UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Érique Moser

Modelagem de processos industriais da indústria de óleo e gás como sistema a eventos discreto para o diagnóstico de falhas: um estudo de caso da planta SMAR PD3

Florianópolis
2023

Érique Moser

Modelagem de processos industriais da indústria de óleo e gás como sistema a eventos discreto para o diagnóstico de falhas: um estudo de caso da planta SMAR PD3

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia em Automação e Sistemas.
Orientador: Prof. Felipe Gomes de Oliveira Cabral, Dr.
Coorientador: Prof. Max Hering de Queiroz, Dr.
Prof. Publio Macedo Monteiro Lima, Dr.

Florianópolis
2023

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Moser, Érique
Modelagem de processos industriais da indústria de óleo e gás como sistema a eventos discreto para o diagnóstico de falhas: um estudo de caso da planta SMAR PD3 / Érique Moser ; orientador, Felipe Gomes de Oliveira Cabral, coorientador, Max Hering de Queiroz, coorientador, Publio Macedo Monteiro Lima, 2023.
80 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Diagnóstico de Falhas. 3. Sistemas a Eventos Discretos. 4. Indústria de Petróleo e Gás. 5. Operação segura. I. Cabral, Felipe Gomes de Oliveira. II. Queiroz, Max Hering de. III. Lima, Publio Macedo Monteiro IV. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. V. Título.

Érique Moser

Modelagem de processos industriais da indústria de óleo e gás como sistema a eventos discreto para o diagnóstico de falhas: um estudo de caso da planta SMAR PD3

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Marcos Vicente de Brito Moreira, Dr.
Instituição Universidade Federal do Rio de Janeiro

Prof. Carlos Barros Montez, Dr.
Instituição Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia em Automação e Sistemas.

Documento assinado digitalmente
Julio Elias Normey Rico
Data: 11/03/2024 15:01:51-0300
CPF: ***.840.859-**
Verifique as assinaturas em https://v.ufsc.br

Coordenação do Programa de Pós-Graduação

Documento assinado digitalmente
Felipe Gomes de Oliveira Cabral
Data: 11/03/2024 15:04:52-0300
CPF: ***.807.927-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Felipe Gomes de Oliveira Cabral, Dr.
Orientador

Florianópolis, 2023.

Dedico este trabalho aos profissionais do setor industrial
que zelam pela segurança, e em especial,
aos 11 petroleiros da P-36 *(in memoriam)*.

AGRADECIMENTOS

Gostaria de agradecer a todos que de alguma forma contribuíram para o desenvolvimento dessa pesquisa, em especial:

-aos meus mestres que me conduziram até aqui, com quem tanto aprendi. Por causa deles eu encontro perguntas e respostas que me inspiram dia após dia. Ao Felipe G. O. Cabral, Max H. de Queiroz e Públio M. M. Lima, que me ampararam nesse desafio, meus sinceros cumprimentos e minha eterna admiração.

-à Agência Nacional de Petróleo e Gás (ANP) e sua formação de engenheiros em automação, controle e instrumentação para petróleo, gás e biocombustíveis do PRH 2.1, por disponibilizar 24 meses de bolsa de mestrado que foram essenciais para realização dessa pesquisa. E à Universidade Federal de Santa Catarina, por disponibilizar tudo que preciso para minha caminhada acadêmica.

-aos meus amigos que são meus exemplos.

-à minha noiva Silvia, que me inspira nos estudos, por seu potencial e determinação. Me apoiou nesse projeto e acredita em mim sem hesitar. Aos seus pais Silvio *(in memoriam)* e Luciane, que estruturaram minha vida em Florianópolis e foram essenciais.

-aos meus pais Joni e Angelita que me ensinaram a firmeza de espírito e brandura de coração. Aos meus irmãos, Thiago e Driele, que compartilham seu amor comigo durante toda minha vida nos momentos bons e ruins. E aos meus familiares, meu muito obrigado.

"O aço é mais duro que a pedra,
que é mais dura que a madeira,
que é mais dura que a água,
que é mais dura que o ar.
Contudo, aquilo que não pode
ser visto é a coisa mais dura.
Que é o espírito interior do homem"
(Autor Desconhecido)

RESUMO

Processos industriais modernos são geralmente controlados hierarquicamente em dois níveis. Uma lógica de controle de tempo contínuo, como um controlador Proporcional Integral Derivativo (PID), por exemplo, é implementada no nível inferior para garantir que o comportamento do sistema atinja especificações de regimes transitório e permanente, enquanto no nível superior, um controle supervisório garante especificações de segurança para operações rotineiras e não rotineiras. Em geral, ambas as estratégias de controle são projetadas sem considerar que os componentes do sistema possam sofrer falhas que modifiquem seu comportamento esperado. Assim, uma estratégia de diagnóstico de falhas é fundamental para indicar a ocorrência de um evento de falha que pode causar um desvio do comportamento esperado do sistema. Essa estratégia também visa adicionar uma camada de proteção no sistema, ao sistema supervisório de segurança existente. Neste trabalho, uma estratégia de diagnóstico de falhas para sistemas a eventos discretos é explorada para um sistema de nível de líquido de tanque controlado em dois níveis. Para tanto, considera-se que pode ocorrer um evento de falha na válvula, o que faz com que a válvula fique travada. É proposto uma estratégia de modelagem para a válvula que considera as consequências da falha e sua interação com os demais componentes. A principal contribuição desse trabalho está na técnica de modelagem da falha, que garante a obtenção do modelo pós falha a partir dos modelos e especificações do projeto de controle supervisório do nível hierárquico superior. Por fim, é mostrado que esta abordagem não interfere no projeto de controle, o que permite que a síntese do diagnosticador seja realizada e implementada em um sistema controlado já em execução.

Palavras-chave: Diagnóstico 1. Diagnosticabilidade 2. Sistema de Eventos Discretos 3. Processo Industrial 4. Operação segura 5.

ABSTRACT

Modern industrial processes are generally hierarchically controlled in two levels. A continuous time control logic, such as a Proportional Integral Derivative (PID) control, for example, is implemented at the lower level to ensure that the system behavior attains transient and steady-state specifications, while at the higher level, a supervisory control ensures safety specifications for routine and non-routine operations. In general, both control strategies are designed without considering that system components can suffer faults that can modify their expected behavior. Therefore, a fault diagnosis strategy is essential to quickly indicate the occurrence of a fault event that may cause a deviation from the expected behavior of the system. This strategy also aims to add a layer of protection to the existing security supervisory system. In this work, a fault diagnosis strategy for discrete event systems is explored for a two-level controlled tank liquid level system. Therefore, it is considered that a fault event can occur in the valve, which causes the valve to be stuck closed. A modeling strategy for the valve is proposed that considers the fault consequences and its interaction with the remaining components. The main contribution of this work is the failure modeling technique, which guarantees the acquisition of the post-failure model based on the models and specifications of the supervisory control project at the higher hierarchical level. Finally, it is shown that this approach does not interfere with the control project, which allows the diagnoser synthesis to be achieved and implemented in a controlled system already running.

Keywords: Diagnosis 1. Diagnosability 2. Discrete-Event System 3. Industrial Process 4. Safe Operation 5.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CLP	Controlador Lógico Programável
DAS	Departamento de Automação e Sistemas
FF	Foundation Fieldbus
HSE	High Speed Ethernet
IEC	International Electrotechnical Commission
IEMA	Instituto de Energia e Meio Ambiente
IHM	Interface Homem Máquina
IPG	Indústria de Petróleo e Gás
ISA	International Society of Automation
ISP	Interoperable System Protocol
LAS	Link Active Scheduler
LEEM	Laboratório Experimental de Escoamento Multifásico
OIE	Oferta Interna de Energia
PD3	Planta Didática III
PID	Proporcional Integral Derivativo
SCADA	Sistema de Supervisão e Aquisição de Dados
SED	Sistema a Evento Discreto
SIS	Sistema Instrumentado de Segurança
TCS	Teoria de Controle Supervisório
UFSC	Universidade Federal de Santa Catarina

LISTA DE SÍMBOLOS

$\mathbb{N}$	Conjunto dos números Naturais
$\mathbb{Z}$	Conjunto dos números Inteiros
$\mathbb{R}$	Conjunto dos números Reais
Σ	Conjunto de Eventos de um SED
$\varnothing$	Conjunto vazio
ε	Sequência vazia
L_aL_b	Operação de concatenação entre as linguagens L_a e L_b
Σ^*	Fecho de Kleene de um conjunto de eventos Σ
L^*	Fecho de Kleene da linguagem L
$\overline{L}$	Prefixo fechamento da linguagem L
L/s	Pós linguagem de L após a sequência s
P_s	Projeção do conjunto de eventos Σ_l para o conjunto de eventos Σ_s
$\Sigma_l \setminus \Sigma_s$	Operação de subtração de conjuntos entre as linguagens Σ_l e Σ_s
P_s^{-1}	Projeção inversa
G	Autômato determinístico de estados finitos
Q	Conjunto de estados de um autômato
q_0	Estado inicial de um autômato
Q_m	Conjunto de estados marcados de um autômato
$\Gamma(q)$	Função de eventos ativos de um estado de um autômato
$L_m(G)$	Linguagem marcada de um autômato G
$Ac(G)$	Operação de Acessibilidade
$CoAc(G)$	Operação de coacessibilidade
$Trim(G)$	Operação *trim*
$G_1 \times G_2$	Composição produto entre os autômatos G_1 e G_2
$G_1 \mid\mid G_2$	Composição paralela entre os autômatos G_1 e G_2
Σ_o	Conjunto de eventos observáveis
Σ_{uo}	Conjunto de eventos não observáveis
$UR(q)$	Alcance não observável de um estado
Υ	Estrutura de controle com todas as ações possíveis de controle
Σ_c	Conjunto de eventos controláveis
Σ_{uc}	Conjunto de eventos não controláveis
S/G	Malha-fechada que representa $\mathbf{S}$ controlando G
$\mathbf{S}$	Supervisor
K	Especificação do sistema
$SupC(E,G)$	Elemento supremo com a máxima linguagem controlável
G_{conf}	Autômato que representa o sistema conflitante
G_{coord}	Autômato Coordenador do sistema conflitante

SUMÁRIO

1 INTRODUÇÃO

1.1 A INDÚSTRIA DE PETRÓLEO E GÁS

A Indústria de Petróleo e Gás (IPG) é fundamental na economia brasileira, correspondendo a cerca de 6,6% do PIB industrial do país (CNI, 2019). Por ser uma indústria de base, diversos processos e serviços dependem da extração e refino do petróleo. Ele é utilizado no transporte que abastece toda a cadeia de circulação de produtos e pessoas, e é uma importante fonte de energia. A Oferta Interna de Energia (OIE) coloca o petróleo como líder da oferta total de energia, representando cerca de 33,11% da contribuição absoluta (EPE, 2021). Esse ranking representa a importância da energia para movimentar a economia do país, portanto, mesmo com o aumento da energia renovável se desenvolvendo com intensidade nos últimos anos, observa-se que o petróleo continua apresentando extrema relevância no âmbito global apesar do esforço para redução de utilização dele e de seus derivados (MDIC, 2022). Além do petróleo ser utilizado para a fabricação de produtos como combustíveis, lubrificantes, plásticos, borrachas sintéticas, tintas, solventes e asfalto, os seus derivados são utilizados em termelétricas. Segundo o Instituto de Energia e Meio Ambiente (IEMA) em 2021, o Brasil viveu a maior crise hídrica registrada até então. Como consequência, a geração de energia elétrica foi afetada e a principal medida adotada para evitar um apagão energético foi acionar as termelétricas fósseis (IEMA, 2021). Justifica-se, portanto, o constante incentivo de pesquisa e desenvolvimento tecnológico na área da IPG, uma vez que ela seleciona a matéria-prima para os inúmeros outros processamentos futuros e consequentemente gera fluxo na economia através dos seus derivados com a geração de empregos, com a comercialização dos produtos pela sociedade, cobrança de impostos e geração de energia. Além disso, a eficiência ambiental na IPG é fundamental para mitigar os impactos ambientais enquanto adequa-se uma matriz energética mais sustentável. Por exemplo: i) a redução de emissões de gases de efeito estufa, contribui para a mitigação das mudanças climáticas, ii) a minimização de destruição de ecossistemas gerados pelo derramamento de petróleo, protege a biodiversidade, iii) a conservação de recursos naturais não renováveis por mais tempo é importante, já que a transição completa para fontes renováveis pode levar tempo, e iv) existem benefícios econômicos ao manter a criação de empregos que auxilia o desenvolvimento de muitas regiões.

1.1.1 Acidentes da Indústria de Petróleo e Gás

Os processos da IPG estão sujeitos à ocorrência de falhas que podem levar a grandes desastres. O acidente mais grave que ocorreu no Brasil foi em 2001, com a explosão e afundamento da plataforma P-36 da Petrobras (considerada até então, a maior plataforma de produção de petróleo do mundo), instalada no campo de Roncador, a 120km da costa, na Bacia de Campos, que provocou a morte de 11 petroleiros e grandes perdas econômicas e ambientais. Só a plataforma estava orçada em 500 milhões de dólares (NUNES, 2012). O desastre foi provocado

por problemas de projeto, manutenção e operação, que ocorreu em uma sequência de duas explosões e afundamento total da plataforma depois de 6 dias após o início do problema (GRABOIS, 2001). O incidente começou às margens da coluna de extração em uma das popas, em uma válvula do Tanque de Drenagem de Emergência. Esse tanque tem a função de armazenar hidrocarboneto em situações de emergência ou eventualidade manutenção dos separadores da planta de processo. Em resumo, em um determinado estágio da atividade, a válvula que regula a entrada de hidrocarbonetos era para estar fechada, porém uma delas não fechou completamente. O motivo pode ter sido erro de operação, porém, não é descartada a possibilidade de que houve o travamento dessa válvula (FIGUEIREDO; ALVAREZ; ADAMS, 2018) (PETROBRAS, 2015). Em consequência disso, o petróleo extraído fez um fluxo reverso que culminou no aumento de pressão do tanque até seu duto explodir. Em um segundo momento, o gás presente no tanque se espalhou nos níveis de operação no qual estavam 11 brigadistas que foram até o local para tentar combater o incidente. Alguns desses compartimentos da plataforma que não eram considerados de risco não apresentavam detectores de gás, e portanto não sinalizou o mesmo. Isso contribuiu para a segunda explosão se transformar na morte dos 11 brigadistas e uma briga judicial que perdura até os dias de hoje (CONSULTOR JURIDICO, 2023).

As falhas podem ter origem no desenvolvimento e dimensionamento do projeto, no manuseio inadequado dos equipamentos por parte dos operadores, no desgaste e quebra dos componentes devido à falta de manutenção, defeito de fabricação ou no não atendimento das especificações dos componentes. Um estudo realizado pela Marsh's Risk Consulting Practice (COCO, 2003), apontou falhas nos equipamentos (38%) e erros humanos (34%) como as maiores causas de acidentes em plataformas "offshore" situados no Golfo do México. O registro desses acidentes na história, e de muitos outros, lança luz sobre a preocupação que surge com a adoção de novas medidas de segurança para o enfrentamento desse problema. No caso do acidente da P-36, a comissão investigadora ANP/DPC recomendou que "os planos de contingência para acidentes de grande proporção e os esquemas de resposta a emergências de grande risco, necessitam ser imediatamente aprimorados, bem como a revisão de critérios de projetos de engenharia em unidades flutuantes de produção para assegurar maior proteção intrínseca". Tal descritivo representa a preocupação das entidades fiscalizadoras em mitigar acidentes que podem levar a grandes tragédias, com consequências muitas vezes irreversíveis. Além disso, evidencia-se a insuficiência de métodos mais seguros nos processos de produção de petróleo para evitar tais desastres. No caso do incidente da P-36, caso a válvula que deveria estar fechada fosse diagnosticada com algum defeito, um alarme poderia auxiliar no desligamento do sistema em virtude do equipamento defeituoso. Portanto, a sequência de eventos irregulares que se procederam após o fluxo reverso do fluido do tanque, contribuiria com a sinalização de que uma situação incomum estava prestes a acontecer.

1.1.2 Camadas de Segurança

Com o objetivo de tornar sistemas de engenharia mais seguros, algumas medidas de segurança são adotadas durante o desenvolvimento desses sistemas. Na prática, a segurança é comumente realizada por meio de níveis chamadas de camadas de proteção, que devem ser acionadas conforme a falha se alastra globalmente, a fim de reduzir o impacto de ações danosas (SUMMERS, 2003) (REIS, 2018). As camadas de proteção são compostas de 7 categorias de equipamentos ou dispositivos capazes de evitar e conter acidentes, como pode ser visto na figura 1, e também impedir situações danosas para a máquina, a empresa e principalmente, o operador. As curvas representam o comportamento do sistema, de modo que cada camada tenta mantê-lo controlado. À medida que o sistema está em um novo comportamento não seguro, pula-se de camada para que medidas mais contundentes sejam acionadas para manter o funcionamento seguro do sistema. Isso acontece até um limite, onde há a ocorrência de um incidente, fazendo com que medidas de mitigação e contenção sejam tomadas para evitar o escalonamento do problema e uma possível catástrofe.

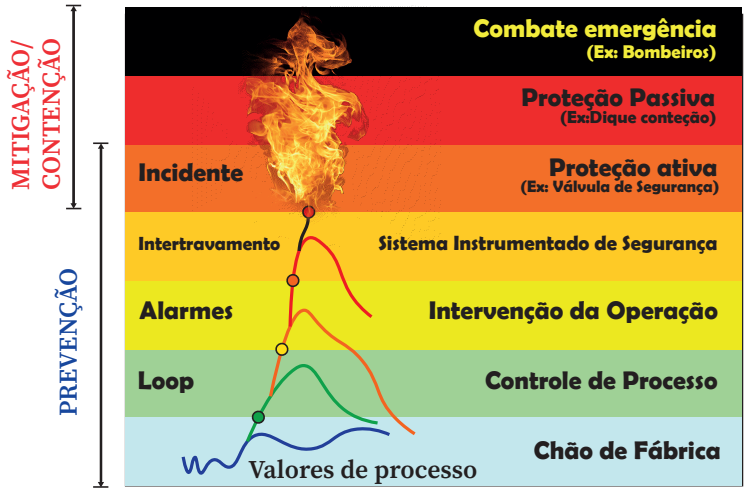


Figura 1 – Camadas de proteção de risco (figura do autor).

Pode-se perceber que o ponto central dessa análise de proteção é o de não concentrar toda a segurança do projeto em uma única camada e nem em um único local físico. Um desastre de grande proporção caracteriza-se por uma série de camadas que não atuaram quando era previsto de forma a não conter as falhas que surgiram. Portanto, quando as camadas de proteção de risco são aplicadas em um projeto, significa que estão sendo consideradas muitas ferramentas de análise, e técnicas de segurança como parte de uma estratégia multifacetada (CORDEIRO, 2021). Em seguida serão descritas as características de cada uma das camadas, apresentadas na figura 1:

- Camada de Chão de Fábrica: representa os sinais que são coletados e atuados, e que fazem parte de um sistema que foi projetado para reduzir ao máximo as falhas, dada a dinâmica do sistema. Pressupõe-se que nessa camada seja utilizado sensor apropriado

com um bom funcionamento a fim de imprimir valores corretos e, com isso, evitar uma leitura errada que poderia provocar a falha;

- Camada de Controle de Processo: representa os equipamentos voltados a controlar de forma segura o sistema, visando sua proteção. Nesse caso a falha poderia se alastrar a partir de um controle impreciso ou com uma lógica errada, que induziria condições não seguras;
- Camada de Intervenção de Operação: se destina à ação dos operadores que percebendo uma anormalidade, tomam ação para evitar um incidente; Na falta de operadores no entorno do equipamento para parar o sistema ou tomar alguma providência, a falha poderia se alastrar.
- Camada de Sistema Instrumentado de Segurança (SIS): apresenta um sistema de controle e monitoramento redundante, realizado com uma série de equipamentos que fazem parte de uma linha chamada *safety*. Uma falha nessa camada poderia se alastrar a partir da quebra do dispositivo, instalação incorreta, instalação em local impróprio e afins;
- Camada de Proteção Ativa: contempla dispositivos do equipamento que evitam que o incidente se torne ainda maior, e podem ser válvula de segurança, disco de ruptura, saída de pressão do equipamento, entre outros. No caso de entupimento de válvula de segurança, dano no sistema de alívio ou obstrução física de outros materiais no entorno, uma falha que seria gerada em outra camada, não poderia ser amenizada;
- Camada de Proteção Passiva: é composta por diques, bacias, proteções e elementos que evitem a propagação do incidente. Em uma situação onde as proteções são subdimensionadas ou algum operador se encontra dentro da zona de perigo ao realizar uma manutenção, por exemplo, a falha seria alastrada para um incidente ainda maior;
- Camada de Combate Emergência: é responsável pelo sistema de sirenes, protocolos de evacuação e até mesmo chamada dos bombeiros. Nesse caso a falha poderia se alastrar no caso de um problema no sistema de comunicação com brigadistas ou falta de tempo para os combatentes chegarem no local.

1.2 DIAGNÓSTICO DE FALHAS DE PROCESSOS INDUSTRIAIS

O diagnóstico de falhas desempenha um papel importante na operação segura dos sistemas de engenharia, pois a ocorrência de um evento de falha pode levar o sistema a comportamentos imprevisíveis, reduzindo a vida útil dos equipamentos, comprometendo a segurança dos operadores e causando danos ambientais. No trabalho seminal de Sampath *et al.* (1995a, 1996), um diagnosticador é proposto para um Sistema a Evento Discreto (SED) modelado como um autômato. SED é um sistema dinâmico de estados discretos cuja transição de estados se

dá por meio da ocorrência, em geral assíncrona, de eventos. Qualquer sistema pode ser modelado como um SED dependendo do nível de abstração desejado e isso o torna útil em diversas áreas de estudo, trazendo grande contribuição em sistemas de informação, computação, automação e manufatura, por exemplo. Como veremos adiante, o autômato será utilizado para representar um conjunto de sequências de eventos como uma máquina de estados.

Neste trabalho, é proposta uma estratégia de modelagem para representar tanto o comportamento livre de falhas quanto o pós-falha para um processo industrial comum da IPG. O diagnóstico de falhas, é utilizado para identificar falhas em um sistema defeituoso através da coleta e análise de dados sobre o estado do sistema, identificando mau funcionamento utilizando medições, testes e outras fontes de informação. Neste sentido, visa-se enquadrar o diagnóstico de falhas entre duas camadas de segurança: i) Intervenção de Operação e ii) SIS. Na primeira, o operador é avisado da existência de alguma irregularidade, que pode ser feito através de um alarme no painel ou por um sistema supervisório, por exemplo. Mesmo que a irregularidade não leve à parada da máquina, ela pode ser prejudicial em algum momento. Já na segunda camada, a ideia é que o diagnosticador atue de maneira a prover informações a um sistema de correção capaz de alterar ativamente o sistema e evitar as consequências danosas da falha. Já que essa é a última camada antes do incidente, o diagnosticador atuaria junto com o SIS como última medida antes da mitigação e contenção de danos. Percebe-se, portanto, que nem sempre é gerada uma situação de parada do sistema, já que o comportamento danoso pode permanecer por um certo tempo antes de ocorrer uma quebra. Por isso, a falha pode ser percebida após uma sequência limitada de eventos. Elas também podem ocorrer de forma imprevisível e relacionada a causas não triviais. Nesse caso, imprevisível no sentido de não saber quando vai ocorrer, pois do ponto de vista da modelagem, a falha diagnosticada é escolhida. Torna-se importante, portanto, o papel do diagnóstico para identificar de forma correta a ocorrência dessas falhas. Através de um diagnosticador de falhas eficiente, é possível evitar situações que geram um funcionamento prejudicial ao sistema, e até impedir situações catastróficas, quando a falha é identificada de forma rápida, com a certeza do acontecimento, com o reconhecimento de múltiplas falhas e classificando-a segundo suas características. Esse trabalho se preocupa com apenas uma falha e com a modelagem pós falha apenas no componente defeituoso. O diagnóstico é importante para evitar que a falha se alastre, poupar a quebra de equipamentos, aumentando sua vida útil, e melhorar a confiança do operador e de quem trabalha no entorno de um processo de alto risco.

A ideia principal apresentada em Sampath *et al.* (1995a, 1996) é modelar o comportamento da planta considerando as falhas como eventos discretos não observáveis, rotular os estados com a informação de ocorrência ou não de falhas e então calcular um observador que permite diagnosticar as falhas pelas sequências de eventos que são diretamente observados. Se o diagnosticador atingir um estado com apenas estados rotulados com falha, a falha é diagnosticada e um sinal é enviado ao operador do sistema. Embora o diagnosticador apresentado em Sampath *et al.* (1995a, 1996) possa ser usado para diagnóstico de falhas, sua construção é, em geral, evitada, pois os observadores podem crescer exponencialmente com o número de estados do

modelo com eventos não observáveis. Vários trabalhos na literatura abordaram o problema de diagnóstico de falhas introduzindo métodos com menor complexidade computacional (CABRAL *et al.*, 2015; BONAFIN; CABRAL; MOREIRA, 2022), métodos para verificar a diagnosticabilidade de SEDs (MOREIRA; JESUS; BASILIO, 2011; MOREIRA; BASILIO; CABRAL, 2015), diferentes arquiteturas de diagnóstico (CABRAL; MOREIRA, 2019; VERAS; CABRAL; MOREIRA, 2021), e outros problemas relacionados, como perdas, atrasos de comunicação e segurança cibernética (NUNES *et al.*, 2018; OLIVEIRA, V. S. L. de; CABRAL; MOREIRA, 2022; LIMA; CARVALHO; MOREIRA, 2023).

1.2.1 Modelagem

Em todos os trabalhos relacionados ao diagnóstico de falhas citados anteriormente, supõe-se que todo o modelo do sistema seja conhecido, incluindo o comportamento pós-falha e a interação pós-falha entre os componentes. Entretanto, modelar o comportamento pós-falha não é uma tarefa fácil, pois o projeto da lógica de controle é realizado considerando apenas o comportamento nominal dos componentes do sistema. Vários trabalhos na literatura visam identificar o modelo de comportamento livre de falhas de um sistema para detecção e isolamento de falhas (KLEIN; LITZ; LESAGE, 2005) (MOREIRA; LESAGE, 2019) (MC MACHADO; VIANA; MOREIRA, 2023). Embora a abordagem de identificação tenha sido aplicada com sucesso para uma variedade de processos de fabricação, o método requer dados de um sistema já em funcionamento até que o cálculo de um modelo livre de falhas tenha convergido. A quantidade de dados e, portanto, o tempo necessário para coletar esses dados, pode ser grande para sistemas com comportamentos complexos e, durante a etapa de coleta de dados, supõe-se que nenhum evento de falha tenha ocorrido. Além disso, o isolamento de um evento de falha não pode ser garantido usando métodos de identificação, uma vez que não há como prever ou identificar o comportamento pós-falha.

A modelagem é uma parte importante do diagnóstico de falhas, portanto, necessita de uma visão de estudo de caso, visão lógica, visão de processo e visão de implementação a fim de aproximar o máximo possível o funcionamento do sistema com o modelo proposto, exigindo afinidade com o sistema que está sendo estudado. Alguns desses sistemas precisam ser de total confiança, dentre eles, os Sistemas Críticos, que demandam atenção redobrada pois causam riscos inerentes a danos físicos, pessoais e financeiros. Especificar uma falha em um sistema com alta criticidade não é um trabalho fácil, e passa a se tornar uma tarefa minuciosa, à medida que grau de periculosidade aumenta.

O modelo de processo abordado na presente pesquisa é parte do sistema a eventos discretos utilizado no projeto de controle supervisório de Rafael G. Oliveira, Queiroz e Cury (2020), sendo necessário, introduzir o evento de falha e o comportamento pós-falha. A principal vantagem da modelagem adotada nesse trabalho é que o evento de falha e seu consequente comportamento local são modelados apenas no componente defeituoso. Assim, o comportamento pós-falha global do sistema é computado pela interação do componente faltoso com outros componentes

do sistema, de acordo com a lógica do projeto de controle supervisório.

Diferente de sistemas contínuos no tempo, a modelagem de sistemas a eventos discretos não utiliza equações diferenciais ordinárias ou equações a diferenças. Sua característica é voltada para a ocorrência de eventos não dependentes do tempo, que pode ser dividida em dois casos: i) modelagem de problemas clássicos, e ii) modelagem de sistemas usualmente feitos com equações diferenciais. No primeiro caso, tem-se como exemplo a chegada de caixas em um armazém ou a presença de uma peça em uma máquina, situações puramente discretas. Já no segundo caso considera-se como exemplo uma dinâmica que já está estabelecida, como um sistema PID que controla o nível em um tanque, e que pode ser modelado como um SED a fim de se impor especificações lógicas ao processo, como intertravamentos de segurança, sequenciamentos de partida e parada e ausência de bloqueio. Nesse caso, o alcance de um determinado nível pode ser utilizado como eventos para realizar o controle de uma válvula. Entre outras palavras: as ações de ligar e desligar pode ser utilizado como eventos para controlar a bomba. As ações de abrir, fechar e regular pode ser utilizado como eventos para controlar a válvula. E, por fim, os eventos do tanque podem ser considerados como a passagem de um nível para outro. Dessa forma, à medida que os sensores detectarem a mudança de nível, os atuadores podem ser ligados ou desligados para assegurar as especificações lógicas de controle.

1.3 TRABALHOS CORRELATOS

Diversos processos da IPG passam por um controle de nível com válvulas. A atuação nesses elementos permite, por exemplo, a estabilização do nível de fluído de dentro de um tanque, e como consequência, a determinação da quantidade exata de líquido. Ela pode ser utilizada para processos que exigem precisão e exatidão nos cálculos. Uma aplicação desse tipo são reservatórios em plataformas de petróleo, que são susceptíveis à explosão caso a variação de volume e dilatação em fator da temperatura não seja equilibrada com um controle rigoroso de gás inerte. Reservatórios também são utilizados em refinarias, e um processo de controle de nível é utilizado para armazenar o combustível extraído proveniente da destilação para decidir em qual momento o abastecimento deve ser encerrado. Ou então, para manter a quantidade de água dentro de um tubulão de uma caldeira, também é necessário determinar o nível e atuar através de uma válvula. As válvulas estão presentes em quase todos os processos da indústria, pois permitem o controle da passagem de fluído através de uma tubulação, e como consequência, agir sobre a vazão sem necessariamente ligar e desligar a bomba hidráulica que realizaria essa tarefa. Tais manobras na bomba, feitas com alta frequência, aumentam sua temperatura, desgastam os componentes internos e danificam o equipamento. A utilização do conjunto válvula e bomba é a mais utilizada para controle de nível e tem bastante representatividade na IPG.

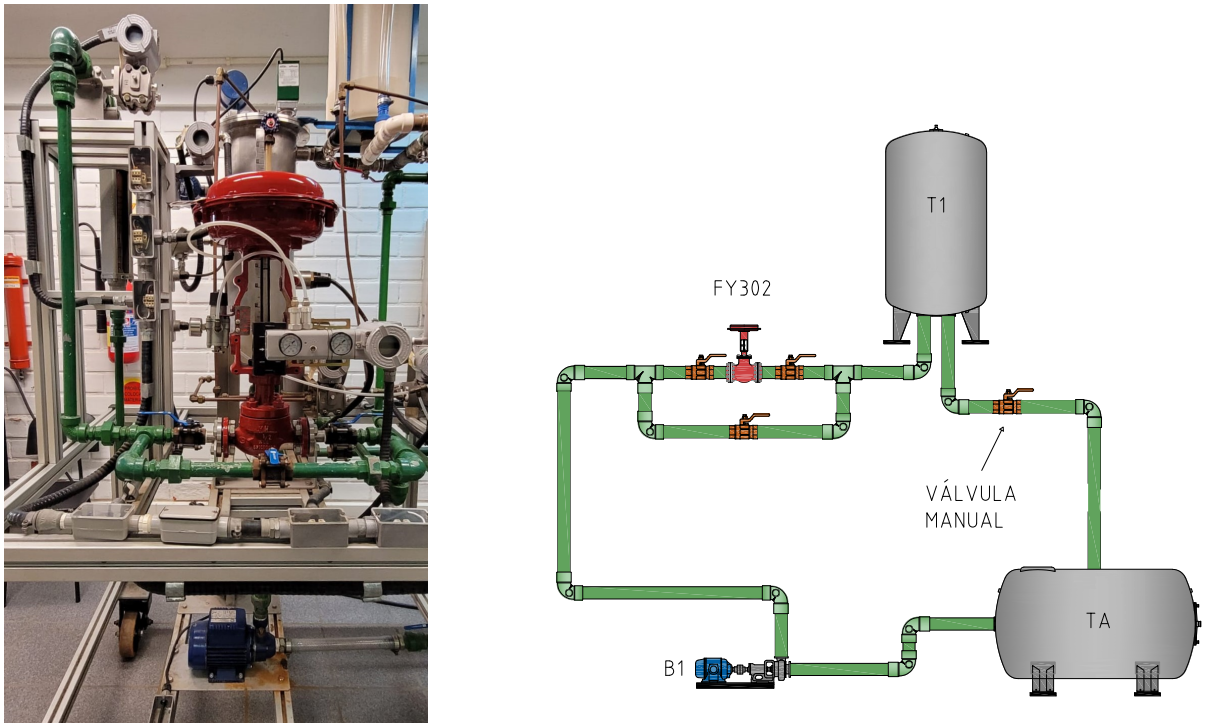
Alguns trabalhos utilizam SEDs em sistemas da IPG. Em Nunes (2012) é estudada uma planta de separação multifásica que utiliza um separador e três hidrociclones, e desenvolve-se um sistema automático de detecção e diagnóstico de falhas baseado na teoria existente de SEDs. Em Martins (2018), um sistema real de plataforma de produção da Petrobrás que opera no

Campo de Marlim na Bacia de Campos é utilizado para o mesmo desenvolvimento de um sistema de diagnóstico de falhas, com o objetivo de ser parte de um sistema inteligente que fornece suporte operacional ao processamento primário de processos de plataforma de petróleo *offshore*. Esses trabalhos não tiveram a aplicação prática do método, e utilizaram softwares de simulação para a comprovação dos resultados. A importância dessas pesquisas é comprovar a eficácia do diagnóstico de falhas utilizando como aplicação teórica os separadores multifásicos, que na IPG tem um papel muito importante na separação do petróleo da água, gás e outras impurezas logo que é extraído. Esse equipamento manipula controle de pressão e nível, ou seja, em situações excessivas de seus comportamentos, poderia ocasionar acidentes envolvendo o entorno do equipamento.

Observa-se que o diagnóstico de falhas é de interesse para a IPG e que os trabalhos encontrados não foram testados na prática. Além disso, é comumente encontrado na indústria sistemas de bomba e válvula para o controle de nível de um processo. Por essa razão, neste trabalho, é utilizado um processo industrial real como estudo de caso e ilustração da modelagem para a estratégia de diagnóstico de falhas proposta nessa dissertação. O processo considerado é constituído da planta didática da SMAR PD3 (Planta Didática 3) de controle de nível, localizada no Laboratório Experimental de Escoamento Multifásico (LEEM) do Departamento de Automação e Sistemas (DAS) da Universidade Federal de Santa Catarina (UFSC). Parte da planta utilizada é apresentada na figura 2(a), com sua respectiva representação gráfica na figura 2(b). A planta apresenta dois conjuntos de bomba, válvula e tanque, mas apenas um conjunto foi suficiente para utilização nesse estudo de controle de nível. Essa planta realiza, entre outros processos, a circulação de líquido por meio de equipamentos industriais semelhante ao que é visto na prática e também integra a rede de campo Foundation Fieldbus (FF) que realiza o controle e gerenciamento dos dispositivos. Além disso, alguns dos sistemas que estão presentes na planta e na IPG são: o controle de fluxo de válvula e de nível de um tanque, aquecimento de fluídos líquidos, controle PID, controle em cascata e leitura de grandezas como vazão, pressão e temperatura.

Entre as vantagens de sua utilização, destaca-se: a planta possui válvula manual *bypass* para representação da quebra da válvula principal e consequentemente pode ser usada para simular a ocorrência da falha. Ela também dispõe de equipamentos vistos na prática mas com a vantagem de estarem em um ambiente acadêmico que permite efetuar diversos testes, e também dispõe dos dispositivos "inteligentes" do protocolo FF que será utilizado para aplicações das arquiteturas de diagnóstico de falhas.

A planta didática da SMAR PD3 é tão relevante para simulações industriais em ambiente acadêmico, que outros trabalhos já a utilizaram no desenvolvimento de suas pesquisas. Pode-se citar o trabalho de Santos, Braga e Saravia (2017) que implementaram duas malhas com controlador PID visando controlar o nível e a vazão do tanque de aquecimento. De forma similar, Gomes, Nicacio e Tôrres (2017) realizaram a modelagem e ajuste dos parâmetros do controlador PI, com resultado satisfatório tanto em termos de resposta rápida quanto em controle



(a) (b)

Figura 2 – Representação da planta utilizada para o controle de nível: (a) Planta instalada no laboratório. (b) Esquema representativo.

de oscilações não desejadas. E o trabalho de Bacovis (2016) realizou uma comparação entre as técnicas de controle PID e *fuzzy*. Na área de SEDs, pode-se citar três referências utilizando os mesmos equipamentos, as quais realizaram a modelagem do sistema. Em Prati (2011) é aplicado o método formal de diagnóstico de falhas através de um exemplo apresentado em Cassandras e Lafortune (2008). Já Simas (2015) realiza a aplicação do controle modular local na planta, fundamentado pela Teoria de Controle Supervisório (TCS) que realiza a síntese e composição de sistemas a fim de garantir uma linguagem controlável minimamente restritiva e não bloqueante, e implementa uma lógica de controle que proporciona segurança para o processo. E o trabalho de Rafael G. Oliveira, Queiroz e Cury (2020), implementou a síntese de supervisores para um controle PID na PD3, explorando a TCS através da síntese minimamente restritiva, não bloqueante e que garante as especificações de segurança.

Observa-se, que nesses últimos trabalhos sobre a planta Planta Didática III (PD3), o diagnóstico de falhas não é implementado atingindo os critérios de diagnosticabilidade que é uma propriedade que permite detectar e localizar a ocorrência de um evento de falha não observável após um número finito de eventos. O diferencial deste trabalho é apresentar uma técnica para modelar o comportamento pós-falha do sistema completo a partir dos modelos obtidos no projeto de controle supervisório, o que possibilita realizar o estudo da diagnosticabilidade e posterior implementação de um diagnosticador. Nesse caso, a falha é modelada apenas no componente que pode sofrer o efeito indesejado (nesse caso uma válvula com travamento), e o comportamento pós-falha completo é obtido pela interação do modelo modificado do componente defeituoso,

com os demais modelos componentes do sistema obtidos do projeto supervisório proposto em Rafael G. Oliveira, Queiroz e Cury (2020).

1.4 OBJETIVOS

Portanto, esta dissertação de mestrado tem entre os objetivos investigar a aplicação de métodos formais de diagnóstico de falhas baseados em sistemas a eventos discretos em processos industriais. Pretende-se avaliar a metodologia proposta por meio de um estudo de caso inspirado em problemas reais da IPG, dirigido à supervisão da PD3, da fabricante SMAR, localizada no LEEM do DAS da UFSC.

O objetivo principal do trabalho pode ser discriminado nos seguintes objetivos específicos:

- identificar problemas característicos de processos industriais para o diagnóstico de falhas;
- modelar e sintetizar diagnosticadores de falha para problemas relevantes de processos industriais;
- propor métodos para implementação eficiente de diagnosticadores considerando a arquitetura de controle e automação típica de processos industriais, incluindo Sistema de Supervisão e Aquisição de Dados (SCADA), Controlador Lógico Programável (CLP) e redes industriais FF;
- modelar a falha traduzindo as consequências diretas do sistema;
- desenvolver e implementar diagnosticadores para uma planta real da IPG;
- modificar as plantas para o diagnóstico de falhas sem alterar o projeto de controle supervisório;
- calcular o modelo completo para verificar diagnosticabilidade do sistema e propor métodos para tornar a planta diagnosticável;
- avaliar experimentalmente os métodos e soluções propostas em estudo de caso.

Este trabalho é organizado como segue. O capítulo 2 apresenta um embasamento acerca da teoria de sistemas a eventos discretos, teoria de diagnóstico de falhas e teoria de controle supervisório. O capítulo 3 apresenta a arquitetura do sistema encontrado, descrição dos modelos construídos e do controle implementado. O capítulo 4 apresenta as modificações realizadas nos modelos para satisfazer a diagnosticabilidade do sistema e realizar o diagnóstico da falha de travamento fechado da válvula. O capítulo 5 apresenta a conclusão e perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SISTEMAS A EVENTOS DISCRETOS

SEDs apresentam duas características distintas: um conjunto discreto como espaço de estados e a transição entre estados é dirigida por eventos. Os sistemas modelados a eventos nem sempre são mais fáceis de modelar e analisar, uma vez que existem vários mecanismos assíncronos de temporização de eventos a serem especificados como parte da compreensão do sistema. Entretanto, considerar sistemas como SEDs traz uma série de vantagens: eles são intuitivos, fáceis de usar, passíveis de operações de composição e também de análise (no caso de estado finito). Se por um lado, podem levar a espaços de estados muito grandes ao modelar sistemas complexos, por outro, apresentam integração com outras técnicas de controle e observação para facilitar sua compreensão.

O fato do estado do sistema ser discreto implica que ele pode assumir valores simbólicos, como por exemplo {ligado, desligado}, {azul, verde, preto}, ou valores discretos tais como valores numéricos pertencentes aos conjuntos $\mathbb{N}$ ou $\mathbb{Z}$, ou ser formado por um subconjunto enumerável de elementos de $\mathbb{R}$. Eventos podem estar associados a ações específicas (por exemplo, o apertar de um botão, uma pessoa passar pela catraca do ônibus, uma peça atinge um determinado ponto de uma linha de produção, o líquido dentro de um tanque atinge uma determinada altura, etc.). Boa parte dos sistemas dinâmicos complexos podem ser vistos como SEDs em algum nível de abstração. Embora seja possível modelar qualquer sistema físico como um SED, determinados sistemas são naturalmente discretos e com evolução determinada pela ocorrência de eventos.

2.2 LINGUAGEM

A linguagem é um dos conceitos mais fundamentais em SEDs. Ela fornece uma estrutura para representar os sistemas com técnicas analíticas para aplicação de métodos, controle e avaliação de desempenho. Para a definição de linguagem, é preciso antes introduzir os conceitos de *alfabeto* e de *palavra*.

Definição 1 (Alfabeto) *Um alfabeto, denotado por Σ, é um conjunto finito. Os elementos de um alfabeto são usualmente denominados de símbolos ou caracteres, mas no âmbito de SED são denominados de eventos.*

O conjunto vazio $\varnothing$ é um alfabeto, e um conjunto infinito não é um alfabeto.

Definição 2 (Palavra) *Uma palavra sobre um alfabeto é uma sequência finita de eventos do alfabeto. Nessa dissertação, será utilizado o termo sequência para designar palavra.*

Uma sequência de comprimento zero, ou seja, sem símbolos, é uma sequência válida sendo denotada por ε.

A definição formal de linguagem é dada a seguir.

Definição 3 (Linguagem) *Uma linguagem formal, ou simplesmente linguagem, é um conjunto de sequências formadas a partir dos eventos de* Σ.

2.2.1 Operações com linguagens

A seguir são definidas formalmente as operações sobre linguagens (CASSANDRAS; LAFORTUNE, 2008).

Definição 4 (Concatenação) *Sejam* L_a, $L_b \subseteq \Sigma^*$, *então a concatenação* $L_a L_b$ *é definida como:*

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

Uma sequência s está em $L_a L_b$ se ela pode ser escrita como uma concatenação de uma sequência em L_a com outra sequência em L_b. Ou seja, a operação de concatenação reúne cada sequência de uma linguagem com cada sequência de outra linguagem.

Σ^* denota o conjunto de todas as sequências possíveis sobre Σ.

Definição 5 (Fecho de Kleene) *Seja* $L \in \Sigma^*$, *então o fecho de Kleene* L^*, *é definido como:*

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

O conjunto de todas as sequências finitas que podem ser formadas com os elementos de L é denominada L^* e inclui a sequência vazia ε. Observa-se que a operação fecho de Kleene é idempotente, ou seja, $(L^*)^* = L^*$.

Ainda existem outras operações que podem ser definidas sobre linguagens, que são apresentadas a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 6 (Fecho do Prefixo ou Prefixo fechamento) *Seja* $L \subseteq \Sigma^*$, *então:*

$$\overline{L} = \left\{s \in \Sigma^* : \left(\exists t \in \Sigma^*\right) [st \in L]\right\}$$

O fecho do prefixo de L é a linguagem que consiste em todos os prefixos de todas as sequências em L, e é denotada por $\overline{L}$. Além disso, L é dita ser prefixo-fechada quando $L = \overline{L}$, ou seja, quando todos os prefixos de todas as linguagens de L também são elementos de L.

Definição 7 (Pós linguagem) *Seja* $L \subseteq \Sigma^*$ *e* $s \in L$. *A pós linguagem de L após s, denotada por* L/s, *é definida como:*

$$L/s = \left\{t \in \Sigma^* : st \in L\right\}$$

Por definição, $L/s = \varnothing$ *se* $s \notin \overline{L}$.

Exemplo 2.2.1 *Seja* $\Sigma = \{f,g,h\}$ *e considere as linguagens* $L_1 = \{\varepsilon,f,fg,ffg\}$ *e* $L_2 = \{h\}$. *Note que* L_1 *e* L_2 *não são prefixo-fechadas, já que* $ff \notin L_1$ *e* $\varepsilon \notin L_2$. *Logo, tem-se que,* $L_1L_2 = \{h,fh,fgh,ffgh\}$, $\overline{L}_1 = \{\varepsilon, f, fg, ff, ffg\}$, $\overline{L}_2 = \{\varepsilon,h\}$, $L_1^* = \{\varepsilon, f, fg, ffg, ff, ffffg, fgf, ffgf,...\}$ *e* $L_2^* = \{\varepsilon,h,hh,hhh,...\}$.

Observação 2.2.1 *Para uma linguagem* $L = \varnothing$, $\overline{L} = \varnothing$, *mas se* $L \neq \varnothing$, *então necessariamente* $\varepsilon \in \overline{L}$. *Além disso,* $\varnothing^* = \varepsilon$ *e* $\{\varepsilon\}^* = \{\varepsilon\}$ *e a concatenação envolvendo uma linguagem e o conjunto vazio tem como resultado o próprio conjunto vazio, ou seja,* $\varnothing L = L\varnothing = \varnothing$.

Outra operação importante é a *projeção* de sequências e linguagens, também chamada de *projeção natural*. As operações de projeção e projeção inversa são definidas a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 8 (Projeção) *A projeção* $P_s : \Sigma_l^* \rightarrow \Sigma_s^*$, *em que* $\Sigma_s \subset \Sigma_l$, *é definida recursivamente da seguinte forma:*

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

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

$$P_s(s\sigma) = P(s)P(\sigma), para\ todo\ s \in \Sigma_l^*, \sigma \in \Sigma_l$$

em que $\setminus$ *denota diferença entre conjuntos.*

Seja Σ_l e Σ_s tal que $\Sigma_s \subset \Sigma_l$, aplicar a projeção em uma sequência s apaga eventos de Σ_l que não pertencem a Σ_s. Ou seja, a operação de projeção apaga todos os eventos $\sigma \in \Sigma_l \setminus \Sigma_s$ das sequências de $s \in \Sigma_l^*$. Por exemplo, considere os conjuntos $\Sigma_l = \{a,b,c\}$ e $\Sigma_s = \{b\}$ e as sequências de eventos $s_1 = ac$ e $s_2 = acb$, em que, $s_1, s_2 \in \Sigma_l^*$. Logo, a projeção $P_s : \Sigma_l^* \rightarrow \Sigma_s^*$ de s_1 e s_2 são iguais a $P_s(s_1) = \varepsilon$ e $P_s(s_2) = b$.

Definição 9 (Projeção inversa) *A projeção inversa* $P_s^{-1} : \Sigma_s^* \rightarrow 2^{\Sigma_l^*}$ *é definida como:*

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

Dada uma sequência s construída a partir do conjunto de eventos Σ_s, a operação de projeção inversa P_s^{-1} realizada sobre s gera como resultado, um conjunto formado por todas as sequências definidas em Σ_l, cuja projeção P_s resulta em s.

Ambas as operações de projeção P_s e projeção inversa P_s^{-1} podem ser aplicadas para linguagens. Para isso, basta aplicá-las a todas as sequências que formam a linguagem. As operações de projeção e projeção inversa estendidas a linguagens são definidas a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 10 (Projeção de linguagens) *Seja $L \subseteq \Sigma_l^*$, então, a projeção de L, $P_s(L)$, é definida como:*

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

Definição 11 (Projeção inversa de linguagens) *Seja $L_s \subseteq \Sigma_s^*$, então a projeção inversa de L_s, $P_s^{-1}(L_s)$, é definida como:*

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

Uma das aplicações da projeção é representar a linguagem observada de um sistema por um observador que apenas possui acesso a eventos registrados por sensores ou eventos de comando de um controlador. Assim, dada uma linguagem gerada por um sistema, é possível obter a linguagem observada realizando-se a projeção da linguagem gerada no conjunto de eventos observáveis.

2.3 AUTÔMATOS

Um dos formalismos utilizados para representar linguagens geradas por SEDs são os autômatos. A seguir é apresentada a definição formal de um autômato (CASSANDRAS; LAFORTUNE, 2008).

Definição 12 (Autômato determinístico) *Um autômato determinístico, denotado por G, é uma quíntupla:*

$$G = (Q, \Sigma, f, q_0, Q_m)$$

em que Q é o conjunto de estados, Σ é o conjunto de eventos, $f : Q \times \Sigma \rightarrow Q$ é a função de transição que pode ser parcial no seu domínio, q_0 é o estado inicial e $Q_m \subseteq Q$ é o conjunto de estados marcados.

Uma vez que a função de transição só determina um único estado evoluído a partir de um mesmo evento, ele é chamado de determinístico. A fim de completar a definição do autômato, podemos ainda definir a função de eventos ativos $\Gamma_G : Q \rightarrow 2^{\Sigma}$, com notação $\Gamma(q) = \{\sigma : \sigma \in \Sigma$ e $f(q,\sigma)!\}$.

A representação gráfica de um autômato pode ser feita através de um grafo orientado, chamado de diagrama de transição de estados. Os vértices do grafo, representados por círculos, são os estados e as arestas, representadas por arcos, são as transições entre os estados. As transições são rotuladas pelos eventos em Σ responsáveis pela transição de estados.

O estado inicial, que representa a condição inicial no diagrama de transição, possui uma seta sem origem. E os estados marcados, que representa estados objetivos, são representados

por círculos duplos concêntricos. Enquanto as arestas representam graficamente a função de transição do autômato.

Um exemplo de um autômato e seu diagrama de transição de estados é apresentado a seguir.

Exemplo 2.3.1 *Seja G um autômato cujo diagrama de estados pode ser visto na figura 3. O conjunto de estados de G é dado por $Q = \{S0,S1,S2\}$ e o conjunto de eventos é dado por $\Sigma = \{a,b,c\}$. A função de transição de estados de G é definida da seguinte forma: $f(S0,c) = S0$; $f(S0,a) = f(S0,b) = S1$; $f(S1,c) = S1$; $f(S1,b) = S2$; $f(S2,b) = S2$; $f(S2,a) = S0$. Note que $f(S1,a)$ e $f(S2,c)$ não são definidas. Assim, a função de eventos ativos em cada estado possui os seguintes resultados: $\Gamma(S0) = \{a,b,c\}$; $\Gamma(S1) = \{b,c\}$; $\Gamma(S2) = \{a,b\}$. Por fim, o estado inicial de G é $q_0 = S0$ e o conjunto de estados marcados é $Q_m = \{S1,S2\}$.*

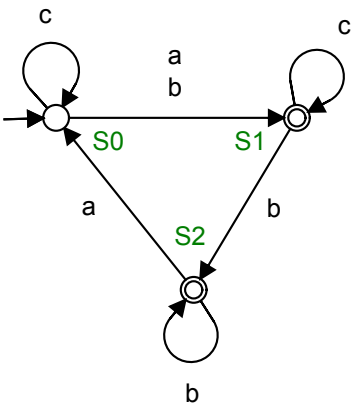


Figura 3 – Diagrama de transição do autômato G.

Na sequência são definidas as linguagens geradas e marcadas por um autômato.

Definição 13 (Linguagem gerada) *A linguagem gerada por um autômato $G = (Q, \Sigma, f, q_0, Q_m)$ é dada por:*

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

Definição 14 (Linguagem marcada) *A linguagem marcada por um autômato $G = (Q,\Sigma,f,q_0, Q_m)$ é dada por:*

$$L_m(G) = \{s \in L(G) : f(q_0,s) \in Q_m\}$$

Nas definições acima é suposto que a função de transição f é estendida, ou seja, $f : Q \times \Sigma^* \rightarrow Q$. Além disso, para qualquer G que possua um conjunto de estados Q não vazio, $\varepsilon \in L(G)$.

Todas as sequências partindo do estado inicial que podem ser executadas no diagrama de transição de estados, compõem a linguagem gerada por G, $L(G)$. Ou seja, a concatenação

dos eventos das transições que compõem um caminho corresponde à sequência de eventos do sistema. Portanto, é importante observar que $L(G)$ é prefixo-fechada por definição, uma vez que um caminho só é possível se todos os seus correspondentes prefixos são também possíveis.

A linguagem marcada $L_m(G)$, é um subconjunto de $L(G)$, que corresponde a todas as sequências s tais que $f(q_0,s) \in Q_m$, ou seja, todas as sequências que levam a um estado marcado no diagrama de transição de estados. É importante observar que a linguagem marcada por G, $L_m(G)$, não é necessariamente prefixo-fechada, já que nem todos os estados de Q precisam ser marcados.

2.3.1 Operações com autômatos

As análises realizadas em SEDs são possíveis graças a modificações do diagrama de transição de estados de acordo com alguma operação correspondente da linguagem gerada. Por isso, é necessário definir operações que permitam combinar autômatos, para construir modelos de sistemas complexos a partir de modelos de componentes do sistema.

A parte acessível de um autômato G é uma operação unária que visa eliminar todos os estados de G que não são alcançáveis a partir do estado inicial q_0 e suas transições relacionadas. A definição formal de parte acessível de um autômato é apresentada a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 15 (Parte acessível) *Seja $G = (Q,\Sigma, f, q_0, Q_m)$. A parte acessível de G, denotada por $Ac(G)$, é definida como:*

$$Ac(G) = (Q_{ac},\Sigma, f_{ac}, q_0, Q_{ac,m})$$

em que $Q_{ac} = \{q \in Q : (\exists s \in \Sigma^*)[f(q_0,s) = q]\}$, $Q_{ac,m} = Q_m \cap Q_{ac}$ e $f_{ac} : Q_{ac} \times \Sigma^* \to Q_{ac}$.

Nesse caso, Q_{ac} contém todos os estados que são possíveis de serem alcançados a partir do estado inicial. Ao tomar a parte acessível de um autômato, o domínio da função de transição se restringe a um domínio menor dos estados acessíveis Q_{ac}. Além disso, a parte acessível não altera as linguagens $L(G)$ e $L_m(G)$.

A seguir é apresentada a definição formal de parte coacessível de um autômato (CASSANDRAS; LAFORTUNE, 2008).

Definição 16 (Parte coacessível) *Seja $G = (Q,\Sigma,f,q_0,Q_m)$, a parte coacessível de G, denotada por $CoAc(G)$, é definida como:*

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

em que $Q_{coac} = \{q \in Q : (\exists s \in \Sigma^*)[f(q,s) \in Q_m]\}$, $q_{0,coac}=q_0$, se $q_0 \in Q_{coac}$ e $q_{0,coac}$ é indefinido, se $q_0 \notin Q_{coac}$, e $f_{coac} : Q_{coac} \times \Sigma^* \to Q_{coac}$.

Um estado $q \in Q$ é dito ser coacessível se existir um caminho a partir do estado q que leve a um estado marcado, ou seja, um estado que pertença a Q_m. Portanto, ao tomar a parte coacessível de um autômato, apaga-se todos os estados em G, e suas transições correspondentes, que a partir dos quais não é possível alcançar um estado marcado.

É importante notar que, assim como na operação de tomar a parte acessível de um autômato, tomar a parte coacessível restringe o domínio da função de transição para os estados coacessíveis Q_{coac}. Note que $L(CoAc(G)) \subseteq L(G)$, contudo $L_m(CoAc(G)) = L_m(G)$.

Um autômato que é tanto acessível quanto coacessível é chamado de Trim. A definição formal da operação trim é apresentada a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 17 (Operação trim) *Seja $G = (Q,\Sigma, f, q_0, Q_m)$, a operação Trim pode ser definida da seguinte forma:*

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

A seguir é apresentado um exemplo que ilustra o resultado das operações de tomar a parte acessível, a parte coacessível e trim de um autômato G.

Exemplo 2.3.2 *Considere o autômato G mostrado na figura 4. As figuras 5(a), 5(b) e 5(c) mostram os autômatos resultantes após as operações de parte acessível, coacessível e trim, respectivamente.*

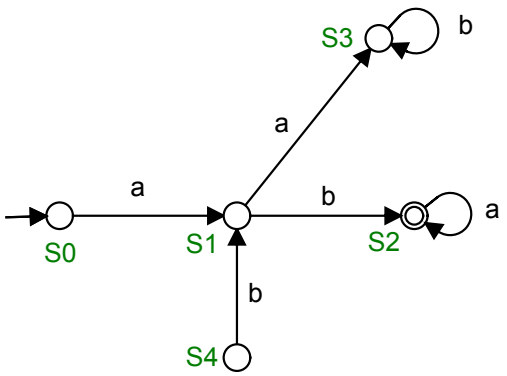


Figura 4 – Autômato G do exemplo 2.3.2.

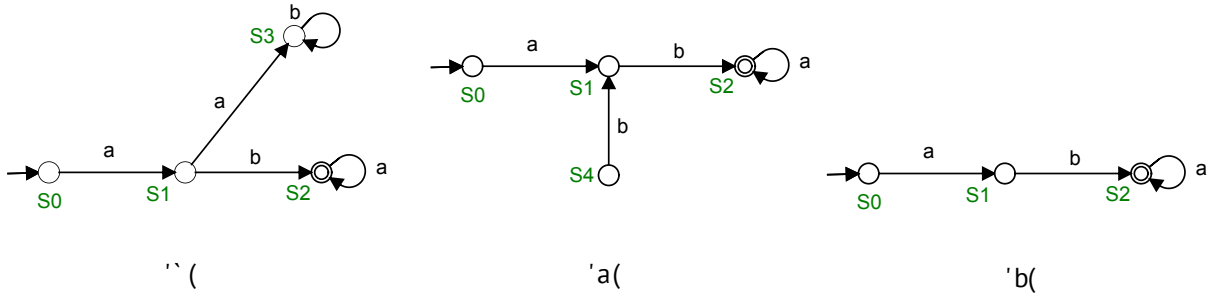


Figura 5 – $Ac(G)$ do exemplo 2.3.2(a), $CoAc(G)$ do exemplo 2.3.2(b) e $Trim(G)$ do exemplo 2.3.2(c).

As operações unárias apresentadas até aqui, como o nome sugere, envolvem apenas um autômato. Mas também existem operações que envolvem dois ou mais autômatos, como: a composição produto e a composição paralela, apresentadas a seguir.

Definição 18 (Composição produto) *Sejam os autômatos* $G_1 = (Q_1,\Sigma_1,f_1,q_{01},Q_{m1})$ *e* $G_2 = (Q_2,\Sigma_2,f_2,q_{02},Q_{m2})$*, então, a composição produto entre* G_1 *e* G_2*,* $G_1 \times G_2$*, é dada por:*

$$G_1 \times G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, f_{1\times 2}, (q_{01},q_{02}), Q_{m1} \times Q_{m2})$$

em que:

$$f_{1\times 2}((q_1,q_2),\sigma) = \begin{cases} (f_1(q_1,\sigma), f_2(q_2,\sigma)), \textit{ se } \sigma \in \Gamma_{G_1}(q_1) \cap \Gamma_{G_2}(q_2) \\ \textit{indefinido, caso contrário} \end{cases}$$

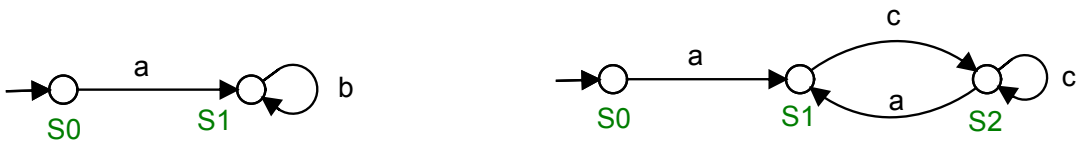
Segundo a definição de composição produto, as transições dos dois autômatos devem ser sempre sincronizadas com um evento em comum, ou seja, um evento que pertença a $\Sigma_1 \cap \Sigma_2$. Por isso, um evento ocorre em $G_1 \times G_2$, se e somente se, o evento ocorrer em G_1 e G_2 ao mesmo tempo.

Os estados do autômato resultante de $G_1 \times G_2$ são denotados em pares, sendo o primeiro componente um estado de G_1 e o segundo componente um estado de G_2. Além disso, como a composição produto sincroniza a evolução dos autômatos, a linguagem gerada e a linguagem marcada por $G_1 \times G_2$ são $L(G_1 \times G_2) = L(G_1) \cap L(G_2)$ e $L_m(G_1 \times G_2) = L_m(G_1) \cap L_m(G_2)$, respectivamente.

A seguir é apresentada a definição formal de composição paralela (CASSANDRAS; LAFORTUNE, 2008).

Definição 19 (Composição paralela) *Sejam os autômatos* $G_1 = (Q_1,\Sigma_1,f_1,q_{01},Q_{m1})$ *e* $G_2 = (Q_2,\Sigma_2,f_2,q_{02},Q_{m2})$*. A composição paralela entre* G_1 *e* G_2*, denotada por* $G_1 \,||\, G_2$*, é dada por:*

$$G_1 \,||\, G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, f_{1||2}, (q_{01},q_{02}), Q_{m1} \times Q_{m2})$$



(a) Autômato G_1. (b) Autômato G_2.

Figura 6 – Autômatos G_1 e G_2 do exemplo 2.3.3.

em que

$$f_{1||2}((q_1,q_2),\sigma) = \begin{cases} (f_1(q_1,\sigma), f_2(q_2,\sigma)), \textit{ se } \sigma \in \Gamma_{G_1}(q_1) \cap \Gamma_{G_2}(q_2) \\ (f_1(q_1,\sigma),q_2), \textit{ se } \sigma \in \Gamma_{G_1}(q_1) \setminus \Sigma_2 \\ (q_1,f_2(q_2,\sigma)), \textit{ se } \sigma \in \Gamma_{G_2}(q_2) \setminus \Sigma_1 \\ \textit{indefinido, caso contrário} \end{cases}$$

A composição paralela, também chamada de composição síncrona, é representada por $||$. Diferente da composição produto, que permite apenas transições rotuladas por eventos comuns, a composição paralela permite transições rotuladas por eventos particulares e sincroniza transições rotuladas por eventos comuns, ou seja, eventos que pertencem a $(\Sigma_1 \setminus \Sigma_2) \cup (\Sigma_2 \setminus \Sigma_1)$ podem ocorrer sempre que forem possíveis. Ela é comumente utilizada, pois se trata da maneira mais comum de se obter o modelo de um sistema complexo, a partir dos modelos de seus componentes, Vale observar, que se $\Sigma_1 = \Sigma_2$, então o resultado da composição paralela é igual ao resultado da composição produto, uma vez que todas as transições serão sincronizadas.

Para caracterizar corretamente as linguagens gerada e marcada pelo autômato resultante da composição paralela é preciso definir as seguintes projeções:

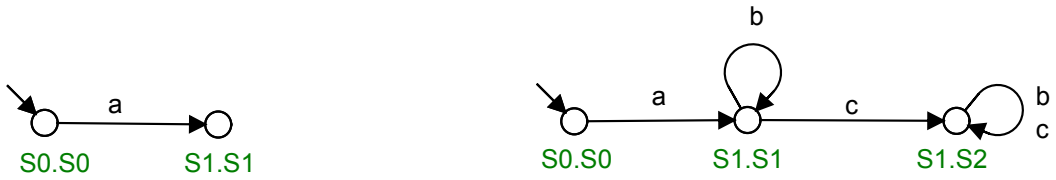
$$P_i : (\Sigma_1 \cup \Sigma_2)^* \rightarrow \Sigma_i^* \text{ para } i = 1,2$$

Com base nessas projeções, as linguagens gerada e marcada resultantes da composição paralela são $L(G_1 \mid\mid G_2) = P_1^{-1}[L(G_1)] \cap P_2^{-1}[L(G_2)]$ e $L_m(G_1 \mid\mid G_2) = P_1^{-1}[L_m(G_1)] \cap P_2^{-1}[L_m(G_2)]$, respectivamente. As operações de composição paralela e produto são ilustradas no exemplo a seguir.

Exemplo 2.3.3 *Considere os autômatos G_1, $\Sigma_1 = \{a,b\}$ e G_2, $\Sigma_2 = \{a,c\}$ mostrados nas figuras 6(a) e (b), respectivamente. As figuras 7(a) e (b) mostram o resultado da composição produto e da composição paralela entre G_1 e G_2, respectivamente.*

2.3.2 Autômatos com observação parcial de eventos

Até aqui, considerou-se que todos os eventos de Σ eram observáveis e conhecidos de alguma forma. Entretanto, um SED pode ser composto por eventos que não podem ser vistos



(a) $G_1 \times G_2$. (b) $G_1 \parallel G_2$.

Figura 7 – Resultados da composição produto e paralela do exemplo 2.3.3.

por observadores recebendo o nome de Sistema a Eventos Discreto Parcialmente Observável. Eventos não observáveis podem ocorrer no sistema, mas não são vistos ou observados, devido à ausência de sensores que registrem a ocorrência do evento, ou por eventos que representem falhas. Também podem originar de um evento que ocorreu em uma localização remota e sua ocorrência não foi comunicada.

Dessa forma, o conjunto de eventos de G pode ser particionado em $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_o denota o conjunto de eventos observáveis e Σ_{uo} denota o conjunto de eventos não observáveis. Para um sistema G com eventos não observáveis, a linguagem gerada observada de G é obtida aplicando-se a projeção $P_o(L(G))$, em que $P_o : \Sigma^* \rightarrow \Sigma_o^*$. Além disso, com o conjunto de eventos particionado entre eventos observáveis e não observáveis, é necessário uma estrutura que identifique os possíveis estados do sistema após a observação de uma sequência de eventos. Essa estrutura é chamada de observador de G e é denotada por $Obs(G)$. Antes de apresentar o algoritmo de construção de $Obs(G)$, é necessário apresentar a seguinte definição de alcance não observável de um estado, denotado por $UR(q)$.

Definição 20 (Alcance não observável) *O alcance não observável de um estado $q \in Q$, denotado por $UR(q)$, é definido como:*

$$UR(q) = \{y \in Q : (\exists t \in \Sigma_{uo}^*)(f(q,t) = y)\}$$

O alcance não observável pode ser definido para um conjunto de estados $B \in 2^Q$ da seguinte forma:

$$UR(B) = \bigcup_{q \in B} UR(q)$$

Definição 21 (Observador) *O observador de um autômato G em relação ao conjunto de eventos observáveis Σ_o, denotado por $Obs(G)$, é dado por:*

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

em que $Q_{obs} \subseteq 2^Q$ e $Q_{m,obs} = \{B \in Q_{obs} : B \cap Q_m \neq \varnothing\}$. Os conjuntos f_{obs}, e $q_{0,obs}$ são obtidos de acordo com o algoritmo 2.3.1 (CASSANDRAS; LAFORTUNE, 2008).

O Algoritmo 2.3.1 mostra os passos para a construção do observador apresentado na definição 20.

Algoritmo 2.3.1 *Seja $G = (Q,\Sigma, f, q_0, Q_m)$ um autômato determinístico, sendo $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$. Então, $Obs(G) = (Q_{obs}, \Sigma_o, f_{obs}, q_{0,obs}, Q_{m,obs})$ é construído da seguinte forma:*

- *Passo 1: Defina $q_{0,obs} = UR(q_0)$. Faça $Q_{obs} = \{q_{0,obs}\}$ e $\widetilde{Q}_{obs} = Q_{obs}$.*
- *Passo 2: $\overline{Q}_{obs} = \widetilde{Q}_{obs}$ e $\widetilde{Q}_{obs} = \varnothing$.*
- *Passo 3: Para cada $B \in \overline{Q}_{obs}$,*
  - *Passo 3.1: $\Gamma_{obs}(B) = (\underset{q \in B}{\cup} \Gamma(q)) \cap \Sigma_o$.*
  - *Passo 3.2: Para cada $\sigma \in \Gamma_{obs}(B)$,*

$$f_{obs}(B,\sigma) = UR(\{q \in Q : (\exists y \in B)[q = f(y,\sigma)]\}).$$

  - *Passo 3.3: $\widetilde{Q}_{obs} \leftarrow \widetilde{Q}_{obs} \cup f_{obs}(B,\sigma)$.*
- *Passo 4: $Q_{obs} \leftarrow Q_{obs} \cup \widetilde{Q}_{obs}$.*
- *Passo 5: Repita os passos 2 a 4 até que toda a parte acessível de $Obs(G)$ tenha sido construída.*
- *Passo 6: $Q_{m,obs} = \{B \in Q_{obs} : B \cap Q_m \neq \varnothing\}$.*

As linguagens gerada e marcada pelo autômato $Obs(G)$ são: $L(Obs(G)) = P_o(L(G))$ e $L_m(Obs(G)) = P_o(L_m(G))$, sendo $P_o : \Sigma^* \rightarrow \Sigma_o^*$.

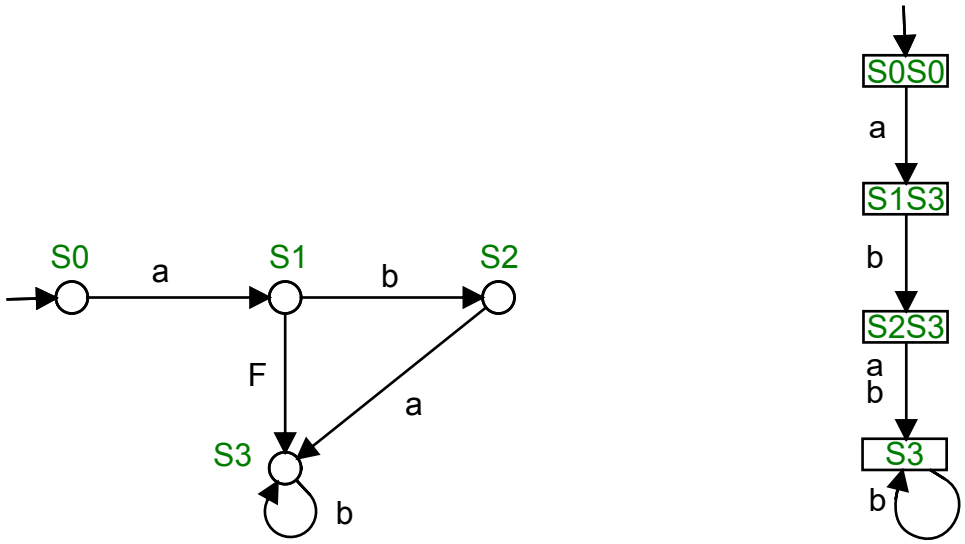
Exemplo 2.3.4 *Seja G o autômato cujo diagrama de transição de estados pode ser visto na figura 8(a). O conjunto de estados de G é $Q = \{S0,S1,S2,S3\}$ e o conjunto de eventos é $\Sigma = \{a,b,F\}$, em que $\Sigma_o = \{a,b\}$ e $\Sigma_{uo} = \{F\}$. O observador de G,$Obs(G)$, pode ser visualizado na figura 8(b). Considere que o sistema tenha executado a sequência de eventos $t = aFb$, a sequência observada será $P_o(t) = ab$, para $P_o : \Sigma^* \rightarrow \Sigma_o^*$. Ao acompanhar a sequência $P_o(t)$ em $Obs(G)$, o estado $\{S2,S3\}$ é alcançado, que corresponde à estimativa de estado de G após a observação dessa sequência.*

O observador fornece uma estimativa dos estados alcançados de G após a observação de uma sequência de eventos gerada pelo sistema, como é notado no exemplo 2.3.4. Na sequência, a Diagnosticabilidade de SEDs é apresentada.

2.3.2.1 Diagnosticabilidade de SEDs

A diagnosticabilidade é a propriedade que permite detectar e localizar a ocorrência de um evento de falha não observável após um número finito de eventos gerados após a falha. Seja G o autômato que modela um sistema e seja $\Sigma_f \subseteq \Sigma_{uo}$ o conjunto de eventos de falha, que corresponde a todas as falhas que podem ocorrer na planta que está sendo considerada.

Para definir a diagnosticabilidade de um sistema é preciso antes definir seu comportamento normal com relação à falha F, como apresentado a seguir.



(a) (b)

Figura 8 – Autômato G com o evento não observável F (a), e autômato observador de G, $Obs(G)$ (b).

Definição 22 *Seja $L(G) = L$ a linguagem gerada pelo autômato G e seja L_N a linguagem prefixo-fechada formada por todas as sequências de L que não contém nenhum evento de falha do conjunto Σ_f. Assim, o comportamento normal do sistema G, em relação à falha do tipo F, é modelado pelo autômato G_N que gera a linguagem L_N. Uma forma de obter G_N é realizando a composição síncrona da planta G com A_N, sendo esse último, um autômato de estado único "N"e com um auto-laço de $\Sigma \setminus \{\Sigma_f\}$ e indica que a ocorrência de todos os eventos menos os de falha. Então G_n representa o autômato G livre de falhas.*

A definição de diagnosticabilidade é apresentada a seguir.

Definição 23 (Diagnosticabilidade) *Sejam L e $L_N \subset L$ as linguagens prefixo-fechadas geradas por G e G_N, respectivamente, e defina a operação de projeção $P_o : \Sigma^* \rightarrow \Sigma_o^*$. Então, L é dita ser diagnosticável com relação à projeção P_o e com relação ao conjunto de eventos de falha Σ_f se*

$$(\exists n \in \mathbb{N})(\forall_s \in L \setminus L_N)(\forall_{st} \in L \setminus L_N)$$
$$(|t| \geq n \Rightarrow \forall \omega \in P_o^{-1}(P_o(st)) \cap L, \omega \in L \setminus L_N)$$

De acordo com a definição 23, L é diagnosticável com relação a P_o e Σ_f se e somente se para todas as sequências st de comprimento arbitrariamente longo após a ocorrência de um evento de falha do conjunto Σ_f, não existirem sequências $s_N \in L_N$, de tal forma que $P_o(s_N) = P_o(st)$. Portanto, se L é diagnosticável então sempre é possível identificar unicamente se a falha ocorreu após um número finito de observações de eventos.

Para se implementar um diagnosticador online, o primeiro passo é verificar a diagnosticabilidade do sistema. Isso significa verificar se após um número finito de observações de eventos após a ocorrência da falha, é possível afirmar que ela ocorreu.

2.3.2.2 Autômato diagnosticador

Para determinar se algum evento de falha possa ter ocorrido em uma sequência qualquer do sistema, realiza-se a diagnose de falhas. O evento de falha, considerado como um evento não observável, modela alguma forma o não determinismo no modelo, gerado por incerteza de qual acontecimento possa ter ocorrido. O conhecimento que qual evento ocorreu, com o auxilio da projeção, é muito importante para monitorar o desempenho do sistema. Portanto, a incerteza é reduzida ao realizar observações do comportamento do sistema, mais precisamente, sobre o prefixo da sequências de eventos que ocorreram.

O diagnosticador é obtido através de adaptações feitas no observador e pode ser construído de forma automática através de procedimento apresentado adiante. O seu objetivo é inferir de forma explícita a ocorrência de falhas a partir da observação de sequências de eventos gerados pelo sistema (SAMPATH *et al.*, 1995b) (SAMPATH *et al.*, 1996). Esse trabalho não tem o objetivo de realizar o diagnóstico com melhor custo computacional mas sim utilizar das ferramentas computacionais que lidam com autômatos. Apesar de existir uma série de diagnosticadores mais eficientes, optou-se por utilizar o diagnosticador apresentado em Sampath *et al.* (1995b), para uma compreensão mais simplificada, e dedicação temporal no diagnóstico do problema proposto.

O autômato diagnosticador $Diag(G)$ se diferencia do observador $Obs(G)$ por adicionar rótulos aos estados de G. Caso um evento de falha $f \in \Sigma_{uo}$ ainda não tenha ocorrido, o rótulo adicionado ao estado da planta é N mas caso tenha ocorrido, é adicionado Y. O rótulo é anexado ao estado $q \in Q$ de G no formato qN e qY, ou (q,N) e (q,Y). Uma mesma planta pode ter mais de um evento não-observável, de modo que, podem ser construídos diagnosticadores para cada falha a ser diagnosticada ou então um único que rastreia todos os eventos como em sistemas de diagnose com arquitetura centralizada.

Duas etapas são necessárias para o cálculo do Diagnosticador $Diag(G)$. Primeiramente é calculada a composição paralela $G_l = G \parallel A_l$, em que $A_l = (\{N,Y\}, \Sigma_f, f_l, N)$ é um autômato rotulador de dois estados mostrado na figura 9 (nesse caso não é necessário considerar estados marcados). Por fim é computado o observador $Obs(G_l)$ da planta. O Exemplo 2.3.5 ilustra a construção de G_{diag} para uma planta G.

Exemplo 2.3.5 *Considere o autômato G, com* $\Sigma_o = \{a,b,c\}$, $\Sigma_{uo} = \{u,f\}$ *e* $\Sigma_f = \{f\}$, *mostrado na figura 10. O primeiro passo para a construção de* G_{diag} *é a composição paralela entre o autômato G e o autômato rotulador mostrado na figura 9. O autômato* G_l, *resultante da composição* $G \parallel A_l$, *pode ser visto na figura 11. O último passo para a construção de* G_{diag} *é o cálculo do observador de* $G \parallel A_l$, $Obs(G \parallel A_l)$, *que é mostrado na figura 12.*

A falha é diagnosticada quando o diagnosticador alcança estados que possuem apenas

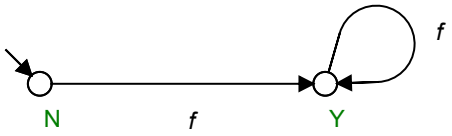


Figura 9 – Autômato rotulador A_l.

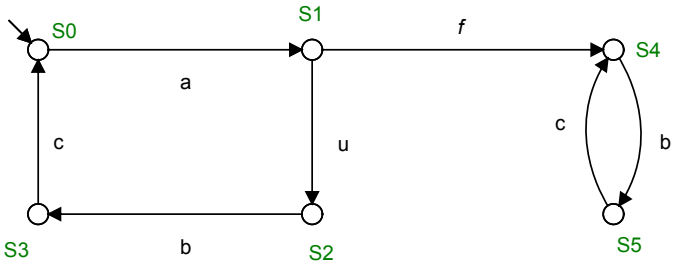


Figura 10 – Autômato G do exemplo 2.3.5.

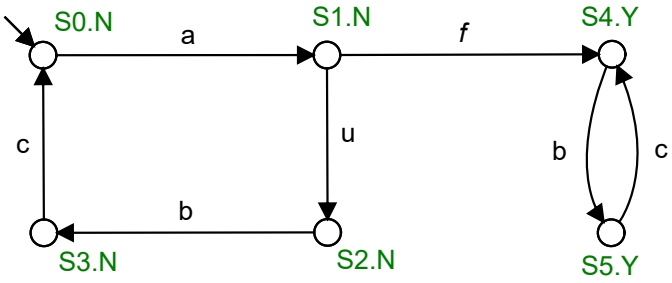


Figura 11 – Autômato G_l do exemplo 2.3.5.

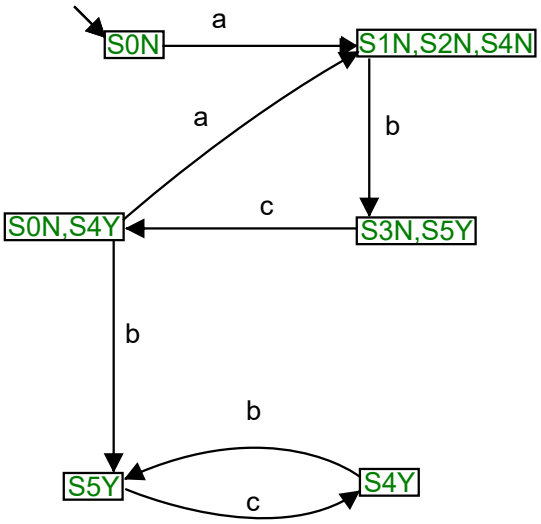


Figura 12 – Autômato diagnosticador G_{diag} do exemplo 2.3.5.

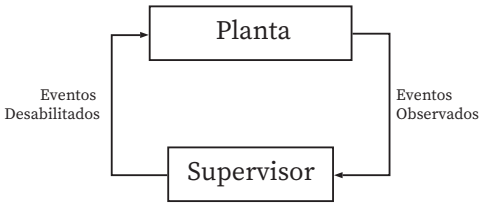


Figura 13 – SED em malha fechada.

rótulos Y. Suponha que a sequência afbcb tenha sido executada pelo sistema. O estado alcançado em G_{diag} a partir de $P_o(afbcb) = abcb$ é o estado 5Y, indicando, assim, que a falha ocorreu.

2.4 TEORIA DE CONTROLE SUPERVISÓRIO

Um sistema a ser controlado, em geral, corresponde a um conjunto de sub-sistemas obedecendo uma dada distribuição. Vistos isoladamente, estes sub-sistemas, possuem um comportamento básico original independente que quando atuando em conjunto com os demais, deve ser restringido de forma a cumprir com a função coordenada a ser executada pelo sistema global (CURY, 2001). A composição dos comportamentos de cada sub-sistema isolado pode então ser identificado com a planta G, com comportamento gerado e marcado, $L(G)$ e $L_m(G)$, respectivamente.

Um conjunto de restrições de coordenação define especificações a serem obedecidas fazendo com que as linguagens $L(G)$ e $L_m(G)$ sejam restritas ao comportamento delimitado pelas especificações. Assim, sequências indesejáveis de eventos, que violam as especificações, devem ser restringidas para que o objetivo de controle seja atingido. Esses estados proibidos em G, ocorrem de forma geral, por provocarem bloqueio ou por serem inadmissíveis. Pode ser também um caso de sequências de eventos que violam uma sequência de acontecimentos desejados, como por exemplo, em uma situação de justiça em compartilhamento de recursos.

Um supervisor é um elemento a fazer com que os sub-sistemas atuem de forma coordenada, e é denotado por **S**. Considera-se que o supervisor **S** interage com a planta G, numa estrutura de malha fechada apresentada na figura 13, na qual **S** observa a ocorrência dos eventos em G e define que eventos, dentro dos fisicamente possíveis, são permitidos de ocorrerem na sequência. Em outras palavras, **S** tem uma função para desabilitar eventos, e nesse sentido, diz-se que ele apresenta um controle de natureza permissiva. O conjunto de eventos habilitados num dado instante pelo supervisor, define uma entrada de controle. A cada nova ocorrência de evento observada em G, a entrada de controle é atualizada. Considera-se ainda que conjunto de eventos que afetam a planta G é particionado num conjunto de eventos desabilitáveis (também chamado de controláveis), e um conjunto de eventos cuja natureza não permite a desabilitação (também chamado de não-controláveis).

A teoria de controle supervisório foi enunciado em (RAMADGE; WONHAM, 1989), em que uma planta G é considerada um autômato determinístico, dotado de uma estrutura de

controle Υ, que corresponde aos eventos que devem ser habilitados em cada momento. O alfabeto de G divide-se em $\Sigma = \Sigma_c \cup \Sigma_{uc}$, em que Σ_c corresponde aos eventos controláveis, passíveis de desabilitação, e Σ_{uc} se refere aos eventos não controláveis, que ocorrem sem interferência do supervisor. A estrutura de controle é definida como

$$\Upsilon = \{\gamma \in 2^{\Sigma} : \gamma \supseteq \Sigma_{uc}\}$$

Define-se um supervisor para a planta G como um mapa

$$\mathbf{S} : L(\mathbf{G}) \rightarrow \Upsilon$$

que associa sequências da planta a eventos a serem habilitados. Ao supervisor $\mathbf{S}$ pode ser associada uma linguagem $M \subseteq L_m(\mathbf{G})$, sendo que o par $(\mathbf{S}, M)$ é definido como supervisor marcador. A linguagem M tem a função de determinar quais sequências da planta G permanecerão marcadas com a ação do supervisor. Para denotar G sob supervisão de $\mathbf{S}$, utiliza-se S/G, cuja linguagem gerada é $L(\mathbf{S}/G) \subseteq L(G)$ e linguagem marcada é $L_m(\mathbf{S}/G) = L(\mathbf{S}/G) \cap M$. Um supervisor $\mathbf{S}$ é dito ser não bloqueante para uma planta G se o sistema em malha fechada não possuir bloqueios, ou seja, $\overline{L_m(\mathbf{S}/\mathbf{G})} = L(\mathbf{S}/\mathbf{G})$.

Além disso, o supervisor $\mathbf{S}$ pode ser representado por um autômato $\mathbf{S} = (X,\Sigma,f,x_0,X_m)$ e um mapa de habilitações $\phi : X \rightarrow 2^{\Sigma}$ que relaciona a cada estado de $\mathbf{S}$ um conjunto de eventos a serem habilitados em G. Na maioria dos casos, qualquer autômato $\mathbf{S'}$ pode ser capaz de aplicar a ação de supervisão sobre G, desde que $L_m(\mathbf{S'} \parallel \mathbf{G}) = L_m(\mathbf{S})$ e $L(\mathbf{S'} \parallel \mathbf{G}) = L(\mathbf{S})$. Em (SU; WONHAM, 2004) é apresentado um método para redução do número de estados de um supervisor, que originalmente pode apresentar um tamanho elevado. A redução de supervisão reduz o número de estados, mantendo a mesma ação de supervisão.

Para que sejam atendidos alguns requisitos de funcionamento do sistema, de maneira geral, analisa-se o sistema em malha fechada. Tais requisitos podem ser representados por meio de uma linguagem-alvo $K \subseteq L_m(G)$, também chamada de especificação, que determina o comportamento desejado para a planta sob supervisão. Afirma-se que uma linguagem $K \subseteq \Sigma^*$ é controlável em relação a $L(G)$ se $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$. Existe um supervisor marcador não-bloqueante que implementa K, tal que $L_m(\mathbf{S}/G) = K$, se e somente se K for controlável em relação a $L(G)$. A classe de linguagens controláveis contidas em uma linguagem E, em relação à uma planta G é denotada por $C(E,G) = \{K : K \subseteq E \; e \; K$ é controlável e.r.a $L(G)\}$. Este conjunto é não vazio e fechado para união, o que implica que possui um único elemento supremo, denominado de $SupC(E,G)$, que é a máxima linguagem controlável contida em E. O supervisor $\mathbf{S}$ pode ser obtido pelo cálculo de $SupC(E,G)$, sendo E é o comportamento desejado em malha fechada para a planta G. O supervisor obtido é tal que $L_m(\mathbf{S}/G) = SupC(E,G)$, cuja ação de supervisão é não bloqueante e ótima. Portanto, o supervisor é concebido, dessa forma, como um agente projetado sobre uma especificação, de modo a impor a dinâmica desejada ao sistema em malha fechada (WONHAM; CAI *et al.*, 2019).

Diferentes estratégias podem ser utilizadas para síntese de supervisor, a depender do grau de complexidade da planta e da arquitetura do sistema envolvido. Uma abordagem utilizada no cálculo de um único supervisor é chamada de monolítica, e é utilizada mesmo nos casos em que a planta é composta por sub-sistemas e a especificação por sub-especificações. Nesta abordagem, a planta G é calculada como a composição paralela dos submodelos G_i e a especificação K é calculada como o produto síncrono das linguagens K_i que impõem restrições ao comportamento da planta. O supervisor monolítico $\mathbf{S}$, não bloqueante e ótimo, é sintetizado por meio da máxima linguagem controlável $SupC(K,G)$.

Para reduzir a complexidade da solução, existe uma alternativa que é a utilização da abordagem modular para síntese de supervisores (WONHAM; RAMADGE, 1988). Calcula-se um supervisor $\mathbf{S}$ para cada especificação K_i, por meio da máxima linguagem controlável. Deseja-se que os supervisores modulares sejam não conflitantes para garantir ausência de bloqueio, condição que se expressa por $\bigcap_{i=1}^{n} \overline{L_m(\mathbf{S}_i/G)} = \overline{\bigcap_{i=1}^{n} L_m(\mathbf{S}_i/G)}$. Em alguns casos a propriedade de não conflito é respeitada, e atinge-se um comportamento em malha fechada não bloqueante e ótimo, tal que $\bigcap_{i=0,n} L_m(\mathbf{S}_i/G) = SupC(K,G)$, em que K é a composição das especificações. Entretanto, nos casos em que não é possível obter supervisores modulares não conflitantes, pode-se calcular um coordenador para resolução de conflitos, que desabilita eventos, do sistema em malha fechada, que levam a situações de bloqueio (WONG; WONHAM, 1998) (QUEIROZ; CURY, 2005). Tomando $G_{conf} = \|_{i=1}^{m} (S_i/G)$ como o sistema em malha fechada que satisfaz a especificação K, mesmo apresentando bloqueio, o coordenador G_{coord} pode ser obtido por meio de $SupC(\Sigma^*,G_{conf})$.

2.5 CONCLUSÃO DO CAPÍTULO

Neste capítulo, foram apresentadas as ferramentas presentes na literatura, que podem ser utilizadas para a modelagem de sistemas, síntese e análise de autômatos (com observação total ou parcial de eventos), supervisores e diagnosticadores. Essa fundamentação é importante para entender a modelagem utilizada para representar o funcionamento da planta piloto analisada nesse trabalho, assim como a ideia por trás do controle supervisório adotado para trazer mais segurança para a planta. No próximo capítulo, a planta SMAR é apresentada, bem como o projeto de controle supervisório apresentado em Rafael G. Oliveira, Queiroz e Cury (2020) que será usado como base para a proposta de modelagem adotada.

3 CONTROLE DO PROCESSO HIERÁRQUICO INDUSTRIAL

No âmbito da exploração e produção de petróleo, inúmeros desafios surgem, desde a perfuração e extração eficiente até o transporte e processamento subsequente. Além disso, a indústria enfrenta constantes pressões ambientais e regulatórias que exigem soluções inovadoras e sustentáveis. A complexidade aumenta mais quando é considerada a interconexão de sistemas em uma instalação completa, envolvendo a otimização de processos, a segurança operacional e a gestão eficiente de recursos. Portanto, seu estudo de caso oferece *insights* valiosos dentro de um contexto mais abrangente e complexo. Embora o estudo de um sistema se concentre em um aspecto específico, os resultados têm o potencial de impactar positivamente a operação global da IPG.

O sistema considerado neste trabalho é parte de uma planta piloto intitulada PD3 fabricada pela Smar Automação Industrial, mostrada na figura 14, instalada na Universidade Federal de Santa Catarina para treinamento e pesquisa de tecnologias avançadas de controle e automação. O processo industrial é utilizado para controlar o nível de líquido de um tanque. Portanto, uma parte dessa planta é suficiente para representar o processo demonstrado, utilizando-se uma bomba, que fornece líquido ao tanque, e uma válvula pneumática, que serve para regular este nível. A planta é instrumentada com dispositivos sensores conectados entre si através de uma rede industrial Foundation Fieldbus (FF). O diagrama de tubulação e instrumentação do sistema é apresentado na figura 15.

O processo é cíclico. A água sai do tanque reservatório impulsionada pela bomba, passando pela válvula Fieldbus FY302, que faz o controle de vazão de entrada no tanque de nível T1. O tanque T1 possui uma válvula manual para realizar o retorno de água para o tanque reservatório.

Figura 14 – O processo industrial de controle de nível de líquido.

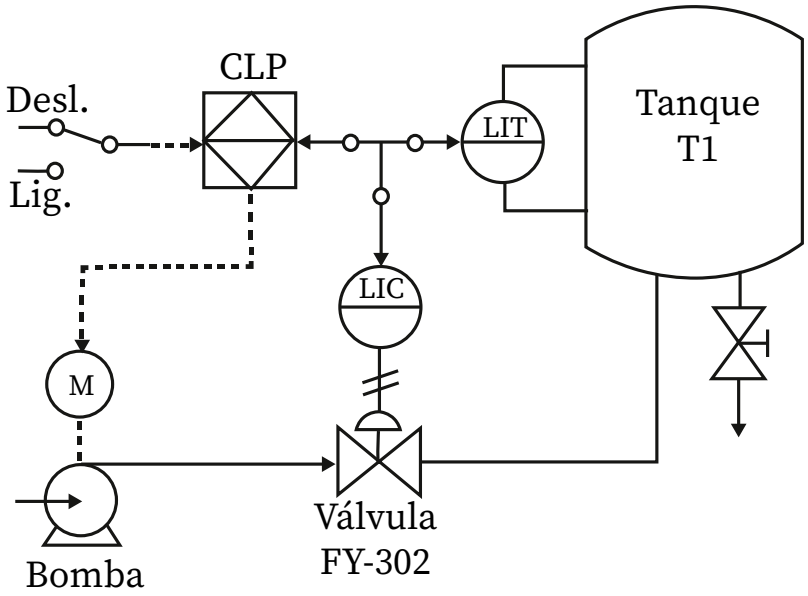


Figura 15 – Diagrama de tubulação e instrumentação (P&ID) do sistema.

A instrumentação do sistema é composta por dois sensores FF, um sensor de nível no tanque (LIT) e um sensor de posição da válvula (LIC) que são utilizados para implementar um sistema de controle PID distribuído. Os parâmetros dos dispositivos LIT e LIC podem ser lidos e escritos por um CLP através da interface FF. Além disso, o CLP também recebe o sinal de uma chave seletora implementada em um painel de Interface Homem-Máquina (HMI) e pode controlar a bomba. Uma válvula manual é posicionada na saída do tanque.

A estratégia de controle supervisório adotada para este sistema visa prevenir consequências relacionadas a perturbações no controlador PID não previstas em sua fase de projeto, que podem ser materializadas devido a erros de operação, como intervenção humana no setpoint e manipulação manual de válvulas (OLIVEIRA, R. G.; QUEIROZ; CURY, 2020). Esses problemas podem ocorrer no estado estacionário do processo ou nos procedimentos não rotineiros, como inicialização e desligamento. As consequências de tais problemas podem causar subfluxo ou transbordamento do nível do tanque. Assim, em Rafael G. Oliveira, Queiroz e Cury (2020) é desenvolvida uma estratégia de controle supervisório modular para garantir que o sistema opere sob restrições de segurança.

Embora o procedimento de controle supervisório projetado em Rafael G. Oliveira, Queiroz e Cury (2020) tenha sido implementado e apresentado resultados satisfatórios, ele se baseia na premissa de que todos os componentes do sistema são perfeitos, $i.e.$, apenas o comportamento nominal dos componentes do sistema é considerado no projeto das estratégias de controle PID e de supervisão. Assim, embora a lógica de controle supervisório possa de fato evitar as consequências de operações incorretas, ela é limitada ao comportamento nominal dos componentes do sistema, como sensores e atuadores, e não pode garantir as especificações do sistema ao lidar com hardware defeituoso. Nesse sentido, caso os sensores ou atuadores não tenham o desempenho esperado, é possível que a lógica hierárquica de controle permita uma operação imprevisível do sistema, podendo levar a falta de líquido ou transbordamento do tanque.

Em vários casos, um controle supervisório pode ser projetado para evitar danos consequentes que podem ser alcançados após a ocorrência de um evento de falha. Nesse contexto, o controle é desenvolvido para prevenir o pior cenário possível, permitindo que o sistema funcione sob comportamento de falha sem atingir consequências severas. Porém, mesmo nesses casos, o comportamento de falha não é desejável, pois pode causar uma deterioração acelerada dos equipamentos e apresenta um desafio à operação segura do sistema. Assim, se ocorrer um evento de falha, sua detecção deve ser computada para permitir a substituição do hardware defeituoso para que o sistema possa operar corretamente. Neste trabalho, é considerado um projeto tradicional para o controle supervisório, em que todas as especificações do sistema são levadas em consideração para o modelo de comportamento livre de falhas. O método proposto visa adaptar os modelos de componentes do sistema utilizados no projeto de controle supervisório para fins de diagnóstico, sem interferir na lógica de controle projetada, agregando uma nova camada para a operação mais segura dos processos industriais. Os modelos utilizados no projeto do controle supervisório, proposto em Rafael G. Oliveira, Queiroz e Cury (2020) para o sistema, representado na figura 14, são apresentados a seguir e um método para modificar esses modelos para fins de diagnóstico de falhas é proposto no Capítulo 4.

3.1 REDE FOUNDATION FIELDBUS

A planta SMAR PD3 possui a arquitetura de Rede Foundation Fieldbus que é utilizada para comunicar os dispositivos de campo, com o controle que realiza as ações necessárias para

manter o estado no regime projetado. Os dispositivos como a válvula e o sensor de nível são ligados por um único barramento, que recebem e enviam informações para o CLP, para os blocos da própria rede FF e para o controle PID para manter o nível no *setpoint* determinado.

Arquiteturas de rede desse tipo são utilizadas, portanto, para comunicar informações do chão de fábrica até o nível gerencial, além de atender requisitos de velocidade e segurança. Algumas vantagens são a melhora no desempenho da produção com tempos de resposta mais rápidos, e a aquisição de dados mais eficiente. Quanto mais informação, melhor uma planta pode ser operada, gerando mais produtos e tornando-se mais lucrativa. A informação digital e os sistemas de comunicação permitem que se coletem os mais diversos tipos de dados para inúmeras finalidades (DIAS, 2015).

O protocolo Fieldbus Foundation foi proposto inicialmente em 1994 por uma fundação internacional, de mesmo nome, sediada nos Estados Unidos, composta pela união de duas organizações de fieldbus existentes: a Interoperable System Protocol (ISP) e uma divisão da WorldFIP francesa, além de um grupo de 85 empresas. Aliados as organizações internacionais de normalização International Electrotechnical Commission (IEC) e International Society of Automation (ISA), a fundação propõe uma arquitetura também baseada no modelo ISO/OSI reduzido de 3 camadas: física, enlace e aplicação (BRANDÃO, 2000). Ele foi concebido principalmente para a aplicação em controle de processos contínuos e atualmente, existem vários softwares de configuração e supervisão para esse sistema. Via de regra, esses softwares utilizam o padrão de comunicação denominado OPC (OLE for Process Control) .

Diferente da maioria das tecnologias de rede em que o controle e supervisão do sistema é realizado por um controlador central, em Fieldbus essas ações são efetuadas de forma distribuída nos próprios equipamentos da rede. Eles possuem microprocessadores integrados, o que permite dispor de blocos de função, além de poderem facilmente se comunicar com outros dispositivos. Isto ajuda a diminuir o número de E/S e a necessidade de equipamentos dedicados de controle (MULER, 2018).

A arquitetura de um dispositivo Fieldbus é baseada em blocos funcionais, que são responsáveis por executar as tarefas necessárias para as aplicações atuais, como aquisição de dados, controle em cascata e feedback, cálculos e atuação. Eles permitem a implementação de estratégias de controle no dispositivo de campo, tornando-o parte integrante do sistema de controle. Cada bloco funcional contém um algoritmo, um banco de dados (de entradas e saídas), um nome e um número de tag que deve ser exclusivo na planta. A Fieldbus Foundation especifica para programação da estratégia de controle, uma linguagem de programação gráfica simples que seria um diagrama de blocos funcionais. Tais blocos são programas residentes nos dispositivos da rede e encapsulam funções e algoritmos básicos de automação e controle de processos. A configuração distribui os blocos funcionais em diferentes equipamentos de campo, caracterizando assim uma estratégia de controle distribuída como mostra a figura 16 (PANTONI, 2006).

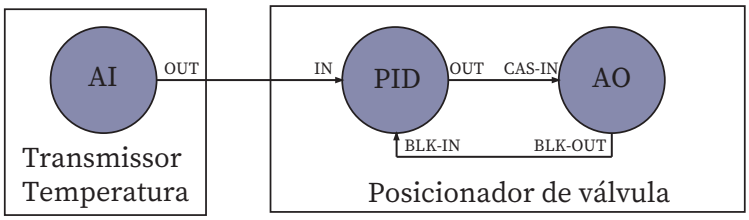


Figura 16 – Diagrama de blocos funcionais de uma estratégia de controle (figura do autor).

Existem três tipos de rede para o protocolo FF, definidos na norma IEC 61784. São eles: H1, H2 e High Speed Ethernet (HSE) (PANTONI, 2006).

O primeiro tipo, denominado H1, é destinado ao uso nos instrumentos de campo. A camada física deste tipo prevê uma taxa de comunicação de 31,25 Kbits/s, half-duplex (fluxo duplo entre as estações de origem e destino, mas não simultâneo) em par trançado blindado com codificação binária do tipo Manchester. Os instrumentos de campo nesse tipo são alimentados pelo barramento e adequado ao uso em áreas classificadas (ambientes com risco de explosão). O FF do tipo H2 é um tipo de rede em desuso, que foi substituído pelo tipo HSE, não existindo produtos compatíveis com tal especificação no mercado. O tipo denominado HSE é destinado ao uso no nível dos controladores de processo e de estações de configuração e de monitoramento de processos. A comunicação em uma rede HSE baseia-se no protocolo Ethernet de camada física com taxa de comunicação de 100 Mbits/s entre dispositivos não alimentados pelo barramento. O mecanismo de controle de acesso ao meio físico é baseado nos protocolos IP (camada de rede) e TCP/UDP (camada de transporte).

Alguns benefícios encontrados na rede FF são:

- interoperabilidade: Permite que os equipamentos sejam fabricados por diferentes fornecedores, mas que funcionem em conjunto formando uma única rede;
- requer somente um barramento para múltiplos dispositivos;
- a alimentação de equipamentos FIELDBUS pode ser feita opcionalmente através dos mesmos condutores de comunicação ou separadamente;
- o controle pode ser executado dentro do dispositivo de campo.

Os dispositivos de campo e equipamentos de controle como, por exemplo, transmissores inteligentes (de pressão, temperatura, densidade), atuadores, controladores lógicos programáveis, estações de supervisão e de configuração, conectam-se em redes de comunicação como representa a figura 17.

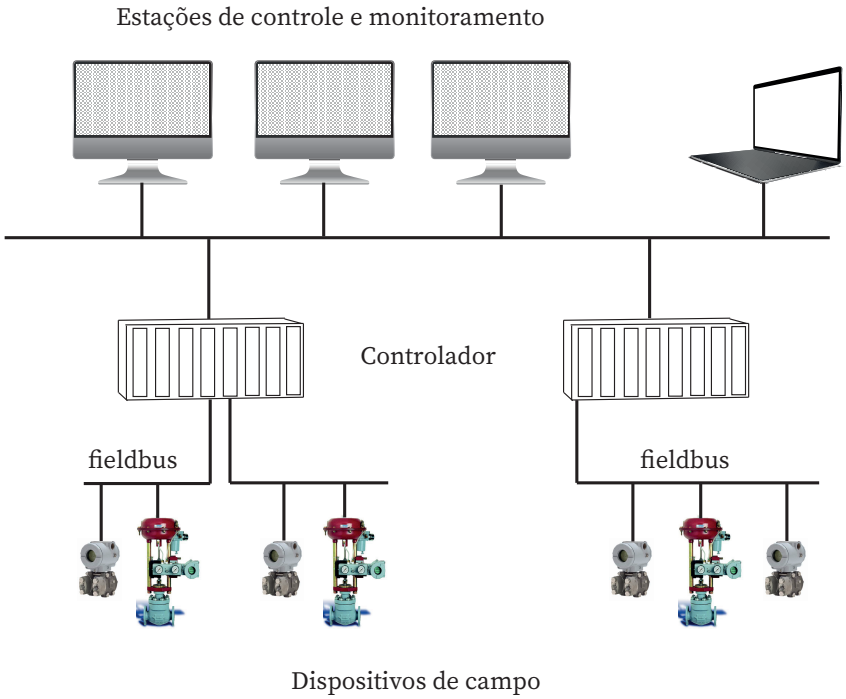


Figura 17 – Sistema Fieldbus e seus componentes (figura do autor).

Nessa figura é possível observar que os dispositivos de leitura e atuação estão interligados por um único barramento. Diferente de dispositivos como a bomba hidráulica ou a válvula tipo globo, esses dispositivos que foram ilustrados, são exclusivos da rede FF e são apresentados a seguir:

Posicionador de válvula FY302: como parte do conjunto da válvula, o posicionador FY302 é utilizado para controlar válvulas pneumáticas em sistema Fieldbus. Ele produz a pressão de saída requerida para situar a abertura da válvula conforme entrada recebida pela rede Fieldbus. A pressão de alimentação do posicionador é de 1.4-7 bar (20-100 psi). A figura 18 apresenta o posicionador FY302.

Figura 18 – Posicionador FY302 utilizado.

Transmissor de Pressão Diferencial LD302D: dispositivo utilizado para medição de grandezas, utiliza técnica de medição de pressão por célula capacitiva e microprocessador no seu circuito eletrônico. Utilizado para medição de pressão diferencial, absoluta, manométrica, nível e vazão. Pode ser configurado localmente utilizando chave magnética ou pelo programa Syscon. A figura 19 apresenta o transmissor LD302D.

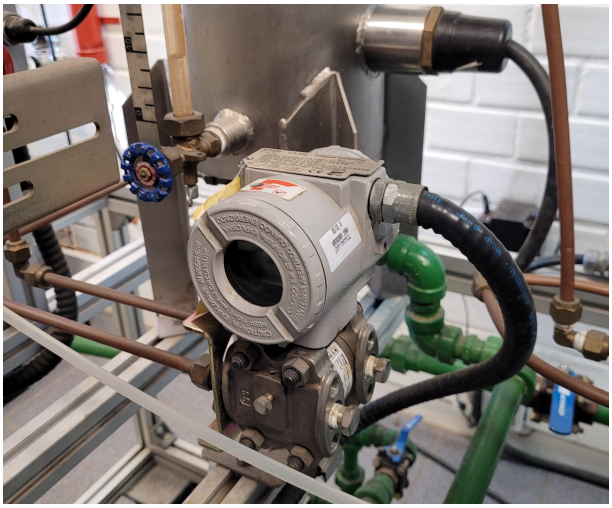


Figura 19 – Transmissor LD302D utilizado.

Arquitetura do DF65

O CLP da fabricante SMAR é representado pelo sistema DF65, no qual realizam-se os cálculos lógicos necessários para controlar a planta. Um sistema DF65 é composto de um módulo de CPU, o módulo de fonte de alimentação e um conjunto de módulos I/O para interagir com os

Modelo	Nome	Descrição
DF50	Módulo de fonte	Fonte de alimentação redundante para alimentar o *backplane*.
DF51	Módulo Processador	Conecta equipamentos Fieldbus no barramento H1, executando a função de Link Active Scheduler (LAS) da rede.
DF52	Fonte Fieldbus	Módulo especialmente desenvolvido para alimentar a rede Fieldbus.
DF53	Impedância de Linha	Módulo desenvolvido para fornecer uma impedância ideal para a rede Fieldbus.

Tabela 1 – Módulos da Bridge DFI302

sinais de campo. Os módulos são plugados nos slots que fazem parte dos racks e conectam entre si utilizando um barramento comum. Um sistema DF65 pode ter até 15 racks que implica em no máximo 60 módulos por sistema. O rack utilizado na planta PD3 é apresentado na figura 20 e os principais elementos da arquitetura do DF65 são:

Figura 20 – Rack e os módulos da planta PD3.

- Coprocessador Lógico DF65: é necessário para interagir os outros módulos do sistema DF65 e no qual realizam-se os cálculos lógicos.
- Módulo de Interface Fieldbus FB700: utilizado para integrar comunicação do CLP com o Fieldbus.
- Módulos de entradas e saídas: são os caminhos pelos quais o CLP lê ou envia comandos da CPU.
- Bridge Universal Fieldbus DFI302: é um elemento chave na arquitetura dos sistemas de controle. Permite acesso direto a I/O e controle avançado para aplicações contínuas e discretas. O DFI 302 é montado sobre um *backplane*, em que todos os módulos são instalados, incluindo o DF50, DF51, DF52 e DF53, que são apresentados na Tabela 1.

3.1.1 Foundation Fieldbus e a Indústria de Petróleo e Gás

O primeiro sistema Foundation Fieldbus do mundo em uma aplicação em alto-mar está na Bacia de Campos, no Rio de Janeiro, na plataforma "Namorado 1", conhecida como PNA-1 (SMAR, 2008). O sistema instalado mede a pressão e a vazão do gás natural. Ele opera com uma rede que conecta os equipamentos de campo e as estações de operação dentro da sala de controle. A tecnologia Fieldbus foi escolhida devido à sua facilidade de integração com todos os dados gerados pelos instrumentos do sistema e pela sua moderna arquitetura de controle distribuído, que é localizado dentro dos equipamentos de campo.

A plataforma PNA-1 é uma das mais antigas da Petrobrás (PETRONOTÍCIAS, 2014). Ela entrou em produção em 1983, não tinha uma sala de controle, somente uma sala de supervisão, e a necessidade de ter operadores no campo, numa área de risco. A equipe de Exploração & Produção da Bacia de Campos escolheu o FF pela garantia de interoperabilidade dada pelo protocolo, e por já estar implementada em várias plantas onshore (CONTROLE & AUTOMAÇÃO, 2000). A Petrobras tem alto nível de exigência no assunto segurança, e modernizar a linha de gás foi uma forma planejada de reduzir riscos de perdas do sistema e paralisações da plataforma.

Figura 21 – Plataforma Namorado 1 (PETRONOTÍCIAS, 2014).

Tamanha é a representatividade do sistema Fieldbus nessa aplicação, que a Petrobras escolheu o System 302 (o mesmo utilizado na planta PD3) para controlar a linha, com cerca de 26 dispositivos fieldbus possuindo sistema de supervisão. O controle de pressão do sistema de importação de gás utilizam dois transmissores e uma única válvula de controle, similar a aplicação encontrada nesse trabalho. Dois PIDs podem dar o sinal de referência para a válvula de controle de pressão, escolhido através de um bloco seletor (SEL). Em plataforma de petróleo, os algoritmos são na maioria das vezes, implementados com PID. Para a escolha de outro protocolo ou tecnologia, segundo o levantamento da equipe de EP da Bacia de Campos, haveria a necessidade de aquisição de mais equipamentos e de maior complexidade na integração do sistema.

A escolha de produtos FF também foi influenciada pela arquitetura de controle distribuído nos dispositivos de campo, reduzindo o número de equipamentos na sala de controle restando ali apenas Interface Homem Máquina (IHM) e equipamento de comunicação em rede (CONTROLE & AUTOMAÇÃO, 2000). Isso significa que para as funções mais simples de controle, não havia

a necessidade de usar um CLP já que os dispositivos podem ser utilizados para controlar o sistema. Uma redução do número de dispositivos também é vantajosa do ponto de vista de peso, pois em plataformas offshore, o peso dos equipamentos e acessórios implica grandes ganhos associados com uma plataforma mais leve.

Estima-se que quando as operações na planta eram utilizando instrumentação pneumática, poderia ocorrer muita variação, em torno de 20% a 30% nas variáveis controladas. Com o Fieldbus, ou instrumentação eletrônica, essa porcentagem não passa de 1% ou 2% (CONTROLE & AUTOMAÇÃO, 2000). Na arquitetura com a qual a Petrobras trabalha - com sintonia de processo com várias malhas interagindo -, fica muito mais fácil trabalhar com Fieldbus que com CLP. Caso se chegue aos ajustes desejados, espera-se Fieldbus em toda a plataforma e não apenas em áreas restritas.

A relevância dos sistemas FF na IPG é evidenciada pela necessidade crucial de otimizar suas aplicações. A evolução contínua dos sistemas e tecnologias na indústria destaca a importância crescente de adotar abordagens inovadoras como o FF. A modelagem, nesse cenário dinâmico, emerge como um componente vital para a integração eficaz desses sistemas que o FF é capaz de implementar. Sendo assim, há uma necessidade de modelagem detalhada para implementação de modelos de sistemas a eventos discretos, e que será visto a seguir na aplicação prática.

3.2 MODELOS DO SISTEMA DE CONTROLE SUPERVISÓRIO

Existe uma necessidade de expressar o modelo do tanque com as situações de estar subindo e descendo enquanto ele está nas posições acima e abaixo do setpoint. Ou seja, o sistema de controle tem uma curva que se comporta de forma peculiar: quando o nível está acima do SP, não necessariamente significa que o mesmo vai descer. O exemplo disso pode ser visto na figura 22, na qual nos primeiros instantes em que o SP é atingido, a variável associada ao eixo y continua crescendo seu valor até um valor de máxima ultrapassagem, devido à dinâmica do sistema e até alcançar seu regime permanente no tempo de acomodação. Dada essa complexidade, realizar uma abstração desse nível em SEDs pode se tornar uma tarefa difícil. Dito isso, optou-se por adotar a passagem de faixas de nível como um evento ao invés de anotar cada instante de nível como um evento.

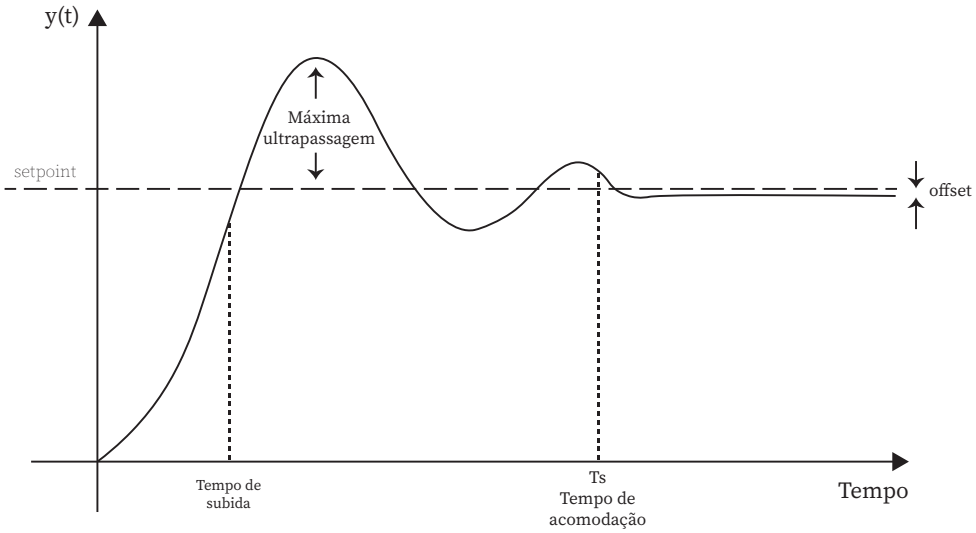


Figura 22 – Gráfico do controle PID.

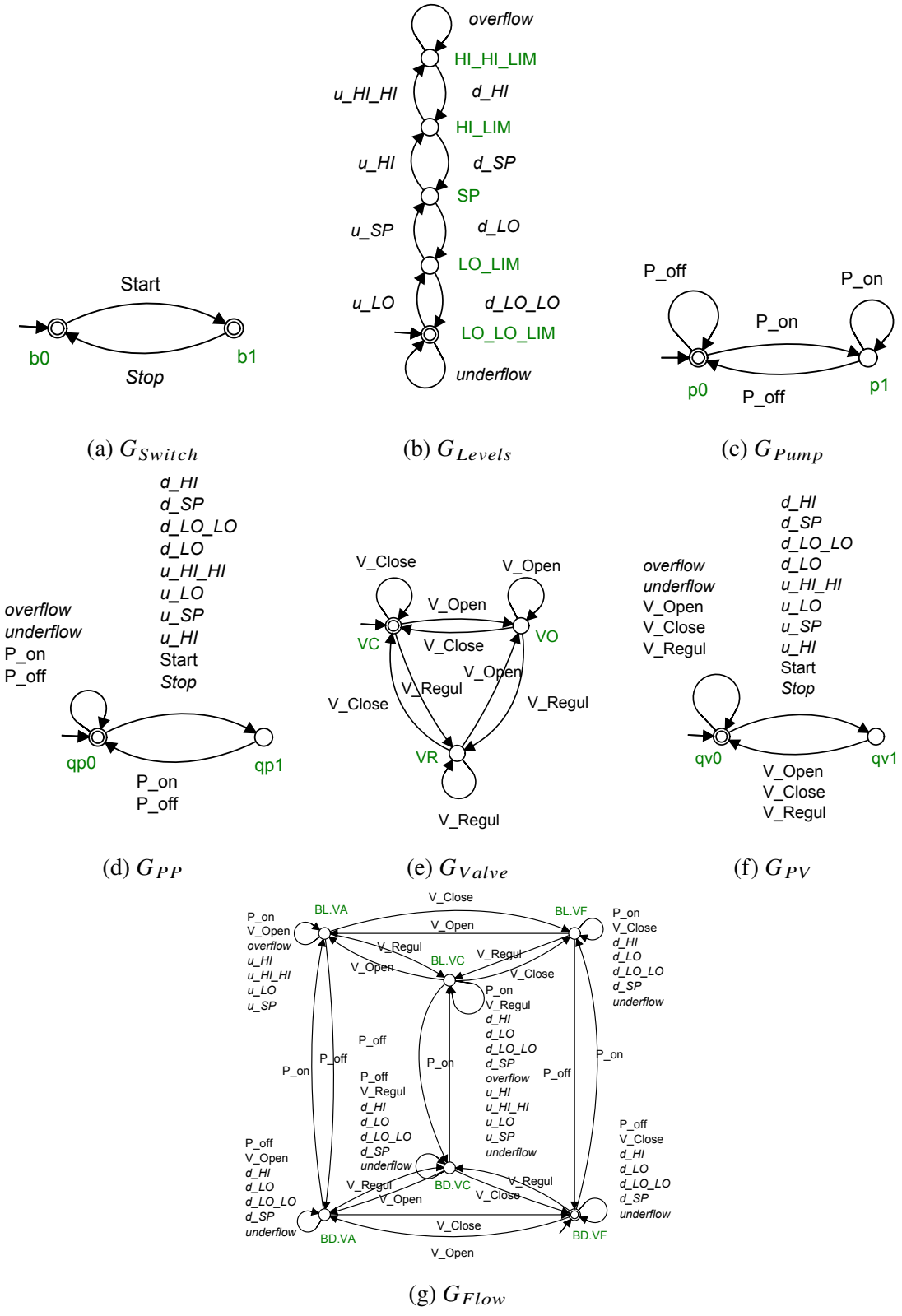
O sistema apresentado na figura 14 é composto por sete módulos (OLIVEIRA, R. G.; QUEIROZ; CURY, 2020), apresentados na figura 23: (a) G_{Switch} modela o comportamento da chave de seleção da HMI; (b) G_{Levels} modela o nível de líquido no tanque; (c) modelos G_{Pump} comandam eventos para a bomba de acordo com mudanças no nível do tanque; (d) G_{PP} modela a preempção das transições de nível por eventos de bombeamento e evita que a bomba fique ligando e desligando ininterruptamente. O mesmo vale para a válvula. (e) G_{Valve} modela o comportamento da válvula; (f) G_{PV} modela a preempção das transições de nível por eventos de válvula; (g) G_{Flow} modela a vazão de líquido no tanque.

O modelo do sistema de comportamento global é definido por $G = G_{Switch} \| G_{Levels} \| G_{Pump} \| G_{PP} \| G_{Valve} \| G_{PV} \| G_{Flow}$. No modelo G_{Switch} (figura 23(a)), os eventos $Start$ e $Stop$ modelam os comandos de inicialização e finalização pelo operador usando o terminal HMI. O evento de stop da chave é não-controlável porque senão tiraria a autonomia do supervisório de desligar. Se fosse controlável seria necessário outro botão não controlável de emergência no modelo. Eventos $Overflow$, d_HI, d_SP, d_LO, d_LO_LO, $underflow$, u_LO, u_SP, u_HI, e u_HI_HI do modelo G_{Levels} (figura 23(b)) representam variações do nível de líquido do tanque, em que os prefixos u e d indicam os cruzamentos dos limiares para cima e para baixo, respectivamente. As partições dos níveis dos tanques são feitas de acordo com as boas práticas descritas pela Fieldbus Foundation (FIELDBUS FOUNDATION, 2002). O estado SP representa que o sistema está próximo do ponto de ajuste escolhido para o controle PID. Para o modelo de bomba G_{Pump} (figura 23(c)), os eventos P_on e P_off representam a ativação e desativação da bomba, respectivamente. No modelo da válvula, G_{Valve} (figura 23(e)), os eventos V_Open e V_Close simbolizam os eventos de comando para abrir e fechar 100% a válvula, respectivamente , enquanto o evento V_Regul é usado para representar a comutação do controle do supervisório para o controle PID. Os modelos G_{PP} (figura 23(d)) e G_{PV} (figura 23(f)) são usados para descrever a preempção do bomba e a válvula na dinâmica de nível, respectivamente. Finalmente, G_{Flow} (figura 23(g)) modela a dinâmica do fluxo de líquido no tanque. Observe

Tabela 2 – Controlabilidade dos eventos.

Modelo	**Evento**	**Tipo**
Levels	d_HI	não-controlável
	d_LO	não-controlável
	d_LO_LO	não-controlável
	d_SP	não-controlável
	overflow	não-controlável
	u_HI	não-controlável
	u_HI_HI	não-controlável
	u_LO	não-controlável
	u_SP	não-controlável
	underflow	não-controlável
Valve	V_close	controlável
	V_open	controlável
	V_Regul	controlável
Switch	Start	controlável
	Stop	não-controlável
Pump	P_off	controlável
	P_on	controlável

que os eventos dos modelos G_{PP}, G_{PV} e G_{Flow} já estão descritos nos demais modelos do sistema. Além disso, na tabela 2 os eventos são apresentados com a classificação em relação a controlabilidade.



(a) G_{Switch} (b) G_{Levels} (c) G_{Pump}

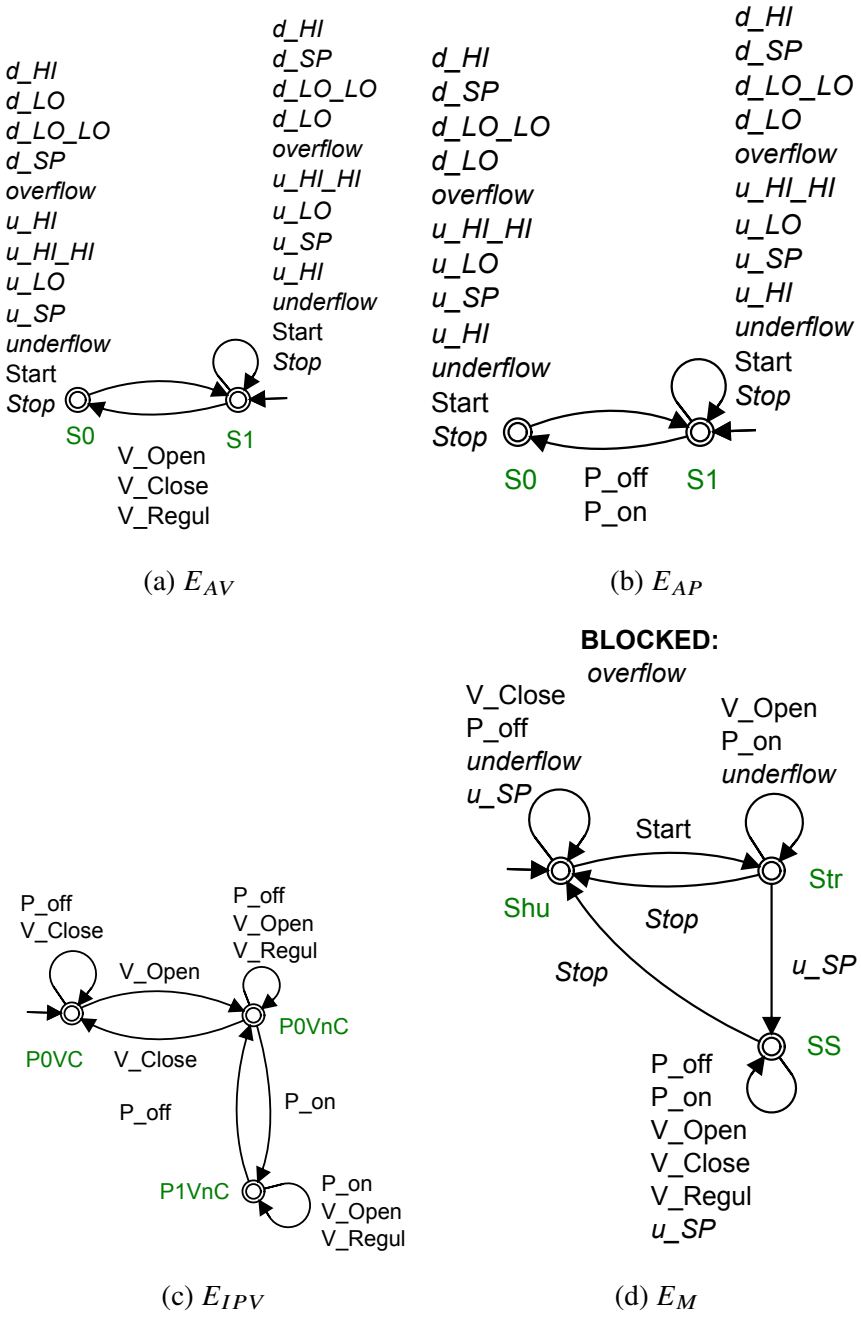
(d) G_{PP} (e) G_{Valve} (f) G_{PV}

(g) G_{Flow}

Figura 23 – Modelos do sistemas para o projeto de controle supervisório.

Para este sistema, quatro especificações, mostradas na figura 24, foram consideradas para garantir restrições de segurança: a ação reativa na válvula (E_{AV}), a ação reativa na bomba (E_{AP}), o intertravamento entre bomba e válvula (E_{IPV}), e os modos de operação, juntamente com os

limites de nível de líquido em cada modo (E_M). Na especificação E_M, o evento $overflow$ está bloqueado, pois está definido para E_M, mas não é viável em nenhum de seus estados.



(a) E_{AV} (b) E_{AP}

(c) E_{IPV} (d) E_M

Figura 24 – Especificações para o projeto de controle supervisório.

3.3 SÍNTESE DOS SUPERVISORES DO SISTEMA DE CONTROLE SUPERVISÓRIO

O funcionamento em malha-fechada depende dos subsistemas da planta interagindo entre si, que formam o autômato da planta. Ele é composto pela composição síncrona dos modelos apresentados na Tabela 3. A composição síncrona de $G = G_{niveis} \parallel G_{chave} \parallel G_{bomba} \parallel G_{PB} \parallel G_{Valvula} \parallel G_{PV} \parallel G_{vazao}$ resulta em um autômato de 240 estados livres de bloqueio. Dado que o controle modular é realizado utilizando a especificação com as plantas que fazem

Tabela 3 – Subsistemas da Planta G.

Subsistema	**Estados**	**Eventos**	**Transições**
Gbomba	2	2	4
Gbotão	2	2	2
Gníveis	5	10	10
Gpb	2	14	16
Gpv	2	15	18
Gválvula	3	3	9

Tabela 4 – Modelos utilizados e seus respectivos números de estados reduzidos.

x	**Ex**	**G**	**Kx**	**Sx**	**Sxr**
Av	2	240	260	260	2
Ab	2	240	258	258	2
Ibv	3	240	200	200	3
M	3	240	199	167	14

parte do seu alfabeto, é gerado um supervisor para cada especificação. No caso de E_{AV},E_{AB} e E_{IPV}, ao realizar a composição paralela de E_x com a planta G_x é observado que resulta em um autômato equivalente ao obtido com a máxima linguagem controlável entre eles, ou seja, controlável e não bloqueante. Isso significa que para essas especificações podem ser utilizadas diretamente como os supervisores da planta, respectivamente, $\mathbf{S}_{AVR}$, $\mathbf{S}_{ABR}$ e $\mathbf{S}_{IPVR}$. Diferente deles, a especificação dos modos de operação necessita do cálculo de $\mathbf{S}_M$ por meio da máxima linguagem controlável. O autômato encontrado apresenta 167 estados e uma maneira de contornar a complexidade gerada pelo número excessivo de estados é a utilização do algoritmo de Su e Wonham (2004). Dessa forma, o autômato resultante apresenta 14 estados. O número de estados de cada modelo da especificação, da planta, K_x,$\mathbf{S}_x$ e $\mathbf{S}_{xR}$ são apesentados na Tabela 4, sendo $K_x = G \parallel E_x$, $\mathbf{S}_x = SupC(E_x,G)$, e $\mathbf{S}_{xR}$ é o supervisor $\mathbf{S}_x$ reduzido. Os eventos desabilitados de cada supervisor são aqueles que estão no alfabeto mas não estão definidos no estado.

Esses supervisores atuando simultaneamente conflitam, portanto, um quinto supervisor pode ser calculado para desabilitar os eventos do sistema que levam a situações de bloqueio. Esse supervisor é chamado de coordenador e ele é projetado sobre a especificação de ausência de bloqueio (WONG; WONHAM, 1998) (QUEIROZ; CURY, 2005). O sistema em malha fechada se torna, portanto, $G_{Conf} = G \parallel \mathbf{S}_{AVR} \parallel \mathbf{S}_{ABR} \parallel \mathbf{S}_{IPVR} \parallel \mathbf{S}_{MR}$ e apresenta 142 estados. Obtendo a máxima linguagem controlável desse autômato encontra-se um modelo de supervisor de 134 estados, o que ainda é um número bastante elevado. E como realizado anteriormente, tal autômato pode ser reduzido resultando no $\mathbf{S}_{CoordR}$ com cinco estados. O número de estados dos supervisores reduzidos facilitam a implementação de tais modelos no CLP. Caso fosse calculado um único supervisor para todo o sistema, chamado de monolítico, a máxima linguagem controlável encontrada seria um modelo com 134 estados, que após passar pelo processo de redução se tornariam 38 estados. Entretanto, optou-se pela utilização dos supervisores modulares que apresentam 26 estados se somados todos os supervisores que serão

Tabela 5 – Supervisores que realizam o controle da Planta.

Modelo	**Estados**	**Eventos**	**Transições**
Sab	2	14	26
Sav	2	15	27
Sibv	3	5	12
Smr	14	16	72
Coordenador	5	13	38

utilizados e são apresentados na Tabela 5, com seus respectivos números de estados, transições e eventos.

Os supervisores são implementados no CLP de forma independente do controle PID implementado nos dispositivos da rede FF, e observam eventos como a mudança de nível e comando de chave seletora para permitir ou desabilitar eventos de ligar ou desligar a bomba, ou atuar na saturação da válvula. o CLP por sua vez, é responsável por gerar os eventos que o supervisor vai observar. Isso é feito comparando os valores do sensor de nível lidos pelo transdutor, por exemplo, e relacionando com faixas de operação de nível do tanque. A figura 25 apresenta os valores estipulados para definir os níveis do tanque. Por exemplo, se o nível estiver abaixo de 5% é considerado *underflow*, entre 5% e 20% é considerado *LO_LO_Lim*, e assim por diante. O *setpoint* pode ser alterado pelo operador a qualquer instante, por meio de um software de supervisão, inclusive fora dos limites do tanque podendo forçar a atuação dos supervisores. Além disso, o operador também pode ligar ou desligar o sistema de controle por meio de uma chave seletora localizada na parte frontal do painel elétrico da máquina.

Outro ponto importante é entender que ocorre um funcionamento cíclico no processo de gerar as desabilitações: as desabilitações influenciam no funcionamento da planta, estas influenciam a máquina de estados dos supervisores e que influenciam as desabilitações. Portanto, qualquer erro de sincronismo pode acarretar em um mal funcionamento do controle supervisório. Esses modelos são programados no CLP uilizando a linguagem Ladder, de acordo com os métodos de Queiroz e Cury (2002) que estrutura uma dinâmica de implementação para evitar tais problemas. Sua proposta é implementar os Supervisores separando a ocorrência das habilitações e as desabilitações. As habilitações evoluirão os estados do autômato supervisor e as desabilitações serão definidas em decorrência desses estados. Já a planta será implementada, separando as transições controláveis e não-controláveis. A planta só poderá evoluir se o evento correspondente a transição analisada não tiver sido desabilitado. A evolução da planta informará ao supervisor qual é o próximo evento habilitado, e a sequência operacional, o que deve ser incitado nos contatos físicos da ligação elétrica. Um evento na planta ocorre se fizer parte do modelo e não tiver sido desabilitado. Sempre que a planta evolui, o supervisor deve ser novamente verificado para manter as habilitações atualizadas. E a última observação referente à planta, é que as transições não-controláveis devem ser verificadas primeiro, já que acontecem antes. Isso se deve ao fato de que eles podem acontecer a qualquer momento devido à dinâmica do sistema, como o alcance de algum nível, em que não há controle sobre a execução. Portanto, processam-se os

eventos que podem ser executados não-controláveis antes de evoluir um novo evento controlável. Por fim, as sequências operacionais vão ser implementadas a fim de traduzir para a parte física da planta o que deve ser ativado ou desativado, como a saída elétrica do CLP, por exemplo.

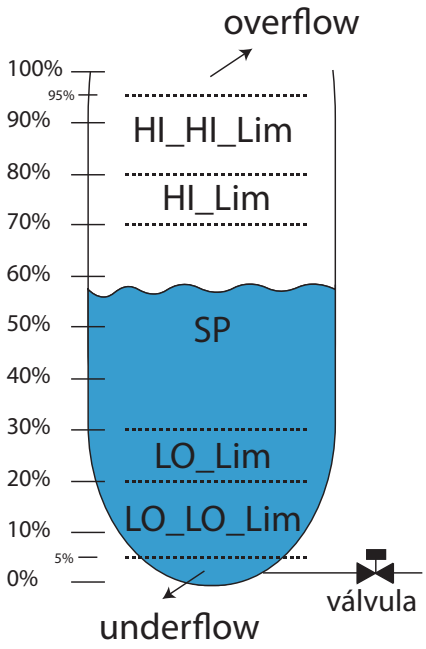


Figura 25 – Faixas estipuladas de níveis do tanque.

Como dito anteriormente, o PID funciona independente do controle supervisório implementado no CLP. Isso ocorre porque ele é programado internamente do dispositivo inteligente Fieldbus, que além de possuir um microprocessador próprio, pode receber e enviar informações para outros dispositivos de campo. Nesse caso, o controle PID está implementado juntamente com a válvula de controle. Tais considerações são construídas através do software Syscon, que permite a adição dos dispositivos presentes em campo e também das funções que eles recebem. A cada dispositivo é atribuído um bloco de função que tem o seu propósito seja coletando informações, configurando exibições no display, enviando informações para outro dispositivo ou controlando o sistema de maneira geral. A relação estabelecida na estratégia de controle, presente na figura 26, apresenta como é realizada a interação dos blocos funcionais de um dispositivo com outro. O bloco MAO, MAI e MDI envia e recebe sinais analógicos para o CLP e recebe sinais digitais, respectivamente. O bloco L_AI disponibiliza os valores de nível para o controlador PID (dentro do bloco V_PI) e para os supervisores implementados no CLP. O bloco V_AO tem a função de disponibilizar sinais referentes a abertura da válvula e o bloco SEL recebe sinais analógicos do PID, do bloco MAI e MDI, e disponibiliza para a válvula um valor calculado pelo PID ou valores pré-definidos, similar a um multiplexador.

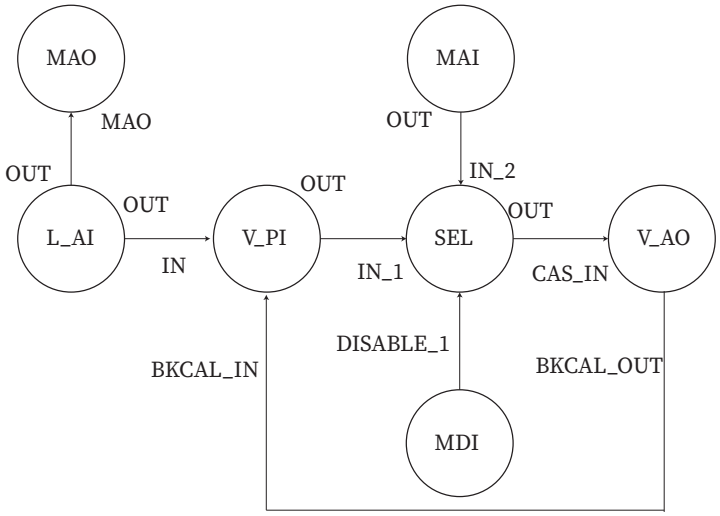


Figura 26 – Estratégia de controle Fieldbus.

3.4 COMPORTAMENTO DO SISTEMA

Para representar o funcionamento do controle supervisório, o sistema foi posto em execução, no qual é possível observar como os supervisores selecionam valores seguros para a válvula de controle e bomba, a fim de manter a planta em uma região segura de operação. Em paralelo ao sistema supervisório, o controle PID age a fim de manter o nível do tanque próximo ao do setpoint. Nesse caso, o setpoint foi ajustado próximo ao limite HI_HI_Lim (79%) para testar a ação dos supervisores de maneira prática. A figura 27 apresenta os resultados obtidos com esse experimento, exibindo em azul o valor de setpoint escolhido, em vermelho a abertura da válvula e em verde o nível do tanque. O tempo de execução durou em torno de 8 minutos e os valores oscilaram entre 0 e 100%. No tempo zero, a abertura e o nível se encontram no valor mínimo. No instante que a chave seletora do painel é ativada para ligar o sistema, a válvula abre em 100% a fim de provocar o alcance do nível perto do setpoint. Por volta dos dois minutos e meio esse valor é alcançado, a válvula reduz a abertura e a bomba é desligada, provocando a queda do nível. A bomba só volta a ligar quando atinge a zona HI_Lim por volta dos 3 minutos, ocasionando a subida de nível do tanque até mais um sobre sinal por volta dos 3 minutos e meio, quando a bomba volta a desligar para garantir a segurança do tanque e evitar o overflow. Quando atinge a zona segura, a bomba volta a ligar pela última vez nesse ciclo e quando ela desliga, por volta dos 4 minutos, a chave seletora desliga o sistema. Nesse momento, mesmo que a chave seletora ligue novamente, o modelo vai procurar uma zona segura para religar a bomba, propriedade obtida dos modelos do controle supervisório. Isso vai ocorrer por volta dos 5 minutos quando a bomba é ligada e o nível volta a subir depois de atingir a zona LO_LIM. Em seguida a válvula volta a abrir de forma saturada para tentar alcançar o setpoint, mas esse é mudado para próximo de 20% para ser testado dessa vez o comportamento dos supervisores na tentativa de evitar o underflow. Por fim, quando o nível chega perto desse valor, a válvula satura novamente aberta para permitir mais entrada de água no tanque e dessa vez a simulação é

finalizada com essa situação em andamento.

Figura 27 – Comportamento do controle do sistema.

3.5 CONCLUSÃO DO CAPÍTULO

Este capítulo visou apresentar os conceitos por trás da rede FF que é utilizada nesse trabalho, trazendo a importância desse protocolo para as plataformas de petróleo. Além disso, foram apresentados os modelos utilizados na PD3 e em seguida a síntese dos supervisores do controle supervisório, a fim de obter um comportamento discreto não bloqueante e seguro em um processo industrial com controle PID distribuído em rede. Os resultados alcançados com a aplicação da teoria de controle supervisório permitiu apresentar a ação antecipatória, reativa e minimamente restritiva dos supervisores modulares.

4 MODELAGEM DA ESTRATÉGIA PARA DIAGNÓSTICO DE FALHAS

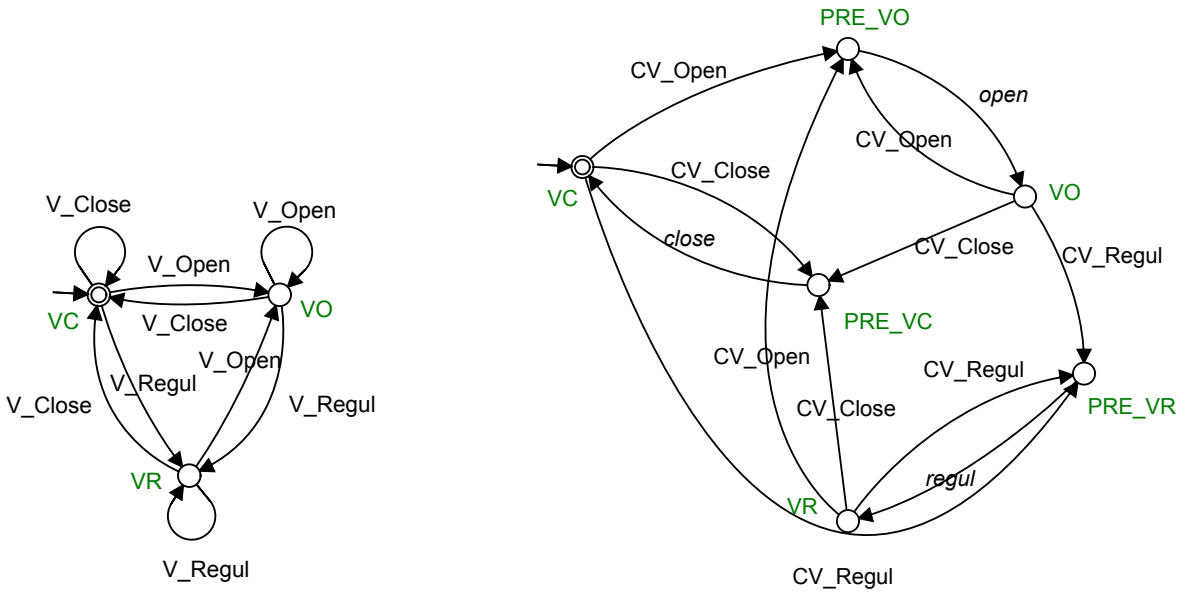
Em geral, o projeto de controle supervisório é realizado considerando apenas o comportamento nominal do sistema (OLIVEIRA, R. G.; QUEIROZ; CURY, 2020), levando aos modelos apresentados na capítulo anterior. Nesse contexto, ambos os comandos de controle e suas respectivas consequências imediatas nos módulos do sistema são modelados como um único evento. Além disso, no projeto de controle supervisório da planta, foi considerado que todos os eventos são observáveis, já que o projeto do controle não levava em consideração a possível ocorrência de falhas. Por exemplo, o evento V_Open do modelo da válvula G_{Valve}, representado na figura 23(e), modela tanto o comando de controle para abrir a válvula quanto sua consequente abertura. Porém, se por consequência de uma falha a válvula travar fechada, o comando de controle para abrir a válvula ainda pode ser enviado pelo controlador, porém, a válvula não abrirá fisicamente.

Esse capítulo trata das modificações necessárias no modelo de controle supervisório para considerar a falha de travamento fechada, e seu diagnóstico.

4.1 MODIFICAÇÃO DOS MODELOS PARA DIAGNÓSTICO

O primeiro passo para modelar o sistema visando o diagnóstico de falhas é separar os eventos em comandos de controle e suas consequências físicas. Nesse sentido, os eventos de comando são considerados controláveis, pois são produzidos pelo CLP, enquanto as consequências físicas são modeladas como não controláveis. Aplicando essa metodologia para o modelo da válvula apresentada na figura 28(a), obtém-se o modelo apresentado na figura 28(b). Nesse modelo, os eventos CV_Open, CV_Close e CV_Regul modelam os comandos para abrir, fechar e regular a válvula, respectivamente, que podem ser enviados pelo controlador. Os eventos $open$, $close$ e $regul$ modelam as consequências físicas da válvula devido a comandos anteriores enviados pelo controlador CV_Open, CV_Close e CV_Regul, respectivamente.

Neste trabalho, é considerado que a válvula do sistema pode falhar e, quando isso acontece, ela trava fechada. A figura 29 apresenta a estrutura e os componentes principais da válvula (SHANG; ZHANG, Y.; ZHANG, H., 2023). Essa falha foi escolhida porque a válvula é usada como o atuador da lógica de controle para regular o nível do líquido no tanque. Também optou-se por este tipo de falha devido às características mecânicas da válvula, pois, por se tratar de um atuador pneumático, não só a própria válvula pode falhar, mas problemas relacionados à baixa pressão no alimentador pneumático levam ao fechamento da válvula, que é indistinguível para a válvula estar permanentemente fechada. Portanto, o travamento da válvula, e a retenção de ar comprimido por mal funcionamento da linha pneumática, provocam as mesmas consequências. Essas características podem ser vistas na figura 29. Note que, caso falte ar no diafragma, seja por problemas no compressor de ar, ou defeito na vedação do compartimento do atuador, a válvula pode travar fechada devido à força elástica da mola ser maior do que a pressão de ar do outro lado do diafragma. Por outro lado, a fricção da haste com a gaxeta também



(a) (b)

Figura 28 – Aletração dos eventos da válvula para comando e consequência: (a) Modelo da válvula original com os eventos únicos. (b) Modelo da válvula modificado com os eventos de comando e consequência.

pode ser um motivo de um travamento se por algum motivo o atrito entre esses elementos for demasiadamente alto. Vale notar que o atrito é o problema de válvula mais comumente encontrado na indústria de processos (CHOUDHURY; THORNHILL; SHAH, 2005) (NAHID; IFTAKHER; CHOUDHURY, 2019). Em todas as situações mencionadas, a válvula seguiria para o comportamento de falha considerado neste trabalho.

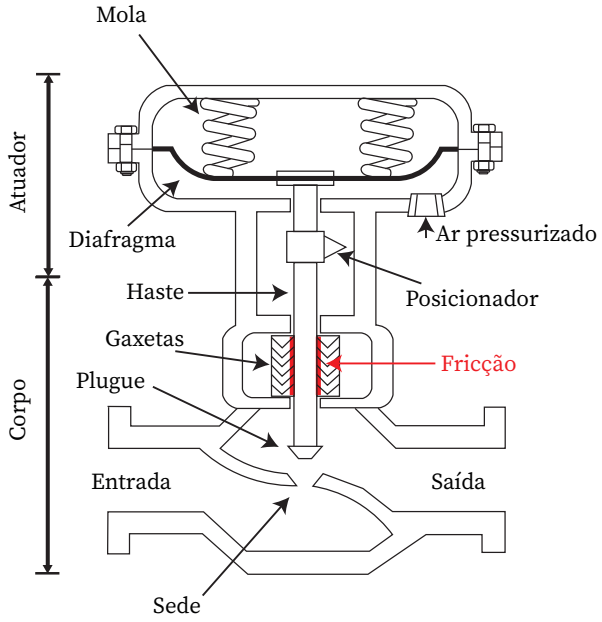


Figura 29 – Componentes da válvula (Figura do autor).

O modelo modificado para a válvula considerando o comportamento pós-falha é apre-

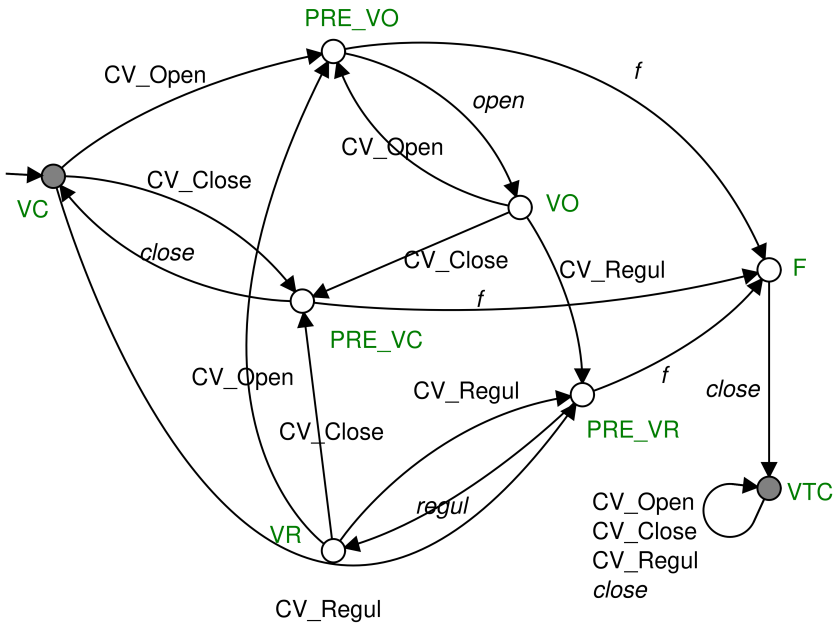


Figura 30 – G_{Valve} modificado para considerar o comportamento de falha da válvula.

sentado na figura 30. Neste modelo, o evento f é usado para representar o evento de falha e considera-se que a válvula pode falhar após qualquer evento de comando e, se isso acontecer, o modelo da válvula evolui para o estado F, em que apenas o evento $close$ pode ocorrer, o que leva o sistema para o estado VTC. No estado VTC, somente eventos de comando e o evento $close$ podem ocorrer, modelando os efeitos permanentes da ocorrência do evento de falha. Esse último evento está presente pelo fato do sistema necessitar da resposta do sistema quando é dado um comando, e a única resposta possível é fechar a válvula. Isso evita um *deadlock* no sistema, pois existem especificações que pedem que quando existe um evento de comando é necessário ver uma resposta. É importante destacar que as modificações aplicadas para representar os efeitos da ocorrência do evento de falha no modelo G_{Valve} são simples e podem ser reproduzidas em uma grande diversidade de atuadores industriais.

No entanto, o evento de falha f não é diagnosticável usando apenas o modelo G_{Valve} como esperado, pois não há informações suficientes para identificar a ocorrência da falha considerando apenas a dinâmica da válvula.

Para levar em consideração a interação do comportamento pós-falha da válvula com os demais componentes do sistema, é necessário modificar os modelos desses componentes para considerar os eventos de comando do controlador e suas consequências físicas introduzidos no modelo da válvula. Portanto, apenas os modelos dos componentes que compartilham eventos com o modelo da válvula devem ser modificados. O primeiro modelo a ser considerado é o G_{Flow}, que representa a interação entre a válvula, a bomba e o tanque. Por se tratar de um modelo da planta, os eventos V_Open, V_Close e V_Regul são substituídos pelos eventos de consequência da válvula, *open*, *close* e *regul*, respectivamente. Os demais eventos de G_Flow são mantidos, uma vez que neste trabalho é considerado que apenas a válvula pode falhar e, portanto, não

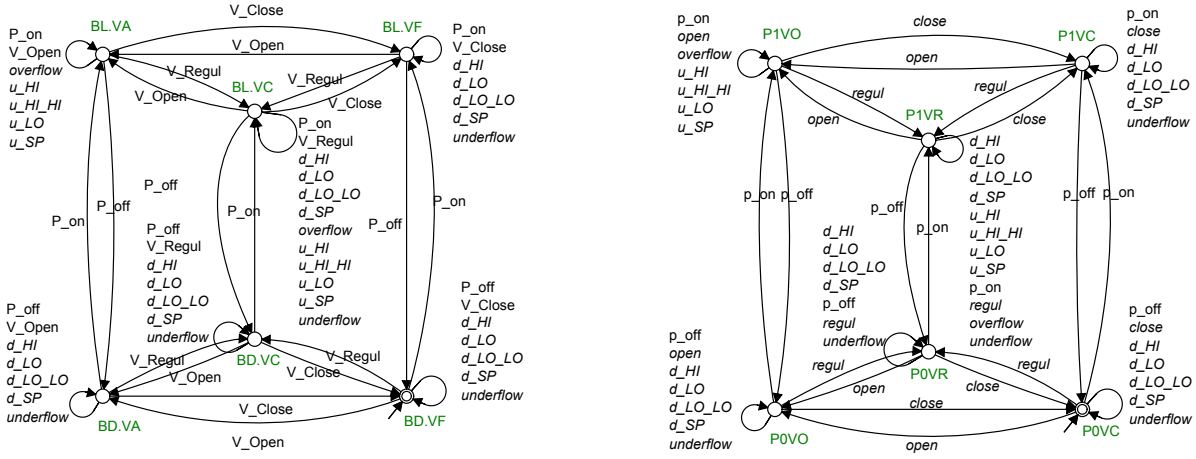


Figura 31 – Alteração dos eventos da válvula em G_{Flow}: (a) Modelo G_{Flow} original. (b) Modelo G_{Flow} modificado.

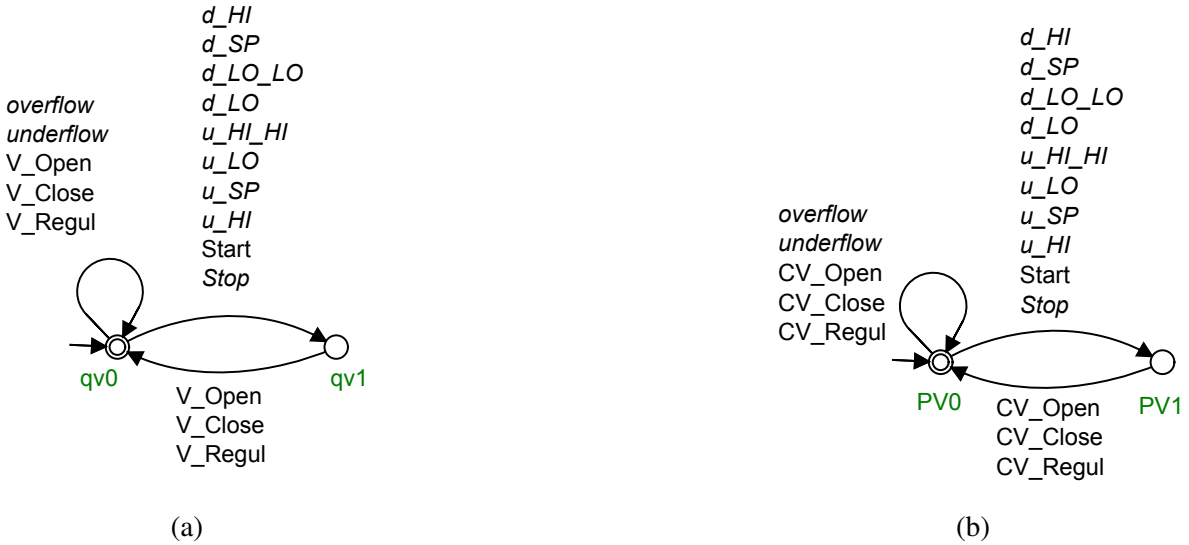


Figura 32 – Alteração dos eventos da válvula em G_{PV}: (a) Modelo G_{PV} original. (b) Modelo G_{PV} modificado.

é necessário considerar a diferença entre eventos de comando e consequência para os demais componentes do sistema. Os modelos original e modificado de G_{Flow} são apresentados nas figuras 31 (a) e 31 (b), respectivamente.

Na figura 32, é apresentado o modelo original e modificado de preempção de válvulas G_{PV}. Como esse modelo é usado para representar que pelo menos um comando de controle é sempre executado na válvula entre duas transições de nível consecutivas, apenas os eventos de comando CV_Open, CV_Close e CV_Regul são usados em substituição aos eventos V_Open, V_Close e V_Regul, respectivamente.

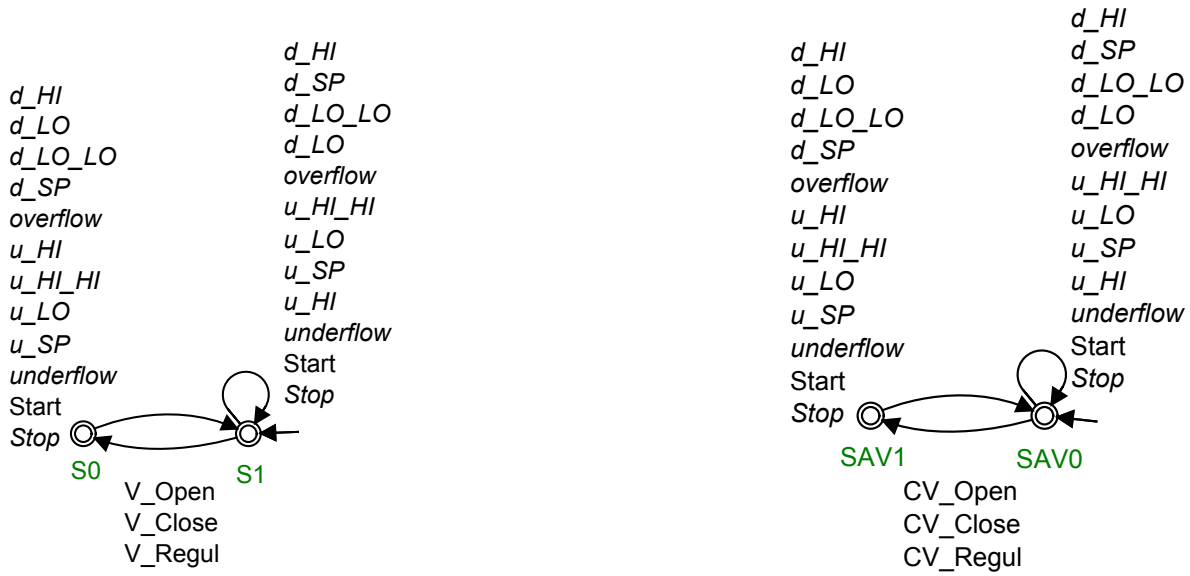
Na estrutura de controle supervisório, as especificações são usadas para calcular o supervisor, que pode desabilitar eventos para garantir que o sistema em malha fechada satisfaça essas especificações. Assim, nos modelos de especificação, eventos relacionados à válvula V_Open ,

V_Close e V_Regul são substituídos por CV_Open, CV_Close e CV_Regul, respectivamente, pois somente esses eventos podem ser desabilitados pelo controlador supervisório. Essas modificações são apresentadas nas figuras 33, 34 e 35 que ilustram as mudanças nos modelos das especificações E_{AV}, E_{IPV} e E_M, respectivamente.

Os modelos originais foram alterados para atender os critérios de diagnosticabilidade. Inicialmente, o modelo da válvula foi modificado inserindo o evento de comando e consequência. Em seguida, os rótulos dos eventos da válvula que fazem parte de outros autômatos também foram trocados. G_{PV} recebeu o evento de comando da válvula, G_{Flow} recebeu o evento de consequência da válvula, e as especificações receberam os eventos de comando da válvula (porque podem ser desabilitados). Os critérios na escolha dos eventos de comando ou de consequência para os modelos anteriormente citados, são apresentados a seguir:

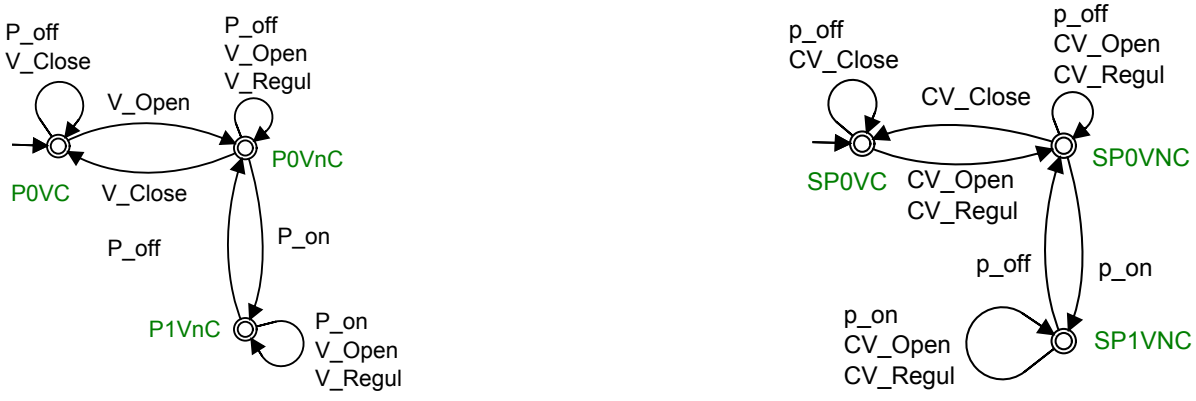
- E_{AV}: utiliza-se os eventos de comando, pois a especificação tem a premissa de esperar uma reação da planta antes de dar um novo comando. Ou seja, o evento de comando referido é o da própria válvula.
- G_{Flow}: utiliza-se os eventos de consequência, uma vez que só haverá vazão quando a válvula abrir, de fato.
- G_{PV}: considerando a premissa desse modelo que é a de que sempre há a ocorrência de um evento de comando entre a ocorrência de 2 eventos não controláveis, utiliza-se nesse caso os eventos de comando da válvula.
- E_{IPV}: Nessa especificação o sistema deve poder controlar a válvula, então deve ser um evento controlável de comando. Caso contrário, a válvula não seria impedida de fechar se a bomba estiver ligada.
- E_M: Seguindo a lógica anterior, utiliza-se o evento de comando pois esse evento tem que ser impedido de acontecer (abrir válvula) caso o sistema não esteja iniciado ainda.

As modificações propostas neste trabalho não alteram a controlabilidade do comportamento livre de falhas do sistema, $i.e.$, o supervisor ainda pode garantir que as especificações sejam atendidas, supondo que o evento de falha não tenha ocorrido, conforme comprovado realizando-se a síntese do supervisor modular considerando-se os modelos modificados no software SUPREMICA (AKESSON *et al.*, 2006). No entanto, notou-se que o modelo é não diagnosticável porque, apesar do objetivo de se evitar o evento de *underflow*, ele pode ocorrer durante o processo de inicialização do sistema, como é possível verificar no modelo E_M da figura 35 (a). Entretanto, esse evento não é esperado depois que o sistema é inicializado com sucesso, indicado pela primeira ocorrência do evento *u_SP* que modela que o sistema alcançou o *setpoint*. Portanto, é preciso diferenciar o comportamento de inicialização do comportamento em regime permanente. Neste trabalho, isso foi feito diferenciando-se os eventos de parada de tal maneira que, para que o sistema seja parado após a inicialização, tanto o controlador quanto o diagnosticador devem ser reinicializados. No controle supervisório isso já é feito com



(a) (b)

Figura 33 – Alteração dos eventos da válvula em E_{AV}: (a) Modelo E_{AV} original. (b) Modelo E_{AV} modificado.



(a) (b)

Figura 34 – Alteração dos eventos da válvula em E_{IPV}: (a) Modelo E_{IPV} original. (b) Modelo E_{IPV} modificado.

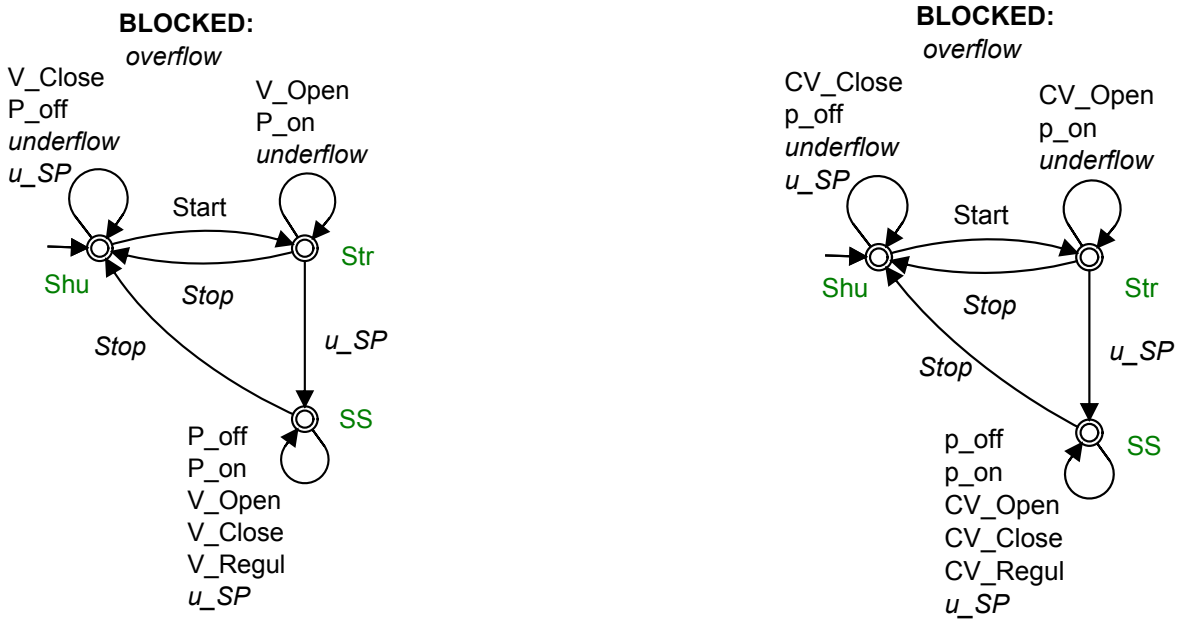


Figura 35 – Alteração dos eventos da válvula em E_M: (a) Modelo E_M original. (b) Modelo E_M modificado.

o evento *stop*, e no caso do diagnosticador, foi considerado que, caso o sistema seja parado, o diagnosticador é reiniciado e, portanto, não é necessário prever a ocorrência de *stop* no modelo dos modos de operação, levando ao modelo de E_m apresentado na figura 36, em que, após o sistema alcançar o primeiro set point, ele pode evoluir para o modo de falha com a ocorrência do evento f.

Em resumo, para atingir este modelo, duas hipóteses são feitas: (i) a inicialização do sistema é concluída após a primeira ocorrência do evento u_SP, que modela que o nível do tanque de líquido atingiu o set-point pela primeira vez, e o evento (ii) $Stop$ modela uma inicialização malsucedida do sistema devido a um problema identificado pelo operador. Assim, o evento de falha é modelado somente após a primeira ocorrência do evento u_SP no modelo de autômato da figura 36. Observe que uma parada de emergência para naturalmente o sistema e o leva ao estado inicial, na qual uma nova inicialização é necessária. Assim, não foi modelada esta situação para fins de diagnóstico e é considerada que uma parada de emergência após o sistema ter sido inicializado com sucesso, também deve zerar o diagnosticador.

É importante ressaltar que o procedimento de diagnóstico é passivo e executado em paralelo com o sistema controlado. Assim, as modificações propostas neste trabalho para os modelos considerados no projeto de controle supervisório não interferem no comportamento do sistema em malha fechada. Além disso, a linguagem gerada pelo modelo global G calculado fazendo a composição paralela de todos os modelos de componentes do sistema é diagnosticável em relação ao evento de falha f. Todos os eventos, com exceção do evento de falha f, são considerados observáveis, uma vez que a planta é instrumentada com dispositivos conectados

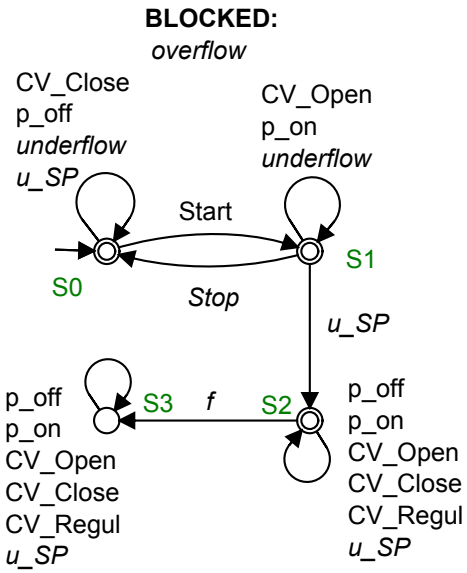


Figura 36 – Especificação dos modos de operação para diagnóstico.

através de uma rede industrial FF, que permite que tanto comandos de controle quanto leituras de sensores estejam disponíveis para um diagnosticador implementado no CLP. Entretanto isso não significa que o sistema é trivial, pois a planta tem válvulas manuais, e se a falha for alguma outra coisa que o sensor não acuse o travamento fechado, como obstrução do tubo ou falta de ar comprimido, a falha é diagnosticada sem necessariamente depender do evento observável de abertura da válvula.

Para facilitar a compreensão das metodologia desenvolvida no trabalho de alteração dos modelos para o diagnóstico, é proposto um conjunto de regras descrita pelo algoritmo 4.1.1 que é apresentado a seguir:

Algoritmo 4.1.1 *Possuindo o modelo de controle supervisório que obedece o comportamento desejado, sem consideração de qualquer falha, as modificações são feitas da seguinte forma:*

- *Passo 1: Divida os eventos do modelo do atuador em: comando e consequência.*
- *Passo 2: Adicione a falha e o comportamento pós-falha no atuador.*
- *Passo 3: Modifique os eventos do atuador nos demais modelos obedecendo os seguintes critérios,*
  - *Passo 3.1: Para modelos da planta, ou seja, que envolva a resposta física do sistema, utilize os eventos de consequência.*
  - *Passo 3.2: Para modelos de preempção, ou seja, que envolva o envio de comando do atuador, utilize os eventos de comando.*
  - *Passo 3.3: Para modelos de especificação, ou seja, que envolva habiliar ou desabilitar eventos, utilize os eventos de comando.*

Tabela 6 – Modelos da planta com número de estados, eventos e transições após as modificações.

Subsistema	**Estados**	**Eventos**	**Transições**
Gbomba	2	2	4
Gbotão	2	2	2
Gníveis	5	10	10
Gpb	2	14	16
Gpv	2	15	18
Gválvula	6	6	12
G	960	20	3880

Na próxima seção são apresentados os cálculos do controle supervisório utilizando os modelos modificados, com o objetivo de apresentar o funcionamento original desejado.

4.2 SÍNTESE DOS SUPERVISORES

Com as modificações realizadas, os modelos de supervisores que foram calculados no sistema original, precisaram ser recalculados para atender as novas necessidades de especificação.

O modelo geral da planta é obtido pela composição paralela de todas as plantas, ou seja, $G = G_{niveis} \parallel G_{botão} \parallel G_{bomba} \parallel G_{PV} \parallel G_{PV} \parallel G_{válvula} \parallel G_{vazão}$ resultando em um autômato com 960 estados. Nesse caso, a Tabela 6 apresenta os modelos da planta com o número de estados, eventos e transições, assim como o resultado da composição paralela G. Em relação a Tabela 3 do capítulo anterior, foi alterado os valores da válvula e do autômato resultante da composição paralela dos subsistemas.

Os eventos de comando e consequência da válvula, foram portanto, não só mudados no modelo $G_{válvula}$ da válvula, como também nos modelos G_{PV},G_{flow},E_{AV},E_{IPV} e E_M no qual antes eles apareciam. Na sequência, é realizada a síntese dos supervisores com base nos novos modelos. A opção para diminuir a complexidade do supervisor monolítico volta a ser a utilização do controle modular e, como visto, a ideia deste é obter um supervisor para cada especificação, com o objetivo de alcançar um comportamento de malha fechada. Ele utiliza um supervisor para cada especificação e suas plantas correspondentes, quando não existe eventos em comum. No nosso caso, como as especificações compartilham eventos em comum, é necessário utilizar a planta todas as especificações.

As três especificações E_{AV}, E_{AP} e E_{IPV} já são verificadas controláveis e não bloqueantes em relação a planta G, então, esses modelos podem ser utilizados diretamente como supervisores modulares S_{AV}, S_{AP} e S_{IPV}, respectivamente. Isso é verificado pois para uma dada composição paralela da especificação com a planta, gera o mesmo número de estados que a síntese $SupC$ desses mesmos modelos. Já a especificação E_M requer a síntese de um supervisor $SupC$ que apresenta o maior comportamento permissivo para ser implementado com a especificação E_M garantida. Dado que o SupC(EM,G) resultou em um autômato com 632 estados, foi aplicado um método para reduzir o número de estados do supervisor, dado por (SU; WONHAM, 2004) o qual resultou em um autômato S_{mR} de 22 estados. O número de estados de cada modelo da

Tabela 7 – Modelos das especificações utilizados com respectivos número de estados reduzidos.

x	E_x	G_x	K_x	S_x	S_{xr}
Av	2	960	1044	1044	2
Ab	2	960	1032	1032	2
Ibv	3	960	780	780	3
M	3	960	788	632	22

Tabela 8 – Supervisores que realizam o novo controle da Planta.

Modelo	**Estados**	**Eventos**	**Transições**
Sab	2	14	26
Sav	2	15	27
Sibv	3	5	13
Smr	22	19	158
Coordenador	24	17	202

especificação, da planta K_x, $\mathbf{S}_x$ e $\mathbf{S}_{xR}$ são apresentados na Tabela 7, em que:

- E_x é o número de estados da especificação pura
- G_x é o número de estados da composição paralela dos autômatos da planta que possuem eventos em comum com essa especificação
- K_x é a composição paralela de $E_{,x}$ com $G_{,x}$
- S_x é o supervisor que é calculado a partir das plantas e especificações (synthesize)
- S_{rx} é o supervisor reduzido que pode ser o mesmo que a especificação ou calculado através de alguma ferramenta computacional (TCT)

Observação: quando K_x é igual a S_x, o S_{rx} recebe E_x pois indica que K_x não gerou nem mau-estado e nem bloqueio. Do contrário, S_{rx} deve ser calculado utilizando alguma ferramenta computacional.

Os eventos desabilitados de cada supervisor são aqueles que estão no alfabeto mas não estão definidos no estado.

Em geral, a conjunção de supervisores modulares não bloqueantes, se torna bloqueante. Para preservar as boas propriedades do controle modular é calculado um coordenador que desabilita os eventos que podem resultar em bloqueio. O sistema em malha fechada se torna, portanto, $G_{Conf} = G \parallel \mathbf{S}_{AVR} \parallel \mathbf{S}_{ABR} \parallel \mathbf{S}_{IPVR} \parallel \mathbf{S}_{MR}$ e apresenta 510 estados. O coordenador é resolvido para o conflito global, que após o processo de redução, apresenta 24 estados. Por curiosidade, o supervisor monolítico para resolução do sistema com a máxima linguagem controlável encontrada, apresentaria 489 estados, ou 63 após redução. Optou-se pela utilização dos supervisores modulares, como apresentado na Tabela 8, que somando o número de estados apresenta um número inferior ao monolítico.

Em seguida após a obtenção dos modelos dos supervisores que farão o controle da planta, é necessário calcular a diagnosticabilidade para verificar se o sistema é capaz de diagnosticar a falha que foi inserida no sistema.

4.3 CÁLCULO DA DIAGNOSTICABILIDADE E SÍNTESE DO DIAGNOSTICADOR

A diagnosticabilidade é calculada utilizando a malha fechada de $G \parallel \mathbf{S}_{AVR} \parallel \mathbf{S}_{ABR} \parallel \mathbf{S}_{IPVR} \parallel \mathbf{S}_{MR} \parallel G_{Conf} \parallel E_{mf1}$ e a especificação da falha. Para o cálculo da diagnosticabilidade foi utilizado o software DESUMA desenvolvido na Universidade de Michigan (RICKER; LAFORTUNE; GENC, 2006) que implementa as principais técnicas de diagnóstico de falhas iniciadas em (SAMPATH *et al.*, 1996) e apresentadas na Seção 2.3.2.2, incluindo rotinas de construção e análise de autômatos diagnosticadores. O DESUMA analisa e controla modelos de SEDs como FSA (autômatos de estado finitos). Dentro dele existe o UMDES que é uma biblioteca de rotinas para criar e manipular SEDs modelados como FSA. Os comandos principais do UMDES estão incorporados ao DESUMA e os usuários podem executar esses comandos a partir da linha de comando. Entretanto, na pasta do UMDES também existe todos os executáveis em que é necessário apenas dar o caminho do arquivo no *prompt*.

Assim, verifica-se que o sistema é diagnosticável e o diagnosticador obtido resulta em 827 estados, 21 eventos e 2394 transições. O resultado pode ser conferido no link: `https://github.com/eriquemoser/mestrado` onde foi disponibilizado o arquivo referente ao diagnosticador obtido pelo software *DESUMA*. Através desse arquivo é possível a visualização gráfica do autômato e também uma análise particular do leitor.

4.4 CONCLUSÃO DO CAPÍTULO

Neste capítulo foram apresentadas as modificações realizadas nos autômatos do capítulo anterior, com o objetivo de calcular um novo supervisor e um diagnosticador para diagnosticar a falha de travamento de válvula fechada. Foram apresentados os procedimentos para, a partir de modelos utilizando eventos de falha não observáveis, obter o mesmo funcionamento dado pela proposta de controle supervisório. A partir de um novo conjunto de modelos respeitando o comportamento desejado, foi calculada a diagnosticabilidade e calculado um diagnosticador para ser utilizado em paralelo ao sistema de controle existente.

A forma de trabalhar - modelar a falha em um único componente - foi escolhida mediante a situação de utilização do controle supervisório. Existem casos que precisam modelar a falha em múltiplos componentes, mas que provavelmente o sistema seja não controlável. Já que o sistema necessita de um controlador e um supervisor, o cálculo deve ser refeito com as modificações das especificações e com o acréscimo do pós falha. Entretanto essa dificuldade não foi considerada grande, em relação ao cálculo dos modelos sem a consideração da falha. O ponto que demandou mais atenção, foi encontrar o modelo E_{mf1} que necessitava da utilização para tornar a planta diagnosticável.

No capítulo seguinte, as conclusões que fazem esse trabalho ser relevante no meio científico e industrial, além das ideias de projetos futuros que podem ser trabalhadas a partir dessa dissertação, são apresentadas.

5 CONCLUSÃO

Processos industriais utilizam sistemas e equipamentos que estão sujeitos à falha. Tendo em vista este fator, nesta dissertação, foi apresentado um método para aplicar um diagnosticador em um sistema voltado à IPG que já apresenta um controle supervisório em seu funcionamento, e é modelado por autômatos. Esse diagnóstico de falhas é apontado dentro das Camadas de Segurança, onde sugere-se uma classificação entre a camada de Intervenção da Operação e Sistemas Instrumentados de Segurança. A aplicação da metodologia se mostrou promissora para qualquer indústria que possui sistema crítico responsável pela segurança da planta, que têm em seu funcionamento, algum sensor ou atuador que não necessariamente possui eventos observáveis para determinação de falha.

A metodologia proposta foi desenvolvida com o objetivo de integrar-se à atual metodologia de desenvolvimento de sistemas de segurança na IPG. Ela visa adicionar uma etapa de diagnóstico para indicar a ocorrência de um evento de falha que pode causar um desvio do comportamento esperado do sistema. O diagnóstico de falhas é um método realizado de maneira sistemática. Utilizou-se a Teoria de Sistemas a Eventos Discretos, que apresenta as ferramentas matemáticas para representar evolução de sistemas utilizados nesse trabalho. Também foi utilizado softwares como o *TCT*, *Supremica* e *Desuma* para sintetizar supervisores e realizar operações com autômatos. Quando o diagnóstico é realizado em conjunto com o controle supervisório, este proporciona a detecção e correção dos erros antes da ocorrência de um incidente, economizando tempo e dinheiro no processo de elaboração de um sistema de segurança.

Para encontrar um modelo de diagnóstico escrito em linguagem formal, foi proposto, neste trabalho, a modificação dos autômatos originais do sistema que passaram por uma série de transformações. Os programas anteriormente citados possibilitaram aplicar as técnicas de diagnóstico de falhas para encontrar a diagnosticabilidade do sistema. O conjunto de passos desenvolvido, possibilita um roteiro para aplicação dessas técnicas em sistemas com características similares. O procedimento de diagnóstico é executado em paralelo com o sistema controlado. As modificações propostas neste trabalho para os modelos considerados no projeto de controle supervisório não interferem no comportamento do sistema em malha fechada

Para manter o sistema controlável, os modelos foram modificados mantendo as propriedades de controlabilidade, ou seja, permitindo ao supervisor desativar os eventos controláveis e mantendo os eventos não-controláveis em execução sempre que precisarem evoluir. Foi criado um *script* que traduz as propriedades expressas dos modelos em regras adotadas para tornar a planta diagnosticável. A partir dos modelos modificados, foi possível utilizar o *software* TCT para síntese dos supervisores e do coordenador, objetivando manter a mesma dinâmica de funcionamento dos reservatórios para utilização nas ferramentas de cálculo de diagnosticabilidade. Através de análise no *software Supremica* foi possível contornar problemas encontrados através da análise de sequências. Os resultados gerados pelas sequências, para as propriedades de diagnosticabilidade, foram úteis para a criação de um novo modelo. Foi necessário, portanto,

diferenciar o comportamento de inicialização do comportamento em regime permanente. Com essa modificação, foi possível utilizar o *software Desuma* para constatar que o exemplo estudado é diagnosticável e passível de diagnóstico.

A metodologia proposta foi testada com sucesso, ao ser aplicada em um exemplo na Planta Didática III, da fabricante Smar, em uma implementação do controle de nível de um reservatório simulando um ambiente industrial. O exemplo didático foi útil para entender a importância das etapas de modificação e definir de forma correta as falhas nos modelos, sendo assim, este exemplo demonstra a eficácia do diagnóstico de falhas.

A fim de tornar a pesquisa mais completa, ainda é necessário a verificação da metodologia para múltiplas falhas a fim de verificar se a consequência pode continuar sendo modelada em um único componente. No exemplo mostrado, apenas a falha de travamento fechada foi realizada, porém acredita-se que procedimento similar possa ser adotado para diagnóstico da falha de travamento aberto. Vale destacar que este trabalho contribui para o aumento da pesquisa de segurança para o emprego de milhares de trabalhadores que lidam com a periculosidade de equipamentos industriais, não só no setor de IPG, mas se mostrou promissor para outras áreas e processos industriais. Evitar um acidente de grande proporção, significa poupar vidas, tragédias ambientais e perdas financeiras. Os resultados dessa pesquisa gerou um artigo em congresso internacional que foi apresentado em novembro do presente ano (MOSER *et al.*, 2023).

Com a conclusão deste trabalho, novas perspectivas e projeções para trabalhos futuros surgiram. A partir dos resultados desta pesquisa, pode ser feito a completação deste trabalho através dos seguintes acréscimos e atividade futuras:

- Considerar outros tipos de falhas que podem ocorrer no sistema;
- Explorar outras arquiteturas de diagnóstico diferentes da abordagem monolítica para aproveitar a modularidade natural dos sistemas industriais complexos;
- Explorar a generalização dessa modelagem para sistemas automatizados, permitindo maior aplicabilidade da solução.

REFERÊNCIAS

AKESSON, Knut; FABIAN, Martin; FLORDAL, Hugo; MALIK, Robi. Supremica-an integrated environment for verification, synthesis and simulation of discrete event systems. *In*: IEEE. 2006 8th International Workshop on Discrete Event Systems. [*S.l.*: *s.n.*], 2006. p. 384–385.

BACOVIS, Otávio Vinicius. **Comparação da utilização do controlador fuzzy e PID aplicados em um uma planta didática de nível de líquido**. 2016. Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná.

BONAFIN, Ana Caroline; CABRAL, Felipe Gomes; MOREIRA, Marcos Vicente. An effective approach for fault diagnosis of Discrete-Event Systems modeled as safe labeled Petri nets. **Control Engineering Practice**, v. 123, p. 105168, 2022.

BRANDÃO, Dennis. **Bloco funcional para controle fieldbus por variáveis de estado**. 2000. Tese (Doutorado) – Universidade de São Paulo.

CABRAL, Felipe Gomes; MOREIRA, Marcos Vicente. Synchronous diagnosis of discrete-event systems. **IEEE Transactions on Automation Science and Engineering**, v. 17, n. 2, p. 921–932, 2019.

CABRAL, Felipe Gomes; MOREIRA, Marcos Vicente; DIENE, Oumar; BASILIO, João Carlos. A Petri net diagnoser for discrete event systems modeled by finite state automata. **IEEE Transactions on Automatic Control**, v. 60, n. 1, p. 59–71, 2015.

CASSANDRAS, C. G.; LAFORTUNE, Stéphane. **Introduction to discrete event systems**. 2. ed. [*S.l.*]: Springer, 2008.

CHOUDHURY, MAA Shoukat; THORNHILL, Nina F; SHAH, Sirish L. Modelling valve stiction. **Control engineering practice**, Elsevier, v. 13, n. 5, p. 641–658, 2005.

CNI, Perfil da Indústria. **Composição setorial: participação percentual do setor no PIB industrial**. [*S.l.*: *s.n.*], 2019. Disponível em: `https://perfildaindustria.portaldaindustria.com.br/composicao_setorial`. Acesso em: 18 jul. 2022.

COCO, James C. **The 100 largest losses 1972-2001: Large property damage losses in the hydrocarbon-chemical industries**. [*S.l.*]: Marsh Risk Consulting, 2003.

CONSULTOR JURIDICO. **STJ mantém multa contra Petrobras por acidente na plataforma P-36**. [*S.l.*: *s.n.*], 2023. Disponível em: `https://www.conjur.com.br/2023-set-13/stj-mantem-multa-petrobras-acidente-plataforma-36/`. Acesso em: 13 nov. 2023.

CONTROLE & AUTOMAÇÃO. **Indústria de óleo e gás: automação aliada à competitividade**. v. 14. [*S.l.*]: Valete Editora, 2000. Disponível em: http://controleinstrumentacao.com.br/arquivo/ed_50/ed_50a.html. Acesso em: 20 set. 2023.

CORDEIRO, Maurício. **A Instrumentação protegendo os processos (Segurança de Processo) parte I**. [*S.l.*: *s.n.*], 2021. Disponível em: https://www.dicasdeinstrumentacao.com/a-instrumentacao-protegendo-os-processos-seguranca-de-processo/. Acesso em: 10 jul. 2022.

CURY, Jose E. R. Teoria de controle supervisório de sistemas a eventos discretos. **V Simpósio Brasileiro de Automação Inteligente (Minicurso)**, p. 8, 2001.

DIAS, Mônica Aparecida. **(Dissertação) Automação de uma Unidade de Experimentação de Escoamento Multifásico utilizando Tecnologia FOUNDATION FIELDBUS**. [*S.l.*], 2015.

EPE, Empresa de Pesquisa Energética (Brasil). Brazilian Energy Balance 2021 year 2020. **Brazilian Energy Balance 2021 year 2020**, p. 292, 2021.

FIELDBUS FOUNDATION. **Function Block Capabilities in Hybrid/Batch Applications**. [*S.l.*]: Field. Foun. App. Guide, 2002.

FIGUEIREDO, Marcelo Gonçalves; ALVAREZ, Denise; ADAMS, Ricardo Nunes. O acidente da plataforma de petróleo P-36 revisitado 15 anos depois: da gestão de situações incidentais e acidentais aos fatores organizacionais. **Cadernos de Saúde Pública**, SciELO Brasil, v. 34, e00034617, 2018.

GOMES, N. C.; NICACIO, J. V.; TÔRRES, A. G. Planta didática SMAR PD3: Modelagem, simulação e ajuste dos parâmetros do controlador de temperatura do tanque de mistura. **The Journal of Engineering and Exact Sciences**, Universidade Federal de Vicosa, v. 3, n. 7, p. 0933–0954, 2017.

GRABOIS, Ana Paula. **Explosão na P-36 teve erros de projeto, manutenção e operação**. [*S.l.*: *s.n.*], 2001. Disponível em: https://www1.folha.uol.com.br/folha/cotidiano/ult95u33144.shtml. Acesso em: 13 nov. 2023.

IEMA, Instituto de Energia e Meio Ambiente. **Crise hídrica, termelétricas e renováveis: Considerações sobre o planejamento energético e seus impactos ambientais e climáticos**. [*S.l.*: *s.n.*], 2021. P. 7. Disponível em: https://energiaeambiente.org.br/wp-content/uploads/2021/09/IEMA_crisehidricatermeletricas.pdf. Acesso em: 29 nov. 2023.

KLEIN, Stéphane; LITZ, Lothar; LESAGE, Jean-Jacques. Fault detection of discrete event systems using an identification approach. **IFAC Proceedings Volumes**, v. 38, n. 1, p. 92–97, 2005.

LIMA, Públio Macedo Monteiro; CARVALHO, Lilian Kawakami; MOREIRA, Marcos Vicente. Ensuring confidentiality of cyber-physical systems using event-based cryptography. **Information Sciences**, v. 621, p. 119–135, 2023.

MARTINS, Ana Elisa Araújo. **Diagnose de falhas de uma unidade de separação trifásica usando modelos a eventos discretos**. 2018. Monografia (TCC (Graduação)) – Universidade Federal do Rio de Janeiro.

MC MACHADO, Thiago H de; VIANA, Gustavo da Silva; MOREIRA, Marcos Vicente. Event-Based Automaton Model for identification of discrete-event systems for fault detection. **Control Engineering Practice**, v. 134, p. 105474, 2023.

MDIC, Ministério do Desenvolvimento Indústria e Comércio Exterior. **Energia**. [*S.l.*: *s.n.*], 2022. Disponível em: `http://mdic.gov.br/index.php/comercio-exterior/contatos/9-assuntos/categ-comercio-exterior/599-energia`. Acesso em: 18 jul. 2022.

MOREIRA, Marcos Vicente; BASILIO, João Carlos; CABRAL, Felipe Gomes. "Polynomial time verification of decentralized diagnosability of discrete event systems" versus "Decentralized failure diagnosis of discrete event systems": A critical appraisal. **IEEE Transactions on Automatic Control**, v. 61, n. 1, p. 178–181, 2015.

MOREIRA, Marcos Vicente; JESUS, Thiago C; BASILIO, João Carlos. Polynomial time verification of decentralized diagnosability of discrete event systems. **IEEE Transactions on Automatic Control**, v. 56, n. 7, p. 1679–1684, 2011.

MOREIRA, Marcos Vicente; LESAGE, Jean-Jacques. Fault diagnosis based on identified discreteevent models. **Control Engineering Practice**, Elsevier, v. 91, p. 104101, 2019.

MOSER, Érique; CABRAL, Felipe Gomes de Oliveira; LIMA, Públio Macedo Monteiro; QUEIROZ, Max Hering de. Modeling of a hierarquical supervisory controlled industrial process for fault diagnosis. *In*: 2023 15th IEEE International Conference on Industry Applications (INDUSCON). São Paulo: [*s.n.*], nov. 2023.

MULER, Otávio Polonio. **Síntese e implementação de controle supervisório de processos industriais com malha de válvulas**. 2018. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

NAHID, Ahaduzzaman; IFTAKHER, Ashfaq; CHOUDHURY, MAA Shoukat. Control Valve Stiction Compensation-Part I: A New Method for Compensating Control Valve Stiction. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 58, n. 26, p. 11316–11325, 2019.

NUNES, Carlos Eduardo Viana. **Sistema Inteligente de Suporte Operacional em Processos de Tratamento Primário de Petróleo**. 2012. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

NUNES, Carlos Eduardo Viana; MOREIRA, Marcos Vicente; ALVES, Marcos V. S.; CARVALHO, Lilian Kawakami; BASILIO, João Carlos. Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. **Discrete Event Dynamic Systems**, v. 28, p. 215–246, 2018.

OLIVEIRA, Rafael G.; QUEIROZ, Max Hering de; CURY, Jose E. R. Synthesis of Supervisors for a PID-Controlled Industrial Process and Implementation on Foundation Fieldbus. **IFAC-PapersOnLine**, v. 53, n. 4, p. 83–88, 2020. ISSN 24058963.

OLIVEIRA, Vinicius Souza Lima de; CABRAL, Felipe Gomes; MOREIRA, Marcos Vicente. K-loss robust codiagnosability of Discrete-Event Systems. **Automatica**, v. 140, p. 110222, 2022.

PANTONI, Rodrigo Palucci. **Desenvolvimento e implementação de uma descrição de dispositivos aberta e não-proprietária para equipamentos FOUNDATION fieldbus baseada em XML**. 2006. Tese (Doutorado) – Universidade de São Paulo.

PETROBRAS. **Acidente da P-36 - Explosão e Naufrágio**. [*S.l.*: *s.n.*], 2015. Disponível em: `https://www.youtube.com/watch?v=Oz10Rsw_bJc&ab_channel=MegadaEngenhariaBR`. Acesso em: 13 nov. 2023.

PETRONOTÍCIAS. **Plataforma da Petrobrás retoma produção após incidente**. [*S.l.*: *s.n.*], 2014. Disponível em: `https://petronoticias.com.br/plataforma-da-petrobras-retoma-producao-apos-incidente/`. Acesso em: 25 set. 2023.

PRATI, Thiago Javaroni. **Desenvolvimento de técnicas formais de diagnóstico de falhas para indústria de petróleo e gás**. 2011. Relatório (Iniciação Científica) – Universidade Federal de Santa Catarina.

QUEIROZ, Max Hering de; CURY, Jose E. R. Modular multitasking supervisory control of composite discrete-event systems. **IFAC Proceedings Volumes**, Elsevier, v. 38, n. 1, p. 91–96, 2005.

QUEIROZ, Max Hering de; CURY, Jose E. R. Synthesis and implementation of local modular supervisory control for a manufacturing cell. *In*: IEEE. SIXTH International Workshop on Discrete Event Systems, 2002. Proceedings. [*S.l.*: *s.n.*], 2002. p. 377–382.

RAMADGE, Peter J.C.; WONHAM, W. M. The Control of Discrete Event Systems. **Proceedings of the IEEE**, v. 77, n. 1, p. 81–98, 1989. ISSN 15582256.

REIS, Luiz Paulo Enadio dos. **Verificação formal de Sistemas Instrumentados de Segurança na indústria de petróleo e gás natural**. 2018. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

RICKER, Laurie; LAFORTUNE, Stéphane; GENC, S. Desuma: A tool integrating giddes and umdes. *In*: IEEE. 2006 8th international workshop on discrete event systems. [*S.l.*: *s.n.*], 2006. p. 392–393.

SAMPATH, Meera; SENGUPTA, Raja; LAFORTUNE, Stéphane; SINNAMOHIDEEN, Kasim; TENEKETZIS, Demosthenis. Diagnosability of discrete-event systems. **IEEE Transactions on automatic control**, v. 40, n. 9, p. 1555–1575, 1995.

SAMPATH, Meera; SENGUPTA, Raja; LAFORTUNE, Stéphane; SINNAMOHIDEEN, Kasim; TENEKETZIS, Demosthenis C. Failure diagnosis using discrete-event models. **IEEE Transactions on Control Systems Technology**, v. 4, n. 2, p. 105–124, 1996. ISSN 10636536.

SAMPATH, Meera; SINNAMOHIDEEN, Kasim; LAFORTUNE, Stéphane; TENEKETZIS, Demosthenis. Diagnosability of Discrete-Event Systems. **IEEE Transactions on Automatic Control**, v. 40, n. 9, p. 1555–1575, 1995. ISSN 15582523.

SANTOS, Talysson Manoel de Oliveira; BRAGA, Márcio Feliciano; SARAVIA, Edgard Gregory Torres. Técnicas de sintonia de PID no controle de vazão e nível na planta didática SMAR PD3 Foundation Fieldbus. October, 2017.

SHANG, Linyuan; ZHANG, Yuyu; ZHANG, Hanyuan. Valve Stiction Detection Method Based on Dynamic Slow Feature Analysis and Hurst Exponent. **Processes**, Multidisciplinary Digital Publishing Institute, v. 11, n. 7, p. 1913, 2023.

SIMAS, Adller. **Síntese e Implementação de Controle Supervisório para Segurança de um Processo Característico da Indústria de Petróleo e Gás**. 2015. Monografia (TCC (Graduação Eng. Elétrica)) – Universidade Federal de Santa Catarina.

SMAR. **O Livro de Referências para Fieldbus**. [*S.l.*: *s.n.*], 2008. P. 63. Disponível em: `https://www.smar.com/public/img/dropzone/arquivos/rbfbbp_14_160.pdf`. Acesso em: 24 jun. 2022.

SU, Rong; WONHAM, W. M. Supervisor reduction for discrete-event systems. **Discrete Event Dynamic Systems**, Springer, v. 14, p. 31–53, 2004.

SUMMERS, Angela E. Introduction to layers of protection analysis. **Journal of hazardous materials**, Elsevier, v. 104, n. 1-3, p. 163–168, 2003.

VERAS, Maria ZM; CABRAL, Felipe Gomes; MOREIRA, Marcos Vicente. Distributed synchronous diagnosis of discrete event systems modeled as automata. **Control Engineering Practice**, v. 115, p. 104892, 2021.

WONG, Kai C; WONHAM, W. M. Modular control and coordination of discrete-event systems. **Discrete Event Dynamic Systems**, Springer, v. 8, p. 247–297, 1998.

WONHAM, W. M.; CAI, Kai *et al.* **Supervisory control of discrete-event systems**. [*S.l.*]: Springer, 2019.

WONHAM, W. M.; RAMADGE, Peter J.C. Modular supervisory control of discrete-event systems. **Mathematics of control, signals and systems**, Springer, v. 1, n. 1, p. 13–30, 1988.