UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
CAMPUS DE FOZ DO IGUAÇU
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

VERIFICAÇÃO FORMAL APLICADA À ANÁLISE DE CONFIABILIDADE DE SISTEMAS HIDRÁULICOS

CLAUDIA BEATRIZ BOZZ

FOZ DO IGUAÇU
2018

Claudia Beatriz Bozz

Verificação Formal Aplicada à Análise de Confiabilidade de Sistemas Hidráulicos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica e Computação.

Área de concentração: Sistemas Dinâmicos e Energéticos.

Orientador: Prof. Dr. Guilherme de Oliveira Kunz

Foz do Iguaçu

2018

Ficha de identificação da obra elaborada através do Formulário de Geração Automática do Sistema de Bibliotecas da Unioeste.

Bozz, Claudia Beatriz
Verificação Formal Aplicada à Análise de Confiabilidade de Sistemas Hidráulicos / Claudia Beatriz Bozz; orientador(a), Guilherme de Oliveira Kunz, 2018.
113 f.

Dissertação (mestrado), Universidade Estadual do Oeste do Paraná, Centro de Engenharias e Ciências Exatas, Programa de Pós-Graduação em Engenharia Elétrica e Computação, 2018.

1. Autômatos híbridos. 2. Model Checking. 3. Verificação formal estatística. 4. Disponibilidade e Manutenção. I. Oliveira Kunz, Guilherme de. II. Título.

Verificação Formal Aplicada à Análise de Confiabilidade de Sistemas Hidráulicos

Claudia Beatriz Bozz

Essa Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação e aprovada pela Banca Examinadora:

Data de defesa pública: 26/07/2018

Prof. Dr. **Guilherme de Oliveira Kunz** - (Orientador)
Universidade Estadual do Oeste do Paraná – UNIOESTE

Prof. Dr. **Jiam Pires Frigo**
Universidade Federal da Integração Latino-Americana – UNILA

Prof. Dr. **Sandro Battistella**
Universidade Estadual do Oeste do Paraná – UNIOESTE

Prof. Dr. **Romeu Reginatto**
Universidade Estadual do Oeste do Paraná – UNIOESTE

Resumo

Sistemas de tempo real que possuem comportamento contínuo associado com elementos de características discretas são chamados de sistemas híbridos. Dentre estes, nesta pesquisa de mestrado, optou-se pelo emprego de um sistema hidráulico como objeto de estudo a fim de realizar a análise de confiabilidade do mesmo a partir de modelagem e verificação formal. Por mais que diversos modelos para a análise de confiabilidade de sistemas complexos tenham sido propostos na literatura, a maioria não são adequados para representar sistemas em que o comportamento é expresso em variáveis contínuas, como é o caso dos sistemas híbridos. De modo geral, para a análise de sistemas, a simulação e os testes experimentais são comumente utilizados, e geram apenas resultados aproximados a partir de uma grande quantidade de amostras. Para eliminar as limitações destas técnicas, a verificação formal é uma alternativa eficaz, visto que é caracterizada por realizar uma varredura em todos os estados possíveis do sistema de forma automática, verificando o comportamento como um todo do mesmo. Neste trabalho, foi utilizada a ferramenta computacional UPPAAL STRATEGO para a modelagem por autômatos estocásticos híbridos e verificação dos modelos, tanto verificação formal clássica como estatística. Um modelo padrão (*benchmark*) foi utilizado como objeto de estudo. Inicialmente o sistema foi modelado e seu comportamento (físico e controlado) verificado através da simulação e verificação formal (especificação de propriedades e verificação de modelos). Os parâmetros de confiabilidade obtidos na análise estatística de falha do sistema foram comparados com outros existentes na literatura, apresentado uma dispersão inferior a 2,5%, logo pôde se verificar que a metodologia empregada e os modelos construídos foram adequados para análise de confiabilidade deste sistema hibrido. Em uma segunda etapa do trabalho, foi modificada a distribuição de probabilidade de falha dos componentes, a fim de tornar o sistema mais fidedigno com sistemas hidráulicos reais, e estimar o tempo médio entre manutenções (*MTBM – Mean Time Between Maintenance*) ideal deste sistema. Portanto, conclui-se que a metodologia empregada foi adequada para realizar a análise de confiabilidade do sistema hidráulico, sendo efetivo levantar os parâmetros de confiabilidade através da verificação de modelos.

Palavras-chaves: Autômatos híbridos, *model checking*, Verificação formal estatística, Disponibilidade, Manutenção.

Abstract

Real time systems that have continuous behavior associated with discrete elements are called hybrid systems. Among them, in this master's research, a hydraulic system has been chosen as an object of study in order to perform the reliability analysis of it through modeling and formal verification. Much as several models for the reliability analysis of complex systems have been proposed in the literature, most of them are not suitable to represent the system when its behavior needs to be expressed by means of continuous variables, like the case of hybrid systems. Generally, simulation and experimental testing are used to analyze systems, and they give only approximate results from a large amount of samples. To eliminate the limitations of these techniques, the formal verification is an effective alternative, since it is characterized by performing a sweep in all possible states of the system automatically, verifying the behavior as a whole. The UPPAAL STRATEGO toolkit for modelling by stochastic hybrid automata and model checking has been used in this work, both classic formal verification and statistical formal verification. A benckmark has been used as object of study. Initially, the system has been modelling and its behavior (physical and controlled) verified through simulation and formal verification (property specification and model checking). The reliability parameters obtained in the statistical analysis of the system failures have been compared with results of literature, presenting a dispersion less than 2.5%, so it can be verify that the methodology used and the models constructed were adequate to analyze the reliability of this system hybrid.In a second step of this work, the probability distribution of failure of the components have been modified, in order to become the system more reliable with real hydraulic systems, and estimate the optimum mean time between maintenance (MTBM) of this system. Thus, it's possible to conclude that the methodology is adequate to perform the reliability analysis of the hydraulic system, being that model checking is effective to estimate the reliability parameters of the hydraulic system.

Keywords: hybrid automata, model checking, statistical formal verification, availability, maintenance.

Dedico esse trabalho ao meu esposo, Horácio Borges.

Agradecimentos

Agradeço acima de tudo à Deus, que em sua infinita bondade, me permitiu concluir mais está etapa de minha vida acadêmica, me proporcionando sabedoria, paciência e saúde.

A minha família e ao meu esposo, que me apoiaram em todos os momentos, me incentivando a nunca desistir e entendendo a minha ausência e falta de tempo.

Ao meu orientador Prof. Dr. Guilherme Kunz de Oliveira por compartilhar comigo seus conhecimentos, dando todo suporte necessário para o desenvolvimento deste trabalho.

Aos meus colegas de mestrado que compartilharam comigo todas as frustações, vitórias e desesperos desse mestrado. Em especial, ao Felipe Crestani, que me auxilio, com muita paciência, na utilização do nosso tão amigável software UPPAAL.

Enfim, agradeço a todos que de uma forma ou outra, me incentivaram e auxiliaram para a conclusão desta pesquisa.

“Portanto, não se preocupem com o dia de amanhã,
pois o dia de amanhã terá suas preocupações.
Basta para cada dia sua própria dificuldade.”
(Mateus 6,34)

Sumário

Lista de Figuras

Lista de Quadros

Lista de Tabelas

Lista de Símbolos

3ASI	*Associazione degli Analisti dell'Ambiente, dell'Affidabilita' e dela Sicurezza Insdustriale*
cdf	Distribuição acumulada de falha
CTL	*Computation tree logic*
ESA	*European Space Agency*
D	Disponibilidade
f(t)	Distribuição densidade de falha
F(t)	Distribuição acumulada de falha
FSPN	*Fluid Stochastic Petri Nets (*Redes de Petri Fluidas Estocásticas)
GSPN	*Generalized Stochastic Petri Nets (*Redes de Petri Estocásticas Generalizadas)
h(t)	Função de risco
IEC	*International Electrotechnical Commission*
LTL	*Linear temporal logic*
MTBF	*Mean Time Between Failure (*Tempo médio entre falha)
MTBM	*Mean Time Between Maintenance* (Tempo médio entre manutenções)
MTTF	*Mean Time To Failure (*Tempo médio para falha)
MTTR	*Men Time To Repair (*Tempo médio em reparo)
pdf	Distribuição densidade de falha
R(t)	Função Confiabilidade
t	Tempo
TCTL	*Timed computational tree logic*
λ	Taxa de Falha
β	Parâmetro de forma da distribuição de Weibull
η	Parâmetro de escala da distribuição de Weibull
γ	Parâmetro de localização da distribuição de Weibull
σ	Desvio-padrão da distribuição Lognormal
μ	Média da distribuição Lognormal

Capítulo 1

Introdução

1.1 Contextualização

A evolução tecnológica decorrente da revolução industrial incorporou à vida cotidiana das pessoas e empresas, os sistemas de tempo real, que trouxeram maior nível de automatização, sendo bastante conveniente e benéfico à sociedade, porém tornou a mesma mais vulnerável e dependente destas tecnologias (Krilavicius, 2006).

Os sistemas de tempo real referem-se a sistemas compostos por sensores e atuadores, que captam informações e atuam sobre o ambiente que está sendo controlado, devendo obedecer rigorosamente às restrições de tempo impostas. Estes sistemas podem ser classificados como sistemas de tempo real brando (*soft real-time systems)* - em que a falha no sistema é aceitável, porém não desejável – e sistemas críticos (*hard real-time)* – em que a falha ocasiona perdas severas, isto é, perdas humanas ou de recursos naturais (Macêdo *et al*, 2004).

Krilavius (2006) demonstra a necessidade do funcionamento correto destes sistemas, citando que sistemas de tempo real brando, como refrigeradores e micro-ondas, na ocorrência de uma falha a sua consequência seria apenas um incômodo ao usuário. Já os sistemas críticos, como sistemas de controle de aviões e plantas nucleares, a ocorrência de uma falha pode ser catastrófica. Estes sistemas devem possuir alta previsibilidade tanto funcional quanto temporal (Macêdo *et al*, 2004).

Tendo em vista sua alta criticidade, estudos relacionados a confiabilidade destes sistemas são de suma importância, sendo que o desenvolvimento e aprimoramento de ferramentas e de metodologias para a análise da confiabilidade, visando avaliar com maior precisão a existência de falhas em equipamentos, processos e sistemas, principalmente quando podem acarretar em prejuízos econômicos, sociais e físicos, independente do seu grau de severidade, são fundamentais.

Segundo Lafraia (2001), a análise de confiabilidade pode ser entendida como uma avaliação da probabilidade de que o sistema mantenha um determinado desempenho, em um determinado tempo e sob condições conhecidas de uso. Melhorias na confiabilidade visam reduzir o índice de falha, desta forma, melhorando a qualidade dos produtos e serviços.

Alguns sistemas de tempo real possuem um comportamento contínuo associado com elementos de características discretas, como por exemplo válvulas liga/desliga, os quais são conhecidos como sistemas híbridos (Caetano, 2011). Por mais que diversos modelos para a análise de confiabilidade de sistemas complexos tenham sido propostos na literatura, a maioria não são adequados para representar sistemas em que o comportamento é expresso em variáveis contínuas, como temperatura e pressão, ou quando a configuração do sistema muda ao longo de sua vida, ou seja, os sistemas híbridos (Codetta-Raiteri e Bobbio, 2006).

Krilavius (2006) propõe que para sistemas híbridos, um modelo abstrato utilizando conceitos matemáticos pode ser criado para descrever seu comportamento. A partir desse modelo é possível analisar as propriedades do sistema e propor melhorias para aumentar a confiabilidade e disponibilidade do mesmo.

Uma estratégia para análise destes sistemas são os métodos formais. Losso *et al* (2010) destaca que a modelagem e verificação formal são importantes ferramentas na tentativa de se garantir a previsibilidade do comportamento do sistema. De modo que, a partir de métodos matemáticos, há uma exaustiva varredura das possibilidades de comportamento do sistema, obtendo-se então uma avaliação da possibilidade de ocorrência de um determinado evento ou sequência de eventos.

Segundo Krilavicius (2006), a análise de sistemas híbridos é caracterizada por duas partes: simulação e verificação formal. Inicialmente o modelo do sistema, contendo um possível comportamento e as exigências deste, é construído e a simulação é realizada a fim de avaliar se o modelo está se comportando conforme o esperado. Em seguida, um conjunto de regras (verificação formal) é aplicado para determinar se o modelo satisfaz os requisitos a ele impostos.

A modelagem pode ser realizada por meio de autômatos temporizados. Segundo Hopcroft, Ullman e Motwan (2002), autômatos são máquinas abstratas, que contém um conjunto finito de estados, e que em cada instante de tempo, está em um determinado estado. O autômato possui condições de transição que devem ser satisfeitas para que o autômato passe de um estado para o outro.

A utilização de verificação formal para análise do projeto de sistemas e predição de confiabilidade já foi empregada por diversos autores, como Peng *et al* (2013), que utilizaram a verificação formal para especificação da confiabilidade, disponibilidade e propriedades de manutenção de um satélite, através do verificador PRISM; Salas (2014) que propôs a aplicação de verificação formal para a análise da concepção de circuitos hidráulicos, utilizando para a modelagem do sistema rede Canal-Agência e Rede de Petri; Yan, Zhang, H. e Zhang Y. (2015), aplicaram *model checking* para a predição da confiabilidade do sistema hidráulico de hélice de passo controlável e, Guo e Yang (2016), que abordam os problemas de análise de desempenho e confiabilidade de um *cluster* de sensores empregando a técnica probabilística de verificação de modelos.

1.2 Motivação

Os sistemas híbridos podem ser encontrados nas mais variadas aplicações, desde uso doméstico até no âmbito industrial (Lazar, 2006). Nesta pesquisa, optou-se pelo emprego de um sistema hidráulico como objeto de estudo, este sistema possui processos contínuos, devido seus componentes mecânicos, e é controlado por elementos digitais que permitem o funcionamento do sistema em diferentes modos de operação, ou seja, o mesmo pode ser caracterizado como um sistema híbrido.

Além disso, Linsingen (2013) menciona que em decorrência de demandas tecnológicas diversificadas e de uma série de características dos sistemas hidráulicos, como a baixa relação peso/potência, facilidade de adaptação a partir de alteração dos parâmetros operacionais, possibilidade de combinação com sistemas mecânicos, elétricos e pneumáticos, estes são utilizados em praticamente todos os ramos de atividade, como maquinários industriais, siderurgia e construção civil, máquinas agrícolas, indústria naval e aeroespacial, veículos de transporte e passeio, entre outros.

Ou seja, os sistemas hidráulicos possuem uma ampla gama de utilização, desde sistemas mais simples até aqueles mais complexos, evidenciando sua importância e justificando o estudo e aprimoramento de técnicas de projeto e análise destes sistemas (Salas, Belan e De Negri, 2013).

Para a concepção de sistemas hidráulicos há possibilidade de se utilizar comandos binários, combinatórios ou sequenciais. Segundo Bollmann (1997), tradicionalmente o projeto destes comandos é realizado de modo intuitivo, em que a escolha dos elementos e de suas interligações são baseados na experiência dos projetistas, em exemplos e correções por tentativa e erro, resultando em diversas soluções intuitivas, perdendo qualidade quando se trata de sistemas de automação mais complexos. Complementando, Nakashima e Baba (1989), afirmam que a abordagem tradicional para o desenvolvimento de sistemas hidráulicos é extremamente tediosa e ineficiente. Ambos os autores concordam que, com esta abordagem, tornam-se mais difícil a supervisão, manutenção e localização de falhas, o que origina uma grande perda de tempo e de produtividade.

Métodos sistemáticos de projetos de comandos binários foram criados visando superar as deficiências do método intuitivo (Bollmann, 1997). Esta metodologia engloba a elaboração de diagrama lógico para comandos combinatórios, utilizando-se dos conceitos de álgebra booleana e tabela verdade, e para comandos sequenciais o diagrama trajeto-passo e diagrama funcional. Embora mais padronizados estes métodos ainda dependem da experiência e criatividade do projetista.

Stein (1998) afirma que a dependência da experiência de um especialista nas fases de criação e de modificação do circuito hidráulico não é um problema se houver uma redução da complexidade na etapa de análise, propondo que haja uma maior dedicação no desenvolvimento de métodos de análise destes sistemas.

De modo geral, para a análise de sistemas a simulação e os testes experimentais são comumente utilizados, e geram apenas resultados aproximados a partir de uma grande quantidade de amostras (Guo e Yang, 2016). Esses métodos não possibilitam a verificação de todos os comportamentos possíveis do sistema, podendo ocultar algum estado de falha do mesmo, pois avaliam apenas o comportamento do sistema para algumas configurações pré-estabelecidas.

Para eliminar as limitações destas técnicas de análise, a verificação formal é uma alternativa eficaz, visto que é caracterizada por realizar a verificação automatizada do sistema (Yan, Zhang, H. e Zhang Y., 2015). Logo, a verificação formal possibilita realizar uma varredura em todos os estados possíveis do sistema, verificando o comportamento como um todo.

Com base ao que foi explanado ao longo desta seção, na presente pesquisa será realizada a análise de confiabilidade de um sistema hidráulico a partir da modelagem por autômatos estocásticos híbridos e verificação formal, por meio de verificação de modelos (*model checking*). Um modelo padrão (*benchmark*) proposto por Aldemir (1987) será utilizado como objeto de estudo, este modelo é um sistema hidráulico teórico, composto por duas bombas, uma válvula, um reservatório e um controlador, construído com o intuito de estudar técnicas de análise de confiabilidade dinâmica.

Inicialmente o sistema será modelado, onde serão desenvolvidos quatro modelos, a saber: o modelo físico – que representa o comportamento físico dos componentes do sistema, bem como a influência da interação entre eles sobre o nível do reservatório; o modelo do controlador – que representa a atuação do controlador sobre os componentes mecânicos; o modelo de falha – que permite o estudo do comportamento do sistema quando submetido a falhas dos componentes mecânicos e por fim; o modelo de manutenção – em que uma politica de reparo é inserida, sendo possível avaliar a influência da manutenção na minimização das falhas do sistema.

Os modelos desenvolvidos serão verificados através da aplicação do verificador de modelos, da ferramenta computacional UPPAAL STRATEGO, para avaliar se os modelos representam o comportamento (físico e controlado) do sistema e para obtenção da característica de falha do mesmo. Os parâmetros de confiabilidade resultantes da análise estatística de falha do sistema serão comparados com outros existentes na literatura a fim de avaliar se a metodologia empregada é adequada para análise de confiabilidade de sistemas híbridos. Em uma segunda etapa do trabalho, serão realizadas modificações dos parâmetros de probabilidade de falha do sistema, a fim de torná-lo mais fidedigno com sistemas hidráulicos reais, e estimar o tempo médio entre manutenções (*MTBM – Mean Time Between Maintenance*) ótimo deste sistema.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem como objetivo aplicar a modelagem por autômatos estocásticos híbridos e verificação formal em um sistema hidráulico, a fim de avaliar se tal metodologia é adequada para a análise de confiabilidade desse tipo de sistemas.

1.3.2 Objetivos Específicos

- Elaborar e verificar formalmente os modelos em autômatos híbridos do comportamento do sistema hidráulico objeto de estudo;
- Realizar a comparação do comportamento físico, do sistema de controle e da característica falha do sistema modelado com os existentes na literatura, com a finalidade de verificar se a metodologia empregada é adequada para análise de confiabilidade;
- Realizar modificações dos parâmetros de falha do sistema, considerando a função de distribuição de falhas mais representativa com sistemas mecânicos, e aplicar a verificação formal para a predição dos dados de confiabilidade e disponibilidade do sistema;
- Estimar o tempo médio entre manutenções (*MTBM*) ótimo do modelo obtido, para obtenção da máxima disponibilidade do sistema.

1.4 Estrutura do Trabalho

Esta dissertação de mestrado está dividida em 6 capítulos, a saber:

- Capítulo 1: introdução, contendo contextualização, motivação e os objetivos do trabalho.
- Capítulo 2: revisão bibliográfica sobre confiabilidade de sistemas, abordando as principais definições e funções de probabilidade, incluíndo a definição do tipo de sistema em estudo e por fim, fundamentação teórica sobre modelagem por autômatos estocásticos híbridos e verificação formal, sendo introduzida a ferramenta computacional UPPAAL STRATEGO, utilizada neste trabalho;
- Capítulo 3: aborda a modelagem e verificação formal do sistema hidráulico em estudo, em que é exposto a especificação de propriedades, modelagem, simulação e verificação formal dos modelos: físico, de controle, de falha e de manutenção;

- Capítulo 4: é realizada a análise comparativa entre os resultados obtidos com aqueles existentes da literatura, avaliando se a metodologia é adequada para análise de confiabilidade.
- Capítulo 5: contempla a segunda etapa do trabalho, onde os parâmetros de falha do sistema são modificados, modelo de manutenção preventiva incluído e o MTBM do sistema é estimado.
- Capítulo 6: considerações finais referentes ao trabalho e proposto trabalho futuro.

Capítulo 2

Fundamentação Teórica

2.1 Confiabilidade de Sistemas

A partir da necessidade de redução na probabilidade de falhas, seja a consequência dessas falhas de maior ou menor grau de severidade, tornou-se crescente o interesse no conhecimento resultante da análise de falhas e da busca de minimização da sua ocorrência. Neste contexto está inserida a confiabilidade (Fogliatto e Ribeiro, 2009).

Segundo Lafraia (2001), a confiabilidade é definida como a "probabilidade de que um componente ou sistema cumpra sua função com sucesso por um período de tempo previsto, sob condições de operação especificadas", ou seja, indica a probabilidade de que o produto ou sistema não venha falhar até um determinado tempo, e é uma medida quantitativa de descrição do desempenho de um sistema (Vaccaro, 1997).

O conceito de confiabilidade deve estar associado a um período de tempo ou duração de missão. Fogliatto e Ribeiro (2009) exemplificam que não faz sentido afirmar que um item apresenta confiabilidade de 70%, por exemplo, sem especificar durante qual período de tempo a análise do item foi realizada.

Fogliatto e Ribeiro (2009) também destacam a importância de especificar as condições de operação deste item. Por exemplo, dois transistores de igual qualidade são usados em um aparelho de televisão e em um equipamento bélico. Provavelmente o transistor utilizado na televisão apresentará uma confiabilidade maior, pois será utilizado de forma mais amena.

A grande vantagem da aplicação da análise de confiabilidade é o aumento da qualidade associada a diminuição de custo, uma vez que possibilita uma melhor gestão dos programas de manutenção, busca a minimização de falhas o que diminuem o número de acidentes, torna os produtos mais confiáveis aumentando a satisfação do usuário, entre outras.

2.1.1 Funções de Confiabilidade

A confiabilidade tem natureza quantitativa e está diretamente ligada a análise de falhas de um sistema. Segundo Siqueira (2005), "a falha consiste na interrupção ou alteração da capacidade de um item desempenhar uma função requerida ou esperada".

Para aferir a confiabilidade do sistema é preciso estimar a distribuição de probabilidade associada com as suas falhas, sendo elas a distribuição acumulada de falha (*F(t)* ou *cdf – cumulative distribution function)* e a distribuição densidade de falha (*f(t)* ou *pdf – probability density function)* (B-Daya, 2009).

A função acumulada de falha representa o somatório das falhas que ocorreram em um determinado período de tempo, ou seja, considerando uma amostra composta por n_o unidades idênticas submetidas a um teste de sobrevida, a cada instante de tempo é possível avaliar quantas unidades já falharam ($n_f(t)$) e, consequentemente quantas sobreviveram ($n_s(t)$). O percentual acumulado de falhas é a distribuição acumulada de falhas (Guzzon, 2009; Vaccaro, 1997):

$$F(t) = \frac{n_f(t)}{n_o} \qquad t \geq 0 \tag{2.1}$$

Segundo Lafraia (2001), a confiabilidade é a probabilidade de que um item sobreviva a este intervalo de tempo, sendo expressa como:

$$R(t) = \frac{n_s(t)}{n_o} = 1 - F(t) \qquad t \geq 0 \tag{2.2}$$

A frequência de ocorrência das falhas é denominada de distribuição densidade de falha e demostra como as falhas se distribuem no decorrer deste período de tempo (Guzzon, 2009), sendo expressa por:

$$f(t) = \frac{dF(t)}{dt} \tag{2.3}$$

A Figura 2.1 ilustra a representação gráfica das três distribuições de probabilidade apresentadas: distribuição acumulada de falha (*F(t)*), função confiabilidade (*R(t)*) e distribuição densidade de falha (*f(t)*), em que se conclui que a confiabilidade diminui com o tempo à medida que o número de falhas aumenta.

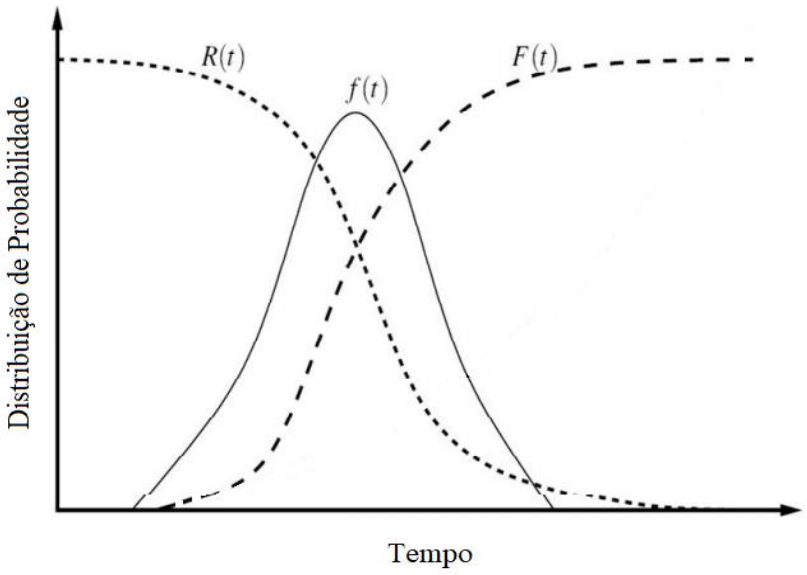


Figura 2.1 – Relação gráfica entre *R(t), F(t) e f(t)*. Fonte: Adaptado de Silva (2012).

2.1.2 Curva da Banheira

A frequência com que as falhas ocorrem num certo intervalo de tempo, medida pelo número de falhas por unidade de tempo, é a taxa de falha, geralmente representada por λ (Lafraia, 2001).

B-Daya (2009) define a taxa de falha como a probabilidade que uma falha por unidade de tempo ocorre no intervalo [t, $t + \Delta t$], dado que uma falha não ocorreu antes de t, o início do intervalo. A função taxa de falha, mais conhecida como função de risco, é dada por:

$$h(t) = \frac{f(t)}{R(t)} \tag{2.4}$$

A função de risco mede a probabilidade de um componente que funcionou até o momento t falhar no próximo instante de tempo (B-Daya, 2009). Uma vez que, se em um dado instante a probabilidade de uma unidade falhar é pequena e sua confiabilidade já atingiu níveis muito baixos, o risco de falha tenderá a ser grande (Vaccaro, 1997).

O inverso da taxa de falhas resulta no tempo médio para falha (*MTTF – Mean Time To Failure*), que indica o valor médio dos tempos até a falha de itens não reparáveis, ou no tempo médio entre falhas (*MTBF – Mean Time Between Failure*), quando os itens podem ser reparados, e é expresso por (Lafraia, 2001; Vaccaro, 1997):

$$MTTF = \frac{1}{\lambda} \quad ou \quad MTBF = \frac{1}{\lambda} \tag{2.5}$$

A Figura 2.2 ilustra três distribuições distintas de falhas sobrepostas, a saber: falhas prematuras, falhas aleatórias e falhas por desgastes, a soma dessas distribuições resulta na curva da banheira, representada na Figura 2.3.

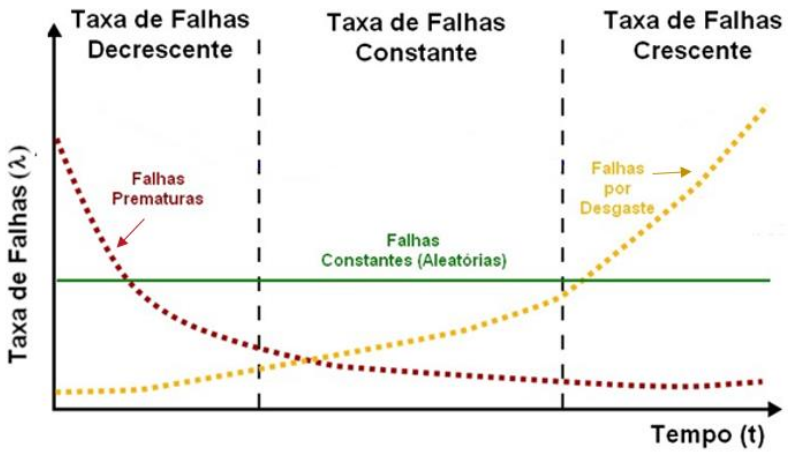


Figura 2.2 – Composição da curva da banheira. Fonte: Adaptado de Natário (2011)

As regiões da curva da banheira estão associadas com os períodos característicos da vida do componente, que são a mortalidade infantil, vida útil e desgaste. Porém nem todo tipo de componente/sistema apresentam sempre todas as fases, por exemplo: softwares são

exemplos típicos de sistema com mortalidade infantil, os componentes eletrônicos apresentam normalmente falhas aleatórias, enquanto que os equipamentos mecânicos geralmente apresentam as três fases (Lafraia, 2001).

No período de mortalidade infantil ocorrem as falhas prematuras, originadas de deficiências no processo de fabricação e defeitos que não foram detectados nos testes de qualidade. Nesse período a probabilidade de ocorrência de falha é alta e decresce com o tempo, à medida que os erros vão sendo corrigidos. São falhas altamente indesejáveis, tanto para o usuário quanto para o fabricante, que deve arcar com os custos de reparo, devido ao período de garantia (Vaccaro, 1997; Lafraia, 2001; Fogliatto e Ribeiro, 2009).

No período de vida útil, as falhas devem-se tipicamente a condições extremas no ambiente de operação e podem ocorrer, uniformemente, em qualquer momento no tempo, os seja, são de natureza aleatória, e são difíceis de serem evitadas. Na vida útil a taxa de falhas é constante (Lafraia, 2001; Fogliatto e Ribeiro, 2009).

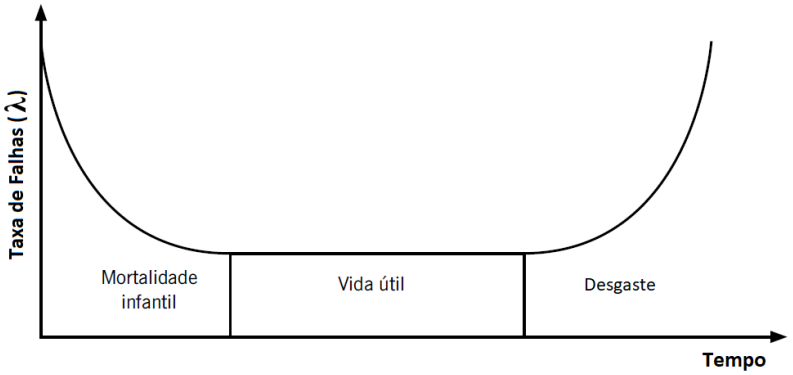


Figura 2.3 – Curva da Banheira. Fonte: Adaptado de Fogliatto e Ribeiro (2009)

O período de desgaste caracteriza o fim da vida do equipamento, e as falhas estão associadas a idade e deterioração do produto. Nesse caso a taxa de falhas cresce continuamente até o fim da vida do equipamento (Lafraia, 2001; Fogliatto e Ribeiro, 2009).

2.1.3 Distribuições de Probabilidade aplicadas a confiabilidade

A partir do comportamento da taxa de falhas de um determinado tipo de sistema, pode-se utilizar uma distribuição de probabilidade para estimar o tempo até a falha desse sistema. Como por exemplo, para sistemas com taxa de falha constante, ou seja, durante a vida útil, é utilizada a distribuição exponencial. Outras distribuições comumente utilizadas em confiabilidade de sistemas são as distribuições de Weibull e Lognormal (Vaccaro, 1997; Fogliatto e Ribeiro, 2009).

Distribuição Exponencial

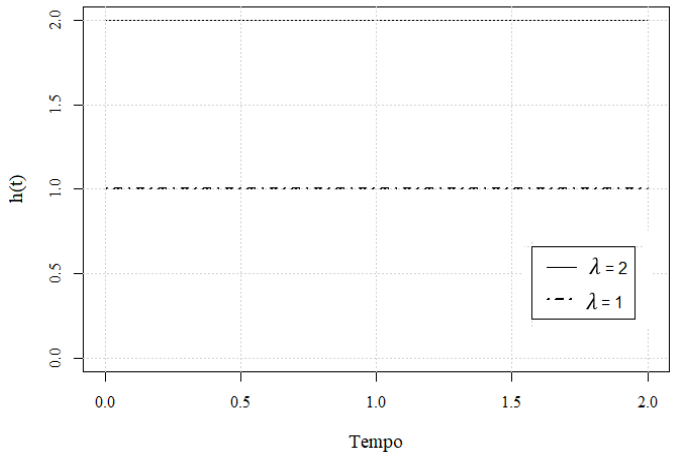
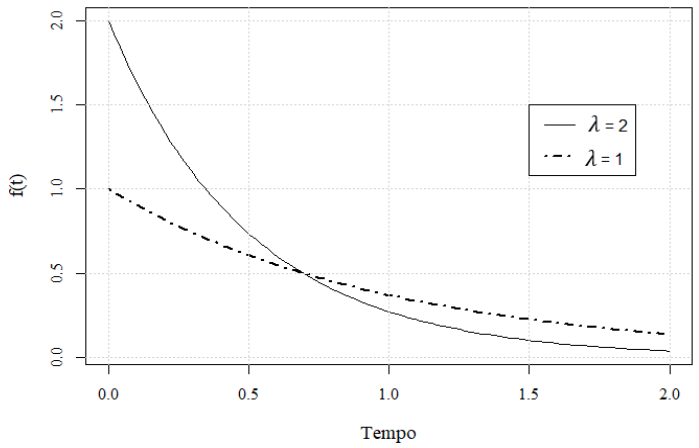
A distribuição exponencial é a distribuição mais utilizada em confiabilidade e avaliação de risco. É a única distribuição com taxa de falha constante e é usada para modelar a vida útil

de muitos sistemas de engenharia (Verma, Ajit e Karanki, 2016). É a distribuição de probabilidade mais simples utilizada, sendo definida apenas por um parâmetro: a taxa de falha (λ) (Guzzon, 2009).

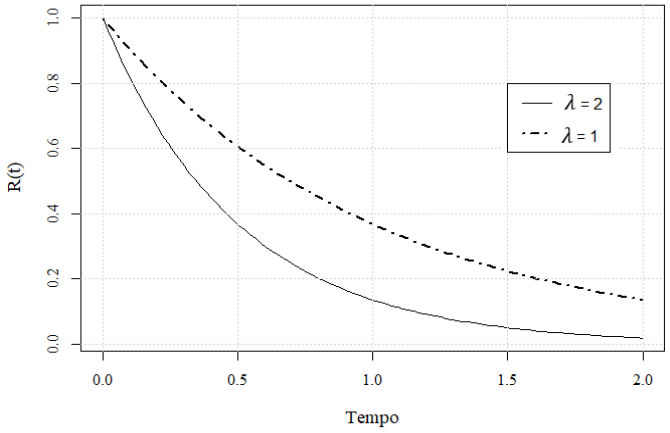
A distribuição exponencial possui a propriedade de ausência de memória, isto é, devido a taxa de falha se manter constante ao longo do tempo, tais sistemas não possuem histórico de eventos passados, consequentemente a probabilidade de um sistema em uso falhar após certo período de tempo é a mesma que a de um sistema novo (Vaccaro, 1997). Tal suposição restringe a aplicação da distribuição exponencial a alguns componentes elétricos. Equipamentos que possuem desgaste ou fadiga são modelados corretamente por essa função apenas no período de vida útil (Fogliatto e Ribeiro, 2009).

A Tabela 2.1 apresenta a representação matemática e gráfica dessa distribuição com relação a sua função de risco, função densidade de falha e função de confiabilidade para o $t \geq 0$.

Tabela 2.1 – Representações de confiabilidade da distribuição exponencial. Fonte: Adaptado de Fogliatto e Ribeiro, 2009

Função	**Representação Matemática**	**Representação Gráfica**
Função de Risco	$h(t) = \lambda$	
Função densidade de falha	$f(t) = \lambda e^{-\lambda t}$	

Continuação da Tabela 2.1 – Representações de confiabilidade da distribuição exponencial. Fonte: Adaptado de Fogliatto e Ribeiro, 2009

Função	Representação Matemática	Representação Gráfica
Função de Confiabilidade	$R(t) = e^{-\lambda t}$	

Distribuição de Weibull

A distribuição de Weibull é "[...] uma das distribuições mais importantes na modelagem de confiabilidade devido à sua flexibilidade e capacidade de representação de amostras de tempos até a falha com comportamentos distintos" (Fogliatto e Ribeiro, 2009).

Três parâmetros descrevem essa distribuição, sendo eles, o parâmetro de forma (β), parâmetro de escala (η) e o parâmetro de posição (γ). A função densidade de falha de Weibull é dada por (Bergamo Filho, 1997):

$$f(t) = \frac{\beta}{\eta}\left(\frac{t-\gamma}{\eta}\right)^{\beta-1} \cdot e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \tag{2.6}$$

O parâmetro de forma permite que a distribuição de Weibull assuma uma variedade de formas, o que permite reproduzir todas as regiões da curva da banheira (Bergamo Filho, 1997; Guzzon, 2009). A Figura 2.4 ilustra os valores de β que resultam nas três regiões da curva banheira, em que $\beta < 1$ representa a mortalidade infantil, $\beta = 1$ representa a vida útil e $\beta > 1$ representa o desgaste.

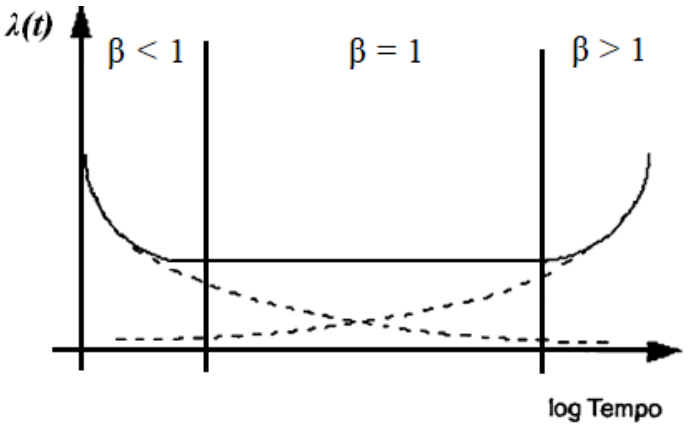


Figura 2.4 – A relação de β e as regiões da curva da banheira. Fonte: Adaptado de Lafraia (2001).

A Tabela 2.2 também é possível observar o comportamento da função taxa de falha para diversos valores de β, comumente encontrados.

Tabela 2.2 - Influência do parâmetro de forma na distribuição de Weibull. Fonte: Lafraia (2001).

β	**Comportamento da Função Taxa de Falha**
< 1	Taxa de falha crescente com o tempo
= 1	Taxa de falha constante – equivale a distribuição exponencial
> 1	Taxa de falha crescente com o tempo
= 2	Taxa de falha linearmente crescente
= 3,2	Distribuição densidade de falha aproxima-se da distribuição normal

A Figura 2.5 apresenta um exemplo da influência do valor do parâmetro de forma sob a distribuição densidade de falha, sendo possível constatar alguns dos comportamentos descritos na Tabela 2.2.

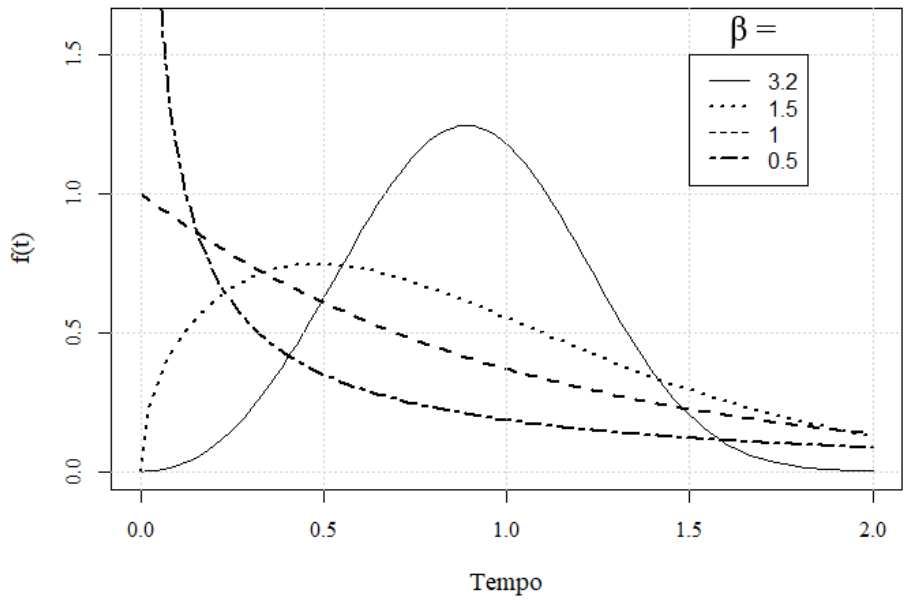


Figura 2.5 – Influência do parâmetro de forma na distribuição de Weibull para $\eta = 1$ e $\gamma = 0$.

O parâmetro de escala (η) determina a largura da curva, de modo que uma variação nesse parâmetro tem como efeito uma mudança no eixo da abcissa, e fará com que a curva se torne mais ou menos achatada (Bergamo Filho, 1997; Guzzon, 2009). A Figura 2.6 ilustra a influência do parâmetro de escala sobre a distribuição de Weibull.

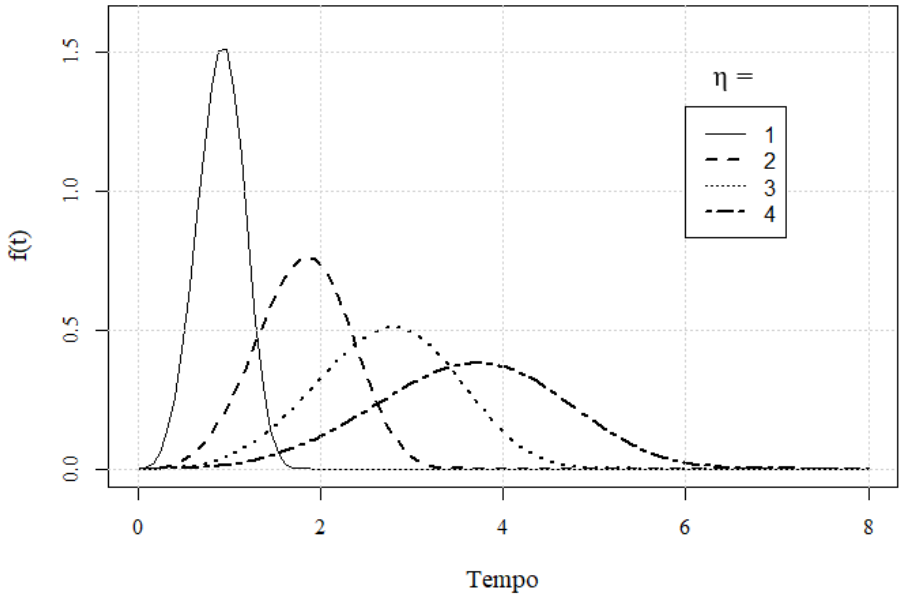


Figura 2.6 – Influência do parâmetro de escala na distribuição de Weibull para $\beta = 4$ e $\gamma = 0$

O parâmetro de localização (γ) indica a posição inicial da distribuição no eixo da abcissa, sendo que para $\gamma = 0$ a distribuição se inicia na origem. Se, $\gamma > 0$ a distribuição começa à direita da origem e se $\gamma < 0$, a distribuição começa à esquerda da origem. A Figura 2.7 ilustra a influência do parâmetro de localização sobre a distribuição de Weibull, ficando claro que valores positivos para este parâmetro são vantajosos, visto que o tempo até a falha será maior (Bergamo Filho, 1997).

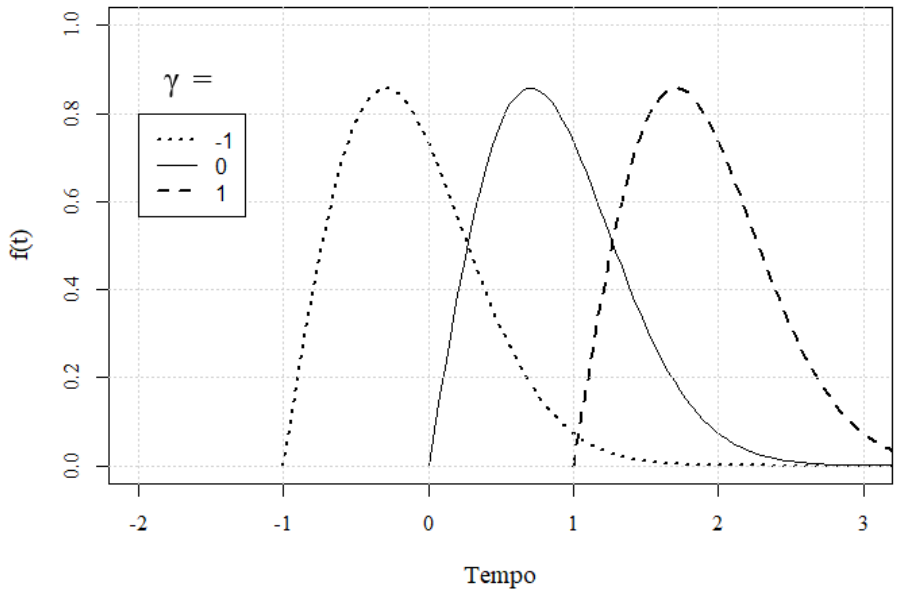
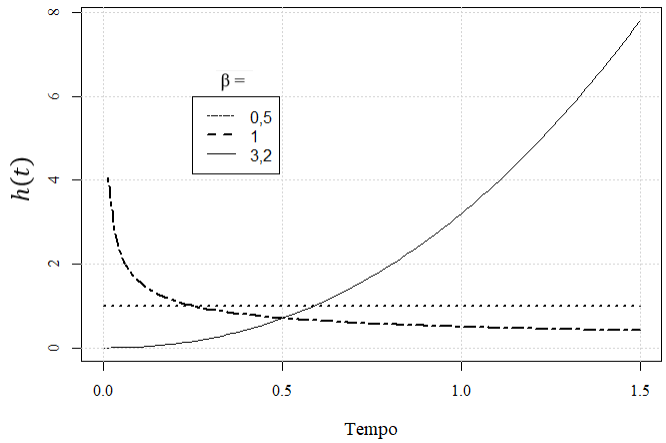
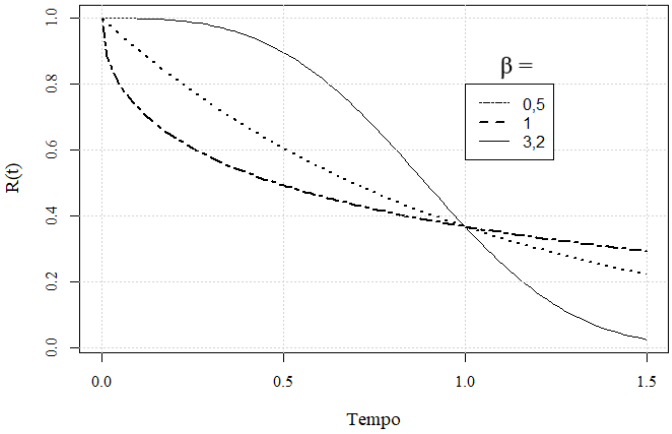


Figura 2.7 – Influência do parâmetro de localização na distribuição de Weibull para $\beta = 4$ e $\eta = 2$.

A Tabela 2.3 apresenta a representação matemática e gráfica dessa distribuição com relação a sua função de risco e função de confiabilidade para o $t \geq 0$, visto que a função densidade de falha já foi apresentada. Para a geração dos gráficos foram mantidos $\eta = 1$ e $\gamma=0$.

Tabela 2.3 - Representações de Confiabilidade da distribuição de Weibull. Fonte: Lafraia (2001)

Função	**Representação Matemática**	**Representação Gráfica**
Função de Risco	$h(t) = \frac{\beta}{\eta}\left(\frac{t-\gamma}{\eta}\right)^{\beta-1}$	
Função de Confiabilidade	$R(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}}$	

Distribuição Lognormal

A distribuição Lognormal é um modelo muito versátil e possibilita o ajuste de muitos tipos de população. É uma distribuição limitada à esquerda, ou seja, não trabalha com valores de $t < 0$. Assim como a distribuição de Weibull, também pode representar qualquer região da curva da banheira, sendo muito utilizada na modelagem de tempos de reparo em unidades reparáveis (Lafraia, 2001; Fogliatto e Ribeiro, 2009; Guzzon, 2009).

É caracteriza por dois parâmetros de distribuição: o parâmetro de forma (σ), que consiste no desvio padrão da função, e o parâmetro de escala (μ), que é a média da distribuição. Possui relação direta com a distribuição normal, em que o logaritmo natural (*ln)*

da variável independente segue a distribuição normal (Lafraia, 2001; Guzzon, 2009; Studart, 2005).

Logo, considerando que *X* é uma distribuição normal dada por $X \sim N(\mu_x, \sigma_x)$ e sabendo que a distribuição Lognormal $Y \sim N(\mu_y, \sigma_y)$ é obtida quando $Y = ln(X)$, tem-se, $X = e^y$. Sabendo que, a distribuição densidade de falha da distribuição normal é dada por (Studart, 2005):

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2} \tag{2.7}$$

Logo, *Y* pode ser determinada, substituindo *x* pelo *y* na equação 2.7:

$$f(y) = \frac{1}{\sigma_y\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{y-\mu_y}{\sigma_y}\right)^2} \tag{2.8}$$

Nota-se que μ_y e σ_y são, respectivamente, a média e o desvio padrão de $ln(x)$, e podem ser obtidas utilizando as seguintes relações:

$$\mu_y = \ln\left(\frac{\mu_x^2}{\sqrt{\sigma_x^2 + \mu_x^2}}\right) \tag{2.9}$$

$$\sigma_y = \sqrt{\ln\left(\frac{\sigma_x^2}{\mu_x^2} + 1\right)} \tag{2.10}$$

É de interesse que se mantenha como variável independente *x*, logo, a distribuição densidade de falha para a distribuição Lognormal é dada por:

$$f(x) = \frac{1}{\sigma_y x\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{\ln x-\mu_y}{\sigma_y}\right)^2} \tag{2.11}$$

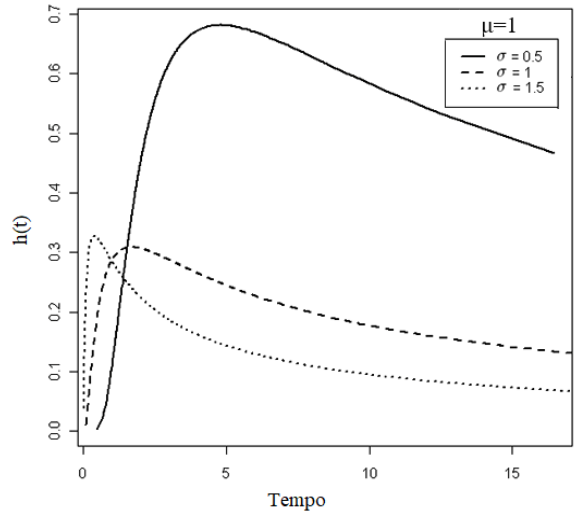
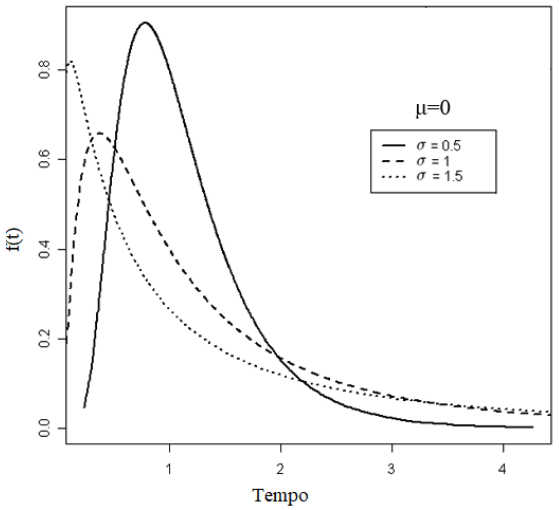
A Tabela 2.4 apresenta a representação matemática e gráfica dessa distribuição com relação a sua função de risco, função densidade de falha e função de confiabilidade, considerando como variável independente o tempo *t*, para simplificação das equações são inseridas as variáveis *z*, φ *e* Φ, que são dados por (Vaccaro, 1997):

$$z = \frac{\ln x - \mu}{\sigma} \tag{2.12}$$

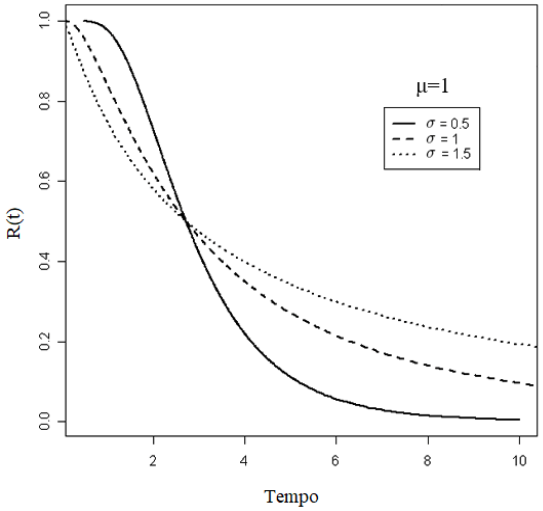
$$\varphi(z) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}z^2} \tag{2.13}$$

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{z} e^{-\frac{1}{2}s^2} \cdot ds \tag{2.14}$$

Tabela 2.4 - Representações de Confiabilidade da distribuição Lognormal. Fonte: Lafraia (2001) e Portal Action (2018)

Função	Representação Matemática	Representação Gráfica
Função de Risco	$h(t) = \frac{0{,}4343}{t \cdot \sigma} \cdot \frac{\varphi(z)}{\Phi(z)}$	
Função densidade de falha	$f(t) = \frac{1}{\sigma_y t\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{\ln t - \mu_y}{\sigma_y}\right)^2}$	

Continuação da Tabela 2.4 - Representações de Confiabilidade da distribuição Lognormal.
Fonte: Lafraia (2001) e Portal Action (2018)

Função	Representação Matemática	Representação Gráfica
Função de Confiabilidade	$R(t) = 1 - \Phi(z)$	

2.1.4 Mantenabilidade e Disponibilidade de Sistemas

Segundo Fogliatto e Ribeiro (2009), "a disponibilidade de equipamentos é um dos principais indicadores de confiabilidade utilizados em programas de manutenção".

Embora, muitos desenvolvimentos teóricos em confiabilidade pressupõem que os componentes são descartados após a primeira falha, na prática, a maioria dos sistemas recebe manutenção, ou seja, são reparadas na ocorrência de falhas (Fogliatto e Ribeiro, 2009; Lafraia, 2001).

Os sistemas reparáveis são aqueles sobre os quais ações de manutenção podem ser aplicadas durante um período de tempo. Essas ações de manutenção podem ser divididas, basicamente em: ações corretivas que tem o objetivo de reparar o equipamento após a falha e trazê-lo ao estado operante no menor tempo possível, e a ações preventivas, que consiste em uma intervenção planejada e programada, que tem por objetivo aumentar a confiabilidade do sistema e retardar a ocorrência de falhas (Fogliatto e Ribeiro, 2009).

A facilidade com que a manutenção é efetuada determina a mantenabilidade de um sistema, que é um parâmetro de projeto e define a facilidade de execução da manutenção, bem como o tempo de manutenção, custos e a função requerida do sistema (Lafraia, 2001).

A norma NBR 5462/1994 define a mantenabilidade como: "capacidade de um item ser mantido ou recolocado em condições de executar suas funções requeridas, sob condições de uso especificadas, quando a manutenção é executada sob condições determinadas e mediante procedimentos e meios prescritos".

Segundo Lafraia (2001), a mantenabilidade afeta diretamente a disponibilidade, visto que o tempo gasto para executar a manutenção retira o sistema do estado disponível. A Figura 2.8 adaptada de Monchy (1989) ilustra a relação entre a confiabilidade e mantenabilidade, e a influência desses parâmetros sobre a disponibilidade do sistema.

Se um sistema possui uma elevada mantenabilidade, significa que caso ocorra uma falha, esta será eliminada rapidamente, colocando o sistema novamente em funcionamento. Uma alta confiabilidade indica que o sistema é pouco suscetível às falhas ao longo de sua vida. Esses dois parâmetros garantem que o sistema terá pouco tempo gasto em "paradas", e consequentemente, maior tempo de operação, ou seja, disponibilidade. Portanto, o aumento na confiabilidade do sistema, bem como o aumento da mantenabilidade, resultam no aumento da disponibilidade do sistema (Sakurada, 2013).

Disponibilidade pode ser definida como a "capacidade de um item estar em condições de executar uma certa função em um dado instante ou durante um intervalo de tempo determinado, levando-se em conta os aspectos combinados de sua confiabilidade, mantenabilidade e suporte de manutenção, supondo que os recursos externos requeridos estejam assegurados" (NBR 5462/1994).

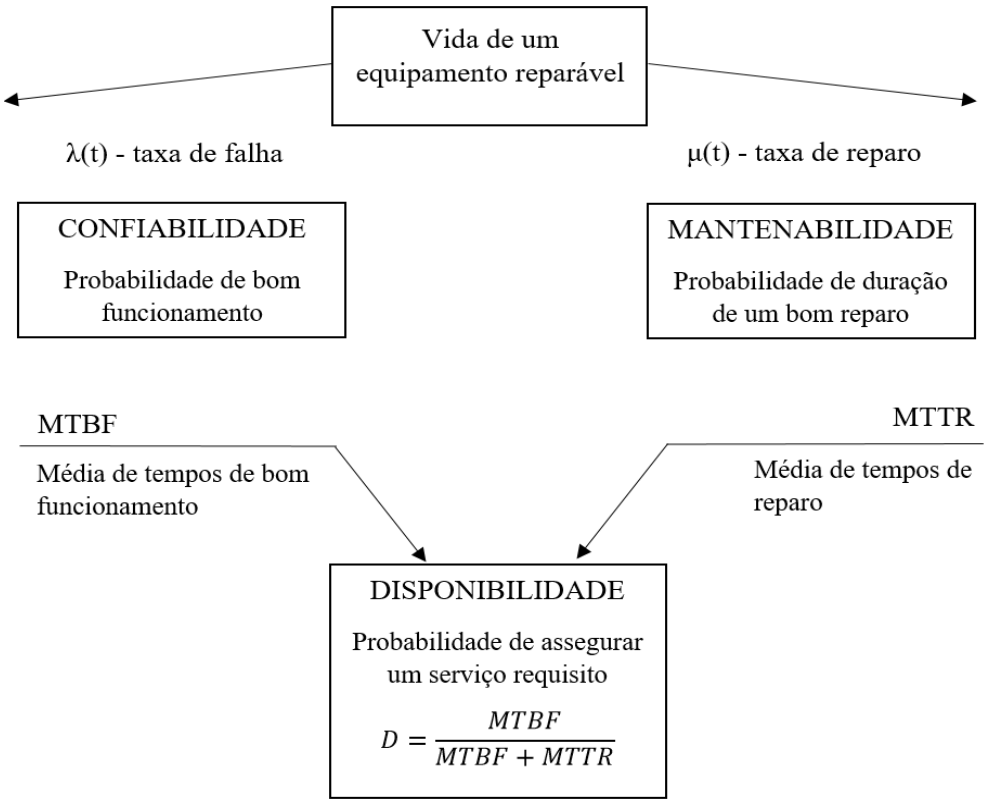


Figura 2.8 – Relação entre confiabilidade, mantenabilidade e disponibilidade. Fonte: Adaptado de Mochy (1989)

Para Fogliatto e Ribeiro (2009), o conceito de disponibilidade varia conforme a capacidade de reparo do sistema. Para sistemas não-reparáveis, a disponibilidade se torna

igual a sua confiabilidade, já em sistemas reparáveis, os possíveis estados do sistema em um tempo *t* de análise são: funcionando ou em manutenção (Ver Figura 2.9).

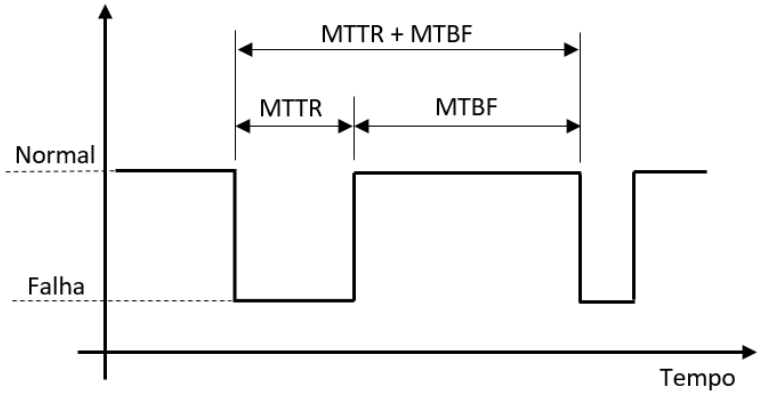


Figura 2.9 – Relação entre disponibilidade e situação operacional do sistema. Fonte: Adaptado de Lafraia (2001)

Nesses casos, costuma-se supor que reparos devolvam o sistema à condição de novo e trabalha-se com um valor médio de disponibilidade, dado por:

$$\mathrm{D} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MTTR}} \tag{2.15}$$

Onde D denota a disponibilidade média do sistema, *MTBF* o tempo médio entre falhas (ou seja, o tempo médio de funcionamento) e *MTTR* o tempo médio até conclusão de reparos (Fogliatto e Ribeiro, 2009).

O *MTBF* está relacionado com a confiabilidade, uma vez que quanto maior for o tempo médio entre as falhas, maior será o período de tempo que o sistema fica operante e, consequentemente, maior será sua confiabilidade. Já o *MTTR* está relacionado com a mantenabilidade, pois quanto menor for o tempo de reparo, maior será a mantenabilidade do sistema (Sakurada, 2013).

Tendo em vista a estreita relação entre mantenabilidade e disponibilidade, a principal função do programa de manutenção é controlar o estado e garantir a disponibilidade do sistema, identificando a frequência ótima de realização de manutenções preventivas. Este é um problema de otimização, visto que o aumento na frequência das ações preventivas aumenta o custo total de manutenção, porém aumenta a confiabilidade (Fogliatto e Ribeiro, 2009).

2.2 Definição de Sistemas Híbridos

Lazar (2006) define os sistemas híbridos como sistemas cujo comportamento é caracterizado por muitos modos de operação, em que cada modo é descrito por processos contínuos, representado por equações diferenciais, e a transição entre os vários modos de operação ocorrem quando um evento particular acontece, caracterizando uma dinâmica discreta do sistema.

Em concordância com essa definição, Costa (2008) afirma que os sistemas híbridos são aqueles que possuem simultaneamente variáveis contínuas e variáveis discretas, e que a evolução de um sistema híbrido se dá tanto em função do tempo como em função da ocorrência de eventos discretos.

Os eventos discretos são causados pela evolução da dinâmica contínua do sistema ou por estímulos externos. Já a mudança na dinâmica contínua é uma resposta aos eventos discretos (Krilavius, 2006). De acordo com Engel (1998) *apud* Caetano (2011), alguns fatores que originam os eventos discretos são:

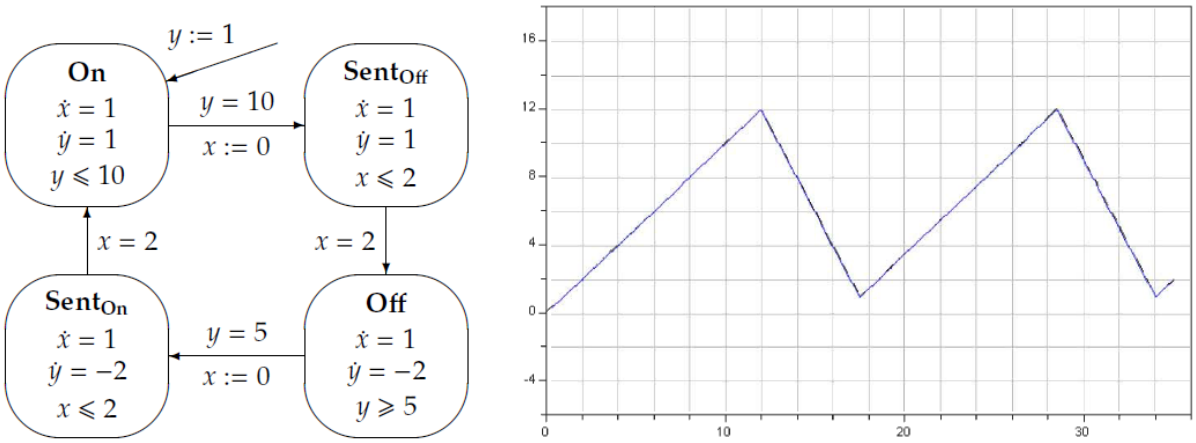
- Transição de fase no sistema, por exemplo mudança de nível de um reservatório;
- Operações de processos descontínuos, ou seja, operação ocorre em etapas;
- Instrumentos que retornam saídas discretas;
- Atuadores com características discretas, como válvulas liga-desliga;
- Início e fim de operação em processos contínuos.

O interesse no estudo de sistemas híbridos se deve ao grande número de sistemas que podem ser descritos com esse tipo de abordagem (Caetano, 2011). Estes sistemas podem apresentar diferentes tamanhos e complexidades, sendo que sua aplicação vai desde sistemas simples com poucos estados discretos e comportamento contínuo simples, até sistemas complexos com dinâmica contínua não linear e grande número de transição entre os modos de operação, como por exemplo: sistemas de controle de trafego aéreo e automotivo, aparelhos eletrônicos como micro-ondas, protocolos de comunicação em tempo real, sistemas contínuos com falhas, dentre outros (Krilavius, 2006).

Como exemplo ilustrativo de sistemas híbridos pode-se citar os sistemas hidráulicos, que variam desde sistemas simples de controle do fluido de um único reservatório até circuitos hidráulicos complexos com combinação de tanques, bombas e válvulas. O objetivo principal desses sistemas é manter o nível de fluido dentro dos valores estipulados (Krilavius, 2006).

Krilavius (2006), através da Figura 2.10a ilustra um exemplo de um o modelo em autômatos híbridos do comportamento contínuo/discreto de um sistema hidráulico, em que o fluido do reservatório é monitorado por um controlador que comanda o ligamento e desligamento da bomba. A mudança de nível (ver Figura 2.10b) ocorre através de uma função linear com o tempo, quando a bomba está desligada (*Off),* o nível do reservatório (*y),*

apresenta uma taxa descrente de 2 unidades por segundo, quando a bomba está ligada a taxa de enchimento do reservatório é de 1 unidade por segundo, dependendo do nível do reservatório o controlador envia os comandos de ligamento (*Sent$_{On}$)* ou desligamento (*Sent$_{Off}$)* para a bomba.



(a) (b)

Figura 2.10 – Exemplo ilustrativo: Modelo em autômatos híbridos. Fonte: Krilavius (2006)

Para o desenvolvimento de um modelo simples e confiável para a representação do sistema híbrido, é fundamental entender a interação entre os processos contínuos, eventos discretos e o controle do sistema (Lazar, 2006; Caetano, 2011).

Pela própria complexidade em que se dá a interação de eventos discretos com variáveis contínuas, não existe uma única forma de realizar a modelagem de sistemas híbridos e diversas abordagens são utilizadas no tratamento deste tipo de sistema. As principais soluções que são trabalhadas atualmente são baseadas em Redes de Petri e Autômatos Híbridos (Costa, 2008).

Segundo Caetano (2011), o autômato híbrido é uma das formas mais empregadas para modelar e analisar os sistemas híbridos, visto que estes sistemas são cada vez mais empregados em aplicações de elevada criticidade que demandam alta confiabilidade do sistema, justificando o emprego de técnicas formais para o estudo destes sistemas.

2.3 Modelagem e Verificação Formal de Sistemas Híbridos

Molossi, Almeida e Vale (2009), destacam que na área da computação é fundamental que na fase de projeto haja a garantia de corretude e completude, a fim de detectar possíveis erros, evitando-os e eliminando-os, uma vez que, quanto mais rápido são detectados menores

serão os custos econômicos para resolvê-los. Esta ideia não se restringe apenas a sistemas computacionais, mas ao projeto de sistemas em geral.

A detecção de erros é feita a partir da análise do sistema. Para tanto, as técnicas mais usuais de verificação são: simulação e testes. A simulação é um processo de construir um modelo computacional de um sistema real e conduzir experimentos computacionais com este modelo com o propósito de analisar o seu comportamento sob determinadas condições de operação. Já os testes se referem a experimentos com protótipos físicos, em que é possível avaliar se o sistema real se comporta de maneira correta nos cenários avaliados.

Estes métodos cobrem apenas um conjunto limitado de comportamentos do sistema, não havendo garantia que todas as falhas foram diagnosticadas. Por isso, o uso da verificação formal se apresenta como uma alternativa (Salas, 2014). Além disso, os métodos formais são altamente recomendados no desenvolvimento de sistemas críticos segundo o guia de melhores práticas da IEC *(International Electrotechnical Commission)* e da ESA *(European Space Agency*) (Baier e Katoen, 2008).

Segundo Molossi, Almeida e Vale (2009), a verificação formal corresponde a um conjunto de técnicas para análise automática de sistemas complexos, sendo duas técnicas bem estabelecidas: a prova de teoremas e a verificação de modelos (*model checking).* A primeira se refere ao uso de axiomas e regras matemáticas para descrever o comportamento do sistema e lógica para avaliar as propriedades do mesmo. Já a verificação de modelos pode ser definida como uma técnica de verificação que explora todos os possíveis estados do sistema, tornando possível determinar se o modelo do sistema realmente satisfaz uma determinada propriedade (Baier e Katoen, 2008).

Baier e Katoen (2008) comparam a verificação de modelos com um programa computacional de xadrez, em que o computador verifica todos os possíveis movimentos que podem ser realizados. Da mesma forma, um verificador de modelos examina todos os cenários possíveis do sistema de modo sistemático.

A verificação formal tem como objetivo avaliar a necessidade de ajuste do sistema em relação a uma determinada propriedade ou especificação de projeto. Para isso, um modelo do sistema é criado e as exigências do sistema são formuladas em linguagem formal. Na sequência, um conjunto de regras é aplicado para determinar se o modelo satisfaz os requisitos a ele impostos, sendo possível avaliar se a combinação do comportamento contínuo e discreto se comportará como desejado (Caetano, 2011).

Para a análise, o verificador de modelos examina todos os estados do sistema relevantes para averiguar se a propriedade é satisfeita. Se em algum estado a propriedade em consideração é violada, o verificador fornece um contraexemplo que indica o caminho que leva o modelo a atingir o estado indesejável, descrevendo a sequência de execução, desde o estado inicial até o estado que viola a propriedade a ser verificada. Esta informação é útil para a correção do modelo (Baier e Katoen, 2008). A Figura 2.11 ilustra um esquemático da metodologia de verificação de modelos.

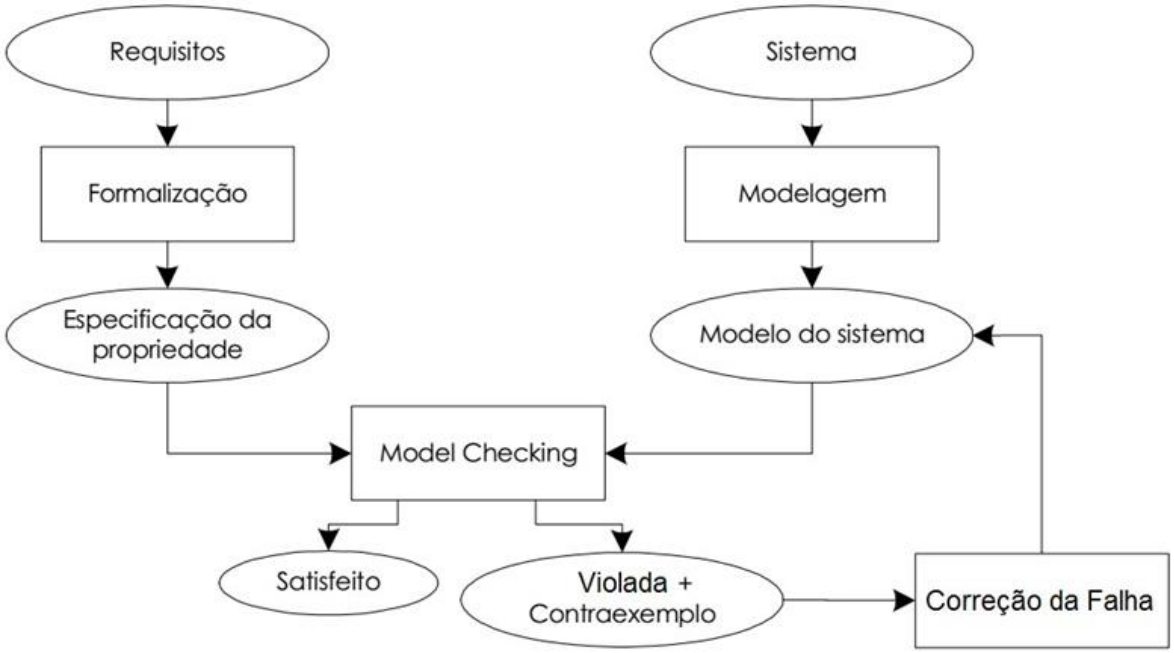


Figura 2.11 – Metodologia da Verificação de Modelos. Fonte: Adaptado de Baier e Katoen (2008).

A verificação formal pode ser dividida em três etapas principais: a modelagem, a especificação e a verificação em si (Molossi, Almeida e Vale, 2009).

2.3.1 Modelagem de autômatos estocásticos híbridos

O modelo do sistema deve ser criado de forma matemática e abstrata, para que os algoritmos de verificação possam ser aplicados (Molossi, Almeida e Vale, 2009). Nessa etapa algumas simplificações podem ser realizadas, uma vez que o modelo deve descrever satisfatoriamente como o sistema se comporta, e não o seu detalhamento estrutural. Portanto, a escolha do formalismo utilizado para a modelagem deve levar em conta as características necessárias para descrever o sistema híbrido com eficiência (Caetano, 2011).

O modelo deve representar de maneira fidedigna o comportamento real das variáveis do sistema sob estudo, sem comprometer a verificação formal por incoerências, tanto devido a modelos simples, que não representam o comportamento desejado, quanto por modelos complexos, que podem demandar grande esforço computacional (Santos, 2017).

A modelagem por autômatos híbridos é a abordagem mais popular para modelar e analisar sistemas híbridos. Basicamente, o autômato híbrido (Figura 2.12) combina os dois tipos de comportamento do sistema, sendo que as mudanças discretas são descritas pelas transições, que são compostas pelas *guardas*, *sincronizações*, *reinicialização e atualizações de variáveis*. O comportamento contínuo é descrito nos locais, que podem ser compostos pelas condições de fluxo e pelas *invariantes* (Krilavius, 2006).

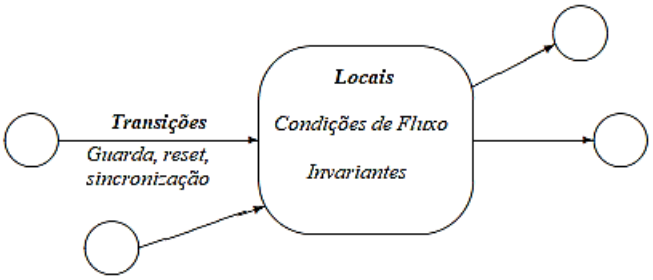


Figura 2.12 – Autômato Híbrido. Fonte: Adaptado de Krilavius (2006).

A descrição formal de autômatos híbridos é dada por Krilavius (2006) como:

Definição 1 (Autômato Híbrido). *Um autômato híbrido é uma tupla H = (X, L, Init, Inv, f, E, Guard, Assign, Σ), onde:*

- $X \subseteq \mathbb{R}^n$ *é o espaço de estados das variáveis contínuas e* $x = (x_1, x_2, \ldots, x_n)$, *onde* $x_i \in \mathbb{R},\ i = 1, 2, \ldots, n$, *representa a dinâmica contínua.*
- *L é o conjunto finito de estados discretos.*
- $Init \subseteq L \times X$ *é o conjunto de condições iniciais do autômato.*
- $Inv: L \to 2^X$ *atribui para cada local l uma invariante que deve ser satisfeita pelo estado x no local l.*
- $f: L \to (X \to \mathbb{R}^n)$ *atribui para cada local l a dinâmica das variáveis contínuas através de equações diferenciais.*
- $E \subseteq L \times \sum \times L$ *é o conjunto de transições do autômato, onde Σ é o conjunto de rótulos de transição.*
- $Guard: E \to 2^X$ *atribui para cada transição uma guarda que deve ser satisfeita para que o estado x faça a transição.*
- $Assign: E \to (X \to X)$ *atribui para cada transição uma tarefa que pode atualizar o valor das variáveis quando ocorre a transição.*

Para modelagem de sistemas complexos é conveniente dividi-lo em vários componentes, cada qual modelado por um autômato híbrido separado. As interações entre eles são determinadas com a operação de composição paralela que resultará em um autômato híbrido único. A interação pode ocorrer de duas formas: cada parte pode se sincronizar por meio de transições discretas, via rótulos de sincronismo ou apenas compartilhando variáveis (Santos, 2017).

A fim de melhorar a qualidade do modelo e avaliar se o comportamento do mesmo está coerente, uma simulação antes da verificação do modelo pode ser realizada. Esta simulação pode auxiliar na identificação de erros simples de modelagem ou até mesmo do sistema. Esta etapa tem como vantagem reduzir os custos e tempo consumido na verificação do modelo (Baier e Katoen, 2008).

2.3.2 Especificação de propriedades

Antes da verificação, é necessário declarar as propriedades que o sistema proposto deve satisfazer. A especificação é geralmente dada em formalismo lógico (Molossi, Almeida e Vale, 2009). Um formalismo lógico que pode ser utilizado para a descrição das propriedades é a lógica temporal, que é uma extensão da lógica proposicional tradicional (Baier e Katoen, 2008). Ou seja, é basicamente uma sentença declarativa, afirmativa ou negativa, na qual pode se atribuir um valor lógico verdadeiro ou falso.

As lógicas temporais mais utilizadas na área de verificação formal são: LTL (*Linear Temporal Logic)*, que estabelece uma regra para um determinado caminho do modelo em análise, e CTL (*Computation Tree Logic)*, que permite a expressão de propriedades sobre um ou todos os caminhos possíveis. Semanticamente, as fórmulas baseadas em LTL são construídas associando-se operadores temporais e proposições atômicas, já em CLT, inclui-se também os quantificadores de caminho (Santos, 2017).

Em termos gerais, a linguagem CLT é composta pelos seguintes operadores (Macêdo et al, 2004; Santos, 2017):

- Lógicos: *and, or, imply, not;*
- Temporais: [ ] *(Sempre),* <> *(futuramente), ( ) (no próximo estado),* ∪ *(até que);*
- Quantificadores de caminho: *A (para todo caminho), E (existe um caminho);*

As proposições atômicas são relacionadas com os operadores supracitados, a fim de formalizar as condições de interesse nos estados do sistema e determinar as propriedades que serão verificadas (Santos, 2017). A sintaxe e semântica de algumas formulações baseadas em CTL são descritas no Quadro 2.1.

Quadro 2.1 – Sintaxe e Semântica de formulações baseadas em CTL. Fonte: Adaptado de Macêdo (2004) e Santos (2017).

Especificação	**Descrição**
`A[]p`	Para todos os caminhos, em todos os estados, *p* sempre acontece.
`A<>p`	Para todos os caminhos, *p* futuramente acontece.
`A()p`	Para todos os caminhos, no próximo estado, *p* acontece.
`A(p∪q)`	Para todos os caminhos, *p* é verdade até que *q* seja satisfeita.
`E[]p`	Existe ao menos um caminho em que, em todos os estados, *p* sempre acontece.
`E<>p`	Existe ao menos um caminho onde *p* futuramente acontece.
`E()p`	Existe ao menos um caminho onde no próximo estado *p* acontece.
`E(p∪q)`	Existe ao menos um caminho onde *p* é verdade até que *q* seja satisfeita.

Para tornar possível uma verificação rigorosa, as propriedades devem ser descritas de forma precisa e inequívoca. A especificação de propriedade prescreve o que o sistema deve fazer e o que ele não deve fazer, enquanto a descrição do modelo aborda como o sistema se comporta (Baier e Katoen, 2008).

2.3.3 Verificação do modelo

Na última etapa, o verificador checa a validade da propriedade no modelo do sistema de maneira completamente automática (Baier e Katoen, 2008). Basicamente, consiste em executar uma análise matemática de todas as trajetórias possíveis para o estado do sistema. Quando o verificador analisa uma trajetória e a rede de autômatos retorna ao estado inicial, significa que naquela trajetória a propriedade foi satisfeita e outra trajetória pode ser iniciada até que todas as transições possíveis de todos os estados sejam executadas (Engell, 1998 *apud* Caetano, 2011).

Segundo Baier e Katoen (2008), na fase de análise dos resultados, segue-se a sequência de procedimentos descrita na Figura 2.13:

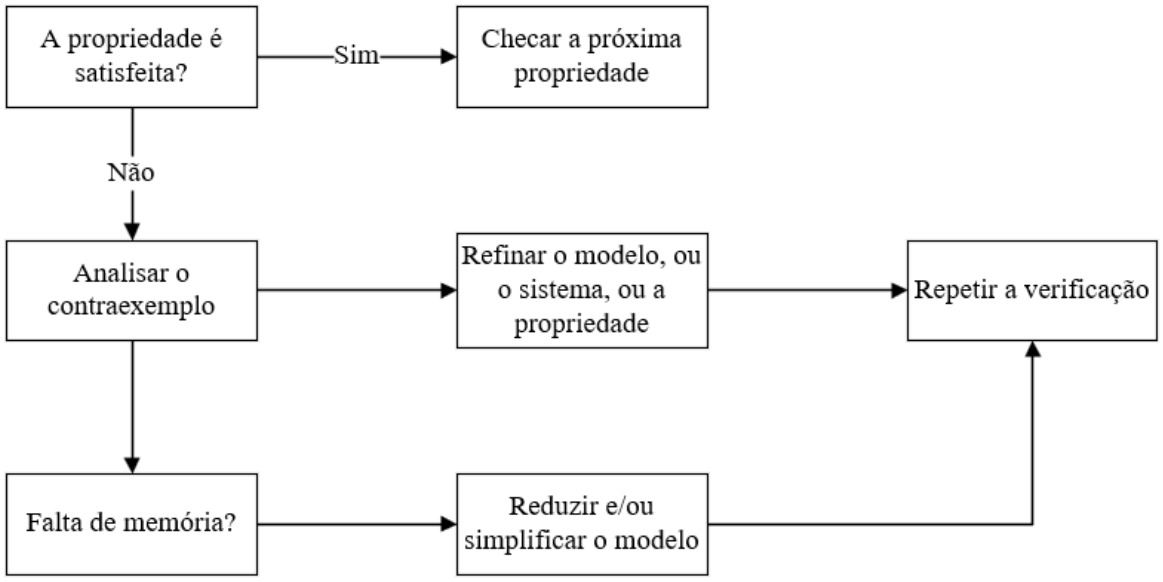


Figura 2.13 - Análise dos resultados do verificador de modelos. Fonte: Adaptado de Baier e Katoen (2008).

Quando se tem a violação da propriedade, a causa pode ser: erro de modelagem, erro de projeto ou a propriedade não reflete a exigência que deve ser validada. Quando se tem um erro de modelagem é necessário corrigir o modelo; caso o modelo seja consistente com o projeto, é necessário reprojetar o sistema e ajustar o seu modelo para essa nova concepção. Nesses casos, é necessário refazer a verificação da propriedade violada e de todas as propriedades verificadas anteriormente. Caso se detecte erro de propriedade, isto implica em uma modificação da mesma, onde o modelo não é alterado, sendo necessário refazer a verificação apenas da propriedade violada (Baier e Katoen, 2008).

Segundo Baier e Katoen (2008), a lógica temporal permite a especificação de uma ampla gama de propriedades relevantes do sistema que podem ser verificadas no verificador de modelos. Santos (2017) descreve algumas das propriedades:

- *Condição de Travamento: é possível acabar em um estado de deadlock?*
  Uma condição de *deadlock* é detectada quando na rede de autômatos não for possível realizar a transição de um estado para o estado sucessor.
- *Acessibilidade: o sistema faz o que deveria fazer?*
  Esta propriedade é utilizada ao projetar um modelo para verificar sua funcionalidade, ou seja, verifica se o funcionamento básico do sistema é possível de ocorrer.
- *Segurança: Algo ruim nunca acontecerá?*
  Geralmente formulada de forma positiva, esta propriedade é usada para expressar que determinada condição nunca irá acontecer.
- *Vivacidade: Alguma coisa boa acontecerá eventualmente?*
  Esta propriedade expressa que determinada condição eventualmente pode ocorrer.

Uma série de vantagens pode ser atribuída à verificação de modelos (Baier e Katoen, 2008), sendo elas:

- É aplicável a uma ampla gama de sistemas;
- Oferece suporte à verificação parcial, ou seja, as propriedades podem ser verificadas individualmente, permitindo assim o foco sobre as propriedades essenciais primeiro.
- Fornece informações de diagnóstico caso uma propriedade seja invalidada (muito útil para fins de depuração);
- É completamente automático;
- Tem um interesse crescente pela indústria, principalmente na área de computação;
- Possui base sólida e matemática, pois é baseada na teoria dos grafos e estruturas de dados e lógica.

Baier e Katoen (2008) apresentam também algumas desvantagens do método:

- Aplicável apenas em sistemas com estados finitos;
- Verifica um modelo de sistema, e não o próprio sistema, ou seja, qualquer resultado obtido é tão bom quanto o modelo do sistema.
- Verifica apenas os requisitos estabelecidos, ou seja, não há garantia de completude. A validade de propriedades que não são verificadas não pode ser julgada.
- Sofre do problema de explosão de estados, ou seja, o número de estados necessários para modelar o sistema com precisão pode facilmente exceder a quantidade de memória do computador disponível.
- Seu uso requer alguma perícia em encontrar abstrações apropriadas para obter modelos menores do sistema e para indicar propriedades no formalismo lógico usado.

- Não é garantido que produza resultados corretos: como qualquer ferramenta, um verificador de modelo pode conter defeitos de software.

Ainda assim, a verificação formal fornece um aumento significativo no nível de confiança no projeto do sistema (Baier e Katoen, 2008). Fica evidente que, com a dependência tecnológica que cresce exponencialmente no atual cenário global, erros em sistemas híbridos estão ficando cada vez mais indesejáveis, visto que são utilizados em uma vasta gama de aplicações. Por isso, é importante evitar que possíveis falhas avancem nas etapas de um projeto, sendo justificável, para tanto, uma quantidade considerável de tempo para que as devidas verificações sejam feitas acerca do problema. Neste aspecto, a verificação formal surge como uma alternativa mais eficiente, uma vez que traz, em sua concepção, a ideia de analisar precisamente um modelo dentro de uma série de condições pré-estabelecidas (Molossi, Almeida e Vale, 2009).

2.3.4 Verificador de Modelos: UPPAAL

Para a modelagem e verificação formal de sistemas híbridos, pode ser utilizada a ferramenta computacional UPPAAL, que foi desenvolvida pela Universidade de Uppsala, na Suécia, juntamente com a Universidade de Aalborg, na Dinamarca. Como justificado por Santos (2017), sua escolha foi motivada, pois:

- A construção dos modelos em autômatos temporizados é realizada através de representação gráfica, tornando-se mais intuitiva;
- O UPPAAL agrega ferramentas para modelagem (gráfica), simulação (gráfica) e verificação (via conferência automática de modelos) em um mesmo ambiente;
- As diversas extensões na linguagem de modelagem, como tipo dos estados da rede de autômatos (*normal, urgente ou commited*), canais de sincronização e tipos de variáveis disponíveis (incluindo a definição de tipos pelo usuário) permitem a representação de diversos comportamentos com um elevado grau de realismo;
- O UPPAAL é uma ferramenta consolidada na área de verificação formal, tendo seu uso disseminado no ambiente acadêmico, com primeira versão datada de 1995, e está em constante atualização por parte de seus desenvolvedores.

No UPPALL, a modelagem é realizada através de uma rede de autômatos de estados finitos com restrição de tempo e suas propriedades são descritas através da linguagem TCTL (*Timed Computational Tree Logic),* que é uma extensão à lógica CTL e proporciona a capacidade de expressar propriedades de tempo real (Kunz, 2012).

Diversas ferramentas são derivadas da sua arquitetura clássica, sendo que o UPPAAL STRATEGO, pode ser utilizado para a representação de modelos através de redes de autômatos cujo comportamento pode depender de características estocásticas e dinâmicas não lineares, ou seja, a modelagem do sistema pode ser realizada por autômatos estocásticos híbridos (Santos, 2017).

A interface gráfica do UPPAAL é dividida: *editor*, utilizado para a construção do modelo; *simulator*, utilizada para avaliar o comportamento do sistema de modo randômico e mostrar o contraexemplo (ou testemunha) gerado na verificação do modelo e, *verifier*, ambiente onde se implementam as propriedades e é realizada sua verificação.

O UPPAAL permite a representação de uma rede de autômatos temporizados, onde os círculos representam os estados que o modelo pode alcançar e as setas indicam a transição entre eles. Os estados podem ser do tipo *inicial, normal, urgente e commited*. A Figura 2.14 ilustra a representação de cada um destes estados, sendo:

- *Estado Inicial:* representa o ponto inicial do autômato, é identificado por um círculo interno;
- *Estado Normal:* estados intermediários do autômato. Deve ser composto por uma *invariante*, que são condições que indicam até que ponto o modelo pode permanecer nesse estado;
- *Estado Urgente:* enquanto o autômato permanece neste estado, os relógios permanecem inalterados, ou seja, não há variação do tempo. Kunz (2012) explica que estes estados são semanticamente semelhantes à adição de um relógio extra *x*, que reinicia em todas as entradas e tem uma invariante x <= 0. É identificado pela letra “U”.
- *Estado Commited:* mas restritivos que os estados *urgentes*, além de manter os relógios inalterados, a transição deve ser realizada imediatamente. É identificada pela letra “C”.

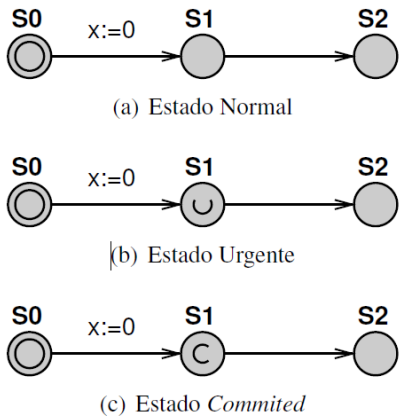


Figura 2.14 – Representação dos estados (a) *normal, (b) urgente* e (c) *commited.* Fonte: Kunz (2012).

Tanto o estado *urgente* como *commited* permitem modelar sequências de ações que não envolvem o tempo.

As *invariantes* associadas aos estados, são condições de permanência do autômato no estado. Caso não haja possibilidade de transição após o término dessa condição, o sistema entrará em *deadlock.* As *invariantes* são expressões que podem envolver relógios, variáveis inteiras e constantes, geralmente utilizam os operadores <, >, ≤, ≥ e ==.

Para estados sem *invariantes* as transições são realizadas de acordo com a taxa de distribuição exponencial, que deve ser declarada no campo *rate of exponencial*. A utilização desta função significa que o autômato pode permanecer no estado até um tempo *t* calculado pela função de distribuição exponencial.

As transições devem indicar sob qual condição é permitido que o autômato transite para o próximo estado. Estas condições podem ser implementadas por meio das *guardas,* que , assim como as *invariantes* são expressões utilizadas para comparação entre variáveis e relógios, que indicam quando a transição deve ocorrer.

Outra forma de ocorrer as transições é através dos *canais de sincronização*, que corresponde ao envio e recepção de sinais entre diferentes modelos que compõe a rede de autômatos. A etiqueta da transição emissora é rotulada pelo final "!" e a da receptora com "?". As sincronizações podem ser declaradas como *binários* ("chan c") ou *broadcast* ("broadcast chan"). A Figura 2.15 ilustra a diferença entre estes dois tipos de canais de sincronização, sendo que:

- *Canais Binários:* Deve haver ao menos uma transição receptora (c?) aguardando o sinal da transição emissora (c!), senão o modelo entra em falha. No caso de haver mais de um receptor e quando mais de uma transição está habilitada para receber o sinal, apenas umas delas o recebe, a escolha de qual receberá o sinal é randômica.
- *Canais broadcast:* Diferente dos canais *binários,* todas as transições que aguardam o sinal de sincronização (c!) o recebem ao mesmo tempo. Neste tipo de canal não é necessário ter receptor do sinal.

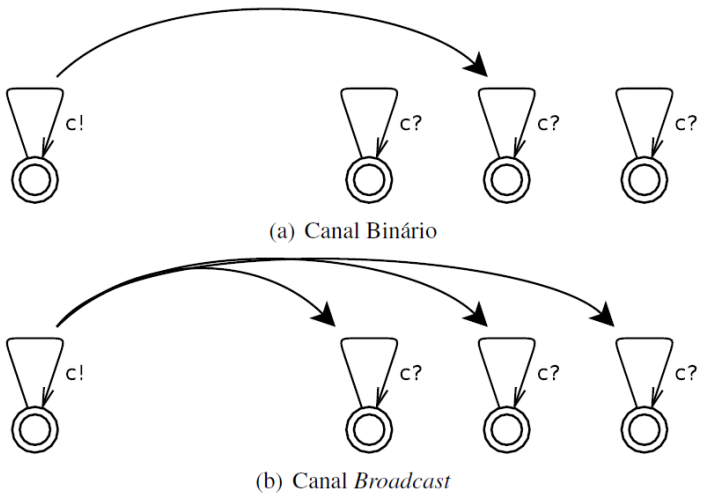


Figura 2.15 – Diferença entre os canais de sincronização *binários* e *broadcast.* Fonte: Kunz (2012).

Nas transições também é possível realizar a atualização de variáveis e a chamada de funções, no campo *update*. O UPPAAL admite diversos tipos de variáveis como booleana, inteira, vetores, constantes, *double*, relógios, relógios híbridos e, além das funções pré-definidas como *sin(), cos(), random(),* permite a implementação de funções em linguagem semanticamente semelhante à linguagem C.

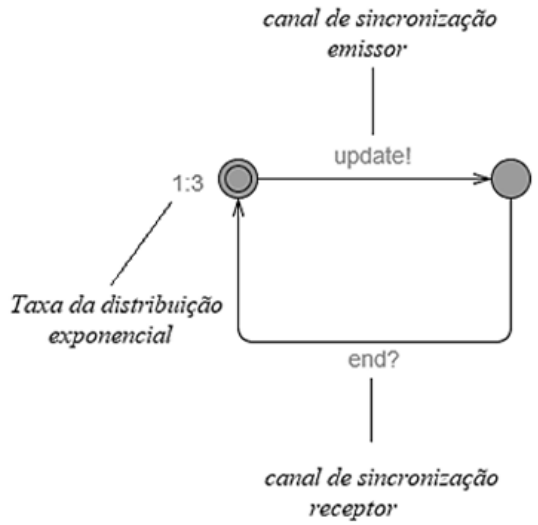
Sistemas complexos ou que envolvem vários processos e componentes podem ser modelados como uma rede de autômatos, sendo que cada autômato representa uma parte da estrutura ou do comportamento global do sistema. Cada autômato é definido por um *template* sendo que a integração entre eles pode ser realizada pelos canais de sincronização ou variáveis compartilhadas.

Na Figura 2.16 é apresentado um exemplo de rede de autômatos modelada no UPPAAL STRATEGO. Nesse modelo, é possível observar a interação de alguns parâmetros supracitados, como estado *commited, invariante, guarda, sincronizações, atualização de variável.* Neste exemplo, o autômato inicia no modelo emissor, que envia um sinal de sincronização *update!* para o modelo receptor, com taxa de envio baseada na função exponencial igual a λ=1/3 (que equivale a 1:3). O modelo receptor recebe o sinal *update?* onde pode permanece no estado intermediário até um tempo igual a 2 unidades de tempo, a guarda permite que a transição só ocorra para um tempo de 2 unidades de tempo, atingindo o estado *commited,* que exige que o autômato saia desse estado imediatamente, enviado desta forma o sinal *end!* para o modelo emissor que retorna ao seu estado inicial, iniciando um novo ciclo.

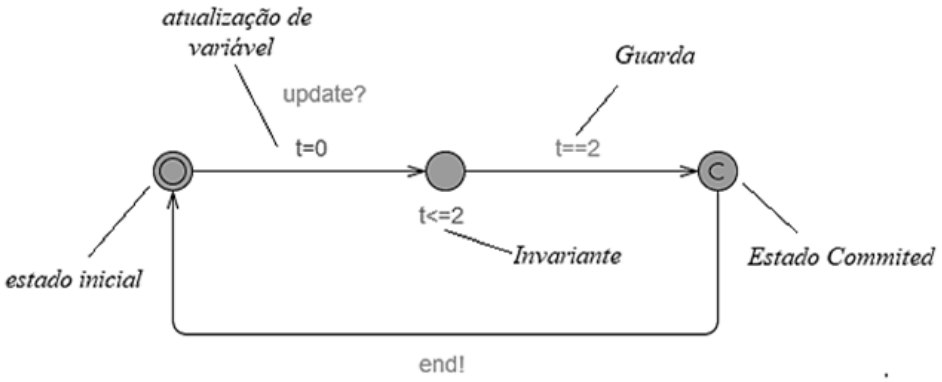
Para realizar a verificação formal do modelo, as especificações são implementadas na guia *verifier* e são denominadas *queries*. As expressões e sintaxe para condição de travamento, propriedades de acessibilidade, vivacidade e segurança podem ser visualizadas no Quadro 2.2.

Quadro 2.2 – *Queries* de verificação formal clássica no UPPAAL.

Propriedade	**Sintaxe**	**Descrição**
Condição de travamento	`A[] not deadlock`	Verifica se em todos os caminhos possíveis inexiste uma condição de travamento.
Acessibilidade	`E<>p`	Verifica se existe um caminho na rede de autômatos que, a partir do estado inicial, satisfaz a fórmula *p*.
Vivacidade	`A<> p`	Verifica se em todos os caminhos, *p* eventualmente pode ocorrer.
	$p --> q$	Verifica se sempre que *p* for satisfeita então *q* será verdade.
Segurança	`A[]p`	Verifica se para todos os caminhos, *p* sempre vai ocorrer.
	`E[]p`	Verifica se existe ao menos um caminho que *p* sempre vai ocorrer.



(a) Modelo Emissor



(b) Modelo Receptor

Figura 2.16 – Exemplo de modelos de rede de autômatos (UPPAAL STRATEGO).

Além dessas especificações, o UPPAAL STRATEGO inclui *queries* para realizar simulações e para verificação formal estatística, em que pode ser definido pelo usuário um grau de confiança e incerteza dos resultados, e o UPPAAL calcula o número de amostras necessárias para análise a fim de atender ao grau de confiança e incerteza solicitados. Estes *queries* podem ser visualizados no Quadro 2.3.

Ressalta-se que na verificação formal clássica é realizada uma varredura automática em todos os caminhos e estados possíveis do modelo, como resultado tem-se uma afirmação de que a propriedade é satisfeita ou não, não havendo incerteza no resultado obtido. Já na verificação foral estatística, devido ao problema de não-decidibilidade ocasionado pelas variáveis contínuas, são utilizados os conceitos de probabilidade para estimar com determinado grau de confiança e erro se a propriedade pode ser satisfeita.

Como exemplo, para um sistema sem variáveis contínuas é possível verificar se o modelo atinge determinado estado utilizando a *query:*

E<> modelo1.localA

Ou seja, existe algum caminho em que o autômato *modelo1* atinja o estado *localA*, se o verificador de modelos indicar que a propriedade foi satisfeita, significa que o modelo pode atingir esse estado com 100% de certeza.

Essa verificação não é possível para sistemas contínuos. Para sanar tal deficiente implantou-se a verificação formal estatística, em que se pode avaliar a seguinte *query*:

Pr[t<=time] (<> modelo1.localA)

Ou seja, qual a probabilidade que em um tempo menor ou igual a *time*, o autômato *modelo1* atinja o estado *localA*. O verificador de modelo apresentará a probabilidade de atingir este estado e o erro associado. Ou seja, não há uma certeza absoluta que a propriedade é satisfeita ou não, apenas uma probabilidade que ela pode ser satisfeita.

Quadro 2.3 – *Queries* de simulação e verificação formal estatística no UPPAAL.

Queries	**Sintaxe**	**Descrição**
Simulação	`Simulate N [<=bound] {E1, …, En}`	Permite a visualização de diversas variáveis (E1, ..., En) para N simulações em um intervalo *bound* de unidades de tempo.
Estimação de Probabilidade	`Pr[bound] (<>p)`	Estima a probabilidade de a rede de autômatos satisfazer *p* em um intervalo de tempo *bound*
Comparação de probabilidade	`Pr[bound](<>p)>=Pr[bound(<>q)`	Verifica se a probabilidade de *p* ocorrer é maior ou igual a probabilidade de *q* ocorrer.
Teste de Hipótese	`Pr[bound](<>p)<=double`	Estima a probabilidade da rede de autômatos que satisfaz *p* em um intervalo de tempo *bound* ser maior ou igual double $\in [0,1]$.

Capítulo 3

Modelagem e Verificação Formal de um Sistema Hidráulico

A proposta deste trabalho refere-se ao emprego da verificação formal para a análise de confiabilidade de sistemas híbridos. Para tanto, neste capítulo serão apresentados: a descrição física e comportamental do sistema hidráulico que é o objeto de estudo desta pesquisa; a metodologia empregada para execução do trabalho; o desenvolvimento e validação dos modelos físico, de controle, de falha e de manutenção do sistema hidráulico.

Os modelos foram desenvolvidos através de uma abstração de autômatos estocásticos híbridos na ferramenta computacional UPPAAL STRATEGO, e somente serão apresentados apenas os modelos finais resultantes da metodologia.

3.1 Descrição do problema de pesquisa

Os sistemas híbridos estão presentes nas mais variadas áreas de utilização, desde o âmbito doméstico até industrial. Dentre esses sistemas, o sistema hidráulico apresenta grande importância, devido sua extensa aplicação e grande gama de características. Sabe-se que o desenvolvimento de projetos de sistemas hidráulicos depende da experiência do projetista. Portanto aprimorar as técnicas de análise desses projetos auxiliam na obtenção de sistemas mais confiáveis e seguros.

Visto que a verificação formal é uma técnica em ascensão, avaliar a sua eficácia para a análise de sistemas hidráulicos é de grande interesse, sendo este o objetivo principal desta pesquisa. Para tanto, neste trabalho optou-se pelo emprego de um sistema hidráulico consagrado para estudo de confiabilidade dinâmica, sendo inclusive apresentado em um *workshop* de análise de confiabilidade dinâmica, realizado em 2004 pela associação italiana 3ASI (*Associazione degli Analisti dell'Ambiente, dell'Affidabilita' e dela Sicurezza Insdustriale)*, com o intuito de comparar várias técnicas de confiabilidade dinâmica existentes (Sakurada, 2013).

O problema usado neste trabalho foi desenvolvido por Aldemir (1987) com o propósito de apresentar um procedimento sistemático para a análise de falhas de sistemas híbridos por meio da análise de Markov assistida por computador.

Diversos autores já trabalharam com esse sistema e obtiveram resultados referentes à característica de falha do mesmo, por exemplo, Marseguerra e Zio (1996) trataram esse

problema utilizando simulação de Monte Carlo, em que diversos cenários foram abordados: 1º) foi considerado que todos os componentes apresentavam a mesma taxa de falha (λ) tanto no estado desligado (*OFF*) como ligado (*ON*); 2º) foi considerado λ diferentes para os estados *OFF* e *ON*; 3º) foi considerado que os componentes poderiam ser reparados, ou seja, passavam por manutenção corretiva; e 4º) foi considerado uma fonte de calor no reservatório, em que a probabilidade de falha de cada componente dependia da temperatura.

Codetta-Raiteri e Bobbio (2006) realizaram o mesmo experimento utilizando Redes de Petri Estocásticas, onde foi simulado o cenário 1 e 3, utilizando duas plataformas das Redes de Petri, a GSPN (*Generalized Stochastic Petri Nets)* e FSPN (*Fluid Stochastic Petri Nets).* Sakurada (2013) utilizou esse problema para validar uma metodologia de análise de confiabilidade dinâmica desenvolvida por ele, denominada de ACoDi, onde simulou o primeiro cenário e comparou seus resultados com aqueles obtidos por Codetta-Raiteri e Bobbio (2006).

A partir do exposto, a fim de verificar se a metodologia proposta, ou seja, a fim de verificar se a modelagem por autômatos estocásticos híbridos e aplicação da verificação formal é adequada para análise de confiabilidade, neste trabalho, será realizada uma análise comparativa dos resultados com aqueles obtidos pelos autores supracitados. Para tanto, todas as informações do sistema e características de falha foram extraídas desses trabalhos, sendo descritas a seguir.

A Figura 3.1 é uma representação ilustrativa do sistema hidráulico objeto de estudo. Não representa nenhum sistema hidráulico real, visto que, é um problema teórico, desenvolvido para análise de técnicas de confiabilidade. Portanto, neste trabalho não será abordada nenhuma aplicação deste sistema, tendo enfoque nos parâmetros de confiabilidade do mesmo.

O sistema hidráulico objeto de estudo é composto por um reservatório contendo fluido, duas bombas (*P1* e *P2*) para encher o reservatório, uma válvula (*V*) para esvaziá-lo e um controlador para monitorar o nível (*H*) do fluido e acionar as bombas e a válvula. Os componentes mecânicos (*P1, P2 e V*) são independentes entre si e podem assumir quatro estados possíveis: 1º) ligado (*ON*); 2º) desligado (*OFF*); 3º) Falha *ON*; e 4º) Falha *OFF*.

Todos os componentes mecânicos apresentam distribuição exponencial de falha, sendo que o tempo médio para a falha (*MTTF*) de cada um deles está descrito na Tabela 3.1. Assume-se que o reservatório e o controlador são perfeitos, ou seja, não apresentam falhas.

Tabela 3.1 - Tempo médio para falhas dos componentes

Componente	**MTTF**
Bomba 1 (P1)	219 h
Bomba 2 (P2)	175 h
Válvula (V)	320 h

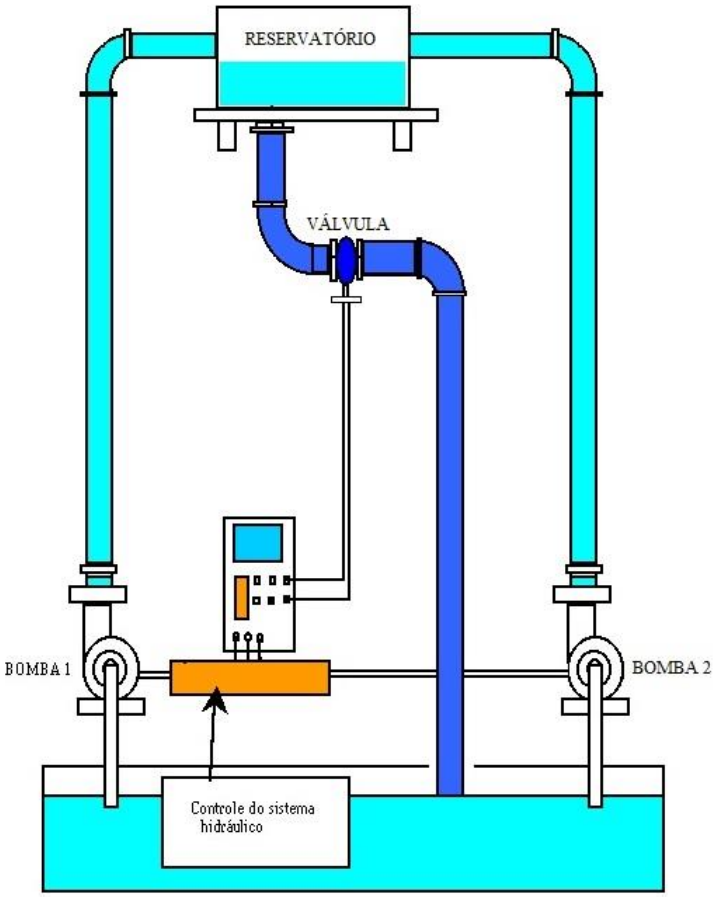


Figura 3.1 – Representação ilustrativa do sistema hidráulico utilizado.

A Figura 3.2 ilustra o esquemático padrão utilizado para o estudo deste sistema, e demostra com mais detalhes os diversos níveis do reservatório. Inicialmente o nível *H*, dado em metros, se encontra em 0, a bomba *P1* ligada, a válvula *V* aberta e a bomba *P2* desligada. A taxa de variação do nível de fluido fornecida pelas bombas é a mesma da válvula, sendo de Y=0,6 m/h. Dessa forma, enquanto nenhum componente entra no estado de falha, o sistema mantém o nível constante em 0.

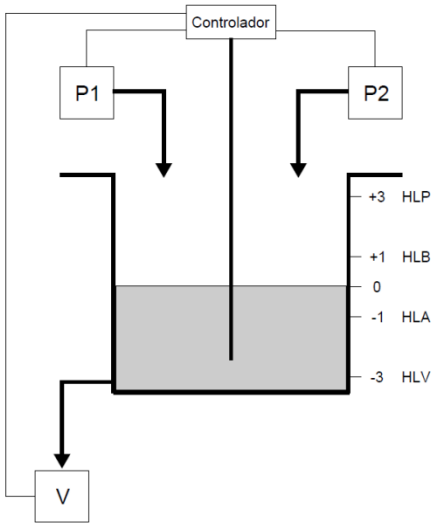


Figura 3.2 - Modelo Padrão (*benchmark*). Fonte: Adaptado de Sakurada (2013).

O nível do reservatório, máximo e o mínimo, respectivamente, é de 3 m e -3 m, se o nível ultrapassar esses limites ocorre a falha do sistema, que fica caracterizada pelos dois cenários: reservatório transbordando e reservatório vazio.

O reservatório possui dois níveis de interesse HLB (+1) e HLA (-1). Esses pontos definem as regiões de controle para a operação do sistema. Se *H* atingir o nível HLB (+1) há o risco de o fluido transbordar pelo reservatório; este evento ocorre quando *H* excede o nível HLP (+3). Para evitar tal condição, o controlador comanda o desligamento das duas bombas e abertura da válvula com o objetivo de reduzir *H*. Se um componente está travado, ele não obedece ao controlador e mantém o seu estado atual.

O outro cenário indesejado é o esvaziamento do reservatório, que ocorre quando *H* está abaixo de HLV (-3). Para evitar isso, quando o nível atinge HLA (-1), o controlador ordena o acionamento das duas bombas e o fechamento da válvula, com o objetivo de aumentar o nível *H*. Portanto, considera-se que o sistema está na região de correto funcionamento quando *H* encontra-se entre HLB e HLA.

O controlador atua no sistema apenas se o nível atingir os pontos HLB e HLA, e segue a lei de controle definida no Quadro 3.1.

Quadro 3.1 – Lei de Controle

Condição	**P1**	**P2**	**V**
$H \leq HLA$	ON	ON	OFF
$HLA < H < HLB$	ON	OFF	ON
$H \geq HLB$	OFF	OFF	ON

3.1.1 Determinação das configurações do sistema

Para a determinação das possíveis configurações do sistema, neste trabalho, foi elaborada a tabela verdade do sistema. Segundo Bollmann (1997), o primeiro passo para o desenvolvimento da tabela verdade é a elaboração da tabela de correspondência, visto que esta apresenta de forma sistemática as variáveis de entrada e de saída, indicando sua descrição, notação e correspondência lógica. A correspondência lógica consiste em determinar qual ação ocorrerá se determinado comando for acionado.

Para o problema proposto, o nível do reservatório é a variável dependente, visto que, haverá variação do nível dependendo do estado de operação dos demais componentes. No Quadro 3.2 é apresentada a tabela de correspondência para o sistema em estudo. As bombas e a válvula são as variáveis de entradas e para variáveis de saída foram considerados os dois estados que o nível pode assumir: transbordamento ou esvaziamento.

Quadro 3.2 - Tabela de Correspondência para o problema proposto

Variáveis de Entrada	**Notação**	**Correspondência Lógica**
Bomba 1	*P1*	Bomba 1 Ligada *P1* = 1
Bomba 2	*P2*	Bomba 2 Ligada *P2* = 1
Válvula	*V*	Válvula aberta *V* = 1
Variáveis de Saída		
Transbordamento do Reservatório	*S1*	Nível *H* aumentando *S1* = 1
Esvaziamento do Reservatório	*S2*	Nível *H* diminuindo *S2* = 1

A partir da tabela de correspondência pode se observar que este é um problema de comando combinatório, com três entradas e duas saídas, ou seja, a combinação das variáveis de entrada determina qual será a saída do sistema.

Segundo Bollmann (1997), a tabela verdade representa a função lógica em forma tabular e deve conter todas a combinações possíveis de todas as variáveis de entradas e quais saídas resultam dessas combinações. "A tabela verdade possui 2^n linhas para representar as combinações possíveis e tantas colunas quantas forem as variáveis de entrada e saída" (Bollmann, 1997). Onde *n* é o número de variáveis de entrada, logo, para este sistema n=3, portanto, a tabela verdade será composto por 8 linhas e 5 colunas, e pode ser visualizada no Quadro 3.3.

Quadro 3.3 – Tabela verdade para o problema proposto.

Linha	**V**	**P2**	**P1**	**S1**	**S2**
00	0	0	0	0	0
01	0	0	1	1	0
02	0	1	0	1	0
03	0	1	1	1	0
04	1	0	0	0	1
05	1	0	1	0	0
06	1	1	0	0	0
07	1	1	1	1	0

A partir da tabela verdade foi possível levantar todas as configurações que o sistema pode assumir, e qual a sua influência na taxa de variação do nível do reservatório. A tabela verdade está de acordo com os valores apresentados por Sakurada (2013). A Tabela 3.2 mostra a correspondência entre a tabela verdade e a taxa de variação do nível H.

Ainda, na Tabela 3.2 é possível observar que o nível do reservatório não se altera se uma das bombas e a válvula estiverem ligadas, o que corresponde as configurações 3 e 8. Também não haverá alteração se todos os componentes estiverem desligados, configuração 5.

Apenas uma configuração possibilita o esvaziamento do reservatório, 7, onde ambas as bombas estão desligadas e a válvula aberta.

São quatro configurações que provocam o transbordamento do reservatório, 1, 2, 4 e 6. Sendo que o caso mais crítico é aquele em que ambas as bombas estão ligadas enquanto a válvula está fechada.

A partir da determinação das configurações, é possível concluir que poderão ocorrer mais eventos de transbordamento do que de esvaziamento, visto que há uma variedade maior de condições para ocorrência da primeira.

Tabela 3.2 - Taxa de variação do nível H.

Configuração	Linha (Quadro 3.3)	P1	P2	V	dH/dt	Nível
1	01	ON	OFF	OFF	0,6 m/h	Aumentando
2	03	ON	ON	OFF	1,2 m/h	Aumentando (+ crítica)
3	05	ON	OFF	ON	0,0 m/h	Constante
4	07	ON	ON	ON	0,6 m/h	Aumentando
5	00	OFF	OFF	OFF	0,0 m/h	Constante
6	02	OFF	ON	OFF	0,6 m/h	Aumentando
7	04	OFF	OFF	ON	- 0,6 m/h	Esvaziando
8	06	OFF	ON	ON	0,0 m/h	Constante

3.2 Metodologia para validação dos modelos

Para o desenvolvimento do modelo em autômatos estocásticos híbridos do sistema hidráulico, apresentado na seção 3.1, para a obtenção dos parâmetros de confiabilidade do mesmo, quatro modelos serão construídos. A Figura 3.3 ilustra tais modelos e qual comportamento será avaliado em cada um deles. Após a simulação e a verificação formal de um modelo, o mesmo é adaptado para a criação do próximo, sendo que os autômatos não são alterados de um modelo para o outro, e, portanto, a verificação das propriedades do modelo anterior se mantém válidas.

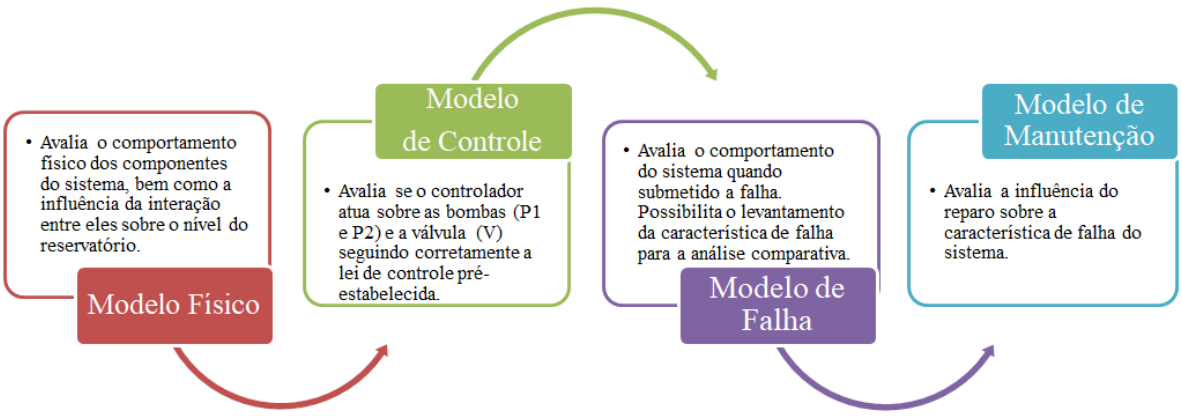


Figura 3.3 – Metodologia para validação dos modelos. Fonte: Adaptado de Kunz (2012).

A metodologia empregada para a criação e validação dos modelos foi descrita na seção 2.3 e será esquematizada de forma sucinta a seguir (Ver Figura 3.4):

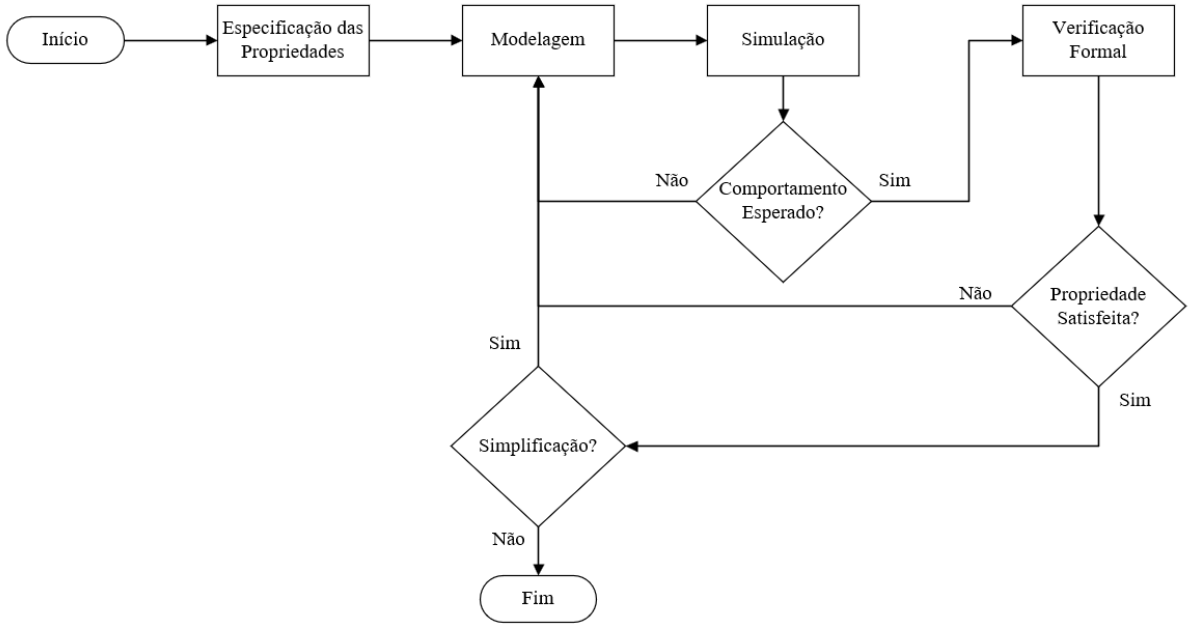


Figura 3.4 – Metodologia para validação dos modelos. Fonte: Adaptado de Kunz (2012).

1. **Especificação das propriedades:** declaração das propriedades que o sistema proposto deve satisfazer, como por exemplo, o controlador deve atuar caso o nível atinja HLB. A partir do que se pretende verificar, os modelos devem ser criados para contemplar tal comportamento, justificando que esta etapa ser anterior à etapa de modelagem.
2. **Modelagem:** Criação dos modelos do sistema em autômatos estocásticos híbridos, sendo eles, o modelo físico, de controle, de falha e de manutenção.
3. **Simulação:** A fim de melhorar a qualidade do modelo e avaliar se o comportamento do mesmo está coerente, uma simulação antes da verificação do modelo é realizada.
4. **Verificação Formal:** o verificador checa a validade da propriedade no modelo do sistema, no caso de não atender as especificações, é necessário retornar à etapa de modelagem.
5. **Simplificação:** algumas simplificações podem ser realizadas, uma vez que o modelo deve descrever satisfatoriamente como o sistema se comporta, e não o seu detalhamento estrutural. Esta etapa possibilita diminuição do esforço computacional, como o tempo de processamento.

Para desenvolvimento dos modelos de acordo com a metodologia supracitada adotou-se a ferramenta computacional UPPAAL STRATEGO, a qual possui os módulos *editor*, *simulate* e *verifier*, que permitem a execução das etapas de modelagem, simulação e verificação formal dos modelos, respectivamente.

3.3 Modelagem do Sistema Hidráulico

Com base no que foi explanado seção 2.3.1, a modelagem do sistema foi realizada por abstração de autômatos estocásticos híbridos. Segundo De Negri (2004), a abstração pode ser entendida como uma descrição simplificadora que enfatiza certas características do sistema e suprime outras. Resultado desta abstração é a obtenção de modelos sob determinadas perspectivas, podendo ser estrutural, funcional e comportamental.

A perspectiva estrutural permite obter um modelo para descrever o sistema fisicamente, considerando seus componentes e as relações entre eles, estabelece as dimensões e propriedades estruturais do mesmo (Salas, 2014).

A perspectiva funcional analisa o sistema sob o ponto de vista da função de cada componente e a inter-relação entre eles, que resulta na função global do sistema (De Negri, 2004). Por exemplo, a bomba quando ligada tem a função de adicionar fluido ao reservatório, já a válvula tem a função de retirar fluido do reservatório, e a interação entre os dois define a função global do sistema que é a variação do nível do reservatório.

Por fim, De Negri (2004) estabelece como perspectiva comportamental, a avaliação do efeito causado sobre o meio externo das diferentes funções cumpridas pelos componentes do sistema. Esta perspectiva propõe analisar o sistema como um todo, ou seja, estabelece quando as diferentes funções devem ser executadas para a obtenção do comportamento correto do sistema. A junção destes três tipos de perspectiva permite fornecer a descrição completa do sistema.

Com base nestas perspectivas, foi realizada a modelagem a partir de quatro modelos: físico, de controle, de falha e de manutenção. O modelo físico apresenta as características do sistema a partir da perspectiva funcional, em que os componentes mecânicos do sistema possuem liberdade para alterar seu estado de forma randômica e não há atuação do controlador sobre o sistema. Desta forma, é possível avaliar se o modelo permite que cada componente mecânico exerça sua função e qual a influência da interação desses componentes sobre o nível do reservatório.

O modelo de controle apresenta uma perspectiva comportamental, em que a partir da inserção do controlador ao modelo físico, é possível avaliar se o modelo exerce todas as funções necessárias para atingir determinado comportamento. Por exemplo, caso o nível do reservatório esteja fora da região de correto funcionamento, o modelo deve ser capaz de monitorar e executar as funções necessárias para que o nível retorne a esta região.

Após a aplicação da metodologia apresentada para os modelos: físico e de controle, o modelo de falha e de manutenção foram desenvolvidos, sendo possível avaliar o comportamento do sistema quando submetido à falha e quando os componentes são reparáveis. Porém, o propósito destes modelos é o levantamento da característica de falha do

sistema que serão comparados com os resultados disponíveis nas literaturas citadas na seção 3.1.

A análise comparativa dos resultados permite validar o modelo construído para o sistema hidráulico em estudo e verificar se metodologia empregada, apresentada na seção 3.2, é adequada para análise de confiabilidade de sistemas híbridos. Caso se obtenha resultados satisfatórios, será possível concluir que a partir de modelagem por autômatos estocásticos híbridos e verificação formal de modelos é possível predizer os parâmetros de confiabilidade e disponibilidade de um sistema híbrido, sendo uma contribuição original.

3.3.1 Modelo Físico

O primeiro modelo construído foi o modelo físico, em que se busca avaliar se os componentes do sistema têm liberdade para mudar de estado, ou seja, se as bombas P1 e P2 e a válvula V podem ligar e desligar.

Este modelo também visa avaliar estes componentes cumprem sua função requerida e para as diferentes combinações de estados dos mesmos, o sistema se comporta de forma esperada, atingindo os níveis do reservatório adequados de acordo com a taxa de variação do nível resultante da combinação do estado de operação de cada componente.

Etapa 1: Especificação das propriedades

Para o modelo em questão as propriedades a serem satisfeitas são:

1. Ausência de *deadlock:* O modelo desenvolvido não pode travar, ou seja, não deve haver estados que impossibilitem a transição do sistema.
2. Acessibilidade:
    a. Deve haver pelo menos um caminho que permita que os componentes mecânicos do sistema (*P1*, *P2* e *V*) passem da condição fechada para aberta, ou, vice-versa.
    b. Deve haver caminhos que permitam que todas as configurações do sistema (Tabela 3.2) possam ocorrer.
    c. O modelo deve permitir alcançar todos os níveis do reservatório (*H*) (Figura 3.2).
    d. Deve haver coerência entre os estados P1, P2 e V e o nível H, ou seja, sempre que o nível do reservatório (*H*) atinge um determinado valor numérico, devido a configuração do sistema, o mesmo deve alcançar determinado estado de acordo com a Figura 3.2.
3. Segurança:
    a. Quando o nível do reservatório (*H*) atinge um determinado valor numérico, devido a configuração de *P1*, *P2* e *V*, o sistema não pode entrar em determinados estados do modelo do reservatório.

A Tabela 3.3 ilustra a lei de transição dos estados do reservatório em relação a variável *H*, dada em cm, sendo utilizada para a verificação das propriedades de acessibilidade, item (d), e segurança, item (a).

Tabela 3.3 – Lei de transição dos estados do reservatório

Estado	**H (cm)**
HLP	$H \geq 300$
HLB	$100 \leq H < 300$
LP0	$-100 < H < 100$
HLA	$-300 < H \leq -100$
HLV	$H \leq -300$

Etapa 2: Modelagem

A primeira consideração realizada refere-se à taxa de variação do nível do reservatório, estipulada em Y = 0,6 m/h. Uma vez que está é uma variável contínua, ela introduz ao modelo um problema de não-decidibilidade à rede de autômatos e impede a representação do comportamento do sistema através de um número finito de estados. A fim de eliminar tal incerteza, a partir da conversão de unidades é possível representar esta quantidade em cm/min, desta forma discretizar a variável tornando-a um valor inteiro Y = 1 cm/min.

Entretanto, o passo de tempo passa a ser de 1 min, acarretando grande esforço computacional para simulação e verificação dos modelos, portanto optou-se por estabelecer uma unidade de tempo (u.t.) de 20 min, e representar a taxa de variação como Y = 20 cm/u.t., ou seja, a cada 20 min (equivalente a 1 unidade de tempo) à uma alteração do nível do reservatório de 20 cm. Desta forma, a variável permanece com um valor inteiro e aumenta-se o passo de tempo, diminuindo tempo de processamento. Desta maneira, é possível o desenvolvimento do modelo para verificação das propriedades postuladas na primeira etapa. O modelo físico é composto por 6 modelos inter-relacionados:

1. *Tank*: Representa o sistema físico do reservatório e é composto por 5 estados: LP0, HLB, HLP, HLA e HLV. Este modelo recebe o comando de atualização (*update)* do modelo *Level* e de acordo com valor da variável *H* ocorre a transição dos estados (Figura 3.5).

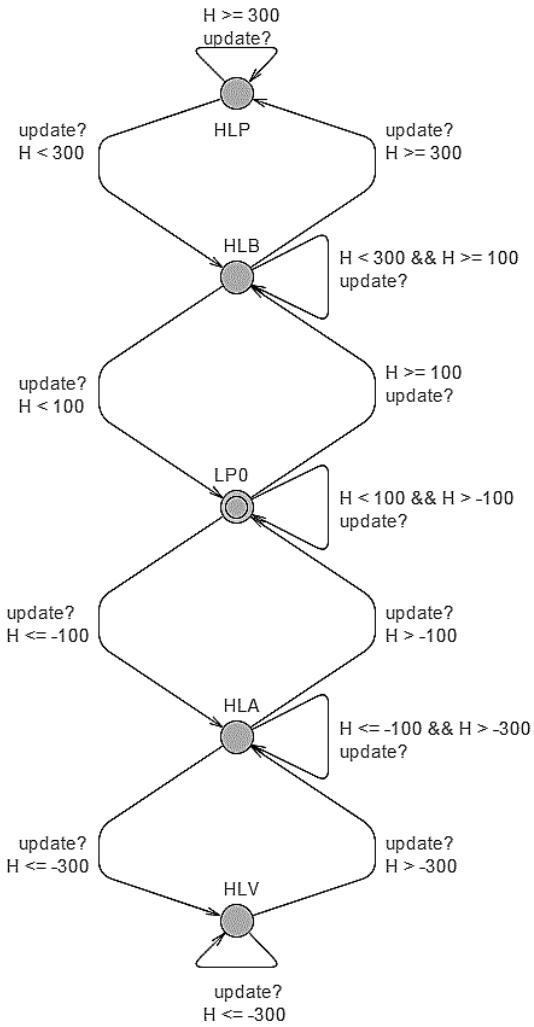


Figura 3.5 – Autômato *Tank* utilizado nos modelos físico, de controle e de falha.

2. *Random:* Este modelo produz comandos para mudança de estado das bombas e da válvula de forma aleatória, visando testar se o modelo permite a abertura e fechamento de tais componentes e verificar as diferentes configurações resultantes da inter-relação entre eles. Este modelo recebe um comando de inicialização do modelo *Level* a cada 1 u.t., e aleatoriamente, por meio do campo *select*, escolhe o componente (*P1*, *P2* e *V*) e o tipo de comando (abertura ou fechamento) que enviará para o mesmo (Figura 3.6).

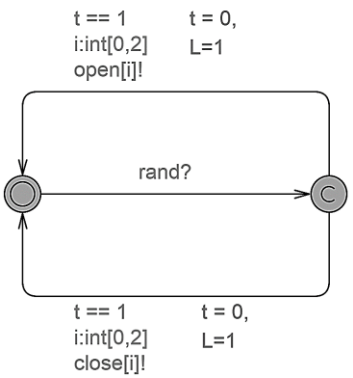


Figura 3.6 – Autômato *Random* do modelo físico.

3. *Level*: Este modelo atualiza o valor do nível do reservatório e envia um comando de atualização (*update)* para o reservatório caso a soma da combinação entre *P1*, *P2* e V não resulte em zero (Figura 3.7).

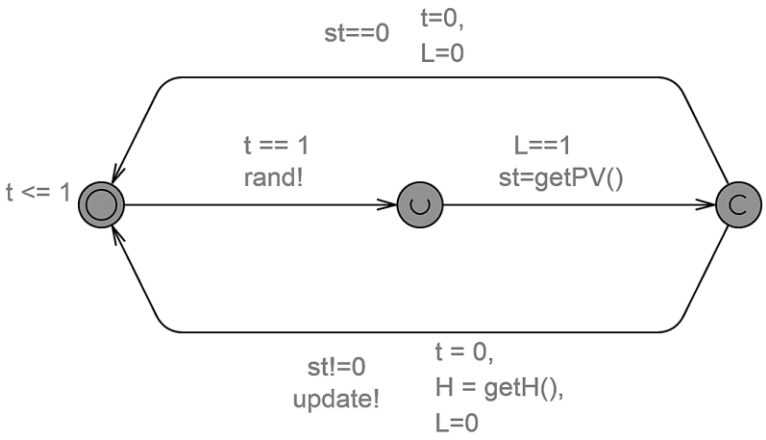


Figura 3.7 – Autômato *Level* do modelo físico

A cada 1 u.t., este modelo envia um comando de inicialização para o modelo *Random*, que executa suas tarefas e atualiza o valor da variável *L*. Só depois de atualizado o valor de *L*, o modelo *Level* calcula a soma das variáveis *P1*, *P2* e *V*, através da função *getPV()*. Enfatiza-se que a variável *V* é subtraída, uma vez que provoca diminuição do nível. Caso a soma seja igual a 0 o modelo retorna ao ponto inicial. Se a soma for diferente de 0 executa a função *getH()* e envia o comando de atualização (*update)* para o reservatório. A implementação dessas funções pode ser visualizada na Figura 3.8 e a Tabela 3.4 ilustra a influência da soma destas variáveis sobre o nível *H*.

```
//Função getPV(): faz a soma das configurações dos elementos que serve de
//           entrada para o cálculo do nivel do reservatório

int getPV() {
  int soma = 0;
  soma = P1 + P2 - V;
  return soma;
}

//Função getH(): calcula a taxa de de nivel (a cada 1 min o nivel varia 1 cm)
int getH() {
  int val = 0;
  if ( H <= 300 && H >= -300) {val = H + (taxa*getPV());}
  if ( val > 300)               {val = 300;}
  if ( val < -300)              {val = -300;}
  return val;
}
```

Figura 3.8 – Implementação das funções *getPV()* e *getH()*

Tabela 3.4 – Influência de *P1*, *P2* e *V* sobre *H*

P1	P2	V	Soma	ΔH
1	0	0	1	+1 cm
1	1	0	2	+2 cm
1	0	1	0	0 cm
1	1	1	1	+1 cm
0	0	0	0	0 cm
0	1	0	1	+1 cm
0	0	1	-1	-1 cm
0	1	1	0	0 cm

4. *Pump1*: Representa o comportamento físico da bomba 1, e possui 2 estados (*OFF* e *ON*). O estado inicial de *P1* é na posição ligada (*ON*), sempre que houver um comando de abertura (*open[0]*) ou fechamento (*close[0]*), onde o índice [0] representa a bomba 1 na rede de autômatos, ocorre a transição do seu estado e a atualização do estado de operação de *P1* (Figura 3.9).

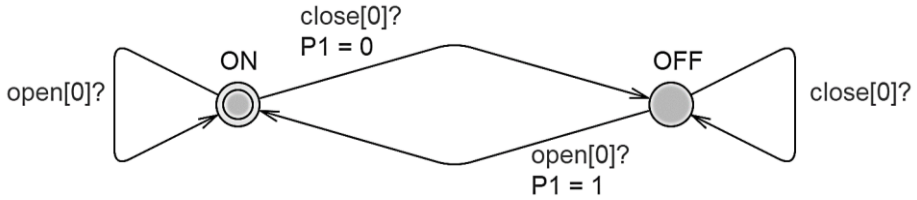


Figura 3.9 – Autômato *Pump1* utilizado em todos os modelos.

5. *Pump2*: Representa o comportamento físico da bomba 2. Assim como *Pump1* possui 2 estados (*OFF* e *ON*), porém seu estado inicial é na posição desligada (*OFF*), e os comandos de abertura (*open[1]*) e fechamento (*close[1]*), onde o índice [1] representa a bomba 2 na rede de autômatos, provocam a transição do seu estado e a atualização do estado de operação de *P2* (Figura 3.10).

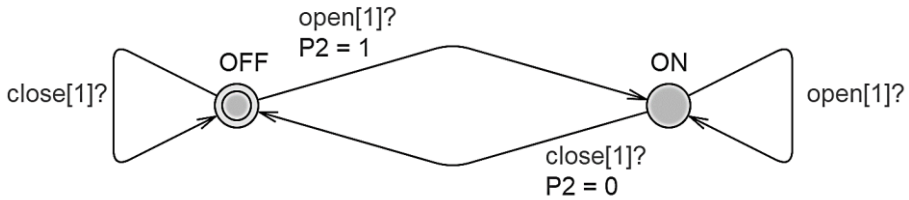


Figura 3.10 – Autômato *Pump2* utilizado em todos os modelos.

6. *Valve*: Representa o comportamento físico da válvula. Assim como *Pump1*, possui 2 estados (*OFF* e *ON*), seu estado inicial é na posição ligada (*ON*), e os comandos de abertura (*open[2]*) e fechamento (*close[2]*), onde o índice [2] representa a válvula

na rede de autômatos, provocam a transição do seu estado e a atualização do estado de operação de *V* (Figura 3.11).

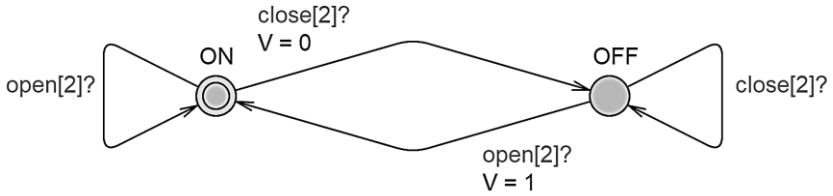


Figura 3.11 – Autômato *Valve* utilizado em todos os modelos.

Etapa 3: Simulação

De forma a avaliar o comportamento do modelo físico, do ponto de vista da simulação, foi utilizada a *query:*

simulate 1 [<=10] {H, P1, P2, V}

A simulação foi realizada para um tempo de missão de 10 u.t., em que foram extraídas informações sobre o nível do reservatório (*H*) e a condição de operação dos componentes mecânicos (*P1*, *P2* e *V*).

O tempo de missão escolhido para esta simulação foi determinado levando em conta os parâmetros que se desejava avaliar, que seria a taxa de variação do nível *H* em decorrência da mudança do estado de operação dos componentes mecânicos. Foi verificado que para um tempo de 10 u.t. foi possível verificar a taxa de variação do nível em todas as configurações possível do sistema, portanto considerou-se este tempo adequado para tal simulação.

Diversas simulações envolvendo a rede de autômatos do modelo físico foram analisadas. Nas Figura 3.12 e Figura 3.13 é apresentada uma destas simulações, onde se observar o comportamento do sistema em todas as combinações possíveis entre *P1, P2* e *V*.

Nas Figura 3.12 e Figura 3.13 é possível observar que quando a soma (S = P1+P2-V) resulta em 0, o nível permanece inalterado, como ocorre nos tempos: 0, 1, 4, e 6 u.t.. Quando a soma resulta em -1, ocorre diminuição do nível (-20 cm), como pode ser visto nos tempos: 2 e 3 u.t.. Quando a soma é igual a 1, o nível aumenta em 20 cm, isto ocorre nos tempos: 5, 7, 9 e 10 u.t. e, por fim, quando a soma é igual a 2, o nível aumenta 40 cm, fato ocorrido no t = 8 u.t., ou seja, em princípio o modelo físico se comporta conforme o esperado.

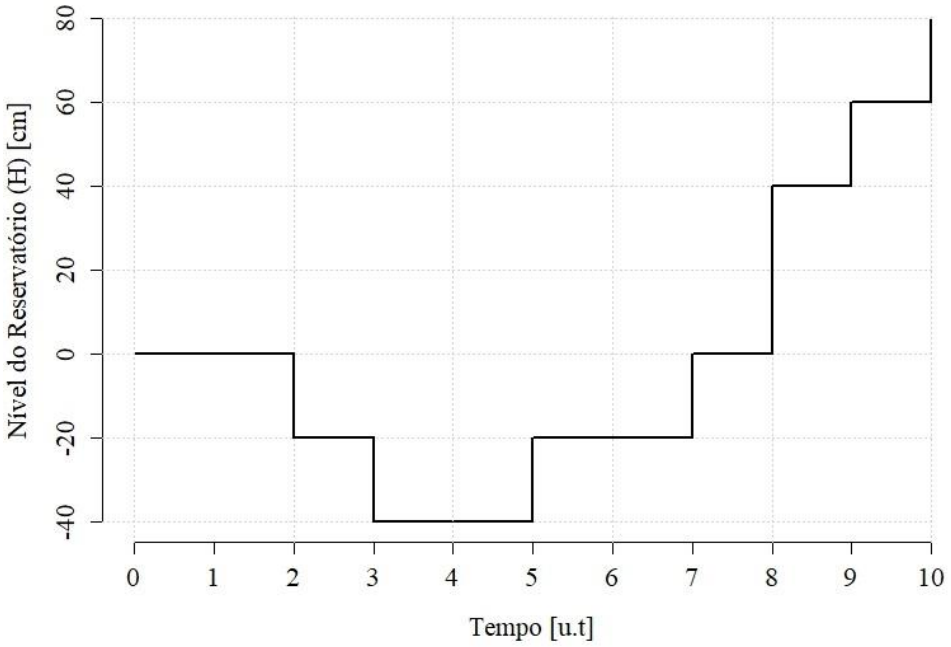


Figura 3.12 – Simulação da taxa de variação do nível do reservatório para o modelo físico.

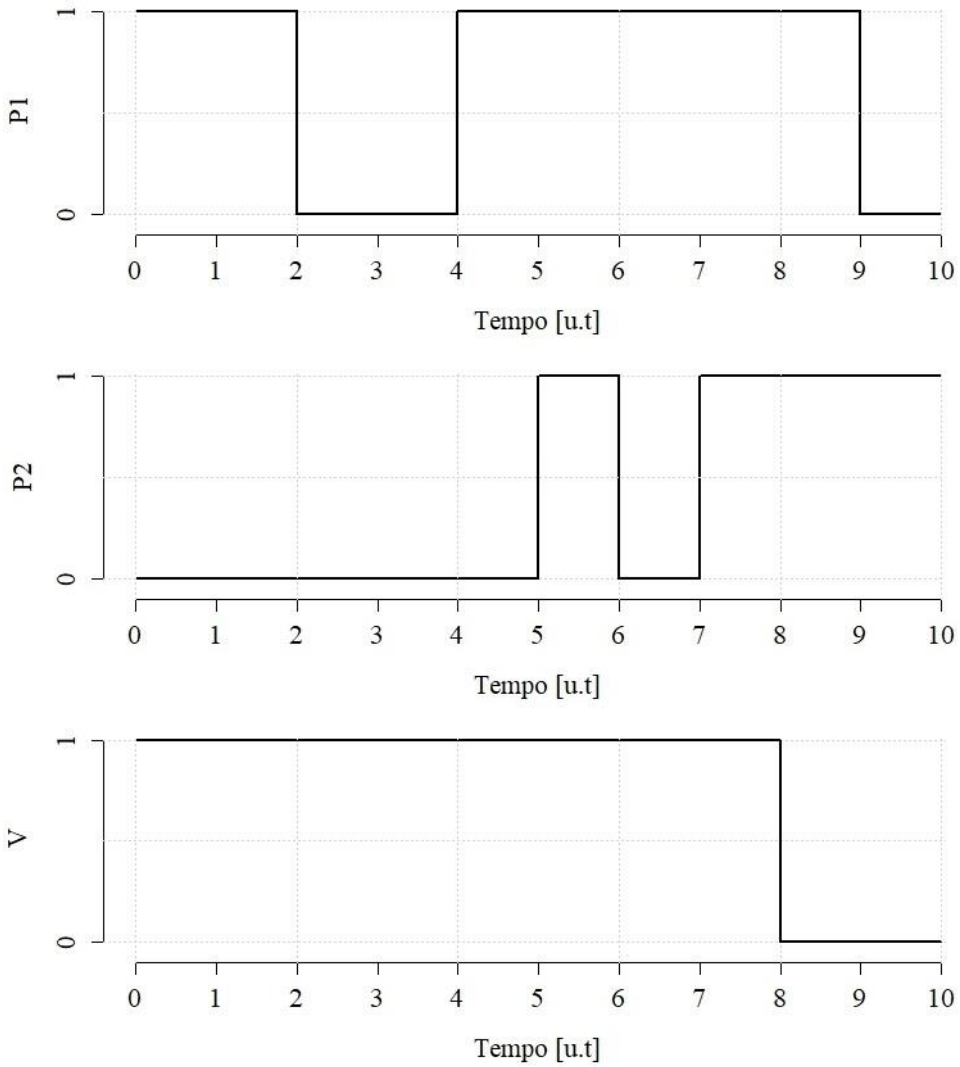


Figura 3.13 – Simulação da condição de operação de P1, P2 e V para o modelo físico.

Etapa 4: Verificação Formal

No Quadro 3.4 estão apresentadas as verificações realizadas sobre o modelo físico, de acordo com a etapa de especificação de propriedades, em que se constata que o modelo atende todos os requisitos a ele impostos.

Quadro 3.4 – Verificação formal para o modelo físico.

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita?**
Verifica a ausência de *deadlock*	`A[] not deadlock`	Sim
Acessibilidade		
Deve existir pelo menos um caminho que possibilite a abertura ou fechamento das bombas e da válvula.	`E<> pump1.ON`	Sim
	`E<> pump1.OFF`	
	`E<> pump2.ON`	
	`E<> pump2.OFF`	
	`E<> valve.ON`	
	`E<> valve.OFF`	
Deve existir pelo menos um caminho que resulte em cada uma das configurações do sistema, especificadas no Tabela 3.2.	`E<> pump1.ON and pump2.OFF and valve.OFF`	Sim
	`E<> pump1.ON and pump2.ON and valve.OFF`	
	`E<> pump1.ON and pump2.OFF and valve.ON`	
	`E<> pump1.ON and pump2.ON and valve.ON`	
	`E<> pump1.OFF and pump2.OFF and valve.OFF`	
	`E<> pump1.OFF and pump2.ON and valve.OFF`	
	`E<> pump1.OFF and pump2.OFF and valve.ON`	
	`E<> pump1.OFF and pump2.ON and valve.ON`	
Deve existir pelo menos um caminho em que o reservatório atinja cada um dos níveis ilustrados na Figura 3.2.	`E<> tank.HLP`	Sim
	`E<> tank.HLB`	
	`E<> tank.LP0`	
	`E<> tank.HLA`	
	`E<> tank.HLV`	
Quando H for 0, o reservatório pode estar no nível LP0.	`E<> H==0 and tank.LP0`	Sim

Continuação Quadro 3.4 – Verificação formal para o modelo físico.

Descrição Informal	Descrição Formal	Propriedade Satisfeita?
Quando H for igual a 100 cm, o reservatório pode estar no nível LP0 ou HLB.	`E<> H==100 and (tank.LP0 || tank.HLB)`	Sim
Quando H for igual a 300 cm, o reservatório pode estar no nível HLB ou HLV.	`E<> H==300 and (tank.HLP || tank.HLB)`	Sim
Quando H for igual a -100 cm, o reservatório pode estar no nível LP0 ou HLA.	`E<> H==-100 and (tank.LP0 || tank.HLA)`	Sim
Quando H for igual a -300 cm, o reservatório pode estar no nível HLA ou HLV.	`E<> H==-300 and (tank.HLA || tank.HLV)`	Sim
Segurança		
Quando H for 0, o reservatório não pode estar nos níveis HLB, HLP, HLA e HLV.	`E[] H==0 and (tank.HLB || tank.HLP || tank.HLA || tank.HLV)`	Sim
Quando H for igual a 100 cm, o reservatório não pode estar nos níveis HLP, HLA e HLV.	`E[] H==100 and (tank.HLP || tank.HLA || tank.HLV)`	Sim
Quando H for igual a 300 cm, o reservatório não pode estar nos níveis LP0, HLA e HLV.	`E[] H==300 and (tank.LP0 || tank.HLA || tank.HLV)`	Sim
Quando H for igual a -100 cm, o reservatório não pode estar nos níveis HLB, HLP e HLV.	`E[] H==-100 and (tank.HLP || tank.HLB || tank.HLV)`	Sim
Quando H for igual a -300 cm, o reservatório não pode estar nos níveis LP0, HLB e HLP.	`E[] H==-300 and (tank.LP0 || tank.HLB || tank.HLP)`	Sim

3.3.2 Modelo de Controle

Após validação do modelo físico, este modelo foi ampliado para inclusão do controlador, resultando no modelo de controle. Este modelo propõe testar se o controlador atua sobre as bombas e a válvula seguindo corretamente a lei estabelecida no Quadro 3.1 (Página 38), e se a partir dessa atuação o sistema retorna à região de correto funcionamento.

Etapa 1: Especificação das propriedades

Neste modelo as seguintes propriedades devem ser satisfeitas:

1. Ausência de *deadlock:* O modelo desenvolvido não pode travar, ou seja, não deve haver estados que impossibilitem a transição do sistema.

2. Acessibilidade:
    a. Sempre que o nível do reservatório (*H*) ultrapassar os limites da região de correto funcionamento, o controlador deve atuar de acordo com a lei estabelecida no Quadro 3.1.
    b. Sempre que o nível do reservatório (*H*) atingir determinado valor numérico, o controlador deve entrar no estado referente a região compatível com este valor.
3. Segurança:
    a. O controlador não pode admitir configurações diferentes daquelas definidas pela lei de controle, quando as variáveis *F1, F2* e *F3* forem iguais a 0. Para o modelo de controle, as variáveis *F1, F2* e *F3*, não tem significado físico, porém no modelo de falha e de manutenção estas variáveis quando assumem o valor de 1, indicam a falha do respectivo componente mecânico. Sendo necessário inseri-las já neste modelo, para fazer a verificação formal do modelo do controlador e mantê-lo valido para todos os modelos subsequentes.
    b. Quando o nível do reservatório (*H*) atingir determinado valor numérico, o controlador não pode entrar nos estados que se referem as regiões não compatíveis com este valor.
    c. Quando *F1, F2* e/ou *F3* forem iguais a 1, o controlador não pode atuar sobre o respectivo componente mecânico, enviado comandos de abertura ou fechamento, sendo que *F1* refere-se a *P1, F2* a *P2* e *F3* a *V*.

O Quadro 3.5 ilustra as regiões de controle, em que de acordo com o valor de *H* o sistema deve atingir tal região e atuar sobre os componentes mecânicos de acordo com a lei de ação apresentada. As propriedades de acessibilidade, item (b), e de segurança, item (b), são especificadas conforme o estabelecido neste quadro.

Quadro 3.5 – Regiões de Controle

Estado	**Definição**	**H**	**Lei de Ação**
R1	Esvaziamento	$H \leq -100$ cm	Ligar P1 e P2 e desligar V.
R2	Correto Funcionamento	$-100 \leq H \leq 100$ cm	Ligar P1 e V e desligar P2.
R3	Transbordamento	$H \geq 100$ cm	Desligar P1 e P2 e ligar V.

Etapa 2: Modelagem

O modelo de controle é composto por 7 modelos inter-relacionados, sendo que os modelos *Pump1*, *Pump2*, *Valve* e *Tank*, não sofreram nenhuma modificação do modelo físico para este, os demais modelos são descritos a seguir:

1. *Random2:* A fim de melhor visualizar a influência do controlador sobre o modelo, o modelo *Random* foi substituído pelo *Random2*, em que ao invés de alterar os estados dos componentes mecânicos, altera-se o valor da variável *H* (Figura 3.14).

A cada 15 u.t., este modelo altera o valor de *H* randomicamente - desta forma, o nível pode assumir qualquer valor entre -300 cm e 300 cm, que são os valores limites do reservatório -, envia um comando de atualização (*run*) tanto para o controlador como para o *Level*. O modelo do controlador avalia se a partir do valor atribuído a *H* o reservatório ultrapassa os limites da região de correto funcionamento, e se necessário atua para retornar a esta região. O tempo de 15 u.t. foi escolhido para garantir que após o sistema receber determinado valor de *H* o modelo tenha tempo suficiente para retornar o sistema dentro dos limites de *R2*, sem que uma nova atualização de *H* ocorra, facilitando na visualização dos resultados da simulação. Neste modelo também é realizada a atualização das variáveis *F1*, *F2* e *F3* de forma aleatória, podendo assumir o valor de 0 ou 1.

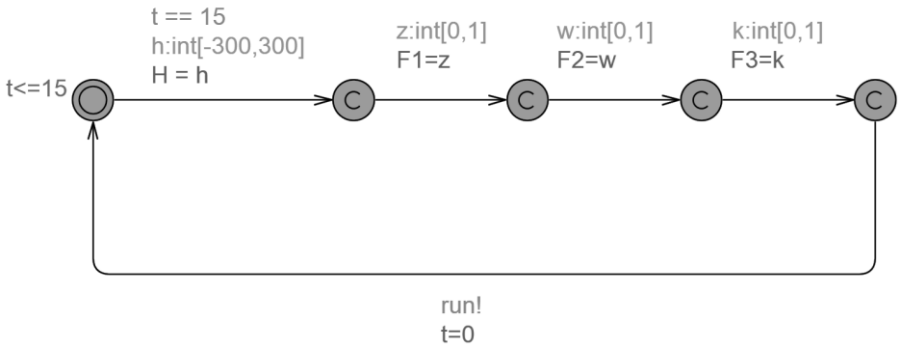


Figura 3.14 – Autômato *Random2* do modelo de controle

2. *Level:* Este modelo mantém a mesma função, que é a atualização do valor do nível do reservatório. Porém, algumas modificações foram realizadas para adequá-lo ao modelo de controle (Figura 3.15).

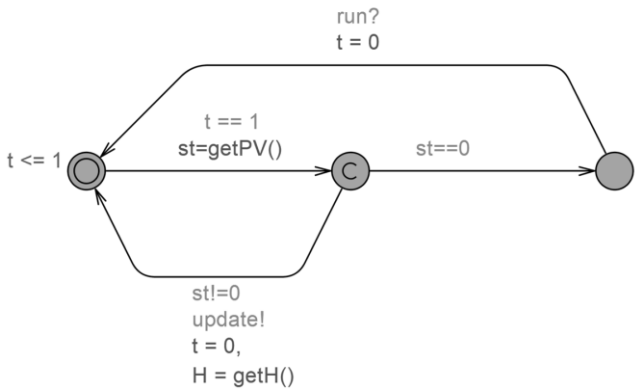


Figura 3.15 – Autômato *Level* do modelo de controle.

Inicialmente, visto que o modelo *Random* não foi utilizado no modelo de controle, foi retirada a variável *L* que relacionava estes dois modelos. Na sequência, com o intuito de diminuir esforço computacional, para que o sistema não fosse atualizado a cada 1u.t., mesmo sem alteração do nível, criou-se um novo estado em que o modelo só executa a transição para atualização do nível quando ocorre uma alteração do sistema. No caso do modelo de controle, a alteração só ocorre pela mudança do valor de *H* pelo modelo *Random2*, e quando isto ocorre o modelo *Level* recebe um

comando de atualização para iniciar monitoração do nível do reservatório (*run)* e o envio do comando de atualização (*update)* para o controlador e reservatório.

3. *Controller:* Tem como função, a cada atualização do nível do reservatório (*H*), verificar se o mesmo ultrapassa os limites da região de correto funcionamento (*R2*), caso isso ocorra, o mesmo deve atuar sobre os componentes mecânicos, enviando comandos de abertura e fechamento de acordo com a lei de controle pré-estabelecida. Retornando a região *R2*, o controlador atualiza a condição de operação de *P1*, *P2* e *V* para a condição inicial (Figura 3.16).

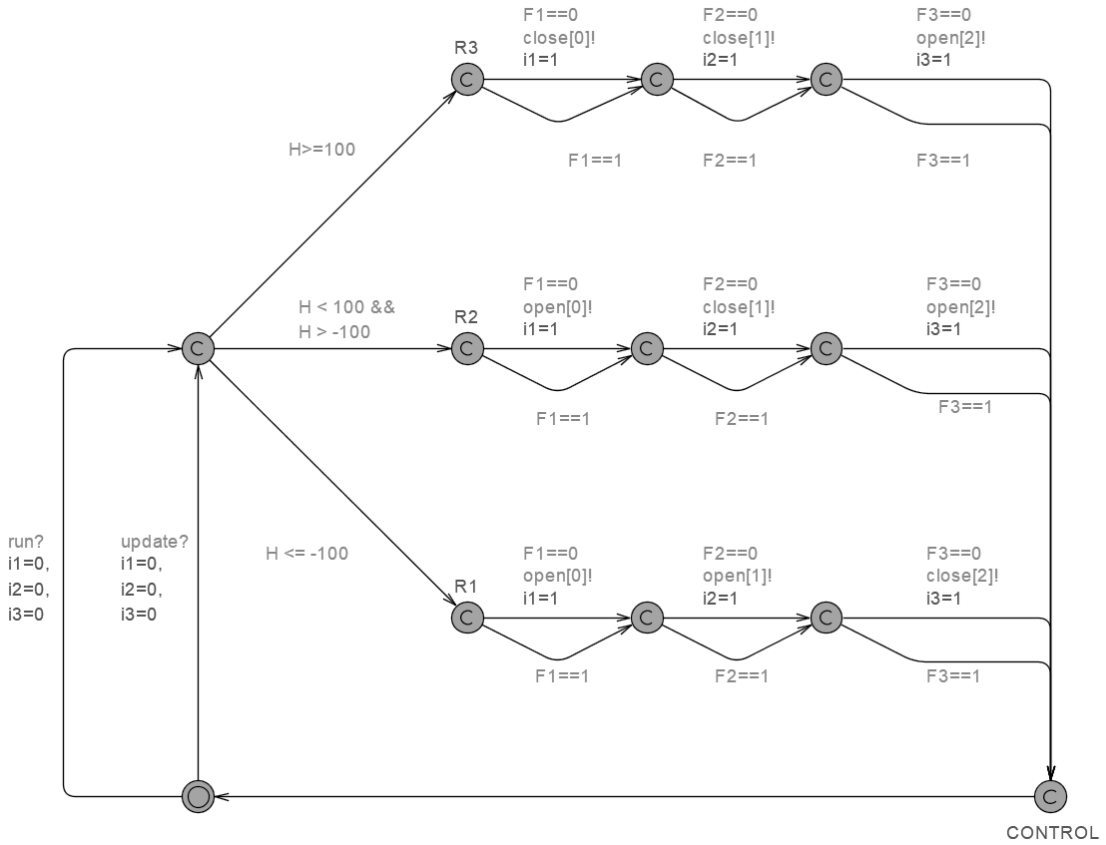


Figura 3.16 – Autômato *Controller* do modelo de controle e de falha.

Este modelo recebe comando de inicialização tanto do modelo *Level* como *Random2*. De acordo com o valor de *H*, ele atinge o estado *R1*, *R2* ou *R3*, descritos no Quadro 3.5 (Página 52). Neste modelo foram inseridas as variáveis *F1*, *F2* e *F3* que, como já mencionado, para o modelo de controle não tem significado físico, porém serão utilizadas no modelo de falha para indicar quando um componente mecânico está em falha, condição esta que impede que o controlador atue sobre o respectivo componente.

Etapa 3: Simulação

De forma a avaliar o comportamento do modelo de controle, do ponto de vista da simulação, dois casos foram simulados em um tempo de missão de 90 u.t.. No primeiro caso, considerou-se que *F1, F2* e *F3* fossem iguais a 0, desta forma foi possível observar se o controlador se comporta de acordo com a lei estabelecida no Quadro 3.5. As Figura 3.17 e Figura 3.18 apresentam uma destas simulações, sendo que a *query* utilizada foi*:*

simulate 1 [<=90] {H, controller.R1,controller.R2, controller.R3}

Nesta simulação foi possível observar 5 variações do nível *H* impostas pelo modelo *Random2*. Nota-se também que a variação do valor de *H*, ocorreu nos instantes de tempo esperados, ou seja, a cada 15 u.t.

Para a escolha do tempo de missão, de 90 u.t., levou-se em conta que a simulação deveria ilustrar as três transições possíveis do modelo do controlador, sendo elas: quando o valor de *H* é maior que 100 cm e o sistema entra na região *R3*, quando *H* é menor que -100 cm e o sistema entra na região *R1* e quando o valor atribuído a *H* está entre 100 cm e -100 cm e dessa forma o sistema se mantém em na região *R2*.

Observa-se que ocorreram duas situações em que o nível é atualizado para valores maiores que 100 cm, nos tempos 15 e 75 u.t.. Nestes mesmos instantes de tempo o sistema entrou na região *R3*, acionando *P1, P2* e *V*, a fim de diminuir *H*.

Com a atuação do controlador a taxa de variação de *H* passou a ser de -20 cm/u.t., logo há o esvaziamento do reservatório até o momento que o nível atinge valores menores que 100 cm, entrando na região *R2* e ali permanecendo até uma nova atualização de *H*.

Pode-se constatar também duas situações em que o nível é atualizado para valores menores que -100 cm, nos tempos 30 e 45 u.t., em que o sistema também se comportou como esperado, entrado na região *R1*. Como para enchimento, a condição mais favorável e utilizada neste caso, apresenta taxa de 40 cm/u.t, o sistema retorna a região *R2* mais rapidamente, conforme pode ser observado pela inclinação da reta na Figura 3.17.

Foi observada uma situação em que *H* é atualizado para 35 cm. Como este valor encontra-se dentro da região *R2,* o controlador não atua e mantém o nível neste valor até a próxima atualização de *H.* Ainda, sobre as regiões de controle, quando não há variação de *H*, o modelo *Level* não emite comando de atualização para o modelo *Controller*, portanto os estados *R1, R2* e *R3* permanecem em 0. Quando o nível H, atinge valores acima ou abaixo dos limites, o controlador entra nas regiões *R3* e *R1*, respectivamente, e ali permanece até que o nível retorne a valores dentro dos limites, sendo que *R2* sempre é acionado neste instante de retorno para retornar *P1, P2* e *V* as condições iniciais.

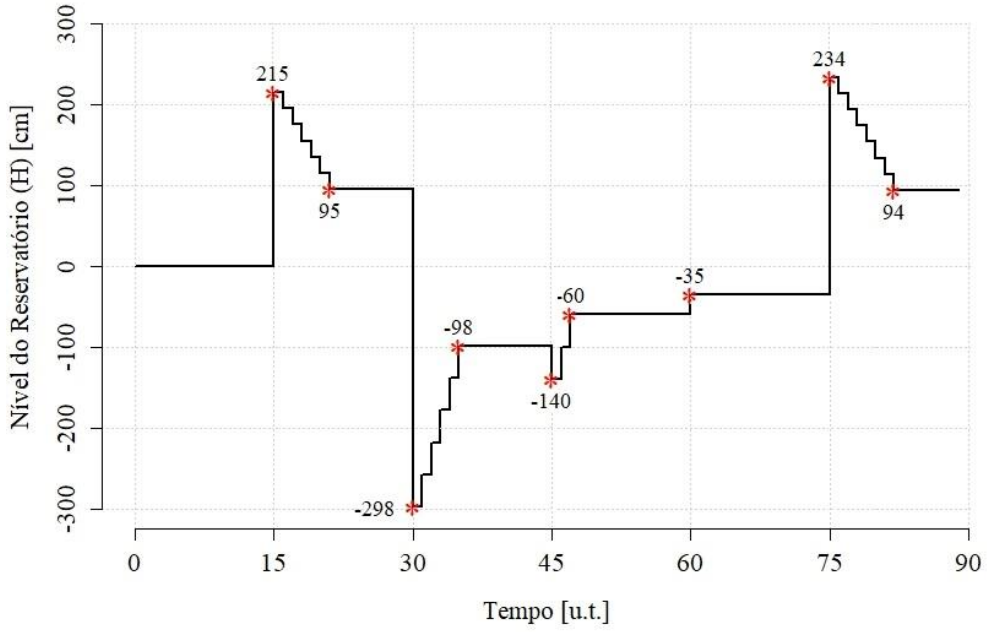


Figura 3.17 – Simulação 1: Variação do nível do reservatório para o modelo de controle.

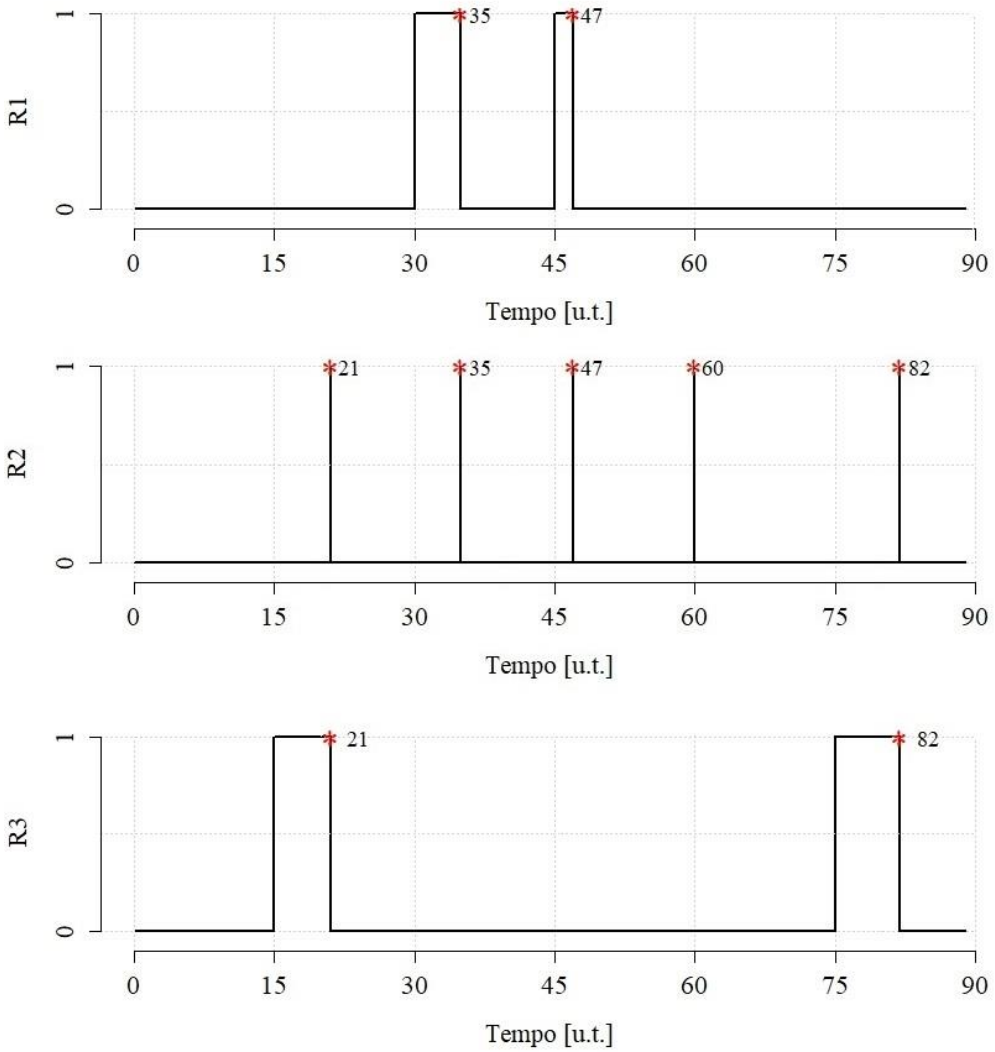


Figura 3.18 – Simulação 1: Regiões de controle *R1*, *R2* e *R3* para o modelo de controle.

No segundo caso simulado não foi restringido o valor das variáveis *F1, F2* e *F3.* Desta forma foi possível observar que quando o valor destas variáveis é igual a 1, o controlador não atua no respectivo componente mecânico, independentemente da região onde o sistema se encontra. Para este caso, foi utilizada a *query:*

simulate 1 [<=90] {H, F1,F2,F3,P,1,P2,V,controller.R1,controller.R2, controller.R3}

Nas Figura 3.19 e Figura 3.20 são demostrados os valores atribuídos as variáveis *F1, F2* e *F3* e os resultados obtidos para a variação do nível do reservatório. Para a correta interpretação dos resultados apresentados a Figura 3.21 ilustra a condição de operação de cada componente mecânica.

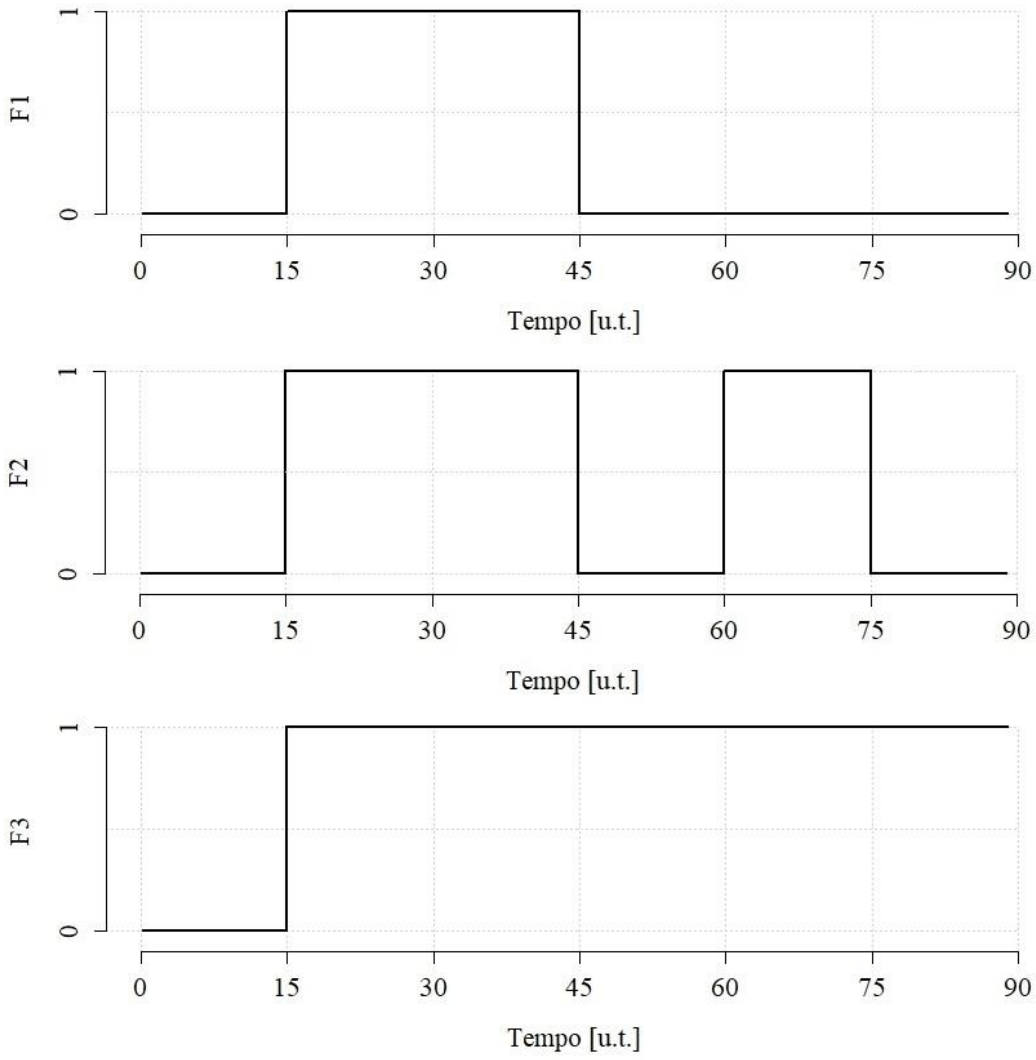


Figura 3.19 – Simulação 2: Valores de *F1, F2* e *F3* para o modelo de controle.

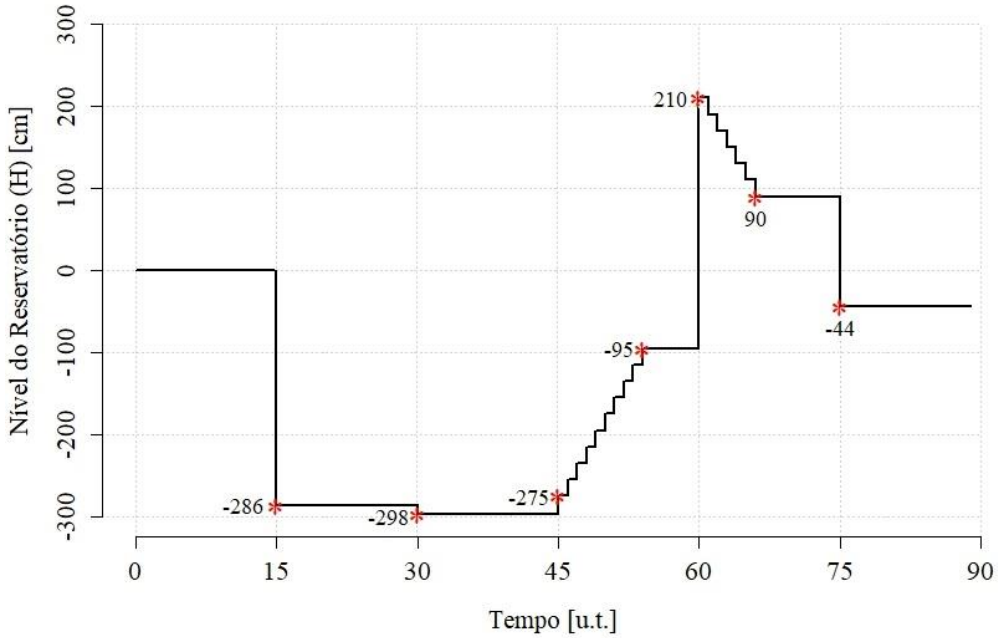


Figura 3.20 – Simulação 2: Variação do nível do reservatório para o modelo de controle.

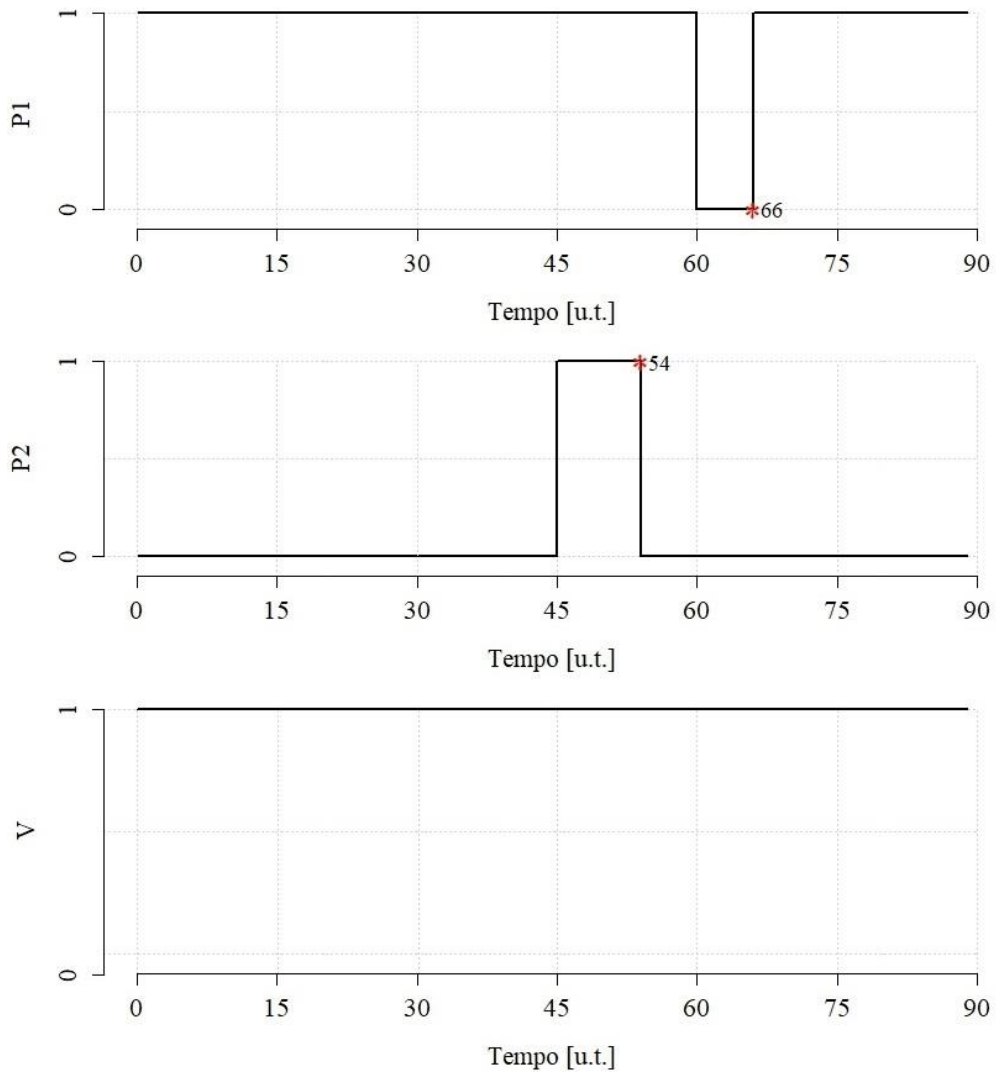


Figura 3.21 – Simulação 2: Condição de operação de *P1*, *P2* e *V* para o modelo de controle.

Nesta simulação pode-se observar que no tempo de 15 u.t. o modelo *Random2* atribuiu os seguintes valores para as variáveis: *H* = -286 cm, *F1* = 1, *F2* = 1 e *F3* = 1, portanto, mesmo o sistema atingindo a região *R1* (Ver Figura 3.22) não houve alteração do nível do reservatório até a próxima atualização visto que o controlador não pode atuar sobre os componentes mecânicos. Pode se observar que em 30 u.t. ocorreu o mesmo fato.

Em 45 u.t. o modelo *Random2* atribuiu os seguintes valores para as variáveis: *H* = -275 cm, *F1* = 0, *F2* = 0 e *F3* = 1. Neste caso, o controlador entrou na região *R1* e pode alterar apenas o estado de operação da bomba 1 e bomba 2, em que ambas deveriam ser ligadas, como a válvula estava na posição aberta, a taxa de variação do reservatório passou a ser de 20 cm/u.t., sendo que quando atingiu o valor de -95 cm em 54 u.t., o sistema entro na região *R2* e retornou os componentes *P1* e *P2* para sua condição inicial.

Em 60 u.t, o nível do reservatório foi atualizado para *H* = 210 cm, *F1* = 0, *F2* = 1 e *F3*=1, como o valor de *H* ultrapassou o valor de 100 cm, o sistema entrou na região *R3*, como o controlador só tinha permissão para alterar o estado de *P1*, a mesma foi desligada, resultado em uma taxa de variação de *H* de 20 cm/u.t.. Por fim, em 75 u.t. as variáveis receberam os valores *H* =-44cm, *F1* = 0, *F2* = 0 e *F3*=1, como o sistema não ultrapassou os valores limites da região *R2*, o nível do reservatório permaneceu neste valor até o final da simulação.

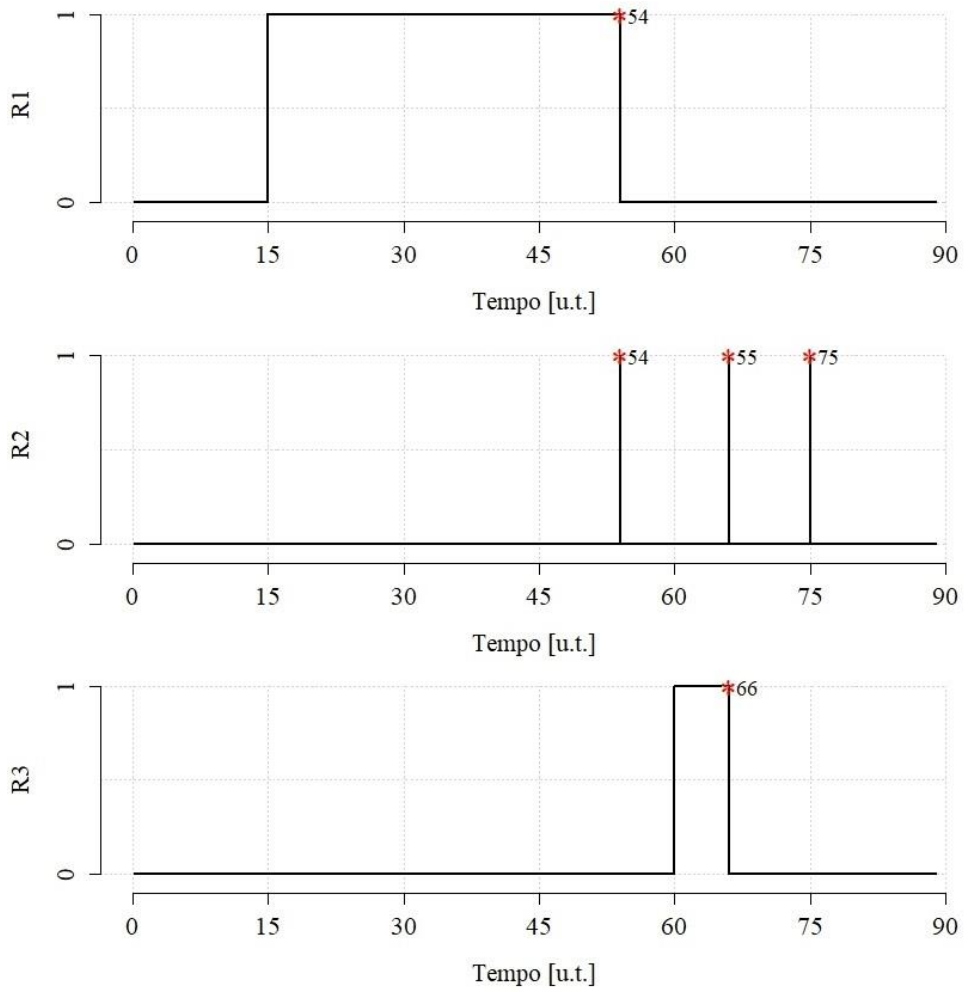


Figura 3.22 – Simulação 2: Regiões de controle *R1, R2* e *R3* para o modelo de controle.

Conclui-se que, do ponto de vista da simulação, o modelo de controle se comportou conforme o esperado para os dois casos simulados.

Etapa 4: Verificação Formal

No Quadro 3.6 estão apresentadas as verificações realizadas para o modelo de controle, de acordo com a etapa de especificação de propriedades, em que se constata que o modelo atende todas as propriedades exigidas.

Quadro 3.6 – Verificação formal para o modelo de controle

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita?**
Verifica a ausência de *deadlock*	`A[] not deadlock`	Sim
Acessibilidade		
Existe um caminho em que o controlador desliga *P1* e *P2* e liga *V*.	`E<> controller.CONTROL and pump1.OFF and pump2.OFF and valve.ON`	Sim
Existe um caminho em que o controlador desliga *V* e liga *P1* e *P2*.	`E<> controller.CONTROL and pump1.ON and pump2.ON and valve.OFF`	Sim
Existe um caminho em que o controlador desliga *P2* e liga *P1* e *V*.	`E<> controller.CONTROL and pump1.ON and pump2.OFF and valve.ON`	Sim
Quando *H* for maior ou igual a 100 cm, o controlador deve entrar no estado referente a região 03.	`E<> H>=100 and controller.R3`	Sim
Quando *H* for menor ou igual a -100 cm, o controlador pode entrar no estado referente a região 01.	`E<> H<=-100 and controller.R1`	Sim
Quando *H* estiver entre 100 cm e -100 cm, o controlador pode entrar no estado referente a região 02.	`E<> (H<100 and H>-100) and controller.R2`	Sim

Continuação do Quadro 3.6 – Verificação formal para o modelo de controle.

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita?**
Segurança		
O controlador não pode permitir nenhuma configuração de *P1*, *P2* e *V* diferentes da lei de controle, quando *F1*, *F2* e *F3* forem iguais a 0.	`E[] (F1==0 and F2==0 and F3==0) and controller.CONTROL and pump1.ON and pump2.OFF and valve.OFF`	Sim
	`E[] (F1==0 and F2==0 and F3==0) and controller.CONTROL and pump1.ON and pump2.ON and valve.ON`	
	`E[] (F1==0 and F2==0 and F3==0) and controller.CONTROL and pump1.OFF and pump2.OFF and valve.OFF`	
	`E[] (F1==0 and F2==0 and F3==0) and controller.CONTROL and pump1.OFF and pump2.ON and valve.OFF`	
	`E[] (F1==0 and F2==0 and F3==0) and controller.CONTROL and pump1.OFF and pump2.ON and valve.ON`	
Quando H for maior ou igual a 100 cm, o controlador não pode entrar nos estados referente a região 01 e 02.	`E[] H>=100 and (controller.R2 or controller.R1)`	Sim
Quando H for menor ou igual a -100 cm, o controlador não pode entrar nos estados referente a região 02 e 03.	`E[] H<=-100 and (controller.R3 or controller.R2)`	Sim
Quando H estiver entre 100 cm e -100 cm, o controlador não pode entrar nos estados referente a região 01 e 03.	`E[] (H<100 and H>-100) and (controller.R3 or controller.R1)`	Sim
Quando *F1* for igual a 1, o controlador não pode atuar sob *P1*.	`E[] F1==1 and controller.i1==1 and controller.CONTROL`	Sim
Quando *F2* for igual a 1, o controlador não pode atuar sob *P2*.	`E[] F2==1 and controller.i2==1 and controller.CONTROL`	Sim
Quando *F3* for igual a 1, o controlador não pode atuar sob *V*.	`E[] F2==1 and controller.i3==1 and controller.CONTROL`	Sim

3.3.3 Modelo de Falha

O modelo físico e de controle foram construídos com o intuito de avaliar se o modelo apresenta comportamento compatível com o sistema em estudo. Após simulação e verificação formal desses modelos, pode-se constatar que o modelo apresenta o comportamento adequado, logo é possível avaliar as características de falha desse sistema.

Para tanto, foi desenvolvido o modelo de falha em que foi realizada a inclusão das falhas dos componentes mecânicos do sistema, sendo possível avaliar o comportamento do sistema quando submetido a falha. Porém, como já mencionado, o propósito deste modelo é o levantamento da característica de falha para comparação com os resultados disponíveis nas literaturas citadas na seção 3.1 e assim poder aferir se a metodologia empregada pode ser utilizada para predição dos parâmetros de confiabilidade de sistemas híbridos.

Etapa 1: Especificação das propriedades

A fim de validar esse modelo alguns requisitos foram levantados, a saber:

1. Ausência de *deadlock:* O modelo desenvolvido não pode travar, ou seja, não deve haver estados que impossibilitem a transição do sistema.
2. Acessibilidade: Deve haver possibilidade de os componentes mecânicos atingirem o estado de falha.

Para este modelo também foram verificadas todas as especificações levantadas para o modelo físico e de controle, que serão repetidas a seguir:

1. Acessibilidade:
    a. Deve haver pelo menos um caminho que permita que os componentes mecânicos do sistema (*P1, P2* e *V*) passem da condição fechada para aberta, ou, vice-versa.
    b. Deve haver caminhos que permitam que todas as configurações do sistema (Tabela 3.2) possam ocorrer.
    c. O modelo deve permitir alcançar todos os níveis do reservatório (*H*) (Figura 3.2).
    d. Deve haver coerência entre os estados P1, P2 e V e o nível H, ou seja, sempre que o nível do reservatório (*H*) atingir um determinado valor numérico, devido a configuração do sistema, o mesmo deve alcançar determinado estado de acordo com a Figura 3.2.
    e. Sempre que o nível do reservatório (*H*) ultrapassar os limites da região de correto funcionamento, o controlador deve atuar de acordo com a lei estabelecida no Quadro 3.1.
    f. Sempre que o nível do reservatório (*H*) atingir determinado valor numérico, o controlador deve entrar no estado referente a região compatível com este valor.

2. Segurança:
    g. Quando o nível do reservatório (*H*) atingir um determinado valor numérico, devido a configuração de *P1*, *P2* e *V*, o sistema não pode entrar em determinados estados do reservatório.
    h. O controlador não pode admitir configurações diferentes daquelas definidas pela lei de controle, quando as variáveis *F1*, *F2* e *F3* forem iguais a 0.
    i. Quando o nível do reservatório (*H*) atingir determinado valor numérico, o controlador não pode entrar nos estados que se referem as regiões não compatíveis com este valor.
    j. Quando *F1*, *F2* e/ou *F3* forem iguais a 1, o controlador não podem atuar sobre o respectivo componente mecânico, enviado comandos de abertura ou fechamento, sendo que *F1* refere-se a *P1*, *F2* a *P2* e *F3* a *V*.

Etapa 2: Modelagem

Para o desenvolvimento do modelo de falha foram realizadas algumas adaptações no modelo de controle e inserido mais 4 modelos, totalizando 10 modelos. Os modelos *Pump1*, *Pump2*, *Valve*, *Tank* e *Controller* permaneceram inalterados enquanto que o modelo *Random2* foi excluído, visto que este modelo foi desenvolvido apenas para testar o comportamento do controlador sob diferentes valores de *H* e não condiz com o comportamento físico do sistema.

Para melhor representar o comportamento físico do nível do reservatório é necessário desenvolver um modelo que presente as falhas dos componentes mecânicos, visto que a mudança no nível do reservatório ocorre devido a essas falhas. Os modelos construídos e modificados para o desenvolvimento do modelo de falha são descritos a seguir:

1. *ExpP1*, *ExpP2*, *ExpV*: Representam a falha dos componentes *P1*, *P2* e *V*, respectivamente. Como já mencionado na seção 3.1, estes componentes apresentam distribuição exponencial de falha. Esta função é nativa ao UPPAAL, portanto a taxa de falha de cada componente foi inserida no campo *rate of exponencial*, sendo a responsável pela transição de cada componente para a condição de falha e é expresso no formato $1/MTTF \equiv 1{:}\,MTTF$.

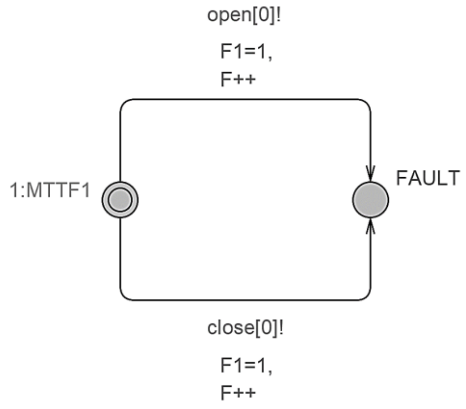


Figura 3.23 – Autômato *ExpP1* do modelo de falha

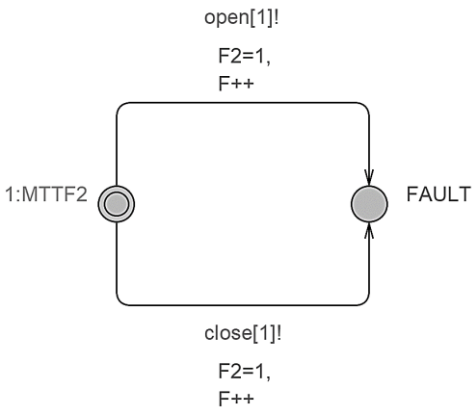


Figura 3.24 – Autômato *ExpP2* do modelo de falha

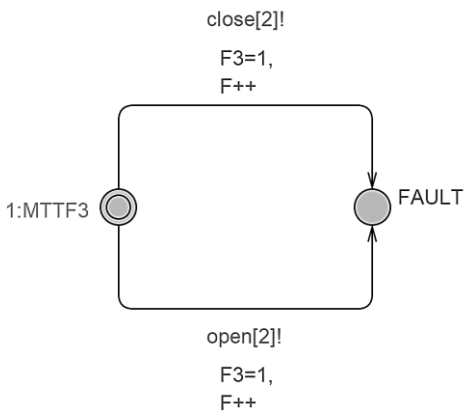


Figura 3.25 – Autômato *ExpV* do modelo de falha

As Figura 3.23, Figura 3.24 e Figura 3.25 representam esses modelos, em que *MTTF1*, *MTTF2* e *MTTF3* são os tempos médios para falha exposto na Tabela 3.1 (Página 36). As variáveis *F1, F2* e *F3*, são atualizadas de 0 para 1 quando o seu respectivo componente entra em falha, e a variável *F* tem seu valor incrementado se qualquer componente entrar em falha. Essas variáveis servem de parâmetro de decisão para o modelo *Controller* e *Level.*

2. *Obs*: Observador da condição de operação de *P1, P2* e *V*. Este autômato foi criado com o intuito de redução de esforço computacional, uma vez que o modelo não precisa monitorar o valor de *H* a cada 1 u.t. Caso ocorra algum comando de abertura ou fechamento para qualquer um dos componentes mecânicos, o observador avalia se essa mudança provoca alteração do nível do reservatório, caso ocorra envia um comando de atualização (*run)* para o modelo *Level*, caso não provoque a alteração de *H*, o *Level* permanece no mesmo estado. Deste modo, tem-se a vantagem de (Figura 3.26).

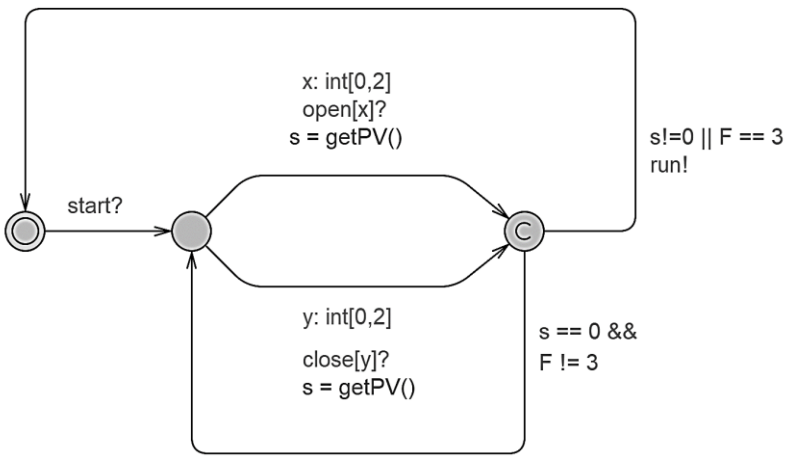


Figura 3.26 – Autômato *Obs* do modelo de falha.

3. *Level:* Neste modelo também foram realizadas algumas modificações para a inclusão da falha (Ver Figura 3.27).

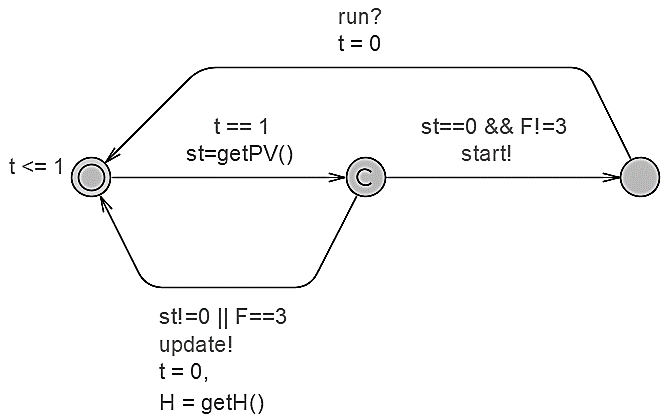


Figura 3.27 – Autômato *Level* do modelo de falha.

Sendo incluída a variável *F* como parâmetro de decisão, caso *F* seja igual a 3, todos os componentes já falharam, e não haverá mais mudança de seus estados, portanto não é permitida a transição para o estado de comunicação entre o *Level* e *Obs.* Caso *F* seja diferente de 3, o autômato pode transitar livremente seguindo a mesma lógica do que para o modelo de controle.

Etapa 3: Simulação

De forma a avaliar o comportamento do modelo de falha, do ponto de vista da simulação, foi utilizada a *query:*

simulate 1 [<=3000] {H, expP1.FAULT, expP2.FAULT, expV.FAULT, P1, P2, V}

A simulação foi realizada para um tempo de missão de 3000 u.t. Para este caso foram simulados a variação de *H*, os tempos que os componentes entram em falha e a posição que eles falharam. O tempo de missão utilizado para este modelo foi o tempo estipulado pelas literaturas consultadas para o modelo de falha. Várias simulações foram realizadas a fim de verificar o comportamento do modelo de falha, a seguir são apresentados os resultados de uma destas simulações que evidencia a falha dos três componentes mecânicos do sistema.

A Figura 3.28 ilustra os tempos que cada um dos componentes mecânicos entrou em falha. O primeiro a falhar foi *V* no tempo de 653 u.t., na condição ligada, ou seja, *V* = 1 (como pode ser observado na Figura 3.29), na sequência, no tempo de 760 u.t., falhou *P2*, também na posição ligada, *P2* = 1, e por fim em 1072 u.t. a bomba *P1* falha ligada, *P1* = 1.

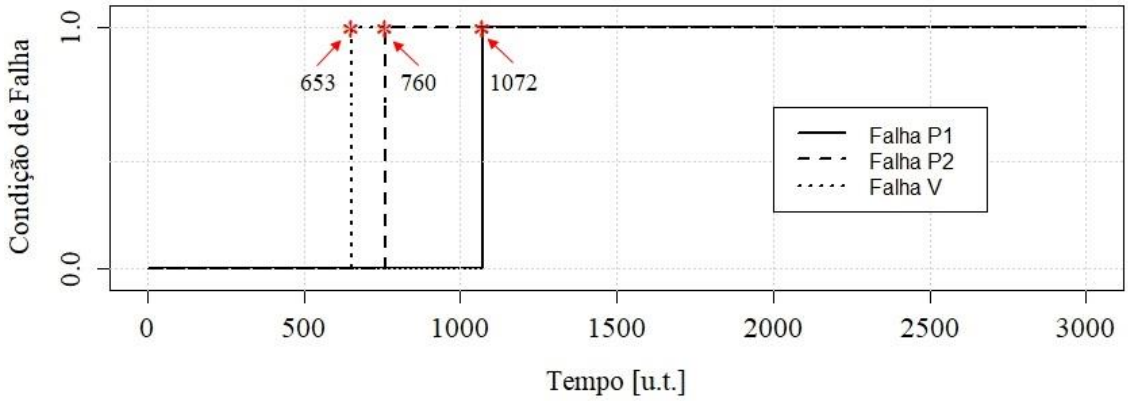


Figura 3.28 – Simulação do tempo para falha de *P1*, *P2* e *V*

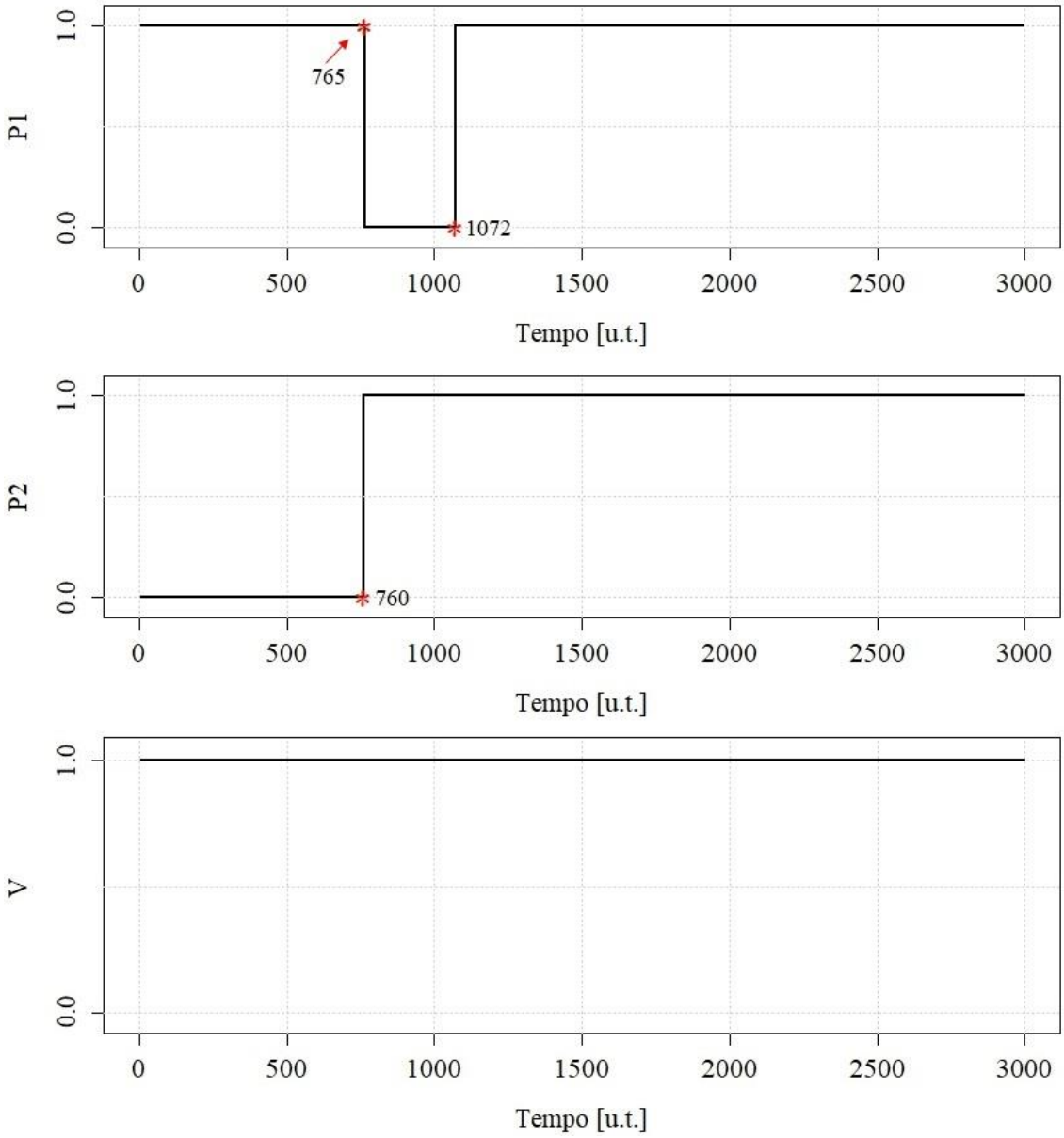


Figura 3.29 - Simulação da condição de operação de P1, P2 e V para o modelo de falha

A primeira falha ocorre em t = 653 u.t., neste ponto, *V* trava aberta, visto que esta é a condição inicial da válvula, o nível permanece constante, isto é, H = 0, e nenhuma alteração no sistema acontece até a próxima falha. Em t = 760 u.t., ocorre a falha de *P2* na posição ligada, a partir desse instante a taxa de variação de *H* passa a ser de 20 cm/u.t., portanto em 5 u.t., o reservatório atinge 100 cm, entrando na região *R3* (risco de transbordamento), este fato pode ser observado na Figura 3.30, que ilustra a variação do nível *H*.

Como o reservatório atinge o estado HLB, o controlador atua sobre o sistema. A ação do controlador deveria desligar *P1* e *P2* e ligar *V*, porém *P2* e *V* estão em condição de falha, logo o controlador atua apenas sobre *P1*, desligando-a. Isto resulta em *P1* = 0, *P2* = 1 e *V* = 1, logo a taxa de variação de *H* passa a ser de 0 cm/u.t., e o nível do reservatório permanece em 100cm até a próxima falha que ocorre em 1072 u.t., em que *P1* trava na posição ligada. Como os 3 componentes mecânicos estão em condição de falha, o controlador não altera seus estados, e a taxa de variação de *H* passa a ser de 20 cm/u.t., deste modo, em 10 u.t., o reservatório atinge 300 cm, transbordando.

A influência das falhas na taxa de variação de *H* pode ser observada na Figura 3.30, onde pode se verificar as duas variações do nível.

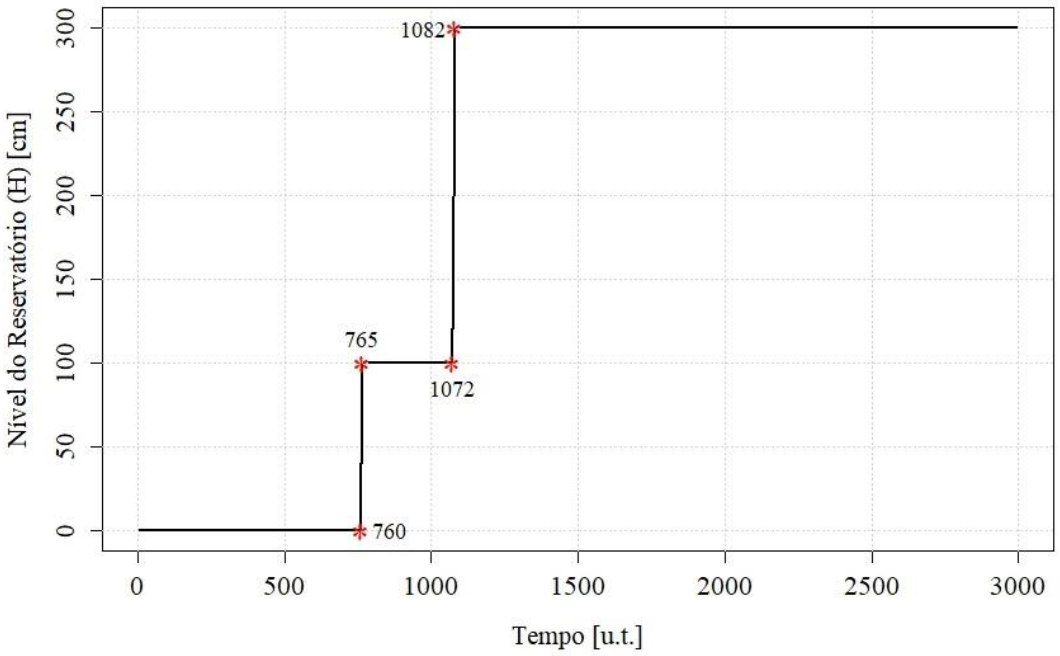


Figura 3.30 – Simulação da variação do nível do reservatório para o modelo de falha

Etapa 4: Verificação Formal

No Quadro 3.7 estão apresentadas as verificações realizadas para o modelo de falha, onde se constata que o modelo cumpre todos os requisitos definidos na etapa de especificação de propriedades.

Quadro 3.7 – Verificação formal para o modelo de falha

Descrição Informal	Descrição Formal	Propriedade Satisfeita?
Verifica a ausência de *deadlock*	`A[] not deadlock`	Sim
Acessibilidade		
Existe pelo menos um caminho que leva *P1, P2* ou *V* para estado de falha.	`E<> expP1.FAULT`	Sim
	`E<> expP2.FAULT`	
	`E<> expV.FAULT`	

As demais propriedades verificadas foram àquelas já apresentadas nos Quadro 3.4 (Página 50) e Quadro 3.6 (Página 60), as quais foram devidamente testadas e aceitas.

3.3.4 Modelo de Manutenção

O modelo de manutenção foi implementado para a inclusão de uma política de reparo nos componentes mecânicos do sistema. A partir desse modelo, é possível avaliar a influência do reparo sobre a característica de falha do sistema. Para fins de comparação dos resultados, este modelo foi desenvolvido seguindo as mesmas considerações de Marseguerra e Zio (1996) e Codetta-Raiteri e Bobbio (2006), que são descritas a seguir:

- Se o nível do reservatório não está dentro da região de correto funcionamento (HLA≤H≤HLB), o controlador suspeita da ocorrência de falha e habilita o processo de reparo dos componentes travados.
- O tempo médio de reparo (*MTTR*) é de 5h e segue a distribuição exponencial.
- O reparo só é permitido enquanto o nível do reservatório estiver fora da região de correto funcionamento.
- O efeito do reparo consiste na remoção da condição de travamento e leva o componente para o estado de funcionamento, *ON* ou *OFF*, de acordo com o requerido para o sistema retornar a região de correto funcionamento.
- Durante o tempo de reparo, o componente permanece travado na posição que falhou.
- Os níveis máximo e mínimo do reservatório foram alterados para +5 m e -5 m, respectivamente, a fim de tornar o tempo que leva para o transbordamento e esvaziamento significativo comparado com o tempo de reparo.

Etapa 1: Especificação das propriedades

A fim de validar esse modelo alguns requisitos foram especificados:

1. Ausência de *deadlock:* O modelo desenvolvido não pode travar, ou seja, não deve haver estados que impossibilitem a transição do sistema.

2. Acessibilidade:
    a. Deve haver possibilidade de os componentes mecânicos atingirem o estado de manutenção.
    b. Os componentes em falha devem entrar em manutenção sempre que o nível ultrapassar os limites da região de correto funcionamento.
    c. O reparo deve ser interrompido caso o sistema retorne à região de correto funcionamento.
3. Segurança:
    a. Os componentes não podem estar em manutenção quando o nível está dentro da região de correto funcionamento.
    b. O reparo não pode ser interrompido se o nível estiver fora da região de correto funcionamento.

Etapa 2: Modelagem

Para atender as considerações apresentadas, foi necessário realizar algumas alterações no modelo de falha, sendo que os modelos *Pump1, Pump2, Valve, Obs* e *Level* não sofreram modificações e as alterações dos demais modelos, bem como os modelos incluídos, são descritos na sequência. Para o modelo de manutenção foram utilizados 13 modelos ao todo.

1. *ExpP1, ExpP2, ExpV*: Além de representarem a falha dos componentes *P1, P2* e *V*, respectivamente, enviam o comando para os modelos de manutenção corretiva para iniciarem o reparo (Figura 3.35, Figura 3.36 e Figura 3.37).

    As Figura 3.31, Figura 3.32 e Figura 3.33 representam esses modelos, onde pode ser observado que após atingir o tempo até a falha o componente trava na posição ligada ou desligada.

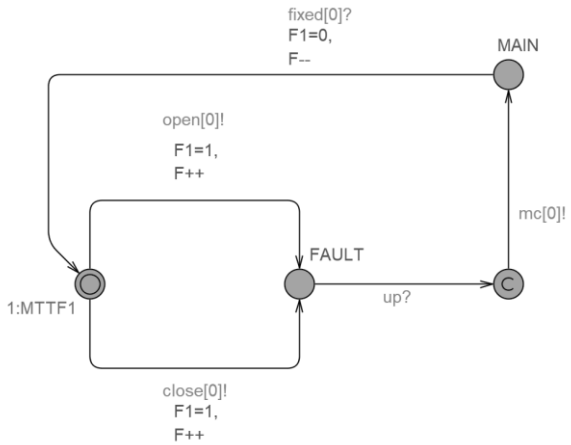


Figura 3.31 – Autômato *ExpP1* para o modelo de manutenção.

Quando este modelo recebe o comando *up* do controlador, significa que *H* ultrapassou o valor limite (100 cm ou -100 cm) para a região *R2*, e, portanto, os

componentes em falha devem entrar em manutenção, dessa forma o modelo envia um comando para iniciar a manutenção do componente (*mc[0],mc[1]* e/ou *mc[2])*, e entra no estado de manutenção (*MAIN*). Quando o reparo é finalizado, recebe um comando de término de manutenção (*fixed[0]*, *fixed[1]* e/ou *fixed[2])*, atualiza o valor das variáveis *F*, indicando que o componente não está mais na condição de falha.

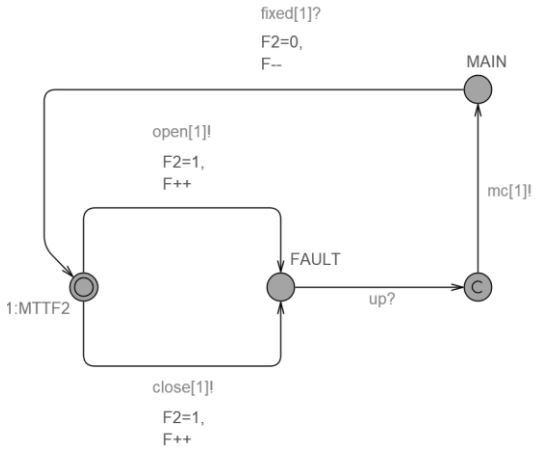


Figura 3.32 – Autômato *ExpP2* para o modelo de manutenção.

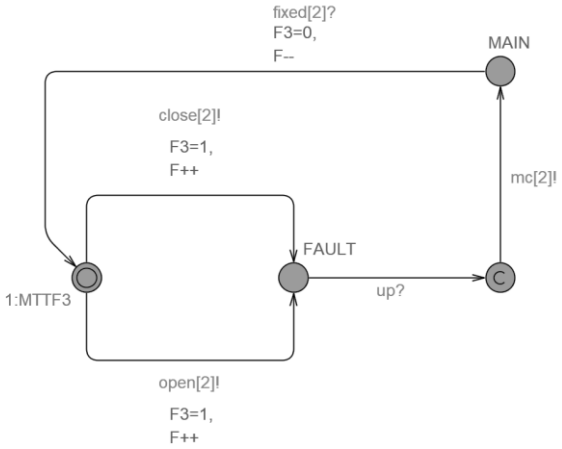


Figura 3.33 – Autômato *ExpV* para o modelo de manutenção.

2. *Controller:* Além da função que este modelo já possuía, para o modelo de manutenção foram acrescentados dois canais de sincronização. O primeiro (*up*!) para enviar aos modelos *ExpP1, ExpP2 e ExpV* o sinal para que o reparo seja iniciado quando *H* sai da região *R2*, e o segundo (*stop*!), para interromper o reparo dos componentes que estão em manutenção quando o reservatório volta a região *R2*, (Figura 3.34). Ressalta-se que as mudanças realizadas não afetam o modelo de controle testado anteriormente, visto que nenhuma modificação estrutural (estados/transição) foi realizada, apenas foram incluídos canais de sincronização que enviam sinais para os demais modelos quando passam por determinado caminho.

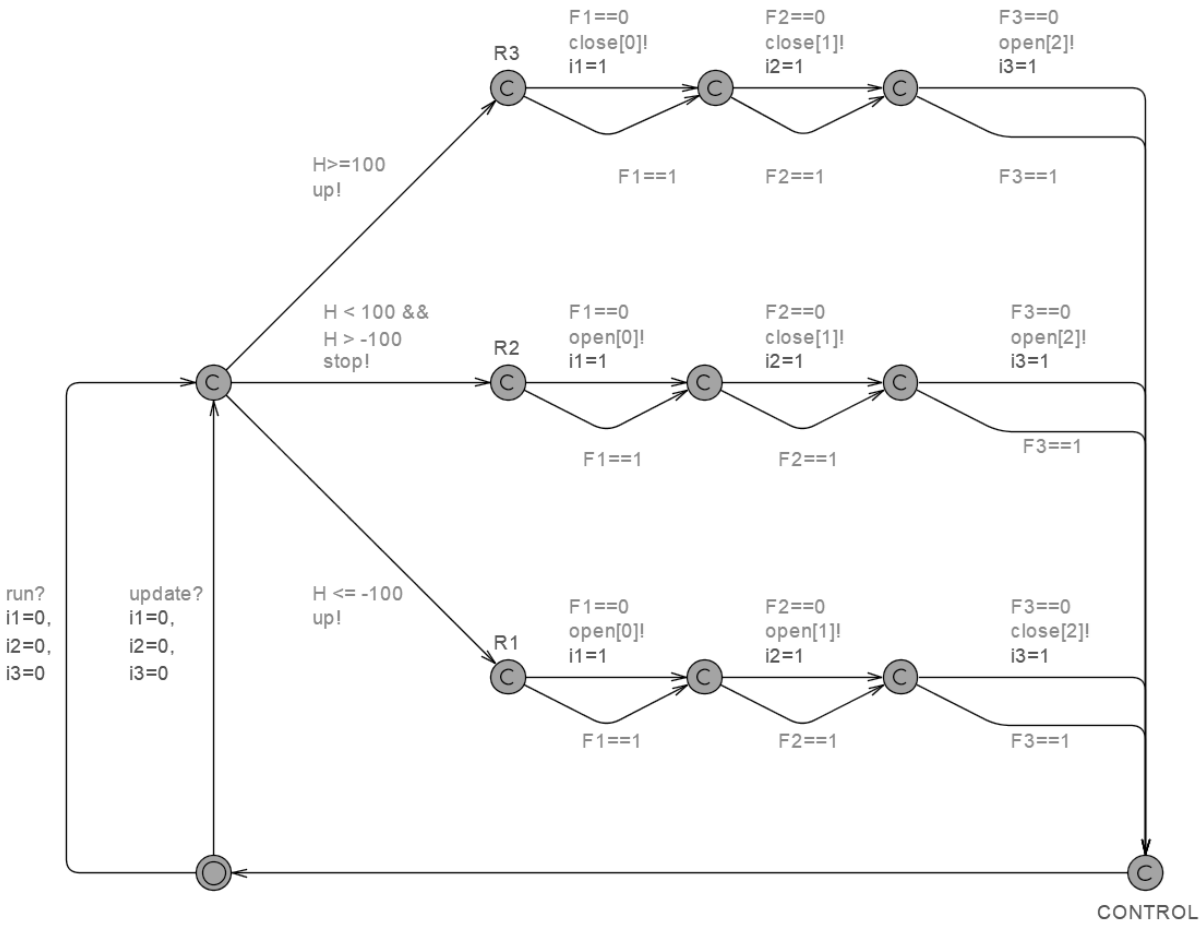


Figura 3.34 – Autômato *Controller* para o modelo de manutenção.

3. *RepairP1, RepairP2 e RepairV:* estes modelos representam a manutenção corretiva que deve ser realizada nos componentes, quando estes estão em condição de falha e o nível *H* está fora da região *R2* (Figura 3.35, Figura 3.36 e Figura 3.37).

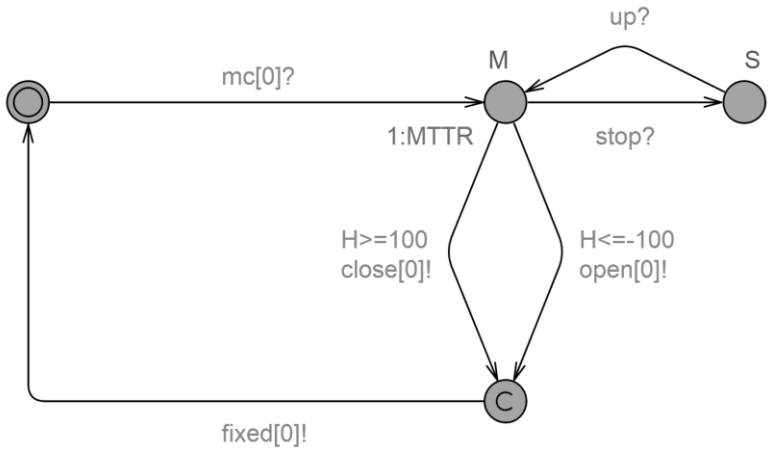


Figura 3.35 – Autômato *RepairP1* para o modelo de manutenção.

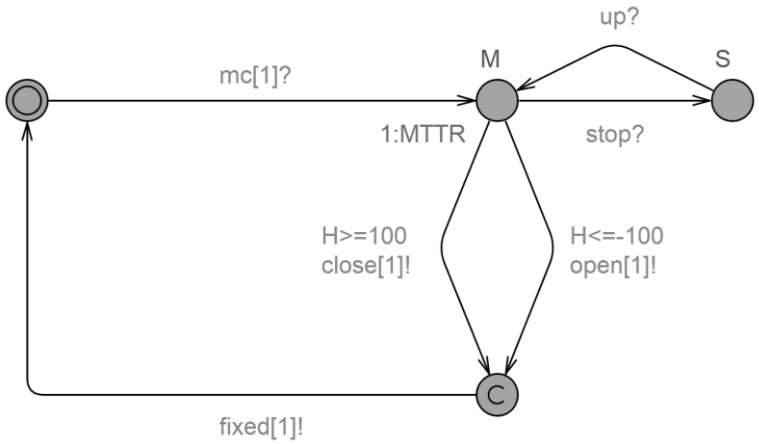


Figura 3.36 – Autômato *RepairP2* para o modelo de manutenção.

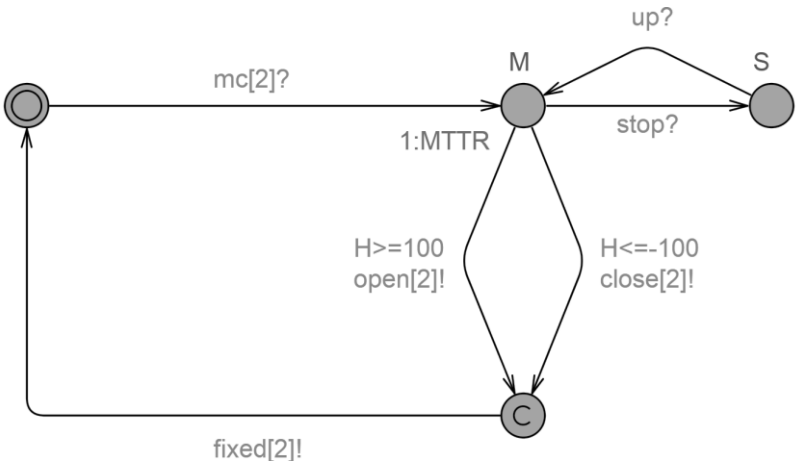


Figura 3.37 – Autômato *RepairV* para o modelo de manutenção.

Estes modelos recebem os sinais de sincronização dos modelos de falhas dos seus respectivos componentes, caso estes estejam em falha. Atingindo o estado de reparo, há três transições possíveis: quando se atinge o tempo de manutenção, estabelecido pela função exponencial (*1:MTTR)*, caso o nível esteja acima de 100 cm ou abaixo de -100 cm, a transição ocorre alterando a condição de operação do componente de acordo com a lei de controle estabelecida no Quadro 3.1 (Página 38), imediatamente envia um final de término de reparo para o componente (*fixed[0]!*, *fixed[1]!* e/ou *fixed[2]!)*.

A terceira transição possível é quando o modelo recebe o sinal de sincronização *stop?*. Neste caso, significa que outro componente já retornou de manutenção e reestabeleceu o nível *H* para a região *R2* e portanto o reparo deve ser interrompido. Estando neste estado, se o sistema atinge as regiões *R1* ou *R3*, o componente entra novamente em manutenção, recebendo o sinal de sincronização *up?*.

4. *Tank:* Uma vez que o nível máximo e mínimo do reservatório foi alterado para 5 m e -5 m, respectivamente, neste modelo foi necessário atualizar o valor de *H* nas invariantes, substituindo os valores de *H*=300 cm por 500 cm e *H*=-300 cm por -500cm (Ver Figura 3.38). Também foi necessária a mesma atualização de valores na função *getH()*.

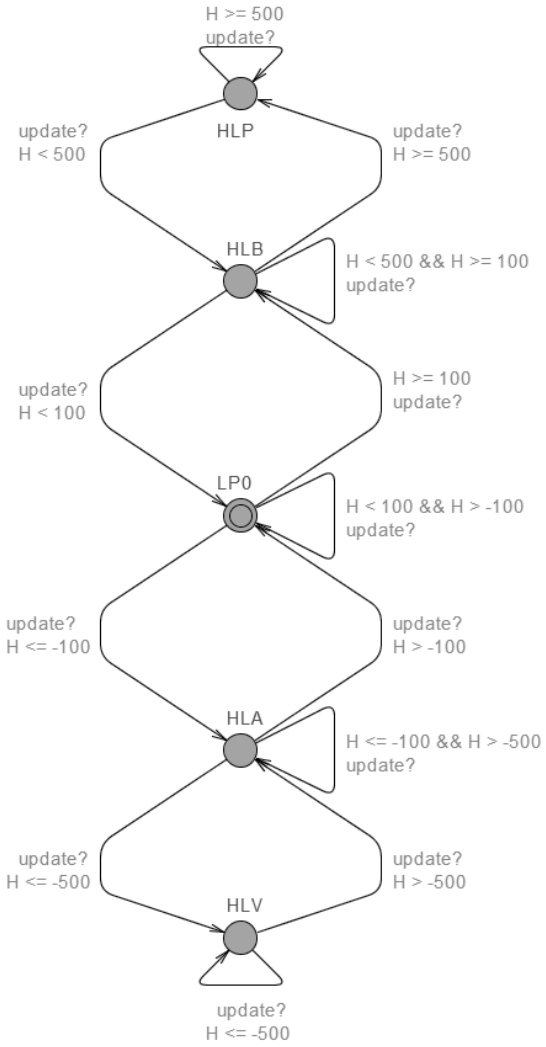


Figura 3.38 – Autômato *Tank* para o modelo de manutenção.

Etapa 3: Simulação

De forma a avaliar o comportamento do modelo de manutenção, do ponto de vista da simulação, foi utilizada a *query:*

simulate 1 [<=1000] {H, expP1.FAULT, expP2.FAULT, expV.FAULT, expP1.MAIN, expP2.MAIN, expV.MAIN, P1, P2, V}

A simulação foi realizada para um tempo de missão de 1000 u.t.. É importante enfatizar que para a simulação de cada modelo desenvolvido, procurou-se adotar um tempo de missão em que as características estudadas ficassem evidentes. Para este caso foram simulados a variação de *H*, os tempos que os componentes entram e saem do estado de falha e de manutenção, e a condição de operação dos componentes ao longo do tempo. Várias simulações foram realizadas a fim de verificar o comportamento do modelo de manutenção. A seguir são apresentados os resultados de uma destas simulações.

As Figura 3.39 e Figura 3.40 ilustram em quais tempos as bombas entraram em estado de falha e de manutenção. A válvula não foi representada, pois para esta simulação dentro do tempo de missão estipulado não houve a falha deste componente, logo, o mesmo não entrou em estado de falha e nem de manutenção. A Figura 3.41 ilustra a condição de operação dos três componentes mecânicos, já a Figura 3.42 ilustra a variação do nível do reservatório para o modelo de manutenção e a Figura 3.43 o detalhamento da variação de *H* nos tempos onde ocorre mudança de estado dos componentes.

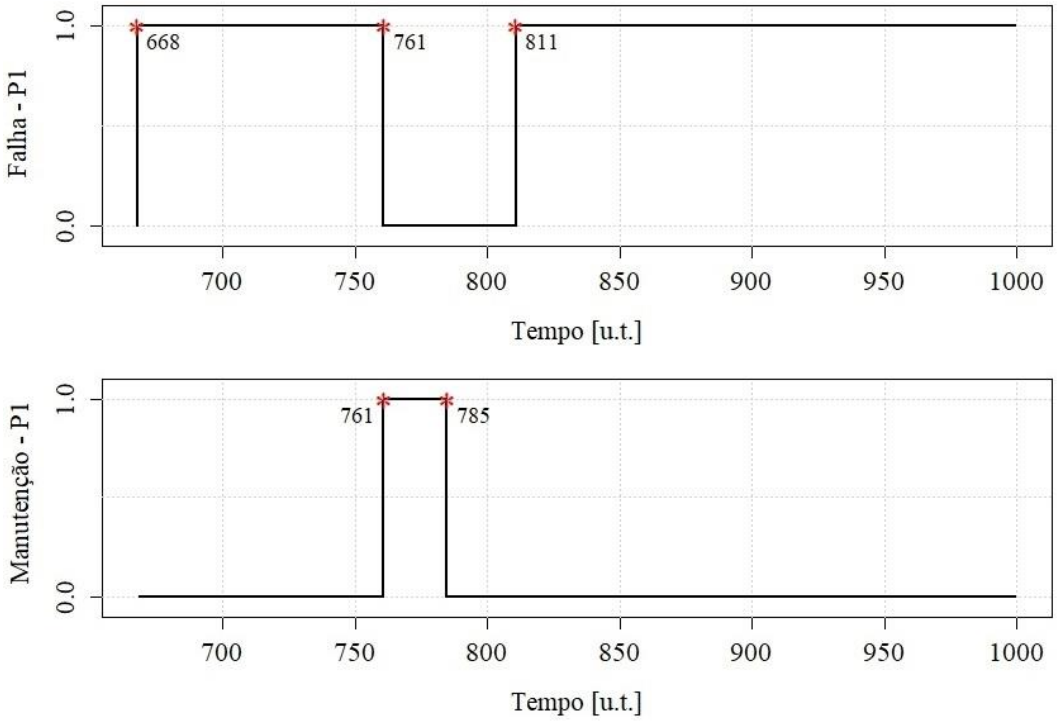


Figura 3.39 – Simulação do Estado de Falha e de Manutenção de *P1*.

Observa-se que a bomba *P1* é o primeiro componente a falhar no tempo de 668 u.t., na posição ligada (*P1*=1), portanto, o sistema mantém sua configuração e não há variação no nível do reservatório.

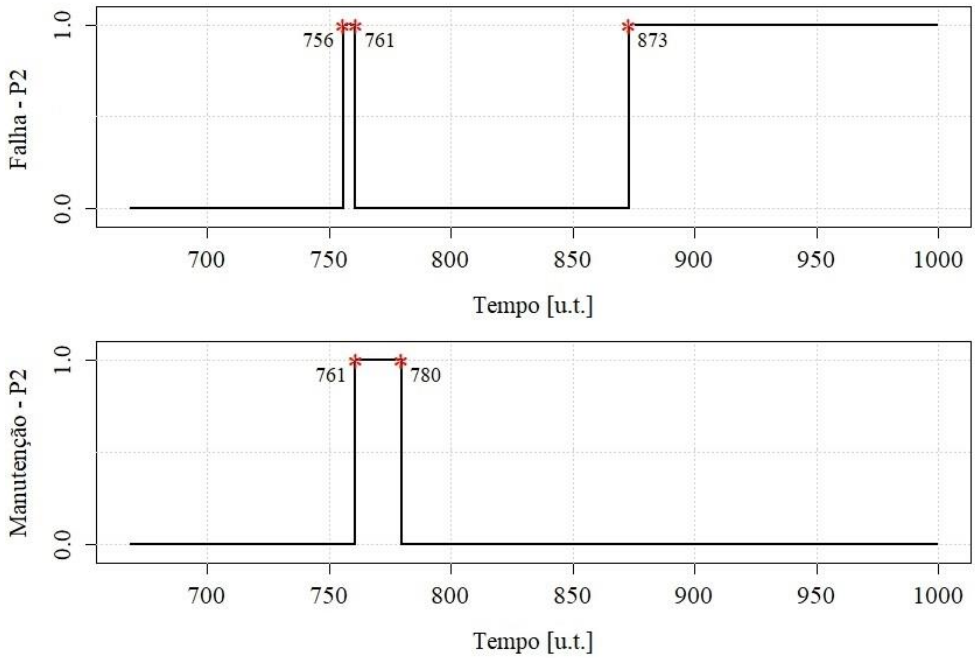


Figura 3.40 – Simulação do Estado de Falha e de Manutenção de *P2*.

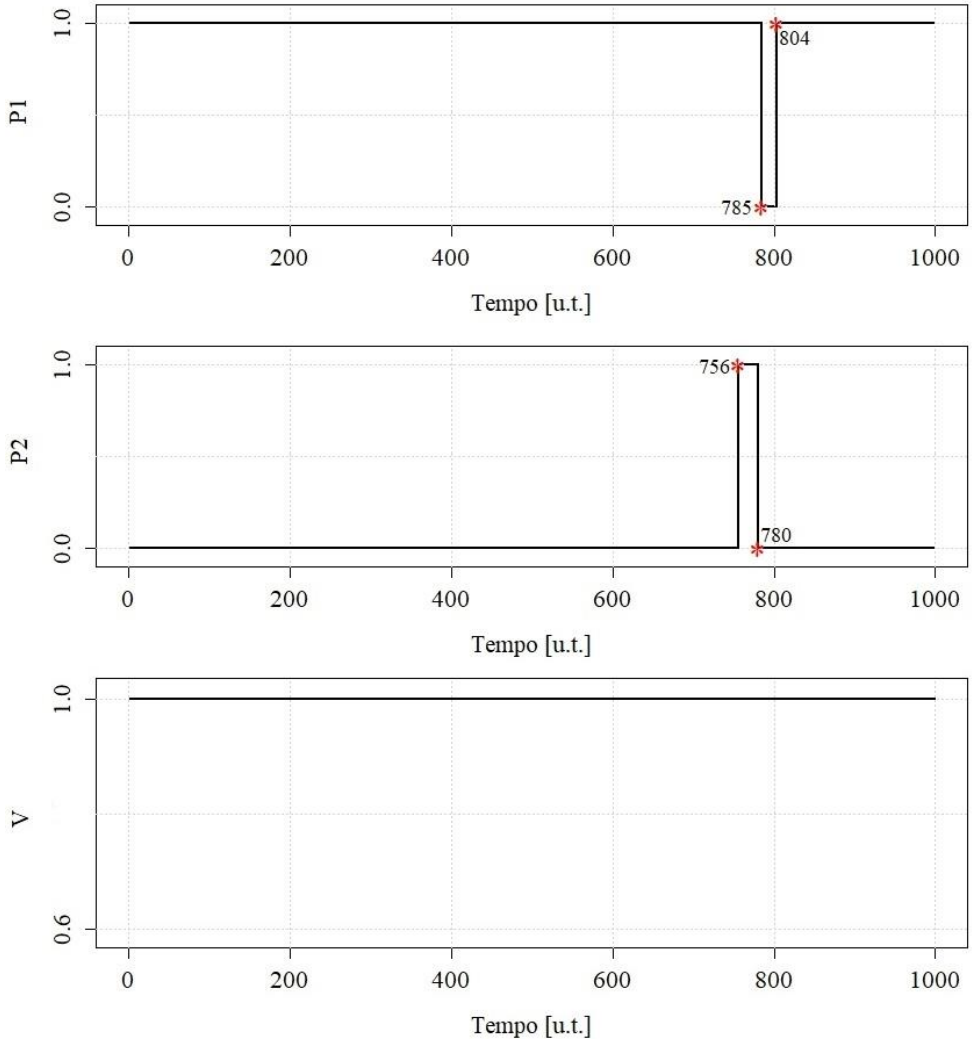


Figura 3.41 - Simulação da condição de operação de *P1*, *P2* e *V* para o modelo de manutenção.

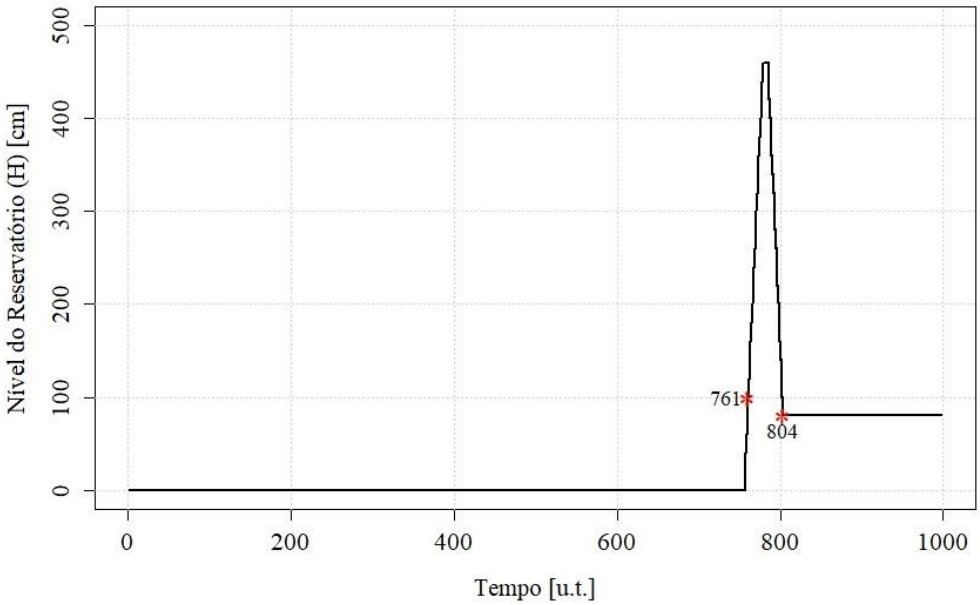


Figura 3.42 - Simulação da variação do nível do reservatório para o modelo de manutenção.

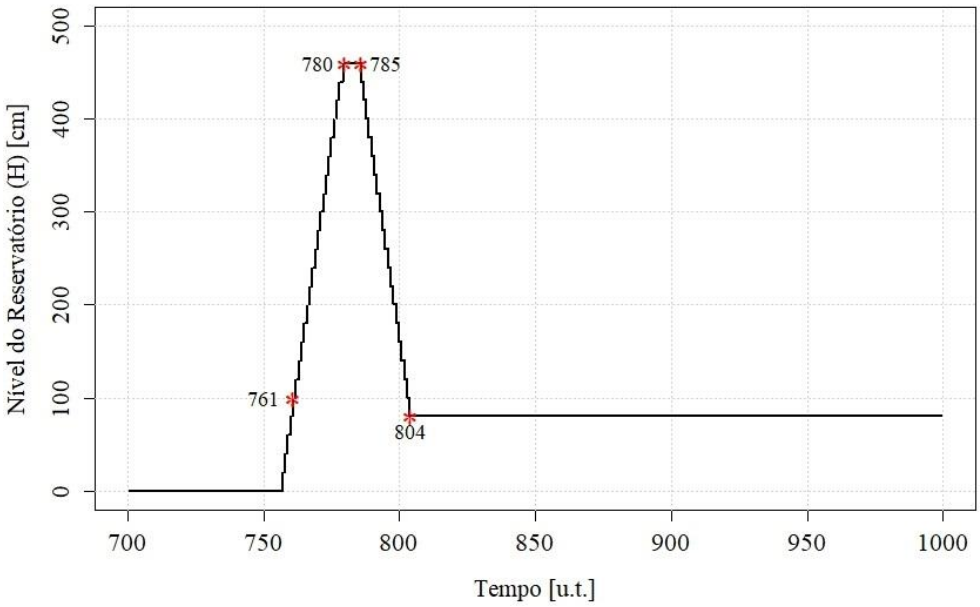


Figura 3.43 - Simulação da variação do nível do reservatório para o modelo de manutenção – detalhamento.

No tempo 756 u.t., a bomba *P2* falha na posição ligada (*P2*=1), e a taxa de variação do reservatório passa a ser de 20 cm/u.t., com isso após 5 u.t., ou seja, em 761 u.t., o nível do reservatório atinge 100 cm, e os componentes em falha (*P1* e *P2*), entram no estado de manutenção, como pode ser observado nas Figura 3.39 e Figura 3.40.

Como durante o tempo de reparo, os componentes permanecem travados na posição que falharam, o nível do reservatório continua aumentando, como pode ser visto na Figura 3.42. Como o efeito do reparo consiste na remoção da condição de travamento e leva o componente para o estado de funcionamento, *ON* ou *OFF*, de acordo com o requerido para o sistema

retornar a região de correto funcionamento, quando a bomba *P2* sai do reparo no tempo de 780 u.t., ela é desligada (*P2* = 0) e com isso não há mais variação do nível do reservatório, até o instante de tempo em que *P1* retorna do reparo, na posição desligada (*P1* = 0), em 785 u.t., em que a taxa de variação do reservatório passar a ser de -20 cm/u.t., desta forma, o reservatório atinge a região *R2* em 11 u.t., isto é, em 804 u.t., e o controlador reestabelece os componentes a sua condição inicial (*P1* = 1, *P2* = 0 e *V* = 1).

Ainda nesta simulação, *P1* e *P2* entram novamente em falha, *P1* travada ligada e *P2* travada desligada e, portanto, não alteram o nível do reservatório. Portanto do ponto de vista da simulação, o modelo de manutenção não apresentou comportamento inadequado.

Etapa 4: Verificação Formal

No Quadro 3.8 estão apresentadas as verificações realizadas para o modelo de manutenção, onde constata-se que o modelo cumpre todas os requisitos definidos na etapa de especificação de propriedades.

Quadro 3.8 – Verificação formal para o modelo de manutenção

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita?**
Verifica a ausência de *deadlock*	`A[] not deadlock`	Sim
Acessibilidade		
Existe um caminho que leve P1, P2 ou V para estado de manutenção.	`E<> expP1.MAIN`	Sim
	`E<> expP2.MAIN`	
	`E<> expV.MAIN`	
Quando H for maior ou igual a 100 cm ou, menor ou igual a -100 cm, os componentes em falha devem entrar no estado de manutenção.	`E<> (controller2.R1 or controller2.R3) and repairP1.M`	Sim
	`E<> (controller2.R1 or controller2.R3) and repairP2.M`	
	`E<> (controller2.R1 or controller2.R3) and repairV.M`	
Quando H estiver entre 100 cm e -100 cm, o reparo deve ser interrompido.	`E<> controller2.R2 and repairP1.S`	Sim
	`E<> controller2.R2 and repairP2.S`	
	`E<> controller2.R2 and repairV.S`	

Continuação do Quadro 3.8 – Verificação formal para o modelo de manutenção.

Descrição Informal	Descrição Formal	Propriedade Satisfeita?
Segurança		
Quando H estiver entre 100 cm e -100 cm, os componentes não podem estar no estado de reparo.	`E[] controller2.R2 and repairP1.M`	Sim
	`E[] controller2.R2 and repairP2.M`	
	`E[] controller2.R2 and repairV.M`	
Não pode haver interrupção do reparo quando H for maior ou igual a 100 cm, ou, menor ou igual a -100 cm.	`E[] (controller2.R1 or controller2.R3) and repairP1.S`	Sim
	`E[] (controller2.R1 or controller2.R3) and repairP2.S`	
	`E[] (controller2.R1 or controller2.R3) and repairV.S`	

Assim como no modelo de falha, as propriedades do modelo físico e modelo de controle reescritas na Página 64 foram verificadas para este modelo, estando apresentadas no Quadro 3.4 (Página 50) e Quadro 3.6 (Página 60), as quais foram devidamente testadas e aceitas.

Capítulo 4

Análise comparativa entre os resultados

No capítulo anterior foi apresentado o desenvolvimento do modelo em autômatos estocásticos híbridos do sistema hidráulico em estudo, a fim de realizar a validação dos modelos foi realizada a simulação e verificação formal clássica, em que se constatou que todos os modelos construídos possuem comportamento esperado e em conformidade com as literaturas consultadas.

Tendo sido verificado o comportamento dos modelos, neste capítulo serão apresentados os resultados obtidos quanto à característica de falha do sistema, isto é, estão apresentadas as distribuições densidade de falha (*pdf*) e densidade acumulada de falha do sistema (*cdf*), e a comparação destes resultados com aqueles obtidos por Marseguerra e Zio (1996), Codetta-Raiteri e Bobbio (2006) e Sakurada (2013). Para a obtenção destes resultados foi utilizada a verificação formal estatística.

Os modelos utilizados para a análise comparativa são: o modelo de falha apresentado na seção 3.3.3, também denominado modelo sem reparo e o modelo de manutenção ou sistema com reparo, apresentado na seção 3.3.4.

4.1 Resultados Comparativos

O UPPAAL STRATEGO, ferramenta computacional utilizada neste trabalho, possui em sua plataforma o módulo UPPAAL SMC que consiste em um verificador formal estatístico, em que por meio da *query* de estimação de probabilidade, detalhado no Quadro 2.3, foi possível a obtenção da *pdf* e *cdf* do sistema. Neste caso, o usuário pode configurar o grau de confiança e a incerteza dos resultados, e o verificador faz a quantidade de simulações necessárias para obtenção do resultado.

Como já mencionado, os resultados obtidos foram comparados com aqueles existentes na literatura para o mesmo sistema utilizando-se das mesmas considerações, sendo eles:

- *Marseguerra e Zio (1996):*Esses autores realizaram simulação de Monte Carlo, para diversos cenários envolvendo o sistema em estudo (ver seção 3.1 – Página 38), em que todos os cenários foram simulados 100 mil vezes. Os resultados obtidos foram a *pdf* e *cdf* para o sistema sem reparo e com reparo, aqui denominados: modelo de falha e modelo de manutenção. Porém este trabalho não apresenta os dados

numéricos, apenas os resultados gráficos, portanto a comparação será quanto à forma da curva e os valores finais obtidos na *cdf*.

- *Codetta-Raiteri e Bobbio (2006):* Estes autores realizaram a modelagem por Redes de Petri para a análise do sistema em estudo, sendo que utilizaram duas técnicas: Redes de Petri Estocásticas Generalizadas (*GSPN)* e Redes de Petri Fluidas Estocásticas (*FSPN).* A diferença entre as duas técnicas é que a *GSPN* permite apenas variáveis discretas e a *FSPN* utiliza variáveis discretas e contínuas. Os resultados obtidos, para 10 mil simulações, foram a *cdf* para o sistema sem reparo e com reparo, em que os resultados obtidos pela *GSPN* foram validados pela técnica *FSPN*. Visto que este trabalho apresenta os dados numéricos foi possível realizar a correlação entre os resultados.
- *Sakurada (2013):* O autor desenvolveu uma metodologia para análise de confiabilidade dinâmica, denominada ACoDi, sendo que a implementação computacional foi realizada utilizando o software *Matlab.* O resultado obtido foi a *cdf* do sistema sem reparo, o autor comparou os valores obtidos com os aqueles obtidos por Codetta-Raiteri e Bobbio (2006), e também realizou 10 mil simulações. Este trabalho também apresenta os dados numéricos obtidos, portanto também foi possível realizar a correlação entre os resultados.

O Quadro 4.1 sintetiza quais resultados serão comparados com cada literatura mencionada acima e o Quadro 4.2 resume qual a metodologia foi empregada por cada um deles.

Quadro 4.1 – Resultados comparados com a literatura.

	pdf	**cdf (sem reparo)**	**cdf (com reparo)**
Marseguerra e Zio (1996)	Sim (Visual)	Sim (Visual)	Sim (Visual)
Codetta-Raiteri e Bobbio (2006)	Não	Sim	Sim
Sakurada (2013)	Não	Sim	Não

Quadro 4.2 – Metodologias utilizadas na literatura consultada.

Autor	**Metodologia**
Marseguerra e Zio (1996)	Simulação de Monte Carlo
Codetta-Raiteri e Bobbio (2006)	Redes de Petri (GSPN e FSPN)
Sakurada (2013)	ACoDi

4.2 Resultados obtidos para o modelo de falha

Para a obtenção dos resultados do modelo de falha, ou sistema sem reparo, foram utilizadas as *queries* apresentadas na Tabela 4.1, sendo que o grau de confiança e a incerteza dos resultados foram estimados, servindo de parâmetros de entrada para a verificação formal estatística, e o número de simulações foi calculada pelo verificador de acordo com tais parâmetros.

Tabela 4.1 – *Queries* e configurações para verificação formal estatística do modelo de falha.

Query	**Descrição**	**Grau de Confiança**	**Incerteza**	**Número de Simulações**
`Pr[<=3000] (<>tank.HLP)`	Qual a probabilidade de o reservatório atingir o nível máximo (transbordar) em um tempo de missão de 3000 u.t. (1000 h)?	99%	0,01	16660
`Pr[<=3000] (<>tank.HLV)`	Qual a probabilidade de o reservatório atingir o nível mínimo (esvaziar) em um tempo de missão de 3000 u.t. (1000 h)?	99%	0,01	6677

A distribuição acumulada de falha obtida para o transbordamento e para o esvaziamento, juntamente com os resultados obtidos a partir das Redes de Petri e Metodologia ACoDi estão apresentados nas Figura 4.1 e Figura 4.2.

O sistema falha mais por transbordamento, com probabilidade de falha de cerca de 50%, do que por esvaziamento, que tem uma probabilidade de falha de aproximadamente 12%. Esta condição já era esperada, visto que a combinação de condições de operações dos componentes mecânicos apresentam mais configurações que levam ao transbordamento do que ao esvaziamento, como pode ser visto na Tabela 3.2 (Página 40).

Por causa disso, as escalas de probabilidade de falha acumulada nos gráficos (eixo vertical) são diferentes. Os gráficos foram gerados no software RStudio, com diferença nas escalas com o intuito de facilitar a visualização do gráfico referente ao esvaziamento.

A partir desses gráficos, é possível concluir que um sistema hidráulico composto por duas bombas e uma válvula, todos com mesma vazão, e sem uma politica de reparo, ao final de 1000 h, aproximadamente 42 dias, apresentará probabilidade de falha em torno de 62%, sendo 50% em transbordamento e 12% em esvaziamento, tendo uma probabilidade de continuar operando após este tempo de 38%.

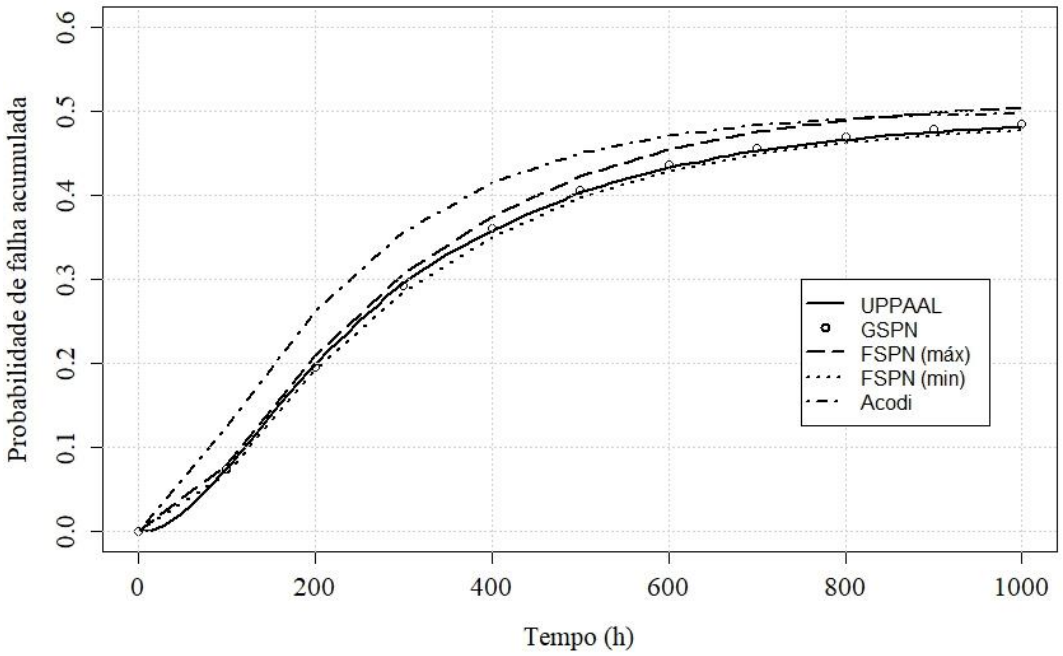


Figura 4.1 – Distribuição acumulada de falha para transbordamento – modelo de falha.

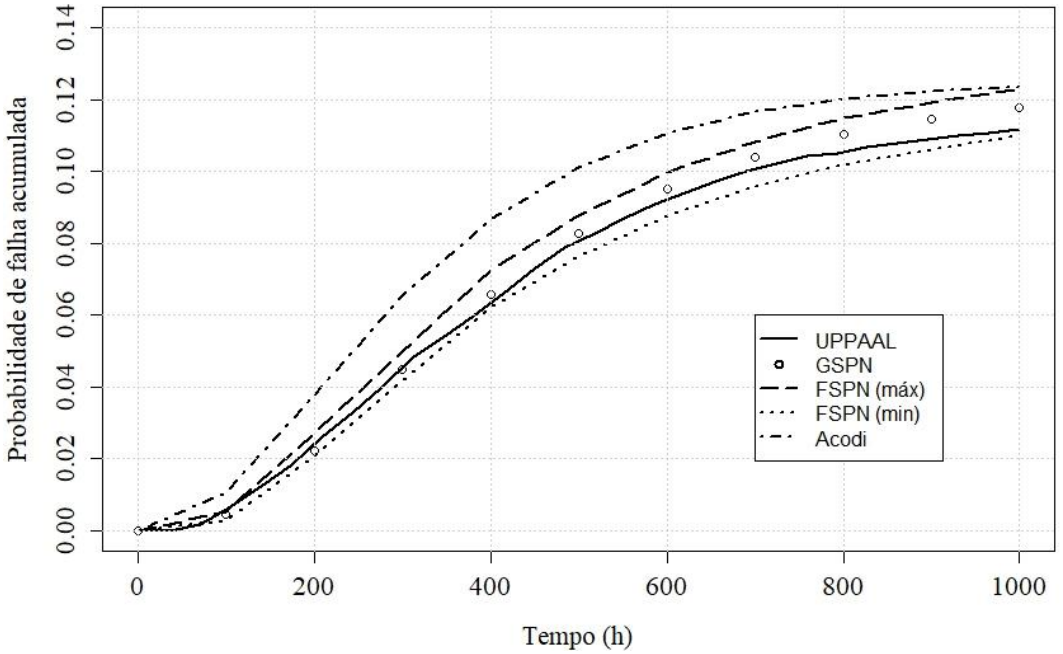


Figura 4.2 - Distribuição acumulada de falha para esvaziamento – modelo de falha.

A Figura 4.3 ilustra os resultados obtidos por Marseguerra e Zio (1996), através de simulação de Monte Carlo, para a *cdf* de transbordamento e esvaziamento do reservatório. Como já mencionando, este autor não disponibilizou os dados numéricos, sendo assim, a comparação foi visual, onde pode-se constatar que tanto a forma da curva quanto o valor final obtido para a probabilidade de falha acumulada estão bem próximos.

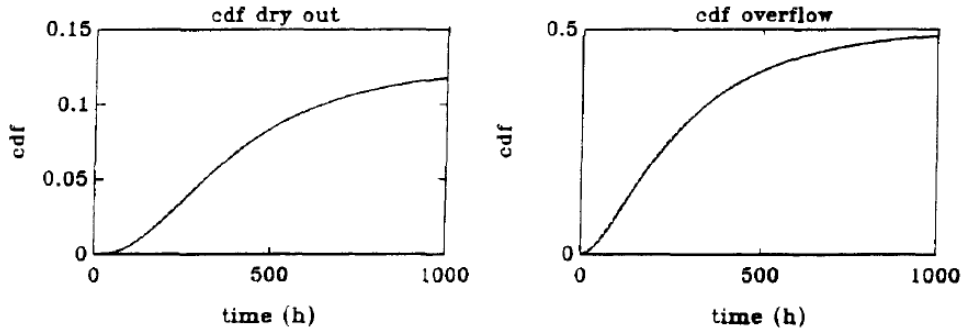


Figura 4.3 – Densidade acumulada de falha para esvaziamento (*dry out)* e transbordamento (*overflow)* obtidos pela simulação de Monte Carlo – modelo de falha. Fonte: Marseguerra e Zio (1996)

Algumas considerações podem ser realizadas a respeito da densidade acumulada de falha do sistema:

1. Percebe-se que as curvas dos resultados obtidos no UPPAAL são muito semelhantes daquelas obtidas pelas Redes de Petri, em especial a técnica GSPN;
2. A curva que apresenta maior dispersão é a referente a metodologia ACoDi, sendo que as maiores diferenças estão nos valores iniciais e à medida que o tempo se aproxima do tempo de missão a diferença é reduzida;
3. Os valores finais da probabilidade acumulada de falha para cada uma das metodologias podem ser visualizados na Tabela 4.2, onde se verifica uma pequena dispersão dos resultados. É possível observar que a maior diferença dos resultados finais obtidos entre o UPPAAL com as demais metodologias, para transbordamento foi com a técnica FSPN(máx), sendo de 2,3882% e para esvaziamento com a metodologia ACoDi, sendo de 1,1944%.

Tabela 4.2 – Resultados finais da densidade acumulada de falha – modelo de falha.

	cdf Transbordamento (1)	**cdf Esvaziamento (2)**	**Dispersão em relação ao UPPAAL**	
			(1)	**(2)**
Simulação de Monte Carlo	≈ 50%	≈ 12%	1,9148%	0,8423%
GSPN	48,5200%	11,7689%	0,4348%	0,6112%
FSPN(máx)	50,4734%	12,3190%	2,3882%	1,1613%
FSPN(mín)	47,7266%	10,9810%	0,3586%	0,1767%
ACoDi	49,7369%	12,3521%	1,6517	1,1944%
UPPAAL	48,0852%	11,1577%	-	

Os valores dos resultados, tanto através do UPPAAL quanto das Redes de Petri e metodologia ACoDi, estão organizados na Tabela 4.3 e Tabela 4.4, que se referem aos valores para a probabilidade acumulada de falha para transbordamento e esvaziamento, respectivamente.

Tabela 4.3 – Dados da *cdf* para o transbordamento – modelo de falha.

Tempo (h)	ACoDi	GSPN	FSPN(max)	FSPN(min)	UPPAAL
0	0	0	0	0	0
100	0,123925	0,074208	0,079228	0,068572	0,074028
200	0,261320	0,195182	0,209277	0,191723	0,197558
300	0,355973	0,292146	0,306257	0,284943	0,295233
400	0,414377	0,359876	0,373996	0,350404	0,356723
500	0,449766	0,405374	0,422347	0,397253	0,403099
600	0,470524	0,435689	0,454625	0,428575	0,432386
700	0,482860	0,455953	0,475018	0,448382	0,453236
800	0,490350	0,469595	0,489827	0,462773	0,465658
900	0,494766	0,478857	0,498549	0,471251	0,474910
1000	0,497369	0,485200	0,504734	0,477266	0,480852

Tabela 4.4 - Dados da *cdf* para o esvaziamento – modelo de falha.

Tempo (h)	ACoDi	FSPN(max)	FSPN(min)	GSPN	UPPAAL
0	0	0	0	0	0
100	0,010633	0,005355	0,002845	0,004463	0,005528
200	0,037664	0,027037	0,020963	0,022077	0,023964
300	0,065436	0,04989	0,04151	0,044846	0,045118
400	0,086567	0,072385	0,062215	0,065827	0,06331
500	0,101232	0,087613	0,076387	0,082568	0,080715
600	0,110622	0,099597	0,087603	0,095014	0,092069
700	0,116519	0,108157	0,095643	0,103939	0,100518
800	0,120112	0,114853	0,101947	0,110227	0,105477
900	0,122330	0,119074	0,105926	0,114622	0,108886
1000	0,123521	0,12319	0,10981	0,117689	0,111577

A partir desses valores foi realizada a correlação dos resultados, através do coeficiente de determinação R^2, que mede a porcentagem que a variável *y* explica a variável *x*. A Figura 4.4 ilustra um quadro de correlação para a probabilidade acumulada de falha para o transbordamento entre as Redes de Petri, metodologia ACoDi e UPPAAL. A partir deste quadro é possível observar que os valores obtidos por Redes de Petri e UPPAAL têm uma forte correlação com valores R^2 maiores que 99,9%. Já a metodologia ACoDi, possui correlação mais baixa em relação a todas as outras metodologias, ainda sim, ultrapassa 98%.

A Figura 4.5 ilustra o quadro de correlação para a probabilidade acumulada de falhas para o esvaziamento, onde se pode tirar as mesmas conclusões daquelas feitas para o transbordamento, ou seja, os valores obtidos por Redes de Petri e UPPAAL têm uma forte correlação com valores R^2 maiores que 99,8%, já a metodologia ACoDi, possui correlação mais baixa em relação a todas as outras metodologias, ainda sim, ultrapassa 97%.

	Acodi	FSPN(máx)	FSPN(min)	GSPN	UPPAAL
Acodi	1	0,9849	0,9818	0,9834	0,9852
FSPN(máx)	0,9849	1	0,9998	0,9999	0,9999
FSPN(min)	0,9818	0,9998	1	0,9999	0,9998
GSPN	0,9834	0,9999	0,9999	1	0,9999
UPPAAL	0,9852	0,9999	0,9998	0,9999	1

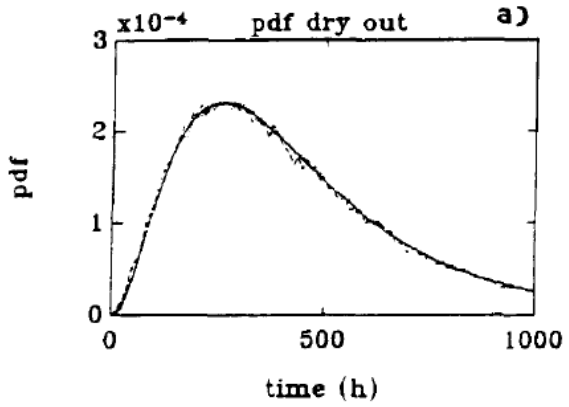
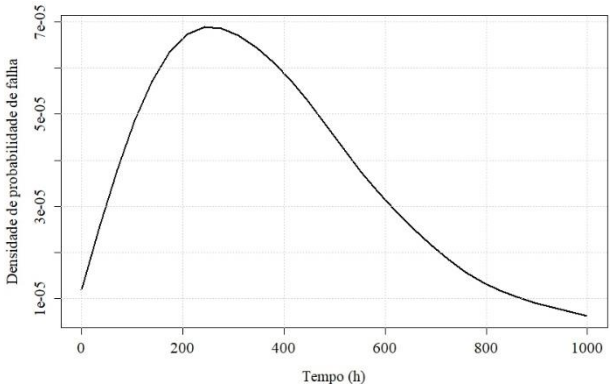
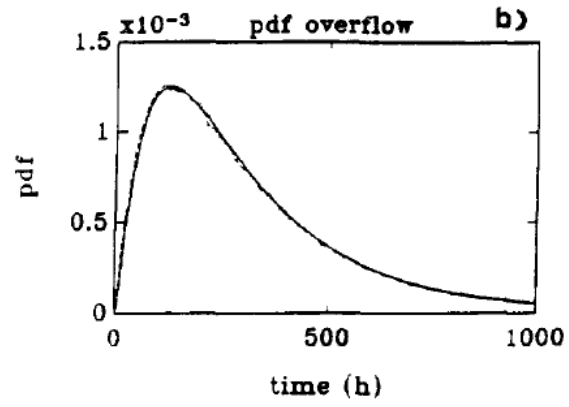
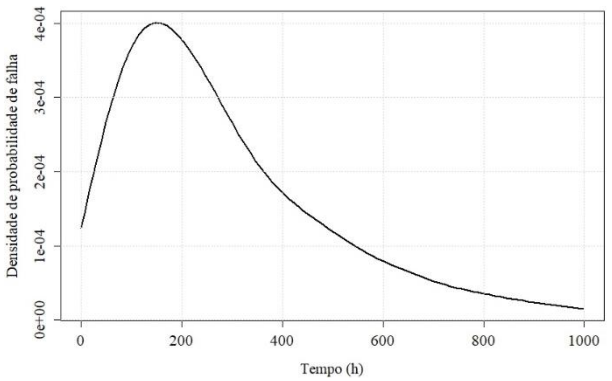
Figura 4.4 – Correlação entre os valores obtidos para transbordamento – modelo de falha.

	Acodi	FSPN(máx)	FSPN(min)	GSPN	UPPAAL
Acodi	1	0,9858	0,9799	0,9788	0,9838
FSPN(máx)	0,9858	1	0,9994	0,9989	0,9992
FSPN(min)	0,9799	0,9994	1	0,9987	0,9990
GSPN	0,9788	0,9989	0,9987	1	0,9994
UPPAAL	0,9838	0,9992	0,9990	0,9994	1

Figura 4.5 - Correlação entre os valores obtidos para o esvaziamento – modelo de falha.

Quanto a distribuição densidade de falha, foi possível realizar comparação em relação a forma da curva com aquela apresentada por Marseguerra e Zio (1996). O Quadro 4.3 ilustra a comparação, tanto para o esvaziamento como para o transbordamento, da *pdf* do sistema entre a Simulação de Monte Carlo e UPPAAL, observa-se que a tendência da distribuição obedece à função Lognormal.

Quadro 4.3 – Comparativo da distribuição densidade de falha – modelo de falha.

Simulação de Monte Carlo	*UPPAAL*
Esvaziamento	
	
Transbordamento	
	

A partir da análise comparativa dos resultados, em que se obtiveram resultados semelhantes a outras metodologias; da simulação e verificação formal, em que se observou comportamento adequando do sistema hidráulico. Conclui-se que, para o modelo de falha, a metodologia adotada neste trabalho foi satisfatória para a obtenção da característica de falha do sistema, logo, foi possível considerar válida a metodologia empregada e os modelos construídos para representação do sistema em estudo.

4.3 Resultados obtidos para o modelo de manutenção

Para a obtenção dos resultados do modelo de manutenção, ou sistema com reparo, foi utilizado um tempo de missão de 500h, como feito pelas literaturas consultadas e foram utilizadas as mesmas *queries* da seção anterior. Porém como a probabilidade de falha diminui consideravelmente quando inserida a manutenção corretiva, para se obter a curva da *cdf* foi necessária diminuir a incerteza dos resultados. As configurações utilizadas para o modelo de manutenção estão resumidas na Tabela 4.5.

Tabela 4.5 – *Queries* e configurações para verificação formal estatística do modelo de manutenção.

Query	**Descrição**	**Grau de Confiança**	**Incerteza**	**Número de Simulações**
`Pr[<=1500] (<>tank.HLP)`	Qual a probabilidade de o reservatório atingir o nível máximo (transbordar) em um tempo de missão de 1500 u.t. (500 h)?	99%	0,001	238334
`Pr[<=1500] (<>tank.HLV)`	Qual a probabilidade de o reservatório atingir o nível mínimo (esvaziar) em um tempo de missão de 1500 u.t. (500 h)?	99%	0,0001	820870

As distribuições acumuladas de falha obtida para o transbordamento e para o esvaziamento juntamente com os resultados obtidos a partir das Redes de Petri estão apresentadas nas Figura 4.6 e Figura 4.7. Para estes gráficos também se optou por manter a escala de probabilidade de falha acumulada diferente entre os gráficos, para melhor visualização dos resultados obtidos para o esvaziamento.

É possível observar que, inserindo-se uma política de reparo ao sistema hidráulico, a probabilidade de falha diminui consideravelmente. Este fato pode ser verificado comparando-se as Figura 4.2 e Figura 4.6 para o transbordamento, em que para o tempo de 500h no modelo sem reparo a probabilidade de falha era é aproximadamente 35%, inserindo-se o reparo passou para menos de 4%. Já nas Figura 4.2 e Figura 4.7 é possível comparar o esvaziamento para o sistema sem e com reparo, em que a probabilidade de falha passou de aproximadamente 8% para valores inferiores a 0,2%, para um tempo de 500h.

Logo, é possível verificar que a manutenção traz ganhos significativos ao sistema quanto à minimização das falhas e, portanto, é válido realizar uma análise de custos, disponibilidade e criticidade do sistema para avaliar o custo-benefício desta prática. Como o sistema em estudo é um problema teórico, neste trabalho não é possível realizar tal avaliação,

mas em sistemas reais está análise é de suma importância, para um eficaz gerenciamento de manutenção.

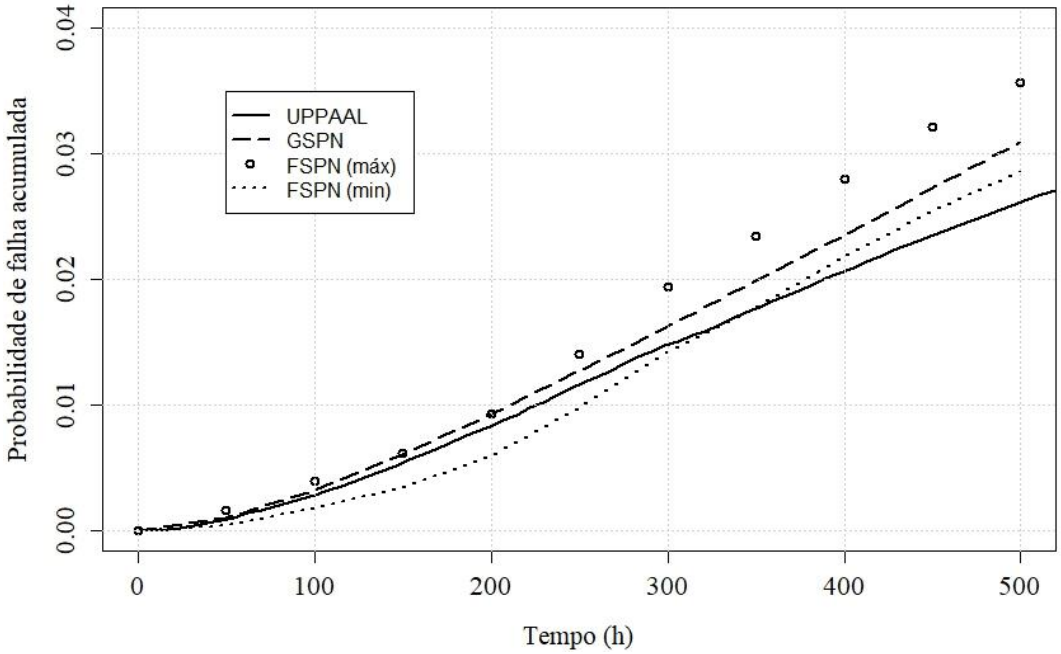


Figura 4.6 – Distribuição acumulada de falha para transbordamento – modelo de manutenção.

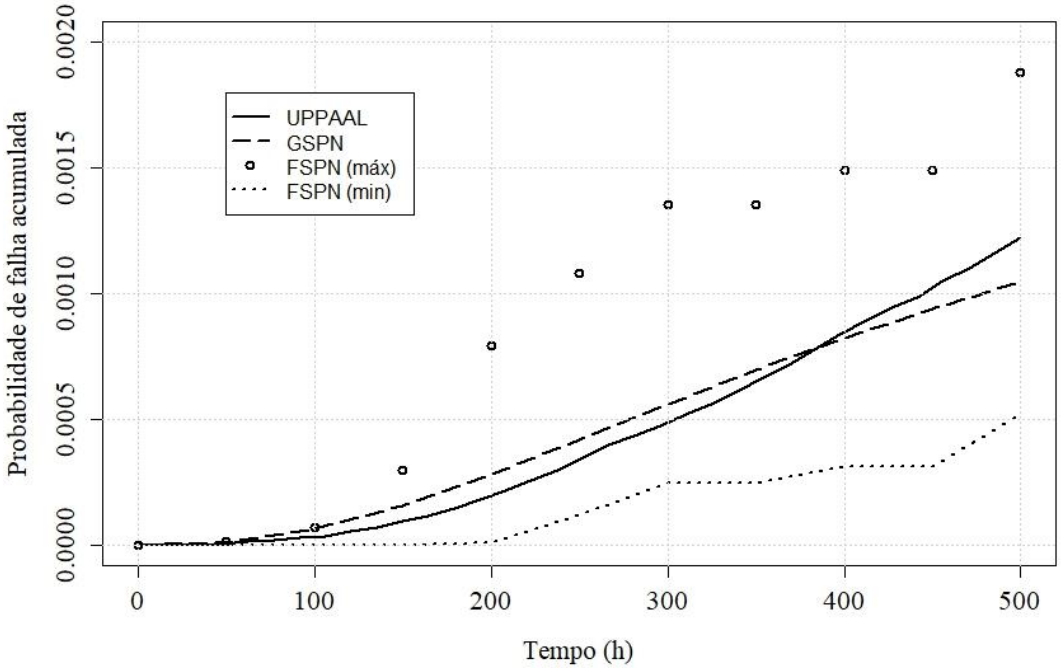


Figura 4.7 - Distribuição acumulada de falha para esvaziamento – modelo de manutenção.

A Figura 4.8 ilustra os resultados obtidos por Marseguerra e Zio (1996), através de simulação de Monte Carlo, para a *cdf* de transbordamento e esvaziamento do reservatório.

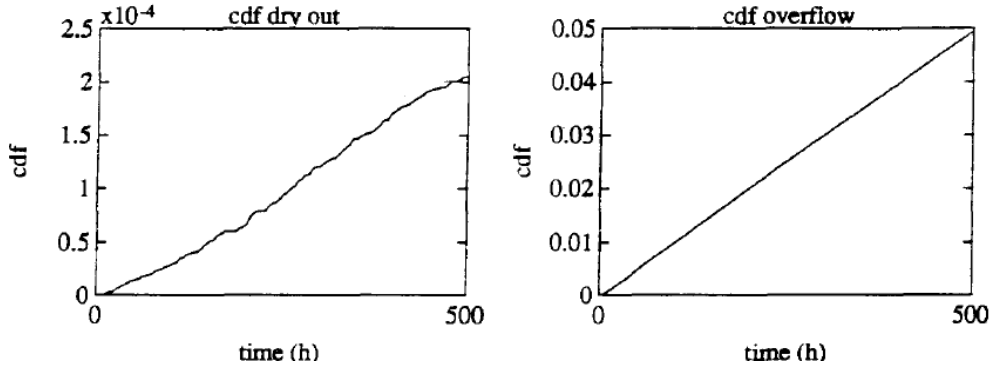


Figura 4.8 – Densidade acumulada de falha para esvaziamento (*dry out)* e transbordamento (*overflow)* obtidos pela simulação de Monte Carlo – modelo de manutenção. Fonte: Marseguerra e Zio (1996)

Algumas considerações podem ser realizadas a respeito da densidade acumulada de falha do sistema com reparo:

1. Observa-se que para o transbordamento, que apresenta maior número de falhas do que o esvaziamento, não há grande dispersão na forma das curvas obtidas para cada um dos métodos. Já para o esvaziamento, observa-se que para o tempo de missão utilizado (500 h), os resultados ainda estão bastante dispersos, em relação ao formato das curvas, embora novamente os resultados do UPPAAL estejam muito próximos do GSPN;
2. Na Tabela 4.6 estão apresentados os valores finais da probabilidade de falha acumulada para cada uma das metodologias e observa-se que a variação dos valores não é tão significante.

Tabela 4.6 – Resultados finais da densidade acumulada de falha – modelo de manutenção

	cdf Transbordamento (1)	**cdf Esvaziamento (2)**	**Dispersão em relação ao UPPAAL**	
			(1)	**(2)**
Simulação de Monte Carlo	≈ 5%	≈ 0,02%	2,3903%	0,1023%
GSPN	3,0860%	0,1045%	0,4763%	0,0178%
FSPN(máx)	3,5612%	0,1879%	0,9515%	0,0656%
FSPN(mín)	2,8588%	0,0521%	0,2491%	0,0702%
UPPAAL	2,6097%	0,1223%	-	

Os valores dos resultados, tanto do UPPAAL quanto das Redes de Petri, estão organizados nas Tabela 4.7 e Tabela 4.8, que se referem aos valores para a probabilidade acumulada de falha para transbordamento e esvaziamento do sistema com reparo, respectivamente.

Tabela 4.7 - Dados da *cdf* para o transbordamento – modelo de manutenção.

Tempo (h)	FSPN(min)	GSPN	FSPN(max)	UPPAAL
0	0	0	0	0
50	0,000380	0,000967	0,001620	0,000903
100	0,001844	0,003168	0,003956	0,002832
150	0,003442	0,006015	0,006158	0,005396
200	0,005891	0,009238	0,009309	0,008343
250	0,009762	0,012682	0,014038	0,011585
300	0,014259	0,016254	0,019341	0,014780
350	0,017787	0,019893	0,023413	0,017659
400	0,021807	0,023558	0,027993	0,020630
450	0,025474	0,027221	0,032126	0,023517
500	0,028588	0,030860	0,035612	0,026097

Tabela 4.8 - Dados da *cdf* para o esvaziamento – modelo de manutenção.

Tempo (h)	FSPN(min)	GSPN	FSPN(max)	UPPAAL
0	0	0	0	0
50	0,000000	0,000010	0,000015	0,000007
100	0,000000	0,000062	0,000070	0,000033
150	0,000000	0,000156	0,000296	0,000095
200	0,000008	0,000280	0,000792	0,000196
250	0,000120	0,000419	0,001080	0,000343
300	0,000246	0,000561	0,001354	0,000488
350	0,000246	0,000697	0,001354	0,000652
400	0,000312	0,000825	0,001488	0,000847
450	0,000312	0,000941	0,001488	0,001023
500	0,000521	0,001045	0,001879	0,001223

Assim como para o modelo de falha foi feita a correlação dos resultados para o modelo de manutenção através do coeficiente de determinação R^2. A Figura 4.9 ilustra um quadro de correlação entre as Redes de Petri e o UPPAAL para a probabilidade acumulada de falha para o transbordamento, a partir deste quadro é possível observar que, embora mais baixa do que para o modelo de falha, a correlação para o transbordamento ainda se mantém alta, com valores R^2 maiores que 98,8%,

A correlação para os dados de esvaziamento é ilustrada na Figura 4.10, que evidencia a dispersão dos resultados já comentada anteriormente.

	FSPN(mín)	GSPN	FSPN(máx)	UPPAAL
FSPN(mín)	1	0,9926	0,9973	0,9883
GSPN	0,9926	1	0,9973	0,9986
FSPN(máx)	0,9973	0,9973	1	0,9958
UPPAAL	0,9883	0,9986	0,9958	1

Figura 4.9 - Correlação entre os valores obtidos para transbordamento – modelo de manutenção.

	FSPN(mín)	GSPN	FSPN(máx)	UPPAAL
FSPN(mín)	1	0,9177	0,8506	0,9457
GSPN	0,9177	1	0,9379	0,9788
FSPN(máx)	0,8506	0,9379	1	0,8691
UPPAAL	0,9457	0,9788	0,8691	1

Figura 4.10 - Correlação entre os valores obtidos para esvaziamento – modelo de manutenção

A partir de todas as comparações realizadas, foi possível observar que todos os resultados obtidos no UPPAAL foram semelhantes aos resultados das demais metodologias, sendo que a diferença entre os resultados não passou de 2,5%. Tendo em vista tais resultados, bem como a verificação do comportamento do modelo através da simulação e verificação formal apresentada no Capítulo 3 e possível concluir que a metodologia utilizada e os modelos construídos podem ser validados.

Foi observado também, que tanto para o modelo de falha como para o modelo de manutenção os resultados obtidos com o UPPAAL foram muito similares com aqueles obtidos pela técnica GSPN.

A partir dos parâmetros de falhas e da precisão dos resultados obtidos, é possível auferir que a metodologia utilizada neste trabalho é adequada para a análise de confiabilidade, portanto, no capítulo que segue, será abordada com mais detalhe a análise de confiabilidade e inclusão de manutenção preventiva.

Capítulo 5

Análise de Confiabilidade do Sistema

No capítulo anterior foi realizada a comparação da característica de falha do sistema modelado com os existentes na literatura, sendo mais um critério para verificar se a metodologia empregada é adequada para análise de confiabilidade. Tendo em vista a precisão dos resultados obtidos, pode-se concluir que a metodologia empregada e os modelos construídos são adequados para tal estudo. Portanto, neste capítulo, serão realizadas modificações dos parâmetros de falha do sistema, considerando a função de distribuição de probabilidades mais representativas com sistemas mecânicos.

Também, por meio da aplicação da verificação formal, será feita a predição dos dados de disponibilidade do sistema, a fim de estimar o tempo médio entre manutenções (*MTBM*) ótimo do modelo obtido, visando a obtenção da máxima disponibilidade do sistema.

5.1 Modificação da distribuição de probabilidade de falha

Como visto na seção 2.1.3, as principais distribuições de probabilidade empregadas no estudo de confiabilidade de sistemas são a distribuição exponencial, distribuição de Weibull e distribuição Lognormal.

A distribuição exponencial, até então utilizada neste trabalho, é a mais empregada devido sua simplicidade e facilidade de utilização, além de representar a maior parte da vida de um equipamento, que é a vida útil, sendo a função mais adequada para representar sistemas que possuem falhas aleatórias, como os componentes elétricos.

O sistema em estudo, um sistema hidráulico, composto por 3 componentes mecânicos, é um sistema que sofre desgaste com o tempo, e, portanto, é melhor representado pela distribuição de Weibull, que tem como características representar de forma satisfatória todas as regiões da curva da banheira, apenas variando o parâmetro de forma, escala e localização da distribuição. Já a manutenção, é adequadamente representada pela distribuição Lognormal.

Portanto, para a estimação do *MTBM*, optou-se por modificar a distribuição de probabilidade tanto da falha quanto da manutenção do sistema, a fim de tornar o sistema mais representativo aos sistemas reais, ou seja, para calcular o tempo médio entre falhas dos componentes, a distribuição exponencial foi substituída pela distribuição de Weibull, e para o tempo de reparo foi substituída pela distribuição Lognormal.

No UPPAAL, as funções estatísticas já implementadas são a distribuição uniforme, utilizada nas *invariantes* e através da função *random()*, e a distribuição exponencial, utilizada por meio do campo *rate of exponential*. Portanto, para utilização das demais distribuições de probabilidade, foi necessário implementá-las.

Neste trabalho, foram utilizadas as funções implementadas por Santos (2017) para a geração de números aleatórios baseados na distribuição de Weibull para o tempo até a falha dos componentes e baseadas na distribuição Lognormal para o tempo de manutenção. Tais geradores foram implementados nas declarações globais do UPPAAL, de modo que podem ser utilizados por qualquer modelo da rede de autômatos e a chamada dessas funções é feita através do campo *update* das transições dos autômatos.

Para inserção da distribuição de Weibull para o cálculo do tempo médio até a falha dos componentes, foi necessário realizar algumas modificações no modelo de falha, apresentado na seção 3.3.3. Foram substituídos os modelos *ExpP1, ExpP2 e ExpV* pelos modelos *WeibullP1, WeibullP2* e *WeibullV*, que podem ser vistos nas Figura 5.1, Figura 5.2 e Figura 5.3, os demais modelos não sofreram modificações.

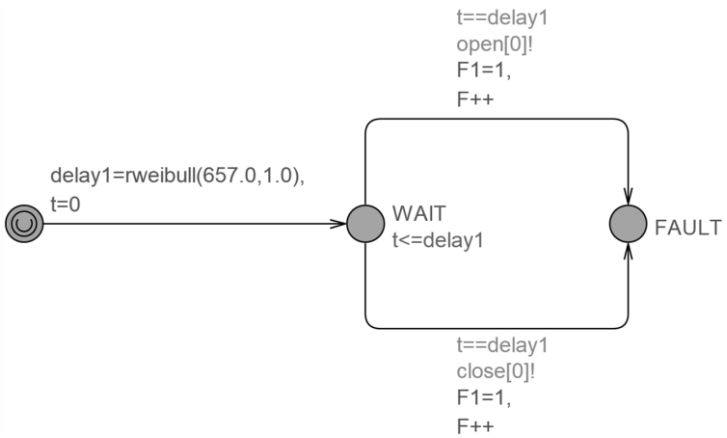


Figura 5.1 – Autômato *WeibullP1* para o modelo de falha.

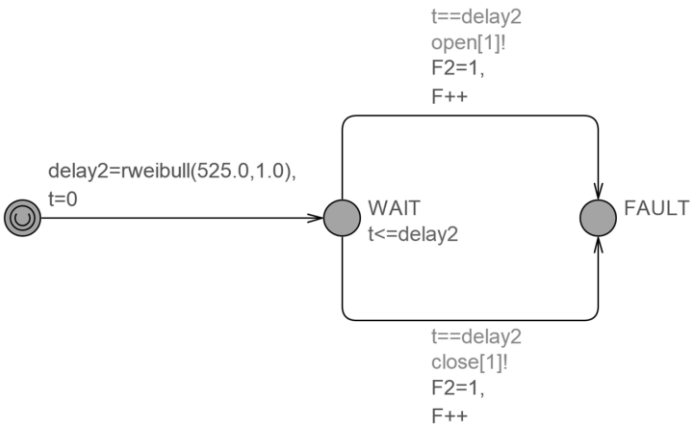


Figura 5.2 – Autômato *WeibullP2* para o modelo de falha.

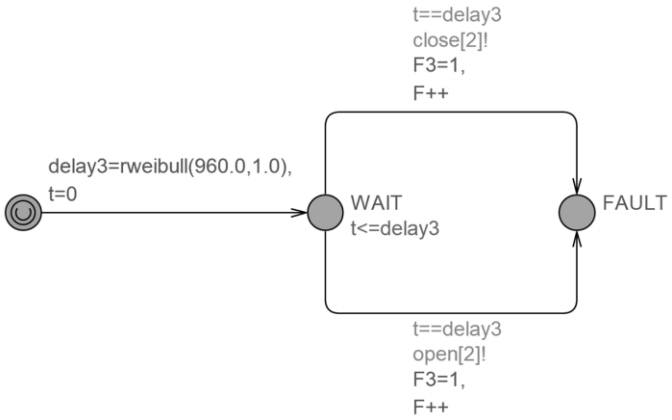


Figura 5.3 – Autômato *WeibullV* para o modelo de falha.

Nestes modelos, a primeira transição que a rede de autômatos realiza é devido ao estado *urgente*. Nesta transição os valores de *delay1*, *delay2 e delay3* são calculados através da função *rweibull*. Esta função gera um número aleatório de acordo com a distribuição de Weibull e seus parâmetros de escala (η) e forma (β), e também reinicializa o relógio *t*. O autômato permanece no estado *WAIT* até que o tempo seja igual ao valor do *delay*, atingindo este tempo o autômato pode seguir dois caminhos: a transição que leva a falha do componente ligado ou a transição que leva a falha do componente desligado.

Visto que o sistema em estudo é um modelo padrão criado para análise de confiabilidade dinâmica, o mesmo não possui um histórico de falhas, necessário para a obtenção do parâmetro de escala e forma da distribuição de Weibull. Portanto, optou-se em assumir o tempo médio entre falhas definido na Tabela 3.1 (Página 38) como o parâmetro de escala, visto que, segundo Lafraia (2001), o mesmo está relacionando com a vida característica do sistema, ou seja, este parâmetro está diretamente relacionado com o tempo de vida dos componentes.

Para o modelo de falha, o parâmetro de forma utilizado foi $\beta = 1$, assim, a distribuição de Weibull está representando a vida útil do componente, ou seja, deve ser equivalente a função exponencial. E portanto, os resultados obtidos para ambas as distribuições devem ser equivalentes.

As Figura 5.4 e Figura 5.5 mostram a distribuição acumulada de falha de transbordamento e esvaziamento para o modelo de falha, tanto para o tempo médio de falha representado pela distribuição exponencial (resultado obtido e apresentado no capítulo 4) quanto pela distribuição de Weibull, com parâmetro de escala igual aos tempos médio para falha e parâmetro de forma igual a 1.

Como esperado, os resultados foram aproximados para as duas distribuições, sendo que a correlação dos dados para o transbordamento foi de R^2 igual a 0,9998 e para o esvaziamento de R^2 igual a 0,9992. Logo, a alteração da distribuição de probabilidade para a falha dos componentes foi adequada, tornando possível a análise do sistema, não apenas na vida útil, mas em todas as regiões da curva da banheira.

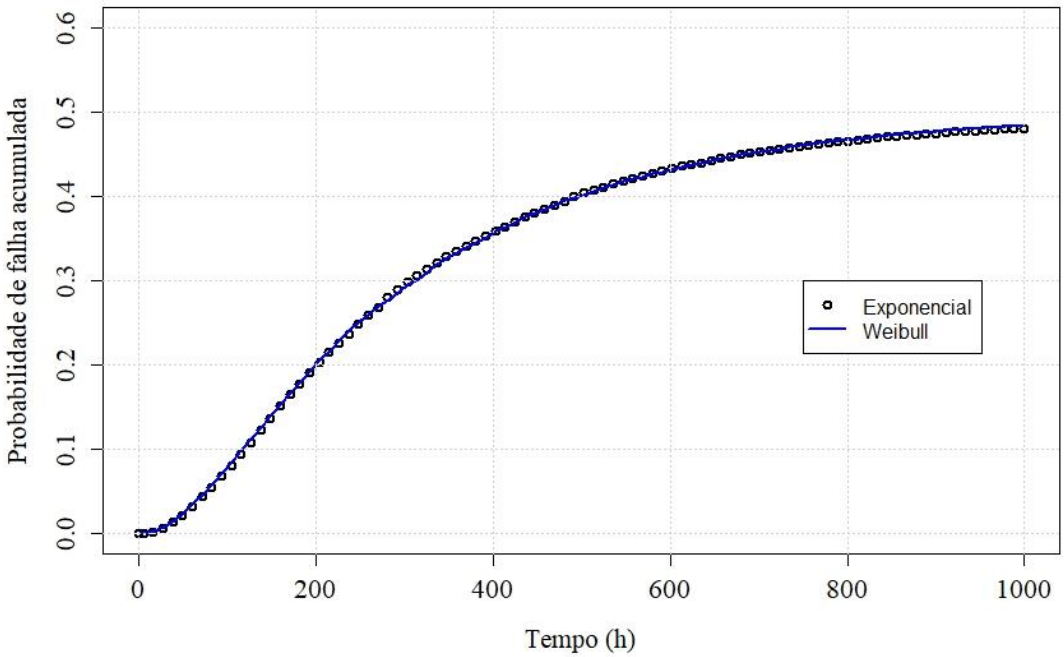


Figura 5.4 – Comparação entre a distribuição exponencial e distribuição de Weibull para densidade acumulada de falha para o transbordamento.

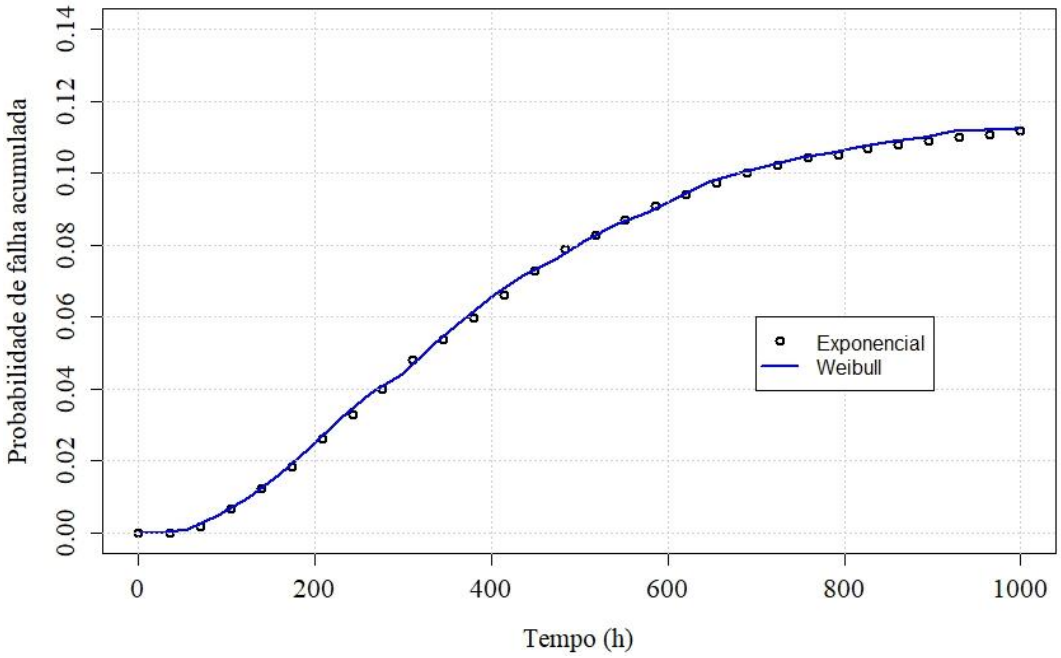


Figura 5.5 - Comparação entre a distribuição exponencial e distribuição de Weibull para densidade acumulada de falha para o esvaziamento.

5.2 Estimação do MTBM do sistema

A maioria dos sistemas reais são reparáveis, ou seja, após a falha passam por reparo e voltam ao seu estado de funcionamento. Em muitos casos se torna interessante realizar uma intervenção antes da ocorrência de falha, e assim reduzir os prejuízos causados por uma

possível falha, como é o caso dos sistemas críticos, em que uma falha no sistema pode causar perdas catastróficas, como perdas humanas ou de recursos financeiros.

A inserção de uma política de manutenção preventiva afeta diretamente a disponibilidade do sistema, visto que durante a intervenção preventiva o sistema torna-se indisponível. Portanto, é importante estimar um tempo ótimo entre manutenções de modo a minimizar a ocorrência de falhas mantendo o sistema o menor tempo possível indisponível.

Partindo dessa problemática, para o sistema em estudo foi realizada a análise de disponibilidade, a fim de determinar o intervalo entre manutenções preventivas mais adequando para este sistema hidráulico. Para tanto, foram realizadas algumas considerações acerca do comportamento do sistema:

- O tempo médio entre falhas dos componentes mecânicos obedece a distribuição de Weibull;
- O tempo médio de manutenção corretiva e de manutenção preventiva obedece a distribuição Lognormal;
- A manutenção corretiva ocorre apenas se o nível do reservatório não está dentro da região de correto funcionamento (HLA≤H≤HLB);
- Quando o processo de manutenção corretiva é habilitado, todo o sistema passa por manutenção e não apenas os componentes em estado de falha;
- Quando o processo de manutenção preventiva é habilitado, todo o sistema entra em manutenção;
- Durante o tempo de reparo e de manutenção preventiva, o sistema encontra-se indisponível;
- O efeito do reparo consiste na remoção da condição de travamento e leva o componente para o seu estado de funcionamento inicial e o reservatório para o nível H=0;
- A manutenção preventiva é perfeita, ou seja, reestabelece o sistema a sua condição inicial;
- Os níveis máximo e mínimo do reservatório são de +5 m e -5 m, respectivamente.

Etapa 1: Modelagem do Sistema

Para realizar a análise de disponibilidade do sistema, utilizou-se como base o modelo de manutenção apresentado na seção 3.3.4, onde foram realizadas modificações para inserção da manutenção preventiva e adequação do comportamento do sistema aos itens supracitados.

A Figura 5.6 ilustra o modelo *Level*, em que foi acrescentado o estado *MAIN*. Quando o sistema entre em manutenção, tanto preventiva como corretiva, este modelo recebe um sinal de sincronização *prev*, e o autômato alcança o estado *MAIN*, desta forma não há alteração de nenhum parâmetro do sistema, mantendo-o indisponível. Ao final da manutenção este modelo recebe o sinal *fix*, e o sistema retorna ao seu estado de funcionamento.

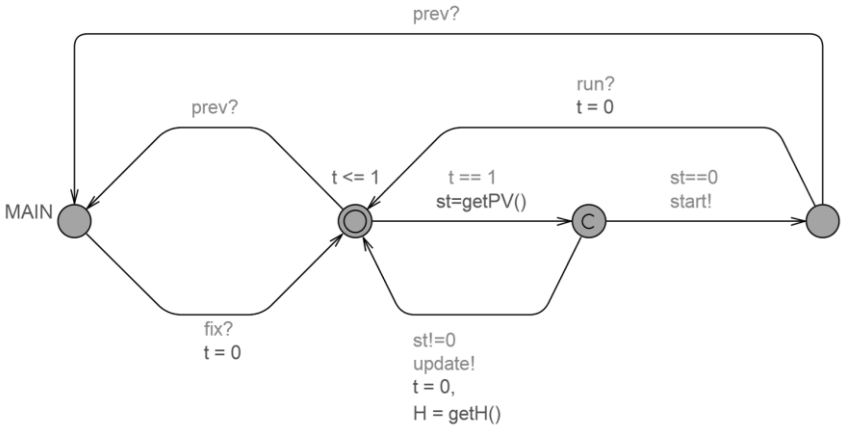


Figura 5.6 – Autômato *Level* para o modelo de estimação do *MTBM*.

No modelo *Obs* foi acrescentada a transição com a sincronização *prev*, como pode ser visto na Figura 5.7. Para que este modelo sempre seja reiniciado após uma intervenção de manutenção.

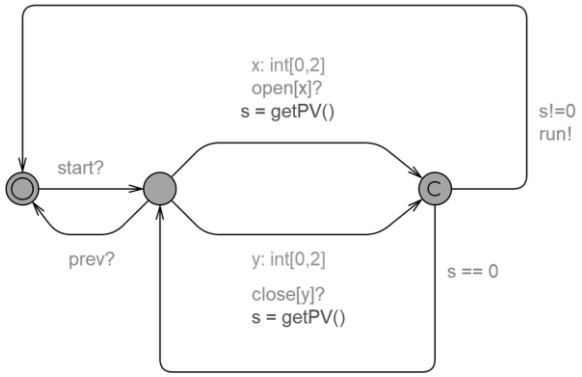


Figura 5.7 – Autômato *Obs* para o modelo de estimação do *MTBM*.

No modelo do controlador (Figura 5.8), foi necessário acrescentar a transição relacionada com a sincronização *fix*, para que quando o sistema retorne da manutenção, imediatamente o estado de operação dos componentes retorne para a sua condição inicial. Também, foi retirada a sincronização *stop*, visto que não há mais pausa durante a manutenção, e foi acrescentada a variável "MC", que tem seu valor atualizado para 1 sempre que o nível sai da região *R2*, indicando que o sistema entrará em manutenção corretiva.

A Figura 5.9 mostra o modelo que relaciona a falha e as manutenções para a bomba *P1*. Inicialmente, o sistema calcula o *delay1*, que é o tempo até a falha do componente, permanecendo no estado *WAIT* até que o tempo seja igual ao valor de *delay1*. A transição seguinte pode levar o componente ao estado *FAULT* na posição ligada (*ON*) ou desligada (*OFF*). Se o nível do reservatório ultrapassar os valores limite da região *R2*, o controlador envia o comando *up*, que é recebido por todos os componentes em falha e imediatamente este modelo envia um sinal de *repair*, que é recebido pelo modelo *Corretiva* (Figura 5.10), que representa a manutenção corretiva do sistema. Após receber o sinal de sincronização, o

modelo da manutenção corretiva, envia o sinal *prev* para os modelos dos componentes, assim aqueles que não estão em falha também entram em manutenção, e para os modelos *Level* e *Obs.*

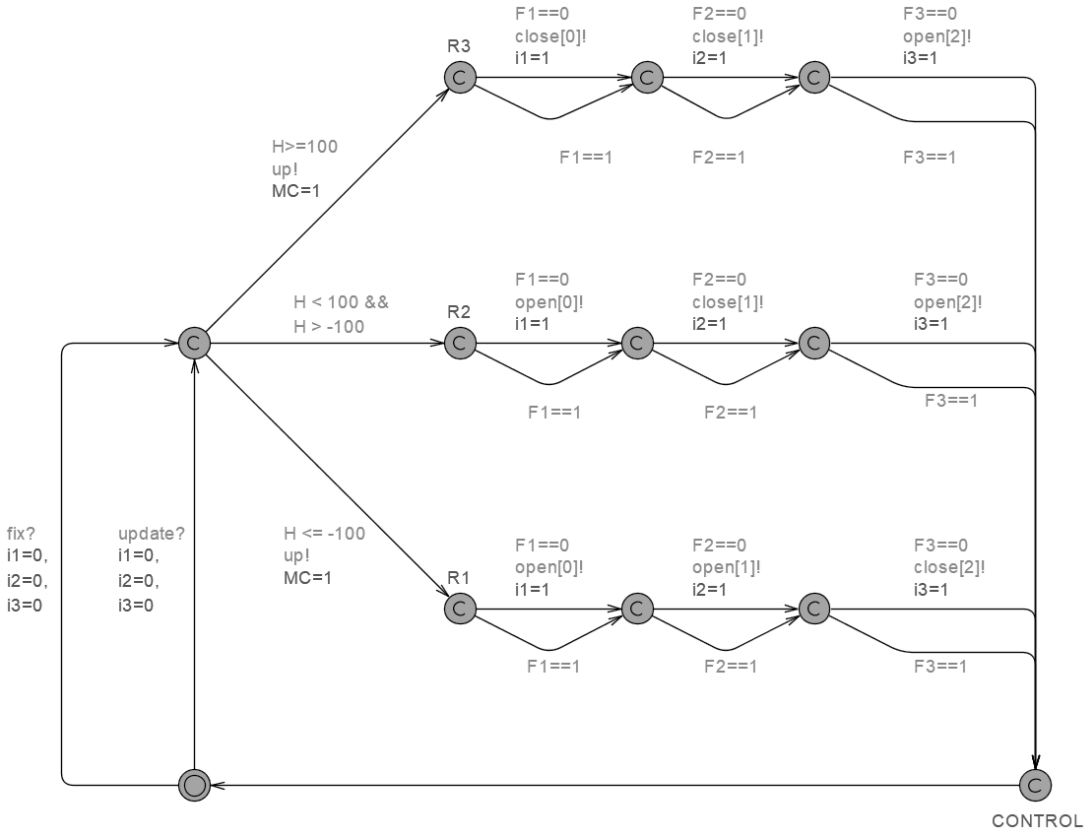


Figura 5.8 – Autômato *Controller* para o modelo de estimação do *MTBM*.

Os modelos da bomba *P2* e da válvula *V* não serão apresentados aqui, mas são semelhantes ao modelo *WeibullP1*, alterando apenas os parâmetros da distribuição de Weibull, e as indicações das variáveis próprias de cada componente.

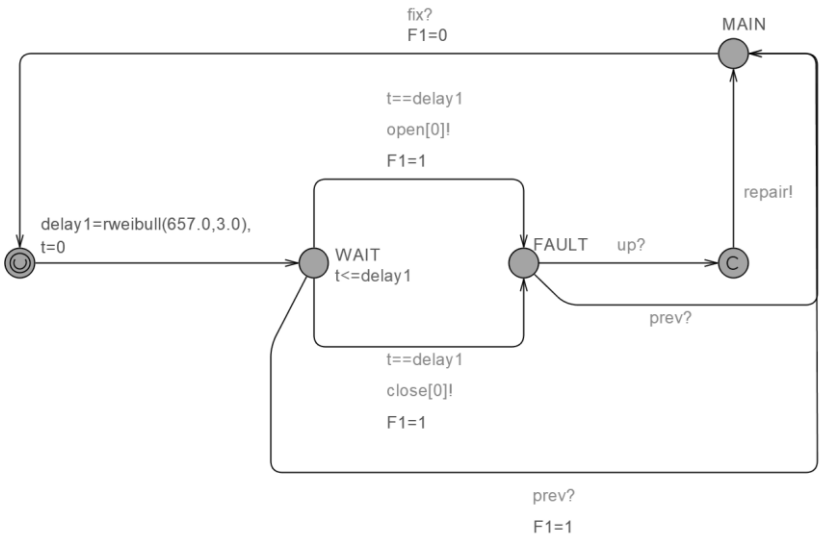


Figura 5.9 – Autômato *WeibullP1* para o modelo de estimação do *MTBM*.

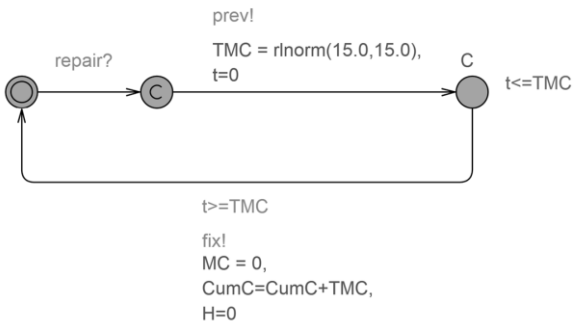


Figura 5.10 – Autômato *Corretiva* para o modelo de estimação do *MTBM*.

A Figura 5.11 ilustra o modelo da manutenção preventiva, o tempo *t* indica o intervalo entre as intervenções. Quando o relógio atinge o valor determinado para o intervalo, o modelo verifica se o sistema está em manutenção corretiva, caso esteja retorna ao estado inicial, caso contrário, envia o sinal de *prev* aos modelos dos componentes, *Level* e *Obs*, e entra no estado de preventiva (indicado pela invariante t<=TMP).

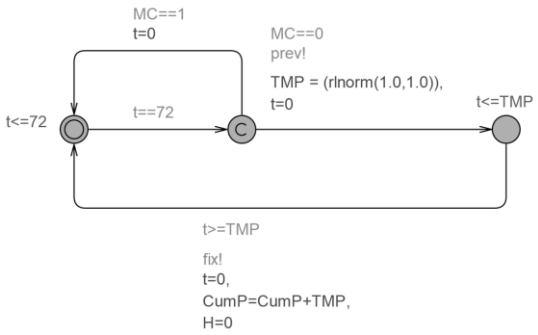


Figura 5.11 – Autômato *Preventiva* para o modelo de estimação do *MTBM*.

As variáveis *TMC* e *TMP* referem-se ao tempo de manutenção corretiva e tempo de manutenção preventiva, e são calculados a partir da distribuição Lognormal. Para tanto é necessário estimar a média e o desvio padrão da distribuição. Visto que, nas literaturas de referência utilizadas para a execução deste trabalho foi usado um tempo médio de reparo de 5h para uma distribuição exponencial, sabendo que a média, variância e desvio-padrão para a distribuição exponencial é calculada pelas fórmulas descritas no Quadro 5.1, chega-se à conclusão que a média e desvio padrão tem mesmo valor numérico, sendo de 5h (equivalente a 15 u.t.). Logo, para a obtenção dos valores de *TMC* foram utilizados estes valores, já para *TMP* foram utilizados valores variados, de modo a permitir avaliar a influência do tempo em manutenção preventiva sobre a disponibilidade do sistema.

Quadro 5.1 – Média, Variância e Desvio-padrão da distribuição exponencial.

Média (μ)	$\mu = \frac{1}{\lambda}$
Variância (var)	$var = \frac{1}{\lambda^2}$
Desvio-padrão (σ)	$\sigma = \sqrt{var}$

As variáveis *CumC* e *CumP* são os acumuladores do tempo que o sistema ficou em manutenção corretiva e preventiva, respectivamente. Estas variáveis são utilizadas para o cálculo da disponibilidade no tempo final de missão do sistema, ou seja, em 1000 h (equivalente a 3000 u.t.), no modelo *Disp*, que pode ser visualizado na Figura 5.12.

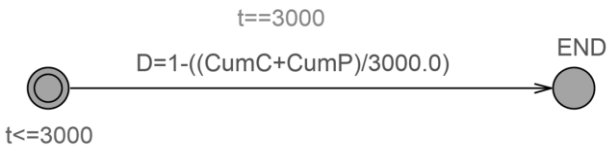


Figura 5.12 – Autômato *Disp* para o modelo de estimação do *MTBM*.

Etapa 2: Análise de Disponibilidade do sistema

A análise de disponibilidade do sistema foi realizada para todas as regiões da curva da banheira, isto é, quando o sistema estava em mortalidade infantil, na vida útil e no desgaste. A configuração da região da curva da banheira é feita através da parametrização do parâmetro de forma da distribuição de Weibull.

Para a escolha do parâmetro de forma (β) foi utilizado o banco de dados de parâmetros de forma e escala da distribuição de Weibull para diversos equipamentos e componentes, disponível no endereço eletrônico: `http://www.barringer1.com/wdbase.htm`. Neste banco de dados, estão disponibilizados três valores estimados para β: baixo (mortalidade infantil), típico (vida útil) e alto (desgaste), tanto para bombas como para válvulas liga-desliga.

Para o parâmetro de escala (η), como já mencionado, foi utilizado os valores de tempo médio para falha de cada um dos componentes. Na Tabela 5.1 consta os valores de β e η utilizados para cada um dos componentes.

Tabela 5.1 – Parâmetros β e η da distribuição de Weibull

Componente	**β**			**η**
	Baixo	**Típico**	**Alto**	
Bomba P1	0,5	1,2	3,0	219 h (657 u.t)
Bomba P2	0,5	1,2	3,0	175 h (525 u.t)
Válvula V	0,5	1,1	3,0	320 h (960 u.t)

Logo, foi realizada a análise de disponibilidade para três casos:

- Caso I: Sistema em mortalidade infantil: todos os componentes foram parametrizados para valores de β baixos;
- Caso II: Sistema em vida útil: todos os componentes foram parametrizados para valores de β típicos;
- Caso III: Sistema em desgaste: todos os componentes foram parametrizados para valores de β altos;

Para todos os casos, foi realizada a variação do intervalo entre manutenções preventivas e do tempo em manutenção preventiva, conforme pode ser visto na Tabela 5.2. Para cada intervalo foi avaliada a disponibilidade para cada um dos tempos em manutenção preventiva, totalizando 64 estimações de disponibilidade para cada caso.

Tabela 5.2 – Variação dos parâmetros de manutenção preventiva.

Intervalo entre preventivas (dias)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30 e 35
Tempo em preventiva (h)	5h (15 u.t.); 3,33h (10 u.t.); 1,67h (5 u.t.) e 0,33h (1 u.t.)

A *query* utilizada para a obtenção do valor de disponibilidade foi:

```
E[<=3000;1500] (max:D)
```

Esta é uma *query* dentro do verificador UPPAAL SMC que calcula o valor médio máximo ou mínimo de determinada variável do modelo, com grau de confiança que pode ser determinado pelo usuário. Esta *query* retorna a média do maior valor atingido pela variável e o erro da medida.

Para o caso em estudo, utilizou um tempo de missão de 1000h (3000 u.t.), 1500 simulações e um grau de confiança de 95%, resultando em valores com erro menor do que 0,001.

Caso I: Sistema em Mortalidade Infantil

A Figura 5.13 ilustra a curva de disponibilidade do sistema quando o mesmo se encontra em mortalidade infantil.

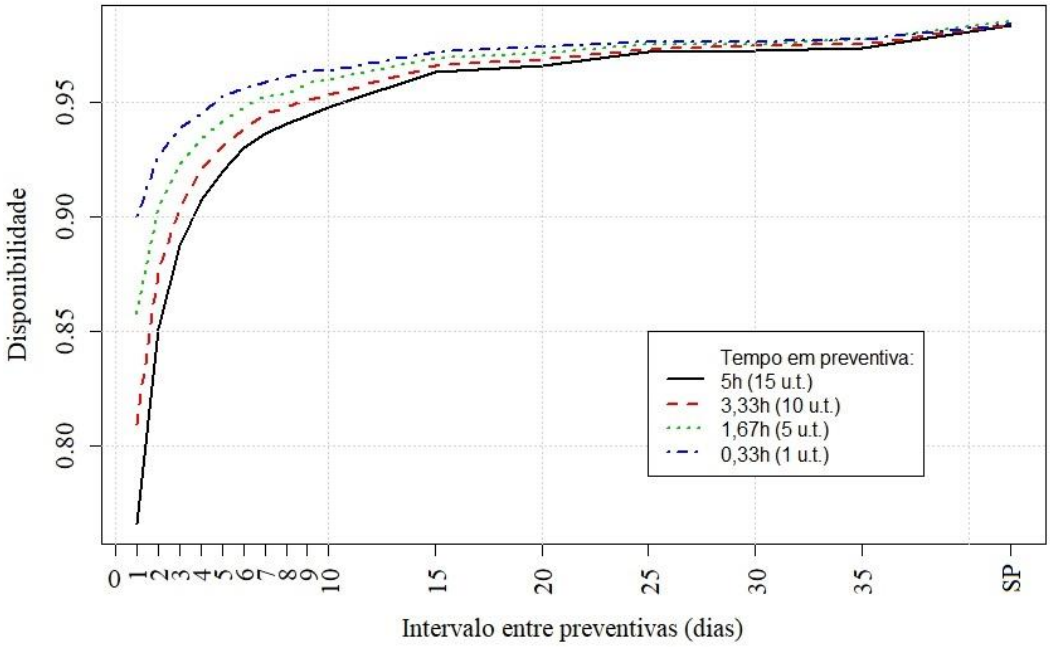


Figura 5.13 - Análise de Disponibilidade do sistema em Mortalidade Infantil.

A inserção de manutenção preventiva provocou a diminuição da disponibilidade do sistema, sendo que quanto menor o intervalo, menor a disponibilidade. Este fato já era esperado, visto que as falhas em mortalidade infantil são decorrentes de erro de projeto e fabricação. Para esta região da curva da banheira, a tendência é que as falhas diminuem com o passar do tempo, logo inserir manutenções periódicas, só tendem a diminuir o tempo disponível do sistema. Ainda, como levam o sistema para a condição como novo, retornam o sistema para valores altos de taxa de falha.

Neste caso, a disponibilidade máxima ocorre quando o sistema não tem intervenções preventivas (SP), permanecendo próxima de 98,5%.

Caso II: Sistema em Vida Útil

A Figura 5.14 ilustra os valores obtidos para a disponibilidade do sistema em cada intervalo de manutenção preventiva considerado. Cada curva representa um tempo em preventiva do sistema em vida útil.

É possível observar que quanto maior for o intervalo entre preventivas maior será a disponibilidade do sistema, sendo que a máxima disponibilidade ocorre quando não há intervenções preventivas no sistema. Também pode ser visto que, quando menor for o tempo despendido na intervenção maior será a disponibilidade do sistema quando em vida útil. O valor máximo de disponibilidade do sistema em vida útil, que ocorre quando não há preventiva, é de aproximadamente 98%.

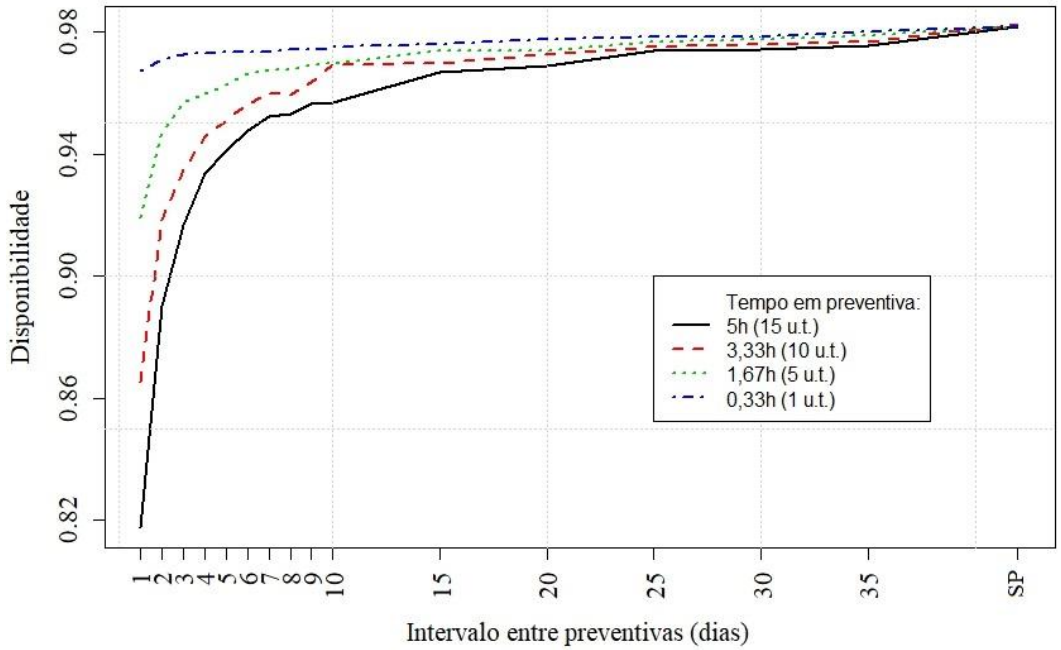


Figura 5.14 – Análise de Disponibilidade do sistema em vida útil.

Portanto, chega-se à conclusão que quando o sistema está em vida útil a manutenção preventiva não traz ganhos na disponibilidade do sistema, confirmando o que diz Lafraia (2001), que não há utilidade em programar manutenções baseadas em horas de operação quando um equipamento está operando na parte plana da curva da banheira, visto que, determinar um intervalo de parada requer previsibilidade, o oposto de aleatoriedade, e como esta região está relacionada a falhas aleatórias a previsão é inviável.

Caso III: Sistema em Desgaste

A Figura 5.15 ilustra os valores obtidos de disponibilidade para o sistema em desgaste, neste caso, já é possível observar alguns picos de máxima disponibilidade para alguns intervalos entre manutenções. Isto se torna mais evidente à medida que o tempo em preventiva é reduzido.

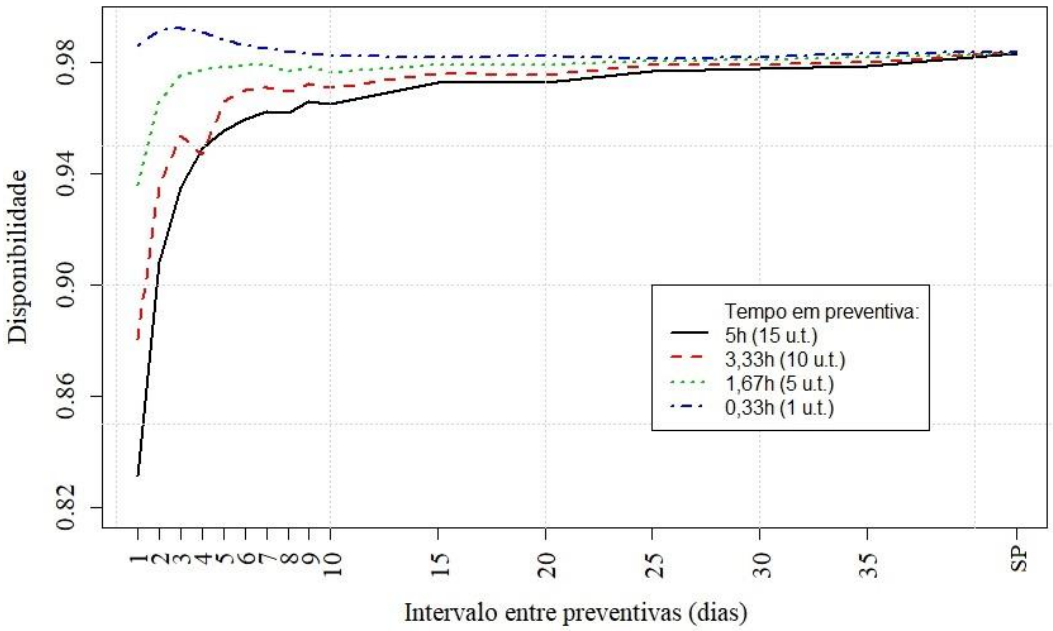


Figura 5.15 - Análise de Disponibilidade do sistema em desgaste.

As Figura 5.16, Figura 5.17, Figura 5.18 e Figura 5.19 mostram em detalhes a curva de disponibilidade para cada tempo em preventiva. Quando o tempo em preventiva é igual ao tempo em corretiva, isto é, *MP* = 5h (15 u.t.), o comportamento do sistema é semelhante a quanto o mesmo estava em vida útil, ou seja, a máxima disponibilidade ocorre quando não há intervenção de manutenção preventiva. O mesmo fato ocorre para quando *MP* = 3,33h (10 u.t.) (1,5 vezes menor que o tempo em corretiva - *MC*).

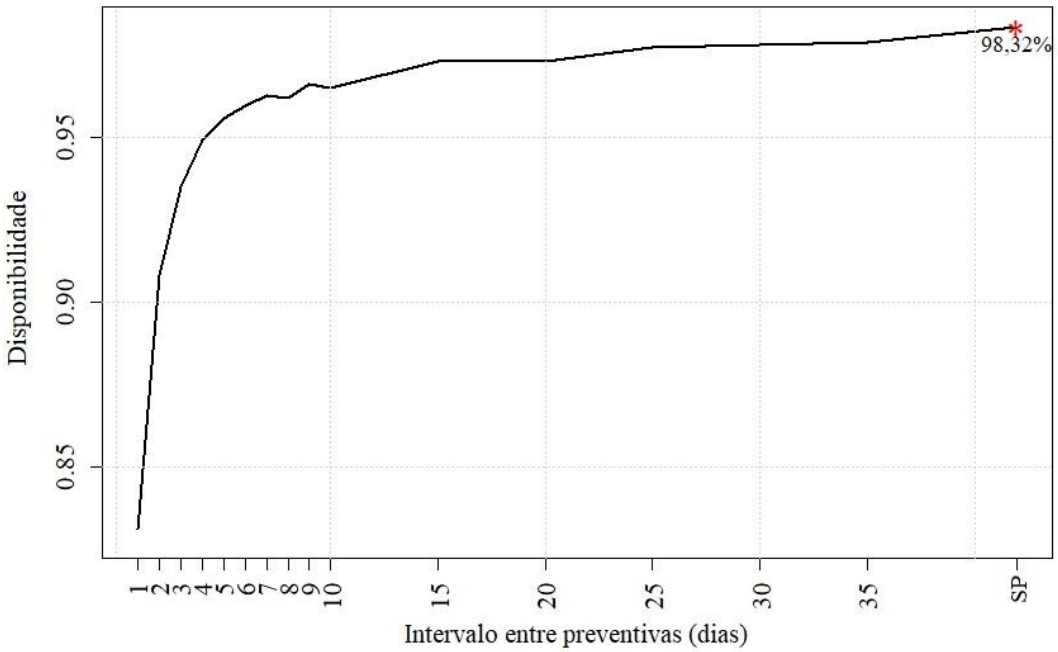


Figura 5.16 - Curva de Disponibilidade para o tempo de preventiva - *MP* = 5h (15 u.t.).

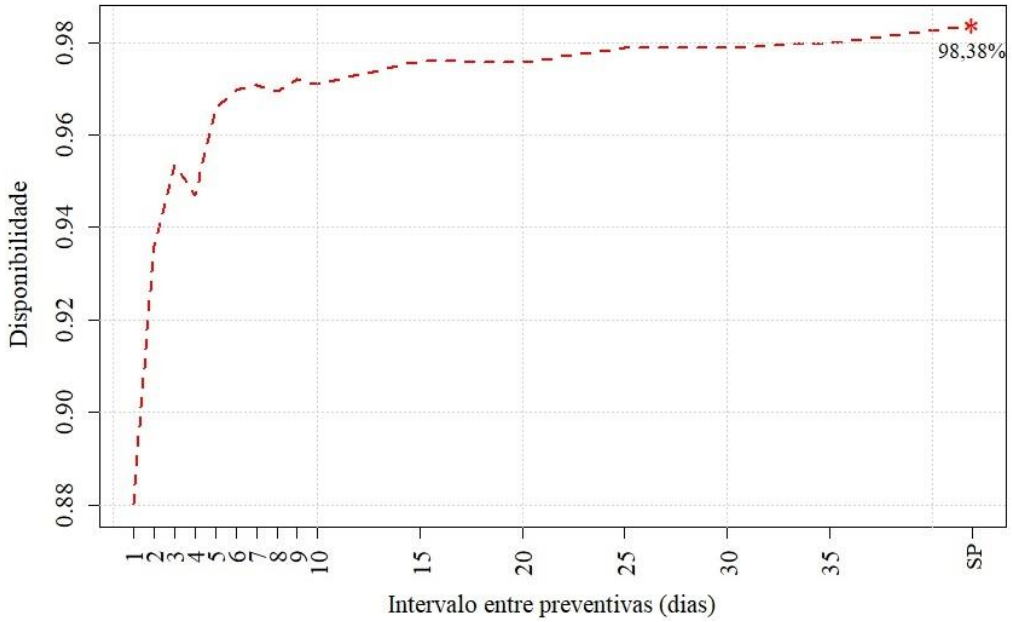


Figura 5.17 - Curva de Disponibilidade para *MP* = 3,33h (10 u.t.).

Quando *MP* é igual a 1,67h (5 u.t.), ou seja, 3 vezes menor que *MC*, já é possível observar alguns picos de máxima disponibilidade, embora, o valor máximo ainda seja quando não há preventivas.

A manutenção preventiva só passa a trazer ganhos efetivos na disponibilidade do sistema, quando seu tempo é muito menor do que o tempo de corretiva, como pode ser observado na Figura 5.19.

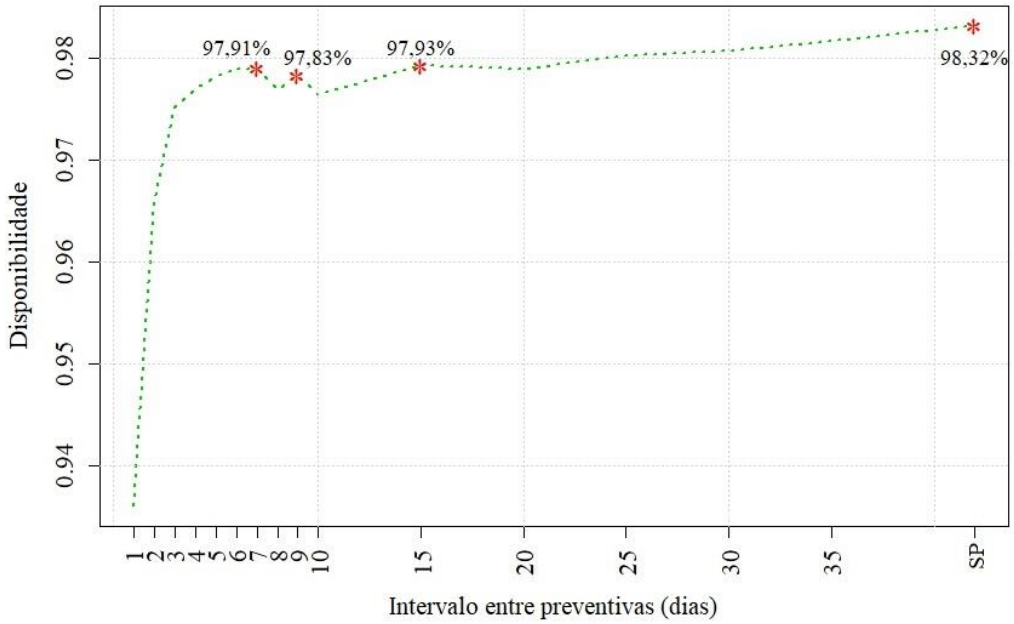


Figura 5.18 - Curva de Disponibilidade para *MP* = 1,67h (5 u.t.).

Quando *MP* é 15 vezes menor que *MC*, neste caso, a máxima disponibilidade de 99,23% ocorre quando são feitas intervenções preventivas a cada 3 dias. Por exemplo, se o tempo médio de corretiva for de 5h, e se a equipe de manutenção realizar ações preventivas, como limpeza e lubrificação em um tempo de 20 min a cada 03 dias, haverá ganhos significativos na disponibilidade do sistema em fase de desgaste.

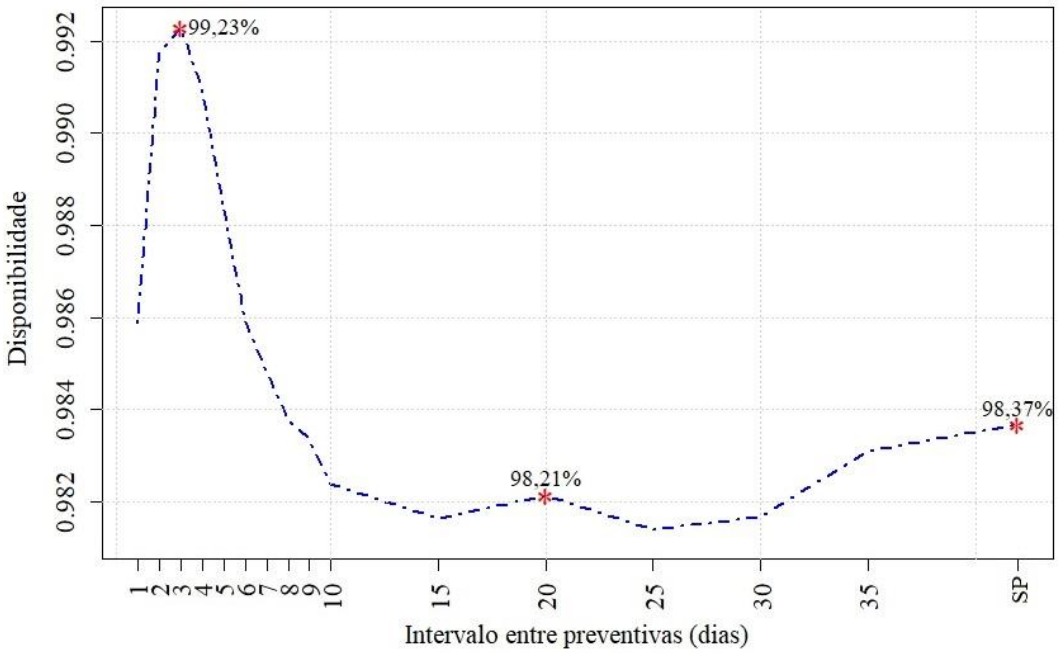


Figura 5.19 - Curva de Disponibilidade para *MP* = 0,33h (1 u.t.).

A fim de determinar para qual relação entre os tempos em preventiva e corretiva, a inserção da manutenção preventiva passa a definir a máxima disponibilidade, foram feitas

simulações para valores de *MP* entre 1,67h (5 u.t.) e 0,33h (1 u.t.), as curvas de disponibilidade podem ser visualizadas nas Figura 5.20, Figura 5.21 e Figura 5.22. Observa-se que para *MP* = 1h (3 u.t.), a máxima disponibilidade ocorre para o intervalo entre manutenções preventivas de 4 dias, ou seja, para o tempo de preventivas 5 vezes menor que tempo de corretiva já se evidencia o ganho de disponibilidade com intervenções preventivas.

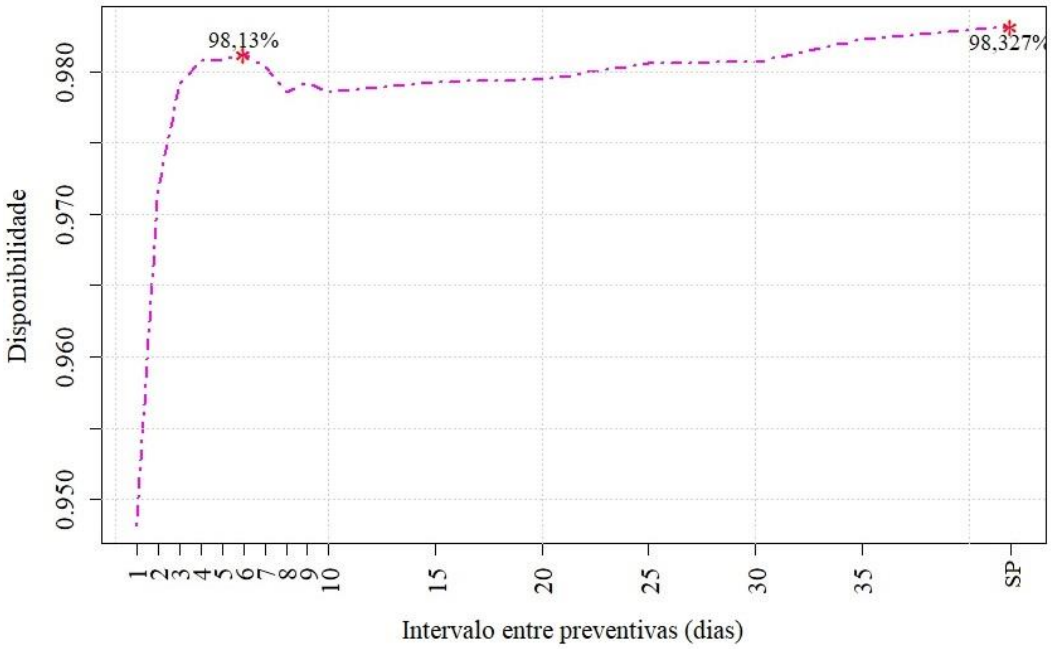


Figura 5.20 - Curva de Disponibilidade para *MP* = 1,33h (4 u.t.).

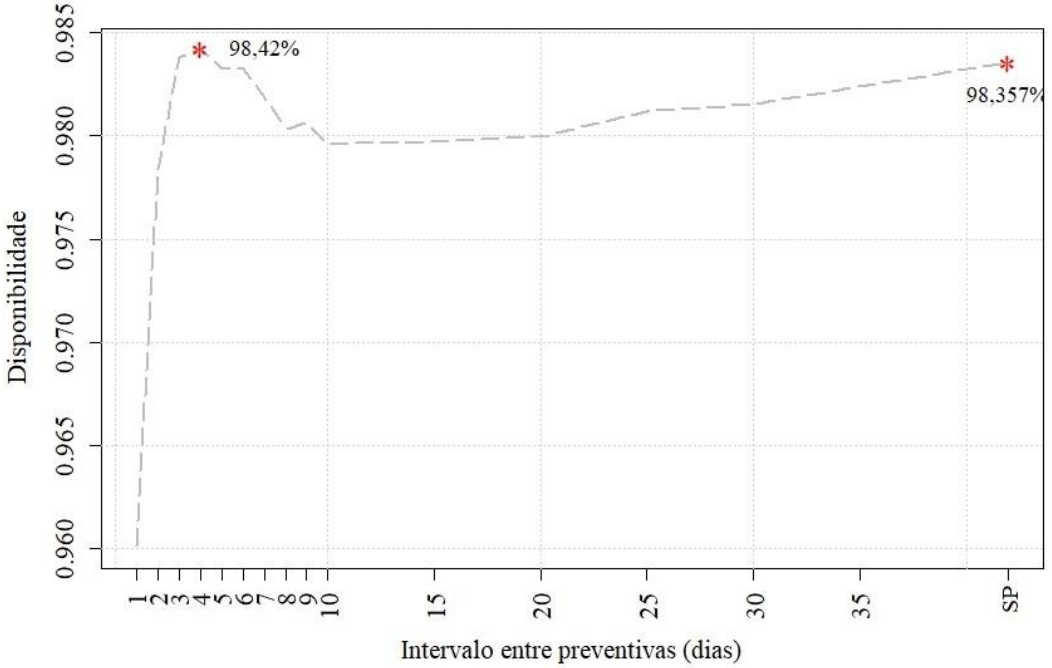


Figura 5.21 - Curva de Disponibilidade para *MP* = 1h (3 u.t.).

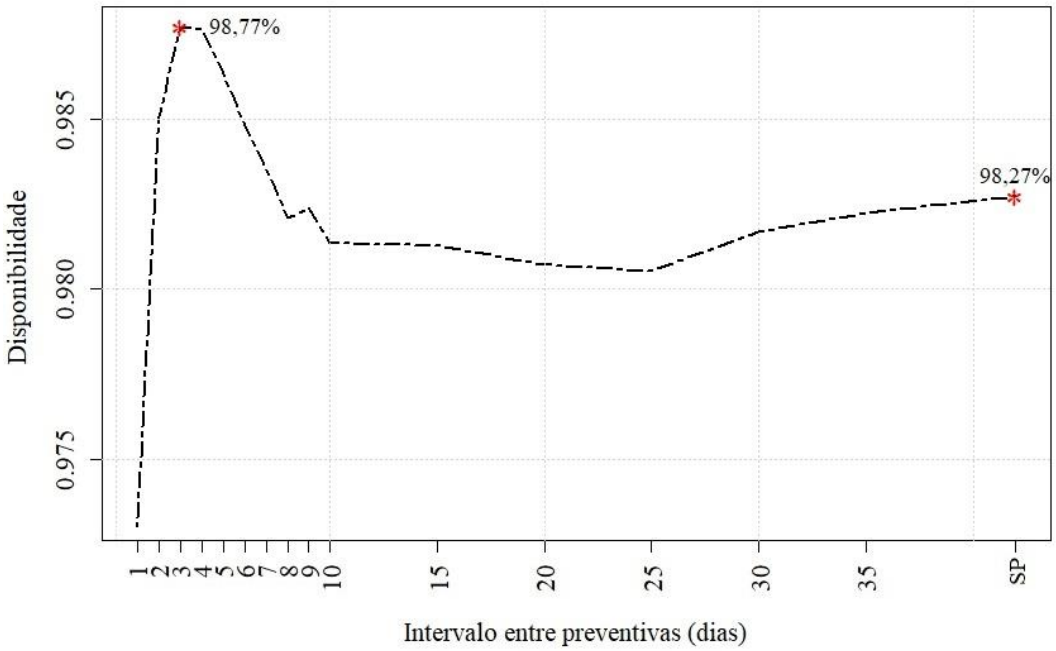


Figura 5.22 - Curva de Disponibilidade para *MP* = 0,67h (2 u.t.).

Portanto, chega-se à conclusão que para sistemas em mortalidade infantil e vida útil a manutenção preventiva não traz ganhos na disponibilidade do sistema. Já para sistemas em desgaste começa a apresentar efeitos benéficos quando o tempo em preventiva é no mínimo 5 vezes menor que o tempo em corretiva, onde o *MTBM* foi de 4 dias com disponibilidade de 98,42%. Para *MP* 7,5 vezes menor que *MC,* o *MTBM* foi de 3 dias com *D* = 98,77%. A maior disponibilidade encontrada foi para *MP* 15 vezes menor do *MC*, onde o *MTBM* também foi de 3 dias com *D* = 99,23%.

Os resultados obtidos nesta análise de disponibilidade vão em consonância ao exposto na literatura sobre confiabilidade de sistemas, em que as manutenções preventivas só passam a ter efeito significativo sobre os parâmetros de confiabilidade quando o sistema está em desgaste. Para tanto, Lafraia (2001) propõe a seguinte estratégia de manutenção:

Quadro 5.2 – Estratégia de Manutenção. Fonte: Lafraia (2001).

Valor de β	**Tendência da taxa de falha**	**Tipo de manutenção**
$\beta < 1$	Taxa de falha decrescente	Manutenção Corretiva
$\beta = 1$	Taxa de falha constante	Manutenção preditiva/corretiva/por oportunidade
$\beta > 1$	Taxa de falha crescente	Manutenção Preventiva

A partir dos resultados obtidos neste capítulo, foi possível verificar que a verificação formal de modelos pode ser utilizada para predição de dados de confiabilidade e disponibilidade de sistemas, podendo ser utilizada para elaboração e atualização de planos de manutenção.

Capítulo 6

Conclusão

Este trabalho teve como objetivo geral realizar uma avaliação da efetividade de se aplicar a metodologia de modelagem por autômatos estocásticos híbridos e verificação formal de modelos para levantamento dos parâmetros de confiabilidade de um sistema hidráulico. Ao término desta pesquisa, conclui-se que tal metodologia foi eficaz para análise de confiabilidade, visto que todos os objetivos específicos foram atendidos satisfatoriamente, como será detalhado a seguir.

Para a análise de comportamento do sistema em estudo e obtenção de sua característica de falha, quatro modelos foram criados, sendo eles: modelo físico, modelo de controle, modelo de falha e modelo de manutenção. Sendo realizada a especificação das propriedades, simulação e verificação dos modelos. Foi possível observar que todos os modelos apresentam comportamento adequado e satisfazem todas as propriedades a eles impostas.

Para a aplicação da metodologia proposta foi utilizada a ferramenta computacional UPPAL STRATEGO, de onde foram extraídas informações sobre a probabilidade de falha do sistema hidráulico com e sem reparo. Estes resultados foram comparados com os valores constantes nas literaturas consultadas, em que foi possível observar que todos os resultados obtidos foram semelhantes aos resultados das metodologias apresentadas pelas demais literaturas, uma vez que a diferença entre eles não passou de 2,5%.

A segunda etapa da pesquisa foi à estimação do *MTBM* ideal para obtenção da máxima disponibilidade do sistema, considerando se uma função de distribuição de falhas mais representativas com sistemas reais, ou seja, para tempos médios para a falha dos componentes mecânicos utilizou-se a distribuição de Weibull e para os tempos médios de manutenção a distribuição Lognormal. Como resultado desta etapa, confirmou-se o que a literatura sobre confiabilidade de sistemas propõe: a manutenção preventiva é benéfica, em termos de disponibilidade, somente quando o sistema está na fase de desgaste. Tanto nas fases de mortalidade infantil como vida útil, a inserção da manutenção preventiva gera decréscimo de disponibilidade do sistema.

Atingido todos os objetivos e tendo em vista que os modelos construídos apresentaram comportamento adequado na etapa de simulação, satisfizeram todas as propriedades na etapa de verificação formal e apresentaram característica de falha semelhante com aquela apresenta pelas demais literaturas para o sistema hidráulico em estudo, pode se concluir que a metodologia utilizada e os modelos construídos podem ser validados e a técnica de verificação de modelos para a análise de confiabilidade é eficaz.

Logo, este trabalho traz como contribuição original a validação da aplicação de modelagem por autômatos estocásticos híbridos e verificação formal de modelos para análise de confiabilidade de sistemas, na ferramenta computacional UPPAL STRATEGO. Sendo possível aplica-la em qualquer sistema hidráulico para predição de dados sobre a confiabilidade e disponibilidade do mesmo.

Sugere-se como trabalho futuro, a aplicação da metodologia empregada em um sistema hidráulico de tempo real com elevada criticidade, uma vez que se entende que se a metodologia foi adequada para o sistema em estudo, que não é classificado como sistema crítico, uma vez que é um problema teórico e não tem criticidade associada à sua função e ao tempo, a mesma pode ser replicada a qualquer sistema hidráulico, independentemente de sua criticidade.

Referências Bibliográficas

Aldemir, T. (1987). Computer-Assisted Markov Failure Modeling of Process Control Systems, *IEEE Transactions on Reliability* **36**(1): 133-144.

Associação Brasileira de Normas Técnicas (1994). *NBR 5462: confiabilidade e mantenabilidade.* Rio de Janeiro, Brasil.

Baier, C.; Katoen, J. P. (2008). *Principles of Model Checking.* Cambridge: Londres, Inglaterra.

Bergamo Filho, V. (1997). *Confiabilidade Básica e Prática.* Blucher, São Paulo, Brasil.

B-Daya, M. (2009). *Handbook of Maintenance Management and Engineering. Vol 7 of Springer,* chapter Failure Statistics, pp. 45-73.

Bollmann, A. (1997). *Fundamentos da Automação Industrial Pneutrônica, Projetos de comandos binários eletropneumáticos,* ABHP, São Paulo.

Caetano, A. O. (2011). *Controle Preditivo de Sistemas Híbridos,* Dissertação de Mestrado, *Universidade Federal de Uberlândia.* Uberlândia, Brasil.

Codetta-Raiteri, D., Bobbio, A. (2006). Stochastic Petri Nets Supporting Dynamic Reliability Evaluation, *Internacional Journal of Materials e Structural Reliability* **4**(1): 65–77.

Costa, G. S. (2008). *Utilização da verificação de modelos para o planejamento de missões de veículos aéreos não-tripulados,* Dissertação de Mestrado, *Instituto Militar de Engenharia.* Rio de Janeiro, Brasil.

De Negri, V. J. (2004). *Introdução aos Sistemas para Automação e Controle Industrial.* Universidade Federal de Santa Catarina, Florianópolis.

Guo, X., Yang, Z. (2016). On the Use of Probabilistic Model Checking for Reliability Evaluation, *3rd ICSAI - International Conference on Systems and Informatics,* Shanghai, China, pp. 566– 571.

Guzzon, S. O. (2009). *Uma proposta para a aplicação de FMEA em estudos de confiabilidade*. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul. Porto Alegre, Brasil.

Fogliatto F. S., Ribeiro, J. L.D. (2009). *Confiabilidade e Manutenção Industrial*. Elsevier, Rio de Janeiro.

Hopcroft, J. E., Ullman, J. D., Motwani, R. (2002). *Introdução à teoria de autômatos, linguagens e computação*. Campus, Rio de Janeiro.

Krilavicius, T. (2006). *Hybrid Techniques for Hybrid Systems*, Tese de doutorado, University of Twente, The Netherlands.

Kunz, G. O. (2012). *Metodologia para desenvolvimento de sistemas de controle de APM (Automated people movers) com aplicação ao sistema aeromovel de transporte de passageiros*, Tese de doutorado, *Universidade Federal do Rio Grande do Sul*, Porto Alegre, Brasil.

Lafraia, J. R. B. (1 ed) (2001). *Manual de confiabilidade, mantenabilidade e disponibilidade*, Qualitymark, Rio de Janeiro.

Lazar, M. (2006). *Model predictive control of hybrid systems: stability and robustness*, Tese de doutorado, Tecnnische Universiteit Eindhoven, Roemenië.

Linsingen, I. V. (4 ed) (2013). *Fundamentos de sistemas hidráulicos*, USFC, Florianópolis.

Losso et al. (2010). Modelagem e verificação formal de sistemas de tempo-real embarcados para aplicações aeroespaciais, *CONEM 2010 - VI Congresso Nacional de Engenharia Mecânica*, Paraíba, Brasil, pp. 1–8.

Macêdo, R. et al (2004). *Tratando a previsibilidade em sistemas de tempo-real distribuídos: especificação, linguagens, middleware e mecanismos básicos*, UFBA - Universidade Federal da Bahia, Salvador, Bahia.

Marseguerra, M., Zio, E. (1996). Monte Carlo approach to PSA for dynamic process systems. *Reliability Engineering and System Safety* **52:** 227-241.

Molossi, L. H., Almeida T., Vale, J. (2009). *Modal Checking (Verificação Formal)*. Universidade Federal do Rio Grande do Sul. Disponível em: <http://www.inf.ufrgs.br/~talmeida/repositorio/etapa2/logica/Verificacao%20Formal.pdf >. Acesso em: 03 mar 2017.

Monchy, F. (1989). *A função manutenção.* Ebras/Durban, São Paulo.

Nakashima, Y., Baba, T. (1989). OHCS: Hydraulic Circuit Design Assistant a Knowledge-Based System for Kayaba Industry Co., Ltd, *Conference on Innovative Applications of Artificial Intelligence,* Los Angeles, Estados Unidos da América, pp. 1–10.

Natário, R. (2011). *Alta Disponibilidade – Medição (II)*. Disponível em: < http://redes-e-servidores.blogspot.com.br/2011/02/alta-disponibilidademedicao-ii.html>. Acesso em 12 mai 2018.

Peng *et al*. (2013). A Probabilistic Model Checking Approach to Analysing Reliability, Availability, and Maintainability of a Single Satellite System. *European Modelling Symposium*. Manchester, United Kingdom, pp. 611-616.

Portal Action (2018). *Distribuição Lognormal.* Disponível em: <http://www.portalaction.com.br/confiabilidade/414-distribuicao-log-normal>. Acesso em: 16 jun 2018.

Sakurada, E. Y. (2013). *Metodologia para análise de confiabilidade dinâmica*, Tese de doutorado, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina.

Salas, K. M., Belan, H. C., De Negri, V. J. (2013). Modeling of electro-hydraulic systems using channel-agency petri nets, *COBEM 2013 – 22nd International Congress of Mechanical Engineering*, Ribeirão Preto, Brasil, pp. 4490-4499.

Salas, K. M. (2014). *Modelagem e análise de circuitos hidráulicos usando redes de Petri,* Dissertação de Mestrado, *Universidade Federal de Santa Catarina,* Florianópolis, Brasil.

Santos, F. C. (2017) *Metodologia de análise de sistemas de proteção com controle distribuído através da ferramenta de modelagem e verificação formal estatística,* Dissertação de Mestrado, *Universidade Estadual do Oeste do Paraná*, Foz do Iguaçu, Brasil.

Silva, J. C. (2012). *Análise de Confiabilidade de britadores e peneiras secundários.* Monografia de Pós-Graduação, *Universidade Federal de Ouro Preto,* Belo Horizonte, Brasil.

Siqueira, I. P. (2014). *Manutenção centrada em confiabilidade: manual de implementação.* Qualitymark, Rio de Janeiro.

Stein, B. (1998). Supporting Hydraulic Circuit Design by Efficiently Solving the Model Synthesis Problem., *International ICSC Symposium On Engineering Of Intelligent Systems*, [S.l], pp. 1274–1280.

Studart, T. M. C. (2005). *Probabilidade e Estatística: aplicada aos recursos hidrícos,* Apostila de Aula, *Universidade Federal do Ceará,* Fortaleza, Brasil.

Vaccaro, G. L. R. (1997). *Modelagem e análise da confiabilidade de sistemas,* Dissertação de Mestrado, *Universidade Federal do Rio Grande do Sul,* Porto Alegre, Brasil.

Verma, A. K., Ajit, S., Karanki, D. R. (2016). *Reliability and Safety Engineering.* 2ed. Springer: USA.

Yan, S., Zhang, H., Zhang, Y. (2015). Reliability Prediction of Hydraulic System with Probabilistic Model Checking. *The First International Conference on Reliability Systems Engineering (2015 ICRSE)*. Beijing, China, pp. 1-7.