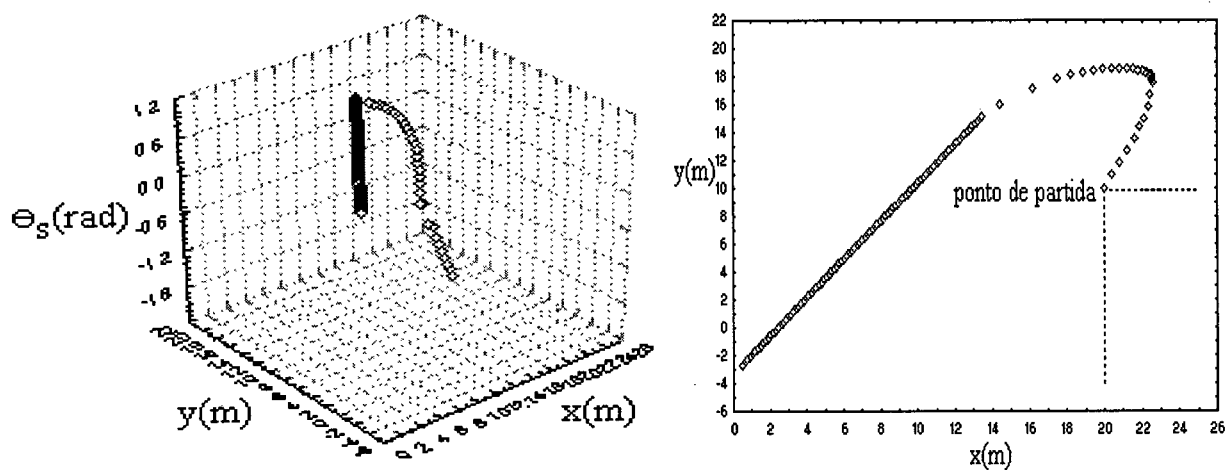


Figura 7.25 - Resultados do controle do caminhão com duas entradas (x,y) via PID-CE (experimento 2).



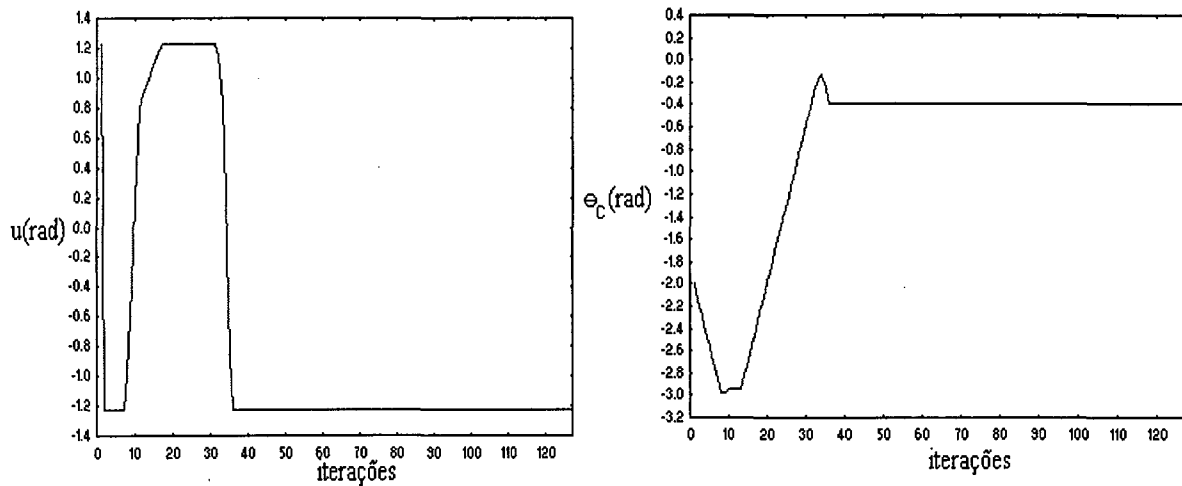


Figura 7.26 - Resultados do controle do caminhão com duas entradas (x,y) via CLN (experimento 2).

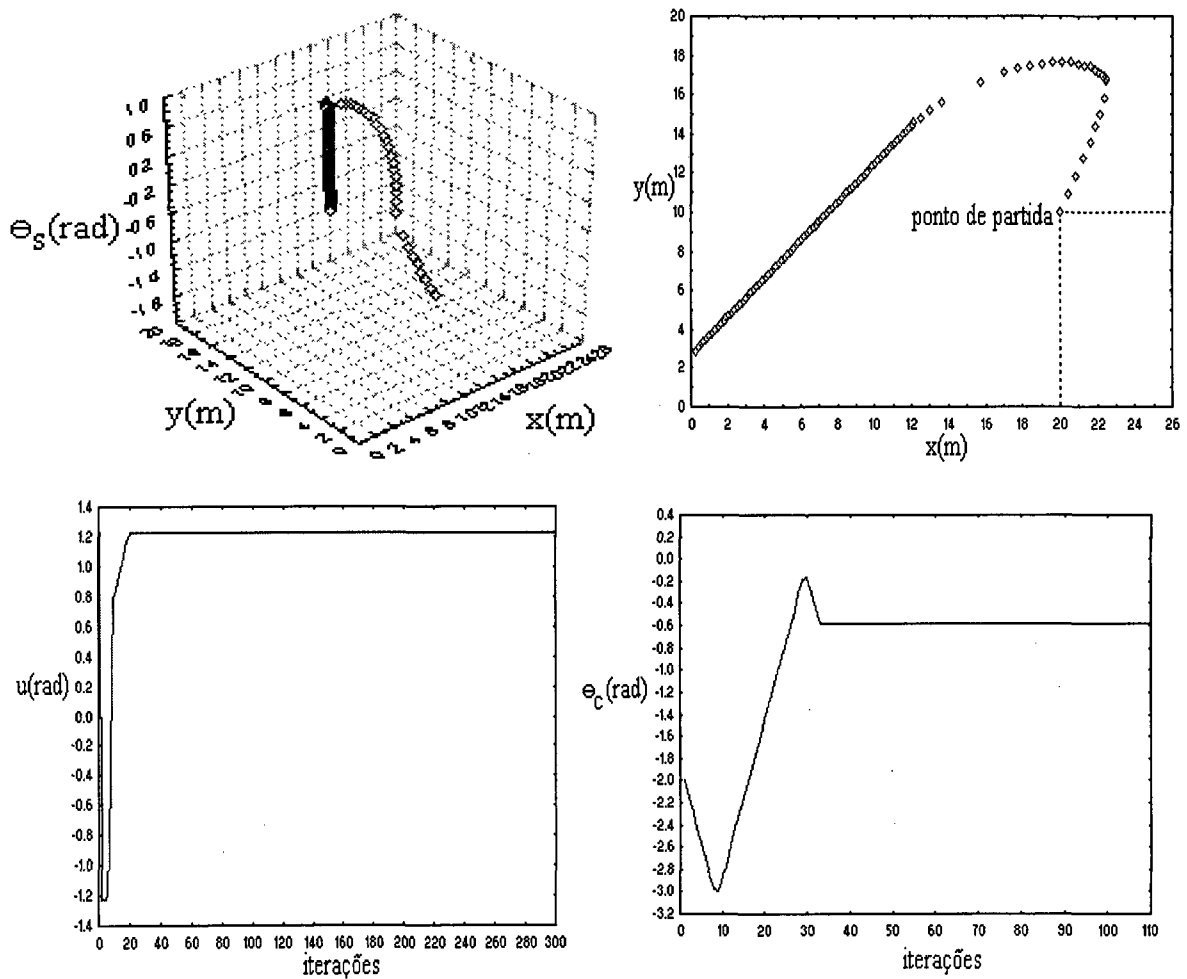


Figura 7.27 - Resultados do controle do caminhão com duas entradas (x,y) via RN-CE (experimento 2).

Caso 2: Três entradas (x,y,θs)

A tabela 7.10 apresenta os resultados do controle do caminhão utilizando-se de três entradas (x,y,θs) para os controladores inteligentes utilizados. A partir dos resultados mais adequados obtidos (tabela 7.10) são traçados os gráficos inerentes ao controle PID-CE (figura 7.28), CLN (figura 7.29) e RN-CE (figura 7.30).

Tabela 7.10 - Resultados do controle do caminhão com três entradas (x,y,θs) (experimento 2).

dados									
número de entradas					3 → (x,y,θs)				
condição inicial					x=20 m, y=10 m, θs=-2 rad, θc=-2 rad				
objetivo					(x,y,θs) = (0,0,0)				
distância mínima da condição inicial até a garagem					22.36				
número máximo de iterações permitido					300				
número de gerações das metodologias da CE					500				
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	nº	parâmetro	ve	vu	traj (m)	erro _{traj}	valor	iteração	J _c
PID-CE	5	Kp ₁ =0.51 Ti ₁ =0.94 Td ₁ =1.00 Kp ₂ =0.50 Ti ₂ =0.99 Td ₂ =0.78 Kp ₃ =0.55 Ti ₃ =0.98 Td ₃ =0.18	5.05	0.49	50.12	2.24	x = 0.368 y = -0.681 θs = 0.603	np=54	6.363
CLN	5		5.97	0.14	37.73	1.68	x = 0.078 y = -0.955 θs = 0.837	np=129	14.519
RN-CE	5	n ₀ = 3 n ₁ = 7 n ₂ = 1	3.86	0.33	52.10	2.33	x = 0.382 y = -1.107 θs = 1.418	np=44	7.189
convenções:									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2,3)}						i = 1 → controle PID ₁ da coordenada x i = 2 → controle PID ₂ da coordenada y i = 3 → controle PID ₃ da coordenada θs			
valor						valores mais adequados obtidos de x (em m), y (em m), θs (em rad) e θc (em rad)			

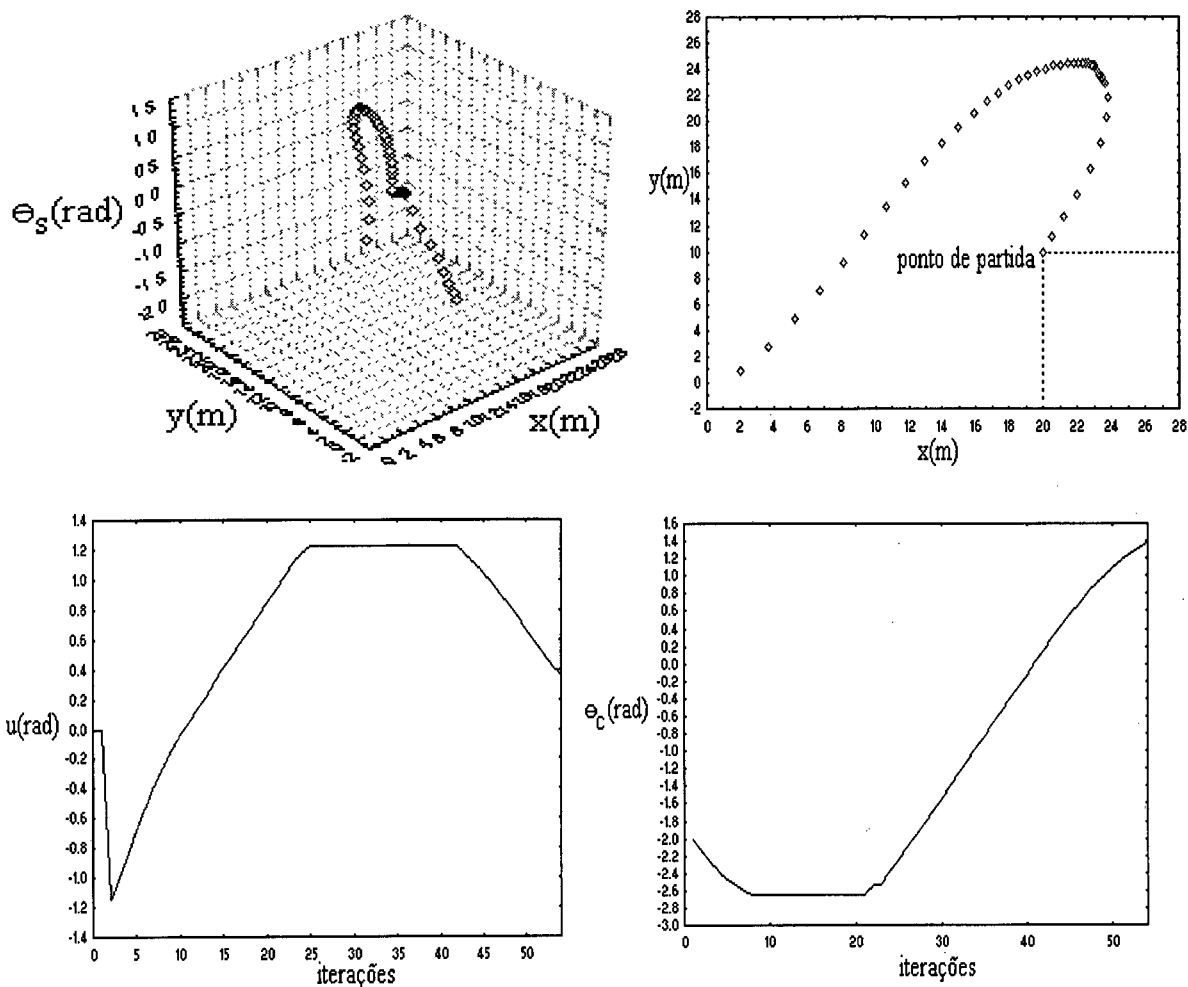
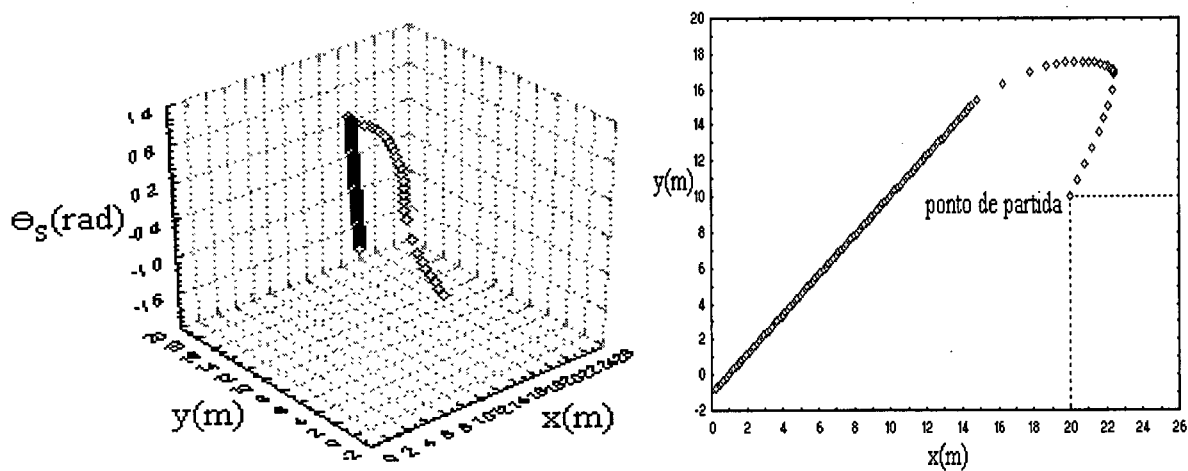


Figura 7.28- Resultados do controle do caminhão com três entradas (x, y, θ_s) via PID-CE (experimento 2).



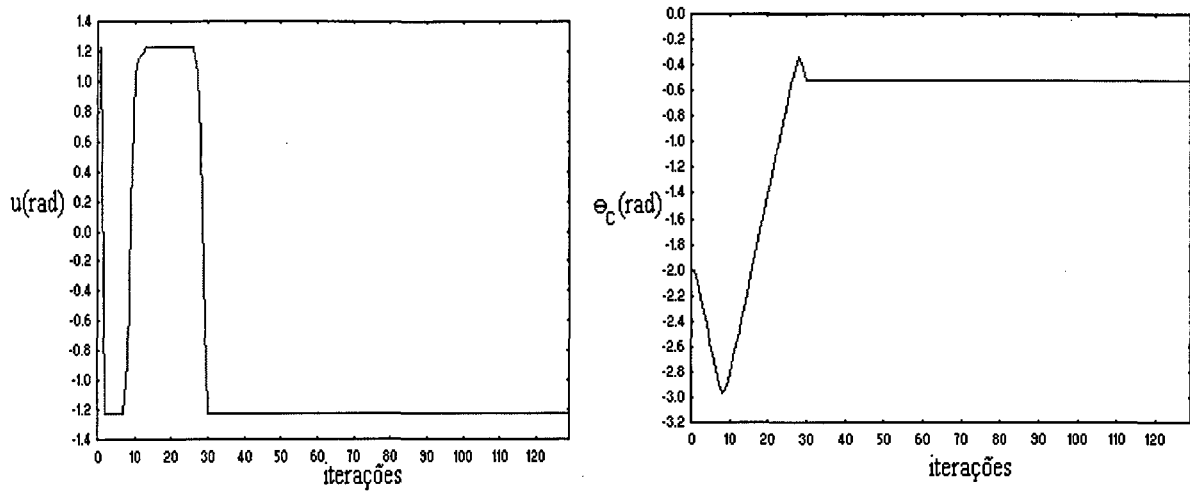


Figura 7.29 - Resultados do controle do caminhão com três entradas (x, y, θ s) via CLN (experimento 2).

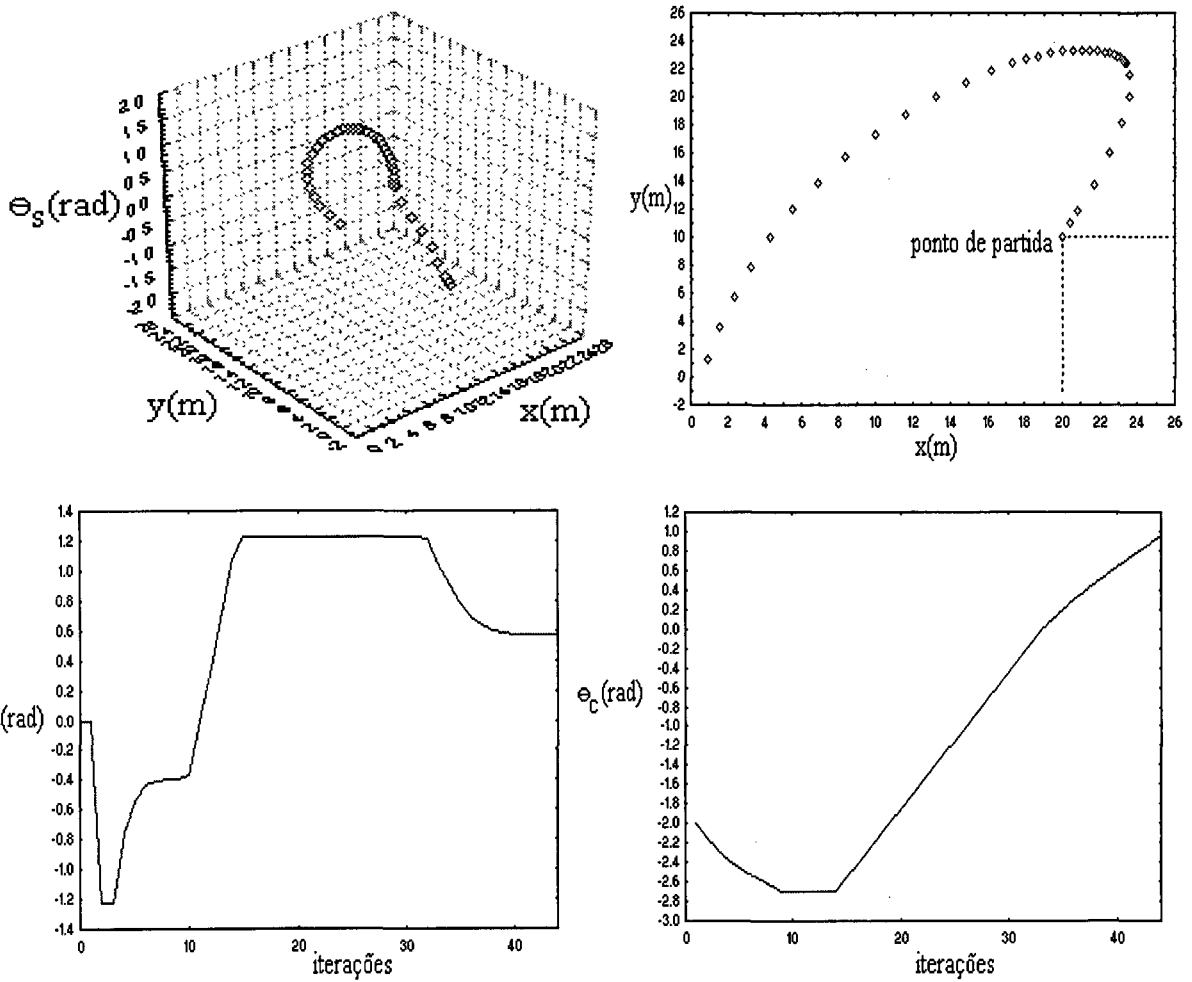


Figura 7.30 - Resultados do controle do caminhão com três entradas (x, y, θ s) via RN-CE (experimento 2).

Caso 3: Quatro entradas (x,y,θs,θc)

A tabela 7.11 apresenta os resultados do controle do caminhão utilizando-se de quatro entradas (x,y,θs,θc) para os controladores inteligentes utilizados. A partir dos resultados mais adequados obtidos (tabela 7.11) são traçados os gráficos inerentes ao controle PID-CE (figura 7.31), CLN (figura 7.32) e RN-CE (figura 7.33).

Tabela 7.11 - Resultados do controle do caminhão com quatro entradas (x,y,θs,θc) (experimento 2).

dados									
número de entradas					4 → (x,y,θs,θc)				
condição inicial					x=20 m, y=10 m, θs=-2 rad, θc=-2 rad				
objetivo					(x,y,θs) = (0,0,0)				
distância mínima da condição inicial até a garagem					22.36				
número máximo de iterações permitido					300				
número de gerações das metodologias da CE					500				
resultados									
configuração			variância		caminhão		melhor resultado		f. custo
algoritmo	nº	parâmetro	ve	vu	traj.(m)	erro _{traj}	valor	iteração	J _c
PID-CE	3	Kp ₁ =0.50 Ti ₁ =1.00 Td ₁ =1.00 Kp ₂ =0.50 Ti ₂ =1.00 Td ₂ =0.53 Kp ₃ =0.57 Ti ₃ =0.52 Td ₃ =0.02 Kp ₄ =0.32 Ti ₄ =0.56 Td ₄ =0.12	4.84	0.45	50.34	2.25	x = 1.198 y = -0.020 θs = 0.802	np=51	7.179
CLN	5		6.41	0.63	54.19	2.42	x = 0.836 y = 1.006 θs = 0.860	np=63	8.750
RN-CE	3	n ₀ = 4 n ₁ = 7 n ₂ = 1	3.72	0.33	44.06	1.97	x = 1.367 y = 0.496 θs = 0.458	np=43	6.624
convencões									
Kp _k , Ti _k , Td _k , onde {k ∈ ℕ / k = (1,2,3,4)}						i = 1 → controle PID ₁ da coordenada x i = 2 → controle PID ₂ da coordenada y i = 3 → controle PID ₃ da coordenada θs i = 4 → controle PID ₄ da coordenada θc			
valor						valores mais adequados obtidos de x (em m), y (em m), θs (em rad) e θc (em rad)			

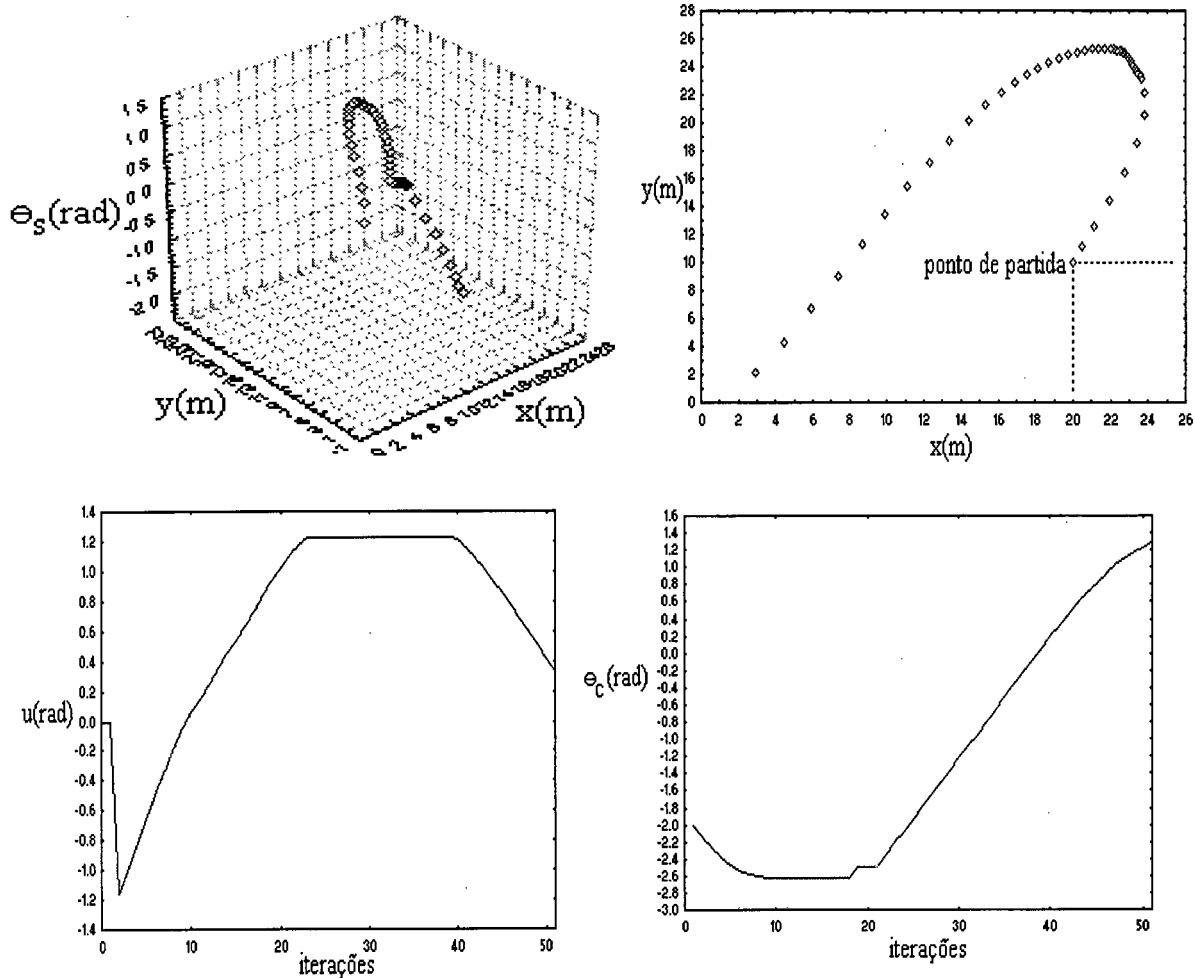
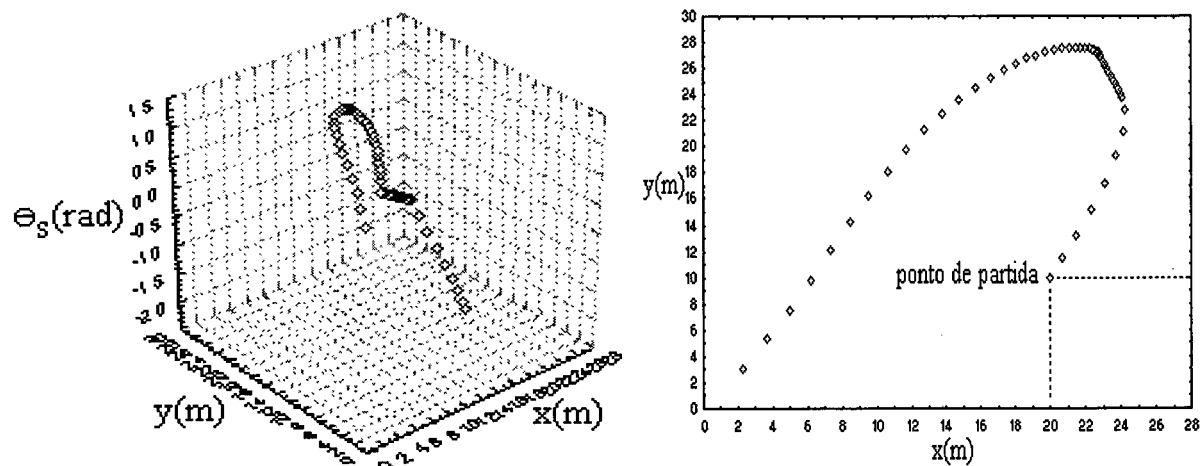


Figura 7.31 - Resultados do controle do caminhão com quatro entradas (x, y, θ_s, θ_c) via PID-CE (experimento 2).



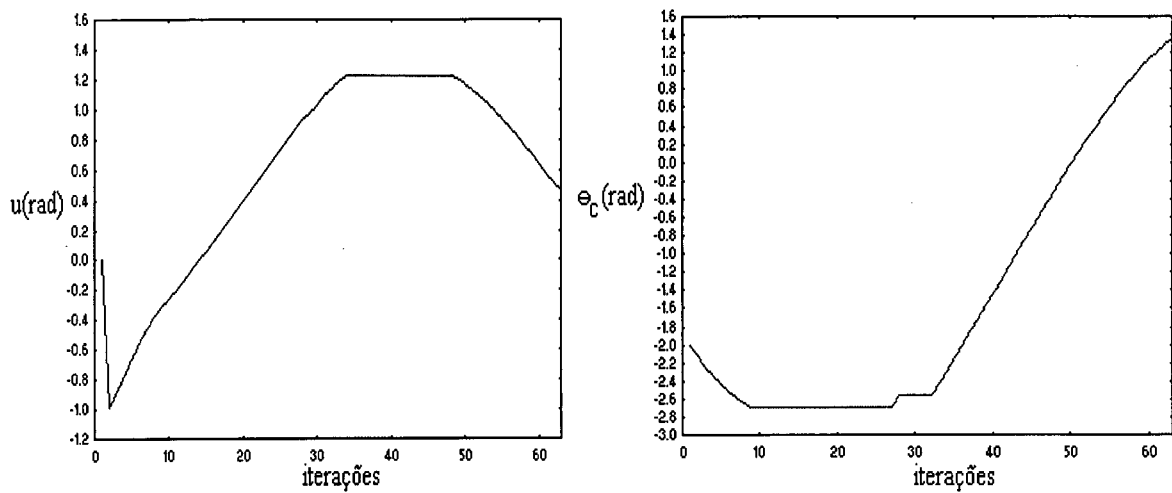


Figura 7.32 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via CLN (experimento 2).

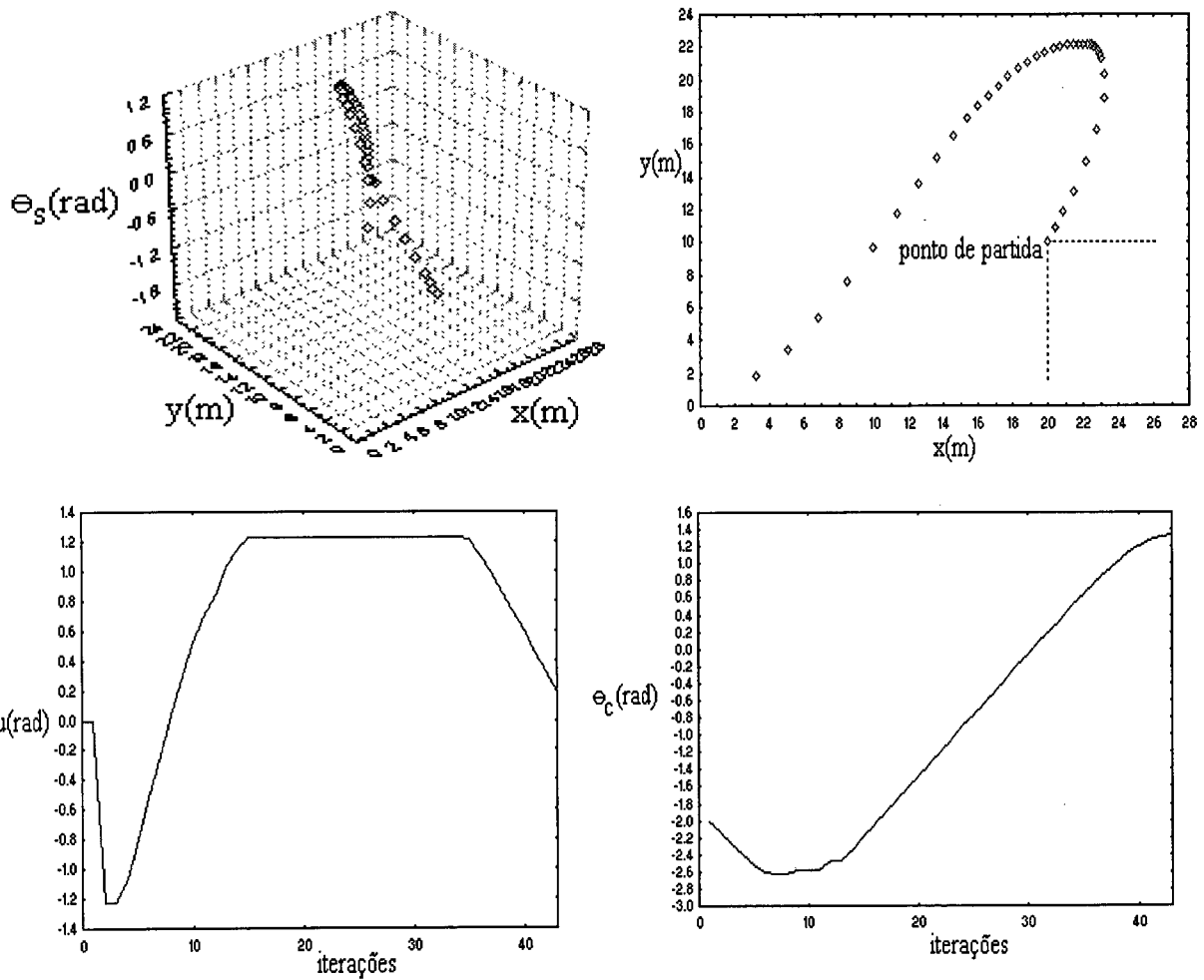


Figura 7.33 - Resultados do controle do caminhão com quatro entradas (x,y,θ_s,θ_c) via RN-CE (experimento 2).

A análise dos resultados obtidos com os três casos abordados no experimento 2 ($\gamma=0.1$) estão ressaltados na tabelas 7.9 a 7.11. Todavia é relevante mencionar que o controlador mais adequado dos estudados para o experimento 2 é o PID-CE com otimização dos pesos via EE-(1,29) com variações correlatas e esquema de adaptação de parâmetros, com a obtenção de função custo $J_c=6.363$ e de uma distância da garagem (0,0,0) dada por $x= 0.368$ m, $y = -0.681$ m e $\theta_s = 0.603$ rad, em 54 iterações, para o caso de três entradas (x,y,θ_s) .

7.4 Conclusão

Este capítulo apresentou a descrição dos processos utilizados nos experimentos em controle de processos. Os processos avaliados foram: processo de nível duplo-tanque acoplado e sistema dinâmico não-linear para a manobra de um caminhão até a garagem.

Os algoritmos inteligentes abordados em controle foram a CE, LN, RNs e sistemas híbridos inteligentes. Os resultados foram efetivados no sistema não-linear de regulação de nível, composto de dois tanques acoplados, sendo que cinco metodologias inteligentes foram aplicadas na configuração de controle inteligente, são elas: PID-CE, PID-CLN, PID-RN, RN-CE e CLIT-CE. Enquanto no controle do sistema dinâmico de controle de um caminhão até a garagem aplicou-se três paradigmas da IC, o PID-CE, CLN e PID-RN.

Obteve-se resultados promissores através dos algoritmos inteligentes aplicados, podendo estas metodologias de controle serem configuradas em controle de outros processos, tanto monovariáveis quanto multivariáveis, contudo deve-se ressaltar a necessidade de análise mais aprofundada de critérios de estabilidade, complexidade algorítmica e convergência das metodologias aplicadas em relação a métodos convencionais de controle.

Capítulo 8

Conclusões gerais e perspectivas para futuros trabalhos

Nesta dissertação foi proposta a descrição, projeto e implementação de metodologias da inteligência computacional em identificação e controle de processos, em escala laboratorial, com propósito ao tratamento adequado das restrições e dos requisitos exigidos no projeto de sistemas.

As metodologias de inteligência computacional utilizadas foram: lógica nebulosa, computação evolucionária (algoritmos genéticos e estratégias de evolução), redes neurais artificiais (“feedforward” e parcialmente recorrente de Elman) e “simulated annealing”. O projeto de sistemas híbridos inteligentes, composto de algoritmos genéticos híbrido com “simulated annealing”, controle nebuloso com ajuste de parâmetros via computação evolucionária e algoritmos combinando redes neurais artificiais com o paradigma da computação evolucionária também foram abordados.

Neste trabalho avaliou-se também os algoritmos de identificação dos mínimos quadrados recursivo e variável instrumental, e as estratégias de controle do tipo PID e linear invariante no tempo.

8.1 Identificação de processos

Os testes experimentais e análise do desempenho dos algoritmos em identificação de processos foram realizados na identificação de um processo de temperatura, um processo de nível mono-tanque e um simulador de turbo-gerador.

A partir dos resultados experimentais e simulações obtidas tratadas no capítulo 6 referente a identificação de processos, concluiu-se que os paradigmas da IC, entre eles: CE (AG, AGSA e EE) e as RNs “feedforward” e RNPREMOM adequaram-se a identificação dos processos propostos.

Contudo cabe ressaltar que a aplicação de cada uma destas metodologias decorre em vantagens e desvantagens. Entre as vantagens de algoritmos convencionais de identificação MQR e VI em relação a CE tem-se: i) possibilidade da aplicação destas em identificação “on-

line”, esta relevante em aplicações de controle adaptativo; ii) embasamento matemático rigoroso; iii) menor complexidade computacional a que os algoritmos inteligentes.

Estes métodos são em essência tratados por algoritmos de busca guiados por gradiente ou informação da derivada. Os algoritmos convencionais requerem um espaço de busca regular ou uma função energia do erro diferenciável. Estes métodos podem facilmente falhar na obtenção de um ótimo global se o espaço de busca é multimodal e diferenciável ou o índice de desempenho não é bem “comportado”.

Os paradigmas CE aplicados apresentam como vantagens a possibilidade de tratamento do atraso de transporte podem ser aplicados a identificação de ambos sistemas de controle discreto ou contínuos (de fase mínima ou não-mínima); possibilidade do tratamento de problemas multiobjetivo e multimodais; tratamento de restrições e, não dependem de derivadas (Kristinsson and Dumont, 1992).

Os pontos negativos em relação a estas técnicas são: i) não há uma teoria formalizada para a prova de convergência e análise estatística sedimentada quanto a CE, que caracterizam-se por serem baseados em procura randômica (estudos mais promissores são referentes ao algoritmo “simulated annealing”; ii) complexidade computacional, o que dificulta sua aplicação em identificação “on-line”; iii) memória requerida para implementação de algoritmos evolutivos; iv) configuração heurística na definição dos parâmetros, tais como tamanho da população, probabilidade do uso dos operadores genéticos, entre outros.

As RNs foram outra metodologia da IC tratada na dissertação. As vantagens da aplicação das RNs estão intrinsecamente ligadas as características de mapeamentos em processos não-lineares, e adequação ao tratamento de processos complexos. Quanto a desvantagens o algoritmo apresenta de forma análoga aos algoritmos da CE, complexidade computacional dos algoritmos de RNs; definição da topologia mais adequada ao tipo de processo abordados e definição do número de entradas, camadas ocultas, funções de ativação e determinação dos coeficientes de aprendizado e momento.

8.2 Controle de processos

Os experimentos em controle abordaram um processo não-linear de nível, composto de dois tanques acoplados e o sistema dinâmico não-linear vinculado a manobrar um caminhão para estacioná-lo em uma garagem.

O capítulo 7 apresentam a aplicação de CE na sintonia de controle PID e linear invariante no tempo. As simulações realizadas frente aos processos propostos, objetivaram a sintonia de controle PID, controle linear invariante no tempo; controle lógico nebuloso com otimização por estratégias de evolução e controle neural evolutivo.

A CE aplicada na sintonia de controle convencional caracterizou-se como um alternativa aos algoritmos convencionais de sintonia. A desvantagem da CE é que para cada geração do algoritmo, toda a população de soluções (parâmetros do controlador) necessita ser avaliada para o cálculo da função de adequabilidade.

No projeto de CLN não necessitou-se do conhecimento de um modelo matemático preciso do processo a ser controlado. Se o modelo contudo existir, pode ser utilizado para a simulação e para teste da estratégia de controle.

As principais vantagens do CLN são: i) não necessitam do modelo matemático do sistema; ii) é possível implementar o conhecimento de especialista humano e experiência utilizando-se regras lingüísticas; iii) possibilidade de controle de processos não-lineares; iv) possibilita sua implementação em processadores dedicados, podendo, assim, ser utilizado para controle de processos rápidos.

Apesar de todas suas vantagens o CLN tem importantes desvantagens. Uma delas é a de não existir um padrão ou método sistêmico para a transformação de conhecimento humano ou experiência dentro da base de regras de um sistema de inferência nebuloso. A não existência de um procedimento geral para a escolha do ótimo número de regras, visto que um largo número de fatores estão envolvidos na decisão, isto é, desempenho do controlador, eficiência de computação, comportamento do operador, escolha de variáveis lingüísticas, entre outros.

Outras desvantagens importantes dos CLNs são: não é possível mostrar a estabilidade do sistema controlado, visto que o modelo não é conhecido; e não há garantia que as regras sejam coerentes (é possível que tenha-se um casamento não adequado entre as regras).

As RNs extraem informação de sistemas para serem aprendidas ou controladas, sendo que estimam o conhecimento numérico de forma não estruturada, enquanto as técnicas nebulosas utilizam a informação verbal de especialistas (ou através de algum algoritmo de otimização) e estimam o conhecimento numérico de forma estruturada (Kosko, 1992).

As RNs apresentam características relevantes de processamento paralelo, memória associativa e distribuída, aprendizado. Outra característica importante é o tratamento de não-linearidades, pois assim as RNs configuram-se em um campo em aberto para controle de

sistemas complexos. As vantagens do aprendizado neural está concentrado na habilidade de modelar e/ou controlar sistemas dinâmicos não-lineares (capacidade de generalização), contudo apresentam pontos negativos como o tempo computacional e análise da convergência do controlador.

O projeto de controle neural tem duas vertentes principais: esquema de aprendizado geral e aprendizado especializado. No esquema de aprendizado geral a RN é treinada “off-line” para aprender a inversa da dinâmica direta de acordo com um conjunto de treinamento determinado pelo projetista, usualmente pela aplicação de sinais de PRBS e ruído branco em diversas faixas de valores. Uma vez treinada a RN pode ser utilizada para configurar uma outra RN como controladora do processo em questão.

O esquema utilizado nesta dissertação foi o especializado que difere do aprendizado geral pois o algoritmo de treinamento ajusta os pesos da RN “on-line” (a cada amostra). O controle neural convencional apresenta contudo variantes como o controlador neural evolutivo que emprega CE para evitar a necessidade da estimação do gradiente da função custo.

8.3 Pesquisa futura e perspectivas na área de controle inteligente

A investigação de formas de projeto de metodologias da IC em configuração de controladores apresenta inúmeras possibilidades. A seguir apresenta-se alguns aspectos que merecem estudos mais aprofundados dentro do contexto dos algoritmos desenvolvidos nesta dissertação, e caminhos alternativos de pesquisa que não foram explorados.

Quanto a análise de convergência e estabilidade de algoritmos inteligentes em automação industrial a literatura menciona algumas propostas de análise de CLNs (baseados na teoria de estabilidade Lyapunov) e em controle neural. Contudo apenas são tratados tópicos específicos e não existe ainda uma teoria mais abrangente.

A complexidade computacional dos algoritmos inteligentes é outro tópico de intensas pesquisas, da forma mais eficiente, robusta e simples de combinar a IC com algoritmos convencionais e adaptativos para a aplicação em processos industriais complexos, principalmente processos químicos que apresentam acentuadas não-linearidades. Uma das alternativas viáveis é a aplicação destes algoritmos em poderosas arquiteturas computacionais de processamento paralelo.

Na área de identificação de processos a aplicação de outros métodos dentro do contexto da CE vem apresentando resultados promissores pela aplicação de algoritmos de programação genética e sistemas classificadores. Em sistemas híbridos inteligentes, tem-se o grande interesse da aplicação de redes neurais-nebulosas e sistemas especialistas nebulosos em identificação e controle de processos, estas com diversas maneiras de serem otimizadas através de CE.

Na área de controle de processos, necessita-se de um estudo mais abrangente de controladores neurais diretos e controle inverso, através da utilização de uma RN identificadora e outra RNs controladora.

Na área de agentes autônomo e robótica existe um vasto campo para aplicações no espectro das metodologias da IC tratadas nesta dissertação (sistema dinâmico não-linear do caminhão). As aplicações em hardware, como projeto de circuitos integrados e “embedded control” com a utilização de IC é outra área emergente com resultados promissores (Chiaberge and Reyneri, 1995; Koza *et. al.*, 1996).

A dissertação que agora conclui-se revelou que o estudo da IC constitui-se de um campo com aplicações multidisciplinares, não só em identificação e controle de processos. Assim, sugere-se caminhos alternativos da aplicação da IC tanto na academia quanto no meio industrial, sendo que existem relevantes projetos em desenvolvimento que vinculam a academia a indústria, a citar: ESPRIT (I a IV), na União Européia; projetos do governo japonês (por exemplo: “The Human Frontiers”) e projetos do DARPA (“Defense Advanced Research Projects Agency”), nos Estados Unidos.

O “ESPRIT III: Neural Adaptive Control Technology”, em parceria da Daimler-Benz Systems Technology Research Berlin e o Departamento de Engenharia Mecânica da Universidade de Glasgow, na Escócia, e o recém iniciado “ESPRIT IV”, coordenado pela Universidade Napier (Reino Unido) conjuntamente a 30 grandes centros de gerenciamento (Dinamarca, França, Inglaterra, Suécia, ...) da indústria e academia da União Européia, que apresenta ênfase para estudos da aplicação de computação evolucionária na indústria, visando a obtenção de resultados relevantes a comunidade científica e aprimoramento do meio industrial.

Deve ser enfatizado, que muitas das aplicações em automação industrial que empregam controle clássico e controle inteligente (nebuloso, evolutivo ou neural) estão combinadas de forma a obter-se eficiente interação e complementaridade mais que competitividade. Uma vez que este ponto é mencionado, não existe razão para argüir se controle clássico é melhor do que controle inteligente ou vice-versa.

Referências bibliográficas

- Al-Akhras, M.A., G.M. Aly and R.J. Green (1996). Neural Network Learning Approach of Intelligent Multimodel Controller. *IEE Proc.-Control Theory Appl.*, vol. 143, no. 4, pp.395-400.
- Anguita, D., M. Pampolini, G. Parodi and R. Zunino (1993). YPROP: Yet Another Accelerating Technique for the Back Propagation. *International Conference on Artificial Neural Networks*, Amsterdam, Holland, pp.500.
- Åström, K.J. (1991). Intelligent Control. *European Control Conference*, Grenoble, France, pp.2328-2339.
- Åström, K.J. and T. Hägglund (1996). PID Control. In Levine, W.S. (ed.). *The Control Handbook*. IEEE Press and CRC Press, Inc, pp.198-209.
- Bäck, T. and F. Hoffmeister (1991). Extended Selection Mechanisms in Genetic Algorithms. *Proc. 4th International Conference on Genetic Algorithms and their Applications*, San Diego, USA. In: R.K. Belew and L.B. Booker, Eds., San Mateo, CA: Morgan Kaufman, USA, pp.92-99.
- Bäck, T. and H.-P. Schwefel (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, vol.1, pp.1-23. The MIT Press.
- Belew, R.K., J. McInemey and N.N. Schraudolph (1990). Evolving Networks: Using the Genetic Algorithm with Connectionist Learning. *CSE Technical Report CS90-174*, Univ. California at San Diego, CA, USA.
- Bonissone, P.P. (1994). Fuzzy Logic Controllers: An Industrial Reality. In: J.M. Zurada, R.J. Marks II and C.J. Robinson (eds.). *Computational Intelligence Imitating Life*. IEEE Press, pp.316-327.

- Boverie, S., B. Demaya and A. Titli (1991). Fuzzy Logic Control Compared with Other Automatic Control Approaches. *Proc. of the 30th Conference on Decision and Control*, Brighton, England, pp.1212-1216.
- Buckley, J.J. and Y. Hayashi (1993a). Fuzzy I/O Controllers as Universal Approximators. *World Congress on Neural Networks*, Portland, Oregon, USA, v.II, pp. 92-96.
- Buckley, J.J. and Y. Hayashi (1993b). Numerical Relationships Between Neural Networks, Continuous Functions, and Fuzzy Systems. *Fuzzy Sets and Systems*, vol. 60, pp.1-8.
- Chen, S., S.A. Billings, C.F.N. Cowan and P.M. Grant (1990). Practical Identification of NARMAX Models Using Radial Basis Function. *Int. Journal of Control*, vol.52, no.6, pp. 1327-1350.
- Chiaberge, N. and L.M. Reineri (1995). Cintia: A Neuro-Fuzzy Real-Time Controller for Low-Power Embedded Systems. *IEEE Micro*, pp.40-47.
- Chipperfield, A. and P. Fleming (1996). Evolutionary Algorithms for Control Engineering. *IFAC 13th Triennial World Congress*, San Francisco, USA, pp.181-186.
- Coelho, A.A.R. (1995). Laboratory Experiments for Education in Process Control. *Workshop in Control Education and Technology Transfer Issues*, Curitiba, PR, Brasil, pp.133-138.
- Coelho, L.S. e A.A.R. Coelho (1995). Síntese de um Controlador Nebuloso para Pré-Processamento ou Pós-Processamento de Dados de uma Rede Neural Artificial. *II Congresso Brasileiro de Redes Neurais*. Curitiba, PR, Brasil, p.13-17.
- Coelho, L.S. e A.A.R. Coelho (1996a). Identificação Experimental de Processos de Nível e Térmico via Técnicas de Computação Evolucionária. *7º Congreso Latinoamericano de Control Automático e XV Simposio Nacional de Control Automático*, Buenos Aires, Argentina, p. 76-81.

- Coelho, L.S. e A.A.R. Coelho (1996b). Metodologias Inteligentes para Configuração de Controladores PID. *7º Congresso Latinoamericano de Control Automático e XV Simposio Nacional de Control Automático*, Buenos Aires, Argentina, p.82-85.
- Coelho, L.S. e A.A.R. Coelho (1996c). Identificação de Processos com Estimação do Atraso de Transporte via Algoritmos Evolucionários e MQR. *XIX Congresso Nacional de Matemática Aplicada e Computacional*, Goiânia, GO, Brasil, p.138-139.
- Coelho, L.S. e A.A.R. Coelho (1996d). Metodologias de Inteligência Artificial para Configuração de Controladores PID. *XIX Congresso Nacional de Matemática Aplicada e Computacional*, Goiânia, GO, Brasil, p.140-141.
- Coelho, A.A.R. e L.S. Coelho (1996e). Técnicas de Controle PID em Algoritmos Adaptativos e Inteligentes. *XIX Congresso Nacional de Matemática Aplicada e Computacional*, Goiânia, GO, Brasil, p.144-145.
- Coelho, L.S. e A.A.R. Coelho (1996f). Estratégias de Configuração Evolucionária, Fuzzy e Neural em Algoritmos de Controle PID. *III Simpósio Brasileiro de Redes Neurais*, Recife, PE, Brasil, p. 109-117.
- Coelho, L.S. e A.A.R. Coelho (1996g). Metodologias Inteligentes em Controle de Processos. *Automação & Indústria*, ano VII, nº 64, p. 13-16.
- De Falco, I., R. Del Balio, A. Della Cioppa and E. Tarantino (1996). A Comparative Analysis of Evolutionary Algorithms for Function Optimisation. *2nd On-line Workshop Evolutionary Computation*.
- Dracopoulos, D.C. and A.J. Jones (1995). Neural Networks and Genetic Algorithms for the Attitude Control Problem. *International Workshop on Artificial Neural Networks*, Malaga, Spain.

- Elman, J.L. (1990). Finding Structure in Time. *Cognitive Science*, 14, pp.179-211.
- EvoNews (1996). *Newsletter of the EvoNet Network of Excellence in Evolutionary Computation*. European Commission's ESPRIT IV Programme. Issue 1.
- Fathi-Torbaghan, M. and L. Hildebrand (1994). Evolutionary Strategies for the Optimization of Fuzzy Rules. *5th International Conference Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Paris, France, pp.671-674.
- Filho, J.L.R., P.C. Treleaven and C. Alippi (1994). Genetic Algorithm Programming Environments. *IEEE Computer*, pp.28-43.
- Fogel, D.B. (1994). An Introduction to Simulated Evolutionary Optimization. *IEEE Transactions on Neural Networks*, vol.5, no.1, pp.3-14.
- Fukuda, T. and T. Shibata (1992). Theory and Applications of Neural Networks for Industrial Control Systems. *IEEE Transactions on Industrial Electronics*, vol. 39, no.6, pp. 472-489.
- Ghoshray, S., K.K. Yen and J. Andrian (1995). Modified Genetic Algorithms by Efficient Unification with Simulated Annealing. *International Conference on Artificial Neural Networks and Genetic Algorithms*, Alés, France, pp. 487-490.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company.
- Haykin, S. (1994). *Neural Networks: a Comprehensive Foundation*. IEEE Press.
- Hirota, K. and M. Sugeno (1995). *Industrial Applications of Fuzzy Technology in the World*. World Scientific Publishing Co.

- Hoffmeister, F. and T. Bäck (1991). Genetic Self-Learning. *Proceedings of the 1st European Conference on Artificial Life*, Paris, France, pp.227-235, 1992, MIT Press.
- Hwang, W.R. and Thompson, W.E. (1993). An Intelligent Controller Design Based on Genetic Algorithms. *Proc. 32nd Conference on Decision and Control*, San Antonio, Texas, USA, pp.1266-1267.
- Janikow, C.Z. and Z. Michalewicz (1991). An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms. *4th International Conference on Genetic Algorithms and their Applications*. In: R.K. Belew and L.B. Booker, Eds., San Mateo, CA: Morgan Kauffman, USA, pp.31-35.
- Johansson, R. (1993). *System Modeling and Identification*. Prentice Hall, Inc.
- Katayama, R., Y. Kajitani, K. Kuwata and Y. Nishida (1993). Developing Tools and Methods for Applications for Incorporating Neuro, Fuzzy and Chaos Technology. *Computer Ind. Engng.*, vol.24, no.4, pp.579-592.
- Kirkpatrick, S., C. D. Gelatt and M.P. Vecchi (1983). Optimization by Simulated Annealing, *Science*, vol.220, pp.671-680.
- Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall International, Inc.
- Koza, J.R., D. Andre, F.H. Bennett III, M.A. Keane (1996). Use of Automacally Defined Functions and Architecture-Altering Operations in Automated Circuit Synthesis Using Genetic Programming. *Genetic Programming 1996: Proc. of the First Annual Conference*, Stanford University, Cambridge, MA: The MIT Press.
- Kristinsson, K. and G.A. Dunmont (1992). System Identification and Control Using Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, vol.22, no.5, september/october, 1992.

- Kröhling, R.A. (1994). *Algoritmos de Controle não Convencionais: Estudo de Caso*. Dissertação de mestrado, UFES/EE, Vitória, ES, Brasil.
- Kwok, D.P., P. Tam, C.K. Li and P. Wang (1990). Linguistic PID Controllers. *11th World Congress International Federation of Automatic Control*, Tallinn, Estonia, USSR, vol.7, pp.192-197.
- Lee, C.C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I and Part II. *IEEE Transactions on Systems, Man and Cybernetics*, vol.20, no.2, March/April, pp.404-435.
- Levine, W.S. (ed.) (1996). *The Control Handbook*. IEEE Press and CRC Press, Inc.
- Li, C.J. and Y.C. Jeon (1993). Genetic Algorithm in Identifying Non Linear Auto Regressive with Exogenous Input Models for Non Linear Systems. *Proc. of American Control Conference*, San Francisco, CA, USA, pp. 2305-2309.
- Li, H.X. and H.B. Gatland (1995). Enhanced Methods of Fuzzy Logic Control. *Proc. of the International Joint Conference of the 4th IEEE International Conference on Fuzzy Systems and the 2nd International Fuzzy Engineering Symposium*, Yokohama, Japan, vol.1, pp.331-336.
- Li, Y. and K.C. Ng (1995a). Uniform Approach to Model-Based Fuzzy Control System Design and Structural Optimisation. *Technical Report CSC-95007*, Centre for Systems and Control, University of Glasgow, U.K., to be in *Genetic Algorithms and Soft Computing*, F. Herrera and J.L. Verdegay (eds.), Physica-Verlag Series "Studies in Fuzziness", 1996.
- Li, Y., K.C. Tan and K.C. Ng (1995). Performance Based Linear Control System Design by Genetic Evolution with Simulated Annealing. *Technical Report CSC-95017*, Centre for Systems and Control, University of Glasgow, submitted to IEEE Conference on Decision and Control, New Orleans, LA, USA.

- Linkens, D.A. and H.O. Nyongesa (1995). Genetic Algorithms for Fuzzy Control, Part 2: Online System Development and Application. *IEE Proc.-Control Theory Appl.*, vol.142, no.3, pp. 177-185.
- Lipton, M.J. and H.P. Rosenof (1996). Genetic Algorithms Solve Batch Scheduling Problems, Maintain Flexibility. *I&CS*, May, pp.43-47.
- Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ.
- Ljung, L. (1995). System Identification. *Technical Report 1763*, Department of Electrical Engineering, Linköping University, Sweden.
- Mayr, O. (1970). The Origins of Feedback Control. *Scientific American*, 223, pp.110-118.
- Mendel, J.M. (1995). Fuzzy Logic Systems for Engineering: A Tutorial. *Proc. of the IEEE*, vol. 83, no.3, pp.345-377.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution. Programs*. Springer-Verlag.
- Montiel, L.L., Ph. de Larminat and M. Guglielmi (1987). Application of the Instrumental Variable Method to the Indirect Adaptive Control. *Preprints 10th World Congress Automatic Control*, vol.10, Munich, Germany, pp.115-120.
- Munakata, T. and Y. Jani (1994). Fuzzy Systems: An Overview. *Communications of the ACM*, vol.37, no.3, pp.69-76.
- Narendra, K.S. and K. Parthasarathy (1990). Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, vol.1, no.1, pp.4-27.

- Narendra, K.S. (1996). Neural Networks for Control: Theory and Practice. *Proc. of the IEEE*, vol.84, no.10, pp.1385-1406.
- Nguyen, D.H. and B. Widrow (1990). Neural Networks for Self-Learning Control Systems. *IEEE Control Systems Magazine*, April, pp.18-23.
- Omatu, S. and S. Akhyar (1994). Self-Tuning PID Control by Neural Networks. *World Congress on Neural Networks*, Portland, Oregon, USA, vol. 3, pp.249-252.
- Pham, D.T. and X. Liu (1992). Dynamic System Modelling Using Partially Recurrent Neural Networks. *Journal of System Engineering*, 2, pp.90-97.
- Passino, K.M. (1996). Intelligent Control. In *The Control Handbook*. IEEE Press and CRC Press, Inc., pp.994-1017.
- Pereira, L.F.A., L.F. Faina, E. Cardoso e T. Yoneyama (1994). Corretor Automático de Parâmetros de Controladores através de Redes Neurais”, X Congresso Brasileiro de Automática, Rio de Janeiro, RJ, Brasil, 1994.
- Pickhardt, R. and H. Unbehauen (1994). An Indirect Adaptive Predictive Controller for Linear and Nonlinear Plants. *Proc. of American Control Conference*, Baltimore, Maryland, USA.
- Psaltis, D., A. Sideris and A.A. Yamamura (1988). A Multi-layered Neural Network Controller. *IEEE Control Systems Mag.*, pp.17-20.
- Ramos, A.R.M. e D.A.C. Barone (1994). Aplicação de Algoritmo Genético no Controle da Trajetória de um Veículo. *I Simpósio Brasileiro de Redes Neurais*, Caxambu, MG, Brasil, pp.147-159.
- RayChaudhuri, T., L.G.C. Hamey and R.D. Bell (1996). From Conventional Control to Autonomous Intelligent Methods. *IEEE Control Systems Mag.*, vol.16, no.5, pp.78-84.

- Romariz, A. (1996). *Introdução às Redes Neurais e suas Aplicações*. Minicurso apresentado à 3ª Semana de Engenharia Elétrica da UnB, Brasília, DF, Brasil.
- Rudolph, G. (1992). On Correlated Mutations in Evolution Strategies. In: R. Männer and B. Manderick (eds), *Parallel Problem Solving From Nature 2*, The Netherlands, Elsevier Science Publishers, pp.105-114.
- Salgado, M.E., G.C. Goodwin and R.H. Middleton (1988). Modified Least Squares Algorithm Incorporating Exponential Resetting and Forgetting. *Int. J. Control*, vol.47, no.2, pp.477-491.
- Saerens, M. and A. Soquet (1991). Neural Controller Based on Back-propagation Algorithm. *IEE Proceedings-F*, vol.138, no.1, pp.55-62.
- Schoenauer, M. and E. Ronald (1994). Neuro-Genetic Truck Backer-Upper Controller. *First Int. Conference on Evolutionary Computation*, Orlando, USA, IEEE Press.
- Seborg, D.E., Edgar, T.F. and Shah, S.L. (1986). Adaptive Control Strategies for Process Control: A survey. *AIChE Journal*, vol.32, no.6, pp. 881-913.
- Simpson, P.K. (ed.) (1995). *Neural Networks: Theory, Technology, and Applications*. Selected Conference Papers. IEEE Technology Update Series, IEEE Press.
- Sjöberg, J., H. Hjararsson and L. Ljung (1994). Neural Networks in System Identification. *Preprint 10th IFAC Symposium on System Identification*, vol.2, Copenhagen, Denmark, pp. 187-190.
- Sjöberg, J. (1995). *Nonlinear System Identification with Neural Networks*. PhD Thesis, Linköping University, Dept. of Electrical Engineering, S-581 83, Linköping, Sweden.

- Soucek, B. and The IRIS Group (1992). *Dynamic, Genetic, and Chaotic Programming: the Sixth Generation*. John Wiley & Sons, Inc.
- Srinivas, M. and L.M. Patnaik (1994). Genetic Algorithms: A Survey. *IEEE Computer*, pp.17-26.
- Tanomaru, J. (1995). Motivação, Fundamentos e Aplicações de Algoritmos Genéticos. *II Congresso Brasileiro de Redes Neurais*. Curitiba, PR, Brasil, p.373-406.
- Tanscheit, R. (1992). Controle Nebuloso. *9º Congresso Brasileiro de Automática*, Vitória, ES, Brasil, pp.82-95.
- Thomas, D.E. and B. Armstrong-Hélouvry (1995). Fuzzy Logic Control - A Taxonomy of Demonstrated Benefits. *Proceedings of the IEEE*, vol. 83, no.3, pp.407-421.
- Uesaka, Y. (1995). Convergence of Algorithm and the Schema Theorem in Genetic Algorithms. *International Conference on Artificial Neural Networks and Genetic Algorithms*, Alés, France, pp.211-213.
- Watkin, T. and A. Rau (1996). Patenting Neural Networks - An Introduction. *Neural Computing & Applications*, vol.4, no.1, pp. 58-62.
- Wellstead, P.E. and M.B. Zarrop (1991). *Self-tuning Systems: Control and Signal Processing*. John Wiley & Sons, Inc.
- Whitley, D., S. Dominic and R. Das (1991). Genetic Reinforcement Learning with Multilayer Neural Networks. *Proc. 4th International Conference on Genetic Algorithms and their Applications*, San Diego, USA. In: R.K. Belew and L.B. Booker, Eds., San Mateo, CA: Morgan Kauffman, USA, pp.562-569.
- Widrow, B., D.E. Rumelhart and M.A. Lehr (1994). Neural Networks: Applications in Industry, Business and Science. *Communications of the ACM*, vol.37, no.3, pp.93-105.

- Ximenes, F.B. (trad.) (1993). *Microsoft Press Dicionário de Informática Inglês-Português e Português-Inglês*. Editora Campus, Rio de Janeiro, RJ, Brasil.
- Yager, R.R. and D.P. Filev (1994). *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons, Inc.
- Zhao, M. (1994). Neural-Net-Based Adaptive PID Regulator with Attenuating Excitation Signal. *Proc. American Control Conference*, Baltimore, Maryland, USA, pp.2931-2932.
- Zuben, F.J.V. e M.L.A. Netto (1994). Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Redes Neurais Recorrentes e Estudo do Comportamento de Convergência do Processo de Treinamento Associado. *X Congresso Brasileiro de Automática*, Rio de Janeiro, RJ, Brasil, pp.1207-1212.