

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

HUDSON DOS SANTOS LAPA

**UMA METODOLOGIA BASEADA EM CONTROLE SUPERVISÓRIO PARA
AUXILIAR NA RECONFIGURAÇÃO DE SMART GRIDS.**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2017

HUDSON DOS SANTOS LAPA

**UMA METODOLOGIA BASEADA EM CONTROLE
SUPERVISÓRIO PARA AUXILIAR NA RECONFIGURAÇÃO DE
SMART GRIDS.**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia Elétrica do Departamento Acadêmico de Elétrica – DAELE – da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. César Rafael Claire Torrico

Coorientador: Prof. Dr. Marcelo Teixeira

PATO BRANCO

2017

TERMO DE APROVAÇÃO

O trabalho de Conclusão de Curso intitulado “**UMA METODOLOGIA BASEADA EM CONTROLE SUPERVISÓRIO PARA AUXILIAR NA RECONFIGURAÇÃO DE SMART GRIDS**”, do aluno **HUDSON DOS SANTOS LAPA** foi considerado **APROVADO** de acordo com a ata da banca examinadora N° **160** de 2017.

Fizeram parte da banca os professores:

Cesar Rafael Claire Torrico

Ricardo Vasques de Oliveira

Ricardo Bernardi

Marcelo Teixeira

A Ata de Defesa assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

DEDICATÓRIA

Dedico este trabalho a Deus, por me dar forças e coragem durante toda esta longa caminhada, aos meus pais, Luiz Antônio da Lapa, Célia Aparecida dos Santos Lapa e meu irmão Bruno Einstein Wyllian Lapa, por todo apoio dedicação paciência e carinho.

AGRADECIMENTOS

Aos professores Marcelo Teixeira e Cesar Rafael Claire Torrico. Pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia.

A todos os professores do curso, que foram de grande importância na minha vida acadêmica.

À minha família, por sua capacidade de acreditar e investir em mim. Célia Aparecida dos santos Lapa, sua dedicação e cuidado foi o que deram, em alguns momentos, a esperança para seguir. Luiz Antônio da Lapa, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Bruno Einstein Wylliam Lapa, sua amizade, companhia e ajuda foram muito importantes e me ajudaram nos momentos mais difíceis.

"Se fracassar, ao menos que fracasse ousando grandes feitos, de modo que a sua postura não seja nunca a dessas almas frias e tímidas que não conhecem nem a vitória nem a derrota."

(Theodore Roosevelt)

RESUMO

LAPA, Hudson. Uma metodologia baseada em controle supervísório para auxiliar na reconfiguração de *smart grids*. 2016. 66 f. Trabalho de Conclusão de Curso de bacharelado em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Este trabalho apresenta uma metodologia baseada na teoria de controle supervísório, para auxiliar *smart grids* em eventos de *self-healing*, que consiste na capacidade de recuperação, após a ocorrência de uma falta no sistema. No trabalho será estudado uma metodologia para representação das *smart grids*, para uma abordagem a eventos discretos onde suas características, restrições e parâmetros técnicos serão modelados segundo a teoria de controle supervísório, com objetivo de ter um supervisor não bloqueante e minimamente restritivo, reduzindo os números de estados que não são desejados na reconfiguração.

Palavras-chave: Controle Supervísório. *Self-Healing*. *Smart Grids*.

ABSTRACT

Lapa, Hudson, A methodology based on supervisory control to aid in the reconfiguration of Smart Grids. 2016. 66 f. Completion of a bachelor's degree in Electrical Engineering, Federal Technological University of Paraná. Pato Branco, 2016.

This work presents an approach that intends to create a methodology which uses supervisory control theory, to aid in self-healing events which consists of the ability to recover, after the occurrence of a fault in the system. In this work, a methodology will be studied for the representation of smart grids, for an approach to discrete events where their characteristics, constraints and technical parameters will be modeled according to the theory of supervisory control, aiming to have a non-blocking and minimally restrictive supervisor, reducing Numbers that are not desired in reconfiguration.

Keywords: Supervisory Control. Self-Healing. Smart Grids.

LISTA DE FIGURAS

Figura 1: Representação de uma Smart-Grid.....	2
Figura 2: Sistema de reconfiguração automática.....	3
Figura 3: Simulador para auto recuperação de rede elétrica pós falta.....	5
Figura 4: Diagrama simplificado do sistema elétrico de potência.....	12
Figura 5: Infraestrutura Elétrica e de Inteligência	13
Figura 6: <i>Self-Healing</i>	15
Figura 7: Exemplo de um diagrama unifilar para cargas conectadas em um barramento.....	17
Figura 8: Curvas de modelo de cargas para análise de correlação P-V.....	19
Figura 9: Autômato Determinístico.....	27
Figura 10: Autômato Acessível e o outro não acessível.....	28
Figura 11: Autômato Co-Acessível e o outro Bloqueante.....	29
Figura 12: SED em malha fechada.....	31
Figura 13: Autômato das plantas das máquinas M1 e M2.....	35
Figura 14: Composição síncrona das plantas M1 e M2.....	35
Figura 15: Autômato da restrição sobre as máquinas	37
Figura 16: Composição da restrição com a planta geral	37
Figura 16: Planta do conjunto de sensores que detectam a falta na região RX.....	40
Figura 17: Planta da região e alimentador respectivamente.....	41
Figura 18: Planta das chaves.....	41
Figura 19: Rede de distribuição para testes.....	42
Figura 20: Detecção e isolamento da falta.....	44
Figura 21: Rede Elétrica Após a Falta na Região 1	45
Figura 22: Região da rede modelado.....	46
Figura 23: Planta das regiões R1, R2 e R3.....	47
Figura 24: Planta do alimentador A1, das regiões alimentadoras R20 e R14 e falta em R1	47
Figura 25: Planta das chaves CH1_2/CH2_1 e CH1_3/CH3_1.....	48
Figura 26: Planta das chaves <i>Tie3_2/2_3</i> , <i>Tie20_2/2_20</i> e <i>tie14_3/3_14</i>	48
Figura 27: Especificação E1.....	50
Figura 28: Especificações E2 e E3.....	50
Figura 29: Especificações E4 e E5.....	50
Figura 30: Especificação E6.....	51
Figura 31: Interface editor Supremica.....	52
Figura 32: Interface simulador Supremica	52
Figura 33: Interface analisador Supremica	53
Figura 34: Possibilidade de Reconfiguração I.....	57
Figura 35: Possibilidade de reconfiguração II.....	58
Figura 36: Possibilidade de reconfiguração III	59

LISTA DE TABELAS

Tabela 1–Número de estado lógico para cada Chave	43
Tabela 2 – Estados da Rede da Figura 19.....	43
Tabela 3– Dados relacionados a composição e SupC	55
Tabela 4– Sequência de manobras de chaves para reconfiguração	56

LISTA DE QUADROS

Quadro 1 - Maiores apagões do mundo desde 1990 4

LISTA DE SIGLAS

TI	TECNOLOGIA DA INFORMAÇÃO
ANEEL	AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA
P&D	PESQUISA E DESENVOLVIMENTO
COPEL	COMPANHIA PARANAENSE DE ENERGIA
SANEPAR	COMPANHIA DE SANEAMENTO DO PARANÁ
COMPAGÁS	COMPANHIA PARANAENSE DE GÁS
LACTEC	INSTITUTO DE TECNOLOGIA PARA O DESENVOLVIMENTO
TECPAR	INSTITUTO DE TECNOLOGIA DO PARANÁ
SED'S	SISTEMA A EVENTOS DISCRETOS
TC'S	TEORIA DE CONTROLE SUPERVISÓRIO
P-V	POTÊNCIA - TENSÃO
P.U	POR UNIDADE
SCADA	SUPERVISÓRIO DE CONTROLE E AQUISICAO DE DADOS

SUMÁRIO

1 - INTRODUÇÃO	1
1.1 Problema e Justificativa	6
1.2 Objetivos	9
1.2.1 Objetivo Geral	9
1.2.2 Objetivos Específicos	9
1.3 Estrutura do Trabalho	10
2 - SISTEMA DE DISTRIBUIÇÃO E REDES ELÉTRICAS INTELIGENTES	11
2.1 <i>Smart Grid</i>	12
2.2 <i>Self-Healing</i> e Redes Elétricas	14
2.3 <i>Self-Healing</i> aplicado em redes elétricas	15
3- MODELAGEM DAS CARGAS E FLUXO DE POTÊNCIA	16
3.1 Definição de Carga	16
3.2 Modelos Estáticos	17
3.2.1 Modelo Corrente Constante	18
3.2.2 Modelo Potência Constante	19
3.2.3 Modelo Impedância Constante	19
3.3 Fluxo de potência	20
3.3.1 Método soma das correntes.....	21
4 - SISTEMA A EVENTOS DISCRETOS	24
4.1 Teoria Das Linguagens Formais	25
4.1.1 Evento.....	25
4.1.2 Alfabeto	26
4.1.3 Cadeia	26
4.1.4 Linguagem	26
4.1.5 Representação de SED`s por linguagens	27
4.2 Autômatos determinísticos de estados finitos.....	27
4.2.1 Linguagens representadas por autômatos	29
4.2.2 Acessibilidade, Co-Acessibilidade e Bloqueio	29

	14
4.2.3 Composição de autômatos	31
4.3 Controle supervisorio de SED	32
4.3.1 Controlabilidade e solução de PCS.....	33
4.3.2 Síntese de supervisores ótimos	35
4.3.3 Modelo da Planta	35
4.3.4 Modelo para especificações.....	37
4.3.5 Algoritmo para cálculo do SupC	39
5- METODOLOGIA	41
6- RESULTADOS	49
7- CONCLUSÃO	64
8 REFERÊNCIAS	66
ANEXOS	70

1. INTRODUÇÃO

O termo *Smart Grid*, ou “rede inteligente”, se refere a uma arquitetura de redes de distribuição, transmissão ou fontes de geração distribuída proveniente de fontes renováveis de energia a partir da qual se torna possível, o envio de dados e informações possibilitando o tratamento e controle acerca do sistema. Essa arquitetura é constituída de alto grau de Tecnologia da Informação (TI) e automação.

As *smart grids* vêm se tornando uma realidade nas redes de transmissão e distribuição de energia, contribuindo para um melhor fornecimento e qualidade do processo e propiciando tornar o sistema mais inteligente, automatizado e integrado.

Seguramente *smart grid* é o futuro da distribuição porque reúne os avanços tecnológicos nas áreas de tecnologia de informação ao mundo da energia, em favor das melhores práticas em termos econômicos e ambientais. Isso principalmente porque possibilita um aproveitamento mais adequado da infraestrutura existente e estimula o aumento da eficiência energética no consumo. (BOCCUZZI, 2010).

Uma das vantagens de se utilizar esse sistema é que o operador tem maiores informações sobre as ocorrências de problemas nas redes tais como, flutuações de tensão, estabilidade e dinâmica ou problemas técnicos, além de propiciar uma integração da geração distribuída com o sistema interligado nacional.

Com as informações obtidas direta e automaticamente da estrutura física, pode-se acompanhar e documentar as eventuais falhas do sistema para uma futura automação. Isso tende a reduzir de maneira significativa a intervenção física do sistema, evitando gastos e minimizando o tempo de recuperação.

A Figura 1 ilustra um sistema de redes inteligentes, que mostra a interligação de geração distribuída, redes de distribuição, consumidores em um centro de dados onde é feito o processamento, análise e integração dos sistemas.

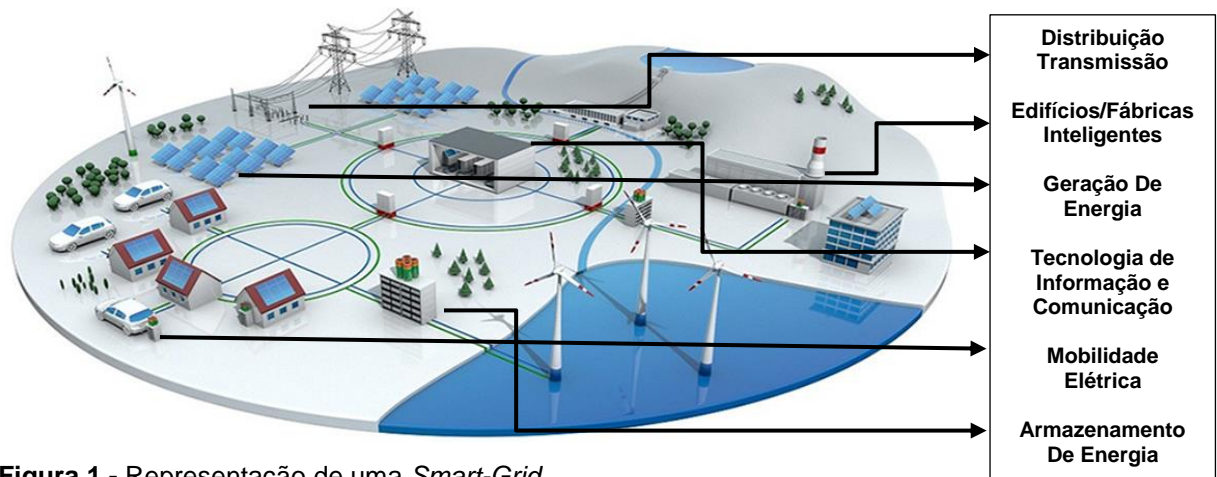


Figura 1 - Representação de uma *Smart-Grid*.
Fonte: 3M SMART GRID (2015).

A ANEEL (Agência Nacional de Energia Elétrica) tem papel importante na implantação de infraestrutura de *Smart Grids*, regulamentando os investimentos em programas de P&D (Pesquisa e Desenvolvimento), além de avaliar, aprovar e acompanhar os resultados obtidos. (ANEEL, 2009). O projeto Paraná *Smart Grid*, por exemplo, é uma iniciativa liderada pela COPEL (Companhia Paranaense de Energia), com a participação da Sanepar, Compagás, Lactec e Tecpar, e tem como objetivo a aplicação de elementos de redes inteligentes na distribuição de energia e outros serviços como água e gás.

Com esse intuito, foi criado um projeto piloto em uma área de alta densidade de carga e visibilidade na cidade de Curitiba, para que sejam feitos testes e correções para uma futura implantação de caráter mais abrangente (PEDRETTI; JUNIOR, 2014).

As pesquisas voltadas a essa área são de suma importância para o avanço do setor e possibilita aos pesquisadores a captação de recursos e a possibilidade de uma implementação real. A Figura 2 mostra uma interface do sistema de reconfiguração automática realizada no projeto piloto em Curitiba.

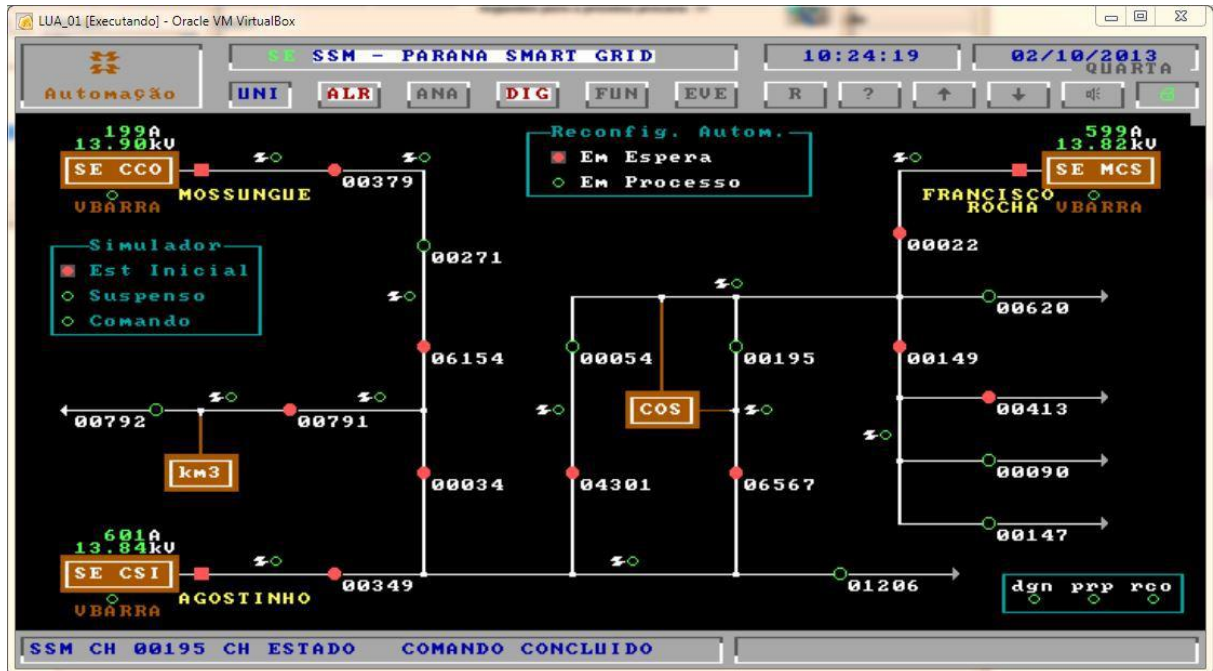


Figura 2 - Sistema de reconfiguração automática.
Fonte: PEDRETTI; JUNIOR (2014).

Basicamente, o projeto piloto da COPEL mapeia os elementos da rede e das cargas e, na ocorrência de faltas, ele pode se reconfigurar da melhor forma possível, além de trazer informações como tensão corrente entre outros parâmetros.

Para a ANEEL:

Esse virtuoso cenário tem propiciado o desenvolvimento de novos sistemas, softwares, materiais, dispositivos e equipamentos para aplicação em diversas áreas do setor. No campo da distribuição de energia elétrica não podemos deixar de destacar o desenvolvimento e as várias aplicações decorrentes desse auspicioso panorama onde, sem sombra de dúvida, se sobressaem as chamadas “redes inteligentes” ou, como querem alguns, as “*smart grids*” que, associadas aos avanços produzidos em outras áreas do conhecimento humano, prometem uma revolução há muito aguardada e desejada na distribuição de energia elétrica no Brasil. Somente podemos falar em “*smart grid*” à medida que nos afastamos do campo teórico ou da liberdade de pensar o futuro como uma retumbante odisséia de devaneios espetaculares sem nenhuma aplicação prática e concreta. (ANEEL, 2009).

O cenário atual está propício para o desenvolvimento dessas tecnologias, tanto em âmbito tecnológico quanto intelectual. Além disso, vale ressaltar a necessidade de se utilizar essas novas tecnologias em nossas redes de distribuição de energia: o Brasil ocupa uma das piores posições em âmbito mundial em questão de número de apagões (GTA/UFRJ, 2013), como mostra o *Quadro 1*.

Países	Data	Causas	Duração	Habitantes afetados
Nova Zelândia	20/02/1998	Falhas técnicas	4 semanas	70.000
Brasil	11/03/1999	Eventos naturais	5 horas	97.000.000
Índia	02/01/2001	Falhas técnicas	12 horas	226.000.000
Estados Unidos e Canadá	14/08/2003	Falhas técnicas e de comunicação	4 dias	50.000.000
Itália	28/09/2003	Falhas técnicas	18 horas	56.000.000
Espanha	29/11/2004	Falhas humanas e técnicas	5 apagões em 10 dias	2.000.000
Indonésia	18/08/2005	Falhas técnicas	7 horas	100.000.000
Sudoeste da Europa	04/11/2006	Falhas humanas e de comunicação	2 horas	15.000.000
Brasil e Paraguai	10/11/2009	Eventos naturais	25 minutos a 7 horas	87.000.000
Brasil	04/02/2011	Falhas técnicas	16 horas	53.000.000

Quadro 1 - Maiores apagões do mundo desde 1990

Fonte: Adaptado de GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO GTA/UFRJ (2013).

Como pode-se observar na Quadro 1 o Brasil teve um dos piores apagões da história afetando milhões de pessoas e a economia do país. A confiabilidade almejada para o fornecimento de energia elétrica é bem elevada, sendo em geral na ordem de 99,999%, o que corresponde a aproximadamente cinco minutos de interrupção de energia anualmente. (GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO GTA/UFRJ, 2013).

Essa confiabilidade interfere, por exemplo, na escolha de certas indústrias por instalarem-se em um determinado país ou região. Portanto uma alta confiabilidade se faz importante, também, para trazer novos investimentos e empreendimento para determinadas regiões.

Ainda assim, as falhas no sistema elétrico são inevitáveis. Mesmo que todas as medidas cabíveis sejam tomadas, em algum momento poderá ocorrer um evento inesperado podendo ser: falhas técnicas, humanas, naturais, etc. Assim, para conseguir manter a continuidade da transmissão da energia para o consumidor final, o sistema precisa ser dotado de um alto poder de recuperação, ou seja, após uma falta, o sistema deve ser capaz de planejar a sua própria reconfiguração para que esse processo seja o mais rápido e eficiente possível.

Na prática, porém, a tarefa de reconfiguração costuma ser árdua e raramente leva a resultados ótimos. Isso porque, após a ocorrência de uma falha no

Sistema Elétrico de Potência, o seu reestabelecimento requer uma sequência de manobras envolvendo chaves e alimentadores. Mais do que isso, tais manobras precisam ser executadas em consonância com um conjunto (eventualmente complexo) de requisitos para o sistema, por razões óbvias de segurança e funcionalidade. Devido ao grande número de chaves, de alimentadores, e à eventual complexidade do conjunto de requisitos que podem estar envolvidos na resolução do problema, a tarefa de reconfigurar implica em um extenso e complexo arranjo, cujo processamento é normalmente associado a técnicas de meta Heurísticas.

Atualmente, no escopo de um projeto de pós-doutorado, do qual o autor deste trabalho participa como colaborador, vem sendo desenvolvida uma estratégia para o reestabelecimento do Sistema Elétrico de Potência. A Figura 3 apresenta uma interface desse sistema.



Figura 3 - Simulador para auto recuperação de rede elétrica pós falta.
Fonte: RIBEIRO, *et al.*, (2017).

O sistema é de natureza estocástica e se utiliza de meta Heurísticas para encontrar um caminho alternativo para que o reestabelecimento seja possível. A figura 3 inclui uma possível amostragem de dados, onde podem ser colocados os alimentadores, chaves, cargas, linhas e restrições operacionais.

1.1 PROBLEMA E JUSTIFICATIVA

Embora abordagens estocásticas sejam úteis às etapas do processo de auto recuperação a faltas (tecnicamente denominados de *self-healing*), duas desvantagens podem ser claramente identificadas:

- I. A aplicabilidade dessas técnicas sobre processos reais em geral limita-se a redes de pequeno porte;
- II. Elas levam a resultados aproximados cuja *otimidade* não é garantida por construção.

No primeiro caso, testes preliminares (MAZZETTO, 2016) dão conta de que não mais que 30 cargas podem ser analisadas em paralelo para que um algoritmo de recomposição convirja para uma resposta aceitável, em tempo aceitável. Como em processos reais uma rede pode comportar uma quantidade substancialmente maior de cargas, pode ser assumido que a abordagem estocástica de reconfiguração é inviável na prática, devido à sua complexidade computacional, e essa complexidade aumenta em função da quantidade de estados da rede, da quantidade de restrições a serem testadas, e da natureza das restrições, que muitas vezes são dependentes, tornando a reconfiguração um problema combinatório.

Já o segundo caso pode ser visto como uma decorrência do primeiro: diante da inviabilidade do cálculo de uma reconfiguração ótima, opta-se por um resultado intermediário, cuja otimidade não se conhece, mas que é viável de ser obtido em um curto período de tempo e que cumpre com a necessidade de se obter uma versão operante da rede. Ademais, a obtenção de uma reconfiguração ótima implicaria no processamento de algoritmos exatos sobre o espaço de estados completo da rede, o que remete a um problema cuja ordem de complexidade é exponencial, no pior caso, sendo esta uma tarefa computacionalmente intratável.

Neste trabalho, será mostrado como o uso de técnicas de controle supervisão pode beneficiar o processo de recuperação de *smart grids*. A ideia central da proposta é diminuir o número de estados, retirando estados que não poderiam ocorrer fisicamente e estados que seriam desnecessários ou ruins para reconfiguração, isto iria auxiliar no algoritmo de recuperação na etapa dispendiosa de

ter de verificar todo o espaço de estados a fim de evitar que alguma restrição seja infligida. Do contrário, a estratégia proposta computa, a partir do modelo da *smart grid*, uma versão desse modelo restrita a um conjunto de requisitos operacionais. O resultado é um submodelo da rede que contém apenas a parte admissível dos estados que sobrevivem aos requisitos. Por fim, esse submodelo pode ainda ser avaliado pelo algoritmo de reconfiguração a fim de estabelecer o arranjo final da rede.

Como resultado da proposta, espera-se contribuir com alternativas para o tratamento dos problemas 1 e 2, apontados acima, nos seguintes aspectos:

- A aplicabilidade dos sistemas de recuperação a faltas poderá ser estendida a processos reais, de médio e grande porte, com número realístico de cargas;
- A abordagem de controle supervisorio possui suporte a modularização, ao contrário da abordagem de reconfiguração estocástica pura. Isso tende a mitigar o problema da complexidade computacional;
- A abordagem de controle supervisorio possui suporte à computação de resultados ótimos em relação ao cumprimento de um conjunto de requisitos. Tais resultados, quando associados aos algoritmos de reconfiguração, podem melhorar o nível de otimismo da solução final;
- Estima-se que a solução de reconfiguração tenderá a apresentar melhores índices de convergência, haja visto que será processada sobre um espaço de estados menor do que o original, em um tempo de processamento também menor;

Para o alcance desses pressupostos, será mostrado que a sequência de ocorrências numa *smart grid* pode ser intuitivamente vista como um Sistema a Eventos Discretos (SEDs) (CASSANDRAS; LAFORTUNE, 2008). Um SED explicita um comportamento cuja dinâmica não depende diretamente do tempo, mas de eventos assíncronos e esporádicos, que podem surgir em intervalos irregulares e imprevisíveis. Normalmente, a modelagem desse tipo de sistema se dá por meio de diagramas de estados, em oposição aos sistemas dirigidos pelo tempo, cuja modelagem é mais naturalmente conduzida por meio de equações diferenciais.

Nesse sentido, este trabalho organiza o comportamento da rede dentro de um arranjo lógico, de modo que a execução conjunta dos elementos da rede irá

determinar o comportamento global do sistema, denominado planta. A planta reflete o comportamento do sistema em malha aberta sem nenhuma restrição. Assim, o próximo passo é fechar a malha, ou seja, implementar as restrições as quais espera-se que o sistema cumpra quando em funcionamento. Uma restrição, ou especificação, quando associada ao modelo da planta interfere em determinados eventos possíveis em malha aberta adequando o comportamento da planta aos requisitos de controle, compondo o comportamento em malha fechada.

Nesse trabalho, o modelo da rede irá compor a planta em malha aberta do sistema avaliado. Requisitos como “observar os limites da linha de transmissão”, “permanecer na radial”, “faltas”, “integridade após a abertura ou fechamento de chaves”, etc., irão compor o modelo das especificações do sistema. Então, a partir da combinação da planta e especificações (restrições físicas e técnicas), será computado um supervisor que funcionará como um controlador para o sistema em malha fechada que atuará de forma minimamente restritiva sobre o sistema evitando estados que podem levar a um comportamento não desejado. Por fim, esse supervisor será combinado com os algoritmos de self-healing para a definição da reconfiguração final, ou seja, dados os estados possíveis e aceitáveis sob controle, serão definidas as alternativas que de fato levarão à reconfiguração física.

O processo de cálculo do supervisor fará o uso da Teoria de Controle Supervisório (TCS). Será mostrado como os eventos provenientes de uma *smart grid* podem ser adequadamente tratados conforme a sua natureza de ocorrência. A TCS provê um mecanismo formal para a obtenção de controladores ótimos para SEDs (RAMADGE; WONHAM, 1987). No caso desse trabalho, o controlador será uma entidade que irá observar as falhas que a rede pode sofrer e, dentre todas as ações possíveis de serem sugeridas pelo *self-healing*, ele irá habilitar apenas as ações cujo desfecho não implique, direta ou indiretamente, em riscos de violação do conjunto de requisitos.

Em suma, a utilização conjunta da TCS de SEDs e dos algoritmos Heurísticos se complementa, tendo em vista que a heurística é uma abordagem probabilística enquanto que a TCS provê um método analítico, exato, para controle. Para isso, os eventos envolvidos numa *Smart grid* precisa ser modelada como um SED, o que é concebível, dada a dinâmica esporádica, assíncrona e não determinística dos seus eventos. Sendo assim, a TCS naturalmente se aplica sobre *smart grids*.

A principal expectativa é a de que a associação dessas duas técnicas traga uma maior segurança e parametrização ao sistema de *self-healing*. Um controlador baseado na TCS, além de ser seguro, no sentido de cumprir com o conjunto de requisitos, teria ainda uma ação maximamente permissiva e não bloqueante sobre a *smart grid*, características desejáveis em um sistema de *self-healing*.

Outra expectativa de resultados é que, como as redes de distribuição são estruturas físicas de elevado custo e que tem um alto grau de planejamento de implantação, elas possuem um tempo de retorno de investimento elevado. Portanto elas permanecem por longos períodos fixas, sem mudanças, o que motiva ainda mais a obtenção de um sistema de *self-healing* robusto e durável, características essas possíveis a partir do método proposto.

1.2 Objetivos

1.2.1 Objetivo Geral

Pretende-se empregar a Teoria de Controle Supervisório de Sistemas a Eventos Discretos para modelar uma *smart grid* e coordena-la de maneira ótima, i.e., minimamente restritiva, controlável, não bloqueante e em consonância com um conjunto de especificações. Com isso pretende-se subsidiar o processo de *self-healing* com um modelo que contempla um subconjunto de estados possíveis, dentro do espaço de estados global da rede, o que tende a simplificar o processo de otimização e reconfiguração.

1.2.2 Objetivos Específicos

- Levantamento bibliográfico de sistemas *smart grid*, *self-healing* e controle supervisório de sistemas a eventos discretos:
- Levantamento dos principais problemas das redes de distribuição, após o processo de recuperação.
- Definição de uma estrutura de modelagem que possa representar uma rede de distribuição e seus requisitos operacionais.

- Analise dos aspectos relacionados as redes, como: fluxo de potência, carga, capacidade de transmissão, topologia da rede etc. Que Poderiam ser integrados ao *framework* de controle supervisorio.
- Levantamento de redes de distribuições radiais mais simplórias para finalidade de testes simples e por fim modelar uma rede real, que possui complexidade maior e tamanho maior.
- Testes e possíveis correções.

1.3 Estrutura do Trabalho

Este trabalho está organizado da seguinte maneira: No Capítulo 2 é introduzido o conceito das redes elétricas inteligentes (*Smart Grid*) e a atuação de sistemas *self-healing* sobre essas redes. No Capítulo 3 são abordados os tipos de modelagem de cargas e o método para cálculo do fluxo de potência. O Capítulo 4 faz uma descrição do estado da arte voltado para estudos de SED, onde a metodologia será aplicada, vale salientar que tal modelagem precisa ser compatível com as técnicas de controle a eventos discretos e usual para resolução de problemas de *Self-Healing*.

No Capítulo 5 é feita metodologia para abordar o problema de modelagem do sistema em formato de SED, onde será identificado os elementos a serem representados, no capítulo 6 é feito a identificação das restrições e suas respectivas modelagem em formato de especificações do sistema e a obtenção do supervisor, onde será analisado a redução do espaço amostral de chaves com e sem o supervisor. Os resultados são discutidos no Capítulo 7.

2. SISTEMA DE DISTRIBUIÇÃO E REDES ELÉTRICAS INTELIGENTES

As redes de energia elétrica convencionais estão sendo modificadas e moldadas às necessidades e demandas da sociedade modernas demandando o desenvolvimento de novas tecnologias. Isso pode ser visto nos estudos que apresentam o uso de fontes renováveis de energia, adaptação e divisão de abastecimento em subestações e a sua importância em casos críticos para a segurança em sistemas de transmissão e de distribuição, etc. (U.S. DEPARTMENT OF ENERGY, 2010).

No cenário nacional a ANEEL (Agência Nacional de Energia Elétrica) reconhece que a tendência mundial de substituição da tradicional infraestrutura do sistema de energia elétrica por uma configuração associada às recentes tecnologias de informação e comunicação vão proporcionar importantes mudanças na forma de relacionamento entre regulador e concessionária de energia e entre esta e seus consumidores. São, portanto, de interesse do regulador as questões decorrentes dessa tendência, que vêm acompanhadas por uma demanda promissora e, de certa forma, iminente (ANEEL, 2010).

Para entendermos melhor as tendências das linhas inteligentes é preciso conhecer melhor a forma tradicional do setor elétrico. O sistema atual é dividido em três partes: Geração, Transmissão e Distribuição, onde:

- **Geração:** A geração é o segmento da indústria de eletricidade responsável por produzir energia elétrica e injetá-la nos sistemas de transporte (transmissão e distribuição) para que chegue aos consumidores. (ABRADEE, 2015). Especialmente no Brasil, as principais fontes para geração são hídricas, porém o país tenta fomentar incentivo para geração distribuída e renovável, tais como: eólica, solar e biomassa.
- **Transmissão:** Pela definição de transmissão é aquele que se encarrega de transportar grandes quantidades de energia provenientes das usinas geradoras. Essas linhas conectam os geradores aos grandes consumidores ou, às empresas distribuidoras. No Brasil, o segmento de transmissão é aquele que se caracteriza por operar linhas em tensão elétrica superior a 230 mil Volts. (ABRADEE, 2015).

- Distribuição: Por sua vez é aquele que recebe grande quantidade de energia do sistema de transmissão e a distribui de forma pulverizada para consumidores médios e pequenos. Existem também unidades geradoras de menor porte, normalmente menores do que 30 MW, que injetam sua produção nas redes do sistema de distribuição. Essas linhas são abaixo de 230 mil Volts. (ABRADEE, 2015)

Tendo ciência de como funciona o básico do setor elétrico a figura abaixo exemplifica de forma visual e simplificada como é feita essa operação dividindo em três zonas, Geração, Transmissão e Distribuição, que estão divididas nas seguintes legendas (Preto, azul e verde) respectivamente.

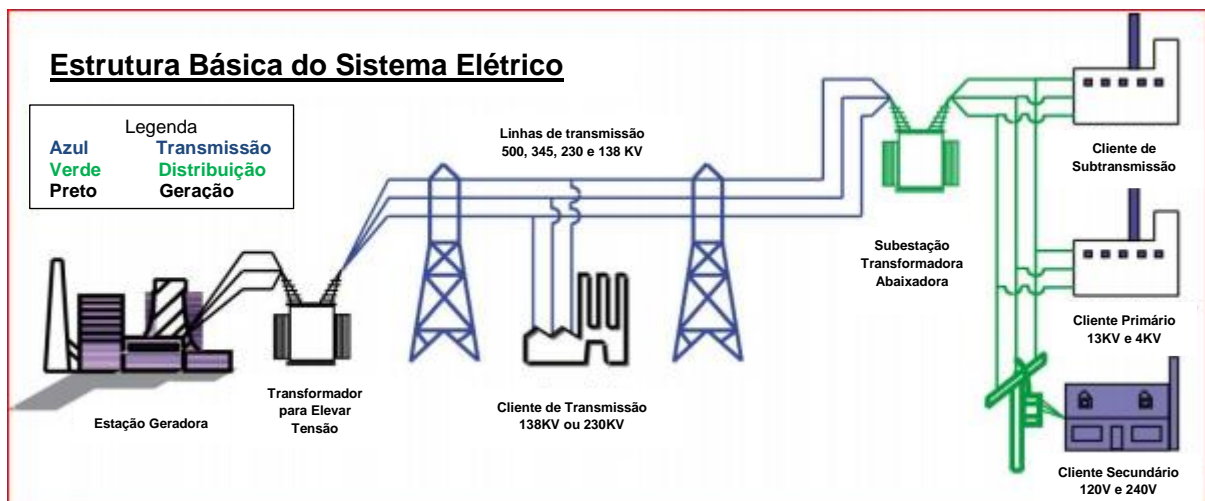


Figura 4 - Diagrama simplificado do sistema elétrico de potência
Fonte: ALJAHANI (2014)

2.1 Smart Grid

Devido ao crescimento da demanda pela energia elétrica, os elementos da infraestrutura das redes de energia estão mais interconectados, causando alterações na concepção da estrutura inicial. A dificuldade de manter os padrões ideais com o crescimento da demanda, fizeram com que algumas alterações pudessem influenciar em determinadas atividades, como por exemplo, a dificuldade no gerenciamento das manutenções devido à ausência de mecanismos para a observação completa da rede (AMIN; WOLLENBERG, 2005 apud MAZZETTO, 2016).

Com a evolução da eletrônica e telecomunicações se fez possível o monitoramento das redes elétricas, no entanto a integração do sistema elétrico de potência com os sistemas de monitoramento/telecomunicação não é tão simples além de possuir um elevado custo. Aos poucos com o avanço tecnológico e as vantagens de se utilizar essa estrutura de rede essa realidade vem mudando em todo o mundo, começando pelos países de primeiro mundo, onde a geração distribuída e fontes renováveis estão em franca evolução e implantação, fazendo necessário essa topologia de rede para gerenciar a entrada dessas cargas, otimizar a utilização da capacidade de transmissão das linhas e minimizar as perdas técnicas. As redes inteligentes têm incorporado nelas, uma serie de equipamentos e toda uma infraestrutura tal como, sensores, fibra ótica, centrais de processamento, transmissores, receptores etc. Além de possuir a estrutura básica para o transporte de energia igual ilustrado na *Figura 4*. A *Figura 5* ilustra a infraestrutura elétrica e de “Inteligência”.

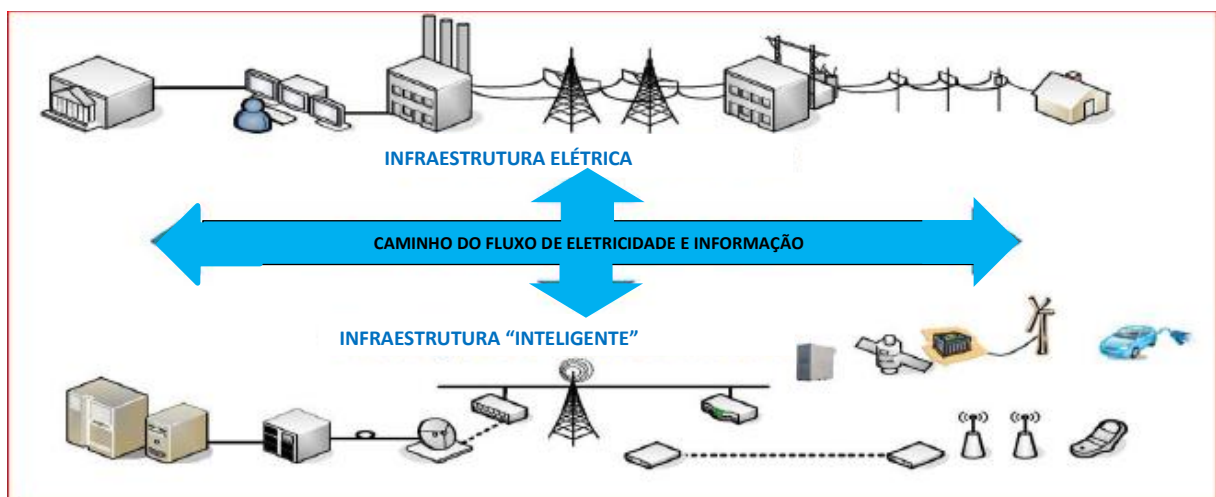


Figura 5 - Infraestrutura Elétrica e de Inteligência

Fonte: ALJAHANI (2014)

Como pode-se observar na figura 5, a infraestrutura que demanda uma *smart grid* engloba uma estrutura elétrica convencional juntamente com elementos que dão “Inteligência” a rede e flexibilização, como sensores, geração distribuída, centros de processamento, transmissão de dados etc. Com isso o custo físico de uma *Smart grid* supera em muito o custo de uma rede convencional.

2.2 *Self-Healing* e Redes Elétricas

Pela tradução, *Self-Healing* significa “Auto Cura”, ou a capacidade de se auto curar, em sistemas elétricos, seria a capacidade de um sistema elétrico de se reconfigurar depois de sofrer uma falta, essa capacidade de se auto configurar ou “Auto Curar”. Para se fazer a auto reconfiguração pode se utilizar ações programadas, para uma ação rápida de manobra de chaves, isolando a região que aconteceu a falta e dando um novo caminho para o fluxo de potência por caminhos alternativos. O termo original foi definido em função dos sistemas inclusos nessa categoria, que podem se ‘curar’ no sentido de se recuperar de situações e retomar seu desempenho original (DEBANJAN GHOSH, 2007)

No ambiente elétrico, até a década de 90, avanços nos sistemas SCADA (Supervisório de controle e aquisição de Dados) trouxeram uma melhor perspectiva em relação à automação das redes. Porém, a capacidade computacional e de comunicação desses sistemas era muito baixa e necessitava de intervenção humana, descaracterizando o princípio da automação. (LAMBIASE, 2012).

Com os avanços tecnológicos no âmbito da computação, telecomunicações e eletrônicos, tornou-se possível a implantação de redes inteligentes com sensores, chaves inteligentes, centro de processamento de dados com computadores que possuem um alto poder de processamento podendo criar soluções mais rapidamente para problemas que estão acontecendo em tempo real. Tal perspectiva tornou possível a implementação de ações de automação nas redes e também empregar novas técnicas computacionais que conseguem redistribuir o fluxo de potência otimizando alguns parâmetros como, maior número de consumidores reestabelecidos mantendo cargas prioritárias, limites de queda de tensão aceitáveis por norma, entre outros parâmetros que podem buscar sua maximização ou minimização.

Os sistemas de *self-healing* tem subentendido em sua definição a característica de autonomia em relação a alguma atividade. Como apresentado na (PSAIER; DUSTDAR, 2011 apud MAZZETTO, 2016).

De modo geral a metodologia utilizada para aplicar a auto cura em algum sistema segue os seguintes passos:

- I. Monitoramento: Coleta e processamento de dados;

- II. Análise: Determina se deve tomar uma ação de acordo com a informação recebida;
- III. Plano: Ações a serem tomadas de acordo com sistema, feitas pelo analisador;
- IV. Execução: Ações estabelecidas previamente pelo plano e análise;
- V. Conhecimento: É a base de conhecimento produzida ou consumida;

2.3 Self-Healing aplicado em redes elétricas

Antes de ajustar um sistema para torná-lo auto reparável, é necessário identificar quais são as características consideradas normais para sua execução. Em sistemas de distribuição, uma rede em funcionamento normal é quando está fornecendo energia para todas as demandas, ou seja, livre de faltas e de acordo com as limitações dos componentes da rede. Critérios que determinam uma rede adequada descrevem os estados normais do sistema, e ao defini-los é possível determinar com maior facilidade os limiares que indicam quando é atingido um estado anormal, que requisitará as funções de auto recuperação (GHOSH et al., 2007).

Uma representação dos passos de como um mecanismo de auto recuperação funciona é ilustrado na Figura 6.



Figura 6 - Processo para o sistema se auto curar.
Fonte: Adaptado de PSAIER; DUSTDAR(2011, P.7).

Deve-se abstrair da Figura 6 o mecanismo que ele propõe e colocá-lo em uma perspectiva do sistema elétrico, pode-se observar que com os sensores que monitoram as *Smart-Grid*, tem-se um monitoramento constante das linhas e quando ocorrer algum evento inesperado, como uma falta, os dados são emitidos para um centro de dados, onde o mesmo faz o diagnóstico e uma solução é gerada, seja por ações pré-estabelecidas ou através de algum nível de inteligência artificial e/ou otimização. Assim que a decisão é tomada a rede será reconfigurada e ajustada para

o novo caminho de fluxo determinado pelos caminhos formados pelas chaves que foram alteradas.

3. MODELAGEM DAS CARGAS E FLUXO DE POTÊNCIA

3.1 Definição de Carga

O termo “Carga” no contexto elétrico de potência, possui diferentes significados dependendo da situação em que ela se enquadra, sendo as principais definições (VISCONTI, 2010):

- Um equipamento conectado ao sistema de potência que consome energia;
- A energia total consumida por todos os equipamentos conectados ao sistema de potência;
- Uma porção do sistema, que não é representada detalhadamente, mas é tratada como um equivalente que consome uma quantidade de potência e está conectado a um barramento.

A definição nos diz que, a partir de um barramento podemos assumir que as cargas conectadas no mesmo podem ser consideradas como um equivalente tendo a tensão no barramento como grandeza de perturbação ou de entrada do sistema, podendo ter como variáveis, tensão, frequência, temperatura ambiente etc. Depende do nível de precisão e detalhamento que o modelo informa ou que precisamos para analisar determinado fenômeno. A **Figura 7** ilustra como funciona o barramento em uma indústria onde estão conectadas várias cargas.

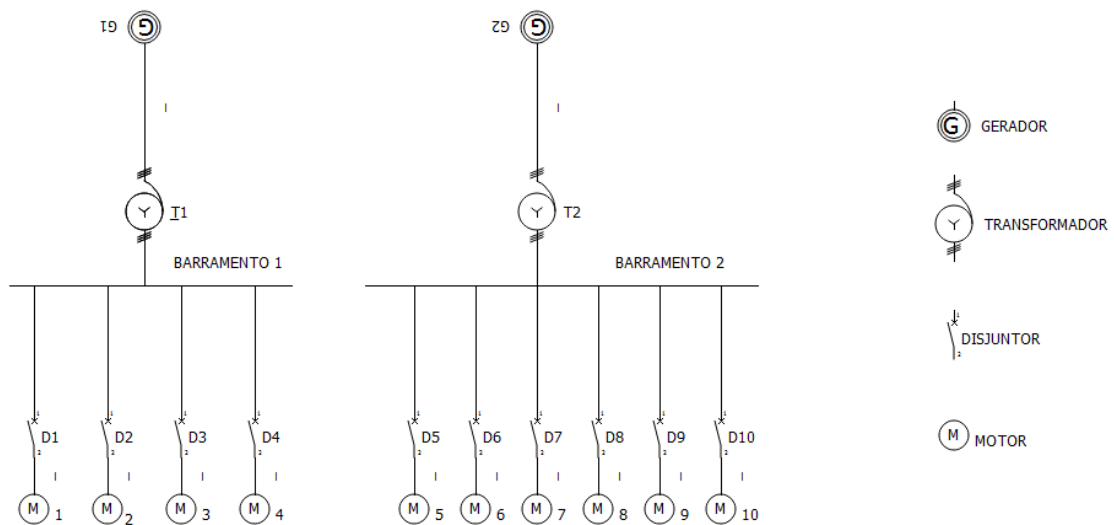


Figura 7– Exemplo de um diagrama unifilar para cargas conectadas em um barramento.
Fonte: Autoria Própria.

Como podemos observar na Figura 7, temos várias cargas, no caso motores conectadas no barramento, podemos tratar as cargas elétricas sendo um elemento consumidor, ou fazer um equivalente entre as barras juntando todos motores e modelar as cargas, tudo depende da quantidade de precisão e informação que você necessita para modelar determinado processo. No trabalho vão ser utilizados métodos estáticos, que serão descritos no próximo tópico.

3.2 Modelos Estáticos

São modelos que expressam as potências ativa e reativa, em qualquer instante do tempo, em função da magnitude de tensão do barramento de carga e da frequência elétrica do sistema, naquele instante (IEEE, 1993 apud VISCONTI, 2010).

Quando se tem pequenas variações ou lentas de tensão e/ou frequência, o sistema retorna ao regime permanente rapidamente, tornando possível modelar a carga por modelos estáticos. A seguir as representações comumente utilizadas para modelar cargas de forma estática, relacionando a potência somente em função da tensão.

3.2.1 Modelo Corrente Constante

A modelagem da corrente constante são cargas das quais o módulo e a defasagem angular entre a tensão e corrente permanecem constante quando o valor da tensão varia. Como mostra a equação (3-1).

$$P(k) = P_0 \left(\frac{V(k)}{V_0} \right)^2 \quad (3-1)$$

$$Q(k) = Q_0 \left(\frac{V(k)}{V_0} \right)^2$$

Este método é utilizado para análise de estabilidade a pequenos sinais

Onde:

- $P(k)$ é a potência ativa expressa em função da variável discreta do tempo k .
- $Q(k)$ é a potência reativa expressa em função da variável discreta do tempo k .
- $V(k)$ é a tensão elétrica expressa em função da variável discreta do tempo k .
- P_0 , Q_0 e V_0 são respectivamente os valores de potência nominal, reativa e a tensão em regime permanente antes do distúrbio ou, fisicamente falando o ponto de equilíbrio do sistema.

3.2.2 Modelo Potência Constante

Este modelo de carga mantém sempre as mesmas injeções de potência ativa e reativa, independente da tensão aplicada. Um exemplo são os motores de indução.

$$P(k) = P_0 \quad (3-2)$$

$$Q(k) = Q_0$$

3.2.3 Modelo Impedância Constante

O modelo de impedância constante mostra a variação da potência proporcionalmente ao quadrado de desvios da tensão $V_{(k)}$ de seu ponto de operação V_0 , como mostra a equação (3-3).

$$P(k) = P_0 \left(\frac{V(k)}{V_0} \right)^2$$

$$Q(k) = Q_0 \left(\frac{V(k)}{V_0} \right)^2$$

(3-3)

Cada tipo de modelagem tem suas vantagens e desvantagens, tudo depende dos dados e objetivos da modelagem. A figura abaixo demonstra os três tipos de modelagens em um gráfico Potência-Tensão, onde demonstra a variação da potência em relação a tensão expressos em P.U.

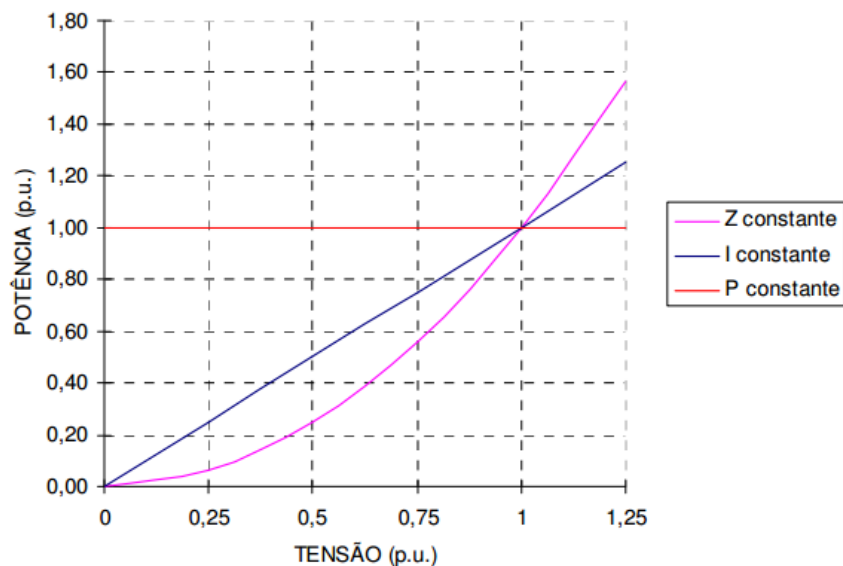


Figura 8 – Curvas de modelo de cargas para análise de correlação.
Fonte - AMARAL (2004).

3.3 Fluxo de potência

O fluxo de potência é um problema matemático composto por um conjunto de equações algébricas não lineares, que permite determinar os valores de tensão complexa nas barras e a potência que flui através das linhas, bem como as perdas e outras variáveis de interesse, para um sistema elétrico de potência em regime permanente. Após determinado o estado operativo do sistema, é possível avaliar se o

mesmo está ou não operando de forma adequada (KAGAN; OLIVEIRA; ROBBA, 2005) e, caso não esteja, determinar as ações corretivas para regularizar a adversidade. Dentre as principais aplicações do fluxo de potência, encontram-se:

- **Análise de segurança:** simula a violação dos limites de operação do sistema, permitindo que ações preventivas, ou corretivas, sejam tomadas, evitando futuras contingências ou possibilitando o reparo do sistema após uma falha.
- **Planejamento:** durante uma expansão, para atender uma demanda maior ou estender a área de cobertura do sistema, o Fator de Potência (FP) pode ser utilizado para verificar o funcionamento da nova configuração de rede e para otimizar as perdas de operação.
- **Otimização de sistemas em operação:** determinar pontos onde as perdas técnicas são altas é fundamental para melhorar a eficiência na distribuição de energia. Transformadores com defeito, cabos fora das especificações e isoladores danificados prejudicam o transporte da energia elétrica, gerando prejuízos que muitas vezes passam despercebidos. O FP permite o cálculo das perdas nos componentes do sistema, possibilitando a identificação e substituição de equipamento avariado. No caso das perdas não técnicas, alguns algoritmos baseados no fluxo de potência têm surgido na literatura de modo a automatizar a detecção de fraudes na rede (RAMOS, 2014).

No trabalho não será utiliza-se o fluxo de potência, porém o mesmo é necessário para o cálculo da potência, requisitos de qualidade de energia tais como limite de fluxo de potência em uma linha, limite da variação da tensão e perdas de potência ativa na rede, logo para reconfigurar uma rede é necessário fazer os cálculos relacionados ao fluxo de potência para analisar a viabilidade da reconfiguração. A rede modelada possui topologia radial, portanto pode-se utilizar o método para o cálculo do fluxo de potência soma das correntes.

3.3.1 Método soma das correntes

Esse método, proposto por (SHIRMOHAMMADI, 1988 apud SILVA, 2016) e depois adaptado para redes trifásicas (CHENG; SHIRMOHAMMADI, 1995 apud SILVA, 2016), utiliza a soma das correntes nas linhas para determinar o fluxo de potência. O algoritmo consiste em duas varreduras, denominadas *backward sweep* e *forward sweep*, sendo uma à montante e outra à jusante. Uma das vantagens do método da soma das correntes (MSC) é que o mesmo pode ser aplicado a redes fracamente malhadas (redes com poucas malhas).

Por convenção é utilizada a normalização em sistema por unidade, expressa em por unidade (p.u), que proporciona o cálculo de todas as grandezas em função de valores de uma tensão V_{base} e uma potência S_{base} . Isso possibilita simplificações nos cálculos de sistemas de potência, como uso de uma constante para sistemas trifásicos, simplificação da representação dos transformadores sobre a rede e similaridade nos valores das grandezas dos componentes do sistema (GRAINGER; STEVENSON, 1994, apud MAZZETTO, 2016).

No trabalho serão utilizadas redes radiais, o método soma das correntes se torna adequado para o processo de cálculo de fluxo de potência. O método consiste nas seguintes etapas:

- Primeiro, é assumir uma referência de tensão inicial para as barras, geralmente 1 p.u.
- Calcula-se a injeção de correntes nas barras requerida pelas cargas, dado por:

$$I_k^i = \left(\frac{S_k^i}{V_k^i} \right)^* \quad (3-4)$$

Onde:

I_k^i : É a corrente injetada na barra k na fase i para uma tensão na mesma fase na barra.

V_k^i : É a tensão da barra k na fase i .

S_k^i : É a carga complexa na barra k na fase i dividida em $S_k^i = P_k^i + jQ_k^i$

- Depois de obter as correntes das linhas que chegam às barras soma-se as corrente de linhas a jusante da barra k e as requeridas pela barra k . O valor de cada ramo é calculado pela lei de corrente de Kirchhoff, descrita na equação (3-5)

$$[I_{ak}] = I_k^i + \sum_{l \in Lm} I_m \quad (3-5)$$

Onde:

I_{ak} : Corrente no seguimento das barras a e k .

$\sum_{l \in Lm} I_j$: é a soma do conjunto de todas as linhas a jusante da barra k .

- A varredura a jusante, partindo da subestação para as barras finais, é realizada para determinar os valores das tensões do sistema. O valor em cada nó é calculado, baseado na lei de tensão de Kirchhoff, descrita na equação (3-6):

$$V_j = V_i - Z_{ij}I_{ij} \quad (3-6)$$

- Após atualizar a tensão com a varredura a jusante, é realizado um teste de erro absoluto entre o valor anterior da tensão V_0 e o valor atual V_1 , para um limite ε previamente especificado, como descrito na equação (3-7):

$$|V_j^i - V_j^{i-1}| < \varepsilon \quad (3-7)$$

- Se todos os valores do erro absoluto convergirem então temos a resposta para a solução do fluxo. Caso contrário, se continua da etapa $i+1$ até convergência.

4. SISTEMA A EVENTOS DISCRETOS

Pela definição de (CURY, 2001), um Sistema a Eventos Discretos (SED) é um sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos irregulares de tempo. SEDs funcionam através de estímulos do mundo externo, que são denominados eventos, por exemplo, o início ou fim de uma tarefa, não sua execução. Um evento pode ser referente a um sinal de um sensor que foi ativado ou o término de uma atividade, ou seja, eventos discretos que acontecem de forma assíncrona ou com certa temporização. Tais eventos causam uma transição de estados no sistema onde o mesmo permanece até que outro evento ocorra.

Foram desenvolvidos até o momento vários formalismos para a modelagem de SEDs, mas nenhum tem se firmado como universal. Esses formalismos refletem diferentes tipos de SEDs bem como diferentes objetivos de análise. Os principais formalismos utilizados para representar SEDs são: (CURY, 2001).

- Redes de Petri com e sem temporização;
- Redes de Petri Controladas com e sem temporização;
- Cadeias de Markov;
- Teoria das Filas;
- Processos Semi-Markovianos Generalizados (GSMP) e Simulação;
- Álgebra de Processos;
- Álgebra Max-Plus;
- Lógica Temporal e Lógica Temporal de Tempo Real;
- Teoria de Linguagens e Autômatos

4.1 Teoria Das Linguagens Formais

Considere um SED, cuja evolução se dá em razão da ocorrência de eventos. Naturalmente, o resultado dessa dinâmica remete a uma trajetória formada por sequências de eventos. Para cada uma dessas sequências dá-se o nome de cadeia. É intuitivo pensar que modelar um SED nada mais é, então, do que descrever tais cadeias e, analogamente, processar operações sobre o modelo de um SED consiste em manipular suas cadeias.

4.1.1 Evento

O primeiro passo para modelar um SED é identificar os eventos que guiam a dinâmica do sistema. Cada um desses eventos é então identificado por um símbolo, distinto dos demais. Identificar eventos em um SED pode ser relativizado ao nível de abstração com que o sistema é interpretado, ou a necessidade de observar determinados eventos para fins práticos. De todo modo, tratam-se de atividades de engenharia e que, portanto, dependem da experiência do projetista.

É comum que o sistema naturalmente aponte para um conjunto de eventos físicos, como por exemplo sinais de início e fim de operações em máquinas, sensores ao identificar uma situação particular, movimentos robóticos, etc.

4.1.2 Alfabeto

Alfabeto é como é conhecido o conjunto de eventos identificados em um SED. Assumimos que o alfabeto é finito, não-vazio e denotado por Σ .

4.1.3 Cadeia

Define-se por cadeia uma sequência finita de eventos pertencentes ao alfabeto Σ . Um caso particular de cadeia é a cadeia vazia (denotada por ϵ), a qual corresponde a uma sequência com nenhum evento. Denota-se por Σ^* o conjunto de todas as cadeias possíveis de serem compostas com eventos em Σ , incluindo a cadeia ϵ . Convencionalmente, o tamanho de uma cadeia $s \in \Sigma^*$ é denotada por $|s|$, representando a quantidade de eventos que compõem s . Por exemplo, $|abcd| = 4$, $|\epsilon| = 0$, e assim por diante.

4.1.4 Linguagem

O conceito de linguagem, não difere da intuitiva noção de linguagem natural, falada. Considerando-se, o vocabulário relacionado a determinado idioma, existe um conjunto de palavras (cadeias) que são coerentes ou corretas no contexto do idioma. Tais palavras são compostas unicamente por letras (símbolos) do seu

alfabeto. Mas, também, é possível realizar a livre composição de novas palavras, embora elas possam não ser reconhecidas no vocabulário (TEIXEIRA, 2015).

A entidade que define esse vocabulário válido dá-se o nome de linguagem. Formalmente, uma linguagem L é um subconjunto de cadeias em Σ^* , i.e., $L \subseteq \Sigma^*$, que determina quais são as combinações possíveis de eventos em Σ que levam a cadeias reconhecidas em determinado contexto (vocabulário). Uma característica importante das linguagens formais é que elas podem contemplar um número infinito de cadeias, embora cada cadeia seja finita e oriunda de um alfabeto também finito. Por exemplo, seja $\Sigma = \{a, b\}$. Uma linguagem L que contempla todas as cadeias possíveis em Σ^* corresponde a $L = \{\epsilon, a, b, aa, bb, ab, ba, aab, \dots\}$. Nesse caso, L contém um número infinito de cadeias finitas.

Sendo uma linguagem um conjunto, por definição, então todas as operações convencionais sobre conjuntos - união, intersecção, potência, entre outras se aplicam às linguagens. Além dessas, outras operações podem ser definidas especificamente para esse contexto. Algumas delas são apresentadas a seguir.

4.1.5 Representação de SED's por linguagens

O comportamento de um sistema a eventos discretos (SED) pode ser descrito através de um par de linguagens. Para isto considera-se um alfabeto Σ como correspondendo ao conjunto de eventos que afetam o sistema. A evolução sequencial do SED, ou seu comportamento lógico, pode então ser modelado através de uma dupla $D = (L, L_m)$.

No modelo D , $L \subseteq \Sigma^*$ é a linguagem que descreve o comportamento gerado pelo sistema, ou seja, o conjunto de todas as cadeias de eventos fisicamente possíveis de ocorrerem no sistema. Por sua vez $L_m \subseteq L$ é a linguagem que descreve o comportamento marcado do sistema, ou seja, o conjunto de cadeias em L que correspondem a tarefas completas que o sistema pode realizar. O comportamento gerado de qualquer sistema a eventos discretos em que não ocorram eventos simultâneos, pode ser representado por uma linguagem prefixo fechada.

As observações acima podem ser sintetizadas formalmente nas seguintes propriedades de linguagens L e L_m que representam um SED:

- $L \supset L_m$, ou seja, o comportamento gerado contém o comportamento marcado de um SED;
- $L = \bar{L}$, ou seja, o comportamento gerado de um SED é prefixo-fechado.

4.2 Autômatos Determinísticos De Estados Finitos

Um Autômato Determinístico de Estados Finitos (ADEF) é uma estrutura formal de diagramação que permite representar problemas de diversas naturezas, tamanhos e complexidades, através de um método prático e intuitivo. Um ADEF pode ser formalmente apresentado como uma quintupla $\langle X, \Sigma, f, x_0, X_m \rangle$ onde:

- X é o conjunto finito de estados do autômato;
- Σ é o conjunto de símbolos (eventos) que definem o alfabeto;
- $f: X \times \Sigma \rightarrow X$ é a função de transição, possivelmente parcial, ou seja, não há necessidade da função ser definida para todo elemento de Σ em cada estado de X ;
- x_0 é o estado inicial;
- X_m é o subconjunto de estados marcados ou finais $X_m \subseteq X$;

Um autômato pode ser representado graficamente como um grafo dirigido, onde os nós representam os estados e os arcos etiquetados representam as transições entre os estados. O estado inicial é identificado através de uma seta apontando para ele e os estados finais são representados com círculos duplos. A figura 9 é um exemplo de um autômato determinístico, cuja descrição formal é representada por:

- $X = \{S0, S1, S2\}$
- $\Sigma = \{A, B, C\}$
- A função de transição é: $f(S0, A) = S1$, $f(S1, B) = S2$, $f(S1, C) = S1$,
 $f(S2, C) = S0$
- $x_0 = \{S0\}$
- $X_m = \{S0, S2\}$

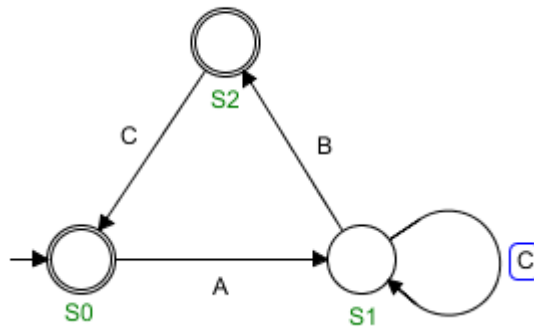


Figura 9– Autômato Determinístico
Fonte: Autoria própria *Software Supremica*.

Neste trabalho vão ser utilizados as estruturas dos ADEF's para representação da SED e o mesmo será modelado pelo *software* Supremica.

4.2.1 Linguagens Representadas Por Autômatos

Um autômato G está associado a duas linguagens, a linguagem gerada $L(G)$ e a linguagem marcada $L_m(G)$. A linguagem gerada por $G = \langle X, \Sigma, f, x_0, X_m \rangle$ é:

$$L(G) := \{s \in \Sigma^* : \hat{f}(x_0, s) \text{ é definida}\}.$$

A linguagem marcada de G é:

$$L_m(G) := \{s \in L(G) : \hat{f}(x_0, s) \in X_m\}.$$

A linguagem $L(G)$ representa todas cadeias que podem ser seguidas no autômato, partindo do estado inicial. A linguagem $L_m(G)$ considera todas as cadeias que partindo do estado inicial chegam a um estado marcado.

Um SED pode ser modelado por um autômato G , onde $L(G)$ é o comportamento gerado pelo sistema e $L_m(G)$ é o comportamento marcado ou conjunto de tarefas completas do sistema.

4.2.2 Acessibilidade, Co-Acessibilidade e Bloqueio

De forma geral, um ADEF $G = \langle X, \Sigma, f, x_0, X_m \rangle$ pode ter estados inacessíveis, isto é, estados que jamais podem ser alcançados a partir do estado inicial. Formalmente, um estado $x \in X$ é dito ser *acessível* se $x = f(x_0, u)$ para algum $u \in \Sigma^*$. G é dito acessível se x é acessível para todo $x \in X$.

Por outro lado G , é dito ser *co-acessível*, ou não bloqueante, se cada cadeia $u \in L(G)$ pode ser completada por algum $w \in \Sigma^*$ tal que $uw \in L_m(G)$, ou seja se cada cadeia $u \in L(G)$ for um prefixo de uma cadeia em $L_m(G)$.

Um ADEF é co-acessível se, a partir de qualquer um de seus estados, existir ao menos um caminho que leve a um estado marcado. A condição de Co-Acessibilidade de um ADEF pode ainda ser descrita pela equação.

$$L(G) = \overline{L_m(G)} \quad (4-1)$$

Quando um autômato é acessível e co-acessível ele é dito ser *trim*.

Um sistema a eventos discretos com comportamento $L(G)$ e $L_m(G)$ é dito ser *não bloqueante*, se satisfaz as condições da equação (4-1). Por outro lado um SED descrito por um autômato G que não satisfaz as condições da equação (4-1) será bloqueante. A condição de bloqueio ($L(G) \neq \overline{L_m(G)}$) corresponde à existência de cadeias geradas pelo sistema ($u \in L(G)$), a partir das quais não se pode completar alguma tarefa no sistema ($u \notin \overline{L_m(G)}$).

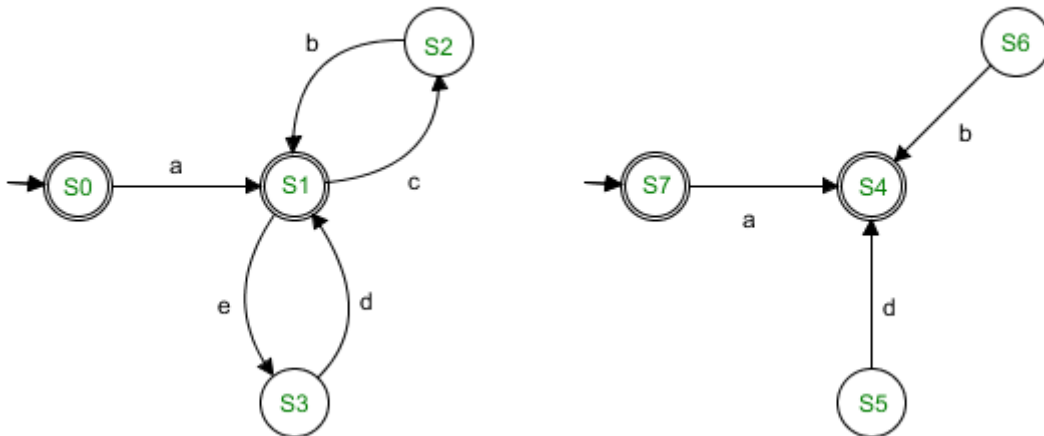


Figura 10– Autômato Acessível e o outro não acessível
Fonte: Autoria própria Software Supremica.

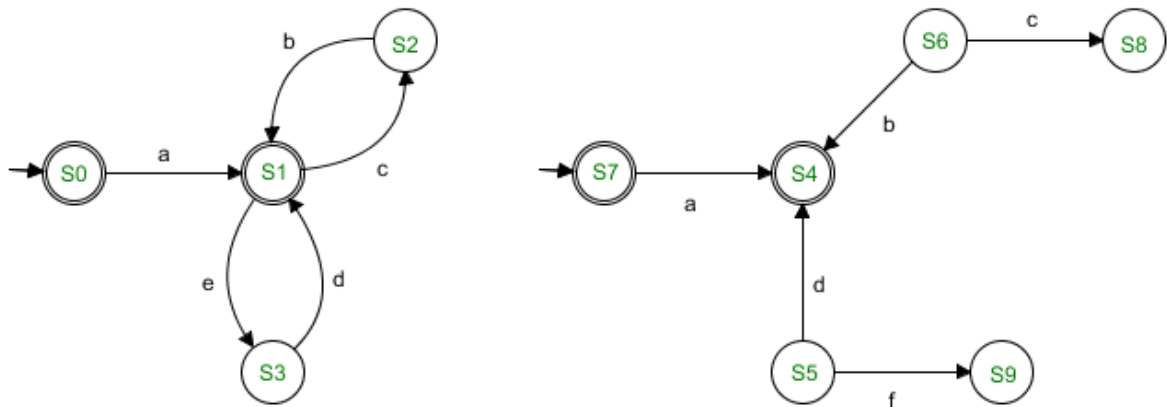


Figura 11– Autômato Co-Acessível e o outro Bloqueante
Fonte: Autoria própria Software Supremica.

Pode-se observar na figura 10, que no primeiro autômato consegue-se chegar a todos os estados a partir do estado inicial então ele é dito acessível, no segundo autômato da figura 10, os estados S5 e S6 não podem ser acessados a partir do estado inicial, então é dito que ele é não acessível. Na figura 11 no primeiro autômato observa-se que para cada estado consegue-se chegar em um estado marcado, no segundo autômato nos estados S8 e S9 não é possível chegar a algum estado marcado, portanto ele é um autômato bloqueante.

4.2.3 Composição de Autômatos

A aplicabilidade da abordagem local para modelagem de SEDs por ADEF é garantida pela operação de Composição de Autômatos, como definida a seguir. Dados dois autômatos $G_1 = \langle X_1, \Sigma_1, f_1, x_{0_1}, X_{m_1} \rangle$ e $G_2 = \langle X_2, \Sigma_2, f_2, x_{0_2}, X_{m_2} \rangle$ define-se a composição síncrona $G_1 || G_2$ como:

$$G_1 || G_2 = Ac(X_1 \times X_2, \Sigma_1 \cup \Sigma_2, f_{1||2}, (x_{0_1}, x_{0_2}), X_{m_1} \times X_{m_2})$$

Onde:

$$f_{1||2} : (X_1 \times X_2) \times (\Sigma_1 \cup \Sigma_2) \rightarrow (X_1 \times X_2)$$

Ou seja:

$$f_{1||2}((x_1, x_2), \sigma) = \begin{cases} (f_1(x_1, \sigma), f_2(x_2, \sigma)) & \text{se } \sigma \in \Sigma_1 \cap \Sigma_2 \text{ e } \sigma \in \Sigma_1(x_1) \cup \Sigma_2(x_2) \\ (f_1(x_1, \sigma), x_2) & \text{se } \sigma \in \Sigma_1 \text{ e } \sigma \notin \Sigma_2 \text{ e } \sigma \in \Sigma_1(x_1) \\ (x_1, f_2(x_2, \sigma)) & \text{se } \sigma \in \Sigma_2 \text{ e } \sigma \notin \Sigma_1 \text{ e } \sigma \in \Sigma_2(x_2) \\ \text{indefinida caso contrário} & \end{cases}$$

Um evento comum a Σ_1 e Σ_2 só pode ser executado sincronamente nos dois autômatos, os demais ocorrem assincronamente, ou seja, de modo independente em cada autômato. Se os alfabetos são iguais $\Sigma_1 = \Sigma_2$, a composição é completamente síncrona, isto é, todos os eventos estão sincronizados. No caso oposto $\Sigma_1 \cap \Sigma_2 = \emptyset$ não existe nenhuma sincronização entre os eventos dos dois autômatos.

4.3 Controle Supervisório de SED.

O conjunto de restrições de coordenação define uma especificação E a ser obedecida, e pode ser interpretado como segue. As linguagens $L(G)$ e $L_m(G)$ contêm cadeias indesejáveis de eventos por violarem alguma condição que se deseja impor ao sistema. Pode ser o caso de estados proibidos em G , por provocarem bloqueio ou por serem inadmissíveis, pode ainda ser o caso de cadeias que violam um ordenamento desejado para os eventos, como por exemplo no caso de justiça no acesso a recursos.

De modo a fazer com que os subsistemas atuem de forma coordenada, introduz-se um agente de controle denominado supervisor, denotado por S . Em nosso paradigma, considera-se que o supervisor S interage com a planta G , numa estrutura de malha fechada (figura 12) onde S observa os eventos ocorridos em G e define que eventos, dentre os fisicamente possíveis de ocorrerem no estado atual, são permitidos de ocorrerem a seguir. Mais precisamente, S tem uma ação desabilitadora de eventos e, neste sentido diz-se que S é um controle de natureza permissiva. O conjunto de eventos habilitados num dado instante pelo supervisor define uma entrada de controle. Esta é atualizada a cada nova ocorrência de evento observada em G .

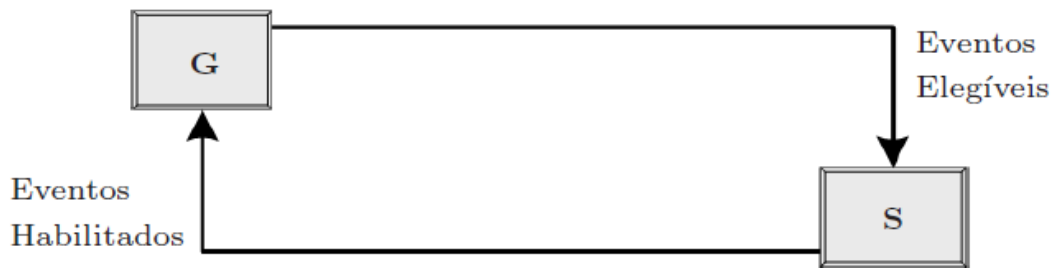


Figura 12 – SED em malha fechada.
Fonte – Adaptado Notas De Aula, (TEIXEIRA,2015).

Considera-se ainda que o conjunto de eventos que afetam a planta G é particionado num conjunto de eventos desabilitáveis, ou controláveis, e um conjunto de eventos cuja natureza não permite a desabilitação.

Finalmente, será considerado aqui que o supervisor pode desmarcar estados marcados da planta G , ou seja, não reconhecer como tarefa do sistema em malha fechada, uma cadeia que corresponde a uma tarefa do sistema em malha aberta. A seguir, o problema de controle será reapresentado, agora formalmente, juntamente com a noção de supervisor.

4.3.1 Controlabilidade e Solução do PCS

Dada uma planta representada por um ADEF G com comportamento dado pelo par de linguagens $(L(G), L_m(G))$ definidas sobre um conjunto de eventos Σ , define-se uma estrutura de controle Γ para G , pelo particionamento de Σ em:

$$\Sigma = \Sigma_c \dot{\cup} \Sigma_u$$

Onde:

- Σ_c é o conjunto de eventos controláveis, que podem ser inibidos de ocorrer no sistema;
- Σ_u é o conjunto de eventos não controláveis, que não podem ser inibidos de ocorrer no sistema.

Uma opção de controle $\gamma \in \Gamma$ aplicada ao sistema contém o conjunto ativo de eventos habilitado a ocorrer no sistema. Com a definição da controlabilidade de eventos, a estrutura de controle toma a forma:

$$\Gamma = \{\gamma \in 2^\Sigma : \gamma \supseteq \Sigma_u\}$$

Onde a condição $\gamma \supseteq \Sigma_u$, indica simplesmente que os eventos não controláveis não podem ser desabilitados. Um supervisor pode ser representado por um autômato S , definido sobre o mesmo alfabeto Σ , cujas mudanças de estado são ditadas pela ocorrência de eventos na planta G . A ação de controle de S , definida para cada estado do autômato, é desabilitar em G os eventos que não possam ocorrer em S após uma cadeia de eventos observada. O funcionamento do sistema controlado $S=G$ pode ser descrito pelo SED resultante da composição síncrona de S e G , isto é, $S||G$. De fato, na composição síncrona $S||G$, somente as transições permitidas tanto no sistema controlado G , como no supervisor S são permitidas. O comportamento do sistema em malha fechada é então dado por:

$$L(S/G) = L(S||G) \quad \text{e} \quad L_m(S/G) = L_m(S||G)$$

Por (CURY, 2001) O Problema de Controle Supervisório pode ser formalmente apresentado como segue, Dada uma planta G , com comportamento $(L(G), L_m(G))$ e estrutura de controle Γ , definidos sobre o conjunto de eventos Σ ; e especificações definidas por $A \subseteq E \subseteq \Sigma^*$; encontre um supervisor não bloqueante S para G tal que:

$$A \subseteq L_m(S/G) \subseteq E$$

As especificações A e E definem limites superior e inferior para o comportamento do sistema em malha fechada.

Essencial para a solução do problema de controle supervisório é o conceito de controlabilidade de linguagens, definidos sobre o conjunto de eventos Σ e a linguagem $E \subseteq L(G)$, E é dita ser controlável com respeito a G , ou simplesmente controlável, se:

$$\bar{E}\Sigma_u \cap L \subseteq \bar{E}$$

O seguinte resultado é fundamental para a solução do problema de controle supervisório como estabelecido anteriormente.

Segundo (CURY, 2001). Dada uma planta G , com comportamento $(L(G), L_m(G))$ e estrutura de controle Γ , definidos sobre o conjunto de eventos Σ e $K \subseteq L_m(G)$, existe um supervisor não bloqueante S para G tal que :

$$L_m(S/G) = K$$

Se e somente se K for controlável.

Para uma linguagem $E \subseteq \Sigma^*$, seja $C(E)$ o conjunto de linguagens controláveis contida em E . Pode-se provar que $C(E)$ possui um elemento supremo $supC(E)$. O seguinte teorema fornece a solução para o problema de controle supervisorio.

O problema de controle supervisorio possui solução se e somente se

$$SupC(E) \supseteq A$$

No caso das condições acima serem satisfeitas, $supC(E)$ representa o comportamento menos restritivo possível de se implementar por supervisão no sistema G , satisfazendo as especificações A e E . Assim, o supervisor ótimo S é tal que $L_m(S/G) = SupC(E)$.

4.3.2 Síntese de supervisores ótimos

A síntese de um supervisor ótimo foi inicialmente proposta por Ramadge e Wonham (RW), e se baseia em três passos. Estes passos serão desenvolvidos a seguir: (CURY, 2001)

- Obtenção de um modelo para a planta a ser controlada;
- Obtenção de um modelo de representação das especificações a serem respeitadas;
- Síntese de uma lógica de controle não bloqueante e ótima.

4.3.3 Modelo Para Plantas

A modelagem de um SED por ADEFs, pode ser em princípio abordada de duas maneiras: uma abordagem global e uma abordagem local. Na abordagem global o sistema é analisado como um todo e procura-se por um ADEF que represente todas

as sequências possíveis de eventos que ele pode gerar e tarefas que pode completar. Para sistemas de maior porte, esta pode ser uma tarefa de grande complexidade. Além disso, qualquer alteração no sistema, por exemplo, pela inclusão ou retirada de equipamentos, ou modificação em sua lógica de controle, requer a reconstrução do modelo como um todo.

Por outro lado, como já discutimos, em geral um SED pode ser visto como a composição de subsistemas, atuando em geral sob ações de um controlador que impõe restrições de coordenação ao sistema. A abordagem local de modelagem parte do princípio de que se pode construir modelos locais para cada subsistema e restrição de coordenação, e que se pode compor os mesmos para obter um modelo do sistema global. Uma abordagem de modelagem localizada, sugere maior facilidade na obtenção de modelos de sistemas de grande porte. Além disso, permite pressupor que alterações num subsistema ou em alguma restrição somente exigirão uma mudança no modelo específico correspondente.

Para modelagem do problema proposto neste trabalho, vai ser utilizado uma abordagem local devido à complexidade do sistema e a facilidade de fazer alterações em algum subsistema, tornando-se possível a incrementar restrições ou alterações na planta sem que seja necessário refazer todo o trabalho, tendo uma maior modularidade.

Para obtenção da planta a ser controlada podemos seguir alguns passos:

1. Identificar o conjunto de subsistemas, equipamentos, sensores etc.
2. Construir o modelo básico ADEF G_i , de cada elemento i envolvido no sistema;
3. Obter o modelo da planta a ser controlada, através da composição síncrona de todos os ADEF G_i , que foi apresentado no tópico 4.2.3;

Para demonstrar a modelagem de uma planta vamos supor o funcionamento de duas máquinas M1 e M2, para isso devemos identificar os estados de funcionamento normal dessas máquinas que são: Ligada e Desligada. A figura 13 é o modelo da planta para as duas máquinas.

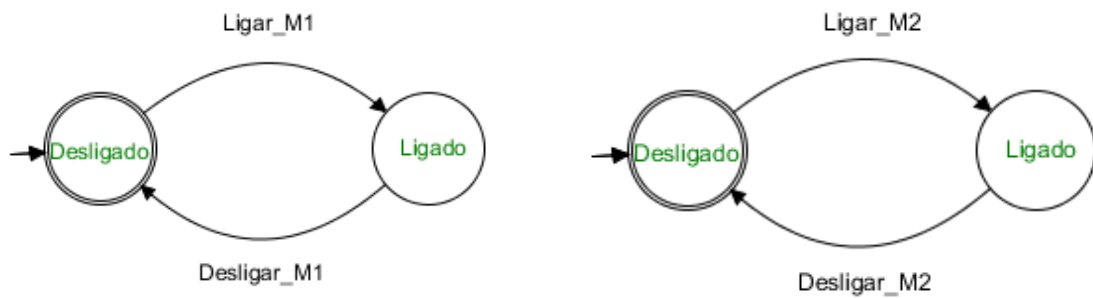


Figura 13 – Autômato das plantas das máquinas M1 e M2.
Fonte: Autoria própria Software Supremaica.

Denominando de G1 e G2 as plantas das máquinas M1 e M2 respectivamente, e para obtenção da planta geral é feita a composição dos dois autômatos mostrado na figura abaixo.

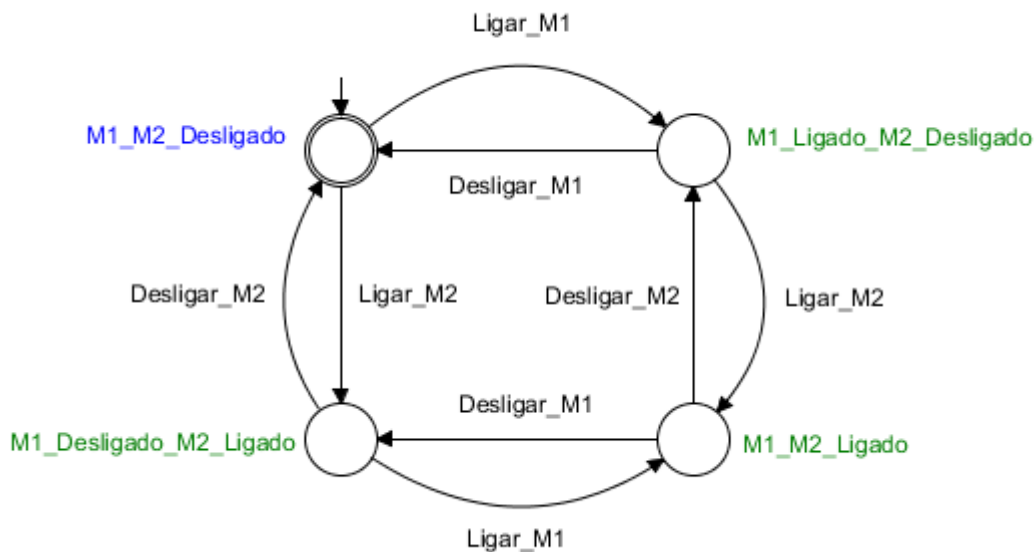


Figura 14– Composição síncrona das plantas M1 e M2.
Fonte: Autoria própria Software Supremaica.

4.3.4 Modelo Para Especificações

Uma vez modelada a planta em malha aberta, o próximo passo é fechar a malha. Para isso, são implementadas restrições as quais espera-se que o sistema cumpra quando em funcionamento. Restringir uma planta passa, basicamente, pela

definição do que restringir e, em seguida, pela implementação da semântica de cada uma das restrições em forma de especificações de controle.

Uma especificação nada mais é do que a representação de uma ação proibitiva no sistema. Quando associada ao modelo da planta, ela interfere em determinados eventos possíveis em malha aberta, adequando o comportamento da planta aos requisitos de controle. Não costuma ser a meta de uma especificação enumerar as possíveis ações no sistema, uma vez que isso, em tese, já faz parte do comportamento da planta. Objetiva-se, por outro lado, cumprir um requisito de controle simplesmente observando eventos na planta e desabilitando aqueles que são considerados proibidos, quando eles puderem ocorrer.

O resultado da ação de uma especificação sobre uma planta remete ao que pode ser informalmente descrito como o comportamento desejado para o sistema sob controle, ou em malha fechada. Assim como os modelos de subsistemas, também as especificações podem ser representadas individualmente através de um autômato, expressando os requisitos de controle na forma de desabilitação de eventos em estados. Os passos para a obtenção da especificação global E , seguindo uma abordagem local, são os seguintes:

1. Construir autômatos E_j para cada restrição de coordenação j , do sistema a ser controlado;
2. Realizar a composição dos autômatos construídos no passo 1; compor o autômato resultante com o autômato da planta Global, gerando o autômato R ;
3. Atualizar R pela eliminação, caso houver, de estados considerados proibidos;
4. Atualizar R pelo cálculo de sua componente co-acessível, ou não bloqueante.

A especificação global E é dada por: $E = L_m(R)$

A Figura 12 mostra um exemplo simples de um modelo de especificação para a planta $G1$ e $G2$ vistas anteriormente.

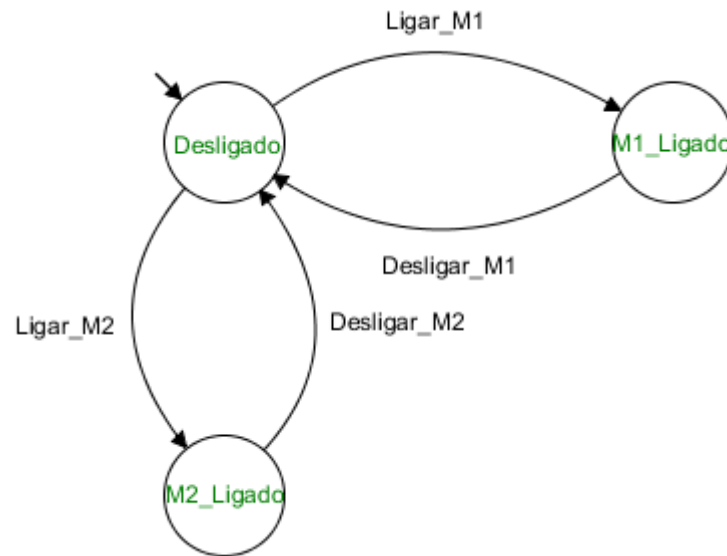


Figura 15 – Autômato da restrição sobre as máquinas
Fonte: Autoria própria *Software Supremica*.

A especificação na figura 14 representa a restrição do funcionamento das duas máquinas simultaneamente, ou seja, uma máquina só pode ser ligada se a outra estiver desligada, a figura 16 é a composição das especificações com a composição da planta.

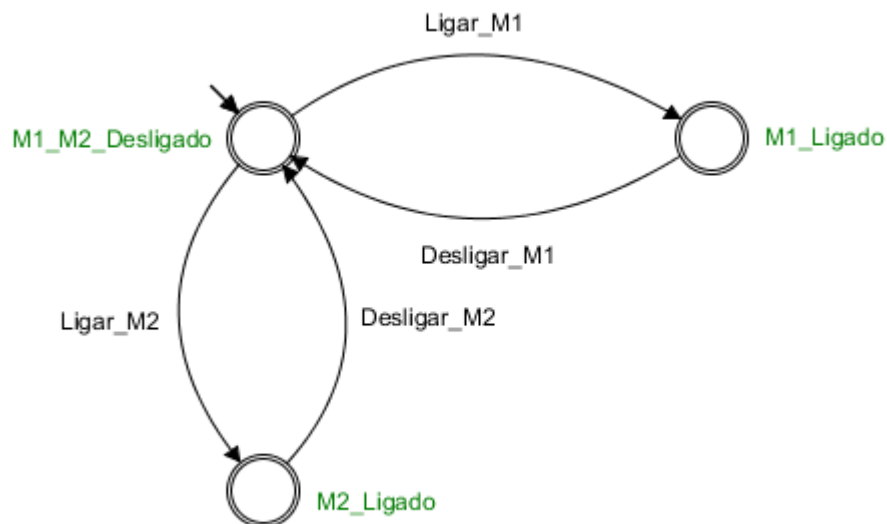


Figura 16 - Composição da restrição com a planta geral
Fonte: Autoria própria *Software Supremica*.

Observa-se na figura 15 e 14 que a especificação global é uma subcadeia do modelo da planta geral e sendo ela acessível e co-acessível, ou seja, não bloqueante não houve a remoção de nenhum estado na composição final.

4.3.5 Algoritmo para cálculo do SupC.

Para implementar um supervisor ótimo e não bloqueante, primeiramente obtém-se a planta geral e as especificações descritas nas seções 4.3.3 e 4.3.4.

O último passo na resolução do Problema de Controle Supervisório (PCS) é a síntese do supervisor S que implementa a lógica não bloqueante ótima, no sentido de menos restritiva possível. Caso $E \subseteq L_m(G)$ seja controlável, as proposições acima garante que existe um supervisor não bloqueante tal que $L_m(S/G)$. No entanto, nem sempre E atende a condição de controlabilidade, o que torna necessário que se calcule a linguagem controlável que mais se aproxime de E , e que não contenha cadeias indesejáveis de eventos. Esta linguagem é dada por $supC(E)$ e representa a lógica ótima de supervisão.

O cálculo de $supC(E)$ baseia-se num procedimento iterativo que identifica “maus estados” no autômato R que modela E , obtido no segundo passo da metodologia, como descrito na secção 4.2.3

Algoritmo de cálculo de $supC(E)$

Dados $G = \langle X, \Sigma, f, x_0, X_m \rangle$ e $R = \langle Q, \Sigma, f_R, q_0, Q_m \rangle$ obtidos como descrito nas seções 4.2.3 e 4.2.4, tais que $L_m(R) = E \subset L_m(G)$;

1. Identificar maus estados em R ; caso não existam faça $S = R$, fim.
 2. Caso existam, atualizar R por eliminação dos maus estados;
 3. Calcular a componente trim de R e voltar ao passo 1.
-

Um estado de R é mau estado se existe $\sigma \in \Sigma_u$ tal que $\sigma \in \Sigma_G(x)$ e $\sigma \notin \Sigma_R(q)$.

No trabalho a síntese do supervisor ótimo e não bloqueante e a composição dos autômatos são feitos pelo *Software Supremica*.

5. METODOLOGIA

Neste trabalho propõem-se uma metodologia e utiliza-se da teoria de controle supervisorio para modelagem de problemas de *self-healing* em *smart grid*. A modelagem visa a redução de estados (número de chaves manobráveis para a reconfiguração), para que sirva de auxílio aos programas de recuperação de faltas, tais como: Meta-Heurística, Força bruta e híbridos.

Utiliza-se do trabalho de (Muriel, 2016), que fez a implementação de algoritmos meta heurísticos para resolução de problemas de auto reconfiguração em redes elétricas, para obter alguns parâmetros de reconfiguração tais como: redes utilizadas, melhores reconfigurações encontradas pelos algoritmos, etapas da reconfiguração e os métodos para cálculo de fluxo de potência, porém a modelagem não utiliza o fluxo de potência nem métodos de otimização e os parâmetros observados no trabalho serão para definir os elementos modelados.

Para iniciar a modelagem é necessário identificar os elementos da rede elétrica e suas premissas. Assume-se que a rede de distribuição não é alterada frequentemente em termos da sua topologia geográfica, altera-se somente o acréscimo ou decréscimo de cargas, além disso é estática no momento em que a falta ocorre, ou seja, possui um arranjo fixo de chaves, alimentadores e cargas energizadas, o sistema fica inerte esperando que uma falta ocorra para iniciar as operações de isolamento e reconfiguração das regiões afetadas. Após determinar as condições da rede o próximo passo são os elementos a se modelar:

- O evento falta que não é controlável, pois o mesmo depende de um estímulo externo esporádico para ocorrer, sendo representado por um conjunto de sensores que detectam a falta e geram um sinal para o sistema, onde ele identifica a região afetada. O retorno da falta ocorre pelo operador que identifica que o problema já foi solucionado e que o sistema já está apto a retornar da falta, logo o evento retorno da falta é um evento controlável.
- Os alimentadores, cada alimentador tem a função de indicar o sentido do fluxo de potência e energizar as cargas.

- Cargas, cada carga pode ser tratada separadamente como uma planta, no entanto devido ao elevado número de cargas isto levaria a muitos estados e transições, uma solução é agrupar todas cargas que estão entre as chaves denominando-as de regiões, reduzindo os eventos e transições, porém perde-se informações sobre as cargas individuais o que não é um problema já que o objetivo é as relações entre as chaves.
- Chaves controladas, fisicamente são utilizadas para manobras, energizando ou desenergizando cargas, podendo alterar qual alimentador vai energizar determinado conjunto de cargas e assim determinar o sentido do fluxo de potência.

Após determinar os elementos a segunda etapa é a modelagem das suas respectivas plantas, onde contém seu funcionamento normal de operação. O primeiro elemento é a falta que é um evento não controlável e ocorre de forma assíncrona e esporádica e o retorno da falta é um evento controlável e determinado pelo operador, a figura 17 mostra a modelagem da planta do conjunto de sensores que detectam a falta em uma determinada região.

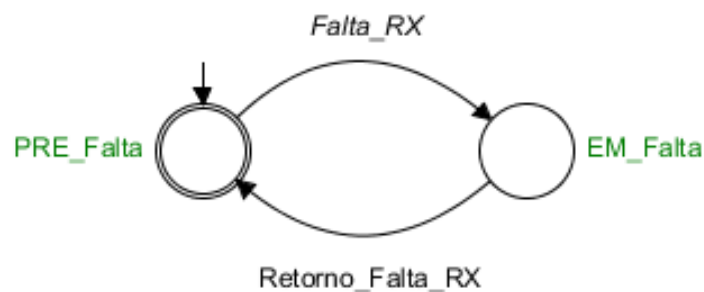


Figura 17– Planta do conjunto de sensores que detectam a falta na região RX.
Fonte: Autoria própria *Software Supremica*.

As regiões são o conjunto de cargas entre as chaves e possuem dois estados, energizadas e desenergizadas, similar a ligar e desligar uma lâmpada, sendo as regiões um conjunto de lâmpadas, as chaves sendo os interruptores e os alimentadores sendo uma fonte de energia. Os alimentadores vistos de uma perspectiva estática sempre vão estar fornecendo energia, ou seja, sempre vão estar ligados, na figura 17 estão as plantas destes dois elementos do sistema.

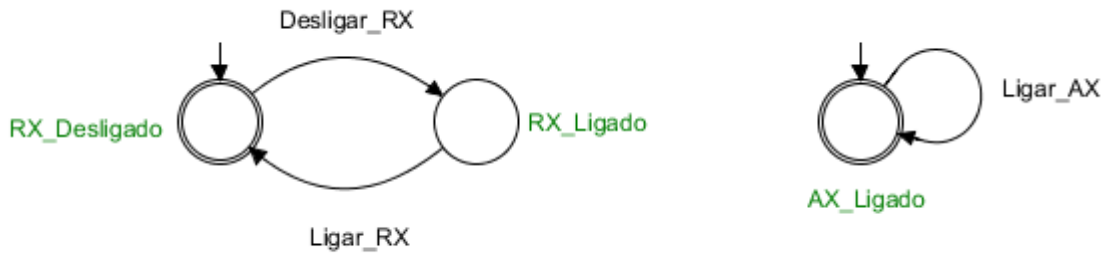


Figura 18 – Planta da região e alimentador respectivamente.
Fonte: Autoria Própria Software Supremica.

Por fim o modelo das chaves, onde elas podem conduzir em duas direções para determinar o sentido do fluxo, se um lado estiver aberto o outro tem que estar fechado. Além disso as chaves que já estão fechadas têm estado inicial fechado no sentido em que o fluxo estará passando e as chaves que estão inicialmente abertas são denominadas de *Tie Switch*. O sentido do fluxo é determinado pelas transições por exemplo: *Fechar_CHY_X*, significa que a chave foi fechada e o sentido do fluxo é da região Y para X.

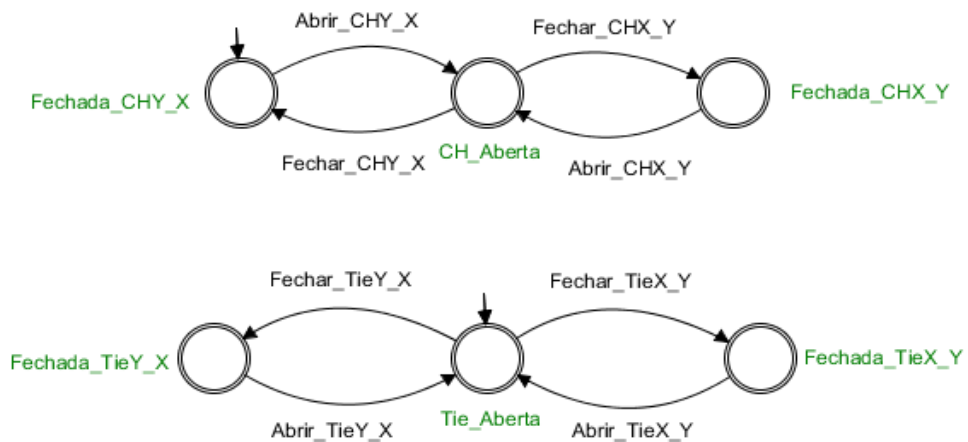


Figura 19 – Planta das chaves.
Fonte: Autoria Própria Software Supremica.

Na figura 18, tem-se uma rede de distribuição, onde os elementos acima modelados foram nomeados, divididos e destacados, a figura 18 é uma rede para testes e dispõem de 22 chaves 4 alimentadores e 18 regiões contendo 72 cargas.

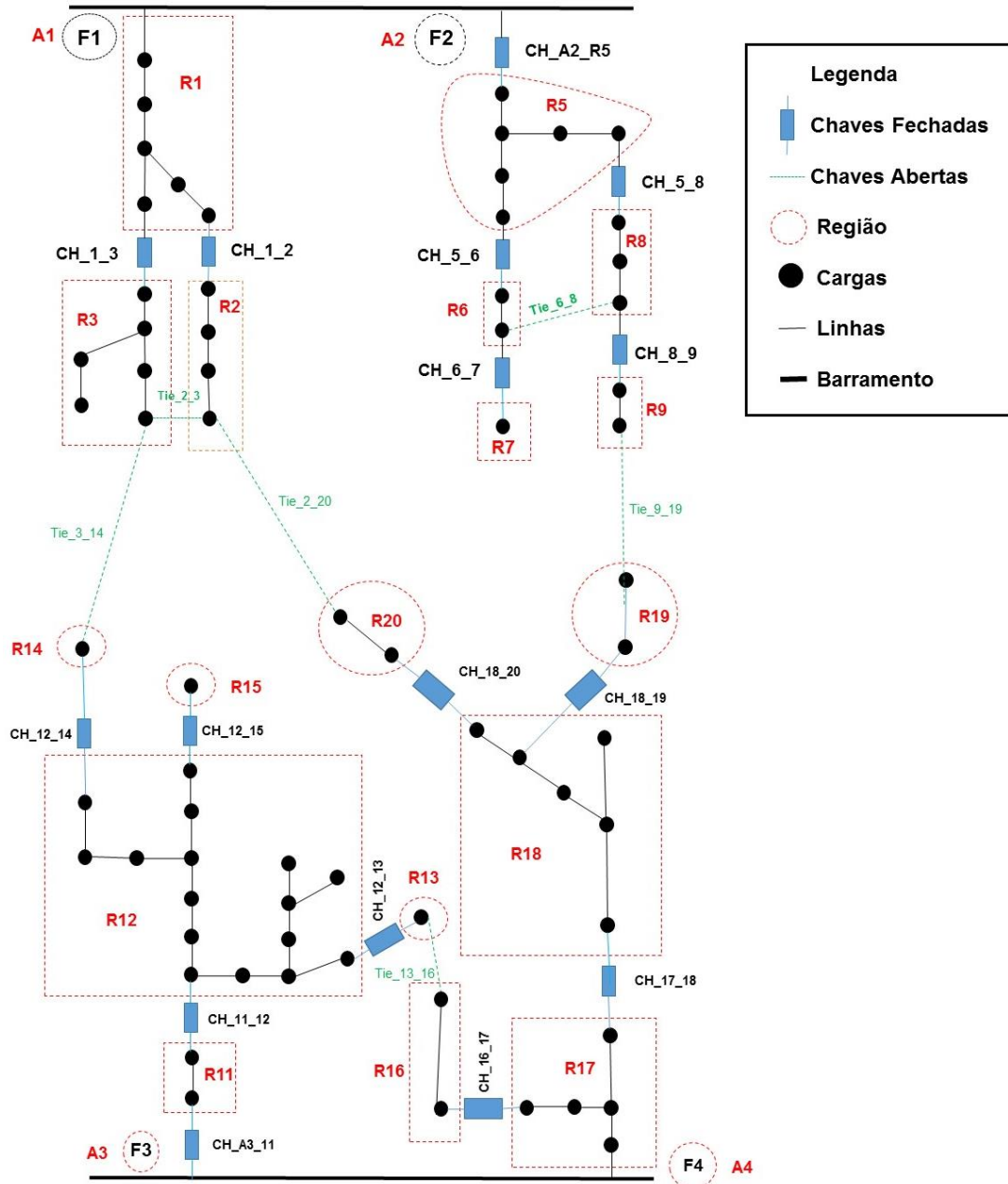


Figura 20 – Rede de distribuição para testes.
Fonte: Autoria própria

Pode-se tratar uma rede de distribuição com chaves controláveis como um problema de combinatória, onde o arranjo de chaves, abertas ou fechadas, que determinam a configuração da rede, e este arranjo depende de algumas restrições que são impostas, tais como, uma carga sendo alimentada por um barramento pode receber alimentação de outro barramento, por exemplo: as chaves *Tie_3_14*, *Tie_13_16* na figura 19, ou laços na alimentação, ou seja, se uma determinada carga, passe a ser alimentada pelo mesmo barramento por dois caminhos diferentes como ilustra na figura 19 se a chave *Tie_2_3* for fechada sem a abertura da *CH1_3* ou

CH1_2. Outra restrição é que todas as regiões, exceto em caso de falta, precisam ser alimentadas, outras especificações como limite de transmissão das linhas, limite máximo e mínimo da variação de tensão é necessário ter dados de fluxo de potência, então não serão abordadas na modelagem.

Pela combinatória tem-se, 22 chaves e cada chave tem dois estados lógicos possíveis que são *ON* e *OFF*. A tabela abaixo ilustra as chaves e seus respectivos número de estado lógico.

Tabela 1–Número de estado lógico para cada Chave

<i>ID</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
<i>Estados</i>	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Tabela 2 – Estados da Rede da Figura 19

CHAVES	STATUS	CHAVES	STATUS
CH_1_2	1	CH_17_18	1
CH_1_3	1	CH_18_19	1
CH_5_6	1	CH_18_20	1
CH_5_8	1	CH_16_17	1
CH_6_7	1	CH_A2_5	1
CH_8_9	1	Tie_3_20	0
CH_A3_11	1	Tie_13_16	0
CH_11_12	1	Tie_9_19	0
CH_12_14	1	Tie_6_8	0
CH_12_15	1	Tie_2_3	0

Fonte – Autoria Própria

Analisando o problema das chaves apresentadas na tabela 1, por uma perspectiva puramente probabilística, uma rede com 22 chaves teria 2^{22} possibilidades de reconfiguração, porém as redes possuem regras que delimitam a quantidade de chaves, a tabela 2 ilustra essas regras, onde são necessárias uma quantidade mínima de chaves para que todas as cargas da rede sejam alimentadas, outra regra que delimita esse conjunto de chaves é a topologia da rede, por isso que o fechamento das chaves de id “*Tie*”, são dependentes da abertura de alguma das chaves que estão diretamente relacionadas com a mesma, por exemplo para fechar *Tie_2_3*, é necessário abrir a chave *CH_1_2* ou *CH_1_3*. Quanto mais regras, mais delimitado ficará o número de combinações por consequente mais rápido será a reconfiguração.

As etapas para o processo de *self-Healing*, são divididas em três partes que são:

1. Detecção da Falta, é determinado o local aonde a mesma ocorreu;
2. Isolamento da Falta, é isolado a região em que a falta ocorreu das outras regiões;
3. Reconfiguração das chaves para abastecimento das regiões desabastecidas;

Para que o sistema possa ser reconfigurado após uma falta, é necessário isolar o local do incidente do restante da rede, buscando minimizar a quantidade de cargas próximas afetadas. Ao isolar as redondezas da falta se previne que o problema possa se alastrar aos demais componentes, danificando aparelhos da rede. Também é necessário deixar o local transparente para o algoritmo, ou seja, como se não fizesse parte do sistema, de modo que ele não exerça nenhuma alteração dentro dessa região. A Figura 21 mostra um exemplo utilizado em (MAZZETTO, 2016).

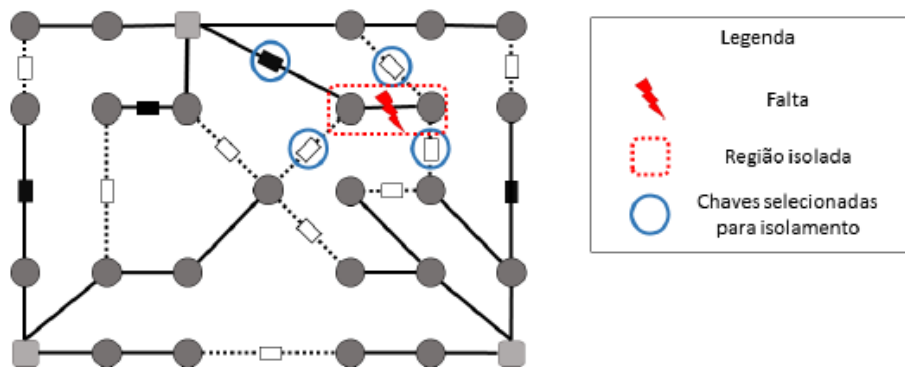


Figura 21 – Detecção e isolamento da falta
Fonte: MAZZETTO(P.30,2016)

A Detecção da falta é constituído pelo conjunto de sensores que uma região possui, então abstrai-se o processo de detecção ilustrando-o como um sinal de controle que um conjunto de sensores envia para o sistema determinando em qual região ocorreu uma falta, a modelagem para o funcionamento desses sensores é ilustrada na figura 18 e cada região possui seus sensores. Após a ocorrência da falta em uma determinada regiões é realizado o processo de isolamento, que consiste em abrir todas as chaves que estão ligadas a aquela região não permitindo que as

mesmas sejam fechadas. Cada região terá pré-definidas as chaves que deverão ser abertas na ocorrência de falta.

Após o isolamento da região, define-se quais as combinações, possíveis e necessárias para a reconfiguração das chaves, visando um novo caminho para alimentar as cargas que foram desenergizadas na etapa anterior. A figura 22 ilustra a falta na região R1 e o isolamento da mesma, habilitando a manobra nas chaves *Tie_2_3*, *Tie_3_14* e *Tie_3_2*, para que um novo caminho seja traçado para alimentar as regiões desabastecidas, e as regiões *R20* e *R14* serão como “Alimentadores” para *R2* e *R3*.

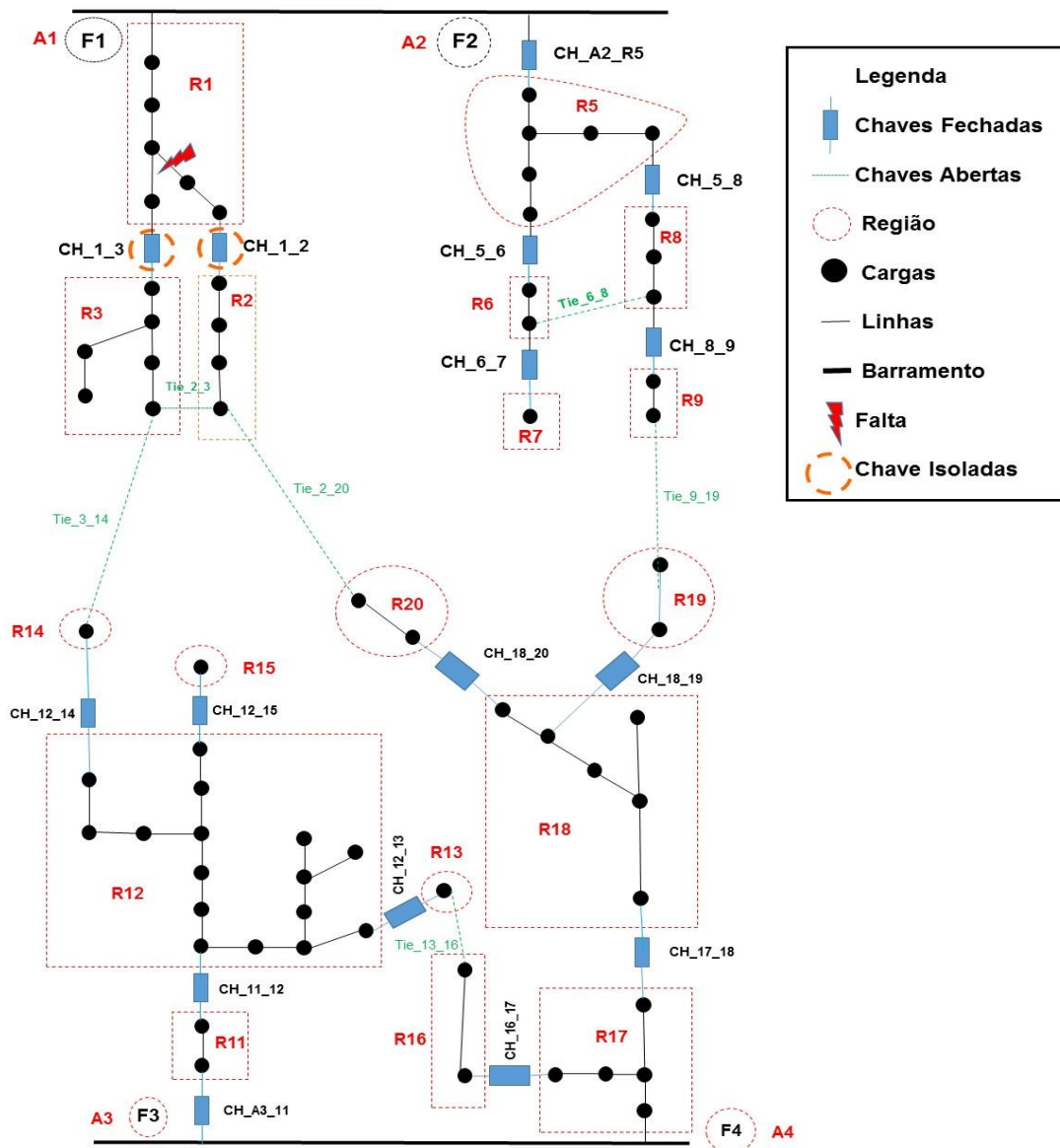


Figura 22 - Rede Elétrica Após a Falta na Região 1
Autor: Autoria própria.

6. RESULTADOS

Para aplicar a metodologia, simula-se um evento de falta na região R1, cada falta e reconfiguração precisa ser analisada e modelada separadamente. Seguindo os passos descritos na metodologia, Primeiro é necessário delimitar a região afetada pela falta, para determinar os elementos da modelagem, onde:

- R1 – Região isolada;
- R2 e R3 – Regiões sem energia;
- R14 e R20 – Regiões alimentadoras;
- CH_1_2 e CH_1_3 – Chaves abertas permanente
- *Tie_2_3*, *Tie_2_20* e *Tie_3_14* – Chaves que podem ser manobradas para abastecer a região;

A figura 23 ilustra a região a ser modelada e os elementos acima citados.

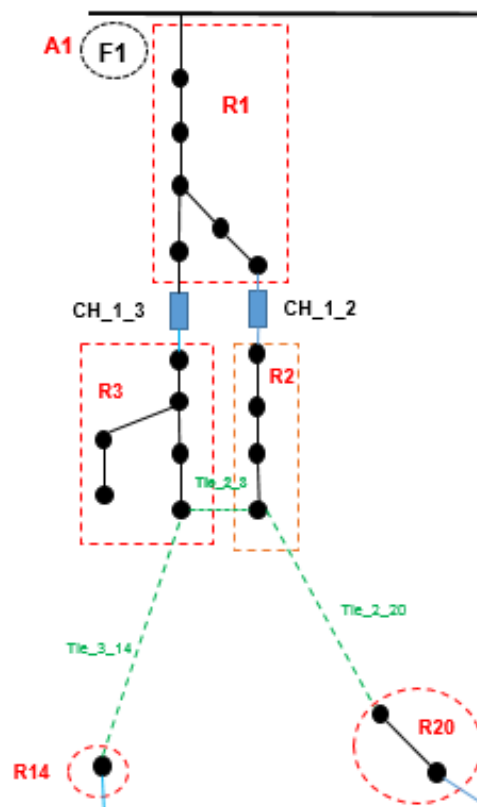


Figura 23 – Região da rede modelado
Fonte: Autoria própria.

Para a síntese do supervisor é preciso obter a planta geral que consiste na composição das plantas locais e o conjunto especificações para cada subsistema. Após a determinação das plantas, mostrado no capítulo 5, não é necessário alterar o mesmo, apenas adaptar para cada elemento, por exemplo: todas regiões vão possuir a mesma lógica de funcionamento, logo os autômatos resultantes de todas as regiões vão possuir o mesmo formato.

A figura 24 mostra os autômatos das regiões modeladas, sendo elas *R1*, *R2* e *R3*, pode-se observar que as plantas para todas as regiões são iguais, isto ocorre pelo fato de que a modelagem local é modular.

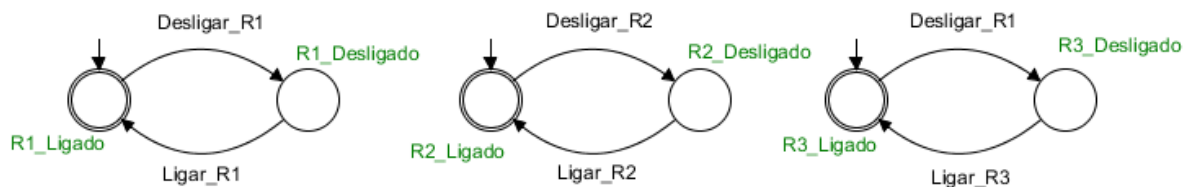


Figura 24 - Planta das regiões R1, R2 e R3.
Fonte – Autoria própria Software Supremica

As regiões *R14* e *R20* estão energizadas pelos alimentadores *A3* e *A4* respectivamente e essas regiões estão ligadas as regiões *R2* e *R3*, após o isolamento os alimentadores *A3* e *A4*, irão suprir as cargas das regiões *R2* e *R3* através das regiões *R14* e *R20*, por esse motivo elas são denominadas de regiões alimentadoras. Na figura 24 estão as plantas do alimentador A1, das regiões alimentadoras *R14* e *R20* e para o sensor de detecção de falta.

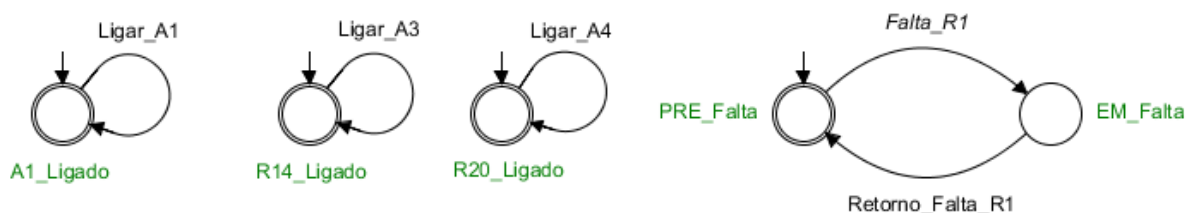


Figura 25 - Planta do alimentador A1, das regiões alimentadoras *R20* e *R14* e falta em *R1*
Fonte: Autoria própria Software Supremica.

As chaves possuem duas formas de plantas, as de ID *CH*, possui estado inicial fechada indicando qual o sentido do fluxo, exemplo estado “*Fechada_CHY_X*”, indica que a chave está fechada e o sentido do fluxo é da região *Y* para a região *X*, as chaves de ID *Tie*, possui estado inicial aberto e pode ser fechada indicando o sentido

do fluxo, exemplo: do estado inicial vai para “*Fechada_TieY_X*” ou “*Fechada_TieX_Y*”, fisicamente a chave não direciona o sentido do fluxo, porém na modelagem é necessário indicar o sentido através das chaves. A figura 26 e figura 27 estão as plantas das chaves.

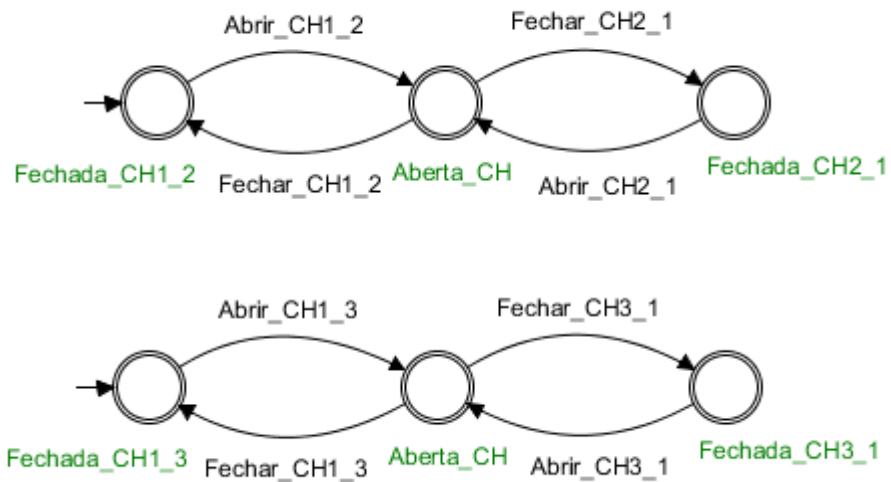


Figura 26 – Planta das chaves CH1_2/CH2_1 e CH1_3/CH3_1.
Fonte – Autoria própria Software Supremica.

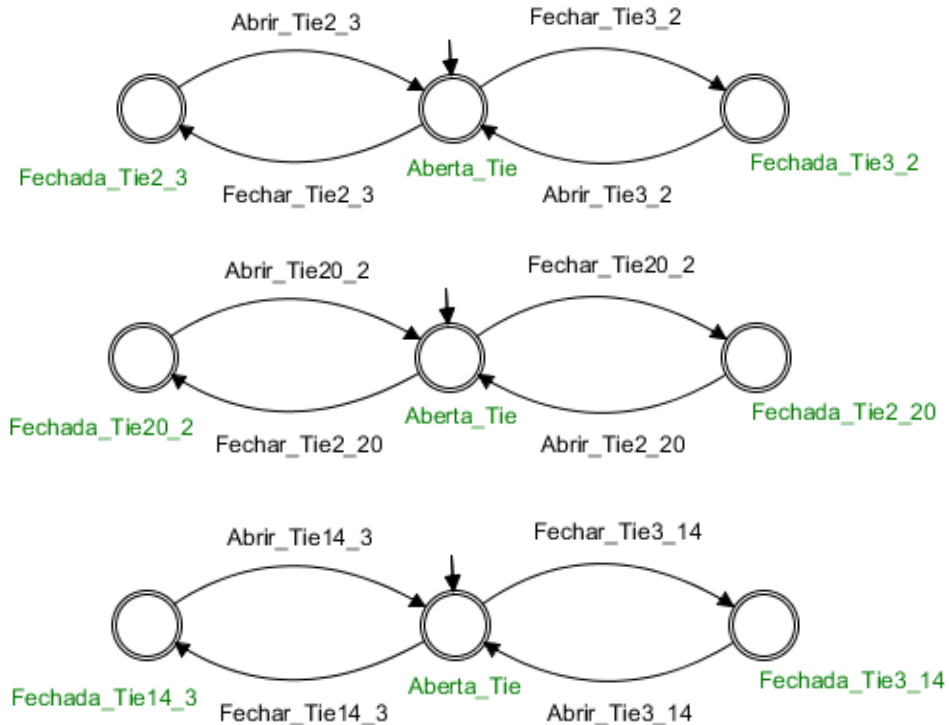


Figura 27 – Planta das chaves Tie3_2/2_3, Tie20_2/2_20 e tie14_3/3_14
Fonte – Autoria própria, Software Supremica.

Para modelar a planta de outras regiões, chaves ou alimentadores o processo é análogo. O segundo passo para determinar o supervisor é identificar e modelar as especificações para cada subsistema, sendo elas: falta, topologia da rede radial, fechamento de chaves que não estão energizadas, energizar as cargas.

Abaixo estão descritos cada especificação com a explicação da ação de controle que a mesma irá fazer na planta.

- **E1:** O sistema fica inativo esperando uma falta, quando ela ocorrer, abrir as chaves que estão ligadas à região e não permitir o fechamento das chaves durante a reconfiguração.
- **E2:** Quando alguma das chaves que fazem ligação com a região R2, forem abertas a região R2 vai para o estado *R2_Desligado*.
- **E3:** Quando alguma das chaves que fazem ligação com a região R3, forem abertas a região R3 vai para o estado *R3_Desligado*.
- **E4:** Quando a região R2 for desligada, permitir o fechamento de uma das chaves que fazem ligação com a região.
- **E5:** Quando a região R3 for desligada, permitir o fechamento de uma das chaves que fazem ligação com a região.
- **E6:** Somente fechar uma chave se a região que ela pretende energizar estiver sem energia, para evitar laços e conflitos entre alimentadores. Somente fechar uma chave se uma das regiões que ela interliga estiver energizada.

As figuras abaixo representam as especificações acima, tanto as plantas quanto as especificações foram feitas no *software* Supremica.

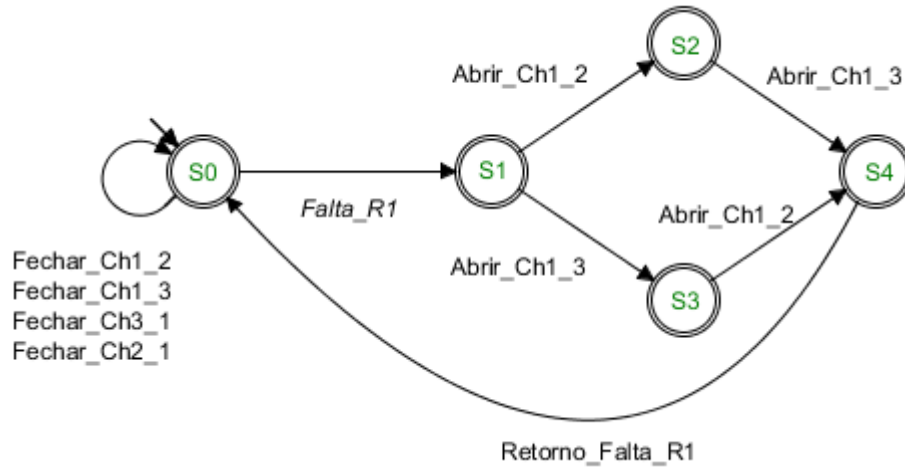


Figura 28 - Especificação E1.
Fonte: Autoria Própria *Software* Supremaica.

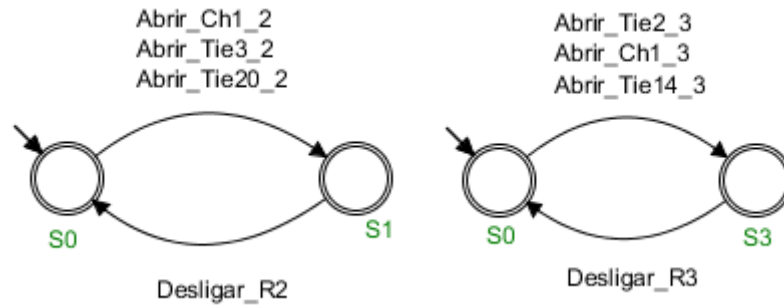


Figura 29 - Especificações E2 e E3.
Fonte – Autoria Própria *Software* Supremaica

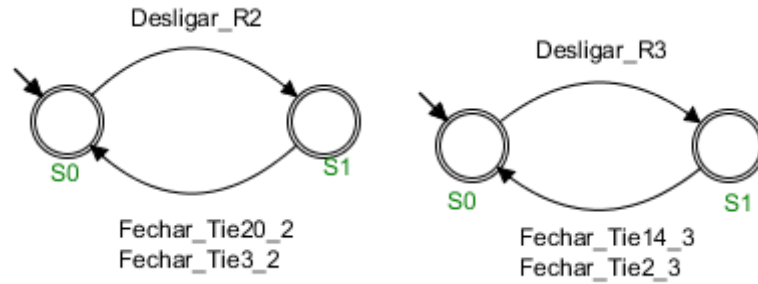


Figura 30 – Especificações E4 e E5.
Fonte – Autoria Própria *Software* Supremaica

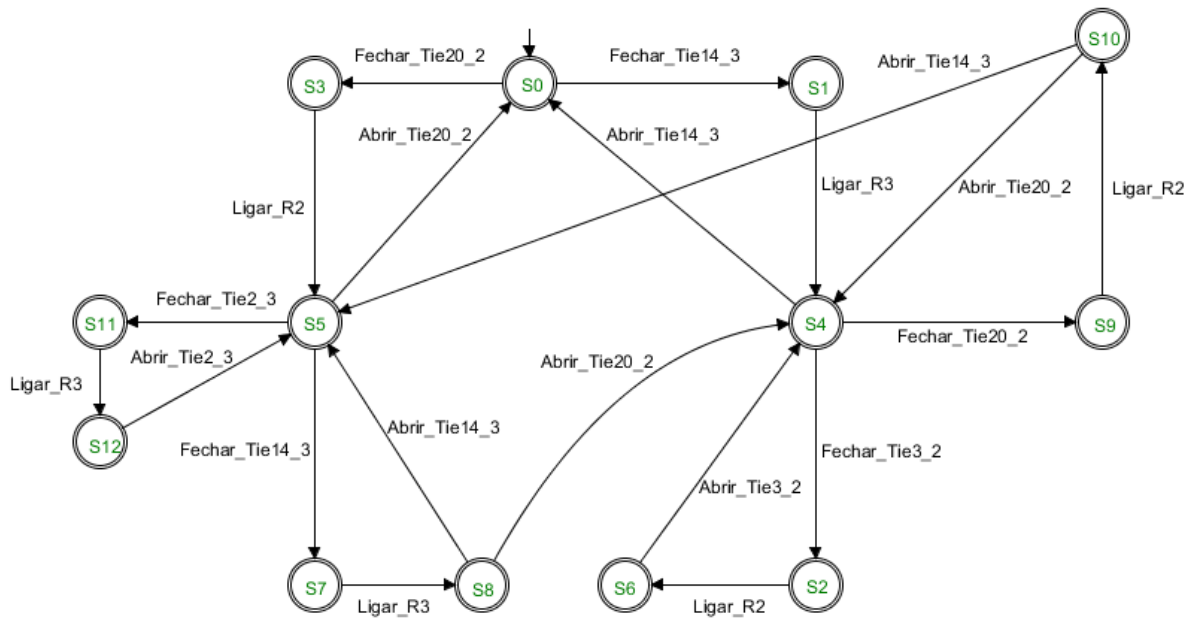


Figura 31: Especificação E6

Fonte – Autoria Própria *Software* Supremaica

O próximo passo na metodologia para obtenção do supervisor é a composição síncrona das plantas encontrando a planta geral G_P e das especificações E_G , porém no programa Supremaica tem-se a opção de síntese do supervisor, onde selecionando as especificações e plantas e marca-se a opção de síntese de supervisor controlável e não bloqueante, ele vai gerar automaticamente todos os passos necessários para síntese do controlador. Para melhor entender o ambiente do programa Supremaica as figuras 32, figura 33 e figura 34, mostram a interface do programa com a modelagem proposta.

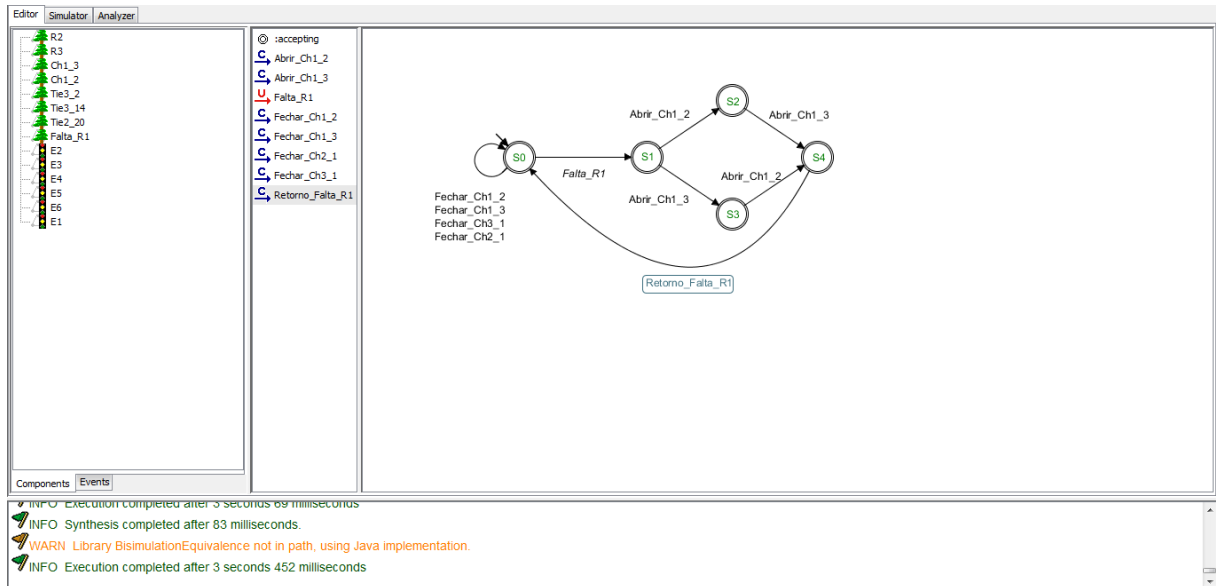


Figura 32: Interface editor Supremica
Fonte – Autoria Própria Software Supremica

O editor do Supremica é onde faz-se a modelagem das plantas e especificações e é definido os eventos, transições, estados, estados marcados, eventos controláveis e não controláveis.

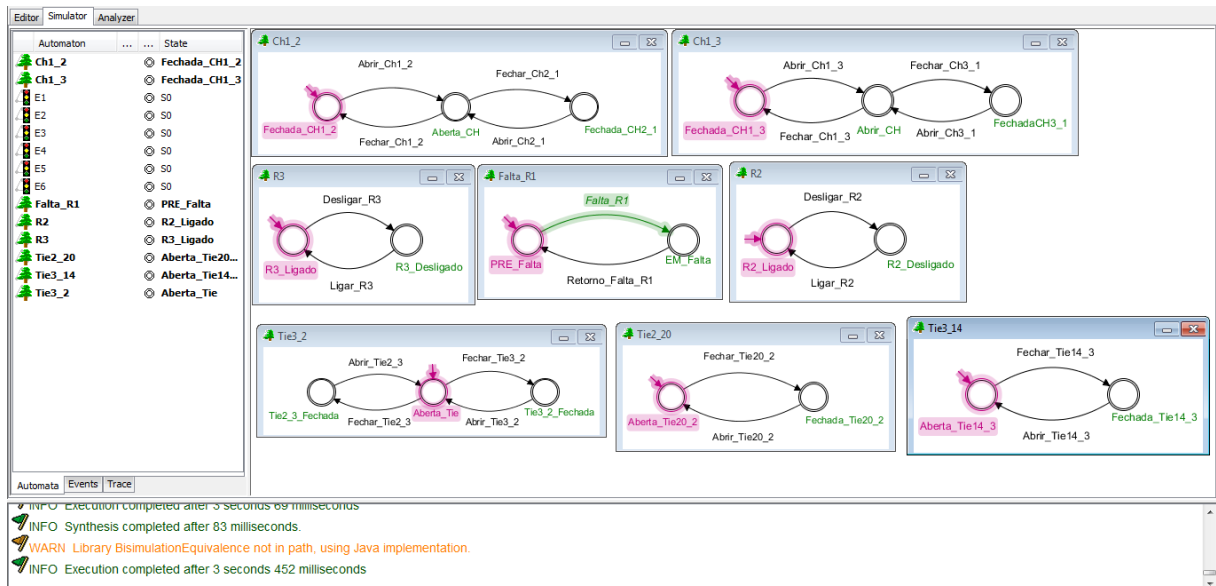


Figura 33: Interface simulador Supremica
Fonte – Autoria Própria Software Supremica

A interface da figura 33 tem por função simular a modelagem, plantas e restrições, para analisar se as especificações estão restringindo a planta de acordo com as necessidades do problema proposto, caso não esteja de acordo é necessário

revisar as restrições, modelagem ou conceito, até que as mesmas estejam de acordo com as necessidades do problema.

Name	Type	Q	E	-
R2	Plant	2	2	2
R3	Plant	2	2	2
Ch1_3	Plant	3	4	4
Ch1_2	Plant	3	4	4
Tie3_2	Plant	3	4	4
Tie3_14	Plant	2	2	2
Tie2_20	Plant	2	2	2
Falta_R1	Plant	2	2	2
E2	Specification	2	4	4
E3	Specification	2	4	4
E4	Specification	2	3	3
E5	Specification	2	3	3
E6	Specification	13	10	20
E1	Specification	5	8	10
sup(Ch1_2 Ch1_3 ...	Supervisor	1123	22	3899
R2 R3 Ch1_3 Ch1...	Plant	864	22	7776
E2 E3 E4 E5 E6 E1	Specification	309	20	1067
R2 R3 Ch1_3 Ch1...	Plant	1265	22	4197

4197 transitions were examined.
1265 reachable states were found.
0 forbidden states were found.
18 deadlocked states were found.

Figura 34: Interface analisador Supremica
Fonte: Autoria Própria *Software* Supremica

Na interface da figura 34 é possível fazer todas as operações entre os autômatos, plantas e especificações, composição síncrona, síntese do supervisor, minimização de autômatos, etc.

Para o trabalho a síntese do supervisor será realizado pelo Supremica, conforme mostra à figura 34, como o supervisor tem um número muito elevado de estados a representação do mesmo por um ADEF fica impossibilitado, no entanto é possível exportar o supervisor para um arquivo, txt, xml, html, entre outros, contendo todos os dados do supervisor, ou seja, para o exemplo proposto ele teria as ações programadas para quando ocorrer um evento falta em na região *R1*, conter os passos para isolar a região em falta das regiões ligadas a ela, bloquear o fechamento das chaves ligadas a região em falta, não permitindo que em alguma reconfiguração as chaves ligada a região afetada seja fechada e por fim as possibilidades de reconfiguração das regiões afetadas. Parte do arquivo “xml” exportado pelo Supremica está nos anexos.

Os passos para obtenção do supervisor estão no capítulo 4, consistem em realizar o produto síncrono entre as plantas locais, conforme equação 6-1, resultando no modelo geral das plantas G_p . Para obtenção das especificações é feito o produto síncrono entre as especificações conforme equação 6-2, resultando em E_G , o Autômato K é proveniente do produto síncrono entre as especificações e plantas

$E_G || G_p$, conforme a equação 6-3, o último passo é o algoritmo iterativo, visto no capítulo 4, que identifica se existem maus estados, caso exista remoção dos maus estados e cálculo da componente *trim*.

Antes de fazer a composição síncrona recomenda-se uma análise entre as especificações e plantas, para verificar se possuem plantas em que não estão relacionadas com nenhuma especificação, pois as mesmas podem ser removidas da composição síncrona, no problema proposto as plantas, *A1*, *R1*, *R20* e *R14*, não possuem relações com as especificações, logo podem ser retiradas da composição. Abaixo segue as equações de composições das plantas, especificações e de *K*:

$$G_p = CH1_2 || CH1_3 || TIE2_3 || TIE3_14 || TIE2_20 || R2 || R3 || \quad (6-1)$$

$$E_G = E1 || E2 || E3 || E4 || E5 || E6 \quad (6-2)$$

$$K = (G_p || E_G) \quad (6-3)$$

Para fins de comparativos, a tabela 3 traz os dados referentes a número de estados, número de eventos, número de transições, para comparar o autômato *K* e as composições em relação ao SupC. Depois das operações de composição e síntese do supervisor foi aplicado a minimização de autômatos fornecido pelo programa Supremica, para verificar a possibilidade de encontrar um autômato menor que o projetado, reduzindo o número de estado e transições.

Tabela 3– Dados relacionados a composição e SupC

	 Q 	 Σ 	 →
<i>G_p</i>	864	22	7776
<i>E_G</i>	309	20	1067
K	1265	22	4197
Sup	1123	22	3899
<i>Min(G_p)</i>	864	22	7776
<i>Min(E_G)</i>	271	20	947
Min(K)	1080	22	3663
Min(SupC)	881	22	3207

|Q| - Número de estados - |Σ| - Numero de Eventos - |→| - Numero de Transições

A metodologia proposta tem por objetivo listar os passos para quando ocorrer uma falta o sistema já tenha pré-definido pelo supervisor as transições possíveis para cada estado, por exemplo se uma chave está aberta e pretende-se fecha-la, no entanto para o fechamento da mesma é necessário a abertura de outra chave, o supervisor vai bloquear o fechamento da chave, possibilitando somente quando a outra chave for aberta.

A tabela 4 ilustra a sequência de transições possíveis para reconfiguração da região modelada, dividida em três etapas, região aonde ocorreu a falta, isolamento da região e reconfiguração das chaves.

Tabela 4– Sequência de manobras de chaves para reconfiguração

Etapas para reconfiguração					
Sequência	Região	Isolar Região		Energizar cargas restantes	
1	Falta_R1	Abrir_CH1_2	Abrir_CH1_3	Fechar_Tie20_2	Fechar_Tie14_3
2	Falta_R1	Abrir_CH1_2	Abrir_CH1_3	Fechar_Tie14_3	Fechar_Tie3_2
3	Falta_R1	Abrir_CH1_2	Abrir_CH1_3	Fechar_Tie20_2	Fechar_Tie2_3
4	Falta_R1	Abrir_CH1_3	Abrir_CH1_2	Fechar_Tie20_2	Fechar_Tie14_3
5	Falta_R1	Abrir_CH1_3	Abrir_CH1_2	Fechar_Tie14_3	Fechar_Tie3_2
6	Falta_R1	Abrir_CH1_3	Abrir_CH1_2	Fechar_Tie20_2	Fechar_Tie2_3
7	Falta_R1	Abrir_CH1_2	Abrir_CH1_3	Fechar_Tie14_3	Fechar_Tie20_2
8	Falta_R1	Abrir_CH1_3	Abrir_CH1_2	Fechar_Tie14_3	Fechar_Tie20_2

Fonte: Autoria própria

A tabela 4, mostra 8 sequências de manobras de chaves que resultariam em uma reconfiguração, porém se retirar as sequências que resultam na mesma reconfiguração, teriam somente 3 sequências possíveis de reconfiguração. As figuras abaixo ilustram os possíveis cenários de reconfigurações contidas no supervisor, sem levar em conta a ordem em que os eventos acontecem.

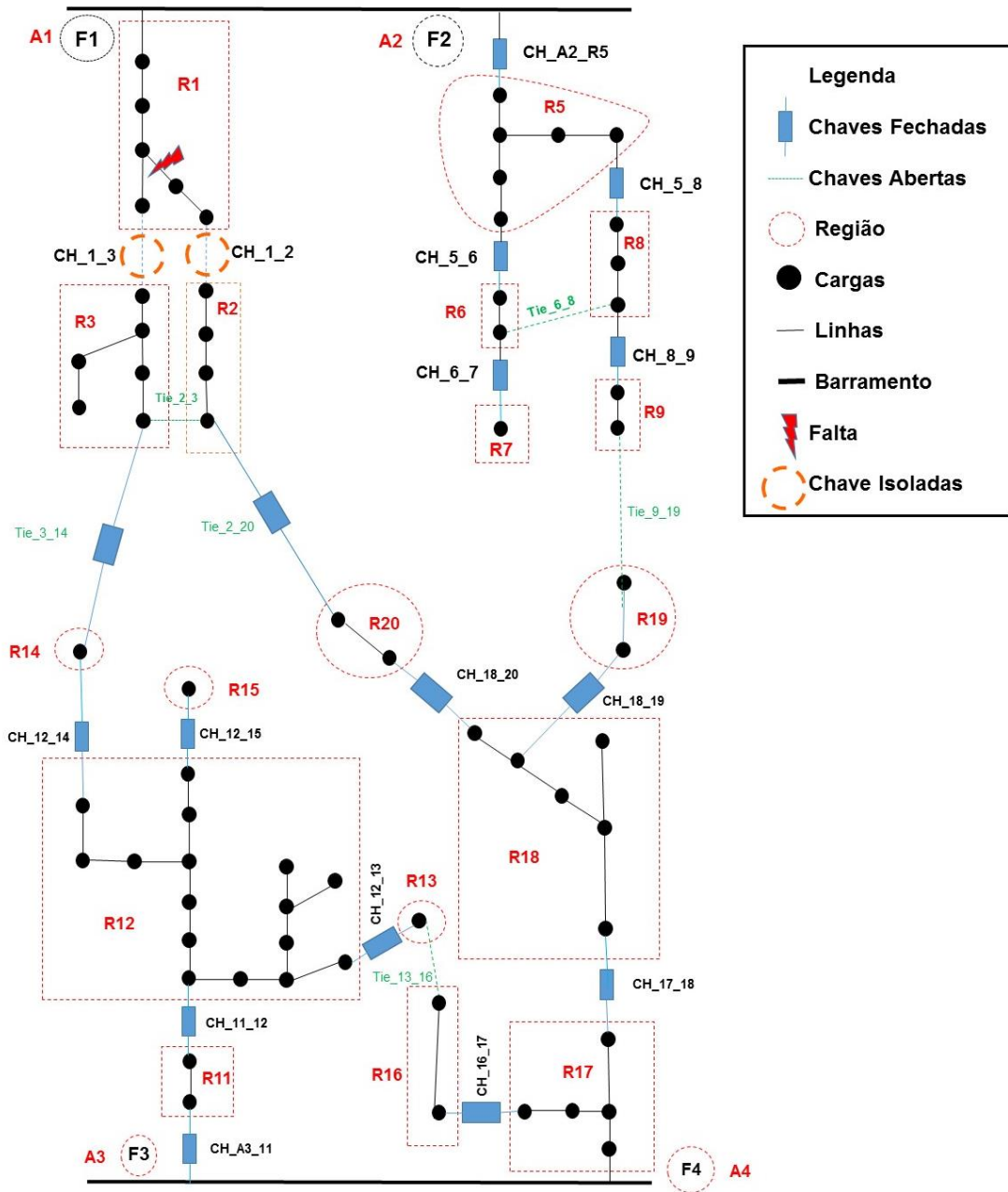


Figura 35 - Possibilidade de Reconfiguração 1
 Fonte: Autoria Própria

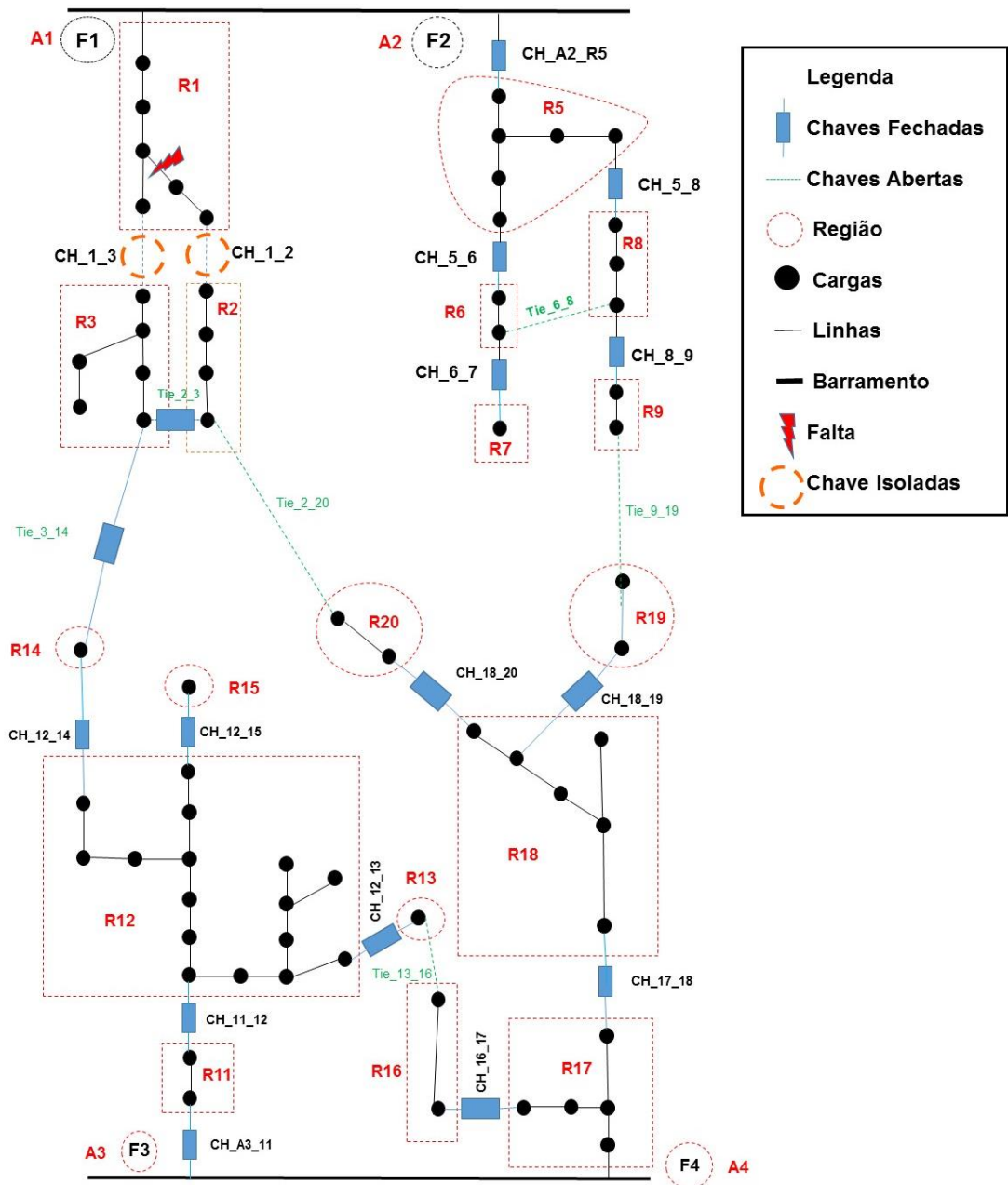


Figura 36: Possibilidade de reconfiguração 2
Fonte – Autoria própria.

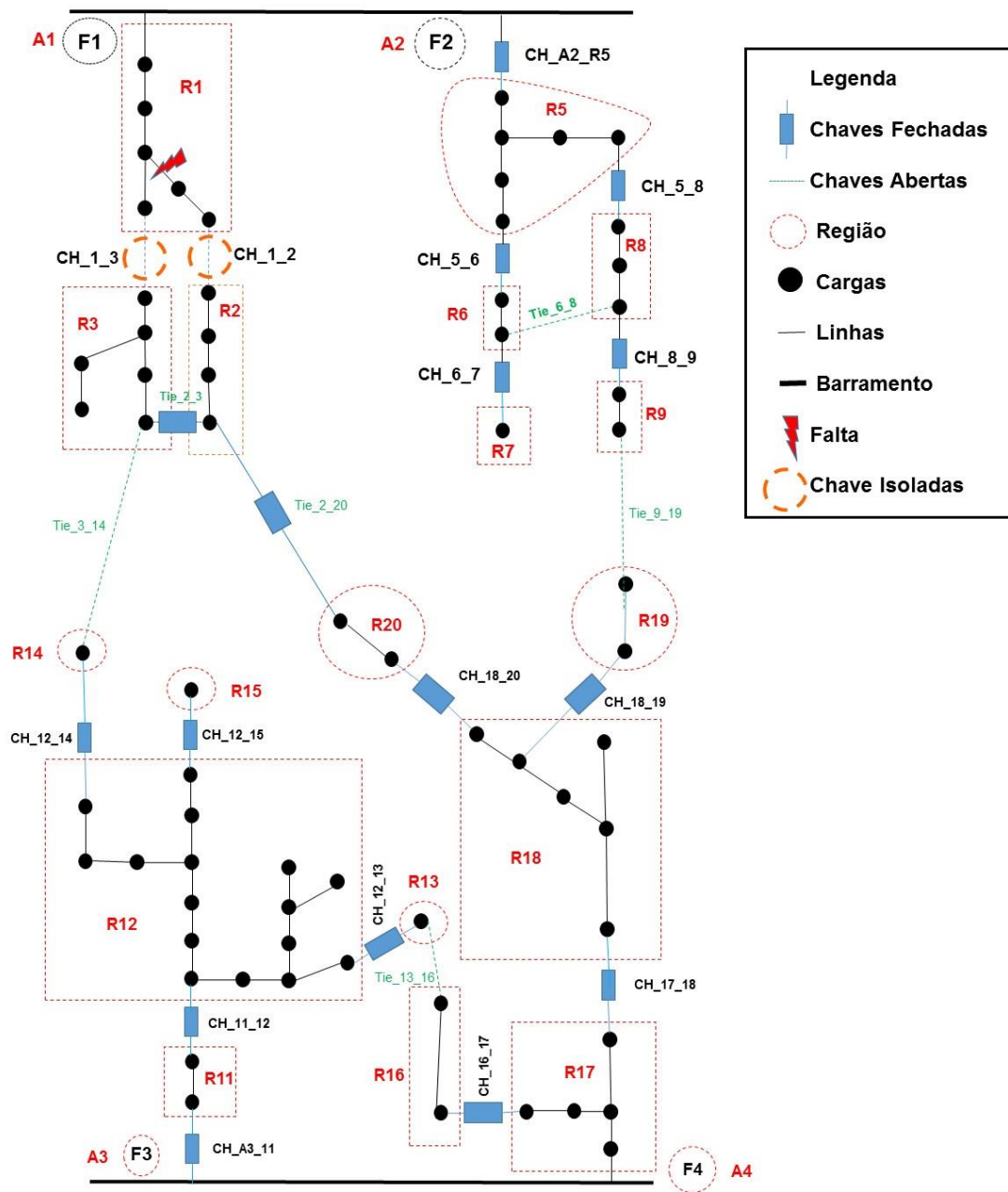


Figura 37: Possibilidade de reconfiguração 3
Fonte – Autoria própria.

A tabela 2 mostra a quantidade de chaves abertas e fechadas na rede, constata-se que a rede possuía 5 chaves para manobras, após a metodologia aplicada sobre a vizinhança da região em que a falta ocorreu consegue-se reduzir para 3 chaves, com a aplicação da modelagem e restrições consegue-se reduzir para 3 cenários possíveis de reconfiguração, isto viabiliza a utilização de algoritmos exatos para o cálculo de fluxo de potência de todos os três cenários e verificar qual cenário possui a maior vantagem de reconfiguração.

7. CONCLUSÕES

Conclui-se que a metodologia é eficaz para modelar problemas de *Self-Healing* em *Smart Grid* possibilita-se através da abordagem delimitar o número de chaves que irão ser utilizadas no processo de reconfiguração. Consegue-se determinar a região em que a falta ocorreu, isolar a mesma das outras regiões e mostrar as possíveis sequências de combinações para a energização das regiões afetadas, logo o sistema proposto cumpre com os objetivos estipulados no seu projeto.

Alguns pontos positivos do sistema é sua modularidade, onde as plantas após definidas podem ser replicadas para todos os elementos similares, o formalismo de controle supervisorio também permite que seja feita alterações no sistemas sem muitas complicações, outro ponto positivo é que pode-se delimitar as regiões afetadas reduzindo o número de elementos a serem analisados, com isso a utilização de algoritmos exatos é possível em determinadas situações e pode-se acrescentar ou retirar plantas e especificações sem ser necessário refazer toda a modelagem, apenas sintetizando novamente o supervisor, isto facilita adicionar mais especificações para modelar novos problemas.

Alguns problemas ficaram em aberto, como: na ocorrência de faltas em múltiplas regiões o supervisor será o produto síncrono entre os supervisores, também verificar na ocorrência de simetria entre regiões as especificações serem as mesmas, logo as especificações seriam modulares, é intuitivo pensar que são possíveis, no entanto necessita-se de mais estudos e testes.

Alguns pontos negativos são que se a topologia geográfica da rede alterar-se é necessário fazer outra modelagem e não foi possível incorporar elementos do fluxo de potência na metodologia, logo é necessário após a determinação das sequências de reconfiguração verificar com algum algoritmo de fluxo se aquela reconfiguração é possível, pois pode acontecer de após a reconfiguração a rede tenha outro problema devido a estabilidade, variação de tensão máximo e mínimo, sobrecarga, limite máximo de transmissão das linhas, questões relacionadas ao fluxo de potência.

Para um estudo futuro recomenda-se a utilização dessa metodologia aplicada em algoritmos Heurísticos e exatos, com redes pequenas e grandes, para observar a melhora que se tem no tempo computacional de um método exato para

uma Heurística. Fazer um levantamento de problemas técnicos e físicos em reconfigurações, como em proteções e geração distribuída e incorporar esses problemas como especificações para observar o impacto no número de estados e transições do supervisor.

8. REFERÊNCIAS

ABRADEE. **Visão Geral do Setor**. Disponível em: <<https://goo.gl/gGK5U8>>. Acesso em: 21 Março 2017.

ALJAHANI, Mohamed E. **An Enhanced Self-Healing Protection System in Smart Grid: Using Advanced and Intelligent Devices and Applying Hierarchical Routing in Sensor Network Technique**. 2014. 1-21p. Western Michigan University. Ann Arbor - Michigan, 2014.

AMARAL, Marcelo B. D. **Modelagem Estática de Carga por Meio de Medição Direta: Proposição de uma Metodologia Sistematizada, seus Impactos e Benefícios**. 2004. 16p.[s.n.]. Juiz de Fora, MG, 2004.

ANEEL. Novos tempos na distribuição de energia elétrica no Brasil. **Revista Pesquisa e Desenvolvimento da ANEEL - P&D**, Brasília (DF), v. 3, p. 129, junho 2009. ISSN 1981-9803.

ANEEL. **PROGRAMA BRASILEIRO DE REDE ELÉTRICA**. Disponível em: <<https://goo.gl/Nh18po>>. Acesso em: 17 Abril 2017.

BECCENERI, José C. **Laboratório Associado de Computação e Matemática Aplicada**. Disponível em: <http://www.lac.inpe.br/elac13/arquivos/MiniCurso_02ELAC2012.pdf>. Acesso em: 22 Março 2016.

BOCCUZZI, Cyro. **Smart grid é o futuro certo da distribuição**. 2010. p. 160p. n. 50 - O Setor Elétrico. São Paulo, 2010. Disponível em: <<http://goo.gl/4CtuhT>>. Acesso em: 5 Abril 2016.

BOUZON, G.; QUEIROZ, M. H. D.; CURY, J. E. R. Exploiting distinguishing sensors in supervisory control of des. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL AND AUTOMATION (ICCA'09), 2009, Christchurch, New Zealand. Christchurch, New Zealand, 2009.

BRANDIN, B. A.; WONHAM, W. M. Modular supervisory control of timed discrete-event systems. In: IEEE. ANNUAL CONFERENCE ON DECISION AND CONTROL (CDC'93)., 1993, San Antonio, USA. San Antonio, USA, 1993.

BRASIL, 3M. 3M SMART GRID. 3M. Disponível em: <http://solutions.3m.com/wps/portal/3M/pt_PT/PT-Smart-Grid/Smart-Grid/>. Acesso em: 10 MARÇO 2016.

CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. 2008. 2. ed - Springer Science. New York, 2008.

CASSANDRAS, Christos G.; LAFORTUNE, Stéphane. **Introduction to Discrete Event Systems**. 2. ed. Nova York: springer, v. 1, 2008. 769 p.

COPEL. **Tarifas de Energia da COPEL**. 2016.[s.n.], 2016. Disponível em: <<http://goo.gl/EpEQw5>>. Acesso em: 11 Março 2016.

CUNHA, A. E. C. D.; CURY, J. E. R. Hierarchical supervisory control based on discrete event systems with flexible marking. In: IEEE TRANSACTIONS AUTOMATIC CONTROL, v. 52., 2007, [S.I.], 2007.

CURY, J. E. R. **Teoria de Controle Supervisório de Sistemas a Eventos Discretos**. 2001.[s.n.]. Canela, RS, 2001.

DEBANJAN GHOSH. **Self-healing systems — survey and synthesis**. 2007. 2164 - 2185p.[s.n.], 2007. Disponível em: <<https://goo.gl/EiaWH5>>. Acesso em: 03 Junho 2017.

FILHO, João M. **Instalações Elétricas Industriais**. 6. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 2002.

GRUPO DE TRABALHO DE REDES ELÉTRICAS INTELIGENTES. **SMART GRID**. Ministério de Minas e Energia. [S.I.]: [s.n.], 2010. 15 p.

GTA/UFRJ. **GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO**. Disponível em: <<http://goo.gl/fO6n5Z>>. Acesso em: 17 Março 2016.

KAGAN, N.; OLIVEIRA, C. C. B. D.; ROBBA, E. J. **INTRODUÇÃO AOS SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA**. 2005. 6-20p. 1. ED. - EDIGARD. SÃO PAULO, 2005. Acesso em: 10 abr. 2017.

LAMBIASE, Clodoaldo D. B. APLICAÇÃO DE SELF HEALING EM SISTEMAS ELÉTRICOS. **LUME - REPOSITÓRIO DIGITAL UFRGS**, Porto Alegre, p. 59, DEZEMBRO 2012.

LIN, F.; WONHAM, W. M. Decentralized control and coordination of discrete-event systems with partial observation. In: IEEE TRANSACTIONS AUTOMATIC CONTROL, v. 35., 1990, [S.I.], 1990.

MAZZETTO, MURIEL. **INTELIGÊNCIA COMPUTACIONAL PARA SISTEMAS SELF-HEALING: UM ESTUDO DE CASO EM REDES ELÉTRICAS INTELIGENTES**. 2016. 14p.[s.n.]. Pato Branco, 2016.

OUEDRAOGO, L. Nonblocking and safe control of discrete-event systems modeled as extended finite automata. In: IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, v. 8., 2011, [S.I.], 2011.

PEDRETTI, André; JUNIOR, João A. B. **COPEL**. Disponível em: <<http://goo.gl/PWFRMj>>. Acesso em: 16 Março 2016.

PSAIER, H; DUSTDAR, S. **A survey on self-healing systems: Approaches and systems**. 2011. p. 43–73p. v. 91 - [s.n.]. Vienna/New York, 2011.

QUEIROZ, M. H. D.; CURY, J. E. R. Modular supervisory control of large scale discrete event systems. In: INTERNATIONAL WORKSHOP ON DISCRETE EVENT SYSTEMS (WODES'00), 2000, Ghent, Belgium. Ghent, Belgium, 2000.

RAMADGE, P. J.; WONHAM, W. M. Modular feedback logic for discrete event systems. **SIAM Journal of Control and Optimization**, v. v. 25, n. n. 5, p. p. 1202–1218, 1987.

RAMADGE, P. J.; WONHAM, W. M. The control of discrete event systems. **IEEE Special Issue on Discrete Event Dynamic**, v. v. 77, n. n. 1, p. p. 81–98, 1989.

RAMADGE, Peter ; WONHAM, Walter. Supervisory control of a class of discrete event processes. **SIAM J. Control and Optimization**, v. 25, n. 1, p. 206-230, 1987.

RAMOS, C. C. O. **CARACTERIZAÇÃO DE PERDAS COMERCIAIS EM SISTEMAS DE ENERGIA**. 2014. 144 Pp.EPUSP. SÃO PAULO - SP, 2014.

RIBEIRO, Richardson et al. **An Advanced Software Tool to Simulate Service Restoration Problems: a case study on Power Distribution Systems**. In: **INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE**, 2017, Zurich, Switzerland. Zurich, Switzerland, 2017. Disponível em: <<https://goo.gl/c6gr6m>>. Acesso em: 10 Agosto 2017.

SILVA, David M. K. **Fluxo de potência para sistemas de distribuição considerando redução de barras**. 2016. 20-23p.[s.n.]. Londrina, 2016.

TEIXEIRA, Marcelo. **Modelagem e Controle de Sistemas a Eventos Discretos - Notas De Aula - 3**. 2015. 1-4p.[s.n.]. Pato Branco, 2015.

U.S. DEPARTMENT OF ENERGY. COMMUNICATIONS REQUIREMENTS. **U.S. DEPARTMENT OF ENERGY**, 2010. Disponível em: <<https://goo.gl/C5KKGx>>. Acesso em: 15 Abril 2017.

VISCONTI, Igor F. **Modelos de Cargas Baseados em Medições para Simulações Dinâmicas em Sistemas Elétricos de Potência**. 2010. 2-16p.[s.n.]. RIO DE JANEIRO, 2010.

ANEXOS

A figura abaixo mostra um dos formato em que o supervisor é exportado, neste caso foi exportado para xml, os dados podem ser utilizados em microcontroladores ou computadores.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <Automata minor="9" major="0" name="Untitled">
- <Automaton name="sup(Ch1_2||Ch1_3||E1||E2||E3||E4||E5||E6||Falta_R1||R2||R3||Tie2_20||Tie3_14||Tie3_2)" type="Supervisor">
+ <Events>
- <States>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S10.PRE_Falta.R2_Ligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="0"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S11.PRE_Falta.R2_Ligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="1"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S12.PRE_Falta.R2_Ligado.R3_Ligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="2" accepting="true"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S2.PRE_Falta.R2_Desligado.R3_Ligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="3"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S6.PRE_Falta.R2_Ligado.R3_Ligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="4"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S7.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="5"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S8.PRE_Falta.R2_Ligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="6"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S9.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="7"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S3.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="8"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S5.PRE_Falta.R2_Ligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="9"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S10.PRE_Falta.R2_Desligado.R3_Desligado.Aberta_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="10"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S4.PRE_Falta.R2_Desligado.R3_Ligado.Aberta_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="11"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S10.PRE_Falta.R2_Desligado.R3_Desligado.Aberta_Tie20_2.Aberta_Tie" id="12"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S10.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="13"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S12.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="14"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S6.PRE_Falta.R2_Desligado.R3_Desligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="15"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S0.S0.S1.S8.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="16"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S0.S3.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="17"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S0.S5.PRE_Falta.R2_Ligado.R3_Ligado.Fechada_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="18" accepting="true"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S0.PRE_Falta.R2_Desligado.R3_Ligado.Aberta_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="19"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S10.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="20"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S11.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="21"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S12.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="22"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S6.PRE_Falta.R2_Desligado.R3_Ligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="23"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S7.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="24"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S8.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="25"/>
<State name="Aberta_CH.Abrir_CH.S0.S0.S3.S1.S5.PRE_Falta.R2_Desligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="26"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S1.PRE_Falta.R2_Ligado.R3_Desligado.Aberta_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="27"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S4.PRE_Falta.R2_Ligado.R3_Ligado.Aberta_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="28"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S10.PRE_Falta.R2_Ligado.R3_Desligado.Aberta_Tie20_2.Aberta_Tie14_3.Aberta_Tie" id="29"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S11.PRE_Falta.R2_Desligado.R3_Ligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="30"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S12.PRE_Falta.R2_Ligado.R3_Desligado.Fechada_Tie20_2.Aberta_Tie14_3.Tie2_3_Fechada" id="31"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S2.PRE_Falta.R2_Desligado.R3_Desligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="32"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S6.PRE_Falta.R2_Ligado.R3_Desligado.Aberta_Tie20_2.Fechada_Tie14_3.Tie3_2_Fechada" id="33"/>
<State name="Aberta_CH.Abrir_CH.S0.S1.S0.S0.S8.PRE_Falta.R2_Ligado.R3_Desligado.Fechada_Tie20_2.Fechada_Tie14_3.Aberta_Tie" id="34"/>
```