

DIAGNÓSTICO DE FALHAS EM SISTEMAS A EVENTOS DISCRETOS:
UMA PROPOSTA DE APLICAÇÃO EM PROCESSOS DE SEPARAÇÃO
ÓLEO-GÁS

Manuel Humberto Manyari Rivera

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Amit Bhaya, Ph.D.

Prof. Eugenius Kaszkurewicz, D.Sc.

Dr. Mario Cesar Mello Massa de Campos, Dr.Eng.

Prof. Antonio Eduardo Carrilho da Cunha, Dr.Eng.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2007

MANYARI RIVERA, MANUEL HUMBERTO

Diagnóstico de Falhas em Sistemas a Eventos
Discretos: Uma Proposta de Aplicação em Pro-
cessos de Separação Óleo-Gás [Rio de Janeiro]
2007

XVII, 134 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia Elétrica, 2007)

Dissertação – Universidade Federal do Rio de
Janeiro, COPPE

- 1 - Sistemas a Eventos Discretos
- 2 - Detecção e Diagnóstico de Falhas
- 3 - Redes de Petri

I. COPPE/UFRJ II. Título (série)

*Aos meus pais Elena e Manuel
e a minha querida irmã Betzabeth.*

Agradecimentos

Agradeço a Deus Pai, quem guia meu caminho sempre. Aos meus pais Elena, Manuel e a minha irmã Betzabeth, pelo seu amor, dedicação e pela força que me deram nestes anos para seguir em frente.

Agradeço aos professores do PEE/COPPE, em especial ao Prof. Amit Bhaya pela sua amizade, paciência, interesse e conselhos na orientação da minha dissertação. Ao Prof. João C. Basílio pela amizade, iniciação na linha de pesquisa e valiosos conselhos; ao Prof. Fernando Lizarralde pelo grande apoio para iniciar meu mestrado na COPPE, ao Prof. Antonio Carrilho pelas suas importantes dicas e ao Prof. Jesús Talavera pelos seus conselhos.

Agradeço aos meus tios Alberto, Lourdes e minha prima Jameny, pela sua ajuda e companhia na minha estadia no Brasil.

Meus agradecimentos às pessoas especiais e amigas que me brindaram sua companhia, apoio moral e amizade durante o mestrado: Erika, Mariela, Kely e Renata. Aos amigos que conheci no Brasil, Ivan, José, Riquelme, Orlando, Napoleón e Enoch, que me fizeram sentir como em casa. À minha galera, que ainda estando longe, sempre me brindaram sua desinteressada amizade Ivan, Alvaro, Walter, Henry e Roberto.

Meus agradecimentos à CNPq pelo apoio financeiro.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DIAGNÓSTICO DE FALHAS EM SISTEMAS A EVENTOS DISCRETOS:
UMA PROPOSTA DE APLICAÇÃO EM PROCESSOS DE SEPARAÇÃO
ÓLEO-GÁS

Manuel Humberto Manyari Rivera

Abril/2007

Orientador: Amit Bhaya

Programa: Engenharia Elétrica

Sistemas de diagnóstico e isolamento *on-line* de falhas têm sido largamente utilizados em processos industriais. A teoria de sistemas a eventos discretos é cada vez mais popular neste contexto. Esta dissertação estuda questões teóricas e práticas no projeto de sistemas automáticos para diagnóstico de falhas em sistemas a eventos discretos modelados por redes de Petri. As principais contribuições do trabalho são uma definição de compatibilidade de modelos (de redes de Petri), extensão de um procedimento existente para mapeamento de sensores, estudo de diagnosticabilidade de modelos e a proposta de uma metodologia integrada para a construção de diagnosticadores. Serão extraídas as características qualitativas de medidas quantitativas para aplicação no projeto de diagnosticadores para sistemas reais que possam ser modelados por sistemas a eventos discretos. Uma aplicação industrial é utilizada como caso de estudo para ilustrar os resultados teóricos obtidos. A aplicação é o projeto de um diagnosticador para o processo de separação de uma planta de produção de óleo em plataforma marítima (*offshore*).

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FAULT DIAGNOSIS IN DISCRETE EVENT SYSTEMS: A PROPOSAL OF
APPLICATION IN OIL-GAS SEPARATION PROCESSES

Manuel Humberto Manyari Rivera

April/2007

Advisor: Amit Bhaya

Department: Electrical Engineering

On-line fault diagnosis and isolation systems have widespread application in industrial processes and the theory of Discrete Events Systems for fault detection and diagnosis is increasingly popular. This dissertation addresses theoretical and practical issues in the design of automatic fault diagnosis systems for discrete event systems modeled by Petri nets. The main contributions are a definition of compatibility of (Petri net) models, extension of an existing sensor mapping procedure, a study of diagnosability of models and proposal of an integrated methodology for the construction of diagnosers. We extract the qualitative characteristics of quantitative measures to apply in the design of diagnosers for real systems that can be modeled by discrete event systems. An industrial application is used as a case study to illustrate the theoretical results obtained. The application consists of the design of a diagnoser for the oil-gas separation process of an oil production plant for offshore platforms.

Resumen de Disertación presentada a la COPPE/UFRJ como parte de los requisitos necesarios para la obtención del grado de Master en Ciencias (M.Sc.)

DIAGNÓSTICO DE FALLAS EN SISTEMAS A EVENTOS DISCRETOS: UNA
PROPUESTA DE APLICACIÓN EN PROCESOS DE SEPARACIÓN
PETRÓLEO-GAS

Manuel Humberto Manyari Rivera

Abril/2007

Orientador: Amit Bhaya

Programa: Ingeniería Eléctrica

Sistemas de diagnóstico y aislamiento *on-line* de fallas tienen amplia aplicación en procesos industriales. La teoría de sistemas a eventos discretos es cada vez más difundida en este contexto. Esta disertación estudia temas teóricos y prácticos en el diseño de sistemas automáticos para diagnóstico de fallas en sistemas a eventos discretos modelados por redes de Petri. Las principales contribuciones de este trabajo son una definición de compatibilidad de modelos (en redes de Petri), extensión de un procedimiento existente para mapeamiento de sensores, un estudio de la diagnosticabilidad de modelos y la proposición de una metodología integrada para la construcción de diagnosticadores. Serán extraídas las características cualitativas de mediciones cuantitativas para aplicar en el diseño de diagnosticadores para procesos reales que pueden ser modelados por sistemas a eventos discretos. Una aplicación industrial es utilizada como caso de estudio para ilustrar los resultados teóricos obtenidos. La aplicación consiste en el diseño de un diagnosticador para el proceso de separación de una planta de producción de petróleo en plataformas marítimas (*offshore*).

Sumário

1	Introdução	1
1.1	Importância dos Sistemas de Detecção e Diagnóstico de Falhas . . .	1
1.2	Métodologias existentes para detecção e diagnóstico de Falhas . . .	3
1.3	Objetivos do Trabalho	6
2	Diagnóstico de Falhas em Sistemas a Eventos Discretos	8
2.1	Sistemas a Eventos Discretos	8
2.1.1	Eventos	9
2.1.2	Características dos Sistemas a Eventos Discretos	9
2.1.3	Linguagem associada a um Sistema a Eventos Discretos . . .	11
2.2	Autômatos	13
2.3	Modelagem por Redes de Petri	15
2.3.1	Matriz de Incidência e Equação de Estado	18
2.3.1.1	Matriz de Incidência	18
2.3.1.2	Equação de Estado	19
2.3.1.3	Alcançabilidade	21
2.3.1.4	Limitabilidade	21
2.3.2	Linguagens em Redes de Petri	22
2.3.2.1	Rede de Petri Rotulada	22
2.3.2.2	Linguagem Gerada por Redes de Petri	22
2.3.3	Projeção	23
2.3.4	Projeção Inversa	23
2.3.5	Lugares iniciais de uma Rede de Petri	24
2.3.6	Operações com Redes de Petri	24
2.4	Compatibilidade de Modelos em Redes de Petri	26

2.5	Mapeamento de Sensores em Redes de Petri	30
2.6	Diagnóstico de Falhas	32
2.6.1	O Diagnosticador	36
2.6.2	Estrutura do Diagnosticador	37
2.6.3	Procedimento de construção de diagnosticadores	38
2.6.3.1	Propriedades do diagnosticador	40
2.6.4	Condições necessárias e suficientes para diagnosticabilidade .	40
2.7	Metodologia do Projeto de Sistemas de Detecção de Falhas	43
2.7.1	Arquitetura do sistema de Diagnóstico	44
2.7.2	Procedimento de Diagnóstico	44
2.8	Comentários e contribuições preliminares	46
3	O Processo de Separação Óleo-Gás	47
3.1	Introdução	47
3.2	Descrição do Processo	48
3.2.1	Subsistema de Produção	48
3.2.2	Subsistema de Aquecimento	50
3.2.3	Primeiro Estágio de Separação	50
3.2.4	Segundo Estágio de Separação	51
3.2.5	Exportação de Óleo	51
3.3	Modelagem Matemática do Processo	52
3.3.1	Trocador de Calor	52
3.3.2	Vasos Separadores	54
3.3.3	Válvulas de Controle	56
3.3.4	Bomba Centrífuga	58
3.3.5	Controladores	59
3.3.6	Placas Orifício	60
3.3.7	Filtros	61
3.4	Falhas no Processo	62
3.4.1	Falhas no trocador de calor	62
3.4.2	Falhas nas válvulas de controle:	62
3.4.3	Falhas nos vasos separadores:	63
3.4.4	Falhas na bomba:	63

3.4.5	Falhas nos filtros:	64
3.5	Simulação	64
4	Projeto dos Diagnosticadores no Processo de Separação Óleo-Gás	69
4.1	Introdução	69
4.2	Diagnosticadores do Processo de Separação Óleo-Gás	70
4.2.1	Trocador de Calor:	70
4.2.2	Primeiro Estagio de Separação:	79
4.2.2.1	Malha de Nível:	80
4.2.2.2	Malha de Pressão:	88
4.2.3	Segundo Estagio de Separação:	94
4.2.3.1	Malha de Nível:	95
4.2.3.2	Malha de Fluxo:	103
4.3	Validação dos Modelos	110
4.4	Simulações do Processo-Diagnosticador	111
4.4.1	Parâmetros da simulação:	114
4.4.2	Avaliação das simulações:	115
4.5	Opções de implementação	121
5	Conclusões e trabalhos futuros	124
	Referências Bibliográficas	127
	Apêndice	131
A	Símbolos P&ID	131
A.1	Linhas de conexão	131
A.2	Instrumentos ou funções gerais	131
A.3	Atuadores e elementos do processo	132

Lista de Figuras

1.1	Arquitetura de um sistema de supervisão	3
2.1	Modelo de Fila de um Sistema de Computador	10
2.2	Figura referente ao exemplo 2.4, <i>este autômato marca a linguagem</i> $L = a, aa, ba, aaa, \dots$ <i>que consiste em todas as seqüências de a e b</i> <i>seguido por a, dado o conjunto de eventos $E = \{a, b\}$</i>	16
2.3	Marcação M_0 da Rede do Exemplo 2.5	19
2.4	Marcação M_2 da Rede do Exemplo 2.5	21
2.5	Composição paralela de três redes de Petri	27
2.6	Lugares/transições em redes de Petri Compatíveis	28
2.7	Redes N e \tilde{N} do exemplo 2.7	29
2.8	Ciclo indeterminado com $P(s_1) = P(s_2) = \sigma_1\sigma_2 \dots \sigma_{nl}$	41
2.9	Arquitetura de um Sistema de Diagnóstico	44
2.10	Procedimento de construção do Sistemas de Diagnóstico	45
3.1	Fluxograma do Processo de Separação Óleo-Gás	49
3.2	Trocador de calor	53
3.3	Detalhe do vaso separador	54
3.4	Curva Característica das válvulas, C_v versus x	57
3.5	Controlador com Antiwindup	61
3.6	Esquema completo de Simulação do trocador de calor	65
3.7	Esquema de Simulação do controlador PID no trocador de calor	65
3.8	Esquema de Simulação da válvula de controle de temperatura do trocador calor	66
3.9	Esquema de Simulação do trocador de calor	66
3.10	Esquema Completo de Simulação do Estágio de Separação	67

3.11	Esquema de Simulação do Controlador de Nível PID, no Vaso Separador	67
3.12	Esquema de Simulação do Vaso Separador	68
3.13	Esquema de Simulação da válvula de Nível, no Vaso Separador	68
4.1	Rede de Petri N_v que representa a Modelagem da válvula TV-01	71
4.2	Rede de Petri N_c que representa a carga de perturbação no trocador de calor	72
4.3	Rede de Petri N_v que representa a modelagem do controlador TIC-01	73
4.4	Rede de Petri N_t do trocador de calor	74
4.5	Diagnosticador G_{tdiag} do trocador de calor representado por N_t	75
4.6	Rede N_{map} , após do mapeamento de sensores em N	77
4.7	Diagnosticador $G_{tmapDIAG}$ da rede N_{map}	79
4.8	Válvula-Filtro de Óleo do primeiro estágio de separação	81
4.9	Carga de Perturbação de Nível no Vaso Separador SG-01	82
4.10	Controlador de Nível do primeiro estágio de separação	83
4.11	Rede N_{ll} que representa a malha de Nível do Vaso Separador SG-01	83
4.12	Modelo que representa a malha de Nível do Vaso Separador SG-01 com o mapeamento de Sensores	86
4.13	Diagnosticador G_{llDiag} da rede N_{llm}	87
4.14	Rede de Petri N_{vp} que representa a Modelagem da válvula PV-01	89
4.15	Rede de Petri N_{cp} que representa a carga na malha de pressão	89
4.16	Rede de Petri N_{pp} que representa a modelagem do controlador TIC-02	90
4.17	Rede de Petri N da malha de pressão do 1ro. estágio de separação	92
4.18	Rede N_{ppm} , após do mapeamento de sensores em N_p	93
4.19	Diagnosticador G_{ppDIAG} da rede N_{ppm}	95
4.20	Válvula-Bomba de Óleo do segundo estágio de separação	96
4.21	Carga de Perturbação de Nível no Vaso Separador SG-02	98
4.22	Controlador de Nível do segundo estágio de separação	98
4.23	Rede N_{lse} que representa a malha de Nível do Vaso Separador SG-02	99
4.24	Modelo que representa a malha de Nível do Vaso Separador SG-02 com o mapeamento de Sensores	101
4.25	Diagnosticador G_{lsDiag} da rede N_{lsm}	102

4.26	Válvula de controle de fluxo de óleo do circuito de exportação	104
4.27	Carga de Perturbação de Fluxo do circuito de exportação	104
4.28	Controlador de fluxo da etapa de exportação de óleo	105
4.29	Rede N_{fs} que representa a malha de fluxo de exportação	106
4.30	Modelo que representa a malha de fluxo da etapa de exportação	108
4.31	Diagnosticador G_{lfDiag} da rede N_{lf_s}	109
4.32	Simulação do trocador de calor no HPSim	110
4.33	A figura mostra a ligação entre o simulador do trocador de calor e o diagnosticador realizado em Stateflow. ambos no ambiente Simulink do MATLAB.	112
4.34	A figura mostra a tela do Stateflow correspondente ao diagnosticador $G_{tmapDIAG}$ (veja Figura 4.7 também).	112
4.35	Simulação dos Sensores Virtuais e Gerador de Eventos no Simulink	113
4.36	Simulação dos Sensores Virtuais e Gerador de Eventos no Simulink	113
4.37	Saídas do Trocador de calor durante comportamento normal	116
4.38	Saídas do gerador de eventos do trocador de calor	116
4.39	Indicador de falhas do diagnosticador durante comportamento normal	117
4.40	Saídas do Trocador de calor durante comportamento com Falha F_1	118
4.41	Saídas do gerador de eventos do trocador de calor com comportamento em falha F_1	118
4.42	Indicador de falhas do diagnosticador com inferência de falha F_1	119
4.43	Saídas do Trocador de calor durante comportamento com Falha F_2	119
4.44	Saídas do gerador de eventos do trocador de calor com comportamento em falha F_2	120
4.45	Indicador de falhas do diagnosticador com inferência de falha F_2	120
A.1	Linhas P&ID de processos e sinais	132
A.2	Símbolos especiais usados para instrumentação no P&ID	132
A.3	Símbolos para elementos finais de controle	134

Notações

Σ	Conjunto de eventos
Σ_o	Conjunto de eventos observáveis
Σ_{uo}	Conjunto de eventos não-observáveis
Σ_f	Conjunto de eventos falha
e_i	Evento
s	Seqüência de eventos
$ s $	Comprimento da seqüência de eventos s
Σ^*	Fecho de Kleene do conjunto de eventos Σ
L_i	Linguagem gerada por um conjunto de eventos
\bar{L}	Prefixo fechamento da linguagem L
G	Autômato determinista
X	Conjunto de estados
E	Conjunto de eventos
f	Função de transição
Γ	Conjunto de evento ativos
x_0	estado inicial
X_m	Conjunto de estados marcados
$\mathcal{L}(G)$	Linguagem gerada pelo autômato G
$\mathcal{L}_m(G)$	Linguagem marcada pelo autômato G
N	Rede de Petri
P	Conjunto de lugares de uma rede de Petri
T	Conjunto de transições de uma rede de Petri
T_o	Conjunto de transições observáveis
T_{uo}	Conjunto de transições não-observáveis
Pre	Função de incidência de arcos de entrada
$Post$	Função de incidência de arcos de saída

M	Marcação ou estado de uma rede de Petri
M_0	Marcação
p_i	Lugar de uma rede de Petri
t_i	Transição de uma rede de Petri
$I(t_i)$	Conjunto de lugares de entrada a t_i
$O(t_i)$	Conjunto de lugares de saída a t_i
$M[t_i >$	Transição t_i habilitada na marcação M
$M[t_i > M_i$	Marcação M_i alcançada desde M através de t_i
$\langle N, M_0 \rangle$	Rede de Petri N com marcação inicial M_0
A	Matriz de incidencia de uma rede de Petri
π	Vetor de disparo
$\mathcal{R}(N, M_0)$	Conjunto alcançável de N
$l(t_i)$	Função de rotulação
P	Projeção
P^{-1}	Projeção inversa
$\mathcal{L}_\sigma(N, M)$	Alcance de M respeito a σ
$S(M, \sigma)$	Conjunto de todas as marcações $M' \in \mathcal{L}_\sigma(N, M)$
p_s	Lugar inicial de uma rede de Petri
\times	Produto de Redes de Petri
\parallel	Composição paralela de Redes de Petri
$\langle N, M_0 \rangle \cong \langle \tilde{N}, \tilde{M}_0 \rangle$	Redes de Petri compatíveis desde seus estados iniciais
h_j	Função de mapeamento sensorial
Y	Conjunto as saídas de um conjunto de sensores
Σ_f	Partição de eventos falha
L/s	Pós-linguagem de L após de s
F_i	Rótulo de falha do tipo i
N_b	Rótulo de comportamento normal
Δ	Conjunto de Rótulos
Q_d	Conjunto de estados de um autômato diagnosticador
δ_d	Função de transição de um autômato diagnosticador
Γ_d	Função de eventos ativos de um autômato diagnosticador
q_0	Estado inicial de um autômato diagnosticador
$LP(M, l, s)$	Função de Propagação de Rótulo Falha
$RA(q, \sigma)$	Função domínio
$LC(q)$	Função de correção de rótulo

Lista de Tabelas

4.1	Medições no Trocador	71
4.2	Válvula do trocador de calor	72
4.3	Carga no trocador de calor	73
4.4	Controlador do trocador de calor	73
4.5	Sensores Virtuais do Trocador de calor	76
4.6	Mapeamento de sensores no trocador de calor	78
4.7	Medições no Primeiro Estágio de Separação	79
4.8	Válvula-Filtro do Controle de Nível do 1ro estágio de Separação	81
4.9	Carga de Nível no Vaso Separador SG-01	82
4.10	Controlador de Nível do Primeiro estágio de separação	83
4.11	Sensores Virtuais do Controle de Nível no Primeiro Estágio de Separação	84
4.12	Mapeamento de Sensores	85
4.13	Válvula de pressão do primeiro estágio de separação	90
4.14	Carga de pressão no primeiro estágio de separação	90
4.15	Controlador do TIC-02	90
4.16	Sensores Virtuais da malha de pressão	92
4.17	Mapeamento de Sensores	94
4.18	Medições no Segundo Estágio de Separação e circuito de re-circulação	95
4.19	Válvula do Controle de Nível do 2do estágio de Separação	97
4.20	Carga de Nível no Vaso Separador SG-02	97
4.21	Controlador de Nível do segundo estágio de separação	98
4.22	Sensores Virtuais do Controle de Nível no Segundo Estágio de Separação	100
4.23	Mapeamento de sensores na malha de nível	102
4.24	Válvula do Controle de fluxo do 2do estágio de Separação	104

4.25	Carga de Fluxo da etapa de exportação	105
4.26	Controlador de fluxo da etapa de exportação de óleo	105
4.27	Sensores Virtuais do Controle de Fluxo de exportação	107
4.28	Mapeamento de Sensores	107
4.29	Sensores Virtuais do Trocador de calor	114
A.1	Letras de identificação dos elementos P&ID	133

Capítulo 1

Introdução

1.1 Importância dos Sistemas de Detecção e Diagnóstico de Falhas

A operação dos sistemas industriais e processos técnicos atuais requerem contínuos avanços na supervisão e diagnóstico das possíveis falhas que podem ocorrer, para melhorar sua confiabilidade, segurança e economia. No controle automático de sistemas e processos de grande complexidade, as funções de supervisão e detecção de falhas servem para indicar os valores das variáveis do processo, indicar estados indesejáveis, proximidade a estados não permitidos bem como para tomar as ações apropriadas com a finalidade de manter uma operação aceitável e evitar danos ou acidentes [13],[27], o que representa uma tarefa desafiadora. É importante diferenciar as funções e hierarquias de controle e supervisão em um sistema automático, entre elas: *i) Monitoramento*: implica na medição de variáveis com considerações de tolerância e a geração de alarmes para o operador. *ii) Proteção automática*: quando os processos entram em um estado perigoso, a proteção automática inicia imediatamente uma ação para diminuir ou sair do estado de risco. *iii) Supervisão com diagnóstico de falhas*: baseado na medição de variáveis e cálculo de características, geram-se sintomas através da detecção de mudanças no processo. Com isto desenvolve-se o diagnóstico e logo é disparada uma ação para conter os efeitos da falha [13].

O monitoramento e a proteção automática são desejáveis para a supervisão de processos; os métodos tradicionais de detecção de falhas checam se algumas variáveis permanecem entre limites máximos ou mínimos, disparando quando

necessário, alarmes e sistemas de proteção. A vantagem destes métodos baseados nos limites de valores é a simplicidade e a confiabilidade. Outros métodos foram criados para sistemas nos quais é preciso uma detecção rápida de pequenas falhas com condutas abruptas e outras de transição lenta, diagnóstico de falhas nos atuadores, sensores e componentes do processo, detecção de falhas em malhas fechadas e supervisão de processos em estados transitórios [13].

Cabe notar que o termo detecção de falha consiste em inferir que uma falha de qualquer natureza ocorreu no sistema, enquanto que o termo diagnóstico de falhas consiste em determinar o tipo de falha e alguns detalhes possíveis como tamanho, localização e tempo de detecção da falha [13].

As falhas, dependendo da sua evolução no tempo, podem ser consideradas como: Falhas abruptas (mudanças rápidas nas variáveis do processo), falhas incipientes (mudanças que evoluem lentamente com o transcorrer do tempo) e falhas intermitentes (mudanças que aparecem e desaparecem no tempo) [13].

O procedimento de diagnóstico de modo geral está baseado na observação dos sintomas analíticos, sintomas heurísticos e do conhecimento heurístico do processo. O conhecimento heurístico e o mecanismo de inferência para diagnóstico são usados para a construção de *Sistema Especialistas* para diagnóstico de falhas. Tais sistemas compreendem a análise do inter-relacionamento de sintomas-falha, estatística de falhas e histórico do processo, diagnóstico usando métodos de classificação de dados mediante reconhecimento de padrões, ou outras técnicas de inteligência artificial, estatística, etc.

A Figura 1.1 mostra a arquitetura de um sistema que contém um subsistema de diagnóstico de falhas [27]. No nível inferior está o sistema com o seu conjunto de controladores e equipamentos de medição; o nível superior é composto do supervisor, que desenvolve tarefas de controle e coordenação dos controladores de nível baixo, diagnóstico de falhas, sistema de recuperação de falhas, reconfiguração, seguida da identificação da falha e a coordenação da operação de todos estes subsistemas. A interface entre estes dois níveis leva informação sobre a ocorrência de eventos observáveis do sistema para o supervisor e comunica os comandos gerados no supervisor para o sistema [27].

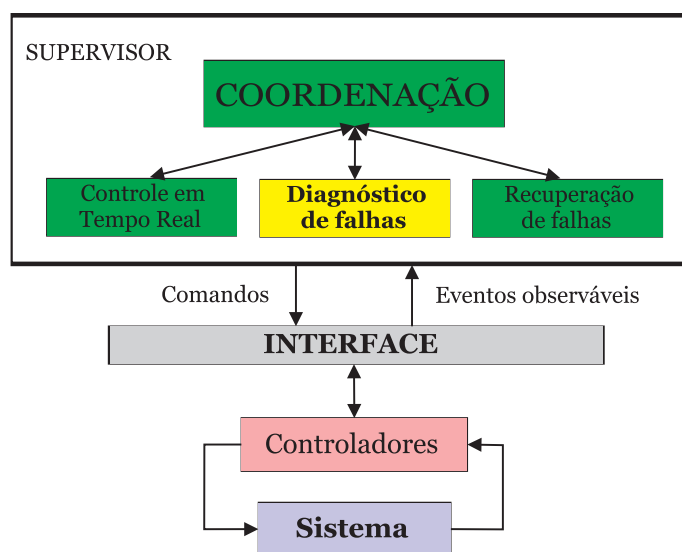


Figura 1.1: Arquitetura de um sistema de supervisão

1.2 Metodologías existentes para detecção e diagnóstico de Falhas

Algumas metodologias de detecção e diagnóstico de falhas utilizam bases de conhecimento extraídas dos operadores do processo e dos dados obtidos do processo real [13][23][30]. Nestas metodologias, as tarefas podem ser divididas em detecção de falhas por geração de sintomas analíticos e heurísticos, e diagnóstico de falhas.

Na geração de sintomas analíticos, o conhecimento analítico do processo é usado para produzir informação quantificável e analisável. Para isto, medições das variáveis do processo devem ser obtidas para gerar valores característicos como: *A Checagem direta de valores limites de tolerância, análise dos sinais medidos diretamente tais como funções de correlação, espectro de frequência, variâncias, amplitudes, etc., análise do processo pelo uso de modelos matemáticos junto com estimação de parâmetros, estimação de estado e estimação de resíduos.* Todas estas informações são comparadas com características do processo em conduta normal.

Os sintomas heurísticos podem ser gerados usando informação qualitativa de operadores da planta ou processo, por intermédio de observações humanas, inspeções, valores característicos heurísticos na forma de ruídos especiais, cores, vibrações, uso, etc. Os sintomas heurísticos são representados por variáveis lingüísticas (pe-

queno, médio, grande). O diagnóstico de falhas consiste em determinar o tipo, tamanho e lugar da falha assim como o seu tempo de deteção baseado nos sintomas analíticos e heurísticos observados.

Além dos métodos baseados em bases de conhecimento, existem métodos de deteção de falhas baseados em modelos. Nos últimos 25 anos diferentes métodos baseados em modelos matemáticos tem sido desenvolvidos [26],[13],[12]. Estas tarefas consistem na deteção de falhas no processo, atuadores e sensores pelo uso de dependências entre as diferentes variáveis medidas. Estas dependências são expressadas por modelos matemáticos. Em [13] e [12] se descrevem os principais métodos de deteção e diagnóstico de falhas baseados em modelos:

- Deteção de falhas com estimação de parâmetros e observadores (Filtro de Kalman, bancos de observadores).
- Deteção de falhas com paridade de equações (Resíduos).
- Diagnóstico de falhas usando métodos de classificação e reconhecimento de padrões, dentro destes métodos existem diferentes aproximações como classificação geométrica e estatística, reconhecimento de padrões com redes neurais, classificação com Clusterização Fuzzy [11].
- Diagnóstico de falhas usando métodos de inferência com aproximações feitas por análise de arvores falha (FTA, fault-tree analysis), análises de eventos falha (ETA, event-tree analysis), aproximações de inteligência artificial, Redes Bayesianas e lógica Fuzzy [16] e [30].
- Diagnóstico de falhas com abordagem de sistemas híbridos [14] e [2].
- Diagnóstico de falhas usando Sistemas a Eventos Discretos [17].

A teoria de controle tem sido tradicionalmente envolvida com o controle de sistemas de variável contínua, modelados por equações diferenciais e a diferenças. Outra importante classe de sistemas é descrita por estados que tomam valores simbólicos ou lógicos ao invés de numéricos, e as mudanças dos estados do sistema são comandadas por eventos de ação externa ou câmbios espontâneos no sistema: Tais sistemas são chamados Sistemas a Eventos Discretos [4].

O diagnóstico de falhas no contexto de Sistemas a Eventos Discretos (SED) iniciou-se nos anos 90. Diferentes autores [26],[27],[28],[31] propõem uma aproximação baseada em estados para realizar o diagnóstico, com problemas de diagnóstico *online* e *offline* onde a idéia principal é testar e observar o comportamento [27]. Os trabalhos de maior importância e impacto na linha de diagnóstico de falhas com SED, foram os trabalhos de Sampath [26],[27],[28]. Sampath *et al.* [26],[27] propuseram uma aproximação para o diagnóstico de falhas aplicável a sistemas que podem ser modelados no contexto de Sistemas a Eventos Discretos. Em [26] e [27] os autores apresentam uma metodologia para a modelagem de sistemas físicos como sistema a eventos, introduzem o conceito de mapeamento de sensores existentes para os modelos em SED, apresentam a definição de diagnosticabilidade de linguagens geradas por SEDs e um procedimento para a construção de diagnosticadores de falhas e, por último, dão as condições necessárias e suficientes para diagnosticabilidade. O diagnóstico de falhas estima a ocorrência a partir de seqüências de eventos parcialmente observadas e se reduz em um problema de estimação da ocorrência de eventos específicos. Em [26],[27],[28] foi feita a hipótese de que os sistemas a eventos discretos são modelados por autômatos de estados finitos. Outros trabalhos usando sistemas modelados por autômatos foram feitos por Lunze e Schröder [19], onde consideraram falhas em sensores e atuadores, visualizaram o problema de diagnóstico como um problema de observação, por intermédio de uma abordagem híbrida usando autômatos estocásticos.

Uma alternativa aos autômatos para modelagem de sistemas a eventos discretos é fornecida pelas redes de Petri. Como será visto, as redes de Petri estão relacionadas aos autômatos no sentido que também representam explicitamente as funções de transição de sistemas a eventos discretos [6]. Uma rede de Petri é uma importante ferramenta que pode ser usada para descrever e estudar o processamento, informação, sistemas de manufatura, etc., modelados como sendo concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e estocásticos [21], [31]. A comparação dos autômatos, estas podem representar a mesma informação que os autômatos, mas com número de estados bem mais reduzidos, isto pela sua capacidade de modelagem paralela.

Ushio *et al.* [31] consideraram algumas extensões do trabalho de Sampath para sistemas modelados por Redes de Petri com um número finito de marcações al-

cançáveis. Wen e Jeng [32] continuaram o estudo apresentado por Ushio e consideraram uma aproximação para verificar a diagnosticabilidade baseada nas propriedades estruturais do diagnosticador; em [31] e [32] supõe-se alguns lugares ou estados da rede de Petri como sendo observáveis ou registráveis e todas as transições de estados ou marcações como sendo não observáveis. Recentemente Giua e Seatzu [10] desenvolveram uma aproximação algébrica para construir um autômato que será utilizado como um diagnosticador de redes de Petri, sem considerações da construção de modelos em redes de Petri. Genç e Lafortune [9], apresentam um algoritmo distribuído para o diagnóstico de Sistemas a Eventos Discretos modelados por redes de Petri sem o estudo das propriedades de diagnosticabilidade. As lacunas nos trabalhos citados, serão objetos de estudo nesta dissertação, que proporrá maneiras de fechá-las.

A indústria do Petróleo tem feito altos investimentos no estudo de fontes de exploração, desenvolvimento de tecnologia em extração de hidrocarbonetos e otimização de níveis de produção [16]. A automação desempenha um papel importante no investimento de novas tecnologias, provendo soluções adequadas a altos níveis de complexidade, segurança e confiabilidade requerida pelos processos industriais. Esta crescente complexidade nas plantas de produção faz necessário o uso de sistemas automáticos, técnicas de controle e supervisão que permitam melhorar a segurança e a qualidade na produtividade durante a operação destas plantas. Em [16], [30], [24], [7] foram construídos sistemas especialistas para a supervisão e diagnóstico de falhas do processo de separação óleo-gás, aplicando diversos métodos de inferência e classificação como, por exemplo, a análise de árvores construídos a partir de uma base de conhecimento.

1.3 **Objetivos do Trabalho**

Este trabalho visa ampliar a teoria existente de Sistemas a Eventos Discretos em detecção e diagnóstico de falhas para sistemas modelados por redes de Petri, e utilizar esta teoria para propor um sistema automático de detecção e diagnóstico de falhas aplicado em um processo industrial real. Especificamente podemos listar os seguintes objetivos:

- Estender a teoria de mapeamento de sensores dada em [26], [27], para sistemas

modelados por Redes de Petri, extraindo as características qualitativas das medições quantitativas.

- Definir uma relação de modelos em redes de Petri, denominada Compatibilidade de modelos, para estabelecer características comparativas entre duas modelagens.
- Integrar o estudo de diagnosticabilidade e técnicas baseadas em sistemas a eventos discretos dadas em [26], [27], [28], [31] para a construção de diagnosticadores de falhas.
- Propor um procedimento sistemático para o projeto de sistemas automáticos de diagnóstico de falhas para sistemas reais modelados por eventos discretos, usando redes de Petri.
- Aplicar o estudo integrado de diagnosticabilidade no Processo de Separação Óleo-Gás de uma plataforma de produção e propor um sistema de diagnóstico de falhas de acordo ao procedimento proposto no presente trabalho.

A organização do texto é como segue:

No Capítulo 1 se mencionam as principais metodologias existentes para detecção e diagnóstico de falhas e os objetivos do presente trabalho.

No Capítulo 2 é apresentado a modelagem de SEDs com aproximações de autômatos e redes de Petri. Se descrevem as propriedades e vantagens de redes de Petri, operações com redes de Petri, compatibilidade de modelos para comparação de duas modelagens e a teoria de mapeamento de sensores em redes de Petri. Também se apresenta a teoria e procedimentos para o diagnóstico de falhas em SED modelados por redes de Petri.

No Capítulo 3 se descreve o processo de separação óleo-gas junto a sua modelagem matemática, são descritas as falhas do processo e se amostra as simulações do modelo construído.

No Capítulo 4 se apresenta os diagnosticadores do processo de separação óleo gás, a validação de modelos e as opções de implementação para o processo real.

No Capítulo 5 são dadas as conclusões finais e a proposta de trabalhos futuros.

Capítulo 2

Diagnóstico de Falhas em Sistemas a Eventos Discretos

O constante desenvolvimento nos sistemas industriais, sistemas de produção, processos de controle e sistemas de comunicação cria uma complexidade de análise na modelagem, planejamento e confiabilidade do sistema em conjunto, assim como na tomada de ações apropriadas para mantê-los em operação e evitar danos ou acidentes. Torna-se necessário trabalhar com ferramentas analíticas aplicadas a estes sistemas para garantir a corretude e confiabilidade da análise feita. Neste contexto, os Sistemas a Eventos Discretos possibilitam a modelagem analítica de uma variada gama de aplicações industriais, e vale notar que a teoria associada foi criada tendo em mente sua utilização no campo prático de Detecção de Falhas, Controle, Supervisão em sistemas de grande e média complexidade.

Neste capítulo será feita a modelagem de sistemas baseada em Redes de Petri e será apresentada a teoria necessária para o desenvolvimento de sistemas de detecção e diagnóstico usando Sistemas a Eventos Discretos, procedimentos de construção de modelos bem como diagnosticadores, considerando as condições necessárias e suficientes para diagnosticabilidade, representando as principais contribuições do presente trabalho.

2.1 Sistemas a Eventos Discretos

Um sistema a Eventos Discretos é um sistema de estados discretos cuja evolução é dirigida pela ocorrência assíncrona de eventos [6]. Quando o espaço de estados de um sistema é naturalmente descrito por um conjunto discreto, (por exemplo

$\{0, 1, 2, \dots\}$, $\{Ocupado, Disponivel, Quebrado\}$) e as transições de estados são somente observadas em pontos discretos no tempo, então associamos cada estado a *eventos* e referenciamos este sistema como um *Sistema a Eventos Discretos (SEDs)* [6].

2.1.1 Eventos

Um Evento pode ser definido como uma ocorrência instantânea que provoca a transição de estados do sistema ou como uma transição assíncrona que modifica o espaço de estados do sistema. O evento pode ser identificado como uma *ação específica* (por exemplo: Apertar um botão, ativação de um sensor de fim de curso), uma ocorrência espontânea (por exemplo: Quebra de uma peça, alarmes de excesso de temperatura) ou como o resultado de condições já esperadas (por exemplo: Chegada do nível de fluido num tanque a um valor estabelecido) [6]. A notação utilizada no presente trabalho segue o seguinte padrão: Para referenciar um evento utilizam-se letras minúsculas (a, e, s), quando consideremos um sistema afetado por diferentes tipos de eventos, definiremos um *conjunto de eventos* Σ , cujos elementos são todos estes tipos de eventos.

2.1.2 Características dos Sistemas a Eventos Discretos

Nas ultimas décadas, a Engenharia de Controle teve grandes sucessos no desenvolvimento enquanto a *Sistemas Dinâmicos de Variável Contínua (SDVC)*, a maioria dos avanços ocorreu em virtude da utilização de modelos baseados em equações diferenciais ou em equações a diferenças, tais modelos têm *espaço de estados contínuo* e a sua transição de estado é *conduzida pelo tempo*. Em contrapartida aos *SDVC*, os *Sistemas a Eventos Discretos (SED)* apresentam um espaço de estado como sendo um *conjunto discreto* e a sua transição de estados é *conduzida por eventos* [21].

Definição 2.1 (*Sistema a Eventos Discretos*)

Um Sistema a Eventos Discretos é um sistema de estados discretos, conduzido por eventos, isto é, a evolução do seu estado depende inteiramente da ocorrência assíncrona de eventos através do tempo [6]. □

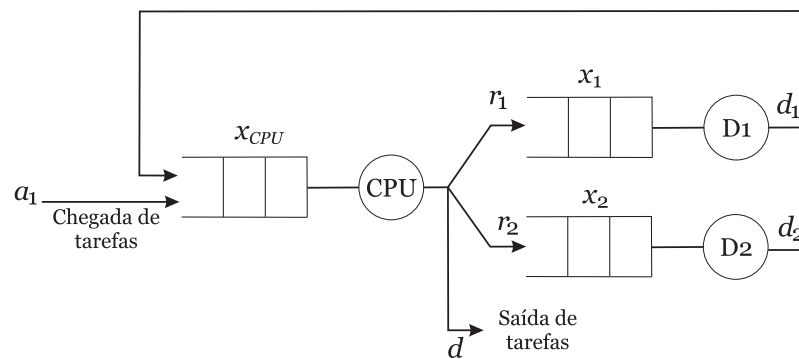


Figura 2.1: Modelo de Fila de um Sistema de Computador

Exemplo 2.1 *Em um típico sistema operacional de computador, os trabalhos, as tarefas ou transações são os clientes quem competem para a atenção de servidores tais como vários processadores, CPU's ou dispositivos periféricos. Quando um servidor está ocupado no instante de requerimento de trabalho, os trabalhos são colocados numa fila, que são uma parte integral do Sistema de Computador. Esta situação é convenientemente representado através de um modelo de fila. Um exemplo de configuração de sistemas de computador é mostrado na Figura 2.1. Neste sistema, trabalhos chegam à fila, uma vez atendidos pela CPU, eles saem ou solicitam acesso a algum dos dois discos, retornando posteriormente para a CPU por mais serviços [6].*

O conjunto de eventos no exemplo normalmente consiste na chegada e saída dos processos aos servidores. Especificamente,

$$\Sigma = \{a, d, r_1, r_2, d_1, d_2\}$$

Sendo

- *a*: a chegada de tarefa proveniente do exterior ao sistema de computador.
- *d*: a saída de tarefa da CPU ao exterior.
- r_1, r_2 : saídas de tarefas da CPU que são encaminhadas aos discos 1 e 2 respectivamente.
- d_1, d_2 : saídas de tarefas dos discos 1 e 2 respectivamente, voltando à CPU.

Uma possível representação de estado do sistema consiste em um vetor $x = [x_{CPU}, x_1, x_2]^T$ correspondendo aos comprimentos das filas da CPU e os discos 1 e 2. Neste caso, o espaço de estados é

$$X = \{(x_{CPU}, x_1, x_2) : x_{CPU}, x_1, x_2 \geq 0\} \quad \blacklozenge$$

2.1.3 Linguagem associada a um Sistema a Eventos Discretos

Nas seções anteriores apresentaram-se algumas das características de Sistemas a Eventos Discretos. Nesta seção será focalizado o estudo de *SEDs* para o desenvolvimento de modelos que descrevem adequadamente o comportamento dos sistemas. Estes modelos servem como base para a análise, a avaliação e o diagnóstico de falhas. Supõe-se que o comportamento dos *SEDs* seja descrito em termos de seqüências de eventos da forma $e_1 e_2 \cdots e_n$. Uma seqüência deste tipo especifica a ordem na qual vários eventos ocorrem no tempo, sem fornecer os instantes de tempo associados à ocorrência desses eventos.

Todo *SED* possui um conjunto subjacente de eventos (Σ) que é o *alfabeto* de uma linguagem, seqüências de eventos são pensadas como as *palavras* na linguagem [6]. Supõe-se que Σ é um conjunto finito de eventos, e ε uma seqüência que consiste em nenhum evento é chamada de *seqüência vazia*. Algumas notações se apresentam a seguir:

- Para s_1 , uma seqüência finita de eventos que pertencem a Σ , tal que $s_1 = \sigma_1 \sigma_2 \sigma_5 \sigma_3$, tem-se que:
 $|s|$, denota comprimento da seqüência s , ou numero de eventos em s
 $\sigma_1 \in s$, denota que σ_1 aparece na seqüência s .
 ε , denota uma seqüência vazia; ou seja $|s| = 0$

Definição 2.2 (*Linguagem*)

Uma linguagem definida sobre um conjunto de eventos Σ é um conjunto de seqüências de comprimento finito, formado pelos eventos de Σ , [6]. \square

A principal operação envolvida na construção de seqüências e linguagens a partir de um conjunto de eventos Σ é a *concatenação*, por exemplo:

Exemplo 2.2 *Sejam as seqüências $s_1 = \sigma_1\sigma_2\sigma_5\sigma_3$ e $s_2 = \sigma_5\sigma_1$, então a concatenação de s_1 com s_2 é dada por:*

$s_1s_2 = \sigma_1\sigma_2\sigma_5\sigma_3\sigma_5\sigma_1$. *O elemento neutro para concatenação é ε : assim, $s_1\varepsilon = \varepsilon s_1 = s_1$.* \blacklozenge

Denota-se por Σ^* , o conjunto de todas as seqüências finitas formadas pela concatenação dos elementos de Σ incluindo a seqüência vazia ε . A operação $(^*)$ é chamada de *Fecho de Kleene*, por exemplo:

Exemplo 2.3 *Seja $\Sigma = \{s_1, s_2, s_3\}$, então:*

$\Sigma^* = \{\varepsilon, s_1, s_2, s_3, s_1s_1, s_1s_2, s_1s_3, s_2s_1, s_2s_2, s_2s_3, s_3s_1, s_3s_2, s_3s_3, s_1s_1s_1, \dots\}$, *representa todas as seqüências contendo nenhum, um ou mais elementos de Σ , permitindo repetições.* \blacklozenge

Esta terminologia será utilizada no desenvolvimento do trabalho:

- **Prefixo, sufixo e sub-seqüência:** Seja $s_1 = \sigma_1\sigma_2\sigma_5\sigma_3$. Então:
 $\sigma_1\sigma_2\sigma_5$ é chamado de prefixo de s_1
 $\sigma_5\sigma_3$ é chamado de sufixo de s_1
 $\sigma_2\sigma_5$ é chamado de uma sub-seqüência de s_1

Operações com Linguagens: As operações usuais de conjuntos, tais como união, intersecção, diferença e complemento com relação a Σ^* , são aplicáveis a linguagens, posto que as linguagens são conjuntos [6]. Adicionalmente usam-se as seguintes operações:

- **Concatenação:** Seja $L_1, L_2 \subseteq E^*$, então:

$$L_1L_2 := \{s \in E^* : (s = s_1s_2) \wedge (s_1 \in L_1) \wedge (s_2 \in L_2)\}$$

Uma seqüência está em L_1L_2 se ela pode ser escrita como a concatenação de uma seqüência de L_1 e uma seqüência de L_2 .

- **Prefixo fechamento:** Seja $L \subseteq E^*$, então:

$$\bar{L} := \{s \in E^* : (\exists t \in E^*)st \in L\}$$

\bar{L} contém todos os prefixos de todas as seqüências de L . Em geral, $L \subseteq \bar{L}$.

- *Fecho de Kleene:* Seja $L \subseteq E^*$, então:

$$L^* := \{\varepsilon\} \cup L \cup LL \cup LLL \cup LLL \dots$$

Qualquer linguagem L sobre Σ é um subconjunto de Σ^* , i.e., $L \subseteq \Sigma^*$ é uma linguagem.

Uma das dificuldades em linguagens é a forma de representá-las. Em outras palavras, precisa-se de um conjunto compacto de estruturas que definam uma linguagem e que possam manipular operações por intermédio de regras bem definidas, e subseqüentemente manipular e analisar linguagens complexas. A seguir se apresenta formalismos de modelamento de autômatos e redes de Petri, com o objetivo de representar e manipular linguagens e solucionar problemas relacionados ao comportamento lógico de SEDs.

2.2 Autômatos

Um Autômato ou Máquina de Estados é um dispositivo capaz de representar uma linguagem de acordo com regras bem definidas. Nesta seção recapitularemos alguns conceitos básicos, seguindo [6].

Definição 2.3 (*Autômato determinista*)

Um Autômato determinista, denotado por G é uma sêxtupla:

$$G = (X, E, f, \Gamma, x_0, X_m)$$

Onde:

X : conjunto de estados.

E : conjunto finito de eventos associados com as transições de G .

$f : X \times E \rightarrow X$, $(x, e) \mapsto y = f(x, e)$, função de transição de estados, indicando que: estando no estado x , existe uma transição rotulada pelo evento e , tal que o próximo estado de G será y .

$\Gamma : X \mapsto 2^E$, conjunto dos eventos ativos.

$$\Gamma(x) = \{e \in E : f(x, e) \text{ é definido}\}$$

x_0 : estado inicial

X_m : conjunto de estados marcados, $X_m \subseteq X$. □

A operação do autômato G segue a seguinte lógica: Começando no estado inicial x_0 e após da ocorrência de um evento $e \in \Gamma(x_0) \subseteq E$ o automato G mudará de estado, representado por $f(x_0, e) \in X$.

Observação 2.1 *Pode-se observar que:*

- Se X é um conjunto finito, chamamos G como Autômato de Estados finitos, ou Máquina de Estados Finitos (MEF).
- A escolha dos estados a serem marcados depende do problema de interesse; geralmente os estados marcados são estados considerados importantes no sistema ou representam o término de alguma tarefa.

Por conveniência, f é sempre estendida do $X \times E$ para o domínio $X \times E^*$ da seguinte maneira recursiva: $f(x, e) := x$ e $f(x, se) := f(f(x, s), e) = x$ para $e \in E$ e $s \in E^*$.

A conexão entre linguagem e autômato é facilmente feita pela inspeção do diagrama de transição de estados de um autômato. Considerando todos os caminhos diretos que podem ser percorridos no diagrama de transição de estado, começando no estado inicial x_0 . Considerando todos os caminhos que iniciam em x_0 e terminam em um estado marcado. Isto nos leva à noção de linguagem gerada e linguagem marcada por um autômato.

Definição 2.4 *(Linguagem gerada por um autômato)*

A linguagem gerada pelo autômato $G = (X, E, f, \Gamma, x_0, X_m)$ é

$$\mathcal{L}(G) = \{s \in E^* : f(x, s) \text{ é definida}\}. \quad (2.1)$$

A linguagem $\mathcal{L}(G)$ representa todos os caminhos direcionados compatíveis com o diagrama de transição de estados, começando do estado inicial, a seqüência correspondente a um caminho é a concatenação dos rótulos dos eventos das transições compondo o caminho. □

Definição 2.5 (*Linguagem marcada por um autômato*)

A linguagem marcada pelo autômato $G = (X, E, f, \Gamma, x_0, X_m)$ é

$$\mathcal{L}_m(G) = \{s \in E^* : f(x, s) \in X_m \text{ é definida}\} \quad (2.2)$$

A linguagem marcada $\mathcal{L}_m(G)$ é o subconjunto de $\mathcal{L}(G)$ consistindo em apenas as trajetórias s nas quais $f(x_0, s) \in X_m$, isto é, as seqüências correspondentes às trajetórias que terminam em um estado marcado do diagrama de transição de estados.

□

Exemplo 2.4 *Considere o autômato da Figura 2.2:*

$$G = (X, E, f, \Gamma, x_0, X_m),$$

sendo $X = \{0, 1\}$ o conjunto de estados, $E = \{a, b\}$ o conjunto de eventos, $x_0 = 0$ o estado inicial, $X_m = \{1\}$ o estado marcado, f é definida como:

$$f(0, a) = 1, \quad f(0, b) = 0$$

$$f(1, a) = 1, \quad f(1, b) = 0$$

O conjunto de eventos ativos é dado por:

$$\Gamma(0) = \{a, b\}, \quad \Gamma(1) = \{a, b\}$$

A linguagem gerada pelo autômato G é o próprio E^* isto é: $\mathcal{L}(G) = \{a, b, aa, ba, bb, aaa, \dots\}$. Posto que $X_m = \{1\}$ então a linguagem marcada é $\mathcal{L}_m(G) = \{a, aa, ba, aaa, aba, baa, bba, \dots\}$.

Pode-se, então, concluir que um autômato G é a representação de duas linguagens $\mathcal{L}(G)$ e $\mathcal{L}_m(G)$, linguagem gerada e linguagem marcada. ♦

2.3 Modelagem por Redes de Petri

As Redes de Petri são uma ferramenta de modelagem gráfica e matemática para uma grande variedade de sistemas, e representam uma ferramenta promissória para descrever e estudar sistemas de processamento, informação, manufatura, etc., os

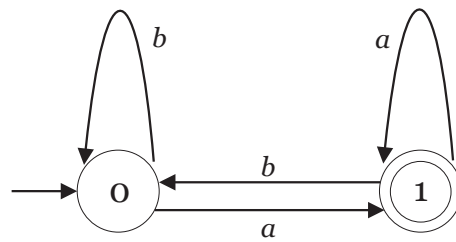


Figura 2.2: Figura referente ao exemplo 2.4, *este autômato marca a linguagem $L = a, aa, ba, aaa, \dots$ que consiste em todas as seqüências de a e b seguido por a , dado o conjunto de eventos $E = \{a, b\}$*

quais freqüentemente são caracterizados por serem concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e estocásticos,[21].

Como ferramenta gráfica, as Redes de Petri podem ser utilizadas como um assistente visual de comunicação, similar aos diagramas de fluxo, diagramas de bloco e redes de comunicação. Adicionalmente, *fichas* (representadas por discos pretos), usadas nas redes de Petri, simulam a dinâmica e concorrência de atividades. Como ferramenta matemática, é possível estabelecer uma equação de estado, equações algébricas, e outros modelos matemáticos que governem o comportamento de um sistema. As Redes de Petri possuem aplicação tanto teórica como prática, sendo assim, elas constituem um poderoso meio de comunicação entre a teoria e a prática, [21].

Considere um SED como sendo modelado por uma Rede de Petri N , definida pela quádrupla [25]:

$$N = (P, T, Pre, Post); \quad (2.3)$$

Onde:

- P , é um conjunto finito, cujos elementos são chamados *lugares*, um tipo de nó no grafo representado por círculos, que são similares a estados em autômatos.
- T , é um conjunto finito, cujos elementos são chamados *transições*, outro tipo de nó no grafo representado por retângulos, sendo: $T = T_o \cup T_u$; onde
 T_o : Conjunto de transições observáveis
 T_u : Conjunto de transições não-observáveis

- $Pre : P \times T \rightarrow \mathbb{N}$; Função de incidência dos arcos de entrada à transição.
- $Post : T \times P \rightarrow \mathbb{N}$; Função de incidência dos arcos de saída da transição.

Para descrever o grafo de uma rede de Petri, é conveniente usar $I(t_j)$ para representar o conjunto de lugares de entrada à transição t_j . Similarmente $O(t_j)$ representa o conjunto de lugares saída desde a transição t_j [6].

Definição 2.6 (*Marcação M*)

A marcação de uma rede de Petri N é um mapeamento $M : P \rightarrow \mathbb{N}$, onde M é um vetor coluna, chamado de vetor marcação que atribui a cada lugar um número inteiro não negativo de fichas (tokens), indicando que o estado associado ao lugar está ativo.

A estrutura do vetor de marcação é da forma:

$$M = [M(p_1) \ M(p_2) \ \dots \ M(p_m)]^T \quad (2.4)$$

Denotamos por $M(p)$ o número de marcações no lugar p (fichas no lugar p), para a marcação $M \in \mathbb{N}^m$, sendo que P contém m lugares. \square

O estado inicial da Rede de Petri é sempre definido como M_0 . Representa-se a Rede de Petri através de um grafo bipartido orientado, cujos vértices são os lugares e as transições da rede. Para cada lugar p e para cada transição t , há $Pre(p, t)$ arcos de p para t , e $Post(t, p)$ arcos de t para p [25].

Notações e terminologia:

Dada uma rede de Petri N e a marcação M , diz-se que a transição t está *habilitada* em M ou então que t é *disparável* em M se:

$$\forall p \in P, \ M(p) \geq Pre(p, t)$$

o que equivale a dizer: seja $t \in T$, a transição t é dita habilitada se e somente se $M(p) \geq Pre(p, t)$; para todo p pertencente ao conjunto de entradas de t . Está habilitação denotamos por $M[t >]$.

Se t está habilitada em M , seja M' a marcação definida por:

$$M'(p) = M(p) - Pre(p, t) + Post(t, p) \quad (2.5)$$

dize-se que t dispara de M para M' , ou que M' é obtida de M após do disparo de t , denota-se isso por: $M[t > M']$ [10],[6].

De modo geral denote-se por $\langle N, M_0 \rangle$, a rede de Petri N com marcação inicial M_0 , , que será a condição inicial para a evolução da dinâmica do sistema.

Dadas uma Rede de Petri N , a seqüência de transições $s = t_1, \dots, t_m$, e a marcação M , dizemos que a seqüência é *disparável para M* se existe uma seqüência de marcações M_0, \dots, M_n , com $M_0 = M$, tal que $\forall i \in 1, \dots, n$, temos que:

$$M_{i-1}[t_i > M_i$$

Se a transição t_1 é disparável em M , e após disparar obtém M_1 , t_2 é disparável para M_1 e obtém M_2 e assim sucessivamente, será denotado como $M[s > M_n$.

2.3.1 Matriz de Incidência e Equação de Estado

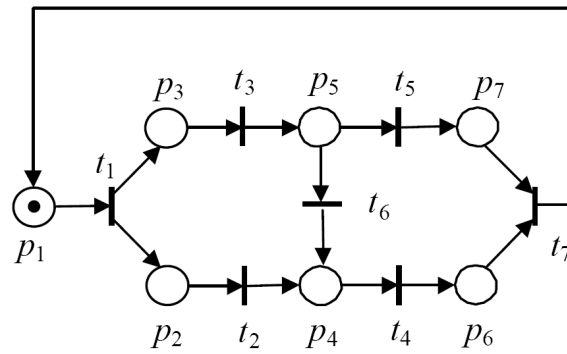
A dinâmica do comportamento de vários sistemas estudados em engenharia podem ser descritos por equações diferenciais ou equações algébricas. De fato é possível descrever e analisar completamente a dinâmica das redes de Petri por equações. Neste sentido, apresentaremos equações matriciais que governam a dinâmica de sistemas concorrentes modelados por Redes de Petri. Entretanto, a solução destas equações é limitada pela natureza não-determinística inerente dos modelos baseados em Redes de Petri. Ditas soluções estão expressadas em restrições posto que as soluções das equações matriciais devem conter números inteiros não negativos como seus elementos.

2.3.1.1 Matriz de Incidência

Para uma Rede de Petri N com n transições e m lugares, a Matriz de Incidência $A = [a_{ij}]$ é uma matriz de números inteiros e seus elementos típicos são dados por:

$$a_{ij} = a_{ij}^+ - a_{ij}^- \quad (2.6)$$

onde $a_{ij}^+ = Post(i, j)$ é a ponderação do arco da transição i até o seu lugar de saída j , e $a_{ij}^- = Pre(j, i)$ é a ponderação do arco até a transição i a partir do seu lugar de entrada i . Construimos as matrizes PRE e $POST$ cujos elementos contém o Pre e $Post$ onde as colunas referenciam as transições e as filas referenciam os lugares. É fácil ver que a regra de transição descrita, representa o número de fichas removidos

Figura 2.3: Marcação M_0 da Rede do Exemplo 2.5

no lugar j quando a transição i dispara. Baseado nas funções de incidência Pre e $Post$, define-se a matriz de incidência A , como sendo:

$$A = POST - PRE \quad (2.7)$$

2.3.1.2 Equação de Estado

Em notação matricial, a marcação M_i é um vetor coluna $n \times 1$. O i -ésimo elemento de M_i denota o número de *fichas* no lugar i imediatamente depois do k -ésimo disparo numa seqüência dada. O k -ésimo disparo ou vetor de disparo u_k é um $n \times 1$ vetor coluna de $n - 1$ 0's (zeros) e um não zero, isto é 1 na i -ésima posição indicando que a transição i disparou no k -ésimo disparo.

Mais formalmente, a definição de um vetor de disparo é dada por $\pi : T^* \rightarrow \mathbb{N}^n$ que é uma função π que associa a alguma transição t um vetor de disparo $u \in \mathbb{N}^n$. Então, $u = \pi(t)$, onde u é da forma $[0 \ 0 \ 1 \dots \ 0 \ 0]^T$.

Posto que a i -ésima linha da matriz de incidência A denota a mudança da k -ésima marcação como o resultado do disparo da transição i , podemos escrever a seguinte equação de estado para uma rede de Petri:

$$M_k = M_{k-1} + A \cdot u_k; \quad k = 1, 2, \dots \quad (2.8)$$

Exemplo 2.5 Considere a seguinte rede de Petri da figura 2.3:

Tem-se que:

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\},$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\};$$

$$T_o = \{t_1, t_4, t_7\}; \quad T_u = \{t_2, t_3, t_5, t_6\};$$

$$M = [M(p_1) \ M(p_2) \ \dots \ M(p_7)]^T \quad e \quad M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$PRE = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$POST = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

A matriz de incidencia é:

$$A = POST - PRE = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

Após o disparo de t_1 :

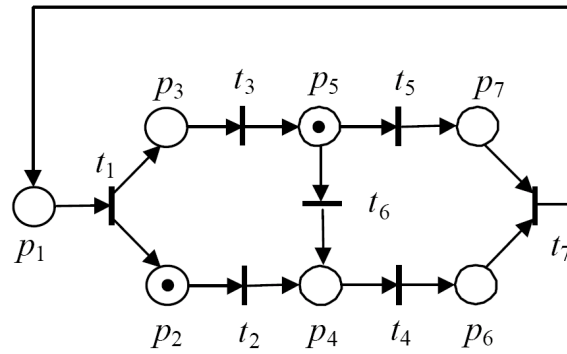
$$u_1 = \pi(t_1) = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T, \text{ obtemos:}$$

$M_1 = M_0 + A \cdot u_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$, que indica a variação de fichas em p_1 , p_2 e p_3 .

Após o disparo de t_3 :

$$u_2 = \pi(t_3) = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T, \text{ obtemos:}$$

$M_2 = M_1 + A \cdot u_2 = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T$, que indica a variação de fichas em p_3 e p_5 . A figura 2.4 mostra o gráfico da rede na Marcação M_2 . ◆

Figura 2.4: Marcação M_2 da Rede do Exemplo 2.5

2.3.1.3 Alcançabilidade

Para um sistema modelado por Redes de Petri é importante saber os possíveis estados aos quais a sua dinâmica pode evoluir. Neste sentido, definiremos o conjunto alcançável a partir de uma marcação dada. Uma marcação M é alcançável em $\langle N, M_0 \rangle$ se e somente se existe $t \in T^*$ tal que $M_0[t > M$ [10]. Baseando-se nessa premissa, mais formalmente definimos o *conjunto alcançável* de $\langle N, M_0 \rangle$, como:

$$\mathcal{R}(N, M_0) := \{M \in \mathbb{N}^m \mid \exists u \in \mathbb{N}^n : M = M_0 + A \cdot u\} \quad (2.9)$$

$\mathcal{R}(N, M_0)$ é o conjunto de todas as marcações M para as quais exista um $n \times 1$ vetor u , inteiro não negativo que satisfaça a equação de estados (2.8).

2.3.1.4 Limitabilidade

Dado um SED modelado por uma Rede de Petri, o problema de limitabilidade consiste em verificar se os lugares podem alcançar um número limitado de marcações ou fichas e determinar seu limite [6]. Isto é:

$$\text{A rede } \langle N, M_0 \rangle \text{ é dita será limitada se } \exists k \geq 0 \mid \forall M \in \mathcal{R}(N, M_0), M(p) \leq k. \quad (2.10)$$

Se a limitabilidade não for satisfeita, é necessário alterar o modelo de forma a assegurar a limitabilidade.

2.3.2 Linguagens em Redes de Petri

Uma vez que, no contexto deste trabalho, as redes de Petri serão utilizadas como um formalismo de modelagem para representar linguagens é necessário especificar mais precisamente a relação entre as transições com os eventos e a respectiva correspondência para cada um deles. Em outras palavras, vamos especificar a linguagem representada por uma Rede de Petri.

Seja Σ o conjunto de eventos do SED em referência, cuja linguagem é modelada por uma rede de Petri, a construção do modelo deve ser tal que, a cada transição em T corresponda um evento distinto do conjunto de eventos Σ , e viceversa. Mas, isto poderia não ser necessariamente restritivo; em autômatos ou máquinas de estados permitimos ter dos arcos diferentes (originando dos estados diferentes) rotulações com o mesmo evento [6]. Isto nos leva à definição de rede de Petri rotulada.

2.3.2.1 Rede de Petri Rotulada

Uma rede de Petri rotulada é uma sêxtupla,

$$N = (P, T, Pre, Pos, \Sigma, l, M_0), \quad (2.11)$$

onde:

- (P, T, Pre, Pos) é o grafo da Rede de Petri.
- Σ é o conjunto de eventos.
- M_0 é o estado ou marcação inicial.
- $l : T \rightarrow \Sigma \cup \{\varepsilon\}$ é a função de rotulação, cada transição é associada a um evento de Σ ou uma seqüência vazia ε . O conjunto de transições $T_u \rightarrow \{\varepsilon\}$, é também chamado *conjunto de transições silenciosas*.

Uma transição só pode ser associada a um único evento, mas um evento não necessariamente corresponde a uma única transição.

2.3.2.2 Linguagem Gerada por Redes de Petri

A linguagem gerada pela rede de Petri Rotulada $N = (P, T, Pre, Pos, \Sigma, l, M_0)$, é:

$$\mathcal{L}(N) := \{l(s) \in \Sigma^* : s \in T^*, M' \in \mathcal{R}(N, M_0), \text{ com } M[s > M']\} \quad (2.12)$$

A linguagem $\mathcal{L}(N)$ representa as seqüências de transições rotuladas que são obtidas por todas as possíveis seqüências de disparos das transições em N iniciando a partir do estado inicial M_0 .

2.3.3 Projeção

Dados dois conjuntos de eventos (Σ, Σ_o) onde um deles é sub-conjunto do outro ($\Sigma_o \subseteq \Sigma$), define-se a projeção neles como: $P : \Sigma^* \rightarrow \Sigma_o^*$ com $\Sigma = \Sigma_o \cup \Sigma_{uo}$, que representa o mapeamento do fecho de Kleene de conjunto de todos os eventos (Σ^*) no fecho de Kleene do conjunto de eventos Observáveis (Σ_o^*):

$$\begin{aligned} P(\sigma) &= \sigma, \text{ se } \sigma \in \Sigma_o \\ P(\sigma) &= \varepsilon, \text{ se } \sigma \in \Sigma_{uo} \\ P(s\sigma) &= P(s)P(\sigma), \text{ } s \in \Sigma^*, \sigma \in \Sigma \end{aligned} \quad (2.13)$$

Na verdade, a Projeção P simplesmente apaga os eventos não observáveis de uma seqüência que contém eventos observáveis e não observáveis. Este tipo de projeção é chamado de projeção natural.

2.3.4 Projeção Inversa

A projeção inversa representa o mapeamento $P^{-1} : \Sigma_o^* \rightarrow 2^{(\Sigma_o \cup \Sigma_{uo})^*}$, onde 2^Σ denota o conjunto das partes de Σ . Dada uma seqüência de eventos correspondente ao subconjunto Σ_o , a projeção inversa retorna o conjunto de seqüências sob o conjunto maior $\Sigma_o \cup \Sigma_{uo}$ que projeta com P a seqüência dada, e é definida como:

$$P^{-1}(y) = \{s \in L : y = P(s)\} \quad (2.14)$$

Consideremos uma marcação $M \in \mathcal{R}(N, M_0)$, uma transição σ rotulada por um evento em Σ_o , uma seqüência de transições u rotuladas por eventos em Σ_{uo} . Então:

$$\mathcal{L}_\sigma(N, M) = \{s = u\sigma : \exists M' \in \mathcal{R}(N, M_0), M[s > M']\} \quad (2.15)$$

Denote-se como $S(M, \sigma)$ ao conjunto de todas as marcações $M' \in \mathcal{L}_\sigma(N, M)$, chamadas *o alcance* de M com respeito a σ .

2.3.5 Lugares iniciais de uma Rede de Petri

Definição 2.7 *Um lugar p_s em uma rede de Petri $\langle N, M_0 \rangle$, é chamado de lugar inicial se:*

- a) p_s não tem transições de entrada, i.e., $I(p_s) = \emptyset$;
- b) A marcação inicial M_0 tem apenas uma ficha no lugar p_s e nenhuma ficha nos demais lugares. □

A seguir será mostrado que toda rede de Petri N pode ser transformada em uma rede de Petri com lugares iniciais N' equivalente, onde a estrutura de N' tem um lugar específico identificado como lugar inicial e denotado por p_s , através do procedimento detalhado a seguir [18]:

Procedimento 1 (Conversão de rede de Petri a rede equivalente com lugar inicial)

PASSO 1 *Iniciar com $N' = N$, e introduzir mais um lugar p_s a P' .*

PASSO 2 *Tornar $T' \leftarrow T' \cup \{t_j'' : t_j \text{ está habilitada em } M_0 \text{ na rede } N\}$.*

PASSO 3 *Para cada transição t'' , fazer $l(t_j'') = l(t_j)$*

$I(t_j'') = p_s$ e $Pre(p_s, t_j'') = 1$ e definir

$$Post(t_j'', p) = M_0 + C(t_j, p), \text{ e}$$

$$O(t_j'') = \{p : Pre'(t_j'', p) > 0\}$$

◇

O propósito do passo 3 é que o disparo da transição t_j'' deve resultar na mesma marcação caso t_j tivesse sido disparada em N . Este processo de conversão a rede com lugares iniciais é importante para realizar as operações com redes de Petri (produto e composição paralela) que se apresentam a seguir.

2.3.6 Operações com Redes de Petri

Serão definidas duas operações importantes para o tratamento e modelagem de redes de Petri, que são *produto* (denotado por \times) e *composição paralela* (denotado por \parallel). Estas operações modelam duas formas de unir o comportamento de um

conjunto de redes de Petri que operam concorrentemente, isto é as redes $N_p = N_1 \times N_2 \times \dots \times N_n$ ou $N_c = N_1 \parallel N_2 \parallel \dots \parallel N_n$, N_p e N_c representam o modelo resultante de dois tipos de interconexão das redes N_1 até N_n .

Sejam duas redes de Petri N_1 e N_2 , definidas como

$$N_1 = (P_1, T_1, Pre_1, Pos_1, \Sigma_1, l_1, M_{20})$$

$$N_2 = (P_2, T_2, Pre_2, Pos_2, \Sigma_2, l_2, M_{10})$$

A construção de N_p , o **produto** das redes de Petri N_1 e N_2 com lugares iniciais p_1 e p_2 respectivamente, é dado pelo seguinte procedimento [18]:

Procedimento 2 (Produto de redes de Petri)

PASSO 1 *Iniciar com N_1 e N_2 de lugares iniciais $p_1 \in P_1$ e $p_2 \in P_2$, definir $P = P_1 \cup P_2$.*

PASSO 2 *Para cada transição $t_j \in T_1$ e $t_k \in T_2$ as quais $l_1(t_j) = l_2(t_k)$, criar uma nova transição t_{jk} a qual dispara se e somente se t_j e t_i disparam da seguinte forma*

$$I(t_{jk}) = I(t_j) \cup I(t_k) \text{ e } Pre(p, t_{jk}) = Pre_1(p, t_j) + Pre_2(p, t_k)$$

$$O(t_{jk}) = O(t_j) \cup O(t_k) \text{ e } Pos(p, t_{jk}) = Pos_1(p, t_j) + Pos_2(p, t_k)$$

T de N_p é o conjunto de todas as t_{jk} .

PASSO 3 *Apagar todos os lugares de P que tenham sido isolados (que não tenham nenhum arco de entrada nem de saída neles).*

PASSO 4 *Definir $M_0(p_1) = M_0(p_2) = 1$ e $M(p_i) = 0, \forall p_i \in P \setminus \{p_1, p_2\}$ ◇*

A construção de N_c , **composição paralela** das redes de Petri N_1 e N_2 com lugares iniciais p_1 e p_2 respectivamente, pode ser obtida de acordo com o seguinte procedimento [20]:

Procedimento 3 (Composição paralela de redes Petri)

PASSO 1 *Iniciar com N_1 e N_2 com lugares iniciais $p_1 \in P_1$ e $p_2 \in P_2$, definir $P = P_1 \cup P_2$, $M_0(p_1) = M_0(p_2) = 1$ e $M(p_i) = 0, \forall p_i \in P \setminus \{p_1, p_2\}$*

PASSO 2 Para cada transição $t_j \in T_1$ e $t_k \in T_2$, definir a transição $t_{jk} \in T_c$, da seguinte forma:

- (i) Se $l_1(t_j) = l_2(t_k)$, criar a nova transição t_{jk} que dispara se e somente se t_j e t_k disparam, e definir $I(t_{jk}) = I(t_j) \cup I(t_k)$ e $Pre(p, t_{jk}) = Pre_1(p, t_j) + Pre_2(p, t_k)$, e $O(t_{jk}) = O(t_j) \cup O(t_k)$ e $Pos(t_{jk}, p) = Post_1(t_j, p) + Post_2(t_k, p)$;
- (ii) Se $l(t_j) \in \Sigma_1 \setminus \Sigma_2$, criar a nova transição t_{jk} que dispara se t_j dispara, e definir $I(t_{jk}) = I(t_j)$ e $Pre(p, t_{jk}) = Pre_1(p, t_j)$, e $O(t_{jk}) = O(t_j)$ e $Post(t_{jk}, p) = Post_2(t_j, p)$
- (iii) Se $l(t_k) \in \Sigma_2 \setminus \Sigma_1$, criar a nova transição t_{jk} que dispara se t_k dispara, e definir $I(t_{jk}) = I(t_k)$ e $Pre(p, t_{jk}) = Pre_2(p, t_k)$, e $O(t_{jk}) = O(t_k)$ e $Post(t_{jk}, p) = Post_2(t_k, p)$.

PASSO 3 Apagar todos os lugares de P que tenham sido isolados (que não tenham nenhum arco de entrada nem de saída neles). \diamond

A importância da composição paralela de redes de Petri é que ela representa o comportamento síncrono dos modelos que compõem o sistema. Na composição paralela, um evento comum somente será executado se ambos os autômatos o executam simultaneamente.

Eventos particulares, isto é, pertencentes a $(\Sigma_1 \setminus \Sigma_2) \cup (\Sigma_2 \setminus \Sigma_1)$ podem ser executados sempre que é possível. Se $\Sigma_1 = \Sigma_2$, então a composição paralela é equivalente ao produto, i.e., $N_1 \parallel N_2 = N_1 \times N_2$.

Se $\Sigma_1 \cap \Sigma_2 = \emptyset$ não há transições sincronizadas, então $N_1 \parallel N_2$ representa comportamento simultâneo de N_1 e N_2 .

Exemplo 2.6 Da figura 2.5, sejam as redes de Petri N_v, N_p e N_c com lugar inicial, a composição paralela é dada pela rede $N_{com} = N_v \parallel N_p \parallel N_c$, ver figura 2.5. Pode-se notar nas redes N_c, N_p que ante o disparo da transição rotulada com evento asp essa evolução se reflete em forma síncrona na rede N_{com} .

2.4 Compatibilidade de Modelos em Redes de Petri

Uma vez feita a modelagem com redes de Petri, uma importante questão é o que pode ser feito com os modelos. As redes de Petri permitem a análise de algumas propriedades de sua funcionalidade, assim como suas possibilidades enquanto a

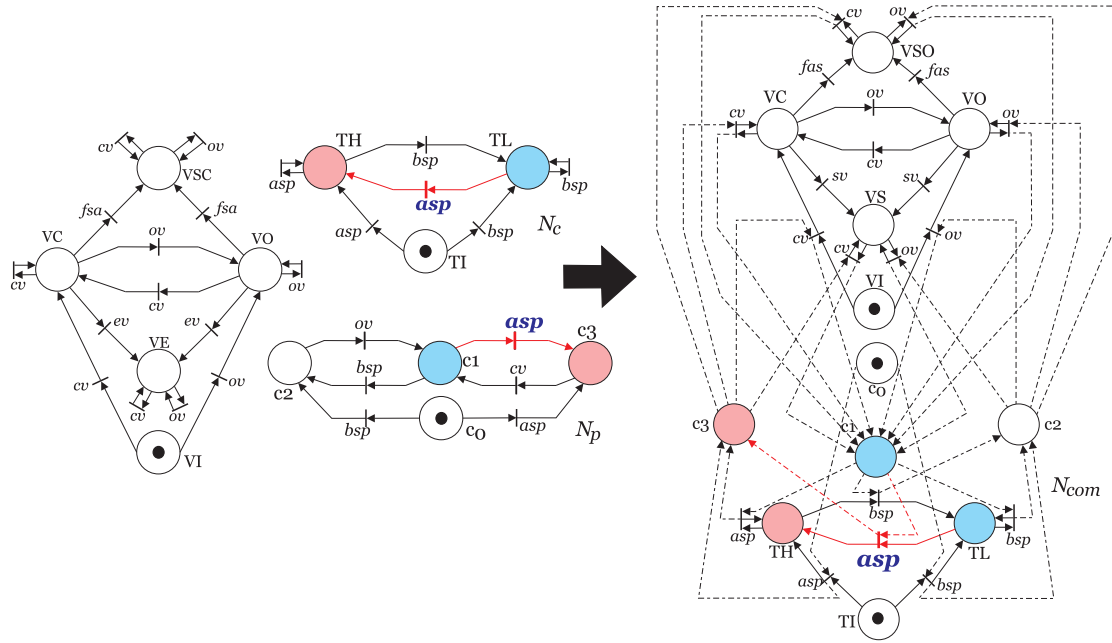


Figura 2.5: Composição paralela de três redes de Petri

variabilidade. Neste ponto nasce a necessidade de comparação entre redes, posto que muitas vezes precisa-se comparar com modelos estabelecidos não só pela linguagem gerada, senão também pela evolução da sua dinâmica interna e a interpretação física das evoluções. Esta observação leva à definição de compatibilidade de redes de Petri, que será importante no desenvolvimento deste trabalho. A definição de compatibilidade de redes de Petri constitui uma das contribuições da presente dissertação, e é apresentada a seguir:

Definição 2.8 *Duas Redes de Petri:* $N = (P, T, Pre, Post, \Sigma, l, M_0)$ e $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{Pre}, \tilde{Post}, \tilde{\Sigma}, \tilde{l}, \tilde{M}_0)$ são compatíveis a partir de M_0 e \tilde{M}_0 , sendo denotado por $\langle N, M_0 \rangle \cong \langle \tilde{N}, \tilde{M}_0 \rangle$, se elas satisfazem as seguintes condições [20]:

- (i) $P = \{p_1, p_2, \dots, p_n\}$ $\tilde{P} = P \cup \{p_{n+1}\}$, $P \in \mathbb{N}^n$; $\tilde{P} \in \mathbb{N}^{n+1}$;
- (ii) $T = \{t_1, t_2, \dots, t_m\}$ e $\tilde{T} = T \cup \{t_{m+1}\}$, $T \in \mathbb{N}^m$; $\tilde{T} \in \mathbb{N}^{m+1}$;
- (iii) $l(t) = \tilde{l}(t)$ para toda transição $t \in T$;
- (iv) $\tilde{M}_0 = [M_0^T \ 0]^T$;
- (v) Existe uma transição $t_p \in T$ tal que $I(t_p) \subset P$ e $O(t_p) = I(t_{m+1}) = \{p_{n+1}\}$;
- (vi) $O(t_{m+1}) = O(t_p) = \{p_i\}$ e $p_i \in P$, ver Figura 2.6 □

Os seguintes fatos são imediatos a partir da definição 2.8:

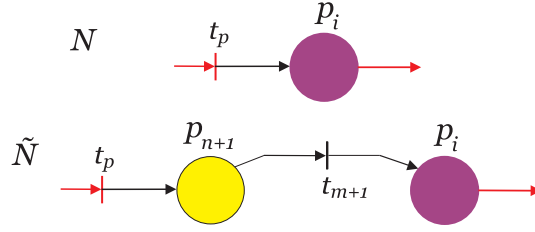


Figura 2.6: Lugares/transições em redes de Petri Compatíveis

Fato 1 : Considere duas rede de Petri N e \tilde{N} tais que $\mathcal{L}(N) = L$ e $\mathcal{L}(\tilde{N}) = \tilde{L}$ e seja $t \in T^*$ uma seqüência de transições cuja ultima transição é t_p e é tal que $l(t) = s \in L$. Existe uma seqüência $\tilde{t} = tt_{m+1} \in \tilde{L}$ que satisfaz $\tilde{l}(\tilde{t}) = l(t)\tilde{l}(t_{m+1}) = ss_{m+1}$ tal que, quando $M_0[t > M$ e $\tilde{M}_0[\tilde{t} > \tilde{M}$, então $\tilde{M} = [M^T \ 0]^T$.

Prova: A prova é imediata a partir da definição, uma vez que existem seqüências t e \tilde{t} , a partir de $M_0 = [M'_0 \mid M(p_i) \cdots M(p_j)]$; onde $\{p_i, \dots, p_j\} \in I(t_p)$, $t_p \in T \cap \tilde{T}$ e $\tilde{M}_0 = [M_0 \mid M(p_{n+1}) \cdots M(p_{n+k})]$; com $\{p_{n+1}\} \in O(t_p)$, $I(t_{m+1})$ ("|" simboliza que alguns elementos são adicionados a um dado vetor), tal que as marcações de \tilde{M} correspondentes à M são iguais e as restantes sejam zero, devido à sua construção. ■

Fato 2 : Em termos de linguagem, temos que $L = P_\Sigma(\tilde{L})$.

Prova: Fazendo uma análise das linguagens geradas $\mathcal{L}(N) = L$ e $\mathcal{L}(\tilde{N}) = \tilde{L}$, podemos notar que L e \tilde{L} não são as mesmas, mas a particularidade interessante é que uma delas pode ser calculada a partir da outra, i.e. Dada $l : T \rightarrow \Sigma$ e $\tilde{l} : \tilde{T} \rightarrow \tilde{\Sigma}$, funções de rotulação de N e \tilde{N} respectivamente, observa-se também por construção, que $\Sigma \subseteq \tilde{\Sigma}$. Aplicando a *projeção natural* de L' com relação ao conjunto de eventos Σ da forma:

$$P_\Sigma(\tilde{L}) := \{t \in \Sigma^* : \exists s \in \tilde{L}, (P_\Sigma(s) = t)\}$$

É fácil ver que a projeção definida acima em duas redes de Petri compatíveis em termos das linguagens geradas caracteriza-se por $L = P_\Sigma(\tilde{L})$. ■

Uma interpretação física para a compatibilidade de modelos em redes de Petri é que N representa uma evolução intermediária na modelagem de \tilde{N} , que não modifica o estado com relação a N , de forma tal que modela um grau de abstração

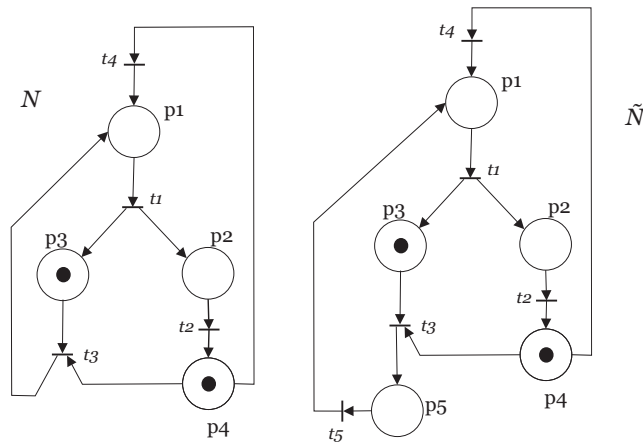


Figura 2.7: Redes N e \tilde{N} do exemplo 2.7

maior, incluindo uma maior quantidade de dados na modelagem do sistema completo. Isto, na prática representa os sensores que serão adicionados para observar o comportamento do sistema, tal como será a mostrado na seguinte seção. Devido à importância da Definição 2.8 no desenvolvimento do presente trabalho, a seguir será apresentado um exemplo de compatibilidade:

Exemplo 2.7 *Sejam N e \tilde{N} as rede de Petri representadas na Figura 2.7, onde*

$$N = (P, T, Pre, Post, l, M_0) \text{ e } \tilde{N} = (\tilde{P}, \tilde{T}, \tilde{Pre}, \tilde{Post}, \tilde{l}, \tilde{M}_0),$$

sendo

$$P = \{p_1, p_2, p_3, p_4\}, \quad \tilde{P} = \{p_1, p_2, p_3, p_4, p_5\} = P \cup \{p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}, \quad \tilde{T} = \{t_1, t_2, t_3, t_4, t_5\} = \{T, t_5\}$$

$$M_0 = [0 \ 0 \ 1 \ 1] = [M'_0 \mid 1 \ 1]$$

$$\{p_3, p_4\} \in I(t_3); t_3 \in T, \tilde{T}$$

$$\tilde{M}_0 = [0 \ 0 \ 1 \ 1 \ 0] = [M_0 \mid 0]$$

$$\{p_5\} \in O(t_3), I(t_5)$$

Quando a transição t_3 dispara, tem-se:

$$M_0[t_3 > M_1; \quad M_1 = [1 \ 0 \mid 0 \ 0]$$

$$\tilde{M}_0[t_3 > \tilde{M}_1; \quad \tilde{M}'_1 = [0 \ 0 \mid 0 \ 0 \mid 1]$$

Pode se concluir que as redes N e \tilde{N} são compatíveis ou $\langle N, M_0 \rangle \cong \langle \tilde{N}, \tilde{M}_0 \rangle$ uma vez que, se uma mesma seqüência de disparos é registrada nas redes N e \tilde{N} a

partir de M_0 e \tilde{M}_0 respectivamente, incluindo o disparo da transição t_5 presente só em \tilde{N} , obtém-se:

$$\tilde{M}'_1[t_5 > \tilde{M}_1; \text{ onde } \tilde{M}'_1 = [1 \ 0 \mid 0 \ 0 \mid 0].$$

Porém $\tilde{M}_1 = [M_1 \mid 0]$, o que significa que o estado geral do sistema não sofreu variação, já que os estados/lugares correspondentes a M_1 são os mesmos a \tilde{M}_1 e dado que $\Sigma \subseteq \tilde{\Sigma}$ então $P_{\Sigma}[\mathcal{L}(\tilde{N})] = \mathcal{L}(N)$. \blacklozenge

2.5 Mapeamento de Sensores em Redes de Petri

O diagnóstico de falhas tem um fato inerente de que quanto maior for a quantidade de informação disponível com relação ao comportamento atual do sistema (medições, variáveis, tempos de ocorrência), maior será o número de falhas que vão ser detectadas. É obvio que o registro do comportamento do sistema é capturado pelo conjunto de sensores instalados no processo. Neste sentido, na presente seção será apresentado um procedimento sistemático para acrescentar a informação proveniente dos possíveis sensores instalados no processo à modelagem do sistema geral feita em redes de Petri. Este procedimento é uma extensão a Redes de Petri, do procedimento criado por Sampath *et al.* para autômatos, representando por tanto uma das contribuições originais da presente dissertação.

Seja uma rede de Petri que modela um SED, denotada por:

$$N = (P, T, Pre, Post, \Sigma, l, M_0).$$

Dado um conjunto de sensores no sistema, definimos o mapeamento associado como $h_j : M \rightarrow Y_j$, $j = 1, \dots, S$, onde S é o número de sensores e Y_j denota o conjunto das possíveis saídas do j -ésimo sensor.

Define-se o produto cartesiano dos conjuntos Y_j como Y :

$$Y = \prod_{j=1}^S Y_j$$

como o conjunto de todas as possíveis saídas do conjunto de sensores, e seja $h : M \rightarrow Y$ o mapeamento dos sensores em cada marcação, tal que:

$$h(M) = (h_1(M), h_2(M), \dots, h_S(M)); \quad \forall M \in \mathcal{R}(N, M_0)$$

Uma vez que $l : T \rightarrow \Sigma$, e supondo sem perda de generalidade, que duas transições associadas ao mesmo evento não podem disparar simultaneamente, então, a partir desse ponto, eventos e transições são referidos de forma indistinta. A rede $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{B}, \tilde{w}, \tilde{\Sigma}, \tilde{l}, \tilde{M}_0)$ é construída a partir de N , de acordo com o seguinte procedimento, o que é uma extensão para redes de Petri do procedimento proposto por Sampath *et.al.* [27]:

Procedimento 4

Seja $\sigma \in \Sigma$ ($\tilde{\sigma} \in \tilde{\Sigma}$) e suponha que $M[\sigma > M']$ ($\tilde{M}[\tilde{\sigma} > \tilde{M}']$) e $M \in \mathcal{R}(N, M_0)$. Para cada evento σ de Σ , uma marcação $\tilde{M} \in \mathcal{R}(\tilde{N}, \tilde{M}_0)$ e um novo evento são obtidos da seguinte forma [20]:

- (i) Se σ for observável, então defina-se $\tilde{\sigma} = \langle \sigma, h(M') \rangle$, e $\tilde{M}[\langle \sigma, h(M') \rangle > \tilde{M}']$; o novo evento $\tilde{\sigma}$ é observável em $\tilde{\Sigma}$;
- (ii) Se σ for não-observável e $h(M) = h(M')$, então $\tilde{\sigma} = \sigma$ e $\tilde{M}[\sigma > \tilde{M}']$; o evento σ é não-observável em $\tilde{\Sigma}$;
- (iii) Se σ é não-observável e se $h(M) \neq h(M')$, então, crie um novo lugar p_{n+1} em \tilde{N} , e substitua σ pelos seguintes eventos:
 - (a) σ , de tal forma que $p_{n+1} \in O(\sigma)$, $\tilde{M}[\sigma > \tilde{M}''$ e
 - (b) $\langle h(M) \rightarrow h(M') \rangle$, de tal forma que $p_{n+1} \in I(\langle h(M) \rightarrow h(M') \rangle)$, $\tilde{M}''[\langle h(M) \rightarrow h(M') \rangle > \tilde{M}']$, onde \tilde{M}^i contém um lugar a mais que M^i que foi aumentado no mapeamento de sensores, e $\langle h(M) \rightarrow h(M') \rangle$ denota a variação no sensor correspondente as marcações M e M' . \diamond

Em outras palavras, o objetivo do mapeamento é incluir um novo lugar \tilde{M}'' e ter duas transições associadas a eventos diferentes, tendo-se uma transição não-observável $\langle \sigma \rangle$ seguida de uma transição observável $\langle h(M) \rightarrow h(M') \rangle$.

A nova Rede \tilde{N} , contém uma composição de eventos do seguinte tipo:

- 1) $\langle \sigma, h(M') \rangle$, observável.
- 2) $\langle h(M) \rightarrow h(M') \rangle$, observável.
- 3) $\langle \sigma \rangle$, não-observável.

A partir disso o seguinte resultado pode ser estabelecido:

Fato 3 : *As redes de Petri \tilde{N} , formada segundo o procedimento 3, e N são compatíveis a partir do seu respectivo estado inicial.*

Prova: Note que, de acordo com o procedimento 3, para cada transição N , corresponde uma transição equivalente em \tilde{N} ou é introduzido um novo lugar e uma nova transição que satisfaz as condições (v) e (vi) da definição 2.8. ■

2.6 Diagnóstico de Falhas

Recentemente Sampath *et.al.*[26], propuseram um método para o projeto de diagnosticadores de falhas baseado em eventos. Neste método, definimos eventos falha e consideramos um modelo de Sistemas a Eventos Discretos incluindo comportamentos de falha. O diagnóstico de falhas estima a ocorrência a partir de seqüências de eventos parcialmente observadas, que se reduz a um problema de estimação da ocorrência de um evento específico. Sampath *et.al.* [26],[27],[28] supõem que o Sistema a Eventos Discretos tem sua modelagem feita por um autômato de estados finitos, tanto no modelo do sistema como no projeto do diagnosticador. Com a finalidade de estender o método de Sampath a Redes de Petri com um infinito número de marcações alcançáveis, é preciso uma aproximação finita de um conjunto de alcançabilidade infinita [31].

É importante notar que no desenvolvimento do trabalho, uma hipótese importante é que os lugares e as transições das redes de Petri podem ser observáveis e não observáveis. Contraste-se esta hipótese com aquela que foi feita por Wen e Yen [32], a saber, todos os lugares são observáveis, porém as transições não observáveis. Nota-se também que esta hipótese está mais de acordo com a realidade do processo, uma vez que dependendo da natureza e das ferramentas para medir as grandezas, são os eventos, alguns observáveis e outros não, aqueles que geram lugares e estados.

O estudo que será feito nesta seção é uma das contribuições do presente trabalho, logo que baixo a premissa que são os eventos os que serão registráveis (observáveis), precisa-se da adequação da teoria de diagnóstico de falhas que foi proposta em sistemas modelados por autômatos, para sistemas modelados por redes de Petri

com um enfoque de eventos observados, sendo diferente da que foi feita por Ushio *et al.* [31] e Wen [32] que consideraram os lugares como sendo observáveis.

Definição 2.9 (*Diagnosticador*)

É uma máquina de estados finitos que tem como característica principal, inferir o comportamento do sistema a partir dos eventos observados on-line. A ocorrência de falhas pode ser detectada pelo análise dos seus estados, onde o diagnosticador contém informação sobre a ocorrência de falhas [26]. \square

Seja o sistema a diagnosticar, modelado pelo *SED* definido pela seguinte Rede de Petri:

$$N = (P, T, A, w, \Sigma, l, M_0);$$

Com:

$$\mathcal{L}(N) = L \subseteq \Sigma^*;$$

Alguns eventos em Σ são observáveis, i.e. a sua ocorrência pode ser observada, e os demais eventos são não-observáveis.

$$\Sigma = \Sigma_o \cup \Sigma_{uo}; \quad \Sigma_f \subseteq \Sigma_{uo}$$

Onde:

- Σ : Conjunto de todos os eventos.
- Σ_o : Conjunto dos eventos observáveis, (por exemplo. Comandos provenientes do controlador, mudança na leitura de sensores)
- Σ_{uo} : Conjunto dos eventos não-observáveis, (por exemplo. Eventos que mudam o estado mas não são registrados, falhas)
- $\Sigma_f \subseteq \Sigma_{uo}$: Conjunto dos eventos falha que são particionados em subconjuntos disjuntos Σ_{f_i} , $i = 1, 2, \dots, n$ com $i \neq j$, i.e. $\Sigma_{f_i} \cap \Sigma_{f_j} = \emptyset$ [27], e as falhas de diferentes tipos:

$$\Sigma_f = \Sigma_{f_1} \cup \Sigma_{f_2} \cup \dots \cup \Sigma_{f_m} \tag{2.16}$$

- $l : T \rightarrow \Sigma \cup \{\varepsilon\}$, função de rotulação que associa a cada transição um evento.

Supomos sem perda de generalidade que o nosso objetivo é identificar a ocorrência de eventos falha, posto que nas seqüências geradas pelos sistemas, apenas os eventos de Σ_o são observados. A partição (2.16) é necessária devido aos diferentes tipos de falhas que podem ocorrer no sistema, seja na instrumentação, nos atuadores, controladores ou perturbações, e também pelo fato de simplificar o diagnóstico de falhas. Cada partição de (2.16) é associada a uma falha de tipo F_i , para $i = 1, 2, \dots, m$. O simples conhecimento ou detecção da ocorrência de ao menos um evento pertencente à partição Σ_{f_i} , levaria a inferência da ocorrência de uma falha do tipo F_i ; posto que todos os eventos de Σ_{f_i} provocam o mesmo efeito no sistema. A modelagem de falhas em Sistemas a Eventos Discretos é realizado usando a premissa de que uma falha do tipo F_i ocorreu se algum dos eventos $\sigma_j \in \Sigma_{f_i}$ ocorreu.

Exemplo 2.8 *Considere o seguinte conjunto de eventos:*

$$\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7\}$$

sendo:

$$\Sigma_o = \{\sigma_1, \sigma_5\} \text{ e}$$

$$\Sigma_{uo} = \Sigma_f = \{\sigma_2, \sigma_3, \sigma_4, \sigma_6, \sigma_7\}$$

Definimos falhas dos tipos F_1 e F_2 , a cada tipo de falha associamos uma partição de eventos $\Sigma_{f_1} = \{\sigma_2, \sigma_6, \sigma_7\}$ e $\Sigma_{f_2} = \{\sigma_3, \sigma_4\}$ respectivamente. Então, ao detectar qualquer evento pertencente a Σ_{f_1} diagnosticamos a ocorrência de uma falha do tipo F_1 e ante a detecção de qualquer evento de Σ_{f_2} diagnosticamos a ocorrência de uma falha do tipo F_2 . \blacklozenge

Denotaremos por L/s ao pós-linguagem de L após de s , i.e.

$$L/s = \{t \in \Sigma^* \mid st \in L\} \quad (2.17)$$

Devido a existência de uma grande diversidade e complexidade de modelos, vamos assumir que o modelo a diagnosticar terá as seguintes características:

A.1 N é *viva*, i.e., existem seqüências de disparo a partir de M_0 tais que todas as transições podem disparar ao menos uma vez.

A.2 Não existem ciclos de marcações cujas transições intermédias observáveis sejam as mesmas, i.e. não existe seqüências infinitas de transições não-observáveis. $(\forall M_i \in \mathcal{R}(N, M_0), t_i \in T, i = 1, 2, \dots, n)$

$$M_1[t_1 > M_2[t_2 > \dots > M_n[t_n > M_1 \Rightarrow \nexists i, j \in \{1, 2, \dots, n\} \mid M_i \equiv M_j$$

Um *SED* modelado por uma rede de Petri, que gera uma linguagem L satisfazendo as assunções A.1 e A.2 é dito diagnosticável se e somente se é possível inferir a ocorrência de uma falha $f_i, \forall i \in \Sigma_f$ após um atraso finito, i.e., em um número finito de disparo transições após da ocorrência da falha f_i , [26], a definição formal de diagnosticabilidade será apresentada a seguir.

Uma linguagem L é diagnosticável se é possível detectar, com um atraso finito, a ocorrência de falhas de qualquer tipo usando o registro de eventos observados. Agora apresentaremos a definição formal de diagnosticabilidade que será importante no desenvolvimento do presente trabalho.

Definição 2.10 (*Diagnosticabilidade*)

A linguagem L é dito diagnosticável com respeito a projeção $P : \Sigma^* \rightarrow \Sigma_o^*, \Sigma_{f_i}$ se somente se:

$$(\forall i \in \Sigma_f)(\exists n_i \in \mathbb{N})[\forall s \in \Sigma_{f_i}](\forall t \in L/s)[\|t\| \geq n_i \Rightarrow D]$$

Onde a condição de diagnosticabilidade D é dada por:

$$w \in P_L^{-1}[P(st)] \Rightarrow \Sigma_{f_i} \in w \quad (2.18)$$

$\|t\|$: denota comprimento da seqüência t .

s : seqüência que termina num evento falha.

t : continuação de s . (*Observável*) □

A definição de diagnosticabilidade dada em [26] significa que, se uma seqüência s - gerada pelo sistema - for finalizada em um evento falha do conjunto Σ_{f_i} e t qualquer subsequência que seja continuação de s , então a condição de diagnosticabilidade D requer que todas as seqüências em L que produzem o mesmo registro de eventos observáveis, como a seqüência st deveria conter nela um evento falha do conjunto Σ_{f_i} . Isto implica que em todas as continuações t de s , podemos

detectar a ocorrência de um evento falha do tipo F_i com um atraso finito, mais especificamente em um máximo de n_i transições do sistema após a sub-sequência s . Diagnosticabilidade requer que cada evento falha conduza a observações distintas de tal maneira a habilitar uma única identificação de evento falha com um atraso finito. Quando mais de uma falha do mesmo tipo F_i ocorre na sequência s , a definição de diagnosticabilidade não requer que cada uma de essas ocorrências seja detectada.

Note que a definição 2.10 pode ser aplicada indistintamente a autômatos e redes de Petri, já que ambas modelagens são modelos formais para representar uma linguagem gerada e marcada por sistemas a eventos discretos.

2.6.1 O Diagnosticador

O diagnosticador G_d é uma Máquina de Estados Finitos (autômato), construído a partir de N , representado pela quintupla:

$$G_d = (Q_d, \Sigma_o, \delta_d, \Gamma, q_0)$$

Sendo

Q_d , é o conjunto de marcações alcançáveis, definido formalmente na próxima subseção.

Σ_o , conjunto de eventos observáveis.

δ_d , função de transição do diagnosticador.

Γ_d , conjunto de eventos ativos para um dado estado.

q_0 , estado inicial do diagnosticador, $q_0 = \{(M_0, N_b)\}$.

Este autômato ou máquina de estados, tem por objetivo fundamental o desenvolvimento do diagnóstico quando observa o comportamento *on-line* de N . O diagnosticador é usado também para estabelecer condições necessárias e suficientes da diagnosticabilidade de um sistema modelado por redes de Petri. Formularemos um diagnosticador fundamental, que é um extensão do método proposto por Sampath *et al.*, e também apresentaremos algumas contribuições do presente trabalho sobre a proposta de Ushio *et al.*

2.6.2 Estrutura do Diagnosticador

Os estados do diagnosticador consistem de marcações e rótulos, os rótulos podem ser N_b ou F_i , onde:

N_b : Significa comportamento *Normal*.

F_i : Significa que a *Falha do Tipo* F_i ocorreu.

E os estados são da forma $q_d = \{(M_1, l_1), \dots, (M_n, l_n)\}$. Inicialmente definimos o conjunto de rótulos falha: $\Delta_f = \{F_1, F_2, \dots, F_m\}$, onde $|\Sigma_f| = m$ e definimos o conjunto de rótulos falha do diagnosticador como segue [25]:

$$\Delta = l' = \{N_b\} \cup 2^{\Delta_f}, \text{ com} \quad (2.19)$$

O estado inicial para um *SED* modelado pela rede de Petri $\langle N, M_0 \rangle$ é $q_0 = \{(M_0, N_b)\}$, e conjunto de marcações alcançáveis Q_d . Consideremos Q_d como o conjunto das marcações alcançáveis, tal que $Q_d \subseteq 2^{\mathcal{R}(M_0 \times \Delta)}$, cujos elementos são conjuntos finitos formados pela Marcação e seu respectivo rótulo, cada elemento de Q_d é da forma: $q_d = \{(M_1, l_1), \dots, (M_n, l_n)\}$; onde $M_i \in \mathcal{R}(N, M_0)$ e $l_i \in \Delta$ tem a forma $l_i = \{N_b\}$ ou $l_i = \{F_{i1}, F_{i2}, \dots, F_{ik}\}$

Isto é:

$$Q_d = \{q \in Q; \exists M \in \mathcal{R}(N, M_0), \delta_d(q_0, M) = q\} \quad (2.20)$$

O diagnosticador consiste em um autômato de estados finitos, onde é preciso se definir a sua função de transição. Com a finalidade de caracterizar a função de transição δ_d , é necessário introduzir três funções: A função de propagação de rotulo falha, a função domínio e a função de correção de rótulo.

Definição 2.11 : *Seja* $M \in \mathcal{R}(M_0), l \in \Delta$ e $s \in \mathcal{L}_\sigma(N, M_0)$ [20].

1. A função Propagação de Rotulo Falha (*LP*, Label Propagation), propaga o rótulo l sobre s , iniciando em M e seguindo a dinâmica do sistema modelado pela Rede de Petri $N, M \in \mathcal{R}(N, M_0)$, i.e., $(M, l, s) \rightarrow l'$, o novo rótulo l' será definido da seguinte forma:

$$LP : \mathcal{R}(N, M_0) \times \Delta \times T^* \rightarrow \Delta$$

$$LP(M, l, s) = l' = \begin{cases} \{N_b\} & \text{se } l = \{N_b\} \wedge \Sigma_{f_i} \notin s, \forall i \\ \{F_i\} & \text{se } F_i \in l \vee \Sigma_{f_i} \in s \\ \{F_i, F_j\} & \text{se } F_i \in l \wedge \Sigma_{f_j} \in s \end{cases} .$$

2. A Função Domínio (RA , Range), obtém todas as possíveis pares de marcações (M_i, l_i) que aparecem nos estados do diagnosticador. RA é definida da seguinte forma:

$$RA: Q_o \times \Sigma_o \rightarrow Q_d$$

$$RA(q, \sigma) = \bigcup_{(M, \ell) \in q} \bigcup_{s \in \mathcal{L}_\sigma(N, M_0)} \{(M[s \succ, LP(M, \ell, s)]\}.$$

3. A função de Correção de Rotulo (LC , Label Correction), é definida como:

$$LC : Q_o \rightarrow Q_o$$

$$LC(q) = \{(M, l) \in q : M \text{ aparece só uma vez em } q\} \cup \{(M, \bigcup l_{i1} \cap \dots \cap l_{ik})$$

sempre que existam pares $(M, l_{i1}), \dots, (M, l_{ik})$ em $q\}$. \square

O rótulo adquirido pela marcação M após da seqüência s indica a ocorrência ou não de um evento falha, dependendo do tipo de rotulação. Suponha que existem dois pares $(M, l), (M, l') \in RA(q, \sigma)$ para algum q do diagnosticador. Então isto implica que a marcação M só deverá aparecer uma vez no estado do diagnosticador, onde o rótulo resultante será a sua união, i.e. $(M, l), (M, l') \rightarrow (M, l, l')$, sendo a representação da informação necessária para o diagnóstico de falhas no sistema.

A partir a definição 2.11, vamos definir a função de transição do diagnosticador.

Definição 2.12 : Seja o autômato diagnosticador $G_d = (Q_d, \Sigma_o, \delta_d, \Gamma, q_0)$, com $\sigma \in \Sigma_o$, $q_1, q_2 \in Q_d$, a função de transição de G_d é definida como:

$$\delta_d : Q_d \times \Sigma_o \rightarrow Q_d$$

$$q_2 = \delta_d(q_1, \sigma) \Leftrightarrow q_2 = RA(q_1, \sigma) , \quad (2.21)$$

\square

onde $\delta_d(q_1, M_d)$ é indeterminado se $RA(q_1, M_d) = \phi$, o que é natural já que uma das características do diagnosticador é que todos os seus estados são acessíveis partir de q_o , o estado que contém as marcação o marcações iniciais.

Usando as definições dos estado do diagnosticador (Equação 2.20) e função de transição (Definição 2.12), na seguinte seção será dado um procedimento para a construção de diagnosticadores.

2.6.3 Procedimento de construção de diagnosticadores

Nesta seção se apresenta um algoritmo recursivo para a construção de diagnosticadores de redes de Petri, que é uma extensão do algoritmo criado por Sampath

et al.[26] - em sistemas modelados por máquinas de estados finitos - para sistemas modelados por Redes de Petri. O procedimento dado nesta seção também apresenta algumas modificações em relação aquele dado em Ushio *et al.*[31]. Estas modificações centram-se no enfoque baseado em eventos observados, que tem diferença de [31], feito em marcações observadas. Posto que as redes de Petri são uma ferramenta com uma modelagem algébrica bem definida, então é possível obter a recursividade computacional, tanto para o cálculo das funções de propagação de etiqueta, função domínio e sua respectiva função de correção de rotulação, assim obter a função de transição do diagnosticador para o posterior análise da sua diagnosticabilidade. O diagnosticador G_d é construído como segue:

Procedimento 5 (Construção de diagnosticador) :

Começar o algoritmo definindo o estado inicial do diagnosticador como $q_0 = \{(M_0, N_b)\}$

A dinâmica no diagrama de transição de estados de G_d evolui segundo: Seja q o estado atual de G_d , e seja $s = u\sigma$, onde $u \in \Sigma_{uo}^$ e $\sigma \in \Sigma_o$. O próximo estado q' do diagnosticador é calculado por:*

PASSO 1 *Para todo $(M, l) \in q$, obter $S(M, \sigma)$ a alcançabilidade de M com relação ao evento observado σ , (Seção 2.15).*

PASSO 2 *Seja $(M', l') \in S(M, \sigma)$. Propagar o rótulo l em l' , como segue:*

- a) Se $l = \{N_b\}$ e s não contém evento falha $\Rightarrow l' = \{N_b\}$*
- c) Se $l = \{F_i\}$ e s não contém evento falha $\Rightarrow l' = \{F_i\}$*
- d) Se $l = \{N_b\}$ e s contém evento falha $\Sigma_{fi}, \Sigma_{fj} \Rightarrow l' = \{F_i, F_j\}$*
- e) Se $l = \{F_i, F_j\}$ e s contém evento falha de $\Sigma_{fk} \Rightarrow l' = \{F_i, F_j, F_k\}$*

PASSO 3 *Seja q_2 o conjunto de todos os (M', l') , substituir por (M', F_i, F_j) todos os $(M', l')(M', l'') \in q_2$ tal que F_i e F_j são componentes de l' e l'' , em outras palavras se a mesma marcação M' aparece mais uma vez em q_2 com diferentes rótulos falha, então associamos a M' todas as componentes comuns de l' e l'' segundo a função de correção de etiqueta. \diamond*

Iterar o procedimento para todo $\sigma \in \Gamma(q_i) \mid q_i \in RA(q, \sigma)$ até que as trajetórias do diagrama de transição de estados tenham uma alcançabilidade finita, isto é $\Gamma(q_i) = \emptyset$ ou que entre em uma trajetória fechada, tal que $\exists s' \mid q_i \in RA(q_i, s')$.

2.6.3.1 Propriedades do diagnosticador

Estabeleceremos algumas das propriedades do diagnosticador que são imediatas do processo de construção descrito no procedimento 5. Estas propriedades serão usadas subsequente para estabelecer as condições de diagnosticabilidade [26].

P.1 Existe $q \in Q_d$ tal que $(M, l), (M, l') \in q$ com $l \neq l'$.

P.2 Seja $q \in Q_d$, então: $(M, l), (M, l') \in q \Leftrightarrow \exists s_1, s_2 \in L$ tal que os últimos eventos da seqüência s_1 e s_2 são observáveis, $M_0[s_1 > M_1, M_0[s_2 > M_2$ e $P(s_1) = P(s_2)$.

P.3 Seja $q_1, q_2 \in Q_d$ e $s \in \Sigma^*$, tal que $(M_1, l_1) \in q_1, (M_2, l_2) \in q_2, M_1[s > M_2$ e $\delta_d(q_1, P(s)) = q_2$ se:

$$F_i \notin l_2 \Rightarrow F_i \notin l_1$$

$$N_b \in l_2 \Rightarrow N_b \in l_1$$

2.6.4 Condições necessárias e suficientes para diagnosticabilidade

Dadas as propriedades *P.1* e *P.2*, os estados do diagnosticador ($q \in Q_d$), podem ser definidos como F_i -certos e F_i -incertos, dependendo da presença ou não de diferentes rótulos falha.

Definição 2.13 : Um estado q do diagnosticador G_d é dito [26]:

- i)* F_i – certo : se $\forall (M, l) \in q, F_i \in l$, o diagnosticador neste estado é capaz de detectar com certeza a ocorrência de uma falta do tipo F_i , i.e. $\forall w \in P_L^{-1}(u), \Sigma_{F_i} \in w$
- ii)* F_i – incerto : se $\exists (M, l)(M', l') \in q$ tal que $F_i \in l, F_i \notin l'$, o diagnosticador neste estado não é capaz de detectar a ocorrência de uma falha do tipo F_i , i.e. $\exists s_1, s_2 \in L \mid \Sigma_{F_i} \notin s_2, P(s_1) = P(s_2), \delta_d(q_0, P(s_1)) = q$. \square

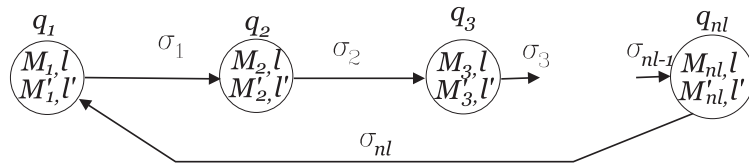


Figura 2.8: Ciclo indeterminado com $P(s_1) = P(s_2) = \sigma_1\sigma_2 \cdots \sigma_{nl}$

Da definição de estado F_i -certo, podemos concluir que, neste estado do diagnosticador, a falta F_i ocorreu, se para uma seqüência de eventos $s \in \mathcal{L}(N)$ tal que $\delta_d(q_0, s) = q$, então $\forall \omega \in P_L^{-1}(s)$, $\Sigma_{F_i} \in \omega$. Este tipo de estado será importante no estudo de diagnóstico baseado em Sistemas a Eventos Discretos, seja pela aplicação em Redes de Petri como em máquinas de estados finitos (Autômatos). A presença de estados F_i -incertos em G_d corresponde a situação de duas seqüências $s_1, s_2 \in L$ geradas pela rede N tal que s_1 contém eventos falha do tipo F_i ($\Sigma_{f_i} \in s_1$) entanto s_2 não contém ($\Sigma_{f_i} \notin s_2$), ademais as seqüências s_1 e s_2 produzem o mesmo registro de eventos observados, i.e. satisfazendo $P(s_1) = P(s_2)$ e $\delta_d[q_0, P(s_1)] = q$. Sempre que a dinâmica do diagnosticador chegue num estado F_i -incerto, concluímos que a falha do tipo F_i pode haver ocorrido mas não é possível afirmar com certeza - a partir da seqüência de eventos observados até esse ponto - se a falha ocorreu realmente.

A análise dos estados do diagnosticador é importante no estudo da diagnosticabilidade dos sistemas e para dar as condições necessárias e suficientes para que linguagem L gerada pela rede de Petri N seja diagnosticável.

Quando um grupo de estados F_i -incertos forma um uma trajetória fechada, então eles podem formar um ciclo F_i -indeterminado. Um ciclo F_i -indeterminado é um ciclo composto exclusivamente de estados F_i -incertos nos quais existem pelo menos duas seqüências $s_1 \in \Sigma^*$ em M_i^k e $s_2 \in \Sigma^*$ em \tilde{M}_i^k que têm a mesma projeção (ver figura 2.8), porém na seqüência s_1 a falha F_i não ocorreu, enquanto que na seqüência s_2 a falha ocorreu [26].

Uma definição formal de ciclos indeterminados e trajetórias indeterminadas são dadas a seguir:

Definição 2.14 (*Ciclos F_i -indeterminados*)

O conjunto de estados F_i -incertos $q_1, q_2, \dots, q_n \in Q_d$ é dito que forma um ciclo F_i – indeterminado [26],[27] se:

1. q_1, q_2, \dots, q_{n_l} formam um ciclo em G_d i.e., $\delta_d(q_l, \sigma_l) = q_{l+1}$, $l = 1, 2, \dots, n_l - 1$ e $\delta_d(q_{n_l}, \sigma_{n_l}) = q_1$, onde $\sigma_l \in \Sigma_o$, $l = 1, 2, \dots, n_l$;
2. Existe $(M_l^k, \ell_l^k), (\tilde{M}_l^r, \tilde{\ell}_l^r) \in q_l$, $l = 1, 2, \dots, n_l$, $k = 1, 2, \dots, n_k$ e $r = 1, 2, \dots, n_r$, tal que:

a) $F_i \in \ell_l^k, F_i \notin \tilde{\ell}_l^r$; $\forall l, k, r$

b) A seqüência de marcações M_l^k e \tilde{M}_l^k satisfazem :

- $M_l^k[\sigma_l > M_{l+1}^k]$, $l = 1, 2, \dots, n_l$ e $k = 1, 2, \dots, n_k$,
 $M_{n_l}^k[\sigma_{n_l} > M_1^k]$, $k = 1, 2, \dots, n_k - 1$, e
 $M_{n_l}^{n_k}[\sigma_{n_l} > M_1^1]$.
- $\tilde{M}_l^r[\sigma_l > \tilde{M}_{l+1}^r]$, $l = 1, 2, \dots, n_l$ e $r = 1, 2, \dots, n_r$,
 $\tilde{M}_{n_l}^r[\sigma_{n_l} > \tilde{M}_1^r]$, $r = 1, 2, \dots, n_r - 1$, e
 $\tilde{M}_{n_l}^{n_r}[\sigma_{n_l} > \tilde{M}_1^1]$. □

Definição 2.15 (Trajetórias F_i -indeterminadas)

A trajetória infinita de estados F_i -indeterminados $r = q_1, q_2, \dots, q_n \in Q_d$ forma uma Trajetória F_i -indeterminada se [31]:

- 1) Existe uma seqüência de marcações observadas M_1, M_2, \dots, M_n tal que:

$$\delta_d(q_j, M_j) = q_{j+1} \quad j = 1, 2, \dots$$

- 2) $\exists (M_j, l_j), (\tilde{M}_j, \tilde{l}_j) \in q_j$ que satisfaçam as seguintes duas condições:

a) $F_i \in l_j, F_i \notin \tilde{l}_j$; $\forall i, j$

b) $M_j[M_j + 1 > M_j + 2]$

$\tilde{M}_j[\tilde{M}_j + 1 > \tilde{M}_j + 2]$ □

Note que qualquer ciclo F_i -indeterminado é considerado como uma trajetória F_i -indeterminada e que qualquer trajetória F_i -indeterminada é reduzida a um ciclo F_i -indeterminado se N tem um conjunto alcançável finito, [31].

Baseados na definição de ciclos F_i -indeterminados e trajetórias F_i -indeterminadas, uma condição necessária e suficiente para diagnosticabilidade pode ser estabelecida.

Teorema 1 (Teorema de diagnosticabilidade [26], [31]): *A linguagem L gerada pela rede de Petri N é diagnosticável se e somente se o diagnosticador G_d de N , não contém ciclos F_i -indeterminados nem trajetórias F_i -indeterminadas, $\forall F_i, i \in \Sigma_f$.*

Prova: A prova é a mesma que foi dada em [26], dado que as redes de Petri e autômatos geram uma linguagem, e dependendo dessa linguagem é que pode-se inferir a ocorrência ou não de uma falha. ■

2.7 Metodologia do Projeto de Sistemas de Detecção de Falhas

O sucesso dos sistemas de diagnóstico de falhas baseia-se na quantidade de dados que se pode obter do processo. Esses dados são capturados por sensores instalados convenientemente para leituras das variáveis do processo. Nos sistemas industriais atuais, considerações econômicas e tecnológicas impõem a necessidade de desenvolver esquemas de diagnóstico confiáveis e seguros, com um custo mínimo na implementação de sensores, particularmente quando todas as variáveis não são acessíveis para medição. Neste sentido, capturar a maior quantidade de informação possível de um sensor real dá origem à obtenção dos denominados *sensores virtuais* [28], desenvolvendo uma aproximação de *diagnóstico híbrido* com característica qualitativa de Sistemas a Eventos Discretos e um nível abstrato da característica quantitativa dos sensores virtuais.

Os *Sensores virtuais* são usados para complementar a informação dada pelos sensores reais, tais informações sendo derivadas por um procedimento analítico. O objetivo dos sensores virtuais é disponibilizar o diagnóstico de falhas com um mínimo de requerimento de sensores, explorando a informação contida nos sinais disponíveis, abrangendo diversas técnicas analíticas. No caso mais simples, o sensor virtual poderia ser um analisador de pontos, que faz uma checagem de algum valor limite ou referencial em relação ao sinal de um sensor real. Alternativamente, o sensor virtual pode ser baseado em técnicas de processamento de sinal, tais como análise de espectro, análises de sinal, vibração, análise de ruído, técnicas de processamento de sinais, técnicas estatísticas como média ou variância.

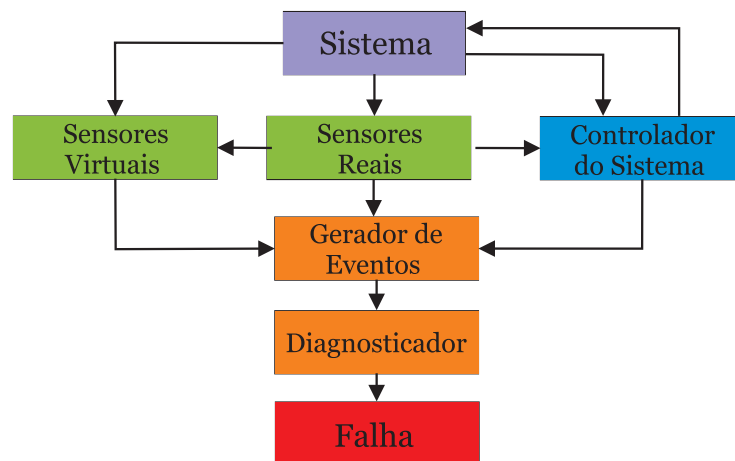


Figura 2.9: Arquitetura de um Sistema de Diagnóstico

2.7.1 Arquitetura do sistema de Diagnóstico

Para detalhar a metodologia proposta, a figura 2.9 mostra a arquitetura básica do sistema de diagnóstico. Inicialmente o comportamento do *sistema* inter-atua com o módulo *controlador* e as suas variáveis são medidas mediante os módulos de *sensores reais* e *sensores virtuais*. Todos estes dados são enviados ao gerador de eventos que utiliza as características qualitativas de todos os dados coletados. Em outras palavras o *módulo gerador de eventos* recebe comandos de controle provenientes do controlador, sinais dos sensores reais e dos sensores virtuais para converter os sinais em eventos reconhecíveis para o *módulo diagnosticador*. O módulo diagnosticador é o coração do sistema de diagnóstico que contém o motor de inferência, baseado na função de transição de estados do autômato diagnosticador e na inspeção dos seus estados. Primeiramente o diagnosticador tem como função determinar o estado do sistema e suas componentes. O estado da informação dada pelo módulo diagnosticador indica se o estado do sistema é normal, se alguma falha ocorreu ou se alguma falha é suspeita.

2.7.2 Procedimento de Diagnóstico

O projeto de um sistema de diagnóstico de falhas pode se realizar em um procedimento que consiste em uma seqüência de passos, este processo é esquematizado na figura 2.10 e detalhado a seguir [20]:

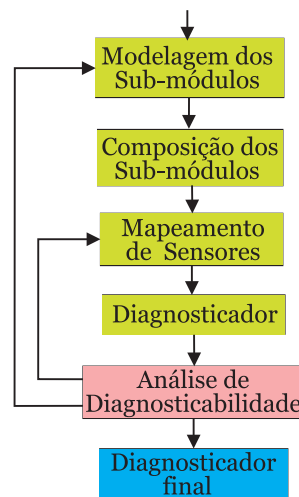


Figura 2.10: Procedimento de construção do Sistemas de Diagnóstico

1. Modelar cada componente do sistema, atuadores e controladores usando redes de Petri com lugares iniciais. Se algum dos modelos em redes de Petri não contém lugares iniciais, aplicar o procedimento 1 (Seção 2.3.5), para converter a rede na equivalente com lugares iniciais.
2. Fazer a composição paralela de todos os componentes modelados individualmente obtidos no passo 1, para gerar a modelagem completo (comportamento síncrono) do sistema.
3. Fazer o mapeamento de todos os sensores presentes no modelo completo, segundo seção 2.5. Isto implica uma prévia análise dos sensores disponíveis e a possível implementação de sensores virtuais.
4. Construir o diagnosticador segundo o procedimento 5 da seção 2.6.3.
5. Analisar a diagnosticabilidade do sistema de acordo ao teorema 1. Caso o análise preliminar conclua que o sistema é não-diagnosticável para alguma falha do tipo F_i , então retornar ao passo 1, e modificar a modelagem dos módulos ou ao incrementar o número de sensores reais e/ou virtuais. Caso contrário conclua-se que o sistema é não-diagnosticável.

2.8 Comentários e contribuições preliminares

Neste capítulo, definiu-se a teoria necessária para o projeto de sistemas automáticos de diagnóstico de falhas em sistemas modelados com Eventos Discretos utilizando uma abordagem por redes de Petri. Foi descrita a necessidade de procedimentos de modelagem em redes de Petri, que a comparação com autômatos, modela comportamentos de sistemas operando conjuntamente com um custo computacional baixo. Deste capítulo destaque-se as contribuições teóricas do presente trabalho:

- Estudou-se o projeto de diagnosticadores de falhas para sistemas modelados por redes de Petri.
- Se deram procedimentos para a obtenção de modelos partindo de submodelos, por intermédio de operações como a composição paralela de modelos em redes de Petri
- Definiu-se compatibilidade de modelos, importante para a comparação de modelos baixo certas restrições.
- Estendeu-se a teoria existente de mapeamento sensores em autômatos para modelos em redes de Petri, se deu um procedimento para mapeamento de sensores e a sua interpretação em termos de linguagens geradas.
- Integrou-se o estudo de modelagem e diagnosticabilidade. para ter uma visão completa do problema de diagnóstico de falhas de modo geral.
- Propôs-se um procedimento sistemático para o projeto de sistemas de diagnóstico de falhas baseado nos procedimentos iterativos dados e a análise de diagnosticabilidade.

Capítulo 3

O Processo de Separação Óleo-Gás

3.1 Introdução

Recentemente o Brasil alcançou a auto-suficiência na produção de Petróleo para o seu consumo interno, que implica que o Brasil deve gerar uma produção superior à demanda a longo prazo. Este crescimento, até chegar à auto-suficiência, caracterizou-se pelos elevados investimentos em avanços tecnológicos e recordes de perfuração em águas profundas que incrementaram as suas reservas *offshore*, além do aperfeiçoamento de diversas atividades da Companhia Nacional Petrobrás.

O presente trabalho tem como um dos objetivos propor o desenvolvimento de um sistema automático de diagnóstico de falhas aplicado ao processo de Separação Óleo-Gás (Separação bifásica) em uma plataforma típica de produção de petróleo, a fim de obter um sistema de isolamento de possíveis falhas e garantir um processo de produção seguro e confiável, de modo a não causar danos, acidentes nem perdas econômicas. A descrição que será apresentada é um subsistema da plataforma completa, posto que toda a planta é bastante complexa. Esta descrição baseia-se no projeto real de uma plataforma de produção da Petrobrás, que encontra-se em operação na Bacia de Campos (Rio de Janeiro) [30], [7].

O petróleo extraído dos poços marítimos consiste de uma mistura de óleo, gás e outras impurezas como água e areia. Esta mistura é tratada em plataformas de produção *offshore* para que o óleo e gás possam ser separados, o referido processo precisa da implementação de uma planta complexa, sendo também necessária a implementação de um sistema de monitoramento, sistema de proteção automática e um sistema de supervisão com diagnóstico de falhas [16].

3.2 Descrição do Processo

O processo a ser estudado é mostrado na Fig.3.1, a nomenclatura utilizada segue a norma ISA e representa um P&ID (*Piping and Instrumentation Diagram*) ver Apêndice A [15], sendo esta ilustração esquemática uma simplificação do processo real. O processo inicia-se no subsistema de aquecimento de óleo cujo objetivo é elevar a temperatura de óleo até um valor predeterminado, em seguida o óleo é bombeado para o primeiro vaso separador onde ocorre a extração de uma fração do gás contido bem como a filtragem das impurezas. Após disso, a mistura é enviada ao segundo vaso separador onde procede-se a completar o processo de separação, para o posterior transporte do gás e óleo para terra através de dutos e navios-tanque.

A separação ocorre nos vasos, a mistura permanece nos vasos por um tempo determinado, durante o qual as bolhas de gás desprendem-se até atingir a superfície, sobre as forças de empuxo, uma vez que encontram-se em um líquido de maior densidade. O processo de separação é feito em duas etapas. Na primeira etapa, os vasos separadores operam sob uma pressão alta, pois é necessário manter uma contra-pressão nas linhas provenientes dos poços, para obter um fluxo desejado. Na segunda etapa, ocorre em uma pressão mais baixa, tipicamente chamada de separação atmosférica, onde se completa o processo de separação óleo-gás [30], [14].

A Fig. 3.1, representa apenas um dos trens de separação. A contribuição dos outros poços foi considerada através do fluxo de "*Óleo e Gás de Outros Separadores*" mostrado na Fig. 3.1. Mais especificamente o processo de separação óleo-gás pode ser dividido em cinco subsistemas:

3.2.1 Subsistema de Produção

Este subsistema compreende a entrada de petróleo proveniente de diversos poços de extração, representados na Fig.[3.1] como *Óleo e Gás dos Poços*. No início do processo, entrada de óleo proveniente dos poços de produção, encontra-se uma válvula de *Shut-down* SDV-01, a qual é fechada caso haja alguma anormalidade que exija a interrupção de todo ou parte do processo. A válvula de Shut-down é acionada devido à atuação dos sensores PSH-01, PSL-01 (alarme de pressão alta e baixa),

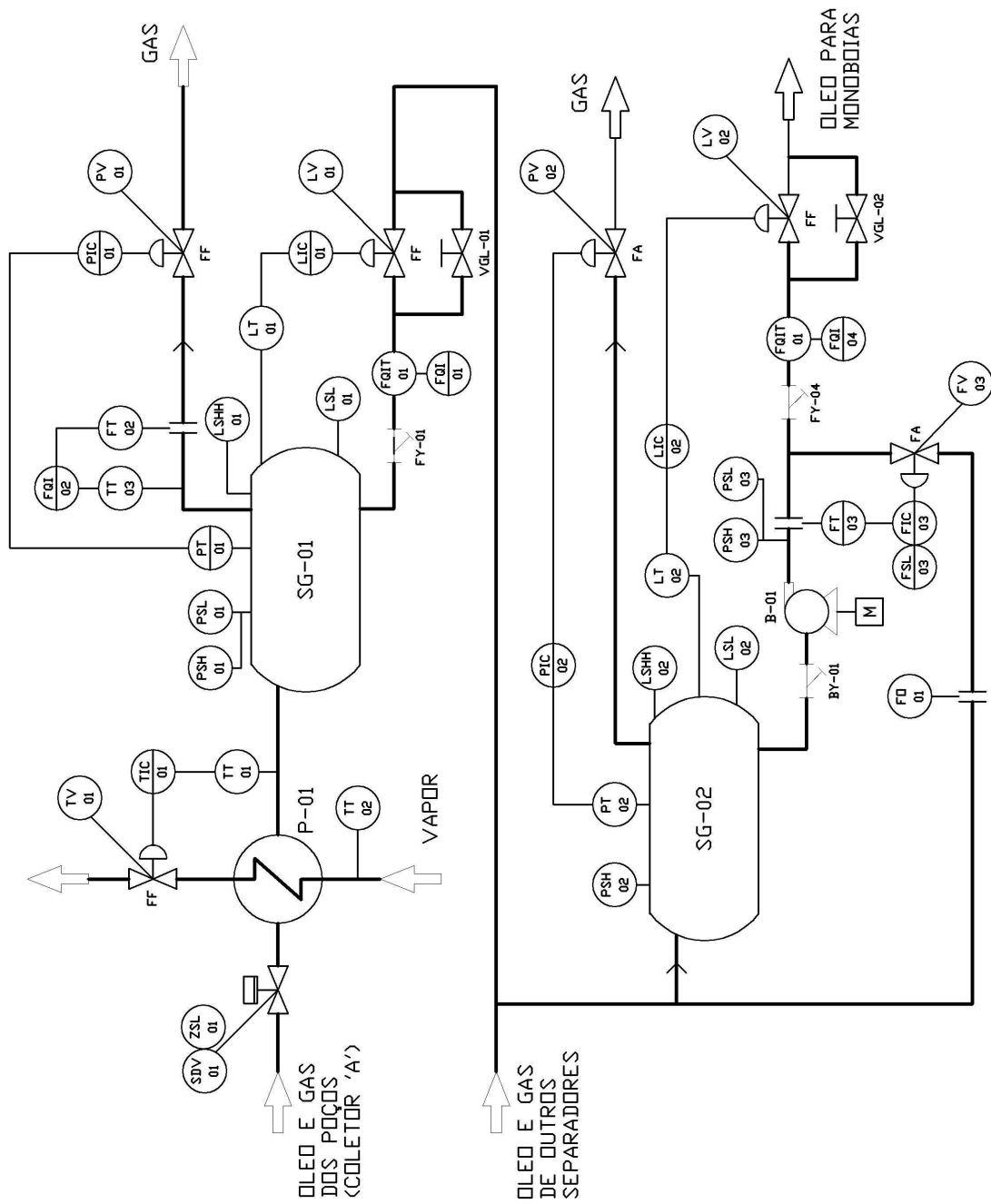


Figura 3.1: Fluxograma do Processo de Separação Óleo-Gás

sensores LSHH-01, LSL-01, LSH-01 (alarmes de nível muito alto, alto e baixo) que estão no vaso separador SG-01, ou também pela interrupção do funcionamento da bomba de exportação B-01.

3.2.2 Subsistema de Aquecimento

O óleo que vem do fundo do mar é recolhido a uma temperatura que está ao redor de 7°C, sendo necessário um aquecimento prévio de forma a diminuir a viscosidade do petróleo e incrementar a sua pressão de vapor. Este aquecimento é feito por intermédio de um trocador de calor tipo *Casco-Tubo* P-01, ver Fig. 3.2, que é alimentado por um fluxo de vapor a uma temperatura de 210°C, e a temperatura de saída do óleo é adquirida pelo transmissor de temperatura TT-02.

O controle de temperatura do petróleo é realizado através do controle do fluxo de vapor, comandada pela ação da malha de controle TIC-01 que age sobre a abertura da válvula TV-01. Esta válvula é acionada por um sistema pneumático a ar com o objetivo de manter uma temperatura constante, especificamente a 60°C, valor de leitura fornecido pelo transmissor de temperatura TT-01, instalado na saída do trocador de calor.

3.2.3 Primeiro Estágio de Separação

Após o aquecimento da mistura, ela é conduzida ao vaso separador SG-01, composto por um tanque cilíndrico que tem uma parede interna transversal de altura mediana a qual divide o tanque em dois. Em uma operação contínua, a mistura enche a primeira parte do vaso, separando água, areia (de densidade maior) no fundo do tanque e óleo (de densidade menor) na superfície, ver Fig. 3.3. O óleo retirado com maior grau de pureza ultrapassa a barreira enchendo a segunda parte do vaso.

Este vaso contém duas malhas de controle. Uma delas que mantém o nível de óleo no interior do vaso em uma altura correspondente à divisão interna, malha de controle de nível LIC-01 tipo PI, agindo sobre a válvula LV-01; um transmissor de nível LT-01 envia a leitura do nível de óleo no tanque para o controlador LIC-01. A outra malha controla a pressão interna no tanque através do controlador de pressão PIC-01, que age sobre a abertura da válvula PV-01, de forma a liberar gás em excesso e alcançar o valor desejado de pressão, valor que é enviado pelo

transmissor de pressão PT-01.

Dessa forma o nível e a pressão do vaso são mantidos nos valores desejados para a operação do processo, mais especificamente em $1.35m$ e $9.07kgf/cm^2abs$ respectivamente. Um computador de vazão FQI-02 calcula a vazão mássica de gás que sai do vaso separador. A vazão volumétrica do óleo é calculada pelo totalizador de vazão FQI-01, e este totalizador é protegido pelo Filtro de Líquidos FY-01 de modo a garantir sua operação mecânica. O gás que passa pela válvula PV-01 segue para as unidades de compressão e após a sua desidratação é exportado pelos gasodutos. O óleo que passa pela válvula LV-01 se junta com o óleo proveniente de outros separadores de primeiro estágio, tal como mostra a Figura 3.1 e posteriormente é enviado ao segundo estágio de separação.

3.2.4 Segundo Estágio de Separação

Após a passagem do petróleo pelo primeiro estágio, este segue ao segundo estágio de separação, composto pelo vaso separador SG-02, que é do tipo atmosférico. Esta etapa de separação é realizada em uma pressão mais baixa que do primeiro estágio, ao redor de $1.15kgf/cm^2$. O gás produzido no vaso SG-02 é enviado para as unidades de recuperação e desidratação nos quais será comprimido para atingir a mesma pressão do gás produzido no primeiro estágio, de modo a ser processado em conjunto e ser exportado para os gasodutos.

Neste estágio de separação existe uma malha de controle de nível LIC-02 que faz a leitura do transmissor de nível LT-02 e age sobre a válvula LV-02 para manter o nível de óleo num valor desejado. A pressão no vaso é controlada pela ação da malha PIC-02 que atua sobre a válvula de gás PV-02, a leitura de pressão interna do vaso separador é feita mediante o transmissor de pressão PT-02 e posteriormente transmitida ao controlador de pressão. Neste estágio de separação não há medição da vazão mássica de gás, só há medição da vazão volumétrica através do totalizador de vazão FQIT-04.

3.2.5 Exportação de Óleo

O óleo é bombeado até as monobóias utilizando a bomba B-01, que eleva a pressão do óleo proveniente do segundo estágio de separação. Nesta etapa é necessário

garantir uma vazão mínima na bomba B-01 sem causar variações consideráveis no nível de óleo no vaso SG-02, motivo pelo qual realiza-se uma etapa de recirculação de óleo. O controle desta etapa de recirculação é comandado pelo controlador de fluxo FIC-03 de tipo PI. O controlador recebe a leitura de fluxo proveniente do transmissor FT-03 e atua sobre a abertura da válvula FV-03, permitindo a recirculação de óleo ao vaso SG-02. Inclui-se também o orifício de restrição FO-01 que absorve a diferença de pressão entre a saída da bomba e a entrada do vaso separador, evitando problemas de vaporização na válvula de recirculação FV-03.

A operação da bomba B-01 é condicionada à atuação de alguns dos seguintes sensores: PSH-02, PSH-03, PSL-03, LSHH-02, LSL-02 ou FSL-05 que são parte do sistema de intertravamento. Adicionalmente, a montante da válvula LV-02 utiliza-se um totalizador de vazão FQI-04; após o óleo sair pela válvula LV-02, este é enviado às monobóias, tal como mostra a Fig. 3.1, terminando desta forma o processo de separação óleo-gás.

3.3 Modelagem Matemática do Processo

Com o objetivo de simular o processo em regime permanente é necessária uma modelagem matemática do sistema, esta simulação é detalhada na seção 3.5. Os modelos apresentados por Teixeira [30] apresentam um comportamento qualitativo aceitável no sentido de levar a um melhor entendimento da planta. Estes modelos incluem fenômenos físicos e químicos, também sendo levadas em consideração algumas hipóteses para simplificar os modelos, sacrificando alguns detalhes para ressaltar as características qualitativas. Apresentaremos uma modelagem das componentes e equipamentos do processo.

3.3.1 Trocador de Calor

A modelagem feita do trocador de calor é uma versão simplificada que reproduz de forma aceitável as características qualitativas do trocador real. Este modelo, dado em [30] e [7], pela modelagem rigorosa própria de um sistema distribuído, resultará em um sistema de equações diferenciais parciais EDP que, na versão simplificada [30] é substituído por um modelo de equações diferenciais ordinárias EDO. A Figura 3.2 mostra a estrutura do trocador de calor tipo tubo/casco.

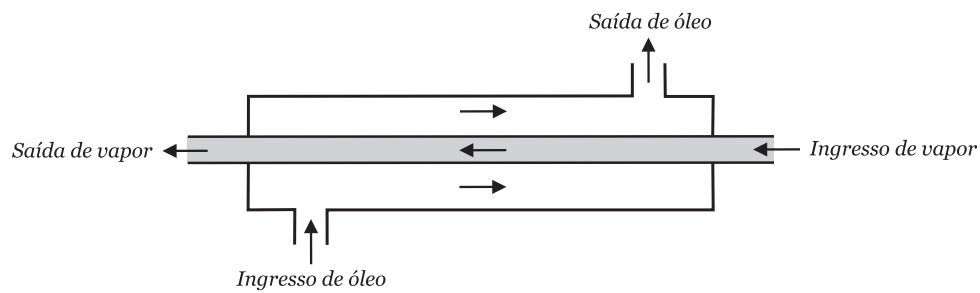


Figura 3.2: Trocador de calor

Para a modelagem consideram-se as seguintes variáveis:

Variáveis:

Q_{Tinput} : Vazão de entrada no tubo

Q_{Tin} : Vazão de vapor no tubo

Q_{Sinput} : Vazão de entrada no casco

Q_{Sin} : Vazão de óleo no casco

T_{Tin} : Temperatura de entrada ao tubo

T_{Tout} : Temperatura de saída do tubo

T_{Sin} : Temperatura de entrada do casco

T_{Sout} : Temperatura de saída de casco

V_G : Volume dos tubos do trocador

V_S : Volume do casco do trocador

C_{Tr} : Taxa de transferência de calor

ρ_T : Densidade do líquido no tubo

ρ_S : Densidade do líquido no casco

c_T : Constante de troca térmica nos tubos do trocador

c_S : Constante de troca térmica nos cascos do trocador

τ_T : Constante de tempo da vazão nos tubos do trocador

τ_S : Constante de tempo da vazão nos cascos do trocador

U_a : Coeficiente de troca térmica

Assumimos que c_T , c_S , ρ_t , ρ_S , U_a são constantes, assim, considera-se como entrada ao casco somente a vazão de óleo. As variações de vazão de óleo e vapor

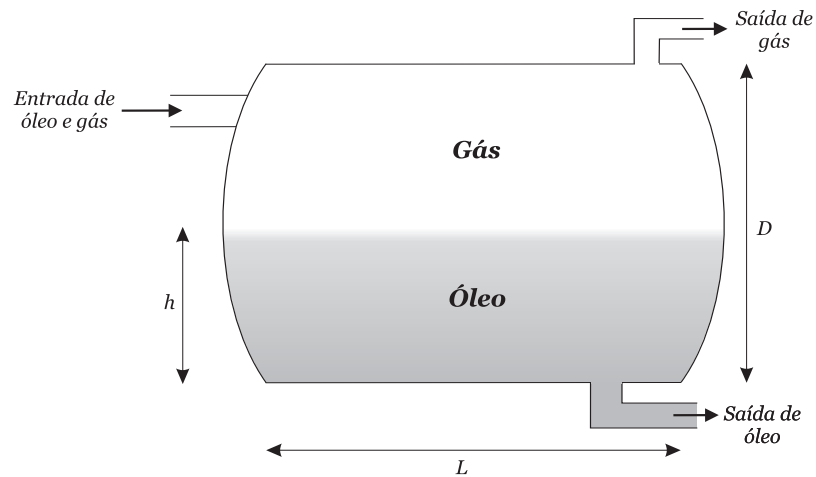


Figura 3.3: Detalhe do vaso separador

no casco e tubo são representadas pelas equações [7]:

$$\frac{dQ_{Tin}}{dt} = -\frac{Q_{Tin}}{\tau_T} + \frac{Q_{Tinput}}{\tau_T} \quad (3.1)$$

$$\frac{dQ_{Sin}}{dt} = -\frac{Q_{Sin}}{\tau_S} + \frac{Q_{Sininput}}{\tau_S} \quad (3.2)$$

A taxa de transferência de calor é expressada por [7]:

$$C_{Tr} = \frac{T_{Tin} - T_{Sin}}{\frac{1}{U_a} + 1800\left(\frac{1}{Q_{Tin} \cdot \rho_T \cdot c_T} + \frac{1}{Q_{Sin} \cdot \rho_S \cdot c_S}\right)} \quad (3.3)$$

Com isso, pode-se estabelecer as variações de temperatura na saída do casco e tubo, representadas por [7]:

$$\frac{dT_{Tout}}{dt} = -\frac{Q_{Tin}T_{Tout}}{3600V_T} + \frac{Q_{Tin}T_{Tin}}{3600V_T} - \frac{C_{Tr}}{\rho_T v_T c_T} \quad (3.4)$$

$$\frac{dT_{Sout}}{dt} = -\frac{Q_{Sin}T_{Sout}}{3600V_S} + \frac{Q_{Sin}T_{Sin}}{3600V_S} - \frac{C_{Tr}}{\rho_S v_S c_S} \quad (3.5)$$

3.3.2 Vasos Separadores

Na modelagem matemática dos vasos separadores, leva-se em consideração que a vazão de óleo e gás proveniente do trocador de calor e bomba de exportação têm

um comportamento relativamente uniforme, incluem-se também as propriedades químicas do óleo e gás de forma simplificada [30]. Para a modelagem, foram utilizadas as seguintes variáveis:

Variáveis:

Q_{Lin} : Vazão de entrada de óleo

Q_{Lout} : Vazão de saída de óleo

Q_{Gin} : Vazão de entrada de gás

Q_{Gout} : Vazão de saída de gás

V_L : Volume de óleo no vaso

V_G : Volume de gás no vaso

P : Pressão no interior do vaso

h : Nível de óleo no vaso

D : Diâmetro do vaso

L : Comprimento do vaso

Z : Compressibilidade do gás

n_G : Número de moles de gás

R : Número de Reynolds

O desenho do vaso com suas respectivas variáveis é mostrado na Figura 3.3. Este vaso corresponde á estrutura básica dos vasos SG-01 e SG-02. A seguinte modelagem pode se encontrar em [30]

Por balanço de volume, pode-se notar que:

$$\frac{dV_L}{dt} = Q_{Lin} - Q_{Lout} \quad (3.6)$$

Pela geometria do vaso e a área da superfície do líquido, a variação do nível de óleo é dado por:

$$\frac{dh}{dt} = \frac{dV_L}{dt} \cdot \frac{1}{A(h)} = \frac{dV_L}{dt} \cdot \frac{1}{2L\sqrt{(D-h)h}} \quad (3.7)$$

Já que o volume do vaso é constante, então a variação no volume do gás é:

$$\frac{dV_G}{dt} = -\frac{dV_L}{dt} \quad (3.8)$$

Das equações dos gases obtém-se:

$$PV_G = Zn_GRT \quad (3.9)$$

Desprezando as variações de Z e na temperatura, a diferenciação da equação 3.9, obtém-se:

$$V_G \frac{dP}{dt} = ZRT \frac{dn_G}{dt} - P \frac{dV_G}{dt} \quad (3.10)$$

O balanço de massa de gás no vaso fornece a equação:

$$ZRT \frac{dn_G}{dt} = P(Q_{Gin} - Q_{Gout}) \quad (3.11)$$

A equação da pressão no vaso é calculada das equações 3.10 e 3.11 :

$$\frac{dP}{dt} = \frac{P}{V_G}(Q_{Gin} - Q_{Gout} + Q_{Lin} - Q_{Lout}) \quad (3.12)$$

3.3.3 Válvulas de Controle

Uma válvula define-se como um aparato mecânico com o qual é possível iniciar, parar ou regular o fluxo de líquidos ou gases mediante uma peça móvel que abre, fecha ou obstrui em forma parcial um ou mais orifícios de condução de fluido. A válvula automática de controle geralmente constitui o último elemento na malha de controle instalada em uma linha de processo e se comporta como um orifício cuja seção de passagem varia continuamente com a finalidade de controlar uma vazão em uma forma determinada.

O Coeficiente de vazão C_v está associado a capacidade máxima que a válvula pode atingir, este valor é fornecido pelos fabricantes mediante uma curva onde C_v é função da abertura da válvula (x), para maior simplicidade serão usados valores normalizados na abertura da válvula, i.e. "0" quando a válvula esteja totalmente fechada e "1" quando a válvula esteja aberta ao 100%. Foram consideradas todas as válvulas do tipo linear, segundo a Figura 3.4 [29]. Para a modelagem, foram utilizadas as seguintes variáveis:

Variáveis:

P_{IN} : Pressão de entrada na válvula.

P_{OUT} : Pressão de saída na válvula.

C_v : Coeficiente de vazão.

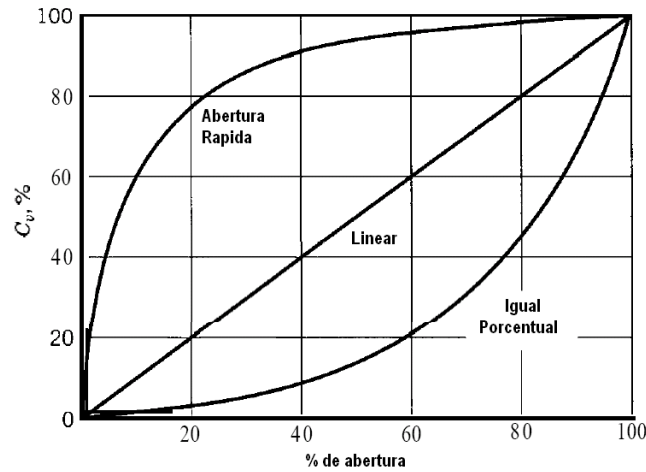


Figura 3.4: Curva Característica das válvulas, C_v versus x

C_{vMAX} : Coeficiente de vazão máxima.

x : Posição da válvula.

y : Set Point da posição válvula.

τ_v : Constante de tempo da válvula.

ΔP : Queda de Pressão na válvula.

γ : Gravidade específica.

T : Temperatura.

C_f : Coeficiente de fluxo.

P_v : Pressão de vapor.

P_c : Pressão crítica.

A válvula considerada é do tipo haste, onde o movimento da haste atua em resposta ao sinal de controle, esse movimento é representado pela seguinte equação diferencial:

$$\frac{dx}{dt} = \frac{y - x}{\tau_v} \quad (3.13)$$

Sendo a constante de tempo τ_v fornecida pelo fabricante, que é o tempo de resposta da válvula. A capacidade de vazão é obtida através da Figura [3.4] e da equação:

$$C_v = C_{vMAX} \cdot f(x) \quad (3.14)$$

Usou-se fórmulas dadas em dimensionamento de válvulas e logo condicionadas para nosso estudo [30]. As equações são dadas a seguir:

Para *líquidos*, primeiramente calcule-se $\Delta P = P_{IN} - P_{OUT}$ e $\Delta P_s = P_{IN} - P_v(0,96 - 0,28\sqrt{P_v/P_c})$, o valor da vazão Q é calculado por:

Se $(\Delta P < C_f^2 \dot{\Delta} P_s)$:

$$Q = \frac{c_v}{1.16} \cdot C_v \sqrt{\frac{\Delta P}{\gamma}} \quad (3.15)$$

Se $(\Delta P \geq C_f^2 \dot{\Delta} P_s)$:

$$Q = \frac{c_v}{1.16} \cdot C_v \sqrt{\frac{\Delta P_s}{\gamma}} \quad (3.16)$$

Para *vapores ou gases*, escolha-se a relação certa mediante o seguinte critério:

Se $(\Delta P < 0,5C_f^2 \dot{P}_{IN})$:

$$Q' = 295C_v \sqrt{\frac{\Delta P(P_{IN+P_{OUT}})}{\gamma T}} \quad (3.17)$$

Se $(\Delta P \geq 0,5C_f^2 \dot{P}_{IN})$:

$$Q' = \frac{257C_v P_{IN} C_f}{\sqrt{\gamma T}} \quad (3.18)$$

Onde o valor de Q' deve se ajustar para ter as unidades em $(m^3/h$ a uma temperatura T em graus Kelvin e uma pressão P em $kgf/cm^2)$:

$$Q = Q' \cdot \frac{1,03T}{288P} \quad (3.19)$$

Em modelos simplificados onde a perda de pressão na válvula é baixa, pode se considerar que a vazão de saída na válvula é proporcional ao movimento do haste e a vazão de entrada.

3.3.4 Bomba Centrífuga

A bomba é uma máquina turbo geradora para líquidos, usada para transformar energia mecânica em energia hidráulica. Um sistema de bombeio pode-ser definido como a adição de energia a um fluido para se movimentar ou se transladar de um ponto a outro, neste caso para movimentar o óleo que está no vaso separador até as

monobóias. A modelagem da bomba centrífuga B-01 foi considerando-a como um gerador de pressão [30], onde a pressão gerada acompanha a curva característica da própria válvula. Assumiu-se também que a velocidade de rotação da válvula se mantém constante.

Variáveis:

Q_{Bomb} : Vazão através da bomba

Q_{FV-02} : Vazão através da válvula FV-02

Q_{LV-02} : Vazão através da válvula LV-02

Q_{VGL-02} : Vazão através da válvula VGL-02

P_{vaso} : Pressão no fundo do vaso

P_{in} : Pressão de sucção da bomba

ΔP_{BY-01} : Perda de pressão no filtro BY-01

ΔP : Pressão gerada na bomba

ρ : Densidade do fluido na Bomba

g : Gravidade

h : Nível de óleo no vaso

A vazão através da bomba B-01 é a soma das vazões nas válvulas que contribuem nela, isto é:

$$Q_{Bomb} = Q_{FV-03} + Q_{LV-02} + Q_{VGL-02} \quad (3.20)$$

Temos que a pressão de sucção da bomba é representada por:

$$P_{in} = P_{vaso} + 10^{-4}h\rho g - \Delta P_{BY-01} \quad (3.21)$$

A pressão de saída na Bomba é dada por:

$$P_{out} = P_{in} + \Delta P \quad (3.22)$$

Onde:

$$\Delta P = 10^{-4}\rho g f(Q_{Bomb}) \quad (3.23)$$

3.3.5 Controladores

Tal como foi explicado anteriormente, o tipo de controladores usados no processo são todos PI, a exceção do TIC-01 que é do tipo PID, todos eles com dispositivo

anti-windup, para não saturar a sinal de controle da contribuição integral do controlador. A modelagem usada é apresentada em [18], onde os controladores (PIs, PIDs) estão de acordo a figura 3.5. Os algoritmos implementam a seguinte lei de controle:

Variáveis:

$u(k)$: Saída do controlador antes da saturação

$v(k)$: Saída do controlador após da saturação

$e(k)$: Sinal de erro

$P(k)$: Contribuição proporcional

$I(k)$: Contribuição integral

$D(k)$: Contribuição derivativa

K : Ganho proporcional

T_i : Constante de tempo integral

T_d : Ganho derivativo

T_t : Ganho de compensação (*antiwindup*)

N : Constante de ajuste derivativo

h : Tempo de amostragem.

$$u(k) = \text{sat}[P(k)] + I(k) + D(k) \quad (3.24)$$

$$P(k) = K \cdot e(k) \quad (3.25)$$

$$I(k) = \begin{cases} I(k-1) + \frac{Kh}{T_i}e(k-1), & \text{se } v(k-1) = u(k-1) \\ I(k-1) + \frac{Kh}{T_i}e(k-1) - \frac{h}{T_t}[u(k-1) - v(k-1)], & \text{c.c.} \end{cases} \quad (3.26)$$

$$D(k) = D(k-1)\frac{T_d}{T_d + Nh} - [e(k) - e(k-1)]\frac{KT_dN}{T_d + Nh} \quad (3.27)$$

3.3.6 Placas Orifício

As instalações atuais dos sistemas industriais incluem dispositivos de proteção de funcionamento de equipamentos, de modo a dar o maior tempo de vida útil de ditos equipamentos. Uma placa de orifício é um dispositivo principalmente usados na medição de vazão e com algumas aplicações para absorver parte da perda de carga entre dois pontos distintos, evitando assim que a carga ou diferença de pressão seja absorvida por equipamentos sensíveis a esta mudança de condições físicas. A placa orifício é uma placa estreita com furo concêntrico que impõe, ao fluxo que a atravessa, uma queda de pressão similar à da válvula. As placas de orifício são

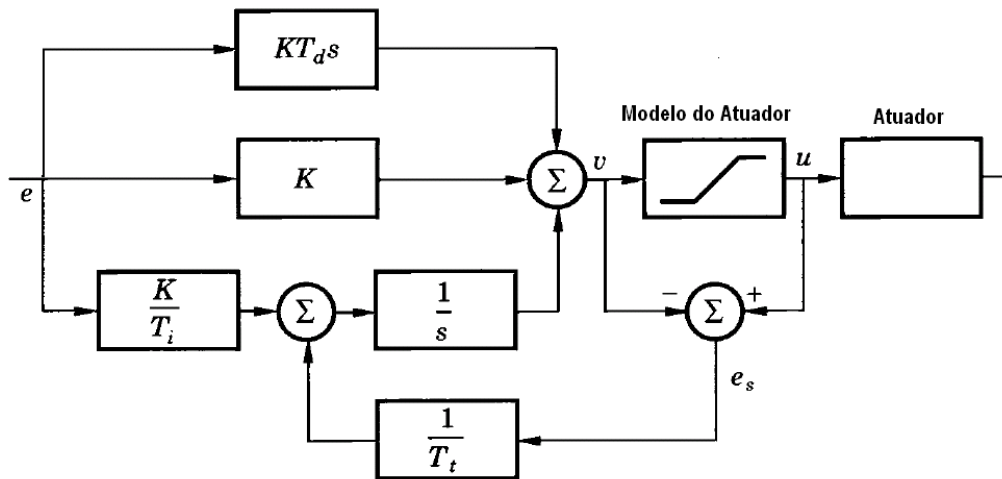


Figura 3.5: Controlador com Antiwindup

Um sistema no qual a saída do atuador é estimada de um modelo matemático

usadas para criar um diferencial de pressão nos medidores de vazão e válvula, em muitas aplicações toma o nome de orifício de restrição, uma vez que a placa é instalada em serie com estes equipamentos de medição.

Variáveis:

ΔP : Diferencial de pressão gerado pela placa.

ΔP_{CALC} : Diferencial de pressão.

Q : Vazão do fluido.

ΔQ_{CALC} : Diferencial de vazão de calculo da placa.

Em [30], se apresenta o diferencial de pressão gerado pela placa ΔP sendo aproximado pela seguinte equação:

$$\Delta P = Q^2 \frac{\Delta P_{CALC}}{Q_{CALC}^2} \quad (3.28)$$

3.3.7 Filtros

O uso dos filtros é para evitar que os equipamentos sejam danificados pelas partículas sólidas que estão suspensas no óleo, o efeito dos filtros no processo é a introdução de uma perda de carga na linha.

Variáveis:

P_{IN} : Pressão de entrada ao filtro.

P_{OUT} : Pressão de saída do filtro.

ΔP : Pressão absorvida no filtro.

A pressão na saída do filtro [30], é dada por:

$$P_{OUT} = P_{IN} - \Delta P \quad (3.29)$$

3.4 Falhas no Processo

O sistema em estudo possui diversos tipos de falhas, alguns ocasionados pelos distúrbios ao que está exposto o processo e outros pelo funcionamento em falha de um o mais equipamentos. Consideraremos uma *falha* como um evento não-observável que causa um dano, degrada e impede a operação de parte ou todo o processo. Esta consideração das falhas como sendo eventos não-observáveis vem pelo fato que as falhas são não-visíveis ou não-registráveis diretamente das variáveis do processo. Em caso de falhas consideradas observáveis a inferência de eventos se reduz em um problema trivial, onde é fácil registrar um tipo de falha determinado a partir de eventos observados.

3.4.1 Falhas no trocador de calor

Neste equipamento há dois tipos de falhas [30]:

Vazamento tubos-casco, este tipo de falha ocorre quando o vapor (água quente) se mistura com o óleo no casco, causado pela ruptura do tubo do trocador. O efeito deste tipo de falha é o aumento da temperatura na saída do trocador, assim como o incremento da vazão da mistura.

Perda de eficiência na troca térmica, este tipo de falha se caracteriza pela perda de eficiência do trocador e a diminuição do coeficiente de troca térmica causada pela aglomeração de sólidos nas paredes dos tubos, restringindo a vazão de vapor no trocador.

3.4.2 Falhas nas válvulas de controle:

No processo de separação em estudo, considerou-se o uso de válvulas de controle do tipo globo com atuador mola-diafragma. O sinal de controle da válvula é um sinal

pneumático proveniente do conversor elétrico-pneumático, deste modo a pressão de (proveniente do transdutor pneumático) age sobre a haste da válvula mediante o diafragma, e contra a força elástica da mola que se situa no extremo da haste. Como resultado destas forças a haste provoca o movimento do obturador que está no outro extremo do haste. Devido ao funcionamento e operação da válvula, foram considerados dois tipos de falhas relacionadas ao seu funcionamento:

Falha por falta de suprimento de ar, este tipo de falha é causado pela falta de ar proveniente do conversor pneumático. Neste caso, a válvula adota duas possíveis posições: Totalmente aberta ou Totalmente fechada, dependendo da configuração da mola. Será considerada que a válvula tem configurada a mola para entrar em posição de totalmente aberta.

Falha por haste emperrada, este tipo de falha é devido a um impedimento mecânico que impossibilita o movimento da haste, impedindo que a válvula se abra ou feche.

3.4.3 Falhas nos vasos separadores:

Considera-se que as possíveis falhas nos vasos separadores são tipificadas como vazamento de óleo e vazamento de gás. Estes tipos de falhas provêm da ocorrência grave de natureza mecânica ou estrutural dos vasos separadores e das tubulações contidas nele, causando uma perda da produção de óleo o gás.

3.4.4 Falhas na bomba:

A bomba considerada no sistema de separação óleo-gás, encarregada da exportação de óleo no segundo estágio de separação é do tipo centrífuga, esta bomba é acionada por motor elétrico. As possíveis falhas neste equipamento são: *Desligamento indevido da bomba* que normalmente ocorre por causa do mal funcionamento do sistema de intertravamento, por averia do motor elétrico que a aciona, por falta de suprimento de energia elétrica ao motor ou por ação de uma operação indevida. Quando esta bomba é desligada, pode ser interrompido todo o processo, uma vez que isto pode acionar o intertravamento o que provoca o fechamento das válvulas de shut-down.

3.4.5 Falhas nos filtros:

A localização dos filtros é à montante dos equipamentos e instrumentos de medição para evitar que sejam danificados pelas partículas sólidas que estão suspensas no óleo, tais como areia. Com o transcorrer do tempo, os filtros progressivamente vão se tornando sujos, o que causa a introdução de uma perda de carga na linha de circulação. As falhas consideradas nestes equipamentos são *filtro sujo*, causada pelo bloqueio parcial da passagem de óleo no filtro, e *filtro entupido*, causada pelo bloqueio total da passagem de óleo no filtro ao estar completamente sujo.

3.5 Simulação

O objetivo da simulação é reproduzir o comportamento quantitativo e qualitativo do processo para sua respectiva análise física-funcional, operação em condições normais e de falha. Na simulação completa do Processo de Separação óleo-gás, considerou-se quatro etapas:

- a) Etapa de aquecimento.
- b) Primeiro estágio de separação.
- c) Segundo estágio de separação e exportação de óleo.

O processo foi simulado em Simulink-MATLAB. A simulação foi baseada nos modelos matemáticos da Seção 3.3, onde os valores das constantes e grandezas foram extraídas de trabalhos anteriores [7] e [30], encontram-se em unidades segundo o Sistema Internacional SI. Deve-se notar que na simulação não foram considerados os diferentes atrasos.

As falhas também foram geradas no simulador, para análise das suas consequências na variação do comportamento normal do processo, e assim construir o modelo do Sistema Automático de Detecção de Falhas que será visto no Capítulo 4. Em geral, o comportamento qualitativo e quantitativo da simulação do processo completo foi aceitável, mas infelizmente não foi possível comparar com dados reais do processo. A continuação se mostram alguns dos esquemas de simulação desenvolvidos em Simulink que foram utilizados para emular e analisar o comportamento (quantitativo e qualitativo) das etapas do processo de separação óleo-gás: Trocador de calor: Figura 3.6, Figura 3.7, Figura 3.8, Figura 3.9; vasos Separadores: Figura 3.10, Figura 3.11, Figura 3.12, Figura 3.13.

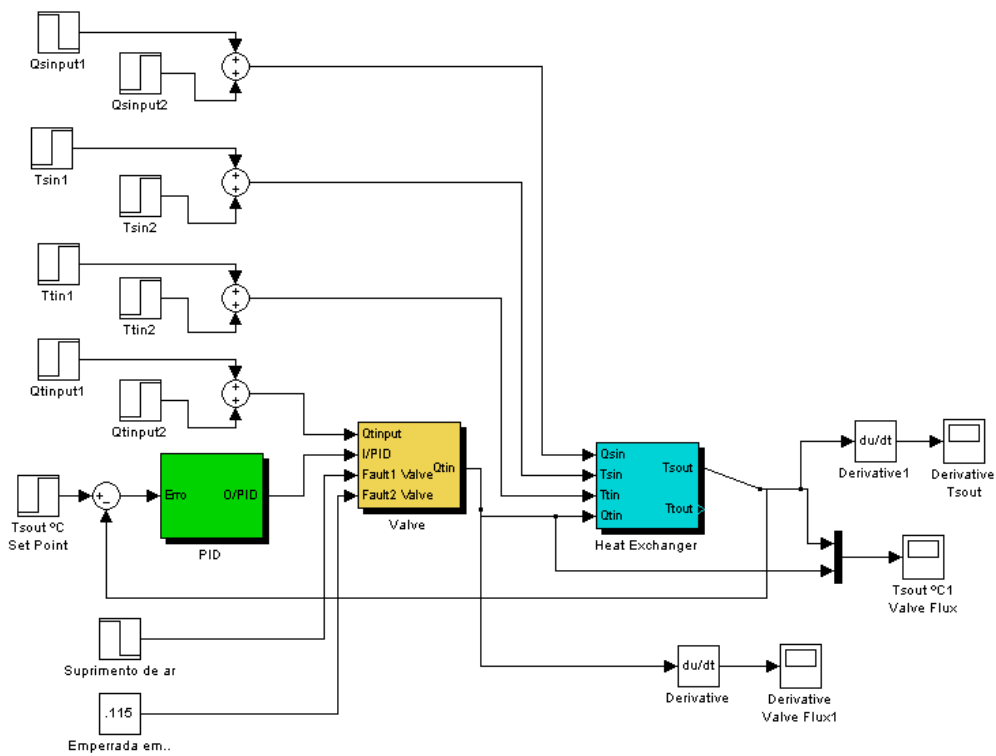


Figura 3.6: Esquema completo de Simulação do trocador de calor

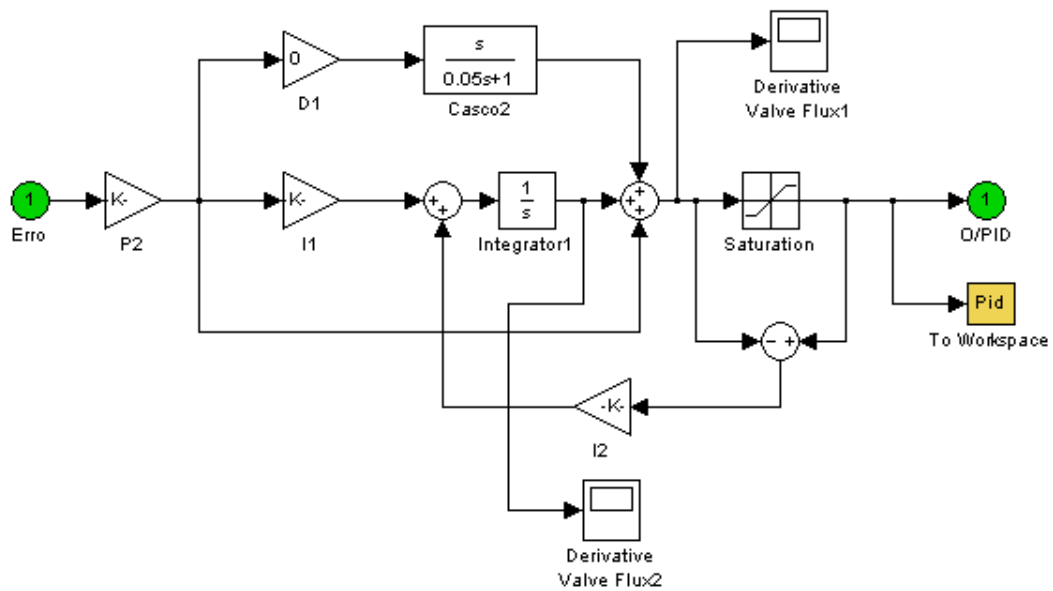


Figura 3.7: Esquema de Simulação do controlador PID no trocador de calor

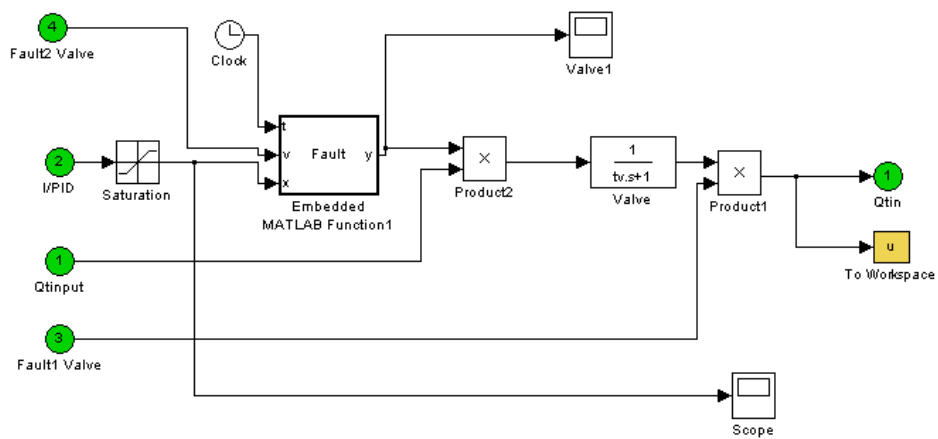


Figura 3.8: Esquema de Simulação da válvula de controle de temperatura do trocador calor

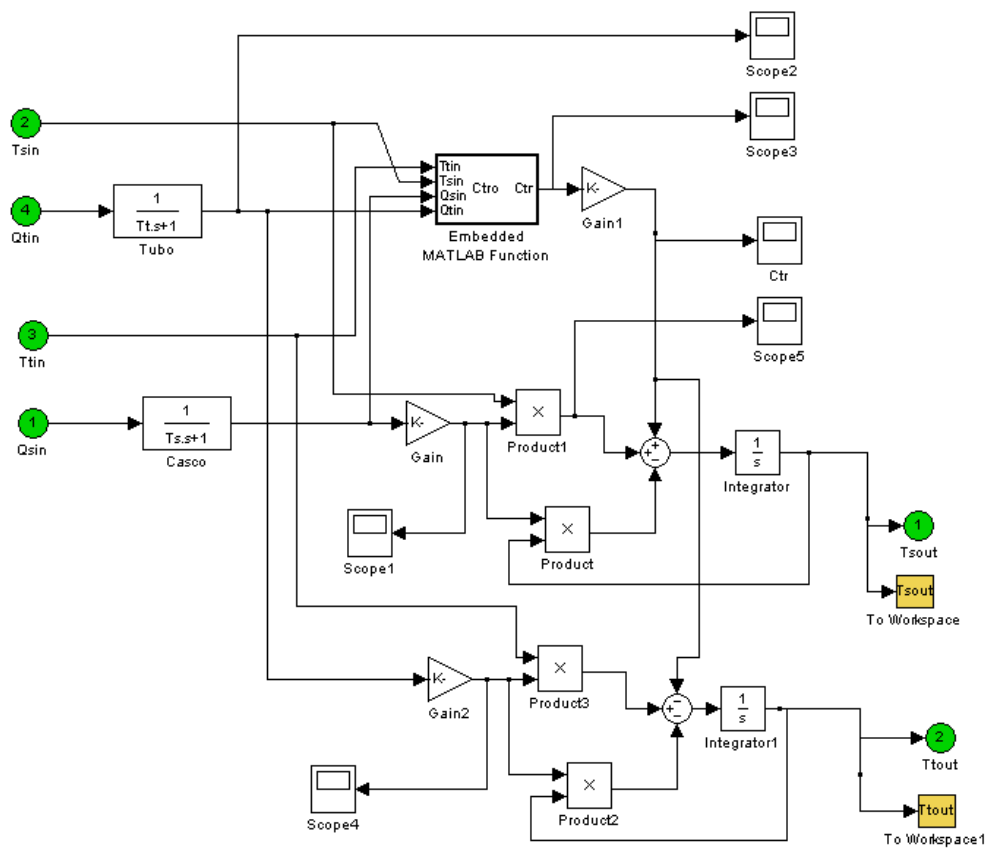


Figura 3.9: Esquema de Simulação do trocador de calor

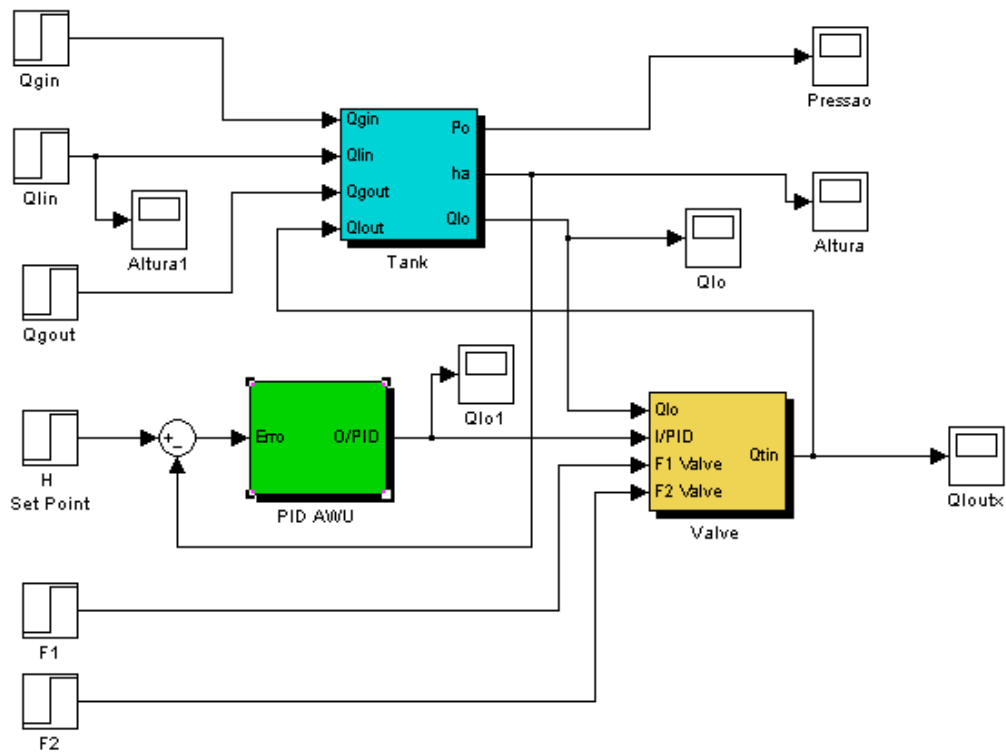


Figura 3.10: Esquema Completo de Simulação do Estágio de Separação

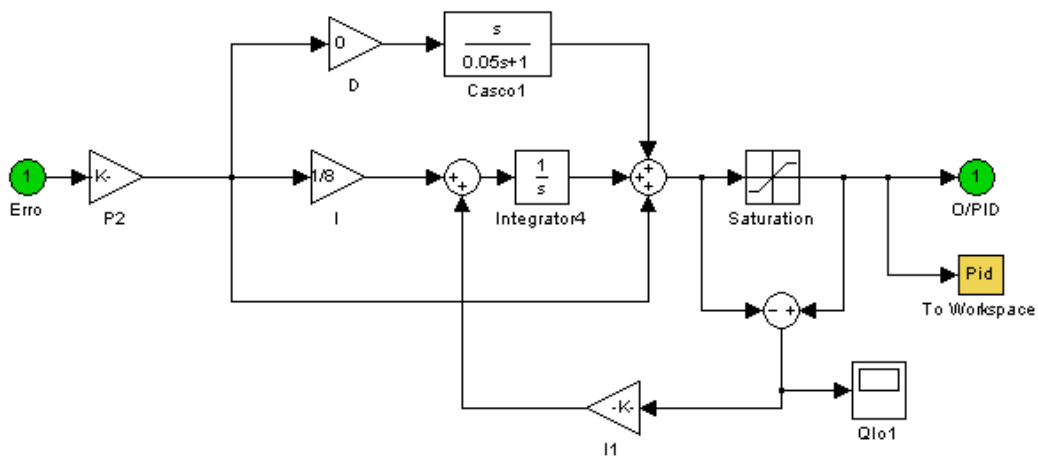


Figura 3.11: Esquema de Simulação do Controlador de Nível PID, no Vaso Separador

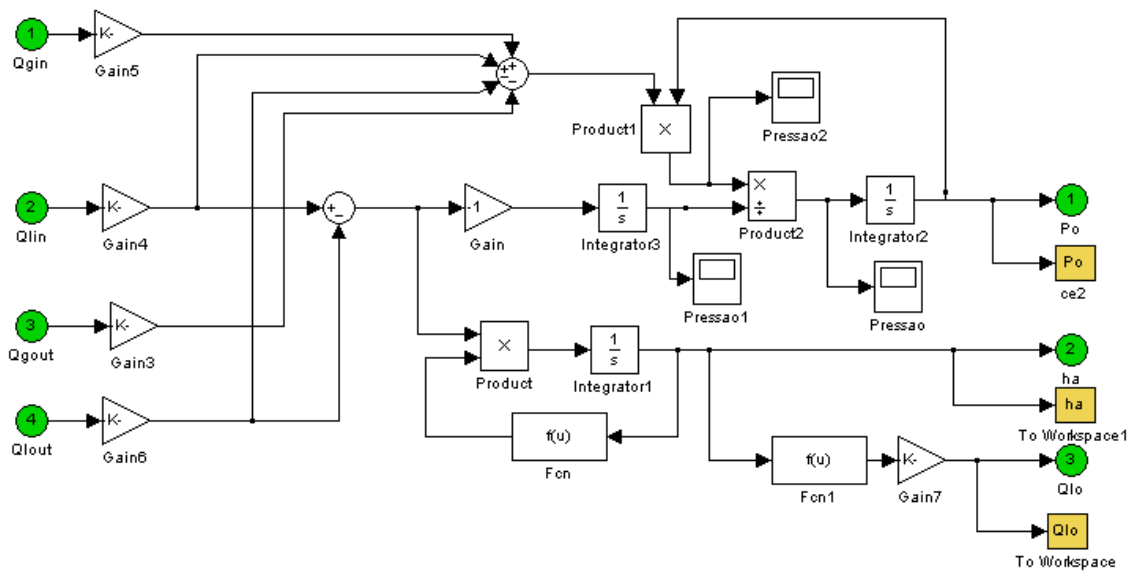


Figura 3.12: Esquema de Simulação do Vaso Separador

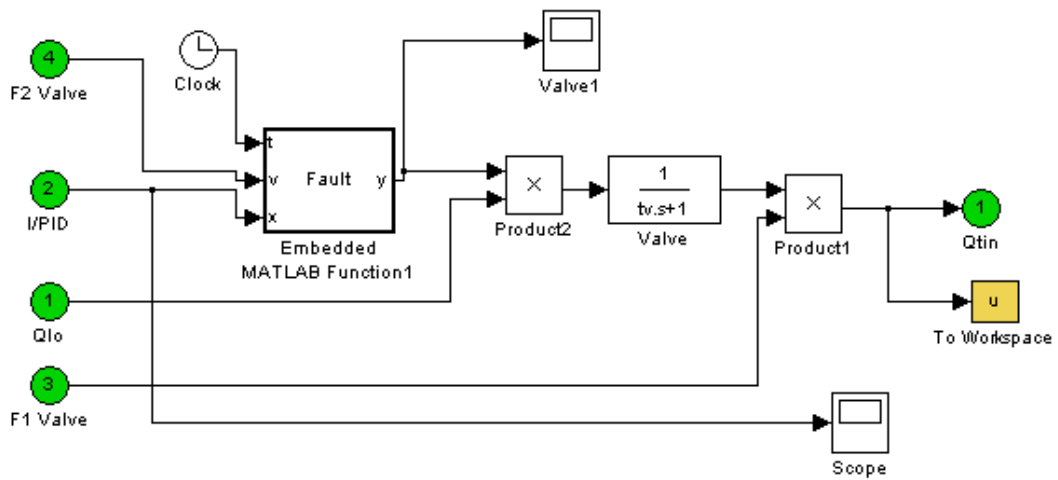


Figura 3.13: Esquema de Simulação da válvula de Nível, no Vaso Separador

Capítulo 4

Projeto dos Diagnosticadores no Processo de Separação Óleo-Gás

4.1 Introdução

No presente capítulo serão apresentados os resultados da aplicação do procedimento dado na Seção 2.7 ao processo de separação óleo-gás descrito no Capítulo 3, sendo esta aplicação uma das contribuições do presente trabalho, tanto à teoria de Sistemas a Eventos Discretos em uma aplicação prática de diagnóstico de Falhas, assim como a modelagem de sistemas com Redes de Petri. Este tipo de diagnóstico é também denominado diagnóstico baseado na construção do modelo [27], onde é fundamental partir de uma modelagem que contenha suficiente informação do comportamento e instrumentação embutida. Como foi visto anteriormente, com relação aos sub-modelos precisou-se começar a modelagem a partir de redes de Petri com lugares iniciais e as que não contemplaram esta restrição foram convertidas para redes com lugares iniciais mediante o Procedimento 1 (Seção 2.3.5).

Para a construção dos modelos foi fundamental a observação das simulações para a análise do comportamento parcial e total sendo utilizada esta informação como uma fonte de conhecimento para nosso diagnosticador. A modelagem do processo foi considerado para uma operação em regime permanente, evitando as fases de partida e parada do processo, já que nestas fases são produzidos muitos transitórios de natureza linear e não linear que torna seu diagnóstico mais complicado, dado que as diferentes condições iniciais produzem-se diferentes comportamentos transitórios. Um sistema de diagnóstico de falhas que compreenda estas fases é matéria de um estudo e análise com maior profundidade, e não será visto neste

trabalho, que abordará apenas a questão dos transitórios que surgem a partir de perturbações que ocorrem quando o sistema está no regime.

Com relação ao processo em estudo: Em uma implementação real é possível que se tenha a necessidade de realizar alguns ajustes dos modelos mas sem variações consideráveis, dado de que as simulações cumpriram com os requerimentos qualitativos esperados do processo real. Considera-se também que possa se precisar de uma filtragem dos dados de medição para diminuir os ruídos do processo. Nesta seção apresentamos o projeto dos diagnosticadores do processo de separação óleo-gás, a validação dos modelos projetados em redes de Petri e a simulação conjunta de parte do processo com o diagnosticador que observa *on-line* o comportamento da planta. Mais adiante serão dadas algumas alternativas de implementação real dos diagnosticadores.

4.2 Diagnosticadores do Processo de Separação Óleo-Gás

Para o melhor entendimento, o projeto do sistema de detecção e diagnóstico de falhas do Processo de Separação Óleo-Gás será feita para cada uma das etapas do processo: Malha de controle de temperatura no trocador de calor, malhas de controle de nível e pressão no primeiro estágio de separação, malhas de controle de nível, fluxo e pressão no segundo estágio de separação, ver figura 3.1; considerando as respectivas falhas que foram descritas na Seção 3.4.

4.2.1 Trocador de Calor:

Consideremos a etapa de aquecimento do sistema de separação óleo-gás feita pelo trocador de calor da Seção 3.4.1. A tabela 4.1 mostra os medidores presentes no trocador de calor. Esta etapa é composta pelo seguintes elementos: Um elemento atuador TV-01 (Válvula), e um controlador de temperatura TIC-01 (Controlador). Adicionalmente mudanças de temperatura causadas por perturbações externas tais como variações nas temperaturas e fluxos de vapor, ingresso de petróleo, etc., serão consideradas no modelo como carga de temperatura. O primeiro passo para a construção de diagnosticadores segundo a seção 2.10 é obter modelos em redes de Petri para cada componente do sistema. As redes de Petri para os três elemen-

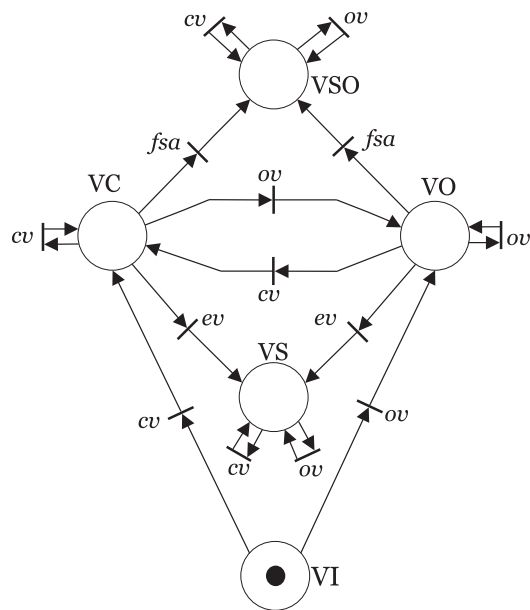
Figura 4.1: Rede de Petri N_v que representa a Modelagem da válvula TV-01

Tabela 4.1: Medições no Trocador

Símbolo	Medição
TT-01	Transmissor de temperatura do óleo na saída do trocador
TT-02	Transmissor de temperatura do vapor na entrada do trocador

tos: Válvula N_v , carga de temperatura N_c e controlador N_p , são mostrados a seguir:

Válvula: (Figura 4.1)

$$N_{vt} = \{P_v, T_v, Pre_v, Post_v, \Sigma_v, l_v, M_{v0}\}$$

Onde:

$$P_V = \{VI, VC, VO, TI, TL, TH, VS, VSO\}$$

$$T_v = \{t_{v1}, t_{v2}, t_{v3}, t_{v4}, t_{v5}, t_{v6}, t_{v7}, t_{v8}, t_{v9}, t_{v10}, t_{v11}, t_{v12}, t_{v13}, t_{v14}\}$$

$$\Sigma_v = \{ov, cv, fsa, ev\}$$

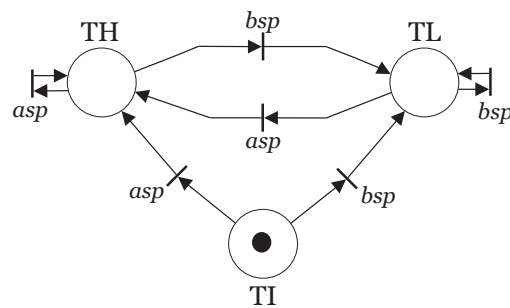
$$l : T_v \rightarrow \Sigma_v = \{cv, ov, ov, cv, ov, cv, ov, cv, fsa, fsa, cv, ov, ev, ev\}$$

$$M_{v0} = [1 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.2 mostra a simbologia dos lugares e eventos da válvula PV-01.

Tabela 4.2: Válvula do trocador de calor

Símbolo	Descrição
<i>VI</i>	Válvula em estado Inicial
<i>VC</i>	Válvula fechando
<i>VO</i>	Válvula abrindo
<i>VS</i>	Válvula emperrada
<i>VSO</i>	Válvula emperrada totalmente aberta
<i>ov</i>	abrir válvula
<i>cv</i>	fechar válvula
<i>fsa</i>	falta de suprimento de ar
<i>ev</i>	emperramento de válvula

Figura 4.2: Rede de Petri N_c que representa a carga de perturbação no trocador de calor

Carga: (Figura 4.2)

$$N_{ct} = \{P_c, T_c, Pre_c, Post_c, \Sigma_c, l_c, M_{c0}\}$$

Onde:

$$P_c = \{TI, TL, TH\}$$

$$T_c = \{t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5}, t_{c6}\}$$

$$\Sigma_c = \{bsp, asp\}$$

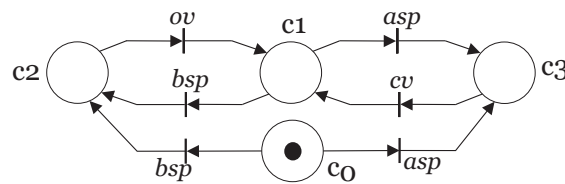
$$l : T_c \rightarrow \Sigma_c = \{bsp, asp, bsp, asp, bsp, asp\}$$

$$M_{c0} = [1 \ 0 \ 0]$$

Os elementos de conjunto de lugares P_p não tem significado ou interpretação física, representam só estados intermediários na ação de controle. A tabela 4.3 mostra a simbologia dos lugares e eventos da perturbação o carga de temperatura.

Tabela 4.3: Carga no trocador de calor

Símbolo	Descrição
TI	Temperatura inicial
TL	Temperatura baixa
TH	Temperatura alta
bsp	Temperatura abaixo do setpoint
asp	Temperatura acima do setpoint

Figura 4.3: Rede de Petri N_v que representa a modelagem do controlador TIC-01

Controlador:(Figura 4.3)

$$N_{pts} = \{P_p, T_p, Pre_p, Post_p, \Sigma_p, l_p, M_{p0}\}$$

Onde:

$$P_p = \{c0, c1, c2, c3\}$$

$$T_p = \{t_{p1}, t_{p2}, t_{p3}, t_{p4}\}$$

$$\Sigma_p = \{ov, cv, bsp, asp\}$$

$$l : T_p \rightarrow \Sigma_p = \{ov, bsp, cv, asp\}$$

$$M_{c0} = [1 \ 0 \ 0 \ 0]$$

A tabela 4.4 mostra a simbologia dos lugares e eventos do controlador de temperatura.

O segundo passo no projeto do diagnosticador é fazer a composição paralela de

Tabela 4.4: Controlador do trocador de calor

Símbolo	Descrição
$c0$	Controlador em estado 0
$c1$	Controlador em estado 1
$c2$	Controlador em estado 2
$c3$	Controlador em estado 3

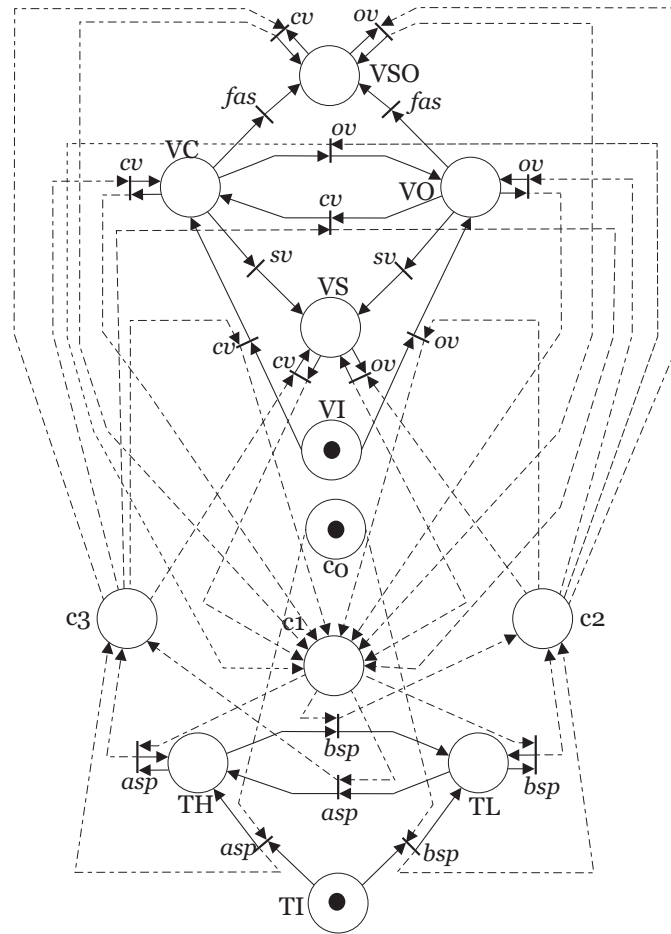


Figura 4.4: Rede de Petri N_t do trocador de calor

todas as componentes individuais. O modelo que representa o comportamento completo do trocador é obtido a partir da composição paralela das redes N_{vt} , N_{pt} e N_{ct} , i.e., a rede $N_t = N_{vt} \parallel N_{ct} \parallel N_{pt}$, onde $N_t = \{P, T, Pre, Post, \Sigma, l, M_0\}$ é mostrada na figura 4.4. Com o propósito de definir marcações, o conjunto de lugares de N_t será escrito como: $P = \{VI, VC, VO, TI, TL, TH, c0, c1, c2, c3, VS, VSO\}$.

Como foi visto na seção 3.4.1, no trocador de calor existem dois tipos de falhas, então definimos as partições falha com eventos não-observáveis: $\Sigma_{f1} = \{fas\}$ por falta de suprimento de ar proveniente do conversor pneumático que têm a posição de segurança pre-especificada "Totalmente Aberta"(Falha abre) e $\Sigma_{f2} = \{ev\}$ falha por emperramento da válvula em alguma posição intermediária; para um conjunto de falhas $\Sigma_f = \{fas, ev\}$.

O próximo passo no projeto do diagnosticador é desenvolver o mapeamento de

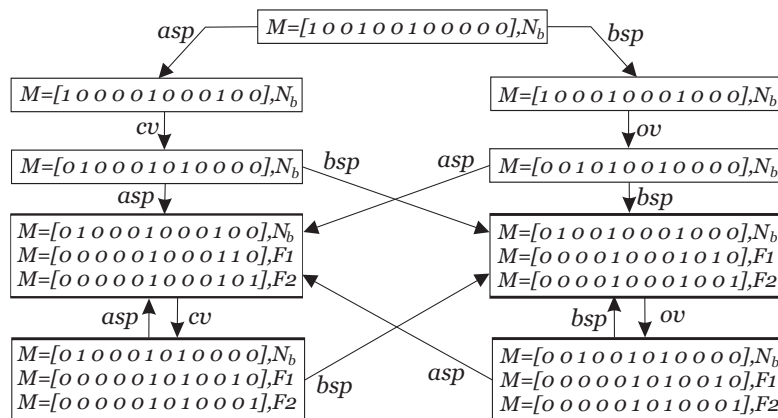


Figura 4.5: Diagnosticador G_{tdiag} do trocador de calor representado por N_t

sensores. Com o objetivo de visualizar mais claramente a necessidade de mapeamento de sensores será considerado inicialmente o caso onde não foi realizado o mapeamento de sensores. Segundo o procedimento 5 construa-se o diagnosticador G_{tdiag} para N_t , onde G_{tdiag} é um autômato que mostra a diagnosticabilidade de N_t ; onde se pode ver que o modelo apresentado de N_t é não-diagnosticável, posto que G_{tdiag} contém ao menos um ciclo indeterminado, sendo assim, então não é possível inferir a ocorrência de uma falha num tempo finito, i.e., considere duas seqüências de eventos:

$$s_1 = asp \ cv \ bsp \ ov \ fsa \ asp \ cv, \quad e \quad s_2 = asp \ cv \ bsp \ ov \ asp \ cv$$

é fácil ver que:

$$P(s_1) = P(s_2) = asp \ cv \ bsp \ ov \ asp \ cv$$

Mas, $\Sigma_{f1} \in s_1$ e $\Sigma_{f1} \notin s_2$ onde $M_{N_0}[s_1] > M_{N_{s_1}}$ e $M_{N_0}[s_2] > M_{N_{s_2}}$ com $M_{N_{s_1}} \neq M_{N_{s_2}}$, o que viola a condição de diagnosticabilidade D , da definição 2.10.

Dado que o sistema modelado por N_t é não-diagnosticável, então é preciso o uso de diagnóstico baseado no modelo, mais precisamente, introduzir o respectivo mapeamento de sensores nos sub-modelos representados pelas redes de Petri N_{vt} , N_{pt} e N_{ct} .

O transmissor de temperatura TT-01 será usado no mapeamento de sensores da modelagem do trocador de calor. Aqui, só consideraremos sensores virtuais, isto é motivado pelo fato de que os eventos asp e bsp são calculados a partir da diferença entre a leitura do TT-01 e o set-point de nível. A seguinte informação pode ser

Tabela 4.5: Sensores Virtuais do Trocador de calor

Símbolo	Descrição
T_{oa}	Temperatura aumentando.
T_{od}	Temperatura diminuindo.
$Err+$	Erro positivo.
$Err-$	Erro negativo.
$Err \sim 0$	Erro de regime próximo a zero.
$Err \gg 0$	Erro de regime muito maior que zero.
$Err > 0$	Erro de regime maior que zero.

obtida de sensores virtuais utilizando a leitura do transmissor TT-01: (i) O sinal da variação da leitura de temperatura, isto é o sinal da derivada da temperatura lida ($sign(dT/dt)$); (ii) O sinal do erro entre o set-point com a temperatura de saída, e (iii) O valor absoluto do erro em regime permanente, obtido quando a derivada da temperatura é quase nula. Definimos portanto o conjunto das possíveis saídas com relação a três sensores virtuais como sendo:

$$Y_1 = \{T_{oa}, T_{od}\}$$

$$Y_2 = \{Err+, Err-\}$$

$$Y_3 = \{Err > 0, Err \gg 0, Err \sim 0\}$$

A notação dos sensores virtuais é mostrada na tabela 4.5. Sendo: $P = \{VI, VC, VO, TI, TL, TH, c0, c1, c2, c3, VS, VSO\}$ o conjunto de lugares de N_t . A posição onde os sensores virtuais serão adicionados na rede N estão definidas pelas chamadas *Marcações Genéricas* [20]. As marcações genéricas são vetores de marcação básicos os quais representam as marcações alcançáveis em N_t . Mediante o uso de marcações genéricas, é possível reduzir o numero de marcações sem comprometer o mapeamento correto de sensores. Nos vetores de marcação genéricos, os números representam as marcações no lugar correspondente que terão que ser considerados para o mapeamento, quando aparece *, significa que não importa o número de marcações no lugar. Definamos o vetor de marcações de N_t como $M = [M(VI) M(VC) \cdot \cdot \cdot]$, pode ser visto que as marcações genéricas para N_t são:

$$M_a = [* * * * 0 1 0 0 0 1 * *]$$

$$M'_a = [* * * * 1 0 0 0 1 0 * *]$$

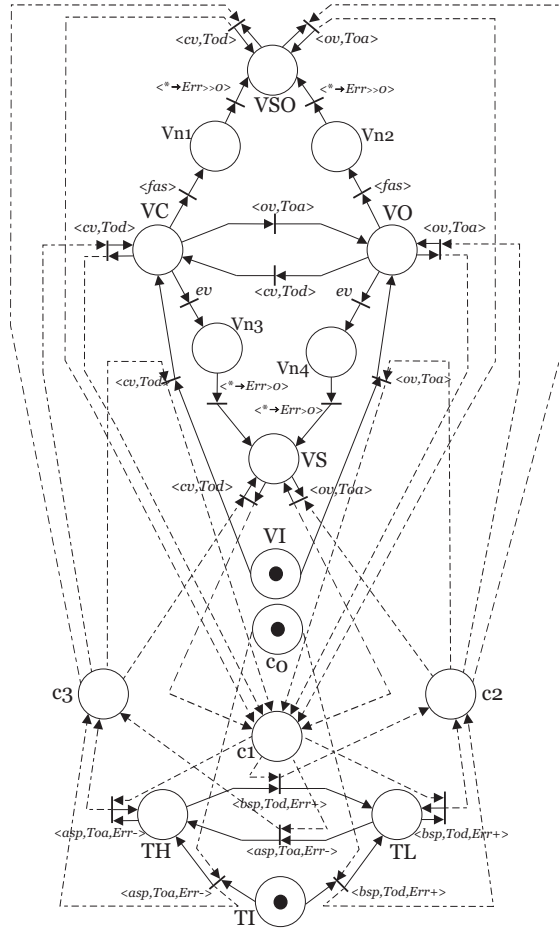


Figura 4.6: Rede N_{map} , após do mapeamento de sensores em N

$$M_b = [* 1 0 * * * 0 1 * * * *]$$

$$M'_a = [* 0 1 * * * 0 1 * * * *]$$

$$M_c = [* * * * * * 0 * * * 1 0]$$

$$M'_c = [* * * * * * 0 * * * 0 1]$$

O mapeamento dos sensores em função das marcações $M_i \in \mathcal{R}(M_0)$ de N_t é dado pela Tabela 4.6.

Observe-se que uma leitura T_{oa}/T_{od} no sensor S_1 indica que é possível registrar ambas as saídas, porém dado que o sensor só tem duas saídas, o mapeamento na marcação M'_c com respeito ao sensor S_1 não é considerado. Como resultado, o mapeamento global de sensores é a união do mapeamento respeito a cada sensor virtual.

Do procedimento descrito para Mapeamento de Sensores em Re-

Tabela 4.6: Mapeamento de sensores no trocador de calor

	h_1	h_2	h_3
M_a	T_{oa}	$Err-$	$Err \sim 0$
M'_a	T_{od}	$Err+$	$Err \sim 0$
M_b	T_{od}	$Err-$	$Err \sim 0$
M'_b	T_{oa}	$Err+$	$Err \sim 0$
M_c	T_{oa}	$Err-$	$Err \gg 0$
M'_c	T_{oa}/T_{od}	$Err - / Err +$	$Err > 0$

des de Petri (procedimento 4) obtemos a rede N_{tmap} (Figura 4.6) definida por: $N_{tmap} = \{P_m, T_m, Pre_m, Post_m, \Sigma_m, l_m, M_{m0}\}$, com $P_m = \{VI, VC, VO, TI, TL, TH, c0, c1, c2, c3, Vn1, Vn2, Vn3, Vn4, VS, VSO\}$. Note que $\langle N_t, M_0 \rangle \cong \langle N_{tmap}, M_{m0} \rangle$, as marcações de N_{tmap} conservam a informação do comportamento de N_t , dado que, tal como foi mostrado no Fato 3, N_t e N_{tmap} são compatíveis com $P : \Sigma_m^* \rightarrow \Sigma^*$.

O autômato $G_{tmapDIAG}$ (Figura 4.7) diagnostica o comportamento de N_{tmap} , por análise da linguagem gerada, e inspeção dos estados do diagnosticador (Teorema 1), se pode notar que $G_{tmapDIAG}$ não contém ciclos indeterminados, porém N_{tmap} é diagnosticável, sendo possível inferir a ocorrência de uma falha após um atraso finito de observações.

Da inspeção do diagnosticador $G_{tmapDIAG}$ note-se a presença de cinco malhas A, B, C, D e E. As malhas A, B e C representam a parte do diagnosticador correspondente às operações normais do trocador de calor. Suponha o registro dos eventos $\langle Err \sim 0 \rightarrow Err \gg 0 \rangle$ e $\langle asp, Tod, Err \rangle$, após disso o diagnosticador entra na malha D, que contém estados F_1 -certos, indicando a ocorrência da falha do tipo F_1 (falta de suprimento de ar). Pode-se notar que nesta malha o controlador tentará diminuir a temperatura através do fechamento da válvula, mas sem obter alguma resposta. Analogamente, quando o diagnosticador entra na malha E, que contém estados F_2 -certos, os estados do diagnosticador indicam que a válvula está em falha do tipo F_2 (emperramento da válvula).

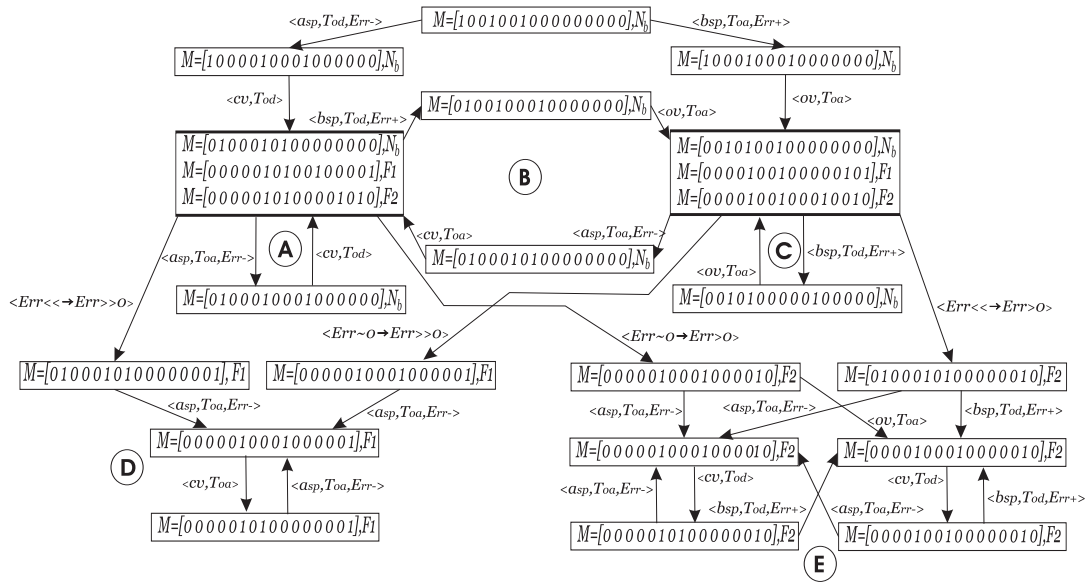


Figura 4.7: Diagnosticador $G_{tmapDIAG}$ da rede N_{map}

Tabela 4.7: Medições no Primeiro Estágio de Separação

Símbolo	Medição
LT-01	Transmissor de Nível de óleo no vaso separador SG-01
PT-01	Transmissor de Pressão de gás no vaso separador SG-01
FT-02	Transmissor de Fluxo de saída gás do vaso separador SG-01
FQI-01	Totalizador de Vazão de óleo de saída do vaso separador SG-01
FQI-02	Totalizador de Vazão de gás de saída do vaso separador SG-01

4.2.2 Primeiro Estágio de Separação:

Para o estudo dos diagnosticadores do primeiro estágio de separação, vamos considerar diagnosticadores nas duas principais malhas desta etapa do processo: Malha de controle de nível e Malha de controle de pressão, uma vez que para nossos efeitos, estas malhas podem ser tratadas como independentes. A tabela 4.7 indica as medições presentes no primeiro estágio de separação. A seguir se dá o projeto dos diagnosticadores:

4.2.2.1 Malha de Nível:

Estudaremos a malha de nível do primeiro estágio de separação visto na Seção 3.2.3. Esta etapa se encarrega de controlar o nível de óleo no vaso separador SG-01; e é composta pelo seguintes elementos: Um elemento atuador LV-01 (Válvula), um filtro, e um controlador de nível LIC-01 (Controlador). Será denominado *carga* às mudanças de nível causadas por perturbações externas, tais como golfadas no ingresso de gás ou óleo ao vaso separador SG-01 ou variações na vazão de saída de óleo do vaso separador. As redes de Petri para os três elementos são: Válvula N_{vl} , carga de pressão N_{cl} e controlador N_{pl} , são mostrados:

Válvula-Filtro:(Figura 4.8)

$$N_{vl} = (P_{vo}, T_{vo}, Pre_{vo}, Post_{vo}, \Sigma_{co}, l_{vo}, M_{vo0})$$

Onde:

$$P_{vo} = \{VI, VC, VO, VS, VSO, FL, FS, FE\}$$

$$T_{vo} = \{t_{v1}, t_{v2}, t_{v3}, t_{v4}, t_{v5}, t_{v6}, t_{v7}, t_{v8}, t_{v9}, t_{v10}, t_{v11}, t_{v12}, t_{v13}, t_{v14}, t_{v15}, t_{v16}, t_{v17}\}$$

$$\Sigma_{vo} = \{ov, cv, fsa, ev, opf, of\}$$

$$l_{vo} : T_{vo} \rightarrow \Sigma_{co} = \{cv, ov, ov, cv, ov, cv, ov, cv, fsa, fsa, cv, ov, ev, ev, opf, of, of\}$$

$$M_{vo0} = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.8 mostra a simbologia dos lugares e eventos da válvula LV-01.

Carga: (Figura 4.9)

$$N_{cl} = (P_{co}, T_{co}, Pre_{co}, Post_{co}, \Sigma_{co}, l_{co}, M_{co0})$$

Onde:

$$P_{co} = \{LI, LL, LH\}$$

$$T_{co} = \{t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5}, t_{c6}\}$$

$$\Sigma_{vo} = \{bsp, asp\}$$

$$l_{co} : T_{co} \rightarrow \Sigma_{co} = \{bsp, asp, bsp, asp, bsp, asp\}$$

$$M_{co0} = [1 \ 0 \ 0]$$

A tabela 4.9 mostra a simbologia dos lugares e eventos da perturbação o carga no nível de óleo.

Tabela 4.8: Válvula-Filtro do Controle de Nível do 1ro estágio de Separação

Símbolo	Descrição
<i>VI</i>	Válvula em estado Inicial
<i>VC</i>	Válvula fechando
<i>VO</i>	Válvula abrindo
<i>VS</i>	Válvula emperrada
<i>VSO</i>	Válvula emperrada totalmente aberta
<i>FL</i>	Filtro limpo
<i>FS</i>	Filtro sujo
<i>FE</i>	Filtro entupido
<i>ov</i>	abrir válvula
<i>vc</i>	fechar válvula
<i>fsa</i>	falta de suprimento de ar
<i>ev</i>	emperramento de válvula
<i>opf</i>	obstrução parcial do filtro
<i>otf</i>	obstrução total do filtro

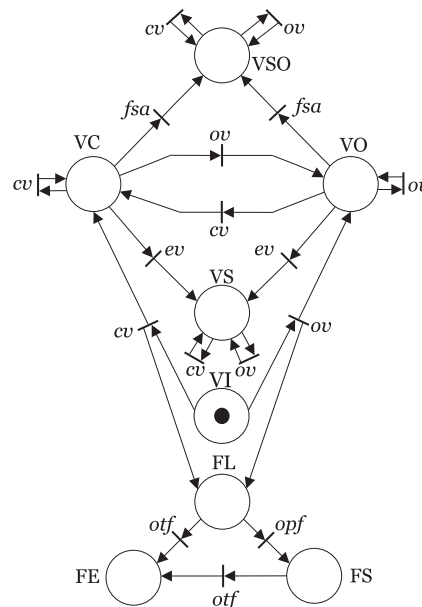


Figura 4.8: Válvula-Filtro de Óleo do primeiro estágio de separação

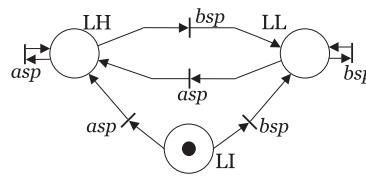


Figura 4.9: Carga de Perturbação de Nível no Vaso Separador SG-01

Tabela 4.9: Carga de Nível no Vaso Separador SG-01

Símbolo	Descrição
LI	Nível inicial
LL	Nível baixo
LH	Nível alto
bsp	Nível abaixo do setpoint
asp	Nível acima do setpoint

Controlador:(Figura 4.10)

$$N_{pl} = (P_{po}, T_{po}, B_{po}, w_{po}, \Sigma_{po}, l_{po}, M_{po0})$$

Onde:

$$P_{po} = \{c0, c1, c2, c3\}$$

$$T_{po} = \{t_{p1}, t_{p2}, t_{p3}, t_{p4}, t_{p5}, t_{p6}\}$$

$$\Sigma_{po} = \{ov, cv, bsp, asp\}$$

$$l_{po} : T_{po} \rightarrow \Sigma_{po} = \{ov, bsp, cv, asp, asp, bsp\}$$

$$M_{po0} = [1 \ 0 \ 0 \ 0]$$

A tabela 4.10 mostra a simbologia dos lugares e eventos do controlador de nível. O modelo que representa o comportamento completo da malha de nível no vaso separador SG-01 e é obtido a partir da composição paralela das redes N_{vl} , N_{pl} e N_{cl} , i.e., a rede $N_{ll} = N_{vl} || N_{cl} || N_{pl}$, onde $N_{ll} = \{P, T, Pre, Post, \Sigma, l, M_0\}$ é mostrada na figura 4.11.

O conjunto de lugares de N_{ll} é escrito como: $P = \{VI, VC, VO, LI, LH, LL, c0, c1, c2, c3, FL, VS, VSO, FE, FS\}$.

Como foi visto na seção 3.4, na malha de pressão existem quatro tipos de falhas, então definimos as partições falha: $\Sigma_{f1} = \{f_{sa}\}$ por falta de suprimento de ar que têm a posição de segurança pre-especificada "Totalmente Aberta", $\Sigma_{f2} = \{ev\}$

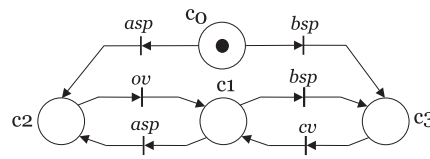


Figura 4.10: Controlador de Nível do primeiro estágio de separação

Tabela 4.10: Controlador de Nível do Primeiro estágio de separação

Símbolo	Descrição
$c0$	Controlador em estado 0
$c1$	Controlador em estado 1
$c2$	Controlador em estado 2
$c3$	Controlador em estado 3

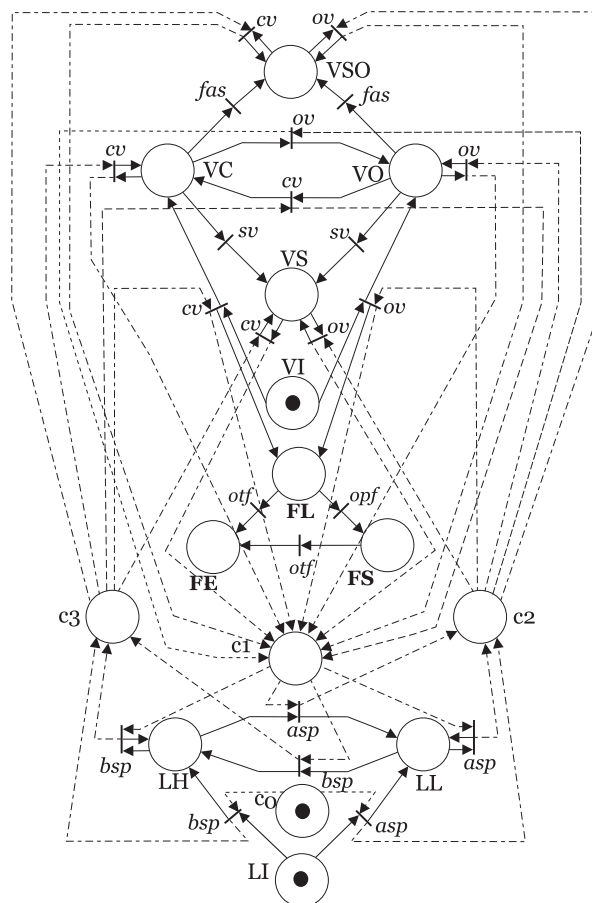


Figura 4.11: Rede N_u que representa a malha de Nível do Vaso Separador SG-01

Tabela 4.11: Sensores Virtuais do Controle de Nível no Primeiro Estágio de Separação

Símbolo	Descrição
N_{oa}	Nível de óleo aumentando.
N_{od}	Nível de óleo diminuindo.
$Err+$	Erro de nível positivo.
$Err-$	Erro de nível negativo.
Voa	Vazão de óleo aumentando.
Vod	Vazão de óleo diminuindo.
$Vo > 0$	Vazão de óleo maior que zero.
$Vo = 0$	Vazão de óleo menor que zero.
$Err \sim 0$	Erro de regime próximo a zero.
$Err \gg 0$	Erro de regime muito maior que zero.
$Err > 0$	Erro de regime pequeno.

falha por emperramento da válvula em alguma posição intermediária, $\Sigma_{f3} = \{otf\}$, falha por obstrução total do filtro e $\Sigma_{f2} = \{opf\}$ falha por obstrução parcial do filtro; para um conjunto de falhas $\Sigma = \{fas, sv, otf, opf\}$.

O transmissor de nível LT-01 e o totalizador FQI-02 serão usados no mapeamento de sensores da modelagem da malha de nível. A tabela 4.11 mostra a informação que pode ser obtida de sensores virtuais utilizando a leitura do transmissor PT-0, transmissor FT-02 e o totalizador FQI-01. Agora, definimos o conjunto das possíveis saídas com relação aos cinco sensores virtuais, sendo:

$$Y_1 = \{N_{oa}, N_{od}\}$$

$$Y_2 = \{Err+, Err-\}$$

$$Y_3 = \{Voa, Vod\}$$

$$Y_4 = \{Vo > 0, Vo = 0\}$$

$$Y_5 = \{Err \sim 0, Err \gg 0, Err > 0\}$$

Sendo: $P = \{VI, VC, VO, LI, LH, LL, c0, c1, c2, c3, FL, VS, VSO, FS, FE\}$ o conjunto de lugares de N_{ll} . A posição onde os sensores virtuais serão adicionados na rede N_{ll} estão definidas pelas marcações genéricas, daqui pode ser visto que as marcações genéricas para N_{ll} são:

$$M_a = [* * * * 1 0 0 0 1 0 1 * * * *]$$

Tabela 4.12: Mapeamento de Sensores

	h_1	h_2	h_3	h_4	h_5
M_a	N_{oa}	$Err-$	Voa/Vod	$Vo > 0$	$Err \sim 0$
M'_a	N_{od}	$Err+$	Voa/Vod	$Vo > 0$	$Err \sim 0$
M_b	N_{oa}	$Err+$	Vod	$Vo > 0$	$Err \sim 0$
M'_b	N_{od}	$Err-$	Voa	$Vo > 0$	$Err \sim 0$
M_c	N_{od}	$Err+$	Voa	$Vo > 0$	$Err \gg 0$
M'_c	N_{oa}/N_{od}	$Err - /Err+$	Voa/Vod	$Vo > 0$	$Err > 0$
M_d	N_{oa}/N_{od}	$Err - /Err+$	Vod	$Vo > 0$	$Err > 0$
M'_d	N_{oa}	$Err-$	$Vod = 0$	$Vo > 0$	$Err \gg 0$

$$M'_a = [* * * * 0 1 0 0 0 1 1 * * * *]$$

$$M_b = [* 1 0 * * * 0 1 * * 1 * * * *]$$

$$M'_a = [* 0 1 * * * 0 1 * * 1 * * * *]$$

$$M_c = [* * * * * * 0 * * * 1 1 0 * *]$$

$$M'_c = [* * * * * * 0 * * * 1 0 1 * *]$$

$$M_d = [* * * * * * 0 * * * 0 * * 1 *]$$

$$M'_d = [* * * * * * 0 * * * 0 * * * 1]$$

O mapeamento dos sensores em função das marcações $M_i \in \mathcal{R}(M_0)$ de N_{ll} é dado pela Tabela 4.12. Como resultado, o mapeamento global de sensores é a união do mapeamento com respeito a cada sensor virtual.

Do procedimento descrito para Mapeamento de Sensores em Redes de Petri (procedimento 4) obtemos a rede N_{llm} (Figura 4.12) definida por: $N_{ppm} = \{P_m, T_m, Pre_m, Post_m, \Sigma_m, l_m, M_{m0}\}$, com $P_m = \{VI, VC, VO, LI, LH, LL, c0, c1, c2, c3, FL, Vn1, Vn2, Vn3, Vn4, Vn5, VS, VSO, FS, FE\}$. O autômato G_{llDiag} (Figura 4.13) diagnostica o comportamento de N_{ppm} . Com uma análise dos estados do diagnosticador, segundo o teorema 1, pode-se concluir que G_{llDiag} não contém ciclos indeterminados, porém N_{llm} é diagnosticável. Por simplicidade, no diagrama de transição de estados de G_{llDiag} foram desenhados os estados contendo as marcações só observáveis, é dizer as marcações que correspondem aos lugares $Vn1, Vn2, Vn3, Vn4, Vn5$ não foram desenhadas.

Da inspeção de G_{llDIAG} , observe-se a presença de onze malhas, A, B, C, D, E, F, G, H, I, J e K. As malhas A, B e C representam a parte do diagnosticador cor-

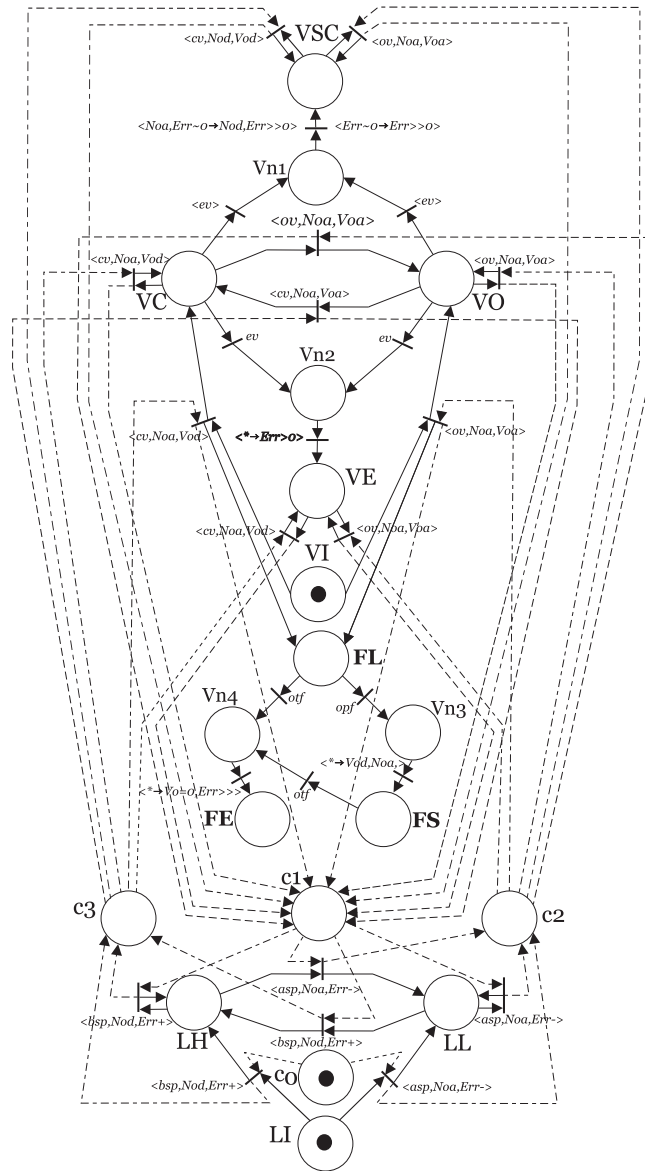


Figura 4.12: Modelo que representa a malha de Nível do Vaso Separador SG-01 com o mapeamento de Sensores

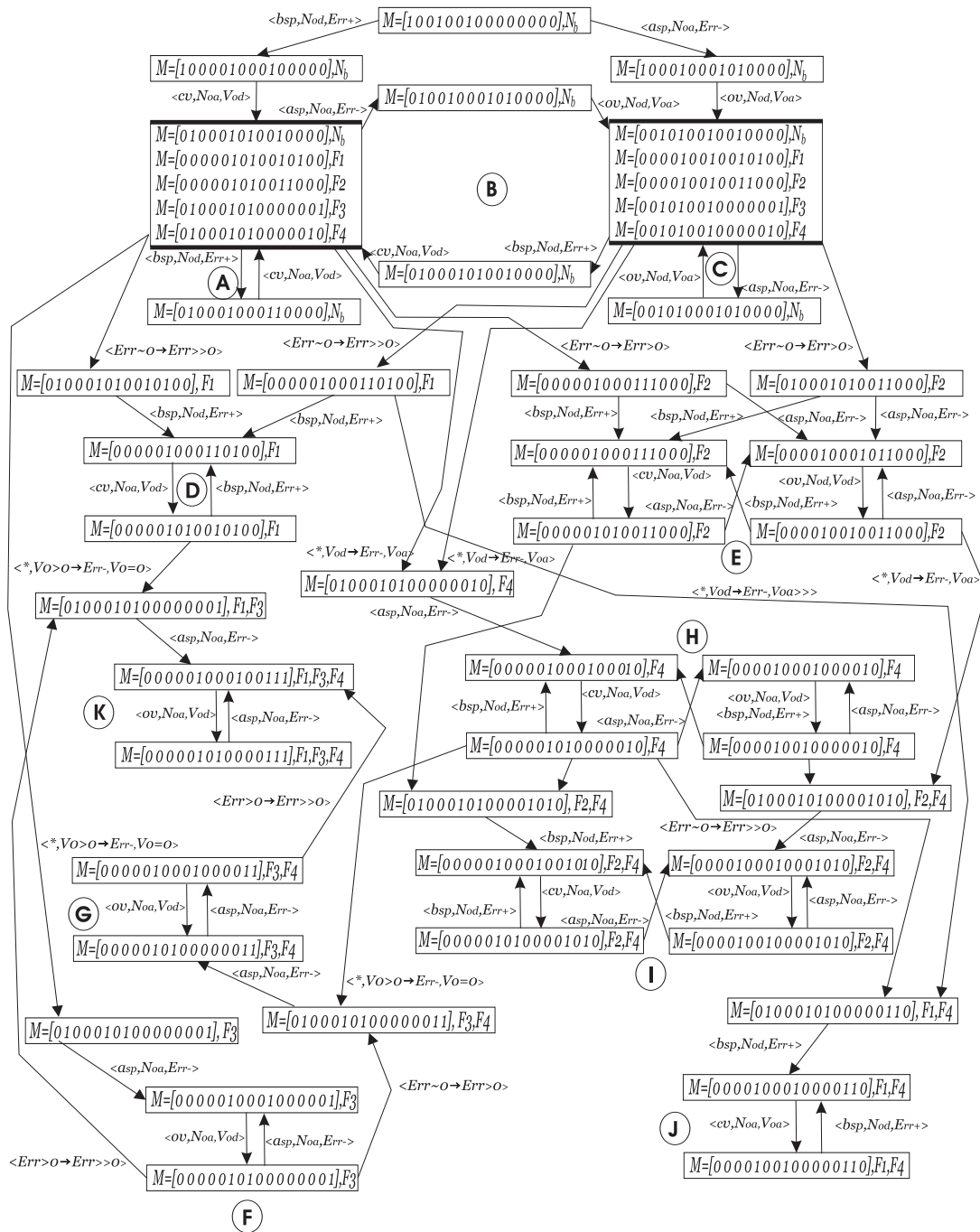


Figura 4.13: Diagnosticador G_{UDiag} da rede N_{Um}

respondente às operações normais da malha de controle de nível. Note-se que, ante o registro do evento $\langle *, V_o > 0 \rightarrow Err \rangle$ o diagnosticador entra na malha F, que contém estados F_3 -certos, que indica a ocorrência da falha tipo F_3 (desligamento indevido da válvula). Logo do registro do evento $\langle Err \rangle 0 \rightarrow Err \gg 0 \rangle$ o diagnosticador entra na malha K, que contém estados F_1, F_3, F_4 -certos, que indicam a ocorrência de falhas de esses tipos (falta de suprimento de ar, filtro entupido e filtro sujo). Pela modelagem feita, uma vez que o diagnosticador inferiu a ocorrência da falha filtro entupido, ele pode inferir também a ocorrência da falha por filtro sujo (que seria lógico), a partir das seqüências de eventos registrados.

4.2.2.2 Malha de Pressão:

Nesta seção, consideremos a malha de pressão do primeiro estágio de separação visto na Seção 3.2.4. Esta etapa é importante para controlar a pressão de gás no interior do vaso separador SG-01, e é composta pelo seguintes elementos: Um elemento atuador PV-01 (Válvula), e um controlador de pressão PIC-01 (Controlador). Será denominado *carga* as mudanças de pressão causadas por perturbações externas como golfadas no ingresso de gás ou óleo ao vaso separador SG-01 ou variações na vazão de saída de óleo do vaso separador. As redes de Petri para os três elementos: Válvula N_{vp} , carga de pressão N_{cp} e controlador N_{pp} , são mostrados:

Válvula:(Figura 4.14)

$$N_{vp} = \{P_v, T_v, Pre_v, Post_v, \Sigma_v, l_v, M_{v0}\}$$

Onde:

$$P_v = \{VI, VO, VC, VSO, VE\}$$

$$T_v = \{t_{v1}, t_{v2}, t_{v3}, t_{v4}, t_{v5}, t_{v6}, t_{v7}, t_{v8}, t_{v9}, t_{v10}, t_{v11}, t_{v12}, t_{v13}, t_{v14}\}$$

$$\Sigma_v = \{ov, cv, fsa, ev\}$$

$$l : T_v \rightarrow \Sigma_v = \{cv, ov, ov, cv, ov, cv, ov, cv, fsa, fsa, cv, ov, ev, ev\}$$

$$M_{v0} = [1 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.13 mostra a simbologia dos lugares e eventos da válvula PV-01.

Carga:(Figura 4.15)

$$N_{cp} = \{P_c, T_c, Pre_c, Post_c, \Sigma_c, l_c, M_{c0}\}$$

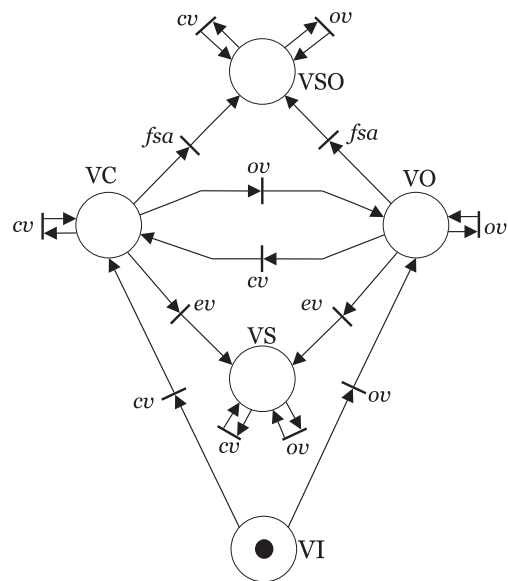


Figura 4.14: Rede de Petri $N_{v,p}$ que representa a Modelagem da válvula PV-01

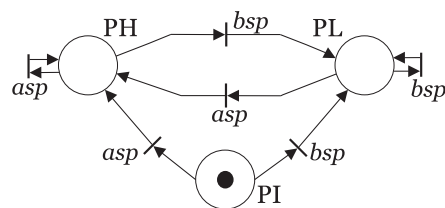


Figura 4.15: Rede de Petri N_{cp} que representa a carga na malha de pressão

Onde:

$$P_c = \{TI, TL, TH\}$$

$$T_c = \{t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5}, t_{c6}\}$$

$$\Sigma_c = \{bsp, asp\}$$

$$l_c : T_c \rightarrow \Sigma_c = \{bsp, asp, bsp, asp, bsp, asp\}$$

$$M_{c0} = [1 \ 0 \ 0]$$

A tabela 4.14 mostra a simbologia dos lugares e eventos da perturbação o carga de pressão.

Controlador:(Figura 4.16)

Tabela 4.13: Válvula de pressão do primeiro estágio de separação

Símbolo	Descrição
<i>VI</i>	Válvula em estado Inicial
<i>VC</i>	Válvula fechando
<i>VO</i>	Válvula abrindo
<i>VS</i>	Válvula emperrada
<i>VSO</i>	Válvula emperrada totalmente aberta
<i>ov</i>	abrir válvula
<i>vc</i>	fechar válvula
<i>fsa</i>	falta de suprimento de ar
<i>ev</i>	emperramento de válvula

Tabela 4.14: Carga de pressão no primeiro estágio de separação

Símbolo	Descrição
<i>PI</i>	Pressão inicial
<i>PL</i>	Pressão baixa
<i>PH</i>	Pressão alta
<i>bsp</i>	Pressão abaixo do setpoint
<i>asp</i>	Pressão acima do setpoint

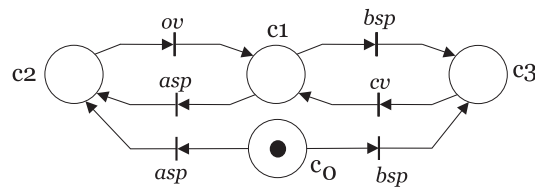
Figura 4.16: Rede de Petri N_{pp} que representa a modelagem do controlador TIC-02

Tabela 4.15: Controlador do TIC-02

Símbolo	Descrição
c_0	Controlador em estado 0
c_1	Controlador em estado 1
c_2	Controlador em estado 2
c_3	Controlador em estado 3

$$N_{pp} = \{P_p, T_p, Pre_p, Post_p, \Sigma_p, l_p, M_{p0}\}$$

Onde:

$$P_p = \{c0, c1, c2, c3\}$$

$$T_p = \{t_{p1}, t_{p2}, t_{p3}, t_{p4}\}$$

$$\Sigma_p = \{ov, cv, bsp, asp\}$$

$$l_p : T_p \rightarrow \Sigma_p = \{ov, bsp, cv, asp\}$$

$$M_{c0} = [1 \ 0 \ 0 \ 0]$$

A tabela 4.15 mostra a simbologia dos lugares e eventos do controlador de pressão. O modelo que representa o comportamento completo da malha de pressão é obtido a partir da composição paralela das redes N_{vp} , N_{pp} e N_{cp} , i.e., a rede $N_p = N_{vp} \parallel N_{cp} \parallel N_{pp}$ com $N_p = \{P, T, Pre, Post, \Sigma, l, M_0\}$ é mostrada na figura 4.17.

Como foi visto na seção 3.4, na malha de pressão existem dois conjuntos de falhas, então definimos as partições $\Sigma_{f1} = \{f_{sa}\}$ falha por falta de suprimento de ar que têm a posição de segurança pre-especificada "Totalmente Aberta" e $\Sigma_{f2} = \{ev\}$ falha por emperramento da válvula em alguma posição intermediária; para um conjunto de falhas $\Sigma = \{fas, sv\}$.

O transmissor de pressão PT-01 e o totalizador FQI-02 serão usados no mapeamento de sensores da modelagem da malha de pressão. A tabela 4.16 mostra a informação pode ser obtida de sensores virtuais utilizando a leitura do transmissor PT-01 e o totalizador FQI-02. Daqui, definimos o conjunto das possíveis saídas com relação aos sensores virtuais:

$$Y_1 = \{P_{ga}, P_{gd}\}$$

$$Y_2 = \{Err+, Err-\}$$

$$Y_3 = \{Err \sim 0, Err \gg 0\}$$

Sendo: $P = \{VI, VC, VO, PI, PH, PL, c0, c1, c2, c3, VSO, VS\}$ o conjunto de lugares de N_p . A posição onde os sensores virtuais serão adicionados na rede N_p estão definidas pelas marcações genéricas. Consideremos o vetor de marcações de

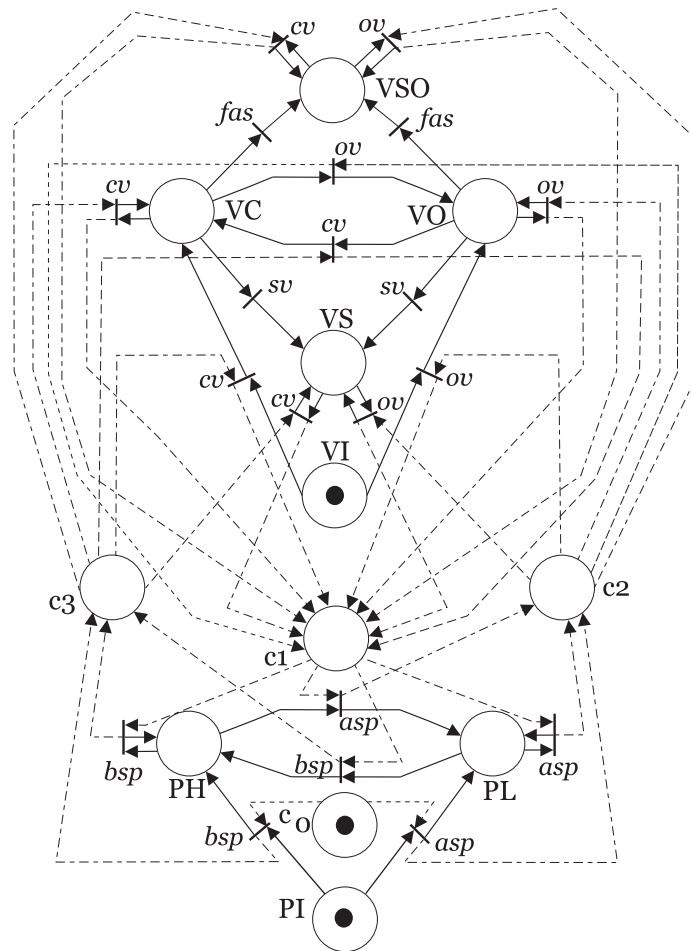


Figura 4.17: Rede de Petri N da malha de pressão do 1ro. estágio de separação

Tabela 4.16: Sensores Virtuais da malha de pressão

Símbolo	Descrição
P_{oa}	Pressão aumentando.
P_{od}	Pressão descendo.
$Err+$	Erro de pressão positivo.
$Err-$	Erro de pressão negativo.
$Err \gg 0$	Erro de regime próximo a zero.
$Err \ll 0$	Erro de regime muito maior que zero.
$Err \sim 0$	Erro de regime pequeno.

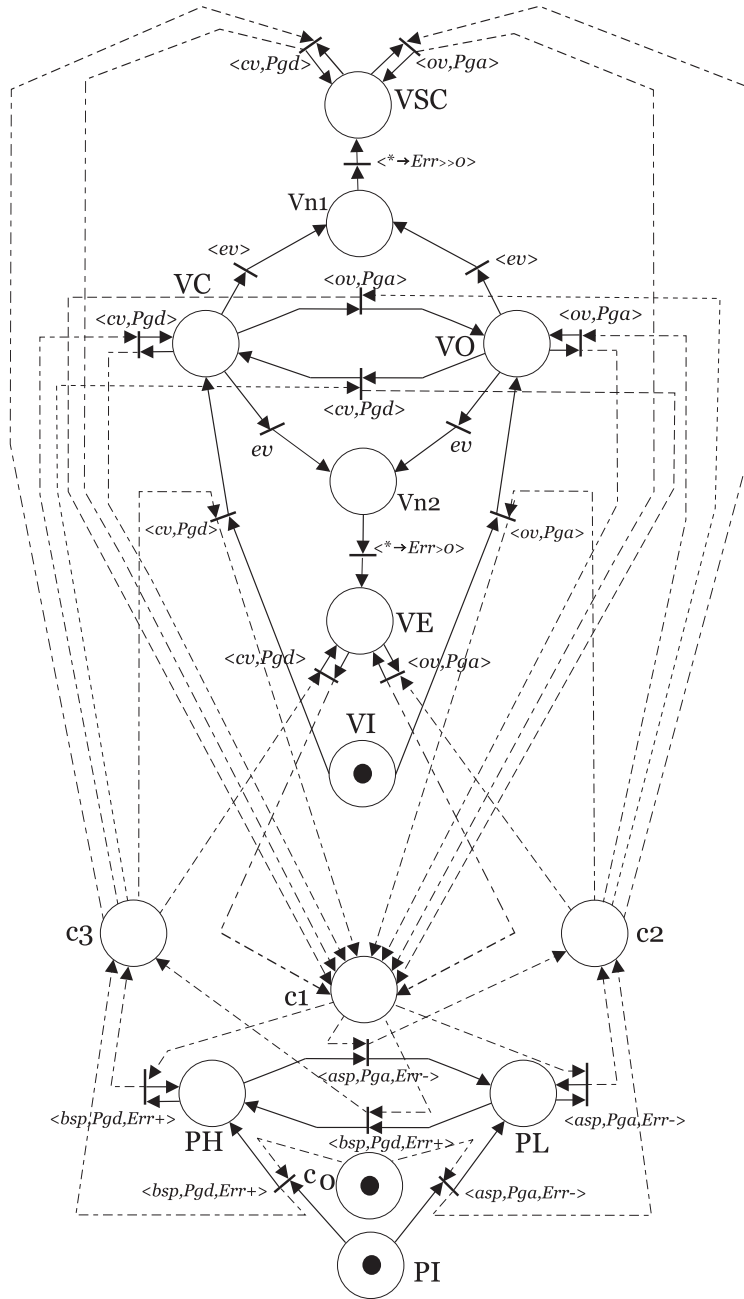


Figura 4.18: Rede N_{ppm} , após do mapeamento de sensores em N_p

Tabela 4.17: Mapeamento de Sensores

	h_1	h_2	h_3
M_a	P_{ga}	$Err-$	$Err \sim 0$
M'_a	P_{gd}	$Err+$	$Err \sim 0$
M_b	P_{ga}	$Err+$	$Err \sim 0$
M'_b	P_{gd}	$Err-$	$Err \sim 0$
M_c	P_{gd}	$Err+$	$Err \gg 0$
M'_c	P_{ga}/P_{gd}	$Err - /Err+$	$Err > 0$
M_d	P_{ga}/P_{gd}	$Err - /Err+$	$Err > 0$
M'_d	P_{ga}	$Err-$	$Err \gg 0$

N_p como $M = [M(VI) \ M(VC) \ \cdot \cdot \cdot]$, pode ser visto que as marcações genéricas para N_p são:

$$M_a = [* \ * \ * \ * \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ * \ *]$$

$$M'_a = [* \ * \ * \ * \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ * \ *]$$

$$M_b = [* \ 1 \ 0 \ * \ * \ * \ 0 \ 1 \ * \ * \ * \ *]$$

$$M'_b = [* \ 0 \ 1 \ * \ * \ * \ 0 \ 1 \ * \ * \ * \ *]$$

$$M_c = [* \ * \ * \ * \ * \ * \ 0 \ * \ * \ * \ 1 \ 0]$$

$$M'_c = [* \ * \ * \ * \ * \ * \ 0 \ * \ * \ * \ 0 \ 1]$$

O mapeamento dos sensores em função das marcações $M_i \in \mathcal{R}(M_0)$ de N é dado pela Tabela 4.17. O mapeamento global de sensores é a união do mapeamento respeito a cada sensor virtual.

Do procedimento descrito para Mapeamento de Sensores em Redes de Petri (procedimento 4) obtemos a rede N_{ppm} (Figura 4.18) definida por: $N_{ppm} = \{P_m, T_m, Pre_m, Post_m, \Sigma_m, l_m, M_{m0}\}$, com $P = \{VI, VC, VO, PI, PH, PL, c0, c1, c2, c3, Vn1, Vn2, VS, VSO\}$. O autômato G_{ppDIAG} (Figura 4.19) diagnostica o comportamento de N_{ppm} . Na análise do diagnosticador $G_{mapDIAG}$, segundo teorema 1, podemos ver que não contém ciclos indeterminados, porém N_{ppm} é diagnosticável. Por simplicidade, no diagnosticador não são mostradas as marcações de $Vn1$ nem de $Vn2$.

4.2.3 Segundo Estágio de Separação:

O segundo estágio de separação compreende as etapas de separação no vaso SG-02 e a etapa de re-circulação do óleo, estas detalham-se nesta seção pelas malhas de

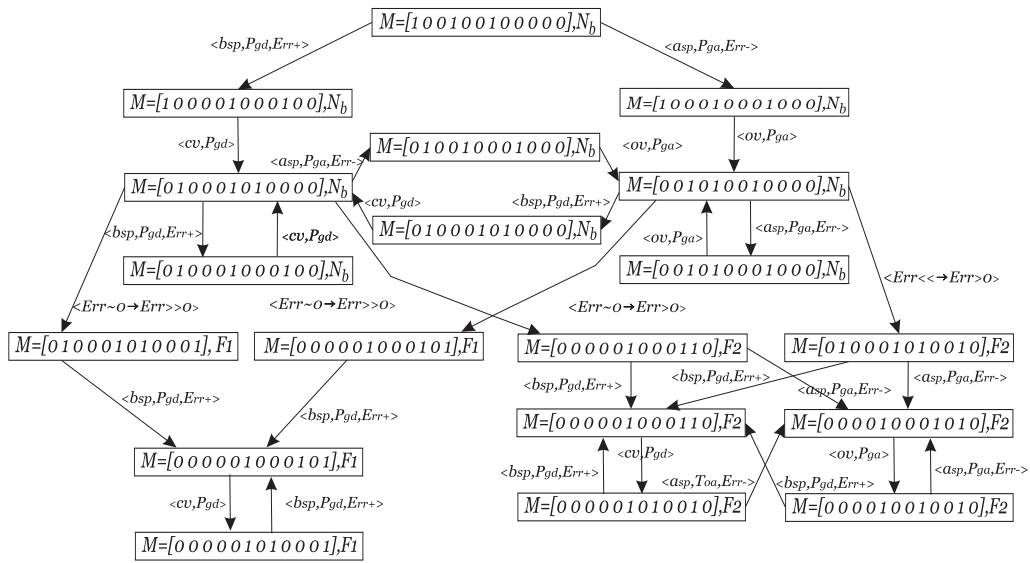


Figura 4.19: Diagnosticador G_{ppDIAG} da rede N_{ppm}

Tabela 4.18: Medições no Segundo Estágio de Separação e circuito de re-circulação

Símbolo	Medição
LT-02	Transmissor de Nível de óleo no vaso separador SG-02
PT-02	Transmissor de Pressão de gás no vaso separador SG-02
FT-03	Transmissor de Fluxo de saída óleo do vaso separador SG-02
FQI-04	Totalizador de Vazão de óleo de saída do vaso separador SG-01

controle de nível, a malha de controle de pressão e a malha de controle de fluxo. A tabela 4.18 indica as medições presentes no segundo estágio de separação e circuito de exportação de óleo. A seguir se dá o projeto dos diagnosticadores:

4.2.3.1 Malha de Nível:

Estudaremos a malha de nível do segundo estágio de separação visto na Seção 3.2.4. Esta etapa realiza o controle de nível do óleo no vaso separador SG-02; e contém os seguintes elementos: Um atuador LV-02 (Válvula), uma bomba BO-1 e um controlador de nível LIC-02 (Controlador). Denominaremos *carga* as mudanças de nível causadas por perturbações externas, tais como golfadas no ingresso de gás ou óleo ao vaso separador SG-02 e à entrada de óleo proveniente da etapa de re-circulação. As redes de Petri para os três elementos são: Válvula N_{vls} , carga de

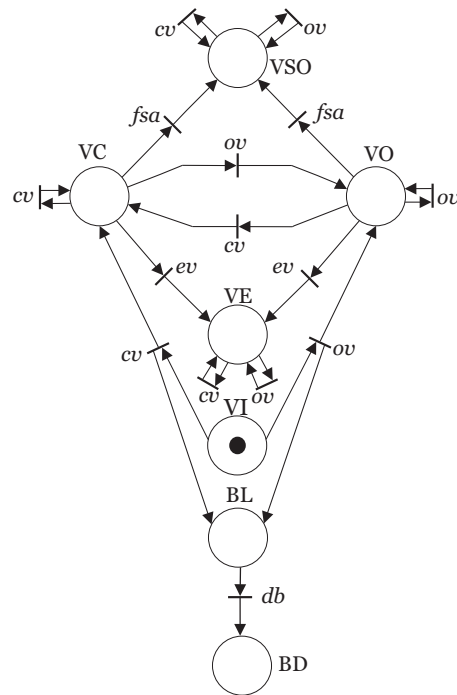


Figura 4.20: Válvula-Bomba de Óleo do segundo estágio de separação

pressão N_{cls} e controlador N_{pls} , são mostrados:

Válvula-Bomba:(Figura 4.20)

$$N_{vls} = (P_{vo}, T_{vo}, Pre_{vo}, Post_{vo}, \Sigma_{co}, l_{vo}, M_{vo0})$$

Onde:

$$P_{vo} = \{VI, VC, VO, VE, VSO, BL, BD\}$$

$$T_{vo} = \{t_{v1}, t_{v2}, t_{v3}, t_{v4}, t_{v5}, t_{v6}, t_{v7}, t_{v8}, t_{v9}, t_{v10}, t_{v11}, t_{v12}, t_{v13}, t_{v14}, t_{v15}\}$$

$$\Sigma_{vo} = \{ov, cv, fsa, ev, db\}$$

$$l_{vo} : T_{vo} \rightarrow \Sigma_{co} = \{cv, ov, ov, cv, ov, cv, ov, cv, fsa, fsa, cv, ov, ev, ev, db\}$$

$$M_{vc0} = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.19 mostra a simbologia dos lugares e eventos da válvula LV-02 e bomba BO-1.

Carga: (Figura 4.21)

$$N_{cls} = (P_{co}, T_{co}, Pre_{co}, Post_{co}, \Sigma_{co}, l_{co}, M_{co0})$$

Onde:

$$P_{co} = \{LI, LL, LH\}$$

Tabela 4.19: Válvula do Controle de Nível do 2do estágio de Separação

Símbolo	Descrição
<i>VI</i>	Válvula em estado Inicial
<i>VC</i>	Válvula fechando
<i>VO</i>	Válvula abrindo
<i>VE</i>	Válvula emperrada
<i>VSO</i>	Válvula emperrada totalmente aberta
<i>BL</i>	Bomba ligada
<i>BD</i>	Bomba desligada
<i>ov</i>	abrir válvula
<i>vc</i>	fechar válvula
<i>fsa</i>	falta de suprimento de ar
<i>ev</i>	emperramento de válvula
<i>db</i>	desligamento da válvula

Tabela 4.20: Carga de Nível no Vaso Separador SG-02

Símbolo	Descrição
<i>LI</i>	Nível inicial
<i>LL</i>	Nível baixo
<i>LH</i>	Nível alto
<i>bsp</i>	Nível abaixo do setpoint
<i>asp</i>	Nível acima do setpoint

$$T_{co} = \{t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5}, t_{c6}\}$$

$$\Sigma_{vo} = \{bsp, asp\}$$

$$l_{co} : T_{co} \rightarrow \Sigma_{co} = \{bsp, asp, bsp, asp, bsp, asp\}$$

$$M_{co0} = [1 \ 0 \ 0]$$

A tabela 4.20 mostra a simbologia dos lugares e eventos da perturbação ou carga no nível de óleo.

Controlador:(Figura 4.22)

$$N_{lse} = (P_{po}, T_{po}, B_{po}, w_{po}, \Sigma_{po}, l_{po}, M_{po0})$$

Onde:

$$P_{po} = \{c0, c1, c2, c3\}$$

$$T_{po} = \{t_{p1}, t_{p2}, t_{p3}, t_{p4}, t_{p5}, t_{p6}\}$$

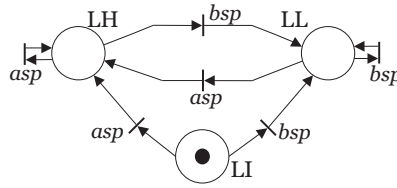


Figura 4.21: Carga de Perturbação de Nível no Vaso Separador SG-02

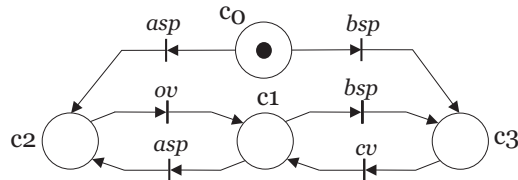


Figura 4.22: Controlador de Nível do segundo estágio de separação

$$\Sigma_{po} = \{ov, cv, bsp, asp\}$$

$$l_{po} : T_{po} \rightarrow \Sigma_{po} = \{ov, bsp, cv, asp, asp, bsp\}$$

$$M_{po0} = [1 \ 0 \ 0 \ 0]$$

A tabela 4.21 mostra a simbologia dos lugares e eventos do controlador de nível. O modelo que representa o comportamento completo da malha de nível no vaso separador SG-02 e é obtido a partir da composição paralela das redes N_{vls} , N_{pls} e N_{cls} , i.e., a rede $N_{lse} = N_{vls} || N_{cls} || N_{pls}$, com $N_{lse} = \{P, T, Pre, Post, \Sigma, l, M_0\}$ é mostrada na figura 4.23.

Como foi visto na seção 3.4, na malha de nível de existem três tipos de falhas. Definiremos as partições falha: $\Sigma_{f1} = \{f_{sa}\}$ falta de suprimento de ar na válvula que têm a posição de segurança pre-especificada "Totalmente Aberta", $\Sigma_{f2} = \{ev\}$ falha por emperramento da válvula em alguma posição intermediária, e $\Sigma_{f3} = \{db\}$, falha por desligamento indevido da bomba BO-1. Definimos o conjunto de falhas

Tabela 4.21: Controlador de Nível do segundo estágio de separação

Símbolo	Descrição
c0	Controlador em estado 0
c1	Controlador em estado 1
c2	Controlador em estado 2
c3	Controlador em estado 3

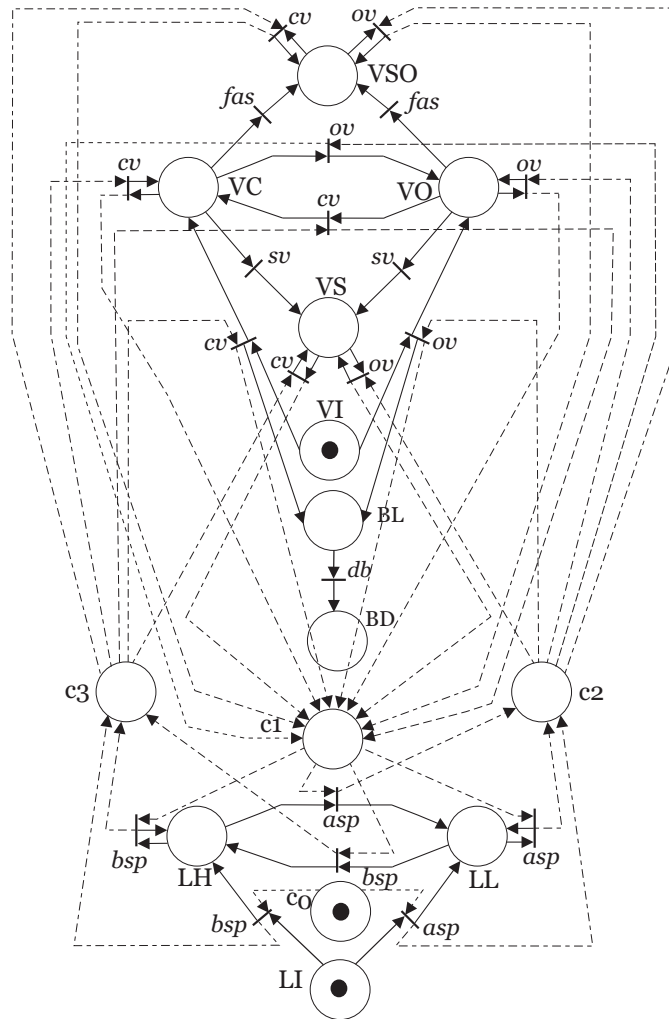


Figura 4.23: Rede N_{lse} que representa a malha de Nível do Vaso Separador SG-02

$$\Sigma_f = \{fas, sv, db\}.$$

O transmissor de nível LT-02, o transmissor de fluxo FT-03 e o totalizador FQI-04 serão usados no mapeamento de sensores da modelagem da malha de nível. A tabela 4.22 mostra a informação que pode ser obtida de sensores virtuais utilizando os equipamentos de medição mencionados. Agora, definimos o conjunto das possíveis saídas com relação aos sensores virtuais, sendo:

$$Y_1 = \{N_{oa}, N_{od}\}$$

$$Y_2 = \{Err+, Err-\}$$

$$Y_3 = \{FP, NF\}$$

Tabela 4.22: Sensores Virtuais do Controle de Nível no Segundo Estágio de Separação

Símbolo	Descrição
N_{oa}	Nível de óleo aumentando.
N_{od}	Nível de óleo descendo.
$Err+$	Erro de nível positivo.
$Err-$	Erro de nível negativo.
FP	Fluxo positivo.
NF	Não fluxo.
$Err \sim 0$	Erro de regime próximo a zero.
$Err \gg 0$	Erro de regime muito maior que zero.
$Err > 0$	Erro de regime pequeno.

$$Y_4 = \{Err \sim 0, Err \gg 0, Err > 0\}$$

Sendo: $P = \{VI, VC, VO, LI, LH, LL, c0, c1, c2, c3, BL, VS, VSO, BD\}$ o conjunto de lugares de N_{lse} . A posição onde os sensores virtuais serão adicionados na rede N_{lse} estão definidas pelas marcações genéricas, onde pode ser visto que as marcações genéricas para N_{lse} são:

$$M_a = [* * * * 0 1 0 0 0 1 1 * * *]$$

$$M'_a = [* * * * 1 0 0 0 1 0 1 * * *]$$

$$M_b = [* 1 0 * * * 0 1 * * 1 * * *]$$

$$M'_a = [* 0 1 * * * 0 1 * * 1 * * *]$$

$$M_c = [* * * * * * 0 * * * 1 1 0 *]$$

$$M'_c = [* * * * * * 0 * * * 1 0 1 *]$$

$$M_d = [* * * * * * 0 * * * 0 * * 1]$$

O mapeamento dos sensores em função das marcações $M_i \in \mathcal{R}(M_0)$ de N_{lse} é dado pela Tabela 4.23.

Segundo o procedimento descrito para Mapeamento de Sensores em Redes de Petri (procedimento 4) obtemos a rede N_{lsm} (Figura 4.24) definida por: $N_{lsm} = \{P_m, T_m, Pre_m, Post_m, \Sigma_m, l_m, M_{m0}\}$, com $P_m = \{VI, VC, VO, LI, LH, LL, c0, c1, c2, c3, BL, Vn1, Vn2, Vn3, VS, VSO, BD\}$. O autômato G_{lsDiag} (Figura 4.25) diagnostica o comportamento de N_{lsm} , e fácil ver que G_{lsDiag} não contém ciclos indeterminados, sendo desta maneira uma rede que gera uma linguagem diagnosticável. Por simplicidade, no diagrama de transição

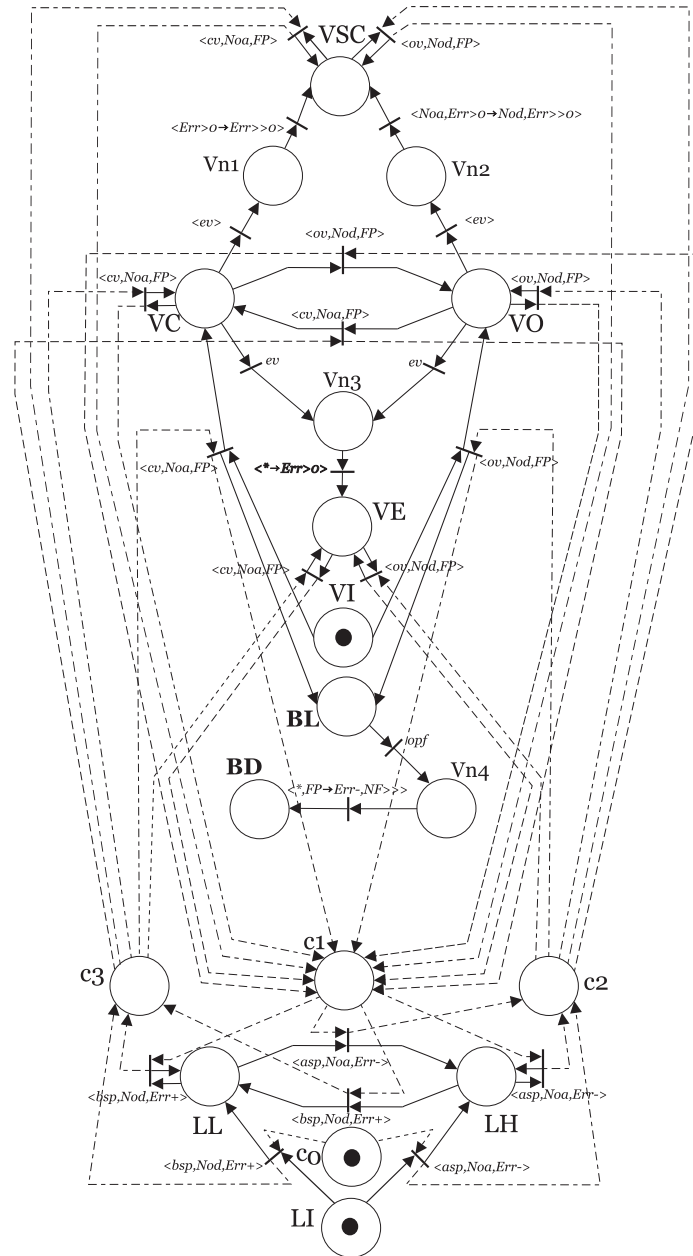


Figura 4.24: Modelo que representa a malha de Nível do Vaso Separador SG-02 com o mapeamento de Sensores

Tabela 4.23: Mapeamento de sensores na malha de nível

	h_1	h_2	h_3	h_4
M_a	N_{oa}	$Err-$	FP	$Err \sim 0$
M'_a	N_{od}	$Err+$	FP	$Err \sim 0$
M_b	N_{oa}	$Err+$	FP	$Err \sim 0$
M'_b	N_{od}	$Err-$	FP	$Err \sim 0$
M_c	N_{od}	$Err+$	FP	$Err \gg 0$
M'_c	N_{oa}/N_{od}	$Err- / Err+$	FP	$Err > 0$
M_d	N_{oa}/N_{od}	$Err- / Err+$	FP	$Err > 0$
M'_d	N_{oa}	$Err-$	FP	$Err \gg 0$

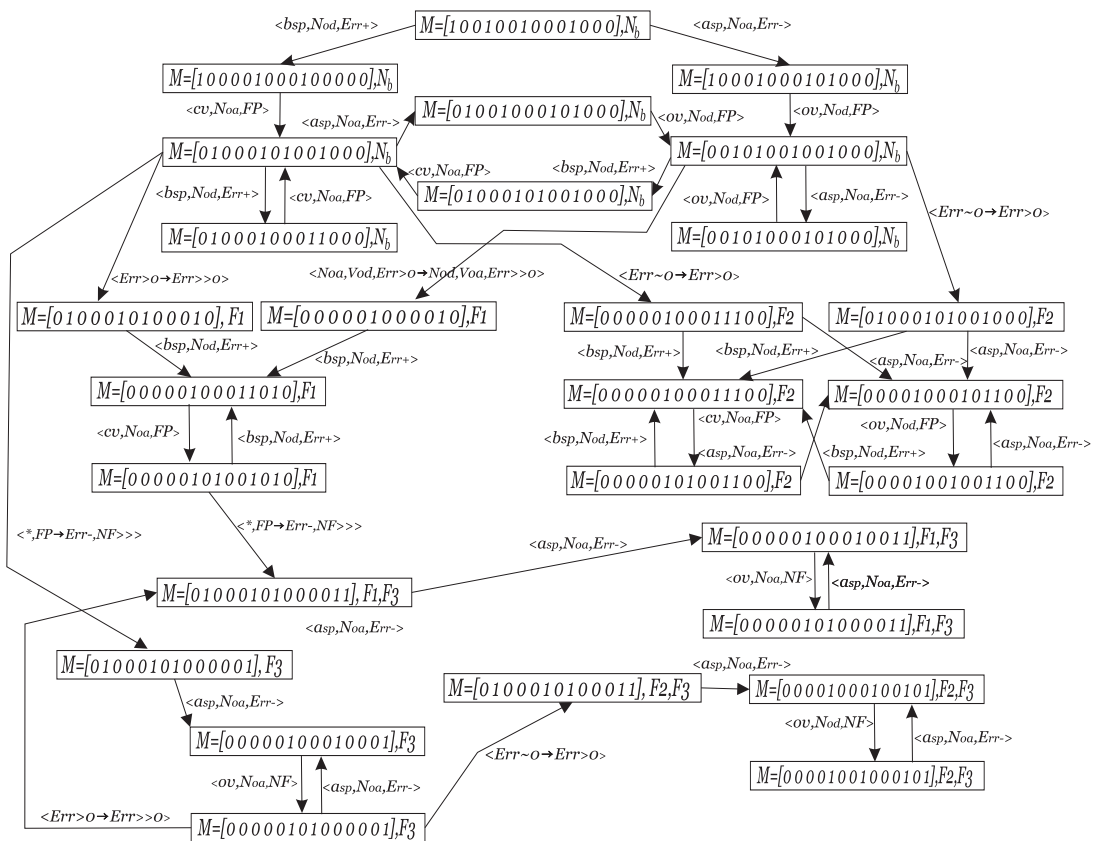


Figura 4.25: Diagnosticador G_{lsDiag} da rede N_{lsm}

de estados de G_{lsDiag} foram desenhados os estados contendo as marcações só observáveis, isto é, as marcações que correspondem aos lugares $Vn1, Vn2, Vn3$ não foram desenhadas, mas supõe-se que são zero.

4.2.3.2 Malha de Fluxo:

Nesta seção, estudaremos a malha de fluxo da etapa de exportação de óleo visto na Seção 3.2.4. Esta etapa encarrega-se de controlar o fluxo de óleo que é exportado e é composta pelo seguintes elementos: Um atuador FV-03 (Válvula), e um controlador de fluxo FIC-03 (Controlador). Será denominado *carga* as mudanças de fluxo causadas por perturbações externas, tais como golfadas no ingresso de gás ou óleo ao vaso separador SG-02 ou variações na vazão de saída de óleo do vaso separador. Cabe ressaltar que a atuação do controlador é ativada pelo ação do sensor FSL (sensor de fluxo baixo) e desligado quando atuar o sensor FSH (sensor de fluxo alto). As redes de Petri para os três elementos são: Válvula N_{vfs} , carga de pressão N_{cfs} e controlador N_{pfs} , são mostrados a continuação:

Válvula:(Figura 4.26)

$$N_{vfs} = (P_{vo}, T_{vo}, Pre_{vo}, Post_{vo}, \Sigma_{co}, l_{vo}, M_{vo0})$$

Onde:

$$P_{vo} = \{VI, VC, VO, VE, VSO\}$$

$$T_{vo} = \{t_{v1}, t_{v2}, t_{v3}, t_{v4}, t_{v5}, t_{v6}, t_{v7}, t_{v8}, t_{v9}, t_{v10}, t_{v11}, t_{v12}, t_{v13}, t_{v14}\}$$

$$\Sigma_{vo} = \{ov, cv, fsa, ev\}$$

$$l_{vo} : T_{vo} \rightarrow \Sigma_{co} = \{cv, ov, ov, cv, ov, cv, ov, cv, fsa, fsa, cv, ov, ev, ev\}$$

$$M_{vc0} = [1 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.24 mostra a simbologia dos lugares e eventos da válvula FV-03.

Carga: (Figura 4.27)

$$N_{cfs} = (P_{co}, T_{co}, Pre_{co}, Post_{co}, \Sigma_{co}, l_{co}, M_{co0})$$

Onde:

$$P_{co} = \{FI, FL, FH\}$$

$$T_{co} = \{t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5}, t_{c6}\}$$

$$\Sigma_{vo} = \{bsp, asp\}$$

Tabela 4.24: Válvula do Controle de fluxo do 2do estágio de Separação

Símbolo	Descrição
<i>VI</i>	Válvula em estado Inicial
<i>VC</i>	Válvula fechando
<i>VO</i>	Válvula abrindo
<i>VS</i>	Válvula emperrada
<i>VSO</i>	Válvula emperrada totalmente aberta
<i>ov</i>	abrir válvula
<i>vc</i>	fechar válvula
<i>fsa</i>	falta de suprimento de ar
<i>ev</i>	emperramento de válvula

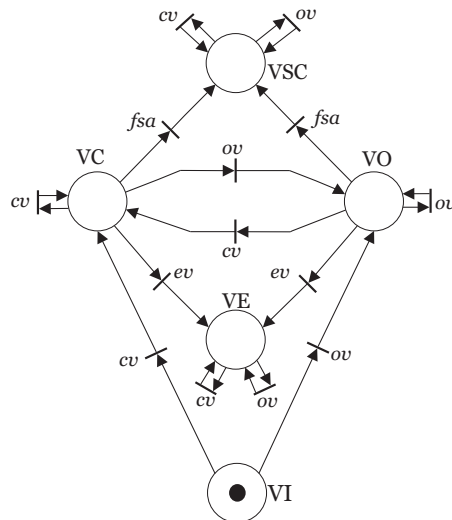


Figura 4.26: Válvula de controle de fluxo de óleo do circuito de exportação

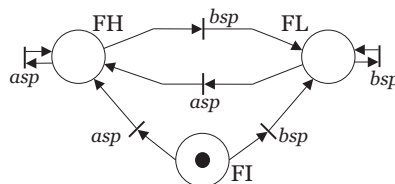


Figura 4.27: Carga de Perturbação de Fluxo do circuito de exportação

Tabela 4.25: Carga de Fluxo da etapa de exportação

Símbolo	Descrição
FI	Fluxo inicial
FL	Fluxo baixo
FH	Fluxo alto
bsp	Fluxo abaixo do setpoint
asp	Fluxo acima do setpoint

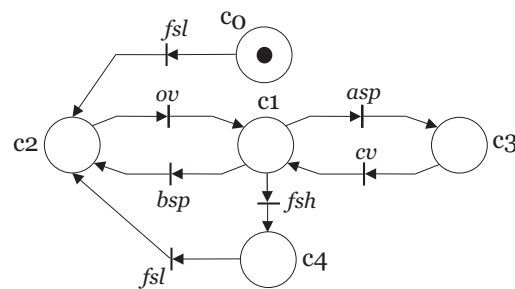


Figura 4.28: Controlador de fluxo da etapa de exportação de óleo

$$l_{co} : T_{co} \rightarrow \Sigma_{co} = \{bsp, asp, bsp, asp, bsp, asp\}$$

$$M_{co0} = [1 \ 0 \ 0]$$

A tabela 4.25 mostra a simbologia dos lugares e eventos da carga no controle de fluxo da etapa de exportação.

Controlador:(Figura 4.28)

$$N_{pfs} = (P_{po}, T_{po}, B_{po}, w_{po}, \Sigma_{po}, l_{po}, M_{po0})$$

Tabela 4.26: Controlador de fluxo da etapa de exportação de óleo

Símbolo	Descrição
$c0$	Controlador em estado 0
$c1$	Controlador em estado 1
$c2$	Controlador em estado 2
$c3$	Controlador em estado 3
$c3$	Controlador em estado 4
fsl	limite inferior de fluxo
fsh	limite superior de fluxo

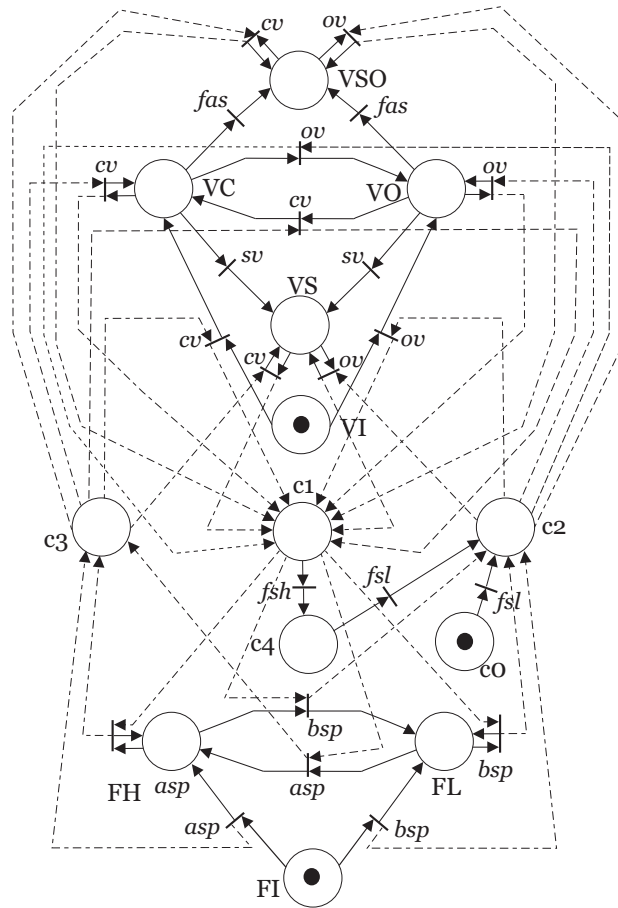


Figura 4.29: Rede N_{fs} que representa a malha de fluxo de exportação

Onde:

$$P_{po} = \{c0, c1, c2, c3, c4\}$$

$$T_{po} = \{t_{p1}, t_{p2}, t_{p3}, t_{p4}, t_{p5}, t_{p6}, t_{p7}\}$$

$$\Sigma_{po} = \{ov, cv, bsp, asp, fsl, fsh\}$$

$$l_{po} : T_{po} \rightarrow \Sigma_{po} = \{ov, bsp, cv, asp, asp, bsp\}$$

$$M_{po0} = [1 \ 0 \ 0 \ 0 \ 0]$$

A tabela 4.26 mostra a simbologia dos lugares e eventos do controlador de fluxo. O modelo completo que representa o comportamento da malha de fluxo é obtido a partir da composição paralela das redes N_{vfs} , N_{pfs} e N_{cfs} , i.e., a rede $N_{fs} = N_{vfs} \parallel N_{cfs} \parallel N_{pfs}$, com $N_{fs} = \{P, T, Pre, Post, \Sigma, l, M_0\}$ é mostrada na figura 4.29.

Como foi visto na seção 3.4, na malha de fluxo existem dois tipos de falhas, então definimos as partições $\Sigma_{f1} = \{f_{sa}\}$ falha por falta de suprimento de ar

Tabela 4.27: Sensores Virtuais do Controle de Fluxo de exportação

Símbolo	Descrição
N_{oa}	Fluxo de óleo aumentando.
N_{od}	Fluxo de óleo diminuindo.
$Err+$	Erro de Fluxo positivo.
$Err-$	Erro de Fluxo negativo.
$Err \sim 0$	Erro de regime próximo a zero.
$Err \gg 0$	Erro de regime muito maior que zero.
$Err > 0$	Erro de regime pequeno.

Tabela 4.28: Mapeamento de Sensores

	h_1	h_2	h_3
M_a	N_{oa}	$Err-$	$Err \sim 0$
M'_a	N_{od}	$Err+$	$Err \sim 0$
M_b	N_{oa}	$Err+$	$Err \sim 0$
M'_b	N_{od}	$Err-$	$Err \sim 0$
M_c	N_{od}	$Err+$	$Err \gg 0$
M'_c	N_{oa}/N_{od}	$Err- / Err+$	$Err > 0$

na válvula que têm a posição de segurança pre-especificada "Totalmente Aberta", $\Sigma_{f2} = \{ev\}$ falha por emperramento da válvula em alguma posição intermediária; para um conjunto de falhas $\Sigma_f = \{fas, sv\}$.

O transmissor de fluxo FT-03 e o totalizador FQI-03 serão usados no mapeamento de sensores da modelagem da malha de fluxo de re-circulação. A tabela 4.27 mostra os sensores virtuais obtidos, utilizando a leitura do transmissor FT-03 e o totalizador FQI-01. Agora, definimos o conjunto das possíveis saídas dos sensores virtuais:

$$Y_1 = \{F_{oa}, F_{od}\}$$

$$Y_2 = \{Err+, Err-\}$$

$$Y_3 = \{Err \sim 0, Err \gg 0, Err > 0\}$$

Sendo: $P = \{VI, VC, VO, FI, FL, FH, c0, c1, c2, c3, VSO, VS\}$ o conjunto de lugares de N_{lfs} . A posição onde os sensores virtuais serão adicionados na rede N_{lfs} estão definidas pelas marcações genéricas, daqui pode ser visto que as marcações genéricas para N_{lfs} são:

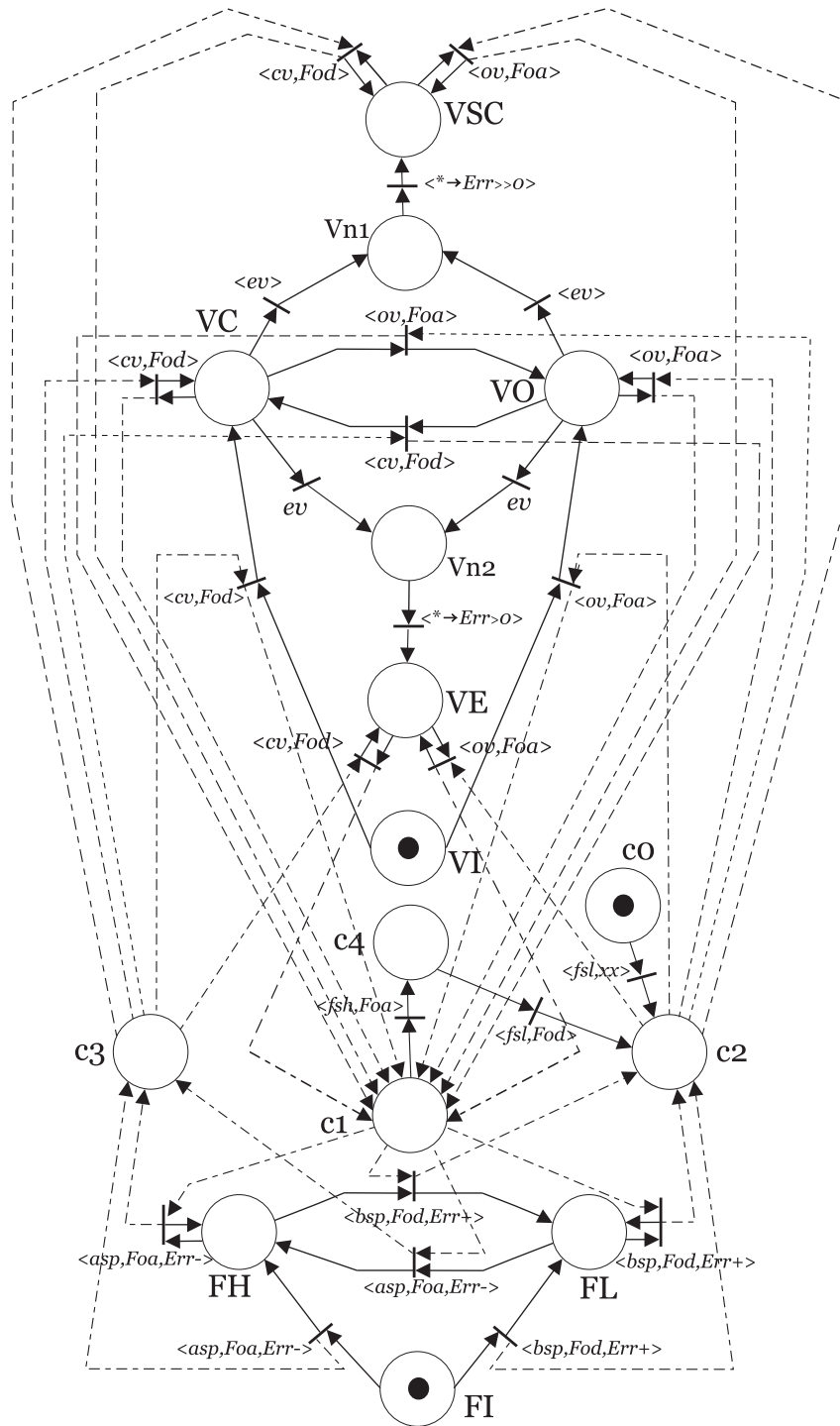
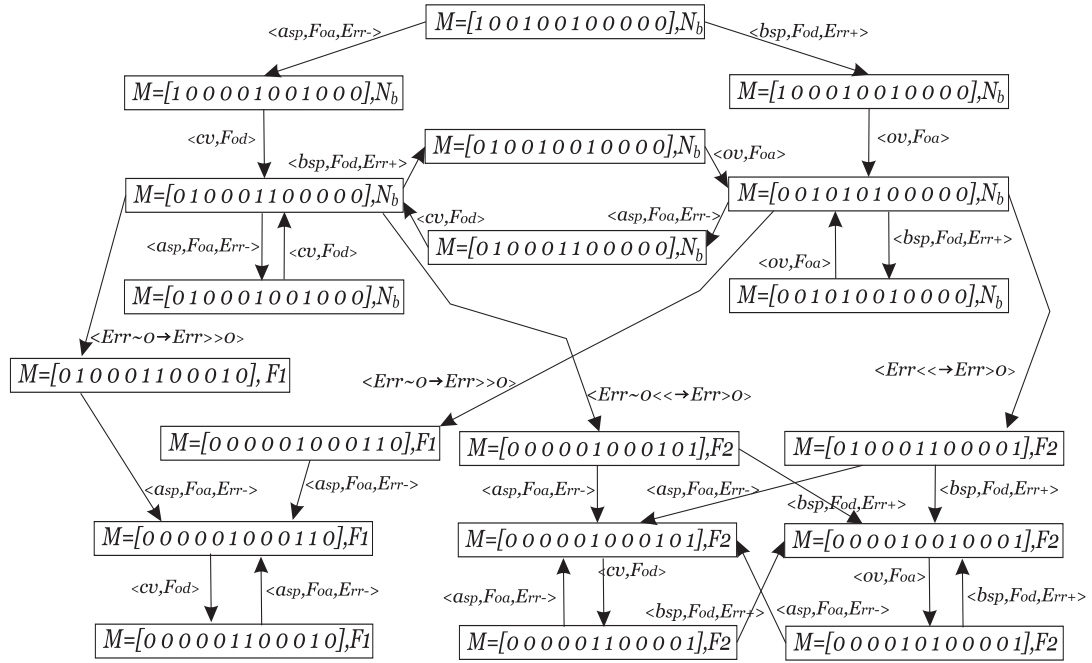


Figura 4.30: Modelo que representa a malha de fluxo da etapa de exportação

Figura 4.31: Diagnosticador G_{lfDiag} da rede N_{lfs}

$$M_a = [* * * * 0 1 0 0 0 1 * *]$$

$$M'_a = [* * * * 1 0 0 0 1 0 * *]$$

$$M_b = [* 1 0 * * * 0 1 * * * *]$$

$$M'_a = [* 0 1 * * * 0 1 * * * *]$$

$$M_c = [* * * * * * 0 * * * 1 0]$$

$$M'_c = [* * * * * * 0 * * * 0 1]$$

O mapeamento dos sensores em função das marcações $M_i \in \mathcal{R}(M_0)$ de N_{lfs} é dado pela Tabela 4.28. Do procedimento descrito para mapeamento de sensores em redes de Petri (procedimento 4), obtemos a rede N_{lfm} (Figura 4.30) definida por: $N_{lfm} = \{P_m, T_m, Pre_m, Post_m, \Sigma_m, l_m, M_{m0}\}$, com $P_m = \{VI, VC, VO, FI, FL, FH, c0, c1, c2, c3, c4, Vn1, Vn2, VSO, VS\}$. O autômato G_{lfDiag} (Figura 4.31) diagnostica o comportamento de N_{lfm} , e fácil ver que G_{lfDiag} não contém ciclos indeterminados, porém N_{lfm} é diagnosticável. Note-se que o sistema de diagnóstico da malha de controle de pressão do vaso separador SG-02 é análogo à malha de pressão do primeiro estágio de separação (Vaso SG-01), motivo pelo qual não foi detalhado nesta seção.

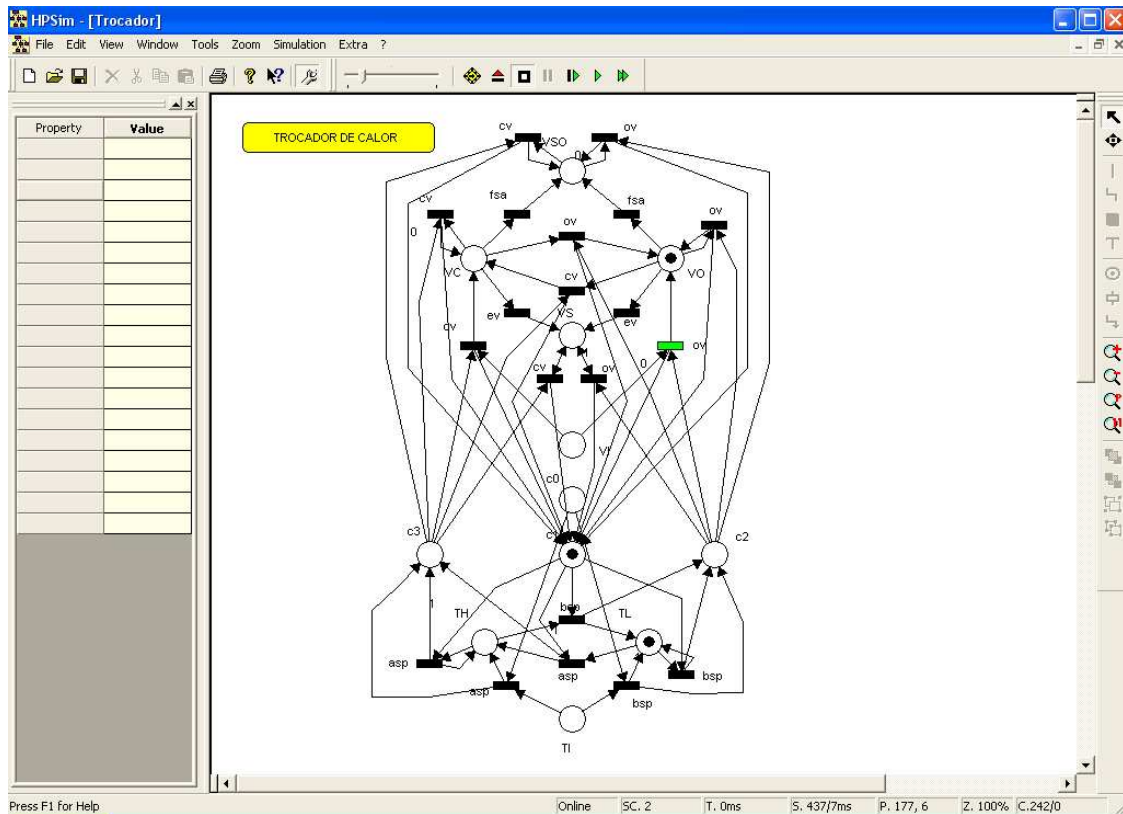


Figura 4.32: Simulação do trocador de calor no HPSim

4.3 Validação dos Modelos

Após da modelagem dos componentes do processo de separação, é preciso avaliar e validar os modelos apresentados na seção anterior. A validação consiste em comprovar a lógica prevista dos modelos e verificar se cumprem algumas propriedades. Para a validação da lógica implementada nas redes de Petri apresentadas usou-se o HPSim como ferramenta de simulação, ver figura 4.32. O HPSim é um software para simulação de redes de Petri que apresenta uma interface intuitiva de fácil utilização. Entre suas vantagens esta a possibilidade do acompanhamento da evolução do estado da rede de uma forma gráfica, o que auxilia no desenvolvimento do modelo e na detecção de erros. Além do modelo básico de redes de Petri (Petri Lugar/Transição), o HPSim permite ainda a simulação de redes de Petri Temporais e redes de Petri estocásticas, além da utilização de arcos inibidores e habilitadores. A diferença entre os três tipos de rede de Petri está nas transições. Foram simuladas as redes obtidas da composição paralela dos sub-modelos, que compreende

as malhas de controle de temperatura no trocador de calor, malhas de controle de nível e pressão no primeiro estágio de separação, malhas de controle de nível, fluxo e pressão no segundo estágio de separação.

Além da comprovação da evolução esperada das redes, também foram analisadas algumas propriedades como vivacidade e limitabilidade. Se todas as transições de uma rede são vivas, então a rede de Petri nunca se bloqueia. Se a rede projetada consegue gerar mais marcações que a sua limitação lhe permite, então o modelo se considera errado. Estas propriedades também foram comprovadas na simulação das redes de Petri. Comprovou-se que as redes são k -limitadas, com $k=1$.

4.4 Simulações do Processo-Diagnosticador

Nesta seção, para efeitos de comprovar o funcionamento de um diagnosticador discreto acoplado a um sistema (contínuo), detalha-se a simulação conjunta de um subsistema contínuo do processo de separação óleo-gás com o seu respectivo diagnosticador de falhas (discreto). Considera-se o trocador de calor, modelado matematicamente e simulado no Capítulo 3, e o diagnosticador de falhas da malha de controle de temperatura, projetado na Seção 4.2.1. Estes modelos correspondem a uma modelagem de sistemas contínuos e um sistema a eventos discretos respectivamente. Note-se que o diagnosticador é produto de uma modelagem que inclui a extração de características qualitativas de grandezas quantitativas.

Como já foi descrito no Capítulo 4, a simulação do trocador de calor foi feita no Simulink do MATLAB. Para o diagnosticador de falhas usou-se o Stateflow como ferramenta de simulação. O Stateflow é um pacote (*toolbox*) de MATLAB que permite modelar sistemas a eventos discretos dentro do Simulink, utilizando cartas de estado (*statecharts*). As cartas de estado são uma generalização de autômatos ou máquinas de estados. O Stateflow possui uma única estrutura denominada Chart, que serve para representar um sistema a eventos discretos. O Chart pode conectar-se com outros blocos do Simulink, de tempo contínuo o discreto, para formar modelos de sistemas híbridos que podem ser muito úteis na análise do comportamento de tais sistemas por intermédio de simulações.

A figura 4.33 mostra o esquema de simulação em Simulink e Stateflow do trocador de calor junto, e seu diagnosticador de falhas.

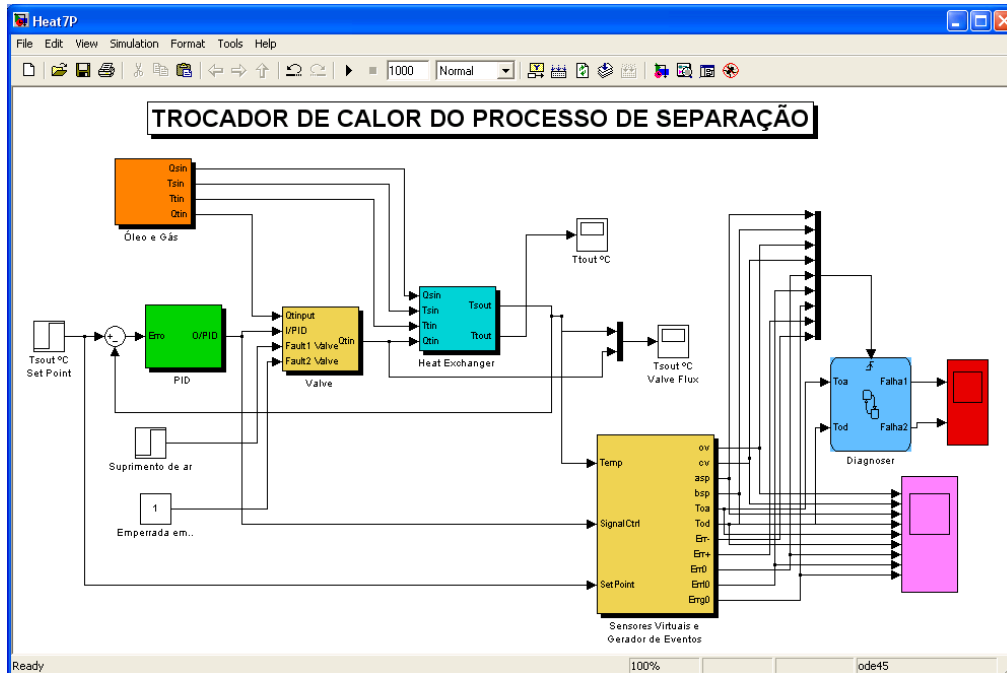


Figura 4.33: A figura mostra a ligação entre o simulador do trocador de calor e o diagnosticador realizado em Stateflow. ambos no ambiente Simulink do MATLAB.

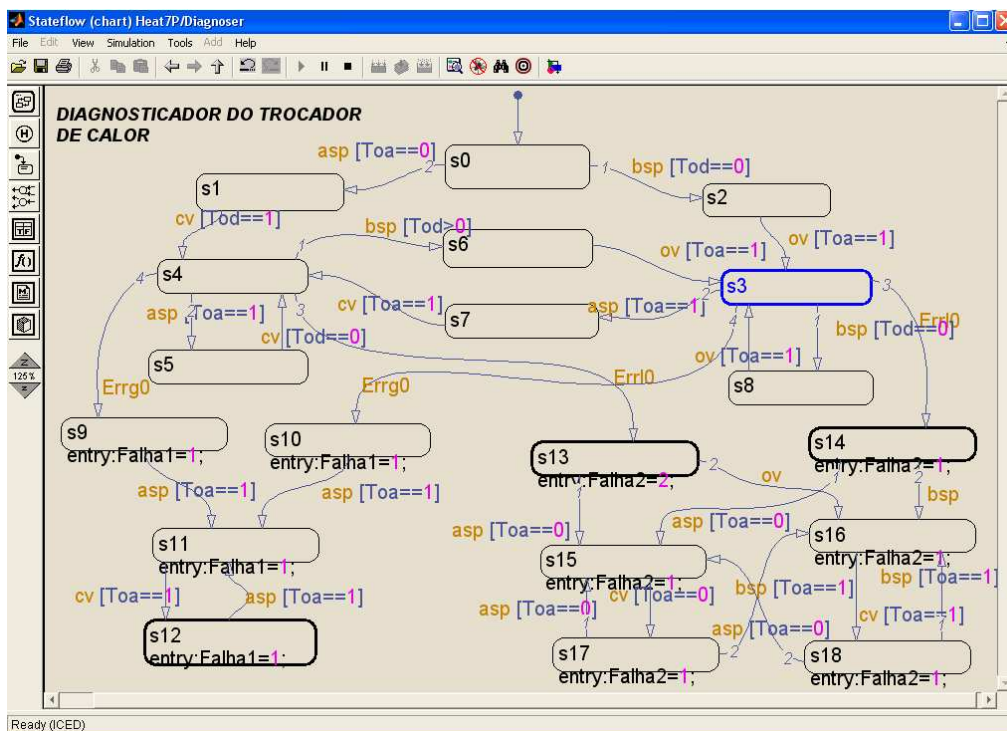


Figura 4.34: A figura mostra a tela do Stateflow correspondente ao diagnosticador $G_{tmapDIAG}$ (veja Figura 4.7 também).

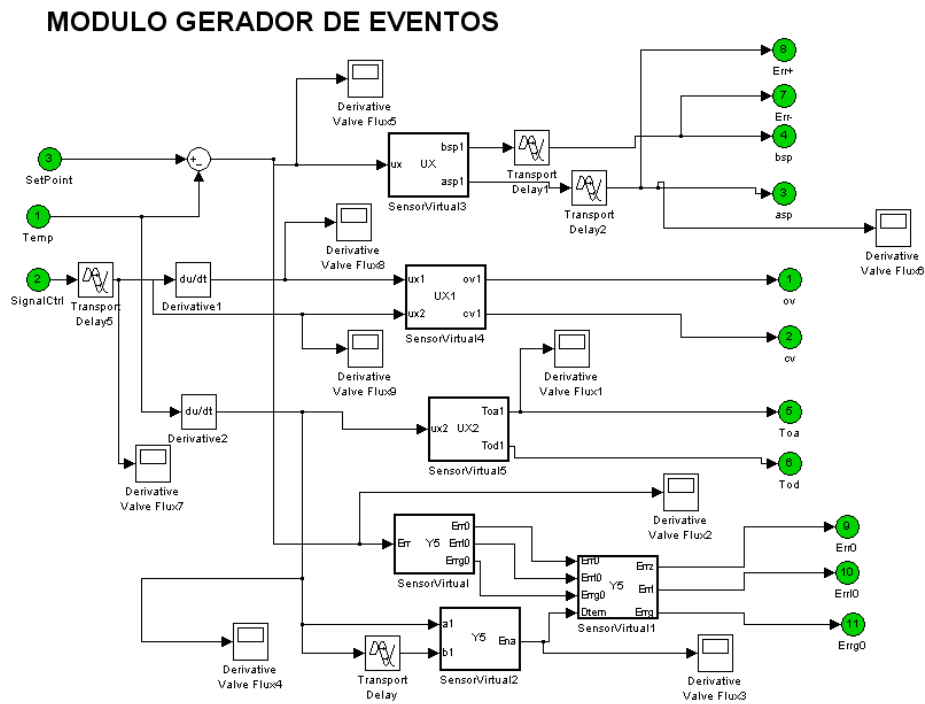


Figura 4.35: Simulação dos Sensores Virtuais e Gerador de Eventos no Simulink

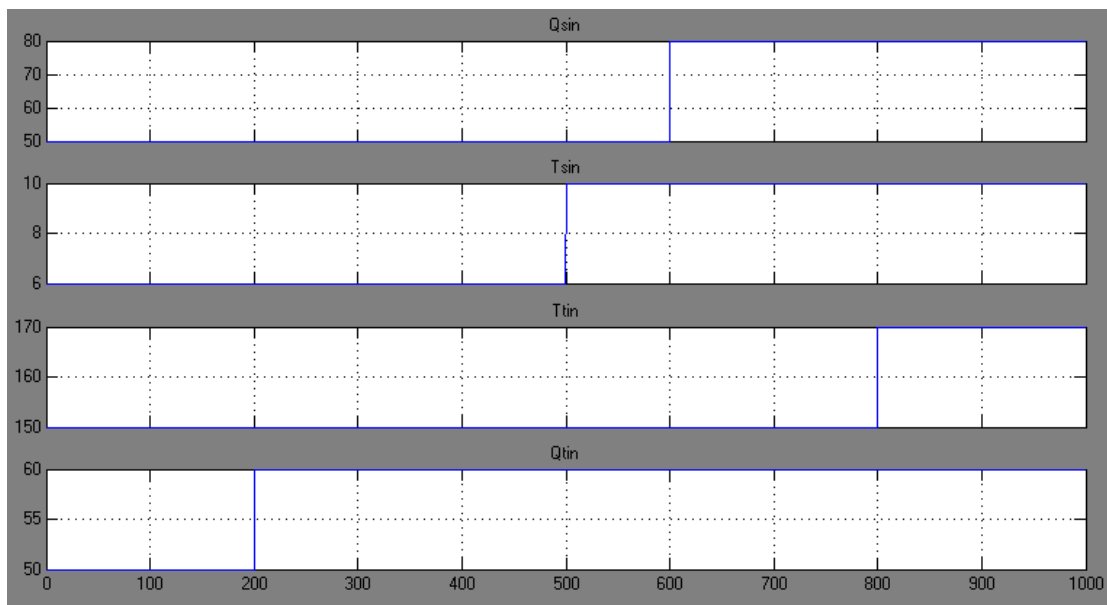


Figura 4.36: Simulação dos Sensores Virtuais e Gerador de Eventos no Simulink

Tabela 4.29: Sensores Virtuais do Trocador de calor

Símbolo	Descrição
T_{oa}	Temperatura de óleo de saída aumentando.
T_{od}	Temperatura de óleo de saída diminuindo.
$Err+$	Erro positivo.
$Err-$	Erro negativo.
$Err \sim 0$	Erro de regime próximo a zero.
$Err \gg 0$	Erro de regime muito maior que zero.
$Err > 0$	Erro de regime maior que zero.

4.4.1 Parâmetros da simulação:

As simulações foram feitas considerando distintas perturbações em um tempo de observação de 1000 segundos. Considerou-se que o trocador de calor como partindo de condições iniciais nulas com as seguintes entradas, figura 4.36:

Q_{Tinput} : Vazão de entrada no tubo (Vapor)

Q_{Sinput} : Vazão de entrada no casco (Óleo)

T_{Tin} : Temperatura de entrada ao tubo (Vapor)

T_{Sin} : Temperatura de entrada do casco (Óleo)

Com:

$$Q_{Tinput} = 50 \text{ (em 0 segs)} + 20 \text{(em 200 segs)} \text{ (m}^3\text{/h)}.$$

$$Q_{Sinput} = 50 \text{ (em 0 segs)} + 30 \text{(em 600 segs)} \text{ (m}^3\text{/h)}.$$

$$T_{Tin} = 150 \text{ (em 0 segs)} + 70 \text{(em 800 segs)} \text{ (}^\circ\text{C)}.$$

$$T_{Sin} = 6 \text{ (em 0 segs)} + 4 \text{(em 500 segs)} \text{ (}^\circ\text{C)}.$$

A arquitetura básica do sistema de diagnóstico consiste dos módulos apresentados na seção 2.7.1 (Módulo de Sensores Virtuais, Gerador de Eventos e Diagnosticador). No módulo de sensores virtuais extraem-se as características de variação das medições tomadas do trocador, i.e. Temperatura de Saída do óleo, Set-point e Saída do Controlador. No modulo gerador de eventos se extraem as características qualitativas dos dados e são apresentados na tabela 4.29.

Por motivos técnicos, considerou-se a obtenção de $Err \sim 0$, $Err \gg 0$ e $Err > 0$ como o erro de regime, calculado quando a derivada da temperatura se encontra

em um faixa próxima a zero ($-0.04 < dT_o/dt > +0.04$), com o objetivo de não criar transições de alta frequência, devido as variações da temperatura de saída do óleo. O modulo de sensores virtuais e gerador de eventos encontra-se no entorno do Simulink, figuras 4.33, 4.35. O módulo diagnosticador, composto pelo autômato diagnosticador, encontra-se no entorno do Stateflow (Figura 4.34), e pela sua evolução no diagrama de transição de estados é que será inferido se algum tipo de falha ocorreu ou não.

4.4.2 Avaliação das simulações:

Para a avaliação das simulações, gerou-se os dois tipos de falha considerados para o trocador de calor: Falta de suprimento de ar na válvula TV-01 (F_1), e falha por emperramento da válvula TV-01 (F_2).

Inicialmente, observou-se as transições de estados do diagnosticador sem gerar falhas no trocador de calor durante um tempo de 1000 segundos. Da figura 4.37 pode-se observar a resposta do sistema com as perturbações causadas pelas entradas detalhadas na seção 4.4.1, e na figura e 4.38 se mostram os eventos gerados durante esse período de avaliação. Notou-se que o diagnosticador não ingressou nos estados F_1 -certos, nem estados F_2 -certos (estados s9-s18, figura 4.34), desta forma, o diagnosticador inferiu que não ocorreu nenhuma das falhas, figura 4.39.

Em outras simulações com falhas geradas durante um tempo de observação de 1000 segundos, foram detectadas a falhas e o tempo de ocorrência, mas com atrasos variáveis, isto dependendo do momento em que se originou a falha. Para uma falha de falta de suprimento de ar, gerada no instante de 400 segundos, o diagnosticador inferiu a falha no tempo 655 segundos, ver Figuras 4.40 e 4.41. Para a mesma falha gerada em 200 segundos o diagnosticador inferiu a falha no instante 480 segundos. Nos dois casos o diagnosticador alcançou todos os estados de comportamento normal (Rotulados com N_b , Estados s0-s8, figura 4.34). Após a falha ser inferida, o diagnosticador ingressa em um ciclo de estados F_1 -certos (Estados s9-s12, figura 4.34) e gerou-se o sinal de ocorrência de falha F_1 , figura 4.42.

Outras simulações com falha por emperramento da válvula, gerada no instante de 550 segundos, o diagnosticador inferiu a falha no tempo 615 segundos. Para a mesma falha gerada em 200 segundos o diagnosticador inferiu a falha no instante

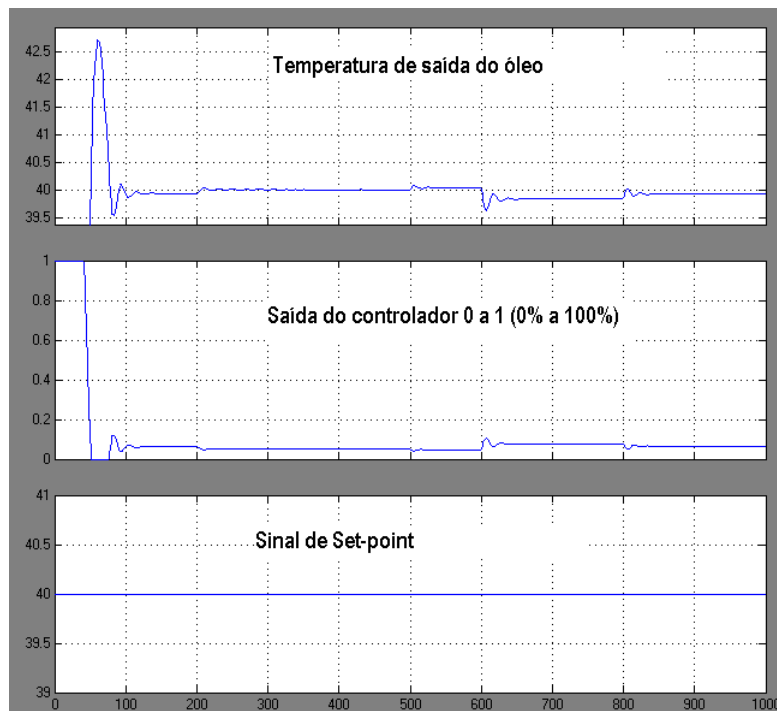


Figura 4.37: Saídas do Trocador de calor durante comportamento normal

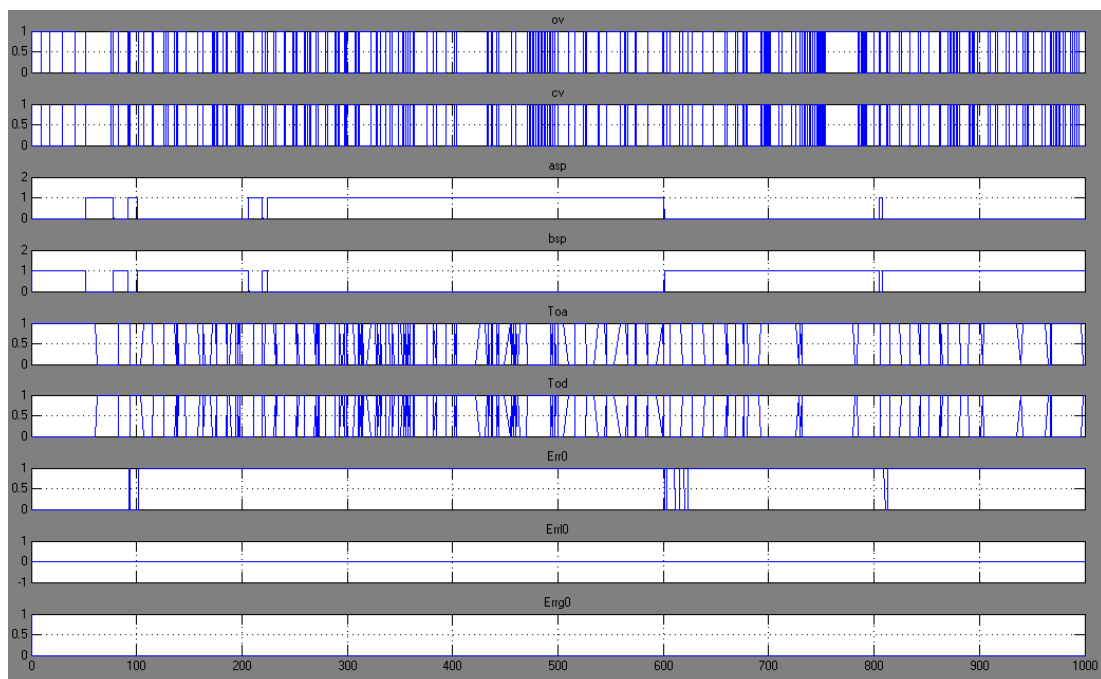


Figura 4.38: Saídas do gerador de eventos do trocador de calor

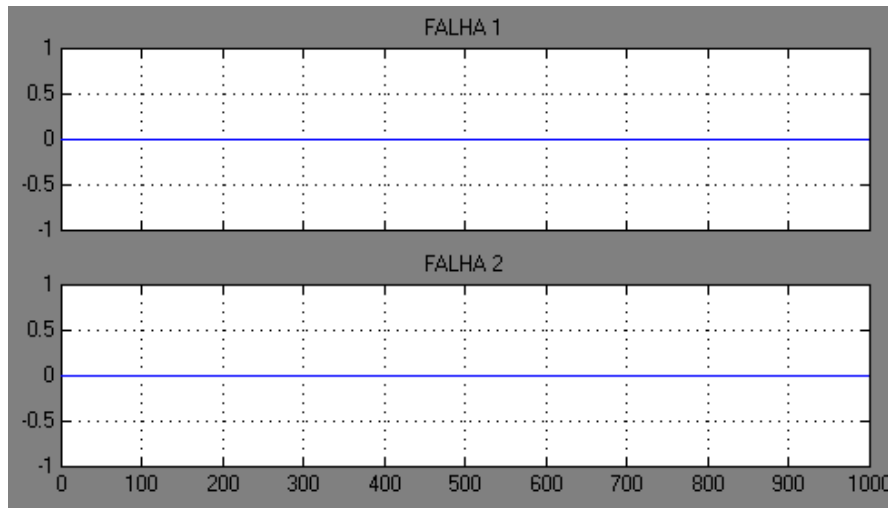
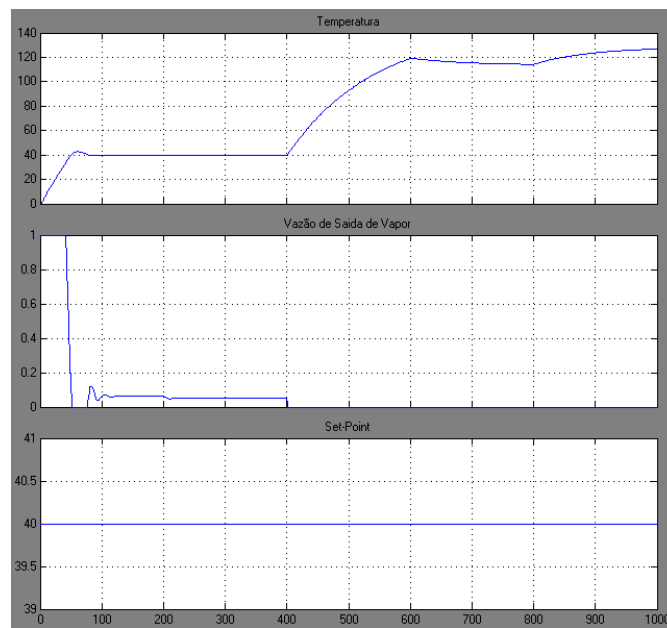
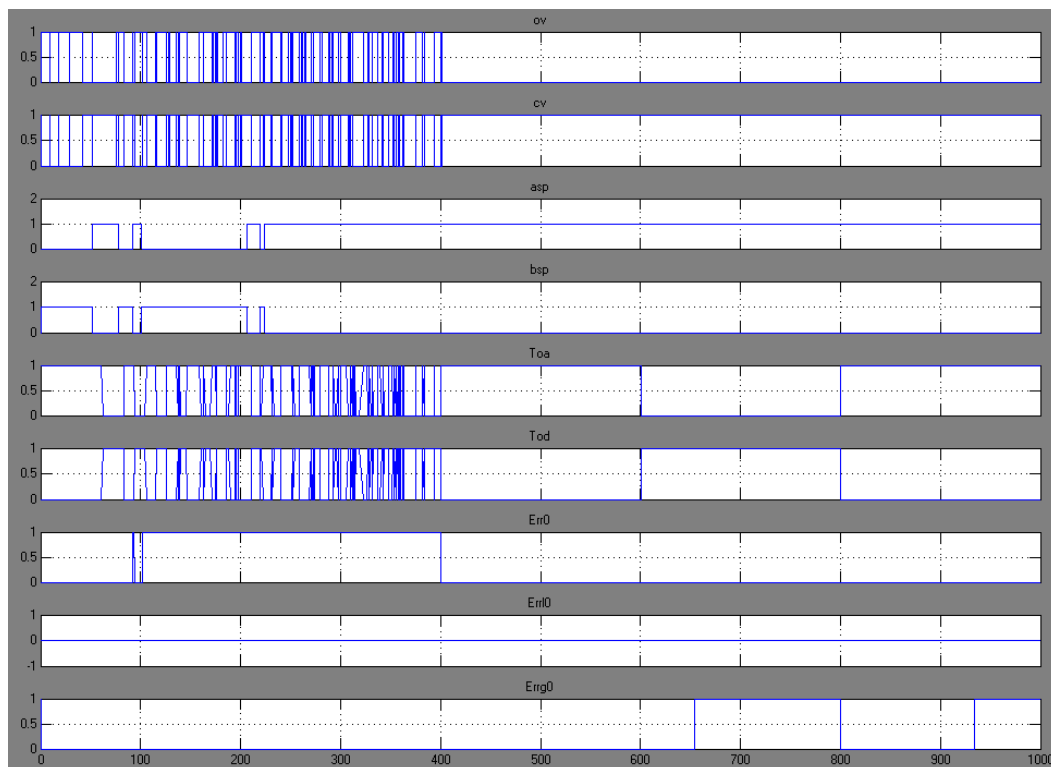


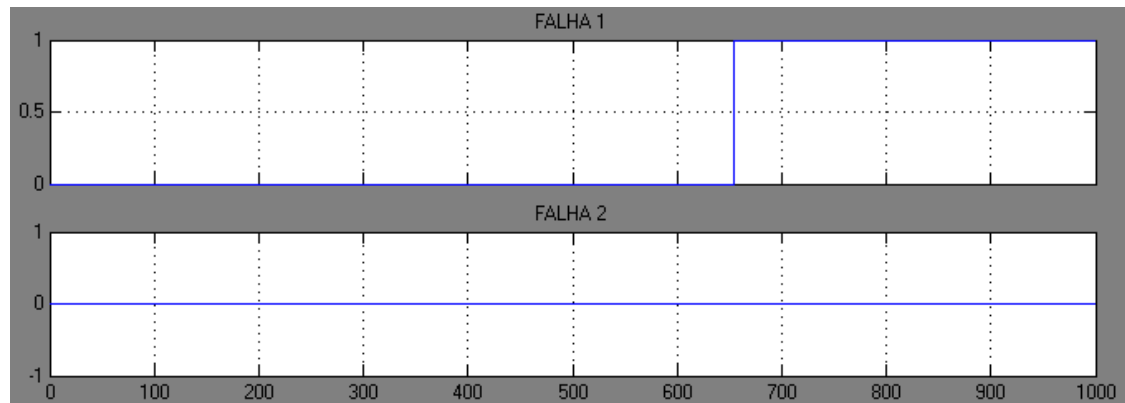
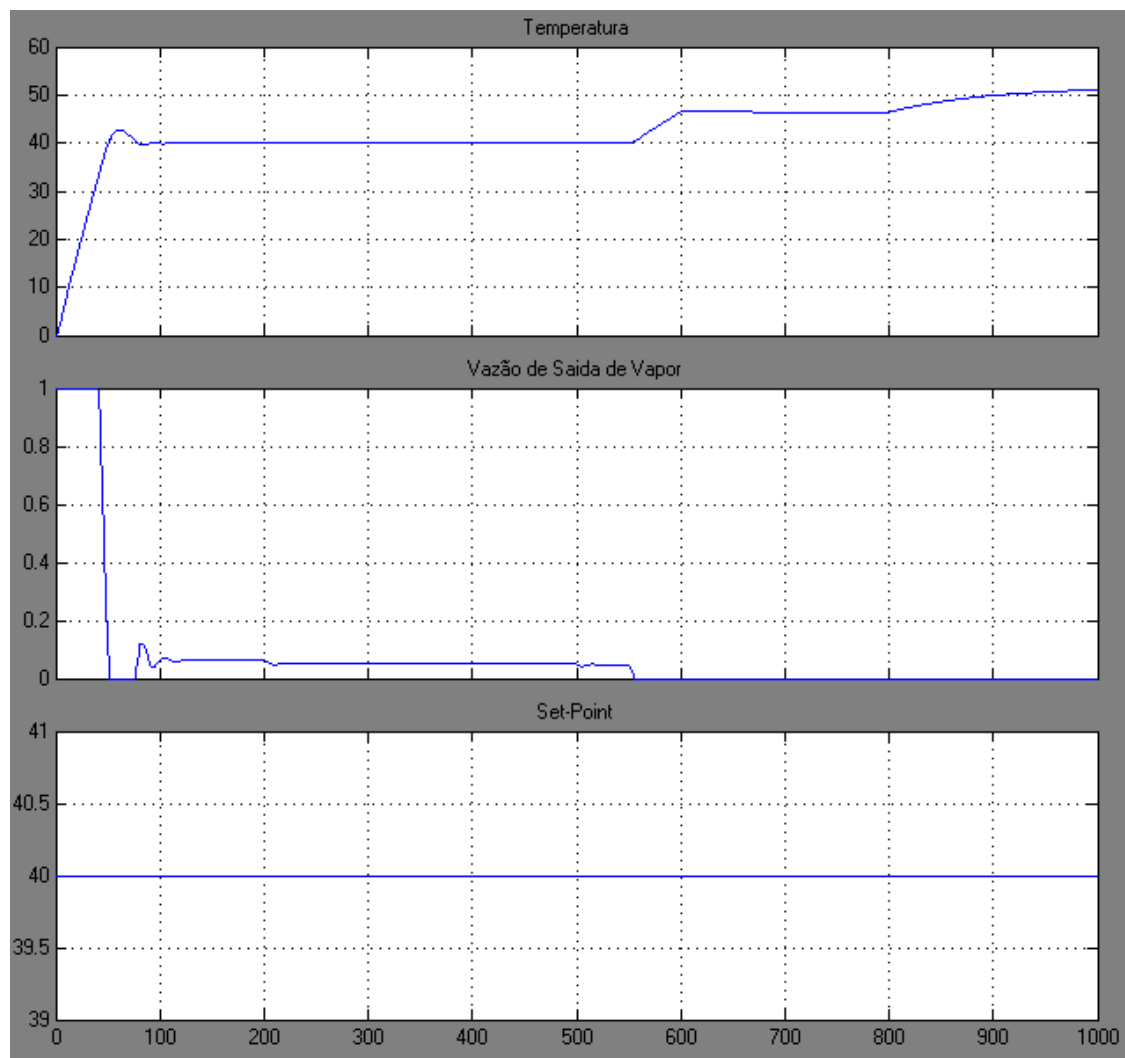
Figura 4.39: Indicador de falhas do diagnosticador durante comportamento normal

370 segundos, ver Figuras 4.43 e 4.44. O diagnosticador alcançou todos os estados de comportamento normal (Rotulados com N_b). Após a falha ser inferida, o diagnosticador entrou em um ciclo de estados F_2 -certos (Estados s13-s18 da figura 4.34) e gerou-se o sinal de ocorrência de falha F_2 , figura 4.45.

A falha por emperramento foi gerada fixando a porcentagem da válvula em um determinado valor. Posto que não se tem os valores reais do processo, a modelagem matemática feita com os dados obtidos em [7], [30], resulta em uma válvula que trabalha em um valor percentual baixo. Em outras palavras, em regime permanente a válvula opera num porcentagem de abertura de 1% a 10%; caso a válvula ficar emperrada entre esses valores percentuais, então a falha por emperramento será inferida pelo diagnosticador. Caso a válvula fique emperrada em valores percentuais superiores a 30% o trocador de calor fornecerá uma vazão de vapor alta, aquecendo o óleo a temperaturas maiores que 70°C. O diagnosticador observando esse comportamento do sistema, reconhece a falha como sendo do tipo falta por suprimento de ar, dado que os eventos gerados são similares aos que se gerariam nesse tipo de falha.

Note-se que na modelagem feita e as considerações físicas, supõe-se que na válvula TV-01 não ocorre os dois tipos de falhas em um mesmo intervalo de tempo. Mas, se isto ocorrer, então só poderá ser inferida a ocorrência da primeira falha que aconteceu ou da falha que causa as maiores transições no trocador.

Figura 4.40: Saídas do Trocador de calor durante comportamento com Falha F_1 Figura 4.41: Saídas do gerador de eventos do trocador de calor com comportamento em falha F_1

Figura 4.42: Indicador de falhas do diagnosticador com inferência de falha F_1 Figura 4.43: Saídas do Trocador de calor durante comportamento com Falha F_2

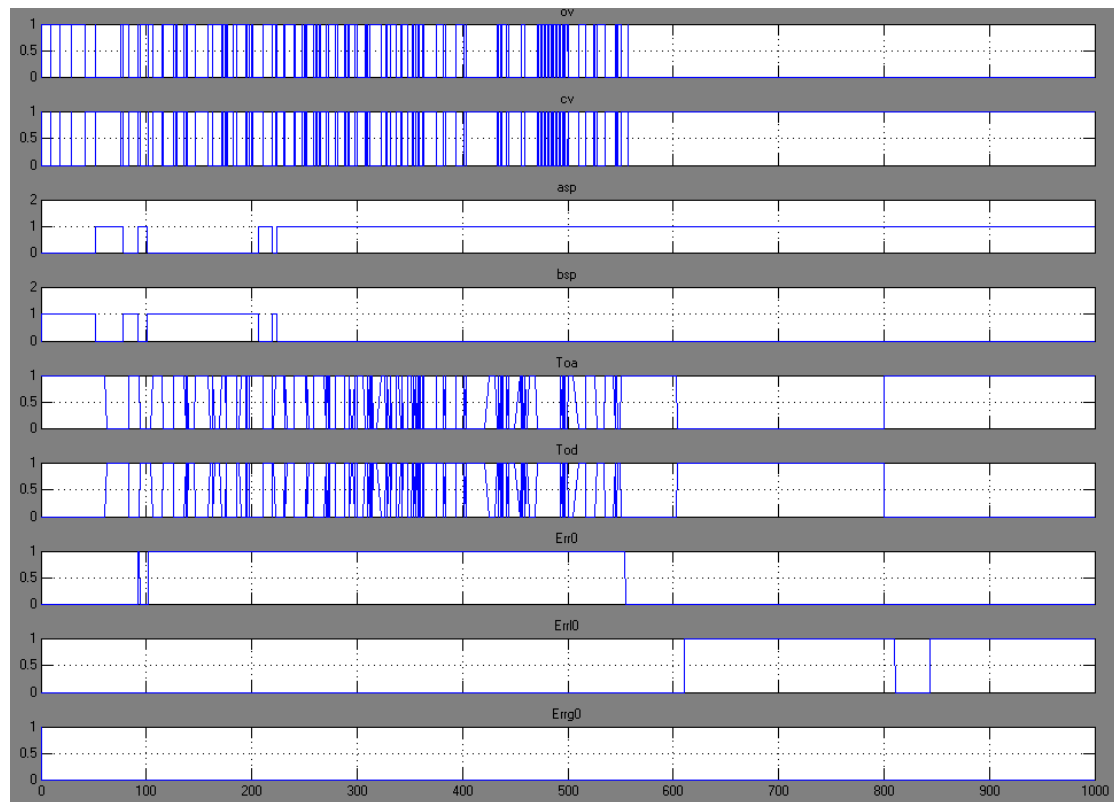


Figura 4.44: Saídas do gerador de eventos do trocador de calor com comportamento em falha F_2

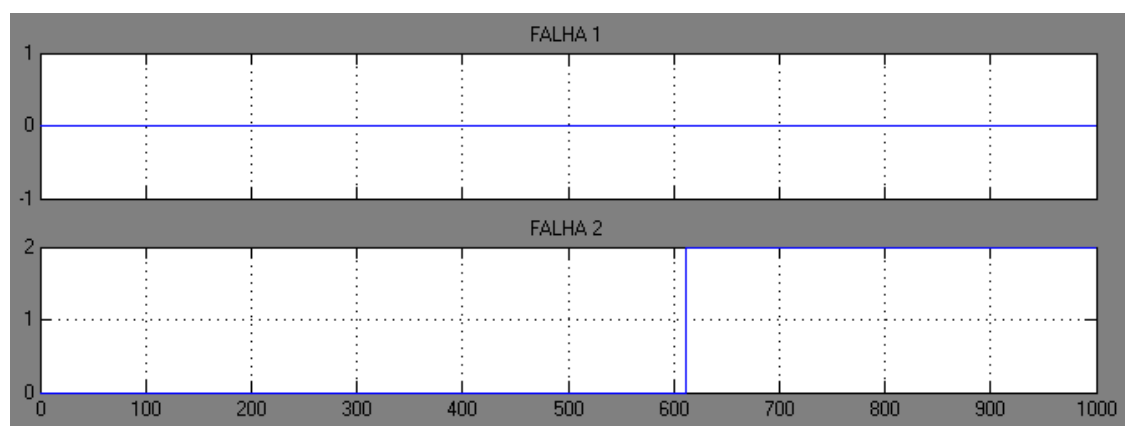


Figura 4.45: Indicador de falhas do diagnosticador com inferência de falha F_2

4.5 Opções de implementação

Uma pergunta muito comum em controle e automação industrial, após de uma abordagem teórica é: O que podemos fazer com o nosso projeto final? Em nosso caso seria, o que podemos fazer com os diagnosticadores projetados? A resposta que achamos tem a ver com as possibilidades de implementação dos diagnosticadores, dados na seção anterior, em ambientes industriais.

É importante que os avanços em pesquisa nas áreas de controle, automação e instrumentação tenham acompanhamento com a evolução das tecnologias aplicadas na indústria atual. O ponto ideal, é que as soluções que sejam geradas no mundo acadêmico devam ter a sua respectiva aplicabilidade no campo industrial ou prático. Isto com o objetivo de suprir as necessidades que as empresas têm, para desenvolver novos produtos nas áreas antes mencionadas. Neste sentido, o presente trabalho tenta fornecer uma possível solução a um problema real como o diagnóstico de falhas através de um autômato diagnosticador. Soluções para a implementação de autômatos supervisores em Controladores Lógicos Programáveis - CLPs foram dadas por [4].

Um Controlador Lógico Programável conhecido também pela sigla de expressão inglesa PLC (*Programmable Logic Controller*) é um computador especializado, baseado em um microprocessador que desempenha funções de controle de diversos tipos e níveis de complexidade. Geralmente as famílias de Controladores Lógicos Programáveis são definidas pela capacidade de processamento de um determinado número de pontos de Entradas e/ou Saídas (E/S).

Controlador Lógico Programável segundo a ABNT (Associação Brasileira de Normas Técnicas), é um equipamento eletrônico digital com hardware e software compatíveis com aplicações industriais. Segundo a NEMA (*National Electrical Manufacturers Association*), é um aparelho eletrônico digital que utiliza uma memória programável para armazenar internamente instruções e para implementar funções específicas, tais como lógica, seqüenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos.

Os CLPs estão muito difundidos nas áreas de controle de processos e de automação industrial. Num sistema típico, toda a informação dos sensores é concen-

trada no controlador (CLP) que, de acordo com o programa em memória, define o estado dos pontos de saída conectados a atuadores.

Uma das desvantagens ou problema atual encontrado nos CLPs é a sua limitada capacidade de armazenagem de informação e velocidade de processamento. Por este motivo torna-se pouco viável a implementação de sistemas de detecção e diagnóstico de falhas em CLPs.

Outros equipamentos de controle que vem sendo utilizados com sucesso são os Sistemas de Controle Distribuído, conhecidos pela sigla inglesa DCSs (*Distributed Control System*). Um DCS integra PLCs e controladores de uma linha de processos em um sistema interativo e coordenado. É possível manejar o processo como um sistema completo, com controle sobre a inter-relação dos vários sub-sistemas. DCSs se referem a sistemas de controle nos quais os elementos de controle não estão numa localização central mas estão distribuídos através de um sistema com cada sub-sistema baixo controle de um ou mais controladores. *Distributed control systems* (DCSs) são usados em aplicações industriais, elétricas e computacionais para monitorar e controlar equipamentos distribuídos com ou sem intervenção humana.

Os Sistemas de Controle Distribuído (DCSs) são sistemas dedicados usados para controlar os processos de manufatura que são contínuos ou grupo-orientados, tais como o refinamento de óleo, os petro-químicos, a geração de energia da estação central, os produtos farmacêuticos, a manufatura de alimento e bebida, produção do cimento, siderurgia, e manufatura do papel.

Em poucas palavras, os DCSs são equipamentos que combinam engenharia simples com uma aberta arquitetura de sistemas modernos. Em detalhes, isto significa:

- Equipamentos de campo (sensores, atuadores, botoeiras, etc) integrados na engenharia de controle do sistema.
- Ferramentas de engenharia, comissionamento e diagnóstico.
- Acesso central a toda a informação de campo.
- Características importantes para a engenharia: Comissionamento, teste, serviços e manutenção.

Um dos aspectos importantes dos DCSs é sua capacidade de armazenamento e velocidade de processamento. Junto ao acesso central da informação de todos

os sensores, torna este equipamento apto para aplicações que precisam manipular quantidades de dados maiores, algoritmos de controle avançados, sistemas de diagnóstico e outros. Desta maneira é possível implementar os sub-sistemas de diagnóstico projetados na seção anterior. Diagnóstico em sistemas a eventos discretos caracteriza-se pelo seu tratamento de informação em forma qualitativa. A aproximação de sistemas de diagnóstico de falha proposta neste trabalho destaca-se pela sua simplicidade na recompilação de dados, onde o resultado é um sistema de diagnóstico de falhas construído através de um método sistemático.

Nos diagnosticadores do processo de separação óleo-gás, as leituras de medições de campo seriam diretamente acessíveis aos DCSs. Os DCSs possuem instruções do tipo algébrico, lógico, diferencial, etc., segundo a norma IEC 61131-3. Nesta norma se detalha o regulamento standard de autômatos programáveis e seus periféricos, incluindo as linguagens de programação que devem-se utilizar (lista de instruções, texto estruturado, diagrama de contatos e diagrama de blocos funcionais) [1]. Considerando sua robustez em ambientes industriais e pelos motivos já antes descritos, é completamente possível implementar sensores virtuais, gerador de eventos bem como o autômato diagnosticador, visto neste capítulo, em um DCS de capacidade média.

Capítulo 5

Conclusões e trabalhos futuros

Os sistemas de diagnóstico e detecção de falhas *on-line* aplicado em diferentes sistemas de manufatura e processos industriais tem por objetivo a determinação do tipo de falha e suas conseqüências. Neste contexto, a teoria de sistemas a eventos discretos oferece uma abordagem sistemática ao diagnóstico de falhas. As redes de Petri são ferramentas de modelagem que geram linguagens de sistemas a eventos discretos não temporizados e temporizados. Este trabalho estudou o projeto de diagnosticadores de sistemas a eventos discretos não temporizados modelados por redes de Petri do ponto de vista tanto teórico, como prático.

Foram feitas análises de trabalhos anteriores de diagnóstico de falhas usando redes de Petri e foram encontradas algumas lacunas teóricas ou práticas, que deram lugar as contribuições desta dissertação.

Estendeu-se a teoria existente de mapeamento de sensores para sistemas modelados por Redes de Petri, partindo de uma teoria dada anteriormente para autômatos. Definiu-se uma relação de modelos em redes de Petri, denominada compatibilidade de modelos. Integrou-se o estudo de diagnosticabilidade e de técnicas baseadas em sistemas a eventos discretos para a construção de diagnosticadores de falhas. Também foi proposto um procedimento sistemático para o projeto de sistemas automáticos de diagnóstico de falha para sistemas reais modelados por eventos discretos. Foi aplicado o estudo de diagnosticabilidade no processo de separação óleo-gás e propôs-se um sistema de diagnóstico de falhas.

Na tarefa de implementação de sistemas de diagnóstico de falhas *on-line*, é preciso a armazenagem do autômato diagnosticador, sensores virtuais e o gerador de eventos em um controlador com capacidade computacional média. Foi feito uma pré

análise na possibilidade de implementação dos diagnosticadores em equipamentos de utilização industrial como são os *Distributed Control Systems* (DCSs).

Observa-se que em bibliografias recentes, Bhowal *et al.* [2], foram desenvolvidos sistemas de diagnóstico usando aproximações híbridas com tempo discreto, baseados em conjuntos finitos de variáveis discretas e contínuas. Bhowal *et al.* [2] analisou a diagnosticabilidade de sistemas híbridos (que são autômatos cujas transições são definidas em função da evolução dinâmica dos estados contínuos do sistema) aplicando as condições necessárias e suficientes dadas em [26], onde achou exemplos onde não se cumprem a condição necessárias e suficiente de não existência de ciclos F_i -indeterminados, devido à abstração utilizada pelos SEDs que omite dinâmicas contínuas no sistema [2]. Bhowal *et al.* [2] procuraram estender a condição de diagnosticabilidade de Sampath *et al.* [26] para sistemas híbridos. Claramente, a não satisfação da condição necessária e suficiente se deve ao fato de não ter uma equivalência entre sistemas a eventos discretos e sistemas híbridos. Nesse caso, é possível que a condição seja apenas necessária, como ocorre quando se estende a condição de diagnosticabilidade para sistemas a eventos discretos estocásticos.

Finalmente concluiu-se que é possível projetar um sistema de diagnóstico de falhas, através de um procedimento sistemático de complexidade razoável, aplicando os procedimentos descritos na dissertação. Diferentemente dos outros métodos (veja seção 1.2), esta metodologia pode se aplicar por intermédio de algoritmos/procedimentos iterativos para a construção de sistemas automáticos de diagnóstico de falhas. Esta metodologia conjuntamente com a teoria de diagnosticabilidade representa uma ferramenta formal para o análise e síntese de sistemas de supervisão e diagnóstico de falhas.

Como trabalhos futuros que podem ser desenvolvidos para a continuação do trabalho iniciado nesta dissertação, sugere-se:

- Elaboração de um pacote de cálculo (*toolbox*) para a realização de operações com redes de Petri tais como produto, composição paralela, mapeamento de sensores, para facilitar e automatizar a construção de diagnosticadores, que no âmbito deste trabalho, foram realizados manualmente.
- Estudo de diagnosticadores de falhas usando modelos temporizados e estocás-

ticos.

- Estudo detalhado da implementação do sistema de diagnóstico proposto.
- Avaliação da possibilidade de construção de redes de Petri diagnosticadoras, junto com os procedimentos de construção para essa abordagem.
- Modelagem de um número maior de falhas no processo de separação óleo-gás.
- Simulação do processo de separação óleo-gás com dados reais e validar os modelos matemáticos.
- Simulação do sistema de diagnóstico projetado junto com um emulador do processo de separação óleo-gás, com o objetivo de validar os autômatos diagnosticadores.
- Implementação industrial do sistema de diagnóstico de falhas proposto, uma vez validado.

Referências Bibliográficas

- [1] ABB, Manual "The compact control System Freelance 800F", Automation Technologies, Germany, 2006.
- [2] Alur, R., Courcoubetis, C., Henzinger, T.A. and Ho, P., Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: LNCS 736, Springer Verlag. pp. 209-229.
- [3] Åström, K., and Wittenmark, B., *Computer Controlled Systems*, Prentice-Hall. Englewood Cliff, 1997.
- [4] Balemi, S., Control of Discrete Event Systems: Theory and Applications, *Ph.D. Thesis*, Swiss Federal Institute of Technology, Zurich, 1992.
- [5] Campos, M. C. M. M., Teixeira, H. C., Controles Típicos de Equipamentos e Processos Industriais, *Editora: Edgard Blucher Brasil*, 2006
- [6] Cassandras, C., Lafortune, S., *Introduction to discrete event systems*, Kluwer, Massachusetts, 1999.
- [7] Dias, A., Avaliação da metodologia "Diagnostic model processor" para Diagnóstico de Falhas em um Processo de Separação, *Tese de Mestrado*, COPPE/UFRJ, 1994.
- [8] Fabian, M., Hellgren, A., PLC-based Implementation of Supervisory Control for Discrete Event Systems, *37th IEEE Conference on Decision and Control*, Florida, USA, 1998 pp 3305-3310.
- [9] Genç, S., and Lafortune, S., A distributed algorithm for on-line diagnosis of place-bordered Petri nets. *In 16th International Federation of Automatic Control World Congress*, Prague, Czech Republic, 2005.

- [10] Giua, A., and Seatzu, C., Fault detection for discrete event systems using Petri nets with unobservable transitions. *IEEE Proc. 44th Int. Conf. on Decision and Control and European Control Conference*, Sevilla, Spain, 2005, pp 6323-6328.
- [11] Halgamuge, S., Advanced Methods for Fusion of Fuzzy System and Neural Networks in Intelligent Data Processing. Fortschr. Ber. VDI Reihe 10, Nr. 401, VDI-Verlag - Diisseldorf, 1996.
- [12] Himmelblau, D. M., *Fault detection and diagnosis in chemical and petrochemical processes*, Elsevier: Amsterdam, 1978, pp 343-393.
- [13] Isermann, R., Supervision fault-detection and fault-diagnosis methods-an introduction, *Control Eng. Practice*, vol. 5, no. 5, 1997, pp 639-652.
- [14] Jiroveanu, G., Boel, R.K., and De Schutter, B., *Fault diagnosis for time Petri nets*, In Proc. WODES 06, Ann Arbor - USA, Julho 2006.
- [15] Jonhson C., *Process Control Instrumentation Tecnology*, 7th Edition, Prentice Hall, New Jersey, 2003.
- [16] Kaszkurewicz, E., Bhaya, A., and Ebecken, F. F., *A Fault Detection and Diagnosis Module for Oil Production Plants in Offshore Platforms*, Expert Systems with Applications, vol. 12, no. 2, 1997, pp 189-194.
- [17] Lafortune, S., Teneketzis, D., Sampath, M., Sengupta, R., and Sinnamohideen, K., Failure diagnosis of dynamic systems: an approach based on discrete event systems. *Proceedings of the American Control Conference*, Arlington, USA, pp. 2058-2071, 2001
- [18] Lafortune, S., Notas de Aula do curso "Discrete event systems", University of Michigan, USA, 2006.
- [19] Lunze, J., Schröder, J., Sensor and actuator fault diagnosis of systems with discrete inputs and outputs, *IEEE Trans. Systems, Man and Cybernetics*, Part B, Vol.34, nro. 3, 2004, pp. 1096-1107.

- [20] Manyari-Rivera, M., Basilio, J. C., and Bhaya, A., Integrated fault diagnosis based on Petri net models, *IEEE Conference on Control Application*, 2007. (Aceito para apresentação)
- [21] Murata, R., Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, vol. 77, no. 4, 1989, pp 541-580.
- [22] Ogata, K., Ingeniería de Control Moderna, 2da. edición. *Prentice Hall Hispanoamericana*, México, 1993.
- [23] Peterson, J., *Petri net theory and the modeling of systems*, Prentice Hall, Englewood Cliffs, NJ, 1981.
- [24] Reider, C. A., Detecção e isolamento de falhas em instrumentação de Processos de Produção de Petróleo, *Tese de Doutorado*, COPPE/UFRJ, 2002.
- [25] Reutenauer, C., *The Mathematics of Petri Nets*, Prentice Hall, Masson, Paris, 1988.
- [26] Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketizis, D., Diagnosability of discrete-event systems, *IEEE Trans. on Automatic Control*, vol. 40, no. 9, 1995, pp 1555-1575.
- [27] Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketizis, D., Failure diagnosis using discrete event models, *IEEE Trans. Control Systems Technology*, vol. 4, no. 2, 1996, pp 105-124.
- [28] Sampath, M., A Hybrid Approach to Failure Diagnosis of Industrial Systems, *Proc. of the American Control Conference*, 2001, pp 2077-2082.
- [29] Smith, C., e Corripio B., Control Automático de Processo, *Editorial Limusa*, México, 1991.
- [30] Teixeira, E., Diagnóstico Inteligente de Falhas em um Processo de Separação Óleo-Gás em Plataformas Offshore, *Tese de Mestrado*, COPPE/UFRJ, 1993.
- [31] Ushio, T., Onishi I., and Okuda, K., Fault detection based on Petri net models with faulty behaviour, *IEEE Proc. of the International Conference on Systems, Man and Cybernetics*, 1998, pp 113-118.

- [32] Wen, Y., and Jeng, M., Diagnosability of Petri nets, *IEEE Proc. of the International Conference on Systems, Man and Cybernetics*, 2004, pp 4891-4896.

Apêndice A

Símbolos P&ID

Assim como a eletrônica tem símbolos para representar componentes em circuitos esquemáticos, o controle de processos têm símbolos para representar elementos de sistemas de controle de processos. No lugar de um esquema, chamamos de diagrama de controle de processos à sigla inglesa P&ID (*Piping and instrumentation diagram*). Os símbolos e seus significados tem sido standarizados na indústria pela ANSI (*American National Standards Institute*) e a ISA (*Instrumentation Systems and Automation*). Este apêndice apresenta uma vista geral do padrão incluindo só os elementos comuns mais usados [15].

A.1 Linhas de conexão

O padrão especifica os tipos de linhas a serem usados para representar conexões na planta, incluindo linhas de fluxo controle e linhas de interconexão entre instrumentos. A figura A.1 mostra a definição das linhas.

A.2 Instrumentos ou funções gerais

Este aspecto do padrão define os símbolos a serem usados pelos vários instrumentos que necessitam medir e controlar processos e plantas. Esta instrumentação inclui sensores, transmissores, conversores de dados, controladores, computadores e controladores lógicos programáveis. A figura A.2 mostra as bolas, retângulos, hexágonos, e rombos utilizados para denotar a instrumentação.

Geralmente, os símbolos também terão combinações de letras e números. As letras servem como abreviatura para indicar o propósito do instrumento no sistema.

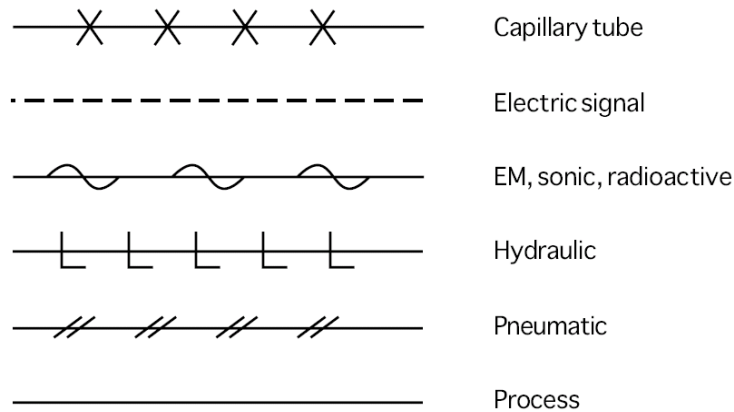


Figura A.1: Linhas P&ID de processos e sinais

	Acessível ao operador	Localizado em campo	Não acessível ao operador
Instrumentos discretos	1	2	3
Indicação/controlado compartilhados	4	5	6
Função Computacional	7	8	9
PLC	10	11	12

Figura A.2: Símbolos especiais usados para instrumentação no P&ID

A tabela A.1 mostra os significados atribuídos das letras dependendo se elas são a primeira letra no símbolo ou as subseqüentes letras.

Os números servem para identificar em que parte da planta opera o instrumento. A atribuição desses números é a critério do usuário, mas geralmente as malhas são numeradas, como 100, 101, etc., e os instrumentos das malhas levam esse número.

A.3 Atuadores e elementos do processo

O último aspecto do P&ID são esses elementos que formam parte do processo mesmo. Isto inclui válvulas de controle, atuadores para o controle de válvulas, esteiras, tanques, etc. A figura A.3 mostra alguns dos elementos mais freqüente-

Tabela A.1: Letras de identificação dos elementos P&ID

	Primeira letra	Letras subsequentes
A	Analysis	Alarm
B	Burner, combustion	
C	Undefined	Control, controller
D	Undefined	
E	Voltage	Sensor, primary element
F	Flow Rate	
G	Undefined	Glass, viewing device
H	Hand	High
I	Current	Indication, readout
J	Power	
K	Time, time schedule	Control station
L	Level	Light, low
M	Undefined	Middle, intermediate
N	Undefined	
O	Undefined	Orifice, restriction
P	Pressure, vacuum	Point, test point
Q	Quantity	
R	Radiation	Recorder, recorder
S	Speed, frequency	Switch
T	Temperature	Transmit, transmitter
U	Multivariable	Multifunction
V	Vibration, mechanical	Valve, damper, louver
X	Weight, force	Well
Y	Event, state	Relay, compute, convert
Z	Position, dimension	Driver, actuator


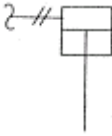

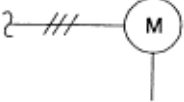
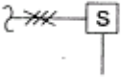


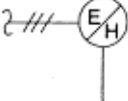

 Atuador Neumático	 Simple ação	 Válvula Mariposa
 Motor rotativo	 Solenóide	 Válvula globo
 Atuador manual	 Eleto-hidráulico	 Válvula de tres vias

Figura A.3: Símbolos para elementos finais de controle

mente utilizados nesta categoria. Em alguns casos, a norma permite ao usuário representar equipamentos especializados no processo, tais como trocadores de calor e bombas, numa maneira consistente com desenhos práticos.

Seja o símbolo PT-101: Segundo a tabela A.1, a P (primeira letra) significa Pressure, a T (segunda letra) significa transmitter, então o símbolo PT-101, significa pressure transmitter (Transmissor de pressão) localizado na malha 101.

Seja o símbolo LC-103: Segundo a tabela A.1, a L (primeira letra) significa Level, a C (segunda letra) significa controller, então o símbolo LC-101, significa level controller (Controlador de Nivel) localizado na malha ou setor da planta 103.