



DIAGNOSTICABILIDADE DE SISTEMAS HÍBRIDOS: ANÁLISE E VERIFICAÇÃO

Eduardo Alexandre dos Reis Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Marcos Vicente de Brito Moreira
Oumar Diene

Rio de Janeiro
Junho de 2015

DIAGNOSTICABILIDADE DE SISTEMAS HÍBRIDOS: ANÁLISE E
VERIFICAÇÃO

Eduardo Alexandre dos Reis Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Oumar Diene, D.Sc.

Prof. Maurício Bezerra de Souza Junior, D.Sc.

Prof. Amit Bhaya, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2015

Silva, Eduardo Alexandre dos Reis

Diagnosticabilidade de sistemas híbridos: Análise e verificação / Eduardo Alexandre dos Reis Silva. – Rio de Janeiro: UFRJ/COPPE, 2015.

XIV, 98 p.: il.; 29, 7cm.

Orientadores: Marcos Vicente de Brito Moreira

Oumar Diene

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 96 – 98.

1. Sistemas a eventos discretos. 2. Autômatos. 3. Diagnóstico de falha em SED. 4. Sistemas Híbridos. 5. Autômatos híbridos. 6. Diagnóstico de falha em SH. 7. Diagnosticabilidade. 8. Distinguibilidade. 9. Diagnosticador. I. Moreira, Marcos Vicente de Brito *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço a Deus. *Porque dele, e por meio dele, e para ele são todas as coisas. A ele, pois, a glória eternamente. Amém!* (Romanos 11. 36).

Agradeço à minha esposa Adriana Alves de Vasconcellos pelo seu amor, carinho e alegria, pois sem ela esta conquista não seria possível.

Agradeço à minha filha Luíza Fátima Vasconcellos Reis Silva pela inspiração e alegria proporcionados pelo seu nascimento e convívio.

Agradeço aos meus pais, Luiz Carlos Silva e Cristina Brandão da Rocha Reis Silva, e à minha avó, Ilda Morgado da Rocha Reis, pelo seu amor e carinho.

Agradeço aos meus professores e orientadores, Marcos Vicente de Brito Moreira e Oumar Diene, por todas as horas gastas de aconselhamento e orientação.

Agradeço à Marinha do Brasil pela oportunidade proporcionada.

Agradeço aos meus amigos da Diretoria de Engenharia Naval pelo companheirismo e pela compreensão durante a conclusão deste trabalho.

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

DIAGNOSTICABILIDADE DE SISTEMAS HÍBRIDOS: ANÁLISE E VERIFICAÇÃO

Eduardo Alexandre dos Reis Silva

Junho/2015

Orientadores: Marcos Vicente de Brito Moreira
Oumar Diene

Programa: Engenharia Elétrica

Sistemas industriais modernos são controlados em tempo real e supervisionados por meio de sistemas de controle automático baseado em computadores. Esses sistemas combinam comportamentos discretos e contínuos, e são melhor modelados como sistemas híbridos (SH). Este trabalho apresenta uma nova definição para a diagnosticabilidade de um sistema híbrido, chamada de h -diagnosticabilidade. Além disso, é apresentado um novo método para a verificação da h -diagnosticabilidade baseado na construção do diagnosticador híbrido G_H , que combina um diagnosticador de sistema a eventos discretos com a distinguibilidade dos modos de um sistema híbrido. Por fim, esse trabalho também apresenta um exemplo de sistema híbrido que é o sistema de controle da máquina do leme de um navio, que é responsável por movimentar o leme e definir o rumo da embarcação. Para esse exemplo são construídos um autômato híbrido e um diagnosticador híbrido para a verificação da h -diagnosticabilidade do sistema híbrido. Após a construção desses autômatos, é verificada que a linguagem gerada pela componente discreta desse sistema híbrido é não diagnosticável, porém ao utilizar o novo método para o diagnóstico de falhas em sistema híbrido, o referido exemplo torna-se h -diagnosticável.

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

DIAGNOSABILITY OF HYBRID SYSTEMS: ANALYSIS AND VERIFICATION

Eduardo Alexandre dos Reis Silva

June/2015

Advisors: Marcos Vicente de Brito Moreira

Oumar Diene

Department: Electrical Engineering

Modern industrial systems are real time controlled and supervised by means of automatic computer-based control systems. These systems combine discrete and continuous behaviors, and are best modeled as hybrid systems (HS). This paper presents a new definition for diagnosability of hybrid systems, called the *h*-diagnosability. In addition, a new method is presented for verification of *h*-diagnosability based on the construction of a diagnoser automaton G_H , called hybrid diagnoser, that combines the discrete-event diagnoser with the distinguishability of the modes of the hybrid system. Finally, this work shows an example of hybrid system, that is the steering gear control system of a ship, which is responsible for moving the rudder and to set the course of the vessel. For this example, a hybrid automaton and a diagnoser G_H are constructed to verify the *h*-diagnosability of the hybrid system. After the construction of these automatons, it is verified that the language generated by discrete component of this hybrid system is not diagnosable. However, by using this new method for diagnostic of faults in hybrid system, this example is *h*-diagnosable.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Símbolos	xii
Lista de Abreviaturas	xiv
1 Introdução	1
2 Sistemas a eventos discretos	7
2.1 Linguagens	8
2.2 Operações com Linguagens	9
2.3 Projeções de sequências e linguagens	9
2.4 Autômato	11
2.4.1 Linguagens representadas por Autômatos	13
2.4.2 Operações com Autômatos	14
2.4.3 Autômatos com observação parcial	18
3 Diagnóstico de falhas em SED	22
3.1 Diagnóstico de falhas em SED	22
3.2 Diagnosticabilidade de falhas em SED	24
3.3 Diagnosticador	24
3.4 Diagnóstico de falhas utilizando mapa de sensores	30
4 Sistemas Híbridos	41
4.1 Autômato Híbrido	41
4.1.1 Conceito de distinguibilidade dos modos	43
4.1.2 Exemplo de um Sistema Híbrido	45
5 Diagnóstico de falhas em Sistemas Híbridos	57
5.1 Diagnosticabilidade em Sistemas Híbridos	57
5.1.1 Conceitos preliminares	59

5.1.2	Definição de h -diagnosticabilidade	60
5.2	Verificação da Diagnosticabilidade de SH	62
5.2.1	Diagnosticador Clustered	62
5.2.2	Verificação da distinguibilidade dos modos em SH	64
5.2.3	Análise da diagnosticabilidade utilizando o diagnosticador clustered G_{CD}	67
5.2.4	Diagnosticador Híbrido G_H	72
6	Conclusões, contribuições e trabalhos futuros	85
6.1	Conclusões	85
6.2	Contribuições	86
6.3	Trabalhos futuros	87
A	Distinguibilidade dos modos de um sistema híbrido	88
A.1	Método do observador dedicado em um SH	88
A.2	Método do espaço de paridade	91
	Referências Bibliográficas	96

Lista de Figuras

2.1	Diagrama de transição de estados do autômato G do Exemplo 2.3.	12
2.2	Autômatos G_1 e G_2 do Exemplo 2.4.	18
2.3	Autômatos $G_1 \times G_2$ e $G_1 \parallel G_2$ do Exemplo 2.4.	18
2.4	Autômato G com evento não-observável do Exemplo 2.5.	19
2.5	Autômato G do Exemplo 2.6.	21
2.6	Autômato $Obs(G)$ do Exemplo 2.6.	21
3.1	Autômato rotulador A_l^{YN}	25
3.2	Autômato G do Exemplo 3.1.	26
3.3	Autômato G_l do Exemplo 3.1.	26
3.4	Autômato G_d do Exemplo 3.1.	27
3.5	Autômato G do Exemplo 3.3.	28
3.6	Autômato G_l do Exemplo 3.3.	29
3.7	Autômato G_d do Exemplo 3.3.	29
3.8	Autômato da válvula G_v do Exemplo 3.4.	32
3.9	Autômato da bomba G_b do Exemplo 3.4.	33
3.10	Autômato do controlador G_c do Exemplo 3.4.	33
3.11	Autômato $G = G_v \parallel G_b \parallel G_c$ do Exemplo 3.4.	34
3.12	Autômato G_d do Exemplo 3.4.	34
3.13	Autômato G_{ms} do Exemplo 3.4.	36
3.14	Autômato G_{ms_l} do Exemplo 3.4.	37
3.15	Autômato diagnosticador G_{ms_d} do Exemplo 3.4.	37
3.16	Autômato G do Exemplo 3.5.	39
3.17	Autômato G_{ms_l} do Exemplo 3.5.	39
3.18	Autômato diagnosticador G_{ms_d} do Exemplo 3.5.	40
4.1	Autômato G do Exemplo 4.1.	47
4.2	Diagrama óleo-hidráulico do sistema da máquina do leme do Exemplo 4.1.	50
4.3	Autômato H do Exemplo 4.1.	56
5.1	Autômato diagnosticador G_d do Exemplo 5.1	58

5.2	Componente discreta G do Exemplo 5.2	68
5.3	Autômato G_{CD} do Exemplo 5.2	69
5.4	Componente discreta G do Exemplo 5.3	70
5.5	Autômato G_{CD} do Exemplo 5.3	71
5.6	Autômato rotulador de cabeça e cauda A_ℓ^{HT}	71
5.7	Autômato $G_\ell = G\ A_\ell^{HT}\ A_\ell^{YN}$ do Exemplo 5.3	71
5.8	Autômato G_{CD} com $G_\ell = G\ A_\ell^{HT}\ A_\ell^{YN}$ do Exemplo 5.4	72
5.9	Autômato G do Exemplo 5.5.	77
5.10	Autômato G_I do Exemplo 5.5.	77
5.11	Autômato G_H do Exemplo 5.5.	77
5.12	Autômato G_I do Exemplo 5.6.	79
5.13	Autômato G_H do Exemplo 5.6.	80
5.14	Variação da posição angular do Modo Normal e Rápido com erro de modelagem e ruído branco em $\{(3HN), (9TY), (9HY)\}$ do Exemplo 5.6.	82
5.15	Variação da posição angular do Modo Normal e Rápido com erro de modelagem e ruído branco em $\{(2HN), (10TY), (10HY)\}$ do Exemplo 5.6.	84
A.1	Configuração básica de um gerador de resíduo através do método do observador dedicado	89
A.2	Configuração básica de um gerador de resíduo para a verificação da distinguibilidade dos modos	90

Lista de Tabelas

3.1	Tabela dos estados de G do Exemplo 3.4.	35
3.2	Mapa de sensores para o Exemplo 3.4.	35
3.3	Mapa de sensores para o Exemplo 3.5.	38
4.1	Tabela com a descrição dos eventos de G do Exemplo 4.1.	48
4.2	Tabela com a nomenclatura do diagrama óleo-hidráulico, conforme figura 4.2, do Exemplo 4.1.	49
4.3	Tabela com as características dos modos de H do Exemplo 4.1.	54
5.1	Tabela com a Tolerância e o Resíduo de $\{(3HN), (9TY), (9HY)\}$ do Exemplo 4.1.	82
5.2	Tabela com a Tolerância e o Resíduo de $\{(2HN), (10TY), (10HY)\}$ do Exemplo 4.1.	83

Lista de Símbolos

σ_f	Evento de falha
Σ	Conjunto de eventos
Σ_o	Conjunto de eventos observáveis
Σ_{uo}	Conjunto de eventos não-observáveis
Σ^*	Fecho de Kleene de Σ
Σ_f	Conjunto de eventos de falha
ε	Sequência de eventos vazia
λ	Sequência de conjunto de estados vazia
s	Sequência de eventos
$ s $	Comprimento de uma sequência de eventos
$s_1 \cdot s_2$	Concatenação de duas sequências s_1 com s_2
G	Autômato
A_ℓ	Autômato rotulador
G_ℓ	Autômato rotulado
G_d	Autômato diagnosticador
G_{sm}	Autômato gerado pelo mapa de sensores
G_{CD}	Autômato diagnosticador clustered
G_H	Autômato diagnosticador híbrido
Q	Conjunto de estados discretos de um autômato
q_0	Estado inicial de um autômato
Q_m	Conjunto de estados marcados de um autômato
\bar{L}	Fecho de prefixo de L
$L(G)$	Linguagem gerada por um autômato G
$L_m(G)$	Linguagem marcada por um autômato G
L_N	Linguagem formada por sequências que não contém o evento σ_f
\mathcal{P}	Caminho em G
\mathcal{P}_d	Caminho em G_d
\mathcal{P}_{CD}	Caminho em G_{CD}
\mathcal{P}_H	Caminho em G_H

$Ac(G)$	Parte Acessível de G
$CoAc(G)$	Parte Coacessível de G
$Trim(G)$	Autômato Trim de G
$G_1 \times G_2$	Composição produto entre os autômatos G_1 e G_2
$G_1 \parallel G_2$	Composição paralela entre os autômatos G_1 e G_2
$UR(q)$	Alcance não observável de um estado q
$Obs(G)$	Observador do autômato G em relação a Σ_o
X	Espaços de estados contínuos
Y	Conjunto das possíveis saídas do sistema contínuo
U	Conjunto dos sinais de controle contínuo
f	Campo vetorial
g	Função de saída do sistema contínuo
Inv	Conjunto das condições de invariância
$guard$	Conjunto das condições de guarda
ρ	Função reset
\mathbf{x}_0	Estado contínuo inicial
T_s	Período de amostragem
\mathcal{M}	Modelo contínuo
$q_i \approx q_j$	q_i é distinguível de q_j
$q_i \sim q_j$	q_i não é distinguível de q_j

Lista de Abreviaturas

<i>SH</i>	Sistema híbrido
<i>SED</i>	Sistema a eventos discretos
<i>RTHS</i>	Real Time Hybrid System
<i>VF</i>	Válvula fechada
<i>VA</i>	Válvula aberta
<i>VT</i>	Válvula travada
<i>BD</i>	Bomba desligada
<i>BL</i>	Bomba ligada
<i>SP</i>	Sem pressão
<i>CP</i>	Com pressão
<i>SV</i>	Sem vazão
<i>CV</i>	Com vazão
<i>BT</i>	Baixa temperatura
<i>AT</i>	Alta temperatura
<i>MAT</i>	Muito alta temperatura
<i>VDP</i>	Válvula direcional principal
<i>VDA</i>	Válvula direcional auxiliar
<i>BB</i>	Bombordo
<i>BE</i>	Boreste

Capítulo 1

Introdução

Sistemas industriais modernos são controlados em tempo real e supervisionados por meio de sistemas de controle automático baseado em computadores. Esses sistemas utilizam sensores para monitorar a planta, (como, por exemplo, termopares, transdutores de pressão e sensores de velocidade), e atuadores (como, por exemplo, motores elétricos, válvulas proporcionais e válvulas solenóides). Esses sistemas combinam comportamentos discretos e contínuos [1–3], e são melhor modelados como sistemas híbridos (SH).

Todos os sistemas, inclusive os híbridos, estão sujeitos a falhas que podem causar danos ao equipamento, bem como risco ao operador. Por esse motivo, o diagnóstico de falhas em sistema híbrido é importante para garantir a confiabilidade e segurança do sistema.

O diagnóstico de falhas tem como objetivo identificar falhas em um sistema defeituoso através da coleta e análise de seus dados, como, por exemplo, as informações dos eventos observáveis de um sistema a eventos discretos. A falha pode ser do tipo abrupta ou incipiente [4]. As falhas abruptas causam mudanças significativas no sistema e devem ser detectadas rapidamente, enquanto as incipientes causam pequenas alterações no sistema, sendo de difíceis detecção. Portanto, dependendo do tipo de falha, pode ser necessário identificá-la rapidamente, a fim de evitar sérios danos ao sistema.

Uma das possibilidades de se detectar a ocorrência de uma falha em um sistema

híbrido é aplicando as ferramentas para o diagnóstico de falhas em sistema a eventos discretos na componente discreta extraída do sistema híbrido. Em sistemas a eventos discretos (SED), a falha a ser diagnosticada é um evento não observável, ou seja, um evento que não pode ser diretamente detectado por sensores. Um evento de falha pode ser diagnosticado se a sua ocorrência puder ser identificada após a ocorrência de um número limitado de eventos observáveis. Com o objetivo de diagnosticar o sistema, são construídos modelos a eventos discretos que capturem o comportamento normal anterior à ocorrência de eventos de falha e o comportamento pós-falha do sistema.

Em [5, 6], uma abordagem para o diagnóstico de falhas em SED modelados por autômatos finitos é apresentada. Nessa abordagem, é apresentada a metodologia para a obtenção de um diagnosticador centralizado que utiliza todos os eventos observáveis do sistema. O diagnosticador proposto pode ser usado para a verificação off-line da diagnosticabilidade do sistema, bem como, para o diagnóstico de falhas online. Entretanto, o espaço de estados do diagnosticador cresce exponencialmente com a cardinalidade do espaço de estados do modelo do sistema [5–8].

Um método para o diagnóstico online é apresentado em [7], cujo objetivo é evitar a construção e o completo armazenamento do autômato diagnosticador. Nessa abordagem, é apresentado um autômato não determinístico G_{nd} , que tem, no pior caso, espaço de estados com cardinalidade $2 \times |Q|$, no qual Q denota o espaço de estados da planta e $|\cdot|$ a cardinalidade. Nesse método, o próximo estado do diagnosticador é obtido após a ocorrência de um evento observável, considerando o estado atual do diagnosticador, bem como, o autômato G_{nd} . Entretanto, um método para a sua implementação em um computador não é apresentado em [7].

Em [9], outro método para verificação da diagnosticabilidade é apresentado. Neste método, é construído um autômato verificador, cujo espaço de estados cresce polinomialmente com a cardinalidade do espaço de estados e com o conjunto de eventos do sistema. Porém, autômatos verificadores não podem ser usados no diagnóstico online de falhas [8].

Todos os trabalhos citados anteriormente, realizam o diagnóstico de falhas na componente discreta de um sistema híbrido, analisando somente as sequências de eventos observáveis geradas pelo sistema, porém nem sempre é possível identificar se a falha ocorreu. Como o estado de um sistema híbrido (SH) é composto por uma parte discreta e uma contínua, chamadas, respectivamente, de modo e medida [1–3], é possível adicionar informações, que são obtidas da parte contínua, para se realizar o diagnóstico de falhas em um sistema híbrido.

Um método de diagnóstico de falhas é apresentado em [10], para uma classe de sistemas híbridos chamada *Real Time Hybrid System* (RTHS). Esse método mostra que a ausência da informação obtida das variáveis contínuas pode acarretar num diagnóstico de falhas incorreta. Nesse método, as transições entre os estados são classificadas como mensuráveis e não mensuráveis. Duas transições são indistinguíveis, se não for possível diferenciá-las através das suas dinâmicas contínuas. Com o objetivo de diagnosticar a falha, um diagnosticador é construído utilizando a relação de indistinguibilidade entre as transições. Apesar de evitar a completa discretização das variáveis contínuas, esse método ainda apresenta um alto nível de abstração da dinâmica contínua [11]. Outra desvantagem é em relação à necessidade de monitorar permanentemente as variáveis contínuas do sistema.

Outro método em [12] é apresentado para o diagnóstico de falhas em sistemas híbridos e baseia-se no conceito de assinatura de falha e na ordem das medições. A assinatura de falha representa os efeitos que a falha provoca nas medições das variáveis contínuas do estado, isto é, a diferença entre as variáveis contínuas medidas e as modeladas. Além disso, a falha pode provocar alterações distintas em cada variável contínua. Com o objetivo de diagnosticar o SH, uma sequência de medições é criada. Nessa sequência, as variáveis contínuas são inseridas, conforme seus efeitos são identificados, e rotuladas de acordo com o seu impacto no sistema contínuo, isto é, se os valores medidos das variáveis contínuas aumentam ou diminuem em relação ao modelo sem falha. Essa sequência de medições é chamada de ordem de medições. Em outras palavras, a ordem de medições modela a sequência de desvios de medição

causados pela ocorrência de falha, bem como, os seus efeitos. Assim, uma falha tem um comportamento característico modelado, conforme a ordem de medições, em cada estado, e este comportamento é chamado de assinatura de falha. Portanto, cada falha para ser diagnosticada deve ter uma assinatura própria. Nesse método, o diagnosticador identifica o estado em que o sistema se encontra e compara com as assinaturas de falha. Se o evento de falha σ_{f_1} tiver ocorrido, e o diagnosticador identificar que o comportamento contínuo no modo em falha é igual ao assinatura desta falha, e esta assinatura não se confunde com nenhuma outra assinatura de falha, o sistema é diagnosticável. Entretanto, esse método depende do número de variáveis contínuas disponíveis para medição, bem como, da necessidade de monitorar permanentemente o sistema, levando a um elevado custo computacional.

Em [11], o conceito de assinatura de modo é introduzido como uma extensão do conceito de assinatura de falha. A assinatura de modo é um vetor de comprimento igual ao número de modos, no qual cada modo é representado como um elemento desse vetor. Como o estado é composto por uma componente discreta, denominada de modo, e uma contínua, denominada de medida, é possível distinguir os estados através da componente contínua, e conseqüentemente distinguir os modos. Os valores um e zero nos componentes desse vetor representam os resultados das comparações das medições das variáveis contínuas dos estados do SH , nos quais os modos estão associados. Se as medições forem iguais, o valor zero é inserido nesse vetor. Caso contrário, o valor um é inserido. Portanto, cada modo possui uma assinatura de modo própria, que pode ser igual ou não ao de outro modo. Com o objetivo de diagnosticar o sistema híbrido, é criado um novo autômato, e as propriedades básicas para o diagnóstico de falhas em SED podem então ser aplicadas para este autômato. Entretanto, a análise da diagnosticabilidade que utiliza a assinatura de modo requer a geração de resíduos de todos os modos do SH, bem como a criação de modos transitórios e eventos de assinatura, resultando em um custo computacional excessivo.

Este trabalho apresenta uma nova definição para a diagnosticabilidade de um

sistema híbrido, chamada de h -diagnosticabilidade. Além disso, é apresentado um novo método para a verificação da h -diagnosticabilidade baseado na construção de um diagnosticador, que combina um diagnosticador de sistema a eventos discretos com a distinguibilidade dos modos de um sistema híbrido.

A diferença entre a abordagem deste trabalho e as que são propostas em [10, 12] é que a análise das variáveis contínuas é realizada em um intervalo de tempo limitado, diferentemente dos outros métodos, no qual a análise é ininterrupta. Este trabalho também se diferencia de [11] no sentido de que a análise das variáveis contínuas é realizada somente entre os estados alcançados pela sequência de eventos observada do sistema, e não em todos os estados.

Uma vantagem desta abordagem em relação aos trabalhos apresentados em [11, 12] é a redução do custo computacional para a verificação da distinguibilidade dos modos, e, conseqüentemente, para o diagnóstico de falhas em sistemas híbridos. Entretanto, é necessário um estudo específico para quantificar a redução deste custo computacional.

Esse trabalho também apresenta um exemplo de sistema híbrido que é o sistema de controle da máquina do leme de um navio, que é responsável por movimentar o leme e definir o rumo da embarcação. Para esse exemplo é construído um autômato híbrido, no qual é modelada primeiramente a componente discreta, e é acrescentada a esse modelo a componente contínua. Após a construção do autômato híbrido, é verificada que a linguagem gerada pela componente discreta desse sistema híbrido é não diagnosticável, porém ao utilizar o novo método para o diagnóstico de falhas em sistema híbrido, o referido exemplo torna-se h -diagnosticável.

Este trabalho está organizado da seguinte forma: no capítulo 2 são apresentados conceitos preliminares necessários para o entendimento desta dissertação, bem como uma revisão sobre SEDs modelados por autômatos. No capítulo 3, o diagnóstico de falhas em SED e uma análise da diagnosticabilidade de um autômato construído utilizando o mapa de sensores, são apresentados. Os fundamentos de um sistema híbrido e de um autômato híbrido, bem como um exemplo, são apresentados no

capítulo 4. Os conceitos de diagnóstico de falhas em sistemas híbridos são apresentados no capítulo 5, bem como, um exemplo para ilustrar os resultados obtidos. Finalmente, as conclusões são apresentadas no capítulo 6.

Capítulo 2

Sistemas a eventos discretos

Neste capítulo são apresentados fundamentos teóricos de sistemas a eventos discretos necessários para a compreensão e elaboração deste trabalho, como por exemplo para a modelagem da componente discreta de um sistema híbrido. Para tanto, neste capítulo são apresentados os formalismos matemáticos usualmente empregados para descrever sistemas a eventos discretos.

Quando o espaço de estado de um sistema é descrito por um conjunto discreto e as transições de estados ocorrem devido à ocorrência de eventos, o chamamos de sistema a eventos discretos (SED) [8, 13]. Os eventos podem estar associados a ações específicas (por exemplo, alguém aperta um botão, acionar um motor elétrico, etc) ou ser o resultado de diversas condições que são satisfeitas (uma peça atinge um determinado ponto de uma linha de produção). A ocorrência de um evento no sistema pode acarretar a transição para o mesmo estado ou para um estado distinto.

A modelagem de um SED deve ser capaz de reproduzir, dentro de limites de tolerância pré-estabelecidos, o comportamento do sistema. As trajetórias dos estados são descritas em função de uma sequência de eventos. Todas as sequências de eventos possíveis de serem geradas por um SED caracterizam a linguagem desse SED, sendo esta definida sobre o conjunto de eventos do sistema Σ . Para descrever o comportamento de um SED é necessário conhecer a sequência de eventos que causam as transições de estados. Desta forma, o modelo de um SED é composto basicamente de dois elementos: estados e eventos.

2.1 Linguagens

O conjunto de eventos Σ de um SED pode ser entendido como um alfabeto e a sequência de eventos formada com os elementos de Σ pode ser entendida como uma palavra. Uma palavra ou sequência que não possui nenhum evento é chamada de sequência vazia e é denotada por ε . O tamanho de uma sequência é baseado no número de eventos existentes nela, levando-se em consideração as múltiplas ocorrências de um mesmo evento. Se s é uma sequência, então o comprimento de s é denotado como $|s|$. Por convenção, o comprimento da sequência vazia ε é zero.

A linguagem é definida da seguinte forma [13]:

Definição 2.1 (*Linguagem*) *Uma linguagem definida sobre um conjunto de eventos Σ é um conjunto de sequências de comprimento finito formadas com os eventos de Σ .* □

Exemplo 2.1 *Os seguintes conjuntos são exemplos de linguagens definidas sobre $\Sigma = \{a, b, c\}$: $L_1 = \{aab, ac\}$ e $L_2 = \{ \text{todas as possíveis sequências de eventos de } \Sigma \text{ terminadas com } c \}$. Observe que L_1 tem apenas dois elementos e L_2 é infinita, porém enumerável.*

A concatenação, denotada por \cdot , é a operação utilizada para formar sequências. A sequência $a \cdot b \cdot b$ é a concatenação da sequência $a \cdot b$ com o evento b , e a sequência $a \cdot b$ é a concatenação do evento a com o evento b . Para simplificar a notação, a concatenação de duas sequências s_1 e s_2 será denotada por $s_1 s_2$.

Note que uma linguagem é um subconjunto do conjunto de todas as sequências de comprimento finito possíveis de serem formadas com os elementos de Σ , incluindo a sequência vazia ε . A esse conjunto dá-se o nome de fecho de Kleene de Σ , que é denotado por Σ^* [8, 13].

Considere a sequência de eventos $s \in \Sigma^*$, no qual $s = tuv$, com $t, u, v \in \Sigma^*$. Então, t é denominado de prefixo de s , u é denominado de subsequência de s e v é denominado de sufixo de s .

2.2 Operações com Linguagens

As operações de conjuntos, tais como união, interseção, diferença e complemento, são aplicáveis para linguagens definidas sobre o conjunto de eventos Σ , uma vez que uma linguagem é um conjunto. A seguir, são apresentadas outras operações com linguagens [8, 13].

Definição 2.2 (*concatenação*): Sejam $L_a, L_b \subseteq \Sigma^*$, a concatenação entre L_a e L_b é uma linguagem $L = L_a.L_b$ formada concatenando-se todas as seqüências de L_a com todas as seqüências de L_b , isto é,

$$L = L_a.L_b := \{s \in \Sigma^* : (s = s_a.s_b) \wedge (s_a \in L_a) \wedge (s_b \in L_b)\}.$$

□

Definição 2.3 (*Fecho de prefixo*): Seja $L \subseteq \Sigma^*$. O fecho de prefixo de uma linguagem L , denotado por \bar{L} , é o conjunto formado por todos os prefixos dos elementos de L , isto é,

$$\bar{L} := \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\}.$$

Em geral, $L \subseteq \bar{L}$. Uma linguagem L é dita ser prefixo-fechada se $L = \bar{L}$. □

Definição 2.4 (*Fecho de Kleene*): Seja $L \subseteq \Sigma^*$, o fecho de Kleene de L é definido como:

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

□

2.3 Projeções de seqüências e linguagens

Outra operação frequentemente utilizada em seqüências e linguagens é a operação de projeção natural, ou simplesmente projeção [14].

Definição 2.5 (*Projeção*) : A projeção $P_s : \Sigma_l^* \rightarrow \Sigma_s^*$, em que $\Sigma_s \subset \Sigma_l$, é definida recursivamente da seguinte forma:

$$P_s(\varepsilon) = \varepsilon,$$

$$P_s(\sigma) = \begin{cases} \sigma, & \text{se } \sigma \in \Sigma_s \\ \varepsilon, & \text{se } \sigma \in \Sigma_l \setminus \Sigma_s \end{cases},$$

$$P_s(s\sigma) = P_s(s)P_s(\sigma), \text{ para todo } s \in \Sigma_l^*, \sigma \in \Sigma_l,$$

em que \setminus denota a diferença entre conjuntos. □

De acordo com a definição 2.5, a operação de projeção aplicada sobre uma sequência de eventos $s \in \Sigma_l^*$ apaga os eventos de s que não pertencem a Σ_s , no qual Σ_s é o menor conjunto de eventos. Por exemplo, considere $\Sigma_l = \{a, b\}$, $\Sigma_s = \{a\}$ e a sequência de eventos $s = aababb$, em que $s \in \Sigma_l^*$. Ao realizar a operação de projeção em s , obtém-se a seguinte sequência de eventos $P_s(aababb) = aaa$.

É possível definir também a operação de projeção inversa da seguinte forma:

Definição 2.6 (*Projeção inversa*) : A projeção inversa $P^{-1} : \Sigma_s^* \rightarrow 2^{\Sigma_l^*}$ de uma sequência $t \in \Sigma_s^*$, é definida como:

$$P^{-1}(t) := \{s \in \Sigma_l^* : P(s) = t\}.$$

□

A projeção P e a sua inversa P^{-1} podem ser estendidas para linguagens. Para tanto, basta aplicar a operação para todas as sequências que formam a linguagem.

Definição 2.7 (*Projeção de uma Linguagem*) : Seja $L \subseteq \Sigma_l^*$,

$$P(L) := \{t \in \Sigma_s^* : (\exists s \in L)[P(s) = t]\}.$$

□

Definição 2.8 (*Projeção inversa de uma Linguagem*) : Seja $L_s \subseteq \Sigma_s^*$,

$$P^{-1}(L_s) := \{s \in \Sigma_l^* : (\exists t \in L_s)[P(s) = t]\}.$$

□

O exemplo a seguir tem o objetivo de ilustrar a aplicação das operações de projeção e de projeção inversa de linguagens.

Exemplo 2.2 *Sejam $\Sigma_l = \{a, b, c\}$, $\Sigma_s = \{a, b\}$ e a linguagem $L \subseteq \Sigma_l^*$, no qual $L = \{a, b, c, ac, cba, abc, abca\}$. Considere a projeção $P_s : \Sigma_l^* \rightarrow \Sigma_s^*$. Ao realizar a projeção P_s da linguagem L , deve-se apagar de todas as sequências de L , todos os eventos que pertencem ao conjunto $\Sigma_l \setminus \Sigma_s$. Neste caso, $\Sigma_l \setminus \Sigma_s = \{c\}$, logo, $P_s(L) = \{\varepsilon, a, b, ba, ab, aba\}$. Além disso, tem-se que $P_s^{-1}(\{aba\}) = \{c\}^*a\{c\}^*b\{c\}^*a\{c\}^*$.*

Uma linguagem deve ser entendida como uma maneira formal de descrever o comportamento de um SED, isto é, deve especificar todas as possíveis sequências de eventos que um SED é capaz de processar ou gerar. Contudo, nem sempre a representação de uma linguagem é fácil de especificar. É preciso um conjunto compacto de estruturas que defina linguagens e que possa ser manipulada com operações bem definidas, tal que seja possível construir, e conseqüentemente manipular e analisar, linguagens complexas. O modelo adotado neste trabalho para descrever SEDs será o autômato.

2.4 Autômato

Um autômato é um dispositivo capaz de representar linguagens de acordo com regras bem definidas [8].

Definição 2.9 (*Autômato Determinístico*): *Um autômato determinístico, denotado por G , é uma sêxtupla*

$$G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m),$$

no qual:

Q é conjunto de estados discretos;

Σ é o conjunto de eventos;

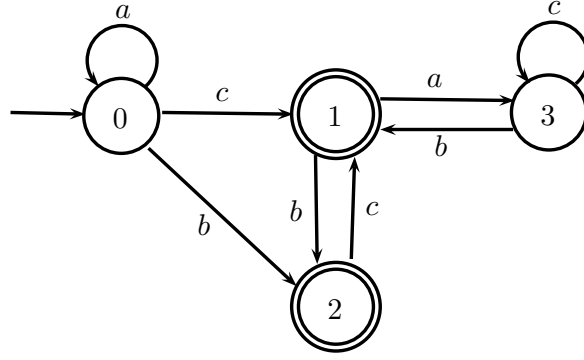


Figura 2.1: Diagrama de transição de estados do autômato G do Exemplo 2.3.

$\phi : Q \times \Sigma^* \rightarrow Q$ é a função de transição de estados;

$\Gamma : Q \rightarrow 2^\Sigma$ é a função de eventos ativos;

$q_0 \in Q$ é o estado inicial;

$Q_m \subseteq Q$ é o conjunto dos estados marcados. □

Os estados marcados se referem a estados de Q , nos quais se desejam dar um significado especial, como, por exemplo, o término de uma tarefa.

Os autômatos são representados graficamente através de diagramas de transição de estados. Nesses diagramas, os estados são representados por circunferências. O estado inicial é indicado por uma seta apontada para este, enquanto os estados marcados são identificados por duas circunferências concêntricas. Os estados são conectados entre si através de arcos direcionados com símbolos, que representam os eventos que determinam a mudança entre esses estados.

Exemplo 2.3 Seja o autômato $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$ em que $Q = \{0, 1, 2, 3\}$, $\Sigma = \{a, b, c\}$, $\phi(0, a) = \{0\}$, $\phi(0, b) = \{2\}$, $\phi(0, c) = \{1\}$, $\phi(1, a) = \{3\}$, $\phi(1, b) = \{2\}$, $\phi(2, c) = \{1\}$, $\phi(3, b) = \{1\}$, $\phi(3, c) = \{3\}$, $\Gamma(0) = \{a, b, c\}$, $\Gamma(1) = \{a, b\}$, $\Gamma(2) = \{c\}$, $\Gamma(3) = \{b, c\}$, $q_0 = \{0\}$ e $Q_m = \{1, 2\}$. A figura 2.1 apresenta o diagrama de transição de estados de G .

Como pode ser observado na figura 2.1, o estado inicial é o estado $\{0\}$, os estados marcados são os estados $\{1, 2\}$ e as transições de estado são indicadas pelas setas rotuladas, como, por exemplo, a transição do estado $\{0\}$ para o estado $\{1\}$ que ocorre

através do evento c .

2.4.1 Linguagens representadas por Autômatos

A seguir, são apresentadas as definições de caminho e de linguagem gerada e marcada por um autômato [8, 13].

Definição 2.10 (*Caminho*): Seja $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$. Um caminho em G é uma sequência da forma:

$$\mathcal{P} = (q_1, \sigma_1, q_2, \dots, q_{n-1}, \sigma_{n-1}, q_n),$$

no qual $q_i \in Q$, para $i = 1, 2, \dots, n$, $\sigma_i \in \Sigma$, e $q_{i+1} = \phi(q_i, \sigma_i)$ para $i = 1, 2, \dots, n-1$. □

Observação 2.1 (*Sequência de eventos associada a um caminho*): Seja $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$ e $\mathcal{P} = (q_1, \sigma_1, \dots, \sigma_{n-1}, q_n)$ um caminho definido em G . A sequência de eventos associada a \mathcal{P} é dada por $s = \sigma_1 \sigma_2 \dots \sigma_{n-1}$.

Definição 2.11 (*Linguagem gerada*): A linguagem gerada por $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$, é dada por:

$$\mathcal{L}(G) = \{s \in \Sigma^* : \phi(q_0, s) \text{ é definida}\}.$$

□

Definição 2.12 (*Linguagem marcada*): A linguagem marcada por $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$, é dada por:

$$\mathcal{L}_m(G) = \{s \in \mathcal{L}(G) : \phi(q_0, s) \in Q_m\}.$$

□

A linguagem $\mathcal{L}(G)$ é o conjunto formado por todas as sequências que podem ser executadas no autômato G , a partir do seu estado inicial q_0 , e permitido pelo

diagrama de transição de estados. Nota-se pela definição 2.11, que $\mathcal{L}(G)$ é prefixo-fechada.

A linguagem marcada $\mathcal{L}_m(G)$ é um subconjunto de $\mathcal{L}(G)$, e consiste somente das sequências s , tais que $\phi(q_0, s) \in Q_m$, isto é, as sequências que iniciam em q_0 e terminam em um estado marcado. A linguagem marcada $\mathcal{L}_m(G)$ não é, em geral, prefixo-fechada.

2.4.2 Operações com Autômatos

As operações com autômatos podem ser divididas em unárias e de composição. As operações unárias alteram o diagrama de transição de estados de um autômato e as operações de composição têm como finalidade representar o comportamento síncrono ou completamente síncrono de dois ou mais autômatos.

2.4.2.1 Operações Unárias

- Parte Acessível

Um estado q_i é dito ser acessível, se existir um caminho em G , no qual \mathcal{P} comece pelo estado inicial q_0 e termine no estado q_i . Caso contrário, o estado q_i não é acessível. Quando é tomada a parte acessível de G , todos os estados não acessíveis e as suas respectivas transições de estados são apagadas de G . A operação de tomar a parte acessível de G , será denotada por $Ac(G)$.

Definição 2.13 : *A parte acessível de G , $Ac(G)$, é um autômato:*

$$Ac(G) := (Q_{ac}, \Sigma, \phi_{ac}, \Gamma, q_0, Q_{ac,m}),$$

em que

$$Q_{ac} = \{q \in Q : (\exists s \in \Sigma^*)[\phi(q_0, s) = q]\},$$

$$Q_{ac,m} = Q_m \cap Q_{ac},$$

$$\phi_{ac} = \phi|_{Q_{ac} \times \Sigma^* \rightarrow Q_{ac}}.$$

A notação $\phi_{ac} = \phi|_{Q_{ac} \times \Sigma^* \rightarrow Q_{ac}}$ significa que ϕ está limitado ao domínio dos estados acessíveis Q_{ac} .

□

- Parte Coacessível

Um estado q_i é dito ser coacessível, se existir um caminho em G , no qual \mathcal{P} comece por q_i e termine em um estado marcado Q_m . Quando é tomada a parte coacessível de G , todos os estados não coacessíveis e as suas respectivas transições de estados são apagadas de G . A operação de tomar a parte coacessível de G , será denotada por $CoAc(G)$.

Definição 2.14 : A Parte Coacessível de G , $CoAc(G)$, é um autômato:

$$CoAc(G) := (Q_{coac}, \Sigma, \phi_{coac}, \Gamma, q_{0,coac}, Q_m),$$

em que

$$Q_{coac} = \{q \in Q : (\exists s \in \Sigma^*)[\phi(q, s) \in Q_m]\},$$

$$q_{0,coac} = \begin{cases} q_0, & \text{se } q_0 \in Q_{coac} \\ \text{indefinido,} & \text{caso contrário} \end{cases},$$

$$\phi_{coac} = \phi|_{Q_{coac} \times \Sigma^* \rightarrow Q_{coac}}.$$

A notação $\phi_{coac} = \phi|_{Q_{coac} \times \Sigma^* \rightarrow Q_{coac}}$ significa que ϕ está limitado ao domínio dos estados coacessíveis Q_{coac} .

□

- Operação Trim

Um autômato que é acessível e coacessível simultaneamente é dito ser *trim*.

Definição 2.15 : A operação trim de G , $Trim(G)$, é um autômato:

$$Trim(G) := CoAc[Ac(G)] = Ac[CoAc(G)].$$

□

2.4.2.2 Operações de Composição

Duas operações de composição são definidas para autômatos: produto, denotado por \times , e composição paralela, denotada por \parallel . A composição paralela é também chamada de composição síncrona e o produto de composição completamente síncrona.

Definição 2.16 (*Produto*) : *Sejam os autômatos $G_1 = (Q_1, \Sigma_1, \phi_1, \Gamma_1, q_{01}, Q_{m_1})$ e $G_2 = (Q_2, \Sigma_2, \phi_2, \Gamma_2, q_{02}, Q_{m_2})$, então a composição produto de G_1 e G_2 é o autômato*

$$G_1 \times G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \phi_{1 \times 2}, \Gamma_{1 \times 2}, (q_{01}, q_{02}), Q_{m_1} \times Q_{m_2})$$

no qual

$$\phi_{1 \times 2}((q_1, q_2), e) := \begin{cases} (\phi_1(q_1, e), \phi_2(q_2, e)), & \text{se } e \in \Gamma_1(q_1) \cap \Gamma_2(q_2) \\ \text{indefinido}, & \text{caso contrário.} \end{cases},$$

Além disso, $\Gamma_{1 \times 2}(q_1, q_2) = \Gamma_1(q_1) \cap \Gamma_2(q_2)$. □

Na operação produto, as transições nos dois autômatos têm que ser sempre sincronizadas em um evento comum, isto é, um evento pertencente a $\Sigma_1 \cap \Sigma_2$. Assim, um evento em $G_1 \times G_2$ irá ocorrer se e somente se o evento ocorrer simultaneamente em ambos os autômatos. Os estados de $G_1 \times G_2$ são denotados por pares, no qual a primeira componente é um estado de G_1 e a segunda componente é um estado de G_2 . Assim, tem-se que:

$$\begin{aligned} \mathcal{L}(G_1 \times G_2) &= \mathcal{L}(G_1) \cap \mathcal{L}(G_2), \\ \mathcal{L}_m(G_1 \times G_2) &= \mathcal{L}_m(G_1) \cap \mathcal{L}_m(G_2). \end{aligned}$$

Note que o conjunto de eventos de $G_1 \times G_2$ é definido como $\Sigma_1 \cup \Sigma_2$.

A operação produto é restritiva, pois as transições só ocorrem com os eventos em comum. Em geral, modelamos um sistema complexo a partir dos modelos dos seus componentes. O conjunto de eventos de cada componente inclui os seus eventos privados, que representam o seu comportamento interno, e os comuns que são compartilhados com outros autômatos e capturam a interação entre os respectivos componentes do sistema. A forma de construir um sistema complexo através dos modelos dos seus componentes é pela operação composição paralela.

Definição 2.17 (*Composição Paralela*): Sejam os autômatos G_1 e G_2 . A composição paralela de G_1 e G_2 é o autômato

$$G_1 \parallel G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \phi_{1\parallel 2}, \Gamma_{1\parallel 2}, (q_{01}, q_{02}), Q_{m_1} \times Q_{m_2})$$

no qual

$$\phi_{1\parallel 2}((q_1, q_2), e) := \begin{cases} (\phi_1(q_1, e), \phi_2(q_2, e)), & \text{se } e \in \Gamma_1(q_1) \cap \Gamma_2(q_2) \\ (\phi_1(q_1, e), q_2), & \text{se } e \in \Gamma_1(q_1) \setminus \Sigma_2 \\ (q_1, \phi_2(q_2, e)), & \text{se } e \in \Gamma_2(q_2) \setminus \Sigma_1 \\ \text{indefinido}, & \text{caso contrário} \end{cases}$$

Além disso, $\Gamma_{1\parallel 2}(q_1, q_2) = [\Gamma_1(q_1) \cap \Gamma_2(q_2)] \cup [\Gamma_1(q_1) \setminus \Sigma_2] \cup [\Gamma_2(q_2) \setminus \Sigma_1]$. \square

Na composição paralela, um evento comum, isto é, um evento pertencente a $\Sigma_1 \cap \Sigma_2$, só pode ocorrer se for executado em ambos os autômatos simultaneamente. Desta forma, os dois autômatos estão “sincronizados” nos eventos comuns. Os eventos privados de G_1 e G_2 , que são os eventos em $[\Gamma_1(q_1) \setminus \Sigma_2] \cup [\Gamma_2(q_2) \setminus \Sigma_1]$, não possuem esta limitação, podendo ocorrer sempre que for possível.

Para caracterizar as linguagens gerada e marcada por $G_1 \parallel G_2$ é preciso definir as projeções $P_i : (\Sigma_1 \cup \Sigma_2)^* \rightarrow \Sigma_i^*$ para $i = 1, 2$. Utilizando essas projeções, as linguagens resultantes da composição paralela são caracterizadas da seguinte forma:

$$\begin{aligned} \mathcal{L}(G_1 \parallel G_2) &= P_1^{-1}[\mathcal{L}(G_1)] \cap P_2^{-1}[\mathcal{L}(G_2)], \\ \mathcal{L}_m(G_1 \parallel G_2) &= P_1^{-1}[\mathcal{L}_m(G_1)] \cap P_2^{-1}[\mathcal{L}_m(G_2)]. \end{aligned}$$

As operações de composição produto e paralela são ilustradas no exemplo a seguir.

Exemplo 2.4 Sejam G_1 e G_2 autômatos, cujos diagramas de transição de estados são apresentados, respectivamente, nas figuras 2.2.(a) e 2.2.(b). As composições produto e paralela desses autômatos são representadas pelos autômatos $G_1 \times G_2$ e $G_1 \parallel G_2$, mostrados nas figuras 2.3.(a) e 2.3.(b), respectivamente.

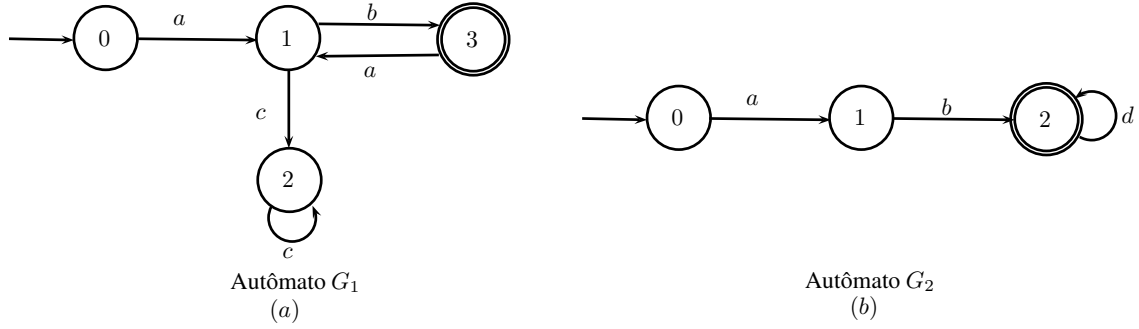


Figura 2.2: Autômatos G_1 e G_2 do Exemplo 2.4.

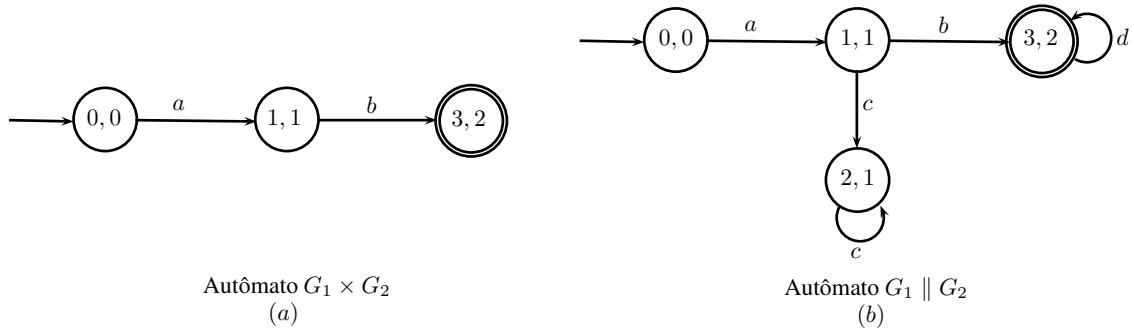


Figura 2.3: Autômatos $G_1 \times G_2$ e $G_1 \parallel G_2$ do Exemplo 2.4.

2.4.3 Autômatos com observação parcial

Inicialmente, é necessário classificar os eventos em observáveis e não-observáveis [8, 13]. Um evento é observável quando sua ocorrência pode ser registrada e comunicada ao observador do sistema, caso contrário o evento é dito ser não-observável.

Suponha que o conjunto de eventos Σ seja particionado como $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_o e Σ_{uo} denotam, respectivamente, o conjunto de eventos observáveis e não-observáveis. Quando o autômato de G do sistema possui eventos não-observáveis diz-se que G é um autômato com observação parcial.

O comportamento observado de um autômato G com observação parcial pode ser descrito por um autômato determinístico, denominado observador ou $Obs(G)$, cujo conjunto de eventos é formado pelos eventos observáveis Σ_o . Os estados do observador são todos os estados em que G pode estar após a observação de uma sequência de eventos observáveis.

Antes de apresentar o algoritmo do observador, é necessário introduzir o conceito

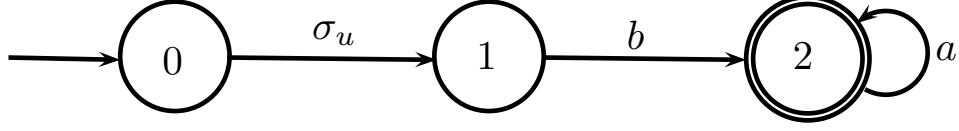


Figura 2.4: Autômato G com evento não-observável do Exemplo 2.5.

de alcance não-observável de um estado $q_i \in Q$, denotado por $UR(q_i)$ [8, 13].

Definição 2.18 (*Alcance não-observável*):

$$UR(q_i) = \{q_j \in Q : (\exists t \in \Sigma_{uo}^*)[\phi(q_i, t) = q_j]\}.$$

□

O alcance não observável de um estado q_i , $UR(q_i)$, gera um conjunto de estados que são alcançados por eventos ou sequências não-observáveis a partir do estado q_i . Esta definição pode ser estendida para um subconjunto de estados $\Theta \subseteq Q$ como:

$$UR(\Theta) = \bigcup_{q_i \in \Theta} UR(q_i). \quad (2.1)$$

Exemplo 2.5 *Seja G representado pelo diagrama de transição de estados, mostrado na figura 2.4, em que $\Sigma_o = \{a, b\}$ e $\Sigma_{uo} = \{\sigma_u\}$. Note que o estado 1 é alcançado a partir do estado 0 por um evento não observável. Desta forma, o alcance não-observável do estado 0 é formado pelo próprio estado 0 e pelo estado 1, ou seja, $UR(0) = \{0, 1\}$.*

A definição do observador de G ou $Obs(G)$ é dada a seguir [13].

Definição 2.19 (*Observador*) *O observador de um autômato G em relação a um conjunto de eventos observáveis Σ_o , denotado por $Obs(G)$, é dado por:*

$$Obs(G) = (Q_{obs}, \Sigma_o, \phi_{obs}, \Gamma_{obs}, q_{0,obs}, Q_{m,obs}),$$

em que $Q_{obs} \subseteq 2^Q$ e $Q_{m,obs} = \{B \in Q_{obs} : B \cap Q_m \neq \emptyset\}$. ϕ_{obs}, Γ_{obs} e $q_{0,obs}$ são obtidos de acordo com o algoritmo 2.1. □

A seguir, é apresentado o algoritmo de construção do observador de G [8].

Algoritmo 2.1 *Seja $G = (Q, \Sigma, \phi, \Gamma, q_0, Q_m)$ um autômato determinístico, em que $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$. Então, $Obs(G) = (Q_{obs}, \Sigma_o, \phi_{obs}, \Gamma_{obs}, q_{0,obs}, Q_{m,obs})$ é construído da seguinte forma:*

- *Passo 1: Defina $q_{0,obs} = UR(q_0)$. Faça $Q_{obs} = \{q_{0,obs}\}$ e $\tilde{Q}_{obs} = Q_{obs}$.*
- *Passo 2: $\bar{Q}_{obs} = \tilde{Q}_{obs}$ e $\tilde{Q}_{obs} = \emptyset$.*
- *Passo 3: Para cada $B \in \bar{Q}_{obs}$,*
 - *Passo 3.1: $\Gamma_{obs}(B) = (\bigcup_{q \in B} \Gamma(q)) \cap \Sigma_o$.*
 - *Passo 3.2: Para cada $e \in \Gamma_{obs}(B)$,*

$$\phi_{obs}(B, e) = UR(\{q \in Q : (\exists \tilde{q} \in B)[q = \phi(\tilde{q}, e)]\}).$$
 - *Passo 3.3: $\tilde{Q}_{obs} \leftarrow \tilde{Q}_{obs} \cup \phi_{obs}(B, e)$.*
- *Passo 4: $Q_{obs} \leftarrow Q_{obs} \cup \tilde{Q}_{obs}$.*
- *Passo 5: Repita os passos 2 a 4 até que toda a parte acessível de $Obs(G)$ tenha sido construída.*
- *Passo 6: $Q_{m,obs} = \{B \in Q_{obs} : B \cap Q_m \neq \emptyset\}$.*

Exemplo 2.6 *Seja G representado pelo diagrama de transição de estados mostrado na figura 2.5, em que $\Sigma = \{a, b, c, \sigma_u\}$, $\Sigma_o = \{a, b, c\}$ e $\Sigma_{uo} = \{\sigma_u\}$. O observador de G é apresentado na figura 2.6. Note que antes da ocorrência de um evento observável, não é possível precisar se o sistema encontra-se no estado inicial 0 ou no estado 1, uma vez que a ocorrência do evento σ_u não pode ser registrada. Assim, o estado inicial do observador é $\{0, 1\}$. Caso o próximo evento observável seja o evento b , pode-se afirmar que o autômato estará no estado 4. Se ao invés de ocorrer o evento b , ocorrer o evento c , pode-se afirmar que o autômato estará ou no estado 2, ou no 3, cuja representação é dada por $\{2, 3\}$. Se ocorrer novamente o evento observável c , o próximo estado do observador será o mesmo, isto é, o estado $\{2, 3\}$.*

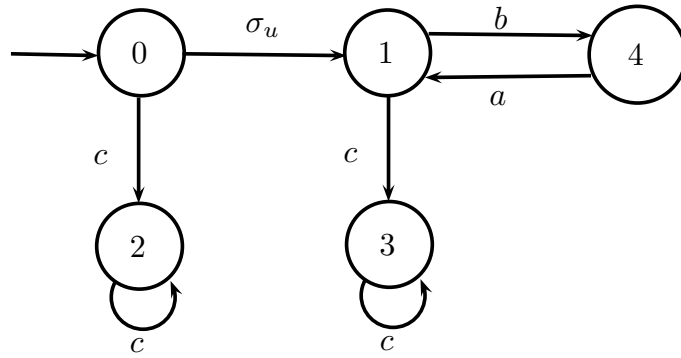


Figura 2.5: Autômato G do Exemplo 2.6.

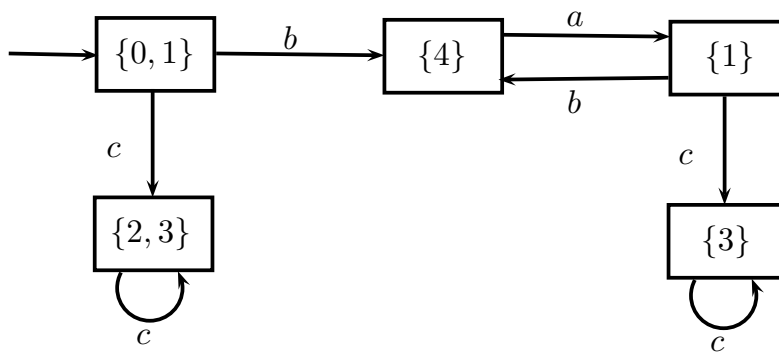


Figura 2.6: Autômato $Obs(G)$ do Exemplo 2.6.

Desta forma, não é possível identificar em qual estado o autômato se encontra após a observação da sequência $s = cc$.

Capítulo 3

Diagnóstico de falhas em SED

Em muitas aplicações no qual o sistema modelado contém eventos não-observáveis, pode-se estar interessado em determinar se certos eventos não-observáveis ocorreram na sequência de eventos executada pelo sistema. Se esses eventos não-observáveis modelam falhas do sistema, então o conhecimento da ocorrência de um desses eventos é muito importante, quando monitoramos o desempenho do sistema. Neste capítulo, o problema de diagnóstico de falhas em sistema a eventos discretos é analisado com a finalidade de verificar a diagnosticabilidade da componente discreta de um sistema híbrido.

3.1 Diagnóstico de falhas em SED

O conjunto de eventos de falha é denominado Σ_f . Supõe-se, sem perda de generalidade, que $\Sigma_f \subseteq \Sigma_{uo}$, uma vez que se o evento de falha fosse observável poderia ser trivialmente diagnosticado. O objetivo do diagnóstico de falhas em SED é identificar a ocorrência de qualquer evento de falha, analisando somente as sequências de eventos observáveis geradas pelo sistema.

As seguintes hipóteses são usualmente feitas nos trabalhos envolvendo o diagnóstico de falhas em SEDs [5, 8]:

H1 A linguagem L gerada por G é viva, ou seja, existe uma transição de estado definida para cada estado $q \in Q$, isto é, uma função de eventos ativos $\Gamma(q_i) \neq \emptyset$

para todo $q_i \in Q$.

H2 Não existe no autômato G nenhum ciclo formado somente por eventos não-observáveis, isto é,

$$\exists n_o \in \mathbb{N} \text{ tal que } \forall ust \in L, s \in \Sigma_{uo}^* \Rightarrow |s| \leq n_o$$

em que $|s|$ denota o comprimento da sequência s .

A hipótese H1 é feita considerando que o sistema está sempre em contínua operação, isto é, não para. A hipótese H2 é necessária para evitar que a ocorrência do evento de falha não seja detectada caso o sistema contenha um ciclo de estados ligados por eventos não-observáveis após a sua ocorrência. Neste trabalho, serão considerados as hipóteses H1 e H2.

Neste trabalho considera-se a existência de um único tipo de falha, isto é, $\Pi_f = \{\Sigma_f\}$, em que $\Sigma_f = \{\sigma_f\}$, e σ_f é o evento de falha. A existência de uma única falha não limita a abrangência deste trabalho, pois esta consideração pode ser estendida para os diversos tipos de falha, no qual cada falha será rotulada de forma diferente e tratada independentemente, ou seja, ao realizar o diagnóstico de falha de um determinado tipo, os eventos de falha de outros tipos são considerados eventos não observáveis comuns que não representam uma falha do sistema.

Após as considerações iniciais, é necessário apresentar os conceitos de sequência normal e de falha, bem como de linguagem normal e de falha.

Definição 3.1 *Uma sequência de falha é uma sequência de eventos que contém o evento de falha σ_f . Caso contrário, a sequência é dita ser normal.*

Definição 3.2 *Seja L a linguagem viva e prefixo fechada gerada por G . Então, a linguagem normal de G , L_N , é uma linguagem prefixo fechada formada por todas as sequências de L que não contêm o evento de falha σ_f .*

3.2 Diagnosticabilidade de falhas em SED

Uma linguagem L é dita ser diagnosticável se é possível detectar a ocorrência do evento de falha após um número limitado de observações de eventos. Conforme [5, 8], a definição formal da diagnosticabilidade de uma linguagem é dada a seguir.

Definição 3.3 : *Seja L a linguagem viva e prefixo fechada gerada por G e seja $L_N \subset L$ a linguagem normal de G . Então L é dita ser diagnosticável com relação à projecção $P_o : \Sigma^* \rightarrow \Sigma_o^*$ e ao conjunto de eventos de falha $\Sigma_f = \{\sigma_f\}$ se*

$$(\exists n \in \mathbb{N})(\forall s \in L \setminus L_N)(\forall st \in L \setminus L_N, |t| \geq n) \Rightarrow D,$$

sendo a condição D dada por

$$(\forall w \in P_o^{-1}(P_o(st)) \cap L, w \in L \setminus L_N).$$

□

Seja s uma sequência qualquer gerada pelo sistema que termina com um evento de falha σ_f , e seja t uma sequência arbitrariamente longa após s . A condição D requer que todas as sequências, cujas projeções em Σ_o são iguais à projeção de s , sejam sequências de falha, isto é, não exista nenhuma sequência de falha que tenha mesma observação que uma sequência normal.

3.3 Diagnosticador

Com a finalidade de realizar o diagnóstico de falhas, a partir da observação do comportamento do sistema online, e para verificar se a linguagem gerada por um autômato G é diagnosticável, pode ser utilizado um autômato determinístico denominado diagnosticador. O diagnosticador de G será denotado neste trabalho por $G_d = (Q_d, \Sigma_o, \phi_d, \Gamma_d, q_{o_d})$.

Assim como os observadores, os diagnosticadores fornecem uma estimativa dos estados do sistema. Contudo, diferentemente do observador, os estados de G são

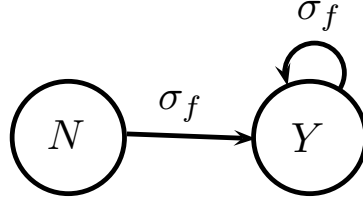


Figura 3.1: Autômato rotulador A_i^{YN}

rotulados para informar se o estado foi alcançado após um evento de falha. Outra diferença é que no diagnosticador, o conjunto dos estados marcados não é definido.

Para a construção do diagnosticador é necessário utilizar um autômato rotulador, conforme ilustrado na figura 3.1, e o diagnosticador pode ser obtido de acordo com o algoritmo a seguir.

Algoritmo 3.1

Passo 1 *Obtenha o autômato $G_l = G \parallel A_i^{YN}$, no qual A_i^{YN} é o autômato rotulador de estados mostrado na figura 3.1;*

Passo 2 *Calcule $G_d = Obs(G_l)$.* □

Note que os estados de G_l , obtidos do passo 1 do algoritmo 3.1, são da forma (q, N) e (q, Y) . Os estados que são alcançados por uma sequência que contenha o evento de falha serão da forma (q, Y) , caso contrário serão da forma (q, N) . Note ainda que um estado de G_d , é formado por um conjunto de estados de G_l fornecendo a estimativa de estado do sistema após a observação de uma sequência de eventos gerada pelo sistema. A construção de um diagnosticador será ilustrada no exemplo a seguir.

Exemplo 3.1 *Seja G o autômato da planta, cujo diagrama de transição de estados está representado na figura 3.2. Seguindo o passo 1 do algoritmo 3.1 obtém-se o autômato G_l apresentado na figura 3.3 e fazendo $G_d = Obs(G_l)$ obtém-se o diagnosticador de G apresentado na figura 3.4.*

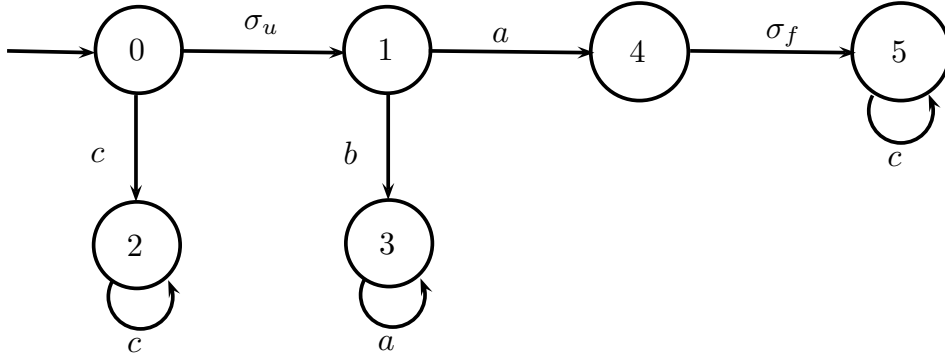


Figura 3.2: Autômato G do Exemplo 3.1.

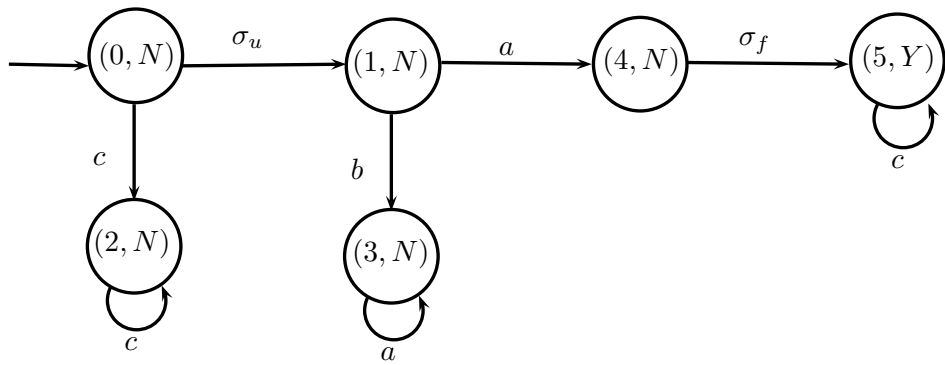


Figura 3.3: Autômato G_l do Exemplo 3.1.

Note que o estado inicial de G_d , $q_{0_d} = \{0N, 1N\}$, é o resultado do alcance não observável do estado inicial de G_l , $q_{0_l} = \{0N\}$, isto é, $UR(0N) = \{0N, 1N\}$.

Note ainda que existem dois caminhos em G_l , $\mathcal{P}_1 = (0N, \sigma_u, 1N, a, 4N)$ e $\mathcal{P}_2 = (0N, \sigma_u, 1N, a, 4N, \sigma_f, 5Y)$, cujas sequências de eventos associada a esses caminhos são, respectivamente, $s_1 = \sigma_u a$ e $s_2 = \sigma_u a \sigma_f$. Os estados alcançados por s_1 e s_2 são, respectivamente, $(4N)$ e $(5Y)$. Ao realizar a operação de projeção em s_1 e s_2 , obtêm-se as seguintes sequências de eventos $P_o(s_1) = P_o(\sigma_u a) = a$ e $P_o(s_2) = P_o(\sigma_u a \sigma_f) = a$. Como $P_o(s_1) = P_o(s_2) = a$, então não é possível identificar se o sistema encontra-se no estado $(4N)$ ou $(5Y)$. Essa incerteza em relação aos estados $(4N)$ ou $(5Y)$ é representada no diagnosticador pelo estado $\{4N, 5Y\}$

Os estados do diagnosticador podem ser classificados, quanto à presença de rótulos Y e N , da seguinte forma [5, 8].

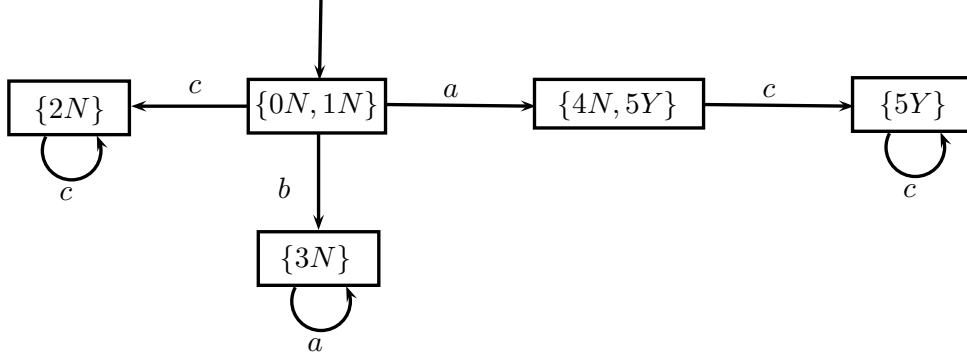


Figura 3.4: Autômato G_d do Exemplo 3.1.

Definição 3.4 Um estado $q_d \in Q_d$ é dito ser um estado de falha (certo), se $\ell = Y$ para todo $(q, \ell) \in q_d$, e um estado normal (ou sem falha) se $\ell = N$ para todo $(q, \ell) \in q_d$. Se existir $(q_i, \ell_i), (q_j, \ell_j) \in q_d$, q_i não necessariamente distinto de q_j , tal que $\ell_i = Y$ e $\ell_j = N$, então q_d é um estado incerto de G_d . \square

A partir das definições 3.3 e 3.4 pode-se concluir que a linguagem gerada por G será diagnosticável com relação a P_o e Σ_f se, e somente se, o diagnosticador sempre alcançar um estado certo após um número limitado de ocorrência de eventos após a ocorrência de um evento de falha σ_f . Se existir uma sequência de comprimento arbitrariamente longo após a ocorrência de σ_f em G , tal que a projeção desta sequência alcance somente estados incertos em G_d , então a linguagem gerada por G será não diagnosticável com relação a P_o e Σ_f . A seguir, as definições de ciclo e de ciclo indeterminado são apresentadas [5, 8].

Definição 3.5 Um conjunto de estados $\{q_1, q_2, \dots, q_n\} \subseteq Q$ forma um ciclo em G se existir em G_d um caminho cíclico $\mathcal{P} = (q_1, \sigma_1, q_2, \sigma_2, \dots, \sigma_{n-1}, q_n, \sigma_n, q_1)$. \square

Definição 3.6 Um conjunto de estados incertos $\{q_{d_1}, q_{d_2}, \dots, q_{d_n}\} \in Q_d$ forma um ciclo indeterminado no diagnosticador G_d , se as seguintes condições forem verdadeiras:

- i) existe um caminho cíclico $\mathcal{P}_d = (q_{d_1}, \sigma_1, q_{d_2}, \sigma_2, \dots, q_{d_n}, \sigma_n, q_{d_1})$ em G_d ;
- ii) $\exists (q_l^{k_l}, \ell_l^{k_l}), (\tilde{q}_l^{r_l}, \tilde{\ell}_l^{r_l}) \in q_{d_l}$, $q_l^{k_l}$ não necessariamente distinto de $\tilde{q}_l^{r_l}$, $l = 1, 2, \dots, n$, $k_l = 1, 2, \dots, m_l$, e $r_l = 1, 2, \dots, \tilde{m}_l$ tal que

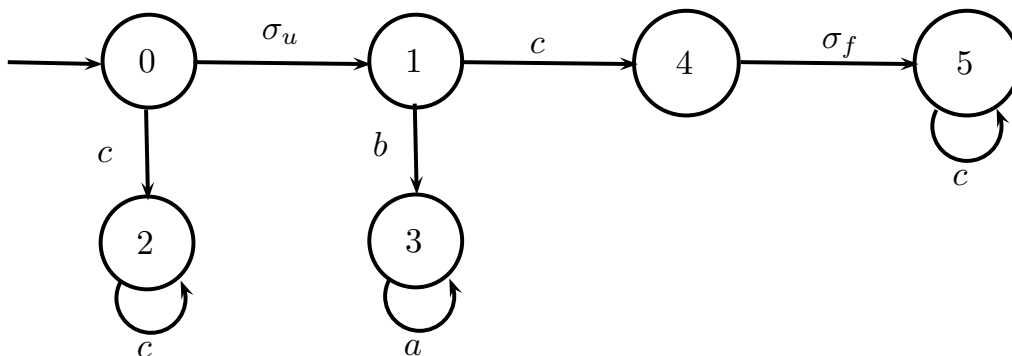


Figura 3.5: Autômato G do Exemplo 3.3.

a) $\ell_l^{k_l} = Y$, $\tilde{\ell}_l^{r_l} = N$, para todo l , k_l e r_l ;

b) Os estados $\{q_l^{k_l}\}$, $l = 1, 2, \dots, n$, $k_l = 1, 2, \dots, m_l$ e $\{\tilde{q}_l^{r_l}\}$, $l = 1, 2, \dots, n$, $r_l = 1, 2, \dots, \tilde{m}_l$ podem ser rearranjados para formar ciclos em G , cujas sequências correspondentes s e \tilde{s} , respectivamente, satisfazem $P_o(s) = P_o(\tilde{s}) = \bar{s}$, em que $\bar{s} = \sigma_1\sigma_2\dots\sigma_n$, no qual $\sigma_1, \sigma_2, \dots, \sigma_n$ são definidos em i). \square

Usando as definições 3.3, 3.5 e 3.6, pode-se enunciar a seguinte condição necessária e suficiente para a diagnosticabilidade de uma linguagem.

Teorema 3.1 *A linguagem viva L , gerada por uma autômato G , será diagnosticável com relação à projeção P_o e $\Sigma = \{\sigma_f\}$ se, e somente se, o seu diagnosticador G_d não tiver ciclos indeterminados.*

Prova: A prova deste teorema é apresentada em [5] \blacksquare

Exemplo 3.2 *Considere o autômato G da figura 3.2 e o seu diagnosticador, mostrado na figura 3.4, ambos obtidos no exemplo 3.1. Observe que G_d não possui ciclos indeterminados, e, portanto, pode-se concluir que a linguagem L é diagnosticável com relação a P_o e $\Sigma_f = \{\sigma_f\}$.*

Exemplo 3.3 *Considere agora o autômato G apresentado na figura 3.5, em que $\Sigma_o = \{a, b, c\}$ e $\Sigma_{uo} = \{\sigma_u, \sigma_f\}$. Os autômatos G_l e G_d são apresentados nas figuras 3.6 e 3.7, respectivamente.*

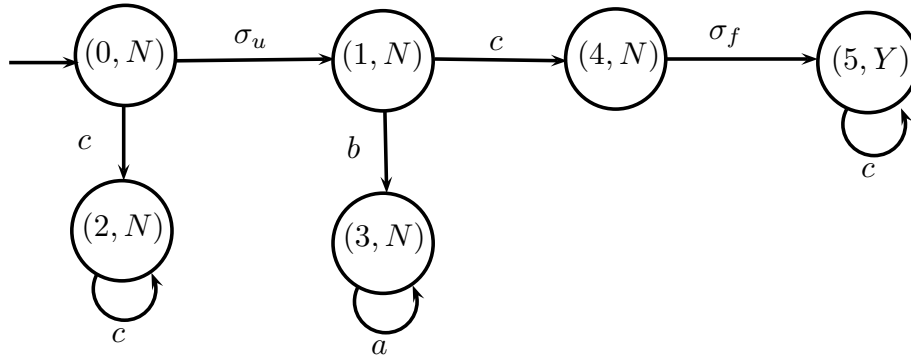


Figura 3.6: Autômato G_l do Exemplo 3.3.

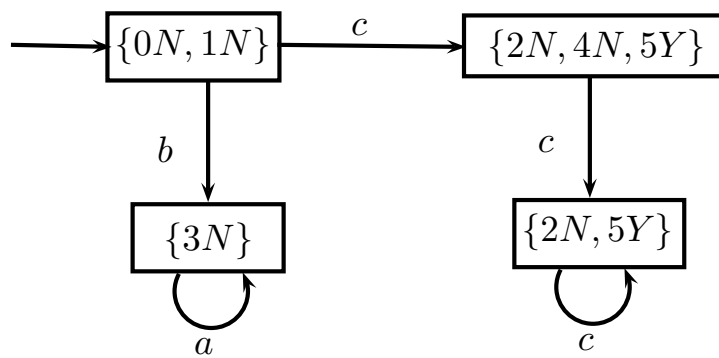


Figura 3.7: Autômato G_d do Exemplo 3.3.

Note que existe um caminho cíclico $\mathcal{P}_d = (\{2N, 5Y\}, c, \{2N, 5Y\})$ em G_d , cuja sequência associada a \mathcal{P}_d é $s_d = c$, e os caminhos normal e de falha em G_l , são, respectivamente, $\mathcal{P}_{l_N} = (2N, c, 2N)$ e $\mathcal{P}_{l_f} = (5Y, c, 5Y)$, cujas sequências associadas a \mathcal{P}_{l_N} e \mathcal{P}_{l_f} são $s_{l_N} = c$ e $s_{l_f} = c$. Como $P_o(s_{l_N}) = P_o(s_{l_f}) = s_d = c$, então existe um ciclo indeterminado em G_d , e portanto a linguagem L , gerada por G , é não diagnosticável com relação a P_o e Σ_f .

O diagnosticador é uma das principais ferramentas para se realizar o diagnóstico de falhas em SED. Atualmente, existem outros trabalhos nessa área que abordam, por exemplo, o diagnóstico robusto de falhas em SED [15–17], o diagnóstico descentralizado [8, 13, 18], o diagnóstico descentralizado robusto [19] e o diagnóstico online de falhas [20–23].

Todos os trabalhos citados anteriormente, realizam o diagnóstico de falhas em SED, analisando somente as sequências de eventos observáveis geradas pelo sistema, porém nem sempre é possível identificar se a falha ocorreu. Desta forma, outros métodos deverão ser utilizados para o diagnóstico de falhas, como, por exemplo, diagnóstico de falhas utilizando mapa de sensores [5, 13]. Esse método utiliza, simultaneamente, as informações das variáveis contínuas e as sequências de eventos observáveis geradas pelo sistema para o diagnóstico de falhas.

3.4 Diagnóstico de falhas utilizando mapa de sensores

No diagnóstico de falhas utilizando mapa de sensores, o autômato de Moore, que é um autômato com características especiais, é convertido em um autômato comum. O autômato de Moore é um autômato com uma função de saída. Essa função atribui um valor de saída para cada estado discreto. Esse valor é “emitido” pelo autômato quando um dos estados desse autômato é alcançado.

A conversão do autômato de Moore para um autômato comum consiste em renomear as transições da seguinte forma: a transição do estado q_i para q_j devido

à ocorrência do evento a , no qual o evento a é renomeado como um novo evento, $(a, \text{saída do estado } q_j)$. Assim, a informação sobre a saída de um estado é inserida no evento que ocasionou essa transição.

Caso o evento a seja observável, e esse evento acontece e é imediatamente seguido pela medição de um sensor, cujo valor medido é igual ao valor de saída do estado alcançado por esse evento, a conversão do evento a para $(a, \text{saída do estado } q_j)$ está correta. Contudo, essa conversão é inadequada se o evento a é um evento não observável. Porém, apesar do evento não ser observável, é possível medir a saída do estado alcançado por esse evento. Assim, para renomear um evento não-observável σ_{uo} , dois casos precisam ser considerados:

1. os valores de saída de q_i e de q_j são iguais; e
2. os valores de saída de q_i e de q_j são diferentes;

No primeiro caso, nenhuma mudança é observada no valor de saída do estado q_j em relação ao de q_i . Assim, o evento σ_{uo} não incorpora o valor de saída de q_j , mantendo a sua representação.

No segundo caso, é observada uma mudança no valor de saída do estado q_j em relação ao de q_i , após a ocorrência do evento não-observável. Então, é necessário criar um novo estado e definir duas transições de estado para representar essa mudança. O estado criado é representado por q_{q_i, σ, q_j} , e as transições de estado são definidas a seguir:

1. evento não-observável σ_{uo} de q_i para q_{q_i, σ, q_j} ; e
2. evento observável de q_{q_i, σ, q_j} para q_j , denominado como a alteração no valor de saída de q_i para q_j .

Na prática, os valores de saída são o resultado dos valores medidos pelos sensores.

O exemplo a seguir ilustra a construção de um modelo utilizando-se o mapa de sensores, bem como o seu diagnosticador. O sistema escolhido é o de aquecimento de água modelado por SED, cuja linguagem gerada pelo seu diagnosticador é não

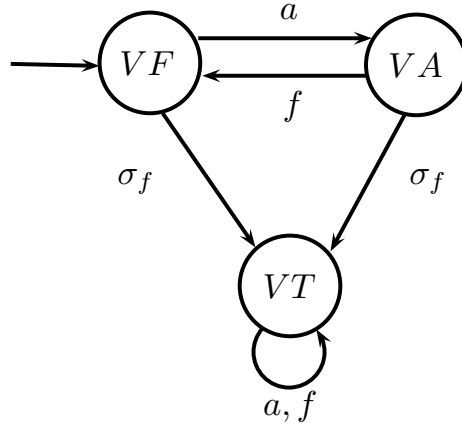


Figura 3.8: Autômato da válvula G_v do Exemplo 3.4.

diagnosticável. Entretanto, ao analisar a linguagem gerada pelo diagnosticador do modelo construído utilizando o mapa de sensores, essa linguagem é diagnosticável.

Exemplo 3.4 *Considere o sistema de aquecimento de água modelado por SED [13], composto por um aquecedor, uma bomba, uma válvula, um controlador, uma bobina de aquecimento e dois sensores, um de pressão e outro de vazão. Para simplificar o sistema, o aquecedor e a bobina não são modelados. Suponha que a saída dos sensores de pressão e de vazão são discretizadas. O sensor de pressão informa que a saída é “com pressão” ou “sem pressão”, enquanto o sensor de vazão informa que a saída é “com vazão” ou “sem vazão”.*

Os modelos da válvula, da bomba e do controlador são ilustrados nas figuras 3.8, 3.9 e 3.10, respectivamente. A válvula é modelada, considerando a possibilidade de ocorrer um evento de falha. Essa falha provoca o travamento da válvula na posição fechada. Os estados do autômato da válvula, G_v , são descritos como: válvula fechada (VF), válvula aberta (VA) e válvula travada na posição fechada (VT). No autômato da bomba G_b , os estados são bomba ligada (BL) e bomba desligada (BD), enquanto no controlador G_c , os estados são: controlador no estágio 1 (C1), estágio 2 (C2), estágio 3 (C3) e estágio 4 (C4).

Os eventos são divididos em $\Sigma_o = \{a, f, d, l\}$ e $\Sigma_{uo} = \{\sigma_f\}$, e representam:

- *a é abrir válvula;*

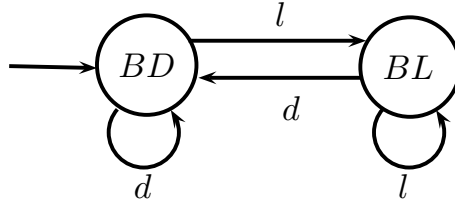


Figura 3.9: Autômato da bomba G_b do Exemplo 3.4.

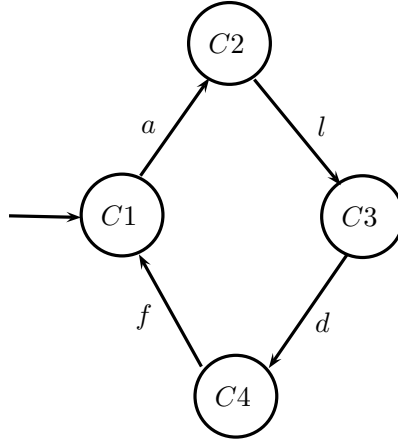


Figura 3.10: Autômato do controlador G_c do Exemplo 3.4.

- f é fechar válvula;
- l é ligar bomba;
- d é desligar bomba;
- σ_f é a falha na válvula, travada na posição fechada.

O autômato $G = G_v || G_b || G_c$ representa o funcionamento do sistema, e é mostrado na figura 3.11.

A seguir será construído o diagnosticador de G , com o objetivo de verificar se a linguagem L , gerada por G , é diagnosticável com relação a P_o e Σ_f . Antes de construir o diagnosticador G_d , os estados de G serão renomeados conforme tabela 3.1. A construção de G_d é ilustrada na figura 3.12.

Note que G_d possui um ciclo indeterminado e, portanto, a linguagem L , gerada por G , é não diagnosticável com relação a P_o e Σ_f . Então, será construído um

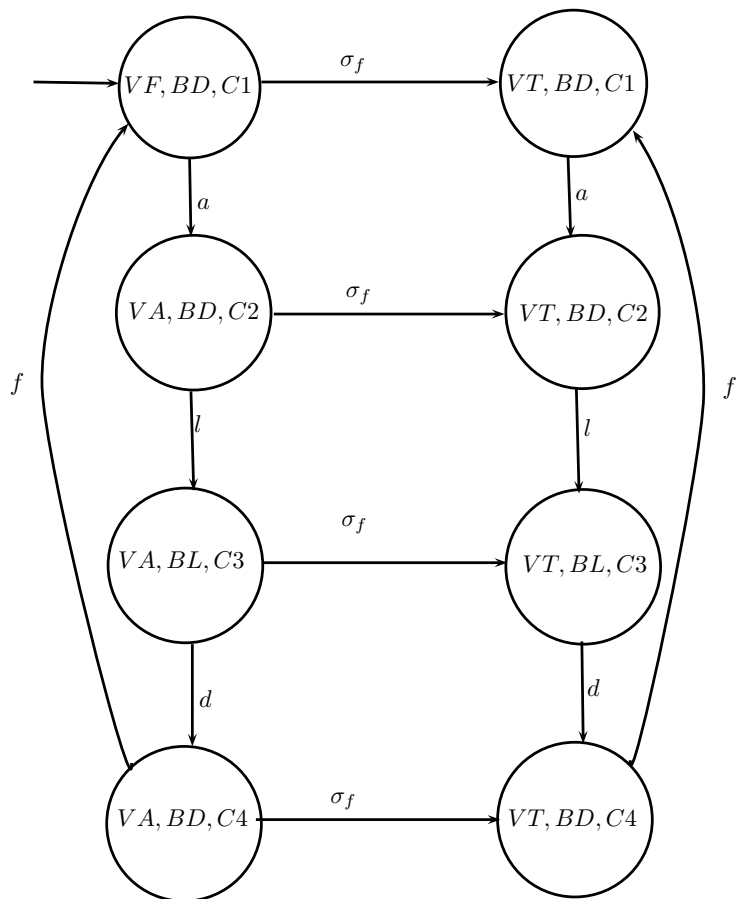


Figura 3.11: Autômato $G = G_v || G_b || G_c$ do Exemplo 3.4.

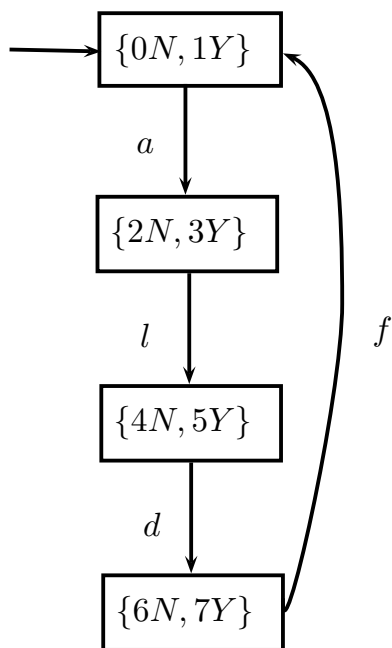


Figura 3.12: Autômato G_d do Exemplo 3.4.

Tabela 3.1: Tabela dos estados de G do Exemplo 3.4.

Q	Estado de G_v	Estado de G_b	Estado de G_c
0	VF - válvula fechada	BD - bomba desligada	C1 - controlador no estágio 1
1	VT - válvula travada	BD - bomba desligada	C1 - controlador no estágio 1
2	VA - válvula aberta	BD - bomba desligada	C2 - controlador no estágio 2
3	VT - válvula travada	BD - bomba desligada	C2 - controlador no estágio 2
4	VA - válvula aberta	BL - bomba ligada	C3 - controlador no estágio 3
5	VT - válvula travada	BL - bomba ligada	C3 - controlador no estágio 3
6	VA - válvula aberta	BD - bomba desligada	C4 - controlador no estágio 4
7	VT - válvula travada	BD - bomba desligada	C4 - controlador no estágio 4

autômato utilizando o mapa de sensores com o objetivo de diagnosticar a falha neste sistema.

Inicialmente, é importante notar na tabela 3.1, que cada estado em G possui um valor de pressão e vazão característicos. Por exemplo, no estado 0, a válvula encontra-se fechada e a bomba desligada, tendo pressão e vazão nulas no sistema, enquanto que no estado 4, a válvula está aberta e a bomba ligada, tendo pressão e vazão positivas no sistema. Na tabela 3.2, as saídas discretizadas dos sensores para os estados de G são apresentadas. Esta tabela é chamada de mapa de sensores.

Tabela 3.2: Mapa de sensores para o Exemplo 3.4.

Q	Q_v	Q_b	Q_c	sensor de pressão	sensor de vazão
0	VF	BD	C1	sem pressão	sem vazão
1	VT	BD	C1	sem pressão	sem vazão
2	VA	BD	C2	sem pressão	sem vazão
3	VT	BD	C2	sem pressão	sem vazão
4	VA	BL	C3	com pressão	com vazão
5	VT	BL	C3	com pressão	sem vazão
6	VA	BD	C4	sem pressão	sem vazão
7	VT	BD	C4	sem pressão	sem vazão

A construção do autômato utilizando o mapa de sensores para G é ilustrada na figura 3.13, e é denominada G_{ms} . Note na figura 3.13, que os valores de saída de $(VA, BL, C3)$ e de $(VT, BL, C3)$ são diferentes. Para representar essa mudança de valores de saída, um novo estado foi criado e duas transições de estado foram definidas. O estado criado é o q_{new} , e as transições são $\phi((VA, BL, C3), \sigma_f) = q_{new}$

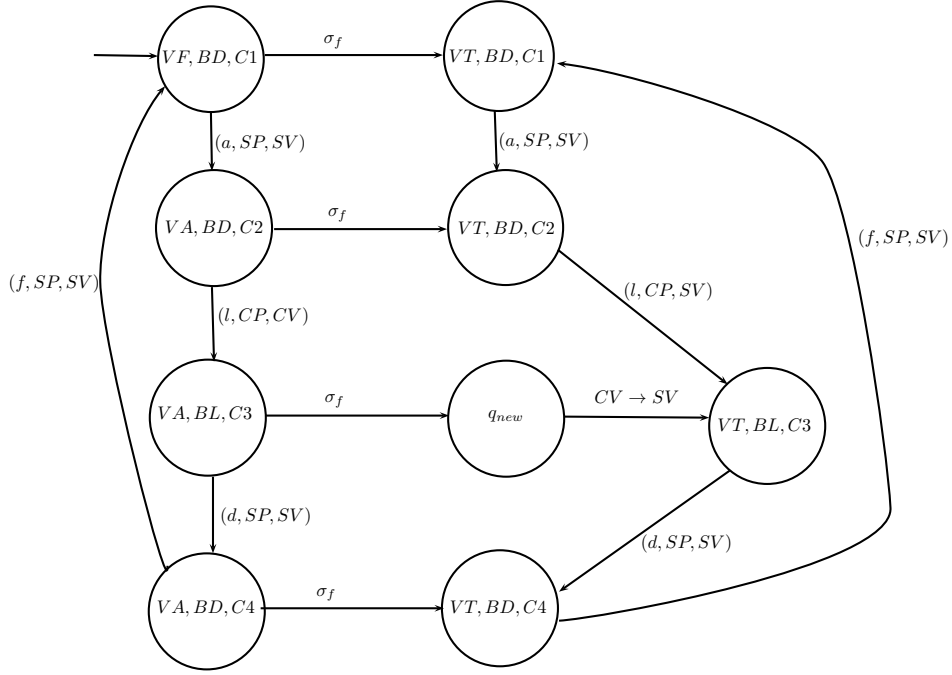


Figura 3.13: Autômato G_{ms} do Exemplo 3.4.

e $\phi(q_{new}, CV \rightarrow SV) = (VT, BL, C3)$. Note ainda que o evento $CV \rightarrow SV$ é um evento observável e representa que o sistema captou a mudança no valor de saída dos estados, isto é, a mudança de “com vazão” para “sem vazão”.

Para a construção do diagnosticador de G_{ms} , é preciso obter o autômato $G_{sm_i} = G_{sm} \parallel A_i$, ilustrado na figura 3.14, e depois calcular $G_{sm_d} = Obs(G_{sm_i})$, ilustrado na figura 3.15.

Note que existe um caminho cíclico $\mathcal{P}_{ms_d} = (\{0N, 1Y\}, (a, SP, SV), \{2N, 3Y\}, (l, CP, CV), \{4N, q_{4N, \sigma_f, 5Y}\}, (d, SP, SV), \{6N, 7Y\}, (f, SP, SV), \{0N, 1Y\})$ em G_{ms_d} , cuja sequência associada a \mathcal{P}_{ms_d} é $s_{ms_d} = (a, SP, SV)(l, CP, CV)(d, SP, SV)(f, SP, SV)$. O caminho normal em G_{ms_i} é $\mathcal{P}_{ms_i} = (\{0N\}, (a, SP, SV), \{2N\}, (l, CP, CV), \{4N\}, (d, SP, SV), \{6N\}, (f, SP, SV), \{0N\})$, entretanto não é possível associar um caminho de falha à \mathcal{P}_{ms_d} . Como o diagnosticador G_{ms_d} não tem ciclos indeterminados, a linguagem gerada por G_{ms_d} é diagnosticável com relação a P_o e σ_f . \square

A mudança de um estado para o outro no autômato construído utilizando mapa de sensores depende da ocorrência de um evento e dos valores das variáveis contínuas

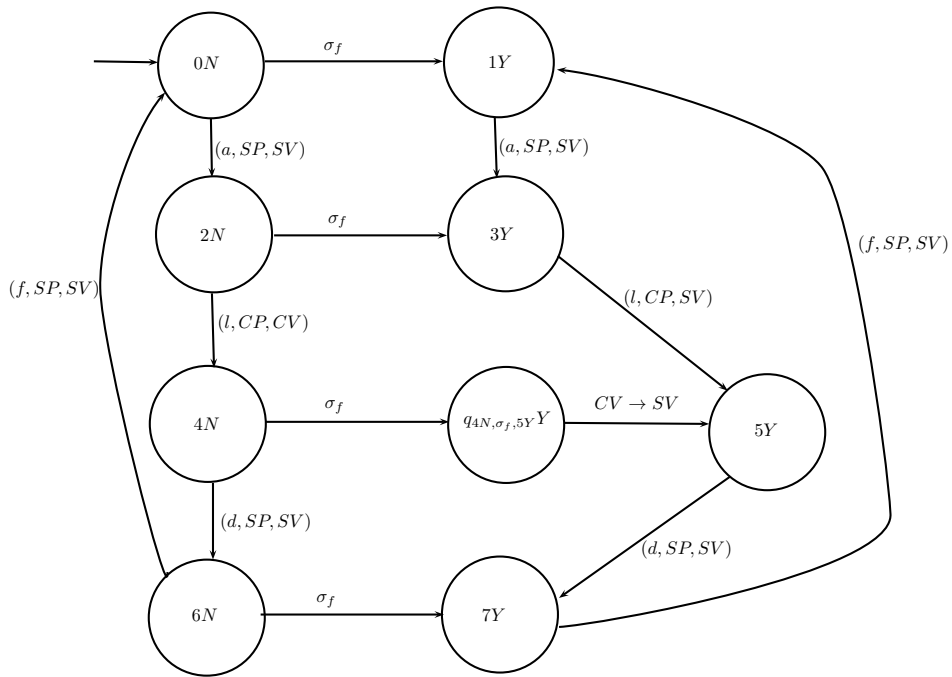


Figura 3.14: Autômato G_{ms_l} do Exemplo 3.4.

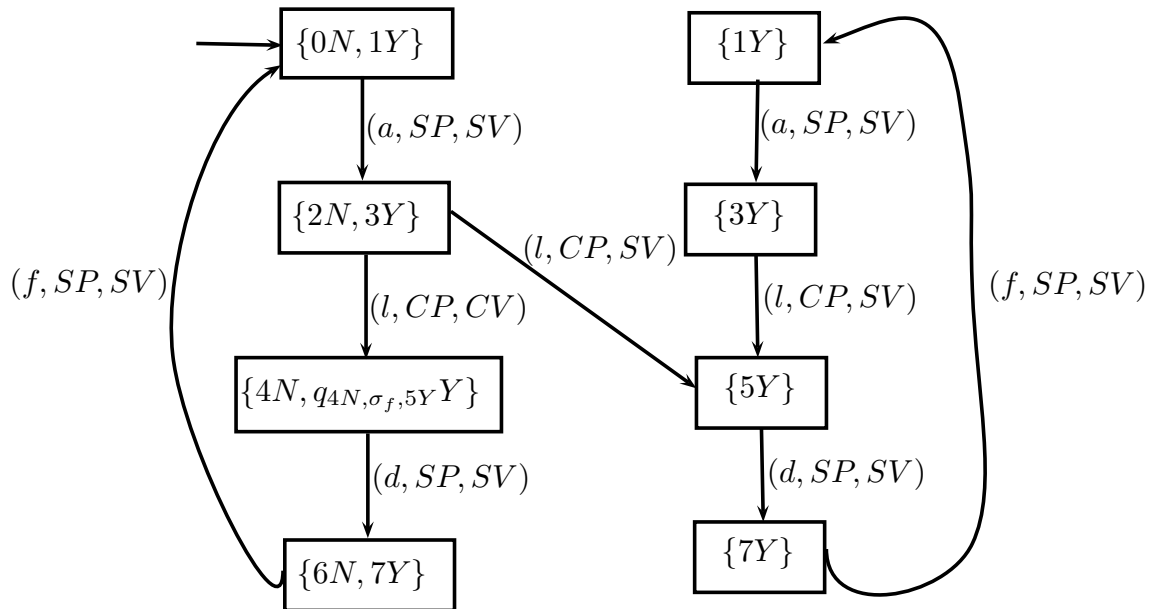


Figura 3.15: Autômato diagnosticador G_{ms_d} do Exemplo 3.4.

medidas pelos sensores. Caso o evento ocorra e os valores medidos não sejam iguais aos valores de saída do estado a ser alcançado, a transição entre os estados não ocorre. A transição só ocorrerá quando o valor medido for igual ao valor de saída. Entretanto, apesar de todas essas condições para a transição entre os estados, é possível que G_{ms_d} faça uma estimativa de estados equivocada, conforme ilustrado no exemplo 3.5.

Exemplo 3.5 *Seja G representado pelo diagrama de transição de estados mostrado na figura 3.16, em que $\Sigma_o = \{a, b\}$ e $\Sigma_{uo} = \{\sigma_f\}$. As saídas discretizadas dos sensores para os estados de G são apresentadas na tabela 3.3. A construção do autômato utilizando mapa de sensores para G com rotulação em relação ao evento de falha σ_f é ilustrada na figura 3.17, e é denominada G_{ms_l} . Após a construção de G_{ms_l} , o diagnosticador G_{ms_d} é construído, conforme mostrado na figura 3.18.*

Tabela 3.3: Mapa de sensores para o Exemplo 3.5.

Q	sensor de temperatura
1	baixa temperatura (BT)
2	baixa temperatura (BT)
3	muito alta temperatura (MAT)
4	alta temperatura (AT)
5	baixa temperatura (BT)

Note que o estado inicial de G_{ms_d} é $\{1N, 2Y\}$. Caso o evento a ocorra, o diagnosticador G_{ms_d} aguardará a leitura do sensor, isto é, o valor de “alta temperatura” ou de “muito alta temperatura”. Como o valor de “alta temperatura” é alcançado antes do valor de “muito alta temperatura”, a estimativa de estado em G_{ms_d} será $\{4N\}$. Suponha agora que ocorra o evento σ_f antes do evento a , o sistema estará no estado $\{2Y\}$. Quando ocorrer o evento a , a transição de $\{2Y\}$ para $\{3Y\}$ somente ocorrerá quando a leitura do sensor alcançar o valor de “muito alta temperatura”. Entretanto, em G_{ms_d} , o sensor atingirá o valor de “alta temperatura” antes do valor de “muito alta temperatura”, acarretando na transição de $\{1N\}$ para $\{4N\}$ pelo evento (a, AT) , em vez da transição de $\{2Y\}$ para $\{3Y\}$ pelo evento (a, MAT) , e

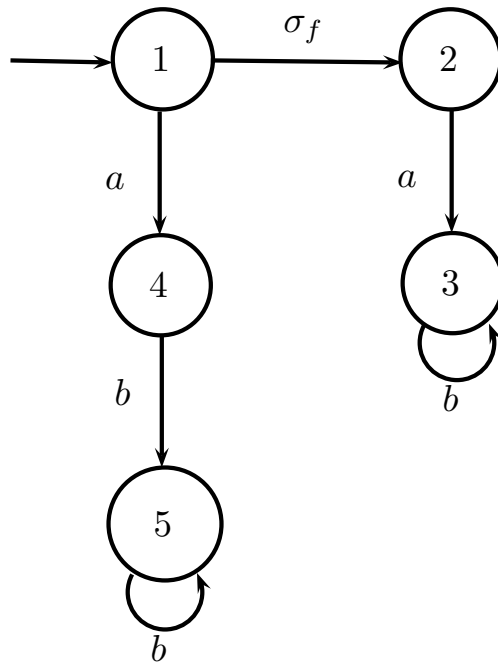


Figura 3.16: Autômato G do Exemplo 3.5.

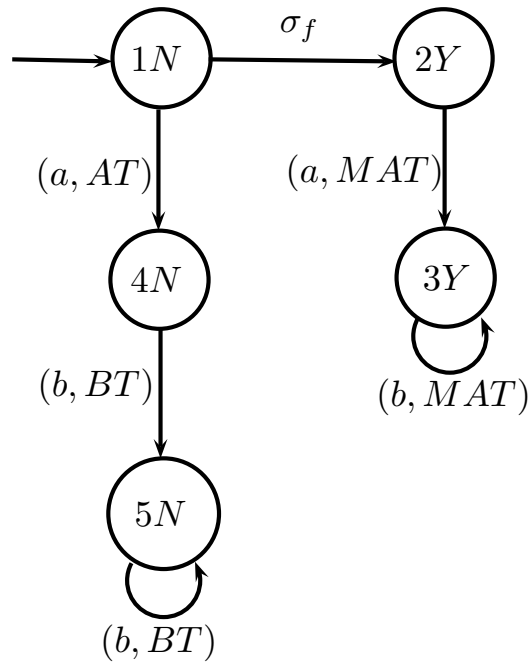


Figura 3.17: Autômato G_{ms_l} do Exemplo 3.5.

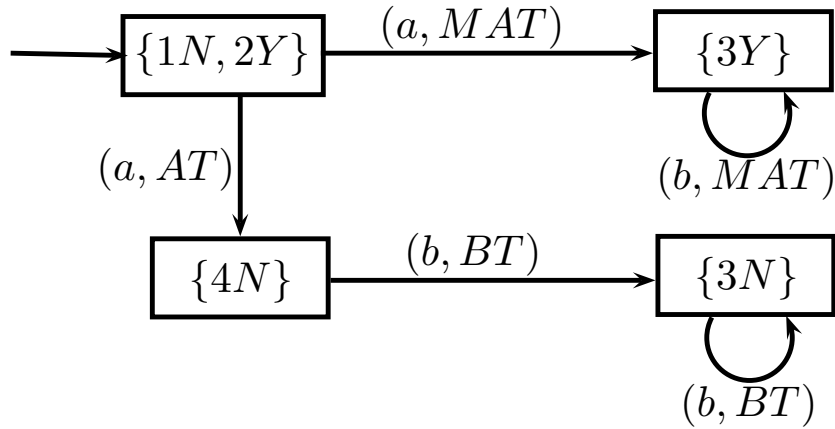


Figura 3.18: Autômato diagnosticador G_{ms_d} do Exemplo 3.5.

portanto, acarretando em uma estimativa de estados errada.

Note ainda que caso ocorra o evento b , cuja sequência observada seja $s = (a, AT)(b, BT)$, a transição de $\{4N\}$ para $\{5N\}$ somente ocorrerá quando a leitura do sensor atingir o valor de “baixa temperatura”. Entretanto, como o sistema encontra-se em $\{3Y\}$ e não em $\{4N\}$, a leitura do sensor não atingirá esse valor, pois sua leitura será de “muito alta temperatura”. Assim, a transição de $\{4N\}$ para $\{5N\}$ pelo evento (b, BT) não irá ocorrer, impedindo a evolução da estimativa de estados em G_{ms_d} .

□

Capítulo 4

Sistemas Híbridos

Sistemas híbridos (SH) são sistemas que combinam comportamentos discretos e contínuos [1–3, 13, 24–27]. A análise e a configuração de um sistema híbrido são mais difíceis do que de um sistema a eventos discretos ou de um contínuo, porque a componente discreta afeta a evolução da componente contínua, bem como, a componente contínua afeta a componente discreta.

A modelagem de sistemas com dinâmica híbrida tem aplicações em diversos ramos da engenharia como: nos circuitos elétricos (por exemplo, quando o carregamento de capacitores são interrompidos por abertura ou fechamento de uma microchave) e nos circuitos hidráulicos (por exemplo, quando a vazão do sistema é controlada por válvulas e bombas).

Em todos esses sistemas, é conveniente modelar a componente discreta e acrescentar a esse modelo a componente contínua. O modelo adotado neste trabalho para o sistema híbrido é o autômato híbrido.

4.1 Autômato Híbrido

Um autômato híbrido é um sistema dinâmico que descreve, simultaneamente, a evolução dos estados e eventos de um sistema a eventos discretos e os valores das variáveis de estado de um sistema contínuo. Como em [1, 13, 26, 27], a definição formal de um autômato híbrido determinístico é dada por:

Definição 4.1 (*Autômato Híbrido*): Um autômato híbrido determinístico, denotado por H , é definido como

$$H = (Q, X, Y, \Sigma, U, f, g, \phi, \Gamma, Inv, guard, \rho, q_0, x_0)$$

sendo

- Q é o conjunto finito de estados discretos;
- $X = \mathbb{R}^n$ é o espaço de estados contínuos;
- $Y \subseteq \mathbb{R}^p$ é o conjunto das possíveis saídas do sistema contínuo;
- Σ é o conjunto de eventos discretos;
- $U \subseteq \mathbb{R}^m$ é o conjunto dos sinais de controle contínuo;
- $f : Q \times X \times U \rightarrow X$ é a função de evolução do estado contínuo;
- $g : Q \times X \times U \rightarrow X$ é a função de saída do sistema contínuo;
- $\phi : Q \times \Sigma \rightarrow Q$ é a função de transição de estados discretos;
- Γ é a função de eventos discretos ativos;
- $Inv : Q \times X$ é o conjunto das condições de invariância;
- $guard \subseteq Q \times Q \times X$ é o conjunto das condições de guarda;
- $\rho : Q \times Q \times X \times \Sigma \rightarrow X$ é a função reset;
- $q_0 \in Q$ é o estado discreto inicial;
- $x_0 \in X$ é o estado contínuo inicial.

□

O estado de um sistema híbrido modelado por H , como em [1–3], é composto de uma parte discreta q_i e uma parte contínua \mathbf{x}_i chamados, respectivamente, modo e medida [1–3], e é representado por (q_i, \mathbf{x}_i) . Do modelo híbrido, é possível extrair

a parte discreta, modelada por um autômato G . Os parâmetros de H que estão contidas em G já foram definidos no capítulo 2, e os outros parâmetros de H serão definidos a seguir.

- Sinais de controle contínuo: A entrada controlável $u \in U$ em que $U \subseteq \mathbb{R}^m$, tal que $\dot{x} = f(q, \mathbf{x}, \mathbf{u})$.
- f é a função de evolução do estado contínuo, cuja representação é dada por uma equação de estado.
- g é a função de saída da dinâmica contínua, cuja representação é dada por uma equação de saída.
- A condição de guarda especifica um subconjunto de X que, quando ocorre, ativa ou permite que a transição do modo q_i para o modo q_j possa acontecer. Esta função de transição é definida através de ϕ .
- A condição invariante (também chamado de domínio) é um subconjunto de X associado a um modo q , tal que a saída do sistema deve pertencer a este conjunto com o objetivo de permanecer neste modo. Caso a saída não pertença ao domínio definido na condição invariante, o sistema é forçado a mudar de modo e uma transição precisa ser definida através ϕ .
- A condição *reset* define o valor inicial de x após a transição de um modo q_i para um modo q_j .

4.1.1 Conceito de distinguibilidade dos modos

Como o estado de um SH é composto por um modo e uma medida, cada estado tem um comportamento contínuo próprio. Portanto, os estados de um SH podem possuir comportamentos contínuos distintos ou não.

Os sistemas contínuos analisados podem ser os sistemas lineares invariantes no tempo ou sistemas não-lineares. Neste trabalho, como em [11], serão analisados os sistemas lineares invariantes no tempo, com representação no espaço de estados.

Com esta condição, o campo vetorial $f(q_i, \mathbf{x}_i, \mathbf{u})$ e a função de saída $g(q_i, \mathbf{x}_i, \mathbf{u})$ são dados pelos modelos contínuos associados a cada modo q_i , da seguinte forma:

$$\begin{cases} \mathbf{x}(n+1) = \mathbf{A}_i \mathbf{x}(n) + \mathbf{B}_i \mathbf{u}(n) \\ \mathbf{y}(n) = \mathbf{C}_i \mathbf{x}(n) + \mathbf{D}_i \mathbf{u}(n) \end{cases}, \quad (4.1)$$

em que

- $\mathbf{x}(n)$ é o vetor de estado no tempo nT_s ;
- $\mathbf{u}(n)$ é o vetor de entrada no tempo nT_s ;
- $\mathbf{y}(n)$ é o vetor de saída no tempo nT_s ;
- T_s é o período de amostragem;
- $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ e \mathbf{D}_i são as matrizes constantes.

O modelo contínuo descrito pela equação (4.1) é denotado como \mathcal{M}_i .

Um modo q_i com equações modeladas \mathcal{M}_i , tem as saídas contínuas modeladas que variam com a tolerância μ_i , isto é, a saída modelada mais as incertezas. Se a saída medida do sistema pertencer ao conjunto de saídas de \mathcal{M}_i , então esta saída é consistente com o modo q_i , caso contrário não.

Para o sistema híbrido modelado por H , o conceito de distinguibilidade é definido a seguir.

Definição 4.2 *Seja q_i e q_j dois modos diferentes de um sistema híbrido modelado por H , associados, respectivamente, aos modelos contínuos \mathcal{M}_i e \mathcal{M}_j . Então, os modos q_i e q_j são distinguíveis em relação a uma medida ψ e a uma tolerância μ , se $\psi(\mathcal{M}_i, \mathcal{M}_j, \mathbf{u}, \mathbf{x}'_0) > \mu$, no qual \mathbf{u} é a entrada definida sobre um intervalo de tempo I e \mathbf{x}'_0 denota a condição inicial de ambos os modelos. Caso contrário, os modos q_i e q_j são não distinguíveis. \square*

Se o modo q_i é distinguível do modo q_j , isto é, possuem medidas diferentes, então sua representação é dada por $q_i \approx q_j$. Caso contrário, é dada por $q_i \sim q_j$.

Note que o modo q_i é distinguível do modo q_j , se todas as possíveis saídas medidas do sistema no modo q_i são consistentes somente com q_i , bem como, todas as possíveis saídas medidas do sistema no modo q_j são consistentes somente com q_j . Caso contrário, o modo q_i é não distinguível do modo q_j .

Os métodos que comparam a saída dos modelos matemáticos com as medições de saída do sistema, como em [4], podem ser utilizados para verificação da distinguibilidade dos modos de um SH . Esses métodos se baseiam na análise de resíduos, em que os resíduos são obtidos através do conhecimento do modelo matemático e das medições de entrada e de saída do sistema em um intervalo de tempo. Alguns dos métodos que podem ser utilizados para verificação da distinguibilidade dos modos de um SH serão apresentados no capítulo 5 e no apêndice A.

A próxima seção tem o objetivo de apresentar um sistema híbrido, bem como a sua modelagem. O modelo adotado é o do autômato híbrido, no qual é modelado primeiramente a componente discreta, e é acrescentada a esse modelo a componente contínua.

4.1.2 Exemplo de um Sistema Híbrido

O exemplo a seguir tem como objetivo ilustrar a estrutura e a dinâmica de um sistema híbrido. São apresentadas separadamente as componentes discretas e contínuas do exemplo, bem como a relação entre elas. O sistema de controle da máquina do leme de um navio, composto de um único leme, é o exemplo escolhido neste trabalho.

Exemplo 4.1 *Seja $H = (Q, X, Y, \Sigma, U, f, g, \phi, \Gamma, \rho, q_0, x_0)$, o autômato híbrido que modela o comportamento do sistema de controle da máquina do leme, cujo sistema óleo-hidráulico do leme é controlado por dispositivos elétricos. As condições de guarda e invariante foram representadas por eventos em H , e portanto foram suprimidas.*

Antes de apresentar o sistema híbrido propriamente dito, alguns conceitos náuticos serão apresentados: bombordo (BB) é o lado esquerdo da embarcação, quando se olha para onde se navega, enquanto boreste (BE) é o lado direito da

embarcação, quando se olha para onde se navega.

O sistema adotado neste trabalho é o comando manual de atuação direta. Esse comando é exercido por uma chave de cinco posições, ou manete, com retorno ao centro. A manete permite que as válvulas solenóides do sistema óleo-hidráulico sejam energizadas diretamente. Nesse modo de comando, enquanto a manete estiver sendo atuada para bombordo (BB) ou para boreste (BE), o leme irá se deslocar no mesmo sentido e irá parar no ângulo desejado quando a manete for liberada. O operador tem 5 opções de operação do leme, que são:

- Posição neutra: o leme fica parado;
- Posição bombordo normal: leme se movimenta para BB com velocidade normal;
- Posição boreste normal: leme se movimenta para BE com velocidade normal;
- Posição bombordo rápida: leme se movimenta para BB com velocidade mais rápida;
- Posição boreste rápida: leme se movimenta para BE com velocidade mais rápida;

Com o objetivo de operar o sistema apropriadamente, algumas ações são necessárias, como, por exemplo, partir o motor elétrico e energizar as solenóides das válvulas direcionais principal e auxiliar. A componente discreta que caracteriza o funcionamento desse sistema, é modelada por $G = (Q, \Sigma, \phi, \Gamma, q_0)$, conforme mostrado na figura 4.1.

Inicialmente, o conjunto de eventos Σ é dado por:

$$\Sigma = \{ON, BB_{ON}, BB_{OFF}, BE_{ON}, BE_{OFF}, RAP_{ON}, RAP_{OFF}, BB35^\circ, BE35^\circ, \sigma_f\},$$

em que

$$\Sigma_o = \{ON, BB_{ON}, BB_{OFF}, BE_{ON}, BE_{OFF}, RAP_{ON}, RAP_{OFF}, BB35^\circ, BE35^\circ\}$$

e $\Sigma_{uo} = \{\sigma_f\}$. A descrição dos eventos é apresentada na tabela 4.1.

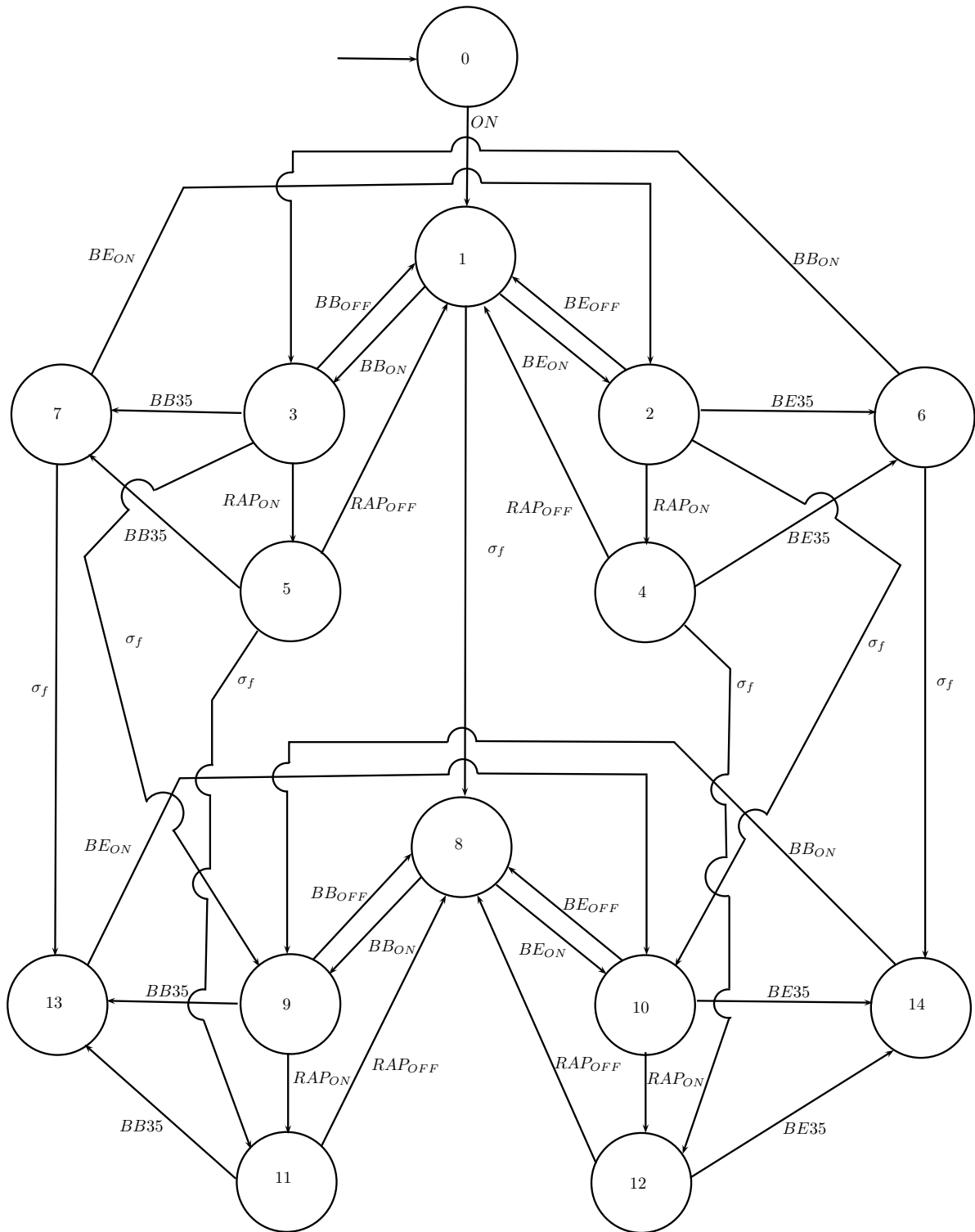


Figura 4.1: Autômato G do Exemplo 4.1.

Tabela 4.1: Tabela com a descrição dos eventos de G do Exemplo 4.1.

Evento	Descrição
ON	partir o motor elétrico e pressurizar o sistema hidráulico
BB_{ON}	energizar a solenóide de BB da válvula direcional principal, com o objetivo de movimentar o leme para bombordo
BB_{OFF}	desenergizar a solenóide de BB da válvula direcional principal
BE_{ON}	energizar a solenóide de BE da válvula direcional principal, com o objetivo de movimentar o leme para boreste
BE_{OFF}	desenergizar a solenóide de BE da válvula direcional principal
RAP_{ON}	energizar a solenóide da válvula direcional auxiliar, com o objetivo de aumentar a velocidade do leme
RAP_{OFF}	desenergizar as solenóides das válvulas direcionais principal e auxiliar, com o objetivo de parar o leme
$BB35^\circ$	desenergizar as solenóides das válvulas direcionais principal e auxiliar, quando a posição do leme atinge 35° a bombordo
$BE35^\circ$	desenergizar as solenóides das válvulas direcionais principal e auxiliar, quando a posição do leme atinge 35° a boreste
σ_f	falha na válvula direcional auxiliar, em que a mesma trava na posição velocidade rápida

Note que o sistema possui uma proteção no sistema elétrico, nos quais as bobinas da válvula direcional principal e auxiliar são desenergizadas quando o leme atinge 35° a bombordo ou a boreste, protegendo o sistema óleo-hidráulico, principalmente o cilindro de trabalhar fora das suas condições normais de operação.

O evento σ_f representa a falha na válvula direcional auxiliar, em que a válvula trava na posição da bobina energizada, mesmo essa não estando. Essa falha aumenta a vazão do sistema, e quando a válvula direcional principal é energizada, a velocidade de movimentação do leme é aumentada.

Com o objetivo de acrescentar as informações da variável contínua à componente discreta do SH, é necessário entender o comportamento da variável contínua do sistema da máquina do leme em cada modo. Inicialmente, é apresentado o seu diagrama óleo-hidráulico, conforme mostrado na figura 4.2, cuja representação dos componentes é dada pela tabela 4.2. O sistema óleo-hidráulico é composto por: uma cana do leme, um cilindro, duas válvulas de esfera, duas mangueiras, duas válvulas de frenagem, uma válvula direcional principal, um motor elétrico, uma bomba principal, uma bomba auxiliar, duas válvulas de retenção, um tanque, uma válvula direcional auxiliar e uma válvula de ventagem. Para não aumentar desnecessariamente

Tabela 4.2: Tabela com a nomenclatura do diagrama óleo-hidráulico, conforme figura 4.2, do Exemplo 4.1.

Número do elemento	Descrição
1	Cana do leme
2	Cilindro
3	Válvula de esfera
4	Mangueira
5	Válvula de frenagem
6	Válvula direcional principal
7	Motor elétrico
8	Bomba
9	Válvula de retenção
10	Tanque
11	Válvula direcional auxiliar
12	Válvula de ventagem

a complexidade do sistema, os demais componentes não serão detalhados.

Note no diagrama óleo-hidráulico, apresentado na figura 4.2, que a válvula direcional principal, ou VDP, é a responsável pelo deslocamento do leme. A VDP é uma válvula de três posições acionada por duas solenóides, cujas posições são descritas a seguir:

- *posição 1: o cilindro se expande devido à diferença de pressão nas câmaras, movimentando o leme para bombordo. O acionamento elétrico é feito através da solenóide de BB, que está energizada;*
- *posição 2: o cilindro fica estático devido a pressão nas duas câmaras serem iguais e pela atuação das válvulas de frenagem, não havendo nenhuma movimentação no leme. As solenóides de bombordo e boreste não estão energizadas;*
- *posição 3: o cilindro se contrai devido à diferença de pressão nas câmaras, movimentando o leme para boreste. O acionamento elétrico é feito através da solenóide de BE, que está energizada.*

A válvula direcional auxiliar, ou VDA, é a responsável pela velocidade de deslocamento do leme. A VDA é uma válvula de duas posições com acionamento elétrico. Caso a solenóide não esteja energizada, o sistema trabalha somente com a vazão

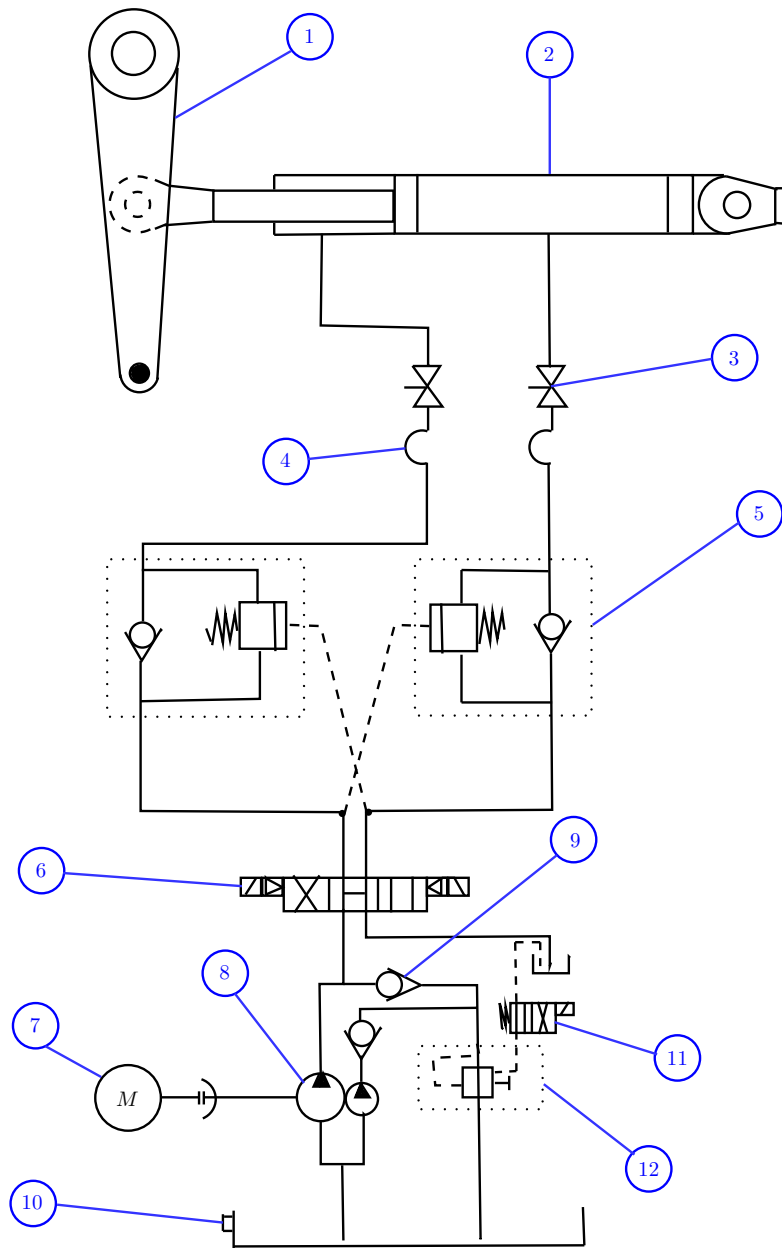


Figura 4.2: Diagrama óleo-hidráulico do sistema da máquina do leme do Exemplo 4.1.

da bomba principal. Caso contrário, o sistema trabalha simultaneamente com as bombas principal e auxiliar, acarretando um aumento de 25% na vazão do sistema. O aumento da vazão provoca um aumento na velocidade de deslocamento do leme, quando a VDP encontra-se na posição 1 ou 3.

Assim, pode-se afirmar que o deslocamento angular do leme, que é a variável contínua escolhida desse sistema, depende da posição escolhida na manete pelo operador. Por exemplo, se o operador escolher a posição neutra, o leme ficará parado.

Em [28], as variáveis contínuas do sistema de controle da máquina do leme com feedback foram modeladas, considerando a vazão fixa e um controlador ON/OFF, e, posteriormente, um controlador PI. O sistema, em [28], é composto por uma válvula proporcional, que é uma válvula acionada eletronicamente, na qual a grandeza de saída (pressão ou vazão) é alterada na mesma proporção como a grandeza de entrada.

Uma das diferenças desse trabalho para o de [28] é que a vazão do sistema pode ser aumentada, se a solenóide da válvula direcional auxiliar for energizada. Outra diferença é que o sistema em [28] é composto por uma válvula proporcional, enquanto neste trabalho essa válvula é substituída por uma válvula direcional de três posições. Com essa alteração, os valores de saída da válvula proporcional que variavam em uma faixa de valores, passam a ser discretizados, podendo assumir somente três valores de saída.

As seguintes hipóteses foram feitas no sistema contínuo da máquina do leme:

1. O comportamento contínuo do sistema óleo-hidráulico mostrado na figura 4.2 será o mesmo de [28], nas condições de solenóide energizada em BB e BE, isto é, para os valores de entrada no limite inferior e superior do intervalo permitido em [28];
2. A velocidade angular do leme será nula quando as solenóides de BB e BE da VDP não estiverem energizadas;
3. A velocidade angular será uniforme para toda a região limitada de 35° BB a

35°BE, isto é, a velocidade angular do leme será igual para qualquer condição inicial que pertença a esse intervalo;

4. O distúrbio externo, referente ao torque de flutuação aleatório [28], é desprezível.

Considerando as hipóteses citadas e a função de transferência dada por [28], as equações de estado do sistema são apresentadas abaixo:

$$\begin{cases} \mathbf{x}(n+1) = \mathbf{A}_i\mathbf{x}(n) + \mathbf{B}_i\mathbf{u}(n) \\ \mathbf{y}(n) = \mathbf{C}_i\mathbf{x}(n) + \mathbf{D}_i\mathbf{u}(n) \end{cases}, \quad (4.2)$$

As variáveis de estado $x_1(n)$, $x_2(n)$ e $x_3(n)$ são, respectivamente, posição, velocidade e aceleração angulares do leme. A entrada $\mathbf{u}(n)$ é a posição nominal da VDP e tem valor unitário. A alteração dos valores de $\mathbf{B}_i\mathbf{u}(n)$ é realizada através da matriz \mathbf{B}_i . A saída $\mathbf{y}(n)$ é a variação da posição angular do leme. O período de amostragem T_s é de 0.001s.

A saída \mathbf{y} representa o deslocamento angular do leme, isto é, quanto o leme se deslocou, e não a posição angular final. Esta é calculada através da posição angular inicial adicionada ao deslocamento angular, ou $\mathbf{y}_f = y_o + \mathbf{y}$, em que y_o é obtido da condição reset.

Note que a posição da manete altera a configuração das válvulas direcionais principal e auxiliar, e portanto alterando as variáveis contínuas do sistema. As matrizes constantes das equações de estado para as diversas posições da manete foram obtidas da função de transferência dada por [28] e das hipóteses citadas anteriormente, e são apresentadas a seguir:

- Posição neutra (\mathcal{M}_1)

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$B_1 = D_1 = 0 \text{ e } C_1 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix};$$

- *Posição Boreste normal* (\mathcal{M}_2)

$$A_2 = \begin{pmatrix} 1 & 0.001 & 0 \\ 0 & 0.9997 & 0.0009 \\ -0.0462 & -0.596 & 0.7665 \end{pmatrix},$$

$$B_2 = \begin{pmatrix} 0 \\ 0.0017 \\ 3.2725 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \text{ e } D_2 = 0;$$

- *Posição Bombordo normal* (\mathcal{M}_3)

$$A_3 = A_2, B_3 = -B_2, C_3 = C_2 \text{ e } D_3 = 0;$$

- *Posição Boreste rápido* (\mathcal{M}_4)

$$A_4 = \begin{pmatrix} 1 & 0.001 & 0 \\ 0 & 0.9996 & 0.0009 \\ -0.0447 & -0.7176 & 0.7176 \end{pmatrix}, B_4 = \begin{pmatrix} 0 \\ 0.0026 \\ 4.9558 \end{pmatrix},$$

$$C_4 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \text{ e } D_4 = 0;$$

- *Posição Bombordo rápido* (\mathcal{M}_5)

$$A_5 = A_4, B_5 = -B_4, C_5 = C_4 \text{ e } D_5 = 0.$$

Os modelos \mathcal{M}_2 e \mathcal{M}_3 são obtidos de [28], desprezando o distúrbio externo, enquanto \mathcal{M}_4 e \mathcal{M}_5 tiverem os seus valores alterados em função do aumento da vazão do sistema.

Caso o leme alcance o ângulo de 35° a bombordo ou a boreste, independentemente da posição que a manete esteja, as equações de estado serão iguais a da posição neutra, isto é, o do modelo (\mathcal{M}_1).

A tabela 4.3 descreve os modos de H , de acordo com o funcionamento do motor elétrico, da VDP e da VDA, bem como pela vazão do sistema, pelo ângulo do leme e pela ocorrência da falha.

Tabela 4.3: Tabela com as características dos modos de H do Exemplo 4.1.

Q	Motor elétrico	tensão na VDP	tensão na VDA	Vazão do sistema	Ângulo do leme	Falha
0	desligado	não	não	sem vazão	$(35^\circ BB, 35^\circ BE)$	não
1	ligado	não	não	normal	$(35^\circ BB, 35^\circ BE)$	não
2	ligado	solenóide de BB	não	normal	$(35^\circ BB, 35^\circ BE)$	não
3	ligado	solenóide de BE	não	normal	$(35^\circ BB, 35^\circ BE)$	não
4	ligado	solenóide de BB	sim	aumentada	$(35^\circ BB, 35^\circ BE)$	não
5	ligado	solenóide de BE	sim	aumentada	$(35^\circ BB, 35^\circ BE)$	não
6	ligado	não	não	normal	$[35^\circ BE]$	não
7	ligado	não	não	normal	$[35^\circ BB]$	não
8	ligado	não	não	aumentada	$(35^\circ BB, 35^\circ BE)$	sim
9	ligado	solenóide de BB	não	aumentada	$(35^\circ BB, 35^\circ BE)$	sim
10	ligado	solenóide de BE	não	aumentada	$(35^\circ BB, 35^\circ BE)$	sim
11	ligado	solenóide de BB	sim	aumentada	$(35^\circ BB, 35^\circ BE)$	sim
12	ligado	solenóide de BE	sim	aumentada	$(35^\circ BB, 35^\circ BE)$	sim
13	ligado	não	não	aumentada	$[35^\circ BB]$	sim
14	ligado	não	não	aumentada	$[35^\circ BE]$	sim

Conforme a tabela 4.3, é possível notar que a dinâmica contínua do SH muda de acordo com o modo que o sistema se encontra, bem como a dinâmica discreta é afetada pela dinâmica contínua. Um exemplo da dinâmica discreta alterando a contínua, é quando o sistema se encontra no modo 1, cujo sistema contínuo está estático e o leme não se movimenta, e ocorre uma transição para o modo 2, cujo sistema contínuo está em evolução e o leme se movimenta. Em relação à dinâmica contínua afetando a discreta, é possível notar esse efeito no modo 3, cujo leme está em movimento. Quando a posição angular do leme atingir 35° a BE, o evento

BE35° ocorre, ocasionando a transição do modo 3 para o 7. Portanto, é possível relacionar os modos com a sua dinâmica contínua, e apresentar o diagrama de transição de estados do autômato híbrido H , conforme a figura 4.3.

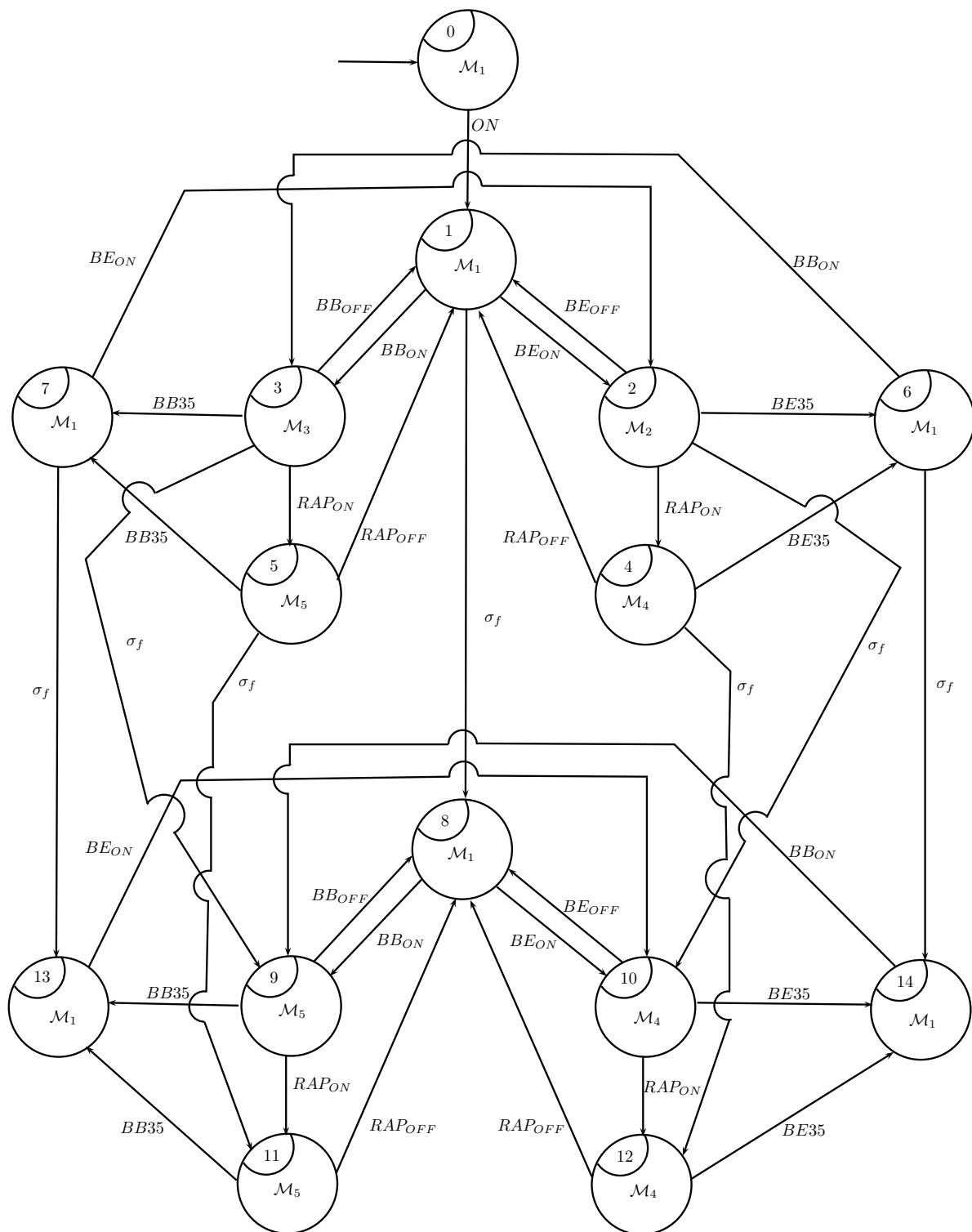


Figura 4.3: Autômato H do Exemplo 4.1.

Capítulo 5

Diagnóstico de falhas em Sistemas Híbridos

Este trabalho apresenta uma nova definição para a diagnosticabilidade de falhas em sistemas híbridos, chamada de h -diagnosticabilidade. Além disso, é apresentado um novo método para a verificação da h -diagnosticabilidade baseado na construção de um diagnosticador, denominado de diagnosticador híbrido, que combina um diagnosticador de sistema a eventos discretos com a distinguibilidade dos modos de um sistema híbrido.

5.1 Diagnosticabilidade em Sistemas Híbridos

Para os casos em que a linguagem gerada pela componente discreta é não diagnosticável de acordo com a definição 3.3, é possível utilizar as informações das variáveis contínuas de um sistema híbrido para verificar a distinguibilidade dos modos, e assim detectar a ocorrência do evento de falha.

Exemplo 5.1 *Dado o autômato híbrido H do exemplo 4.1, em que G é a componente discreta desse sistema. Calcule o diagnosticador de G . O diagnosticador G_d é dado conforme figura 5.1.*

Note que a linguagem de G é não diagnosticável com relação a P_o e σ_f , pois existem ciclos indeterminados em G_d como, por exemplo, os ciclos $((1N, 8Y), BB_{ON},$

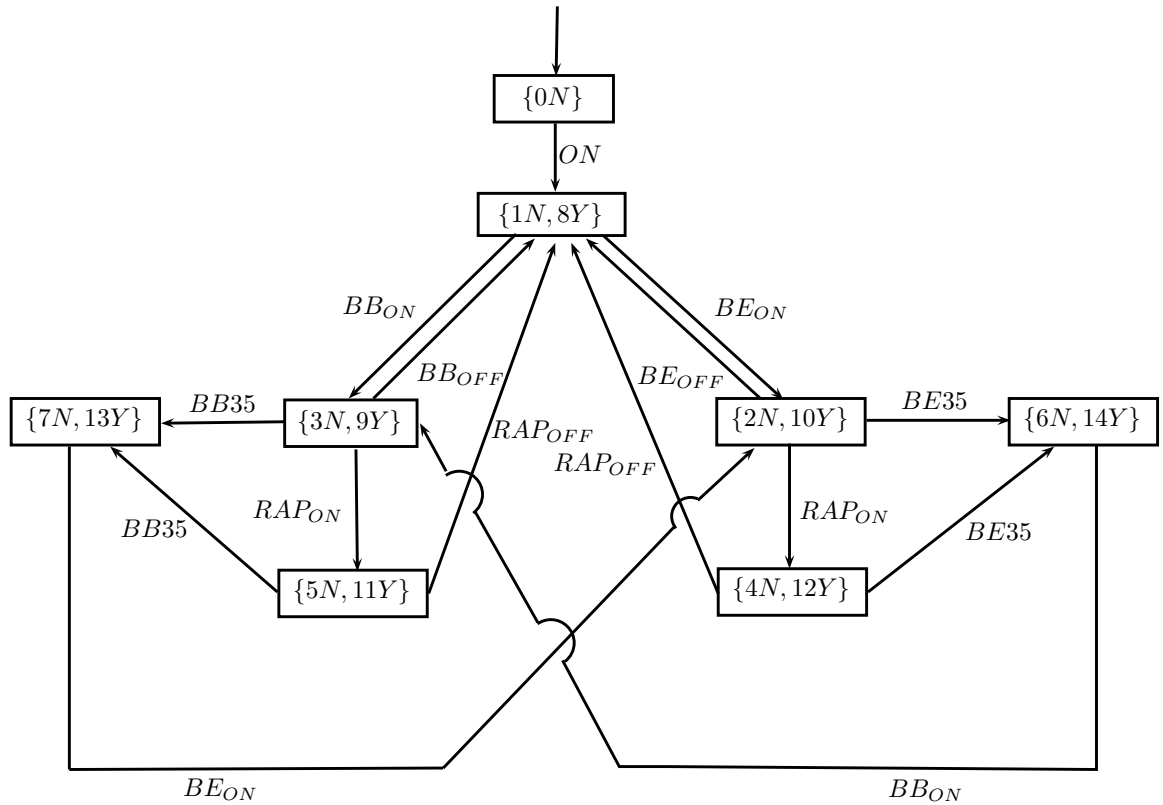


Figura 5.1: Autômato diagnosticador G_d do Exemplo 5.1

$(3N, 9Y), BB_{OFF}, (1N, 8Y)$). Note ainda que as seqüências $s_N = ON \cdot BB_{ON} \in L_N$ e $s_f = ON \cdot \sigma_f \cdot BB_{ON} \in L \setminus L_N$ possuem a mesma projeção, ou seja, $P_o(ON \cdot BB_{ON}) = P_o(ON \cdot \sigma_f \cdot BB_{ON}) = ON \cdot BB_{ON}$. Após a ocorrência da seqüência $s = ON \cdot BB_{ON}$, o diagnosticador G_d , mostrado na figura 5.1, indica dois possíveis estados: $3N$ e $9Y$. Note que não é possível determinar se a falha σ_f ocorreu ou não após a observação da seqüência $ON \cdot BB_{ON}$. Por outro lado, se após a ocorrência do evento BB_{ON} for possível distinguir os estados $3N$ e $9Y$, antes da ocorrência de qualquer outro evento, inclusive o evento não-observável σ_f , então é possível identificar que a falha σ_f ocorreu.

A seguir é apresentado o método de diagnóstico de falhas em sistemas híbridos proposto neste trabalho, baseado na observação dos eventos e na distinguibilidade dos modos de um sistema híbrido. O método consiste nas seguintes etapas:

1. Calcular $G_l^{YN} = G \parallel A_l^{YN}$, isto é, rotular os estados de G para informar se o estado foi alcançado após um evento de falha;

2. Definir o estado inicial do diagnosticador como o alcance não observável do estado inicial de G_l^{YN} , ou seja, $UR(q_{0_l}^{YN})$;
3. Após a observação de um evento σ_o , calcular o alcance não observável dos estados alcançáveis por σ_o para obter a estimativa de estados de G_l^{YN} . Os estados da estimativa de estados do diagnosticador que são alcançados por σ_o são chamados de estados cabeças.
4. Verificar a consistência dos estados cabeças da estimativa de estados utilizando as medições das variáveis de estado contínuo, e eliminar os modos cabeças não consistentes da estimativa de estados e todos os estados que não pertençam ao alcance não observável de um modo cabeça consistente. Se uma estimativa de estados composta somente por estados rotulados com Y é alcançada, então a falha é diagnosticada.

Antes de apresentar o conceito de h -diagnosticabilidade e o método para a verificação da h -diagnosticabilidade em sistemas híbridos, alguns conceitos são necessários e serão apresentados na seção abaixo.

5.1.1 Conceitos preliminares

Um sistema híbrido, modelado por um autômato híbrido H , tem a sua componente discreta modelada por um autômato determinístico G . Para a componente discreta G , é definida a função que determina os possíveis modos de G alcançados após a ocorrência de uma sequência observável, conforme [20]. Essa estimativa é denotada $Reach(G, \nu)$, sendo $\nu = v\sigma_o = P_o(s)$ a sequência observada no autômato G após a ocorrência de uma sequência $s \in L$ em que o último evento observável é σ_o , e pode ser calculado recursivamente como

$$Reach(G, \varepsilon) = UR(q_0), \quad (5.1)$$

$$Reach(G, v\sigma_o) = UR(\Delta(Reach(G, v), \sigma_o)), \quad (5.2)$$

em que $\Delta(\text{Reach}(G, v), \sigma_o) = \bigcup_{i=1}^{\kappa} \delta(q_i, \sigma_o)$, com $q_i \in \text{Reach}(G, v)$, $\kappa = |\text{Reach}(G, v)|$, e $\delta(q_i, \sigma_o) = \{\phi(q_i, \sigma_o)\}$, se $\phi(q_i, \sigma_o)$ é definido, e $\delta(q_i, \sigma_o) = \emptyset$, caso contrário.

Note que $\Delta(\text{Reach}(G, v), \sigma_o)$ são todos os modos alcançados pela sequência observável $v\sigma_o$ em G , cujo último evento é observável.

Um caminho em G é uma sequência $\mathcal{P} = (q_{p_1}, \sigma_{p_1}, q_{p_2}, \dots, \sigma_{p_{n-1}}, q_{p_n})$, em que $\sigma_{p_i} \in \Sigma$, $q_{p_{i+1}} = \phi(q_{p_i}, \sigma_{p_i})$ e $i = 1, 2, \dots, n-1$. Um caminho cíclico é um caminho tal que $q_{p_1} = q_{p_n}$. Seja $G_{\mathcal{P}} = (Q_{\mathcal{P}}, \Sigma, \phi_{\mathcal{P}}, \Gamma_{\mathcal{P}}, q_{\mathcal{P}_0})$ um subautômato de G formado somente com o caminho \mathcal{P} tais que $Q_{\mathcal{P}}$ é o conjunto de estados discretos em \mathcal{P} , $\phi_{\mathcal{P}}$ é a função de transição de estados definida através de \mathcal{P} , $\Gamma_{\mathcal{P}}$ é a função de eventos ativos e $q_{\mathcal{P}_0}$ é o estado inicial de \mathcal{P} .

Para a verificação da h -diagnosticabilidade de um SH , as seguintes hipóteses usuais, **H1**, **H2** [5–8] e **H3** [11], são feitas:

H1 A linguagem L gerada por G é viva, ou seja, existe uma transição de estado definida para cada modo $q \in Q$, isto é, $\Gamma(q_i) \neq \emptyset$ para todo $q_i \in Q$.

H2 Não existe no autômato G nenhum ciclo formado somente por eventos não-observáveis, isto é,

$$\exists n_o \in \mathbb{N} \text{ tal que } \forall ust \in L, s \in \Sigma_{uo}^* \Rightarrow |s| \leq n_o$$

em que $|s|$ denota o comprimento da sequência s .

H3 Nenhum evento pode ocorrer durante a verificação da distinguibilidade dos modos;

5.1.2 Definição de h -diagnosticabilidade

Após a apresentação dos conceitos preliminares, a diagnosticabilidade de sistemas híbridos, chamada de h -diagnosticabilidade, pode ser definida a seguir.

Definição 5.1 *Seja L a linguagem gerada pela componente discreta G de um sistema híbrido modelado por H e $L_N \subset L$ a linguagem normal de G . O sistema híbrido H é dito ser h -diagnosticável se*

$$(\exists n \in \mathbb{N})(\forall s \in L \setminus L_N)(\forall st \in L \setminus L_N, |t| \geq n) \Rightarrow D_1 \vee D_2,$$

no qual D_1 é dado por:

$$(\forall \omega \in P_o^{-1}(P_o(st)) \cap L, \omega \in L \setminus L_N)$$

e D_2 é dado por

$$(\forall s_N \in L_N, P_o(s_N) = P_o(st) = \nu)(\exists \eta \sigma_o \in \bar{\nu}, q_F \approx q_N),$$

em que $q_F \in \Delta(\text{Reach}(G_{P_F}, \eta), \sigma_o)$ e G_{P_F} é o subautômato de G_ℓ associado com caminho \mathcal{P}_F , cuja sequência é st , enquanto $q_N \in \Delta(\text{Reach}(G_{P_N}, \eta), \sigma_o)$ e G_{P_N} é o subautômato de G_ℓ associado com caminho \mathcal{P}_N , cuja sequência é s_N . \square

De acordo com a definição 5.1, H é h -diagnosticável se for possível distinguir todos os caminhos de falha dos caminhos normais após um número limitado de ocorrência de eventos. Existem duas possibilidades para distinguir os caminhos:

- as sequências de eventos têm diferentes projeções, conforme condição D_1 ;
- Existem pelo menos dois estados distinguíveis alcançados após a observação de um evento observável em ambos os caminhos, conforme condição D_2 ;

Note que se a condição D_1 for verdadeira para todo $st \in L \setminus L_N$, então a linguagem gerada pela componente discreta é diagnosticável de acordo com a definição 3.3, e portanto o sistema híbrido é h -diagnosticável.

Em relação à condição D_2 , note que o sistema híbrido modelado por H é h -diagnosticável se a observação das sequências associadas ao caminho normal \mathcal{P}_N e de falha \mathcal{P}_f forem diferentes, após a ocorrência de k eventos. De acordo com a

condição D_2 , esses caminhos podem ser distinguidos antes da ocorrência do k -ésimo evento, se esses caminhos têm modos distinguíveis após um evento observável.

Uma vez que h -diagnosticabilidade de um SH tenha sido definida, o próximo passo é a verificação da h -diagnosticabilidade. Na próxima seção, dois métodos para a verificação da h -diagnosticabilidade são apresentados.

5.2 Verificação da Diagnosticabilidade de SH

Os métodos para a verificação da h -diagnosticabilidade utilizando os diagnosticadores G_{CD} , chamado de diagnosticador clustered, e G_H , chamado de diagnosticador híbrido, são apresentados. Esses diagnosticadores combinam o diagnosticador de sistema a eventos discretos com a distinguibilidade dos modos de um sistema híbrido.

5.2.1 Diagnosticador Clustered

A construção do autômato diagnosticador $G_{CD} = (Q_{CD}, \Sigma_{CD}, \phi_{CD}, \Gamma_{CD}, q_{0_{CD}})$ é apresentado a seguir.

Algoritmo 5.1

- *Passo 1:* Calcule $G_\ell = G \| A_\ell^{YN} = (Q_\ell, \Sigma, \Gamma_\ell, \phi_\ell, q_{\ell_0})$.
- *Passo 2:* Defina $q_{0_{CD}} = UR(q_{\ell_0})$, $\Sigma_{CD} = \Sigma_o$ e $Q_{CD} = \emptyset$.
- *Passo 3:* Defina $\tilde{Q} = \{q_{0_{CD}}\}$.
- *Passo 4:* $Q_{CD} \leftarrow Q_{CD} \cup \tilde{Q}$.
- *Passo 5:* Defina $\hat{Q} = \tilde{Q}$ e $\tilde{Q} = \emptyset$.
- *Passo 6:* Para cada $B \in \hat{Q}$:
 - *Passo 6.1:* $\Gamma_{CD}(B) = (\bigcup_{q_\ell \in B} \Gamma_\ell(q_\ell)) \cap \Sigma_o$
 - *Passo 6.2:* Para cada $e \in \Gamma_{CD}(B)$:
 - *Passo 6.2.1:* $H_{cd} = \{q_\ell \in Q_\ell : (\exists \tilde{q}_\ell \in B)[q_\ell = \phi_\ell(\tilde{q}_\ell, e)]\}$.

- . Passo 6.2.2: $\phi_{CD}(B, e) = UR(H_d)$ e $\bar{Q} = UR(H_d)$.
- . Passo 6.2.3: $Q_{CD} \leftarrow Q_{CD} \cup \{\phi_{CD}(B, e)\}$.
- . Passo 6.2.4: $[h_C, s_C] = Distinguibibilidade(H_d)$.
- . Passo 6.2.5: Se $\|h_C\| = 1$, então
 - $\tilde{Q} \leftarrow \tilde{Q} \cup \{\phi_{CD}(B, e)\}$.
- . Passo 6.2.6: Se $\|h_C\| > 1$, então
 - * Passo 6.2.6.1: $\Sigma_C = \bigcup_{i=1}^{|s_C|} \{cl_i\}$, em que cl_i é o i -ésimo elemento da sequência de eventos s_C .
 - * Passo 6.2.6.2: $\Sigma_{CD} \leftarrow \Sigma_{CD} \cup \Sigma_C$.
 - * Passo 6.2.6.3: Para $i = 1$ a $\|h_C\|$ faça
 - . Passo 6.2.6.3.1: $Q_{CD} \leftarrow Q_{CD} \cup \{UR(H_i)\}$, em que H_i é o i -ésimo elemento da sequência de conjunto de estados h_C .
 - . Passo 6.2.6.3.2: $\phi_{CD}(\bar{Q}, cl_i) = UR(H_i)$.
 - . Passo 6.2.6.3.3: $\tilde{Q} \leftarrow \tilde{Q} \cup \{\phi_{CD}(\bar{Q}, cl_i)\}$.
 - * Passo 6.2.6.4: $\Gamma_{CD}(\bar{Q}) = \Sigma_C$.
- Passo 7: Repita os passos 5 a 6 até a total construção de G_{CD} .

Função 5.1 $[h_C, s_C] = Distinguibibilidade(H_d)$

- Passo 1: Crie uma sequência de estados h , com todos os elementos de H_d , sem repetição. Defina $h_C = \lambda$, e $s_C = \varepsilon$.
- Passo 2: Enquanto $h \neq \lambda$ faça
 - Passo 2.1: $\tilde{H} = \emptyset$.
 - Passo 2.2: Seja h_j o j -ésimo elemento da sequência h . Para $j = 1$ a $\|h\|$ faça
 - * Passo 2.2.1: Se $h_1 \sim h_j$, então $\tilde{H} \leftarrow \tilde{H} \cup \{h_j\}$.
 - Passo 2.3: Se $\tilde{H} \neq H_d$:

- * *Passo 2.3.1: Crie um evento cl_{h_1} .*
- * *Passo 2.3.2: $s_C \leftarrow s_C \cdot cl_{h_1}$, no qual \cdot denota a operação concatenação.*
- * *Passo 2.3.3: $h_C \leftarrow h_C \cdot \tilde{H}$.*

– *Passo 2.4: Crie uma nova sequência h , removendo de h os estados de \tilde{H} .*

Observação 5.1 *Note na função 5.1 que a sequência de conjunto de estados h é formada pela concatenação de estados, ou conjuntos de estados. O tamanho de h é denotado por $\|h\|$. Neste trabalho, a sequência vazia de estados é denotada por λ , e $\|\lambda\| = 0$.*

De acordo com o algoritmo 5.1, é possível observar que a verificação da distinguibilidade dos modos é feita imediatamente após a ocorrência de um evento observável e supõe-se que não ocorra nenhuma mudança de modo durante essa verificação.

O objetivo da função 5.1 é agrupar os conjuntos de estados não distinguíveis de um conjunto de estados discretos de H_{cl} . A saída dessa função é h_c , que é uma sequência de conjuntos de estados não distinguíveis, tal que cada elemento de h_c é um subconjunto de H_{cl} , e s_C que corresponde a uma sequência de eventos criada para representar o resultado da verificação da distinguibilidade dos modos.

A verificação da distinguibilidade dos modos deve ser realizada antes da implementação do diagnosticador online com o objetivo de verificar se os modos do sistema podem ser identificados pelas medições das variáveis de estado contínuo do sistema. A subseção a seguir apresenta as informações necessárias para se verificar a distinguibilidade dos modos.

5.2.2 Verificação da distinguibilidade dos modos em SH

Em um sistema híbrido, o modo está associado a uma dinâmica contínua. A distinguibilidade dos modos q_i e q_j deve ser válida para todo o domínio de funcionamento do sistema, caso não seja possível, os modos são não distinguíveis. Portanto, será necessário determinar as seguintes condições:

- tempo de amostragem;

- modelos matemáticos;
- entrada do sistema;
- o domínio das variáveis de estado;
- tolerância μ

O primeiro item se refere ao tempo necessário para verificar a distinguibilidade dos modos, para todas as condições iniciais possíveis e com a entrada do sistema definida. Os modelos matemáticos representam as variáveis contínuas do sistema. O terceiro item significa que a entrada é conhecida para o sistema. O domínio do sistema significa a faixa de valores que as variáveis de estado podem assumir. A tolerância μ representa as incertezas do sistema, como os erros de modelagem e os ruídos.

Um modo q_i com equações modeladas \mathcal{M}_i , tem as saídas contínuas modeladas que variam com a tolerância μ_i , isto é, a saída modelada mais as incertezas. Se a saída medida do sistema pertencer ao conjunto de saídas do \mathcal{M}_i , esta saída é consistente com o modo q_i , caso contrário não.

Agora, considere os modos q_i e q_j com equações modeladas \mathcal{M}_i e \mathcal{M}_j , respectivamente. Se a saída medida do sistema for consistente com as saídas modeladas de q_i e q_j , não é possível distinguir em qual modo o sistema se encontra, e portanto os modos são não distinguíveis. Portanto, pode-se notar que a distinguibilidade tem relação direta com as saídas possíveis do sistema. Ao verificar a distinguibilidade dos modos utilizando modelos, é necessário considerar todas as incertezas inerentes do sistema, a fim de que os resultados obtidos na análise online sejam coerentes com os obtidos por simulação. Em outras palavras, q_i é distinguível de q_j , se todas as possíveis saídas medidas do sistema no modo q_i são consistentes somente com q_i , bem como, todas as possíveis saídas medidas do sistema no modo q_j são consistentes somente com q_j . Caso contrário, o modo q_i é não distinguível do modo q_j .

A tolerância do sistema é calculada por simulação, comparando a saída do modelo sem erro com a dos modelos com erro de modelagem e ruído branco gaussiano. Das

diferenças calculadas, a que possuir maior valor é considerada a tolerância do modelo. A tolerância do modo q_i é denotada por μ_i .

A verificação da distinguibilidade dos modos pode ser realizada através da saída contínua y e/ou das variáveis de estados contínuos x . A seguir é apresentado o método do $\|e_y\|$, enquanto os métodos do $\|e_x\|$ e do espaço de paridade são apresentados no apêndice A.

Método $\|e_y\|$

Este método é baseado no valor de saída dos modelos \mathcal{M}_i e \mathcal{M}_j , considerando os erros de modelagem e ruído. O valor da tolerância μ_i é obtido da análise de y_i , saída do modelo \mathcal{M}_i com erros de modelagem e ruído, e de y_{i_M} , saída do modelo sem erros, conforme a equação a seguir:

$$\mu_i = \|y_i - y_{i_M}\| \quad (5.3)$$

em que μ_i é a norma euclidiana da diferença de $y_i - y_{i_M}$.

A tolerância de \mathcal{M}_i será o maior valor obtido da equação 5.3, considerando todas as possíveis saídas de \mathcal{M}_i com erros de modelagem e ruído.

O valor do resíduo r_{ji} é obtido da comparação da saída de \mathcal{M}_j com erros de modelagem e ruído, denotada de y_j , com a saída de \mathcal{M}_i sem erros, denotada de y_{i_M} , conforme a equação a seguir:

$$r_{ji} = \|e_{ji}\| = \|y_j - y_{i_M}\| \quad (5.4)$$

em que o resíduo r_{ji} é a norma euclidiana do erro de $y_j - y_{i_M}$.

Os valores de r_{ji} são obtidos por simulação considerando todos os casos. Caso todos os valores de r_{ji} sejam maiores que μ_i , então os modos podem ser distinguíveis, caso contrário não.

Depois, é necessário comparar as saídas possíveis de \mathcal{M}_i com incertezas e as do \mathcal{M}_j sem erro. Se todos os valores de $r_{ij} > \mu_j$, então os estados podem ser

distinguíveis, caso contrário não.

Se todos os valores de $r_{ij} > \mu_j$ e de $r_{ji} > \mu_i$, então os modos são distinguíveis, caso contrário não.

Em outras palavras, a equação (5.4) compara os valores de saída dos modelos \mathcal{M}_j com incertezas e \mathcal{M}_i sem erro. O resultado desta equação é o resíduo, que representa a relação entre os dois modelos. Caso este resíduo seja menor que a tolerância μ_i , então os valores de saída do modelo \mathcal{M}_j pertencem aos possíveis valores de saída de \mathcal{M}_i , e portanto consistente com o \mathcal{M}_i . Neste caso, os valores de saída do modelo \mathcal{M}_j são consistentes com \mathcal{M}_i e \mathcal{M}_j , e portanto não distinguíveis.

Note que $r_{ji} > \mu_i$, não garante a distinguibilidade dos modos, pois se $r_{ij} < \mu_j$ os modos são não distinguíveis.

5.2.3 Análise da diagnosticabilidade utilizando o diagnosticador clustered G_{CD}

A seguir, é apresentado um exemplo, cujo sistema híbrido é h -diagnosticável.

Exemplo 5.2 Considere o SH modelado por $H = (Q, X, Y, \Sigma, U, f, g, \phi, \Gamma, q_0, \mathbf{x}_0)$, cuja componente discreta $G = (Q, \Sigma, \phi, \Gamma, q_0)$ é apresentada na figura 5.2. Seja $\Sigma_o = \{a, b\}$ e $\Sigma_{uo} = \{\sigma_u, \sigma_f\}$. Para simplificar o exemplo, Inv , $Guard$ e ρ serão omitidos. Os modelos contínuos associados aos modos q_i , $i = 1, \dots, 5$, são dados por:

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} -3 & 1 \\ 0 & -3 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

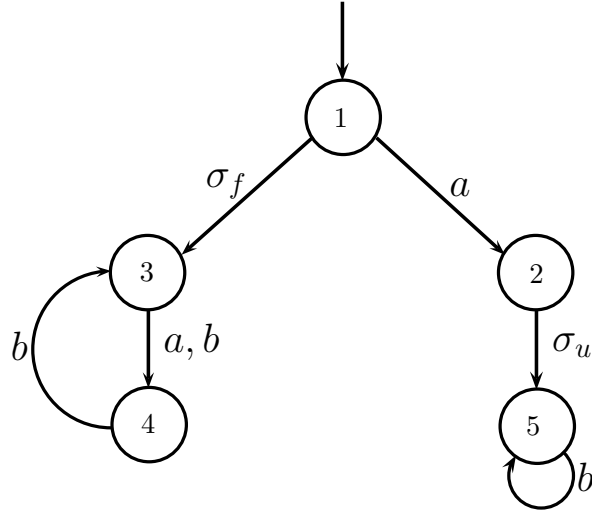


Figura 5.2: Componente discreta G do Exemplo 5.2

$$\mathbf{B}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{B}_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \mathbf{C}_3 = \begin{bmatrix} 1 & 1 \end{bmatrix},$$

$$\mathbf{C}_4 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \mathbf{C}_5 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{D}_1 = \mathbf{D}_2 = \mathbf{D}_3 = \mathbf{D}_4 = \mathbf{D}_5 = 1.$$

Ao aplicar o algoritmo 5.1, é obtido o autômato diagnosticador clustered G_{CD} , conforme ilustrado na figura 5.3. Note que G_{CD} não tem ciclos indeterminados, e, portanto, o sistema híbrido é h -diagnosticável.

Entretanto, o método de verificação da h -diagnosticabilidade em sistemas híbridos apresentado no algoritmo 5.1 somente pode ser utilizado para sistemas híbridos que não possuem estados com distinguibilidade ambigua em G_{CD} , isto é, um estado em G_{CD} que é alcançado por diferentes caminhos, tal que cada caminho gera um conjunto diferente de estados distinguíveis em G_{CD} . Para ilustrar o problema associado à existência de estados em G_{CD} com distinguibilidade ambígua, considere o seguinte exemplo.

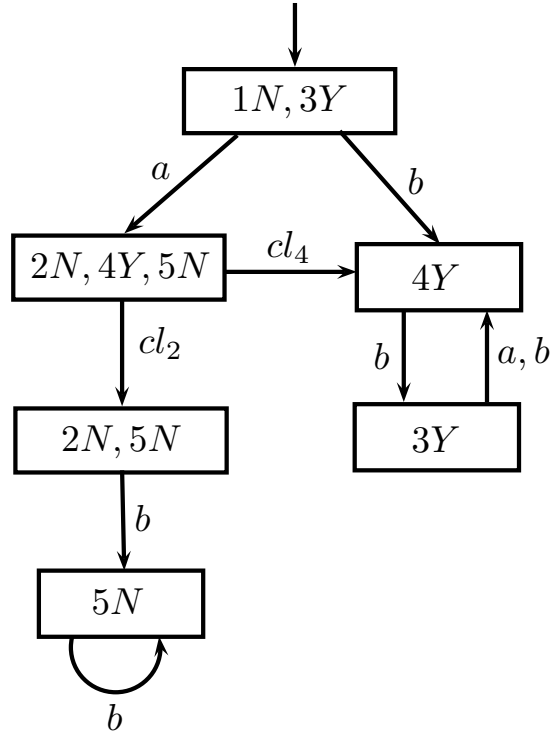


Figura 5.3: Autômato G_{CD} do Exemplo 5.2

Exemplo 5.3 Seja G , mostrado na figura 5.4, a componente discreta de um sistema híbrido modelado por H . Dado $\Sigma_o = \{a\}$ e $\Sigma_{uo} = \{\sigma_u, \sigma_f\}$ e assumamos que os modos 1, 2 e 5 são não distinguíveis e os modos 3, 4 e 6 são todos distinguíveis ($1 \sim 2 \sim 5$ e $3 \approx 4 \approx 6$). Então, aplicando o método proposto pelo algoritmo 5.1 para a verificação da h -diagnosticabilidade, o diagnosticador G_{CD} é obtido, conforme mostrado na figura 5.5. De acordo com o método proposto, os eventos cl_i são criados para representar a distinguibilidade entre os estados cabeças de G . Por exemplo, dois conjuntos de estados cabeças distinguíveis $\{3N, 4Y, 6N\}$ e $\{4Y, 6N\}$ podem ser associados com o estado $\{3N, 4Y, 6N\}$ de G_{CD} , que são alcançados, respectivamente, após a ocorrência das sequências $s_1 = a$ e $s_2 = aa$. Uma consequência da existência do estado de distinguibilidade ambígua $\{3N, 4Y, 6N\}$ em G_{CD} é o fato que a linguagem gerada por G_{CD} contém sequências que não são possíveis de ocorrer no procedimento de diagnóstico de falhas. Por exemplo, as sequências $acl_{6N}cl_{3N}$ e $acl_{6N}cl_{4Y}$ que pertencem a linguagem gerada por G_{CD} , porém não é possível que essas sequências ocorram no procedimento de diagnóstico de falhas, uma vez que os

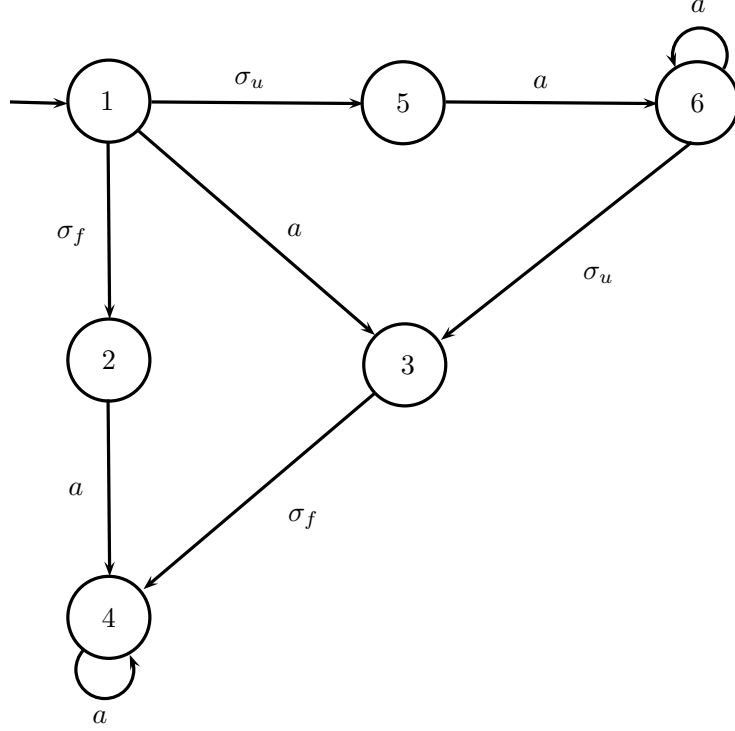


Figura 5.4: Componente discreta G do Exemplo 5.3

eventos cl_i são gerados somente após a ocorrência de um evento observável.

Com o objetivo de remover os estados de distinguibilidade ambígua em sistemas híbridos modelados por H , é introduzido o autômato rotulador de cabeça e cauda, $A_\ell^{HT} = (\{H, T\}, \Sigma, \phi_\ell^{HT}, \Gamma_\ell^{HT}, \{H\})$, em que $\phi_\ell^{HT}(H, \sigma) = \phi_\ell^{HT}(T, \sigma) = H$, se $\sigma \in \Sigma_o$, e $\phi_\ell^{HT}(H, \sigma) = \phi_\ell^{HT}(T, \sigma) = T$, se $\sigma \in \Sigma_{uo}$; $\Gamma_\ell^{HT}(T) = \Gamma_\ell^{HT}(H) = \Sigma$, conforme ilustrado na figura 5.6.

Note que a composição paralela de G com A_ℓ^{HT} rotula os estados de G alcançados por um evento observável com H , chamados de estados cabeças, e os estados de G alcançados por um evento não observável com T , chamados de estados caudas. Um novo autômato G_ℓ é construído pela operação paralela de G com os autômatos rotuladores A_ℓ^{HT} e A_ℓ^{YN} , isto é, $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$, como mostrado na figura 5.7 do exemplo 5.3.

Note que houve um aumento no número de estados de $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$ em relação a $G_\ell = G \parallel A_\ell^{YN}$ do Exemplo 5.3. Esse aumento se deve ao fato da rotulação dos estados de G em relação aos eventos observáveis e em relação ao evento de falha

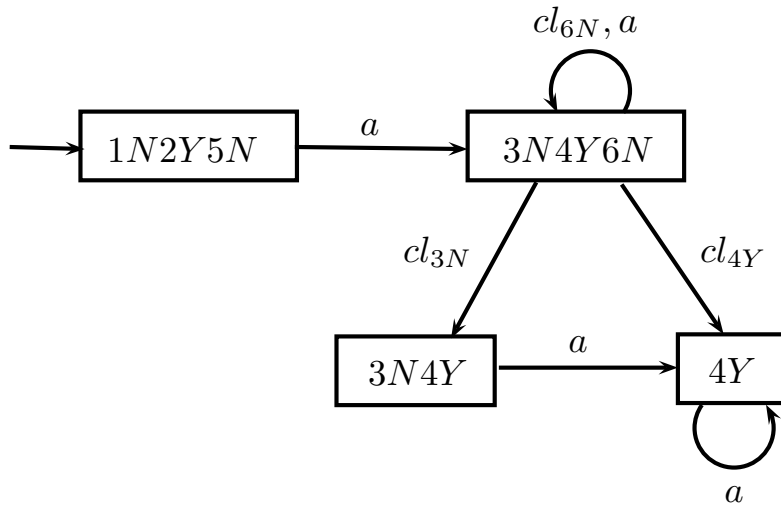


Figura 5.5: Autômato G_{CD} do Exemplo 5.3

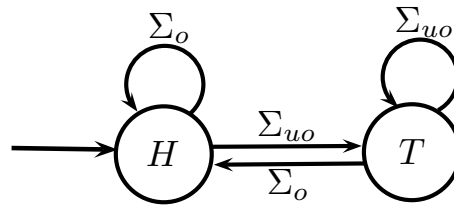


Figura 5.6: Autômato rotulador de cabeça e cauda A_ℓ^{HT}

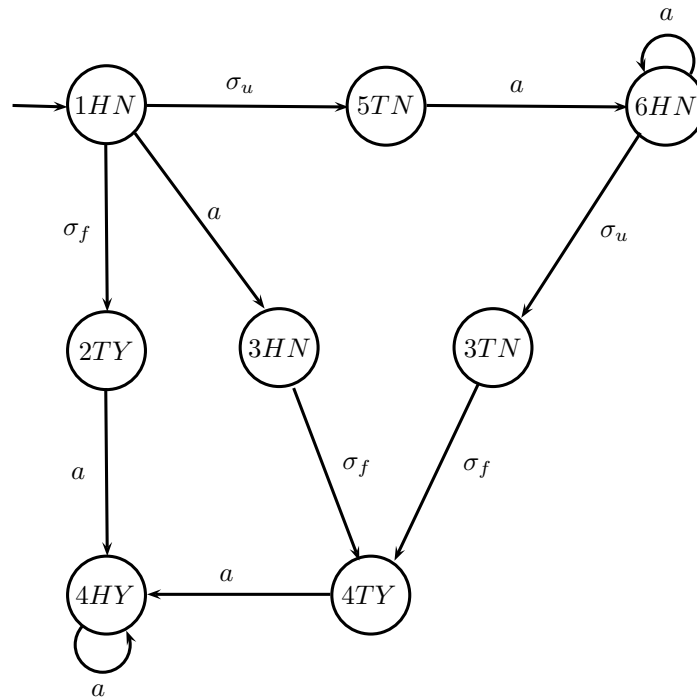


Figura 5.7: Autômato $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$ do Exemplo 5.3

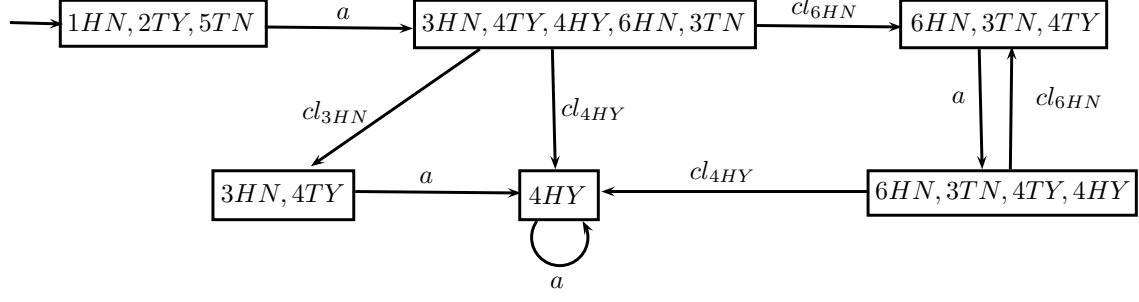


Figura 5.8: Autômato G_{CD} com $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$ do Exemplo 5.4

σ_f . Note ainda que os estados 3 e 4 de G aparecem duas vezes em $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$. O estado 3 aparece como $3HN$ e $3TN$, representando que esse estado é alcançado tanto por um evento observável como por um não observável. O mesmo acontece com o estado 4. Com a rotulação dos estados de G em relação ao eventos observáveis e em relação ao evento de falha σ_f , o problema associado à existência de estados em G_{CD} com distinguibilidade ambígua é resolvido, conforme mostrado no exemplo abaixo.

Exemplo 5.4 *Considere o mesmo sistema híbrido do exemplo 5.3. Então, aplicando o algoritmo 5.1 com $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$, para a verificação da h -diagnosticabilidade, o diagnosticador G_{CD} é obtido, conforme mostrado na figura 5.8. Note que todas as sequências geradas por G_{CD} podem ocorrer no procedimento de diagnóstico de falhas, e, portanto, resolvendo o problema associado à existência de estados em G_{CD} com distinguibilidade ambígua.*

Ao substituir o autômato $G_\ell = G \parallel A_\ell^{YN}$ por $G_\ell = G \parallel A_\ell^{HT} \parallel A_\ell^{YN}$ no algoritmo 5.1, o algoritmo 5.2 é criado e será apresentado na próxima seção.

5.2.4 Diagnosticador Híbrido G_H

O novo autômato diagnosticador $G_H = (Q_H, \Sigma_H, \phi_H, \Gamma_H, q_{0_H})$ utilizado na verificação da h -diagnosticabilidade de um SH , removendo o problema associado à existência de estados em G_{CD} com distinguibilidade ambígua, é apresentado a seguir.

Algoritmo 5.2

- *Passo 1:* Calcule $G_\ell = G \|A_\ell^{HT} \|A_\ell^{YN} = (Q_\ell, \Sigma, \Gamma_\ell, \phi_\ell, q_{\ell_0})$.
- *Passo 2:* Defina $q_{0_H} = UR(q_{\ell_0})$, $\Sigma_H = \Sigma_o$ e $Q_H = \emptyset$.
- *Passo 3:* Defina $\tilde{Q} = \{q_{0_H}\}$.
- *Passo 4:* $Q_H \leftarrow Q_H \cup \tilde{Q}$.
- *Passo 5:* Defina $\hat{Q} = \tilde{Q}$ e $\bar{Q} = \emptyset$.
- *Passo 6:* Para cada $B \in \hat{Q}$:
 - *Passo 6.1:* $\Gamma_H(B) = (\bigcup_{q_\ell \in B} \Gamma_\ell(q_\ell)) \cap \Sigma_o$
 - *Passo 6.2:* Para cada $e \in \Gamma_H(B)$:
 - *Passo 6.2.1:* $H_{cl} = \{q_\ell \in Q_\ell : (\exists \tilde{q}_\ell \in B)[q_\ell = \phi_\ell(\tilde{q}_\ell, e)]\}$.
 - *Passo 6.2.2:* $\phi_H(B, e) = UR(H_{cl})$ e $\bar{Q} = UR(H_{cl})$.
 - *Passo 6.2.3:* $Q_H \leftarrow Q_H \cup \{\phi_H(B, e)\}$.
 - *Passo 6.2.4:* $[h_C, s_C] = Distinguiabilidade(H_{cl})$.
 - *Passo 6.2.5:* Se $\|h_C\| = 1$, então $\tilde{Q} \leftarrow \tilde{Q} \cup \{\phi_H(B, e)\}$.
 - *Passo 6.2.6:* Se $\|h_C\| > 1$, então
 - * *Passo 6.2.6.1:* $\Sigma_C = \bigcup_{i=1}^{|s_C|} \{cl_i\}$, em que cl_i é o i -ésimo elemento da sequência de eventos s_C .
 - * *Passo 6.2.6.2:* $\Sigma_H \leftarrow \Sigma_H \cup \Sigma_C$.
 - * *Passo 6.2.6.3:* Para $i = 1$ a $\|h_C\|$ faça
 - *Passo 6.2.6.3.1:* $Q_H \leftarrow Q_H \cup \{UR(H_i)\}$, em que H_i é o i -ésimo elemento da sequência de conjunto de estados h_C .
 - *Passo 6.2.6.3.2:* $\phi_H(\bar{Q}, cl_i) = UR(H_i)$.
 - *Passo 6.2.6.3.3:* $\bar{Q} \leftarrow \bar{Q} \cup \{\phi_H(\bar{Q}, cl_i)\}$.

* Passo 6.2.6.4: $\Gamma_H(\bar{Q}) = \Sigma_C$.

- Passo 7: Repita os passos 5 a 6 até a total construção de G_H .

Note que a diferença do algoritmo 5.2 para o algoritmo 5.1 é no passo 1, no qual é obtido um novo autômato G_l , devido à rotulação em relação ao evento observável e em relação ao evento de falha σ_f . O novo autômato rotulado é obtido por $G_\ell = G \| A_\ell^{HT} \| A_\ell^{YN}$.

Uma condição necessária e suficiente para a h -diagnosticabilidade de um sistema híbrido pode ser encontrada utilizando as seguintes definições.

Definição 5.2 *Seja $q_H = \{(q_1, \ell_{1_1}, \ell_{2_1}), (q_2, \ell_{1_2}, \ell_{2_2}), \dots, (q_n, \ell_{1_n}, \ell_{2_n})\} \in Q_H$, em que $q_k \in Q$, $\ell_{1_k} \in \{H, T\}$ e $\ell_{2_k} \in \{Y, N\}$, para $k = 1, 2, \dots, n$. Então q_H é dito ser um estado discreto de falha, se $\ell_{2_k} = Y$ para $k = 1, 2, \dots, n$, e um estado discreto normal se $\ell_{2_k} = N$ para $k = 1, 2, \dots, n$. Se existir $(q_i, \ell_{1_i}, \ell_{2_i}), (q_j, \ell_{1_j}, \ell_{2_j}) \in q_H$, (q_i, ℓ_{1_i}) não necessariamente distinta de (q_j, ℓ_{1_j}) tal que $\ell_{2_i} = Y$ e $\ell_{2_j} = N$, então q_H é um estado discreto incerto de G_H . \square*

Definição 5.3 *Um conjunto de estados discretos incertos $\{q_{H_1}, q_{H_2}, \dots, q_{H_n}\} \subseteq Q_H$ forma um ciclo indeterminado no diagnosticador híbrido G_H , se as seguintes condições forem verdadeiras:*

- i) existe um caminho cíclico $\mathcal{P}_H = (q_{H_1}, \sigma_1, q_{H_2}, \sigma_2, \dots, q_{H_n}, \sigma_n, q_{H_1})$ em G_H ;*
- ii) $\exists (q_l^{k_l}, \ell_{1_l}^{k_l}, \ell_{2_l}^{k_l}), (\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}, \tilde{\ell}_{2_l}^{r_l}) \in q_{H_l}, (q_l^{k_l}, \ell_{1_l}^{k_l})$ não é necessariamente distinto de $(\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}), l = 1, 2, \dots, n, k_l = 1, 2, \dots, m_l, e r_l = 1, 2, \dots, \tilde{m}_l$ tal que*
 - a) $\ell_{2_l}^{k_l} = Y, \tilde{\ell}_{2_l}^{r_l} = N$, para todo l, k_l e r_l ;*
 - b) Os estados $\{(q_l^{k_l}, \ell_{1_l}^{k_l}, \ell_{2_l}^{k_l})\}, l = 1, 2, \dots, n, k_l = 1, 2, \dots, m_l$ e $\{(\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}, \tilde{\ell}_{2_l}^{r_l})\}, l = 1, 2, \dots, n, r_l = 1, 2, \dots, \tilde{m}_l$ podem ser rearranjados para formar um caminho cíclico em G_ℓ , tal que as correspondentes sequências s e \tilde{s} satisfazem $P_o(s) = P_o(\tilde{s}) = P(\bar{s})$, em que $P : \Sigma_H^* \rightarrow \Sigma^*$ e $\bar{s} = \sigma_1 \sigma_2 \dots \sigma_n$, nos quais $\sigma_1, \sigma_2, \dots, \sigma_n$ são definidos em i). \square*

O seguinte teorema fornece uma condição necessária e suficiente para a h -diagnosticabilidade de um sistema híbrido.

Teorema 5.1 *Seja G_H o diagnosticador híbrido de um sistema híbrido modelado por H . Então, H é h -diagnosticável se, e somente se, o diagnosticador híbrido G_H não tiver ciclos indeterminados.* \square

Prova: Suponha que exista um ciclo indeterminado em G_H e que H é h -diagnosticável. Se G_H tem um ciclo indeterminado, então existe $s_N \in L_N$ e $s_f \in L \setminus L_N$, tal que $P_o(s_N) = P_o(s_f) = \nu$. Sejam \mathcal{P}_F e \mathcal{P}_N os caminhos de falha e normal, associados, respectivamente, com s_f e s_N . De acordo com a definição 5.1, se existe uma sequência normal s_N , cuja $P_o(s_N) = P_o(s_f)$, então H é h -diagnosticável, se e somente se, existir $\eta\sigma_o \in \bar{\nu}$, tal que $q_F \approx q_N$, em que $q_F \in \Delta(\text{Reach}(G_{\mathcal{P}_F}, \eta), \sigma_o)$ e $q_N \in \Delta(\text{Reach}(G_{\mathcal{P}_N}, \eta), \sigma_o)$. Conforme o algoritmo 5.2, os estados q_F e q_N são separados após a verificação da distinguibilidade realizada no passo 6, evitando, assim, a existência de um ciclo indeterminado associado com \mathcal{P}_F e \mathcal{P}_N , o que contradiz a suposição da existência de um ciclo indeterminado em G_H .

Suponha agora que H é não h -diagnosticável. Então, de acordo com a definição 5.1, existe $s_N \in L_N$ e $s_f \in L \setminus L_N$ tal que $P_o(s_N) = P_o(s_f) = \nu$, e não existe $\eta\sigma_o \in \bar{\nu}$, tal que $q_F \approx q_N$, em que $q_F \in \Delta(\text{Reach}(G_{\mathcal{P}_F}, \eta), \sigma_o)$ e $q_N \in \Delta(\text{Reach}(G_{\mathcal{P}_N}, \eta), \sigma_o)$. Então, de acordo com o algoritmo 5.2, existirá um caminho em G_H associado com os caminhos \mathcal{P}_F e \mathcal{P}_N . Sendo s_f arbitrariamente longa após a ocorrência de um evento de falha, e G_H um autômato finito, então existirá um ciclo de estados incertos em G_H associado com st e s_N , que caracteriza um ciclo indeterminado. \blacksquare

De acordo com o teorema 1, o sistema híbrido modelado por H é h -diagnosticável se não existirem ciclos indeterminados em G_H . Então, com o objetivo de verificar a h -diagnosticabilidade de um SH , é necessário procurar ciclos incertos em G_H , e verificar se existem ciclos de estados normais e de falha em G_l , associados com o ciclo incerto em G_H . Essa tarefa tem um alto custo computacional, pois é necessário reconstruir ciclos de G_l oriundos dos ciclos em G_H . Uma condição necessária para um conjunto de estados discretos incertos formar um ciclo indeterminado é apresentada no teorema 5.2. Se nenhum dos ciclos incertos de G_H satisfizer essa condição, então

o sistema híbrido modelado por H é h -diagnosticável, e, portanto, não é necessário procurar os correspondentes caminhos cíclicos em G_l .

Teorema 5.2 *Se o conjunto de estados discretos incertos $\{q_{H_1}, q_{H_2}, \dots, q_{H_n}\} \subseteq Q_H$ forma um ciclo indeterminado em G_H , então existe um caminho cíclico $\mathcal{P}_H = (q_{H_1}, \sigma_1, q_{H_2}, \sigma_2, \dots, q_{H_n}, \sigma_n, q_{H_1})$ em G_H tal que $\exists (q_l^{k_l}, \ell_{1_l}^{k_l}, \ell_{2_l}^{k_l}), (\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}, \tilde{\ell}_{2_l}^{r_l}) \in q_{H_l}$, em que $\ell_{2_l}^{k_l} = Y$, $\tilde{\ell}_{2_l}^{r_l} = N$, e $\ell_{1_l}^{k_l} = \tilde{\ell}_{1_l}^{r_l} = H$, para $l = 1, 2, \dots, n$. \square*

Prova: Suponha que exista um estado $q_{H_l} \in \mathcal{P}_H$ tal que $\nexists (q_l^{k_l}, \ell_{1_l}^{k_l}, \ell_{2_l}^{k_l}), (\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}, \tilde{\ell}_{2_l}^{r_l}) \in q_{H_l}$ satisfazendo $\ell_{2_l}^{k_l} = Y$, $\tilde{\ell}_{2_l}^{r_l} = N$ e $\ell_{1_l}^{k_l} = \tilde{\ell}_{1_l}^{r_l} = H$. Uma vez que um estado cauda de q_{H_l} está no alcance não observável de um estado cabeça de q_{H_l} , e a rotulação Y se propaga após a ocorrência do evento de falha σ_f , então a única possibilidade para a não existência de $(q_l^{k_l}, \ell_{1_l}^{k_l}, \ell_{2_l}^{k_l}), (\tilde{q}_l^{r_l}, \tilde{\ell}_{1_l}^{r_l}, \tilde{\ell}_{2_l}^{r_l}) \in q_{H_l}$ satisfazendo $\ell_{2_l}^{k_l} = Y$, $\tilde{\ell}_{2_l}^{r_l} = N$ e $\ell_{1_l}^{k_l} = \tilde{\ell}_{1_l}^{r_l} = H$, é que o estado cauda de falha de q_{H_l} pertença ao alcance não observável de um estado cabeça normal de q_{H_l} . Assim, não é possível obter um caminho cíclico de estados de falha em G_l associado com o caminho cíclico \mathcal{P}_H , e portanto $\{q_{H_1}, q_{H_2}, \dots, q_{H_n}\}$ não é um ciclo indeterminado. \blacksquare

Note que a condição expressa no teorema 5.2 é apenas necessária, ou seja, é possível que haja um ciclo incerto que satisfaça à condição do teorema 5.2, e que não seja um ciclo indeterminado. O exemplo a seguir, apresentado em [13], ilustra esse fato.

Exemplo 5.5 *Seja G , a componente discreta do sistema híbrido modelado por H , cuja transição de estados é dada na figura 5.9, em que $\Sigma_o = \{a, b, c, d\}$ e $\Sigma_{uo} = \{\sigma_f\}$ [13]. Assuma que todos os modos são não distinguíveis. Executando o passo 1 do algoritmo 5.2, que rotula os modos em relação aos eventos observáveis na forma H e T e em relação ao evento de falha σ_f na forma N e Y , obtém-se o autômato G_l ilustrado na figura 5.10. Com o autômato rotulado G_l , execute o demais passos do algoritmo 5.2 até a obtenção do diagnosticador híbrido G_H ilustrado na figura 5.11.*

Note na figura 5.11, que existe um caminho cíclico $\mathcal{P}_H = (\{(3, H, Y), (6, H, N), (7, T, Y)\}, b, \{(4, H, Y), (8, H, Y), (9, H, N)\}, d, \{(3, H, Y), (6, H, N), (7, T, Y)\})$ em

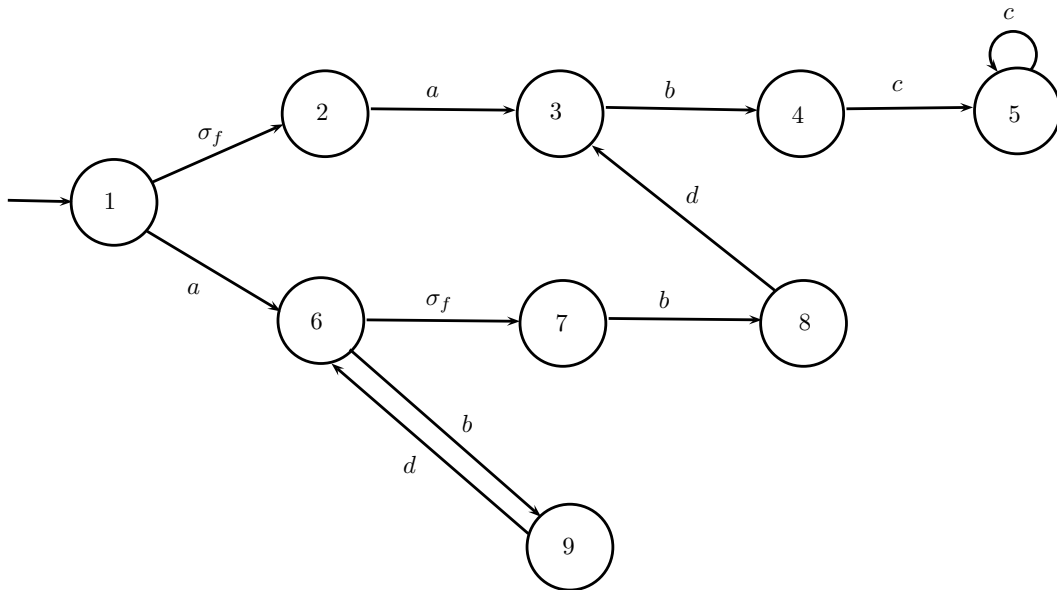


Figura 5.9: Autômato G do Exemplo 5.5.

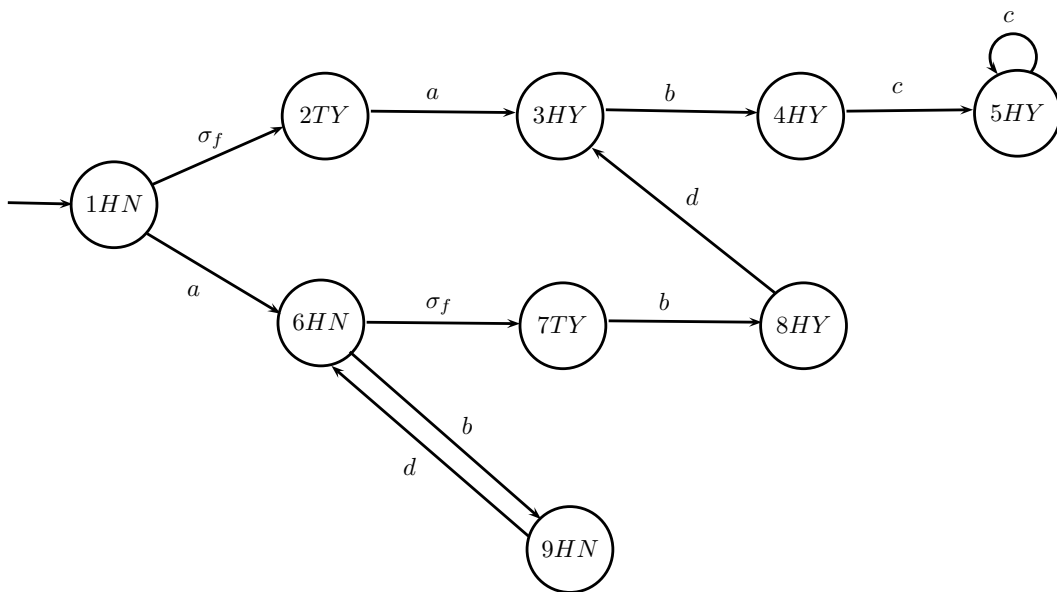


Figura 5.10: Autômato G_l do Exemplo 5.5.

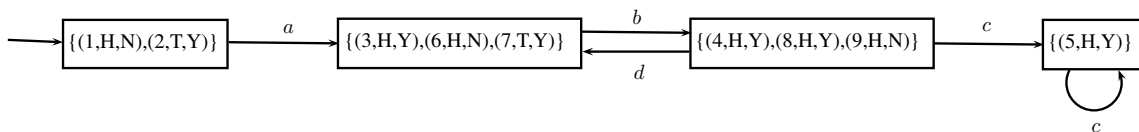


Figura 5.11: Autômato G_H do Exemplo 5.5.

G_H . Note ainda que em todos os estados do caminho cíclico, existem estados cabeças normais e de falha, conforme o teorema 5.2, e portanto é possível que exista um caminho cíclico indeterminado em G_H . O caminho cíclico da sequência normal em G_l é dado por $\mathcal{P}_N = ((6, H, N), b, (9, H, N), d, (6, H, N))$, enquanto os caminhos de falha são dados por: $\mathcal{P}_{f_1} = ((3, H, Y), b, (4, H, Y))$ e $\mathcal{P}_{f_2} = ((7, T, Y), b, (8, H, Y), d, (3, H, Y))$. Note que \mathcal{P}_{f_1} e \mathcal{P}_{f_2} não formam caminhos cíclicos em G_l , e, consequentemente, não formam um ciclo indeterminado em G_H . Portanto, o sistema híbrido modelado por H é h -diagnosticável.

O exemplo 5.6 tem o objetivo de mostrar que, ao adicionar as informações do sistema contínuo para o diagnóstico de falhas, um sistema híbrido H , cuja componente discreta G é não diagnosticável, torna-se h -diagnosticável.

Exemplo 5.6 *A partir do autômato híbrido do exemplo 5.1, deseja-se verificar se o sistema híbrido modelado por H é h -diagnosticável. Suponha que o erro de modelagem seja de 2%, o erro de medição seja um ruído branco gaussiano, o tempo para a verificação do modos é de 1s, os modos 3 e 9 distinguíveis bem como os modos 2 e 10 ($3 \approx 9$, $2 \approx 10$), e os demais modos são não distinguíveis ($1 \sim 8$, $5 \sim 11$, $4 \sim 12$ e $6 \sim 14$). De acordo com o algoritmo 5.2, o primeiro passo é rotular o autômato híbrido em relação a H e T e em relação Y e N . O autômato G_l , resultante da composição $G_{\ell_1} = G \| A_{\ell}^{HT} \| A_{\ell}^{YN}$, pode ser visto na figura 5.12. Em seguida, o autômato diagnosticador híbrido G_H é construído, após a execução dos passos 2 a 7, conforme mostrado na figura 5.13.*

Note que o número de estados aumentou de 15 em G para 22 em G_{ℓ} . Esse aumento se deve ao fato da necessidade de rotular os estados em relação a H e T , com a finalidade de solucionar o problema associado à existência de estados com distinguibilidade ambígua no diagnosticador.

O estado inicial de G_H é o estado $\{0HN\}$, pois não existe nenhum evento não-observável que pertença a função de eventos ativos de q_{l_0} , isto é, $\Gamma(\{0, H, N\}) \notin \Sigma_{uo}$.

Note ainda que em $\{(3HN, 9TY, 9HY)\}$, a componente contínua do sistema híbrido é analisada, através da verificação da distinguibilidade dos modos $\{3HN\}$ e

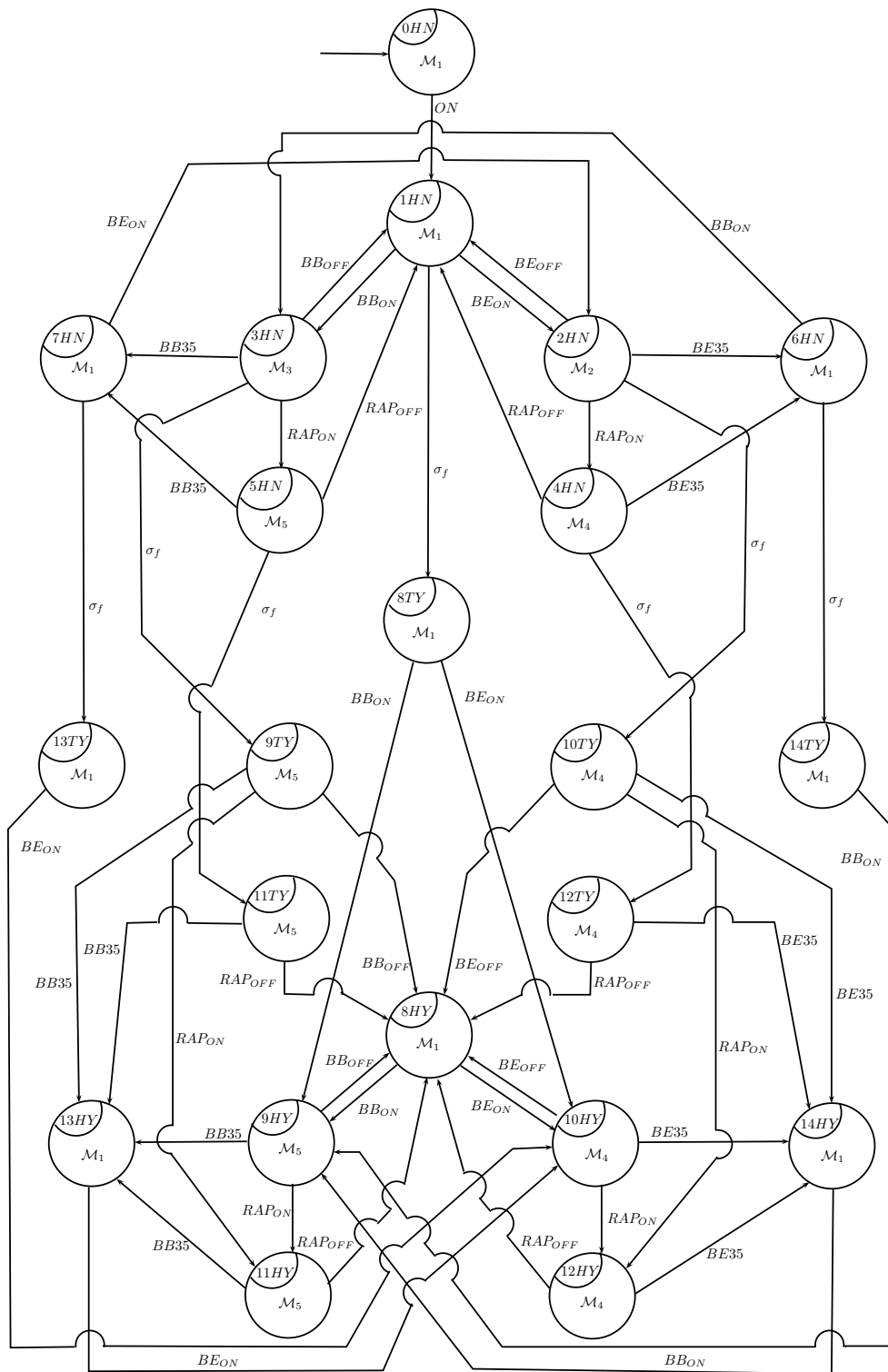


Figura 5.12: Autômato G_l do Exemplo 5.6.

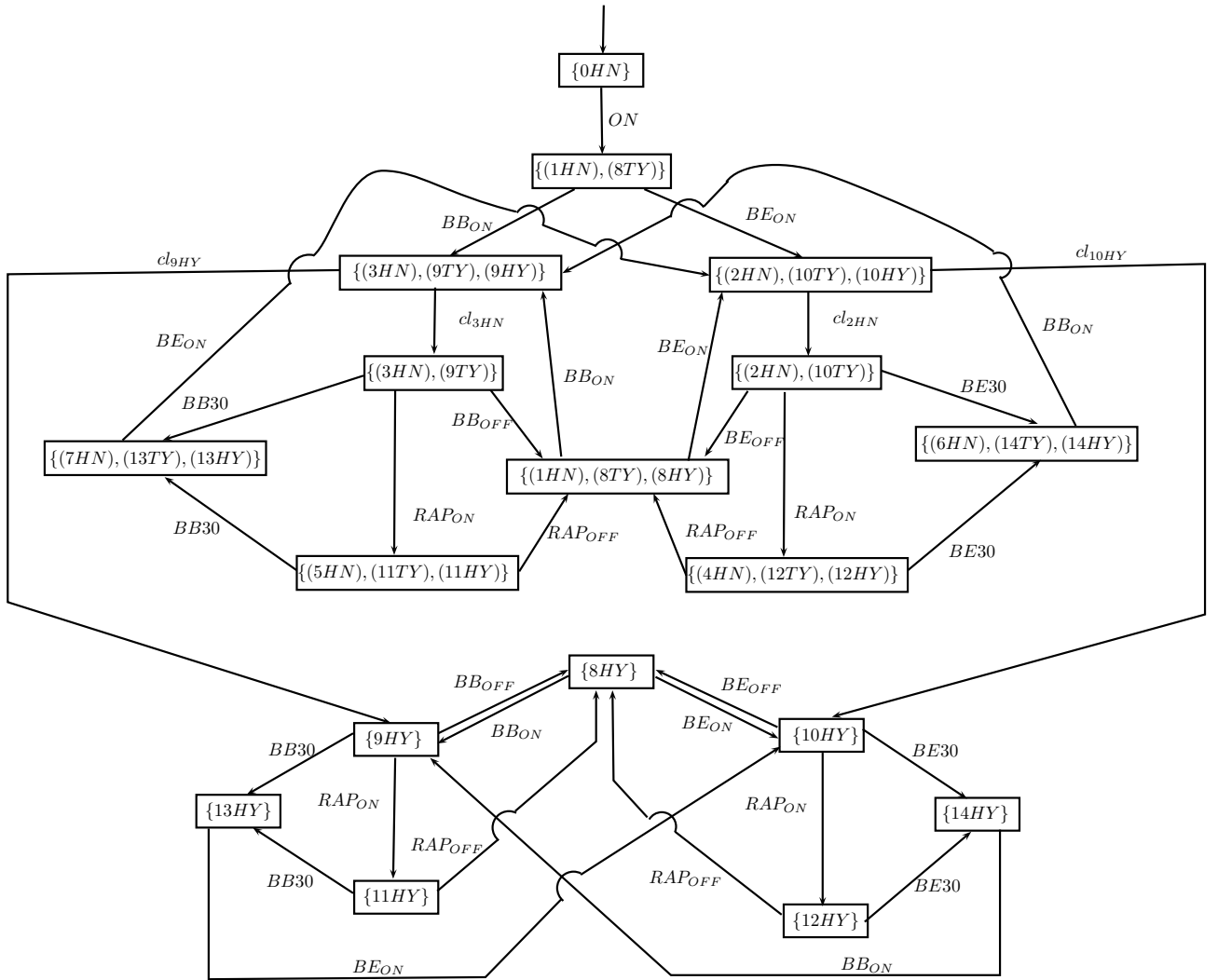


Figura 5.13: Autômato G_H do Exemplo 5.6.

$\{9HY\}$. Como $3HN \approx 9HY$, então a distinção entre modos é representada através dos eventos cl_{3HN} e cl_{9HY} .

A distinguibilidade dos modos $\{(3HN)\}$ e $\{(9HY)\}$ é mostrada na tabela 5.1, cujo tempo de avaliação é de 1 segundo. Note que o estado inicial em $\{(3HN)\}$ e $\{(9HY)\}$ é sempre $x^T = [0 \ 0 \ 0]$. A posição angular é zero, devido à hipótese da velocidade angular ser uniforme em toda a região limitada de $35^\circ BB$ a $35^\circ BE$, enquanto a velocidade e aceleração angular são iguais a zero porque o leme encontra-se parado.

O resíduo do modo normal é o resultado da comparação do modelo matemático de $\{(9HY)\}$ com erros de modelagem e de medição com o modelo matemático de $\{(3HN)\}$, de acordo com a equação 5.4. Enquanto a tolerância do modo normal é definida como sendo o maior valor que μ_{3HN} pode assumir, isto é, o maior resultado da comparação do modelo matemático de $\{(3HN)\}$ com e sem erros, de acordo com a equação 5.3.

Como o valor do resíduo do modo normal é maior que a tolerância do modo normal, bem como o valor do resíduo do modo rápido é maior que a tolerância do modo rápido, os modos são distinguíveis. Em outras palavras, os possíveis valores de saída em $\{(3HN)\}$ não pertencem aos possíveis valores de saída de $\{(9HY)\}$, e vice-versa. Dessa forma, quando diagnosticador G_H realizar a leitura do sensor de posição angular do leme, o resultado dessa leitura será consistente somente com o modelo matemático do modo normal ou do rápido. A figura 5.14 ilustra a variação da posição angular do modo normal e rápido em $\{(3HN), (9TY), (9HY)\}$, no qual é possível notar que as duas curvas não se cruzam no final do período de avaliação. A identificação dos estados $\{(3HN)\}$ e $\{(9HY)\}$ é representada no diagrama de transição de estados através de cl_{3HN} e cl_{9HY} .

Em relação ao estado $\{(2HN), (10TY), (10HY)\}$, esse também é separado em $\{(2HN)\}$ e $\{(10HY)\}$, pois os modos são distinguíveis, conforme tabela 5.2. Note que o estado inicial em $\{(2HN)\}$ e $\{(10HY)\}$ é sempre $x^T = [0 \ 0 \ 0]$, como em $\{(3HN)\}$ e $\{(9HY)\}$. A figura 5.15 ilustra a variação da posição angular do modo

Tabela 5.1: Tabela com a Tolerância e o Resíduo de $\{(3HN), (9TY), (9HY)\}$ do Exemplo 4.1.

Descrição	Valor
Tolerância Modo Normal	3.9449
Tolerância Modo Normal com ruído	4.3703
Resíduo Modo Normal	9.4091
Resíduo Modo Normal com ruído	9.5767
Tolerância Modo Rápido	4.9686
Tolerância Modo Rápido com ruído	5.2802
Resíduo Modo Rápido	10.1448
Resíduo Modo Rápido com ruído	10.2910

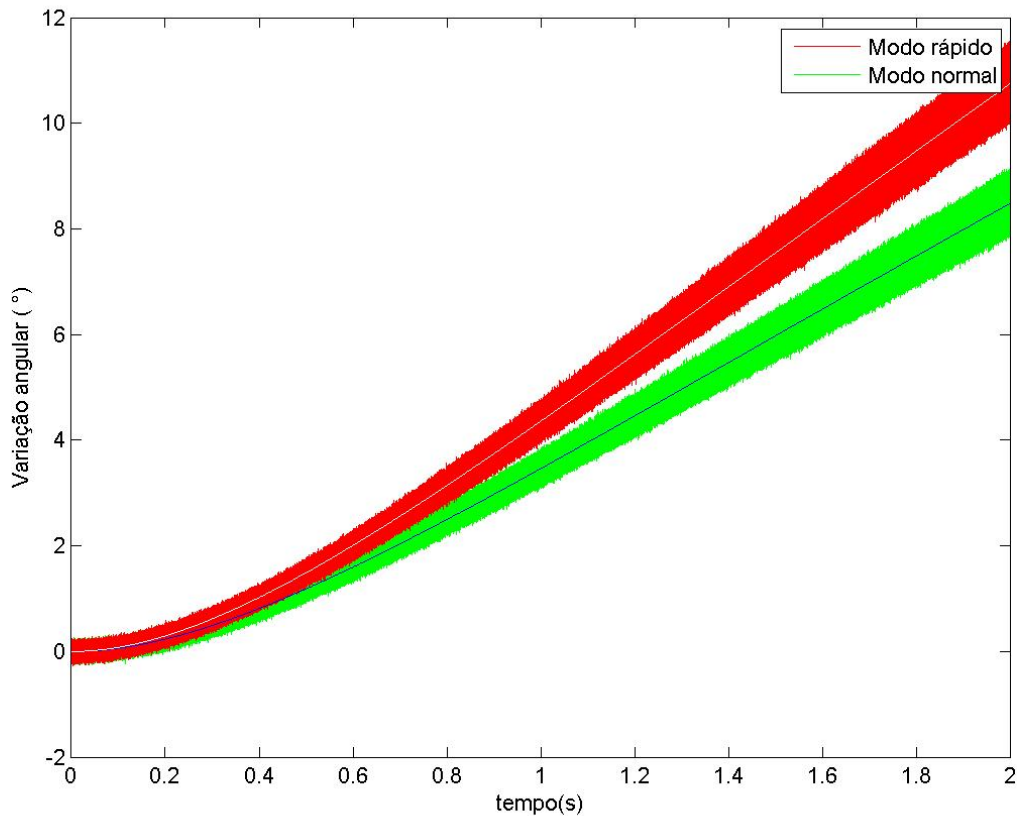


Figura 5.14: Variação da posição angular do Modo Normal e Rápido com erro de modelagem e ruído branco em $\{(3HN), (9TY), (9HY)\}$ do Exemplo 5.6.

Tabela 5.2: Tabela com a Tolerância e o Resíduo de $\{(2HN), (10TY), (10HY)\}$ do Exemplo 4.1.

Descrição	Valor
Tolerância Modo Normal	3.9449
Tolerância Modo Normal com ruído	4.3534
Resíduo Modo Normal	9.4091
Resíduo Modo Normal com ruído	9.5699
Tolerância Modo Rápido	4.9686
Tolerância Modo Rápido com ruído	5.2892
Resíduo Modo Rápido	10.1448
Resíduo Modo Rápido com ruído	10.3048

normal e rápido em $\{(2HN), (10TY), (10HY)\}$, no qual é possível notar que as duas curvas se separam antes de atingir 1s.

Note na figura 5.13, que o caminho $\mathcal{P}_H = ((3HN, 9TY, 9HY), cl_{3HN}, (3HN, 9TY), BB_{OFF}, (1HN, 8TY, 8HY), BB_{ON}, (3HN, 9TY, 9HY))$ em G_H é um caminho cíclico incerto, porém não é indeterminado. É possível formar um caminho cíclico normal em G_I através do $\mathcal{P}_N = (3HN, BB_{OFF}, 1HN, BB_{ON}, 3HN)$, porém não é possível rearrumar um caminho cíclico de falha, pois $\mathcal{P}_{F_1} = (9TY, BB_{OFF}, 8HY, BB_{ON}, 9HY)$ não é um caminho cíclico. O outro estado de falha que pertence a $\{(3HN, 9TY, 9HY)\}$ é $\{9HY\}$, porém este é identificado e separado pelo algoritmo 5.2 através de cl_{9HY} , não formando um caminho cíclico. Como não existem ciclos indeterminados em G_H , o sistema híbrido modelado por H é h -diagnosticável, conforme o teorema 5.1.

Outra forma de identificar se existem ciclos indeterminados em G_H é pelo teorema 5.2. Note que $\mathcal{P}_H = ((3HN, 9TY, 9HY), cl_{3HN}, (3HN, 9TY), BB_{OFF}, (1HN, 8TY, 8HY), BB_{ON}, (3HN, 9TY, 9HY))$ em G_H é um caminho cíclico, no qual cada estado desse caminho está associado a um estado normal e de falha, conforme a condição ii)a). Entretanto, a condição ii)b) é violada pelo estado $(3HN, 9TY)$, visto que $9TY$ não é um estado cabeça. Como não existem ciclos indeterminados em G_H , o sistema híbrido modelado por H é h -diagnosticável, conforme o teorema 5.2.

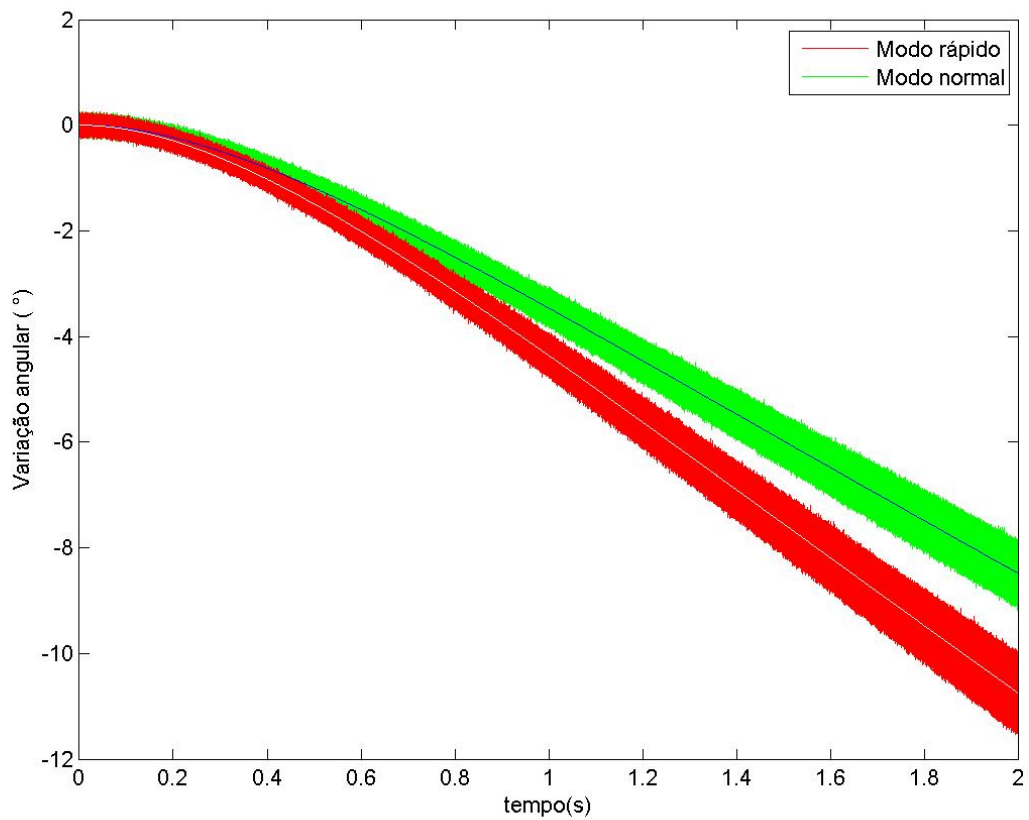


Figura 5.15: Variação da posição angular do Modo Normal e Rápido com erro de modelagem e ruído branco em $\{(2HN), (10TY), (10HY)\}$ do Exemplo 5.6.

Capítulo 6

Conclusões, contribuições e trabalhos futuros

6.1 Conclusões

Este trabalho apresentou uma nova definição para a diagnosticabilidade de um sistema híbrido, chamada de h -diagnosticabilidade. Além disso, foi apresentado um novo método para a verificação da h -diagnosticabilidade utilizando o diagnosticador híbrido G_H . Para a construção de G_H , a componente discreta G de um sistema híbrido modelado por autômato híbrido H foi rotulada em relação aos eventos observáveis e não-observáveis e em relação ao evento de falha σ_f , denotado de autômato G_l . Após a construção de G_l , foi construído o diagnosticador G_H que combina um diagnosticador de sistema a eventos discretos enriquecido com novos eventos associados a distinguibilidade dos modos de um sistema híbrido.

O diagnosticador proposto pode ser utilizado tanto no diagnóstico offline quanto no diagnóstico online de falhas. Uma das vantagens desse diagnosticador é que só verifica a distinguibilidade dos modos de um sistema híbrido H que são alcançados por eventos observáveis e não entre todos os modos. Outra vantagem é em relação ao tempo para executar essa verificação, pois essa é feita somente em um intervalo de tempo. Devido a essas particularidades, o método proposto neste trabalho, tem um

custo computacional menor do que os outros métodos de diagnóstico de falhas em sistemas híbridos propostos na literatura, porém é necessário um estudo específico para quantificar essa redução.

Além disso, foram propostos três métodos para verificar a distinguibilidade dos modos de um sistema híbrido H , conforme apresentado no capítulo 5 e no apêndice A. Um dos métodos de verificação da distinguibilidade dos modos foi empregado neste trabalho e seus resultados foram apresentados no exemplo 5.6.

Por fim, foi apresentado um exemplo de sistema híbrido que é o sistema de controle da máquina do leme de um navio, que é responsável por movimentar o leme e definir o rumo da embarcação. Para esse exemplo foi construído um autômato híbrido, no qual foi modelado primeiramente a componente discreta, e foi acrescentada a este modelo a componente contínua. Após a construção do autômato híbrido, foi verificada que a linguagem gerada pela componente discreta desse sistema híbrido era não diagnosticável, porém ao utilizar o novo método para o diagnóstico de falhas em sistema híbrido, o referido exemplo se tornou h -diagnosticável.

6.2 Contribuições

As principais contribuições desta dissertação no diagnóstico de falhas em sistemas híbridos são enumeradas a seguir:

- Uma nova definição para diagnosticabilidade em sistemas híbridos foi definida, denotado de h -diagnosticabilidade; e
- um novo método para a verificação da h -diagnosticabilidade utilizando o diagnosticador híbrido G_H foi apresentado.

Os resultados e as contribuições alcançados nesta dissertação decorreram na publicação dos seguintes artigos:

1. SILVA, E. A. R., DIENE, O., MOREIRA, M. V. “Diagnosability of Hybrid systems: Analysis and verification cast in DES framework”, Anais do XX Congresso Brasileiro de Automática, pp. 3136-3143, 2014.

2. DIENE, O., SILVA, E. A. R., MOREIRA, M. V. “Analysis and Verification of the Diagnosability of Hybrid Systems”, *53rd IEEE Conference on Decision and Control*, pp. 1-6, 2014.
3. DIENE, O., MOREIRA, M. V., ALVAREZ, V. R., SILVA, E. A. R. “Computational Methods for Diagnosability Verification of Hybrid Systems”, *IEEE Multi-Conference on Systems and Control (MSC)*, Submetido para publicação, 2015.
4. SILVA, E. A. R., ALVAREZ, V. R., DIENE, O., MOREIRA, M. V. “Modelagem do sistema de controle da máquina do leme de um navio utilizando o autômato híbrido”, *Simpósio Brasileiro de Automação Inteligente (SBAI)*, Submetido para publicação, 2015.

6.3 Trabalhos futuros

Como trabalho futuro é proposto:

- projetar um diagnosticador online híbrido baseado no método proposto de verificação da h -diagnosticabilidade em sistemas híbridos.
- Comparar o custo computacional do método proposto com os demais métodos de diagnóstico de falhas em sistemas híbridos propostos na literatura.
- Estudar a metodologia de diagnóstico de falhas em SH nos sistemas biológicos.

Apêndice A

Distinguiabilidade dos modos de um sistema híbrido

A.1 Método do observador dedicado em um SH

O método do observador dedicado estima as variáveis de estado baseado nas medidas das variáveis de saída e de controle. A figura A.1 ilustra a configuração de um estimador de estado de ordem completa.

Considere a planta definida por $\dot{x} = Ax + Bu$ e $y = Cx$. O modelo matemático do observador é o mesmo da planta, exceto pela inclusão da matriz de ganho do observador, cujo objetivo é incorporar as incertezas nas matrizes A e B, bem como pela ausência do erro inicial. O erro é a diferença entre a saída medida y e a estimada \hat{y} .

A entrada é u e a matriz H é matriz de ganho do observador, que representa a matriz de penalização do termo de correção que envolve a diferença entre a saída medida y e a saída estimada \hat{y} . Como o observador dedicado estima os valores das variáveis de estado, é possível calcular o erro destas variáveis da seguinte forma

$$\|e_x\| = \|x - \hat{x}\| \tag{A.1}$$

Sendo o erro de saída $e_y = y - \hat{y}$, em que $y = Cx$ e $\hat{y} = C\hat{x}$, obtém-se

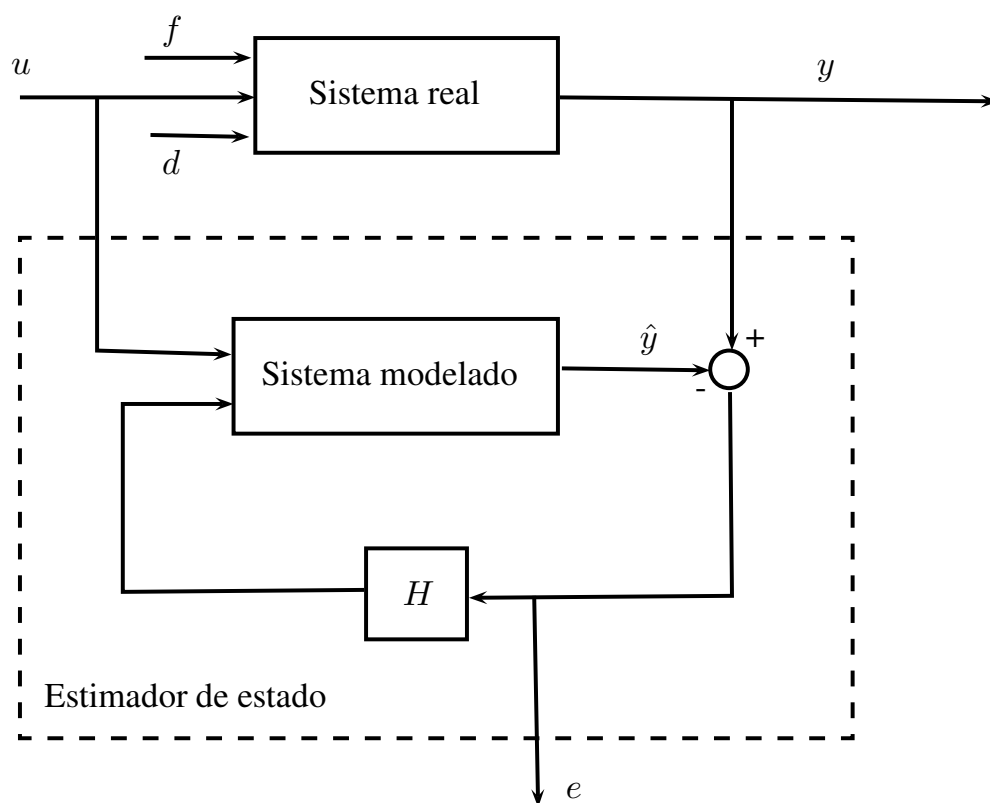


Figura A.1: Configuração básica de um gerador de resíduo através do método do observador dedicado

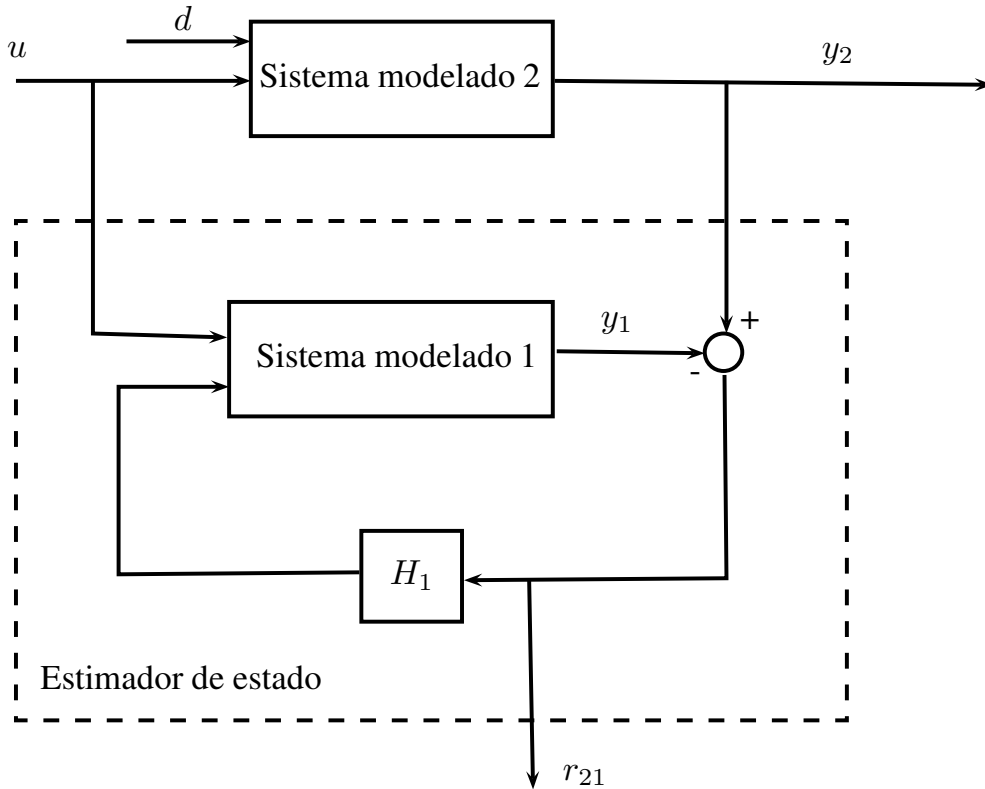


Figura A.2: Configuração básica de um gerador de resíduo para a verificação da distinguibilidade dos modos

$$\|e_y\| = \|y - \hat{y}\| = \|C e_x\| \quad (\text{A.2})$$

Note ainda que esta configuração compara o sistema real com o sistema modelado, e caso o erro $\|e_y\|$ seja maior do que o definido, o sistema contínuo modelado não é consistente com o sistema medido. Este deve ser utilizado para o cálculo online do erro e para a verificação da consistência. Como o objetivo do método proposto é verificar a distinguibilidade dos modos, é necessário realizar algumas modificações na estrutura do observador, conforme figura A.2.

Note na figura A.2, que o resíduo r_{21} é dado por

$$r_{21} = \|e_{21}\| = \|y_2 - y_1\| = \|C_2 x_2 - C_1 x_1\| \quad (\text{A.3})$$

em que x_2 é o valor das variáveis de estado de \mathcal{M}_2 com erros de modelagem e

ruído e x_1 é a estimativa das variáveis de \mathcal{M}_1 sem erro.

Note que nesse método o valor da saída de \mathcal{M}_2 , influencia na estimativa das variáveis de estado de \mathcal{M}_1 , e a diferença entre os valores de saída é dada pelo resíduo. Caso os resíduos obtidos para os casos de \mathcal{M}_2 com erros de modelagem e ruído e de \mathcal{M}_1 sem erro, sejam maiores que a tolerância μ_1 ($r_{21} > \mu_1$), os modelos podem ser distinguíveis, caso contrário não.

Depois, calcule r_{12} , para os casos de \mathcal{M}_1 com erros de modelagem e ruído e de \mathcal{M}_2 sem erros, e verifique se todos os valores de $r_{12} > \mu_2$. Caso afirmativo, os modelos 1 e 2 são distinguíveis, e, conseqüentemente, os seus modos são distinguíveis.

A.2 Método do espaço de paridade

Inicialmente, é apresentada a abordagem da redundância analítica para o isolamento e detecção da falha em um sistema contínuo [4]. O objetivo é estender alguns conceitos apresentados nessa abordagem e verificar a distinguibilidade dos modos de um SH . Assim, a geração do resíduo para a verificação da distinguibilidade de dois modos é realizada comparando os modelos matemáticos associados a estes modos, enquanto na abordagem da redundância analítica para o isolamento e detecção da falha em um sistema contínuo, o resíduo é gerado comparando a saída do modelo matemático do sistema em funcionamento normal com a saída medida do sistema.

A abordagem através da redundância analítica requer que o gerador de resíduos compare a saída do modelo matemático com a saída medida, ambos utilizando as medições de entrada do sistema. Desta comparação surge o resíduo. Se uma falha acontecer, as relações de redundâncias não são satisfeitas, e um residual, $r \neq 0$, ocorre. Assim, o residual pode ser utilizado com o objetivo detectar o tempo de ocorrência da falha, bem como sua localização. Para a geração do resíduo, são necessários três tipos de modelos: nominal, real ou observado, e o de falha. O modelo de falha deve representar o comportamento do sistema contínuo em falha.

A base para a decisão de uma ocorrência de uma falha é a assinatura de falha, conforme [4].

Definição A.1 (*Assinatura de falha*): é o resíduo que é obtido através do modelo matemático do sistema em falha. \square

Desta forma, a assinatura de falha é utilizado para identificar o tamanho, tipo e tempo de falha, no qual o resíduo é diferente de zero. Neste trabalho, a informação do resíduo tem como objetivo distinguir os modos de um sistema híbrido. Se o resíduo gerado pela comparação dos modelos matemáticos associados aos modos for zero, isto significa que os valores medidos das variáveis contínuas associados a esses modos são iguais, e portanto não é possível distingui-los, e conseqüentemente, os modos são não distinguíveis.

No método do espaço de paridade, a principal idéia é verificar a paridade, ou consistência, das equações matemáticas de um sistema com as saídas medidas, esta relação é chamada de relações de redundância analítica ou *analytical redundancy relation*. A falha é declarada se os limites de tolerância b_i são ultrapassados. Esta relação foi generalizada para o caso das relações de redundância temporal em sistemas contínuos, com objetivo de detectar falhas nestes sistemas, conforme [4, 29, 30].

Este conceito é estendido neste trabalho. Em [4, 29, 30], o resíduo é gerado pela comparação entre as equações matemáticas de um sistema com as saídas medidas, enquanto neste trabalho, o resíduo é gerado pela comparação entre as equações matemáticas associadas de um modo com outro de um SH . Assim, a distinguibilidade é detectada se o resíduo gerado ultrapassar os limites de tolerância μ .

Dado o sistema híbrido H , em que q_i e q_j são dois modos de H , associados, respectivamente, aos modelos contínuos \mathcal{M}_i e \mathcal{M}_j . O modelo \mathcal{M}_i é dado por

$$\begin{cases} \mathbf{x}(n+1) = \mathbf{A}_i \mathbf{x}(n) + \mathbf{B}_i \mathbf{u}(n) \\ \mathbf{y}(n) = \mathbf{C}_i \mathbf{x}(n) + \mathbf{D}_i \mathbf{u}(n) \end{cases}, \quad (\text{A.4})$$

e o modelo \mathcal{M}_j é dado por

$$\begin{cases} \mathbf{x}(n+1) = \mathbf{A}_j \mathbf{x}(n) + \mathbf{B}_j \mathbf{u}(n) \\ \mathbf{y}(n) = \mathbf{C}_j \mathbf{x}(n) + \mathbf{D}_j \mathbf{u}(n) \end{cases}. \quad (\text{A.5})$$

Para um determinado valor de $k \in \mathbb{N}$, em que k representa o número de interações do sistema, e calculando recursivamente o valor de y_i conforme a equação (A.4), obtém-se a seguinte equação em função do valor inicial $x(n)$

$$\begin{pmatrix} y_i(n) \\ y_i(n+1) \\ y_i(n+2) \\ \vdots \\ y_i(n+k) \end{pmatrix} = \begin{pmatrix} C_i \\ C_i A_i \\ C_i A_i^2 \\ \vdots \\ C_i A_i^k \end{pmatrix} x(n) + H_i \begin{pmatrix} u(n) \\ u(n+1) \\ u(n+2) \\ \vdots \\ u(n+k) \end{pmatrix} \quad (\text{A.6})$$

em que

$$H_i = \begin{pmatrix} D_i & 0 & & 0 \\ C_i B_i & D_i & 0 & \\ C_i A_i B_i & C_i B_i & D_i & 0 \\ \vdots & & & \\ C_i A_i^{k-1} B_i & \dots & \dots & C_i A_i B_i & C_i B_i & D_i \end{pmatrix} \quad (\text{A.7})$$

De forma análoga, é possível determinar y_j ,

$$\begin{pmatrix} y_j(n) \\ y_j(n+1) \\ y_j(n+2) \\ \vdots \\ y_j(n+k) \end{pmatrix} = \begin{pmatrix} C_j \\ C_j A_j \\ C_j A_j^2 \\ \vdots \\ C_j A_j^k \end{pmatrix} x(n) + H_j \begin{pmatrix} u(n) \\ u(n+1) \\ u(n+2) \\ \vdots \\ u(n+k) \end{pmatrix} \quad (\text{A.8})$$

em que

$$H_j = \begin{pmatrix} D_j & 0 & & 0 \\ C_j B_j & D_j & 0 & \\ C_j A_j B_j & C_j B_j & D_j & 0 \\ \vdots & & & \\ C_j A_j^{s-1} B_j & \dots & \dots & C_j A_j B_j & C_j B_j & D_j \end{pmatrix} \quad (\text{A.9})$$

Após determinar a evolução de $y_i(n+k)$ e $y_j(n+k)$, é necessário definir a matriz w_i , da seguinte forma:

$$w_i = \begin{pmatrix} C_i \\ C_i A_i \\ \vdots \\ C_i A_i^k \end{pmatrix} \quad (\text{A.10})$$

em que k representa o número de interações do sistema.

Com w_i definido, é necessário definir a matriz v_i cujas colunas geram o espaço nulo de w_i^T . Esta matriz contém todos os vetores v tal que $w_i^T v_i = 0$. Considere o espaço de paridade de ordem k definido por:

$$P = \{v_i | v_i^T w_i = 0\} \quad (\text{A.11})$$

Note que os vetores v são ortogonais a w_i e o produto desses vetores é igual a zero. Note ainda que a matriz w_i aparece na equação (A.8), associada à condição inicial $x(n)$. Reordenando esta equação, obtém-se

$$w_i x(n) = \begin{pmatrix} y_i(n) \\ y_i(n+1) \\ y_i(n+2) \\ \vdots \\ y_i(n+k) \end{pmatrix} - H_i \begin{pmatrix} u(n) \\ u(n+1) \\ u(n+2) \\ \vdots \\ u(n+k) \end{pmatrix} \quad (\text{A.12})$$

Multiplicando a matriz v_i^T por $w_i x(n)$, obtém-se o espaço de paridade de k através

do resíduo, conforme

$$r = v_i^T w_i x(n) = 0. \quad (\text{A.13})$$

Note que $v_i^T w_i = 0$ e portanto o resíduo será zero, independentemente do estado inicial $x(n)$. Note ainda que o cálculo do resíduo se restringiu somente ao cálculo das matrizes modeladas e do estado inicial.

Substitui-se a matriz w_i por w_j na equação (A.13), com o objetivo de verificar o espaço de paridade de ordem k do modelo \mathcal{M}_j . O novo resíduo é dado

$$r_{ji}(n+k) = v_i^T \begin{pmatrix} C_j \\ C_j A_j \\ \dots \\ C_j A_j^k \end{pmatrix} x(n) \quad (\text{A.14})$$

Calcule r_{ji} e r_{ij} , considerando o erro de modelagem. Caso os valores de r_{ji} sejam maiores que μ_i e os de r_{ij} sejam maiores que μ_j , então os modos q_i e q_j são distingueís, caso contrário não.

Note que o valor da tolerância foi definido considerando os erros de modelagem. Note ainda que se a condição inicial for nula, o resíduo é zero, independentemente do valor de $v_i^T w_j$.

Referências Bibliográficas

- [1] LYGEROS, J. *Lecture Notes on Hybrid Systems*. Lecture Notes, University of Patras, 2004.
- [2] ABRAHAM, E. *Modelling and Analysis of Hybrid Systems*. Lecture Notes, RWTH Aachen University, 2012.
- [3] VAN DER SCHAFT, A., SCHUMACHER, H. “An Introduction to Hybrid Dynamical Systems”. v. 2, *Lecture Notes in control and information sciences*, London Berlin Heidelberg, Springer-Verlag, 2000.
- [4] FRANK, P. M. “Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy- A Survey and Some New Results”, *Automatica*, v. 26, n. 3, pp. 459–474, 1990.
- [5] SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., et al. “Diagnosability of discrete-event systems”, *IEEE Trans. on Automatic Control*, v. 40, n. 9, pp. 1555–1575, 1995.
- [6] SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., et al. “Failure Diagnosis using discrete-event systems”, *IEEE Trans. on Control Systems Technology*, v. 4, n. 2, pp. 105–124, 1996.
- [7] QIU, W., KUMAR, R. “Decentralized failure diagnosis of discrete event systems”, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, v. 36, n. 2, 2006.
- [8] BASILIO, J., CARVALHO, L. K., MOREIRA, M. V. “Diagnose de falhas em Sistemas a eventos discretos modelados por autômatos finitos”, *Revista Controle e Automação*, v. 21, n. 5, pp. 510–533, 2010.
- [9] MOREIRA, M. V., JESUS, T. C., BASILIO, J. C. “Polynomial time verification of decentralized diagnosability of discrete event systems”, *IEEE Transactions on Automatic Control*, pp. 1679–1684, 2011.

- [10] BISWAS, S., SARKAR, D., MUKHOPADHYAY, S., et al. “Diagnosability analysis of real time hybrid systems”, *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 104–109, 2006.
- [11] BAYOUDH, M., TRAVE-MASSUYES, L. “Diagnosability analysis of hybrid systems cast in a discrete-event framework”, *Discrete Event Dynamic Systems: Theory and Applications*, v. 24, pp. 309–338, 2014.
- [12] DAIGLE, M., KOUTSOUKOS, X., BISWAS, G. “An event-based approach to integrated parametric and discrete fault diagnosis in hybrid systems”, *Transactions of the Institute of Measurement and Control*, v. 32, n. 5, pp. 487–510, 2010.
- [13] CASSANDRAS, C., LAFORTUNE, S. *Introduction to Discrete Event System*. Secaucus, NJ, Springer-Verlag New York, Inc., 2008.
- [14] RAMADGE, P. J., WONHAM, W. M. “The control of discrete-event systems”, *Proceedings of the IEEE*, , n. 77, pp. 81–98, 1989.
- [15] CARVALHO, L. K., MOREIRA, M. V., BASILIO, J. “Generalized robust diagnosability of discrete event systems”, *IFAC World Congress*, pp. 8737–8742, 2011.
- [16] CARVALHO, L. K., BASILIO, J., MOREIRA, M. V. “Robust diagnosis of discrete event systems against intermittent loss of observations”, *Automatica*, v. 48, n. 9, pp. 2068–2078, 2012.
- [17] CARVALHO, L. K., MOREIRA, M. V., BASILIO, J. “Robust diagnosis of discrete event systems against permanent loss of observations”, *Automatica*, v. 49, n. 1, pp. 223–231, 2013.
- [18] DEBOUK, R., LAFORTUNE, S., TENEKETZIS, D. “Coordinated decentralized protocols for failure diagnosis of discrete event systems”, *Discrete Event Dynamic Systems: Theory and Applications*, v. 10, n. 1, pp. 33–86, 2000.
- [19] BASILIO, J., LAFORTUNE, S. “Robust codiagnosability of discrete event systems”, *American Control Conference*, pp. 2202–2209, 2009.
- [20] MOREIRA, M. V., CABRAL, F. G., DIENE, O. “Petri Net Diagnoser for Discrete Event System Modeled by Finite State Automata”, *51st IEEE Conference on Decision and Control*, pp. 6742–6748, 2012.

- [21] CABRAL, F. G., MOREIRA, M. V., DIENE, O., et al. “A Petri Net Diagnoser for Discrete Event Ssystem Modeled by Finite State Automata”, *IEEE Transactions on Automatic Control*, v. 60, n. 1, pp. 59–71, 2015.
- [22] BASILE, F., CHIACCHIO, P., DE TOMMASI, G. “An efficient approach for online diagnosis of discrete event systems”, *IEEE Transactions on Automatic Control*, v. 54, n. 4, pp. 748–759, 2009.
- [23] RAMIREZ-TREVINO, A., RUIZ-BELTRAN, E., RIVERA-RANGEL, I., et al. “Online fault diagnosis of discrete event systems. A Petri net-based approach”, *IEEE Transactions on Automation Science and Engineering*, v. 4, n. 1, pp. 31–39, 2007.
- [24] BAYOUDH, M., TRAVE-MASSUYES, L., OLIVE, X. “Hybrid systems diagnosis by coupling continuous and discrete event techniques”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*, v. 17, 2008.
- [25] DAVID, R., ALLA, H. *Discrete, Continuous and Hybrid Petri Nets*. Springer, 2005.
- [26] SILVA, E. A. R., DIENE, O., MOREIRA, M. V. “Diagnosability of Hybrid systems: Analysis and verification cast in DES framework”, *Anais do XX Congresso Brasileiro de Automática*, pp. 3136–3143, 2014.
- [27] DIENE, O., SILVA, E. A. R., MOREIRA, M. V. “Analysis and Verification of the Diagnosability of Hybrid Systems”, *53rd IEEE Conference on Decision and Control*, pp. 1–6, 2014.
- [28] XIROS, N. I., TSOURAPAS, V. P., MOURTZOUCOS., K. K. “A Hydraulic Steering Gear Simulator for analysis and Control”, *World Scientific and Engineering Academy and Society -WSEAS*, v. 1, 2004.
- [29] CHOW, E. Y., WILLSKY, A. S. “Analytical redundancy and the design of robust failure detection systems”, *IEEE Transactions on Automatic Control*, v. 29, n. 7, 1984.
- [30] XI-CHENG LOU, A. S. W., VERGHESE, G. C. “Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems”, *Automatica*, v. 22, n. 3, 1986.