



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



# **Sistema de Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseado em Identificação Paramétrica**

**Diego Rodrigo Cabral Silva**

Orientador: Prof. DSc. Luíz Affonso Henderson Guedes de Oliveira

Co-orientador: Prof. DSc. Adrião Duarte Dória Neto

**Tese de Doutorado** apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de ordem PPgEE: D36  
Natal, RN, dezembro de 2008

Divisão de Serviços Técnicos

Catálogo da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Silva, Diego Rodrigo Cabral.

Sistema de Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseado em Identificação Paramétrica / Diego Rodrigo Cabral Silva - Natal, RN, 2008  
115 p.

Orientador: Luíz Affonso Henderson Guedes de Oliveira

Co-orientador: Adrião Duarte Dória Neto

Tese (Doutorado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Ciência e Engenharia Elétrica e de Computação.

1. Detecção de falhas - Tese. 2. Filtro de Kalman unscented - Tese. 3. Identificação paramétrica - Tese. I. Oliveira, Luiz Affonso Henderson Guedes de. II. Dória Neto, Adrião Duarte. III. Universidade Federal do Rio Grande do Norte. IV. Título.

RN/UF/BCZM

CDU 681.5.09(043.2)

# **Sistema de Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseado em Identificação Paramétrica**

**Diego Rodrigo Cabral Silva**

Tese de Doutorado aprovada em 11 de dezembro de 2008 pela banca examinadora composta pelos seguintes membros:

---

Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira (orientador) .....  
DCA/UFRN

---

Prof. Dr. Adrião Duarte Dória Neto (co-orientador) ..... DCA/UFRN

---

Prof. Dr. André Laurindo Maitelli ..... DCA/UFRN

---

Prof. Dr. Oscar Gabriel Filho ..... UnP

---

Prof. Dr. Roberto Célio Limão de Oliveira ..... UFPA



*Aos meus pais, Jorge e Graça.*



---

# Agradecimentos

---

Ao meu orientador e ao meu co-orientador, professores Affonso e Adrião, sou grato pela orientação.

Aos colegas de pós-graduação, pelas críticas e sugestões.

À minha família pelo apoio durante esta jornada.





---

# Resumo

---

O presente trabalho visa contribuir com a área de detecção e diagnóstico de falhas em sistemas dinâmicos através da proposta de uma nova arquitetura de sistemas de detecção e isolamento de falhas (FDI, *Fault Detection and Isolation*). A arquitetura proposta traz inovações no que se refere à maneira como as grandezas físicas do processo monitorado são relacionadas ao sistema FDI e, em consequência disso, à maneira como as falhas são detectadas, isoladas e classificadas. Uma busca por ferramentas matemáticas capazes de satisfazer os objetivos da arquitetura proposta apontou para o uso do filtro de Kalman e seus derivados EKF (*Extended Kalman Filter*) e UKF (*Unscented Kalman Filter*). O uso do primeiro algoritmo mostra-se eficaz no caso em que o processo monitorado apresenta uma relação linear entre suas grandezas físicas a serem monitoradas e sua saída. Os outros dois, caso a dinâmica seja não linear. Posteriormente, um comparativo entre o EKF e o UKF mostra que o segundo se adequa melhor às necessidades da arquitetura proposta. Os resultados mostrados no final da tese são referentes a plantas lineares e não-lineares, onde se pode observar a eficácia da arquitetura proposta quando a mesma foi aplicada a processos simulados e reais.

**Palavras-chave:** Detecção de Falhas; Filtro de Kalman Unscented; Identificação Paramétrica.



---

# Abstract

---

The present research aims at contributing to the area of detection and diagnosis of failure through the proposal of a new system architecture of detection and isolation of failures (FDI, *Fault Detection and Isolation*). The proposed architecture presents innovations related to the way the physical values monitored are linked to the FDI system and, as a consequence, the way the failures are detected, isolated and classified. A search for mathematical tools able to satisfy the objectives of the proposed architecture has pointed at the use of the Kalman Filter and its derivatives EKF (*Extended Kalman Filter*) and UKF (*Unscented Kalman Filter*). The use of the first one is efficient when the monitored process presents a linear relation among its physical values to be monitored and its output. The other two are proficient in case this dynamics is no-linear. After that, a short comparative of features and abilities in the context of failure detection concludes that the UKF system is a better alternative than the EKF one to compose the architecture of the FDI system proposed in case of processes of no-linear dynamics. The results shown in the end of the research refer to the linear and no-linear industrial processes. The efficiency of the proposed architecture may be observed since it has been applied to simulated and real processes. To conclude, the contributions of this thesis are found in the end of the text.

**Keywords:** Failure Detection; Unscented Kalman Filter; Parametrical Identification.



---

# Sumário

---

<b>Sumário</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Contribuições . . . . .	3
1.4 Organização do Documento . . . . .	4
<b>2 Detecção e Isolamento de Falhas</b>	<b>5</b>
2.1 Supervisão de Processos Industriais . . . . .	5
2.1.1 Tarefas Básicas da Supervisão . . . . .	6
2.1.2 Faltas, Falhas e Mau Funcionamentos . . . . .	7
2.1.3 Confiabilidade, Disponibilidade e Segurança . . . . .	8
2.1.4 Tolerância a Faltas e Redundância . . . . .	8
2.1.5 Tipos de Falhas . . . . .	9
2.1.6 Etapas do Processo de Detecção e Diagnóstico de Falhas . . . . .	10
2.2 Métodos de Detecção de Falhas . . . . .	11
2.2.1 Detecção de Faltas com Verificação de Limites . . . . .	11
2.2.2 Detecção de Falhas com Equações de Paridade . . . . .	12
2.2.3 Modelos de Processos e Modelagem das Falhas . . . . .	12
2.2.4 Detecção de Falhas com Métodos de Identificação de Processos . . . . .	14
2.2.5 Detecção de Faltas com Observadores de Estado e Estimadores de Estado . . . . .	15
2.2.6 Detecção de Faltas Através da Análise da Assinatura de Sinais . . . . .	15
2.2.7 Outros Métodos de Detecção de Falhas . . . . .	15
2.3 Métodos de Diagnóstico de Falhas . . . . .	17
<b>3 Identificação de Sistemas</b>	<b>21</b>
3.1 Introdução . . . . .	21
3.1.1 Os Passos de um Processo de Identificação de Sistemas . . . . .	22
3.2 Representações Lineares . . . . .	22
3.2.1 Funções de Transferência . . . . .	23

3.2.2	Resposta Temporal . . . . .	23
3.2.3	Espaço de Estados . . . . .	23
3.2.4	Representações em Tempo Discreto . . . . .	24
3.3	Representações Não-lineares . . . . .	26
3.3.1	A Série de Volterra . . . . .	26
3.3.2	Modelos de Hammerstein e de Wiener . . . . .	27
3.3.3	Representações NARX . . . . .	27
3.3.4	Modelos Polinomiais Contínuos . . . . .	28
3.3.5	Redes Neurais Artificiais . . . . .	29
3.3.6	Funções Radiais de Base . . . . .	29
3.3.7	O Neurônio <i>neuro-fuzzy</i> . . . . .	30
3.4	Algoritmos de Identificação . . . . .	30
3.5	O Filtro de Kalman . . . . .	31
3.5.1	O Filtro de Kalman Discreto . . . . .	32
3.6	O Filtro de Kalman Estendido . . . . .	36
3.7	O Filtro de Kalman <i>Unscented</i> . . . . .	40
3.7.1	A transformação <i>unscented</i> . . . . .	40
3.8	Validação de Modelos . . . . .	43
3.8.1	Simulação . . . . .	43
3.8.2	Análise de Resíduos . . . . .	44
3.8.3	Validação para Aplicações em Malha Fechada . . . . .	44
<b>4</b>	<b>Proposta do Trabalho</b>	<b>45</b>
4.1	Metodologia Utilizada . . . . .	48
4.2	Síntese da Arquitetura . . . . .	50
<b>5</b>	<b>Resultados</b>	<b>53</b>
5.1	EKF <i>versus</i> UKF . . . . .	53
5.2	Arquitetura Clássica <i>versus</i> Arquitetura Proposta . . . . .	58
5.2.1	Resultados com a Arquitetura Clássica . . . . .	62
5.2.2	Resultados com a Arquitetura Proposta . . . . .	65
5.2.3	Análise dos Resultados . . . . .	68
5.3	Simulação de Falhas com <i>Threshold</i> Variável . . . . .	70
5.4	Processos Dinâmicos Reais . . . . .	72
5.4.1	Um Motor DC . . . . .	73
5.4.2	O Motor com Parâmetros Físicos . . . . .	78
<b>6</b>	<b>Conclusões</b>	<b>87</b>
6.1	Trabalhos Futuros . . . . .	89
6.2	Publicações . . . . .	89
	<b>Referências bibliográficas</b>	<b>91</b>

---

# Lista de Figuras

---

2.1	<i>Esquema das conexões entre um processo, a supervisão do mesmo e um sistema FDI.</i>	6
2.2	<i>Mapa de conceitos relacionados a faltas e falhas.</i>	8
2.3	<i>Sistema tolerante a faltas com módulos redundantes em paralelo.</i>	9
2.4	<i>Classificação de falhas quanto a localização (atuador, processo ou sensor).</i>	10
2.5	<i>Etapas de um sistema FDI em funcionamento.</i>	10
2.6	<i>Verificação do sinal <math>Y(t)</math> entre os limites inferior <math>Y_{min}</math> e superior <math>Y_{max}</math>.</i>	12
2.7	<i>Esquema de blocos representando o método de detecção de falhas por equações de paridade.</i>	13
2.8	<i>Esquema de detecção de falhas baseado em modelos.</i>	13
2.9	<i>Esquema de blocos representando o método de detecção e classificação de falhas através de identificação de sistemas.</i>	14
2.10	<i>Representação da detecção de falhas por análise de sinais.</i>	16
2.11	<i>Resumo das tecnologias usadas na detecção de faltas em sistemas dinâmicos.</i>	16
2.12	<i>Esquema geral do diagnóstico de faltas.</i>	17
2.13	<i>Relacionamento falta-sintoma.</i>	18
2.14	<i>Métodos de diagnóstico de faltas.</i>	18
3.1	<i>Representação esquemática do modelo ARX.</i>	25
3.2	<i>Representação esquemática do modelo ARMAX.</i>	26
3.3	<i>(a) modelo de Hammerstein, (b) modelo de Wiener.</i>	27
3.4	<i>Topologia do neurônio neuro-fuzzy.</i>	30
3.5	<i>Funções de pertinência do neurônio neuro-fuzzy.</i>	31
3.6	<i>Ciclo de predição-correção do algoritmo do filtro de Kalman discreto.</i>	34
3.7	<i>Uma ilustração completa da operação do filtro de Kalman, combinando o diagrama da Figura 3.6 com as equações das Tabelas 3.1 e 3.2.</i>	35
3.8	<i>Uma ilustração completa da operação do EKF, combinando o diagrama da Figura 3.6 com as equações das Tabelas 3.3 e 3.4.</i>	39
4.1	<i>Arquitetura clássica de sistemas de detecção de falhas.</i>	45
4.2	<i>Arquitetura da ferramenta proposta.</i>	47
5.1	<i>Estimativa de parâmetros com EKF.</i>	54
5.2	<i>Estimativa de parâmetros pelo algoritmo UKF.</i>	55
5.3	<i>Estimativa de parâmetros pelo EKF após uma falha abrupta.</i>	56
5.4	<i>Estimativa de parâmetros pelo UKF após uma falha abrupta.</i>	56

5.5	<i>Estimativa de parâmetros pelo EKF após uma falha incipiente.</i>	57
5.6	<i>Estimativa de parâmetros pelo UKF após uma falha incipiente.</i>	57
5.7	<i>Processo com dois tanques em cascata.</i>	59
5.8	<i>Rede Neural utilizada na identificação do tanque 1.</i>	60
5.9	<i>Rede Neural utilizada na identificação do tanque 2.</i>	61
5.10	<i>Rede Neural utilizada na classificação de falhas.</i>	61
5.11	<i>Identificação do sistema utilizando redes neurais recorrentes.</i>	62
5.12	<i>Erro da identificação por usar a entrada do processo em ponto de operação diferente.</i>	63
5.13	<i>Detecção da falha 1 da Tabela 5.2 pela arquitetura clássica.</i>	64
5.14	<i>Detecção da falha 1 da Tabela 5.2 pela arquitetura clássica em um ponto de operação diferente do treinamento da identificação.</i>	65
5.15	<i>Detecção da falha 4 da Tabela 5.2 pela arquitetura clássica.</i>	65
5.16	<i>Detecção da falha 4 da Tabela 5.2 pela arquitetura clássica em um ponto de operação diferente do treinamento da identificação.</i>	66
5.17	<i>Detecção da falha 1 da Tabela 5.2 pela arquitetura proposta.</i>	67
5.18	<i>Detecção da falha 1 da Tabela 5.2 pela arquitetura proposta em um ponto de operação diferente ao caso da Figura 5.17.</i>	67
5.19	<i>Detecção da falha 4 da Tabela 5.2 pela arquitetura proposta.</i>	68
5.20	<i>Detecção da falha 4 da Tabela 5.2 pela arquitetura proposta em um ponto de operação diferente ao caso da Figura 5.19.</i>	68
5.21	<i>Tanque com parâmetro variável em função do ponto de operação.</i>	70
5.22	<i>Parâmetro C em funcionamento normal.</i>	71
5.23	<i>Parâmetro C refletindo uma falha ocorrida no processo.</i>	72
5.24	<i>Circuito elétrico do motor DC.</i>	73
5.25	<i>Identificação paramétrica do motor discretizado usando O filtro de Kalman.</i>	75
5.26	<i>Falha abrupta na resistência do motor.</i>	75
5.27	<i>Falha incipiente na resistência do motor.</i>	76
5.28	<i>Falha intermitente na resistência do motor.</i>	77
5.29	<i>Outra falha intermitente na resistência do motor.</i>	77
5.30	<i>Identificação paramétrica do motor discretizado usando UKF.</i>	79
5.31	<i>Detalhe do começo da execução do algoritmo no processo de identificação paramétrica do processo descrito pela equação 5.8.</i>	79
5.32	<i>Superfície de erro relacionando os parâmetros físicos reais do motor <math>k_m</math> e R com a saída <math>\omega</math> do mesmo.</i>	80
5.33	<i>Falha abrupta gerada no parâmetro <math>k_m</math> da Tabela 5.4.</i>	81
5.34	<i>Visão detalhada da falha abrupta em <math>k_m</math>.</i>	81
5.35	<i>Falha abrupta gerada no parâmetro R da Tabela 5.4.</i>	82
5.36	<i>Visão detalhada da falha abrupta em R.</i>	83
5.37	<i>Falhas abruptas simultâneas.</i>	83
5.38	<i>Detalhe do parâmetro <math>k_m</math> em situação de falhas abruptas simultâneas.</i>	84
5.39	<i>Falha incipiente na constante do motor <math>k_m</math>.</i>	84
5.40	<i>Falha intermitente refletida constante do motor <math>K_m</math>.</i>	85



5.41	<i>Visualização detalhada do momento em que a estimativa do parâmetro <math>k_m</math> é não confiável.</i>	86
------	---	----



---

# Lista de Tabelas

---

2.1	<i>Exemplos de proteção automática.</i>	6
3.1	<i>Equações de predição do filtro de Kalman discreto.</i>	34
3.2	<i>Equações de correção do filtro de Kalman discreto.</i>	35
3.3	<i>Equações de predição do EKF.</i>	39
3.4	<i>Equações de correção do EKF.</i>	39
5.1	Parâmetros da Planta	59
5.2	Falhas e assinatura de resíduos.	62
5.3	Parâmetros da Planta	71
5.4	Parâmetros da Planta	73



---

# Capítulo 1

## Introdução

---

A demanda sempre crescente por eficiência, qualidade dos produtos, redução de custos e a integração dos vários sistemas de controle no setor industrial, aliada aos preços elevados envolvidos em seus processos e à inquestionável necessidade de segurança, justificam a importância da supervisão (ou monitoração) e dos sistemas de detecção e diagnóstico de falhas envolvendo tais processos [Isermann 2006]. A maneira clássica de supervisionar um processo, ou seja, checar se determinadas variáveis ultrapassaram seus respectivos limites, pode ser substituída por sistemas mais elaborados baseados em várias técnicas modernas provenientes das engenharias e da inteligência computacional [Russell 2003], por exemplo, com o intuito de manter sempre os sistemas em funcionamento com todos os seus parâmetros sob controle.

Esta tese propõe uma nova arquitetura de sistemas de detecção e isolamento de falhas (FDI) que apresenta contribuições em relação as já utilizadas e encontradas na literatura.

### 1.1 Motivação

A partir de 1960 [Isermann 2006] a automação industrial, aplicada à operação e projeto de processos técnicos, aumentou progressivamente influenciada pelos motivos já citados anteriormente. Algumas conseqüências como o funcionamento de processos sem a necessidade da presença de operadores (humanos) e a liberação desses da obrigação de realizar tarefas repetitivas e monótonas, foram valorizadas principalmente por causa da redução dos custos. O grau de automação envolvido nos processos industriais aumentou drasticamente a partir de 1975 [Isermann 2006] em conseqüência da disponibilidade dos microcomputadores, que eram capazes de resolver vários problemas da automação. Essa evolução foi seguida pelo avanço na área de sensores, atuadores, barramentos de comunicação e interfaces máquina-usuário. O avanço no entendimento da teoria existente nos processos e funções da automação também deu a sua contribuição para tal fato.

Toda a tecnologia envolvida nos processos industriais, nas linhas de montagem e de produção e até mesmo nos outros setores das indústrias atuais resulta em um ambiente cada vez mais preciso, eficiente, confiável e seguro, quando comparados aos mesmos encontrados no passado. Além disso, o uso das tecnologias originadas na eletrônica digital e na informática tornam possível o monitoramento dos produtos desde a sua fase inicial de produção, no chão de fábrica, até o seu contato com o mercado, o que possibilita um maior

controle de tudo o que acontece no processo de fabricação de produtos industrializáveis dos mais diferentes setores.

Mesmo com todo esse aparato tecnológico, motivos para continuar aumentando o grau de automação na supervisão dos processos ainda sobram. Além da concorrência cada vez mais acirrada em diversos setores da indústria, existem ainda as leis ambientais altamente rigorosas que prevêm multas para danos à natureza, além do desperdício de produto e de possíveis danos causados a equipamentos. Todos esses motivos, quando sintetizados, indicam perdas financeiras indesejáveis para qualquer indústria.

Além do aumento do controle e da supervisão, a tecnologia envolvida cria uma infraestrutura que disponibiliza informações detalhadas sobre os processos. Tais informações podem ser utilizadas em sistemas auxiliares que têm o intuito de detectar possíveis falhas em alguma fase desses processos.

É nessa área, de detecção e isolamento de falhas, que este trabalho vem a contribuir. Considerando o interesse em descobrir, em tempo hábil, possíveis problemas nos processos industriais, deseja-se criar uma ferramenta de auxílio à supervisão que traga contribuições significativas ao ambiente industrial, aumentando a eficiência e a precisão no que se trata de detecção e isolamento de falhas. O que pode vir a evitar perdas bastante consideráveis, como: paralisação de processos, danos a equipamentos de custo elevado e de fundamental importância, acidentes envolvendo vidas humanas, etc. O que se traduz, entre outras coisas, em perdas financeiras que podem chegar a valores bastante elevados, dependendo do tamanho e do setor da indústria em questão.

Os sistemas FDI (*Fault Detection and Isolation*) são geralmente implementados em duas etapas (ver Capítulo 2). A primeira delas é a detecção da falta ou falha<sup>1</sup>, ou seja, a percepção de que algo está errado e pode estar levando o processo a algum desvio mais significativo do seu funcionamento normal. Neste ponto ainda não se sabe a localização ou o motivo que ocasionou a falta. E se conclui que existe alguma anormalidade através da comparação entre valores provenientes da planta física e valores estimados, geralmente por modelos matemáticos do processo [Venkatasubramanian et al. 2003].

A segunda fase do sistema FDI é o isolamento da falta, ou seja, a identificação do componente ou local físico dentro do sistema onde se encontra o problema detectado no passo anterior. Geralmente, para a fase de identificação, os sistemas FDI fazem uso de um classificador de faltas (ver Capítulo 2). Este, por sua vez, analisa os sinais provenientes da etapa anterior, também chamados de *resíduos* para chegar a alguma conclusão.

No presente trabalho, encontra-se uma abordagem diferente da descrita acima, onde a etapa de classificação de faltas ocorre de maneira simples e direta, podendo ser executada até pelo operador do sistema. Para isso, o sistema se baseia nos sinais de entrada e saída do processo real, além do seu modelo matemático, que servirá de base para a identificação das possíveis anormalidades.

A proposta detalhada da arquitetura do sistema FDI proposto pode ser encontrada no Capítulo 4. Esta faz uso de algoritmos recursivos de identificação de sistemas. Esses algoritmos são o Filtro de Kalman e seus derivados, Filtro de Kalman Estendido (EKF) e Filtro de Kalman *Unscented* (UKF), utilizados aqui como estimadores de parâmetros.

Com isso, visa-se colaborar com a área de detecção e isolamento de falhas apresen-

---

<sup>1</sup>Os conceitos detalhados de faltas e falhas serão definidos no Capítulo 2

tando a abordagem acima descrita e suas respectivas contribuições. Essas, por sua vez, estão destacadas na Seção 1.3.

## 1.2 Objetivos

Em termos científicos, o principal objetivo desta tese é propor e analisar o desempenho de técnicas de identificação paramétrica de modelos não-lineares aplicados a uma nova arquitetura de sistemas de detecção de falhas.

Em termos tecnológicos, o objetivo desse trabalho é contribuir para a área de detecção automática de falhas em sistemas dinâmicos, através da proposta de uma nova arquitetura de sistemas FDI. Esta, por sua vez, trará benefícios que podem vir a complementar as ferramentas que hoje atuam nos parques industriais.

## 1.3 Contribuições

Entre as contribuições desse trabalho podem-se citar:

- além de identificar e isolar faltas e falhas em processos industriais, o que é, neste caso, apontado pelas estimativas dos parâmetros, a presente arquitetura é capaz de quantificar a falha, relacionando-a diretamente a uma grandeza física do processo em questão. Ou seja, ela é capaz de encontrar os parâmetros da equação do processo em modo de falha;
- a arquitetura também apresenta a capacidade de detectar múltiplas falhas simultâneas, quantificando-as e relacionando-as com grandezas físicas do sistema real, mesmo que as estas tenham sido originadas por razões descorrelacionadas;
- possibilidade da monitoração da formação de falhas incipientes<sup>2</sup> desde o começo da formação das mesmas, através da estimativa dos parâmetros em tempo de execução, ou seja, mesmo pequenas variações da relação entrada/saída do processo que refletirem na alteração de algum parâmetro monitorado, serão detectadas de forma *on-line* ao ocorrido;
- capacidade de aplicação em ambientes reais, considerando ruído. Como a solução aqui proposta se baseia nos algoritmos estocásticos de estimação, uma consequência natural e direta é a capacidade de seu funcionamento mesmo diante da presença de ruído
- uma outra consequência do uso dos algoritmos derivados do Filtro de Kalman original, o EKF e o UKF, é a capacidade de estimar parâmetros mesmo quando a relação entre esses e a saída do processo obedece uma dinâmica não-linear e variante no tempo;
- considerar a tarefa de detecção e isolamento de falhas como um problema de identificação paramétrica de sistemas variantes no tempo, cujas dinâmicas possivelmente têm relações não-lineares entre os parâmetros e o sinal de saída. Isto se deve ao fato

---

<sup>2</sup>Falhas incipientes são aquelas que começam lentamente, desviando o processo do seu comportamento normal pouco a pouco, às vezes, mascaradas pelo controlador do sistema.

de pretender-se, neste trabalho, estimar os parâmetros de um modelo fenomenológico, e não de um modelo tipo caixa preta (ver Capítulo 3). No caso, falhas serão modeladas como variações bruscas (falhas abruptas) e/ou suaves (falhas incipientes) (ver Capítulo 2) no modelo fenomenológico.

## 1.4 Organização do Documento

No próximo capítulo encontra-se uma panorâmica da área de detecção e isolamento de falhas em sistemas dinâmicos, os principais conceitos e a teoria envolvida, além das tecnologias que já foram utilizadas no projeto e implementação dos mesmos. No Capítulo 3, encontra-se a compilação de várias publicações na área de identificação de sistemas, o que é parte fundamental para a construção de sistemas FDI. No Capítulo 4 é detalhada a proposta do trabalho, fazendo referência às teorias apresentadas nos capítulos anteriores. No Capítulo 5 são mostrados os resultados obtidos, utilizando-se vários experimentos, entre eles, alguns experimentos com plantas reais, onde se encontram algumas dificuldades que inexistem em ambientes simulados. E, finalmente, no Capítulo 6 encontram-se as conclusões do trabalho juntamente com algumas considerações para trabalhos futuros.



---

## Capítulo 2

# Detecção e Isolamento de Falhas

---

Os sistemas de detecção e isolamento de falhas, ou apenas sistemas FDI (*Fault Detection and Isolation*) como são comumente denominados na literatura, começaram a ser introduzidos nas plantas industriais por volta da década de 1970 [Witczak 2005], quando começou uma movimentação por parte dos engenheiros em prol do combate ao desperdício de produtos, aos desastres naturais, e ao elevado número de acidentes envolvendo vidas humanas.

Os primeiros sistemas FDI eram bastante simples [e Antonio Pietrosanto 2000], porém já traziam grandes ganhos à segurança das pessoas envolvidas no processo de produção. Com o passar dos anos, equipamentos eletrônicos como processadores e memórias ficaram cada vez mais acessíveis, permitindo-se a ampliação de seu uso no cenário da indústria, o que refletiu também, no avanço dos sistemas FDI.

Atualmente, o nível de complexidade envolvido em um processo industrial exige sistemas de supervisão e proteção tão avançados quanto a engenharia consegue oferecer, demandando um esforço contínuo por parte dos pesquisadores e engenheiros que dedicam seu tempo de trabalho ao desenvolvimento de novas tecnologias que vêm contribuir com a melhora do desempenho dos sistemas FDI.

Este capítulo está organizado da seguinte forma: a seguir, encontra-se uma explanação sobre a supervisão de processos, seguido da definição de alguns conceitos básicos da área de detecção de falhas, necessários para o entendimento do presente trabalho. Mais adiante são mostradas várias ferramentas usadas para a detecção de falhas em sistemas dinâmicos e, para finalizar, mostram-se algumas técnicas utilizadas no diagnóstico das mesmas.

### 2.1 Supervisão de Processos Industriais

A supervisão de um processo industrial tem o objetivo de mostrar o presente estado do mesmo, incluindo os estados não desejados e não permitidos. Além disso, a supervisão tem como uma de suas principais funções atuar de forma a evitar danos ou acidentes. Os desvios do comportamento normal do processo são atribuídos a distúrbios ou erros que podem ser causados por várias razões. Eles podem resultar em períodos curtos ou longos de mau funcionamento do sistema ou até mesmo falhas, caso nenhuma contra-ação seja tomada. Uma outra função da supervisão é evitar quaisquer mau funcionamentos ou falhas. A Figura 2.1 mostra a organização, através de um esquema de blocos, de um

processo industrial supervisionado e com um sistema FDI acoplado ao mesmo.

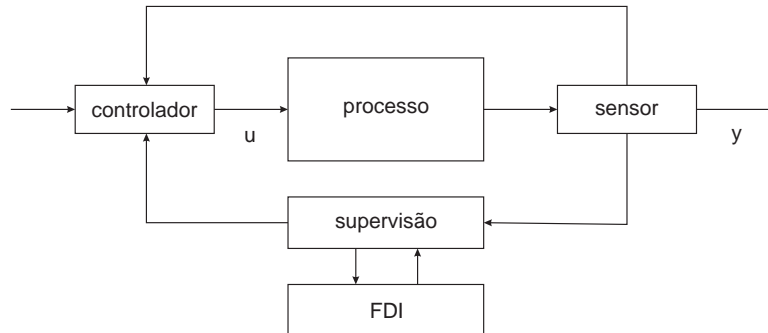


Figura 2.1: Esquema das conexões entre um processo, a supervisão do mesmo e um sistema FDI.

A seguir serão brevemente descritas as tarefas básicas da supervisão e, na seqüência, alguns termos usados nessa área serão apresentados [Isermann 2006].

### 2.1.1 Tarefas Básicas da Supervisão

As principais tarefas desempenhadas por um sistema de supervisão, são mostradas abaixo:

1. *Monitoramento*: As variáveis mensuráveis são verificadas considerando-se suas tolerâncias e, então, gerando-se alarmes ao operador caso algum limite seja ultrapassado. Depois que um alarme é disparado, o operador pode tomar alguma decisão apropriada.
2. *Proteção automática*: em caso de o processo ir para um estado perigoso, a função de monitoramento automaticamente inicia uma contra-ação apropriada. Usualmente, o processo é conduzido a um estado de segurança, que é freqüentemente um desligamento de emergência. A Tabela 2.1 mostra alguns exemplos de proteção automática.

Tabela 2.1: Exemplos de proteção automática.

Processo	Falta	Contra-ação
cabo elétrico	curto circuito	interrupção da corrente
motor elétrico	superaquecimento	interrupção da corrente
turbina a vapor	alta velocidade	fechamento de válvulas
caldeira	superaquecimento	interrupção de combustível

3. *Supervisão com diagnóstico de falhas*: geração de atributos (através de processamento de sinais, estimação de estados, identificação de sistemas, entre outros), detecção de falhas (através da análise dos sintomas), avaliação de falhas, classificando-as em diferentes classes de periculosidade e, finalmente, tomadas de decisões, dependendo do possível grau de perigo da falha em questão.

4. *Ações de supervisão e gerenciamento de faltas*: Dependendo da avaliação das faltas diagnosticadas, as seguintes ações podem ser tomadas: *operação segura* (desligar se existir um risco iminente ao processo ou ao ambiente), *operação de confiança* (mudanças no ponto de operação, carga, velocidade, pressão ou temperatura mais baixos), *reconfiguração* (manter o processo sob controle usando sensores, atuadores e componentes redundantes), *inspeção* (executar um diagnóstico detalhado com métodos adicionais), *manutenção* (alterar parâmetros de sistemas ou partes desgastadas o mais rápido possível), *reparação* (remover a falha assim que possível).

A seguir encontram-se alguns conceitos importantes para o entendimento e familiarização com a área em questão [Witczak 2005].

### 2.1.2 Faltas, Falhas e Mau Funcionamentos

Nesta seção encontra-se alguns dos principais conceitos utilizados na área de detecção e isolamento de falhas. As definições a seguir podem ser encontradas em [Isermann 2006].

#### **Falta:**

“Uma falta é um desvio não permitido de pelo menos uma característica do sistema a partir do aceitável, usual, ou condição padrão.”

#### *Observações:*

- uma falta é um estado dentro do sistema;
- o desvio não permitido é a diferença entre o valor da falta e o limite violado da zona de tolerância para seu valor usual.

#### **Falha:**

“Uma falha é uma interrupção permanente da habilidade do sistema em executar uma determinada função sob certas condições de operação.”

#### *Observações:*

- uma falha é a terminação da habilidade de uma unidade funcional de executar uma função requerida;
- uma falha resulta de uma ou mais faltas.

#### **Mau funcionamento:**

“Um mau funcionamento é uma irregularidade intermitente na execução da função desejada do sistema”

A Figura 2.2 mostra graficamente alguns desses conceitos.

Na literatura internacional [Isermann 2006], o termo “falta” é utilizado para designar um leve desvio do funcionamento normal do processo, mas que permita que o mesmo ainda realize as funções para o qual foi projetado. Já o termo “falha” é utilizado para interrupções permanentes, ou seja, que impedem o sistema de executar suas tarefas.

Neste trabalho, considera-se faltas e falhas sendo situações indesejadas, independente da capacidade do processo poder ou não executar suas funções. Visto que o sistema FDI aqui proposto, pode quantificar a falta ou falha, através da estimação dos parâmetros em modo de falha, e estas podem ser interpretadas pelo operador ou pelo sistema de recuperação da maneira mais apropriada para cada processo real. Ou seja, o limiar entre falta e falha pode ser definido para cada processo em que o sistema FDI for aplicado.

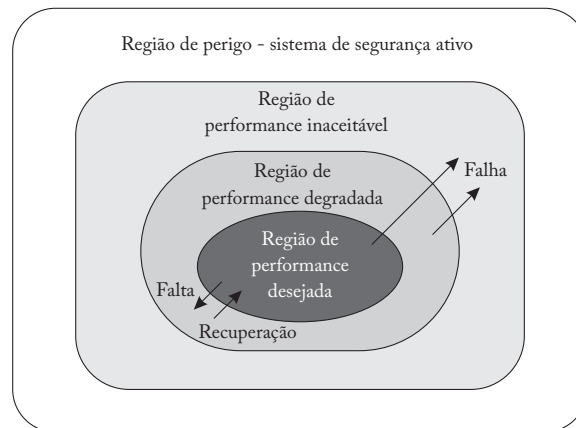


Figura 2.2: Mapa de conceitos relacionados a falhas e falhas.

### 2.1.3 Confiabilidade, Disponibilidade e Segurança

#### **Confiabilidade:**

“Habilidade de um sistema de executar uma função requerida sob certas condições, dentro de um dado escopo, durante um dado período de tempo.”

#### *Observações:*

- pode ser resumidamente definida como a habilidade de executar uma função requerida por um certo período de tempo;
- a confiabilidade pode ser afetada por mau funcionamentos e falhas.

#### **Segurança:**

“Habilidade de um sistema de não causar perigo às pessoas envolvidas, aos equipamentos ou ao ambiente.”

#### *Observações:*

- resumidamente, pode ser definida como a habilidade de não causar perigo;
- a segurança é baseada nos efeitos perigosos das faltas, falhas e mau funcionamentos.

#### **Disponibilidade:**

“Probabilidade de um sistema ou equipamento operar satisfatoriamente e efetivamente em qualquer período de tempo.”

#### *Observações:*

- quanto maior a disponibilidade, maior será a satisfação do usuário de um sistema;
- disponibilidade leva em consideração que falhas e mau funcionamentos acontecem e levam tempo para serem reparados.

### 2.1.4 Tolerância a Faltas e Redundância

Mesmo depois de aplicadas a análise de confiança e segurança para a melhoria do projeto, testes e controle de qualidade, sabe-se que o aparecimento de certas faltas e falhas não foram totalmente evitados. Conseqüentemente, essas faltas inevitáveis devem

ser tolerados por esforços adicionais do projeto. O que quer dizer que sistemas de alta integridade devem apresentar a habilidade da tolerância a falhas<sup>1</sup>. Isso significa que falhas são compensadas de maneira que não cheguem a se tornar falhas. Uma das maneiras mais usadas de se garantir isso é usando redundância de equipamentos, geralmente em paralelo, como mostra a Figura 2.3.

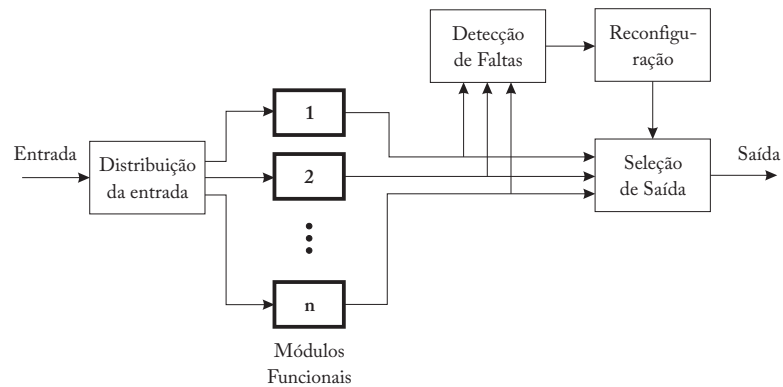


Figura 2.3: Sistema tolerante a falhas com módulos redundantes em paralelo.

Os módulos funcionais podem ser componentes de *hardware* ou partes de *software*, podendo estes serem idênticos ou diferentes. Existem vários esquemas de sistemas tolerantes a falhas, com redundância dinâmica ou estática. Em geral, os módulos funcionais são supervisionados com capacidade de detecção de falhas seguidos por um mecanismo de reconfiguração para trocar os módulos com problema pelos módulos redundantes. Estes módulos são geralmente atuadores, sensores, computadores, motores, bombas, etc.

### 2.1.5 Tipos de Falhas

As falhas em um processo industrial podem ser classificadas em relação a vários aspectos, dentre eles estão o tempo e a localização. Em se tratando de tempo, as falhas podem ser abruptas, incipientes ou intermitentes. As definições de cada classificação são encontradas a seguir:

- **falhas abruptas:** são aquelas que surgem repentinamente, podendo ser decorrentes de imprevistos ou até mesmo de acidentes. Essas falhas mudam o comportamento do processo rapidamente, exigindo contra-ações velozes e eficazes que possam minimizar as conseqüências do ocorrido.
- **falhas incipientes:** são falhas iniciadas a partir de pequenos desvios do comportamento normal do sistema. Elas podem ser mascaradas pelos controladores, e muitas vezes passam despercebidas pelos operadores ou até mesmo pelos sistemas FDI.
- **falhas intermitentes:** são aquelas que se alternam no tempo. Podem ser causadas por alguma perturbação periódica, ou por alguma situação que se repita ciclicamente, caracterizando-se um cenário de falha.

<sup>1</sup>Na literatura brasileira, encontra-se o termo “sistemas tolerantes a falhas” em vez de “sistemas tolerantes a falhas”, como encontrado na literatura internacional

Considerando-se a localização, as falhas podem se encontrar no atuador [Lunze & Schroder 2004], no sistema dinâmico (processo) ou no sensor [Mehranbod et al. 2004] e [Lunze & Schroder 2004]. A Figura 2.4 representa a ligação entre esses três componentes que fazem parte de um parque industrial típico.

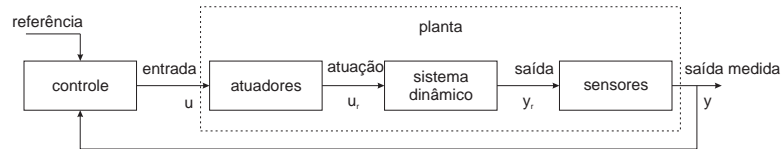


Figura 2.4: Classificação de falhas quanto a localização (atuador, processo ou sensor).

As definições de cada classificação estão a seguir:

- **falhas em atuadores:** podem ser vistas como qualquer mau funcionamento do equipamento que atua no sistema.
- **falhas no sistema dinâmico:** ocorrem quando alguma alteração no sistema muda de alguma forma a relação original de entrada e saída do mesmo.
- **falhas em sensores:** podem ser observadas através de variações específicas nas medidas que possibilitem desconsiderá-las como variações válidas do sistema.

Existem ainda outras classificações mais específicas para falhas. Estas fogem do escopo deste trabalho podendo ser consultadas em outras publicações como [Gertler 1998].

### 2.1.6 Etapas do Processo de Detecção e Diagnóstico de Falhas

A tarefa de monitorar processos industriais desde o seu funcionamento normal até a detecção e o diagnóstico de falhas envolve várias fases. Estas, são mostradas na Figura 2.5.

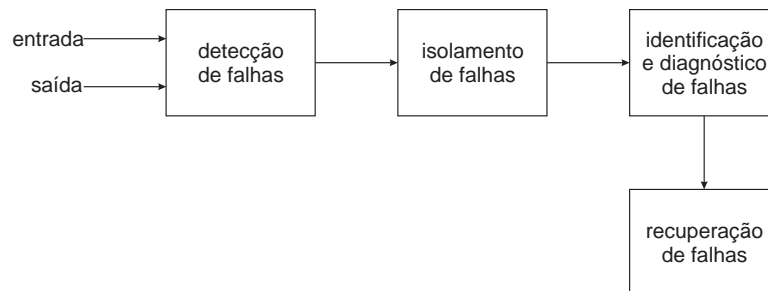


Figura 2.5: Etapas de um sistema FDI em funcionamento.

Pode-se perceber que os sinais de entrada e saída do processo real são passados para o sistema FDI. É através deles que se realiza a monitoração do estado do mesmo. A percepção de que algo fugiu do funcionamento normal refere-se à etapa de detecção de faltas e falhas. Neste momento não se tem muitas informações do problema ocorrido, sabe-se apenas que a relação entre a entrada e a saída do processo físico não representa

o funcionamento normal. A segunda etapa refere-se ao isolamento da falha, ou seja, à descoberta do local físico envolvido na falha. Assim, ele pode ser isolado dos outros componentes do processo. A terceira etapa corresponde à identificação, o que quer dizer a classificação da falha e a amplitude da mesma. Com todas essas informações, a etapa de recuperação procura tomar decisões que minimizem os danos ou que leve o processo para um estado de segurança.

O presente trabalho tem seu foco nas três primeiras etapas ilustradas pela Figura 2.5, não se atendo às ações que podem ser tomadas para recuperar o processo.

A próxima seção apresenta uma explanação sobre as ferramentas utilizadas na etapa de detecção de falhas em sistemas dinâmicos. Ficando para a seção seguinte a apresentação de alguns métodos utilizados na etapa de identificação das falhas.

## 2.2 Métodos de Detecção de Falhas

Diversas são as técnicas já utilizadas na construção de ferramentas com o objetivo de detectar faltas e falhas em processos industriais. Vale destacar que cada autor tem sua visão particular de como categorizar essas técnicas, como em [Lou et al. 2003], [Gertler 1998], [Angeli & Chatzinikolaou 2004] e [Venkatasubramanian et al. 2003]. Neste trabalho, escolheu-se seguir a abordagem encontrada em [Isermann 2006] por se considerar mais intuitiva quanto à classificação das técnicas. Em geral, os sistemas FDI podem ser baseados tanto em redundância de *hardware* quanto em redundância analítica. Neste trabalho, apenas considera-se o segundo caso, sendo apresentadas a seguir, as principais abordagens baseadas em redundância analítica.

### 2.2.1 Detecção de Faltas com Verificação de Limites

O método mais simples e frequentemente usado para detecção de faltas é a verificação do limite de uma variável  $Y(t)$  medida diretamente. As variáveis de um processo são monitoradas e comparadas aos valores limites, também chamados de *thresholds*.

Geralmente, dois valores são escolhidos para serem os *thresholds* máximo  $Y_{max}$  e mínimo  $Y_{min}$ . O valor normal caracteriza-se quando

$$Y_{min} < Y(t) < Y_{max}$$

o que quer dizer que o processo se encontra numa situação normal se a variável monitorada se encontra dentro da faixa de tolerância. Exceder um dos *thresholds* significa que o processo encontra-se em modo de falta ou falha. Esse método é aplicado em quase todos os sistemas de automação de processos e as variáveis podem ser: pressão de óleo, temperatura da água, pressão de gases, entre outras. Os *thresholds* são selecionados a partir de experiências e geralmente indicam alguma margem de segurança. A Figura 2.6 mostra a monitoração de uma variável hipotética  $Y(t)$  e seus respectivos limites inferior  $Y_{min}$  e superior  $Y_{max}$ .

Apesar de eficiente, esse tipo de verificação é bastante simplificado. Na prática, nem sempre que o sinal ultrapassa um dos limites significa a presença de uma falha e, nesses

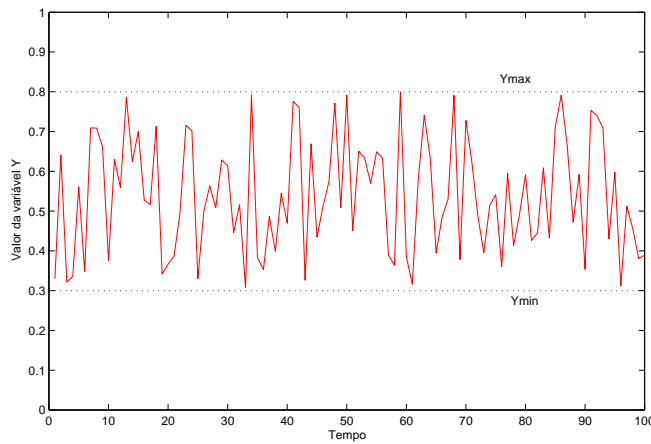


Figura 2.6: Verificação do sinal  $Y(t)$  entre os limites inferior  $Y_{min}$  e superior  $Y_{max}$ .

casos, os alarmes são gerados desnecessariamente. Outro aspecto importante é que os valores dos *thresholds* podem ser funções do ponto de operação do processo, o que leva a política da programação *on-line* do *threshold*. Isto pode aumentar bastante a complexidade do procedimento.

### 2.2.2 Detecção de Falhas com Equações de Paridade

Uma maneira direta de detectar falhas em processos é comparar o seu comportamento real, com o seu comportamento nominal, ou seja, sem estar em modo de falha. Nesse caso, a comparação dos sinais de saída do processo real com os sinais de saída das equações do mesmo fornecem os chamados *resíduos*<sup>2</sup>, que são os sinais que serão analisados para a decisão sobre falhas ou não. Esta abordagem tem a capacidade de indicar anormalidades no processo através das discrepâncias entre o sinal de saída do processo e das equações. Porém, é necessário o conhecimento prévio da equação que governa o processo e de seus parâmetros. Devido à dificuldade de se obter equações precisas em casos reais e das mesmas não levarem em consideração os ruídos encontrados na natureza dos processos, esse método não é muito utilizado, porém o seu princípio de funcionamento por comparação com um “processo nominal livre de falhas” é amplamente popular em outras técnicas. A Figura 2.7 ilustra em esquema de blocos essa abordagem, onde a variável  $\theta$  representa os resíduos.

### 2.2.3 Modelos de Processos e Modelagem das Falhas

Métodos de detecção de faltas baseados em modelos usam a relação entre variáveis provenientes do modelo e do processo real para extrair informação de possíveis mudanças

<sup>2</sup>Resíduos são sinais gerados a partir da comparação entre valores mensuráveis do processo real e valores estimados pelos modelos ou equações quando os mesmos estão sujeitos à mesma entrada. Eles são a base de várias abordagens de sistemas FDI.



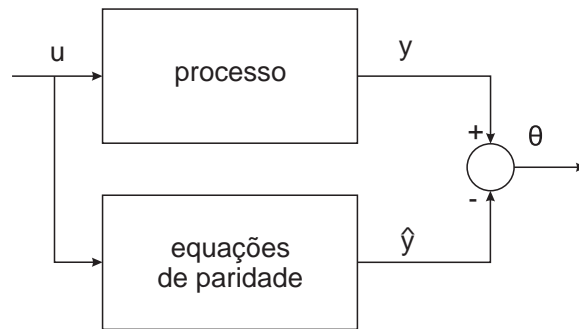


Figura 2.7: Esquema de blocos representando o método de detecção de falhas por equações de paridade.

ocasionadas por faltas. Essas relações são, na maioria das vezes, relações analíticas na forma de equações dos modelos de processos. A Figura 2.8 mostra o esquema básico para o sistema de detecção de falhas baseado em modelo. As relações entre os sinais de entrada  $U$  e os sinais de saída  $Y$  são representados pelo modelo matemático do processo. Métodos de detecção de faltas extraem atributos especiais como parâmetros  $\theta$ , variáveis de estado  $x$ , e resíduos  $r$ . Comparando esses valores observados com os valores nominais, e aplicando métodos de detecção de diferenças, podem ser gerados os sintomas analíticos  $s$ .

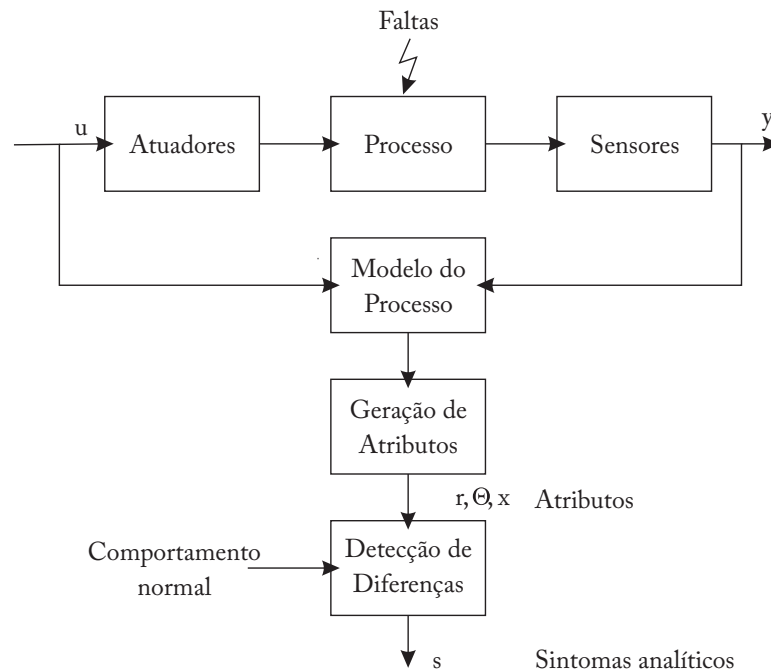


Figura 2.8: Esquema de detecção de falhas baseado em modelos.

Esses sintomas são a base para o diagnóstico de faltas. Eles servirão como entrada no processo de diagnóstico e são diretamente responsáveis por indicar todos os detalhes das faltas ocorridas.

### 2.2.4 Detecção de Falhas com Métodos de Identificação de Processos

Um modelo matemático de um processo descreve a relação entre as entradas  $u(t)$  e as saídas  $y(t)$ . Tais modelos são fundamentais para essa modalidade de detecção de falhas, dado que eles servirão, neste caso, como referência do processo industrial quando o mesmo está em situação normal de funcionamento. Em muitos casos os modelos não são totalmente conhecidos, enquanto que em outros apenas alguns parâmetros são desconhecidos. O Capítulo 3 tratará esse assunto de maneira mais aprofundada.

Uma questão importante a ser considerada quando se utiliza essa abordagem para sistemas FDI é a qualidade dos modelos. Esses devem ser precisos para expressar minimamente os desvios provocados por possíveis faltas no processo. Conseqüentemente, métodos de identificação de processos precisam ser aplicados freqüentemente antes de se aplicar qualquer técnica de detecção de falhas. Conteúdo adicional sobre a área de identificação de sistemas pode ser encontrado em [Aguirre 2007], [Ljung 1987], [Fernandes et al. 2006], [Billings 1980] e [Van Overschee 1994]. A Figura 2.9 mostra a arquitetura geralmente utilizada quando se trata dessa abordagem.

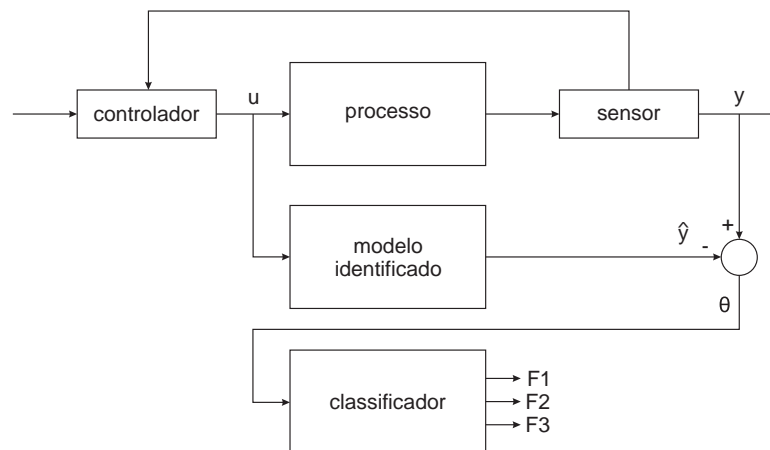


Figura 2.9: Esquema de blocos representando o método de detecção e classificação de falhas através de identificação de sistemas.

Observando-se a Figura 2.9 pode-se perceber que o módulo rotulado “classificador” recebe resíduos, representados por  $\theta$ , que são gerados a partir das saídas do processo real e do modelo. Estes são a base para a identificação da falha. Por fim, a saída do classificador indica qual falha ocorreu. Nesse caso, elas estão representadas por  $F1$ ,  $F2$  e  $F3$ . Neste caso, a etapa de classificação envolve as fases de isolamento e identificação das falhas.

Vale lembrar que, na maioria das vezes, os modelos utilizados com essa abordagem são do tipo caixa preta, ou seja, seus parâmetros não têm nenhum significado físico, isto é, nenhuma relação direta com o modelo fenomenológico do processo em questão. A técnica de detecção de falhas por identificação de processos foi utilizada em [Fernandes et al. 2007c], [Silva et al. 2007], [Armengol et al. 2003], [Simani 2005], [Gertler 1998] e [Ding et al. 2005].

### 2.2.5 Detecção de Falhas com Observadores de Estado e Estimadores de Estado

Como observadores de estado usam uma saída de erro entre a saída do processo medido e uma saída do modelo ajustável, eles são uma alternativa adicional aos métodos baseados em modelo. É assumido, no caso da abordagem de paridade de equações, que a estrutura e os parâmetros do modelo são precisamente conhecidos. Observadores de estado ajustam as variáveis de estado de acordo com as condições iniciais e o curso do tempo dos sinais medidos de entrada e saída.

Abordagens têm sido propostas, na literatura, para sistemas de detecção de falhas baseadas nos observadores de estados de Luemberger e Filtros de Kalman [An & Sepeshri 2003] e [Walker & Huang 1995]. Estes últimos serão bastante utilizados neste trabalho tendo sua teoria apresentada na Seção 3.5. Uma importante diferença entre as abordagens encontradas na literatura e a proposta nesta tese, é o fato de aqui se usar o filtro de Kalman para a identificação paramétrica dos processos, e não para a estimativa de estados, como encontrado nas publicações supracitadas.

### 2.2.6 Detecção de Falhas Através da Análise da Assinatura de Sinais

Para processos de larga escala, como plantas químicas, o desenvolvimento de métodos de detecção de falhas baseados em modelo requerem um esforço muito grande. Então, métodos baseados em análise de dados oferecem uma maneira alternativa. Especialmente métodos baseados em análise estatística multivariável como Análise dos Componentes Principais (PCA) e Projeção por Estruturas Latentes (PLS).

Esses métodos são atrativos quando as medições dos processos são altamente correlacionadas e apenas alguns eventos (falhas) produzem dados não usuais, ou descorrelacionados. Os modelos PCA são basicamente lineares e estáticos e são desenvolvidos a partir do funcionamento normal do processo. Entretanto, eles podem ser expandidos para outras situações.

Existem ainda outras técnicas amplamente utilizadas para análise de sinais como *Wavelets* e FFT (*Fast Fourier Transform*). Assim como PCA e PLS, elas não necessitam do modelo do processo. Essas ferramentas matemáticas são utilizadas para fazer uma transformação em um sinal, tornando-o mais apropriado para a tomada de decisão sobre a existência ou não de uma falha. Trabalhos utilizando transformadas *Wavelets* no contexto de sistemas FDI podem ser encontrados em [Freitas et al. 2004], [Martins 2007], [Jones & Corbin 1989], [Mallet et al. 1997], [de Souza et al. 2000], [Galvão & Yoneyama 1998], [Zhang & Yan 2001] e [Paiva 2003]. A Figura 2.10 ilustra o esquema da detecção de falhas por análise de sinais.

### 2.2.7 Outros Métodos de Detecção de Falhas

Na literatura são encontrados vários trabalhos envolvendo uma vasta gama de técnicas aplicadas à tarefa de detectar falhas em sistemas dinâmicos. A Figura 2.11 dá uma noção geral dessas tecnologias.

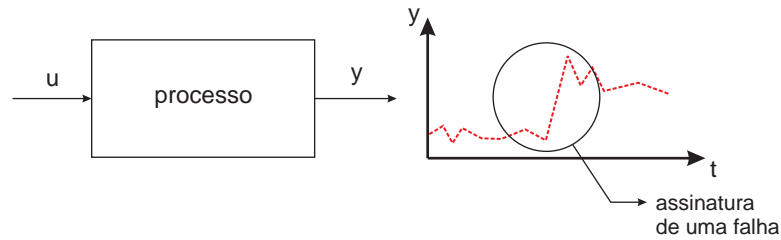


Figura 2.10: Representação da detecção de falhas por análise de sinais.

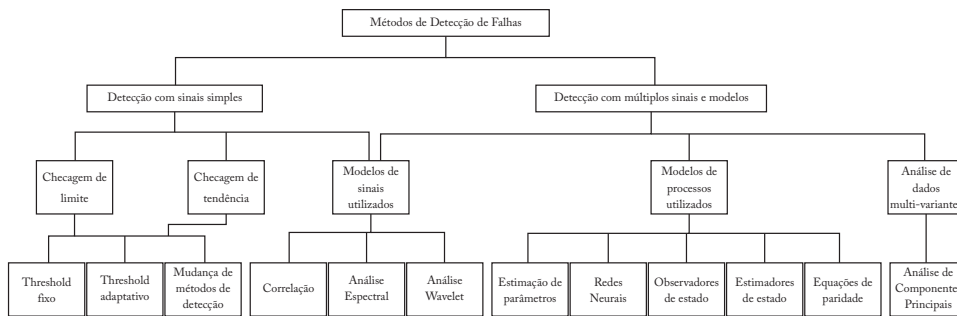


Figura 2.11: Resumo das tecnologias usadas na detecção de faltas em sistemas dinâmicos.

Uma ferramenta amplamente utilizada na tarefa de detectar falhas são as redes neurais artificiais. Trabalhos envolvendo esta ferramenta podem ser encontrados em [Tinos & Terra 1998], [Naughton et al. 1996], [Shin et al. 2005], [IEE 1994], [Srinivasan & Batur 1994], [Linarié & Koroman 2003] e [Gao et al. n.d.]. Uma classificação das técnicas utilizando redes neurais em sistemas de detecção de falhas pode ser encontrada em [Madani 1999].

Os sistemas especialistas, outro tipo de ferramenta amplamente utilizado quando se aplica técnicas de sistemas inteligentes, também têm suas aplicações na área de detecção de falhas, como em [W. & White 1989], [Al-Taani 2005], [Monsef et al. 1997], [Hu et al. 1999] e [Tatara et al. 2000].

A lógica fuzzy aparece nos sistemas FDI em [Rich & Knight 1994], [Koscielny et al. 1999], [Fries & Graham 2003] e [Monsef et al. 1997].

Em geral, as técnicas baseadas em inteligência artificial, como redes neurais, lógica fuzzy e sistemas especialistas, podem ser empregadas pelas diversas abordagens de detecção de falhas (assinatura de falhas, identificação de sistemas, etc.).

A detecção de faltas, falhas ou anormalidades é uma etapa de extrema importância para os sistemas FDI. É nela que se encontra a parte do sistema responsável por indicar se o processo está em funcionamento normal, ou, em modo de falha. Porém, nesta etapa, informações vitais acerca da falha, como localização física, amplitude e intensidade da mesma não são fornecidas ao operador. Desta forma, surge a necessidade de uma outra etapa no processo de tratamento das falhas, chamada diagnóstico.

Esta segunda etapa apresenta seus próprios desafios e dificuldades, podendo ser tratada de maneira independente da primeira. Ela exige suas próprias técnicas e soluções que, neste trabalho, serão abordados na próxima seção.

## 2.3 Métodos de Diagnóstico de Falhas

A tarefa de diagnosticar uma falha consiste na determinação do tipo da falha com todos os detalhes possíveis como, tamanho da falha, localização e hora de detecção. O procedimento de diagnóstico é baseado nos sintomas analíticos e heurísticos observados e no conhecimento heurístico do processo.

A Figura 2.12 resume os passos para as variáveis medidas automaticamente e também para a observação humana. Em ambos os casos a extração de atributos e a detecção de alterações da situação normal ou nominal são necessárias. Sintomas analíticos ou heurísticos devem ser trazidos para uma representação unificada de sintomas a fim de se executar o diagnóstico.

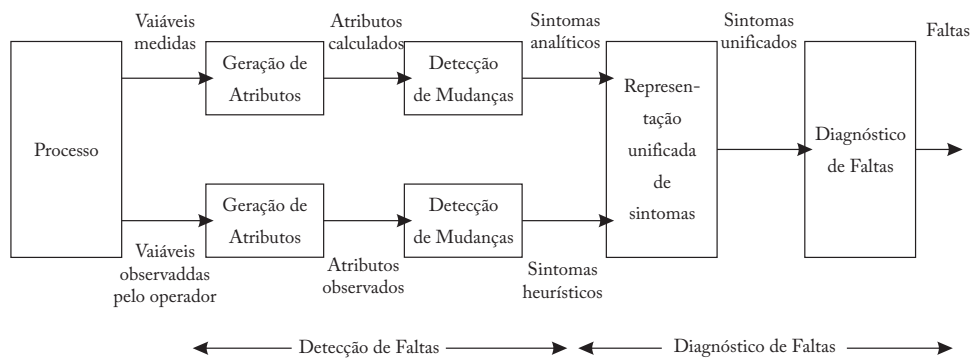


Figura 2.12: Esquema geral do diagnóstico de falhas.

Entende-se por *atributos* valores extraídos de sinais ou modelos de processos que descrevem o estado atual do processo (como parâmetros, variáveis de estado, etc.) e por *sintomas* as possíveis alterações de atributos dos seus valores normais. No caso de um processo sem falhas os sintomas devem ser próximos a zero.

As entradas de um procedimento de diagnóstico baseado em conhecimento são os sintomas disponíveis e algum outro conhecimento relevante para falhas sobre o processo. São eles:

**Sintomas analíticos:** são os resultados das checagens de limites dos sinais mensuráveis, ou os resultados da detecção de falhas.

**Sintomas heurísticos:** são as observações do pessoal de operação na forma de ruído acústico, oscilações ou impressões ópticas como cores ou fumaça, obtidos por inspeção. Esses indicativos podem geralmente ser representados na forma de medidas qualitativas, como expressões lingüísticas do tipo “pouco”, “médio” ou “muito”.

**Representação unificada de sintomas:** para o processamento de todos os sintomas no mecanismo de inferência, é vantajoso utilizar uma representação unificada, que será a entrada do procedimento de diagnóstico, como mostra a Figura 2.12.

**Relacionamento falta-sintoma:** a propagação de falhas para sintomas observáveis segue o relacionamento físico geral de causa-efeito. A Figura 2.13a mostra que uma falta influencia eventos, que por sua vez, influenciam nos sintomas medidos e observados. O diagnóstico de falhas procede da maneira inversa. O sistema deve indicar as falhas de acordo com os sintomas, Figura 2.13b.

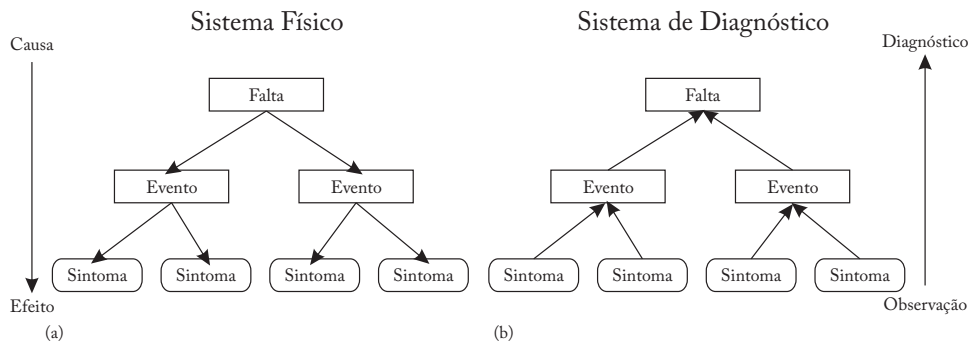


Figura 2.13: *Relacionamento falta-sintoma.*

A Figura 2.14 mostra as tecnologias que vêm sendo utilizadas no diagnóstico de falhas. Essa categorização dos métodos de diagnóstico está de acordo com o descrito por Isermann em [Isermann 2006]. Na Figura 2.14 pode-se ver que tais métodos são divididos em duas grandes categorias: métodos de classificação e métodos de inferência. A primeira é subdividida em quatro subcategorias, todas elas decidem qual falha ocorreu através de comparações entre características das falhas ocorridas. Algumas delas compara tais características com as características das falhas previamente analisadas para enquadrá-la em algum caso já conhecido. Como pode ser visto ainda na Figura 2.14, várias são as ferramentas utilizadas na categoria de classificação, como: tabelas de decisão, classificador bayesiano, árvores de decisão, etc. A segunda categoria, utiliza a inferência para classificar os ocorridos. A primeira subcategoria, entitulado “Métodos de Inteligência Artificial” se utiliza de várias ferramentas provenientes da inteligência artificial [Russell 2003] como por exemplo a lógica nebulosa e as redes neurais artificiais [Haykin 2001b]. As outras duas subcategorias: decisão binária e decisão por aproximação utilizam como ferramentas matemáticas a lógica de predicados e as já citadas lógica nebulosa e redes neurais artificiais.

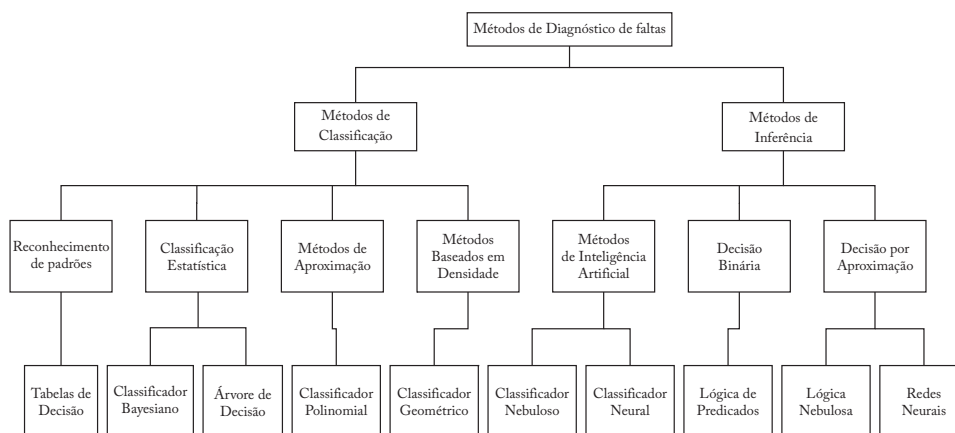


Figura 2.14: *Métodos de diagnóstico de falhas.*

Como visto, grande parte dos sistemas de detecção e diagnóstico de falhas empregam duas etapas: detecção de falhas via geração de resíduos e classificação das falhas. Diferentemente da abordagem convencional, neste tese propõe-se utilizar uma abordagem

baseada em identificação *on-line* dos parâmetros físicos do processo. Então, a partir da saída do processo de identificação é possível monitorar variações dos parâmetros do processo físico, o que corresponde a atividades de detecção, isolamento e identificação de falhas. Assim, a etapa de classificação é bastante simplificada, podendo ser realizada pela observação direta dos valores dos parâmetros estimados pelo sistema FDI proposto. Uma outra possibilidade, é de se implementar um módulo de diagnóstico que sintetize as informações fornecidas pelo sistema FDI em gráficos ou informações mais conclusivas. Tal alternativa não faz parte dos objetivos deste tese, podendo ser desenvolvida como trabalho futuro.

Devido ao fato de utilizar-se uma abordagem baseada em identificação paramétrica, no próximo capítulo é apresentada uma panorâmica sobre a área de identificação de sistemas.





---

# Capítulo 3

## Identificação de Sistemas

---

### 3.1 Introdução

Identificação de sistemas pode ser definida como uma caracterização de um sistema dinâmico através da observação de seu comportamento mensurável. Identificação tem sido estudada a décadas por uma variedade de comunidades de pesquisa, cada uma com seu ponto de vista e interesse, como: regressão estatística (estimação), processamento de sinais (filtragem) e engenharia de controle (controle adaptativo) [Aguirre 2007]. Quando informação *a priori* sobre as regras que governam um sistema não existem ou são muito complexas, técnicas de identificação são usadas para construir um modelo usando apenas dados observados de entrada e saída.

Então, o sistema é chamado de caixa-preta, se o comportamento interno é desconhecido. Caso contrário, em muitos sistemas físicos, nosso conhecimento das leis mecânicas, químicas e elétricas permitem-nos formular um modelo, que é uma ferramenta fundamental para estudo do sistema, analiticamente ou através de simulações [Aguirre 2007]. De qualquer forma, não importa o quão detalhado é o seu modelo, os parâmetros de qualquer modelo apresentam imprecisões.

Identificação de sistemas consiste em uma técnica capaz de encontrar um modelo matemático ou uma representação de um sistema baseando-se em seus dados de entrada e saída. Essa técnica é usada sempre que há a necessidade de se usar uma cópia do sistema ou que se deseja saber os parâmetros da equação matemática que rege determinado sistema.

Quando não se conhece a composição do sistema que se quer modelar ou identificar, chama-se o processo de modelagem caixa-preta. Quando se tem ao menos a equação de referência e se pretende achar os parâmetros corretos, chama-se o processo de modelagem de caixa-cinza. E, finalmente, quando se conhece, ao final do processo de modelagem, a equação e os valores de todos os parâmetros envolvidos, chama-se de modelagem matemática ou caixa branca.

Independente de qual seja o controlador usado, é normalmente útil conhecer como a variável manipulada afeta a variável controlada. Frequentemente, a relação entre essas variáveis é dinâmica, o que exige um modelo dinâmico para quantificar a relação de causa e efeito que existe entre essas variáveis. Essa é a principal motivação para se obter modelos que descrevam os sistemas reais a serem trabalhados em alguma aplicação.

### 3.1.1 Os Passos de um Processo de Identificação de Sistemas

Existem vários aspectos importantes envolvidos em problemas reais de identificação. Porém, pode-se dizer que em linhas gerais o processo de identificação de um sistema dinâmico segue as seguintes etapas [Aguirre 2007]:

1. *testes dinâmicos e coleta de dados.* Já que, como dito anteriormente, a identificação se propõe a encontrar modelos a partir de dados de entrada e saída, esses necessitam, em alguma das etapas serem gerados, escolhidos ou coletados. Entretanto, na maioria das vezes, só estão disponíveis dados que representam a operação normal do sistema. Para qualquer outra situação, será necessário gerar tais dados através de testes com o intuito de explorar a dinâmica do sistema. Conseqüentemente, os sinais de excitação serão um ponto chave desta etapa, assim como a escolha do tempo de amostragem;
2. *escolha da representação matemática.* Como será visto nas próximas seções, existem diversas representações para sistemas dinâmicos, tanto os lineares quanto os não-lineares. Porém cada representação apresenta diversas características, que devem estar em harmonia com o sistema real a ser identificado, como por exemplo a ordem e a complexidade do sistema. As possibilidades de combinações de representações e suas características tornam essa etapa complexa, mas de extrema importância;
3. *determinação da estrutura do modelo.* A escolha da estrutura do modelo nada mais é do que a definição das características da representação matemática, definida no passo anterior. Ela está diretamente ligada à capacidade de se construir um modelo capaz de mapear os dados de entrada e saída coletados na primeira etapa;
4. *estimação de parâmetros.* Essa etapa também envolve a escolha de um algoritmo a ser utilizado na busca dos parâmetros corretos. O algoritmo de mínimos quadrados é amplamente utilizado na literatura relacionada e já foram desenvolvidas diversas variações do original. Esse método, e suas variações, são eficientemente utilizados em representações matemáticas lineares nos parâmetros;
5. *validação do modelo.* Após ter seguido os quatro passos anteriores, o que se espera ter é pelo menos uma estrutura matemática, capaz de representar o sistema original. A etapa de validação trata de comparar os desempenhos de tais estruturas, em caso de se ter mais de uma, com o objetivo de escolher a mais representativa, ou a que melhor descreve o sistema original. Essa etapa pode ser muito subjetiva, pois depende da aplicação a qual se destina o modelo e da quantidade de dados disponível do sistema original.

## 3.2 Representações Lineares

Existem diversas maneiras de se representar o mesmo modelo matemático, ou seja, existem várias formas de se escrever as equações que descrevem o comportamento dinâmico de um sistema. Esta seção aborda as representações mais comuns para sistemas lineares. As várias maneiras de se representar sistemas não-lineares serão vistas na próxima seção.

### 3.2.1 Funções de Transferência

Funções de transferência são funções que modelam o comportamento dinâmico de um par entrada-saída de um sistema, ou seja, descrevem como uma entrada é dinamicamente “transferida” para a saída do sistema.

A função de transferência (FT) de um sistema é, por definição, a transformada de Laplace da sua resposta ao impulso. Elas são normalmente representadas como a razão de dois polinômios em  $s$ . Como por exemplo:

$$G(s) = \frac{N(s)}{D(s)} = \frac{b_0 + b_1s + \dots + b_qs^q}{a_0 + a_1s + \dots + a_ns^n} \quad (3.1)$$

onde os zeros de  $G(s)$  são os valores de  $s = z_1, z_2, \dots, z_q$ , para os quais  $G(s)$  é nula. Em outras palavras, os zeros de  $G(s)$  são os zeros do polinômio  $N(s)$ . Semelhantemente, os pólos de  $G(s)$  são os zeros do polinômio  $D(s)$ , isto é, valores de  $s = p_1, p_2, \dots, p_n$ , para os quais  $D(s)$  é zero.

Um problema típico em modelagem de sistemas lineares é o de obter a função de transferência do sistema em estudo. Isso é conseguido aplicando-se a transformada de Laplace à equação diferencial que descreve o sistema. A equação diferencial, por sua vez, é obtida através das leis que descrevem os fenômenos envolvidos no mesmo.

### 3.2.2 Resposta Temporal

Uma outra maneira de se encontrar a função de transferência de um sistema, é aplicando-se a transformada de Laplace à sua resposta ao impulso [Aguirre 2007]. Em consequência disso, a resposta temporal ao impulso pode ser obtida usando-se a transformada inversa de Laplace da respectiva função de transferência.

A resposta temporal  $y(t)$  de um sistema, cuja resposta ao impulso é  $g(t)$ , quando excitado por uma entrada  $u(t)$ , pode ser obtida da seguinte forma:

$$y(t) = \mathcal{L}^{-1}Y(s) = \mathcal{L}^{-1}G(s)U(s), \quad (3.2)$$

onde  $Y(s)$ ,  $G(s)$  e  $U(s)$  são, respectivamente, as transformadas de Laplace de  $y(t)$ ,  $h(t)$  e  $u(t)$ . A Equação 3.2 presume que é necessário conhecer, além da função de transferência do sistema, a transformada de Laplace do sinal de entrada.

### 3.2.3 Espaço de Estados

Uma das características básicas das funções de transferência é a descrição da dinâmica do sistema relacionando a entrada e a saída do mesmo, ou seja, ela descreve a relação de causa e efeito entre uma entrada e uma saída de um dado sistema. Porém, a função de transferência não mostra o que está acontecendo dentro do sistema, entre os pontos de entrada e saída.

Diferentemente da função de transferência, a representação por espaço de estados tem a capacidade de modelar relações entre variáveis internas do sistema. Essa representação

descreve o sistema no domínio do tempo e permite também representar sistemas não-lineares e multivariáveis, diferentemente da função de transferência. Um modelo linear típico tem a seguinte forma, quando representado em espaço de estados:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du,\end{aligned}\tag{3.3}$$

sendo que  $x \in R^n$  é o vetor de estado  $n$ -dimensional, o ponto indica a derivada temporal, ou seja,  $\dot{x} = dx/dt$ ,  $u(t) \in R^r$  é o vetor de entradas formado por  $r$  funções temporais,  $y(t) \in R^m$  é o vetor  $m$ -dimensional de saídas medidas e  $A$ ,  $B$ ,  $C$  e  $D$  são matrizes constantes. O modelo descrito no conjunto de equações 3.3 será multidimensional se  $r > 1$  e/ou  $m > 1$ . De maneira que, se houver apenas uma entrada  $r = 1$  e uma saída  $m = 1$ , o modelo é dito monovariável.

Duas observações importantes sobre a representação em espaço de estados são:

- o conhecimento do vetor de estado em qualquer instante  $t_0$  especifica o estado ou condição do sistema nesse instante;
- a representação em espaço de estado não é única. Ou seja, é possível representar o mesmo sistema com mais de um modelo no espaço de estados.

É importante salientar que representações por variáveis de estados, diferentemente de funções de transferência, não estão limitadas a sistemas lineares. No caso geral a representação por variáveis de estado de um sistema é dada por:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= g(x, u),\end{aligned}\tag{3.4}$$

onde  $f(\cdot) : R^n \rightarrow R^n$  e  $g(\cdot) : R^n \rightarrow R^m$ .

Entre todas as representações lineares e não-lineares, a por espaço de estados tem uma relevância particular para o presente trabalho. O capítulo seguinte faz uso de algoritmos que utilizam essa representação para realizar a identificação de sistemas, o que será a base para a ferramenta aqui proposta.

### 3.2.4 Representações em Tempo Discreto

Existem algumas representações matemáticas que são especialmente adequadas à identificação de sistemas através do uso de algoritmo estimadores de parâmetros. Nesta seção, serão apresentadas algumas dessas representações.

Considerando-se o seguinte modelo geral:

$$\begin{aligned}A(q)y(k) &= \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k), \\ y(k) &= \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}v(k), \\ y(k) &= G(q)u(k) + V(q)v(k),\end{aligned}\tag{3.5}$$

onde  $A(q)$ ,  $B(q)$ ,  $C(q)$ ,  $D(q)$  e  $F(q)$  são os polinômios definidos a seguir:

$$\begin{aligned} A(q) &= 1 - a_1q^{-1} - \dots - a_{n_y}q^{-n_y}; \\ B(q) &= b_1q^{-1} + \dots + b_{n_u}q^{-n_u}; \\ C(q) &= 1 + c_1q^{-1} + \dots + c_{n_v}q^{-n_v}; \\ D(q) &= 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d}; \\ F(q) &= 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}. \end{aligned} \quad (3.6)$$

Sendo  $q^{-1}$  o operador de atraso, de forma que  $y(k)q^{-1} = y(k-1)$  e  $v(k)$  ruído branco. As funções  $G(q)$  e  $V(q)$  são referidas como as funções de transferência do processo e do ruído, respectivamente.

### Modelo ARX

O modelo auto-regressivo com entradas externas (ARX, do inglês *autorregressive with exogenous inputs*) pode ser obtido a partir do modelo geral da Equação 3.5, assumindo-se  $C(q) = D(q) = F(q) = 1$  sendo  $A(q)$  e  $B(q)$  polinômios arbitrários, o que resulta em:

$$A(q)y(k) = B(q)u(k) + v(k). \quad (3.7)$$

que pode ser reescrito como segue:

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k), \quad (3.8)$$

que coloca em evidência as funções de transferência do sistema  $H(q) = B(q)/A(q)$  e de ruído  $1/A(q)$ , como pode ser visto na Figura 3.1.

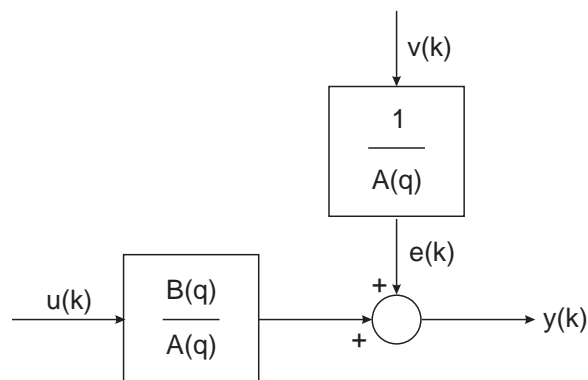


Figura 3.1: Representação esquemática do modelo ARX.

### Modelo ARMAX

O modelo auto-regressivo com média móvel e entradas exógenas (ARMAX, do inglês *autoregressive moving average with exogenous inputs*) também pode ser obtido a partir

do modelo geral da Equação 3.5, assumindo  $D(q) = F(q) = 1$  sendo  $A(q)$ ,  $B(q)$  e  $C(q)$  polinômios arbitrários, resultando em

$$A(q)y(k) = B(q)u(k) + C(q)v(k). \quad (3.9)$$

ou

$$\begin{aligned} y(k) &= \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}v(k) \\ y(k) &= H(q)u(k) + e(k), \end{aligned} \quad (3.10)$$

sendo  $e(k)$  não branco, como ilustrado na Figura 3.2.

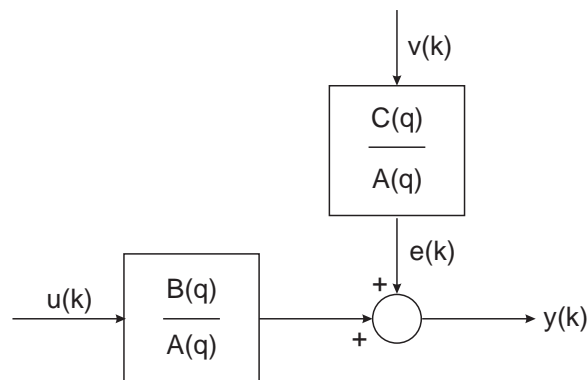


Figura 3.2: Representação esquemática do modelo ARMAX.

Nesse caso, o erro é modelado como um processo de média móvel (MA), e o ruído adicionado à saída,  $e(k)$ , é modelado como ruído branco filtrado por  $C(q)/A(q)$  (um filtro ARMA).

### 3.3 Representações Não-lineares

A maioria dos sistemas dinâmicos encontrados no mundo real são não-lineares. Em alguns casos, no entanto, é possível usar aproximações lineares dos mesmos e ainda assim obter bons resultados. Porém, nem sempre essa tática é satisfatória. Nesses casos deve-se realmente procurar um modelo mais próximo, que inevitavelmente será um modelo não-linear. O uso de modelos não-lineares exige mais dos algoritmos utilizados em sua identificação, resultando num aumento da complexidade desses.

Nesta seção encontra-se uma breve descrição de algumas representações não-lineares. Há uma grande variedade de representações não-lineares que podem ser usadas na identificação de sistemas. A seguir, algumas delas são sucintamente apresentadas.

#### 3.3.1 A Série de Volterra

A série de Volterra tem a capacidade de relacionar a saída  $y(t)$  com a entrada  $u(t)$  de um sistema não-linear. Ela é definida como segue:

$$y(t) = \sum_{j=1}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_j(\tau_1, \dots, \tau_j) \prod_{i=1}^j u(t - \tau_i) d\tau_i, \quad (3.11)$$

onde as funções  $h_j$  são os chamados núcleos ou *kernels* que consistem de generalizações não-lineares da resposta ao impulso  $h_1(t)$ . No caso do sistema ser linear, onde  $j = 1$ , a equação acima equivale à integral de convolução.

Um dos problemas da utilização da série de Volterra é o grande número de parâmetros que precisa ser definido [Aguirre 2007].

### 3.3.2 Modelos de Hammerstein e de Wiener

Essas representações consistem do uso de um sistema dinâmico linear acoplado em cascata a uma função estática não-linear. No caso de Hammerstein a função não-linear precede o modelo dinâmico linear.

Os modelos de Hammerstein e de Wiener podem ser também definidos em tempo discreto, como mostra a Figura 3.3.

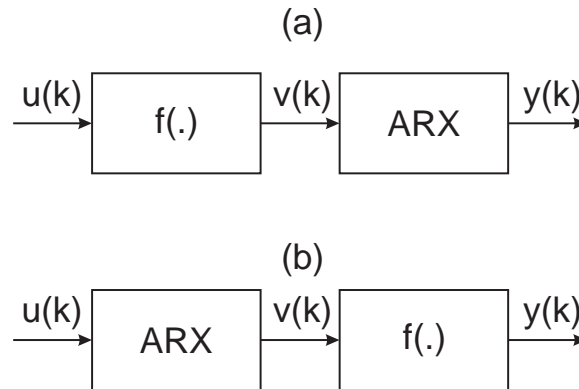


Figura 3.3: (a) modelo de Hammerstein, (b) modelo de Wiener.

### 3.3.3 Representações NARX

Os modelos NARX (do inglês *nonlinear autoregressive model with exogenous inputs*) são modelos discretos que mapeiam a saída  $y(k)$  de um sistema como uma função dos valores prévios dos sinais de entrada e da saída, ou seja,  $y(k) = F[y(k-1), \dots, y(k-n_y), u(k-\tau_d), \dots, u(k-n_u)]$ , onde  $n_y$ ,  $n_u$  e  $\tau_d$  são os maiores atrasos em  $y$ , em  $u$  e o tempo de atraso, respectivamente. Com o intuito de evitar polarização de parâmetros é comum incluir termos de ruído no modelo, transformando o modelo em NARMAX (*nonlinear autoregressive moving average model with exogenous inputs*), que é normalmente representado da seguinte maneira:

$$y(k) = F[y(k-1), \dots, y(k-n_y), u(k-\tau_d), \dots, u(k-n_u), e(k), e(k-1), \dots, e(k-n_e)], \quad (3.12)$$

onde  $e(k)$  é o ruído e  $n_e$  é o maior atraso no modelo do ruído. Devido à generalidade demasiada do modelo representado pela Equação 3.12 e à conseqüente dificuldade para se determinar a função  $F$ , pode-se considerar a possibilidade de usar a representação polinomial ou racional para  $F$ . Um modelo polinomial NARMAX sem atraso puro de tempo tem a seguinte forma:

$$y(k) = \sum_i c_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} u(k-r) \prod_{q=1}^{n_e} e(k-q). \quad (3.13)$$

Vale ressaltar que quando todos os termos não-lineares forem do tipo  $y(k-i)u(k-j)$ ,  $\forall i, j$  o modelo chama-se bilinear se caracterizando como um caso particular do modelo polinomial representado pela Equação 3.13.

Já os modelos racionais, são formados pela razão entre dois polinômios, como mostrado a seguir:

$$y(k) = \frac{\sum_i c_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} u(k-r) \prod_{q=1}^{n_e} e(k-q)}{\sum_i d_i \prod_{j=1}^{d_y} y(k-j) \prod_{r=1}^{d_u} u(k-r) \prod_{q=1}^{d_e} e(k-q)} + e(k). \quad (3.14)$$

Apesar de permitir utilizar regressores não-lineares nos sinais de saída, entrada e ruído, os modelos polinomiais do tipo mostrado na Equação 3.13 são lineares nos parâmetros, o que os torna capazes de serem representados na forma  $y(k) = \psi^T \theta$ , e, conseqüentemente, muito sobre a teoria de sistemas lineares é válido para esses casos. Já os modelos racionais são não-lineares nos parâmetros e, nesse caso, algoritmos alternativos precisam ser derivados para estimá-los.

### 3.3.4 Modelos Polinomiais Contínuos

Um modelo polinomial, para um sinal  $y(t)$  contínuo no tempo, pode ser formado utilizando-se o sinal e suas derivadas como base. Considerando o caso de ordem três, escolhe-se  $\dot{X} = Y = \dot{y}(t)$ ,  $\dot{Y} = Z$  e  $\dot{Z} = F(X, Y, Z)$ . Nesse caso, os regressores  $(X, Y, Z)$  são o sinal  $y(t)$  e suas derivadas primeira e segunda  $\dot{y}$  e  $\ddot{y}$ . A função  $F$  pode então ser aproximada por uma expansão polinomial ou racional, por exemplo, de combinações não-lineares dos regressores. No caso polinomial, o modelo pode ser representado da seguinte forma:

$$\begin{aligned} \dot{X} &= \dot{y}(t) \\ \dot{Y} &= \ddot{y}(t) \\ \dot{Z} &= \sum_{l=1}^{n_\theta} \theta_l \psi^l, \end{aligned} \quad (3.15)$$

onde  $\psi^l = X^i Y^j Z^k$  e  $i, j, k \in N$ . O problema de definir o número e quais termos devem ser utilizados para compor a expansão polinomial neste caso é muito semelhante ao caso discreto. A dificuldade a mais é a necessidade de se estimar as derivadas do sinal medido  $y(t)$ .



### 3.3.5 Redes Neurais Artificiais

As rede neurais são mais um tipo de representação não-linear [Haykin 2001b]. Elas são compostas por estruturas de processamento chamadas neurônios (inspiradas nas células neurais humanas) interconectados e organizadas em camadas. A saída de cada neurônio com  $n$  entradas é do tipo

$$y = f\left(\sum_{j=1}^n w_j x_j + b\right), \quad (3.16)$$

onde o bias  $b$  e  $w_j$  são constantes e  $f$  é chamada de função de ativação. Existem vários tipos de funções de ativação, sendo a mais comum a sigmóide, mostrada a seguir

$$f(z) = \frac{1}{1 + e^{-z}}. \quad (3.17)$$

No caso de redes neurais multicamadas a saída de um neurônio é conectada à entrada de outro neurônio. Caso a rede tenha apenas um nó na camada de saída e apenas uma camada oculta, a saída da rede é uma função não-linear nos parâmetros do tipo

$$y(k) = f_s\left\{\sum_{i=1}^m w_i f_i\left(\sum_{j=1}^n w_{ij} x_j + b_i\right) + b_s\right\} \quad (3.18)$$

onde  $f_s$  é a função de ativação do neurônio da camada de saída.  $f_i, i = 1, \dots, m$ , são as funções de ativação dos neurônios da camada escondida e  $b_s$  é o bias do neurônio da camada de saída,  $w_i$  são os pesos que ponderam a saída da camada oculta e  $w_{ij}$  são os pesos da entrada  $j$ , vista pelo  $i$ -ésimo neurônio da camada oculta.

### 3.3.6 Funções Radiais de Base

As funções radiais de base (RBF, *radial basis functions*) [Haykin 2001b] são mapeamentos do tipo

$$f(y) = w_0 + \sum_i^{N_c} w_i \phi(\|y - c_i\|), \quad (3.19)$$

onde  $y \in \mathfrak{R}^{d_e}$ ,  $\|\cdot\|$  é a norma euclidiana,  $w_i \in \mathfrak{R}$  são pesos,  $N_c$  é o número de centros  $c_i \in \mathfrak{R}^{d_e}$  e  $\phi(\cdot) : \mathfrak{R}^+ \rightarrow \mathfrak{R}$  é uma função, normalmente escolhida *a priori*, como, por exemplo:

$$\phi(\|y - c_i\|) = \exp\left(-\frac{\|y - c_i\|^2}{\sigma_i^2}\right), \quad (3.20)$$

sendo  $\sigma_i$  constante e  $\|y - c_i\|^2 = (y - c_i)^T (y - c_i)$ . A função de base mostrada na Equação 3.20 é chamada gaussiana.

Essa representação pode ser interpretada como uma técnica de interpolação global com boas propriedades locais. As funções radiais de base são casos particulares de redes neurais, porém, diferente destas, são lineares nos parâmetros  $w_i$

### 3.3.7 O Neurônio *neuro-fuzzy*

O neurônio *neuro-fuzzy* é constituído por três camadas: a camada de entrada, a camada de agregação de regras e a camada de saída, como mostrado na Figura 3.4. Onde cada neurônio possui  $p$  partições, representadas por  $p$  funções de pertinência triangulares, fixas e regularmente distribuídas no espaço de excursão da variável associada, como mostradas na Figura 3.5. Pode-se notar que, no máximo duas dessas funções estarão ativas para uma dada entrada  $x_i$  e a soma de seus respectivos valores  $f_j(x_i) + f_{j+1}(x_i)$  é sempre igual a um.

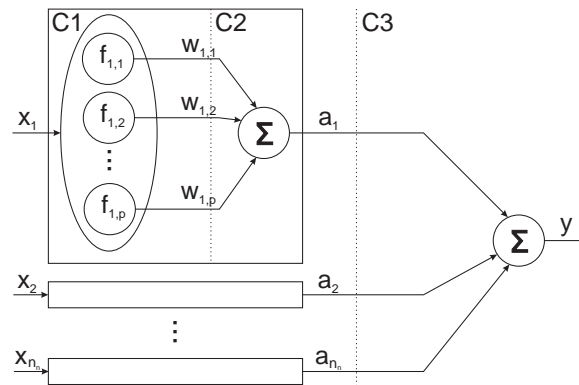


Figura 3.4: Topologia do neurônio *neuro-fuzzy*.

A  $j$ -ésima partição de um dado nó de entrada  $i$  é ponderada por um peso  $w_{i,j}$ , então é possível escrever que a saída do  $i$ -ésimo nó de entrada após a agregação de regras é:

$$a_i = \sum_{j=1}^p f_{i,j}(x_i)w_{i,j}, \quad (3.21)$$

sendo  $w_{i,j}$  pesos ajustáveis na hora do treinamento do neurônio *neuro-fuzzy*.

A última camada realiza a composição de todas as regras por meio de um somatório, ou seja,

$$y = \sum_{i=1}^{n_n} a_i, \quad (3.22)$$

A Equação 3.21 é linear, porém, deve-se notar que, devido a existência de partições de entrada, o neurônio *neuro-fuzzy* é linear por partes, configurando uma representação globalmente não-linear.

## 3.4 Algoritmos de Identificação

Um problema comum a todas as representações de sistemas dinâmicos apresentadas anteriormente é a necessidade de se determinar a sua estrutura, ou seja, a topologia de cada modelo quando aplicado a um sistema em particular.

Depois de se definir a estrutura de um modelo, deve-se determinar seus parâmetros. Essas duas etapas também podem ser definidas ao mesmo tempo. Existem diversos algo-

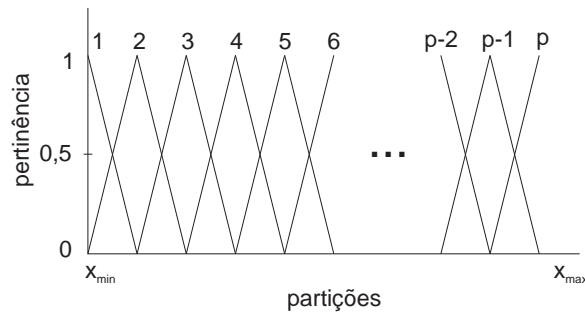


Figura 3.5: Funções de pertinência do neurônio neuro-fuzzy.

ritmos para a determinação de parâmetros de modelos matemáticos, vários dos quais são derivados dos algoritmos de Mínimos Quadrados, ou, MQ, como são mais frequentemente encontrados na literatura. Eles apresentam características interessantes como robustez e facilidade de implementação. Porém, só podem ser aplicados à classe de sistemas que são lineares nos parâmetros. Para as representações não-lineares nos parâmetros, devem ser usados algoritmos específicos e, naturalmente, mais complexos.

Neste trabalho, foram investigados alguns algoritmos para a determinação de parâmetros em representações não-lineares. Entre eles, o *backpropagation*, amplamente utilizado no treinamento de redes neurais artificiais e facilmente encontrado na literatura [Haykin 2001b], [Fu 1994] e [Miller et al. 1995].

Outros algoritmos considerados para o ajuste de representações não-lineares neste trabalho, foram os baseados no Filtro de Kalman. Já que este só é aplicado a sistemas lineares nos parâmetros, algumas variações conhecidas como Filtro de Kalman Estendido (EKF) e Filtro de Kalman *Unscented* (UKF) são aplicados a sistemas não-lineares com êxito, como pode ser consultado em [Haykin 2001a], [Welch & Bishop 2006], [Julier & Uhlmann n.d.], [An & Sepehri 2003] e [Walker & Huang 1995].

O filtro de Kalman e suas extensões são amplamente utilizados neste trabalho. Eles terão sua teoria apresentada na próxima seção.

### 3.5 O Filtro de Kalman

O filtro de Kalman foi publicado em 1960 por R. E. Kalman [Kalman 1960] e consiste em uma solução recursiva para o problema de filtragem linear de dados discretos.

O filtro de Kalman pode ser visto como um conjunto de equações matemáticas que provêm uma maneira eficiente de se estimar os estados de um processo, de maneira que minimize o erro médio quadrado [Welch & Bishop 2006]. O algoritmo é muito poderoso em vários aspectos:

- suporta estimação de estados passados, presentes e até mesmo futuros;
- funciona mesmo quando a natureza precisa do sistema modelado é desconhecida.

O objetivo desta seção é introduzir os conceitos referentes ao filtro de Kalman discreto, incluindo uma descrição e discussões sobre o filtro de Kalman discreto básico. Alguns exemplos serão mostrados e direcionados ao tema da tese.

### 3.5.1 O Filtro de Kalman Discreto

O filtro de Kalman aborda o problema de estimar os estados  $x \in R^n$  de um processo discreto controlado, que é governado por uma equação de diferenças linear estocástica, como por exemplo:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.23)$$

com a seguinte medida  $y \in R^m$

$$y_k = Hx_k + v_k \quad (3.24)$$

onde as variáveis aleatórias  $w_k$  e  $v_k$  representam os ruídos de processo e de medição, respectivamente. Assume-se que elas são independentes entre si, brancas, e com distribuição de probabilidade normal, ou seja:

$$p(w) \sim N(0, Q) \quad (3.25)$$

$$p(v) \sim N(0, R) \quad (3.26)$$

onde  $Q$  e  $R$  são as matrizes de covariância dos ruídos de processo e medição, respectivamente.

A matriz  $n \times n$   $A$  na Equação 3.23 relaciona o estado no instante anterior ( $k - 1$ ) com o estado presente ( $k$ ), na ausência de uma função ou ruído de processo. A matriz  $n \times l$   $B$  relaciona o sinal de controle na entrada  $u \in R^l$  ao estado  $x$ . A matriz  $m \times n$   $H$  na Equação 3.24 relaciona o estado ao valor medido  $y_k$ .

#### A Origem Computacional do Filtro de Kalman

Define-se  $\hat{x}_k^- \in R^n$  como uma estimativa do estado *a priori* no instante  $k$  dado o conhecimento do processo no instante anterior ao instante  $k$ , e  $\hat{x}_k \in R^n$  a estimativa dos estados *a posteriori* no instante  $k$ , dado a medida  $y_k$ . Assim, pode-se definir erros *a priori* e *a posteriori* da seguinte forma:

$$e_k^- \equiv x_k - \hat{x}_k^-, \quad (3.27)$$

e

$$e_k \equiv x_k - \hat{x}_k. \quad (3.28)$$

dessa forma, a covariância da estimativa do erro *a priori* é:

$$P_k^- = E[e_k^- e_k^{-T}] \quad (3.29)$$

e a covariância da estimativa do erro *a posteriori* é:

$$P_k = E[e_k e_k^T] \quad (3.30)$$

Derivar as equações para o filtro de Kalman começa com o objetivo de encontrar uma equação que calcula uma estimativa do estado *a posteriori*  $\hat{x}_k$  como uma combinação linear de uma estimativa *a priori*  $\hat{x}_k^-$  e uma diferença entre a medida observada atual  $y_k$  e

a medição da predição  $H\hat{x}_k^-$ , como

$$\hat{x}_k = \hat{x}_k^- + K(y_k - H\hat{x}_k^-) \quad (3.31)$$

A diferença  $(y_k - H\hat{x}_k^-)$  é chamada de inovação da medida ou resíduo e reflete a discrepância entre a predição da medida  $H\hat{x}_k^-$  e a medida observada  $y_k$ . Quando o resíduo é zero significa que esses dois sinais estão em completa concordância.

A matriz  $K$  na Equação 3.31 é escolhida, para ser o ganho ou o fator de mistura que minimiza a covariância do erro *a posteriori* (3.30). Essa minimização pode ser atingida substituindo (3.31) na definição de  $e_k$  em (3.28), depois, substituindo o resultado disso em (3.30), executando as expectativas indicadas, derivando o resultado em relação a  $K$ , igualando o resultado a zero, e, então, resolvendo para  $K$ . Para mais detalhes veja [Maybeck 1979], [Brown & Hwang 1992] e [Jacobs 1993]. Uma forma resultante de  $K$  que minimiza (3.30) é dada por:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (3.32)$$

Analisando-se a Equação 3.32, pode-se notar que quanto mais a covariância do erro de medição se aproxima de zero, mais a matriz de ganhos  $K$  valoriza o resíduo em (3.31). Especificamente,

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (3.33)$$

Por outro lado, quanto mais a matriz de covariância da estimativa do erro *a priori*  $P_k^-$  se aproxima de zero, menos o ganho  $K$  valoriza o resíduo em (3.31). Especificamente,

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (3.34)$$

Uma outra maneira de interpretar a ponderação por parte de  $K$  é que quanto mais a covariância do erro medido  $R$  se aproxima de zero, a atual medida  $y_k$  é mais confiável, enquanto que a predição da medida  $H\hat{x}_k^-$  é menos confiável. Por outro lado, quanto mais a covariância da estimativa do erro *a priori*  $P_k^-$  se aproxima de zero, a medida observada  $y_k$  é menos confiável, enquanto a medida predita  $H\hat{x}_k^-$  é mais confiável.

### A Origem Probabilística do Filtro de Kalman

A justificativa para a Equação 3.31 é baseada na probabilidade da estimativa *a priori*  $\hat{x}_k^-$  condiciona em todas as medidas observadas anteriores  $y_k$  (regra de Bayes). Isso é suficiente para se notar que o filtro de Kalman mantém os dois primeiros momentos da distribuição do estado.

$$E[x_k] = \hat{x}_k \quad (3.35)$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k \quad (3.36)$$

A estimativa do estado *a posteriori*, na Equação 3.31, reflete a média (o primeiro momento) da distribuição do estado - ele é normalmente distribuído se as condições nas Equações 3.25 e 3.26 são conhecidas. A covariância da estimativa do erro *a posteriori*,

na Equação 3.30, reflete a variância da distribuição do estado (o segundo momento não central). Em outras palavras,

$$p(x_k|y_k) \sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) = N(\hat{x}_k, P_k) \quad (3.37)$$

Para mais detalhes sobre as origens probabilísticas do filtro de Kalman, pode-se consultar [Maybeck 1979], [Brown & Hwang 1992] e [Jacobs 1993].

### O Algoritmo do Filtro Discreto de Kalman

O filtro de Kalman estima um processo usando uma forma de retorno. O filtro estima o estado do processo para algum instante no tempo e, então, obtém o retorno na forma de medições (ruidosas). Assim, as equações do filtro de Kalman se dividem em dois grupos: equações de predição e equações de correção. As primeiras são responsáveis por projetar (através do tempo) estado no instante atual, assim como a covariância do erro para obter estimativas *a priori* para o próximo instante de tempo. As equações de correção são responsáveis pelo retorno, ou seja, por incorporar a nova medição à estimativa *a priori* para obter uma melhor estimativa *a posteriori*. A Figura 3.6 mostra o ciclo predição-correção do algoritmo final.

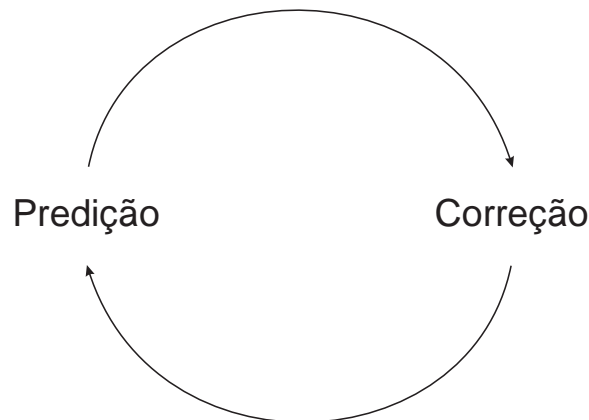


Figura 3.6: *Ciclo de predição-correção do algoritmo do filtro de Kalman discreto.*

As equações específicas para as duas etapas do algoritmo estão presentes nas Tabelas 3.1 e 3.2 abaixo.

Tabela 3.1: *Equações de predição do filtro de Kalman discreto.*

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned}$$

Observando-se a Tabela 3.1, pode-se notar como as equações de predição projetam as estimativas do estado e da covariância do erro do instante  $k - 1$  para o instante  $k$ . As matrizes  $A$  e  $B$  vêm da Equação 3.23, enquanto  $Q$  vem da Equação 3.25.

Tabela 3.2: Equações de correção do filtro de Kalman discreto.

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (y_k - H \hat{x}_k^-) \\ P_k &= (I - K_k H) P_k^- \end{aligned}$$

O primeiro passo na etapa de correção é calcular o ganho de Kalman,  $K_k$ . Pode-se perceber que a primeira equação na Tabela 3.2 é igual à equação 3.32. O próximo passo é medir a saída do processo, obtendo-se  $y_k$  e, então, calcular a estimativa do estado *a posteriori* incorporando a medida à segunda equação da Tabela 3.2, que, por sua vez, é igual à Equação 3.31. O último passo consiste de obter a covariância do erro *a posteriori* através da terceira equação da Tabela 3.2.

Após cada ciclo predição-correção, o processo é repetido com a estimativa *a posteriori* do instante anterior usada para projetar ou predizer os novos estados *a priori*. Essa natureza recursiva é uma das características mais atraentes do filtro de Kalman. Ela torna a implementação muito mais viável que, por exemplo, a implementação de um filtro de Wiener [Brown & Hwang 1992] que é projetado para operar com todos os dados para cada estimativa. O filtro de Kalman, por sua vez, condiciona a estimativa do instante atual em todas as medições passadas. A Figura 3.7 mostra uma ilustração da operação completa do filtro, combinando o diagrama de alto nível mostrado na Figura 3.6 com as equações das Tabelas 3.1 e 3.2.

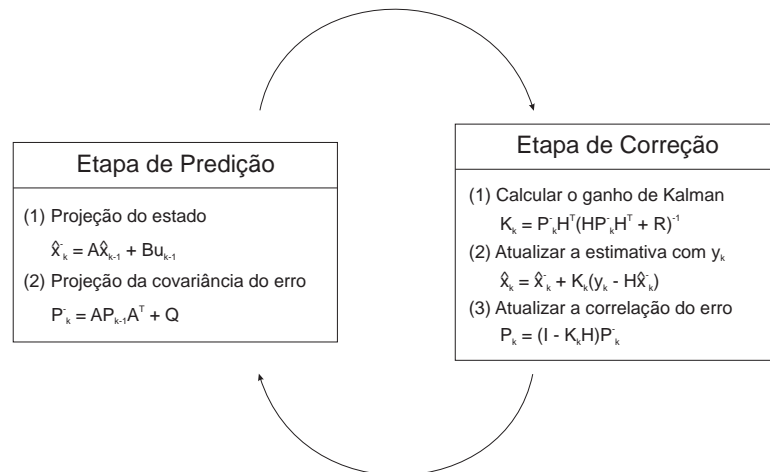


Figura 3.7: Uma ilustração completa da operação do filtro de Kalman, combinando o diagrama da Figura 3.6 com as equações das Tabelas 3.1 e 3.2.

Para concluir essa primeira explanação sobre o filtro de Kalman, pode-se perceber que quando  $Q$  e  $R$  forem constantes, tanto a estimação da covariância do erro  $P_k$  quando o ganho de Kalman  $K_k$  irão se estabilizar rapidamente e se manter constante (veja as equações de correção na Figura 3.7). Se esse é o caso, esses parâmetros podem ser pré-calculados rodando-se o filtro *off-line* ou determinando-se o valor de equilíbrio de  $P_k$  como descrito em [Grewal & Andrews 1993].

É comum, contudo, o caso em que o erro medido não se mantém constante. O ruído de processo  $Q$  também pode mudar dinamicamente durante a operação do filtro, tornando-se  $Q_k$ , para que se possa ajustar para diferentes instantes durante a execução do filtro. Este caso é tratado com filtros adaptativos, como em [Qi & Jian-Da 2008].

### 3.6 O Filtro de Kalman Estendido

Como descrito na seção anterior, o filtro de Kalman soluciona o problema de se tentar estimar o estado  $x \in R^n$  de um processo discreto controlado que é governado por uma equação de diferenças estocástica linear. Porém, o que acontece se o processo a ser estimado ou se a relação com a saída do processo é não-linear? Algumas das aplicações mais interessantes do filtro de Kalman apresentam essa realidade. Um filtro de Kalman que lineariza em torno da média e da covariância é conhecido como o Filtro de Kalman Estendido ou EKF, do inglês *Extended Kalman Filter* [Welch & Bishop 2006].

Em alguns aspectos, o EKF se assemelha a uma série de Taylor, podendo-se linearizar a estimativa em torno da estimativa do instante atual, usando-se as derivadas parciais das funções do processo e medição para calcular estimativas mesmo se as relações forem não-lineares. Para executar tal idéia, deve-se modificar o que foi exposto na seção anterior. Deve-se assumir que o processo, neste caso, tem um vetor de estados  $x \in R^n$ , mas que o processo é agora governado por uma equação de diferenças estocástica não-linear

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}), \quad (3.38)$$

com a saída observada  $y \in R^m$  sendo

$$y_k = h(x_k, v_k) \quad (3.39)$$

onde as variáveis aleatórias  $w_k$  e  $v_k$  novamente representam os ruídos de processo e medição como descritos pelas Equações 3.23 e 3.24. Nesse caso, a função não-linear  $f$  na equação de diferenças 3.38 relaciona o estado no instante anterior  $k - 1$  com o estado no instante atual  $k$ . Ela inclui como parâmetro uma função direcionadora  $u_{k-1}$  e o ruído do processo  $w_k$  de média zero. A função não-linear  $h$  na Equação de medição 3.39 relaciona o estado  $x_k$  à medida observada  $y_k$ .

Na prática, é claro que não se sabe os valores dos ruídos  $w_k$  e  $v_k$  a cada iteração. De qualquer forma, pode-se aproximar o estado e a saída sem eles

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0), \quad (3.40)$$

e

$$\tilde{y}_k = h(\tilde{x}_k, v_k) \quad (3.41)$$

onde  $\hat{x}_k$  é a estimativa do estado *a posteriori* (do instante anterior a  $k$ ).

É importante ressaltar que o principal vício do EKF é que as distribuições das variáveis aleatórias deixam de ser normais após as suas transformações não-lineares. O EKF é simplesmente um estimador de estados *ad hoc* que aproxima a otimalidade da regra de



Bayes através de linearização.

### A Origem Computacional do EKF

Para estimar um processo com relações não-lineares nos estados ou na saída, começa-se por escrever novas equações que linearizam uma estimativa em torno de (3.40) e (3.41).

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \tilde{x}_{k-1}) + W w_k - 1, \quad (3.42)$$

$$y_k \approx \tilde{y}_k + H(x_k - \tilde{x}_k) + V v_k. \quad (3.43)$$

onde,

- $x_k$  e  $y_k$  são os vetores de estado e de saída,
- $\tilde{x}_k$  e  $\tilde{y}_k$  são os vetores de estado e de saída aproximados de 3.40 e 3.41,
- $\hat{x}_k$  é uma estimativa *a posteriori* do estado no instante  $k$ .
- as variáveis aleatórias  $w_k$  e  $v_k$  representam os ruídos de processo e medição como em (3.25) e (3.26).
- $A$  é a matriz Jacobiana de derivadas parciais de  $f$  com respeito à  $x$ , que é

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0),$$

- $W$  é a matriz Jacobiana de derivadas parciais de  $f$  com respeito à  $w$ , que é

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0),$$

- $H$  é a matriz Jacobiana de derivadas parciais de  $h$  com respeito à  $x$ , que é

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0),$$

- $V$  é a matriz Jacobiana de derivadas parciais de  $h$  com respeito à  $v$ , que é

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0).$$

Pode-se perceber que, por simplicidade, não se usou o índice subscrito  $k$  nas matrizes Jacobianas  $A$ ,  $W$ ,  $H$ ,  $V$ , mesmo elas sendo diferentes para cada instante de tempo.

Agora, define-se uma nova notação para o erro de predição,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k, \quad (3.44)$$

e para o resíduo de medição,

$$\tilde{e}_{y_k} \equiv y_k - \tilde{y}_k. \quad (3.45)$$

Não se deve esquecer que, na prática, não se tem acesso a  $x_k$  em (3.44), e ele representa o vetor de estados atual, ou seja, o que está se tentando estimar. Por outro lado, tem-se acesso a  $y_k$  em (3.45), que é a medida atual que se está usando para estimar  $x_k$ . Usando-se (3.44) e (3.45) pode-se escrever equações para o erro de processo como,

$$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k, \quad (3.46)$$

$$\tilde{e}_{y_k} \approx H\tilde{e}_{x_k} + \eta_k, \quad (3.47)$$

onde  $\varepsilon_k$  e  $\eta_k$  representam novas variáveis aleatórias independentes com média zero e matrizes de covariância  $WQW^T$  e  $VRV^T$ , com  $Q$  e  $R$  como em (3.25) e (3.26), respectivamente.

Pode-se perceber que as equações 3.46 e 3.47 são lineares e que elas são bastante semelhantes às equações 3.23 e 3.24 do filtro de Kalman discreto. Dessa forma, pode-se usar o resíduo de medição atual  $\tilde{e}_{y_k}$  em (3.45) e um segundo (hipotético) filtro de Kalman para estimar o erro de predição  $\tilde{e}_{x_k}$  dado por (3.46). Essa estimativa, chama-se  $\hat{e}_k$ , pode ser usada em (3.44) para obter estimativas do estado *a posteriori* para o processo original não-linear como

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k. \quad (3.48)$$

As variáveis aleatórias em (3.46) e (3.47) têm aproximadamente as seguintes distribuições de probabilidade:

$$p(\tilde{e}_{x_k}) \sim N(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T])$$

$$p(\varepsilon_k) \sim N(0, WQ_kW^T)$$

$$p(\eta_k) \sim N(0, VR_kV^T)$$

Dadas essas aproximações e deixando que o valor predito de  $\hat{e}_k$  seja zero, a equação do filtro de Kalman utilizada para estimar  $\hat{e}_k$  é

$$\hat{e}_k = K_k \tilde{e}_{y_k}. \quad (3.49)$$

Substituindo 3.49 em 3.48 e fazendo uso de 3.45, pode-se ver que o segundo (hipotético) filtro de Kalman não é necessário.

$$\hat{x}_k = \tilde{x}_k + K_k \tilde{e}_{y_k} = \tilde{x}_k + K_k (y_k - \tilde{y}_k) \quad (3.50)$$

A equação 3.50 pode ser utilizada agora para a atualização da medição no filtro de Kalman estendido com  $\tilde{x}_k$  e  $\tilde{y}_k$  que vêm de (3.40) e (3.41) e o ganho de Kalman  $K_k$  que vem da primeira equação da Tabela 3.2, com a substituição apropriada da covariância do erro de medição.

O conjunto completo de equações do EKF é mostrado nas Tabelas 3.3 e 3.4. Percebe-se que o  $\tilde{x}_k$  foi substituído por  $\hat{x}_k^-$  para que se mantivesse a notação do estado *a priori* utilizada anteriormente, e que agora foram inseridos nas matrizes Jacobianas  $A$ ,  $W$ ,  $H$  e  $V$ , o subscrito  $k$ , para reforçar a idéia que ele são diferentes, e precisar ser recalculados a cada instante de tempo.

Tabela 3.3: *Equações de predição do EKF.*

$$\begin{aligned}\hat{x}_k^- &= f(\hat{x}_{k-1}, u_{k-1}, 0) \\ P_k^- &= A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T\end{aligned}$$

Assim como o filtro de Kalman discreto, as equações de predição na Tabela 3.3 projetam as estimativas do estado e da covariância do instante anterior  $k - 1$  para o instante atual  $k$ . A função  $f$ , na primeira equação da Tabela 3.3 vem da equação 3.40,  $A_k$  e  $W_k$  são as Jacobianas do processo no passo  $k$ , e  $Q_k$  é a covariância do ruído de processo (ver equação 3.25) no instante  $k$ .

Tabela 3.4: *Equações de correção do EKF.*

$$\begin{aligned}K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (y_k - h(\hat{x}_k^-, 0)) \\ P_k &= (I - K_k H_k) P_k^-\end{aligned}$$

Assim como no filtro discreto de Kalman, as equações na Tabela 3.4 corrigem as estimativas do estado e da covariância com a medida  $y_k$ . A função  $h$  na segunda equação da Tabela 3.4 vem da equação 3.41,  $H_k$  e  $V_k$  são as Jacobianas de medição no instante  $k$ , e  $R_k$  é a covariância do ruído de medição (ver equação 3.26), no instante  $k$ .

A operação básica do EKF é a mesma do filtro de Kalman linear discreto mostrado na Figura 3.7. A Figura 3.8 oferece uma visão da operação do EKF, combinando o diagrama da Figura 3.6 com as equações das Tabelas 3.3 e 3.4.

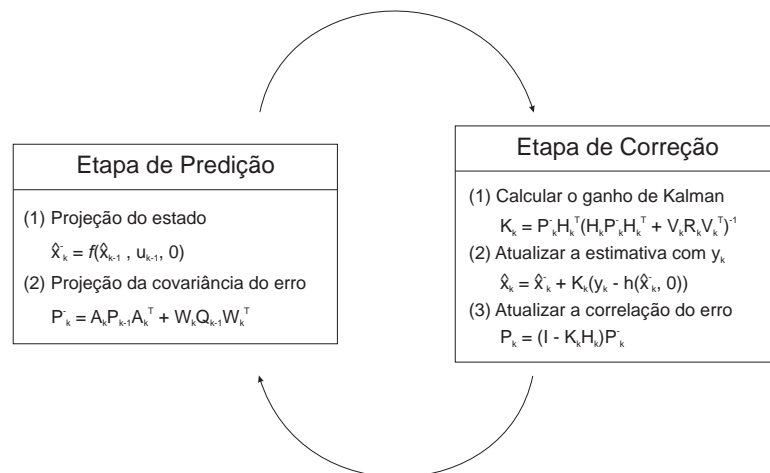


Figura 3.8: *Uma ilustração completa da operação do EKF, combinando o diagrama da Figura 3.6 com as equações das Tabelas 3.3 e 3.4.*

Uma importante característica do EKF é que a matriz Jacobiana  $H_k$  na equação do ganho de Kalman  $K_k$  serve para propagar corretamente apenas a componente relevante da informação medida. Por exemplo, se não existe um mapeamento um-para-um entre a

medida  $y_k$  e o estado, via a função  $h$ , a Jacobiana  $H_k$  afeta o ganho de Kalman de maneira que ele apenas magnifique a porção do resíduo  $y_k - h(\hat{x}_k^-, 0)$  que afeta o estado. É claro que se todas as medidas não têm um mapeamento um-para-um entre a medida  $y_k$  e o estado via  $h$ , então, como pode-se esperar o filtro irá rapidamente divergir. Nesse caso, o processo é não observável.

### 3.7 O Filtro de Kalman *Unscented*

O filtro de Kalman *unscented* baseia-se em uma transformação que foi denominada transformação *unscented* e que será apresentada na seção a seguir. Para explicar o UKF, do inglês (*Unscented Kalman Filter*), refere-se à abordagem encontrada em [Aguirre 2007].

Considerando-se dois vetores de variáveis aleatórias  $x$  e  $y$ , que se relacionam de maneira não-linear através de uma função  $f$ , tal que  $y = f(x)$ . Se for possível obter um número relativamente grande de realizações de  $x$  e  $y$ , então, torna-se possível calcular as médias amostrais  $\bar{x}$  e  $\bar{y}$ , assim como suas matrizes de covariância (amostral). Deve ser observado que para executar tais cálculos, o conhecimento da função  $f$  não é necessário.

Observando a descrição do filtro de Kalman, nas seções anteriores, percebe-se que a maior dificuldade, no caso de sistemas não-lineares é a inexistência de expressões matemáticas para o cálculo da matriz de covariância *a posteriori*. No caso do EKF, a solução foi linearizar o sistema em torno de uma estimativa do vetor de estado e lançar mão das expressões disponíveis para as matrizes de covariância de sistemas lineares.

Uma outra solução para casos não-lineares seria a de tomar um número relativamente grande de realizações de  $x$  e propagá-las através da função não-linear  $f(\cdot)$ , obtendo, assim, o mesmo número de realizações de  $y$ . Depois, no lugar de tentar derivar uma expressão que descreva a matriz de covariância de  $x$  e da função  $f$ , pode-se determinar numericamente as referidas matrizes, usando-se as definições de covariância.

Um dos efeitos colaterais dessa abordagem está na necessidade de se manipular, a cada iteração, um grande número de realizações de  $x$  e  $y$ . Porém, uma solução foi proposta por Julier e Uhlmann, em 1994 [Julier et al. 1995]. Eles apontam para o uso de poucas, porém representativas realizações de  $x$  e  $y$ , de maneira que não haja grandes diferenças nas matrizes de covariâncias, caso estas tenham sido calculadas utilizando-se todas as realizações ou apenas estas poucas representativas. A escolha das realizações representativas, bem como a formulação resultante, são hoje conhecidas como a transformação *unscented*, que será apresentada na próxima seção.

#### 3.7.1 A transformação *unscented*

A transformação *unscented* reduz um potencialmente grande número de vetores de estado (as realizações de  $x$ ) a um pequeno número de vetores, chamados de pontos sigma. Deve ser notado que, cada ponto sigma refere-se a um vetor  $\in R^n$ . Os pontos sigma são representados por  $X_i, i = 0, 1, \dots, 2n_a$ , em que  $n_a$  é a dimensão do vetor de estado aumentado, que será explicado posteriormente.

A escolha dos pontos sigma  $\mathcal{X}_i$  deve ser tal que a média e a covariância desses pontos sejam iguais à média e à covariância do vetor de estado aumentado, ou seja,

$$\frac{1}{2\eta_a} \sum_{i=1}^{2\eta_a} \mathcal{X}_i = \bar{x} \quad e \quad cov\{\mathcal{X}_i\} = P, \quad (3.51)$$

onde  $P$  é a matriz de covariância de  $x$ , e  $\bar{x}$  indica a média.

Os pontos sigma, que satisfazem a Equação 3.51, podem ser propagados pelo sistema não-linear  $f(\cdot)$  de modo a gerar um conjunto de “pontos sigma propagados”  $\mathcal{Y}_i = f(\mathcal{X}_i)$ . A média e a covariância desse novo conjunto de pontos  $\mathcal{Y}_i$  podem ser determinadas de maneira numérica utilizando relações conhecidas para a média e a variância amostrais. Os valores, assim determinados, podem então ser tomados como aproximações (estimativas) da média e da matriz de covariância do vetor de estado propagado  $y$ , ou seja,

$$\frac{1}{2\eta_a} \sum_{i=1}^{2\eta_a} \mathcal{Y}_i \approx \bar{y} \quad e \quad cov\{\mathcal{Y}_i\} \approx P_{yy}.$$

O vetor de estado aumentado é dado por:

$$x_k^a = [x_k^T w_k^T v_k^T]^T, \quad (3.52)$$

cuja dimensão é  $\eta_a = 2n + m$ , portanto,  $x_k^a \in R^{2n+m}$  sendo  $n$  a dimensão do espaço de estado original, e  $m$ , o número de saídas. Como o vetor de estado foi aumentado, a matriz de covariância dos erros de estimação, deve também ser aumentada, referindo-se ao vetor aumentado, da seguinte forma:

$$P^a = \begin{bmatrix} P & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}, \quad (3.53)$$

em que  $P^a \in R^{(2n+m)(2n+m)}$ .

Os pontos sigma são escolhidos da seguinte forma:

$$\begin{cases} \mathcal{X}_0^a(k|k) = \hat{x}^a(k|k) \\ \mathcal{X}_i^a(k|k) = \hat{x}^a(k|k) + [\sqrt{(\eta_a + \lambda)P^a(k|k)}]_i \\ \mathcal{X}_{i+\eta_a}^a(k|k) = \hat{x}^a(k|k) - [\sqrt{(\eta_a + \lambda)P^a(k|k)}]_i, \end{cases} \quad (3.54)$$

onde a nomenclatura foi alterada para evitar conflito de índices. Deve ser notado que a nomenclatura é equivalente à utilizada em seções anteriores, ou seja,  $\hat{x}^a(k|k)$  indica  $\hat{x}_k^{a+}$ . Os pesos relativos ao cálculo da média e covariância, são escolhidos de tal forma que

$$\begin{cases} \omega_0^{(m)} = \frac{\lambda}{\eta_a + \lambda} \\ \omega_0^{(c)} = \frac{\lambda}{\eta_a + \lambda} + 1 - \alpha^2 + \beta \\ \omega_i^{(m)} = \omega_i^{(c)} = \omega_{i+\eta_a}^{(m)} = \omega_{i+\eta_a}^{(c)} = \frac{1}{2(\eta_a + \lambda)}, \end{cases} \quad (3.55)$$

sendo  $i = 1, \dots, \eta_a$  e  $[\sqrt{(\cdot)}]_i$  podendo ser a  $i$ -ésima linha ou coluna da matriz resultante da raiz quadrada na Equação 3.54 (triangular inferior da fatorização de Cholesky [Haykin 2001a]), dependendo da formação da matriz. Especificamente, se a matriz for do tipo  $A^T A$ , as linhas de  $A$  devem ser usadas. No caso de a matriz ser  $AA^T$ , as colunas de  $A$  devem ser usadas para compor os pontos sigma (de acordo com Julier e Uhlmann [Julier & Uhlmann n.d.]). Os pesos calculados na Equação 3.55 serão usados para ponderar os pontos sigma no cálculo da estimativa de  $\hat{x}$ ,  $\hat{y}$  e nas matrizes de covariância, como mostrado em 3.56.

O parâmetro  $\lambda = \alpha^2(L + \kappa) - L$  pode ser visto como um parâmetro de escala. A constante  $\alpha$  determina o espalhamento dos pontos sigma em volta de  $\hat{x}^a(k|k)$ , e é usualmente definido como um valor positivo próximo a zero ( $1 \geq \alpha \geq 10^{-4}$ ). A constante  $\kappa$  é um parâmetro de escala secundário, que é geralmente definido como  $3 - L$  [Julier et al. 1995], e  $\beta$  é usado para incorporar conhecimento *a priori* sobre a distribuição de  $x$  (para distribuições gaussianas,  $\beta = 2$  é ótimo).

A escolha dos pontos sigma de acordo com as equações em (3.54) garante que os dois primeiros momentos (média e covariância) sejam exatamente ajustados. Além disso, se a distribuição de densidade de probabilidade do vetor de estado for simétrica, o terceiro momento, que é nulo, também é ajustado de maneira exata. Existem ainda outras maneiras de se escolher os pontos sigma. Julier e Uhlmann discutem algumas delas em [Julier & Uhlmann n.d.].

O filtro de Kalman *unscented* pode ser então definido com as seguintes equações

$$\left\{ \begin{array}{l} \mathcal{X}_i^x(k+1|k) = f[\mathcal{X}_i^x(k|k), u(k), \mathcal{X}_i^w(k|k)] \\ \mathcal{Y}_i(k+1|k) = h[\mathcal{X}_i^x(k+1|k), \mathcal{X}_i^r(k|k)] \\ \hat{x}(k+1|k) = \sum_{i=0}^{2\eta_a} \omega_i^{(m)} \mathcal{X}_i^x(k+1|k) \\ \hat{y}(k+1|k) = \sum_{i=0}^{2\eta_a} \omega_i^{(m)} \mathcal{Y}_i(k+1|k) \\ P(k+1|k) = \sum_{i=0}^{2\eta_a} \omega_i^{(c)} [\mathcal{X}_i^x(k+1|k) - \hat{x}_i(k+1|k)][\mathcal{X}_i^x(k+1|k) - \hat{x}_i(k+1|k)]^T \\ P_{yy}(k+1|k) = \sum_{i=0}^{2\eta_a} \omega_i^{(c)} [\mathcal{Y}_i(k+1|k) - \hat{y}_i(k+1|k)][\mathcal{Y}_i(k+1|k) - \hat{y}_i(k+1|k)]^T \\ P_{xy}(k+1|k) = \sum_{i=0}^{2\eta_a} \omega_i^{(c)} [\mathcal{X}_i^x(k+1|k) - \hat{x}_i(k+1|k)][\mathcal{Y}_i(k+1|k) - \hat{y}_i(k+1|k)]^T \\ K(k+1) = P_{xy}(k+1|k)P_{yy}^{-1}(k+1|k), \end{array} \right. \quad (3.56)$$

sendo  $i = 0, \dots, 2\eta_a$  e  $\mathcal{X}_i^a(k-1|k-1)$  são dados por (3.54) e

$$\mathcal{X}^a = [(\mathcal{X}^x)^T [(\mathcal{X}^w)^T [(\mathcal{X}^r)^T]^T]^T$$

As duas primeiras equações em (3.56) fazem a propagação dos pontos sigma pelo modelo não-linear. As próximas duas equações fornecem estimativas das médias dos vetores propagados. A quinta equação é responsável pela estimativa da matriz de covariância (amostral). As próximas duas são equações intermediárias, necessárias para o cálculo do ganho de Kalman na última equação.

As equações de correção podem ser escritas da seguinte maneira:

$$\begin{cases} \hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[y(k+1) - \hat{y}(k+1|k)] \\ P(k+1|k+1) = P(k+1|k) - K(k+1)P_{yy}(k+1|k)K^T(k+1), \end{cases} \quad (3.57)$$

lembrando que  $\hat{x}(k+1|k+1) = \hat{x}_{k+1}^+$  e  $P(k+1|k+1) = P_{k+1}^+$ . Pode-se perceber que as equações de correção do KF, EKF e UKF são idênticas, mesmo a propagação sendo realizada por um modelo linear no caso do KF e modelos não-lineares nos casos do EKF e UKF. Pode-se perceber também que no caso do EKF, assim como no UKF, a média e a covariância são aproximações. No primeiro a aproximação é analítica, enquanto que o outro recorre a uma abordagem estatística.

A escolha da matriz de covariância geralmente requer determinar valores para  $Q$  e  $R$ .  $Q$  pode ser utilizada para refletir o grau de confiança no modelo utilizado na implementação do UKF, enquanto que  $R$  deve refletir o nível de ruído nas medições, uma vez que consiste da matriz de covariância do ruído. Em aplicações práticas, pode-se tomar  $Q$  como uma matriz diagonal em que os elementos da diagonal são as variâncias dos erros de predição de um passo a frente, de cada elemento do vetor de estados, obtidos utilizando o modelo em questão.

## 3.8 Validação de Modelos

A validação dos modelos tem o objetivo de avaliar se a obtenção do modelo do processo foi bem sucedida, ou seja, se para um determinado uso, o modelo representa satisfatoriamente o processo real em questão. Como dito no início do capítulo, nenhum modelo representa o sistema real em todos os aspectos. Dito isso, a validação passa a ser dependente da aplicação. Deve-se verificar se para determinado uso o modelo é válido, ou seja, se para àquelas situações ele representa satisfatoriamente o sistema real.

Esta seção tem como objetivo apresentar algumas técnicas que podem ser usadas na verificação da qualidade de um modelo, onde cada uma delas fornecerá uma medida de qualidade que também variará com a aplicação. Na prática, deve-se procurar validar um modelo considerando um conjunto de indicadores no lugar de apenas um. Além disso, deve-se atentar para o fato de que nem sempre pode-se usar todas as ferramentas em qualquer situação, ou em qualquer modelo. Finalmente, o fato mais importante é observar a utilidade do modelo para o uso prático da aplicação em questão.

### 3.8.1 Simulação

Na simulação, deseja-se saber se o modelo reproduz, ao longo do tempo, os dados observados do sistema real. A simulação é um procedimento relativamente simples e muito comum, porém deve-se atentar para alguns detalhes, como:

- os dados utilizados na etapa de validação, não devem ser os mesmos utilizados na obtenção do modelo. Uma vez obtido um modelo final, é interessante verificar sua capacidade de generalização, ou seja, a capacidade de responder como o sistema

real a estímulos cujos quais ele nunca foi excitado. Esse conjunto de dados pode ser chamado de conjunto de validação.

- ao efetuar a validação, deve-se certificar de que os dados de validação foram obtidos na mesmas condições de operação que o conjunto de dados usados na obtenção do modelo, principalmente quando o sistema real, geralmente não-linear, é representado por modelos lineares. Nesse caso, o uso de dados de validação coletados em situações de operação diferentes seguem uma outra dinâmica e certamente não corresponderá à que o modelo foi obtido.
- em outras situações, problemas semelhantes podem ser observados mesmo quando ambos os conjuntos de dados forem obtidos em iguais condições de operação, no caso de o sistema ser variante no tempo.

### 3.8.2 Análise de Resíduos

O procedimento de análise de resíduos tem o objetivo de informar se os parâmetros do modelo identificado foram ou não estimados corretamente, ou seja, tal procedimento tem a capacidade de indicar se os erros observados no processo de simulação ocorreram devido a problemas na estimação de parâmetros ou se a dinâmica que produziu os dados não pode ser completamente representada pelo modelo em questão.

Esse procedimento é realizado fazendo-se uma análise estatística no vetor de resíduos  $\xi = y - \psi\hat{\theta}$ , que resume-se a verificar se os resíduos são aleatórios ou não.

Do ponto de vista de validação de modelos, a motivação de se verificar quão aleatórios são os resíduos pode ser entendida considerando-se que os resíduos são a parte dos dados que o modelo não conseguiu explicar.

Se ao testar o vetor de resíduos for verificado que se trata de uma variável aleatória branca, significa que não há informação útil nos resíduos, e que o modelo está representando bem a dinâmica do sistema. Por outro lado, se os ruídos não forem brancos, haverá informação neles, indicando que o modelo não conseguiu explicar tudo que havia na dinâmica dos dados. Nesse caso, pode-se tentar escolher um novo conjunto de regressores e estimar os parâmetros novamente.

### 3.8.3 Validação para Aplicações em Malha Fechada

Caso o modelo em questão vá ser utilizado em aplicações de malha fechada, é recomendado que se observe a resposta do modelo nessa situação, visto que modelos que apresentam uma resposta próxima a resposta do sistema real em malha aberta nem sempre apresentam uma resposta aceitável em malha fechada. Mais informações e exemplos sobre esse caso, consultar [Aguirre 2007].

O próximo capítulo descreve com detalhes a arquitetura do sistema FDI proposto, além de compará-la com a abordagem baseada em resíduos, também apresentada a seguir. A teoria apresentada no Capítulo 2 e no atual é essencial para o entendimento da arquitetura proposta e das demais, apresentadas com intuito de comparação.



---

## Capítulo 4

# Proposta do Trabalho

---

Uma grande parte das soluções encontradas na literatura para sistemas de detecção e isolamento de falhas é composta de duas fases principais: estimação de resíduos e classificação de falhas. A Figura 4.1 mostra um diagrama de blocos que representa essa abordagem. No módulo de estimação de resíduos normalmente se utiliza um modelo do processo, que tem como propósito simular a resposta do sistema real, que está sendo executado em paralelo a este. O bloco representando o modelo é o principal componente do módulo estimador e a qualidade da identificação realizada (ver Capítulo 3) influenciará diretamente na confiabilidade dos resíduos fornecidos para o próximo módulo, o classificador. Como visto no Capítulo 3, existem diversas técnicas já consolidadas na literatura usadas para a identificação de processos reais e pode-se decidir entre uma delas para ser utilizada nessa fase.

Resíduos, como já visto no Capítulo 2, são sinais que representam os desvios de comportamento do processo real em relação ao seu próprio comportamento em situações normais (fora de situações que representem falhas ou mau funcionamentos), que foi “aprendido” pelo modelo em sua fase de treinamento. Os resíduos podem ser obtidos calculando-se a diferença entre a saída ou estados do processo e a saída ou estados estimados pelo modelo. Estes, por sua vez, são fornecidos como entrada para o módulo classificador, que é responsável por fazer a detecção, identificação e classificação das falhas. A partir deste ponto, se chamará esta abordagem de arquitetura clássica.

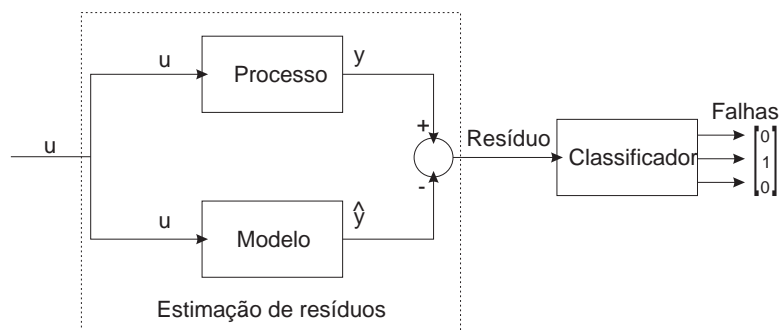


Figura 4.1: *Arquitetura clássica de sistemas de detecção de falhas.*

Para efeito de comparação com a proposta central desta tese, desenvolveu-se um sistema FDI baseado na arquitetura clássica [Fernandes et al. 2007c], [Silva et al. 2007],

[Fernandes et al. 2007a]. Ele foi aplicado a um processo com dois tanques em cascata, controlado por um ambiente de redes industriais Foundation Fieldbus [*Technical Information: Foundation Fieldbus* n.d.]. A descrição completa desse processo pode ser encontrada no Capítulo 5, onde mostra-se também resultados comparativos entre a arquitetura clássica e a arquitetura proposta nesta tese.

A abordagem baseada em resíduos, apesar de ser bem estabelecida na literatura, apresenta algumas deficiências, como: a possibilidade de haver mais de uma falha com o mesmo padrão de resíduo e a tendência à divergência, com o passar do tempo, entre o modelo e o processo real. A primeira deficiência dificulta a identificação de falhas diferentes porém, com mesma assinatura, ou seja, mesmo comportamento dos resíduos, enquanto a segunda dificulta a detecção de falhas incipientes (ver Capítulo 2). Outro aspecto limitador é a deficiência em se identificar falhas simultâneas.

Diferentemente da arquitetura clássica, a presente proposta pretende empregar um módulo de identificação paramétrica do processo, fazendo uso de seu modelo fenomenológico ao invés do módulo de estimação de resíduos. Com isto, pretende-se extinguir as deficiências apresentadas pela abordagem baseada em resíduos. Mais especificamente, a proposta desse trabalho consiste em uma arquitetura para um sistema de detecção e isolamento de falhas para processos industriais que seja capaz de precisar informações importantes, não disponibilizadas pelos sistemas FDI atualmente encontrados na literatura. São elas: a quantificação do desvio sofrido pelo processo em consequência da falha, a localização física da mesma e a tendência à falha, quando os desvios são ainda muito suaves para se indicar uma falha propriamente dita. Considera-se essas informações de extrema utilidade para o acompanhamento da saúde do processo monitorado. Além disso, elas poderão ser usadas para alimentar outras ferramentas (como: analisadores de tendência, preditores, etc.) a fim de se extrair informações complementares sobre o processo em questão.

O diferencial do trabalho está na identificação paramétrica do processo em estado de falha, ou seja, ele será capaz de mostrar o modelo matemático do sistema no momento em que ele se altera em consequência de uma falha, no caso em que estas se refletem em pelo menos um dos parâmetros monitorados. A Figura 4.2 mostra o esquema que pretende-se empregar para se conseguir inferir os parâmetros do modelo fenomenológico do processo durante a sua execução. Dessa forma, pode-se monitorar as alterações dos parâmetros da equação, caso hajam, que servem como indicativo de alguma anormalidade (faltas e falhas).

O bloco rotulado “processo” representa o sistema dinâmico ou planta industrial ao qual se deseja aplicar o sistema FDI. Naturalmente, este bloco tem como entrada o sinal  $u$  que pode ser o sinal do controlador, caso esteja em malha fechada [Ogata 1998], e fornece como saída o sinal chamado de  $y$ . O bloco rotulado “modelo fenomenológico nominal”, representa a equação matemática que rege o processo físico em questão, com os valores dos parâmetros  $\theta$  corretos, representando o funcionamento normal do processo. Esta equação deve ser previamente conhecida.

O bloco rotulado “algoritmo de estimação” refere-se ao algoritmo responsável por estimar os parâmetros do modelo, baseando-se apenas na entrada  $u$  e na saída  $y$  do processo. O modelo do processo, cuja equação é conhecida, está representado por um bloco menor

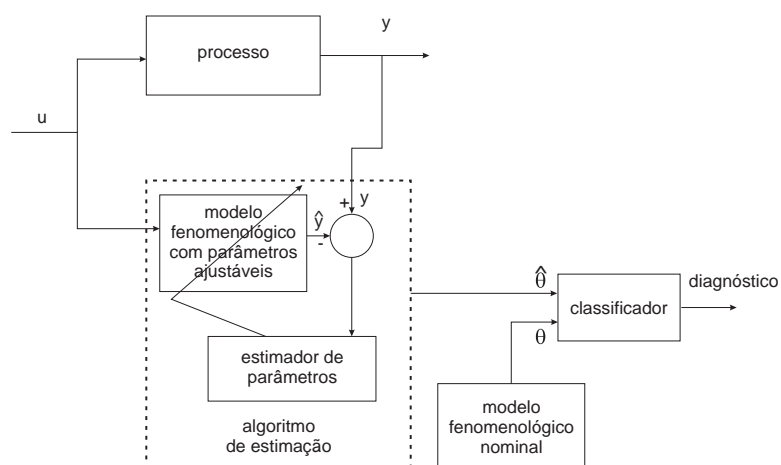


Figura 4.2: Arquitetura da ferramenta proposta.

que se encontra dentro do algoritmo de estimação. Neste caso, seus parâmetros serão ajustados pelo bloco “estimador de parâmetros” que é responsável por fornecer estimativas de parâmetros  $\hat{\theta}$  que minimizem o erro entre a saída do processo  $y$  e a saída do modelo que se encontra dentro do algoritmo de estimação.

Os parâmetros  $\theta$ , fixos e previamente conhecidos, juntamente com os parâmetros estimados pelo algoritmo  $\hat{\theta}$  são fornecidos a um bloco chamado “classificador” responsável pelo diagnóstico das falhas. Vale lembrar que nesta abordagem, a etapa de classificação é bastante simplificada e direta, podendo até mesmo ser realizada pelo operador do sistema por meio da interpretação dos valores de  $\theta$  e  $\hat{\theta}$ .

A proposta se baseia na idéia de que os parâmetros do modelo  $\hat{\theta}$  serão iguais aos parâmetros do processo físico  $\theta$ , quando este último estiver em funcionamento normal, ou seja, livre de faltas, falhas ou quaisquer mau funcionamentos que perturbem o seu comportamento normal. Este comportamento é representado na Figura 4.2 pelo bloco “modelo fenomenológico nominal”. Neste caso, a etapa de classificação receberá valores iguais para  $\theta$  e  $\hat{\theta}$ , devendo-se concluir, então, que o processo está em funcionamento normal.

De maneira análoga, quando o comportamento do sistema começar a apresentar irregularidades, o algoritmo responsável pela estimação dos parâmetros, baseando-se nas novas entradas e saídas  $u$  e  $y$ , respectivamente, encontrará uma combinação de parâmetros que satisfaça a essa relação de entrada/saída. Se os novos parâmetros estimados diferirem daqueles previamente encontrados ou fornecidos pelo modelo do processo, o classificador terá condições de diagnosticar uma anormalidade no processo. A amplitude da diferença entre  $\theta$  e  $\hat{\theta}$ , será usada para decidir entre faltas ou falhas, além de classifica-las. Adicionalmente, como cada parâmetro  $\hat{\theta}$  está diretamente ligado a alguma grandeza física do processo, simplifica-se consideravelmente a etapa de isolamento, ou localização da anormalidade.

A possibilidade de se estimar os parâmetros que satisfaçam a relação de entrada/saída dos dados, obedecendo a equação do modelo, permite monitorar divergências entre os parâmetros do processo  $\theta$  e parâmetros estimados  $\hat{\theta}$  desde o começo de sua formação. E

mesmo que esta diferença ainda não seja suficiente para gerar um alarme ou acusar algo mais grave, ela não deixa de ser um indicativo de um pequeno desvio de comportamento. O acompanhamento desse parâmetro estimado, pode servir para a predição de problemas mais sérios e, dependendo de como ele for usado, indicar períodos de manutenção preventiva, troca de componentes, entre outros. O que pode significar a ausência de surpresas indesejadas e diminuição de prejuízos.

Em síntese, a abordagem proposta traz as seguintes vantagens, se comparada às soluções clássicas:

- a capacidade de rastrear os valores corretos dos parâmetros da equação que rege o processo, quando este sai do seu comportamento normal e passa sofrer desvios causados por faltas ou falhas;
- em consequência da primeira, o sistema FDI proposto é capaz de quantificar a alteração sofrida pelo processo causada pela falha, através da comparação entre os parâmetros nominais (em funcionamento normal) e os parâmetros estimados quando o sistema encontra-se em modo de falha;
- a precisão de isolar e identificar a falha diretamente, de acordo com o modelo fenomenológico em modo de falha. Isso quer dizer que pode-se localizar a falha diretamente através da observação dos parâmetros que sofreram alterações significativas, já que estes são associados diretamente a alguma grandeza física envolvida no processo;
- A possibilidade de observar a formação de falhas incipientes e fazer previsões sobre possíveis problemas que venham a surgir em consequência desses desvios de comportamento. Como o algoritmo de estimação de parâmetros aqui proposto funciona *on-line*, elimina-se a tendência à divergência, problema este encontrado na abordagem clássica proveniente da etapa de modelagem do processo. Neste caso, a arquitetura proposta não apresenta problemas de detecção de falhas incipientes, mesmo com o passar do tempo.

Para que a implementação do sistema de detecção e isolamento de falhas citado acima seja possível, com todas as vantagens citadas, deve-se satisfazer algumas condições relacionadas ao processo físico que será monitorado. São elas:

- o conhecimento prévio da estrutura do modelo matemático do processo (equação de referência) ou a necessidade da modelagem a partir de leis físicas e químicas já conhecidas; e
- ter grandezas físicas relacionadas com as possíveis falhas que se deseja monitorar. Estas serão refletidas nos parâmetros da equação do modelo do processo.

## 4.1 Metodologia Utilizada

Várias ferramentas matemáticas foram utilizadas na tentativa de se chegar à arquitetura proposta na Figura 4.2 ou pelo menos a uma arquitetura equivalente com a capacidade de quantificar as falhas com precisão de acordo com o modelo matemático do processo.

A primeira abordagem utiliza uma rede neural MLP (*Perceptron de Múltiplas Camadas*) [Haykin 2001b], [Fu 1994] e [Miller et al. 1995] para mapear a relação entre a entrada e saída do sistema real e os parâmetros do mesmo. Dessa maneira, a saída da rede neural corresponderia aos parâmetros da equação do processo que indicam grandezas físicas envolvidas no mesmo como massas, velocidades, posições, etc. Assim, no momento em que uma falha ocorrer, causará variações em pelo menos uma das grandezas físicas, o que se refletirá em algum parâmetro da equação do processo, e conseqüentemente, será indicado por pelo menos uma das saídas da rede neural. Estas últimas, quando comparadas aos parâmetros do sistema em operação normal (parâmetros nominais), indicarão situações anormais de funcionamento, caso apresentem diferenças significativas. Ou seja, o sistema FDI nesse caso é capaz de indicar uma mudança na dinâmica do sistema, que é refletida pela variação de algum componente do processo. Além disso, ele tem como apontar o novo valor dos parâmetros que, satisfazendo a equação de referência do sistema, mapeiam os dados de entrada e saída observados do processo real. Isso quer dizer que o sistema FDI é capaz de achar a equação do processo quando o mesmo se encontra em modo de falha, e que ele pode ser usado para medir o desgaste ou o estado de componentes do sistema que são considerados invariantes no tempo.

Esta abordagem foi implementada e testada. Sua descrição detalhada e seus resultados completos podem ser encontrados em [e Luiz Affonso Guedes e Adrião D. D. Neto 2008].

Essa abordagem, apesar de teoricamente satisfazer todos os objetivos da proposta, apresenta algumas limitações que, na prática, podem ser decisivas para o funcionamento eficaz do sistema FDI. Entre elas estão as apontadas a seguir:

- a necessidade do treinamento *off-line*. Como se usa uma rede MLP com o algoritmo de treinamento supervisionado *Backpropagation* [Haykin 2001b], o treinamento da rede necessita da Figura do “professor”, ou o conjunto de dados de treinamento. Este, por sua vez, é formado por dados de entrada e saída do processo, que serão usados na entrada da rede neural, e os parâmetros do mesmo, que serão usados como referência no treinamento da rede. Ou seja, são as respostas corretas que a rede deve fornecer no momento de seu uso. Para gerar esse conjunto de dados é necessário simular o processo em modo de falha. Para isso deve-se assumir todos os possíveis processos diferentes em que o processo real pode vir a se tornar, quando o mesmo estiver em modo de falha. Dessa forma, a eficácia da implementação fica amarrada aos sistemas previstos e, à probabilidade de as falhas reais levarem o processo físico para um desses sistemas dos quais a rede foi treinada.
- na prática, como o treinamento precisa ser feito previamente de maneira *off-line*, pode-se encontrar situações não previstas durante o treinamento. Devido à necessidade de se prever os possíveis modos de falha em que o sistema pode assumir, é inevitável o risco de, em operação, o processo físico passar por uma situação não prevista e assumir um estado que a rede neural não tem como mapear seus parâmetros. Dessa forma, todas as informações sobre o sistema em modo de falha seriam inatingíveis.

Considerando os desafios citados acima, para que a proposta inicial fosse capaz de funcionar sob condições adversas em toda a escala dos parâmetros, seria necessário um

algoritmo que realizasse o ajuste dos parâmetros em tempo de execução, em outras palavras, *on-line*. Além disso, o algoritmo deve levar em consideração a dinâmica do processo, ou seja, que a saída  $y$  do mesmo depende de saídas passadas. Assim, buscou-se uma alternativa que fosse iterativa, ou seja, capaz de considerar a dinâmica do processo através apenas das medidas atuais de entrada  $u$  e saída  $y$ .

A procura por um método capaz de satisfazer todos os requisitos supracitados, onde a cada instante de tempo discreto, fosse necessário apenas fornecer os dados atuais de entrada e saída do sistema e, baseado nisso, o algoritmo fosse capaz de atualizar suas estimativas, resultou no algoritmo RLS (*Recursive Least Square*) [Haykin 2002]. Este mantém uma matriz, internamente, que é atualizada a cada iteração, sendo necessário passar para o algoritmo apenas dados do instante atual.

O RLS é normalmente utilizado em filtros adaptativos para encontrar os coeficientes do mesmo através do erro médio quadrático, que é produzido recursivamente. Porém, esse algoritmo pode ser facilmente adaptado para encontrar os parâmetros de uma representação de um sistema dinâmico, que é o problema em questão.

Dessa maneira, basta fornecer os dados de entrada e saída atuais para o algoritmo que, de posse da estrutura matemática do processo real, ele estimará os parâmetros de tal forma que o erro médio quadrático entre a saída do processo físico e do modelo seja mínimo. Baseado nesse erro, pode-se gerar uma medida de confiabilidade dos parâmetros. A diferença entre os parâmetros do processo e do modelo são, nesse caso, os sinais principais de análise para detecção e identificação de falhas.

O problema observado na abordagem supra citada se baseia na limitação do RLS de apenas contemplar sistemas lineares, ou lineares nos parâmetros. Porém, a maior parte dos sistemas dinâmicos encontrados no mundo real são, de fato, não-lineares. Muitas aplicações de engenharia são satisfeitas através de linearizações em torno de pontos particulares de operação dos sistemas reais. Porém no caso de identificação de sistemas, as linearizações causam erros grosseiros na resposta dos modelos e, muitas vezes, a dinâmica do processo real não pode ser modelada por sistemas lineares [Aguirre 2007].

A proposta mostrada na Figura 4.2, com todas as suas contribuições previamente citadas para a área de detecção e isolamento de falhas, terá seus requisitos satisfeitos com um método capaz de modelar sistemas dinâmicos não-lineares, e que não apresente os problemas encontrados nas abordagens anteriores. Com essas especificações, chegou-se aos algoritmos do Filtro de Kalman Estendido (EKF) e o Filtro de Kalman Unscented (UKF)[Haykin 2001a] [Julier & Uhlmann n.d.], derivados do Filtro de Kalman [Kalman 1960] original que, assim como o RLS, só funciona para sistemas lineares. O EKF e o UKF se utilizam de artifícios diferentes para contemplarem a estimação em sistemas não-lineares. O EKF é mais popular na literatura [Lerro & Bar-Shalom 1993], porém, o UKF apresenta algumas vantagens [Qi & Jian-Da 2008] em relação ao primeiro, como mostrado no Capítulo 5.

## 4.2 Síntese da Arquitetura

O foco desta tese está em propor uma arquitetura que traga as contribuições previamente citadas neste capítulo e listadas na Seção 1.3. A Figura 4.2 ilustra a solução em

esquema de blocos, não prendendo um algoritmo de estimação em particular à mesma. Porém, as análises realizadas na seção anterior, apontam, no caso de processos lineares, para os algoritmos RLS e KF, e no caso de processos não-lineares, para os algoritmos EKF e UKF.

A teoria acerca do filtro de Kalman e seus derivados foi apresentada na Seção 3.5. Até este ponto, sabe-se que ele funciona para estimação de estados de um sistema dinâmico linear, e no caso do EKF e UKF para sistemas não-lineares.

Porém, como observado em algumas aplicações [Haykin 2001a] ele pode ser facilmente adaptado para estimar parâmetros, ou até mesmo, estados e parâmetros. No presente caso, utiliza-se o filtro de Kalman para a estimação de parâmetros e, como consequência, as equações 3.23 e 3.24 passam a ser:

$$\begin{cases} \hat{\theta}(k+1) = I \times \hat{\theta}(k) + W(k) \\ \hat{y}(k+1) = f(u(k), \theta(k+1), y(k)) + v(k) \end{cases} \quad (4.1)$$

onde  $\hat{\theta}$  é o vetor de parâmetros estimados. Nesse caso, o ruído do processo  $w$  pode ser considerado nulo, e como a tendência dos parâmetros é de se estabilizarem, a primeira equação simplesmente estima que o vetor de parâmetros da próxima iteração  $\hat{\theta}(k+1)$  é igual ao vetor de parâmetros estimado no passo anterior  $\hat{\theta}(k)$ . Na segunda equação de (4.1) tem-se a estimativa da saída do modelo  $\hat{y}(k+1)$  onde  $f(\cdot)$  é uma função, que pode ser não linear, do sinal de entrada  $u(k)$ , dos parâmetros estimados pelo algoritmo  $\hat{\theta}(k+1)$  e da saída no instante anterior  $y(k)$ . A variável  $v(k)$  representa o ruído de medição.

Dessa forma, o algoritmo do filtro de Kalman e seus derivados EKF e UKF podem ser usados para a estimação de parâmetros e não de estados, como foi originalmente proposto. O que torna o esquema da solução ilustrada na Figura 4.2 viável e pronta para os testes de validação, encontrados no próximo capítulo.





---

# Capítulo 5

## Resultados

---

Neste capítulo são apresentados os resultados obtidos através da aplicação da arquitetura proposta em processos simulados e reais. Além disso, mostra-se o confronto entre algoritmos e arquiteturas de sistemas FDI. Primeiramente, compara-se o EKF ao UKF. Estes resultados têm o intuito de confrontar os dois algoritmos indicados pelo capítulo anterior para o caso de processos não-lineares. Esta etapa usa um processo simulado, onde confrontam-se os dois algoritmos nas tarefas de identificação e detecção de falhas abruptas e incipientes em situações idênticas. Será feita uma análise nos resultados provenientes desta etapa, apontando-se as principais deficiências das duas técnicas e suas conseqüências para um sistema FDI, com o objetivo de se escolher, definitivamente, o algoritmo que será usado nas ocasiões posteriores. Em seguida, mostra-se um comparativo de resultados e de abordagens entre a arquitetura clássica, apresentada na Figura 4.1 e a arquitetura proposta, mostrada na Figura 4.2. Depois, mostra-se um exemplo em que a arquitetura proposta monitora um processo simulado, onde este apresenta um parâmetro que varia em função do ponto de operação. Este caso ilustra o uso da técnica de *threshold* variável, indicando que, nem sempre, a variação de um determinado parâmetro, indica a ocorrência de uma falha no processo. Por fim, mostra-se os resultados da arquitetura proposta quando a mesma foi aplicada a processos reais, nesse caso, um motor DC.

### 5.1 EKF *versus* UKF

A arquitetura proposta no Capítulo 4 e detalhada na Figura 4.2 faz uso de um bloco rotulado “algoritmo de estimação”, porém, não define um algoritmo específico para ser usado neste bloco. Ainda no Capítulo 4, defende-se o uso do EKF ou UKF para o caso de se monitorar processos com dinâmica não-linear. Nesta seção, confronta-se os dois algoritmos e define-se qual deles será definitivamente utilizado nos exemplos posteriores.

Como o EKF necessita do cálculo de matrizes Jacobianas (ver Capítulo 3), escolheu-se um processo não-linear que contribuísse com um cálculo simples de sua matriz Jacobiana, visto que este processo apenas será usado a fins de comparação entre os dois algoritmos. Ele é descrito a seguir:

$$Y(k+1) = atan(a(k)U(k)) + b(k)U(k) \quad (5.1)$$

onde os valores dos parâmetros  $a$  e  $b$  são 1,2 e 0,5 respectivamente. Este processo

será utilizado com os dois algoritmos em situações similares.

A primeira comparação, naturalmente, consiste da identificação paramétrica do processo em funcionamento normal. Neste caso, os parâmetros estimados partiram dos valores  $\hat{a} = 0$  e  $\hat{b} = 0$ . Além disso, foi adicionado um ruído branco de medição com média zero e variância  $\sigma^2 = 0,04$ . Os dois algoritmos devem encontrar os valores corretos dos parâmetros, baseando-se apenas nos sinais de entrada e saída.

A Figura 5.1 mostra o desempenho do EKF na identificação do processo mostrado na Equação 5.1.

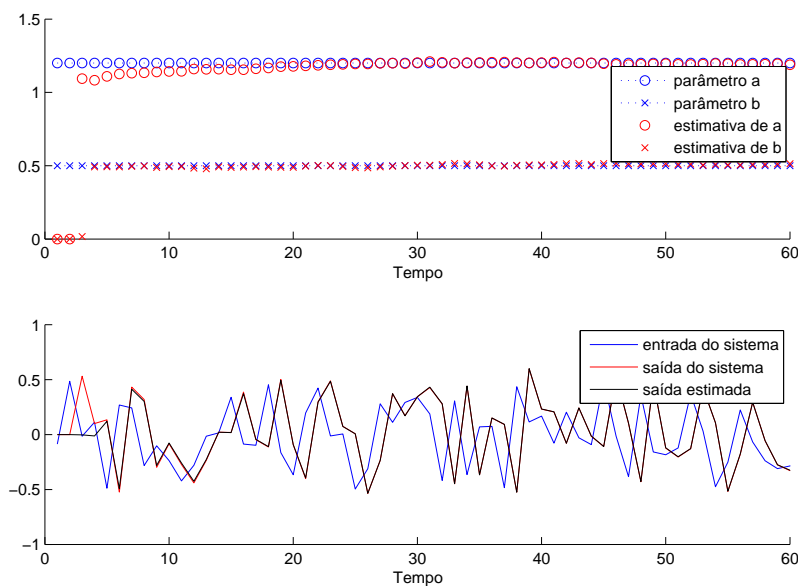


Figura 5.1: *Estimativa de parâmetros com EKF.*

O resultado do EKF na identificação do processo descrito em (5.1) deixa clara a eficácia do algoritmo. Pode-se observar, no gráfico superior, que na quarta iteração, o algoritmo dá um grande salto na direção da solução, em seguida, passa por ajustes finos, chegando a valores satisfatórios na iteração 20. O gráfico inferior, mostrando a entrada e a saída do processo, além da saída estimada pelo algoritmo, reforça o que já foi dito. Pode-se perceber que a saída e a saída estimada (linhas contínuas vermelha e preta, respectivamente) ficam muito próximas antes da iteração 10, e com diferenças imperceptíveis, nesta escala, após a iteração 20. O que confirma a confiabilidade dos parâmetros estimados.

A Figura 5.2 mostra o desempenho do UKF na tarefa de identificar o processo descrito em (5.1). O UKF foi testado com várias combinações de valores de seus parâmetros internos  $\alpha$ ,  $\beta$  e  $\kappa$  (ver Seção 3.5). Porém tais variações não mostraram diferenças significativas no desempenho do mesmo. Os valores usados daqui em diante foram  $\alpha = 0, 1$ ,  $\beta = 2$  e  $\kappa = 0$ .

Pode-se perceber, através da Figura 5.2, que, assim como o EKF, o UKF estima precisamente os parâmetros. Porém, de maneira mais eficiente. A partir da quinta iteração os

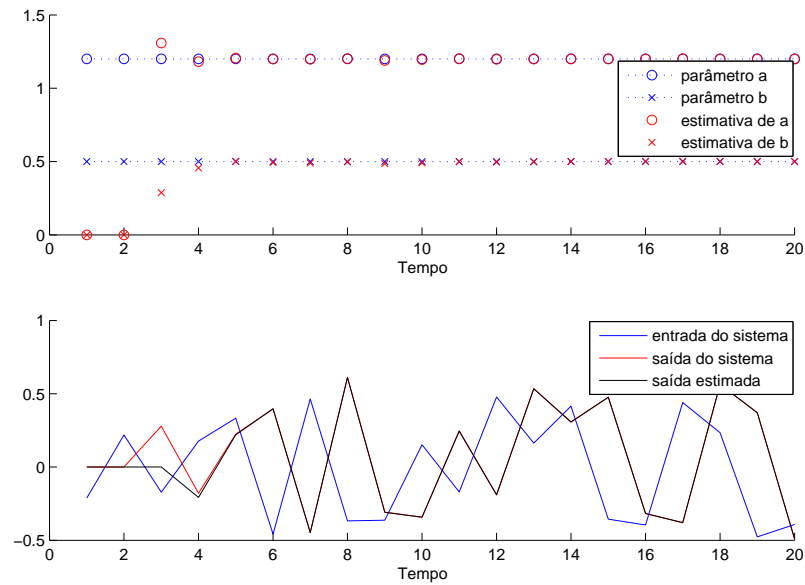


Figura 5.2: Estimativa de parâmetros pelo algoritmo UKF.

valores se estabilizam nos valores corretos. O sinal de saída estimada, encontra-se muito próximo ao sinal da saída medida, confirmando assim, a eficácia do algoritmo UKF nesta etapa.

Embora os dois algoritmos tenham mostrado eficácia na identificação do processo definido em (5.1), pode-se perceber a maior eficiência por parte do UKF. Este chega a valores precisos dos parâmetros estimados mais rapidamente.

O próximo ponto a ser comparado é a eficiência na detecção de falhas abruptas e o respectivo tempo de estabilização dos novos valores de parâmetros estimados. Para simular um situação com comportamento equivalente ao causado por uma falha abrupta, altera-se bruscamente um dos parâmetros do processo mostrado em (5.1). Isso faz com que a saída do processo torne-se diferente da saída estimada, o que caracteriza uma falha, e faz com que o algoritmo inicie uma busca pelos novos valores dos parâmetros, ou seja, pelos valores que façam a saída estimada voltar a ser similar a saída real do processo.

A Figura 5.3 mostra o desempenho do EKF na situação descrita acima.

Pode-se perceber, claramente, que o EKF encontra-se em uma situação de funcionamento normal até que na iteração de número 100 uma alteração no parâmetro  $a$  causa divergências entre a saída do processo e a saída estimada pelo algoritmo. Apesar disso, pode-se ver que, até a iteração 200 o EKF não consegue chegar ao novo valor do parâmetro.

A mesma situação foi simulada para o UKF. Tais resultados são mostrados na Figura 5.4.

Percebe-se, claramente, a eficiência do UKF nesse tipo de situação. Os valores dos parâmetros estimados, após o acontecimento da falha, são rapidamente encontrados.

O último ponto a ser confrontado entre os dois algoritmos é na detecção de falhas

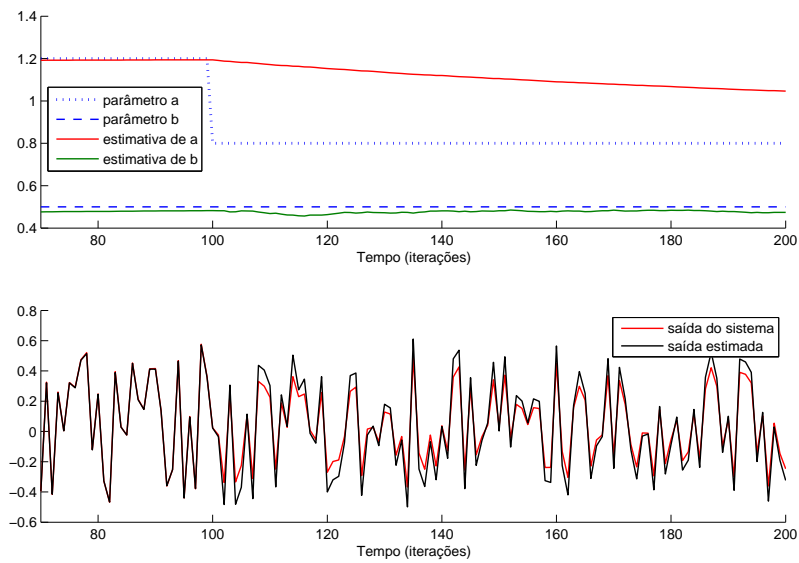


Figura 5.3: *Estimativa de parâmetros pelo EKF após uma falha abrupta.*

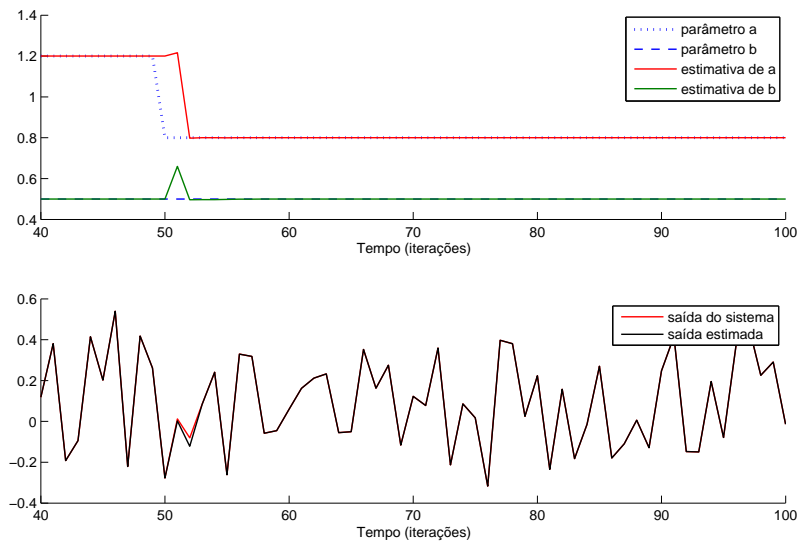


Figura 5.4: *Estimativa de parâmetros pelo UKF após uma falha abrupta.*

incipientes, e a conseqüente estimativa dos novos valores dos parâmetros após o início da falha.

A Figura 5.5 mostra o desempenho do EKF nesta situação.

Pode-se perceber que o EKF não consegue acompanhar a variação suave sofrida pelo parâmetro do processo. A Figura 5.6 mostra o desempenho do UKF em situação semelhante.

O UKF, mais uma vez, mostra-se apropriado para a tarefa de estimação de parâmetros,

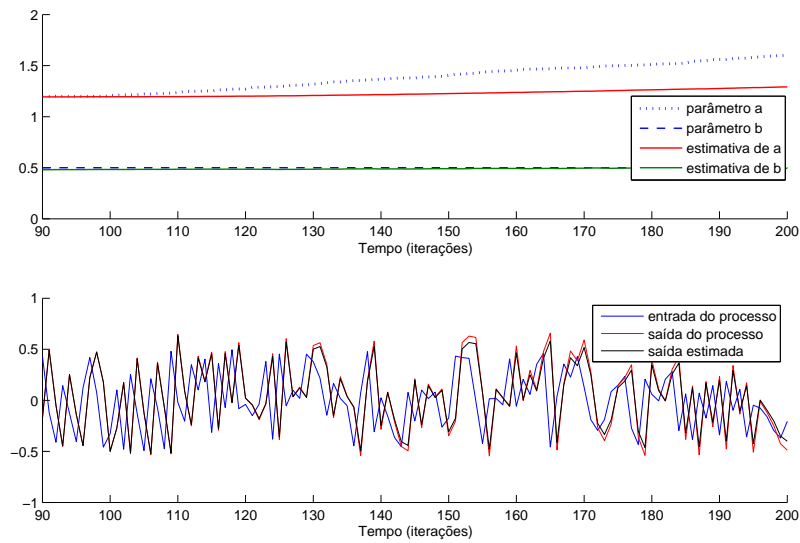


Figura 5.5: Estimativa de parâmetros pelo EKF após uma falha incipiente.

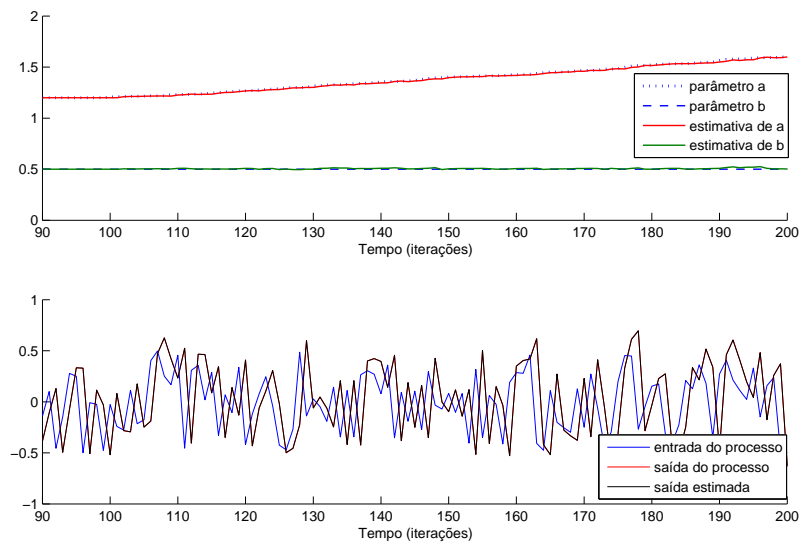


Figura 5.6: Estimativa de parâmetros pelo UKF após uma falha incipiente.

mesmo que o processo real sofra variações suaves, o que caracterizam falhas incipientes.

As publicações apontam o algoritmo EKF como o mais popular, quando se trata de estimação em sistemas não-lineares [Lerro & Bar-Shalom 1993]. Apesar disso, o EKF apresenta algumas deficiências, tais como a necessidade da diferenciação da dinâmica do sistema, assim como a susceptibilidade a *bias* e divergência das estimativas [Qi & Jian-Da 2008]. O *Unscented Kalman Filter* (UKF), por sua vez, usa o próprio modelo não-linear, em vez de linearizá-lo [Julier & Uhlmann n.d.].

Considerando-se que os modelos não-lineares são usados sem linearização, o UKF não exige o cálculo de Jacobianas ou Hessianas, e mesmo assim tem precisão de segunda ordem, enquanto que a precisão do EKF é de primeira ordem [Qi & Jian-Da 2008]. Dessa maneira o UKF torna-se mais simples de ser adaptado a cada processo que se deseja monitorar, além de ter a precisão mais apurada, o que pode ser percebido no desempenho da solução. Desta maneira, justifica-se a escolha do UKF no uso da solução, quando a mesma for aplicada a processos não-lineares.

### Análise dos resultados

Nos resultados apresentados, demonstra-se que os algoritmos EKF e UKF são eficazes na identificação paramétrica do processo definido em (5.1). Porém, além do EKF necessitar do cálculo de Jacobianas, o que, dependendo do processo, pode aumentar significativamente a complexidade da implementação, mostra-se que este não satisfaz os requisitos da arquitetura proposta, não conseguindo estimar, em tempo hábil, os valores dos parâmetros após uma falha, seja esta abrupta ou incipiente. Já o UKF mostra-se adequado para o contexto de detecção e isolamento de falhas, visto que este mostrou-se eficiente nos dois casos de falhas abordados na comparação, além de ser adaptado mais facilmente a cada processo que pretende-se monitorar.

## 5.2 Arquitetura Clássica *versus* Arquitetura Proposta

Esta seção mostra como duas diferentes arquiteturas de sistemas FDI abordam o problema de detecção e diagnóstico de falhas em processos industriais. A primeira delas refere-se a arquitetura apresentada na Figura 4.1, que é amplamente encontrada na literatura, com o uso de diferentes ferramentas empregadas nas fases de detecção e classificação de falhas. A segunda, é a arquitetura proposta no Capítulo 4 e mostrada na Figura 4.2, fazendo uso do UKF como estimador de parâmetros.

Neste comparativo leva-se em consideração, além do desempenho na detecção de falhas, a simplicidade da implementação e a precisão acerca de informações referentes à qualidade de funcionamento do processo monitorado.

As duas arquiteturas, foram acopladas a um mesmo processo dinâmico e avaliadas sob diversas condições de falhas. O processo em questão consiste de dois tanques em cascata, como mostrado na Figura 5.7.

O ciclo de funcionamento deste processo é descrito a seguir: o tanque 1 recebe água do reservatório, através da bomba. A água, então, passa pelo orifício do tanque 1 chegando ao tanque 2, onde atravessará outro orifício, até chegar novamente ao reservatório. Os dois tanques têm características físicas similares. Porém, o tanque 2 tem o nível controlado por um controlador PID [Ogata 1998].

As equações que regem o comportamento dos níveis dos dois tanques são as seguintes:

$$\dot{L}_1 = -\frac{a_1}{A_1} \sqrt{2gL_1} + \frac{K_m}{A_1} V_p \quad (5.2)$$

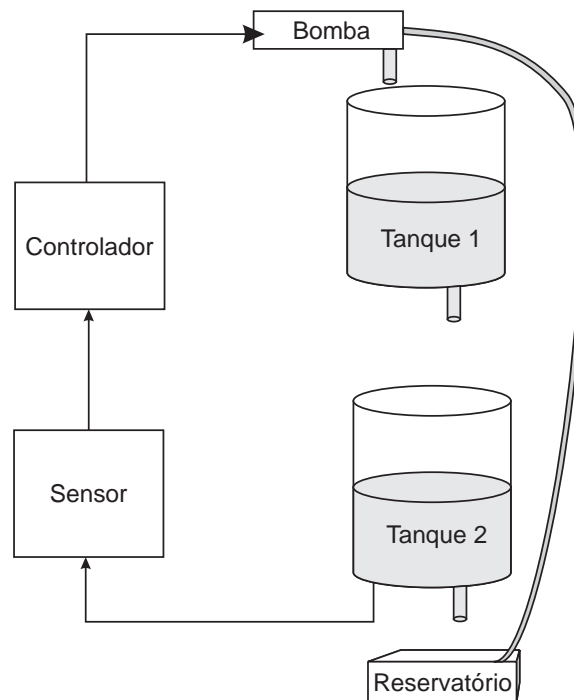


Figura 5.7: Processo com dois tanques em cascata.

$$\dot{L}_2 = -\frac{a_2}{A_2} \sqrt{2gL_2} + \frac{a_1}{A_2} \sqrt{2gL_1} \quad (5.3)$$

Todas as variáveis envolvidas nas Equações 5.2 e 5.3 são descritas na Tabela 5.1.

Tabela 5.1: Parâmetros da Planta

Nome	Significado	Valor
$K_m$	Constante da bomba	$4,6(cm^3/s)/V$
$V_p$	Tensão aplicada à bomba	$-22 < V_p < 22$
$a_1$	Diâmetro do orifício 1	$0,178139cm$
$a_2$	Diâmetro do orifício 2	$0,178139cm$
$A_1$	Área do tanque 1	$15.5179cm^2$
$A_2$	Área do tanque 2	$15.5179cm^2$
$g$	Aceleração da gravidade	$980cm/s^2$

A arquitetura clássica, nesse caso, usará redes neurais nas etapas de geração de resíduos e diagnóstico. Na primeira, duas redes neurais MLP (*Multilayer Perceptron*) farão a identificação dos tanques 1 e 2. Na segunda etapa, uma rede MLP implementa a classificação.

A identificação dos dois tanques é executada de forma *off-line*. Primeiramente excita-se o processo com um sinal e captura-se as medidas dos níveis dos dois tanques. Em seguida, treina-se a primeira rede para se comportar como o primeiro tanque, sendo a sua

entrada o sinal aplicado à bomba e a medida do nível do tanque 1, no instante anterior ao atual. A saída refere-se ao nível do tanque 1 no instante atual. A segunda rede deve-se comportar como o segundo tanque, sendo a sua entrada o nível do tanque 1 e a medida do nível do tanque 2, no instante anterior ao atual. A saída da rede, analogamente ao tanque 1, refere-se ao nível do tanque 2 no instante atual. Este processo também é descrito em [Fernandes et al. 2007c] e [Silva et al. 2007]. Para efeito de simulação computacional, utilizou-se um período de amostragem de 0,1 segundos.

A Figura 5.8 ilustra a rede neural utilizada na identificação do primeiro tanque.

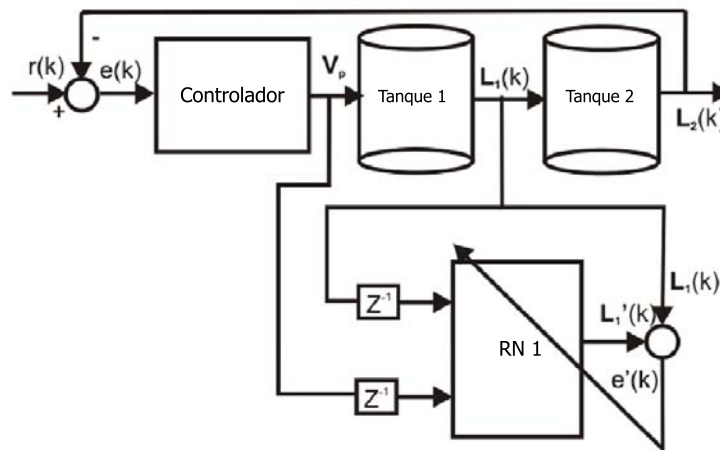


Figura 5.8: Rede Neural utilizada na identificação do tanque 1.

O bloco rotulado “RN 1” se refere a rede neural que identifica o tanque 1. Sua arquitetura é a seguinte:

- dois nós na camada de entrada  $V_p(k-1)$  e  $L_1(k-1)$ ;
- uma camada escondida com dez neurônios e função de ativação sigmóide [Haykin 2001b];
- um neurônio na camada de saída com função de ativação linear, representando o nível do tanque 1 no instante atual  $k$ ,  $L_1(k)$ .

A Figura 5.9 mostra a rede neural utilizada na identificação do segundo tanque:

O bloco rotulado “RN 2” representa a rede neural que identifica o tanque 2, com a seguinte arquitetura:

- dois nós na camada de entrada  $L_1(k-1)$  e  $L_2(k-1)$ ;
- uma camada escondida com dez neurônios e função de ativação sigmóide;
- um neurônio na camada de saída com função de ativação linear, representando o nível do tanque 2 no instante atual  $k$ ,  $L_2(k)$ .

Uma terceira rede neural aparece na arquitetura fazendo o papel de classificador de falhas. Esta, é do tipo MLP, e sua arquitetura é composta por: dois nós na camada de entrada, que receberão os resíduos provenientes da comparação dos níveis dos tanques



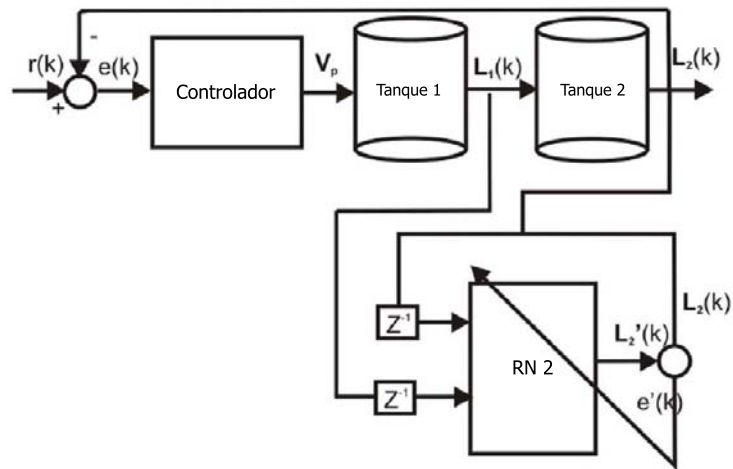


Figura 5.9: Rede Neural utilizada na identificação do tanque 2.

reais e seus respectivos modelos; oito neurônios na camada escondida, com função de ativação sigmóide, e cinco neurônios na camada de saída, com função de ativação sinal [Haykin 2001b], sendo que cada uma representa uma das situações predefinidas do processo. A Figura 5.10 mostra um diagrama de blocos representando as três redes neurais envolvidas na arquitetura.

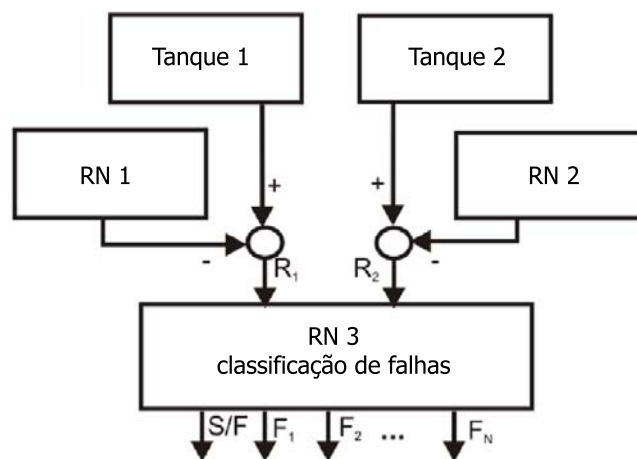


Figura 5.10: Rede Neural utilizada na classificação de falhas.

A primeira saída do classificador, rotulada “S/F” indica uma situação normal do processo, ou seja, sem a presença de qualquer falha. As outras saídas são associadas a situações que indicam falhas. A Tabela 5.2 relaciona cada falha a um número, além de mostrar o que acontece com os resíduos em cada situação.

Um zero (0) na Tabela 5.2 indica que naquela situação o resíduo não representa um desvio, ou seja, o valor da saída do tanque real deve ser muito próxima à saída da rede neural que o identifica. Um sinal de “mais” (+) quer dizer que o resíduo será maior que zero, indicando um desvio do comportamento normal em um dos tanques. Analogamente,

Tabela 5.2: Falhas e assinatura de resíduos.

Falha	Tipo da falha	R1	R2
0	ausência de falha	0	0
1	entupimento do orifício do tanque 1	+	-
2	entupimento do orifício do tanque 1	0	+
3	corrosão no orifício do tanque 1	-	+
4	corrosão no orifício do tanque 2	0	-

um sinal de “menos” (-) indica que o resíduo será menor que zero, indicando também, um desvio de comportamento de um dos tanques.

### 5.2.1 Resultados com a Arquitetura Clássica

Primeiramente, mostra-se os resultados da identificação do processo. Nesta abordagem, a identificação é de extrema importância, pois a geração dos resíduos dependem diretamente da etapa de identificação. Qualquer erro na identificação, passará como um valor “significativo” para o classificador.

A Figura 5.11 mostra resultados da identificação dos dois tanques.

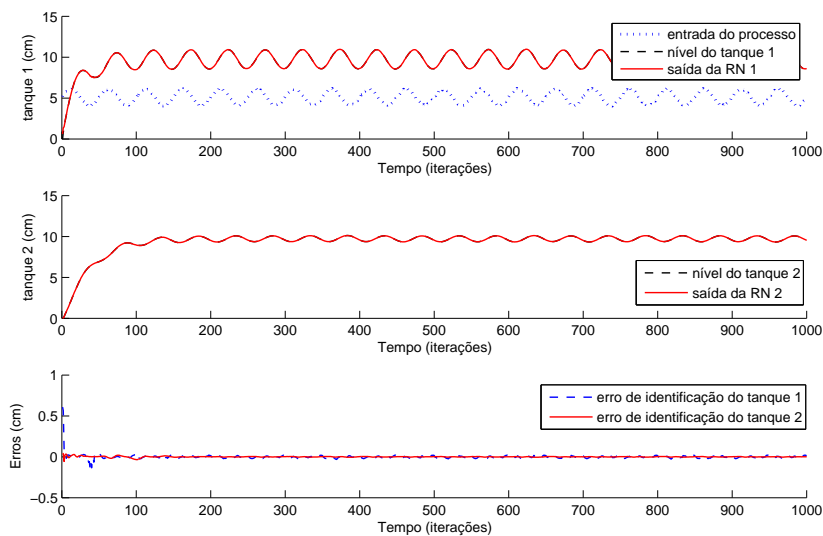


Figura 5.11: Identificação do sistema utilizando redes neurais recorrentes.

O dois primeiros gráficos mostram os níveis dos tanques 1 e 2 (em centímetros) comparados às saídas das redes neurais 1 e 2, respectivamente. No segundo, pode-se ver também o sinal de entrada aplicado ao processo. O terceiro gráfico mostra os erros entre os tanques e os modelos durante a validação dos mesmos. Os resultados de identificação mostram um desempenho satisfatório para este caso. Vale salientar que a validação foi

executada com um sinal diferente daquele usado no treinamento das redes neurais, porém, ele possui as mesmas características de média, frequência e variância, o que leva o processo para o mesmo ponto de operação do sinal de entrada usado no treinamento das redes neurais.

A Figura 5.12 testa a identificação com um sinal que apresenta a média maior que o anterior, o que leva o processo a um ponto de operação diferente do mostrado na Figura 5.11.

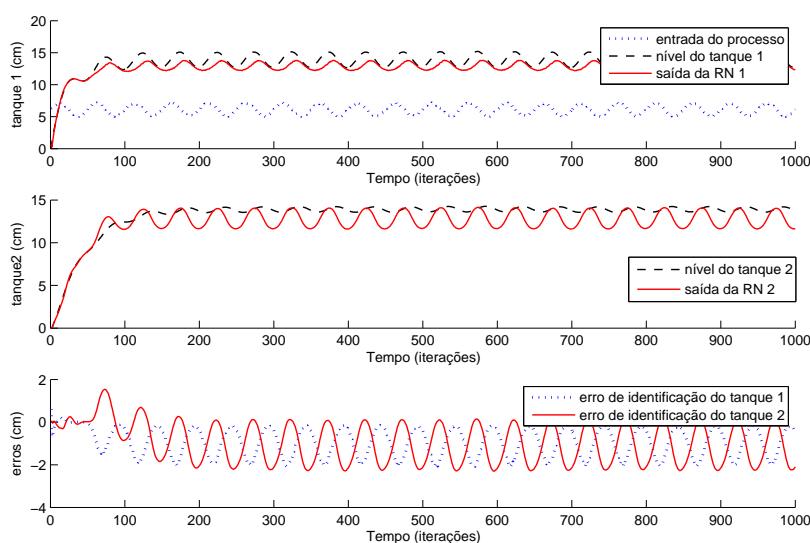


Figura 5.12: Erro da identificação por usar a entrada do processo em ponto de operação diferente.

Neste caso, pode-se perceber que as redes neurais apresentam erros na identificação. Isso acontece porque a identificação *off-line* precisa de um sinal de excitação do sistema bastante representativo. Ou seja, que leve o processo a diferentes situações de seu comportamento dinâmico. Porém, realizar uma identificação satisfatória, como mostrada na Figura 5.11, para todos os pontos de operação é um desafio extremamente complexo quando se trata de processos dinâmicos não-lineares e, além disso, essa complexidade varia conforme o processo que se deseja monitorar.

No caso dos sistemas FDI que seguem a arquitetura clássica, a identificação do processo é uma etapa de extrema importância e está diretamente ligada ao desempenho do classificador. Erros de identificação podem se confundir com resíduos, e quando passados ao classificador, este acusará falhas que não ocorreram ou, classificará erroneamente as falhas ocorridas no processo, como mostram os resultados da classificação, a seguir.

A Figura 5.13 mostra o momento em que ocorre a falha 1 da Tabela 5.2, ou seja, uma obstrução parcial no orifício de saída do tanque 1. As conseqüências desta falha são: o aumento inesperado do nível do tanque 1 e a redução momentânea do nível do tanque 2, como pode ser visto nos dois primeiros gráficos. O terceiro gráfico, mostra a geração dos resíduos, conforme indicado pela Figura 5.10

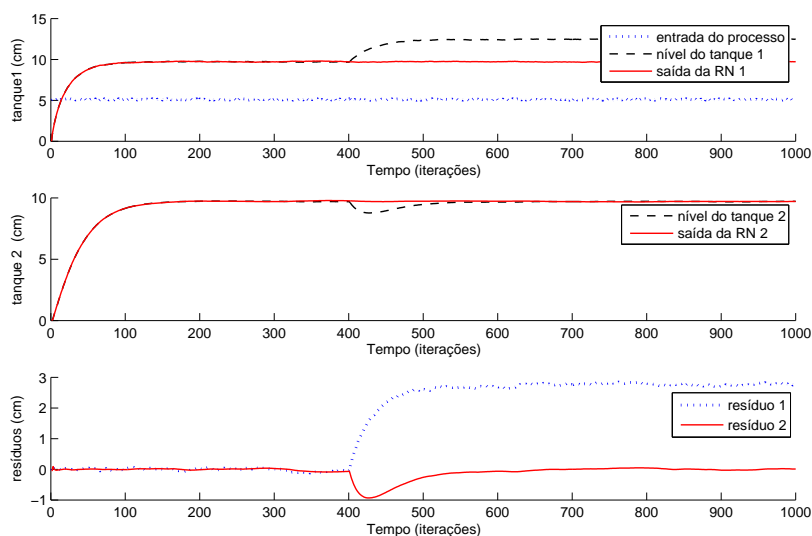


Figura 5.13: Detecção da falha 1 da Tabela 5.2 pela arquitetura clássica.

Neste caso, a identificação está satisfatória, como pode ser visualizado antes do momento da falha, que ocorre na iteração 400. Isto pode ser concluído de acordo com os valores dos resíduos muito próximos a zero em funcionamento normal do processo, como é indicado pela Tabela 5.2. No momento que a falha ocorre, os resíduos refletem o comportamento esperado, de acordo com a Tabela 5.2, e o classificador não tem dificuldades de indicar a ocorrência da falha 1. Porém, quando a identificação não está satisfatória, o que é uma realidade quando se trata de identificação *off-line* em processos dinâmicos não-lineares, o classificador pode não conseguir diagnosticar corretamente as falhas, o que dependerá do erro de identificação. A Figura 5.14 mostra uma situação onde o erro de identificação causa o diagnóstico errado pelo classificador.

Pode-se observar que mesmo antes da falha ocorrer, os resíduos já estão fora da situação normal, o que leva o classificador a indicar outra situação que não a de funcionamento normal. No momento que a falha 1 ocorre, os resíduos apresentam um comportamento semelhante ao da situação mostrada na Figura 5.13, porém, ambos continuam positivos, não gerando os valores apropriados para o classificador indicar a falha em questão.

A mesma situação simulada com a falha 1, foi também aplicada à falha 4 da Tabela 5.2. A Figura 5.15 mostra o sistema FDI monitorando os resíduos em uma situação onde a identificação do processo está satisfatória. Neste caso, os resíduos gerados são coerentes com o apresentado na Tabela 5.2 e o classificador não tem dificuldade em apontar a falha correta.

A Figura 5.16 mostra a falha 4 ocorrendo em um ponto de operação diferente daquele em que as duas redes neurais foram treinadas. Nesse caso, assim como ocorrido no exemplo da Figura 5.14, pode-se perceber erros na identificação.

A classificação, como esperado, é inconsistente devido aos valores dos resíduos não corresponderem aos resíduos esperados pelo classificador para a falha 4.

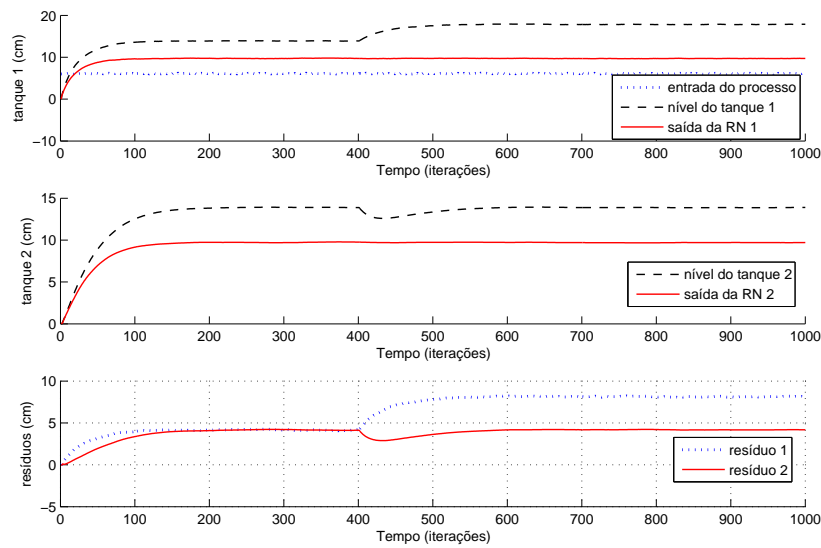


Figura 5.14: Detecção da falha 1 da Tabela 5.2 pela arquitetura clássica em um ponto de operação diferente do treinamento da identificação.

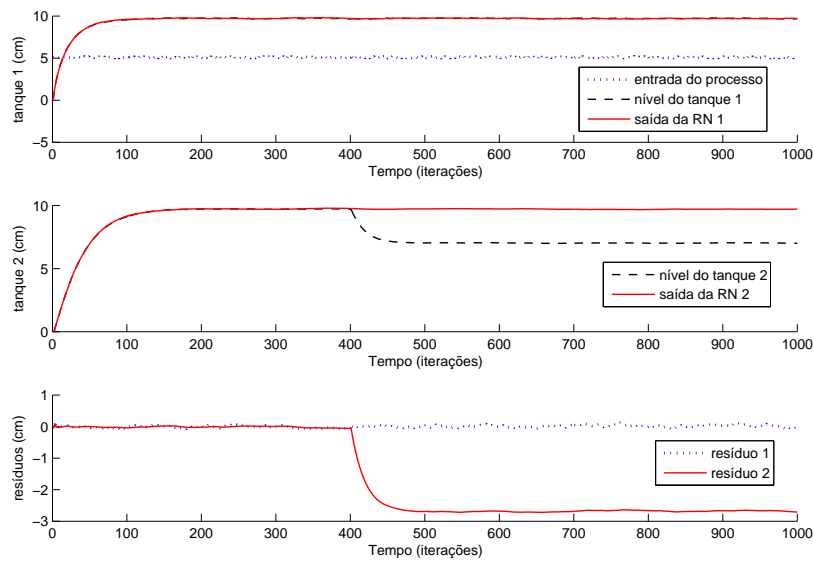


Figura 5.15: Detecção da falha 4 da Tabela 5.2 pela arquitetura clássica.

## 5.2.2 Resultados com a Arquitetura Proposta

A arquitetura proposta está descrita no Capítulo 4. Ela consiste de um algoritmo de identificação de sistemas responsável por estimar os parâmetros do modelo fenomenológico do processo em questão. Neste caso, para efeito de comparação à arquitetura clássica,

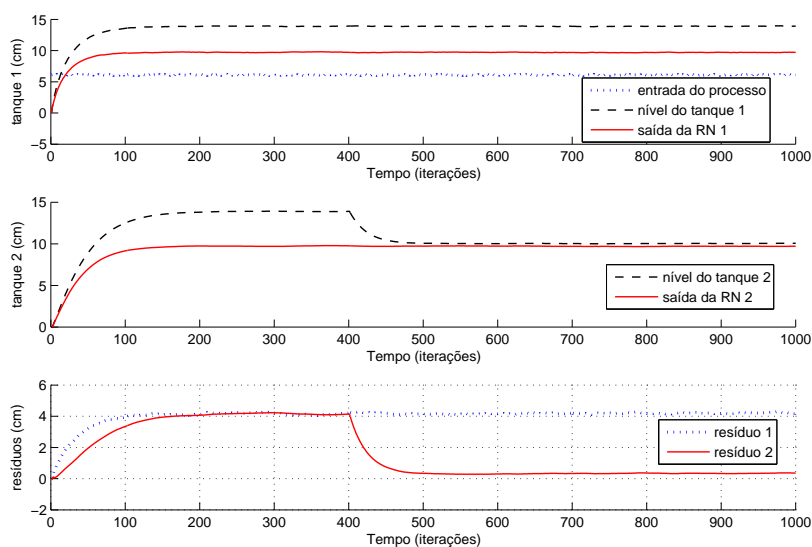


Figura 5.16: Detecção da falha 4 da Tabela 5.2 pela arquitetura clássica em um ponto de operação diferente do treinamento da identificação.

vai se aplicar o sistema FDI proposto, com identificação paramétrica via UKF, ao mesmo processo ao qual foi aplicado à arquitetura clássica. Além disso, para a comparação ser direta, submeteu-se o processo às mesmas falhas, com os mesmos pontos de operação aos aplicados à primeira arquitetura.

A Figura 5.17 mostra a simulação da falha 1 da Tabela 5.2, como o sinal de entrada do processo similar ao aplicado no caso da Figura 5.13.

Neste caso, além da falha ser detectada, indicada pela mudança na estimativa do parâmetro  $a_1$ , pode-se observar que o novo valor estimado pelo algoritmo é exatamente igual ao valor do parâmetro do processo. Isso significa que, a arquitetura proposta é capaz de mensurar as variações sofridas pelo processo em consequência das falhas. E, além disso, fornecer a equação do modelo fenomenológico do modelo em modo de falha, caso a falha seja refletida nos parâmetros monitorados.

A Figura 5.18 pode ser diretamente comparada à Figura 5.14, onde a arquitetura clássica não consegue classificar corretamente a falha ocorrida.

Nesse caso, como a identificação da arquitetura proposta é paramétrica, os parâmetros do modelo não sofrem variação em função do ponto de operação, não alterando a estimativa da saída do processo. Além disso, o fato da identificação ser *on-line* faz com que a arquitetura proposta encontre o novo valor do parâmetro, após o momento de ocorrência da falha 1, independente do ponto de operação do processo.

As Figuras 5.19 e 5.20 podem ser comparadas diretamente às Figuras 5.15 e 5.16, respectivamente. A primeira mostra a ocorrência da falha 4 com o processo funcionando no mesmo ponto de operação em que mostra a Figura 5.15. Neste caso, ambas as arquiteturas detectaram a falha, com a vantagem para a arquitetura proposta dizer qual a nova medida do orifício do tanque 2.

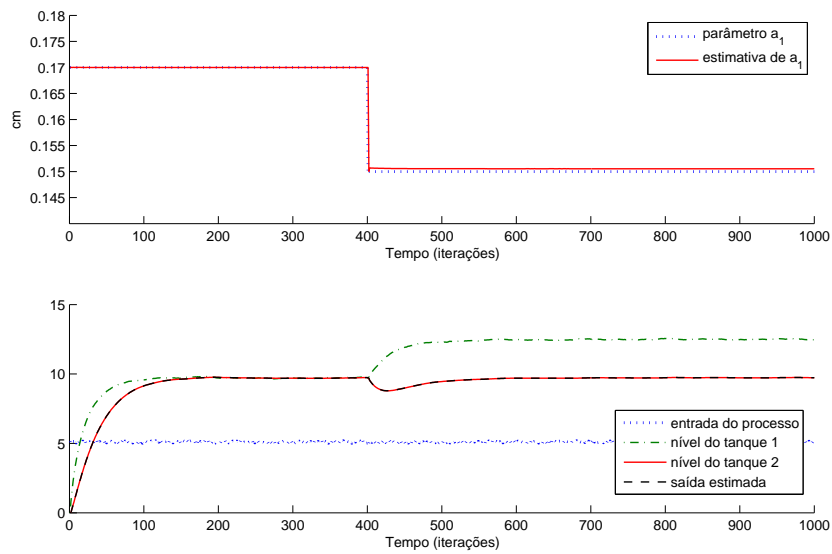


Figura 5.17: *Detecção da falha 1 da Tabela 5.2 pela arquitetura proposta.*

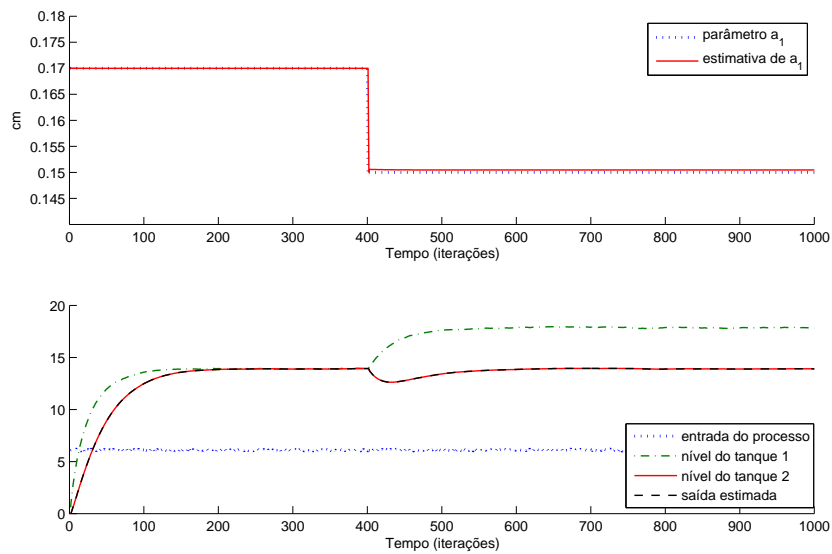


Figura 5.18: *Detecção da falha 1 da Tabela 5.2 pela arquitetura proposta em um ponto de operação diferente ao caso da Figura 5.17.*

No caso da Figura 5.20, a arquitetura proposta detecta adequadamente a falha e o novo valor do parâmetro, enquanto que a arquitetura clássica, na mesma situação, não consegue classificar o ocorrido no processo.

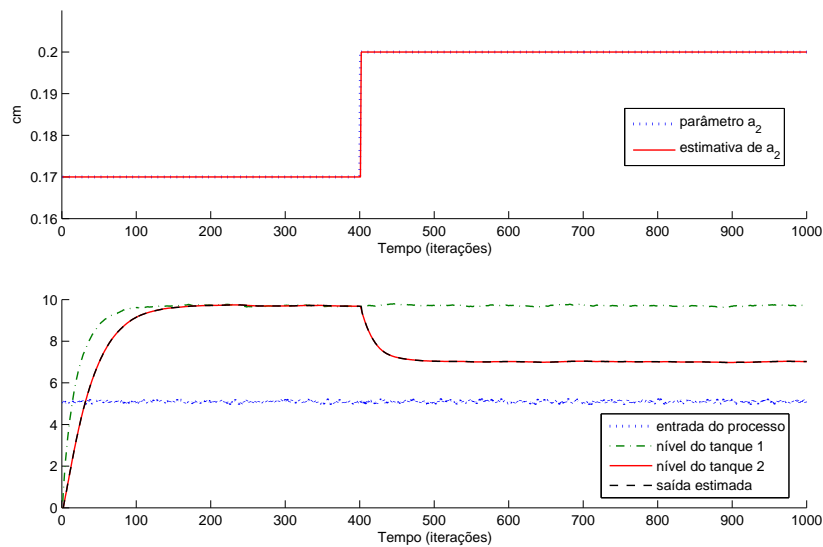


Figura 5.19: Detecção da falha 4 da Tabela 5.2 pela arquitetura proposta.

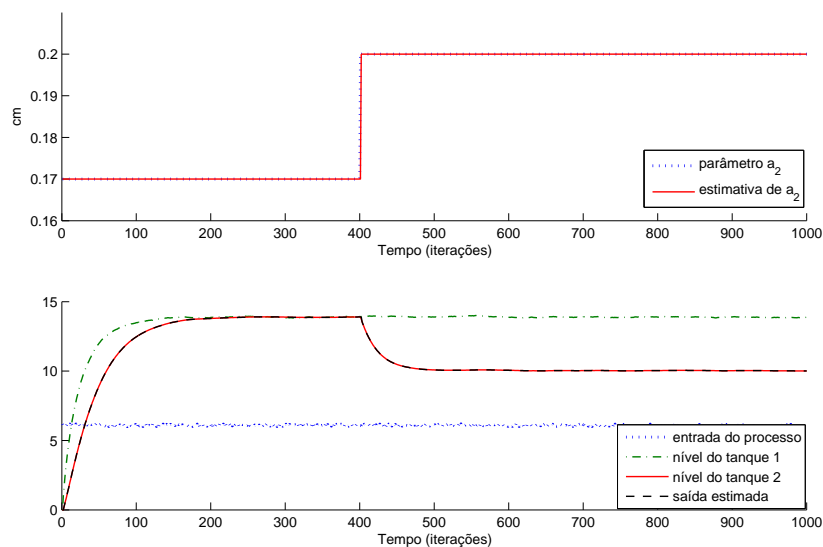


Figura 5.20: Detecção da falha 4 da Tabela 5.2 pela arquitetura proposta em um ponto de operação diferente ao caso da Figura 5.19.

### 5.2.3 Análise dos Resultados

Observando-se os resultados referentes à arquitetura clássica, verifica-se que o desempenho da etapa de identificação está diretamente associado ao desempenho do sistema FDI como um todo. A identificação caixa-preta *off-line*, como a aplicada nesta arquitetura apresenta algumas deficiências como:



- a necessidade de uma sinal de entrada bastante representativo, que faça o processo percorrer a maior quantidade possível de seus estados;
- os parâmetros ajustáveis (pesos) das redes neurais responsáveis pela identificação do processo real só são ajustados na etapa de treinamento. Isso significa que, para qualquer situação que venha a ocorrer no processo, se esta não tiver sido prevista na fase de treinamento (identificação), as redes neurais vão gerar sinais de saída diferentes dos sinais do processo real, mesmo em situações livre de falhas. Como consequência disso, o classificador fará diagnósticos errôneos, como mostrado nas Figuras 5.14 e 5.16.

Já a arquitetura proposta, faz uso de identificação paramétrica, onde os parâmetros têm uma relação direta com alguma grandeza física do processo monitorado (caixa-cinza, ver Capítulo 3). Isso traz contribuições importantes como:

- os parâmetros de um modelo fenomenológico são constantes em todos os pontos de operação do processo. Portanto, não há motivo para o sistema FDI não funcionar em pontos de operação distintos;
- a identificação *on-line*, dispensa uma etapa prévia de identificação e a necessidade de se excitar o processo para que o mesmo percorra a maior parte dos estados possível;
- qualquer falha ocorrida, refletirá nos parâmetros do modelo fenomenológico. Essa alteração será percebida pela diferença entre os sinais de saída do processo e a saída estimada pelo algoritmo que, naturalmente, forçará o mesmo a iniciar uma nova estimativa dos parâmetros. Os novos valores servem para quantificar o desvio do comportamento original do processo.

Além disso, existe um custo associado à implementação da arquitetura clássica, principalmente em situações reais, como:

- a coleta de dados para o treinamento das redes de identificação, o que, em casos reais, pode ser proibitivo, pois gerar sinais representativos em processos já em funcionamento nem sempre é possível;
- o uso de estruturas computacionais para simular o processo ou parte dele. Além disso, a complexidade dessas estruturas crescem de acordo com a complexidade do processo a ser monitorado;
- a necessidade de simular falhas previamente para se extrair o comportamento dos resíduos;
- a necessidade de treinamento de um classificador, com modelos simulados, pois não se pode gerar falhas em processos reais simplesmente para observar seu comportamento. O que pode não representar a situação real.

Já a arquitetura proposta, tem uma série de vantagens operacionais, como:

- a não necessidade de treinamento prévio dos modelos do processo;
- a inexistência de treinamento para a etapa de classificação e diagnóstico de falhas;

- a quantificação do desvio sofrido pelo processo através da estimativa dos parâmetros do modelo fenomenológico em modo de falha;
- a praticidade de se encontrar o local atingido pela falha, facilitando o isolamento do mesmo, através da observação direta dos parâmetros afetados pelas falhas.

Os pontos citados acima reforçam as contribuições trazidas à área de detecção e isolamento de falhas pela arquitetura proposta. As próximas seções mostram os resultados desta arquitetura quando a mesma foi aplicada a um processo que simula um parâmetro variante em função do ponto de operação e, finalmente, quando ela foi aplicada a processos reais.

### 5.3 Simulação de Falhas com *Threshold* Variável

A técnica de verificação de limites, ou *threshold*, mostrada na Seção 2.2.1, apesar de simples, também pode ter aplicações nas soluções mais complexas. Neste caso mostra-se o uso do *threshold* quando o mesmo é variável. Esta técnica é útil quando os valores dos parâmetros, mesmo variando, não indicam falhas. Em situações reais, parâmetros podem variar em função do ponto de operação, como o caso mostrado a seguir. Os limites inferior e superior do parâmetro são variáveis de acordo com o valor nominal do parâmetro, ou seja, ele pode variar sem indicar falha, desde que respeite os respectivos limites.

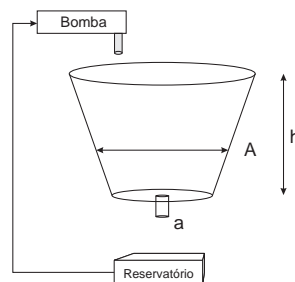


Figura 5.21: Tanque com parâmetro variável em função do ponto de operação.

O processo em questão é um tanque de formato cônico, onde o valor do diâmetro da secção transversal depende do nível de líquido no tanque. A Figura 5.21 ilustra o processo. A Equação 5.4 mostra a relação dos parâmetros com a saída do mesmo:

$$\dot{h} = -C\sqrt{2gh} + \frac{K_m}{A}V_p \quad (5.4)$$

onde

$$C = \frac{a}{A(h)}$$

e

$$A(h) = 2h + 3$$

Como pode-se notar,  $C$  é um parâmetro que depende do ponto de operação e, para exemplificar, é o parâmetro que será monitorado. Os outros parâmetros são mostrados na Tabela 5.3.

Tabela 5.3: Parâmetros da Planta

Nome	Significado	Valor
$K_m$	Constante da bomba	$4,6(cm^3/s)/V$
$V_p$	Tensão aplicada à bomba	$-22 < V_p < 22$
$a$	Diâmetro do orifício	$0,17cm$
$A$	Diâmetro do tanque	$3cm - 23cm$
$h$	Altura do tanque	$10cm$
$g$	Aceleração da gravidade	$980cm/s^2$

A Figura 5.22 mostra o comportamento do parâmetro  $C$  em caso de funcionamento normal, quando estimado pelo UKF.

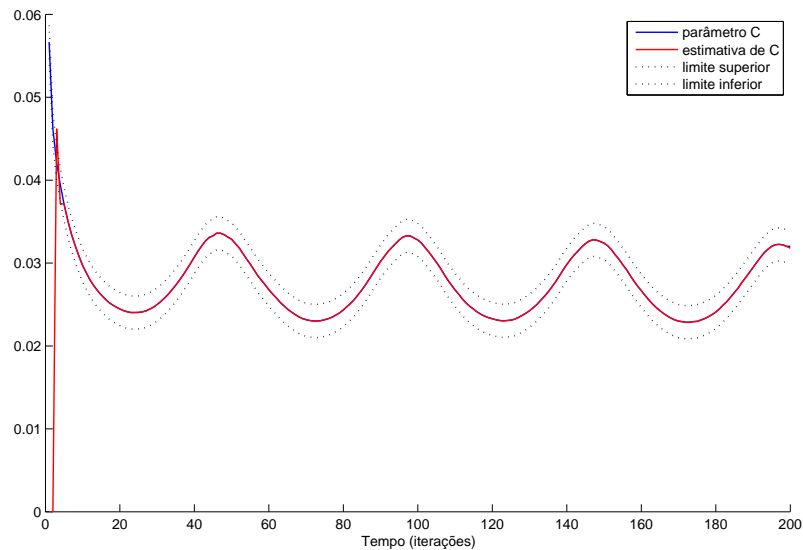


Figura 5.22: Parâmetro  $C$  em funcionamento normal.

Pode-se perceber, através do gráfico, que o parâmetro  $C$  varia, mesmo em situações de funcionamento normal do processo. As linhas pontilhadas representam os limites inferior e superior de situações normais de funcionamento. Neste caso, a linha contínua vermelha, que representa a estimativa do parâmetro pelo algoritmo, está de acordo com a linha contínua azul, que representa o parâmetro  $C$  do processo. Neste caso, ambos estão dentro do limite de funcionamento normal do processo.

Já a Figura 5.23 mostra uma falha ocorrendo no processo sendo refletida no parâmetro  $C$ .

Pode-se perceber que, neste caso, a linha contínua azul, que representa o parâmetro  $C$  do processo simulado, ultrapassa o limite superior, indicando a ocorrência de uma falha. Em um caso real, não se tem acesso a essa medida. Porém, pode-se observar que a estimativa de  $C$ , dada pelo algoritmo, consegue acompanhar o valor do parâmetro real,

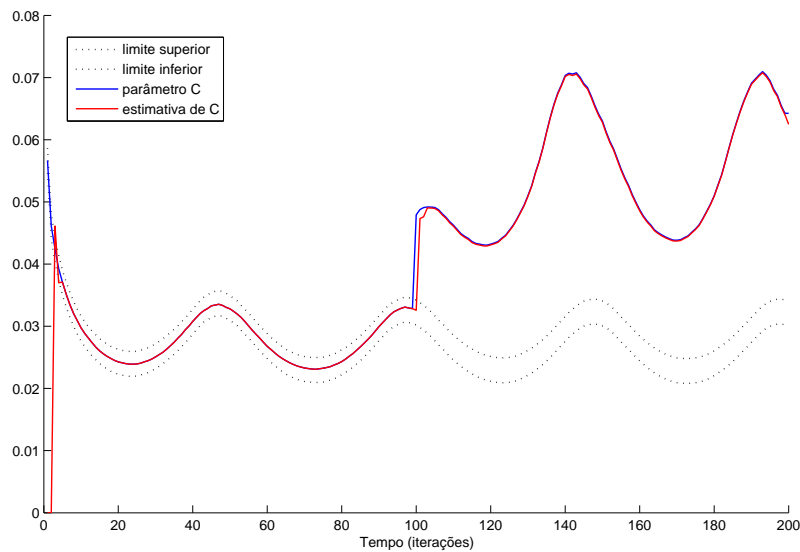


Figura 5.23: Parâmetro  $C$  refletindo uma falha ocorrida no processo.

indicando a falha e quantificando-a.

### Análise dos Resultados

Pode-se observar que nem sempre variações dos parâmetros indicam ocorrência de falhas. Nesse caso, se o ponto de operação do processo mudar, é esperado, como comportamento normal, que o parâmetro mude. A solução para esse tipo de situação é usar *thresholds* variáveis. Estes últimos funcionam como valores limitantes, inferior e superior, usados no diagnóstico de falhas.

## 5.4 Processos Dinâmicos Reais

Os resultados encontrados nesta seção foram obtidos de uma planta real, o que os torna definitivos no que diz respeito à validação da arquitetura proposta e de sua implementação.

O primeiro exemplo, trata de uma planta linear. Isto simplifica a explicação do funcionamento do sistema FDI proposto, assim como a análise dos resultados. O segundo exemplo trata de uma planta não-linear. Esta representa a maior parte dos sistemas reais e, por isso, mostra que a solução tem aplicabilidade prática.

Para casos lineares, algoritmos RLS e KF podem naturalmente ser usados na estimação dos parâmetros e, conseqüentemente, na detecção das faltas e falhas. Já em processos que apresentam uma dinâmica não-linear, a comparação entre o EKF e o UKF mostrada na Seção 5.1 aponta o UKF como algoritmo mais apropriado para a referida tarefa.

### 5.4.1 Um Motor DC

Esta seção trata da implementação da arquitetura proposta no Capítulo 4 aplicada ao controle de velocidade de um motor DC. A descrição do processo é encontrada a seguir, onde mostra-se a equação que rege a sua dinâmica e a definição de todas as variáveis envolvidas. Depois mostra-se a identificação do mesmo, através do ajuste dos parâmetros monitorados pelo algoritmo. E, finalmente, o desempenho da solução quando o processo foi submetido a falhas reais.

#### Descrição do Processo

O motor aqui utilizado é um motor DC didático fabricado pela Quanser. Toda a descrição é encontrada em seu manual de instruções [Astrom & Apkarian n.d.] que foi utilizado aqui para uma breve descrição. A Figura 5.24 ilustra o circuito elétrico do motor, indicando o seu funcionamento e identificando as grandezas físicas envolvidas no processo.

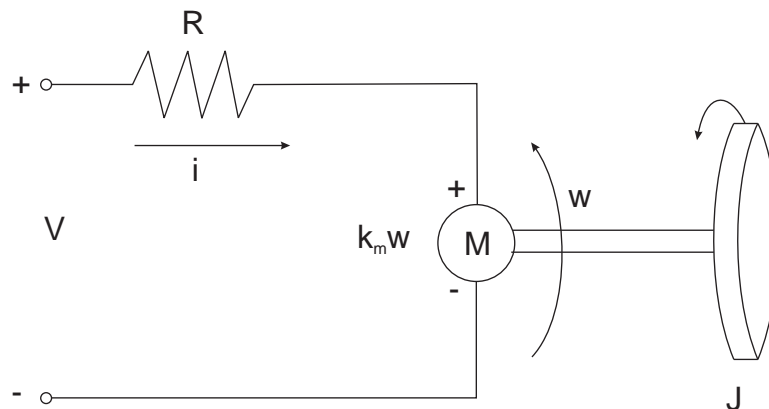


Figura 5.24: Circuito elétrico do motor DC.

onde cada grandeza é descrita pela Tabela 5.4

Tabela 5.4: Parâmetros da Planta

Nome	Significado	Unidade
$K_m$	Constante do motor	$N.m/A$
$J$	Momento de Inércia do rotor	$Kg.m^2$
$R$	Resistência	$\Omega$
$i$	Corrente	$A$
$V$	Tensão de entrada	$V$
$\omega$	Velocidade Angular	$rad/s$
$h$	Período de amostragem	$s$

a função de transferência que relaciona a tensão de entrada  $V$  com a velocidade angular  $\omega$  é a seguinte:

$$G_{\omega,V}(s) = \frac{K_m}{R(Js + \frac{k_m^2}{R})} \quad (5.5)$$

Nesse primeiro exemplo real, a fim de simplificação e para validação da solução para processos lineares, o modelo utilizado usará a representação de função de transferência, e os parâmetros a serem estimados pelo algoritmo serão os pólos e os zeros da mesma (ver seção 3.2.1). A função de transferência simplificada do motor é:

$$G(s) = \frac{a}{s + b} \quad (5.6)$$

onde  $a = k_m/RJ$  e  $b = k_m^2/RJ$ . Como pode-se perceber, esta é a versão contínua (polinômios em  $s$ ) da função de transferência do motor.

Discretizando essa função de transferência tem-se:

$$G(z) = \frac{a'}{z + b'} \quad (5.7)$$

onde

$$a' = \frac{1}{K_m} (1 - e^{-\frac{K_m^2}{J.R}h})$$

e

$$b' = -e^{-\frac{K_m^2}{J.R}h}$$

Porém, neste exemplo, para validar a solução com processos lineares, como já foi dito, as grandezas físicas não serão os parâmetros estimados, como sugere a proposta. Este caso está mostrado na próxima seção.

Para este exemplo, os parâmetros a serem estimados são  $a'$  e  $b'$ . O período de amostragem utilizado foi  $h = 0,01s$ .

### Identificação do Processo

Antes de qualquer teste com falhas propriamente ditas, deve-se verificar a convergência do algoritmo em situações de funcionamento normal. Visto que esta é uma condição necessária para o funcionamento da solução. A Figura 5.25 mostra os parâmetros da Equação 5.7 sendo estimados no gráfico superior, e as saídas do motor e estimada no gráfico inferior.

### Funcionamento sob Condições Adversas

Esta seção mostra os resultados do desempenho da solução quando o processo monitorado sofre desvios de comportamento provenientes de falhas de várias naturezas. A Figura 5.26 ilustra uma falha abrupta gerada propositalmente na resistência  $R$  do motor. A falha é gerada depois que a estimativa dos parâmetros se encontra estável, próximo a iteração mil.

Pode-se perceber que os dois parâmetros sofrem alterações no momento da falha, e que até o algoritmo estimar os novos valores de  $a'$  e  $b'$  a saída real do motor diverge da

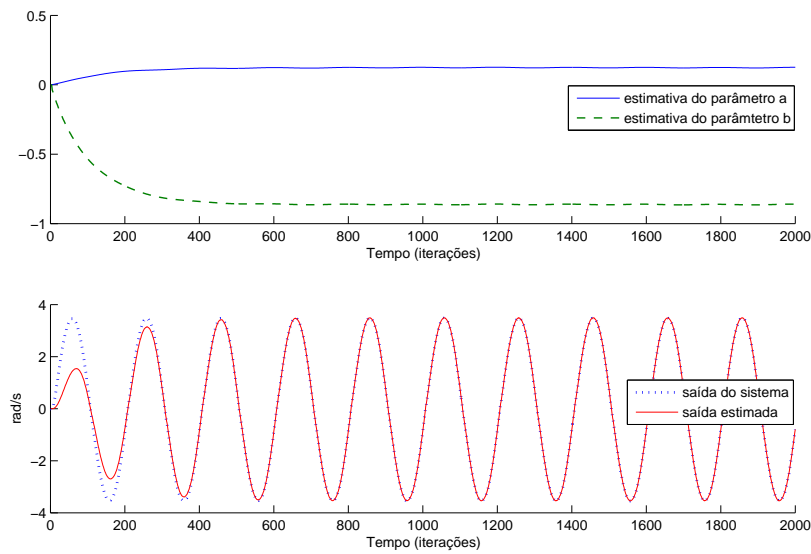


Figura 5.25: Identificação paramétrica do motor discretizado usando O filtro de Kalman.

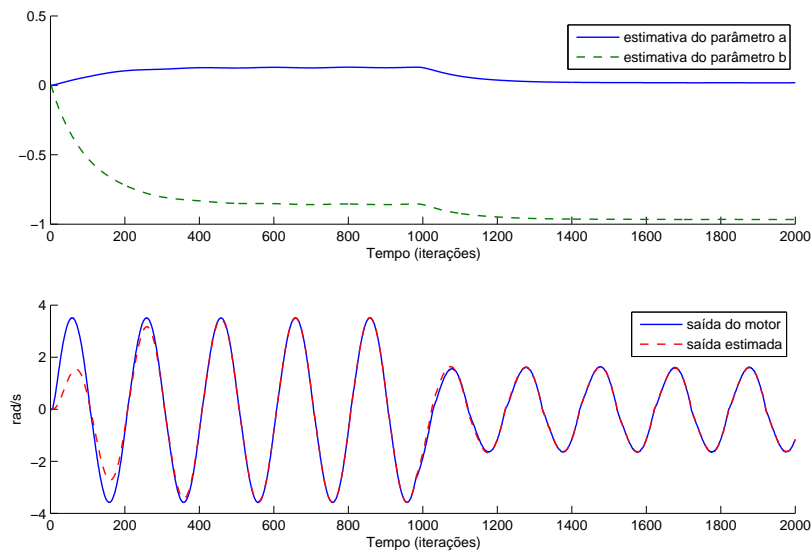


Figura 5.26: Falha abrupta na resistência do motor.

saída estimada, mesmo que discretamente. A partir da iteração mil e duzentos observa-se os valores dos parâmetros estabilizados e a saída estimada próxima a saída real do motor, o que indica que os novos valores dos parâmetros foram corretamente estimados.

A Figura 5.27 ilustra o comportamento do processo sob uma falha incipiente, ou seja, uma mudança sutil em sua relação entrada/saída, neste caso, causada propositalmente pela suave variação da resistência  $R$  do circuito do motor.

Em soluções que usam a identificação *off-line* do processo [Fernandes et al. 2007b],

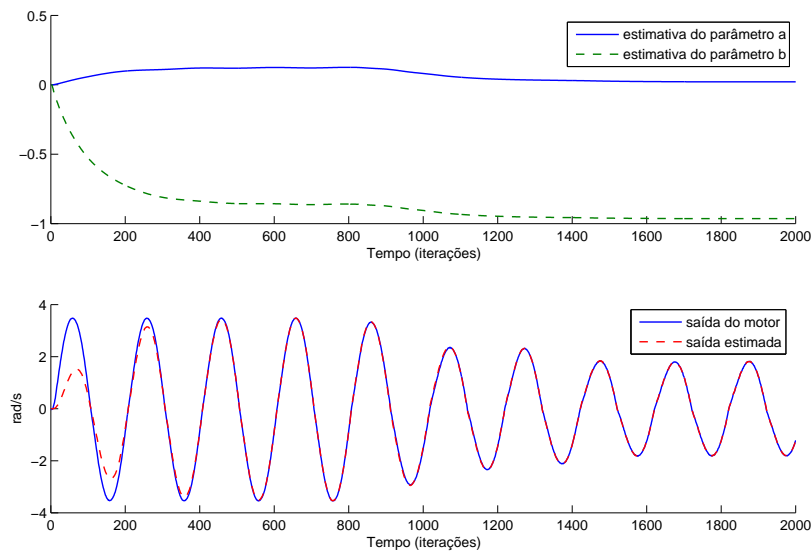


Figura 5.27: *Falha incipiente na resistência do motor.*

há uma tendência à divergência entre a saída do modelo e a saída do processo real com o passar do tempo, causada pela realimentação de valores estimados pelo próprio modelo. Esse efeito pode causar o que se chama de falso positivo, indicando uma falha incipiente que na verdade não existe. Na abordagem proposta nesta tese, a identificação do processo ocorre de maneira contínua, com os parâmetros sendo estimados a cada iteração. Isto faz com que não haja erros entre a saída estimada e a saída real provocados simplesmente pelo tempo. A detecção de falhas incipientes, nesse caso, é percebida através da leve variação que esta causa nos parâmetros, pois o algoritmo, as vezes, é rápido o suficiente para não acusar diferença entre os sinais de saída e saída estimada.

Ao observar a Figura 5.27 pode-se notar entre as iterações oitocentas e mil, a leve mudança dos dois parâmetros e a suave mudança na amplitude da saída do processo. Porém, neste caso é imperceptível a diferença entre a saída estimada e a saída real.

Uma outra categoria de falhas, são as intermitentes (ver Capítulo 2). Estas, por sua vez são causadas por algum distúrbio que interfere ciclicamente no processo. As Figuras 5.28 e 5.29 mostram o comportamento do motor quando este está sob os efeitos de interferências cíclicas.

As duas figuras mostram interferências diferentes. O que causa uma resposta diferente do processo em cada caso. Porém, em ambas, pode-se notar que os parâmetros têm um comportamento próximo de um sinal periódico. Vale salientar que, para estes dois casos, a saída estimada está sempre muito próxima da saída real, o que confere a confiabilidade dos valores estimados em modo de falha.



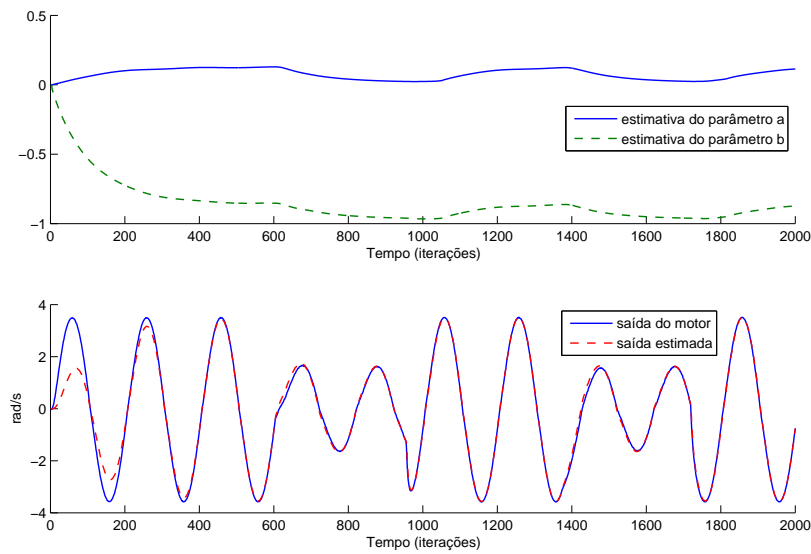


Figura 5.28: Falha intermitente na resistência do motor.

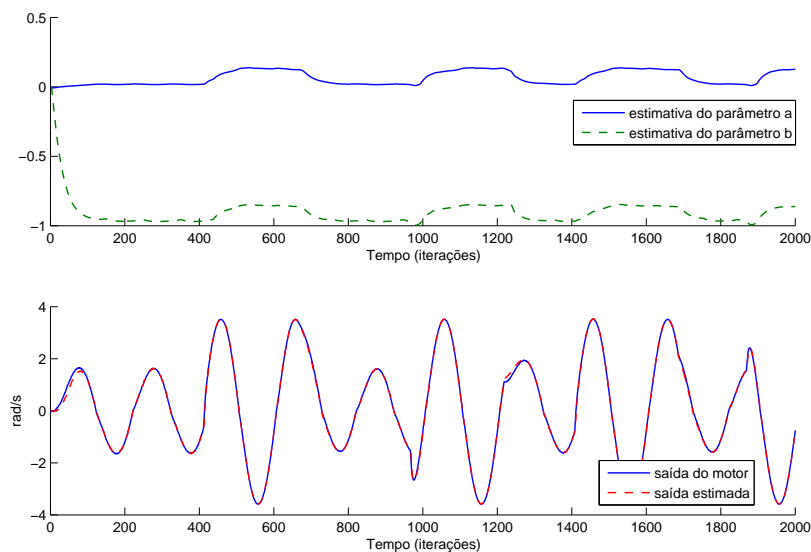


Figura 5.29: Outra falha intermitente na resistência do motor.

### Análise dos Resultados

Pode-se concluir, analisando-se os gráficos mostrados acima, que a solução é eficaz quando usada em ambientes reais, levando em considerações todas as adversidades não encontradas em ambientes simulados. Além da identificação, que é uma etapa necessária para as demais, a ferramenta, estimou os parâmetros, como esperado, em todos os casos de falha ao qual foi submetido, mostrando-se capaz, de ser usada, quando o processo for

linear.

O caso onde existe uma dinâmica não-linear entre os parâmetros e a saída da planta é mostrado na seção seguinte, onde se relaciona as grandezas físicas do processo com a saída do mesmo. Situação onde a solução traz todas as vantagens citadas no Capítulo anterior.

## 5.4.2 O Motor com Parâmetros Físicos

### Descrição do Processo

A equação de diferenças discreta do motor apresentada na seção anterior é a seguinte:

$$y(k+1) = a_1 \cdot u(k) + a_2 \cdot y(k) \quad (5.8)$$

onde os parâmetros  $a_1$  e  $a_2$  são:

$$a_1 = \frac{1}{K_m} (1 - e^{-\frac{K_m^2}{J.R}h}) \quad (5.9)$$

e

$$a_2 = -e^{-\frac{K_m^2}{J.R}h} \quad (5.10)$$

A Tabela 5.4 descreve todas as grandezas físicas relacionadas com as variáveis  $a_1$  e  $a_2$ .

substituindo as equações 5.9 e 5.10 em (5.8) tem-se:

$$y(k+1) = \frac{1}{K_m} (1 - e^{-\frac{K_m^2}{J.R}h}) \cdot u(k) - e^{-\frac{K_m^2}{J.R}h} \cdot y(k) \quad (5.11)$$

Para este caso, pode-se notar que há não-linearidades expressas por quocientes e exponenciais que relacionam as grandezas físicas com a saída do motor. A constante do motor  $K_m$  e a resistência  $R$  foram escolhidos para serem monitorados pelo sistema FDI. A resistência  $R$  pode ser facilmente usada para a simulação de falhas através da ligação de uma resistência extra em série. O período de amostragem utilizado foi  $h = 0,01s$ .

### Identificação do Processo

Antes de monitorar faltas e falhas em um processo real, deve-se verificar o desempenho do algoritmo na identificação do processo. Dado a representação do mesmo e os parâmetros que serão monitorados, deve-se usar o algoritmo para que este estime os parâmetros reais do processo, baseando-se para isso, nos sinais de entrada e saída do mesmo.

A Figura 5.30 mostra o algoritmo do filtro de Kalman Unscented aplicado a estimação dos valores dos parâmetros do processo quando o mesmo encontra-se fora de falhas. Pode-se perceber que, mesmo dando valores iniciais errados, o algoritmo converge, levando-os aos valores fixos do motor real, também apresentados no gráfico.

A partir da Figura 5.30 pode-se ver claramente que o parâmetro  $R$  da Equação 5.11 atinge um valor aceitável por volta da iteração 200. Já o parâmetro  $K_m$ , tem o seu valor inicial muito próximo do valor de destino, dificultando sua visualização nesta Figura. Por

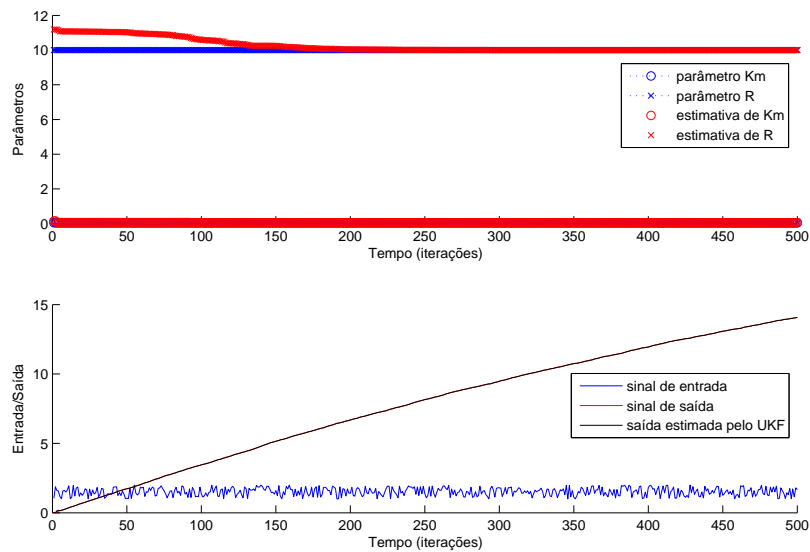


Figura 5.30: Identificação paramétrica do motor discretizado usando UKF.

esta razão, a Figura 5.31 mostra uma aproximação do começo da execução do algoritmo, onde pode-se perceber a tendência à convergência do parâmetro  $K_m$  já na terceira iteração.

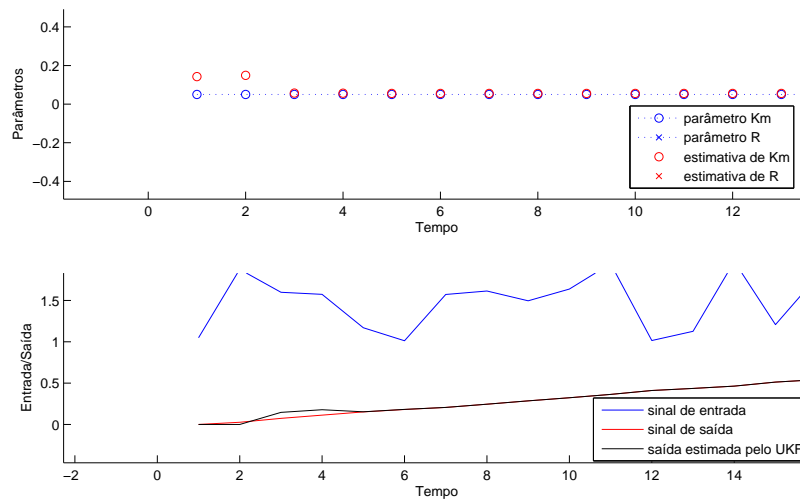


Figura 5.31: Detalhe do começo da execução do algoritmo no processo de identificação paramétrica do processo descrito pela equação 5.8.

A diferença na convergência dos parâmetros se dá pela representatividade dos mesmos na saída do processo. Ou seja, O parâmetro  $K_m$  quando alterado leva a saídas do processo a alterações mais bruscas, quando comparada ao efeito na saída produzido pela alteração do parâmetro  $R$ . A superfície de erro para este processo em funcionamento normal é

ilustrada na Figura 5.32, onde pode-se perceber esta relação dos parâmetros  $R$  e  $K_m$  com a saída do sistema.

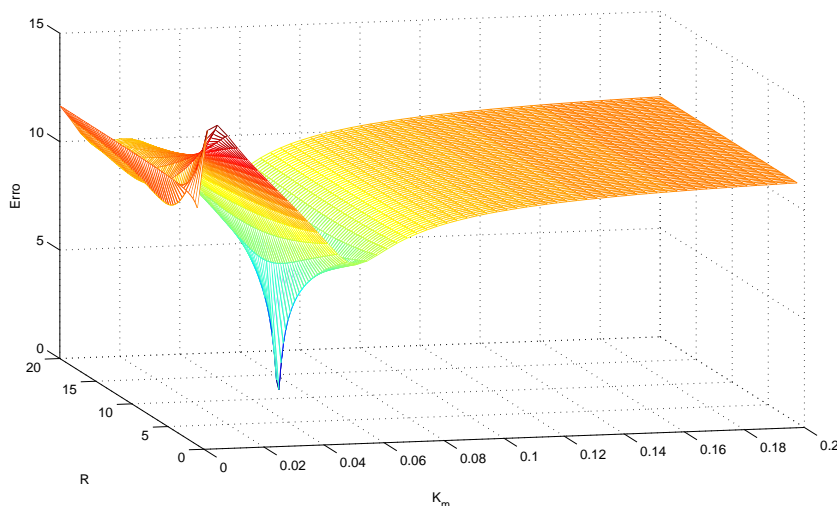


Figura 5.32: Superfície de erro relacionando os parâmetros físicos reais do motor  $k_m$  e  $R$  com a saída  $\omega$  do mesmo.

Neste caso, esse gráfico foi gerado calculando-se o erro entre a saída do motor em funcionamento normal (fora de faltas e falhas) e seu funcionamento simulado utilizando-se os parâmetros indicados nos eixos do gráfico. Pode-se perceber que existem áreas do gráfico onde a variação do parâmetro  $R$  mantém o erro constante, não direcionando o algoritmo para o mínimo e justificando o maior tempo em sua convergência.

### Desempenho sob Condições Adversas

Esta seção mostra o desempenho da solução baseada em UKF quando o processo físico encontra-se em modo de falha. Para este processo, várias falhas foram simuladas, entre elas, falhas abruptas nas duas grandezas físicas monitoradas, sendo uma de cada vez. Falhas abruptas simultâneas, quando os dois parâmetros sofrem alterações bruscas ao mesmo tempo, falhas incipientes e falhas intermitentes.

Primeiramente, encontram-se as falhas abruptas individuais. A falha provocada no parâmetro  $K_m$  pode ser visualizada na Figura 5.33.

Como pode ser observado, a estimativa do novo parâmetro em modo de falha foi eficientemente calculada pelo algoritmo UKF. A Figura 5.34 mostra uma visão detalhada tanto da convergência do parâmetro para o novo valor (gráfico superior), quanto a saída do sistema (gráfico inferior) comparada à saída estimada. Pode-se observar que em pouco mais de dez iterações o novo valor do parâmetro encontra-se estabilizado e a diferença entre os sinais de saída e saída estimada volta a diminuir. O que representa confiabilidade na estimativa do parâmetro em modo de falha.

Isso significa dizer que a constante do motor  $K_m$  sofreu alteração, o que é considerado uma falha, pois o sistema não mais responde como antes. A solução aqui proposta é

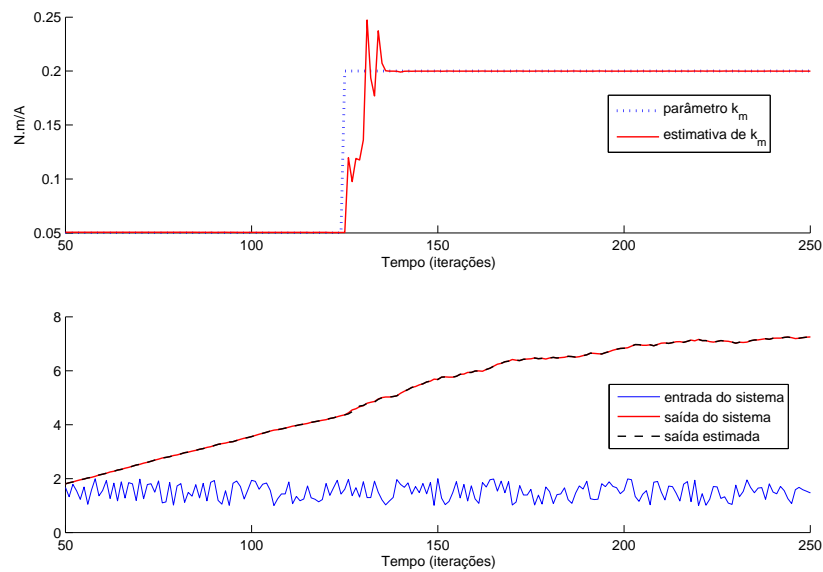


Figura 5.33: Falha abrupta gerada no parâmetro  $k_m$  da Tabela 5.4.

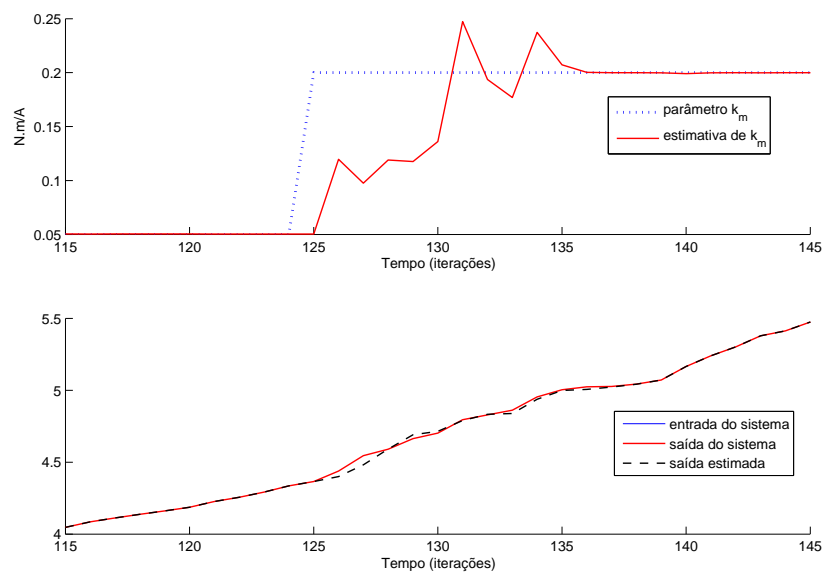


Figura 5.34: Visão detalhada da falha abrupta em  $k_m$ .

capaz de estimar o novo valor da variável que sofreu o desvio quando o processo físico encontra-se em modo de falha, o que permite perceber diretamente a amplitude da falha, apenas comparando o novo valor de  $k_m$ , estimado pelo algoritmo, ao seu valor nominal. Além é claro, de isolar fisicamente o local do processo atingido pela falha, através da alteração apenas dos parâmetros afetados pela mesma.

A falha abrupta gerada no parâmetro  $R$  (resistência de armadura do motor) pode ser visualizada na Figura 5.35.

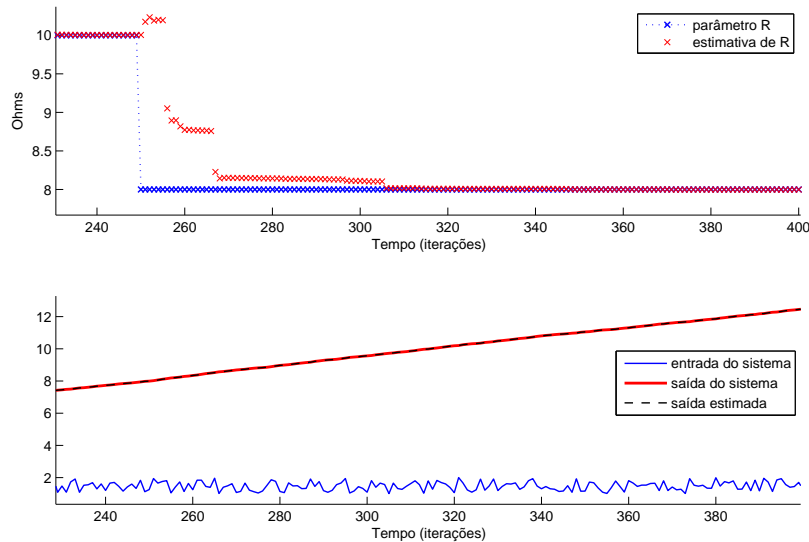


Figura 5.35: Falha abrupta gerada no parâmetro  $R$  da Tabela 5.4.

Pode-se observar que o algoritmo converge para o novo valor do parâmetro  $R$ , porém pode-se perceber também que este precisa de mais iterações que o parâmetro  $K_m$  para convergir. Esta é uma consequência já esperada, de acordo com o que foi previamente explicado sobre este processo, já que o parâmetro  $K_m$  causa distúrbios maiores que o parâmetro  $R$  na saída do processo, ao sofrer variação. Na Figura 5.36 pode-se ter uma visão detalhada entre os sinais de saída e saída estimada no momento em que a falha ocorre. A ausência de diferenças visíveis entre os dois sinais justifica a demora na convergência para o novo valor.

Para finalizar as falhas abruptas, a Figura 5.37 mostra o comportamento do sistema FDI sob o efeito de falhas abruptas que ocorre nos dois parâmetros monitorados ao mesmo tempo, chamada previamente de falhas abruptas simultâneas.

Percebe-se claramente a alteração no parâmetro  $R$ , porém, por questões de escala do gráfico, a visualização da alteração no parâmetro  $K_m$  foi dificultada. Por este motivo, uma visualização detalhada da variação de  $k_m$  pode ser encontrada na Figura 5.38.

Pode-se perceber que mesmo quando os dois parâmetros sofrem alterações ao mesmo tempo, o sistema FDI é capaz de estimar, de maneira eficiente, os novos valores dos parâmetros em modo de falha. A saída estimada, rapidamente volta a seguir a saída do processo, indicando a confiabilidade das estimativas.

A Figura 5.39 mostra uma falha incipiente acontecendo com a constante do motor  $K_m$ , que pode ser causada por algum aumento leve e contínuo do atrito entre componentes do motor. Isso causa um lento desvio do funcionamento normal do motor, as vezes não percebido, devido à ausência de sintomas visíveis em seu funcionamento, consequência essa das alterações muito suaves em sua saída.

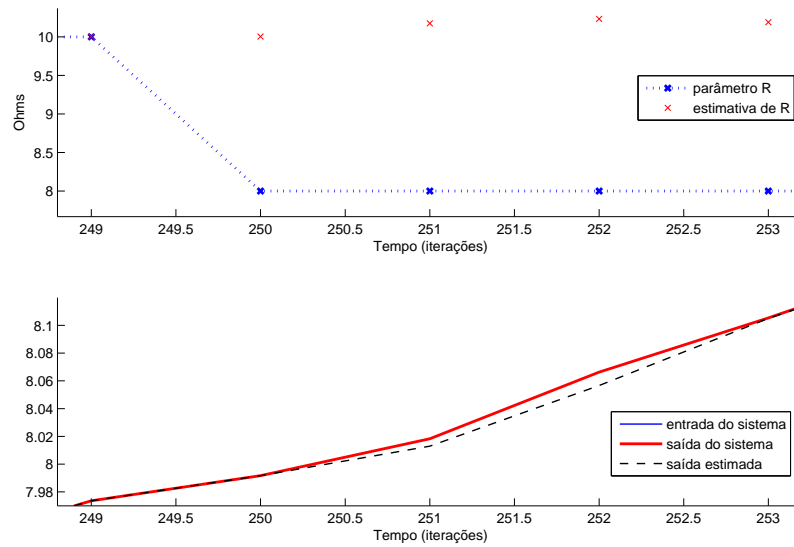


Figura 5.36: *Visão detalhada da falha abrupta em R.*

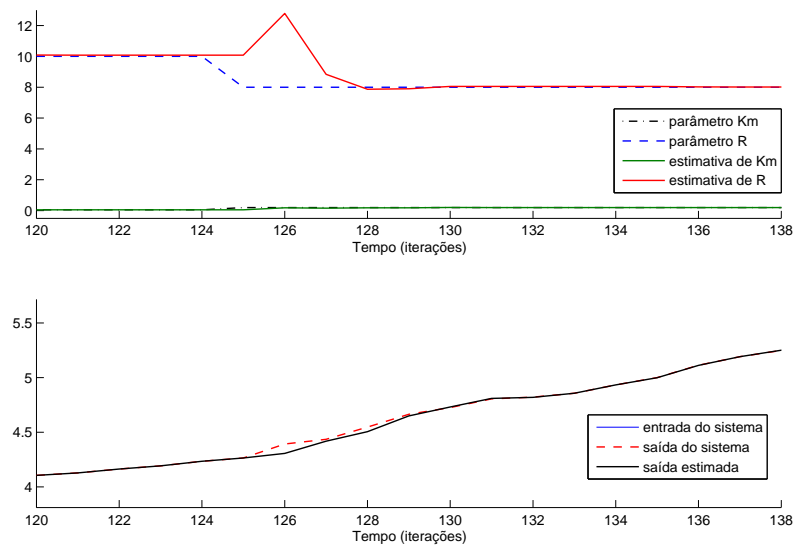


Figura 5.37: *Falhas abruptas simultâneas.*

Observando a Figura 5.39 fica claro que a partir da iteração cem o parâmetro começa a aumentar o seu valor lenta e continuamente. Porém, pode-se perceber também que a estimativa do parâmetro  $K_m$  dada pelo algoritmo consegue seguir o valor real do parâmetro, fazendo o seu papel de mostrar, a cada iteração, a estimativa atual do valor do parâmetro do motor.

A estimativa dos parâmetros, em todos os casos de falha, podem contar com a diferença entre a saída do processo e a saída estimada pelo algoritmo, como uma medida de

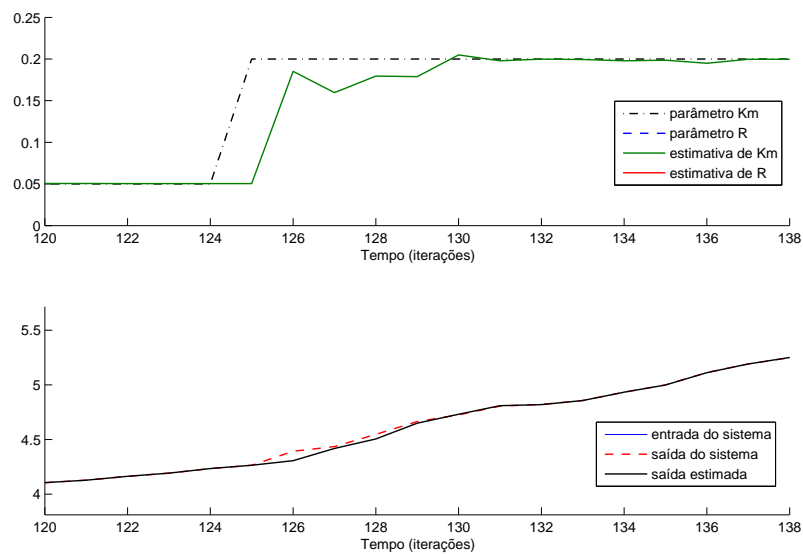


Figura 5.38: Detalhe do parâmetro  $k_m$  em situação de falhas abruptas simultâneas.

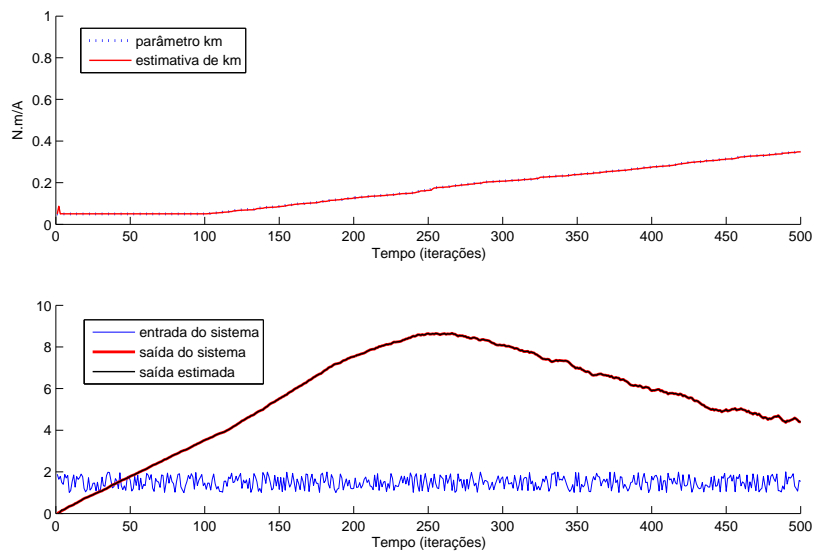


Figura 5.39: Falha incipiente na constante do motor  $k_m$ .

confiabilidade. Neste último caso pode-se observar que, durante a falha incipiente, tanto o parâmetro estimado está sempre com valores muito próximos ao parâmetro real quanto a saída estimada está sempre muito próxima à saída real. O que pode-se concluir que durante toda a falha, a medida estimada do parâmetro é confiável.

As falhas intermitentes, como já definidas no Capítulo 2. são aquelas que interferem no processo de maneira cíclica. A Figura 5.40 mostra uma falha intermitente ocorrendo no motor e refletindo no parâmetro  $K_m$ . Esta falha pode ocorrer por desgaste irregular de



algum componente do motor.

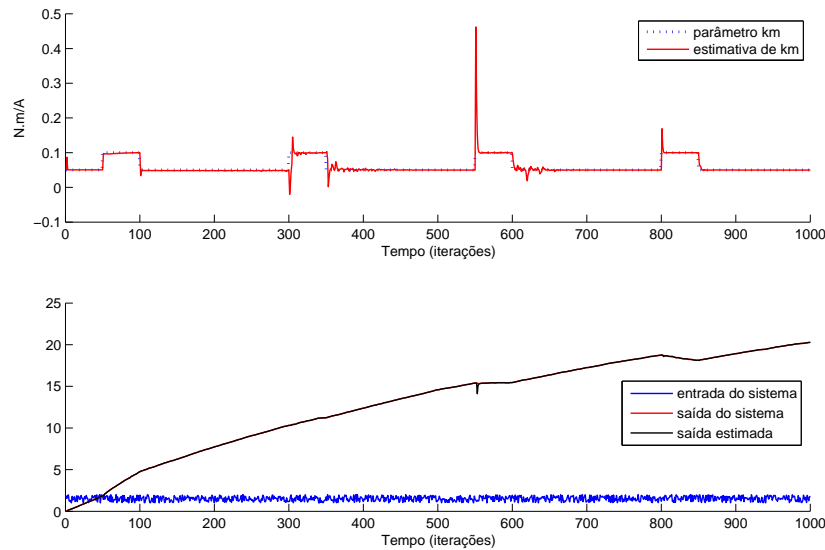


Figura 5.40: Falha intermitente refletida constante do motor  $K_m$ .

Como dito anteriormente, a diferença entre os sinais de saída e sua estimativa, podem indicar o grau de confiabilidade da estimativa dos parâmetros. Neste caso, em alguns pontos o algoritmo estimou erroneamente o valor do parâmetro  $K_m$ . Isto acontece geralmente assim que ocorre uma alteração brusca no processo físico, como o momento de início de uma das falhas intermitentes. Este comportamento também pode ser observado nos casos anteriores. Neste período, o algoritmo está recalculando as estimativas dos parâmetros, e estas levam um tempo até se estabilizarem. Na Figura 5.41 pode-se ver nitidamente a diferença entre os sinais de saída e sua estimativa, indicando que naquele momento a estimativa do parâmetro  $K_m$  não é confiável.

### Análise dos Resultados

Como pode ser observado nos gráficos de resultados referentes à versão que usa os parâmetros físicos do motor, o algoritmo UKF implementado na solução do sistema FDI proposto obteve resultados satisfatórios desde a sua identificação, em funcionamento normal, passando por todas as situações de falhas mostradas. Vale ressaltar que a eficiência da estimativa dos parâmetros está intimamente ligada à sensibilidade de cada parâmetro em relação à saída do processo. Visto que esta, quando comparada à saída estimada, serve como indicativo de satisfação da estimativa dos parâmetros para o algoritmo. Podendo também ser utilizada como medida de confiabilidade da estimativa em cada instante de tempo.

Como mostrado na seção 5.4.1, em um ambiente real, as falhas que ocorrem em um processo físico são refletidas nos parâmetros do modelo que o governa. Estes, por sua

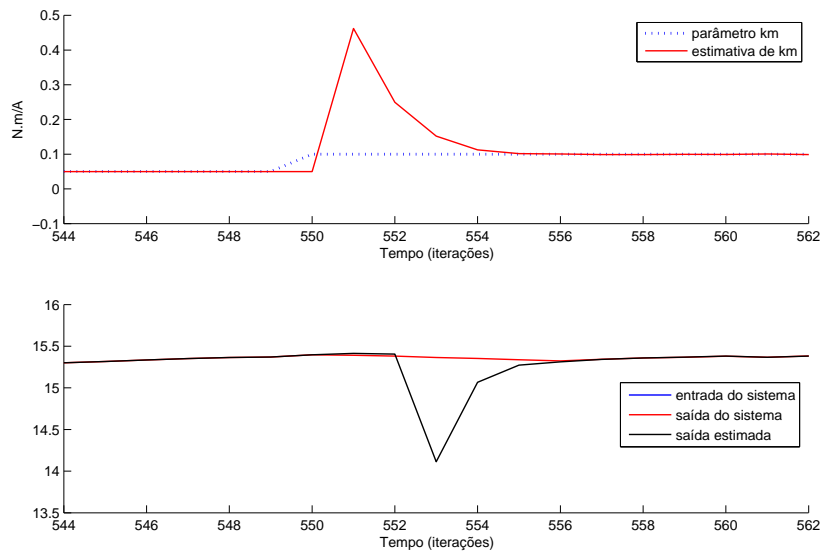


Figura 5.41: *Visualização detalhada do momento em que a estimativa do parâmetro  $k_m$  é não confiável.*

vez, são monitorados através de estimativas em tempo real e a cada iteração transparecem a qualidade de funcionamento do processo monitorado.

Os gráficos que mostram a estimativa dos parâmetros comprovam a eficácia da arquitetura, quando a mesma faz uso dos algoritmos do filtro de Kalman, para o caso de processos lineares, e Filtro de Kalman Unscented para o caso de processos não-lineares em todos os casos mostrados, ou seja, casos de falhas abruptas, incipientes e intermitentes, fazendo com que a mesma alcance os seus objetivos.

---

# Capítulo 6

## Conclusões

---

Como mostrado no Capítulo 2, uma grande variedade de tecnologias é, atualmente, utilizada no projeto e implementação de sistemas FDI. A maioria dessas soluções têm objetivos e preocupações em comum. As mais importantes são listadas a seguir:

- a detecção, em tempo hábil, de quaisquer anormalidades ocorridas no processo monitorado;
- o isolamento da área afetada e o diagnóstico da gravidade do ocorrido; e
- a recuperação do processo para uma situação ideal, ou pelo menos aceitável, de funcionamento.

No geral, esses são os objetivos de um sistema de detecção e isolamento de falhas, porém, novas técnicas aplicadas aos sistemas FDI podem trazer contribuições importantes aos mesmos, abrindo uma série de novas possibilidades ou melhorando o desempenho das ações já conhecidas.

A arquitetura do sistema de detecção e isolamento de falhas proposta neste trabalho mantém e satisfaz todas as preocupações supracitadas (sendo a etapa de recuperação dependente do processo real a ser monitorado e, por isso, não sendo tão discutida nesta tese), além de prover informações adicionais consideradas extremamente importantes para a etapa de diagnóstico da respectiva anormalidade. Dentre as contribuições observadas, pode-se destacar:

- a capacidade de se visualizar sintomas que indicam o início da formação de uma falha, caso esta apresente uma dinâmica lenta, se caracterizando como uma falha incipiente. Neste caso, alguns sistemas FDI falham devido as alterações do comportamento so processo serem pequenas e, na maioria dos casos, mascaradas pelo controlador;
- a possibilidade de se fazer previsão de quando um determinado desvio do comportamento do processo se caracterizará em uma falha, possibilitando a manutenção programada e preventiva de partes do processo, como sensores, atuadores, ou componentes que sofram desgaste;
- isolar, de maneira direta, os locais físicos afetados pelas falhas;
- a capacidade de mensurar, com precisão, o desvio do comportamento do processo causado pela falha;

- a arquitetura proposta, é menos sensível à divergência, com o passar do tempo, do que a abordagem clássica, pois a identificação é feita de maneira *on-line*, ou seja, ajustando-se o modelo a cada passo de tempo discreto;
- uma outra vantagem em relação à arquitetura clássica é a possibilidade de realizar tarefas de detecção e diagnóstico de falhas, precisamente, em todos os pontos de operação, o que, no modelo clássico, apresentou dificuldades devido ao erro de identificação em algumas situações.

As informações adicionais providas pela arquitetura proposta vêm enriquecer a forma como se monitora os processos industriais. A partir delas torna-se possível, além dos objetivos gerais de todo sistema FDI, a quantificação das anormalidades detectadas e a localização física destas no processo de maneira direta, já que cada parâmetro monitorado se relaciona com uma grandeza física envolvida no processo. A quantificação é feita comparando-se o valor do parâmetro estimado com o seu valor nominal (em funcionamento normal), enquanto que a localização física é ainda mais direta, assumindo-se que o operador conhece o processo em questão e seus parâmetros monitorados.

Uma outra possibilidade é o acompanhamento de pequenos desvios de comportamento do processo que, devido a sua pequena amplitude, não são considerados falhas. Apesar desses leves distúrbios não serem considerados um problema, eles podem ser entendidos como o começo de um. Dessa forma, pode-se acompanhar o desenvolvimento de uma falha, evitando que a mesma surpreenda e cause danos mais expressivos. Baseado nisso, pode-se fazer previsões e agendamento de manutenções preventivas, ou o que parecer melhor em cada situação.

Somando-se às novas funcionalidades acrescidas pela arquitetura proposta, a tese traz ainda as seguintes contribuições:

- proposta de uma arquitetura para detecção de falhas em sistemas dinâmicos baseada em identificação paramétrica, onde se relaciona, de maneira representativa, as grandezas físicas do processo monitorado com os parâmetros do modelo;
- simplificação da etapa de diagnóstico, a partir de conclusões diretas indicadas pela representatividade física dos parâmetros estimados em modo de falha;
- análise de satisfação de critérios e desempenho com diversas ferramentas que, de algum modo, satisfazem o diagrama de blocos da arquitetura proposta, mostrado na Figura 4.2;
- validação da proposta através de modelos simulados e processos reais detectando falhas abruptas, incipientes e intermitentes;
- possibilidade de se detectar a ocorrência de diferentes falhas, mesmo que essas estejam ocorrendo simultaneamente;
- possibilidade de detectar falhas incipientes, que atualmente se apresentam como um grande desafio na área;
- aplicação em ambientes reais, onde se verifica a presença de ruídos de medição e de processo;
- aplicação direta na indústria, em casos onde se tem os modelos fenomenológicos conhecidos, como por exemplo: tanques, motores, etc.;

- o uso do algoritmo UKF usado de forma a estimar parâmetros mostrando-se ideal aos objetivos da arquitetura proposta, sendo inclusive comparado ao EKF, onde este, por sua vez, apresentou deficiências na estimação de parâmetros em situações de falhas;

As contribuições supracitadas, apontam uma nova direção a ser desenvolvida e aperfeiçoada dentro da área de detecção e isolamento de falhas. A abordagem paramétrica, utilizando modelos caixa cinza, traz uma série de conseqüências positivas às soluções que a adotam. Outras melhorias ainda podem ser implementadas na solução proposta. Elas são indicadas como trabalhos futuros na próxima seção.

## 6.1 Trabalhos Futuros

Apesar de trazer contribuições e estar bem validada, a proposta ainda pode receber melhorias e adaptações para serem aplicadas a outros tipos de processos. A seguir, alguns pontos que ficarão para serem desenvolvidos posteriormente à apresentação desta tese:

- implementar os conceitos de filtros adaptativos ao UKF. O que vai trazer o benefício da estimação das matrizes de covariância dos ruídos de processo e medição  $Q$  e  $R$ , respectivamente. Isso simplifica a tarefa de tratar o ruído, o que pode ser decisivo em situações onde o ruído é muito alto;
- a adaptação da proposta visando adicionar a funcionalidade de apontar falhas nos sensores e atuadores, mesmo quando o objeto principal de monitoramento é o processo.

## 6.2 Publicações

Até o presente momento, foram obtidas as seguintes publicações relacionadas a esta tese:

- “Identificação Neural de um Sistema de Níveis em Ambiente Foundation Fieldbus” - Congresso Latino Americano de Automática, 2006 [Fernandes et al. 2006];
- “Faults Neural Classifier Applied to Levels Control Real System” - International Symposium on Industrial Electronics, 2007 [Fernandes et al. 2007c];
- “An Implementation of a Fault Detection and Isolation System on Foundation Fieldbus Environment” - Emerging Technologies, Robotics and Control Systems, 2007 [Silva et al. 2007];
- “Faults Detection and Isolation Based On Neural Networks applied to a Levels Control System” - International Joint Conference on Neural Networks (IJCNN), 2007 [Fernandes et al. 2007b];
- “Detecção e Isolamento de Falhas em um Sistema de Níveis Real” - Simpósio Brasileiro de Automação Inteligente (SBAI), 2007 [Fernandes et al. 2007a];

- “Implementação de um Sistema de Detecção de Falhas Incipientes para um Sistema de Nível” - Congresso Brasileiro de Automática, 2008 [e Luiz Affonso Guedes e Adrião D. D. Neto 2008].

As publicações referentes à abordagem que utiliza o UKF encontram-se em fase de submissão.

---

# Referências Bibliográficas

---

- Aguirre, Luís Antônio (2007), *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*, 3ª edição, UFMG.
- Al-Taani, Ahmad T. (2005), An expert system for car failure diagnosis., em 'IEC (Prague)', pp. 457–560.
- An, L. & N. Sepeshri (2003), 'Hydraulic actuator circuit fault detection using extended kalman filter', *Proceedings of the American Control Conference* .
- Angeli, C. & A. Chatzinikolaou (2004), 'On-line fault detection techniques for technical systems: A survey', *International Journal of Computer Science and Applications* **I**(1), 12–30.
- Armengol, Joaquim, Josep Vehf, Miguel Àngel Sainz & Pau Herrero (2003), Fault detection in a pilot plant using interval models and multiple sliding time windows., em 'SAFEPROCESS'.
- Astrom, Karl Johan & Jacob Apkarian (n.d.), *Quanser Engineering Trainer (QET) Series: USB QICii Laboratory Workbook - DC Motor Control Trainer*.
- Billings, S. A. (1980), 'Identification of nonlinear systems - a survey', *IEE Proceedings-D Control Theory and Applications* **127**(6), 272–285.
- Brown, R. G. & P. Y. C. Hwang (1992), *Introduction to Random Signals and Applied Kalman Filtering*, second edition<sup>a</sup> edição, John Wiley and Sons, Inc.
- de Souza, João A. M. F., R. K. H. Galvão & T. Yoneyama (2000), A novel wavelet-based scheme for fault detection, em 'Proceedings of the UK Control Conference'.
- Ding, S. X., S. Schneider, E. L. Ding, A. Rehm & Robert Bosch GmbH (2005), Advanced model-based diagnosis of sensor faults in vehicle dynamics control systems, em '16th IFAC World Congress'.
- e Antonio Pietrosanto, Giovanni Betta (2000), 'Instrument fault detection and isolation: State of the art and new research trends', *IEEE Transactions on Instrumentation and Measurement* **49**(1), 100–107.
- e Luiz Affonso Guedes e Adrião D. D. Neto, Diego R. C. Silva (2008), 'Implementação de um sistema de detecção de falhas incipientes para um sistema de nível', *Congresso Brasileiro de Automática* .

- Fernandes, Raphaela Galhardo, Diego Rodrigo Cabral Silva, Luiz Affonso H. Guedes de Oliveira & Adrião Duarte Dória Neto (2006), 'Identificação neural de um sistema de níveis em ambiente foundation fieldbus', *Congresso Latino Americano de Automática* .
- Fernandes, Raphaela Galhardo, Diego Rodrigo Cabral Silva, Luiz Affonso H. Guedes de Oliveira & Adrião Duarte Dória Neto (2007a), 'Detecção e isolamento de falhas em um sistema de níveis real', *Simpósio Brasileiro de Automação Inteligente (SBAI)* .
- Fernandes, Raphaela Galhardo, Diego Rodrigo Cabral Silva, Luiz Affonso H. Guedes de Oliveira & Adrião Duarte Dória Neto (2007b), 'Faults detection and isolation based on neural networks applied to a levels control system', *International Joint Conference on Neural Networks (IJCNN)* .
- Fernandes, Raphaela Galhardo, Diego Rodrigo Cabral Silva, Luiz Affonso H. Guedes de Oliveira & Adrião Duarte Dória Neto (2007c), 'Faults neural classifier applied to levels control real system', *International Symposium on Industrial Electronics* .
- Freitas, R. D. G., A. L. Maitelli & A. O. Salazar (2004), 'An wavelet approach to pipeline leak detection by pressure analysis', *Proceedings of OMAE2004: 23rd International Conference on Offshore Mechanics and Arctic Engineering* .
- Fries, T.P. & J.H. Graham (2003), A fuzzy intelligent agent approach to fault diagnosis., em 'Proceeding (388) Intelligent Systems and Control'.
- Fu, LiMin (1994), *Neural Networks In Computer Intelligence*, McGraw Hill.
- Galvão, R. K. H. & T. Yoneyama (1998), A method for fault detection in control systems based on the wavelet transform, em 'Anais do XII Congresso Brasileiro de Automática', pp. 1741–1746.
- Gao, X. Z., S. J. Ovaska & X. Wang (n.d.), 'Neural networks-based fault detection with applications in ink jet printers'.
- Gertler, J. (1998), A survey of model-based failure detection and isolation in complex plants., em 'IEEE Control Systems Magazine', Vol. 8, pp. 3–11.
- Grewal, Mohinder S. & Angus P. Andrews (1993), *Kalman Filtering Theory and Practice.*, Prentice Hall.
- Haykin, Simon (2001a), *Kalman Filtering and Neural Networks*, Simon Haykin.
- Haykin, Simon (2001b), *Redes neurais: princípios e prática*, 2nd<sup>a</sup> edição, Bookman.
- Haykin, Simon (2002), *Adaptive Filtering Theory*, Prentice Hall.
- Hu, W., M. Schroeder & A.G. Starr (1999), A knowledge-based real-time diagnostic system for plc controlled manufacturing systems, em 'IEEE SMC '99 Conference Proceedings'.



- IEE (1994), *Fault Diagnosis in Nonlinear Dynamic Systems via Neural Networks*.
- Isermann, Rolf (2006), *Fault-Diagnosis Systems An Introduction from Fault Detection to Fault Tolerance*, Springer.
- Jacobs, O. L. R. (1993), *Introduction to Control Theory*, second edition<sup>a</sup> edição, Oxford University Press.
- Jones, J. G. & M. J. Corbin (1989), *Band-limiting filter approach to fault detection. Fault Diagnosis in Dynamics Systems - Theory and Application*, Prentice Hall.
- Julier, S. J. & J. K. Uhlmann (n.d.), 'Unscented filtering and nonlinear estimation', *Proceedings of the IEEE* pp. 401–422.
- Julier, S. J., J. K. Uhlmann & H. Durrant-Whyte (1995), 'A new approach for filtering nonlinear systems', *Proceedings of the American Control Conference* pp. 1628–1632.
- Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Transactions of the ASME Journal of Basic Engineering* pp. 35–45.
- Koscielny, J. M., M. Syfert & M. Bartys (1999), Fuzzy-logic fault diagnosis of industrial process actuators., *em 'Journal of Applied Mathematics and Computer Science'*.
- Lerro, D. & Y. Bar-Shalom (1993), 'Tracking with unbiased consistent converted measurements versus ekf', *IEEE Transactions on Aerospace and Electronic Systems* pp. 1015–1022.
- Linarié, Davor & Vladimir Koroman (2003), Fault diagnosis of a hydraulic actuator using neural network, *em 'IEEE, ICIT / Karel Jezernik'*.
- Ljung, L. (1987), *System Identification: Theory for the user*, Prentice-Hall, Englewood Cliffs, NJ.
- Lou, S.J., H. Budman & T.A. Duever (2003), 'Comparison of fault detection techniques', *Journal of Process Control* **13**, 451–464.
- Lunze, J. & J. Schroder (2004), 'Sensor and actuator fault diagnosis of systems with discrete inputs and outputs', *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS* **34**(2), 1096–1107.
- Madani, Kurosh (1999), 'A survey of artificial neural networks based fault detection and fault diagnosis techniques', *Proceedings of IEEE*.
- Mallet, Yvette, Danny Coomans, Jerry Kautsky & Olivier De Vel (1997), Classification using adaptive wavelets for feature extraction, *em 'IEEE Transactions on Pattern Analysis and Machine Intelligence'*, Vol. 19, pp. 1058–1066.
- Martins, R. S.; Duarte, V. J. L.; Maitelli A. L.; Salazar A. O.; Dória Neto A. D.; Medeiros T. Q. e Avelino A. M. (2007), 'Método de detecção de vazamentos com base em wavelet e software embarcado em dsp', *Rio PipeLine 2007*.

- Maybeck, Peter S. (1979), *Stochastic Models, Estimation, and Control*, Academic Press, Inc 1.
- Mehranbod, Nasir, Masoud Soroush & Chanin Panjapornpon (2004), 'A method of sensor fault detection and identification', *Journal of Process Control*.
- Miller, W. T., R. Sutton & P. J. Werbos (1995), *Neural Networks for Control*, The MIT Press.
- Monsef, H., A. M. Ranjbar & S. Jadid (1997), Fuzzy rule-based expert system for power system fault diagnosis, em 'IEE Proceedings - Generation, Transmission and Distribution'.
- Naughton, J. M., Y. C. Chen & J. Jiang (1996), A neural networks application to fault diagnosis for robotic manipulator, em 'Ryerson Polytechnic University Toronto and University of Western Ontario London'.
- Ogata, Katsuhiko (1998), *Engenharia de Controle Moderno*, LTC.
- Paiva, Henrique Mohallem (2003), Detecção de falhas em sistemas dinâmicos empregando transformadas wavelets adaptativas, Dissertação de mestrado, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil.
- Qi, Song & Han Jian-Da (2008), 'An adaptive ukf algorithm for the state and parameter estimations of a mobile robot', *Acta Automatica Sinica* 34(1), 72–79.
- Rich, Elaine & Kevin Knight (1994), *Inteligência Artificial*, Makron Books.
- Russell, Stuart (2003), *Inteligência Artificial*, Campus.
- Shin, Hyun Joon, Dong-Hwan Eom & Sung-Shick Kim (2005), 'One-class support vector machines - an application in machine fault detection and classification', *Computer and Industrial Engineering* pp. 395–408.
- Silva, Diego Rodrigo Cabral, Raphaela Galhardo Fernandes, Luiz Affonso H. Guedes de Oliveira & Adrião Duarte Dória Neto (2007), 'An implementation of a fault detection and isolation system on foundation fieldbus environment', *Emerging Technologies, Robotics and Control Systems* 2, 62–68.
- Simani, Silvio (2005), Identification and fault diagnosis of a simulated model of an industrial gas turbine, em 'IEEE Transactions on Industrial Informatics', Vol. 1, pp. 202–216.
- Srinivasan, Arvind & Celal Batur (1994), Hopfield/art-1 neural network-based fault detection and isolation, em 'IEEE Transactions on Neural Networks', Vol. 5.
- Tatara, Eric, Cenk Ündey, Bruce Williams, Gülnur Birol & Ali Çinar (2000), A real-time knowledge-based system for automated monitoring and fault diagnosis of batch processes, em 'AIChE'.

*Technical Information: Foundation Fieldbus* (n.d.), Samson.

Tinos, Renato & Marco H. Terra (1998), Fault detection and isolation in robotic manipulators and the radial basis function network trained by the kohonen's self-organizing map., *em* 'SBRN', pp. 85–90.

Van Overschee, P De Moor, B. (1994), 'N4sid: Subspace algorithms for the identification of combined deterministic-stochastic system', *Automatica* **30**(1), 75–93.

Venkatasubramanian, Venkat, Raghunathan Rengaswamy, Kewen Yin & Surya N. Kavuri (2003), 'A review of process fault detection and diagnosis part i: Quantitative model-based methods', *Computers and Chemical Engineering* (27), 293–311.

W., Sherer & C. White (1989), 'A survey of expert systems for equipment maintenance and diagnostics', *Knowledge-based system Diagnosis, Supervision and Control*. .

Walker, B.K. & Kuang-Yang Huang (1995), 'Fdi by extended kalman filter parameter estimation for an industrial actuator benchmark', *Control Eng. Practice* **3**(12), 1769–1774.

Welch, Greg & Gary Bishop (2006), 'An introduction to the kalman filter'.

Witczak, Marcin (2005), 'Soft computing in fault detection and isolation'.

Zhang, O. & Y. Yan (2001), A wavelet-based approach to abrupt fault detection and diagnosis of sensors, *em* 'IEEE Transactions on Instrumentation and Measurement', Vol. 50, pp. 1389–1396.