MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

METODOLOGIA PARA DESENVOLVIMENTO DE SISTEMAS DE CONTROLE DE APM (*AUTOMATED PEOPLE MOVERS*) COM APLICAÇÃO AO SISTEMA AEROMOVEL DE TRANSPORTE DE PASSAGEIROS

por

Guilherme de Oliveira Kunz

Tese para obtenção do Título de
Doutor em Engenharia

Porto Alegre, junho de 2012

METODOLOGIA PARA DESENVOLVIMENTO DE SISTEMA DE CONTROLE DE APM (*AUTOMATED PEOPLE MOVERS*) COM APLICAÇÃO AO SISTEMA AEROMOVEL DE TRANSPORTE DE PASSAGEIROS

por

Guilherme de Oliveira Kunz
Engenheiro Mecânico

Tese submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Doutor em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Eduardo André Perondi (UFRGS)
Coorientador: Prof. Dr. José Mendes Machado (UMINHO)

Aprovada por:

Prof. Dr. Flávio José Lorini (UFRGS)

Prof. Dr. Luis Antonio Lindau (UFRGS)

Profa. Dra. Morgana Pizzolato (UFSM)

Prof. Dr. Francis Henrique Ramos França
Coordenador do PROMEC

Porto Alegre, 13 de junho de 2012

RESUMO

Este trabalho consiste na proposição de uma metodologia que garanta o projeto efetivo de controladores para utilização em sistemas de proteção, operação e supervisão de sistemas APM (*Automated People Movers*), utilizando como estudo de caso o Sistema Aeromovel. *Automated People Movers* (APM) são sistemas de transporte de passageiros com operação completamente automatizada (sem tripulação), trafegando em vias exclusivas e com alta frequência de serviço. Geralmente são operados em via elevada, acima dos obstáculos encontrados no nível do solo, de maneira a manter o espaço urbano de circulação. A automação completa de trens para transporte de passageiros é geralmente condição essencial para sua viabilidade financeira nos casos onde tem-se veículos pequenos para prover alta frequência de serviço. Em parte, a confiabilidade de um veículo com operação completamente automatizada aumenta com o uso de vias exclusivas impedindo a interferência de outros veículos ou pessoas na via, porém, uma metodologia de desenvolvimento do sistema de proteção, operação e supervisão que diminua a taxa de falhas e suas consequências é necessária por tratar-se de um sistema crítico. Este trabalho apresenta uma revisão das soluções utilizadas na construção de Sistemas APM e daquelas adotadas no controle digital de sistemas automáticos. Descreve-se o processo de construção do sistema de proteção, operação e supervisão baseados na proposta de ampliação da norma IEC 61850 e prova-se a importância do uso do processo de simulação, verificação formal e os testes de conformidade no desenvolvimento de um sistema de controle seguro para aplicação em Sistemas APM. Para cada etapa do trabalho são apresentados os testes de conformidade para verificação do modelo de controle proposto.

Palavras-chave: Sistemas APM, IEC 61850, Autômatos Temporizados.

ABSTRACT

This work consists in the proposition of a methodology that guarantees the effective design of controllers for use in protection, operation and monitoring Automated People Movers System, using as a case study Aeromovel system. Automated People Mover (APM) are systems of passenger transport with entirely automated operation (without crew), traveling in exclusive ways and with high service frequency. They are usually operated at elevated highway above the obstacles encountered on the ground level, so as to keep the urban space circulation. The complete automation of trains for passenger condition is necessary for their economic viability in the cases where they have small vehicles to supply high frequency of service. The reliability of an fully automated vehicle operation increases with the use of exclusive right-of-way preventing access from other vehicles or people on the track, however, the development of a methodology for protection, operation and monitoring systems is necessary to decrease the rate of failures and their consequences. This work presents a review of the solutions used in the construction of APM systems and the digital control of automated systems. We describe the construction process of the protection, operation and supervision systems based on the proposed extension of IEC 61850 and proves the importance use of the simulation, formal verification and conformity tests to develop a safe control system. For each stage of the work, the compliance tests to verify the proposed control model are presented.

Palavras-chave: APM systems, IEC 61850, Timed Automata.

AGRADECIMENTOS

Aos professores e funcionários da Universidade Federal do Rio Grande do Sul (UFRGS) onde tive o privilégio de cursar a graduação, mestrado e doutorado.

Aos colegas professores e aos funcionário da Universidade do Oeste do Paraná (UNIOESTE) pela oportunidade de ampliar a qualificação profissional e pela confiança depositada.

A Universidade do Minho e aos moradores da cidade de Guimarães (Portugal) pelo acolhimento e oportunidade de um excelente intercâmbio pesquisístico.

Ao Prof. Dr. Eduardo André Perondi e Prof. Dr. José Mendes Machado pela amizade, exemplarismo, confiança, orientação, dedicação e paciência.

Aos professores membros da banca Profa. Dra. Morgana Pizzolato, Prof. Dr. Luis Antonio Lindau e Prof. Dr. Flávio José Lorini, pelas dicas valiosas e pela paciência na avaliação e correção do trabalho.

A Fundação do Parque tecnológico da ITAIPU e equipe, pela concessão da bolsa de estudos e auxílio para publicação dos resultados da pesquisa.

Ao Centro Nacional de Supercomputação (CESUP/UFRGS) pelo uso dos recursos computacionais permitindo a conclusão do trabalho.

A equipe do Aeromóvel pelo apoio e acesso as informações essenciais para o desenvolvimento da tese.

A equipe de desenvolvedores da ferramenta UPPAAL e aos pesquisadores incansáveis que trabalham diariamente para melhorar a segurança nos sistemas de transporte de passageiros.

Agradeço a minha família pelo arrimo essencial na minha formação, em especial aos meus pais Fernando e Miriam pelo incentivo constante e apoio incondicional. Ao meu irmão Roberto por sempre me incentivar a seguir em frente. A minha avó Marta pelo exemplo de vida, a minha avó Enriqueta pelo carinho e acolhimento e, em especial, ao meu avô Roberto pelo exemplo de hombridade.

A minha esposa Milena pela companhia serena, incentivo constante, paciência incansável e orientação sábia.

Em retribuição a confiança depositada
e ao conhecimento compartilhado.
Serei eternamente grato.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
ABS	*Anti-Blocking Brake System*
ACSI	*Abstract Communication Service Interface*
AD	Analógico-Digital
AMV	Aparelhos de Mudança de Via
ANSI	*American National Standards Institute*
AP	*Access Point*
APB	*Absolute Permissive Block*
APM	*Automated People Movers*
ASCE	*American Society of Civil Engineers*
ASN	*Abstract Syntax Notation*
ATC	*Automated Train Controller*
ATCS	*Advanced Train Control Systems*
ATO	*Automatic Train Operation*
ATP	*Automatic Train Protection*
ATS	*Automatic Train Supervision*
DBM	*Difference Bound Matrices*
CBTC	*Communications-Based Train Control*
CCTV	*Closed-circuit television*
CEE	Comissão de Estudo Especial
CLP	Controlador Lógico Programável
CPU	*Central Processing Unit*
CSV	*Comma-Separated Values*
CTC	*Centralized Traffic Control*
CTL	*Computation Tree Logic*
DA	Digital-Analógico
DSP	*Digital Signal Processor*
FLOPS	*Floating-Point Operations per Second*
GMP	Grupo Moto Propulsor
GOOSE	*Generic Object Oriented Substation Event*
GPS	*Global Positioning System*

HMI	*Human Machine Interface*
IEEE	*Institute of Electrical and Electronics Engineers*
IEC	*International Electrotechnical Commission*
IED	*Intelligent Electronic Devices*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
IRIG	*Inter-range instrumentation group time codes*
ISO	*International Organization for Standardization*
LN	*Logical Node*
LSB	*Local Sensor Bus*
LVB	*Local Vehicle Bus*
MMS	*Manufacturing Messaging Specification*
MVB	*Multifunction Vehicle Bus*
NRT	*Not Real Time*
NTP	*Network time protocol*
PPHPD	Passageiros por Hora por Direção
RAM	*Random-access memory*
RF	Rádio Frequência
RFC	*Request for Comments*
RT	*Real Time*
RTAI	*RealTime Application Interface*
RTDS	*Real Time Digital Simulator*
RTSJ	*Real Time Specification for Java*
SA	Sistema de Atuação
SAM	Sistema de Atuação e Medição
SAS	Sistemas de Automação de Subestações
SCADA	*Supervisory Control and Data Acquisition*
SCL	*System Configuration Language*
SCSM	*Specific Communication Service Mapping*
SM	Sistema de Medição
SMV	*Sample Value*
SMP	*Symmetric Multi Processors*
SP	Sistema de Pedido
SPR	Sistema de Pedido e Resposta
SR	Sistema de Resposta
TCTL	*Timed Computation Tree Logic*

T&DI	*Transportation & Development Institute*
UDP	*User Datagram Protocol*
VA	*Válvula Atmosférica*
VATO	*Vehicle Automatic Train Operation*
VATP	*Vehicle Automatic Train Protection*
VDF	*Válvula de Direcionamento de Fluxo*
VIT	*Válvula de Isolamento de Trecho*
VLAN	*Virtual Lan*
RATO	*Region Automatic Train Operation*
RATP	*Region Automatic Train Protection*
WLAN	*Wireless Local Area Network*
WTB	*Wire Train Bus*

LISTA DE SÍMBOLOS

A	área útil do êmbolo do pistão [m^2]
F	força [N]
F_a	força de atrito [N]
F_e	força externa [N]
k_n	relógios
M	massa [kg]
p	pressão absoluta [Pa]
s_n, c_n	canais de comunicação
S_n	estado
t	tempo [s]
y	posição do embolo do pistão [m]
$\dot{y}$	velocidade do embolo do pistão [m/s]
$\ddot{y}$	aceleração do embolo do pistão [m/s^2]
$()?$	receptor de sincronismo
$()!$	emissor de sincronismo
φ	fórmula de verificação
A	operador de caminho que significa: para todos os caminhos
E	operador de caminho que significa: existe um caminho
G	operador temporal que significa: sempre ou globalmente
F	operador temporal que significa: no futuro
U	operador temporal que significa: até
W	operador temporal que significa: a menos que
X	operador temporal que significa: próximo
R	operador temporal que significa: habilita
$\rightarrow$	conector implica da lógica
$\rightsquigarrow$	operador *leads to* do UPPAAL. $p \rightsquigarrow q$ equivale a $AG(p \rightarrow AFq)$
&	conector lógico "e".
&&	conector lógico "e" quando escrito na guarda de um arco no UPPAAL
==	operador de teste de igualdade
|	conector lógico "ou"
$!()$	símbolo lógico da negação

1 INTRODUÇÃO

Neste capítulo estão descritos os principais aspectos referentes à motivação, objetivos e organização geral do presente trabalho. Esta introdução engloba uma breve descrição geral sobre APM (*Automated People Movers*) e, mais especificamente, sobre a automação do Sistema Aeromovel. Além disso, neste capítulo é apresentada a estrutura geral do texto.

1.1 Motivação

Mais de 90% da inovação dos veículos atuais está no *software* embarcado [Roychoudhury, 2009]. Está afirmação demonstra a importância de sistemas embarcados no controle de veículos modernos como, por exemplo, os sistemas aplicados no controle de aeronaves e trens.

A automação completa de trens para transporte de passageiros é condição *sine qua non* para sua viabilidade econômica, principal nos casos onde tem-se veículos pequenos para prover alta frequência de serviço [Vuchic, 2007].

Em parte, a confiabilidade de um veículo de operação completamente automatizada, aumenta com o uso de vias exclusivas (*Right-of-Way* de categoria A) impedindo acesso de outros veículo ou pessoas na via, porém, uma metodologia de desenvolvimento do sistema de proteção, operação e supervisão que diminua a alta taxa inicial de falhas, típicas de sistemas computacionais, e suas consequências, é necessária por trata-se de um sistema crítico [Lafraia, 2006].

1.2 *Automated People Movers*

Automated People Movers (APM) são sistemas de transporte de passageiros com operação completamente automatizada (sem tripulação), trafegando em vias exclusivas e com alta frequência de serviço. Geralmente são operados em via elevada, acima dos obstáculos encontrados no nível do solo, de maneira a manter o espaço urbano de circulação [IEEE, 2004].

Aplicações típicas se dão em terminais aeroportuários, centros comerciais, ligação a grandes estacionamentos periféricos, complexos turísticos, parques ecológicos, campi universitários, hospitais, alimentação de sistemas troncais de transporte (metrô, trem e

corredores de ônibus), centros urbanos com alta densidade de fluxo de pessoas, dentre outras. Usualmente são linhas do tipo *shuttle-loop* [Vuchic, 2007].

A capacidade de sistemas APM na aplicação em aeroportos chega a 9000 pphpd (passageiros por hora por sentido) assumindo 75 passageiros por veículo, considerando bagagens e realizado por uma composição com quatro módulos e dois minutos de intervalo. Segundo Branco, 2008, a ferrovia é um dos meios de transporte que mais progrediu tecnologicamente desde seus primeiros exemplares comerciais, saindo do trem a vapor para o elétrico e agora alcançando a tecnologia de levitação magnética, multiplicando por dez ou mais sua velocidade, ao mesmo tempo em que atingiram altos níveis de segurança. Com relação à variável consumo, a Tabela 1.1 apresenta um quadro comparativo entre o consumo de diferentes meios de transporte consumindo cinco litros de combustível por tonelada transportada. Nesta comparação levam-se em consideração somente veículos nos quais o sistema de propulsão está situado no próprio veículo.

Tabela 1.1 Quadro comparativo de eficiência entre meios de transporte

Tipo	Distância
Navegação Fluvial	500 km
Ferrovias	333 km
Rodovias	100 km
Aerovias	6,6 km

Fonte: Branco, 2008

Automated People Movers são divididos em quatro grupos: propulsão por aderência, propulsão por membro de tensão, propulsão por fluxo de ar e gerador eletromagnético [ANSI, 2008a]. No caso do sistema Aeromovel, que possui propulsão externa, utiliza de sistema de propulsão por fluxo de ar conforme apresentado a seguir.

1.3 O Sistema Aeromovel

As principais características da tecnologia Aeromovel são a exclusividade de trânsito no percurso e a alta relação carga útil/peso transportado e tração externa (independente das rodas), conforme descrito por Lindau et al., 1986. Estas características são decorrentes, respectivamente, do fato de o carro deslocar-se acima do solo em uma via exclusiva e de ter seu sistema externo de potência. Isto torna seu veículo mais leve que os de outros modos de transporte similares, possibilitando que as vigas por sobre as quais se desloca sejam menos robustas, diminuindo os custos de construção e de instalação do sistema [Britto, 2008]. A Figura 1.1 mostra o Aeromovel em operação.

Figura 1.1 Aeromovel na Alemanha (Feira de Hannover, 1980)

A solução adotada para remover o sistema de potência do veículo foi utilizar tecnologia pneumática de propulsão. O veículo desloca-se sobre uma viga vazada no interior, caracterizando um duto fechado, no qual circula ar pressurizado por ventiladores. O veículo possui anteparos, chamados de "aletas", que bloqueiam a passagem do ar e dividem o duto em câmaras distintas. Válvulas presentes no duto, chamadas de válvulas atmosféricas (VA), permitem o controle da passagem do ar existente nas câmaras à atmosfera. Pela atuação destas válvulas e dos ventiladores cria-se uma diferença de pressão nas câmaras que proporciona a movimentação do veículo. Assim, o sistema comporta-se analogamente a um grande pistão pneumático, fornecendo pressão a um dos lados da câmara enquanto o outro é despressurizado através da sua conexão à atmosfera. As válvulas atmosféricas assumem atualmente apenas duas configurações de operação, abertas ou fechadas [Britto, 2008].

Na Figura 1.2 é apresentado um diagrama esquemático do veículo estacionado em uma estação. Parte da seção da viga está em corte, onde é possível visualizar uma das aletas do veículo e uma válvula atmosférica (VA). Os trilhos, os truques do veículo e a vedação longitudinal, por onde passa o mastro da aleta, são também apresentados [Britto, 2008].

O Aeromovel utiliza a tecnologia ferroviária na sua interface entre o veículo e o piso. Desta forma, tem-se um menor consumo energético por ser o atrito metal/metal inferior ao atrito, por exemplo, borracha/concreto. O veículo possui truques com quatro rodas independentes. A independência das rodas permite que o Aeromovel faça curvas com menores raios que os trens convencionais, que têm rodas fixas nos eixos. As aletas são articuladas, o que permite que o veículo faça curvas e se desloque em aclives e declives sem que esta se choque com a parede do duto [Britto, 2008].

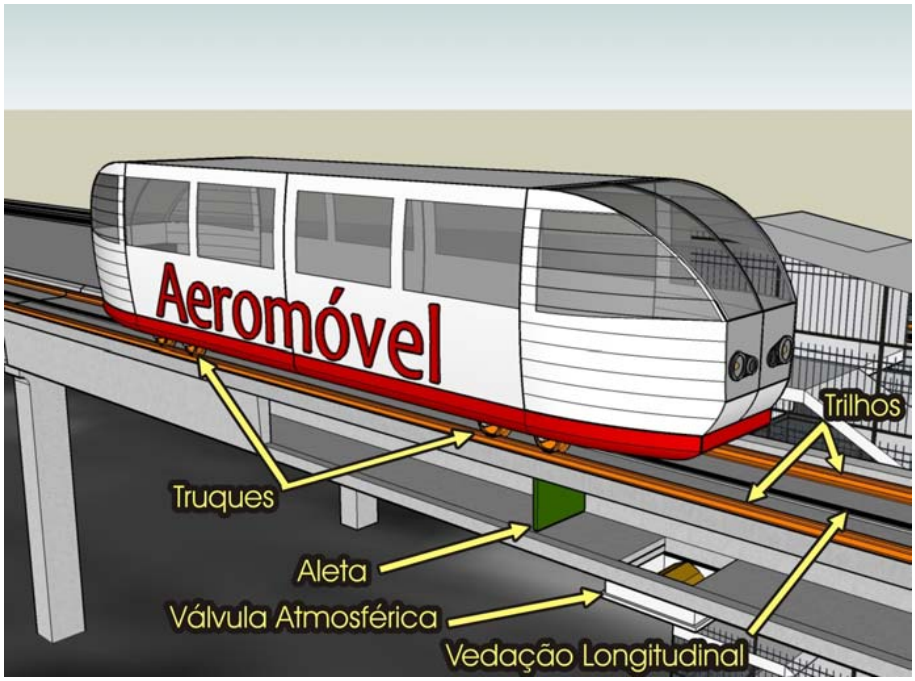


Figura 1.2 Veículo Aeromovel [Britto, 2008]

Em cada roda estão instalados freios a disco. O sistema de frenagem conta com o sistema de controle anti-bloqueio (*anti-blocking brake system* - ABS), garantindo a parada do veículo na estação e o não travamento das rodas [Britto, 2008]. O seu acionamento é hidráulico e com intertravamento com o acionamento das portas e controlado por CLP (Controlador Lógico Programável) [Furtado, 1994].

A unidade de potência, denominada como Grupo Moto Propulsor (GMP), é responsável pela geração da pressão diferencial que atua na aleta. Os GMP são constituídos basicamente por um motor elétrico assíncrono que aciona o ventilador industrial centrífugo [Furtado, 1994]. Cada GMP é conectado ao duto principal através de uma canalização [Britto, 2008].

O sistema de potência fluídica é formado por um ventilador industrial centrífugo e um conjunto de quatro válvulas de quatro posições, chamadas de Válvulas de Direcionamento de Fluxo (VDF). As válvulas possibilitam alternar o efeito do ventilador sobre o duto principal por onde se movimenta o veículo, podendo realizar a insuflação ou a exaustão do ar. As válvulas podem assumir as seguintes posições: abertas, 1/3 do curso, 2/3 do curso e fechadas. Isto possibilita o controle escalonado da pressão e, consequentemente, da força imposta ao veículo [Britto, 2008].

Na Figura 1.3 é apresentado um diagrama esquemático do GMP com dois motores instalados no mesmo eixo. Neste diagrama, as válvulas VDF estão configuradas de modo a exaurir o ar do duto para a atmosfera. A inversão nas atuações das válvulas produz o efeito contrário, de insuflação [Britto, 2008].

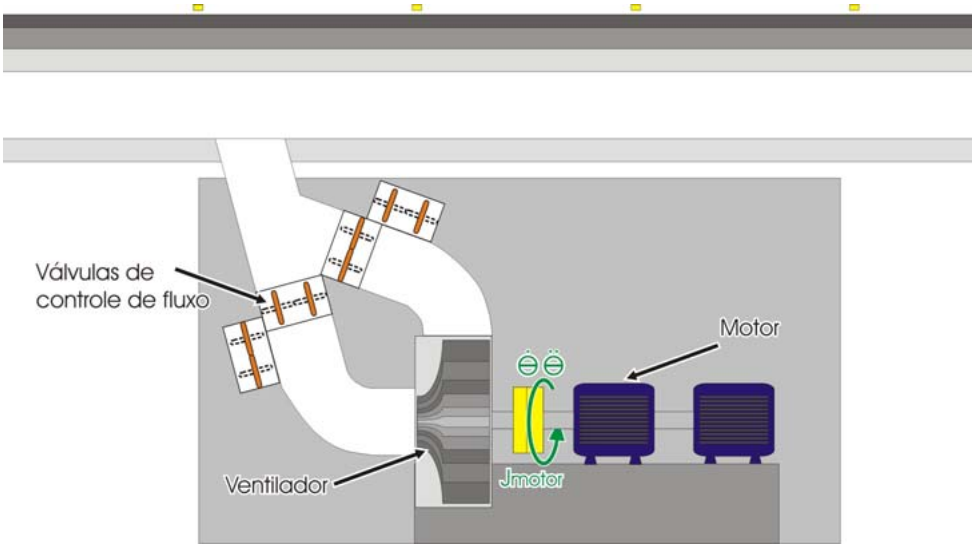


Figura 1.3 Desenho esquemático do Grupo Moto Propulsor (GMP) [Britto, 2008]

Outros elementos utilizados pelo sistema Aeromovel são as Válvulas de Isolamento de Trecho (VIT) e os Aparelhos de Mudança de Via (AMV). As VIT são válvulas que se encontram no interior dos dutos. Elas proporcionam a sua divisão e formação de câmaras internas de volumes menores. Assim, os veículos podem locomover-se de forma independente em um mesmo circuito. Já, os AMV possibilitam a mudança do veículo de uma via para outra [Britto, 2008].

Segundo Aeromovel, 1999, o sistema padrão completo pode ser segmentado em trechos compreendidos por duas estações, sendo estes denominados de "Blocos-Padrões". O Bloco Padrão é formado por dois GMP, um em cada estação, duas VIT, duas VA e um veículo. Esta configuração permite três tipos de operação do sistema, que são:

- Operação por Pressão ("Push") - o veículo é "empurrado" pela pressão provocada pela operação do GMP à montante do veículo. Na câmara a jusante do veículo, a VA é aberta, comunicando o duto à atmosfera.
- Operação por Depressão ("Pull") - o veículo é "puxado" pela depressão provocada pela operação do GMP à jusante do veículo. Na câmara a montante do veículo, a VA é aberta, comunicando o duto à atmosfera.
- Operação por Pressão-Depressão ("Push-Pull") - Os dois GMP estão conectados ao duto e as duas válvulas atmosféricas são fechadas. Assim, o veículo se locomove devido à pressão a montante e à depressão a jusante. Nesta forma de operação, o veículo pode desenvolver as maiores velocidades. Na Figura 1.4 é apresentado o desenho esquemático do Bloco Padrão.

Além destes três tipos de operação, o veículo apresenta opções distintas de frenagem. O

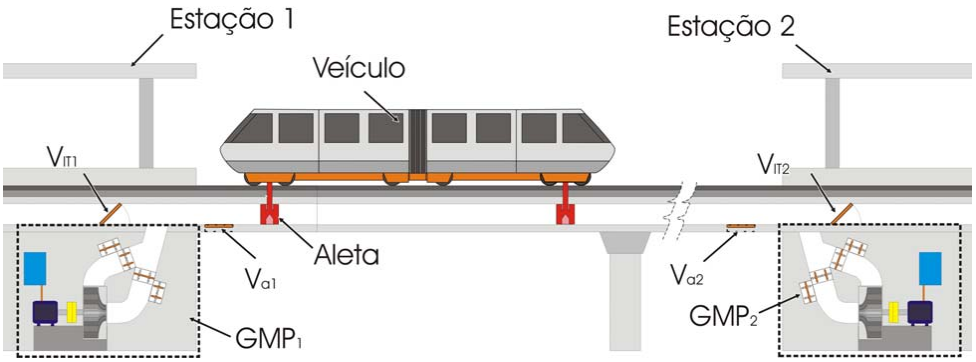


Figura 1.4 Desenho esquemático do percurso entre duas estações [Britto, 2008]

Aeromovel pode frear através do acionamento do freio a disco, com as válvulas atmosféricas abertas e dos GMP fechadas, ou pode frear fechando uma ou as duas válvulas atmosféricas, diminuindo sua velocidade por efeito do diferencial de pressão devido à movimentação do veículo, a qual ocasiona a compressão do ar à jusante e a expansão do ar a montante. Outro modo de frenagem é pela própria ação do GMP, de forma a fornecer força contrária ao movimento do veículo. Todos estes meios de atuação proporcionam uma redundância importante de segurança de operação [Britto, 2008].

Na Tabela 1.2, são apresentados os tempos de acionamento de cada uma das válvulas presentes no sistema GMP de acordo com Aeromovel, 1999.

Tabela 1.2 Intervalos de tempo aproximados, relativos ao Sistema Aeromovel

Tempo de abertura ou fechamento de válvulas (on/off)	1,5 s
Tempo de abertura ou fechamento das válvulas proporcionais	5 s
Acionamento de válvulas em casos de emergência	0,7 s

Fonte: Aeromovel, 1999

Na Tabela 1.3, são apresentados as velocidades e distâncias envolvidas no sistema controle de acordo com Aeromovel, 1999.

Tabela 1.3 Velocidades e distâncias relativas ao sistema Aeromovel

Aceleração	0.6 a 1.3 m/s^2
Velocidade de cruzeiro	80 km/h
Desaceleração em condições normais	0.7 a 0.85 m/s^2
Desaceleração máxima em condição de emergência	1.3 m/s^2
Distância entre propulsores	400 a 1400 m

Fonte: Aeromovel, 1999

As grandezas de velocidade, distância e principalmente o tempo de acionamento dos equipamentos responsáveis para o controle do veículo são essenciais para a posterior modelagem e dimensionamento do sistema de controle.

Sendo o sistema Aeromovel de transporte classificado como um APM, a seguir é apresentada a organização e os benefícios da aplicação deste conceito de automação em sistemas de transporte.

1.4 Controle do Sistema APM

O controle do Sistema APM realiza automaticamente o controle de movimento, a execução da segurança e operações de direcionamento das composições. A realização automática destas funções é de responsabilidade do sistema ATC (*Automated Train Controller*) o qual inclui os seguintes os subsistemas:

- ATP - *Automatic Train Protection*. Proteção contra colisões, excesso de velocidade, invasões de vias, entre outras condições de perigo.
- ATO - *Automatic Train Operation*. Controle de velocidade, paradas programadas e controle de portas entre outras funções que de outra maneira seriam de responsabilidade do operador.
- ATS - *Automatic Train Supervision*. Monitora e realiza o ajuste de desempenho individual de cada composição para manter a agenda de saídas e chegadas.

Um ATC deve incluir obrigatoriamente o subsistema ATP e, opcionalmente pode incluir ATO ou ATS. Para estabelecer uma comunicação entre estes três subsistemas utiliza-se a norma *IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements* [IEEE, 2004] que descreve os requisitos funcionais e performance da comunicação para controle de composições. A característica básica do CBTC inclui:

- Determinação da localização da composição com alto grau de precisão e independente de sensores nos trilhos.
- Comunicação contínua entre a composição e os sistemas externos a ele.
- Processo de verificação das condições e controle para o ATP (*Automatic Train Protection*). Funcionalidades do ATO (*Automatic Train Operation*) e ATS (*Automatic Train Operation*) também podem ser realizadas.

Dentre os benefícios de uma operação automatizada tem-se [Yelloz, 2007]:

- Redução de custos no uso de rádio frequência (RF) e do Controle dos Trens Baseados em Comunicação (CBTC) necessários para o uso da metodologia de controle baseada em blocos móveis (ver Capítulo 2).

- Melhor qualidade de serviços com o aumento da frequência de composições, visto que não há condutor e não é necessária a espera de resposta do operador do sistema supervisório.
- Diminuição da espera nas estações por manter a regularidade (*headways* mínimos). Conforto, segurança e pontualidade em viagens curtas são também atrativos dos sistemas não tripulados.

Assim, Yelloz, 2007, observa que sistemas automatizados através do uso de Radio Frequência e Controle de Trens Baseados em Comunicação (CBTC) é o caminho para o melhor custo/benefício para aumentar a capacidade de transporte, não requisitando modificações significativas na infraestrutura.

1.5 Objetivos

O objetivo geral deste trabalho consiste na proposição de uma metodologia que garanta o projeto efetivo de controladores para utilização em sistemas de proteção, operação e supervisão de sistemas APM (*Automated People Movers*), utilizando como estudo de caso o Sistema Aeromovel.

Os objetivos específicos deste trabalho são os seguintes:

- determinar o estudo de caso para aplicação da metodologia;
- determinar as características físicas do Sistema APM;
- desenvolver o modelo de automação para APM utilizando Perspectiva Funcional/Estrutural para aplicação no Sistema Aeromovel;
- analisar e validar ferramentas para realização do desenvolvimento de algoritmos de controle e proteção;
- identificar as possíveis arquiteturas para a configuração dos respectivos controladores;
- desenvolver e validar modelos detalhados de comunicação, controle e proteção para aplicação no Sistema Aeromovel;
- aplicar técnicas de análise de forma a garantir segurança de funcionamento destes sistemas;
- propor a ampliação normalização envolvida para aplicação em Sistemas APM (IEC 61850);

- realizar a verificação formal e a simulação em busca das condições de falha no sistema;

1.6 Metodologia

Os objetivos deverão ser atingidos considerando a execução dos seguintes pontos:

- Pesquisa bibliográfica sobre sistemas APM e sobre o Aeromovel.
- Pesquisa sobre as normas aplicadas na automação de APM, identificando os requisitos de funcionamento e as lacunas a serem exploradas para o desenvolvimento de uma metodologia para automação de APM.
- Levantamento das funcionalidades necessárias para a operação do sistema de automação.
- Aplicação e implementação computacional da perspectiva funcional/estrutural.
- Realização de simulação e verificação utilizando os modelos de controle do sistema APM juntamente com os modelos do sistema da comunicação.

1.7 Organização Geral

No Capítulo 2 é apresentada uma revisão bibliográfica sobre as soluções já desenvolvidas para o sensoriamento e controle de sistemas APM, assim como é feita uma descrição do sistema proposto, envolvendo a norma IEC 61850 e sua integração para aplicação em sistemas APM. No Capítulo 3 é formalizada a proposta para automação do sistema Aeromovel de transporte. Ainda no Capítulo 3 é apresentada uma análise sobre as soluções utilizadas para o controle de sistemas APM, assim como são propostas soluções funcionais para o mesmo. Neste mesmo capítulo são também analisados e descritos os algoritmos utilizados e as funcionalidades fornecidas pelo sistema. Nos Capítulos 4 e 5 serão simulados e verificados formalmente, respectivamente, os modelos do sistema de controle de sistemas APM e do sistema de comunicação. Uma vez comprovados isoladamente os comportamentos dos modelos, os mesmos serão verificados em conjunto no Capítulo 6, enquanto que no Capítulo 7 é descrita a proposta de ampliação da norma IEC 61850 para aplicação em sistemas APM e no Capítulo 8 é apresentada a metodologia dos testes de conformidade do sistema implementado para uso em campo. No Capítulo 9 são apresentadas as conclusões do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentada a revisão bibliográfica visando a estabelecer a fundamentação teórica necessária para o pleno entendimento dos assuntos que seguem tais como sistemas pneumáticos, controle digital de sistemas, modelagem de sistemas automáticos e sistemas APM. É também apresentado o estudo do estado da arte quanto às soluções já desenvolvidas para o sensoriamento e controle de sistemas APM.

2.1 Fundamentação Teórica

Nesta seção são apresentadas as fundamentações teóricas sobre sistemas pneumáticos, controle digital de processos, modelagem de sistemas automáticos e sistemas APM.

2.1.1 Sistemas Pneumáticos de Posicionamento

Como descrito anteriormente, a propulsão do Aeromovel é exercida pela diferença de pressão sobre uma aleta conectada ao veículo, sendo análogo a um sistema de servoposicionamento pneumático, herdando, portanto, muitas de suas características.

2.1.1.1 Servoposicionamento Pneumático

O uso da pneumática em sistemas de posicionamento é comum em diversos sistemas automáticos tais como: acionamentos de robôs industriais, manipulação e movimentação de peças e material, máquinas industriais alimentícias, linhas de montagem e pequenas máquinas e sistemas automáticos utilizados em diferentes processos da manufatura [Virvalo, 1995].

Como vantagens da utilização de sistemas pneumáticos, observam-se o baixo custo quando comparados com outros sistemas tecnológicos, boa relação peso/potência, facilidade de manutenção, rápida atuação com altas acelerações, flexibilidade de instalação e utilização de tecnologia com fonte de energia disponível em grande parte das plantas industriais [Cruz, 2003].

Em contraste com as vantagens descritas acima, encontra-se presente em sistemas pneumáticos um comportamento oscilatório proveniente do atrito nos atuadores e da compressibilidade do ar [Andrighetto et al., 2005a]. Este comportamento é crítico principalmente em baixas velocidades de deslocamento [Guoliang e Xuanyin, 2003],

dificultando o seguimento de trajetórias com paradas intermediárias. O emprego da pneumática é, portanto, mais comum para situações de posicionamento ponto-a-ponto.

A servopneumática consiste de uma técnica de acionamento relativamente de baixo custo, onde o componente mais caro é a servoválvula. Devido a problemas oriundos da compressibilidade do ar e do atrito, a maior dificuldade de desenvolver aplicações precisas utilizando esta tecnologia está no seu controle. Sistemas pneumáticos geralmente necessitam de bons modelos matemáticos que representem as não linearidades inerentes ao sistema [Bobrow e McDonell, 1988]. Pesquisas sobre a compensação do atrito em atuadores pneumáticos vem sendo desenvolvidas por Perondi, 2002, por exemplo.

A evolução tecnológica, principalmente na eletrônica e no campo de *software*, tem proporcionado recursos computacionais que viabilizam o desenvolvimento de algoritmos mais complexos, sendo uma tendência do mercado o investimento maior em soluções em *software* frente aos investimentos em *hardware* [Sun Microsystems, 2004].

A solução proposta por parte dos pesquisadores na área [Perondi, 2002; Rajendran e Bolton, 2003] para superar as dificuldades encontradas no emprego da servopneumática está no desenvolvimento de algoritmos de controle. Estes algoritmos estão cada dia mais sofisticados e complexos, necessitando de sistemas de controle digital com maior flexibilidade e poder de processamento matemático. Trabalhos anteriores, como Vieira, 1998, demonstram a necessidade da utilização de frequências de controle maiores que 1 kHz devido à rápida resposta dos sistemas pneumáticos.

2.1.1.2 O Atuador Pneumático

Os sistemas que possibilitam posicionar uma carga mecânica a partir de um conjunto de coordenadas (x, y, z) são chamados de sistemas de posicionamento. Esta localização pode ser fixa ou variável no tempo e, nestes casos, os sistemas de posicionamento são, algumas vezes, chamados de seguidores.

Um componente importante do sistema é o elemento que aplica a força sobre a carga mecânica para levá-la até a posição desejada. Este elemento é chamado de motor ou atuador, podendo ser elétrico, hidráulico ou pneumático.

Atuadores pneumáticos têm como vantagem sobre acionamentos elétricos, a ausência de motores pesados, geralmente com sistemas complexos de transmissão por engrenagens [Liu e Bobrow, 1988]. O acionamento elétrico também é relativamente caro devido ao alto custo dos servomotores. Em aplicações de robótica móvel, a ausência de engrenagens, como nas

soluções nas quais se utiliza *músculos pneumáticos*, torna a servopneumática atrativa para o desenvolvimento de robôs flexíveis e com tamanho reduzidos [Granosik e Borenstein, 1998].

As equações de equilíbrio dinâmico de um cilindro pneumático, como mostrado no modelo da Figura 2.1, descrevem o funcionamento básico do sistema.

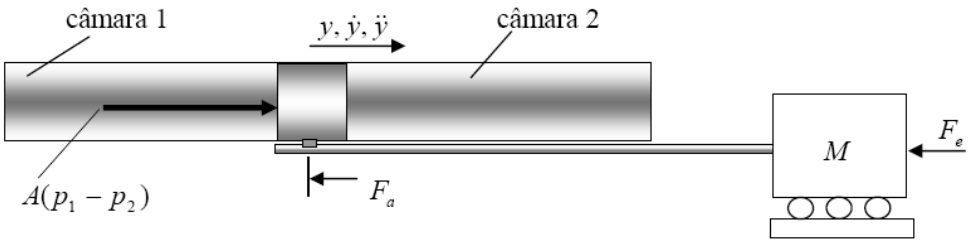


Figura 2.1 Equilíbrio de forças no êmbolo do cilindro atuador [Perondi, 2002]

Este equilibro de forças é obtido pela aplicação da 2^a lei de Newton:

$$M\ddot{y} + F_a + F_e = A(p_1 - p_2) \tag{2.1}$$

onde M é a massa deslocada, F_a é a força de atrito, F_e é a força externa, p_1 é a pressão na câmara 1, p_2 é a pressão na câmara 2, A é a área do êmbolo e $A(p_1 - p_2)$ é a força resultante da diferença entre as câmaras.

Para controlar a variação destas pressões assim como a posição, velocidade e aceleração do êmbolo, utilizam-se uma ou mais servoválvulas. Quando o sistema é realimentado com a posição (permitindo o controle de deslocamento), ele é denominado servoposicionador pneumático.

2.1.2 Problemas de Controle Digital de Sistemas

Para o controle preciso de seguimento de trajetórias é necessário utilizar tecnologias que permitam a aquisição de dados (posição e, muitas vezes, também as pressões nas câmaras), seu tratamento por algoritmos de controle e a posterior atuação no sistema físico. Este processo constitui o chamado controle digital de processos.

2.1.2.1 Controle Digital de Processos

A utilização de sistemas de controle avançados possibilita, por exemplo, a compensação do atrito, aumentando a precisão no uso da pneumática para fins de posicionamento preciso na indústria.

Este controle é realizado de forma digital e é atualmente, na maioria dos casos, baseado em microprocessadores ou microcontroladores, que executam processos continuamente [Kilian, 2000]. O controle de processo consiste, neste caso, na leitura da trajetória requerida e dos valores dos sensores e na utilização destes dados para calcular a saída enviada para o atuador, como mostra a Figura 2.2.

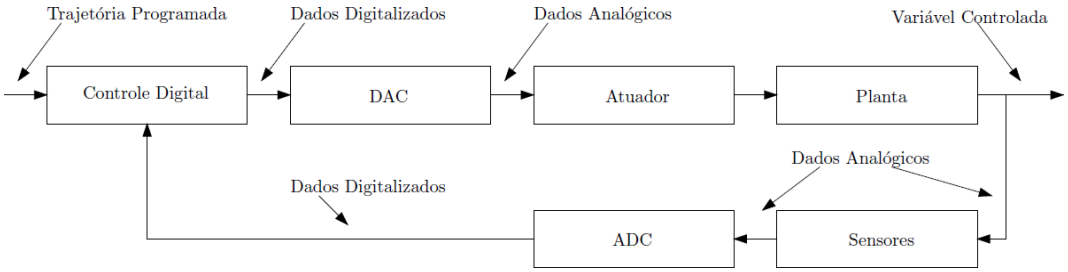


Figura 2.2 Controle digital de processos

Quanto menor o período de tempo necessário para a repetição deste processo, maior a possibilidade de precisão no posicionamento. Esta é a diferença fundamental quanto ao processo analógico que é contínuo e possui resposta muito mais rápida do que a grande maioria dos sistemas físicos controlados. Também, nos processos digitais, busca-se que o tempo entre cada interação seja muito pequeno quando comparado com o tempo de resposta do sistema físico controlado [Kilian, 2000].

Como já enfatizado, o avanço da tecnologia atual direciona-se mais à utilização de algoritmos de controle complexos comparado ao desenvolvimento de *hardware* específico. Isto torna o sistema flexível e diminui os custos de atualização ou manutenção, conforme ilustrado na Figura 2.3.

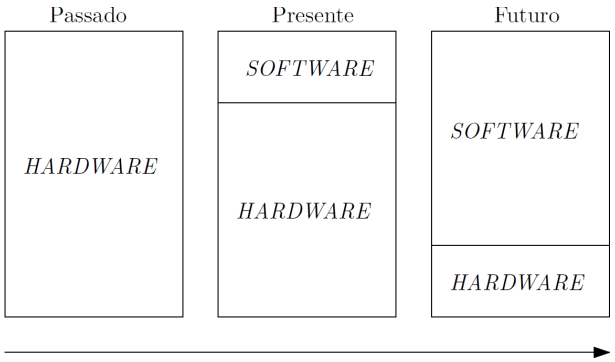


Figura 2.3 Tendências de desenvolvimento de *Software versus Hardware* [Sun Microsystems, 2004]

O *software* em sistemas APM é um componente essencial para operacionalização do sistema. *Firmware*, sistema operacional, *drivers*, compiladores, editores, interfaces de usuário,

sistema de *backup*, entre outros elementos, são necessários para disponibilizar um serviço seguro ao passageiro [Aldrich, 2005].

2.1.2.2 Soluções para o Controle Digital

Segundo Edge, 1997, pode-se desenvolver sistemas de controle baseados em três diferentes tipos de implementação: Analógica, Híbrida e Digital. Observa-se também uma tendência na utilização de controladores com maior flexibilidade (ver Figura 2.3), ou seja, controladores com maior percentual de soluções adequadas a cada tipo de aplicação. Muitos dos controladores digitais para movimento oferecem facilidades de programação, porém, os algoritmos de controle são relativamente fixos e não modificáveis depois de implementados [Edge, 1997].

O principal sinal adquirido em grande parte das técnicas de controle de servoposicionadores pneumáticos, é a posição do êmbolo do cilindro pneumático. A partir da derivação numérica deste sinal pode-se obter informação sobre velocidade e aceleração. Estas informações são geralmente utilizadas pelos algoritmos de controle para determinar a intensidade do sinal de atuação na servoválvula. Grande parte dos trabalhos estudados determinam o tempo de atualização do sinal de controle em 1 ms [Vieira, 1998; Andrighetto et al., 2005a, 2005b; Perondi, 2002; Giberti et al., 2001]. Na bibliografia encontra-se também valores de 3 ms [Latino e Sandoval, 1996], 4 ms [Pu et al., 1992] e 5 ms [Virvalo, 1989].

Devido à significativa queda nos preços e avanço da eletrônica, a utilização de centrais únicas de processamento (CPU - *Central Processing Unit*) para controle de processos tem sido mais difundida e, dependendo das necessidades da aplicação, pode-se utilizar diferentes tipos de implementação: Controlador Dedicado, Controle Distribuído e Controle Orientado a Objetos.

Controlador Dedicado

O controlador dedicado (Figura 2.4) é um sistema isolado (*stand-alone*) com uma única central de processamento, responsável por processar todas as funções do sistema, incluindo o monitoramento, interfaceamento com o usuário e operações em tempo real (tais como a realimentação de controle, decisões quanto a segurança, tratamento de dados, etc.). Para isto são necessárias políticas de sincronização entre as tarefas, assim como, acesso aos dispositivos (sensores e transdutores).

Vários trabalhos utilizam controladores dedicados para a realização das tarefas de desenvolvimento de controladores e monitoramento dos sistemas. Como exemplo, pode-se

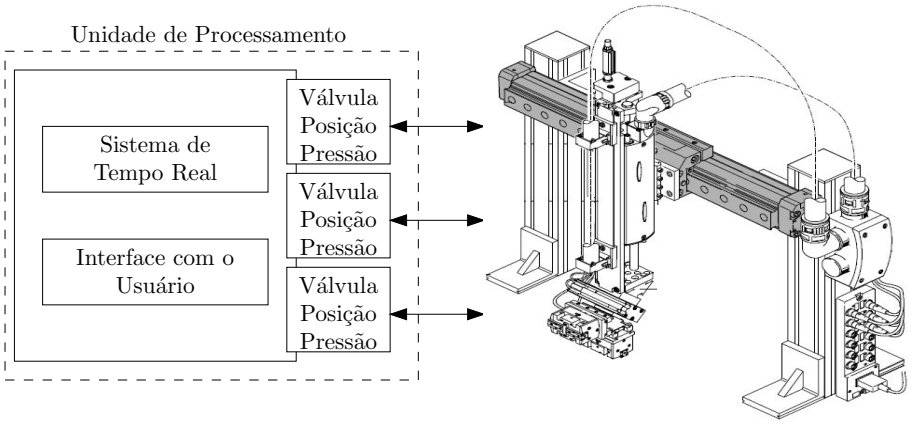


Figura 2.4 Controle dedicado

citar Perondi, 2002, Vieira, 1998, e Andrighetto et al., 2005b (estes autores utilizam placas de controle baseadas em DSP (*Digital Signal Processor*)). Em Kunz, 2006, descreve-se a montagem de uma bancada pneumática de testes onde a programação e o processamento em tempo real dos algoritmos foram baseados no sistema operacional GNU/Linux com as modificações do projeto RTAI [DIAPM, 2005], e o software de programação dos algoritmos de controle e visualização dos dados de operação é o Scilab-Scicos [Bucher et al., 2004]. Neste caso, o tempo de ciclo de controle utilizado foi de 1 ms e a interface com o sistema físico foi realizada por uma placa comercial (CIO-DAS Jr) AD/DA acoplada ao computador.

Controladores Distribuídos

Os controladores distribuídos são unidades de processamento em tempo real dedicados individualmente a cada dispositivo. As unidades estão ligadas a um barramento de comunicação que estabelece a troca de informações com outra unidade de processamento responsável pelo interfaceamento com o usuário, diminuindo assim a solicitação de processamento individual do sistema e facilitando a utilização de interfaces mais *amigáveis*, por disponibilizar maiores recursos. Na Figura 2.5 é apresentado o exemplo do controlador distribuído aplicado ao servoposicionador pneumático.

Em geral, a decisão de utilizar um sistema de controle distribuído é motivada pela redução nos custos e aumento da flexibilidade do sistema de controle. Em Lages e Alt, 2003, é apresentada uma arquitetura de controle distribuída para um robô manipulador através de uma rede IP (*Internet Protocol*) convencional. Nesse trabalho, são propostas técnicas para aumentar o desempenho do controlador, diminuindo os atrasos inerentes às comunicações em redes IP.

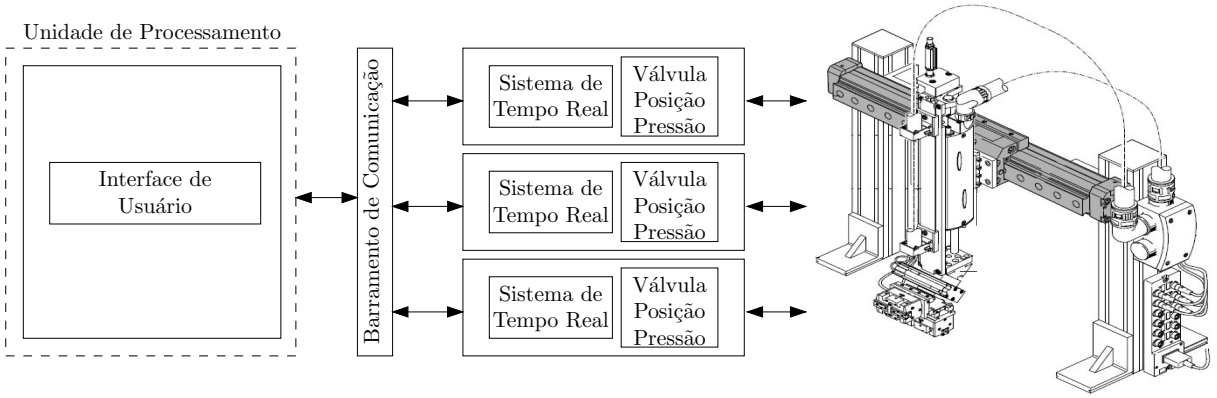


Figura 2.5 Controle distribuído

Controladores Orientados à Objetos

O esquema de controle orientado a objetos é similar ao de controladores distribuídos, porém, o tratamento dispendido a cada dispositivo (válvula, sensor de posição, sensor de pressão e configuração do controle em tempo real) utiliza a filosofia de orientação a objetos, abstraindo informações sobre implementações específicas de cada dispositivo, resultando em uma maior padronização de interfaces e de *softwares* para construção de algoritmos de controle [Edge, 1997].

No controle orientado a objetos encontram-se características intrínsecas da utilização desta filosofia de programação, tais como: alto grau de reutilização, portabilidade, modularidade e fácil aprendizagem. A orientação a objetos, a partir do desenvolvimento de novas ferramentas, vem sendo crescentemente utilizada em aplicações de automação [Pereira et al., 2005].

A utilização de ferramentas orientadas a objetos utilizando política e requisitos de tempo real apresentam-se bastante promissoras. Um exemplo disto é a RTSJ (*Real Time Specification for Java*) que permite o desenvolvimento de aplicações de tempo real utilizando a linguagem Java. Porém, trabalhos como Pereira et al., 2005, destacam alguns aspectos que merecem especial atenção devido a existência de melhorias necessárias para sua utilização em aplicações críticas.

Visto que atrasos na comunicação podem ter efeitos críticos no sistema de proteção do sistema de controle de tráfego, faz-se necessário utilizar requisitos de tempo crítico para tais funcionalidades.

2.1.2.3 Sistemas de Tempo Real Críticos

Tanto o sistema de aquisição e acionamento, quanto a técnica de controle necessitam de garantias temporais para a eficiência de seu funcionamento. Para possibilitar um controle temporal seguro é necessário que a implementação destes sistemas esteja baseada em um sistema de tempo real.

O termo *tempo real* é aplicado de diferentes formas, de acordo com a área de conhecimento onde é utilizado. No presente trabalho, utiliza-se esse termo no sentido consagrado na Ciência da Computação, onde ele é dividido em duas grandes áreas: tempo real "probabilístico" (*soft real-time*) e tempo real "determinístico" (*hard real-time*) [Dozio e Mantegazza, 2003].

Sistemas *soft real-time* são caracterizados pela habilidade do sistema de executar um processo garantindo a média do tempo determinado durante sua execução. Processos *soft real-time* geralmente são empregados em soluções onde variações, ou mesmo atrasos no tempo previsto, ocorrem sem comprometimento do funcionamento do sistema. Dispositivos de vídeo geralmente utilizam *soft real-time*. O dispositivo de vídeo troca informações de modo que pequenas variações no desempenho ou mesmo a supressão de *frames* (imagens estáticas), são aceitáveis, pois são "quase" imperceptíveis ao olho humano. Em geral, as interfaces homem-máquina não necessitam de determinismo temporal acurado.

Sistemas *hard real-time* são caracterizados pela existência de garantias temporais mais rígidas, ou seja, maior grau de confiabilidade no sentido de que não serão ultrapassados os limites preestabelecidos durante o desenvolvimento do sistema. Assim, sistemas *hard real-time* não utilizam a média para compensar possíveis atrasos. Um exemplo de aplicação típica está no controle de plantas industriais.

Em sistemas *soft real-time* ocorrem variações quanto a estimativa do instante de tempo em que determinada ação será executada, sendo assim, pode ser representada na forma de uma função de distribuição probabilística, enquanto em sistemas *hard real-time* estas variações são nulas. Pode-se comparar sistemas *hard real-time* (determinísticos) e *soft real-time* (probabilísticos) através do gráfico da Figura 2.6.

Outra característica de sistemas *hard real-time* é a baixa latência, ou seja, o baixo tempo de atraso na resposta entre o comando de execução e a real execução do processo. O conceito de latência está ilustrado na Figura 2.7.

Na Figura 2.7, além da latência, pode-se observar outros conceitos fundamentais na análise de sistemas de tempo real. O "processo recorrente" é o conjunto de comandos que devem

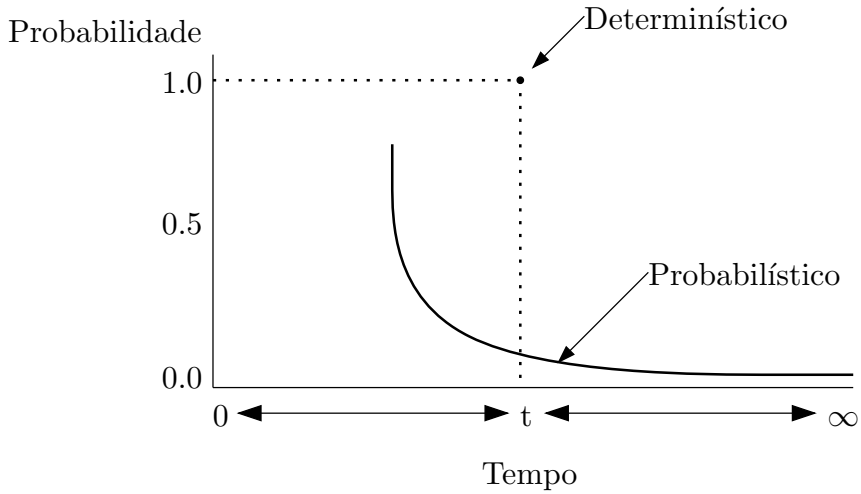


Figura 2.6 Sistemas determinísticos *versus* probabilísticos [Sun Microsystems, 2004]

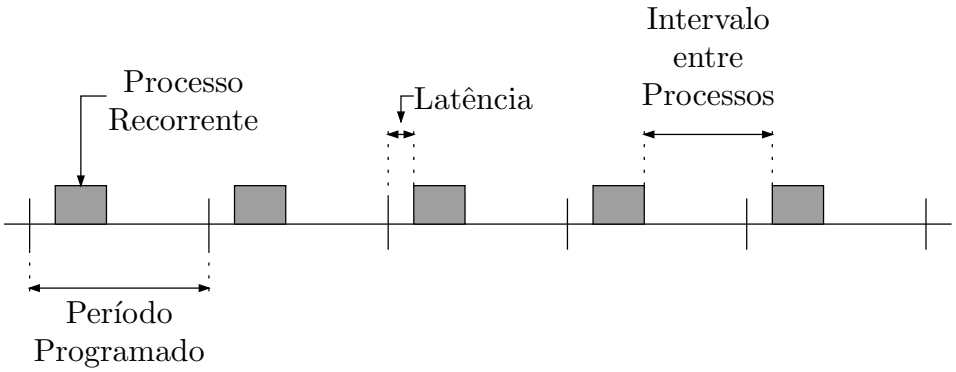


Figura 2.7 Informações temporais sobre processos

ser repetidos de acordo com o "período programado". Observa-se que o processo recorrente não utiliza todo o tempo disponível para sua execução, o restante é utilizado por outros possíveis processos em execução no sistema, tais como interrupções, acesso a disco, etc.

Ao determinar o "período programado", devem ser considerados, além da latência do sistema, e do tempo necessário para completar os comandos do processo recorrente, os possíveis atrasos e a possibilidade de falhas no sistema. O tratamento das possíveis falhas aumenta a integridade do sistema e pode ser realizado a partir de mensagens de alarme ou parada de dispositivos atuantes, evitando danos ao sistema controlado. As mensagens de falha são programadas para o caso em que o tempo de execução de um processo recorrente ultrapasse o tempo previsto.

Geralmente, centrais de processamento, como estações de trabalho, não são projetados para aplicações do tipo *hard real-time*. Isto ocorre devido à falta de disponibilidade de informações sobre as latências do *hardware* utilizado. Assim, grande parte dos controladores com requisitos de tempo real são implementados em sistemas dedicados, como processadores digitais de sinais (*DSPs - Digital Signal Processors*) e microcontroladores, os quais possuem

maiores garantias quanto a latência devido à baixa quantidade de instruções utilizadas para processar as interrupções [Dozio e Mantegazza, 2003].

Os maiores benefícios da utilização de microcomputadores para o controle digital está na flexibilidade e na grande capacidade de processamento, chegando a milhões de operações de ponto flutuante por segundo (FLOPS - *Floating-Point Operations per Second*), que aumenta significativamente a eficiência no processamento matemático. Esta capacidade possibilita a rápida resposta do ciclo de controle, mesmo com o uso de técnicas de controle complexas. Acrescenta-se também a possibilidade de utilizar sistemas de alto desempenho, como *Symmetric Multi Processors* (SMP), com arquitetura de *hardware* com mais de um processador na mesma unidade de processamento, resultando em uma melhor relação entre custo e capacidade de processamento.

Devido ao rápido desenvolvimento tecnológico (tanto em *software* quanto em *hardware*), o acesso a microcomputadores está muito facilitado. Como existe uma grande disponibilidade de microcomputadores no mercado, alguns projetos vêm sendo desenvolvidos para a construção de sistemas operacionais com suporte a *hard real-time* em computadores pessoais. Muitos destes sistemas, como será visto nas seções que seguem, funcionam em equipamentos com tecnologias consolidadas e bastante difundidas, diminuindo os custos para o desenvolvimento de soluções para controle em tempo real.

Atualmente, encontram-se disponíveis boas soluções de sistemas com suporte a *hard real-time* abertas ou *Open Source*. Estas soluções possuem como características:

- livre utilização e distribuição (gratuito);
- permitem acesso também livre às tecnologias utilizadas (código fonte);
- permitem a alteração do código fonte.

2.1.2.4 Sincronização de Tempo

A sincronização dos relógios dos diferentes sistemas computacionais envolvidos no controle de Sistemas APM é de vital importância, principalmente quando os mesmos estão distantes fisicamente entre si e há uma interdependência de funções. Por exemplo, no caso de veículos automáticos, quando o sistema computacional de controle de freio está fisicamente distante do sistema que lê sinais de posição de outras composições que trafegam na mesma via.

Dentre as principais soluções pesquisadas estão as tecnologias IRIG-B, *Network time protocol* (NTP) e o IEEE 1588. Para o caso do uso de IRIG-B ou IEEE 1588 há a necessidade

de equipamento especial para sincronismo [Eidson, 2006]. De acordo com o fabricante de equipamentos para sincronismo RuggedCom [Ruggedcom, 2010] obtém-se 1 ms de resolução para o protocolo NTP, 100 ns para o IEEE 1588 e 100 μs para o GPS (IRIG-B).

Será avaliada a necessidade de resolução de tempo no sincronismo dos relógios, de acordo com dados obtidos das normas de controle de Sistemas APM e testes de campo a serem realizados. Inicialmente, será utilizado o protocolo NTP, pois este não necessita de equipamentos dedicados para tal função, visto que esta tecnologia utiliza a infra-estrutura de rede *ethernet*.

2.1.2.5 A Norma IEC 61850

A seguir, é apresentada a norma IEC 61850, pois, conforme o estudo realizado sobre normas para APM, que será apresentado na Seção 2.1.4, esta satisfaz os requisitos para o sistema de proteção, operação e supervisão de Sistemas APM.

Segundo Hewings, 2008, os sistemas de proteção e controle de composições são tradicionalmente baseadas em circuitos cabeados e centralizados e apesar de estes sistemas apresentarem, geralmente, um *design* simples, eles comportam significativas dificuldades na instalação e na manutenção. Assim, conforme há um aumento de demanda ao sistema, há vantagens na escolha de uma arquitetura aberta, com um sistema de comunicação simples. Estes conceitos são largamente abordados no desenvolvimento da norma IEC 61850, criada para ser um padrão de comunicação para subestações de energia elétrica baseada no uso de IEDs (*Intelligent Electronic Devices*), que ocupam o lugar de antigos relés de proteção, combinando funções de proteção, controle e comunicação no mesmo equipamento. De forma geral, sua aplicação resulta nos seguintes benefícios [Hewings, 2008]:

- redução de cabeamento;
- redução do custo e tempo de instalação;
- aumento da capacidade de monitoramento dos sistemas de controle e proteção;
- infra-estrutura separada da funcionalidade;
- interoperabilidade;
- uso de linguagem de configuração do sistema (*System Configuration Language* - SCL) abrangente para o ciclo de vida desde a concepção à engenharia, operação e manutenção.

A norma IEC 61850 apresenta requisitos tais como controle em tempo-real, distribuído

e orientado a objetos. Em suas aplicações são encontrados problemas similares aos de sistemas de transporte, tais como, o controle de sistemas distribuídos geograficamente, onde o tempo de transmissão de um sinal elétrico é significativo e, portanto, considerado nas especificações do projeto.

É importante ressaltar o fato de que sistemas APM, assim como Sistemas de Automação de Subestações (SAS), necessitam de tomadas de decisões rápidas que, frequentemente, devem ser realizadas a partir de dados transmitidos a grandes distâncias. A IEC 61850 supre este requisito através de protocolo baseado no padrão *Ethernet* associado ao uso de fibra óptica. Entretanto, diferentemente dos demais protocolos com requisito de tempo real para longas distâncias [Larsson, 2005], a IEC 61850 difere pelo fato de, além de tratar também o modelo de comunicações de dados, trata o modelo de dados e serviços. O modelo de dados apresenta um dicionário de nomes padrão e uma estrutura hierárquica de objetos baseados nas funcionalidades existentes nas subestações e usinas do sistema elétrico [Freitas e Carmo, 2009].

A norma IEC 61850 provê um padrão para integração de subestações a partir da especificação dos requisitos de comunicação, características funcionais, estrutura de dados nos dispositivos e nomenclatura para os dados. Provê também padrões para características operacionais, tais como, a forma das aplicações interagirem com o controle dos dispositivos e como devem ser testados para avaliar a conformidade do sistema.

Com relação aos requisitos de controle de sistemas APM, a IEC 61850, sendo baseada em ASN.1, está em conformidade com a norma *IEEE Trial-Use Standart for Message Set Template for Intelligent Transortation Systems* [IEEE, 2000b]. A norma IEC 61850 utiliza os seguintes protocolos de comunicação:

- *Generic Object Oriented Substation Event* (GOOSE). Utilizada para mensagem de falha. Esta mensagem é enviada para rede com alta prioridade e requisitos de tempo real;
- *Sample Value* (SMV). Utilizada para troca de dados entre equipamentos. Através desta mensagem é possível realizar processamento de sinais de equipamentos distribuídos. Também possui requisitos de tempo real;
- *Manufacturing Messaging Specification* (MMS). Norma internacional ISO 9506 [ISO, 2003], utilizada para comunicação do sistema supervisório e equipamentos e para configuração remota. Não possui requisitos de tempo real.

A norma IEC 61850 é dividida em 15 documentos agrupados em 10 diferentes partes:

- IEC 61850-1 *Communication networks and systems in substations - Introduction and overview*. Este documento apresenta uma introdução sobre a norma [IEC, 2003a];

- IEC 61850-2 *Communication networks and systems in substations - Glossary*. Este documento apresenta a terminologia, definições e termos utilizados em diferentes partes padrão IEC 61850 [IEC, 2004a];
- IEC 61850-3 *Communication networks and systems in substations - General requirements*. Descreve os requisitos de confiabilidade, mantenabilidade, disponibilidade entre outros itens relacionados à segurança do sistema [IEC, 2002a];
- IEC 61850-4 *Communication networks and systems in substations - System and project management*. Qualidade e responsabilidade de serviço, tipos de testes, sistemas de teste, bem como a vida útil aceitável para os equipamentos [IEC, 2002b];
- IEC 61850-5 *Communication networks and systems in substations - Communication requirements for functions and device models*. Resolve os requisitos de interoperabilidade, funções e desempenho [IEC, 2003b];
- IEC 61850-6 *Communication networks and systems in substations - Configuration description language for communication in electrical substations related to IEDs*. Define a linguagem utilizada para configuração do sistema de automação. A linguagem utilizada é a XML (*eXtensible Markup Language* [IEC, 2004b];
- IEC 61850-7-1 *Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Principles and models*. Define a estrutura básoca de comunicação [IEC, 2003c];
- IEC 61850-7-2 *Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Abstract communication service interface* (ACSI). Define as interfaces de serviços da estrutura básica de comunicação [IEC, 2003d];
- IEC 61850-7-3 *Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Common data classes*. Define a organização de dados da estrutura básica de comunicação [IEC, 2003e];
- IEC 61850-7-4 *Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Compatible logical node classes and data classes*. Este documento descreve 92 nós lógicos ou funções necessárias e opcionais para o controle incluindo a forma de associação entre elas tais como proteção, controle e medição [IEC, 2003f];
- IEC 61850-7-410 *Communication networks and systems for power utility automation -*

Hydroelectric power plants - Communication for monitoring and control. Define os nós lógicos necessários para aplicação em usinas hidroelétricas [IEC, 2007];

- IEC 61850-8-1 *Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to* ISO/IEC 8802-3. Especifica a comunicação entre os equipamentos e o sistema de supervisão da planta. O protocolo utilizado é o MMS (*Manufacturing Message Specification*) e o ISO/IEC 8802-3 [IEC, 2004c];
- IEC 61850-9-1 *Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Sampled values over serial unidirectional multidrop point to point link*. Mapeia os elementos de comunicação de tempo-real que utilizam de valores amostrados (*sampled measured values*) [IEC, 2003g];
- IEC 61850-9-2 *Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Sampled values over* ISO/IEC 8802-3. Mapeia os elementos de comunicação baseados em comunicação em tempo-real de valores amostrados (*sampled measured values*). Mapeia os elementos de comunicação de tempo-real que utilizam as mensagens GOOSE (*generic object-oriented system events*) [IEC, 2004d];
- IEC 61850-10 *Conformance Testing*. Define os procedimentos para teste de conformidade dos dispositivos IEC 61850 [IEC, 2005].

2.1.2.6 Hierarquia e Classificação

Segundo a norma IEC 61850, o sistema de controle é dividido em três níveis de acordo com a hierarquia de funcionalidade:

- Funções do Nível de Processo: onde se localizam todas os equipamentos que apresentam interfaces diretas com o processo por meio do uso de sinais analógicos ou binários para indicação do estado do sistema. Tradicionalmente, estes sinais são transmitidos via fiação na forma de intensidade de corrente ou tensão elétrica.
- Funções do Nível de Função: onde se localizam todas as funções que atuam diretamente nos equipamentos do nível de processo. O chaveamento de equipamentos é uma das suas funcionalidades típicas. Neste nível se encontram as lógicas de controle dos equipamentos do nível de processo.
- Funções do Nível de Estação: onde se localizam todas as demais funcionalidades. Estas

são divididas em dois grupos:

- Relacionadas aos Processos: Funções que utilizam dados de mais de um nível de função. Um exemplo é uma função que, para ser exercida, necessita de dados provenientes de múltiplos controladores.
- Relacionadas às Interfaces: Funções relacionadas a comunicação com sistemas HMI (*Human Machine Interface*), SCADA (Sistemas de Supervisão e Aquisição de Dados) ou com uma estação remota.

Atualmente, há aplicações da IEC 61850 nas áreas de hidroelétricas, geração distribuída e energia eólica. Propõem-se, no presente estudo, a ampliação da norma IEC 61850 para aplicação em sistemas APM, realizando um controle CBTC.

Uma das principais partes da IEC 61850 consiste no documento IEC 61850-10 (*Conformance Testing*), o qual define os procedimentos para teste de conformidade dos dispositivos IEC 61850 [IEC, 2005]. Assim, podem-se utilizar equipamentos já existentes e também uma normativa de conformidade para a implantação e avaliação da automação dos equipamentos utilizados nos sistemas APM.

A Figura 2.8 ilustra a organização dos dados da norma IEC 61850, onde se pode observar que cada dispositivo físico (o *hardware* propriamente dito) contém um ou mais dispositivos lógicos (IED) e que cada dispositivo lógico contém um ou mais nós lógicos (funções do IED). Cada nó lógico contém seus respectivos dados organizados em uma classe de dados. Esta hierarquia de dados está descrita abaixo:

- Dispositivo Físico (*Physical Device*). Objeto acessado através de um endereço de rede correspondente.
- Dispositivo Lógico (*Logical Device*). Contém uma coleção de nós lógicos implementados em um IED, isto é, os nós lógicos associados a um dispositivo lógico são do tipo não distribuídos.
- Nó Lógico (*Logical Node*). Funções que representam as funções reais no sistema.
- Dados e Atributos (*Data/Data Attribute*). Propriedades dos nós lógicos como, por exemplo, dados de sensores ou configuração do equipamento.

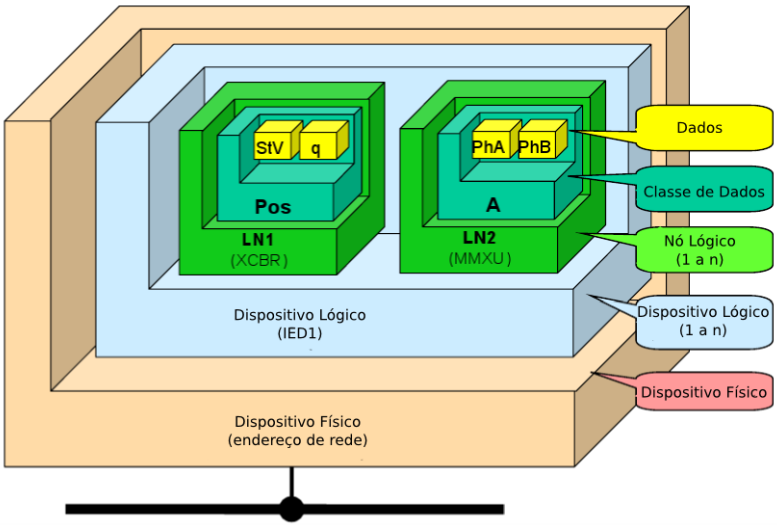


Figura 2.8 Organização dos dados [IEC, 2007]

2.1.3 Representação e Modelagem de Sistemas Automáticos

Com relação a representação e modelagem de sistemas automáticos, a pesquisa bibliográfica se centrou nas estruturas, formalismos e *softwares* que foram utilizados para o desenvolvimento do trabalho, em virtude do vasto campo em estudo.

Segundo Negri, 1996, a descrição de sistemas automáticos é subdividida em perspectivas estruturais, funcionais e comportamentais, de tal modo que a junção dos mesmos forneça a descrição completa do sistema.

2.1.3.1 Perspectiva Funcional

A estrutura funcional é uma representação diagramática que serve para descrever os relacionamentos do sistema técnico por meio de fluxos de energia, material e de sinal. A partir da função global trabalha-se no desdobramento de funções mais simples [Matos, 2007].

Em um primeiro momento, a representação funcional é suficiente para o entendimento do sistema técnico, mas, com o aumento da complexidade e a integração de diferentes tecnologias, pode surgir a necessidade de uma abordagem mais detalhada [Matos, 2007].

A função global é desempenhada pela interação de vários dispositivos, cada um desempenhado uma função particular. Um modelo é funcional quando estabelece de forma inequívoca a função de cada componente no sistema e a inter-relação entre elas. Este tipo de modelo responde à pergunta “O que o sistema faz?”. A Figura 2.9 exibe a representação de uma função genérica com seus respectivos elementos de entradas e saídas.

Figura 2.9 Representação esquemática de uma Função Global [Negri, 2004]

2.1.3.2 Perspectiva Estrutural

Um modelo estrutural é aquele que representa o conjunto de elementos em um sistema e o conjunto de relações que conectam estes elementos com outros. Este tipo de modelo responde à pergunta "Onde as funções são implementadas?" [Negri, 2004].

Alguns exemplos são: Desenho técnico projetivo (resultantes de projeções de uma peça sobre um ou mais planos) e Diagrama E/R (estabelece relacionamento entre objetos ou conceitos) e Diagrama de Classes (representação de estrutura e relações das classes de dados e serviços).

2.1.3.3 Perspectiva Comportamental

O comportamento de um sistema é observado por meio de uma perspectiva que responde à pergunta "Como ou quando a função do sistema é executada?" [Negri, 2004]. Os modelos comportamentais possuem uma semântica dinâmica, ou seja, descrevem as mudanças de estado e as saídas que ocorrem ao longo do tempo [Paes e Negri, 2002].

Ainda sob a perspectiva comportamental, os modelos podem ser subdivididos em de estado contínuo e de estado discreto. Os modelos a estado contínuo descrevem o sistema através de equações elementares e de suas interconexões, sendo estas equações definidas para todo o tempo (modelos contínuos no tempo) ou definidas em pontos discretos no tempo (modelos discretos no tempo) [Paes e Negri, 2002]. Alguns exemplos são: Equações diferenciais e diagramas de blocos. Por sua vez, os modelos a estado discreto representam os possíveis estados que o sistema pode assumir, sendo a mudança de estado e as saídas produzidas pelo sistema decorrente da combinação lógica de entradas e do estado em que o sistema encontra-se [Paes e Negri, 2002]. Estes tipos de modelo (como, por exemplo, a rede de Petri marcada, o diagrama de transição de estados (diagrama de estados) e o diagrama funcional), mostram explicitamente a sequência com que ocorrem os estados e saídas, sem, normalmente, estabelecer necessariamente uma relação direta com o tempo [Paes e Negri, 2002]. Alguns exemplos são: Equações Booleanas, tabela verdade e diagramas lógicos e Diagrama de Contatos.

2.1.3.4 Perspectiva Funcional/Estrutural

O nível de estruturação obtido com a modelagem em Rede Canal Agência (Rede C/A) é constituída basicamente de três elementos [Matos, 2007]:

- Canal: elemento passivo da rede onde circulam a matéria, energia e informação. É representado pela forma geométrica de um círculo;
- Agência: elemento ativo da rede onde são processadas a matéria, energia e informação. É representado pela forma geométrica de um retângulo;
- Arcos de direção: elemento gráfico indicativo dos fluxos de energia, matéria e informação.

Os símbolos referentes aos elementos básicos e à interconexão entre os elementos estão descritos, respectivamente, nas tabelas 2.1 e 2.2.

Tabela 2.1 Elementos básicos

Elementos Básicos			
Símbolo	Nome Genérico	Perspectiva Funcional	Perspectiva Estrutural
	Unidade Ativa	Atividade (Função)	Agência
	Unidade Passiva	Recurso	Canal

Tabela 2.2 Interconexão de elementos

Interconexão de Elementos	
Símbolo	Tipo de Recurso
	Fluxo de Informação
	Fluxo de Energia
	Fluxo de Matéria
	Fluxo de Energia e Matéria

Sob uma perspectiva funcional, as unidades funcionais passivas correspondem aos recursos que fluem através do sistema, ou seja, a energia, a matéria e a informação ou suas formas de manifestação, tais como eletricidade, peças, ferramentas, sinais, dados, etc. [Negri, 1996]. Sob uma perspectiva estrutural, as unidades funcionais passivas são designadas de canais, indicando os componentes do sistema que dão suporte para que os recursos possam fluir, sem causar modificação no seus estado [Negri, 1996].

A Figura 2.10 mostra a estrutura funcional de um sistema automático genérico, onde é possível observar a troca de informações entre as agências Sistema de Informação e Sistema Energético/Material.

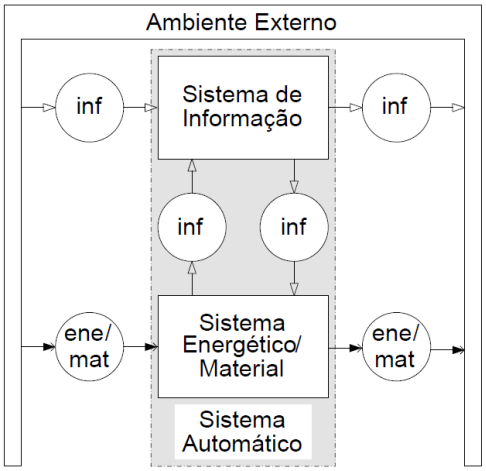


Figura 2.10 Modelo funcional/estrutural de um sistema automático [Souto, 2005]

A Figura 2.11, apresentada a seguir, consiste um detalhamento da Figura 2.10. O Sistema de Medição (SM) atua no sentido "processamento de energia/matéria" para o "processamento de informação", o Sistema de Atuação (SA) no sentido inverso [Matos, 2007].

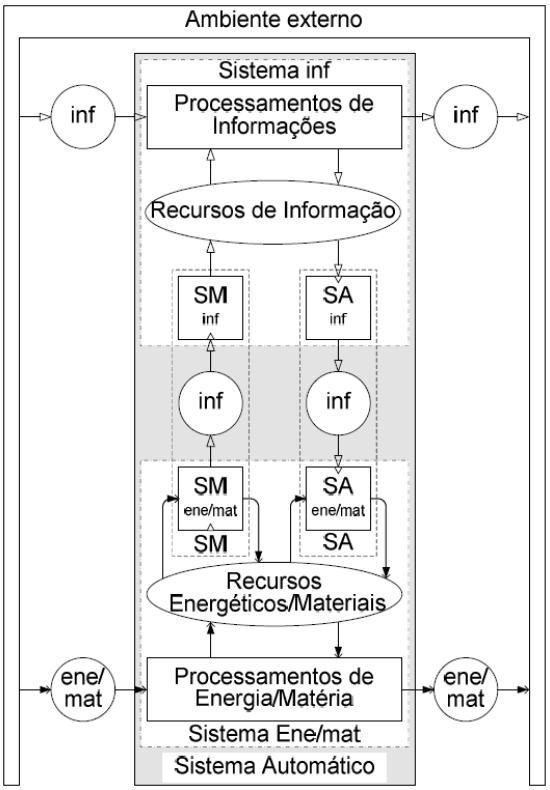


Figura 2.11 Sistema automático - modelo funcional refinado [Souto, 2005]

Em Souto, 2005, são propostas novas agências (ver Figura 2.12), Sistema de Atuação e Medição (SAM), Sistema de Pedido (SP), Sistema de Resposta (SR) e Sistema de Pedido e Resposta (SPR).

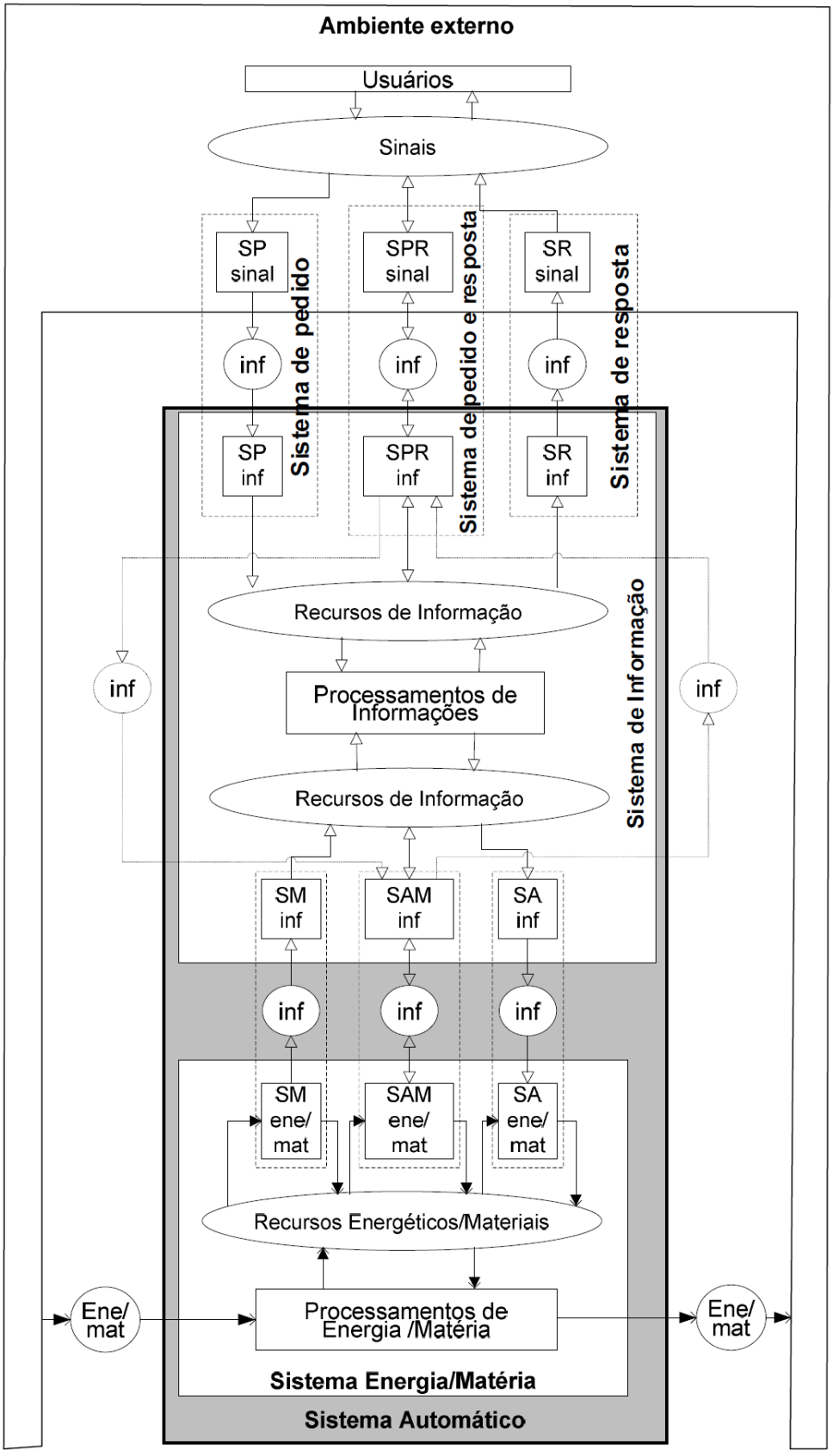


Figura 2.12 Modelo estrutural e funcional de um sistema automático estendido [Souto, 2005]

Observa-se que os sistemas de atuação (SA), medição (SM) e atuação/medição (SAM) são substituídos, respectivamente, por sistema de pedido (SP), resposta (SR) e sistemas mistos de pedido e resposta (SPR) na interface de comunicação entre usuário e sistema de informação. Desta forma, permite-se que o problema seja trabalhado em partes menores baseadas em subfunções, obtendo resultados mais aprimorados e sem perder a visão do sistema como um todo [Souto, 2005].

Assim, neste trabalho, os sistemas APM são analisados sob as perspectivas funcionais, estruturais e comportamentais discretas. A perspectiva comportamental contínua do sistema Aeromovel foi trabalhada em Britto, 2008, e Sarmanho, 2009.

As perspectivas acima citadas, somente permitem representar o sistema, não sendo possível simular ou testar seu comportamento. Para o detalhamento, simulação e validação é necessário o uso de ferramentas que permitam modelar e simular seu funcionamento de forma a verificar se as respostas do sistema em questão estão de acordo com suas especificações.

2.1.3.5 Autômatos Temporizados

Automato finito é um sistema de estados finitos (portanto possui um número finito e predefinido de estados) o qual constitui um modelo computacional do tipo sequencial [Menezes, 2008]. Este formalismo não se aplica a avaliação de sistemas críticos (*hard real-time*) pela ausência da variável tempo. Este formalismo operacional pode ser:

- *determinístico*, onde, dependendo do estado corrente e do símbolo lido, o sistema pode assumir um único estado bem determinado.
- *não-determinístico*, onde, dependendo do estado corrente e do símbolo lido, o sistema pode assumir um conjunto de estados alternativos.

Os autômatos híbridos são aqueles compostos por sistemas contínuos e discretos [Henzinger, 1996]. Pelo sistema em análise envolver elementos analógicos e digitais, como por exemplo os sensores e sistema de comunicação, este formalismo é o mais apropriado para utilização, porém, ainda é necessário que se desenvolvam ferramentas mais eficientes, que suportem o grande número de variáveis que um sistema complexo possui quando consideram-se todos os detalhes de implementação [Oliveira, 2003].

Um autômato temporizado, proposto por Alur e Dill, 1991, é um autômato finito acrescido de restrições temporais, sendo assim, um sistema de estados finitos que possui um modelo computacional do tipo sequencial e não determinístico, pois dependendo do estado

corrente e do símbolo lido, o sistema pode assumir um conjunto de estados alternativos acrescidos de restrições temporais.

Bons resultados podem ser obtidos, utilizando simulação e verificação formal de forma complementar [Machado et al., 2011]. Enquanto que a simulação permite obter resultados de uma forma mais célere, porque só são analisadas algumas possibilidades de evolução do comportamento do sistema, a verificação formal permite a obtenção de resultados mais completos, em virtude de permitir a análise de todas as possibilidades de evolução do comportamento de um sistema. Entre os formalismos a utilizar, deve ser escolhido um formalismo que permita modelar o tempo.

Relativamente aos simuladores e *model-checkers* existentes no mercado optou-se pela utilização do UPPAAL, por dois motivos principais: porque permite a utilização direta, como *input*, de um formalismo que modela o tempo (Autômatos Finitos Temporizados) e permite simular e verificar formalmente, no mesmo ambiente, o mesmo modelo, sem necessidade de translação entre formalismos que poderia levar a erros na análise, se dois ambientes diferentes, para simulação e verificação formal fossem utilizados.

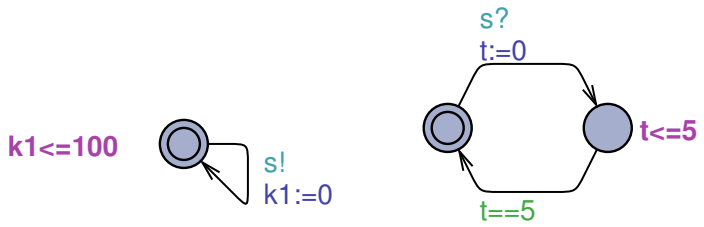
UPPAAL é uma ferramenta madura (primeira versão datada de 1995) para verificação de sistemas de tempo-real desenvolvida pelas Universidades de Uppsala (Suécia) e Aalborg (Dinamarca). Esta ferramenta tem como objetivo realizar a verificação formal de sistemas modelados em redes de autômatos temporais através de linguagem baseada em TCTL (*Timed Computation Tree Logic*) [Behrmann et al., 2006].

A lógica TCTL [Alur e Dill, 1991] é uma extensão à lógica CTL (*Computation Tree Logic*), proporcionando a capacidade de expressar propriedades de tempo real como, por exemplo: "*A porta deve ser aberta em até 5* s *após a parada completa do veículo na estação*".

Tal formalismo é aplicado em uma variedade de aplicações de controle distribuído, críticas e de tempo-real, tais como em protocolos de áudio altamente dependentes de tempo real [Havelund et al., 1997], interrupção entre comunicação de sistemas de áudio e vídeo [Havelund et al., 1999], transmissão de dados com perda de pacotes D'Argenio et al., 1997, algoritmos de sincronização labial [Bowman et al., 1998], análise de protocolos para sistemas industriais [David e Yi, 2000], controladores distribuídos em plantas industriais [Hune et al., 2000], processadores [Iversen et al., 2000], controladores de motores para veículos [Lindahl et al., 2001] e protocolo TDMA (*Time division multiple access*) [Lnn e Pettersson, 1997].

As figuras 2.13(a) e 2.13(b) exemplificam a representação de uma rede de autômatos temporizados através do uso da ferramenta UPPAAL, onde os círculos representam estados e as

setas a transição entre eles. O estado inicial é representado por um circulo interno.



(a) Modelo Emissor (b) Modelo Receptor

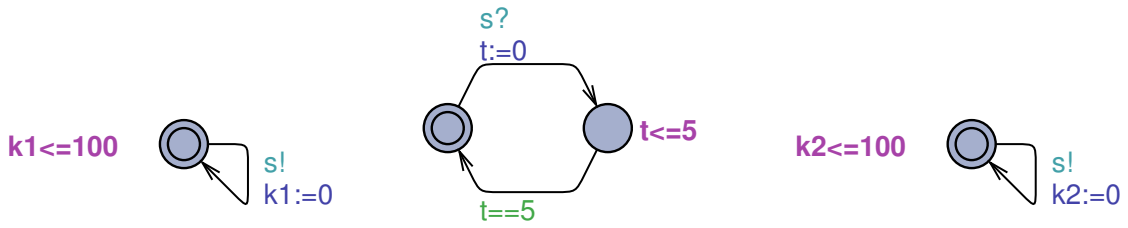
Figura 2.13 Rede de autômatos temporais

Na Figura 2.13(a) é modelado um emissor que envia sinal em um período não maior do que 100 unidades de tempo ($k_1 <= 100$), onde k_1 representa um relógio e um sinal de comando descrito pelo canal s, onde um sinal de sincronização acionado é representado através do uso de um ponto de exclamação ($s!$) e recebe o sinal representado por um ponto de interrogação ($s?$). Ao enviar o sinal, o modelo representado na Figura 2.13(a) reinicializa o relógio k_1. Observa-se que ocorre restrição somente quanto ao período máximo de tempo para envio. Assim, o sinal é enviado entre 0 e 100 unidades de tempo.

Na Figura 2.13(b) a mudança de estado ocorre somente ao receber o sinal de sincronização através do comando $s?$. Neste instante, o temporizador t é reinicializado. A mudança para o estado inicial ocorre somente no instante $t = 5$, visto que não é permitido que este estado permaneça ativo por mais de 5 unidades de tempo ($t <= 5$) e que há uma restrição para transição (o relógio deve ser igual a 5), ou seja, para cada envio de um novo dado do sensor, tem-se 5 unidades de tempo de processamento.

Na Figura 2.14, uma nova simulação é apresentada. Neste caso, cada um dos emissores representa um sensor críitfluxo co para o movimento de um veículo, onde, o veículo leva 5 ms para processar um sinal recebido e quando o intervalo de recebimento é superior a 100 ms, é considerado que o sistema está em estado de falha. Na simulação desta rede de autômatos temporais, o sistema pode não apresentar falhas, visto que, por definição, a simulação não implica em testar, necessariamente, todas as possibilidades de eventos. Desta forma, a simulação pode apenas indicar que o sistema funciona de acordo com o esperado. Porém, quando se realiza a verificação formal, ou seja, a validação matemática das possibilidades de combinação de eventos assíncronos, visando a testar possíveis falhas do sistema, podem ser identificados erros não observados durante a simulação. Um exemplo típico pode ser exemplificado através da operação do relógio apresentado na Figura 2.14(a). Assim, quando ele se encontra acima de 95 ms e o modelo da Figura 2.14(c) envia um dado para o receptor (Figura 2.14(b)), como o receptor estará ocupado pelos próximos 5 ms, resulta na incapacidade

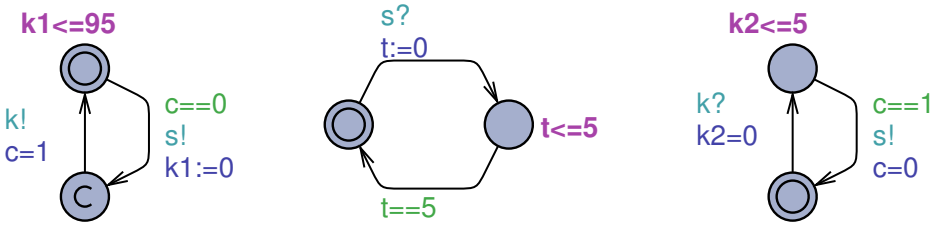
de envio do sinal pelo modelo da Figura 2.14(a), dentro do intervalo de 100 ms. Assim, nesta condição, o sistema está em estado de falha que não produz erro durante a simulação mas que é identificada através da verificação formal por autômatos temporizados.



(a) Modelo Emissor 1 (b) Modelo Receptor (c) Modelo Emissor 2

Figura 2.14 Rede de autômatos temporais ampliada

Uma vez encontrada a possibilidade de um evento não desejado, volta-se para os modelos para avaliar a possibilidade de correção. Uma solução para o problema apresentado acima é apresentada na Figura 2.15, onde é inserida uma sincronização para o envio dos sinais pelos sensores, onde k e s são canais de sincronismo, c é uma variável booleana e t, k_1 e k_2 são relógios. Desta forma, o modelo apresentado na da Figura 2.15(a) deve enviar um sinal para o veículo em um intervalo de tempo inferior a 95 ms e, somente após este envio, o modelo da Figura 2.15(c) enviará seu sinal para o veículo, atendendo assim a simulação e a verificação formal.



(a) Modelo Emissor 1 (b) Modelo Receptor (c) Modelo Emissor 2

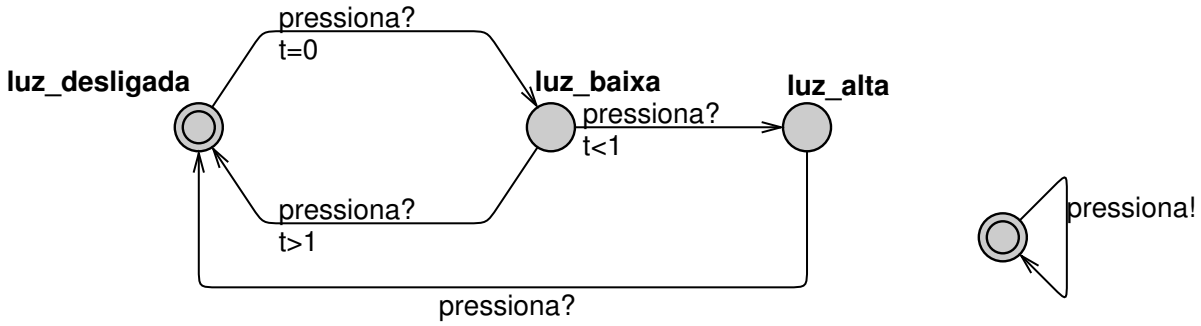
Figura 2.15 Rede de autômatos temporais corrigida

Nas figuras 2.16(a) e 2.16(b) é exemplificada uma rede de autômatos temporizados de interrupção de corrente, onde *pressiona* é um canal de comunicação e t um relógio. A Figura 2.16(a) representa a lâmpada com dois diferentes níveis de iluminação (baixa e alta). Este modelo segue as seguintes regras de funcionamento:

- O estado inicial é de lâmpada desligada.
- Para ligar a lâmpada na intensidade baixa deve ser pressionado o interruptor apenas uma única vez. O acionamento do interruptor é identificado pelo sinal *pressiona*?.
- Ao pressionar o interruptor (saindo do estado de desligado) é iniciado um temporizador ($t = 0$).

- Para ligar a lâmpada na intensidade alta, deve ser pressionado o interruptor duas vezes seguidas em um intervalo menor do que 1 s ($t <= 1$).
- Uma vez em funcionamento, a lâmpada é desligada somente pelo acionamento do interruptor.
- Para desligar a lâmpada, deve ser pressionado o interruptor uma única vez.

A Figura 2.16(b) representa o acionamento do interruptor. Observa-se que o acionamento, representado pelo sinal *pressiona*!, ocorre sem restrições de tempo, ou seja, de forma aleatória.

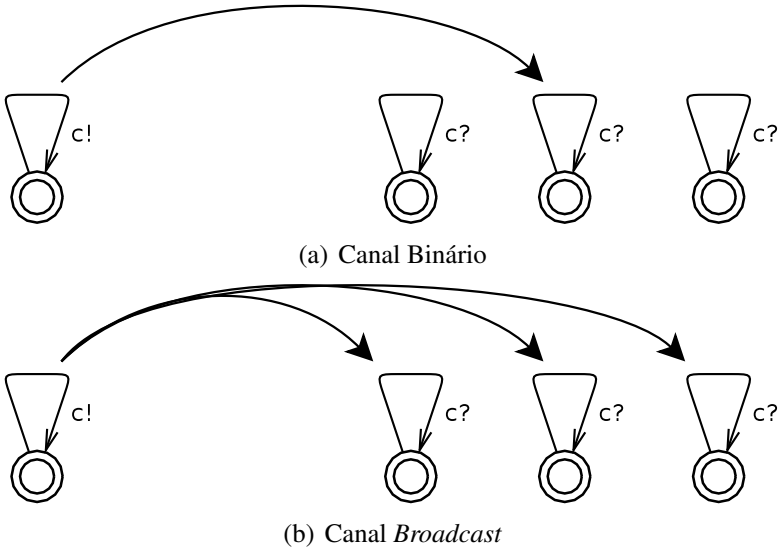


(a) Lâmpada (b) Interruptor

Figura 2.16 Modelo exemplificativo de uma rede de autômatos temporais de um interruptor de corrente

O UPPAAL possui extensões em sua linguagem de modelamento para uso com autômatos temporizados (ver detalhes no Anexo A.1). Dentre as ampliações da linguagem, as seguintes características relativas à sincronização de uma rede de autômatos temporizados são importantes para a compreensão dos modelos [Behrmann et al., 2004]:

- **Canais de sincronização binários** são declarados como "`chan c`". Uma transição rotulada com $c!$ sincroniza com outra rotulada com $c?$. A sincronização entre canais não é determinística se mais de uma combinação é possível, ou seja, se para um $c!$ existem mais de um $c?$. A Figura 2.17(a) ilustra o funcionamento de canais binários onde apenas um receptor por vez recebe o sinal de sincronismo. Neste caso, não há ordenação entre os possíveis receptores, agindo em momento e sequencia indeterminada. Se nenhum dos receptores estiver aguardando o sinal de sincronização considera-se falha do sistema.
- **Canais de sincronização *broadcast*** são declarados na forma de "`broadcast chan c`". Em sincronização tipo *broadcast*, um emissor $c!$ pode sincronizar com mais de um receptor $c?$. Se não houver receptores, o emissor mantém a execução, isto é, envios *broadcast* são não bloqueantes, de forma diversa da sincronização binária. Diferentemente dos canais binários, a Figura 2.17(b) indica o funcionamento de canais



(a) Canal Binário

(b) Canal *Broadcast*

Figura 2.17 Diferença entre canais binários e *broadcast*

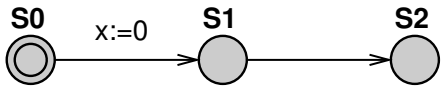
broadcast, onde todas as tarefas que aguardam o sinal de sincronização *c* recebem ao mesmo tempo. Em canais *broadcast*, não é necessário haver um receptor da mensagem.

- **Canais de sincronização urgentes** são declarados com a adição do prefixo *urgent*. Atrasos não devem ocorrer se uma transição de sincronização de um canal urgente está habilitado. Transições usando canais urgentes para sincronização não devem utilizar restrições de tempo.

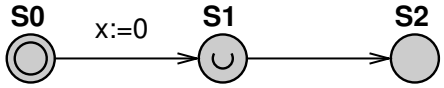
- **Estados urgentes** são semanticamente semelhantes à adição de um relógio extra x, que reinicia em todas as entradas e tem uma invariante $x <= 0$ no estado. Assim, os relógios permanecem inalterados enquanto o sistema está em um estado urgente. É identificado por uma letra "U", conforme pode ser observado através da Figura 2.18(b).

- **Estados *committed*** são mais restritivos do que os estados urgentes. O estado geral do sistema é *committed* se algum estado *committed* estiver ativado. Quando o sistema está no estado *committed* não pode haver mudança de tempo e a próxima transição deve envolver a saída deste estado, ou, ao menos, de um dos estados que estão na condição *committed*. Nos modelos, estes estados são utilizados para representar mudanças que não envolvem tempo, ou seja, que não representam mudanças físicas no sistema. O uso de estados *committed* representa ganhos em termos de processamento durante o trabalho de verificação formal do sistema. Estes estados são identificados por uma letra "C", conforme pode ser observado na Figura 2.18(c).

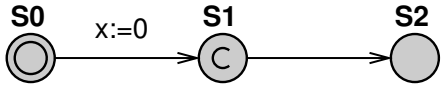
Na Figura 2.18 estão ilustradas as diferenças entre os estados normais, urgentes e *commited*, onde x é um relógio. Podem ser observados os seguintes comportamentos destes modelos funcionando em conjunto:



(a) Estado Normal



(b) Estado Urgente



(c) Estado *Commited*

Figura 2.18 Diferença entre estados normais, urgentes e *commited*

- Sempre que atingir o estado *commited* a próxima transição será a saída deste estado.
- É possível que o modelo da Figura 2.18(a) esteja no estado S_1 e que o relógio x seja maior do que zero.
- Sempre que o modelo da Figura 2.18(b) estiver em S_1, tem-se que o relógio x é igual a zero, pois o tempo não pode passar enquanto o sistema estiver em um estado urgente.

Nos modelos de autômatos temporizados no UPPAAL é possível utilizar as seguintes variáveis de controle, tanto nas transições quanto nos estados:

- **Proteção.** Uma proteção é uma expressão particular que satisfaz as seguintes condições: não possui efeito externo; resulta em uma condição booleana; somente relógios, variáveis inteiras e constantes podem ser referenciadas. Representa, basicamente, uma comparação entre variáveis ou relógios da qual depende para poder realizar a transição de um estado para outro. Um exemplo é a expressão $t == 5$ da Figura 2.13(b) indicando que a transição deve ocorrer somente quando o relógio t for igual a 5.
- **Sincronização.** O rótulo de sincronização utiliza a forma *expressão!*, *expressão?* ou vazio. A expressão deve ser sem efeito externo e somente pode fazer referência a inteiros, constantes e canais de sincronização. Um exemplo são os sinais de comunicação $s!$ e $s?$ entre os modelos das figuras 2.13(a) e 2.13(b).
- **Atribuição.** Uma atribuição contém uma lista de expressões separadas por ponto-e-vírgula que realizam mudanças em variáveis externas; estas expressões devem referir-se somente a relógios, variáveis inteiras e constantes. Um exemplo é a expressão $t := 0$ da Figura 2.13(b), indicando a atribuição do valor zero ao relógio t.
- **Invariante.** Um invariante é uma expressão que segue as seguintes condições: não

possui efeitos externos; somente relógios, variáveis inteiras e constantes podem ser referenciadas. Pode ser utilizado em um estado para inserir uma condição de tempo limite da forma $x < t$ ou $x <= t$, onde x é um relógio. Se não for possível sair do estado dentro do tempo determinado, o sistema assume uma condição de falha. Um exemplo é a expressão $t <= 5$ da Figura 2.13(b), indicando que o tempo máximo de permanência no estado em questão é de 5 (podendo ser inferior a isto).

O principal objetivo de um verificador formal é testar o modelo para um conjunto de especificações. Assim como os modelos, a especificação do requisito deve ser expressa em uma linguagem formal e legível para o sistema. Existem na literatura, muitas lógicas para linguagem sendo que o UPPAAL utiliza uma versão simplificada da TCTL (*Timed Computation Tree Logic*).

Como poderão ser visto a seguir, a verificação formal é dividida em testes de acessibilidade, segurança e invariabilidade:

- **Acessibilidade.** Dada uma formulação (φ), serve para verificar a possibilidade de uma especificação ser satisfeita. Verifica também a possibilidade de φ ser eventualmente satisfeito ao longo da trajetória de estados. Pode ser utilizado para verificar, por exemplo, se uma válvula pode passar da condição de aberta para fechada ou se uma mensagem pode sair do emissor e chegar ao receptor. Esta propriedade não garante se o protocolo ou a válvula em teste funciona corretamente, mas verifica se o seu funcionamento básico é possível de ocorrer. No UPPAAL, a sintaxe para esta propriedade é dada por "`E<>`φ", que significa se existe um caminho onde φ pode ocorrer.

- **Segurança.** A propriedade de segurança pode ser expressa da seguinte forma: "verificar se um evento não desejado pode ocorrer". Por exemplo, em uma planta nuclear, verificar se a temperatura vai estar sempre abaixo de um determinado valor. Considerando que φ é a formulação do teste, no UPPAAL, a verificação de segurança é expressa na forma "`A[]` φ", testando, assim, se para todos os possíveis caminhos, φ sempre vai ocorrer.

- **Vivacidade.** Vivacidade significa que algo eventualmente pode ocorrer, ou seja, por exemplo, ao pressionar o botão do controle remoto, eventualmente, a televisão vai ligar, ou do conjunto de mensagens enviadas pelo emissor, eventualmente, alguma vai chegar ao receptor. A vivacidade é expressa pela fórmula "`A<>`φ", significando que para todos os caminhos φ eventualmente pode ocorrer. Outra forma é $\varphi \rightsquigarrow \psi$, ou seja, sempre que φ é satisfeito, então, eventualmente, ψ vai ser satisfeito. No UPPAAL, a vivacidade é expressa na forma "$\varphi --> \psi$".

No UPPAAL, a condição de intertravamento é expressa utilizando uma fórmula específica. Essa fórmula consiste no uso do comando "`deadlock`", que verifica todos os possíveis estados de falha para essa condição. Por exemplo, a expressão "`A[] not deadlock`" testa se, para todos os caminhos possíveis, inexiste intertravamento (observa-se que a expressão "`not`" designa negação).

Adicionalmente, pode-se usar a expressão "`imply`" para dedução. Na Figura 2.18(b), onde foram apresentados os estados urgentes, foi descrito que sempre que o modelo estiver em S_1 implica que o relógio x é igual a zero, ou em linguagem CTL, "`A[] S1 imply x==0`".

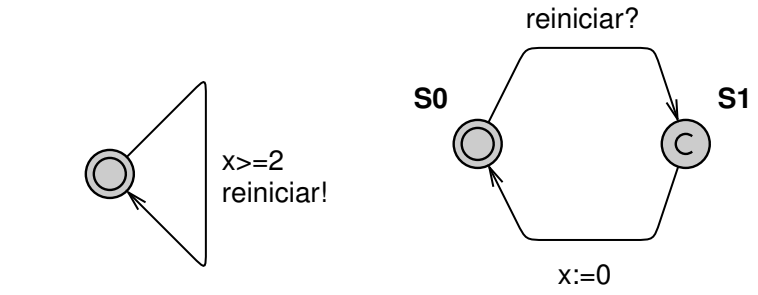
A ferramenta UPPAAL utiliza DBM (*Difference Bounded Matrix*) como estrutura de dados para armazenar os relógios e suas evoluções com o tempo [Bengtsson e Yi, 2004].

Sendo o tempo variável primordial de observância nos modelos, a seguir serão exemplificados 3 diferentes casos de restrições temporais em autômatos. Nas figuras 2.19(a) e 2.19(b) estão expostos, respectivamente, os modelos de emissão e recepção, onde x é um relógio e *reiniciar* é um canal de comunicação binário. Observa-se que, no modelo da Figura 2.19(a), sempre que $x >= 2$ (proteção), é enviado o sinal *reiniciar*! para o modelo da Figura 2.19(b) que, ao receber o sinal *reiniciar*?, sai do estado S_1 para o estado *commited* S_2, reiniciando o relógio x ao voltar para o estado S_1. Observa-se pelo gráfico da Figura 2.19(c) que, sempre que o modelo estiver em S_1, implica que $x >= 2$ ou "`A[]` $receptor.S_1$ `imply` $x >= 2$". Porém, não há restrições quanto ao tempo limite para esta transição, de modo que, eventualmente quando o modelo estiver em S_1, o relógio x pode ser maior do que 1000, ou, "`E<>` $receptor.S_1$ `and` $x > 1000$".

O modelo do emissor, conforme mostrado na Figura 2.20(a), foi alterado através da inserção de uma invariante $x <= 3$, designando que o modelo emissor tem tempo máximo para sair do estado inicial. Desta forma, a Figura 2.20(c) mostra os possíveis resultados do relógio, podendo-se concluir que, quando o receptor estiver em S_1 o relógio sempre estará entre os valores 2 e 3.

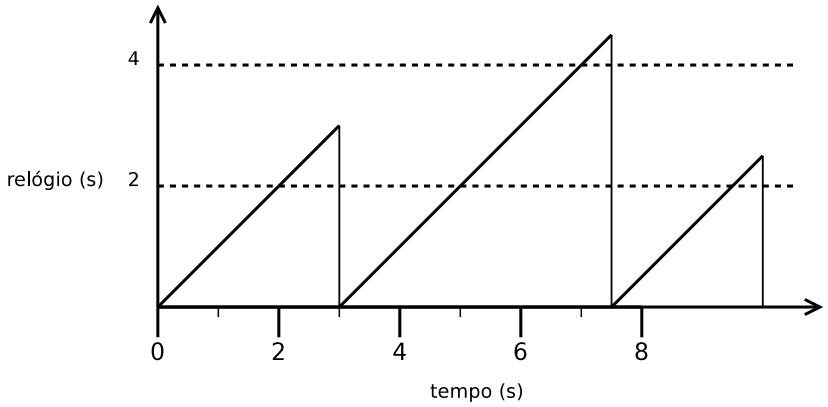
Se, ao invés de inserir uma invariante no modelo do emissor, fossem utilizados somente as proteções, como pode ser observado através da Figura 2.21(a), o resultado da simulação, poderia, por exemplo, apresentar comportamento conforme o apresentado na Figura 2.21(c), onde, se dentro do intervalo entre 2 e 3 unidades de tempo do relógio não houver transição de estados, o mesmo não ocorrerá mais, pois estará fora dos requisitos de proteção de transição.

O UPPAAL é uma ferramenta que permite a simulação e verificação formal de autômatos temporais. Assim, para sua aplicação, o desenvolvimento dos modelos apresentados neste trabalho segue o método apresentado na Figura 2.22 e detalhado a seguir:



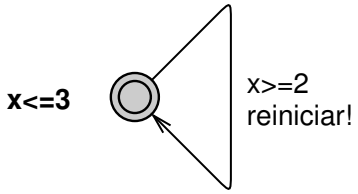
(a) Autômato Emissor

(b) Autômato Receptor

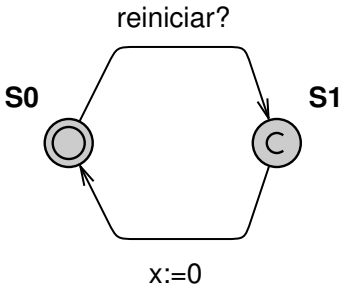


(c) Resultado de Simulação

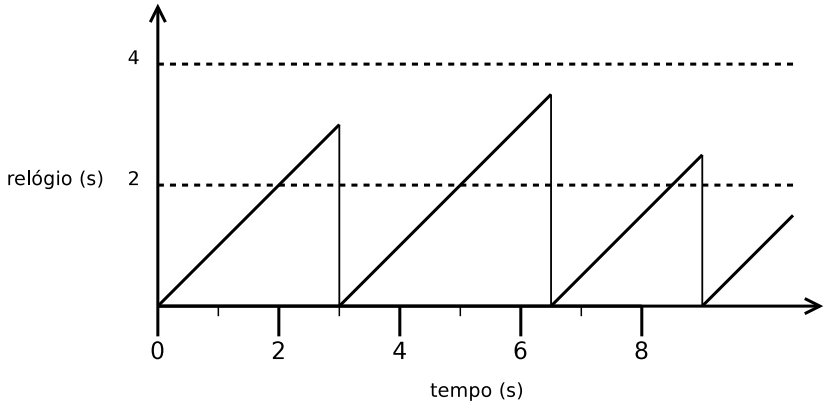
Figura 2.19 Autômatos com restrição de tempo mínimo para transição



(a) Autômato Emissor

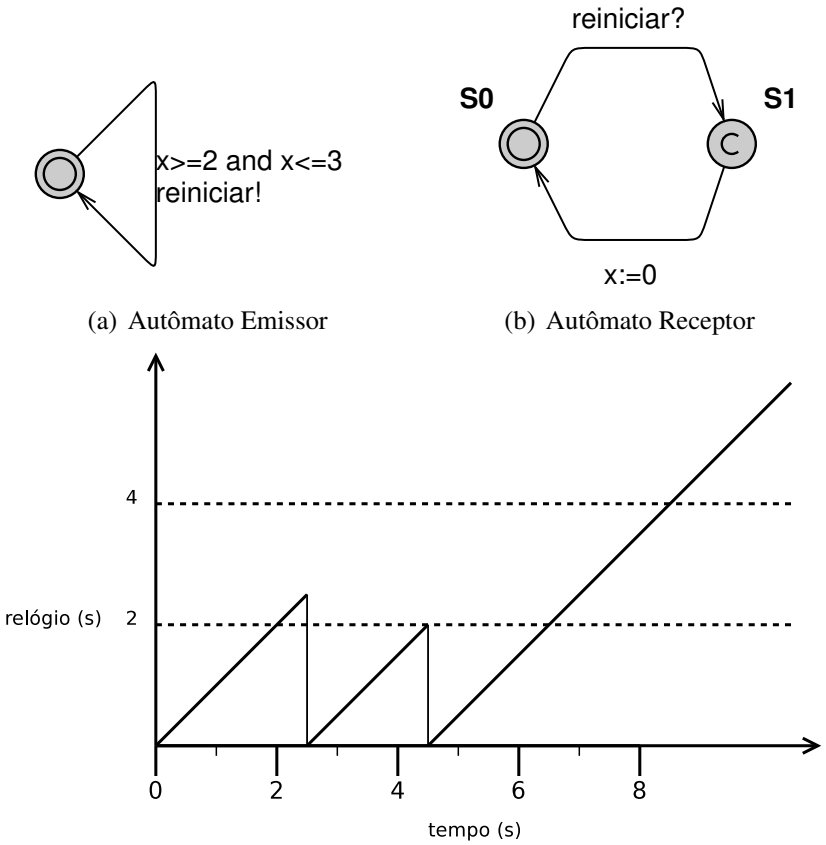


(b) Autômato Receptor



(c) Resultado de Simulação

Figura 2.20 Autômatos com intervalo de tempo fixo para transição



(a) Autômato Emissor (b) Autômato Receptor

(c) Resultado de Simulação

Figura 2.21 Autômatos com intervalo de tempo desejável para transição

1. Especificação. Levantamento dos requisitos do equipamento ou controlador a ser modelado, tais como, por exemplo, tempo de resposta, entradas e saídas de dados.

2. Desenvolvimento. Modelagem em autômatos temporizados dos requisitos do equipamento ou controlador.

3. Simulação. A etapa de simulação tem como objetivo verificar se o sistema possui comportamento esperado conforme especificado. No caso de não possuir comportamento de forma a representar o equipamento ou controlador, deve-se voltar à etapa de desenvolvimento. A ferramenta UPPAAL permite, durante simulação, visualizar a mudança de estado nos modelos e a produção de arquivo de eventos que possibilita a produção gráfica dos estados, facilitando a verificação visual do comportamento do modelo.

4. Verificação. A etapa de verificação tem por objetivo verificar formalmente se os requisitos do modelo são completamente atendidos. No caso de não atender a alguma dos requisitos, deve-se voltar à etapa de desenvolvimento.

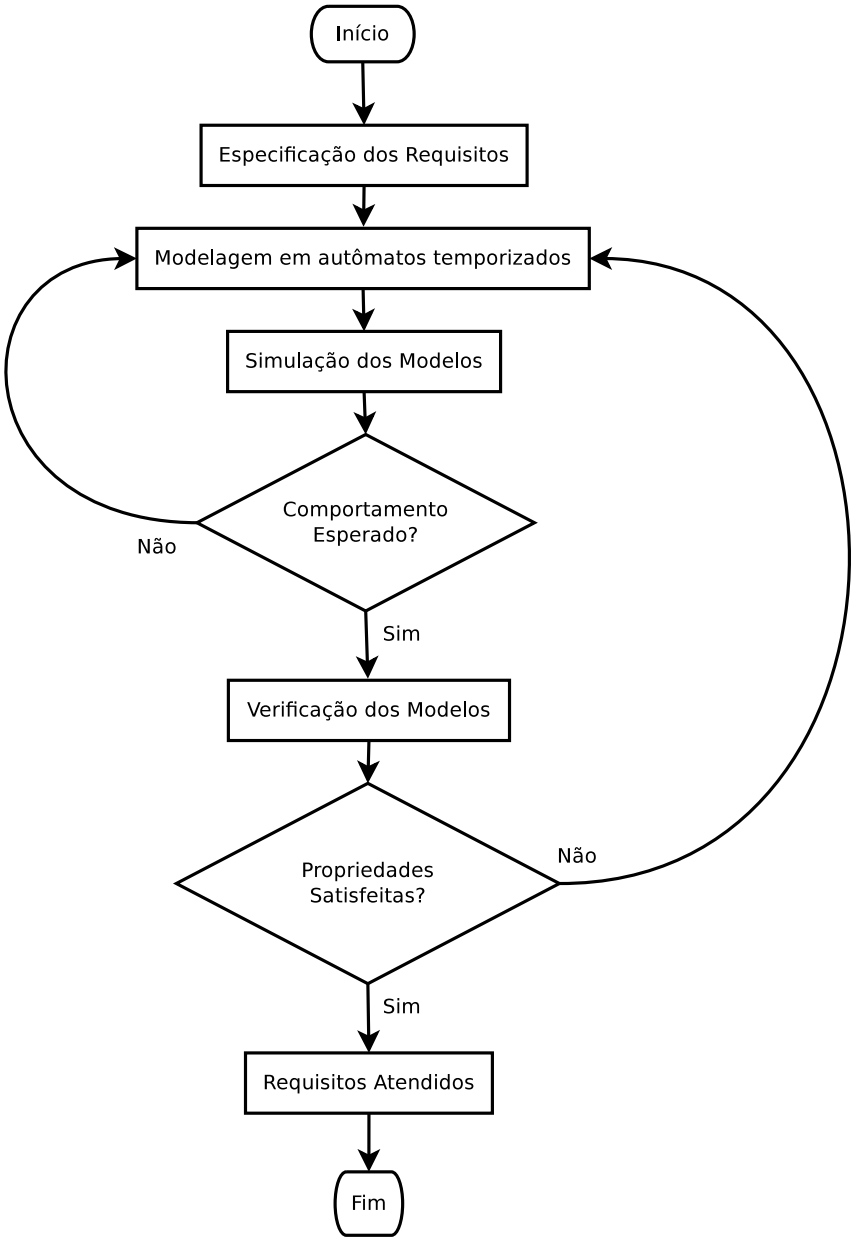


Figura 2.22 Método de desenvolvimento

Testes de Conformidade de Modelos Implementados em CLPs

Testes de conformidade tem por objetivo verificar se uma aplicação satisfaz fielmente aos requisitos de um padrão ou especificação. Existem muitos tipos de testes, incluindo testes de desempenho, robustez, comportamento, funções e interoperabilidade [Gray et al., 2000]. Trabalhos visando à geração automática de testes de conformidade vem sendo desenvolvidos tais como em Silva, 2008, Oliveira, 2009, e Sampaio, 2011, por exemplo.

Para verificar se o modelo, por exemplo, do controlador do GMP, permanece com comportamento conforme ao comportamento modelado na rede de autômatos temporizados após sua implementação em CLP, propõe-se a utilização da ferramenta UPPAAL TRON [Larsen

et al., 2009]. A primeira versão da UPPAAL TRON é datada de 2004 e baseada no trabalho de Mikucionis e Sasnauskaite, 2003. Esta ferramenta foi escolhida por sua maturidade e por ser compatível com os modelos implementados no UPPAAL [Larsen et al., 2004a; Hessel et al., 2008a; Mikucionis et al., 2004].

2.1.4 Sistemas APM (*Automated People Mover*)

A seguir, visando a identificação da existência de um padrão quanto a requisitos de comunicação, funcionalidades, estrutura de dados, convenção de nomes e testes de conformidade, estão descritas as principais normas técnicas relativas aos sistemas APM.

2.1.4.1 Sistema de Sinalização e Segurança

Nesta seção são apresentadas algumas das metodologias e tecnologias utilizadas na sinalização visando ao controle de tráfego.

Blocos

No início da organização do transporte ferroviário, os trilhos eram divididos em segmentos de linha e cada segmento permitia a passagem de uma composição por vez, sendo a disponibilidade de tal recurso sinalizada por semáforos. Para evitar a existência de sinalização conflitante, foi desenvolvido um método baseado em intertravamentos (*interlocking*), que permite automatizar com segurança a troca de estados dos semáforos [TDC, 1998].

Este método, também conhecido como sinalização de bloqueio (*Block Signal*) ou sinalização automática de bloqueio (*Automatic Block Signal* - ABS) foi melhorado com o advento do bloqueio absoluto de permissão (*Absolute Permissive Block* - APB), o qual permitia o controle de blocos em ambas as direções [TDC, 1998].

Distance-To-Go

Com o uso de um sistema computacional, a localização da composição é reconhecida e, assim, é possível calcular a força de frenagem necessária para a parada antes do próximo obstáculo, geralmente no final do segmento de linha que está ocupando [Proenca, 2003].

Neste tipo de configuração, sistemas computacionais centralizados são geralmente utilizados (*Centralized Traffic Control* (CTC) *Systems*)

Blocos Móveis

Unindo os métodos de Blocos e *Distance-To-Go*, foi desenvolvido o método de blocos móveis, onde, ao invés de segmentar os trilhos, fixa-se os blocos com referência nas composições, assim criando uma região da linha que é momentaneamente exclusiva de acesso a composição [Proenca, 2003]. Esta região é móvel, acompanhando o deslocamento da composição.

Assim, para o uso desta tecnologia, faz-se necessária a comunicação entre os veículos em trânsito. A Figura 2.23 ilustra o funcionamento da metodologia de blocos móveis.

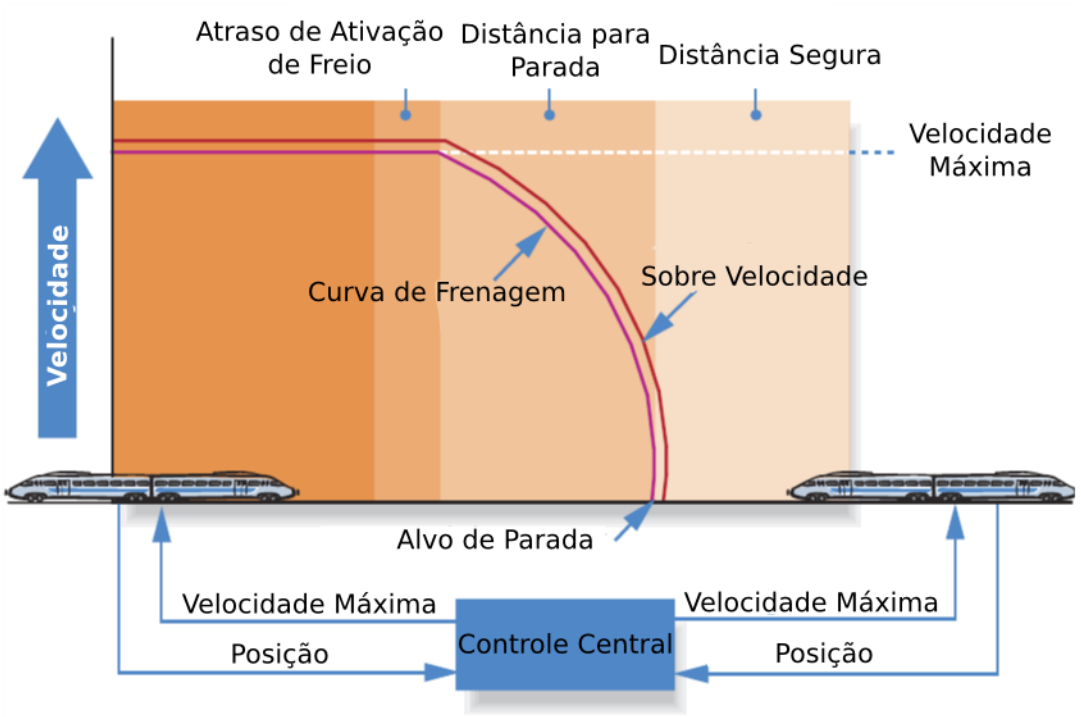


Figura 2.23 Blocos móveis [Alcatel, 2003]

Para a ativação satisfatória do sistema de frenagem do trem, é necessário que o controle central possua, constantemente atualizadas as informações da velocidade e da posição atual de cada veículo em trânsito na via e o tempo necessário para a ativação do freio de forma a realizar a frenagem de acordo com a curva de frenagem (conforme a Figura 2.23), evitando assim o choque entre os veículos.

Sistemas de comunicação avançados são utilizados (incluindo comunicação a rádio frequência). Assim, para viabilizar esta metodologia, foi desenvolvido o sistema controle de composições baseado em comunicação CBTC (*Communications-Based Train Control*) que é definida segundo norma específica, descrita a seguir, na Seção 2.1.4.2.

2.1.4.2 Normas Técnicas

Visto que entre os objetivos deste trabalho está o desenvolvimento dos sistemas de Proteção, Controle e Operação de Sistemas APM sob o ponto de vista da norma IEC 61850, faz-se necessário realizar um estudo sobre as normas existentes para sistemas de automação de composições, visando a levantar subsídios para apoiar e justificar a presente proposta.

No decorrer desta seção são referenciadas normas técnicas de acordo com a sua importância e impacto no sistema de automação, controle, supervisão e proteção de Sistemas APM.

Em *IEEE standard for the functioning of and interfaces among propulsion, friction brake, and train-borne master control on rail rapid transit vehicles* [IEEE, 1999b] são descritos o funcionamento das interfaces entre propulsão, freio a fricção e o controle central de bordo. As interfaces são basicamente divididas em 3 diferentes tipos:

- sinais "liga-desliga". Transmitidos por sinal elétrico, tendo como característica a disponibilização de um fluxo unidirecional de informação, que é considerado insuficiente para anúncio de falha e registro de dados, sendo portanto, aplicado para indicação luminosa e aplicações similares;
- sinais digitais e analógicos proporcionais. Tem também como característica o fluxo unidirecional, sendo usado apenas para o anúncio de falha (limitado a indicação luminosa ou similar), sendo possível o registro das falhas;
- comunicação serial. Tem como característica o fluxo bi-direcional, sendo possível o registro de falhas e a integração entre os níveis de automação do veículo.

Nesta norma são determinadas as interfaces para os sistemas de freio de emergência, direção, propulsão, aceleração e velocidade, frenagem combinada de dois ou mais métodos (como, por exemplo, regeneração e atrito), carregamento do veículo (alguns sistemas utilizam este sinal para cálculo preciso da distância de parada), freio de emergência, detecção de não-movimento e estado das portas em relação ao controle central e entre elas. Observa-se que, para todas as interfaces, a comunicação serial é válida.

Em *IEEE Standard for User Interface Requirements in Communications-Based Train Control (CBTC) Systems* [IEEE, 2003] são descritos os requisitos das interfaces com os usuários, tanto dos equipamentos localizados a bordo do trem, quanto fora do trem. São abordados requisitos de ergonomia, interface visual (*display*) e interface auditiva.

São obrigatórias, na interface a bordo, a visualização dos seguintes dados:

- Modo de operação do trem;
- Condição de operabilidade do CBTC;
- Velocidade atual do trem;
- Velocidade atual máxima autorizada;
- Condição de alarme de sobrevelocidade.

É obrigatória, na interface a bordo, as seguintes possibilidades de entrada de dados pelos usuários:

- Seleção de modo de operação;
- Aviso da condição de alarme de sobrevelocidade.

É obrigatória, na interface a bordo, a seguinte saída audível:

- Condição de sobrevelocidade.

São obrigatórias, na interface fora do trem, a visualização dos seguintes dados:

- Informações sobre o plano de trajetória, incluindo intertravamento, localização das estações, localização dos cruzamentos de trilhos e localização dos limites de território de atuação do CBTC;
- Estados do trem, como atributos, modo de operação e localização para todas as composições equipadas com o sistema CBTC e que operam dentro do envelope de controle;
- Informações de limite de autoridade para cada trem equipado com sistema CBTC e sobre as rotas autorizadas;
- Informações sobre restrições na operação do trem (como trilhos bloqueados) e de limite de velocidade.

São obrigatórias, na interface localizada fora do trem, as seguintes entradas de dados pelos usuários:

- Entradas para requisitar e cancelar rotas, incluindo a autorização para controlar e movimentar o trem;
- Entradas para inserir e remover áreas bloqueadas e, temporariamente, restrições de velocidade.

Além dos itens descritos, são definidos requisitos de cores, *layout* de janelas, controle de acesso e informações necessárias para a manutenção do sistema.

A norma *IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements* [IEEE, 2004] estabelece os requisitos funcionais e de desempenho necessários para sistemas que utilizam CBTC. Esta norma apresenta valores típicos, tais como, por exemplo, a resolução de velocidade, o intervalo de comunicação e o tempo de reação de equipamentos utilizados no CBTC. A norma *IEEE Recommended Practice for Communications-Based Train Control (CBTC) System Design and Functional Allocations* [IEEE, 2008] estabelece as recomendações para sistemas CBTC determinadas em IEEE, 2004, através do detalhamento dos sistemas ATP, ATO e ATS.

A norma *IEEE Standard for Communications Protocol Aboard Trains* [IEEE, 1999a] define o protocolo de comunicação interna ao veículo e entre veículos. Este padrão define duas soluções de acordo com a aplicação: o protocolo 1473-L (LonWorks) e 1473-T (TCN). O tipo 1473-L é baseado na EIA 709.1-1998 e EAI 709.3-1998. Pode ser configurado para suportar barramentos de sensores (*local sensor bus* - LSB) ou de aplicações (*local vehicle bus applications* - LVB). O LSB conecta os sensores e o LVB conecta dispositivos embarcados ao sistema de operação do veículo.

O tipo 1473-T é baseado na IEC 61375-1-1999 e é direcionado para aplicações que necessitam de determinismo de tempo, sendo dividido nos protocolos WTB (*wire train bus*) e MVB (*multifunction vehicle bus*). O WTB interconecta as unidades de operação das composições e o MVB interconecta os dispositivos embarcados ao sistema de operação do veículo. Um dos principais benefícios deste padrão é a possibilidade de utilizar equipamentos de diferentes fabricantes [Moreno et al., 2007]. O sistema MVB opera a 1,5 Mbit/s sobre a comunicação serial RS-485 para distâncias curtas (até 200 m) e fibra ótica para distâncias de até 2000 m [Schafers e Hans, 2000].

Em Sullivan, 2001, são observados que nos tipos 1473-T e 1473-L falta suporte para novas demandas de transmissão de vídeo e a falta de interfaces IP, impedindo a comunicação via *Ethernet*, por exemplo, além da falta de integração com protocolos utilizados para sistemas avançados de controle de composições (*Advanced Train Control Systems* - ATCS). Em Sullivan, 2002, Sullivan, 2003, e Laplante e Woolsey, 2003, é observado que empresas tem investido na inserção do protocolo 1473-L através de comunicação IP (*Internet Protocol*).

Em IEEE, 2011, atualização da norma IEEE 1473, é inserido o protocolo 1473-E destinado para redes *Ethernet*. Desta forma, torna-se possível a integração com redes de alta velocidade para transferência de dados, tais como comunicação de áudio e vídeo, sugerindo o

uso de NTP (ver IETF RFC 1305) para sincronização de tempo e UDP (ver IETF RFC 768) para dados de controle. Observa-se que está fora do escopo da norma questões como organização, padronização de variáveis e de tipos de mensagens relativas ao sistema de controle.

Em *IEEE standard for passenger information system for rail transit vehicles* [IEEE, 1998], são descritas as informações oferecidas aos passageiros como, por exemplo, sinais sonoros emitidos antes de fechar as portas e a comunicação via rádio com a estação central.

Em *IEEE Trial-Use Standart for Message Set Template for Intelligent Transortation Systems* [IEEE, 2000b], são definidos os requisitos para troca de dados, tais como identificador de mensagem, relatório descritivo da mensagem, nome do usuário, etc. Define também que a apresentação dos dados deve ser feita usando a ASN.1 (*Abstract Syntax Notation One*), que é uma notação formal que permite definir tipos de dados e especificar valores que estes podem assumir.

Em *IEEE Standart for the Interface Between the Rail Subsystem and the Highway Subsystem at a Highway Rail Intersection* [IEEE, 2002], é definido o protocolo de comunicação entre o sistema de controle ferroviário e outros sistemas de controle de transporte, visando, principalmente, a uma solução para a intersecção de trajetórias entre composições e outros sistemas de transporte como rodovias.

A norma *IEEE Standard for Verification of Vital Functions in Processor-Based Systems Used in Rail Transit Control* [IEEE, 2000a], determina as verificações necessárias para equipamentos microprocessados utilizados em aplicações críticas de sistemas de transporte. A verificação é realizada nos níveis conceitual, funcional e de implementação. O nível conceitual refere-se a basicamente à identificação dos requisitos do sistema. O nível funcional refere-se à identificação de todas as funções vitais requeridas. O nível de implementação refere-se à identificação de todos os *softwares* e *hardwares* que executam funções vitais.

Em *IEEE standard for the functioning of and interfaces among propulsion, friction brake, and train-borne master control on rail rapid transit vehicles* [IEEE, 1999b] são determinadas as interfaces e funcionalidades dos sistemas de controle, propulsão e parada. As interfaces são divididas em sinais do tipo "*on-off*", analógico/proporcional e de comunicação serial.

Na tabela 2.3 é apresentada uma listagem simplificada das normas IEEE, acima apresentadas, entre outras relacionadas, mas de menor impacto no trabalho em questão.

Tabela 2.3 Síntese das normas relativas a Sistemas APM

16™-2004	*IEEE Standard for Electrical and Electronic Control Apparatus on Rail Vehicles*
1473™-1999	*IEEE Standard for Communications Protocol Aboard Trains*
1475™-1999	*IEEE Standard for Functioning of and Interfaces Among Propulsion, Friction Brake, and Train-Borne Master Control on Rail Rapid Transit Vehicles*
1476™-2000	*IEEE Standard for Passenger Train Auxiliary Power Systems Interfaces*
1477™-1998	*IEEE Standard for Passenger Information System for Rail Transit Vehicles*
1478™-2001	*IEEE Standard for Environmental Conditions for Transit Rail Car Electronic Equipment*
1482.1™-1999	*IEEE Standard for Rail Transit Vehicle Event Recorders*
1483™-2000	*IEEE Standard for Verification of Vital Functions in Processor-Based Systems Used in Rail Transit*
1536™-2002	*IEEE Standard for Rail Transit Vehicle Battery Physical Interface*
1558™-2004	*IEEE Standard for Software Documentation for Rail Equipment and Systems*
1568™-2003	*IEEE Recommended Practice for Electrical Sizing of Nickel-Cadmium Batteries for Rail Passenger*
1698™-2009	*IEEE Guide for the Calculation of Braking Distances for Rail Transit Vehicles*
1474.1™-2004	*IEEE Standard Method for Communications-Based Train Control (CBTC) Performance and Functional Requirements*
1474.2™-2003	*IEEE Standard for User Interface Requirements in Communication Based Train Control*
1474.3™-2008	*IEEE Recommended Practice for Communications-Based Train Control (CBTC) System Design and Functional Allocations*
1570™-2002	*IEEE Standard for the Interface Between the Rail Subsystem and the Highway Subsystem at a Highway Rail Intersection*
1628™-2009	*IEEE Standard Recommended Practice for Maintenance for DC Overhead Contact Systems for Transit Systems*
1653.2™-2009	*IEEE Standard for Uncontrolled Traction Power Rectifiers for Substation Applications Up to 1500 V DC Nominal Output*

2.1.4.3 Grupo de trabalho da ABNT/CEE-121

Em 31 de março de 2009, coordenada pela Profa. Dra. Morgana Pizzolato, foi instalada a Comissão de Estudo Especial de Sistema APM - ABNT/CEE-121, tendo como objetivo a normalização brasileira de sistemas APM (*Automated People Mover*), compreendendo requisitos mínimos de segurança e desempenho, bem como informações sobre veículos e sistemas de propulsão e de frenagem.

A principal referência utilizada para elaboração da norma brasileira é a ANSI / ASCE /

T&DI 21, a qual é dividida nas seguintes partes:

- ANSI/ASCE/T&DI 21-05: *Automated People Mover* - parte 1. Fornece informações sobre as condições ambientais de operação, disponibilidade, confiabilidade de serviço, requisitos da comunicação de áudio e vídeo e sistema ATC.
- ANSI/ASCE/T&DI 21.2-08: *Automated People Mover* - parte 2. Fornece informações sobre o veículo, sistema de propulsão e frenagem.
- ANSI/ASCE/T&DI 21.3-08: *Automated People Mover* - parte 3. Fornece informações sobre os requisitos dos equipamentos elétricos, estação e via.
- ANSI/ASCE/T&DI 21.4-08: *Automated People Mover* - parte 4. Fornece informações sobre segurança, preparação para emergências, verificação do sistema e demonstração, monitoramento da operação, manutenção e treinamento.

A norma brasileira, assim como a ANSI/ASCE/T&DI 21, será ampla, tratando desde a terminologia, requisitos ambientais, disponibilidade, confiabilidade, mantenabilidade, análise de perigos, comunicação com passageiros, sistema ATC, requisitos do veículo, propulsão, frenagem, equipamentos elétricos, estação e via. O projeto de norma é dividida em 6 partes:

1. terminologia, símbolos e abreviaturas [ABNT, 2012a];
2. requisitos do ambiente de operação, dependabilidade, segurança e comunicação de áudio e vídeo [ABNT, 2012b];
3. requisitos do Sistema de Controle Automático [ABNT, 2012c];
4. requisitos do veículo, propulsão e freio;
5. requisitos dos equipamentos elétricos;
6. requisitos da estação e da via.

Com relação aos requisitos da comunicação de áudio e vídeo do Sistema APM [ABNT, 2012b], uma vez que o sistema é baseado no barramento *ethernet*, a comunicação de áudio e vídeo pode ser realizada respectivamente por meio de VoIP (voice over IP) e câmeras IP, facilitando, inclusive o processo de gravação de transmissões de áudio e vídeo e no sistema de comunicação direcionado ao público nas estações e na comunicação direcionada aos passageiros na composição.

No que tange o controle automático de sistemas APM, a norma, em desenvolvimento pela ABNT/CEE-121, trata do projeto de falha segura do sistema ATC, do sistema de proteção automática, de operação automática e supervisão automática.

Com relação ao projeto de falha segura do sistema ATC, a norma determina que o projeto e a implementação de todos os elementos de *hardware* e *software* críticos de segurança do sistema devem ser submetidos à verificação e validação, o que deve incluir a verificação da redundância (múltiplas versões), diversidade de auto-teste e garantia numérica. Em termos de projeto, o uso de autômatos temporizados, por sua característica de modelagem próxima do sistema implementado final e sua capacidade de verificação formal, permite a realização de testes para todas as possibilidades lógicas de entrada na condição de falha.

Com o foco na validação do sistema implementado, de forma a verificar se o modelo está em conformidade com o código implementado a partir da verificação formal, a ferramenta UPPAAL TRON [Hessel et al., 2008b] permite a realização de testes de conformidade através da comunicação direta entre a rede de autômatos temporizados e o CLP programado.

Segundo a ABNT/CEE-121, as funções do ATP devem ter prioridade sobre as funções do ATO e ATS. Cada função do sistema ATP pode ser verificada e validada isoladamente. Nos casos em que o ATS venha a se tornar inoperante, o ATP e o ATO devem permanecer operáveis. As funções dos sistemas ATP, ATO e ATS estão listadas a seguir:

- Funções do ATP:
  - Detecção de presença - visa a garantir a separação das composições e/ou interruptores de bloqueio da via. A detecção deve ser contínua em qualquer trecho da via.
  - Garantia de separação - visa a fornecer proteção contra aproximação excessiva entre composições. Assim, o sistema deve calcular continuamente a distância necessária, entre veículos e entre eles e as estações, para uma parada segura.
  - Detecção de movimento involuntário - visa a evitar, através do uso do freio de emergência, a movimentação de uma composição sem que esta seja comandada, ou, se for detectado o movimento de uma composição em sentido de deslocamento inverso ao permitido.
  - Proteção contra sobrevelocidade - visa a evitar que composições ultrapassem a velocidade limite determinada em projeto.
  - Proteção contra deslocamento excessivo - visa a evitar que composições ultrapassem a extremidade final da via.
  - Proteção contra separação de veículos - visa a detectar a separação de veículos acoplados.

  - Proteção de perda de sinal - no caso de perda de sinal de comunicação deve ocorrer a frenagem automática pelo ATP.
  - Detecção de velocidade zero - a velocidade é considerada nula quando inferior a 0,30 m/s.
  - Proteção de abertura não programada das portas - visa a frenagem até parada total no caso de destravamento de portas enquanto a composição estiver em movimento.
  - Bloqueios da proteção de controle das portas - determina que as portas podem ser abertas somente quando a composição estiver na plataforma da estação e a velocidade zero, com os motores do sistema de propulsão desenergizados, desacoplados ou bloqueados e a composição bloqueada, impedindo seu movimento.
  - Bloqueios de partida - determina que, para a partida do veículo, todas as portas deve estar fechadas e travadas.
  - Bloqueios de inversão de sentido - determina que a inversão de sentido deve ocorrer somente quanto a velocidade zero é identificada e em locais determinados.
  - Intertravamento de propulsão e frenagem - determina que, uma vez que o freio de emergência seja acionado, deve permanecer ativado até a parada completa da composição. Os controles do freio de emergência devem ser intertravados com os controles da propulsão, de maneira que os comandos do freio dominem.
  - Intertravamento do sistema de mudança de via - visa a assegurar o travamento do aparelho de mudança de via (AMV) quando ocorre a passagem da composição.

- Funções do ATO: envolve o controle de movimento (aceleração, desaceleração e *jerk*), a parada programada, assim como, o controle das portas e tempo de espera na estação.
- Funções do ATS: envolve o monitoramento da situação e desempenho, o cancelamento do controle automático, o rastreamento da composição, o gerenciamento do *headway* e os alarmes do sistema.

2.2 Estado da Arte de Sistemas APM

A seguir são apresentadas as soluções encontradas para o controle de Sistemas APM. Cabe ressaltar que, na maior parte dos artigos analisados, as informações são pouco detalhadas sobre a implementação do sistema ATC, logo, foram referenciados somente os artigos mais significativos quanto à descrição da tecnologia utilizada.

Em Ghosh et al., 1995, é apresentada uma arquitetura e uma metodologia para desenvolvimento de controladores para segurança de trens de transporte de passageiros. A metodologia é baseada na simulação de falhas no sistema e é aplicada para detectar erros tanto no *software* quanto no *hardware* de controle. O sistema é desenvolvido em VHDL (*VHSIC Hardware Description Language*).

Conley, 2001, apresenta o sistema $Flexiblok^{TM}$, no qual as funções de ATC são divididas entre funções internas e externas do trem. O ATC externo consiste do ATS, do RATO (*Region Automatic Train Operation*) e do RATP (*Region Automatic Train Protection*). No trem, o ATC é constituído pelo VATO (*Vehicle Automatic Train Operation*) e pelo VATP (*Vehicle Automatic Train Protection*). Um sistema de rádio é utilizado para estabelecer a comunicação entre o ATC interno com o externo. O ATC externo é responsável por garantir uma região de segurança para todas as composições em funcionamento, levando em consideração, por exemplo, o tempo para frenagem e o cálculo da relação entre posição e velocidade em curvas. Uma segunda rede de comunicação provê conectividade entre os sistemas ATC, localizado fora da via, com a central de controle. A principal função desta rede consiste na realização da comunicação entre diferentes ATC na troca de regiões. A Figura 2.24 ilustra a comunicação entre o veículo e a central de controle.

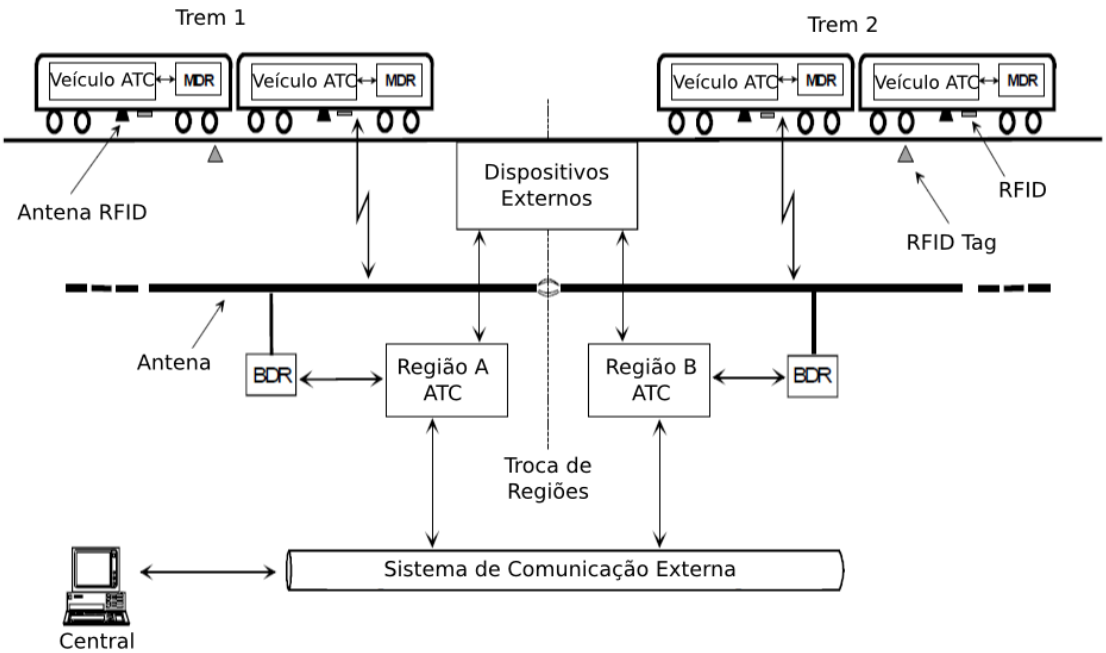


Figura 2.24 Sistema Flexiblok da Bombardier [Conley, 2001]

Na Figura 2.24 pode-se observar que os veículos possuem antena RF-ID para leitura dos sensores de posição, mantendo o sistema de controle embarcado atualizado quanto à localização

e velocidade do veículo. É permitida a comunicação entre os veículos e o controle central é efetuado através de antenas organizadas por regiões.

Em Amendola et al., 2007, é descrita a arquitetura de funcionamento de um APM (ATC) onde o controle vital do sistema é realizado pelos controladores MICROLOK II *Interlocking Controller* e AF-902 *Track Circuit Controller*. Os sistemas ATP e ATO estão centralizados em equipamentos localizados dentro do veículo.

Em Bailey, 2007, é descrita a utilização do CITYFLO 650 (CBTC). Esta solução utiliza uma rede de alta velocidade para comunicação entre o sistema ATP, o sistema ATO e a central de controle. Caminhos redundantes são disponibilizados através do uso de fibra ótica.

Em Freitas e Mira, 2007, é descrita a arquitetura utilizada para comunicação audiovisual com os passageiros via sistema de comunicação. Neste caso, tanto o sistema de controle, quanto o sistema de comunicação audiovisual utilizam o mesmo meio de comunicação. Os autores ressaltam que deve-se avaliar os custos para que o sistema de controle possa compartilhar dados menos críticos (como a comunicação de áudio e vídeo) com a comunicação de controle de distanciamento entre composições.

Na maioria dos sistemas CBTC, os dados trocados entre o trem e o sistema de controle central são transferidos bidirecionalmente via comunicação sem fio. Principalmente para trajetos a longas distâncias, utiliza-se o *Global System for Mobile Communications Railway* (GSM-R) para realizar a comunicação entre o trem e a central de controle, porém, para deslocamentos urbanos, a *Wireless Local Area Network* (WLAN) é a melhor escolha segundo Cao et al., 2007.

Segundo Cao et al., 2007, existem, porém, algumas dificuldades no uso de WLAN. Estas dificuldades estão exemplificadas através da Figura 2.25.

Quando o trem 1 sai da área de cobertura da antena 1 (*Access Point* 1 - AP1) e entra na área da antena 2 (AP2) há necessidade de um tempo para a troca de antenas. Durante este tempo, o trem não consegue comunicar-se com a central de controle. Isto afeta diretamente o sistema de frenagem de emergência do trem 2, tornando importante conhecer com precisão este tempo de troca de comunicador.

Nos estudos realizados por Cao et al., 2007, o tempo necessário para troca de comunicador ficou entre 70 e 120 ms, com a probabilidade do tempo de troca estar dentro deste intervalo de 96,49% e confiança de 96,49%.

Além destes estudos, é ainda possível destacar o trabalho de Kuun, 2004, no qual é descrito com considerável nível de detalhes o uso de rádio frequência para sistemas CBTC e

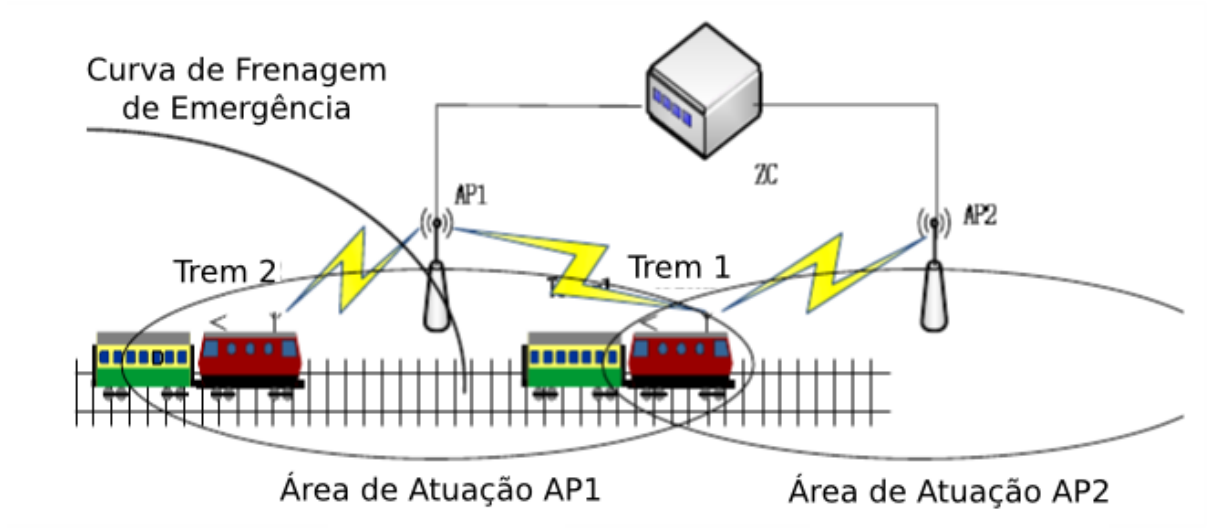


Figura 2.25 Exemplo do uso de WLAN e CBTC [Cao et al., 2007]

CCTV para velocidades acima de 120 km/h e Sullivan, 2003, onde é observado que o uso da IEEE P802-20 apresenta potencial de comunicação para veículos com velocidade máxima de cruzeiro de até 250 km/h.

Conclui-se a partir das referências citadas anteriormente, que as informações existentes sobre as soluções para Sistemas ATC restringem-se a organização dos equipamentos de comunicação e de controle do Sistema APM. Desta forma este estudo auxilia na concepção da proposta de organização física de controle e operação do Sistema APM para o Aeromovel conforme pode ser observado no Capítulo 3.

3 AUTOMAÇÃO DO SISTEMA AEROMOVEL DE TRANSPORTE

Neste capítulo são apresentados as estruturas físicas e funcionais do sistema de controle proposto para o Aeromovel.

3.1 Estrutura Física de Comunicação

A Figura 3.1, apresenta a estrutura física do sistema de controle proposta para o GMP com base no revisão apresentada na Seção 2.2. Os equipamentos designados como controle 1, 2 e 3 são sistemas computacionais de processamento de dados em tempo real. O controle 1 é responsável pelos nós lógicos de controle e sensoriamento do veículo (como, por exemplo, o controle de portas, velocidade, frenagem, etc.). Os controles 2 e 3 são responsáveis pelas funcionalidades de controle dos GMP (como, por exemplo, o controle de abertura de válvulas). Cada controlador pode conter diferentes nós lógicos. Cada um dos controladores anteriormente citados utiliza mensagens em tempo real (*real-time*) assim como mensagens não tempo real. As mensagens de tempo real são utilizadas principalmente para a comunicação entre os nós lógicos de controle e as mensagens de não tempo real para comunicação com o sistema supervisório (ATS).

Para comunicação entre o controlador 3 e os demais sistemas, faz-se necessário o uso de um meio de comunicação sem fio (*wireless*). Para a comunicação entre os demais equipamentos utiliza-se cabeamento por fibra ótica, levando em consideração a necessidade de redundâncias com topologia de anel duplo. Os *switches* utilizados para comunicação entre equipamentos devem estar de acordo com a norma IEC 61850 devido à necessidade de controle de prioridade entre mensagens de tempo real (Goose e SMV) e de não tempo real (MMS).

Uma possibilidade de modelo para simulação do sistema é apresentada na Figura 3.2. Observa-se que, para este caso, é necessária a utilização de um modelo matemático para o sistema GMP. Esta simulação envolve a verificação de requisitos de tempo na comunicação entre os controladores, logo, sugere-se a utilização do simulador RTDS (*Real Time Digital Simulator*).

3.2 Estrutura de Controle

Na Figura 3.3 é exemplificada a estrutura física de controle do Sistema Aeromovel de Transportes necessária para o controle de posicionamento do veículo. Neste caso utiliza-se

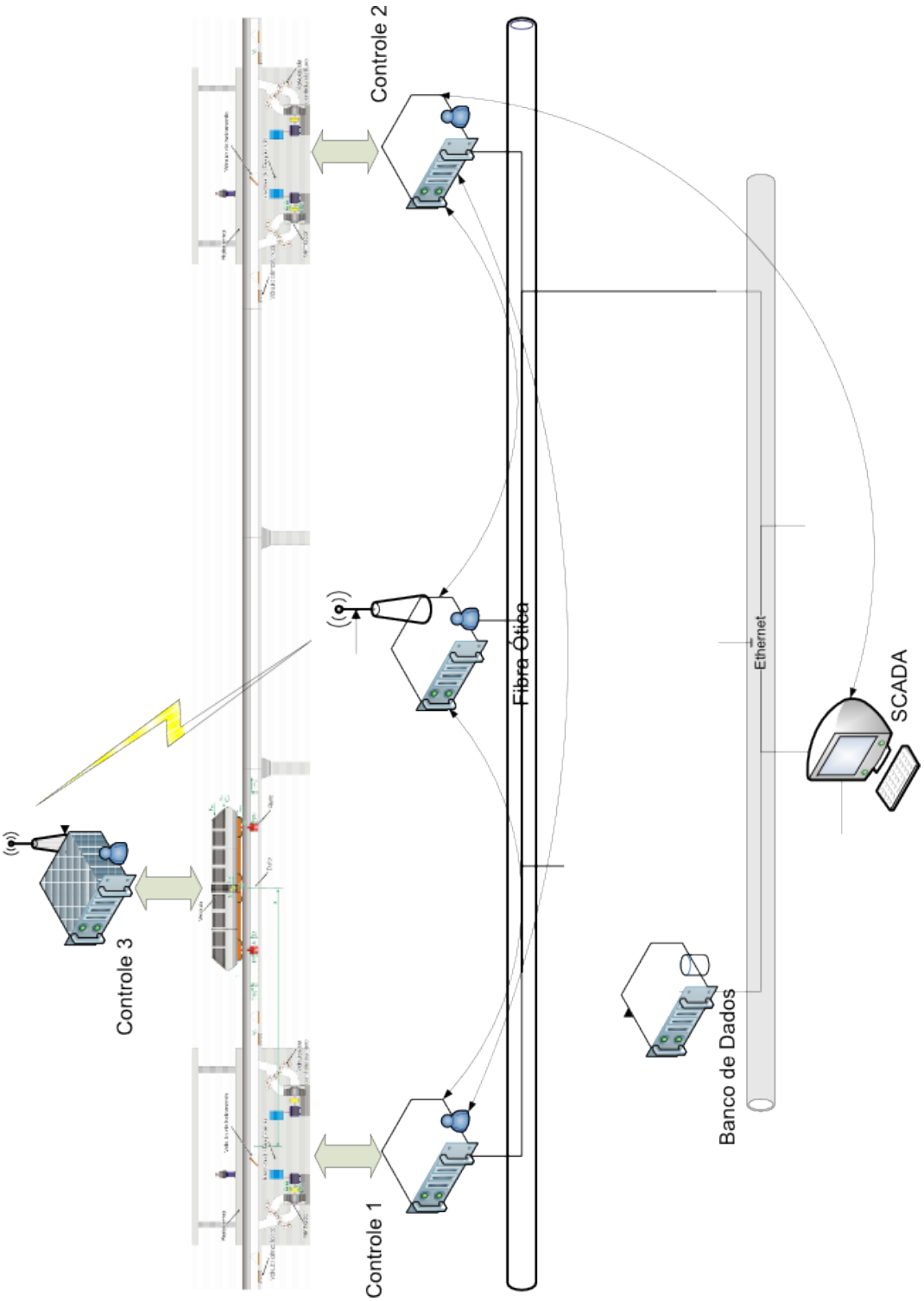


Figura 3.1 Proposta para automação do Sistema Aeromovel de Transporte

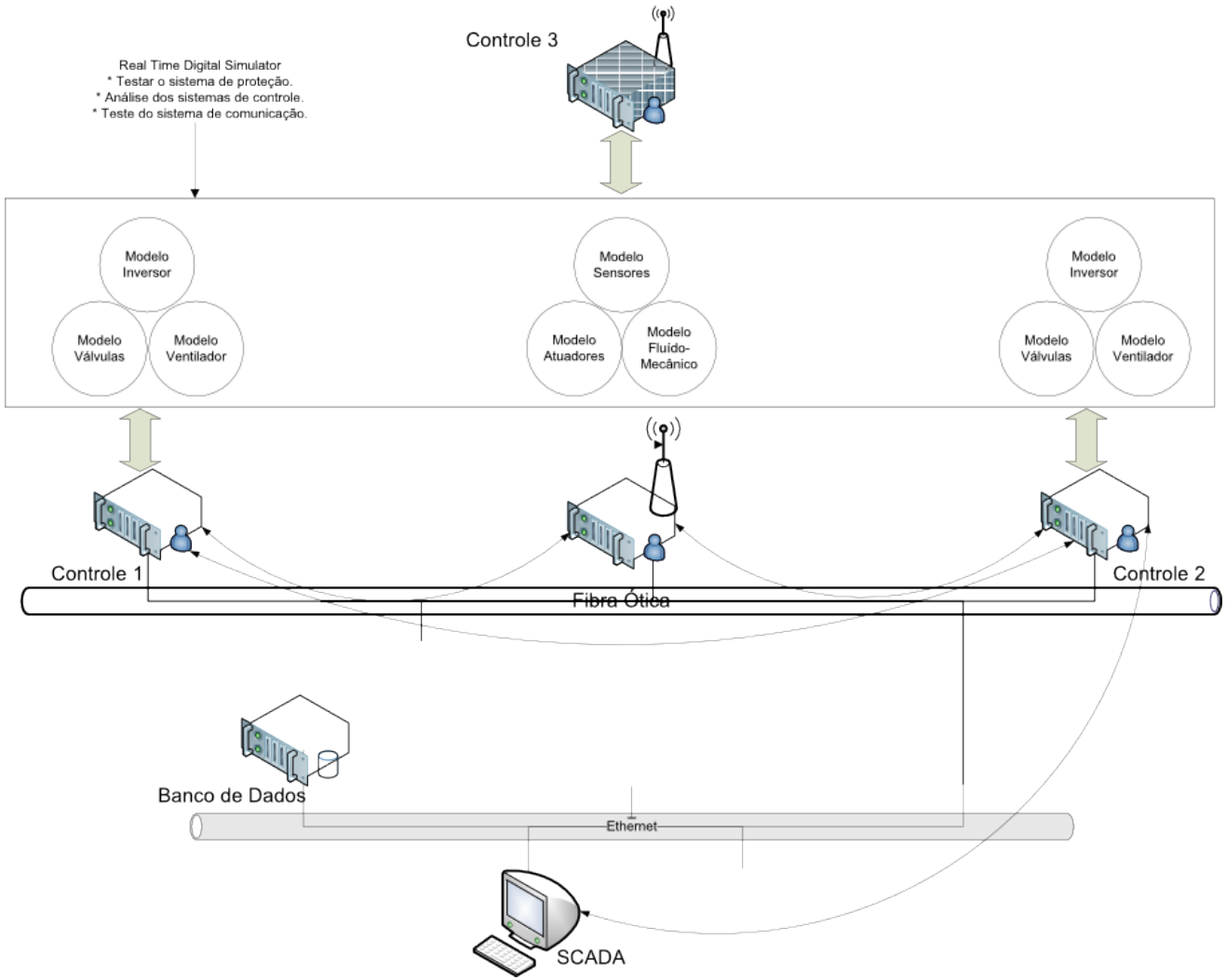


Figura 3.2 Proposta para simulação do Sistema Aeromovel de Transporte

dois diferentes métodos para localização do veículo na via: um *encoder* e um *transponder*. O *encoder* é utilizado juntamente com o *transponder*, de forma que o *encoder* reinicia sua contagem a cada *transponder* encontrado, evitando assim o acúmulo de erros ocasionados pela contagem do sistema localizado no trem, devido, por exemplo, à mudança do diâmetro das rodas ocasionado pelo desgaste da mesma [Sarmanho, 2009]. A Figura 3.3 apresenta um esquema representativo desta configuração.

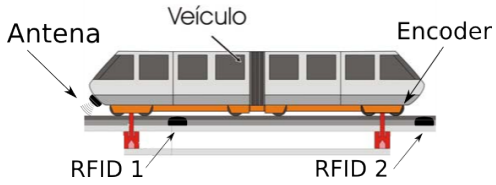


Figura 3.3 Utilização de *encoder* juntamente com *transponders*

Os sinais do *encoder* e do *transponder* são representados como dispositivos lógicos. O controle de posição utiliza dados adquiridos pelo sistema de medição baseado em *transponder* e *encoder*. Neste caso, foi utilizado o conceito de *Merge Unit* para os sistemas de instrumentação devido à necessidade de tradução de diferentes protocolos utilizados na instrumentação para o padrão da norma IEC 61850. A *Merge Unit* realiza a tradução de protocolos utilizados em instrumentos para a IEC 61850, conforme apresentado na Figura 3.4.

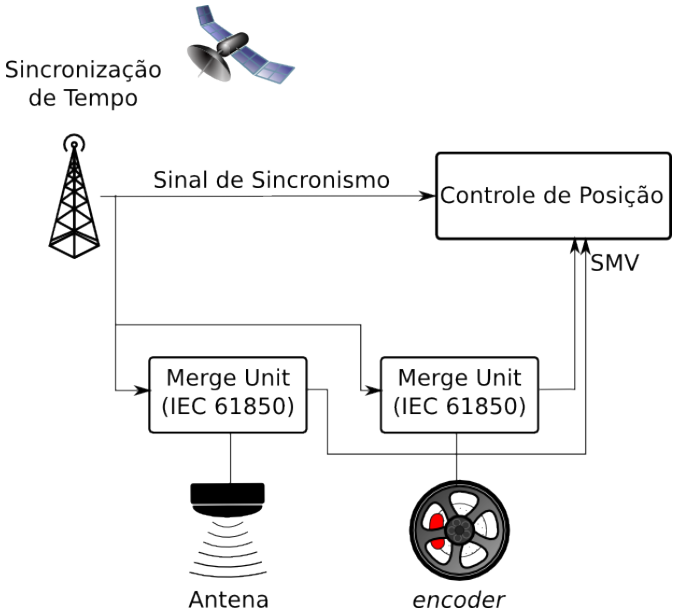


Figura 3.4 Sincronização de tempo no Sistema Aeromovel de Transporte

Os instrumentos podem ser instalados distantes do Controlador de Posição, logo, é necessário que cada instrumento possua seu próprio sistema de comunicação, evitando a inserção de ruídos devidos à transmissão de sinais na forma analógica ao longo do veículo até o conversor analógico digital. Neste caso, o *timestamp* (o instante de tempo no qual o dado

foi amostrado) é inserido no momento da conversão analógico-digital, via sinal de sincronismo de tempo.

O Controle de Posição relaciona-se com os demais nós lógicos através dos protocolos especificados na IEC 61850. Neste caso, o protocolo MMS (representado na Figura 3.5 como NRT - Não Tempo Real) é utilizado para comunicação com o sistema supervisório e configuração e os protocolos SMV e Goose (representados na Figura 3.5 como RT - Tempo Real) são, respectivamente, utilizados para os dados dos sensores e eventos de falha no sistema.

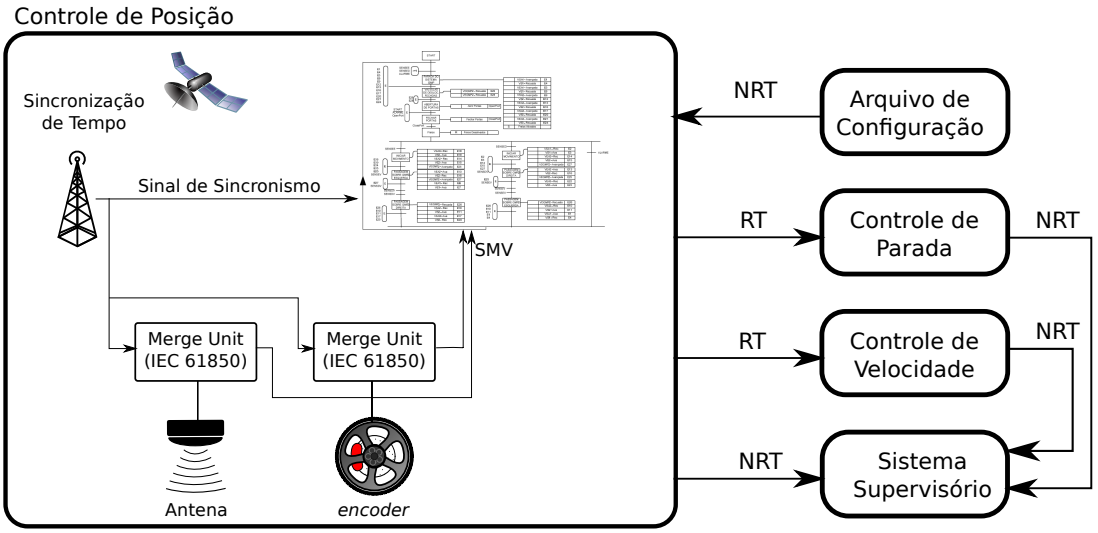


Figura 3.5 Controle de posicionamento

Observa-se através da Figura 3.5, que o cálculo de posição depende dos dispositivos lógicos. Os controles de velocidade e de parada dependem do cálculo realizado pelo controlador de posição, sendo que a lógica do controle de posição está representada por um diagrama funcional.

Na Tabela 3.1 são apresentadas as informações referentes ao dispositivo lógico "Sensor de Deslocamento" que alimenta o controle de posição do veículo [IEC, 2007]. Para todos os nós lógicos há herança do nó lógico padrão (*Common Logocal Node*) que possui informações e serviços independentes da função realizada (ver [IEC, 2003d]). Os tipos ING (*Integer Status Setting*) e SAV (*Sampled Value*) são classes de dados e estão definidos em IEC, 2003e, onde ING é utilizado para representar valores inteiros utilizados para configuração do nó lógico e SAV é uma classe de dados utilizada para representar valores amostrados que são usualmente transmitidos através de mensagens de tempo real SMV (*Sampled Value*).

A exemplo das informações apresentadas na Tabela 3.1, devem ser determinadas as classes para os demais dispositivos utilizados no sistema APM, onde, neste caso, ING indica

Tabela 3.1 Nó lógico TDST

Nome do Atributo	Tipo	Descrição	Opcional/Obrigatório
SmpRteRng	ING	Possibilidades de Frequências de Amostragem	Opcional
Dis	SAV	Distância [m]	Obrigatório
SmpRte	ING	Frequência de Amostragem	Opcional

Integer Status Settings e SAV indica *Sample Analog Value*.

A visão geral do sistema pode ser entendida através da Figura 3.6, na qual observa-se que o GMP recebe energia elétrica e a transforma em energia pneumática que realiza a movimentação do veículo. O veículo tem como entradas e saídas, além dos passageiros, a comunicação com os sistemas ATO e ATP através dos sistemas SA (acionamento de portas e freio), SM (medição de posição através dos *encoders* e *transponders*, como visualizado na Figura 3.4) e SAM (sistema de comunicação com os passageiros).

Neste caso, observa-se que os Sistemas de Atuação e Medição (SAM), de Atuação (SA) e de Medição (SM) são dispositivos lógicos que realizem o interfaceamento do veículo (via sistema *wireless*) e do GMP (via fibra ótica) com os nós lógicos de controle (ATP e ATO). Neste caso, a comunicação de controle é prioritariamente em tempo real utilizando-se as mensagens Goose e SMV, visto que cada sensor ou atuador é apresentado no sistema de controle como um dispositivo lógico. No caso de o equipamento de atuação ou medição não estar de acordo com a norma IEC 61850, são utilizados adaptadores de protocolo (*Merge Unit*), como apresentado na Figura 3.4.

Os Sistemas de Pedido (SP), Resposta (SR) e Pedido e Resposta (SPR) representam a comunicação entre os nós lógicos e o sistema supervisório e utilizam mensagens MMS (sem requisitos de tempo real), como ilustrado na Figura 3.5.

Os blocos ATO e ATP representam, respectivamente, a operação e proteção do veículo. O bloco veículo tem como entrada e saída os usuários do sistema de transporte e o bloco grupo propulsor tem como entrada energia elétrica e como saída diferença de pressão de ar para o ambiente e para o veículo. Ambos os blocos ATO e ATP possuem parte de suas funções localizadas fisicamente no veículo e parte no GMP. A comunicação entre os nós lógicos responsáveis pela proteção e operação devem possuir requisitos de tempo crítico.

Observa-se a existência de dois tipos de usuários: o usuário operador e usuário do transporte. O usuário operador utiliza o sistema ATS para monitoramento do veículo. O usuário do transporte, além da interface com o veículo tem possibilidade de realizar comunicação com o usuário operador em caso de necessidade. Essa comunicação é realizada através do canal SAM.

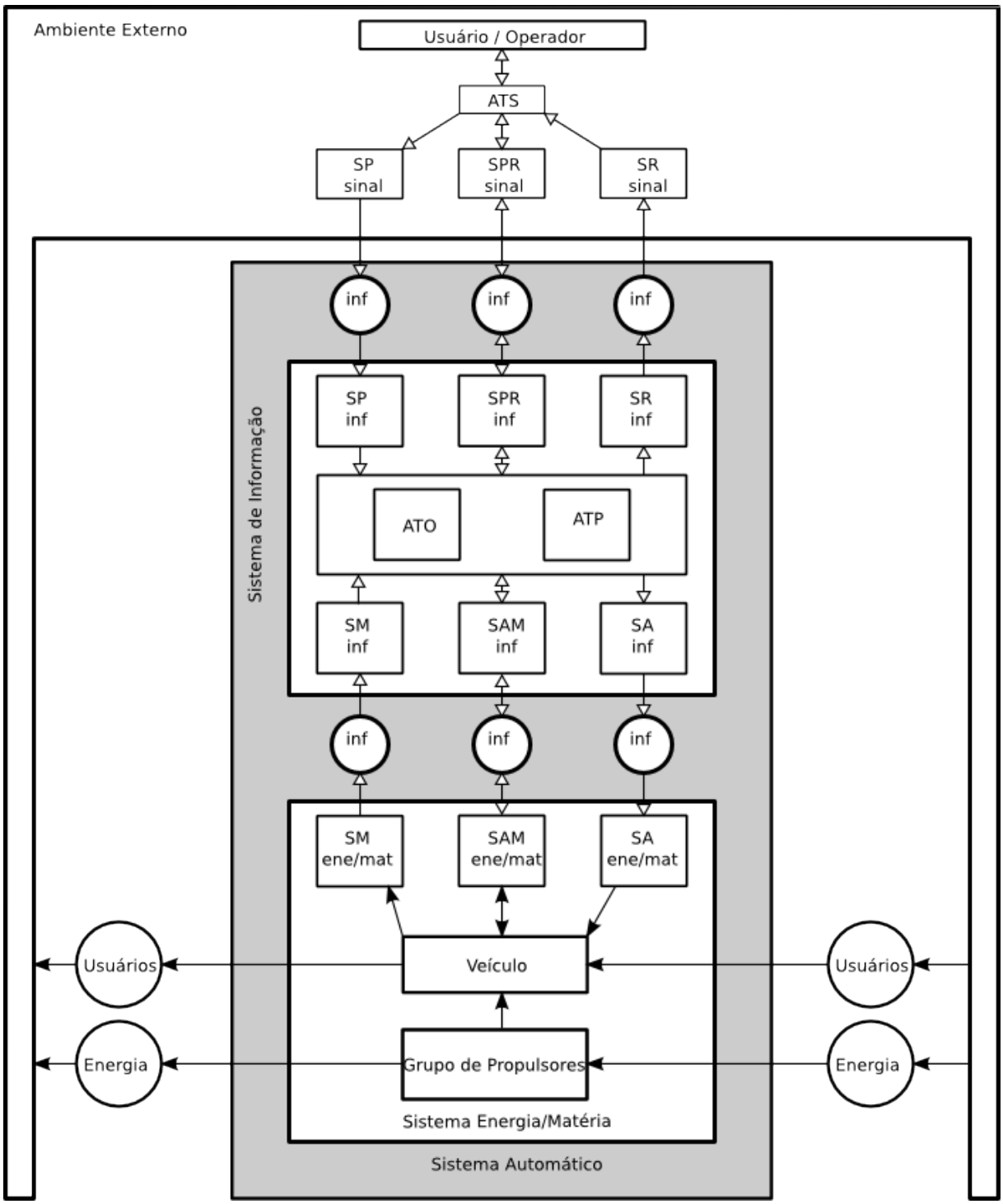


Figura 3.6 Visão geral da automação do Sistema Aeromovel de Transporte

4 TRANSPORTE DE PASSAGEIROS - AEROMOVEL

Neste capítulo é apresentada a representação dos modelos dos equipamentos e controladores necessários para a movimentação do sistema Aeromovel. Para a modelagem foram utilizados autômatos temporizados visando à realização de simulações e da verificação formal para validação dos mesmos. Estão apresentados os modelos das válvulas pneumáticas, ventiladores centrífugos, freios e portas operando em regime normal de funcionamento com seus respectivos controladores.

4.1 Visão Geral

Escolheu-se uma trajetória e um regime de funcionamento considerado adequado para a realização dos testes de validação da metodologia, não representando necessariamente o *modus operandi* vigente no sistema Aeromovel de transporte de passageiros. Para tanto, os modelos foram desenvolvidos a partir dos esquemas encontrados na bibliografia e modificados de maneira a diminuir a quantidade de estados presentes na modelagem em autômatos temporizados, evitando a condição de "explosão" de estados durante a análise computacional, mantendo, porém, o comportamento dos equipamentos, sob o ponto de vista da automação, inalterado.

Com a finalidade de validar a metodologia, foi idealizado um deslocamento na forma de um circuito fechado, contendo duas estações para embarque e desembarque de passageiros, em pontos diametralmente opostos. Adicionalmente, para testes de proximidade entre veículos, foram inseridos dois veículos e dois Grupo Moto Propulsores (GMP) de forma que, para cada veículo, exista apenas um GMP responsável pelo seu movimento. O critério para identificação do GMP responsável por um determinado veículo é o de proximidade, ou seja, de acordo com a posição do veículo. Se, inicialmente, o GMP "A" empurrando (estado de *PUSH*) o veículo, a partir de determinada posição o GMP "B" vai começar a puxar (estado de *PULL*) o mesmo veículo. Observa-se que, neste caso, não foi incluído o funcionamento empurra e puxa (*PUSH/PULL*) para operar concomitantemente no acionamento do mesmo veículo. Desta forma o veículo é puxado ou empurrado somente por um dos GMP em cada instante. Os motivos para não utilizar o sistema *PUSH/PULL* são os seguintes:

1. Aumento da quantidade de estados, pois em um sistema circular com dois veículos (necessários para teste de proximidade) seriam necessários ao menos três GMP, o que acarretaria em um aumento significativamente o tempo e os recursos computacionais

necessários para a verificação formal dos modelos.

2. Aumento da complexidade dos GMP resultante do aumento do número de válvulas necessárias para gerenciar o movimento dos veículos.

Com o objetivo de dividir a área de influência (ou domínio) entre os GMP, o trajeto definido pelos trilhos foi seccionado em 6 diferentes segmentos, conforme pode ser observado na Figura 4.1.

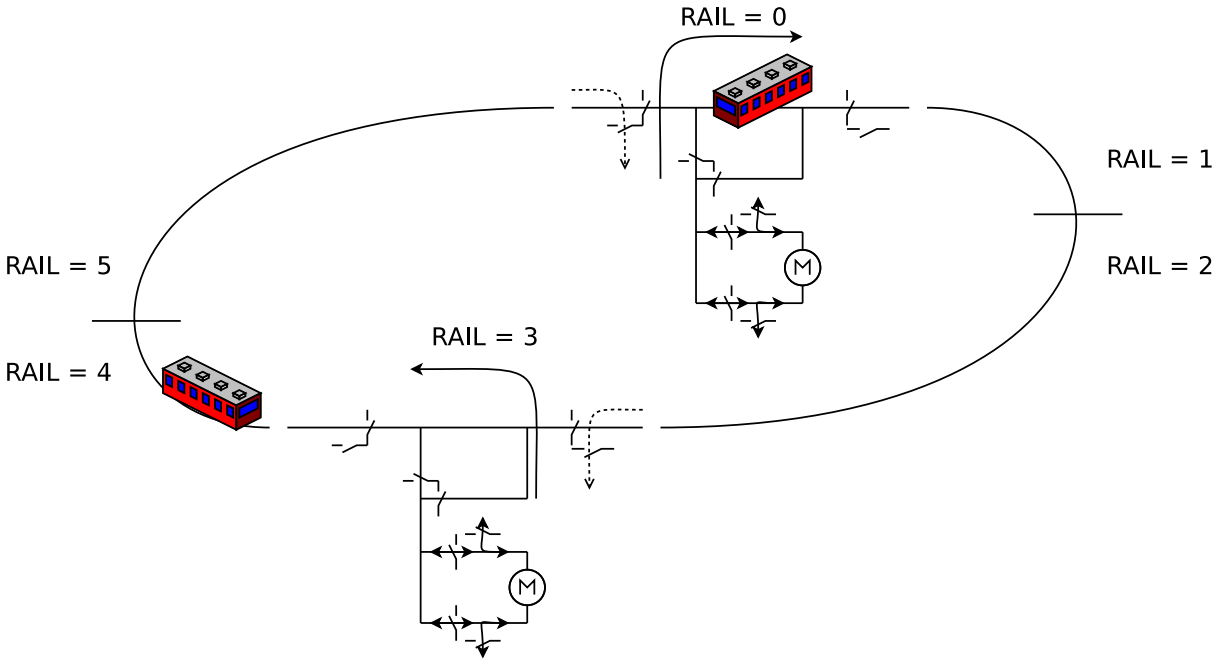


Figura 4.1 Divisão do trilho em segmentos

Observa-se na Figura 4.1 que os segmentos 0 e 3 representam os trechos das estações e as seções 1, 2, 4 e 5 estão associadas às divisões dos trilhos. Estes segmentos não descrevem equipamentos existentes nos trilhos, mas sim divisões não físicas utilizadas como limitadores de trecho e reconhecidas pelos GMP para a identificação da posição dos veículos e definição da responsabilidade dos GMP na propulsão dos veículos.

A lógica de associação utilizada prevê que cada GMP é responsável pelo veículo mais próximo em condições normais de funcionamento. Observa-se que não são consideradas possíveis otimizações para minimizar a perda de potência resultante das diferenças entre empurrar e puxar o veículo. Também não são consideradas como variáveis as perdas do sistema de propulsão resultante das curvaturas da trajetória, visto que o objetivo do trabalho é focado na metodologia e no funcionamento do veículo em termos de proteção visando ao aumento da confiabilidade, não fazendo parte do trabalho encontrar configurações ótimas visando a uma maior eficiência energética do sistema presente.

Considerando dois veículos T_0 e T_1 e dois Grupo Moto Propulsores GMP_0 e GMP_1,

foram listadas, abaixo, as regras existentes no modelo proposto para relacionar veículos ao GMP de acordo com a posição dos mesmos no trilho:

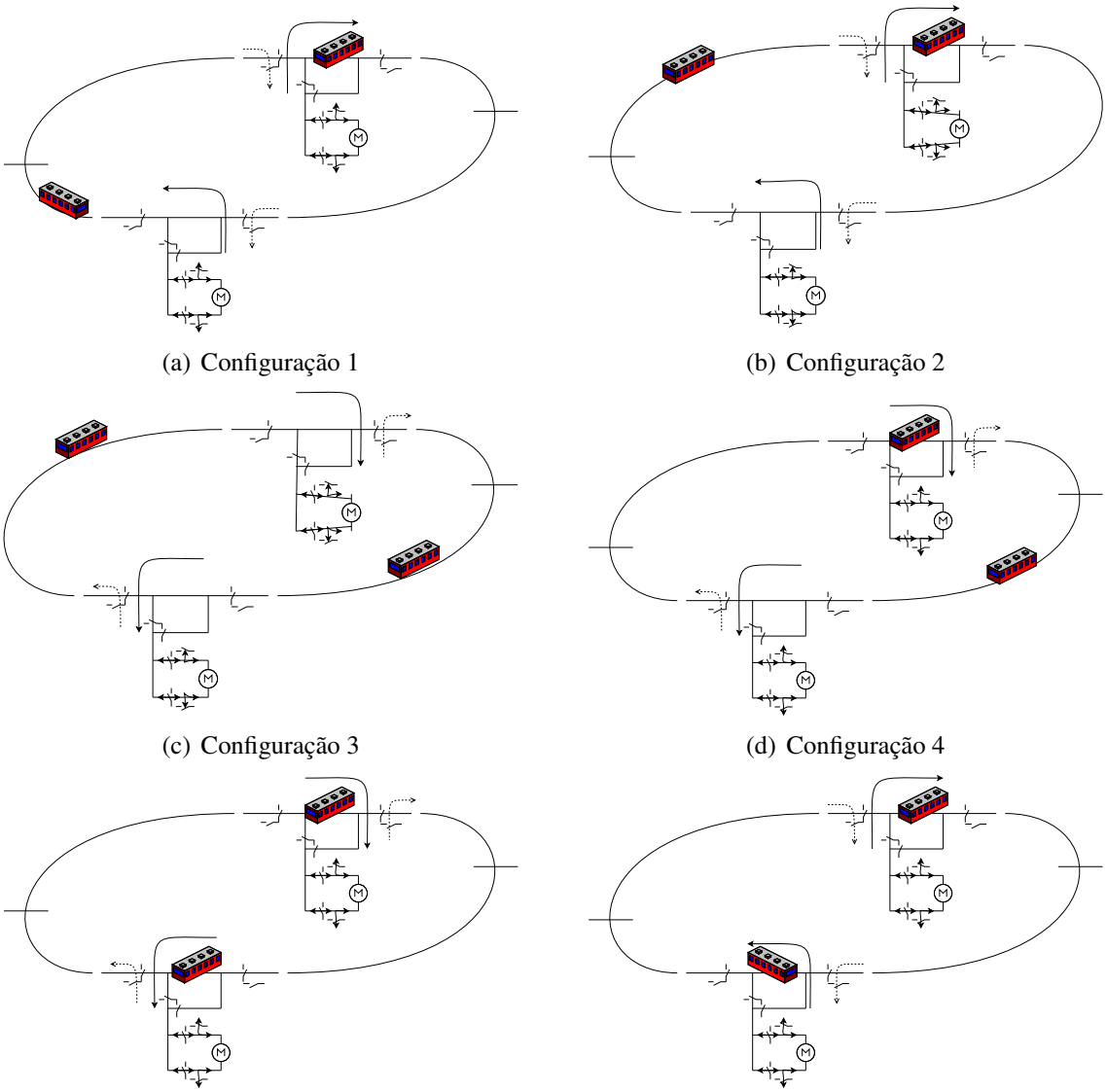
- Se T_0 está no segmento 0, o responsável pelo movimento é o GMP_0 e o responsável pelo movimento do veículo T_1 é o GMP_1, independente do segmento que T_1 estiver (ver Figura 4.2(a)).
- Se T_1 está no segmento 5 e GMP_0 está movimentando o veículo T_0 nos segmentos 0 ou 1, T_1 deve parar (ou diminuir a velocidade) e aguardar até a liberação do GMP_0 (ver Figura 4.2(b)).
- Se T_0 está no segmento 1, o responsável pelo movimento é o GMP_0. Se T_1 estiver no segmento 5, então GMP_1 deve ser responsável pelo movimento de T_0.
- Se T_0 está no segmento 2, o responsável pelo movimento é o GMP_1. Se T_1 estiver no segmento 4, então GMP_0 deve ser responsável pelo movimento de T_0 (ver Figura 4.2(c)).
- Se T_0 está no segmento 3, o responsável pelo seu movimento é o GMP_1 e do T_1 é o GMP_0, independentemente do segmento que T_1 estiver (ver Figura 4.2(e)).
- Se T_1 está no segmento 2, ele deve esperar até liberar o GMP_1 (ver Figura 4.2(d)).
- se T_0 está no segmento 4, o responsável pelo movimento é o GMP_1. Se T_1 estiver no segmento 2, então GMP_0 será responsável pelo movimento de T_0 (ver Figura 4.2(a)).
- Se T_0 está no segmento 5, o responsável pelo movimento é o GMP_0. Se T_1 estiver no segmento 1, então GMP_1 será responsável pelo movimento de T_0 (ver Figura 4.2(c)).

De acordo com a norma *IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements* [IEEE, 2004], os sensores de posição devem estar no veículo e não nos trilhos. Desta forma, o sistema foi modelado de maneira que, para cada atualização da posição tanto do veículo T_0 ou do veículo T_1, verifica-se qual GMP é responsável pelo movimento do mesmo.

Uma vez observada a necessidade de troca entre GMP devido a mudança de jurisdição dos veículos, é estabelecida uma troca de informações entre os GMP de variáveis tais como velocidade e posição evitando, assim, alterações na dinâmica do movimento do veículo que podem ser sentidas pelos passageiros durante o processo de troca de GMP ou entre a troca de *PUSH* para *PULL*, ou vice-versa.

O processo de troca entre estados dos GMP segue a seguinte sequência de eventos:

1. Uma mensagem do tipo *broadcast* vinda do controle geral avisa os GMP da necessidade



(a) Configuração 1 (b) Configuração 2

(c) Configuração 3 (d) Configuração 4

(e) Configuração 5 (f) Configuração 6

Figura 4.2 Possíveis posições relativas dos veículos

de mudança de estado devido à mudança de jurisdição dos veículos.

2. Uma vez que mensagem é recebida pelos controladores dos GMP, cada controlador envia uma nova mensagem para seus componentes (válvulas) avisando para mudarem seus estados para *standby* ou *offline* (estado no qual o GMP não interfere nos veículos). Este estado é determinado pela abertura das válvulas atmosféricas.

3. Os comandos do T_0 ou T_1 são redirecionados para os respectivos GMP de acordo com a posição dos veículos.

4. No caso de troca de GMP, ocorre também a troca das variáveis de controle do veículo (entre os GMP) de forma a não resultar em mudanças abruptas de velocidade (para o bem estar dos passageiros).

5. Após o GMP atingir o estado de *offline*, as válvulas de direcionamento de vazão são reconfiguradas para que o GMP possa atingir o estado de *PUSH* ou *PULL*.
6. A vazão do ventilador é configurada para manter aproximadamente constante a velocidade do veículo. Neste caso, não é levada em consideração a perda de velocidade resultante do período em que o veículo permaneceu em resposta livre.
7. Após a configuração das válvulas de direcionamento de vazão, as válvulas de divisão de segmento são reconfiguradas para sair da condição de *offline*.

Observa-se que há uma hierarquia entre os controladores, visto que cada GMP envia comandos somente para o seu conjunto de válvulas e motor de acordo com o segmento da via e velocidade do veículo (mesmo sendo as mensagem do tipo *broadcast*). Detalhes do funcionamento do fluxo de mensagens estão apresentadas no Capítulo 5.

4.2 Modelos

A seguir serão apresentados os modelos do sistema de propulsão, assim como os resultados das simulações e da verificação formal.

4.2.1 Grupo Moto Propulsor

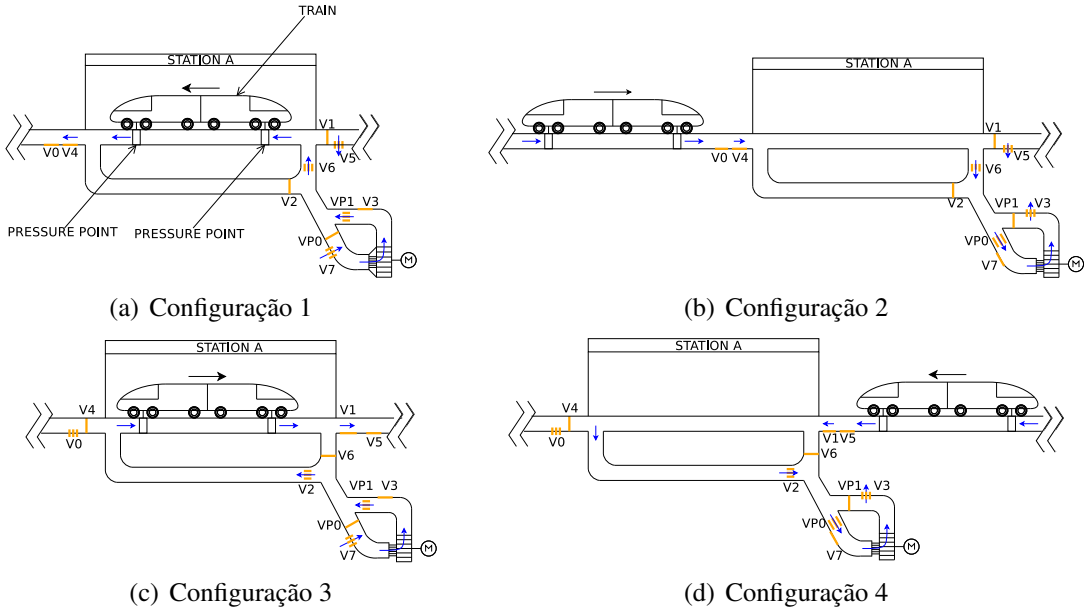
A unidade de potência, também conhecida como Grupo Moto Propulsor ou Sistema de Propulsão, é responsável pela geração de pressão diferencial e é constituída basicamente de um motor elétrico assíncrono que aciona um ventilador centrifugo industrial [Furtado, 1994]. Cada GMP é conectado ao duto principal com 1 m^2 de área de seção.

O sistema de propulsão possui capacidade de vazão de ar de até 10^6 m^3/h. Em conjunto, duas válvulas de controle proporcional (VP_0 e VP_1) permitem o controle da variação da pressão e, consequentemente, da força imposta ao veículo. O sistema de propulsão possui oito válvulas do tipo liga-desliga (V_{0-7}), e, a partir da alteração dos seus estados, o controlador pode mudar a direção da vazão de ar nos dutos, resultando no aumento ou na diminuição da pressão de ar no duto principal, como pode ser observado através da Figura 4.3(a). As válvulas utilizadas no sistema Aeromovel são similares às válvulas borboletas usualmente encontradas em aplicações industriais. Pistões pneumáticos são utilizados para movimentar as aletas das válvulas às altas vazões de ar envolvidas no processo.

Conforme visto no Capítulo 1, de acordo com Aeromovel, 1999, o bloco padrão é formado por dois grupos moto propulsores que disponibilizam três diferentes possibilidades de

configurações: *Push* (ver Figura 4.3(a) e Figura 4.3(c)), *Pull* (ver Figura 4.3(b) e Figura 4.3(d)) e *Push-Pull*. Observa-se que o funcionamento *Push-Pull* não foi implementado pois, conforme citado anteriormente, o sistema foi planejado com a restrição de que um grupo moto propulsor deve controlar somente um veículo de cada vez, simplificando, desta forma, os modelos para permitir a verificação formal com mais de um veículo em funcionamento.

Considerando que os Grupos Moto Propulsores funcionam de forma sincronizada nas mudanças de jurisdição dos veículos, trocando informações com os demais GMP, no caso do uso do sistema *Push-Pull* (onde há necessidade de sincronia entre 2 GMP para controlar o mesmo veículo), em hipótese, esta configuração não traria impacto em termos de arquitetura do sistema, pois a sincronização entre os GMP já está prevista na configuração proposta.



(a) Configuração 1 (b) Configuração 2

(c) Configuração 3 (d) Configuração 4

Figura 4.3 Possíveis configurações do sistema propulsor

Uma das dificuldades encontradas no trabalho com os Grupo Moto Propulsores está associada as mudanças de estados, como, por exemplo, na mudança do estado *push* para o estado de *pull*, pois, devido ao fato de o modelo de automação ser baseado na norma IEC 61850, os controladores devem ser distribuídos. Assim, foi determinado que cada válvula deve possuir um controlador e os comandos de mudança de estado para as válvulas devem ser enviados ao mesmo tempo para todos os equipamentos envolvidos, tendo como benefício uma mudança não sequencial e, por sua vez, maior eficiência do conjunto de válvulas.

Porém, durante este processo de mudanças em paralelo (mensagem do tipo *broadcast*) de estados das válvulas, o sistema, dependendo de atrasos no barramento de comunicação, pode assumir configurações diferentes dos três estados mencionados anteriormente, podendo assim ocorrer comportamentos indesejados e problemas para os passageiros e para os equipamentos.

Procurando evitar uma mudança sequencial do conjunto de válvulas, o que implicaria em maior quantidade de tempo para a conclusão do comando, visto que a válvula seguinte somente mudaria de estado após a válvula anterior terminar seu movimento, foi proposto a inclusão de um estado denominado *offline*, onde o GMP não exerce influencia no movimento do veículo, permanecendo "desconectado" do duto onde está localizado o veículo, independentemente do estado do ventilador centrífugo, desde que as válvulas V_1 e V_4 permaneçam fechadas enquanto que as válvulas V_0 e V_5 permanecem abertas.

Nesta configuração, independente do estado do restante das demais válvulas do GMP, o veículo não é influenciado pelo GMP. Este estado é utilizado como intermediário na mudança do GMP entre os estados de *PUSH* e *PULL* ou quando o veículo permanece no estado estacionado na estação.

4.2.1.1 Modelos

Visando o desenvolvimento de modelos fidedignos, que representem o comportamento do sistema real sem comprometer a verificação formal por escassez de recursos computacionais e com excessivo tempo de processamento, buscou-se simplificar, quando possível, os modelos sem comprometer seu funcionamento adequado.

Neste caso, buscou-se analisar o GMP (motor e conjunto de válvulas) de forma a verificar formalmente se, o sistema de propulsão, quando aplicado com os demais elementos do sistema geral (como veículos e equipamentos de comunicação de dados) possa ser simplificado na forma de um modelo com somente três diferentes estados: *OFFLINE*, *PUSH* e *PULL*. Neste caso, o modelo simplificado poderia representar os comportamentos das válvulas, motor e seus respectivos controladores.

Como já foi comentado, os sistemas de controle de composições são usualmente centralizados. Porém, buscando aplicar uma solução baseada na norma IEC 61850 [Hewings, 2008], os modelos foram desenvolvidos baseados em controle distribuído. Tendo sido, dessa forma, modelados com controladores dedicados contemplando requisitos de tempo real para cada dispositivo. As unidades estão conectadas a um barramento de comunicação que provê a troca de informação com outras unidades (este barramento de comunicação está detalhado no Capítulo 5). A decisão de usar um sistema de controle distribuído é motivada pela redução de custo e aumento da flexibilidade no controle. Neste caso, em especial, pela grande distância entre os elementos dos sistema.

Os modelos dos dispositivos e controladores foram desenvolvidos utilizando autômatos

temporizados e analisados através da ferramenta para simulação e verificação formal UPPAAL. Os modelos foram divididos nos seguintes moldes:

- *Valvs_Control*. Representa os controladores das válvulas *on-off*, havendo um controlador para cada uma das oito válvulas (ver Figura 4.5).
- *Valvs*. Representa o sistema físico das válvulas *on-off* do sistema de propulsão e possuem 4 diferentes estados (*closed*, *closing*, *open* e *opening*). O estado inicial depende da posição da válvula e do estado geral do GMP. O tempo necessário para mudança de estado é fixa e o sistema é iniciado com todas as válvulas em estados pré-determinados. Este modelo é repetido para cada uma das oitos válvulas *on-off* presentes no GMP (ver Figura 4.4).
- *Valvs_Prop_Control*. Representa o controlador das válvulas proporcionais e é repetida para cada uma das duas válvulas proporcionais do sistema GMP (Figura 4.7).
- *Valvs_Prop*. Representa o modelo dos estados físicos das válvulas proporcionais, possuindo dois diferentes estados (*moving* ou *stationary*). O tempo para mudança é proporcional à quantidade de movimento necessária. Este modelo é repetido para cada uma das duas válvulas proporcionais existentes no GMP (ver Figura 4.6).
- *GMP_Control*. Representa o controlador geral do GMP. Este modelo é único para o bloco padrão (Figura 4.8) e é responsável por receber mensagens de outros controladores de GMP e veículos e reenviá-los para os controladores das válvulas localizadas internamento no GMP. Este controlador é essencial para realizar a simplificação de estados no sistema. Na verificação formal do sistema ampliado, incluindo veículos e o barramento de comunicação, este modelo, uma vez comprovado que independente dos comandos recebidos, apresentará somente um entre os três possíveis estados (*PUSH*, *PULL* e *OFFLINE*). Este modelo que será utilizado para realizar simulações durante o tempo necessário para a mudança de estado das válvulas e controladores internos ao GMP.
- *Motor*. Representa o controlador do motor e é modelado com três diferentes estados (Figura 4.10). O tempo necessário para a mudança entre os estados é fixo e é assumida a hipótese de que o ventilador trabalha em estado estacionário. Este molde é único para o bloco padrão.
- *Random*. Este modelo produz comandos para mudança de estado do GMP sem sequência pré-determinada, visando a testar o controlador para as diversas possibilidades de sequências de comandos, porém ocorrendo em intervalos de tempo fixos necessários

para a mudança de estado do GMP (ver Figura 4.9).

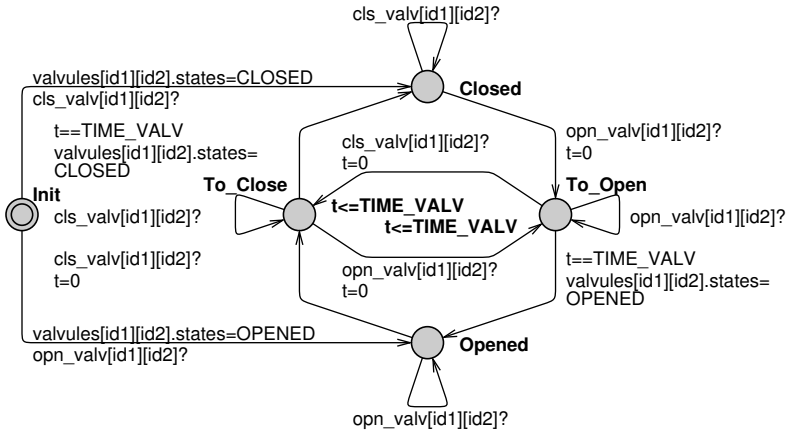


Figura 4.4 Modelo representativo das válvulas ON/OFF

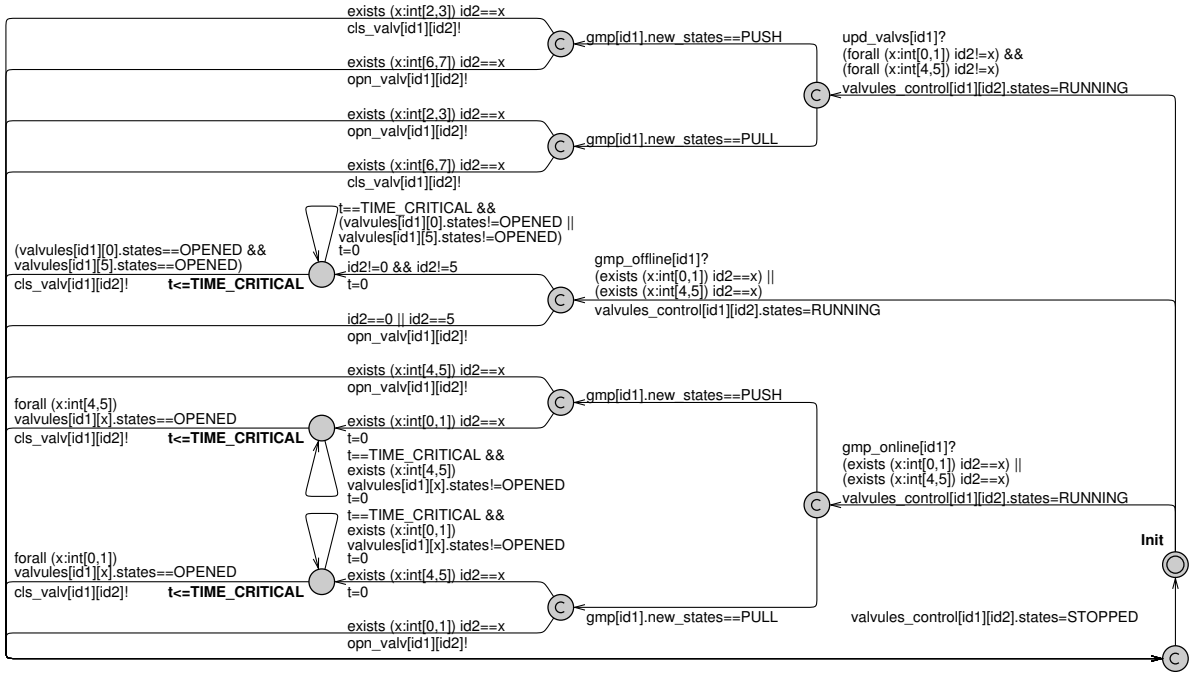


Figura 4.5 Modelo representativo do controlador das válvulas ON/OFF

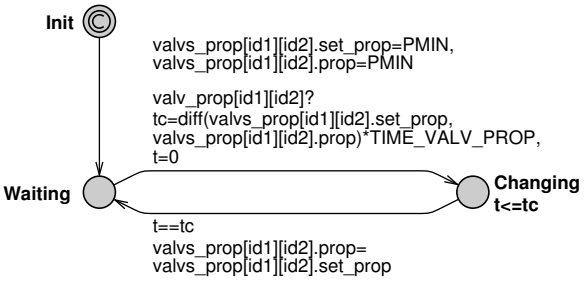


Figura 4.6 Modelo representativo das válvulas proporcionais

Os modelos que representam o funcionamento físico dos equipamentos (*Motor*, *Valv* e *Valv_Prop*) foram desenvolvidos de forma a permitir movimentos, sem restrições. Enquanto

que os modelos dos controladores (*GMP_Control*, *Valv_Control* e *Valv_Prop_Control*) são os responsáveis pelas restrições de movimento dos sistemas físicos, prevenindo o ocorrência de comportamentos não desejados. Um total de 23 modelos foram simulados e verificados formalmente.

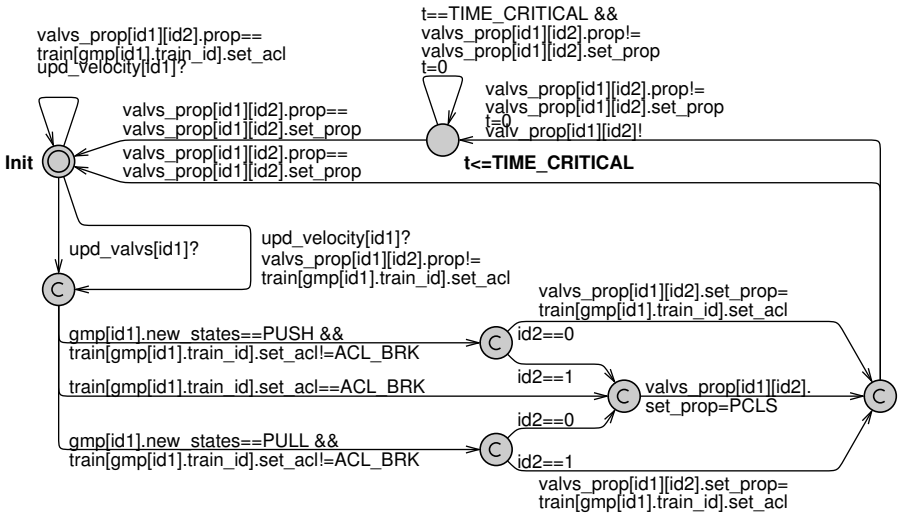


Figura 4.7 Modelo representativo do controlador das válvulas proporcionais

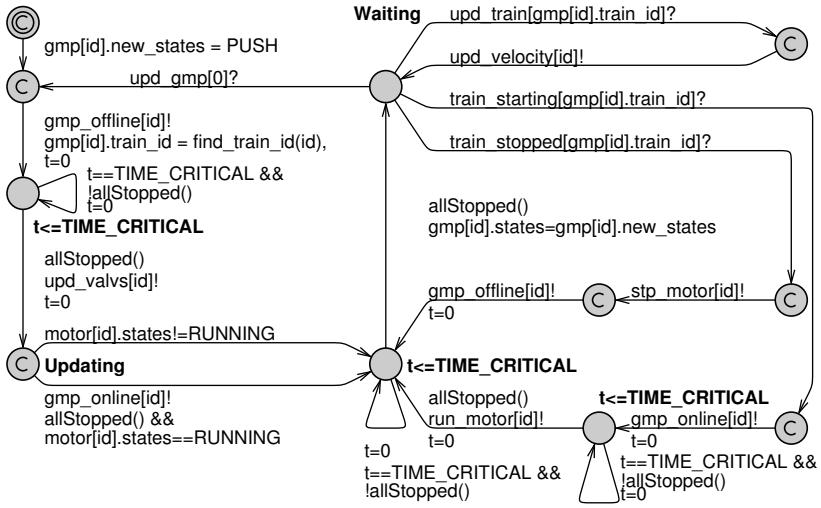


Figura 4.8 Modelo representativo do controlador do GMP

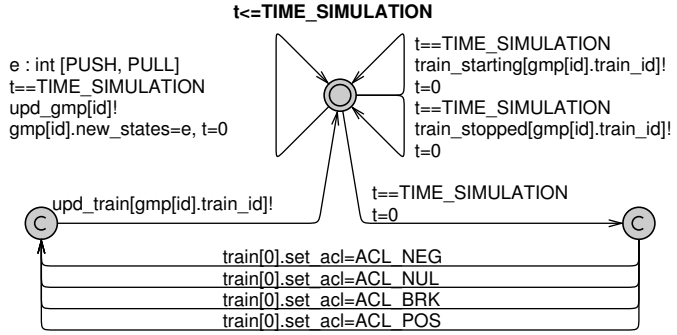


Figura 4.9 Modelo para simulação dos comandos de entrado do GMP

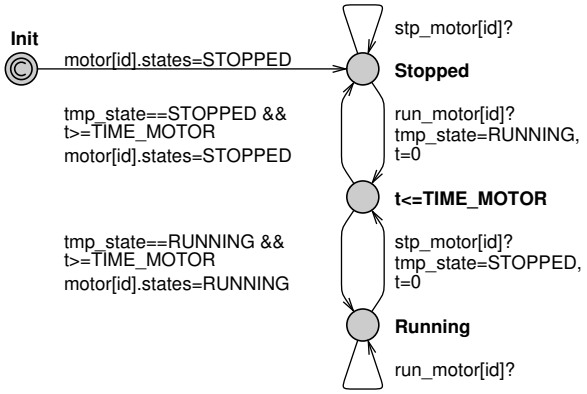


Figura 4.10 Modelo representativo do motor

4.2.1.2 Simulação

Através dos registros de simulação (arquivo no formato UPPAAL tipo *XTR*) obteve-se o diagrama apresentado na Figura 4.11. Uma rotina computacional foi desenvolvida para converter arquivos no formato *XTR* para o formato *CSV - Comma-Separated Values*. O diagrama da Figura 4.11 ilustra o comportamento de todas as válvulas do GMP quando ocorre mudanças entre os estados: *OFF*, *PUSH* e *PULL*.

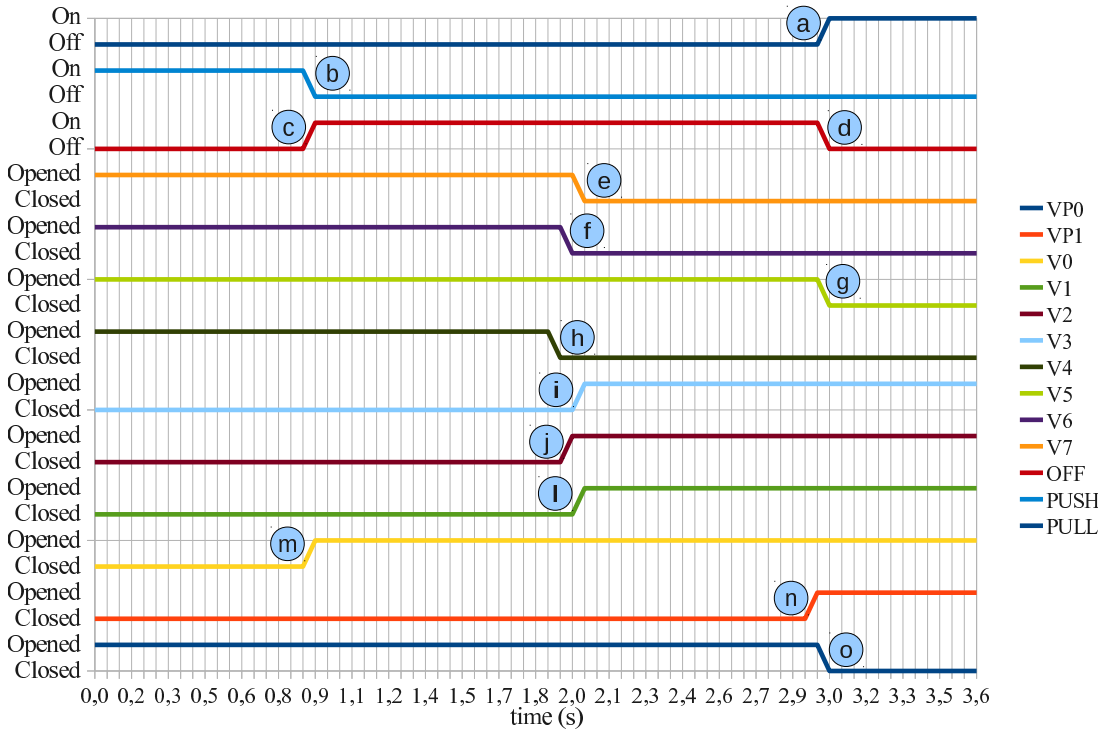


Figura 4.11 Resultado de simulação do GMP

Na Figura 4.11, VP_0 e VP_1 representam as válvulas proporcionais pneumáticas aparecendo ou totalmente abertas ou totalmente fechadas. Nessa figura, V_0, V_1, V_2, V_3, V_4, V_5, V_6 e V_7 são válvulas *on-off* e *OFF*, *PUSH* e *PULL* representam os estados *OFFLINE*,

PUSH e *PULL* do GMP, respectivamente, do conjunto de válvulas e do motor do ventilador centrífugo. O motor elétrico de acionamento do ventilador não está representado na Figura 4.11, pois supõe-se que ele está sempre em funcionamento e em estado estacionário durante a análise.

Através da análise da Figura 4.11 pode ser observado que o sistema está inicialmente no estado *PULL*. Quando ocorre a mudança de estado da válvula V_0, de *closed* para *opened* (indicado pela mudança "m", na Figura 4.11), a válvula V_4 muda do estado de *opened* para *closed* (identificado pela mudança "h", na Figura 4.11). Em paralelo com a mudança da válvula V_4, começa a mudança de configuração para o estado *PUSH*, definido pela mudança das válvulas V_1, V_2, V_3, V_6 e V_7 (identificados, na Figura 4.11, respectivamente, pelas letras "l", "j", "i", "f" e "e"). As válvulas pneumáticas proporcionais também mudam seu estado. A válvula proporcional VP_0 muda do estado *opened* para o estado *closed* (identificado pela letra "o", na Figura 4.11) e a válvula proporcional VP_17 muda do estado *closed* para o estado *opened* (identificado pela letra "n", na Figura 4.11). Uma vez que o sistema é reconfigurado, a válvula V_5 muda do estado *opened* para o estado *closed* (identificado pela letra "g", na Figura 4.11). Assim, o sistema de propulsão passa para o estado *PUSH* (identificado pela mudança "a", na Figura 4.11).

Os resultados da simulação demonstram que o sistema apresenta comportamento de acordo com o esperado, executando as ações na sequência apropriada. O próximo passo consiste na realização da verificação formal a fim de avaliar mais profundamente o comportamento do sistema de propulsão.

4.2.1.3 Verificação

Em relação a tarefas de verificação formal, foram identificados comportamentos esperados do sistema de propulsão do APM. Os modos de funcionamento foram descritos em linguagem natural e formal utilizando a linguagem de entrada do verificador de modelos UPPAAL (ver Tabela 4.1). Para a dedução das propriedades foi utilizada uma ferramenta descrita em [Campos e Machado, 2009].

Todas as propriedades observadas na Tabela 4.1, foram verificadas utilizando representação do espaço de estados do tipo *Difference Bound Matrices* (DBM) em um PC Intel® Core™ 2 Duo CPU 2.10 GHz (4 Gb RAM) em cerca de 250 minutos.

Como será visto a seguir, o uso de forma complementar das técnicas de simulação e verificação formal mostrou-se efetivo na análise da operação de uma parte dos controladores distribuídos do sistema Aeromovel. O comportamento do sistema de propulsão foi verificado e

Tabela 4.1 Propriedades comportamentais do sistema de propulsão

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
O sistema GMP deve atingir sempre os estados *PUSH* ou *PULL*	`E<>((forall (x:int[0,3]) Valvs(0,x).Closed) && (forall (x:int[4,7]) Valvs(0,x).Opened) && valvs_prop[0][1].set_prop == PCLS) ||((forall (x:int[0,3]) Valvs(0,x).Opened) && (forall (x:int[4,7]) Valvs(0,x).Closed) && valvs_prop[0][0].set_prop == PCLS)`	Sim
Se o ventilador do sistema de propulsão estiver em funcionamento e se o controlador do sistema de propulsão não estiver processando informações e se o sistema não está na situação de *OFFLINE* então o sistema de propulsão deve estar, necessariamente ou no estado de *PUSH* ou no estado de *PULL*	`A[] (Motor(0).Running and GMP_Control(0).Waiting and !Valvs(0,0).Opened and !Valvs(0,5).Opened) imply (((forall (x:int[0,3]) Valvs(0,x).Closed) && (forall (x:int[4,7]) Valvs(0,x).Opened) && valvs_prop[0][1].set_prop == PCLS) ||((forall (x:int[0,3]) Valvs(0,x).Opened) && (forall (x:int[4,7]) Valvs(0,x).Closed) && valvs_prop[0][0].set_prop == PCLS))`	Sim
As válvulas V_1 e V_5 não devem estar fechadas simultaneamente	`A[] not (Valvs(0,1).Closed and Valvs(0,5).Closed)`	Sim
As válvulas V_0 e V_4 não devem estar fechadas simultaneamente	`A[] not (Valvs(0,0).Closed and Valvs(0,4).Closed)`	Sim
Se o ventilador estiver em funcionamento e se o controlador do sistema de propulsão estiver processando informações então o sistema de propulsão deve estar necessariamente no estado de *OFFLINE*	`A[] (Motor(0).Running and GMP_Control(0).Updating) imply (Valvs(0,0).Opened and Valvs(0,5).Opened and (Valvs(0,4).Closed ||Valvs(0,4).To_Close) and (Valvs(0,1).Closed ||Valvs(0,1).To_Close))`	Sim
O sistema nunca deve atingir a situação de *deadlock*	`A[] not deadlock`	Sim

os resultados indicam que estão de acordo com as especificações de operação dos sistemas.

Assim, a partir do estudo de parte dos controladores do veículo foi possível propor uma simplificação do comportamento do sistema de propulsão quando da análise do sistema completo (incluindo veículo e barramento de comunicação) situação na qual as grandes

necessidades de recursos de tempo e capacidade de processamento computacional tenderiam a inviabilizar a pesquisa.

4.2.2 Sistema de Operação

Como visto anteriormente, um ATC deve incluir obrigatoriamente o subsistema ATP e, opcionalmente, pode incluir o ATO ou o ATS. Nesta seção serão apresentados os modelos relativos ao sistema automático de operação (ATO) para a verificação e simulação. O sistema automático de operação do veículo (ATO) exerce funções tais como: controle de movimento, parada programada na estação e controle das portas e tempo de parada.

Como observado na norma *IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements* [IEEE, 2004], que descreve os requisitos funcionais e performance da comunicação para controle de composições, uma das característica básica do CTBC inclui a determinação da localização do trem com alto grau de precisão e independente de sensores nos trilhos.

Para isto foi elaborado um modelo de sensor que está apresentado na Figura 4.12. Neste sensor, a leitura da posição é realizada em um tempo pré-fixado (*TIME_TRAIN_SENSOR*). A cada intervalo de tempo é enviada uma mensagem de sincronização "`upd_train[id]!`" (sendo "`id`" o identificador do veículo, ou seja, 0 para T_0 e 1 para T_1). Foi incorporado um modelo de sensor para cada veículo.

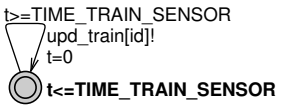


Figura 4.12 Modelo do sensor de posição do veículo

O canal de sincronização "`upd_train[id]`" é utilizado para atualizar uma rotina que converte a força pneumática aplicada ao veículo em um valor de aceleração, como pode ser visualizado na Figura 4.13. Há um modelo para cada veículo.

O modelo de cálculo da força apresenta uma versão simplificada da relação pressão *versus* aceleração, onde a relação do estado do GMP responsável pelo veículo analisado com as respectivas ações é determinado da seguinte forma:

- Empurrando: o valor da aceleração torna-se proporcional à abertura da válvula que está ativada no GMP.
- Puxando: o valor da aceleração torna-se proporcional à abertura da válvula ativada no GMP.

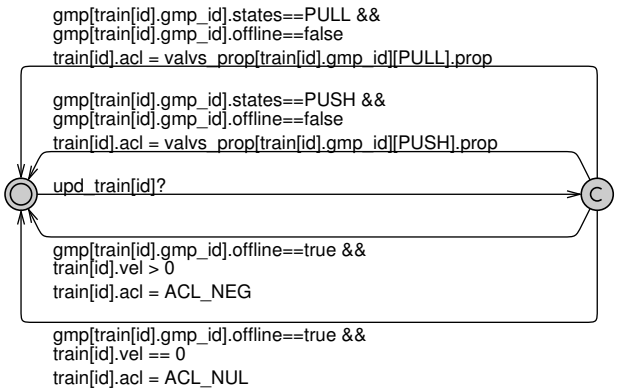


Figura 4.13 Modelo simulador da força atuante no veículo

- *Offline*: quando a velocidade do veículo é maior do que zero, a aceleração é fixada em um valor negativo diferente do valor de quando o veículo está frenando. Não há opção para o caso de velocidade menor do que zero, pois no modelo testado, uma das restrições é a de que o veículo deve mover-se em apenas uma direção (no caso do circuito circular, apenas no sentido horário).
- *Offline*: quando a velocidade do veículo é nula, a aceleração torna-se também nula.

Uma vez atualizada a aceleração, ocorre também a atualização da velocidade através do modelo apresentado na Figura 4.14. O modelo em questão é ativado através do comando "`upd_train[id]?`" e as atualizações da posição e da velocidade são realizadas através das equações 4.1 e 4.2:

$$train[id].pos += train[id].vel \tag{4.1}$$

$$train[id].vel += train[id].acl \tag{4.2}$$

Onde "`train[id].pos`", "`train[id].vel`" e "`train[id].acl`" são, respectivamente, posição, velocidade e aceleração dos veículos identificados por *id*.

Observa-se que o comando de atualização de posição é um canal do tipo *broadcast*. Desta forma, uma vez que o sinal *broadcast* é acionado pelo modelo do sensor, é recebido paralelamente pelo modelo de atualização da aceleração e pelo modelo de atualização da velocidade e da posição. O estado seguinte ao recebimento do sinal de sincronização pelo canal *broadcast* "`upd_train[id]?`" por ambos modelos das figuras 4.13 e 4.14 é indeterminado, ou seja, a ordem de atualização da aceleração ou da velocidade e posição não é determinístico.

No circuito para dois veículos e dois GMP, conforme descrito anteriormente, é necessário que o controlador dos GMP verifique qual GMP realiza o push ou o *pull* de acordo com a posição dos veículos. Para simplificação, durante o processo de definição da relação entre

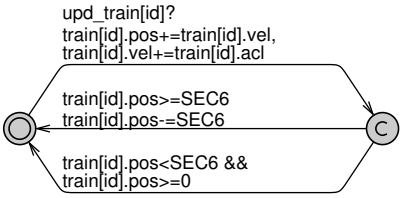


Figura 4.14 Modelo representativo da atualização da posição associado a cada veículo.

GMP e veículos, o circuito foi dividido em 6 segmentos (conforme descrito anteriormente). A associação da posição com os segmentos é expressa através do modelo apresentado na Figura 4.15. Há um modelo de relação posição e segmento para cada veículo.

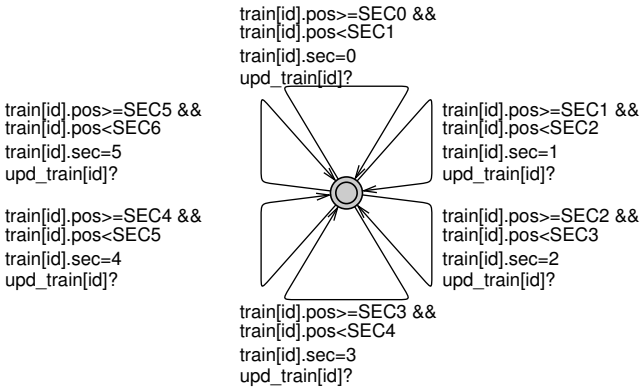


Figura 4.15 Modelo verificador da relação posição *versus* segmento

Para cada recebimento do canal de sincronização "`upd_train[id]?`", o segmento é atualizado (variável "`train[id].sec`") de acordo com a posição atual do veículo (variável "`train[id].pos`"), onde *SEC0* é o ponto inicial e *SEC6* equivale a 2200 m de comprimento. Os segmentos 0 e 3 possuem 100 m e os segmentos 1, 2, 4 e 5 possuem 500 m de comprimento.

Durante a simulação no UPPAAL, não é prático observar a posição dos veículos pois, por padrão, se é permitido visualizar a posição durante a observação das variáveis de posicionamento do veículo. Assim, auxiliar na visualização da posição do veículo foi desenvolvido o modelo apresentado na Figura 4.16. Um modelo de representação da posição foi associada a cada veículo.

Observa-se que a atualização da posição do veículo é realizada conforme o recebimento do sinal provindo do canal de sincronização *upd_train[id]?*. Este modelo não executa ações sobre os demais modelos e a verificação formal do conjunto dos modelos pode ser executada sem seu uso. Porém, durante a simulação, é muito importante para permitir a observação da posição do veículo.

O modelo em questão, assim como os demais modelos que representam sistemas físicos, deve ser desenvolvido com liberdade de movimento. No caso do modelo da Figura 4.16,

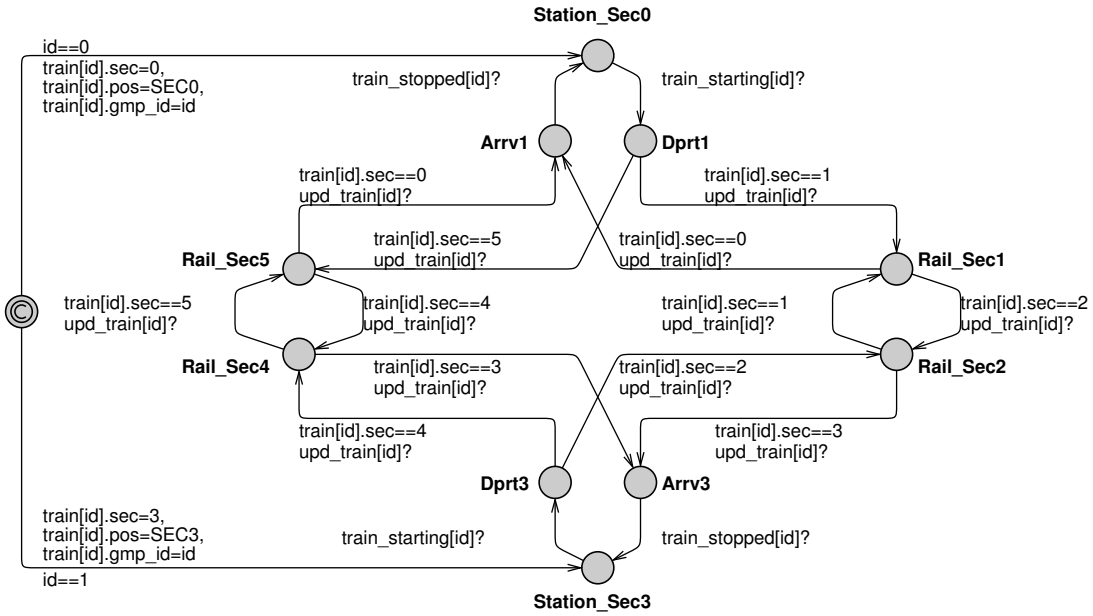


Figura 4.16 Modelo representativo da posição do veículo

observa-se que é possível que o veículo mova-se em sentido horário e anti-horário, não havendo restrições quanto ao seu movimento livre. Conforme já comentado, as restrições foram impostas nos modelos dos controladores. Este aspecto é importante, pois durante a simulação e verificação formal, quando observados, efeitos indesejados, as modificações podem ser feitas somente nos modelos dos controladores, não necessitando alterar os modelos que representam os sistemas físicos.

O modelo apresentado na Figura 4.17 é único para o conjunto de veículos e GMP. A função deste modelo é associar os GMP aos veículos para os quais são responsáveis. A figura apresenta o modelo para apenas um GMP e um veículo atuando na troca entre os estados de *push* e *pull* do GMP.

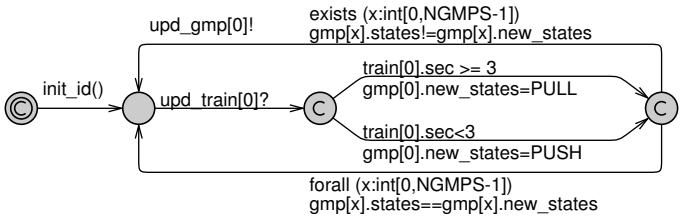


Figura 4.17 Modelo delimitador do escopo do GMP

Porém, no caso de dois GMP e dois veículos, o modelo utilizado é apresentado na Figura 4.18. Observe que este modelo é único e deve ser alterado de acordo com a quantidade veículos e sistemas de propulsão. O comando para troca do estado dos GMP é enviado de acordo com canais individuais (não-*broadcast*) para cada controlador de GMP. Os demais equipamentos, inclusive os veículos, não recebem comunicado da mudança neste nível de

controle, o que não impede de haver avisos provindos do sistema de supervisão.

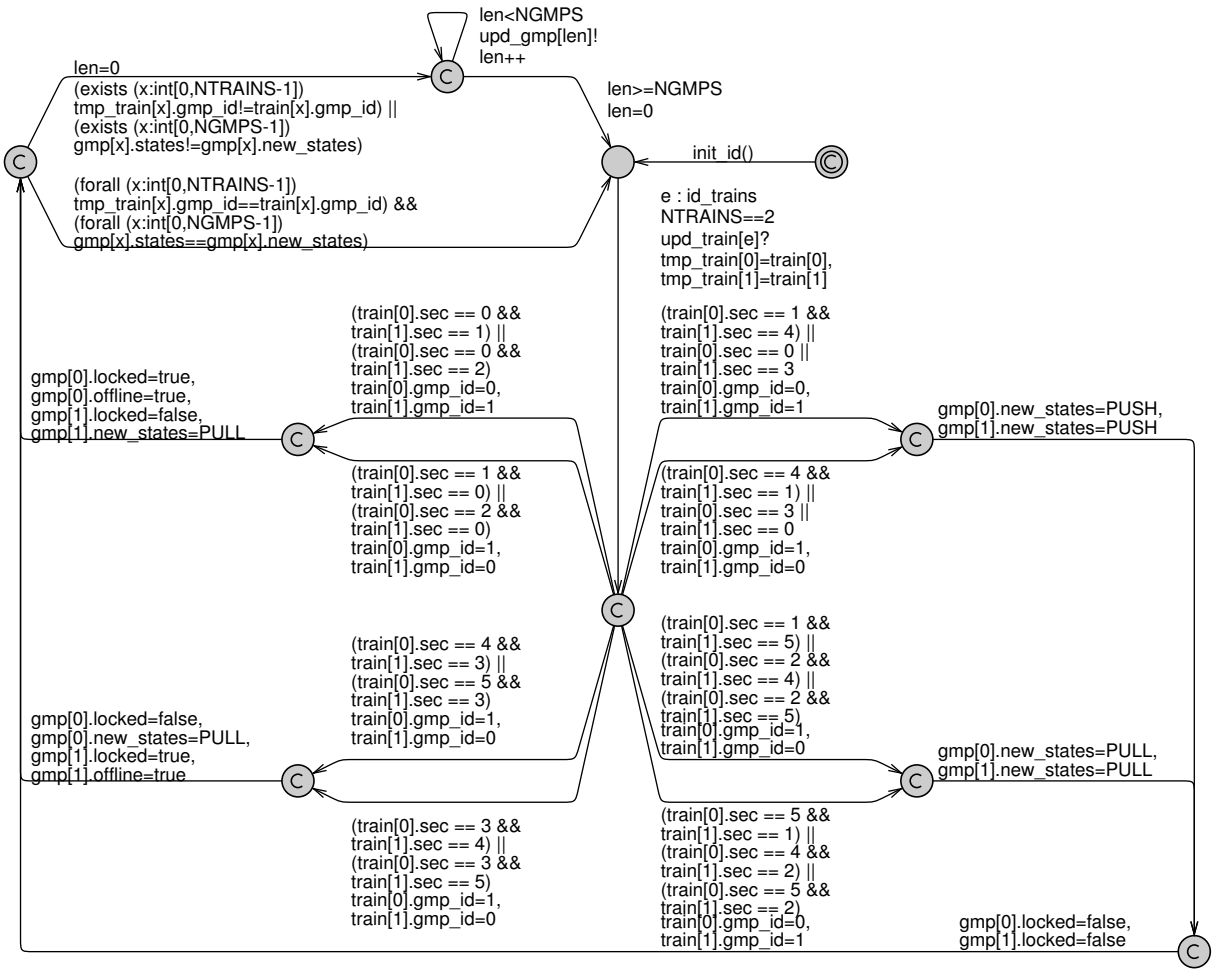


Figura 4.18 Modelo delimitador do escopo dos GMP para dois veículos

O modelo é executado a partir da mudança de posição (“`upd_train[e]?`”) onde “`e`” representa qualquer um dos veículos. O comando para atualização dos GMP é enviando somente na identificação da necessidade de alteração de configuração dos mesmos. Caso contrário, nenhum comando é enviado.

O modelo necessário para o funcionamento do veículo é o que designa seu regime de funcionamento, como pode ser observado através da Figura 4.19. Há um modelo de regime de funcionamento para cada veículo. O modelo segue as seguintes etapas de funcionamento:

1. Aguarda o recebimento do sinal “`train_starting[id]?`”, onde “`id`” é o identificador do veículo.
2. Envia o sinal cls_ports[id]! para fechamento das porta.
3. Aguarda que as portas estejam devidamente fechadas.
4. Libera os freios através do canal “`rls_brk[id]!`”.

5. Determina a aceleração do veículo através da variável "`train[id].set_acl`". Esta variável não representa diretamente a aceleração do veículo, mas sim o *setpoint* da aceleração.

6. Após a velocidade atingir o limite determinado para o trecho, modifica o *setpoint* da aceleração para zero, tornando a velocidade constante.

7. Após entrar no segmento das estações (segmentos 0 ou 3) é determinado que o *setpoint* da aceleração seja negativo e o freio seja acionado através do comando "`prs_brk[id]!`".

8. Após atingir velocidade inferior à velocidade mínima e os freios estejam acionados, é enviado o comando *broadcast* "`train_stopped[id]!`" e as portas são abertas novamente.

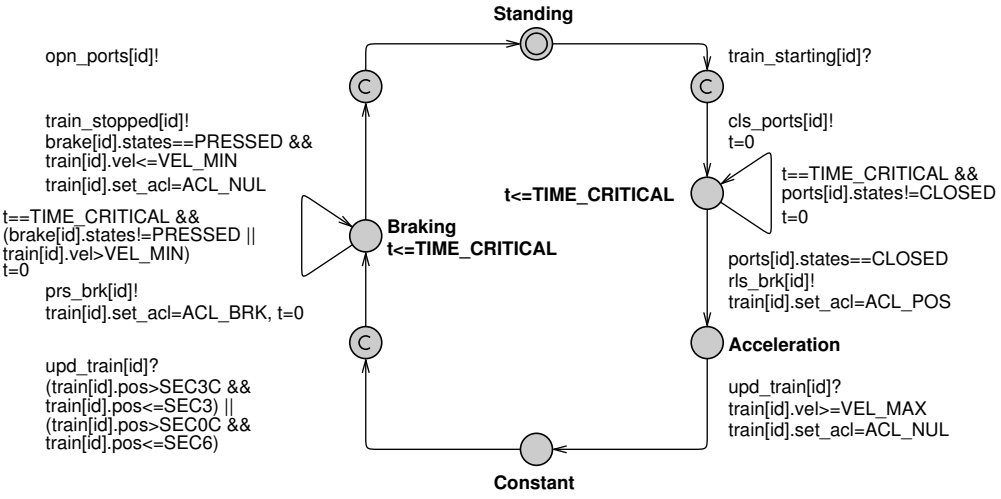


Figura 4.19 Modelo do regime de funcionamento do veículo

O sinal que indica o momento de partida do veículo é emitido pelo modelo apresentado na Figura 4.20. A partida do veículo é definida por um período constante.

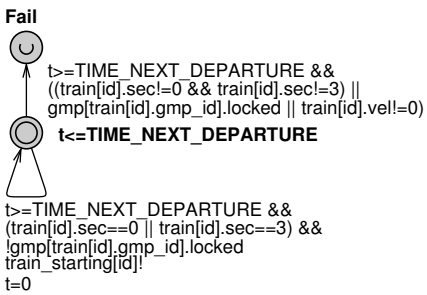


Figura 4.20 Modelo indicador de partida do veículo

Os modelos dos freios e das portas estão representados, respectivamente, nas figuras 4.21 e 4.22. Ambos são representados através de três estados, com possibilidade de

inversão antes da conclusão dos comandos.

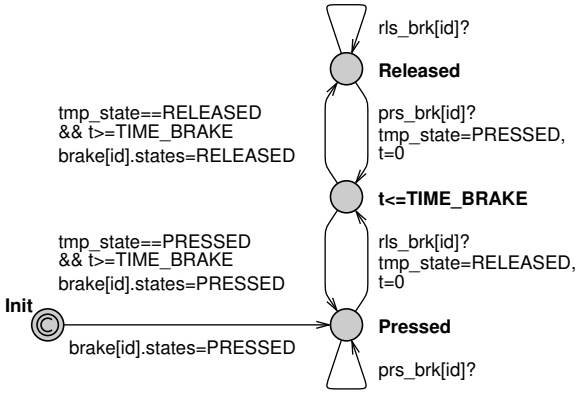


Figura 4.21 Modelo do sistema de frenagem do veículo

O modelo de frenagem possui os estados *pressionado*, *liberado* e em *mudança*. Para cada acionamento do sistema de frenagem (liberar ou pressionar), foi determinado um tempo fixo para os estados acionado ou liberado, definido respectivamente, pelos comandos de sincronização "`prs_brk[id]?`" e "`rls_brk[id]?`". O sistema de freio inicia sempre no estado acionado.

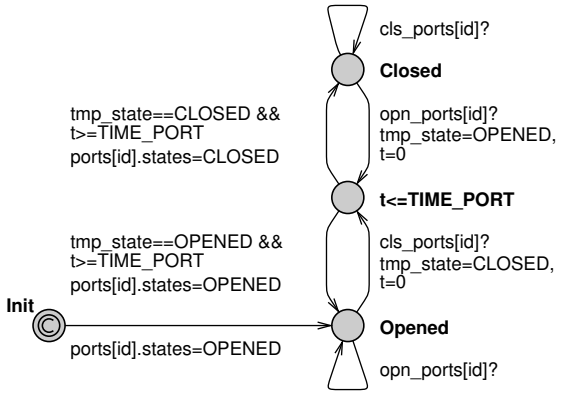


Figura 4.22 Modelo do sistema de portas do veículo

O modelo de controle de portas apresenta os estados aberto, fechado e em mudança de estados. Para cada acionamento do sistema de controle de portas (abrir ou fechar), foi determinado um tempo fixo associado, respectivamente, aos comandos "`opn_ports[id]?`" e "`cls_ports[id]?`". O sistema de controle de portas inicia-se sempre no estado aberto.

Uma vez ocorrida a mudança de estado, seja do sistema de frenagem ou de controle das portas, as variáveis "`ports[id].states`" (para as portas) e "`brake[id].states`" (para o sistema de frenagem) são modificadas, porém nenhum sinal de sincronização é emitido.

Lembrando que a validação da simplificação de estados do GMP foi apresnetada na

seção 4.2.1 um modelo simplificado do GMP foi utilizado, com o objetivo de evitar aumento excessivo de estados e diminuir o tempo de processamento durante a verificação formal. Este modelo apenas registra as solicitações para mudanças entre os estados *push*, *pull* e *offline*, assim como as mudanças de velocidade. Os dados registrados são utilizados na simulação realizada para análise do comportamento e na verificação formal. Com isso reduz-se de um conjunto inicial de 10 válvulas e 1 controlador para apenas um controlador resultando em um total de 4 estados, como pode ser observado na Figura 4.23.

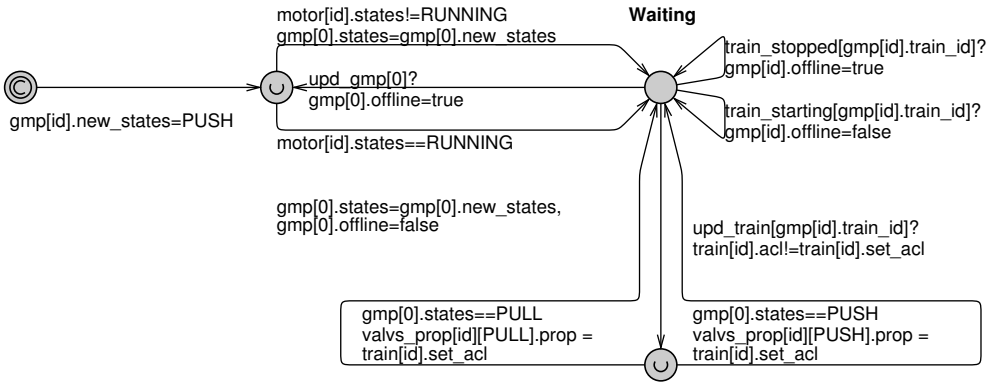


Figura 4.23 Modelo do GMP simplificado

4.2.2.1 Simulação e Verificação Formal

Para a avaliação de que o funcionamento do sistema esteja de acordo com as especificações foi realizada a simulação com o uso do software UPPAAL. Para apresentar o resultado das simulações em um formato gráfico foi necessário o desenvolvimento de um *software* específico. Um exemplo da sua aplicação está apresentada na Figura 4.24, onde OFF é o estado *offline*, GMP é representado através dos estados "`PUSH=0`" e "`PULL=1`", PRT representa o estado das portas (sendo 0 fechada e 1 aberta) e BRK os freios (sendo 0 liberado e 1 acionado).

A seguir são apresentados os resultados das simulações para as variáveis posição (Figura 4.25), velocidade (Figura 4.26), aceleração (Figura 4.27) e segmento de via (Figura 4.28), onde o veículo completa uma volta com duas paradas, uma em cada estação (segmentos 0 e 3). Os resultados da simulação indicam que o veículo apresenta possui comportamento adequado às especificações.

A verificação formal foi realizada para a identificação dos travamentos através da utilização do comando "`A[] not deadlock`". Para a verificação dos modelos que contém apenas um veículo para um GMP foram necessários aproximadamente 8 s para processamento,

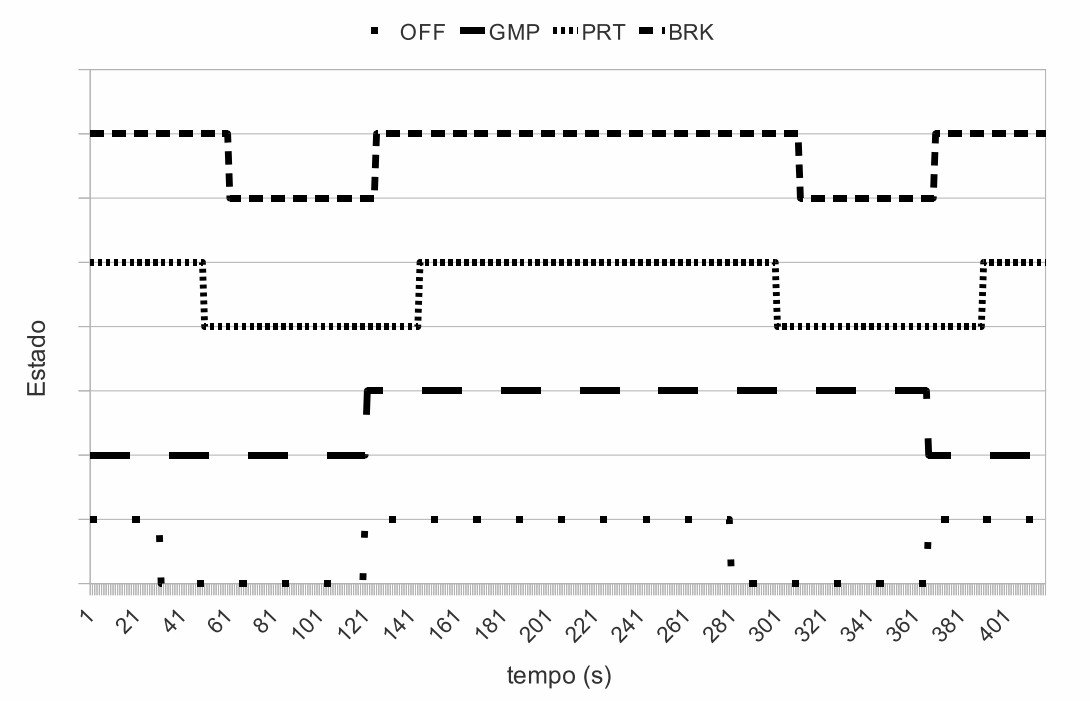


Figura 4.24 Resultado da posição

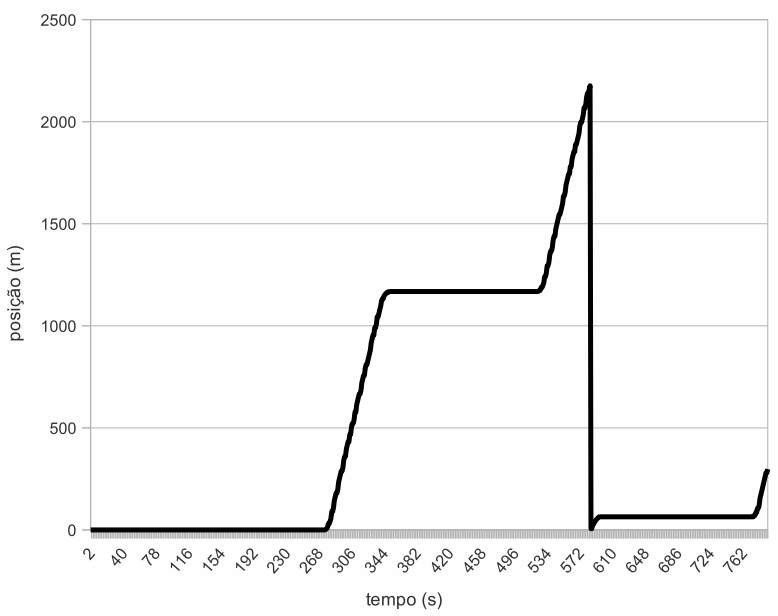


Figura 4.25 Resultado da simulação: posição

enquanto que, para dois veículos com dois GMP foi necessário aproximadamente 23 s. Observa-se que sem a simplificação do modelo do GMP (através das simplificações apresentadas nas Seção 4.2.1) não foi possível verificar os modelos pois o limite de memória RAM disponibilizado pelo UPPAAL (4 Gb) foi excedido.

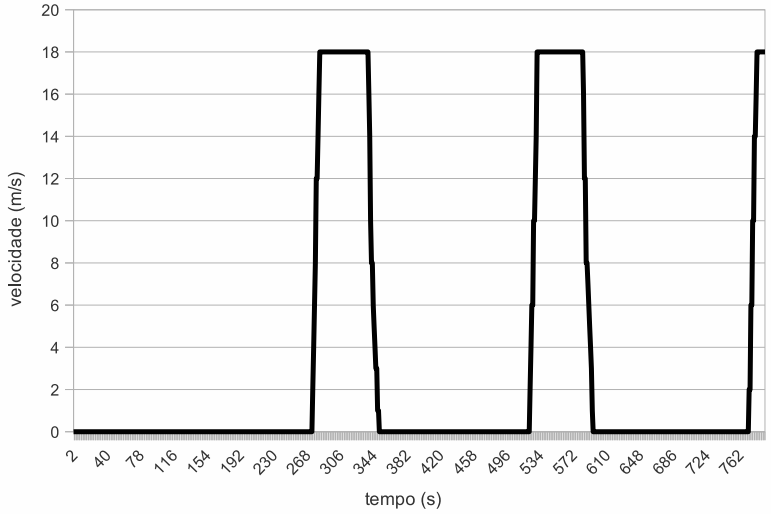


Figura 4.26 Resultado da simulação: velocidade

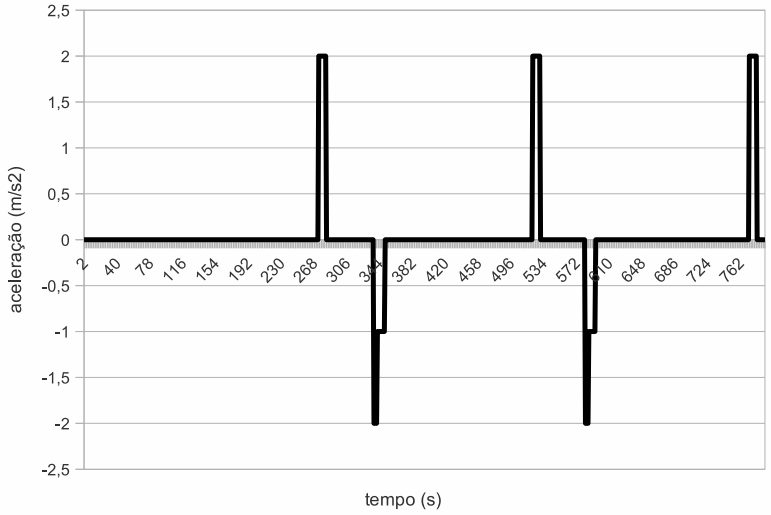


Figura 4.27 Resultado da simulação: aceleração

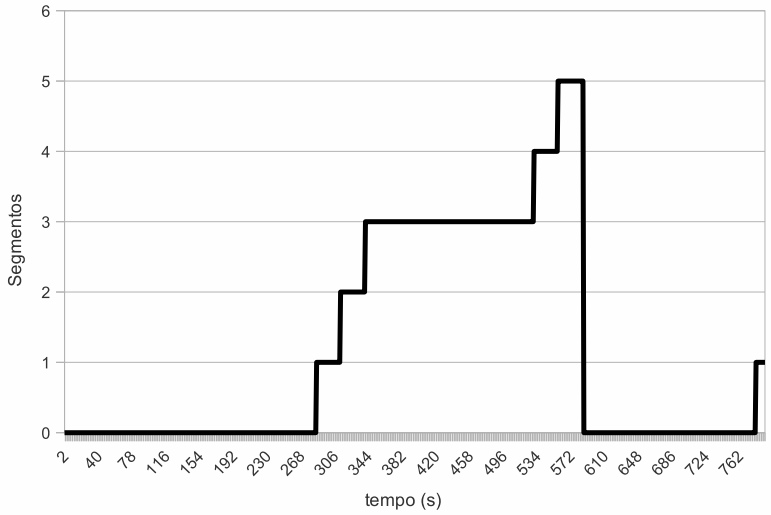


Figura 4.28 Resultado da simulação: segmento

5 MODELOS IEC 61850

Neste capítulo é detalhado o procedimento da modelagem em autômatos temporizados da comunicação de dados entre os equipamentos de controle dos sistemas ATP e ATO. Além disso, através da execução da verificação formal, utilizando o modelo proposto, é demonstrado que as implementações dos protocolos GOOSE e SMV estão de acordo com a norma IEC 61850.

O sistema de controle de composições é usualmente centralizado, porém, como o presente trabalho se propõe uma solução baseada na norma IEC 61850 [Hewings, 2008], os modelos foram desenvolvidos utilizando conceitos de controle distribuído. Dessa forma, foram modelados controladores dedicados com requisitos de tempo real para cada dispositivo. As unidades são conectadas a um barramento de comunicação que provê a troca de informação com outras unidades (este barramento de comunicação será detalhado no Capítulo 5). A decisão de usar um sistema de controle distribuído é motivada pela redução de custos e aumento da flexibilidade no controle. Neste caso, em especial, também pela grande distância entre os elementos dos sistema.

Todos os modelos apresentados neste capítulo foram desenvolvidos utilizando formalismo de autômatos temporizados com as simulações e verificação formal realizadas através do *software* UPPAAL.

Conforme apresentado no Capítulo 2, a norma IEC 61850 indica que deve haver um nó lógico para cada funcionalidade ou equipamento no sistema. Um nó lógico consiste de uma representação virtual de um equipamento. Desta forma é estabelecido um padrão de comunicação entre equipamentos que independe do fabricante dos mesmos.

Para exemplificar o funcionamento dos nós lógicos realizou-se um estudo de caso considerando o controle de frenagem do veículo. Para tanto, a relação entre nós lógicos com as mensagens GOOSE e os controladores de equipamentos foi explicitada, o que pode ser observado através da Figura 5.1.

Neste caso, se o controlador de frenagem necessita, por exemplo, mudar o estado da válvula de segurança, um comando (pacote de dados no formato GOOSE) é enviado para o nó lógico do controlador da válvula de segurança através do emissor (*publisher*) de mensagens GOOSE.

Esta mensagem é enviada para o barramento de dados, que, de acordo com os atrasos

intrínsecos envolvidos na transmissão de informação, será recebido pelo receptor (*subscriber*) de mensagens GOOSE associado à válvula de segurança.

O receptor de mensagens GOOSE, após verificar o pacote de dados recebido, envia a mensagem por canais de comunicação para o controlador da válvula que a aciona, alterando o seu estado de acordo com a mensagem de solicitação.

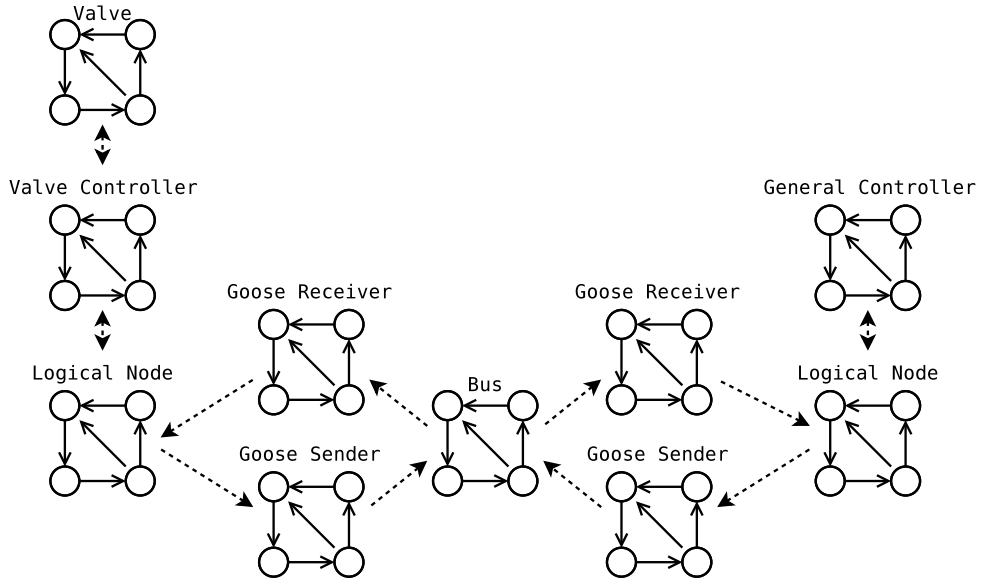


Figura 5.1 Nós lógicos integrados

É importante salientar que, para cada nó lógico, há somente um modelo emissor de mensagens GOOSE e nenhum ou vários receptores de mensagens GOOSE. Os emissores de mensagens GOOSE enviam mensagens do tipo *broadcast*, ou seja, sem destino exato. Desta forma, todos os equipamentos pertencentes à mesma rede do emissor recebem a mensagem.

Já, os receptores de mensagens GOOSE são criados para receber mensagens de emissores pré-determinados. Se, por exemplo, o controlador de frenagem necessita conhecer o estado de 10 diferentes válvulas, logo serão necessários 10 receptores de mensagens GOOSE, um para cada válvula. Esta forma de funcionamento é idêntica à aplicada às mensagens SMV.

5.1 Comportamento do Sistema de Comunicação

Para os testes de conformidade, foi proposto um ambiente hipotético que envolve os protocolos GOOSE e SMV. Este ambiente foi proposto para estudar o sistema de controle redundante da velocidade, o qual consiste de uma das partes críticas do sistema de comunicação. Este ambiente foi modelado considerando a comunicação entre três equipamentos: dois nós lógicos (modelos *LNA* e *LNC*) que enviam, periodicamente, o estado binário (GSE - GOOSE) e um sinal de amostragem (SMV) de um determinado equipamento, como, por exemplo, um

sensor de posição cujo sinal de amostragem indica a posição do veículo e o estado binário indica se os sensores estão, ou não, funcionando corretamente. O terceiro dispositivo (modelo *LNB*) recebe os valores do estado binário e do sinal de amostragem dos modelos *LNA* e *LNC*.

Para cada um dos protocolos foi inserido um modelo de barramento (modelos *BUS-VLAN0* e *BUS-VLAN1*) representando diferentes VLANs (*Virtual Lan*). Os barramentos são diferenciados, pois as mensagens dos protocolos GOOSE e SMV recebem diferentes prioridades.

Através desse ambiente foram simulados e verificados formalmente 2 modelos de barramento, 3 modelos de nós lógicos, 2 modelos de servidores GOOSE, 2 modelos de clientes GOOSE, 2 modelos de servidores SMV e 2 modelos de clientes SMV usados para estabelecer a comunicação entre os nós lógicos. Além dos modelos que representam as funções de controle, 4 modelos foram inseridos como requisito dos testes de conformidade. Consequentemente, foram simulados e verificados formalmente um total de 17 modelos.

A Figura 5.2 mostra a relação entre os 13 modelos funcionais, onde as setas indicam o fluxo e sentido da comunicação:

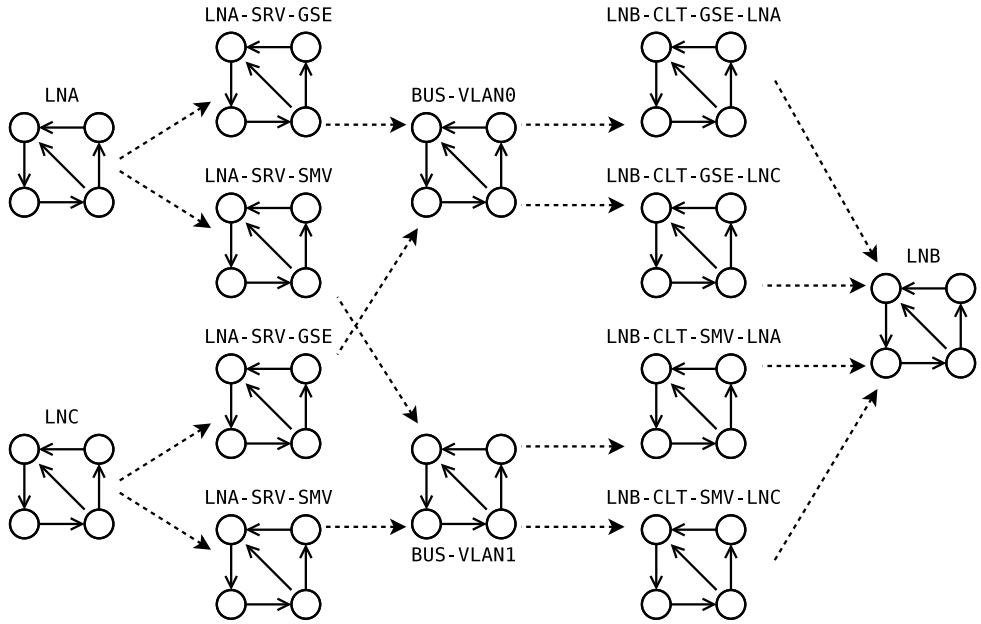


Figura 5.2 Comportamento dos nós lógicos integrados

Observa-se na Figura 5.2, que os modelos *LNA* e *LNC* são iguais e enviam sinais para o mesmo nó lógico (*LNB*) através de 2 barramentos *BUS-VLAN0* e *BUS-VLAN1*. Para isso, utilizam os modelos *LNA-SRV-GSE* e *LNC-SRV-GSE* para enviar, respectivamente, os dados binários dos nós lógicos *LNA* e *LNC* através de mensagens GOOSE para o nó lógico *LNB*. Isto é realizado através dos respectivos clientes GOOSE *LNB-CLT-GSE-LNA* e *LNB-CLT-GSE-LNC*. Para as mensagens SMV, são utilizados os modelos *LNA-SRV-SMV* e *LNC-SRV-SMV* para

enviar, respectivamente, os sinais dos nós lógicos *LNA* e *LNC* através de mensagens SMV para o nó lógico *LNB*. Isto é realizado através dos respectivos clientes SMV *LNB-CLT-SMV-LNA* e *LNB-CLT-SMV-LNC*.

A seguir, são descritos os modelos dos nós-lógicos (*Logical Nodes*), barramento de comunicação (*Communication Bus*), mensagens GOOSE (GOOSE *Messages*), mensagens SMV (*Sampled Valued Messages*) e observador para verificação de mensagens (*Check Messages Observer*).

5.1.1 Nós Lógicos

De acordo com a norma IEC 61850, a arquitetura de comunicação utilizada deve ser a *publish/subscribe*. Diferentemente da arquitetura cliente/servidor, onde um cliente solicita a informação de um determinado servidor através dos métodos *request/reply*, na arquitetura *publish/subscribe*, o método *publisher* (*producer* ou *sender*) envia as mensagens sem destino específico. O método *subscriber* (comumente referido como *consumer* ou *receiver*) somente irá processar as mensagens de interesse.

Desta forma, os modelos *LNA* e *LNC* enviam periodicamente as mensagens sem destino determinado. Como pode ser observado na Figura 5.3, os modelos de envio transmitem periodicamente mensagens SMV e assincronicamente mensagens GOOSE. Ambos os modelos *LNA* e *LNC* são baseados no modelo apresentado na Figura 5.3, alterando somente o valor *ln_id* que é a identificação do nó lógico.

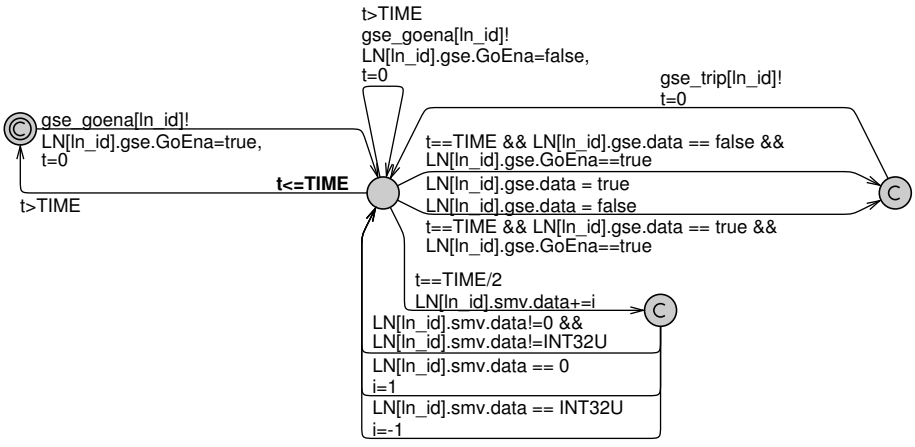


Figura 5.3 Modelo emissor

O modelo é inicializado com o temporizador igual a zero e este tempo (t) é reinicializado quando $t > TIME$. Quando $t = TIME/2$, o valor do sinal amostrado *SMV* é incrementado ou decrementado, resultando em uma onda aproximadamente triangular. Quando $t = TIME$, o

estado do sinal binário (GOOSE) é alterado e o *timer* reinicializado. Se, eventualmente, $t > TIME$, o estado de *GoEna* (que indica se o servidor GOOSE está ou não enviando mensagem) é alterado, podendo ir para o estado 0 ou 1. Se resultar em 0, o servidor cancela o envio de mensagens.

O modelo apresentado na Figura 5.4 (*LNB*) descreve o nó lógico que recebe os dados dos protocolos GOOSE e SMV de ambos os nós lógicos *publishers* (*LNA* e *LNC*).

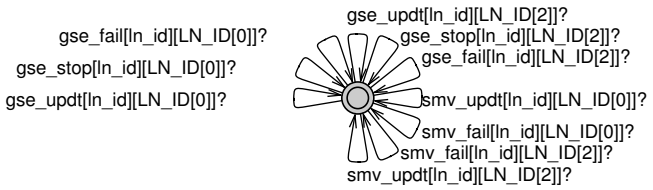


Figura 5.4 Modelo receptor

Este modelo não processa as informações recebidas, somente recebendo os valores enviados para a rede pelos modelos *LNA* e *LNC*, onde *ln_id* é o identificador do *LNB*. O modelo recebe os seguintes comandos enviados pelos *subscribers* GOOSE e SMV:

- *gse_updt* - indica a ocorrência de alguma alteração do estado do sinal GOOSE.
- *gse_stop* - indica que o *publisher*, cancela, conscientemente, o envio de mensagens.
- *gse_fail* - indica que o *publisher* deixa de enviar mensagens dentro de um intervalo de tempo estipulado, normalmente devido a falhas na rede ou no próprio *publisher*.
- *smv_fail* - indica que o *publisher* de mensagens SMV deixa de enviar mensagens dentro de um intervalo de tempo estipulado.
- *smv_updt* - indica que houve alteração do estado do sinal SMV.

5.1.2 Barramento de Comunicação

O modelo do barramento de comunicação (Figura 5.5) foi definido como uma estrutura FIFO (*First In, First Out*) considerando que, como são utilizadas VLAN (*Virtual LAN* - redes logicamente independentes), todas as mensagens terão a mesma prioridade. Mensagens com prioridades diferentes, como protocolos relativos ao sistema de supervisão e ao sistema de proteção que possuem prioridades diferenciadas, utilizarão VLANs separadas. Todo o atraso na transmissão (na rede e no processamento da comunicação) é inserido pelo modelo de barramento. Observa-se que o atraso total típico é de 4 ms (Figura 5.6).

É definido previamente se o barramento vai tratar mensagens GOOSE ou SMV. Este modelo opera de forma que, ao receber as mensagens da rede, adiciona a mensagem em uma

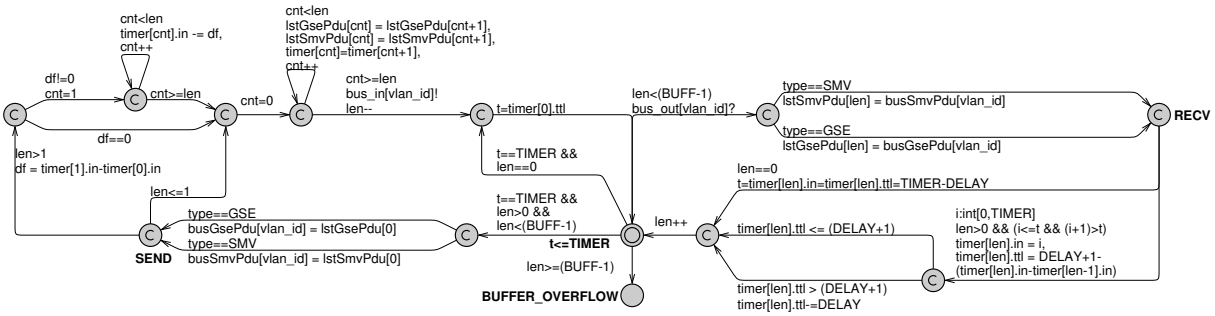


Figura 5.5 Modelo do barramento de comunicação

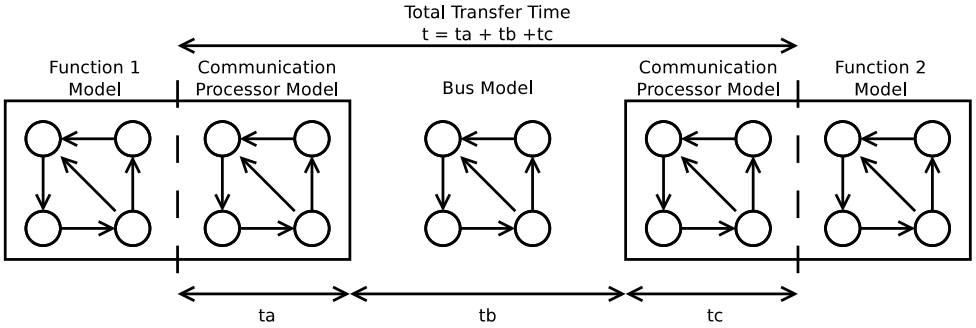


Figura 5.6 Atraso de comunicação

fila e, independentemente do momento da chegada da mensagem, aguarda 4 ms para remover a mensagem da fila e enviar novamente para a rede.

5.1.3 Mensagens GOOSE

Os modelos para comunicação GOOSE são classificados como: GOOSE Emissor, GOOSE Receptor, Barramento de Comunicação e Nós Lógicos. As seguintes características foram observadas para a implementação dos modelos de comunicação GOOSE:

- As mensagens são assíncronas e não solicitadas;
- O protocolo GOOSE é encapsulado diretamente no camada *Ethernet*. As mensagens são do tipo sem conexão, desta forma, os modelos não verificam a estabilidade da conexão, ou seja, não ocorre confirmação de recebimento dos pacotes enviados.
- As mensagens são do tipo *multicast*, assim, somente os equipamentos que se encontram na mesma VLAN (rede virtual) podem enviar ou receber pacotes e deve haver um modelo de barramento de comunicação para cada VLAN.
- Como este tipo de mensagem não possui confirmação de recebimento, o método de retransmissão pelos emissores é utilizado para aumentar a probabilidade de sucesso no recebimento das mensagens.

O emissor de mensagens GOOSE possui três diferentes estados básicos: *NON-EXISTENT*, *RETRANSMIT-PENDING* e *RETRANSMIT*. O nó lógico é configurado para enviar mensagens GOOSE somente quando *GoEna=true*. O emissor transmite o primeiro pacote de mensagem com o contador *SqNum* com valor zero. Esta variável é incrementada a cada transmissão de pacote, mas é reiniciada para o valor 1 quando atingir o limite do contador e recebe o valor zero quando a variável *StNum* é atualizada. A variável *StNum* é utilizada para definir quantas vezes o equipamento mudou de estado. Na primeira emissão, *StNum* recebe o valor 1 e *timeAllowedtoLive* o valor 2. A variável *timeAllowedtoLive* define o tempo para a próxima retransmissão. Desta forma, o receptor sabe quando deverá receber a próxima mensagem. Estas variáveis fazem parte da estrutura de dados nomeada *SendGooseMessage*. O tempo de espera para a próxima transmissão de dados (*timeAllowedtoLive*) é configurada para 2^n, com "n" iniciado em 1, sendo incrementado por 1 até atingir 1024 ms.

A Figura 5.7 ilustra o tempo de espera para a próxima retransmissão de dados. A variável *th* é o *heartbeat*, sinal que indica que o sistema está funcionando corretamente. A ausência deste sinal indica que houve falha no emissor ou no meio de transmissão de dados. O valor de *heartbeat* é de 1024 ms. Na ocorrência de um evento (mudança de estado), $t0$ é indeterminado pela natureza dos eventos e pelo fato de as mensagens de comunicação serem assíncronas, $t1$ é igual a 2^1ms, $t2$ é igual 2^2ms e assim por diante.

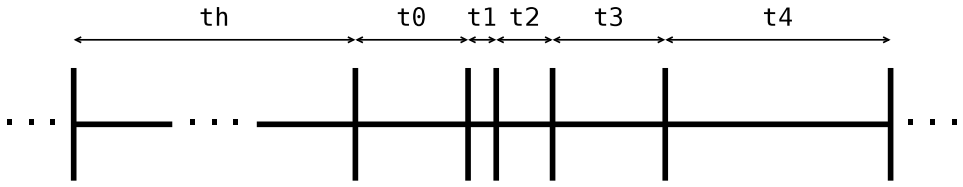


Figura 5.7 Tempo de espera para retransmissão de mensagens GOOSE

O emissor GOOSE (ver Figura 5.8), para enviar mensagem para outros nós lógicos, copia a estrutura de dados *SendGooseMessage* para a estrutura *busGsePdu* e envia um sinal por um sistema de canais de comunicação para o modelo de barramento, avisando que está disponível um novo pacote de dados. O modelo de barramento de comunicação recebe o sinal e copia a estrutura *busGsePdu* para uma pilha de mensagens e registra o momento (tempo) em que recebeu a mensagem. Após um período (simulação de atraso) de 4 ms, o modelo de barramento de comunicação remove os dados da pilha e copia os mesmos para a estrutura *busGsePdu*, enviando sinal do tipo *broadcast* para todos os receptores GOOSE que compartilham a mesma VLAN. Observa-se que este método é o mesmo utilizado para mensagens *Sample Value*, sendo que a única diferença consiste das estruturas utilizadas na pilha FIFO do barramento.

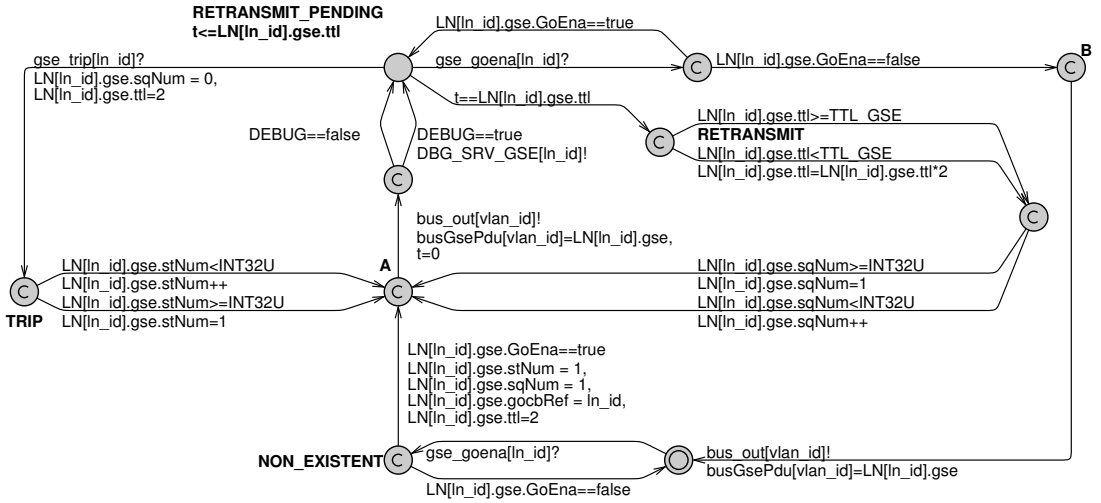


Figura 5.8 Modelo de emissor de mensagens GOOSE

O sinal chega ao receptor GOOSE (ver Figura 5.9) via canal *broadcast* e, se houver interesse na mensagem, os dados da estrutura *busGsePdu* são copiados localmente. Nem toda mensagem recebida por um receptor GOOSE é processada, pois os controladores necessitam conhecer o estado de determinados dispositivos, de acordo com sua função, mas não de todos. Quando é criado um receptor GOOSE, é determinado que o mesmo deve processar mensagens de um determinado nó lógico, havendo um receptor GOOSE para cada nó lógico de interesse. Se a mensagem recebida não é importante, a mesma é descartada e o receptor volta ao estado de espera, aguardando a próxima mensagem. Se a informação recebida é importante, então o receptor GOOSE envia mensagem para o controlador do nó lógico que pertence e volta ao estado de espera da rede.

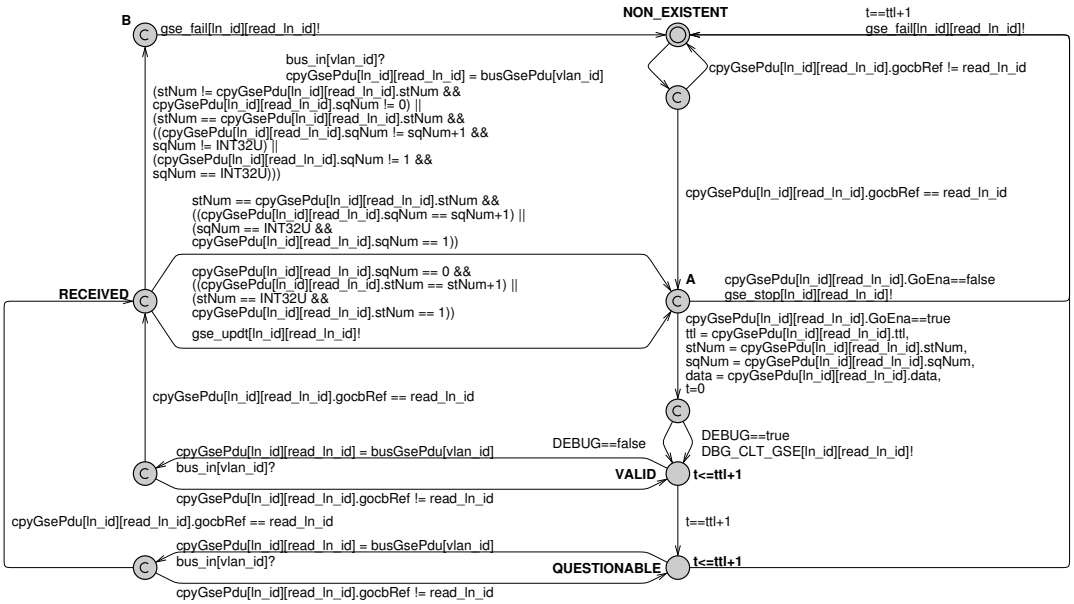


Figura 5.9 Modelo receptor GOOSE

5.1.4 Mensagens de Valores Amostrados

Como pode ser observado através da Figura 5.10, o modelo de recebimento de mensagens *Sampled Value* inicia aguardando a primeira mensagem destinada ao seu nó lógico (identificado por *ln_id*). Uma vez recebida a primeira mensagem, é verificado o intervalo de recebimento (*smpRate*), o contador da mensagem (*smpCnt*) e reinicializado o *timer* do modelo. Caso o *timer* seja maior do que a variável *smpRate* ou se for identificada uma perda de pacotes através da verificação do contador *smpCnt*, é enviada uma mensagem de falha para o nó lógico de recebimento através do comando *smv_fail*.

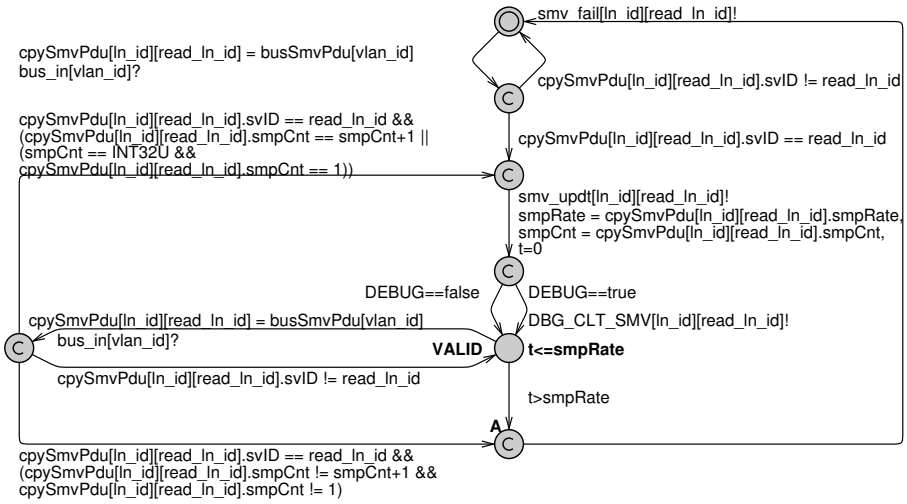


Figura 5.10 Modelo do receptor de SMV

A Figura 5.11 mostra o modelo de envio de mensagens *Sampled Value*. Este modelo envia, de acordo com uma frequência pré-configurada (TTL_SMV), o valor atualizado dos modelos *LNA* e *LNC*, independente do recebimento das mensagens pelos demais nós lógicos. O envio é feito através do método *multicast*, visto que as mensagens estão restritas aos nós lógicos da VLAN.

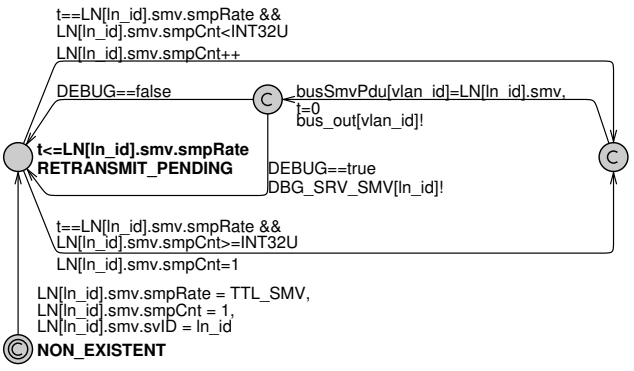


Figura 5.11 Modelo do emissor de SMV

5.1.5 Observador para Verificação de Mensagens

O modelo apresentado na Figura 5.12 foi utilizado para auxiliar na verificação formal realizando verificações através dos seguintes objetos:

1. *CHECK_LNA_LNB_GSE* é utilizado para verificação da integridade do pacote e do atraso entre o envio e o recebimento de mensagens GOOSE entre os nós lógicos LNA e LNB.

2. *CHECK_LNC_LNB_GSE* é utilizado para realizar a verificação da integridade do pacote e do atraso entre o envio e o recebimento de mensagens GOOSE entre os nós lógicos LNC e LNB.

3. *CHECK_LNA_LNB_SMV* é utilizado para realizar a verificação da integridade do pacote e do atraso entre o envio e o recebimento de mensagens *Sampled Value* entre os nós lógicos LNA e LNB.

4. *CHECK_LNC_LNB_SMV* é utilizado para realizar a verificação da integridade do pacote e do atraso entre o envio e o recebimento de mensagens *Sampled Value* entre os nós lógicos LNC e LNB.

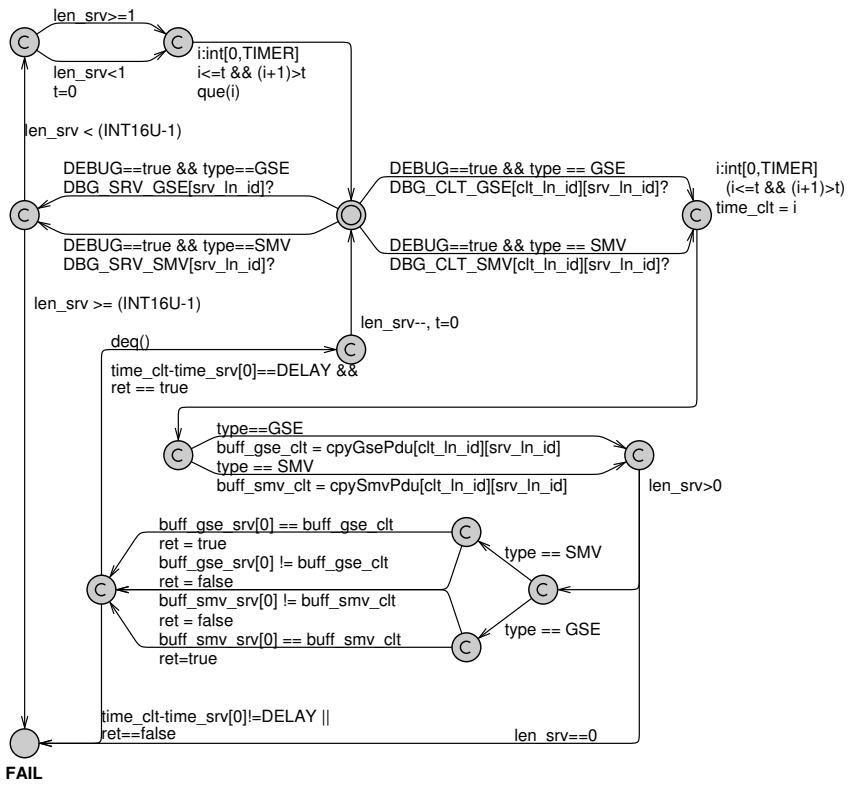


Figura 5.12 Modelos observador para verificação de mensagens

Uma vez enviada uma mensagem GOOSE ou *Sampled Value*, a mesma mensagem e o tempo referente ao envio, são copiados para uma fila do tipo *Firt In First Out* (FIFO). Quando a mensagem chega ao destino, é verificada a integridade do pacote através da comparação da mensagem copiada e é verificado se o intervalo de envio não ultrapassou os 4 ms, de acordo com os requisitos da norma IEC 61850.

5.2 Simulação

Para todos os modelos, os valores máximos dos contadores foram restringidos com o objetivo de minimizar o impacto de tempo no processamento computacional, viabilizando a simulação e a verificação formal. Por exemplo, o contador *stNum*, que por definição possui 2^{32} bytes, nos modelos foi definido com tendo 2^8 bytes. Para todos os estados dos modelos envolvidos, com exceção dos estados *committed*, é necessário determinar o intervalo de tempo para que a mudança ocorra.

Com relação à simulação, o registro dos seus resultados (arquivo no formato UPPAAL tipo *XTR*) foi utilizado para obter os gráficos apresentados nas figuras 5.13 e 5.14. Uma rotina computacional foi desenvolvida para converter dados do formato *XTR* para *CSV - Comma-Separated Values*. Através da Figura 5.13, é possível observar que nas retransmissões de mensagens há um incremento da variável *SqNum* e na mudança de configuração ocorre o incremento da variável *stNum*.

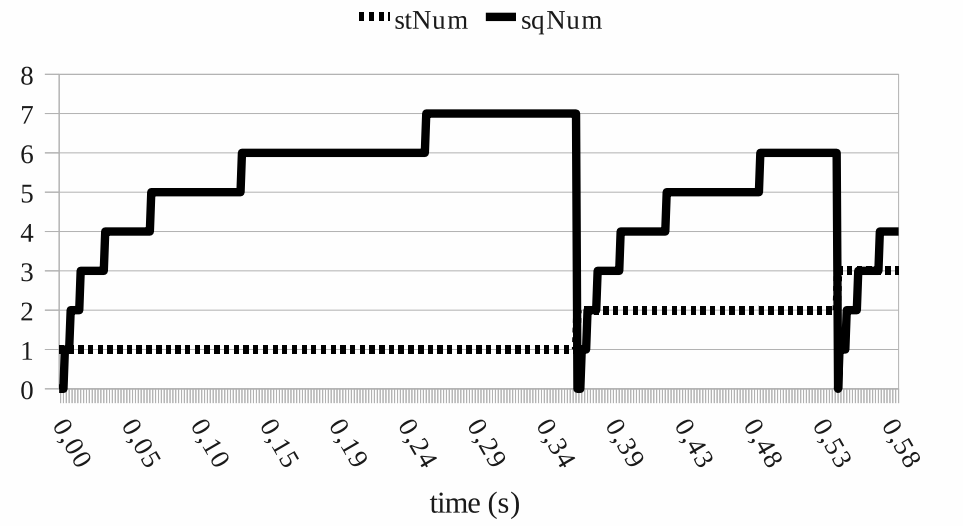


Figura 5.13 Resultados de simulação - GOOSE

Nota-se que *stNum* é incrementada em cada envio de mensagem e *SqNum* é incrementada somente quando *stNum* reinicia a contagem devido ao limite do contador ou à modificação do estado do equipamento.

Na Figura 5.14 pode-se observar que o valor de *smpCnt* é incrementado em cada envio de mensagem e que *data* é modificado de forma não determinística, simulando a aleatoriedade de um dado sinal amostrado.

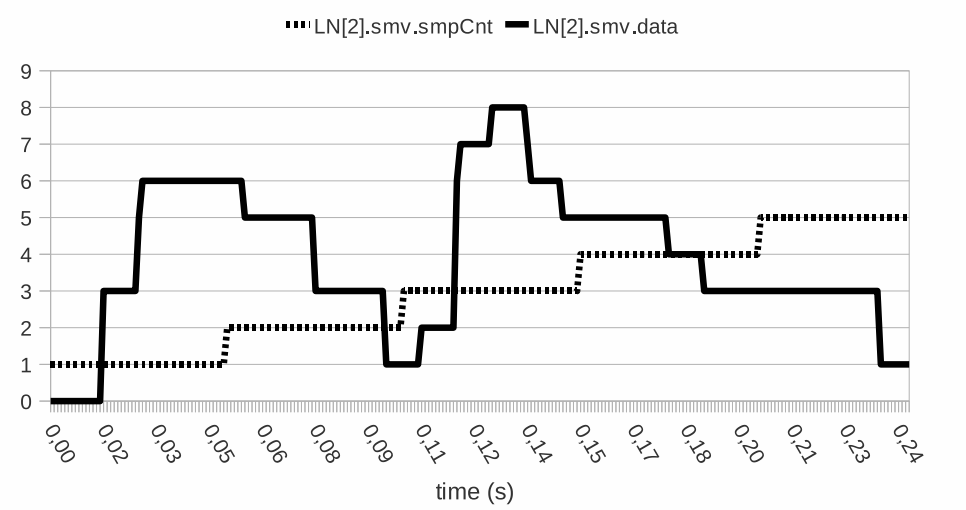


Figura 5.14 Resultados de simulação - SMV

Uma vez observado o comportamento esperado dos protocolos GOOSE e SMV durante a simulação, torna-se necessária a verificação formal, com o objetivo de avaliar o comportamento dos protocolos GOOSE e SMV nas demais possibilidades de interação entre eventos assíncronos.

5.3 Verificação Formal

No que diz respeito à verificação formal dos modelos, alguns comportamentos dos protocolos de comunicação GOOSE e SMV são esperados. Os comportamentos testados estão descritos em linguagem natural e formal, de acordo com o padrão de entrada para o verificador de modelos UPPAAL. Foi utilizada a representação em espaço de estados DBM (*Difference Bounded Matrices*) [Dill, 1989]. As verificações foram definidas a partir das informações contidas em [IEC, 2005], como pode ser observado através das tabelas 5.2 e 5.1.

Os resultados obtidos através da simulação e da verificação formal são satisfatórias e demonstram que os modelos implementados estão de acordo com a norma IEC 61850. Uma vez verificados os modelos, os mesmos são novamente utilizados em simulação e verificados formalmente em conjunto com os controladores do sistema APM.

Tabela 5.1 Verificação das propriedades do sistema de comunicação

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
Verifica o comportamento do sistema no caso de atraso no envio, duplicação de mensagens, perda de dados e pacotes fora de ordem para mensagens GOOSE	`A[] (LNB_CLT_GSE_LNA.VALID) imply LNB_CLT_GSE_LNA.t <= LNB_CLT_GSE_LNA.ttl`	Sim
	`A[] (LNB_CLT_GSE_LNC.VALID) imply LNB_CLT_GSE_LNC.t <= LNB_CLT_GSE_LNC.ttl`	Sim
	`A[] not LNB_CLT_GSE_LNC.B`	Sim
	`A[] not LNB_CLT_GSE_LNA.B`	Sim
	`A[] not LNB_CLT_GSE_LNC.QUESTIONABLE`	Sim
	`A[] not LNB_CLT_GSE_LNA.QUESTIONABLE`	Sim
Verifica o atraso no envio, duplicação de mensagens, perda de dados e pacotes fora de ordem para mensagens SMV	`A[] not LNB_CLT_SMV_LNA.A`	Sim
	`A[] not LNB_CLT_SMV_LNC.A`	Sim
Verifica a ocorrência de estouro de pilha no barramento de comunicação	`A[] not BUS_VLAN1.BUFFER_OVERFLOW`	Sim
	`A[] not BUS_VLAN0.BUFFER_OVERFLOW`	Sim
Verifica se o conteúdo das mensagens estão coerentes entre emissores e receptores para os protocolos GOOSE e SMV	`A[] not CHECK_LNC_LNB_SMV.FAIL`	Sim
	`A[] not CHECK_LNA_LNB_SMV.FAIL`	Sim
	`A[] not CHECK_LNC_LNB_GSE.FAIL`	Sim
	`A[] not CHECK_LNA_LNB_GSE.FAIL`	Sim
Verifica a impossibilidade de o sistema entrar no estado de intertravamento	`A[] not deadlock`	Sim

Tabela 5.2 Verificação das propriedades do sistema de comunicação

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
Verifica se as mensagens GOOSE tem suas variáveis *sqNum* e *stNum* iniciando com valor 1	`A[] (LNA_SRV_GSE.A and LNA_SRV_GSE.t == 0) imply LN[LN_ID[0]].gse.sqNum == 1 and LN[LN_ID[0]].gse.stNum == 1`	Sim
	`A[] (LNC_SRV_GSE.A and LNC_SRV_GSE.t == 0) imply LN[LN_ID[2]].gse.sqNum == 1 and LN[LN_ID[2]].gse.stNum == 1`	Sim
Verifica se *GoEna* habilita e desabilita o envio de mensagens GOOSE	`A[] (LNA_SRV_GSE.B) imply LN[LN_ID[0]].gse.GoEna == false`	Sim
	`A[] (LNA_SRV_GSE.RETRANSMIT) imply LN[LN_ID[0]].gse.GoEna == true`	Sim
	`A[] (LNC_SRV_GSE.B) imply LN[LN_ID[2]].gse.GoEna == false`	Sim
	`A[] (LNC_SRV_GSE.RETRANSMIT) imply LN[LN_ID[2]].gse.GoEna == true`	Sim
Verifica se mensagens GOOSE são publicadas periodicamente, se *sqNum* é incrementado, se *stNum* não é modificado (se não ocorre nenhuma mudança de estado) e quando há mudança de estado do equipamento *stNum* é incrementado e envia mensagens GOOSE com os novos valores e verifica se as mensagens recebidas após o evento estão com seus conteúdos atualizados	`A[] (LNB_CLT_GSE_LNA.A and LNB_CLT_GSE_LNA.stNum != cpyGsePdu[1][0].stNum) imply (cpyGsePdu[1][0].stNum == LNB_CLT_GSE_LNA.stNum+1 and cpyGsePdu[1][0].sqNum == 0) or (cpyGsePdu[1][0].stNum == 1 and LNB_CLT_GSE_LNA.stNum == INT32U and cpyGsePdu[1] [0].sqNum==0) and (cpyGsePdu[1][0].data != LNB_CLT_GSE_LNA.data)`	Sim
	`A[] (LNB_CLT_GSE_LNC.A and LNB_CLT_GSE_LNC.stNum != cpyGsePdu[1][2].stNum) imply (cpyGsePdu[1][2].stNum == LNB_CLT_GSE_LNC.stNum+1 and cpyGsePdu[1][2].sqNum == 0) or (cpyGsePdu[1][2].stNum == 1 and LNB_CLT_GSE_LNC.stNum == INT32U and cpyGsePdu[1][2].sqNum==0) and (cpyGsePdu[1][2].data != LNB_CLT_GSE_LNC.data)`	Sim

6 SISTEMA GERAL

Neste capítulo são apresentadas as modelagens da integração entre os controladores do veículo e dos sistemas de comunicação entre os equipamentos. Para a modelagem foram utilizados autômatos temporizados visando a simulação e verificação formal para a validação dos mesmos.

6.1 Visão Geral dos Modelos

Procurando facilitar a visualização do sistema, é apresentado o esquema da Figura 6.1, onde pode-se observar os modelos e as conexões realizadas através de canais de comunicação.

Nessa figura, os losangos não tracejados são os canais de comunicação *broadcast*, os losangos tracejados indicam os canais de comunicação *peer-to-peer*, enquanto que as elipses representam os modelos. Observa-se que não estão apresentados os dados compartilhados. No total são 23 instâncias de 17 diferentes modelos, os quais estão listados a seguir:

- Modelo de Barramento de Comunicação - este modelo é utilizado para duas instâncias diferentes:
    1. Barramento de dados que representa a VLAN 0 para comunicação de mensagens GOOSE (`BUS_VLAN0=Bus`).
    2. Barramento de dados que representa a VLAN 1 para comunicação de mensagens SMV (`BUS_VLAN1=Bus`).
- Nós Lógicos - cada modelo de nó lógico é único, havendo 3 modelos para o sistema:
    1. Controlador do Veículo (“`LNA=Trains_Control_LN`”).
    2. Nó Lógico do Simulador do Controlador GMP (“`LNB=GMP_Simulator_LN`”).
    3. Nó Lógico do Controlador da Relação Veículo e GMP (“`LNC=GMP_vs_Trains_LN`”).
- Modelo de Emissor GOOSE. Este modelo é utilizado em 3 instâncias:
    1. Emissor GOOSE para o Nó Lógico do Controlador do Veículo (“`LNA_SRV_GOOSE=Goose_Server`”).
    2. Emissor GOOSE para o Nó Lógico do Simulador do Controlador GMP (“`LNB_SRV_GOOSE=Goose_Server`”).

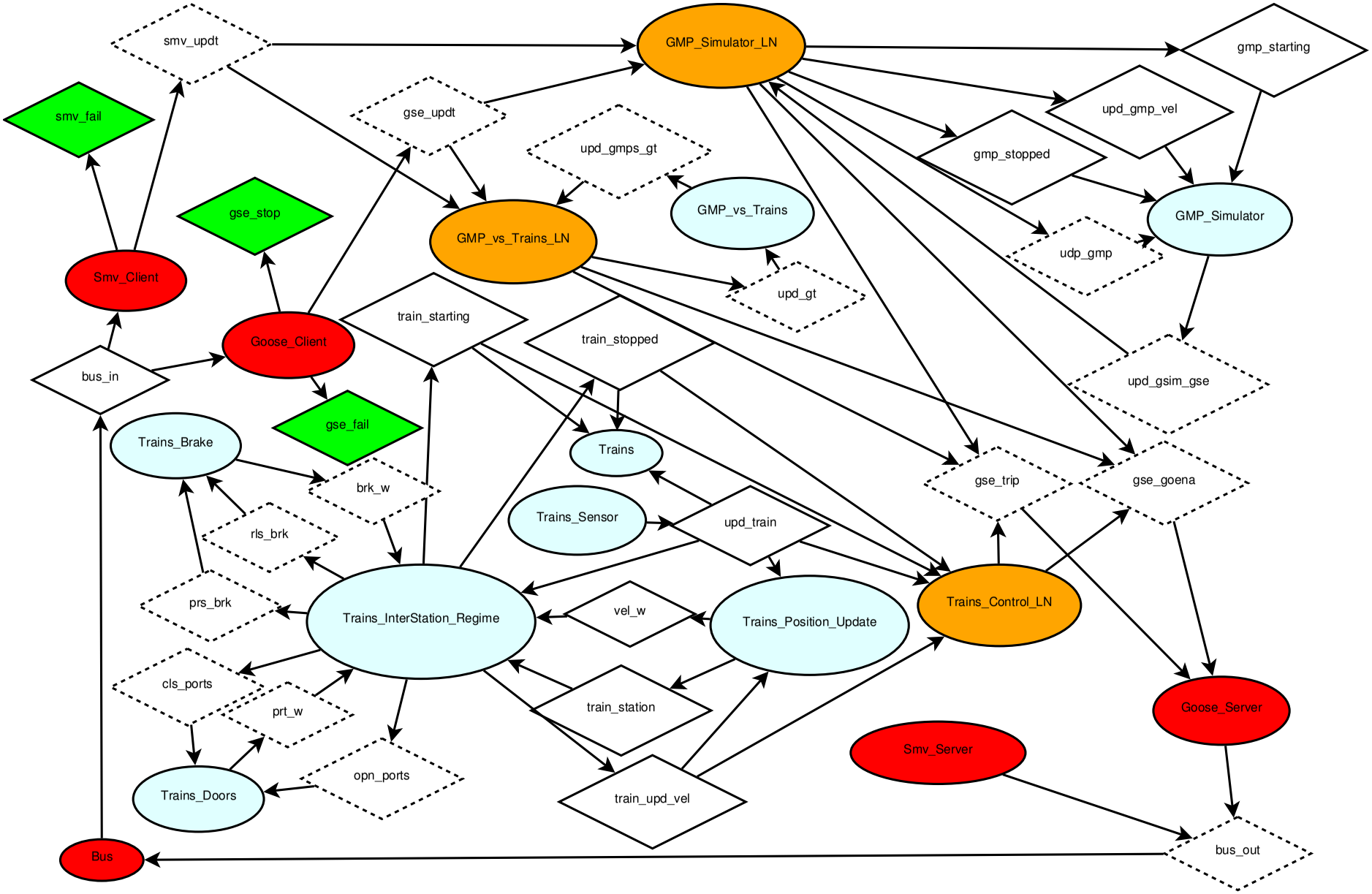


Figura 6.1 Conexão entre modelos de controle e comunicação

3. Emissor GOOSE para o Nó Lógico do Controlador da Relação Veículo e GMP (“LNC_SRV_GOOSE=Goose_Server”).

- Modelo de Emissor SMV. Este modelo é unicamente pelo modelo controlador do veículo (“LNA_SRV_SMV=Smv_Server”).
- Modelo de Receptor GOOSE. Este modelo é utilizado em 3 instâncias:
  1. Receptor GOOSE para o Nó Lógico do Simulador do Controlador GMP receber comandos do Nó Lógico do Controlador do Veículo (“LNB_CLT_GOOSE_LNA=Goose_Client”).
  2. Receptor GOOSE para o Nó Lógico do Simulador do Controlador GMP receber comandos do Nó Lógico do Controlador da Relação Veículo e GMP (“LNB_CLT_GOOSE_LNC=Goose_Client”).
  3. Receptor GOOSE para o Nó Lógico do Controlador da Relação Veículo e GMP receber comandos do Nó Lógico do Simulador do Controlador GMP (“LNC_CLT_GOOSE_LNB=Goose_Client”)
- Modelo de Receptor SMV. Este modelo é utilizado em 2 instâncias:
  1. Receptor SMV para o Nó Lógico do Simulador do Controlador GMP receber dados do Nó Lógico do Controlador do Veículo (“LNB_CLT_SMV_LNA=Smv_Client”).
  2. Receptor SMV para o Nó Lógico do Controlador da Relação Veículo e GMP receber dados do Nó Lógico do Controlador do Veículo (“LNC_CLT_SMV_LNA=Smv_Client”).
- Modelo Representativo da Posição dos Veículos (“LNA_TRAINS=Trains”).
- Modelo do Sensor de Posição dos Veículos (“LNA_TRAINS_SENSOR=Trains_Sensor”).
- Modelo de Atualização da Posição dos Veículos (“LNA_TRAINS_POSITION=Trains_ Position_Update”).
- Modelo de Regime de Deslocamento dos Veículos (“LNA_TRAINS_INTERSTATION=Trains_InterStation_Regime”).
- Modelo do Controlador de Freio dos Veículos (“LNA_TRAINS_BRAKE=Trains_Brake”).
- Modelo do Controlador de Porta dos Veículos (“LNA_TRAINS_PORTS=Trains_Doors”).
- Modelo do Simulador do GMP (“LNB_GMP_SIM=GMP_Simulator”).

- Modelo do Controlador da Relação Veículo e GMP (“LNC_GMP_VS_TRAINS=GMP_vs_Trains()”).
- Modelo de Verificação da Validade das Mensagens. Este modelo possui 5 diferentes instâncias:
    1. Verificar o envio de mensagens GOOSE entre o Nó Lógico do Controlador do Veículo e o Nó Lógico do Simulador do Controlador GMP (“CHECK_LNA_LNB_GSE=Check_Msg”).
    2. Verificar o envio de mensagens GOOSE entre o Nó Lógico do Controlador da Relação Veículo e GMP e o Nó Lógico do Simulador do Controlador GMP (“CHECK_LNC_LNB_GSE=Check_Msg”).
    3. Verificar o envio de mensagens GOOSE entre o Nó Lógico do Simulador do Controlador GMP e o Nó Lógico do Controlador da Relação Veículo e GMP (“CHECK_LNB_LNC_GSE=Check_Msg”).
    4. Verificar o envio de mensagens SMV entre o Nó Lógico do Controlador do Veículo e o Nó Lógico do Simulador do Controlador GMP (“CHECK_LNA_LNB_SMV=Check_Msg”).
    5. Verificar o envio de mensagens SMV entre o Nó Lógico do Controlador do Veículo e o Nó Lógico do Controlador da Relação Veículo e GMP (“CHECK_LNA_LNC_SMV=Check_Msg”).

Observa-se que através dos modelos apresentados será feita a simulação e a verificação formal com apenas um veículo em movimento, uma vez que o objetivo é verificar o comportamento do sistema de comunicação juntamente com o veículo lembrando que já foi verificada a funcionalidade do GMP isolado, bem como os controladores dos GMP e de trajetória com dois veículos em conjunto e o sistema de comunicação isoladamente.

Como neste caso é utilizado um conjunto de 23 autômatos temporizados, é necessário reduzir a quantidade de estados para diminuir o esforço computacional. Assim os modelos apresentados no Capítulo 4 foram modificados para representar apenas o deslocamento do veículo de uma estação para a seguinte, ao invés do trajeto circular completo entre as estações.

O objetivo, neste caso, é, partindo de uma primeira estação, verificar se é possível alcançar sem falhas e com o comportamento especificado a próxima estação. Para isso os modelos que foram feitos para processamento no UPPAAL foram modificados para utilização na sua variante UPPAAL TIGA [Cassez et al., 2005; Liu e Smolka, 1998]. UPPAAL

TIGA é uma extensão do UPPAAL e implementa algoritmo *on-the-fly*, eficiente para resolver jogos baseados em autômatos temporizados com respeito a propriedades de acessibilidade e segurança. Esta extensão permite concluir a verificação antes mesmo que tenha explorado todo o espaço de estados do modelo, pois o foco está em encontrar uma estratégia que atinja com sucesso o objetivo final (acessibilidade).

Para a verificação formal foram adicionados outros testes para definir o que é uma condição vencedora:

- Acessibilidade - o estado alvo deve sempre ser atingido ("`control: A<> acerto`").
- Acessibilidade Estrita com Prevenção - a condição tem que ser atingida e tem que evitar o erro ("`control: A[ not(falha) U acerto ]`")
- Acessibilidade Fraca com Prevenção - a condição deve ser atingida e tem que evitar o erro ("`control: A[ not(falha) W acerto ]`")
- Segurança - a condição de erro tem que ser evitada ("`control: A[] not(falha)`")

A seguir serão apresentados os modelos modificados devido à necessidade de integração do sistema de operação dos veículos e do sistema de comunicação entre os equipamentos. Conforme já comentado, alterações também foram realizadas para adaptar os modelos ao sistema para simulação e verificação formal do UPPALL TIGA.

6.1.1 Modelos

O modelo apresentado na Figura 6.2 representa o nó lógico do simulador do GMP.

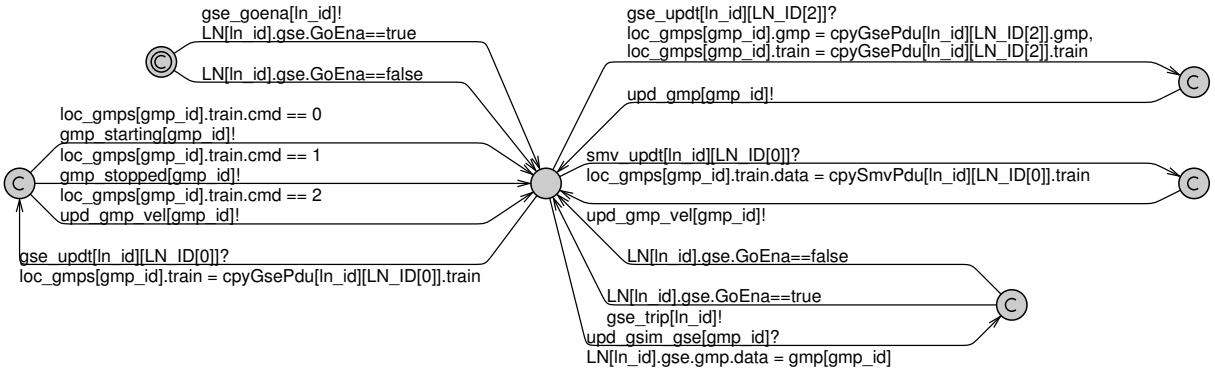


Figura 6.2 Modelo do nó lógico do simulador do GMP

Este nó lógico realiza as ações de troca de informação entre o emissor e o receptor GOOSE e SMV para o controlador do GMP (Figura 6.3).

O modelo apresentado na Figura 6.3 possui as seguintes características:

- Na inicialização é verificado através da variável "GoEna" se há necessidade de envio de mensagens GOOSE. Havendo necessidade, uma mensagem "gse_goena" é enviada para o emissor GOOSE, como pode ser observado através da Figura 6.1. Caso o nó lógico não tenha sido configurado para iniciar enviando mensagens GOOSE, uma nova verificação é realizada a cada atualização do GMP através do canal "upd_gsim_gse". Por definição, todos os nós lógicos são configurados para começar a comunicação enviando seu estado através de mensagens GOOSE.
- As mensagens "gse_updt" recebidas pelos receptores GOOSE são convertidas em comandos de acordo com a origem da mensagem.
  - As mensagem recebidas oriundas do nó lógico de controle do veículo são convertidas, de acordo com o conteúdo das mesmas, nos comandos para iniciar o funcionamento do GMP "gmp_starting", desligar o GMP "gmp_stopped" ou atualizar a velocidade do GMP "upd_gmp_vel".
  - As mensagens recebidas pelo nó lógico do integrador entre GMP's e veículos são convertidas em mensagens para atualização do veículo pelo qual o GMP irá controlar "upd_gmp".
- As mensagens "smv_updt" recebidas pelos receptores SMV são utilizadas exclusivamente para a realização da atualização de velocidade "upd_gmp_vel" e são enviadas pelo nó lógico de controle de trajetória do veículo.

O modelo apresentado na Figura 6.3 consiste de uma modificação do modelos apresentados na Figura 4.23. Esta alteração é devido à necessidade de integração a estrutura de nós lógicos. As modificações são as seguintes:

- Integração com o modelo apresentado na Figura 4.13 - desta forma, quando há solicitação de atualização de velocidade (upd_gmp_vel), é calculada a diferença entre a velocidade atual e a velocidade solicitada, resultando na aceleração necessária. Assim, respeitando os limites de aceleração (positiva ou negativa), atualiza-se a aceleração do veículo através do aumento ou diminuição da potência disponibilizada.
- As mensagens "upd_gmp", "gmp_starting" e "gmp_stopped" são, respectivamente, para troca de veículo a ser controlado, início de funcionamento e parada do GMP.

Como, para a verificação do sistema de comunicação em conjunto com o sistema de controle, foi utilizado somente um GMP para mover um único veículo de uma estação para outra. Assim, o GMP foi posicionado na região central entre as estações, puxando ou empurrando o veículo de acordo com a posição do mesmo. O modelo apresentado na Figura 6.4

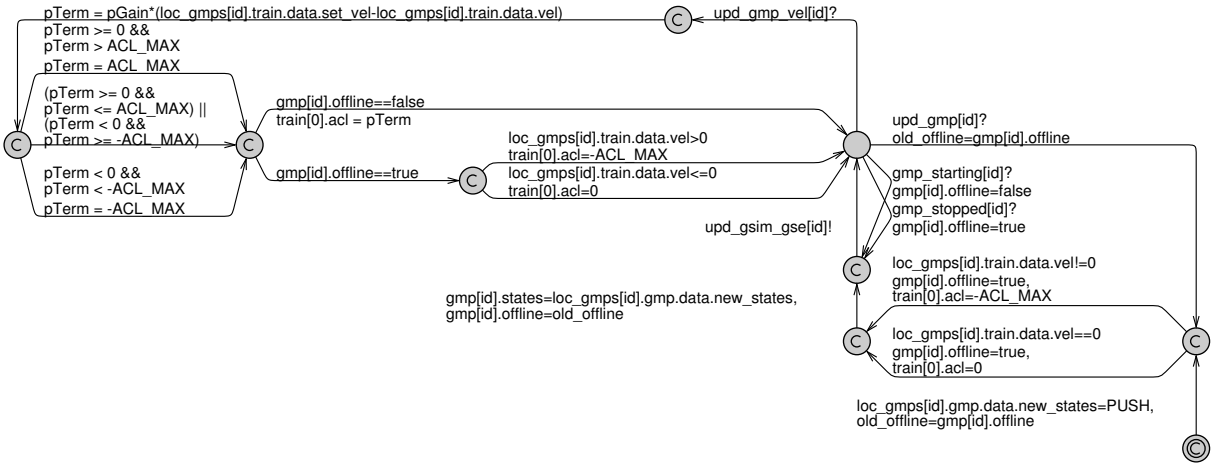


Figura 6.3 Modelo modificado para o simulador do GMP

representa o nó lógico do sistema integrador do GMP com o veículo. Este nó lógico realiza a interface entre o emissor e os receptores GOOSE e SMV e o integrador do GPM com o veículo (Figura 6.5). Este modelo possui as seguintes características:

- No caso de ocorrer alguma modificação no estado do GMP, o nó lógico recebe a mensagem "`upd_gmps_gt`" que envia mensagem para os demais equipamentos avisando sobre a mudança de estado através da mensagem "`gse_trip`".
- As mensagens "`gse_updt`" e "`smv_updt`" são recebidas quando, respectivamente, há atualização no estado do GMP e atualização da posição do veículo. Para ambas, uma mensagem é enviada para o controlador da integração do GMP com o veículo através do comando "`upd_gt`".

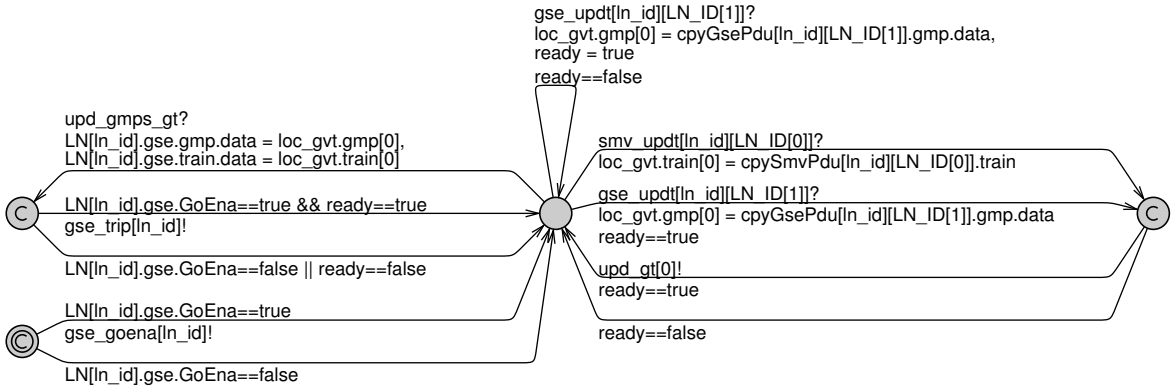


Figura 6.4 Modelo do nó lógico de integração entre GMP e veículos

O modelo do controle da integração entre o GMP e o veículo pode ser visto na Figura 6.5. Este modelo, simplificado por integrar um GMP e um veículo, solicita mudança no GMP somente na troca do estado *empurrar* ou *puxar*. Esta verificação é feita somente quando há atualização no estado do GMP ou atualização da posição do veículo.

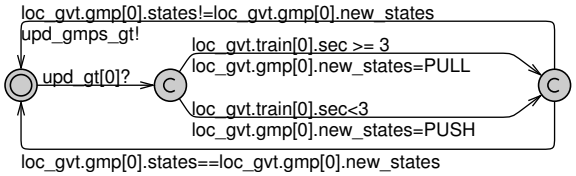


Figura 6.5 Modelo do modificado para a Integração entre GMP e veículos

O modelo apresentado na Figura 6.6 representa o nó lógico para o controlador do veículo para a situação de deslocamento entre as paradas (regime entre paradas). Este nó lógico recebe mensagens "`train_starting`", "`train_stopped`" e "`train_upd_vel`" do controlador de regime entre paradas. Estas mensagens, representam, respectivamente, o início de movimento do veículo, a parada do veículo e a atualização de velocidade do veículo. A atualização de posição do veículo (`upd_train`) é enviada concomitantemente para o nó lógico e para o controle de regime entre paradas, por ser parte integrante do conjunto de informações associadas ao veículo.

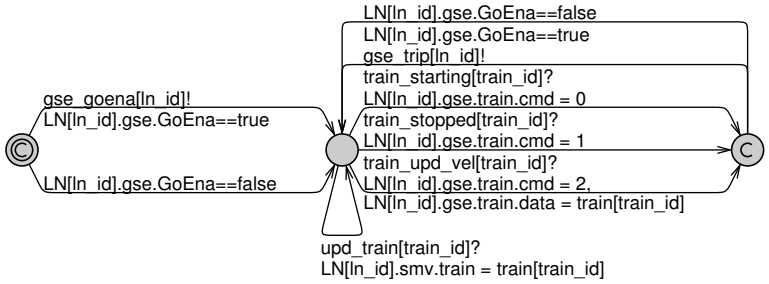


Figura 6.6 Modelo para o nó lógico do contolador de regime entre paradas

O modelo de regime entre paradas (Figura 6.7) foi modificado de forma a poder controlar o deslocamento do veículo de uma estação para a seguinte e, ao alcançar a estação alvo, divide-se em dois diferentes estados: o denominado *falha*, associada a comportamentos que não estão de acordo com o previsto e o denominado *sucesso* onde o veículo alcança a estação alvo de acordo com o comportamento especificado. Estas duas possibilidades são utilizadas na verificação formal, na qual é utilizado o algoritmo *on-the-fly*, projetado, conforme já comentado, para resolver jogos baseados em autômatos temporizados através do uso do UPPALL TIGA. O modelo em questão segue os seguintes passos para deslocar o veículo de uma estação a outra:

1. Após tempo de parada na estação excedido, envia comando avisando que o veículo está pronto para iniciar o deslocamento (`train_starting`).
2. É solicitado o fechamento das portas (`cls_ports`) e aguarda sua conclusão.
3. É solcitado a liberação dos freios (`rls_brk`) e aguarda-se a conclusão.
4. Solicita-se que o veículo atinja velocidade máxima (`train_upd_vel`). Observa-se que

o GMP controla a acelaração máxima permitida.

5. Uma vez atingida a velocidade solicitada (`vel_w`), que não necessariamente é atingida, dependendo da distância entre as estações, o controlador do veículo inicia o processo de aguardar a chegada à seção anterior a estação.

6. Próximo à estação, é solicitada a velocidade mínima de deslocamento do veículo. Novamente, o GMP controla a aceleração máxima permitida.

7. Ao atingir a velocidade mínima e posicionar-se na região da estação (`train_station`) é solicitada a velocidade nula para execução da parada.

8. Atingida a velocidade mínima, os freios são acionados (`prs_brk`) e as portas abertas (`opn_ports`).

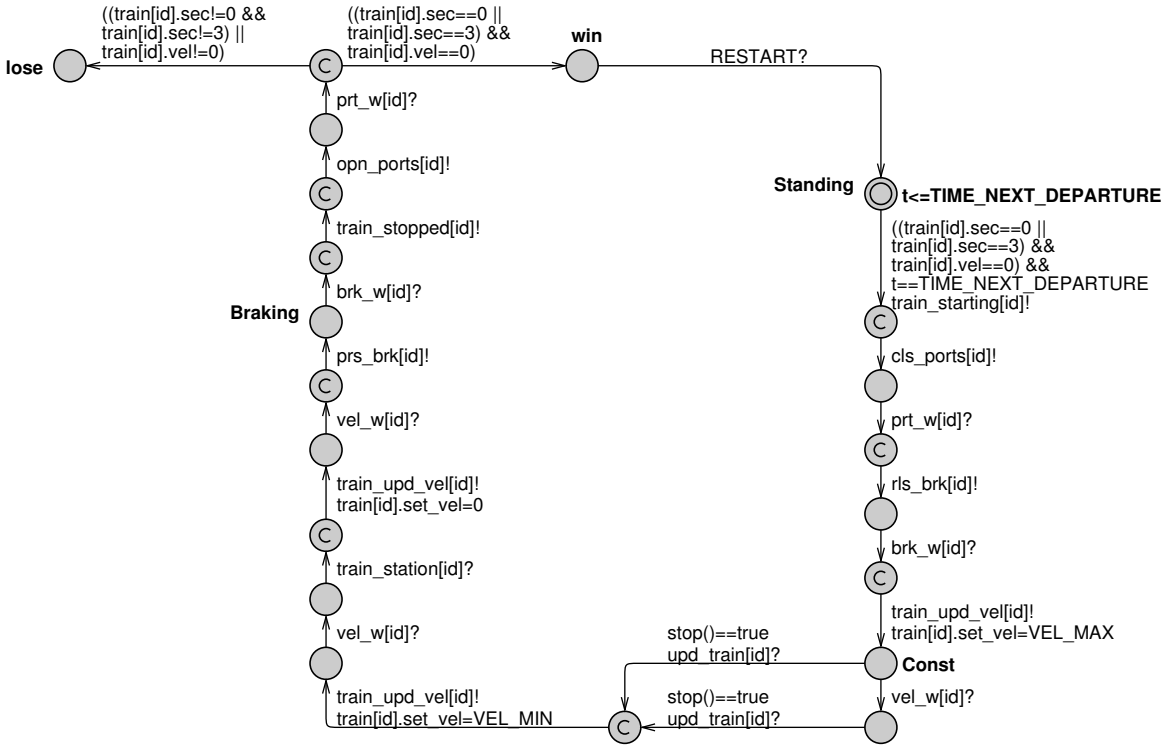


Figura 6.7 Modelo modificado para o regime entre paradas

Na Figura 6.7, o sinal de sincronização "`RESTART`", foi inserido para indicar a possibilidade de reinício de deslocamento do veículo, porém, para fins de simulação e verificação formal, este sinal de sincronização não é utilizado visto que, garantindo que o deslocamento do veículo está de acordo com o especificado, não é necessário que se realize o retorno do veículo, uma vez que as condições se mantém idênticas no caso em estudo. Neste caso em específico o objetivo foi reduzir a quantidade tempo e recursos computacionais para processamento da rede de autômatos temporizados.

Os modelos de atualização de força (Figura 4.13), de posição (Figura 4.14) e da relação posição *versus* segmento (Figura 4.15), apresentados no Capítulo 4, foram integrados em um modelo único (Figura 6.8). Adicionalmente foi adicionado um divisor ("`DIV`") nas equações de posição e velocidade (respectivamente equações 4.1 e 4.2) com objetivo de aumentar a resolução da atualização do movimento do veículo no tempo.

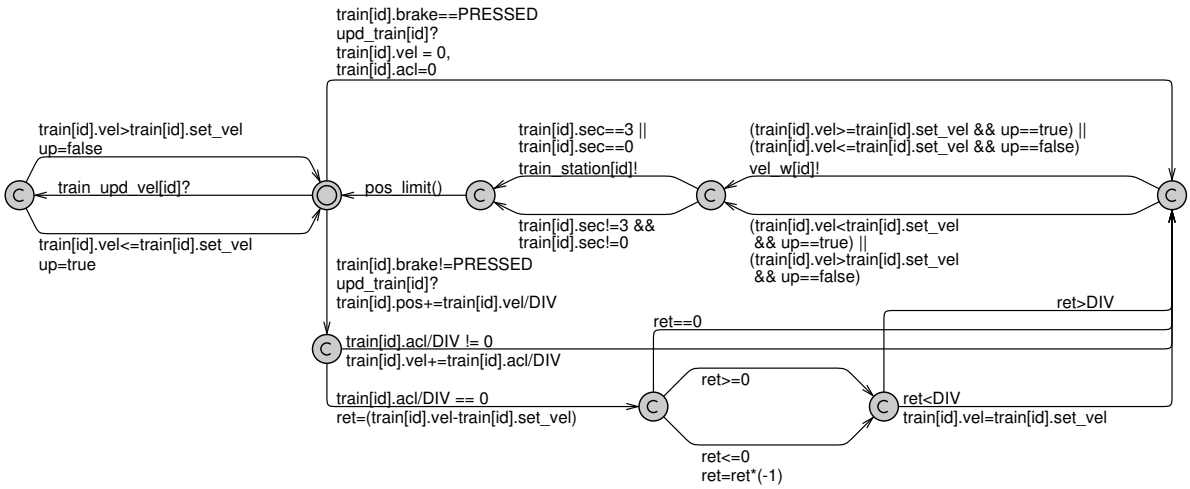


Figura 6.8 Modelo modificado para atualização da posição, velocidade e aceleração

A função "`pos_limit()`", na Figura 6.8, realiza o cálculo do segmento de acordo com a posição do veículo, atualizando a variável "`train[id].sec`".

6.2 Simulação

Para avaliação do funcionamento do sistema proposto foi realizada simulação com o uso do *software* UPPAAL. Os resultados das simulações do deslocamento do veículo para uma trajetória ponto-a-ponto podem ser observados nas figuras 6.9, 6.10, 6.11 e 6.12 (respectivamente, posição, velocidade, aceleração e segmento associado à posição do veículo). É possível observar que, utilizando os protocolos de comunicação IEC61850, o veículo comporta-se de acordo com o esperado.

Na Figura 6.10, a linha contínua indica a velocidade enquanto que a linha pontilhada representa a velocidade alvo (*set-point*).

Conforme comentado, no modelo apresentado na Figura 6.7, a aceleração limite (Figura 6.11) é controlada pelo GMP e, ao aproximar-se do local de parada (estação), o veículo diminui a velocidade até alcançar a velocidade mínima considerada antes da parada completa, que é ocasionada pelo acionamento do sistema de frenagem.

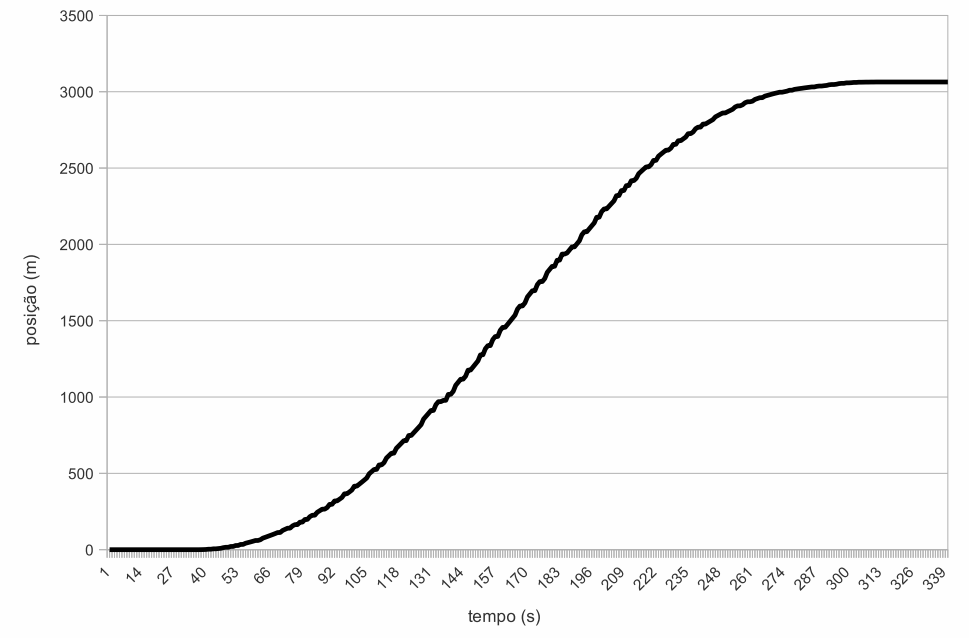


Figura 6.9 Resultado da simulação: posição

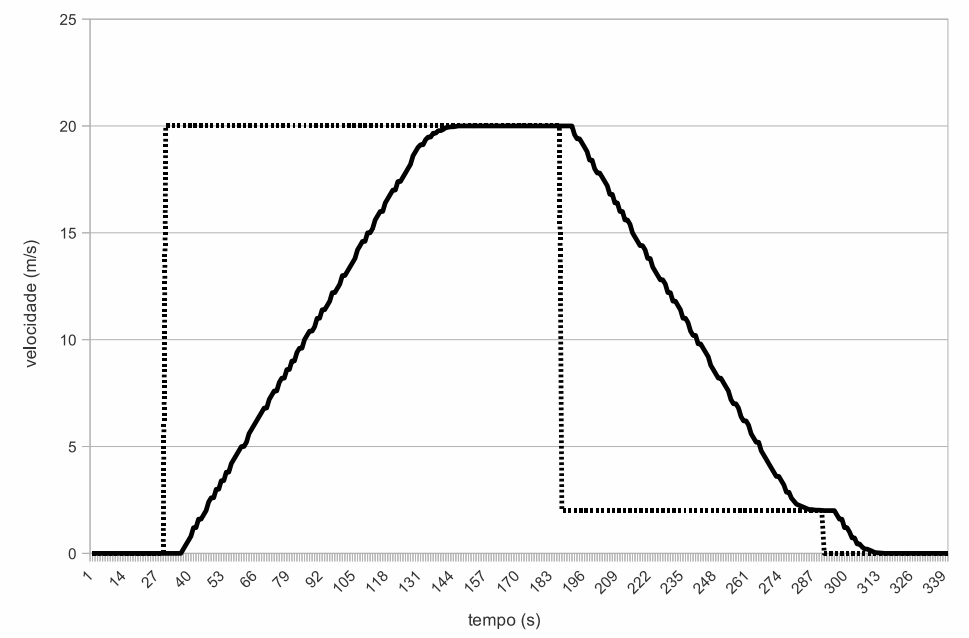


Figura 6.10 Resultado da simulação: velocidade

Na Figura 6.12 são apresentadas as evoluções dos estados das portas do veículo, sistema de frenagem e segmento de posicionamento no trilho. Observa-se que a sequência é respeitada, sendo o sistema de frenagem liberado somente após as portas estarem fechadas. A condição de portas abertas ocorre somente nos segmentos 0 e 3, onde localiza-se a estação, desde que a velocidade seja nula e os freios estejam acionados.

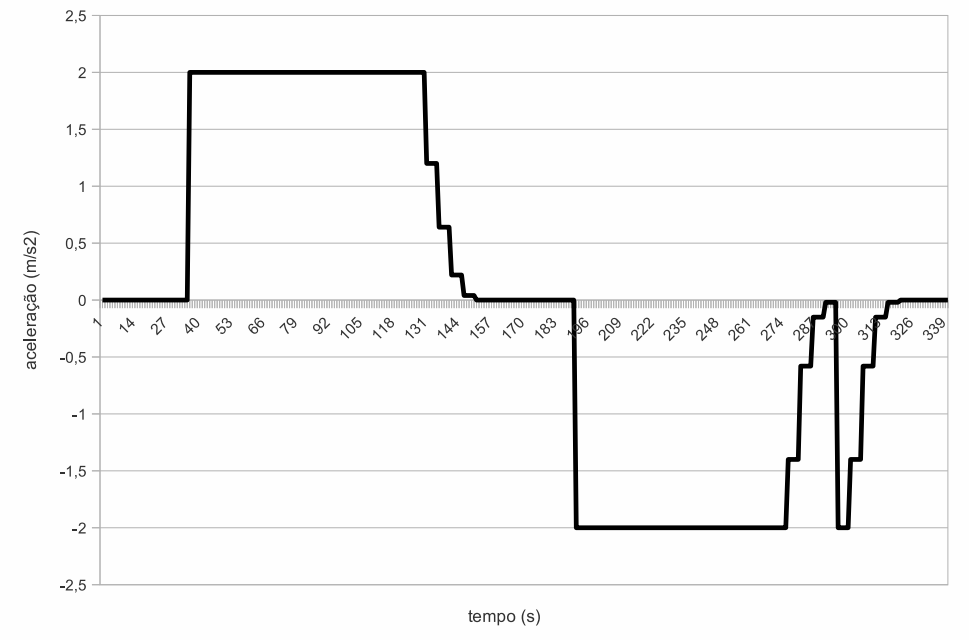


Figura 6.11 Resultado da Simulação: Aceleração

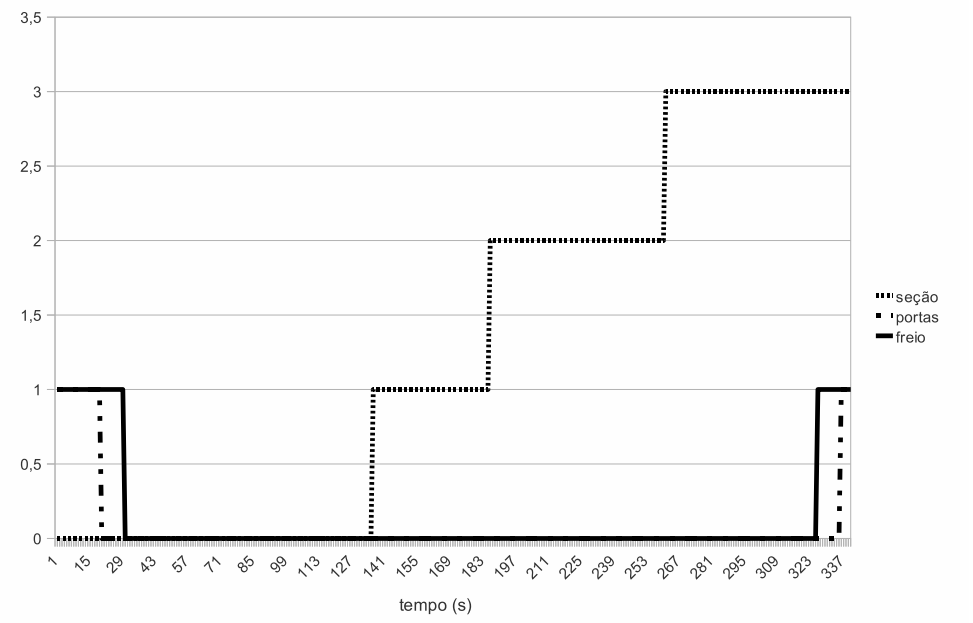


Figura 6.12 Resultado da simulação: segmento

6.3 Verificação Formal

No que diz respeito à verificação formal dos modelos, alguns comportamentos dos protocolos de comunicação GOOSE e SMV são esperados, assim como do veículo. Os comportamentos testados estão descritos em linguagem natural e formal, de acordo com o padrão de entrada para o verificador de modelos UPPAAL. Conforme já comentado, foi utilizada a representação em espaço de estados DBM (*Difference Bounded Matrices*) [Dill,

1989]. Os resultados das verificações podem ser observados na Tabela 6.1.

Tabela 6.1 Verificação das propriedades do sistema de comunicação

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
Verifica se o modelo alcança sempre o estado denominado *sucesso* que representa que o veículo chega com velocidade nula, portas abertas, freios acionados na seção referente a estação de destino	`control: A<> LNA_TRAINS_INTERSTATION.win`	Sim
Verifica se o conteúdo das mensagens estão coerentes entre emissores e receptores para os protocolos GOOSE e SMV	`A[] not CHECK_LNC_LNB_SMV.FAIL`	Sim
	`A[] not CHECK_LNA_LNB_SMV.FAIL`	Sim
	`A[] not CHECK_LNC_LNB_GSE.FAIL`	Sim
	`A[] not CHECK_LNA_LNB_GSE.FAIL`	Sim
Verifica a impossibilidade de o sistema entrar no estado de intertravamento	`A[] not deadlock`	Sim

7 AMPLIAÇÃO DA NORMA IEC 61850

Neste capítulo, de acordo com os objetivos gerais do presente trabalho, são apresentados os procedimentos propostos para a ampliação da norma IEC 61850 no desenvolvimento dos sistemas de comunicação, supervisão, operação e controle de sistemas APM.

7.1 Organização dos Nós Lógicos

De acordo com o trabalho desenvolvido pode-se classificar os modelos nos seguintes grupos de nós lógicos:

- Equipamentos mecânicos - os equipamentos, por norma, iniciam com a letra “K”.
  - Sistema de frenagem do veículo (KBRK).
  - Ventilador do sistema de potência (KFAN).
  - Sistema de portas do veículo (KDRS).
  - Válvulas Proporcionais do GMP (KVLV).
  - Válvulas *On/Off* do GMP (KVLV).
- Controladores - os controladores, por norma, iniciam com a letra “C”.
  - Regime Interestações (CTRN).
  - Controle da Relação GMP e Veículos (CGTR).
  - Controlador do GMP (CGMP).
- Sensores - os sensores, por norma, iniciam com a letra “T”.
  - Sensor de Posição (TDST).
- Monitoramento - os blocos de monitoramento, por norma, iniciam com a letra “M”.
  - Cálculo do segmento em que o veículo se encontra a partir do conhecimento da sua posição (MSEG).

Para cada dispositivo lógico existe um nó lógico denominado LN0 e para cada dispositivo físico há um nó lógico com informações sobre o dispositivo. De acordo com a IEC, 2003f este nó lógico é denominado de LPDH. O nó lógico LN0 é responsável pelo envio e recebimento de mensagens GOOSE e SMV. Foi considerado, nos capítulos anteriores, um

dispositivo lógico para cada dispositivo físico. Outros equipamentos necessários para o sistema APM e para norma IEC 61850 como, por exemplo, a comunicação com a interface SCADA e o protocolo para sistemas supervisórios não foram implementados por estarem fora do escopo do trabalho.

Na Figura 7.1 é apresentada a relação entre os nós lógicos e o nó lógico IHMI (interface com sistema SCADA), descrito em IEC, 2003f. Pode-se observar que, para cada CTRN, há um KBRK, um KDRS e um TDST. Já, para o nó lógico CGMP, há dez KVLV e apenas um KFAN.

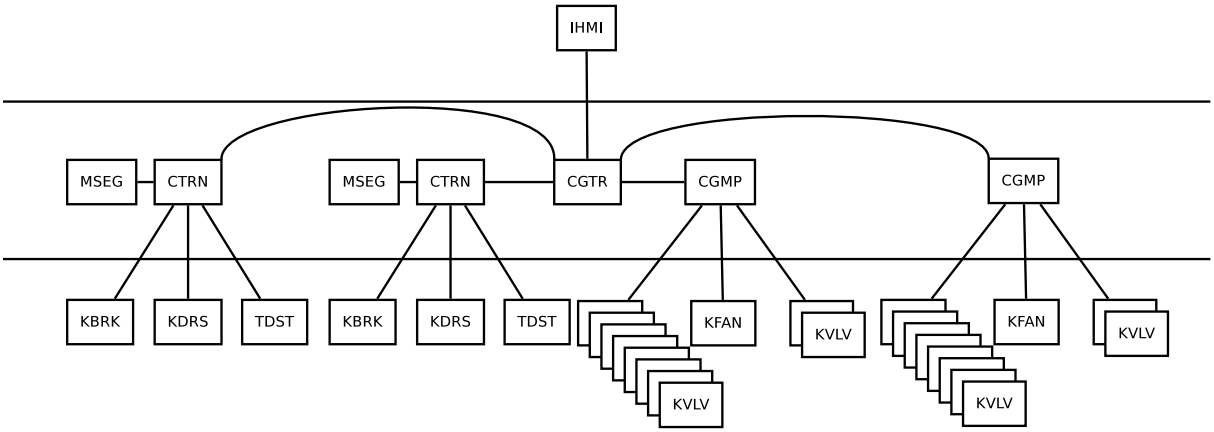


Figura 7.1 Relação entre nós lógicos

De acordo com os modelos simulados e verificados formalmente, estão detalhados abaixo os atributos necessários para cada um dos nós lógicos, com exceção dos atributos dos nós lógicos IHMI, LLN0 e LPHD, que estão descritos em IEC, 2003f e dos atributos dos nós lógicos KVLV e TDST, que estão descritos em IEC, 2007. Observa-se que, para todos os nós lógicos, há herança do nó lógico padrão (*Common Logical Node*) que possui informações e serviços independentes da função realizada [IEC, 2003d].

De acordo com os modelos das figuras 4.21 e 4.22, os nós lógicos KBRK e KDRS estão descritos, respectivamente, nas Tabelas 7.1 e 7.2, onde apresentam variável indicadora do estado do equipamento do tipo SPS (*Single Point Status*) e sinais de controle do tipo SPC (*Controllable single point*) definidos na norma IEC, 2003e.

O nó lógico do cálculo do segmento onde o veículo se encontra (MSEG), de acordo com o modelo apresentado na Figura 4.15, é descrito através da Tabela 7.3, onde MV indica o valor medido de acordo com IEC, 2003e.

O nó lógico de regime de deslocamento entre as estações (CINT), de acordo com o modelo apresentado na Figura 6.7, é descrito através da Tabela 7.4, onde INC (*Controllable Integer Status*) é definido de acordo com IEC, 2003e.

Tabela 7.1 Sistema de frenagem (KBRK)

Classe KBRK			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Controles			
PrsBrk	SPC	Aciona sistema de frenagem	Sim
RlsBrk	SPC	Libera sistema de frenagem	Sim
Sinais de Estado			
Brk	SPS	Estado de freio acionado ou liberado	Sim

Tabela 7.2 Sistema de portas (KDRS)

Classe KDRS			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Controles			
OpnDrs	SPC	Abre portas	Sim
ClsDrs	SPC	Fecha Portas	Sim
Sinais de Estado			
Drs	SPS	Estado de portas fechadas ou abertas	Sim

Tabela 7.3 Cálculo de segmento (MSEG)

Classe MSEG			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Sinais Amostrados			
Seg	MV	Cálculo de segmento de trilho indicando posição do veículo atualizado pelo sensor de posição (TDST)	Sim

O nó lógico de controle da relação entre GMP e veículos (CGTR), de acordo com os modelos apresentado nas figuras 4.18 e 4.17, é descrito através da Tabela 7.5.

Tabela 7.4 Regime interestações (CINT)

Classe MSEG			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Controles			
SpVel	INC	Objetivo de velocidade	Sim
Stp	SPC	Inicio do deslocamento	Sim
Str	SPC	Fim do deslocamento	Sim
Sinais Amostrados			
Acl	MV	Cálculo da aceleração do veículo atualizado pelo sensor de posição (TDST)	Sim
Vel	MV	Cálculo da velocidade do veículo atualizado pelo sensor de posição (TDST)	Sim

Tabela 7.5 Controle da relação GMP e veículos (CGTR)

Classe CGTR			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Controles			
UpdGt	SPC	Atualizar relação GMP e Veículos	Sim
Configurações			
Trn	ING	Quantidade de veículos em funcionamento	Sim
Gmp	ING	Quantidade de GMP disponíveis	Sim

O nó lógico do controlador do GMP (CGMP), de acordo com o modelo apresentado na Figura 6.3, é também descrito através da Tabela 7.6, onde ING (*Integer Status Setting*) é definido de acordo com IEC, 2003e.

Tabela 7.6 Controlador do GMP (CGMP)

Classe CGTR			
Nome do atributo	Tipo	Descrição	Obrigatório
LNName		Deve ser herdado da classe nó lógico padrão (ver IEC, 2003d)	
Dados			
Informações do nó lógico padrão			
		Deve ser herdado os dados obrigatórios da classe nó lógico padrão	Sim
Controles			
UpdGt	SPC	Atualizar relação GMP e Veículos	Sim
Configurações			
Sinais de Estado			
Off	SPS	Estado de GMP desconectado do duto central	Sim
PuPl	SPS	Estado atual do GMP (empurra ou puxa)	Sim
StpPP	SPS	Estado alvo do GMP (empurra ou puxa)	Sim
Lck	SPS	Estado GMP por controle manual	Sim
Configurações			
AclMax	ING	Aceleração máxima do veículo	Sim
TrnID	ING	Identificador do veículo	Sim

7.2 Etapas de Desenvolvimento

A metodologia para desenvolvimento de sistemas de controle de APM utilizando conceitos da norma IEC 61850 pode ser sumarizada através das etapas descritas a seguir. Estas etapas são propostas baseadas no estudo de caso apresentado no trabalho (sistema Aeromovel).

Os nós lógicos básicos considerados são resultado da metodologia de desenvolvimento do sistema de controle do sistema Aeromovel. Para cada etapa (Figura 7.2(a)) são desenvolvidos os passos indicados na Figura 7.2(b). As etapas propostas são as seguintes:

1. Levantamento dos requisitos de controle do sistema APM.
2. Desenvolvimento dos modelos que representem o comportamento dos equipamentos pertencentes ao sistema de potência, sem restrições de controle, como válvulas, motores e movimento do veículo.
3. Desenvolvimento dos modelos dos controladores na forma de autômatos temporizados, de forma que os modelos dos sistemas eletromecânicos do sistema de potência tenham o comportamento esperado, de acordo com os requisitos definidos. Nesta etapa, são especificados os elementos do sistema de proteção, como, por exemplo, a exclusão de estados do sistema de potência que possam ocasionar danos aos equipamentos.

4. Desenvolvimento dos modelos que representem o comportamento dos equipamentos pertencentes ao sistema do veículo, sem restrições de controle, como válvulas e movimento do veículo.

5. Desenvolvimento dos modelos dos controladores na forma de autômatos temporizados, de modo que os modelos dos sistemas eletromecânicos do sistema do veículo tenham o comportamento esperado de acordo com os requisitos definidos, como, por exemplo, elementos de operação do sistema associados ao controle de movimento, parada programada na estação e controle das portas, além de elementos de proteção do sistema, como de detecção de movimento involuntário e proteção contra sobrevelocidade.

6. Integração entre os modelos dos equipamentos e controladores dos sistemas de potência e de movimentação do veículo.

7. Desenvolvimento de modelos que representem o comportamento dos equipamentos de comunicação de dados, como o barramento de dados, tipos de mensagens e nós lógicos.

8. Integração entre os modelos dos equipamentos de controle dos sistemas de potência, movimentação do veículo e comunicação de dados. Neste ponto, é necessária a adaptação dos modelos para compatibilizar os controladores com os nós lógicos. Como resultado desta etapa, tem-se a definição dos nós lógicos.

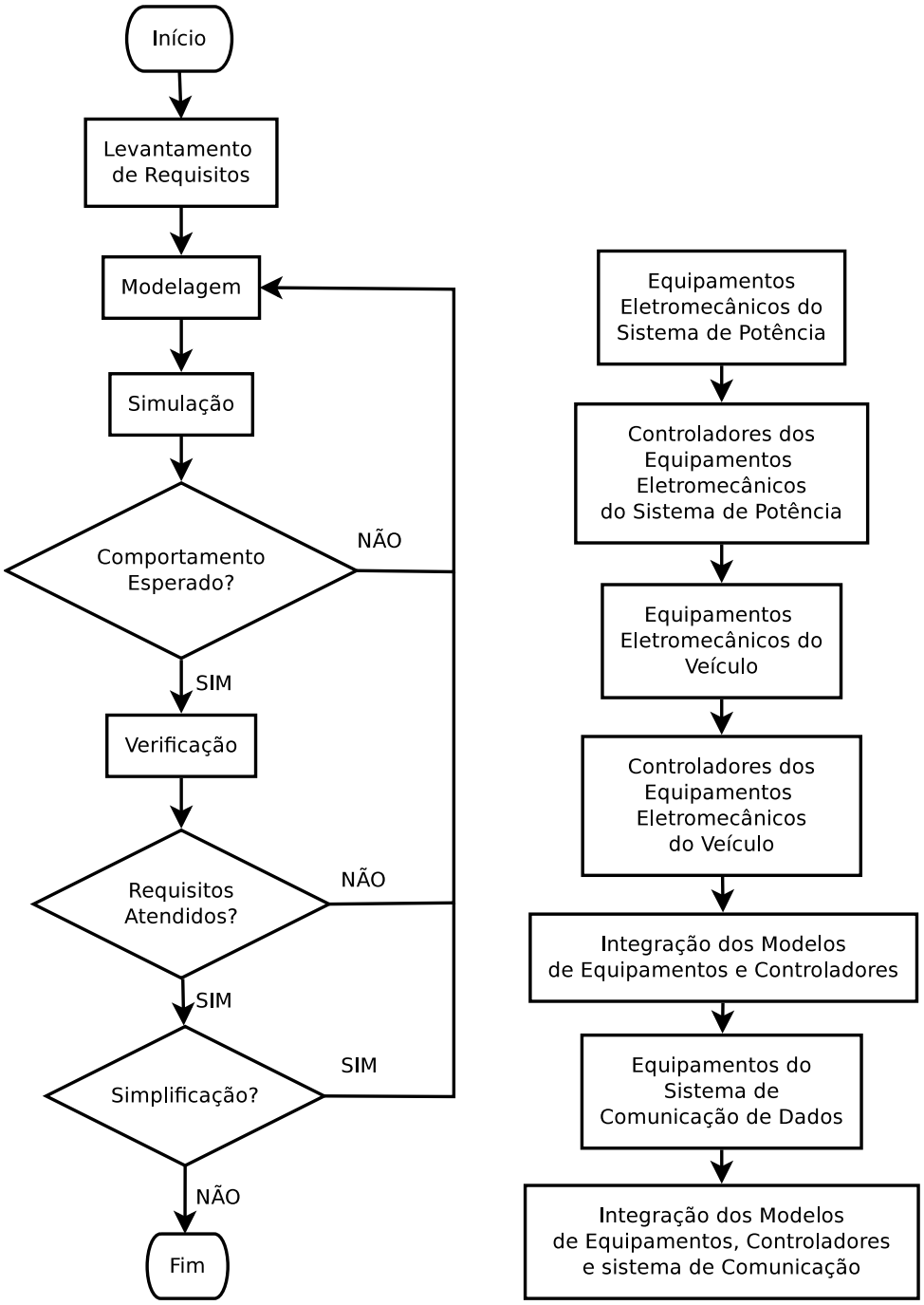
9. Desenvolvimento dos modelos descritivos do sistema de supervisão (ATS).

10. Integração entre os modelos dos equipamentos controladores dos sistemas de potência, movimentação do veículo, comunicação de dados e sistema de supervisão (ATS).

Para todas as etapas, a verificação do comportamento esperado (em linguagem TCTL "E<>φ") e do evento de falha ("A[]φ") devem estar de acordo com o requisitos levantados. A simulação auxilia na avaliação do comportamento esperado.

Esta metodologia fornece um *crescendum* sólido da solução para controle de APM, permitindo simulações e testes de conformidade para cada um dos modelos desenvolvidos (etapa por etapa). Como resultado final, tem-se um sistema graficamente modelado em autômatos temporizados, e verificável até a sua implementação para aplicação no sistema APM, permitindo assim, um rastreamento preciso de possíveis falhas.

Com a aplicação da norma IEC 61850, tem-se a abstração de *hardware*, uma vez que o planejamento e operação se dão, em grande parte, em um sistema virtualizado. Conforme afirmado anteriormente, o uso de linguagem de configuração SCL (*System Configuration Language*) abrange o ciclo de vida do projeto desde a concepção à engenharia, operação



(a) Local (b) Global

Figura 7.2 Etapas de desenvolvimento

e manutenção, facilitando a comunicação entre equipes. Adicionalmente, o uso de uma nomenclatura normatizada auxilia na legibilidade.

Como desvantagem, tem-se a alta necessidade de processamento computacional quando se trata de sistemas complexos. Porém, como alternativa, tem-se a possibilidade de realizar simplificações (como foi feito no sistema GMP) de forma a reduzir o tempo de processamento. Atualmente, devido ao rápido desenvolvimento tecnológico (tanto em *software* quanto em

hardware), o acesso a computadores está muito facilitado e ferramentas como UPPAAL tem ganho novos recursos tais como uso de sistemas 64-bits (permitindo o uso maior de memória) e processamento distribuído para checagem estatística de modelos (data base: julho de 2011).

8 TESTES DE CONFORMIDADE

Neste capítulo é apresentada a proposta de validação dos controladores implementados a partir de autômatos temporizados.

8.1 Conformidade de Sistemas

A simulação e a verificação formal dos autômatos temporizados resultam em garantias de que os modelos propostos estão de acordo com os requisitos solicitados. Porém, os modelos não são aplicáveis diretamente ao sistema físico e para isso é necessária a tradução da rede de autômatos em uma linguagem de programação que permita o interfaceamento com os dispositivos a serem controlados. Durante o processo de tradução de autômatos temporizados para a linguagem do CLP, por exemplo, deve ser verificado se o código produzido está de acordo com o sistema modelado, simulado e formalmente verificado, com a finalidade de aumentar a confiança e a segurança do sistema [Oliveira, 2009].

A etapa de tradução dos autômatos temporizados em questão neste trabalho para linguagem dos equipamentos que irão efetivamente controlar o sistema é complexa. As maiores dificuldades estão associadas à dependência de desempenho (sistema *hard real-time*) e à necessidade de interfaces com sistemas externos através de do uso de sensores e atuadores, o que depende das características de acesso de entradas e saídas do sistema operacional.

A UPPAAL TRON permite testar se o comportamento do programa produzido está de acordo com o comportamento do modelo em autômatos temporizados pela análise dos sinais de saída e entrada, ignorando o funcionamento interno do sistema [Oliveira, 2009], criando um assim chamado de teste de “caixa preta”.

A Figura 8.1 mostra a rede de autômatos temporizados divididos entre um modelo de ambiente e de IUT (*Implementation Under Test*). UPPAAL TRON substitui o modelo da IUT por um programa externo, e com base na sequência temporizada de entrada e de saída, ações do restante da rede de autômatos estimulam a IUT. Também, em tempo real, verifica se a produção de sinais da IUT está em conformidade com os requisitos. Neste sistema assume-se que o IUT é uma “caixa-preta” cujo estado interno não é diretamente observável e que apenas os sinais de entrada e saída são observáveis [Larsen et al., 2005].

O adaptador é o componente de *software* que conecta o UPPAAL TRON à IUT. Ele é responsável por traduzir os eventos da rede de autômatos em sinais de entrada em estímulos para

a IUT e os sinais de saída da IUT em sinais para rede de autômatos. Dependendo da construção do adaptador, UPPAAL TRON pode ser conectado ao *hardware* diretamente (possivelmente através de sensores e atuadores) ou pode ser ligado diretamente a um *software*.

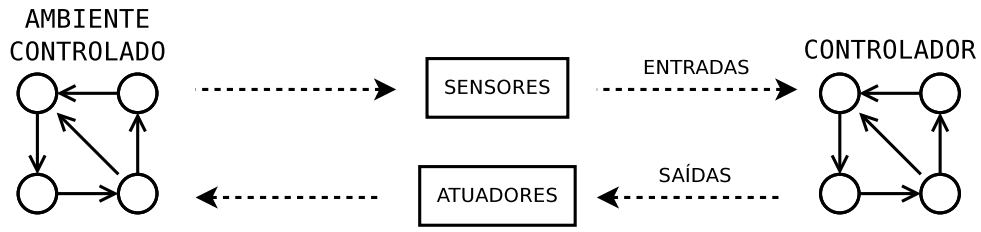


Figura 8.1 Modelo de funcionamento da ferramenta UPPAAL TRON

A Figura 8.1 mostra a configuração do sistema típico durante a implantação do sistema: o ambiente é uma planta que precisa ser dirigida e controlada, e a implementação é um controlador de *software/hardware* tendo como entrada os sensores e saída os atuadores que agem sobre o ambiente. É importante destacar que UPPAAL TRON pode assumir a perspectiva do controlador para validar o modelo de ambiente ou assumir a perspectiva do ambiente se o objetivo for o de avaliar a conformidade do controlador.

Na Figura 8.2, o modelo de ambiente é executado pelo UPPAAL e os canais são interceptados pelo UPPAAL TRON que, através de seu *driver*, envia sinais para o adaptador que, por sua vez, comunica-se diretamente com o IUT (unidade sob teste). Na configuração do teste genérico, o adaptador traduz mensagens de entrada abstratas em ações físicas e reconhece as realizações físicas codificando-as em mensagens abstratas compreendidas pela ferramenta UPPAAL TRON. O *software* adaptador é uma implementação específica para cada objeto em estudo. Observa-se que o adaptador pode conter falhas que são identificadas pelo UPPAAL TRON, sendo uma parte integrante da aplicação sob teste. O adaptador deve ser suficientemente rápido para replicar o comportamento dos sensores e atuadores, e o testador para emular o ambiente e, portanto, fornecer testes confiáveis [Larsen et al., 2005].

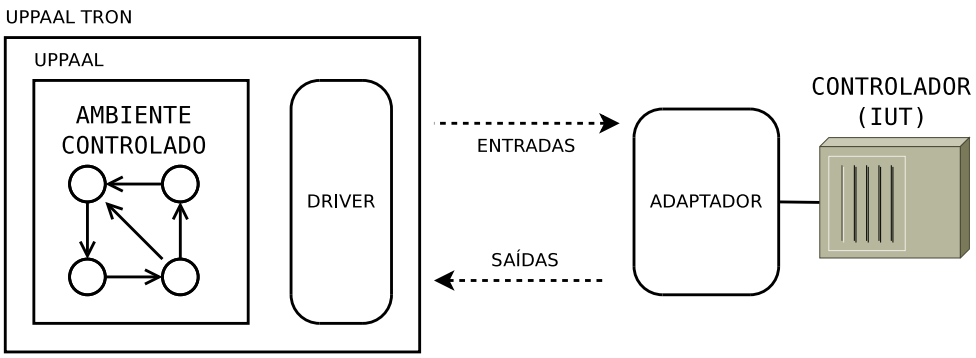


Figura 8.2 Modelo de funcionamento da ferramenta UPPAAL TRON com sistema IUT

A comunicação entre o *driver* e o adaptador pode ocorrer de uma das seguintes formas:

- Bibliotecas (*Dynamic-link library* - DLL). Meio mais rápido de comunicação entre *driver* e adaptador, onde todos os recursos de dados são compartilhados.
- TCP/IP (*Transmission Control Protocol/Internet Protocol*). Permite comunicação remota, assim, a IUT não necessita estar no mesmo computador ou arquitetura.
- Textual. Comunicação realizada através de comandos escritos em um arquivo de texto. É um método mais lento de comunicação e tem a particularidade de poder produzir uma sequencia de eventos pré-determinados (*script*), permitindo assim a avaliação, por exemplo, da resposta do sistema a uma sequencia de falhas durante a validação do sistema.

O UPPAAL TRON permite o uso de relógio virtual e relógio de tempo real. O uso relógio virtual tem como finalidade proporcionar condições de teste onde há um relógio de referência global que é controlado e não dependente do tempo físico. Desta forma, o foco da verificação permanece relacionado, principalmente, a funcionalidade. O relógio de tempo real, além da avaliação da funcionalidade, também avalia desempenho. O uso sequencial destes dois métodos permite isolar as falhas relacionadas à funcionalidade das falhas relacionadas ao desempenho.

A seguir são apresentados dois exemplos de verificação de conformidade relativos ao sistema de controle de GMP e do sistema de comunicação da norma IEC 61850 (mensagens GOOSE).

8.2 Conformidade do Controlador do GMP

O controlador do GMP foi escolhido para ser apresentado por se tratar de um elemento crítico para o controle do veículo. A organização do teste de conformidade do GMP está apresentada na Figura 8.3.

Observa-se que algumas mudanças foram realizadas como resultado da adaptação dos modelos apresentados no Capítulo 4. A primeira trata-se da substituição por interrupções dos estados nos quais era realizada a técnica de *pooling*. Assim, nos estados do modelo do controlador de válvulas onde eram realizados *loopings* de controle com intervalos de tempo mínimo (críticos) através do uso de invariantes (com o objetivo de verificar a mudança de estado da válvula controlada), os *loopings* foram trocados pelo método de interrupção, onde o sistema aguarda a mensagem emitida pela válvula indicando a mudança de estado. Desta forma, o processamento é reduzido pois enquanto aguarda a interrupção, o processador permanece liberado para outras atividades. Esta necessidade não foi identificada durante o processo de

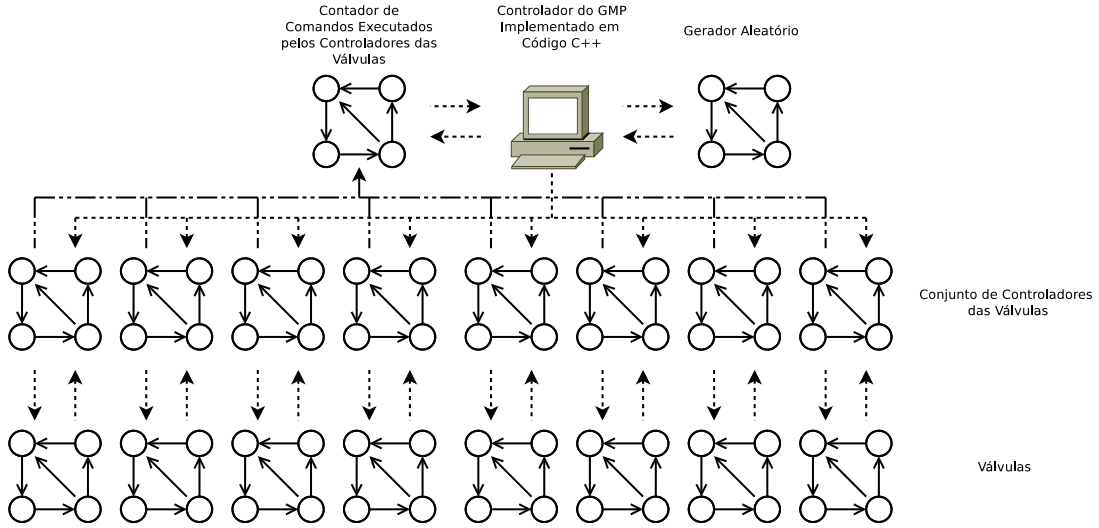


Figura 8.3 Avaliação de conformidade do GMP

simulação ou verificação formal, tendo sido identificada somente no teste de conformidade com relógio de tempo real.

Desta forma, canais do tipo *broadcast* foram inseridos, indicando se a válvula alcançou o estado de aberta ("`valv_opn_ack`") ou fechada ("`valv_cls_ack`"). Os modelos atualizados das válvulas e dos controladores das válvulas podem ser vistos, respectivamente, nas figuras 8.4 e 8.5.

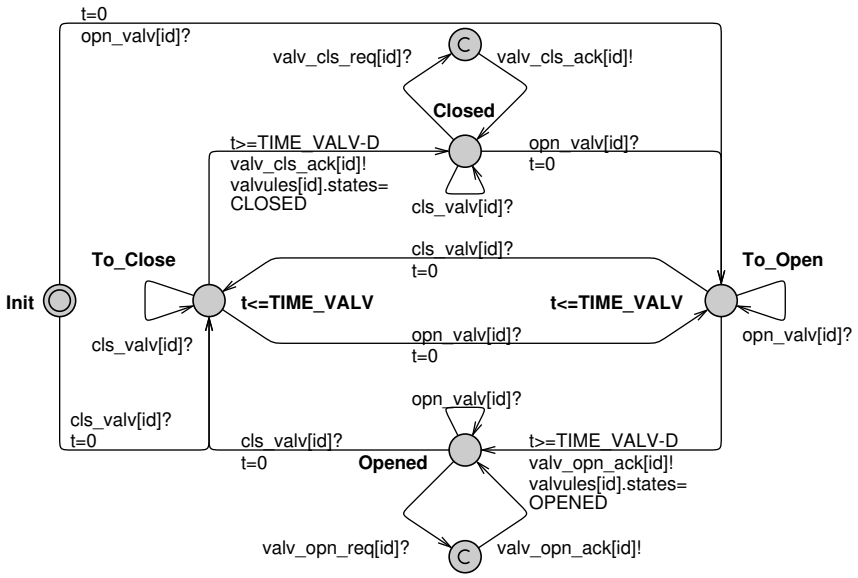


Figura 8.4 Modelo de válvulas para uso de interrupções

Similarmente, as mudanças ocorridas nos modelos das válvulas do tipo "*on-off*", as válvulas proporcionais também foram modificadas para o uso de interrupções (como pode ser visto através das figuras 8.6 e 8.7), com o uso do canal de comunicação "`valv_prop_ack`". O modelo do motor permaneceu inalterado.

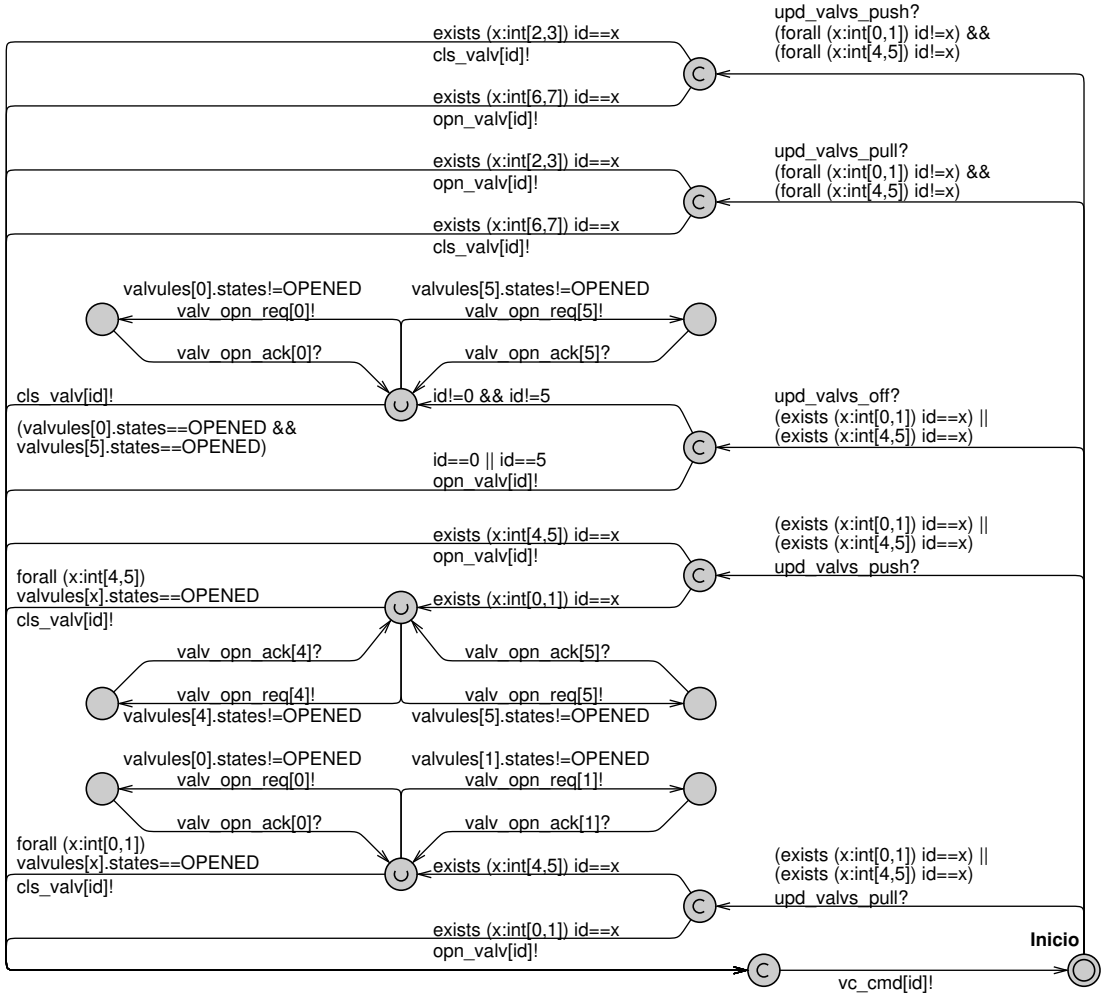


Figura 8.5 Modelo de controladores de válvulas para uso de interrupções

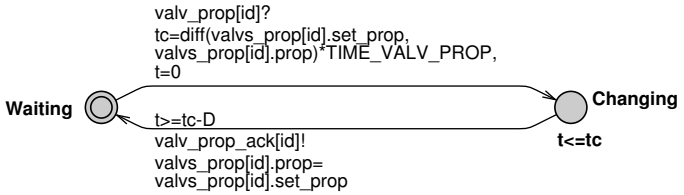


Figura 8.6 Modelo de válvulas proporcionais para uso de interrupções

Para verificar se a mudança de estado do GMP foi concluída, ou seja, se todas as válvulas acionados pelo comando já atingiram seu estado final, foi desenvolvido o modelo apresentado na Figura 8.8. Para cada comando há um número determinado de válvulas a serem acionadas. Assim, o modelo em questão verifica se o controlador das válvulas concluiu a execução e envia o comando "`all_vc_end`" para o controlador do GMP.

Os modelos modificados do controlador do GMP (ou IUT) e o gerador aleatório de comandos estão apresentados, respectivamente, nas figuras 8.9 e 8.10.

Para estar de acordo com a proposta apresentada na Figura 8.3, foram determinados que os canais de entrada na IUT são: "`upd_gmp_off`", "`upd_gmp_pull`", "`upd_gmp_push`",

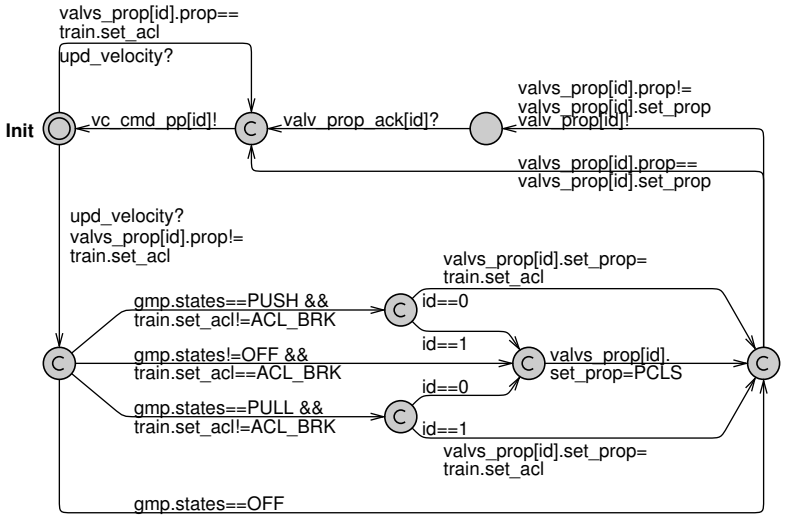


Figura 8.7 Modelo de controladores de válvulas proporcionais para uso de interrupções

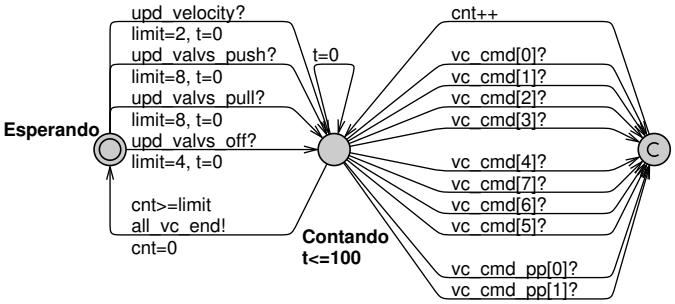


Figura 8.8 Modelo que verifica se o comando foi concluído pelo conjunto de válvulas

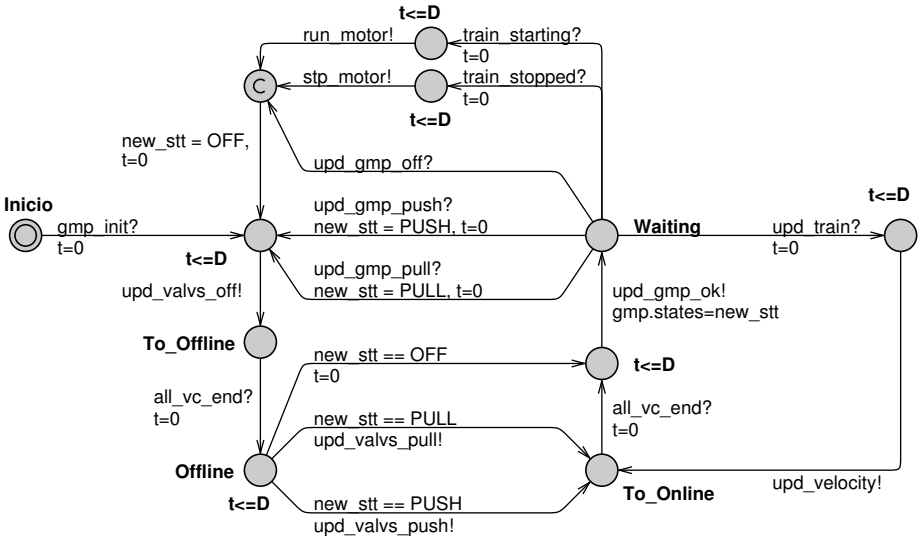


Figura 8.9 Modelo do controlador do GMP

"`all_vc_end`", "`gmp_init`", "`train_starting`", "`train_stopped`" e "`upd_train`". Os canais de saída do IUT (*Implementation Under Test*) são "`upd_valvs_off`", "`upd_valvs_pull`", "`upd_valvs_push`", "`upd_gmp_ok`", "`upd_velocity`", "`run_motor`" e "`stp_motor`". Na Figura 8.11, é apresentada a relação entre modelos, sinais de sincronização

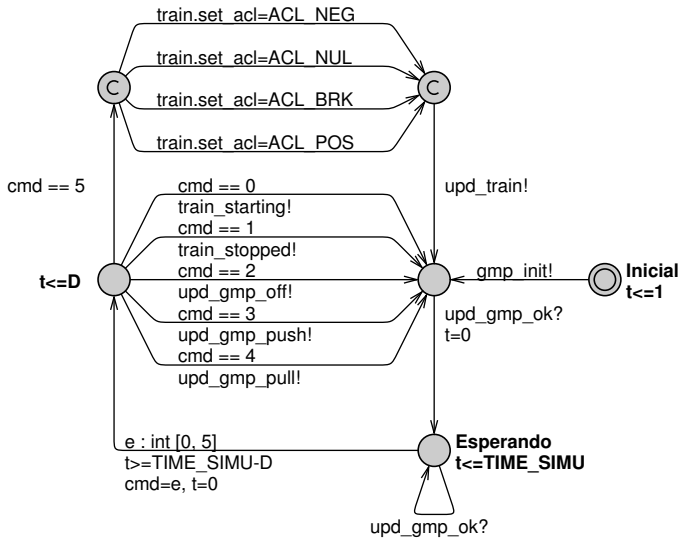


Figura 8.10 Modelo de geração aleatório de comandos para o GMP

e variáveis compartilhadas em uma rede de autômatos temporizados de forma a dimensionar o grau de complexidade existente no controle do GMP.

Esta avaliação gráfica permite a verificação de coerência entre os modelos simulados e a IUT através da avaliação dos canais de comunicação e variáveis acessadas por cada um dos elementos da rede de autômatos temporizados. Por exemplo, o GMP não poder utilizar uma variável compartilhada do estado da válvula visto que são processos diferentes. As trocas de informações devem ser realizadas através de canais de comunicação ou variáveis associadas a canais de comunicação. Observa-se que a IUT pode ser um modelo ou um conjunto de modelos em autômatos temporizados, o que pode, eventualmente, devido a restrições inertes à ferramenta UPPAAL quando trabalha com escopos de variáveis, mascarar uma falha de acesso a informações de outros autômatos, o que não é identificável através dos métodos de simulação e verificação formal.

Uma vez realizadas as modificações, a verificação formal do sistema é novamente requerida. Os comportamentos testados são descritos em linguagem natural e formal de acordo com o padrão de entrada para o verificador de modelos UPPAAL. Foi utilizada a representação em espaço de estados DBM (*Difference Bounded Matrices*) [Dill, 1989]. Os itens verificados estão descritos nas tabelas 8.1 e 8.2.

Todas as propriedades observadas na Tabela 4.1 foram verificadas através do processamento em um PC Intel® Core™ 2 Duo CPU 2.10 GHz (4 Gb RAM) em cerca de 160 minutos.

Sendo o conjunto dos modelos aprovados nos testes de verificação formal, o controlador do GMP foi traduzido manualmente para código C++ e foi produzida uma biblioteca para

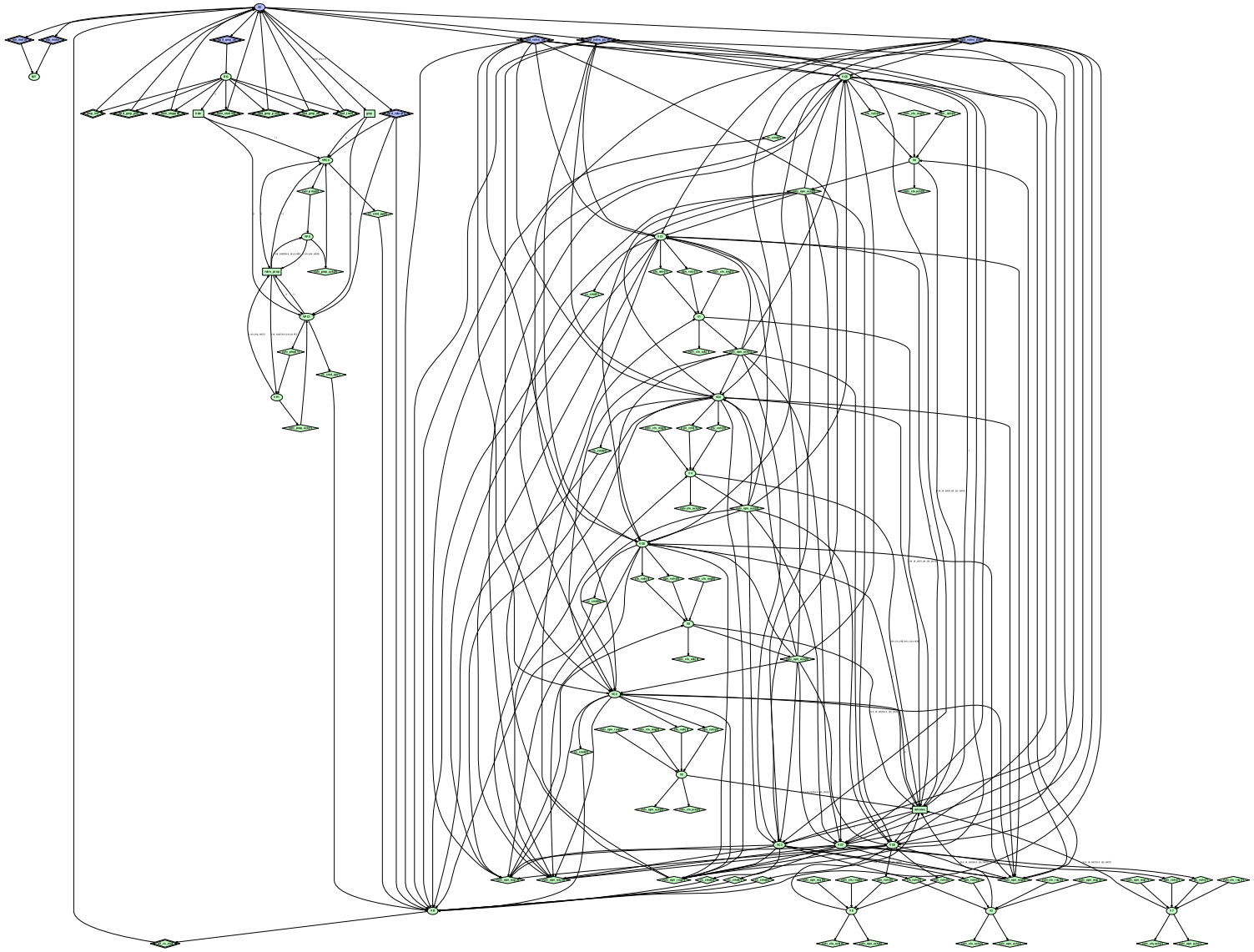


Figura 8.11 Relação entre os modelos do GMP e a IUT

Tabela 8.1 Verificação das propriedades do sistema GMP modificado

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
Verifica se as válvulas proporcionais são modificáveis	`E<> VP0.Changing`	Sim
	`E<> VP1.Changing`	Sim
Verifica a possibilidade de o GMP atingir a configuração de PUSH, PULL e OFF	`E<> (V0.Opened && V1.Opened && V2.Opened && V3.Opened ) && (V4.Closed && V5.Closed && V6.Closed && V7.Closed)`	Sim
	`E<> (V0.Closed && V1.Closed && V2.Closed && V3.Closed) && (V4.Opened && V5.Opened && V6.Opened && V7.Opened)`	Sim
	`E<> (V0.Opened && V5.Opened && V1.Closed && V4.Closed)`	Sim
	`E<> (GC.Waiting && gmp.states == PUSH)`	Sim
	`E<> (GC.Waiting && gmp.states == PULL)`	Sim
	`E<> (GC.Waiting && gmp.states == OFF)`	Sim
	`E<> GC.new_stt == PUSH`	Sim
	`E<> GC.new_stt == PULL`	Sim
	`E<> GC.new_stt == OFF`	Sim
Verifica a possibilidade da válvula 4 estiver fechada enquanto a 0 está aberta	`E<> V4.Closed and V0.Opened`	Sim
Verifica a possibilidade da válvula 1 estiver fechada enquanto a 5 está aberta	`E<> V1.Closed and V5.Opened`	Sim
Verifica a impossibilidade das válvulas 1 e 5 estiverem fechadas	`A[] not (V1.Closed and V5.Closed)`	Sim
Verifica a impossibilidade das válvulas 0 e 4 estiverem fechadas	`A[] not (V0.Closed and V4.Closed)`	Sim
Verifica a impossibilidade de o sistema entrar no estado de intertravamento	`A[] not deadlock`	Sim

Tabela 8.2 Verificação das propriedades do sistema GMP modificado (continuação)

Descrição Informal	**Descrição Formal**	**Propriedade Satisfeita**
Verifica se o estado de PUSH é atingido corretamente	`A[] (GC.Waiting && gmp.states == PUSH) imply (((V0.Closed || V0.To_Close)&& (V1.Closed || V1.To_Close) && (V2.Closed || V2.To_Close) && (V3.Closed || V3.To_Close)) && ((V4.Opened || V4.To_Open) && (V5.Opened || V5.To_Open) && (V6.Opened || V6.To_Open) && (V7.Opened || V7.To_Open)))`	Sim
Verifica se o estado de PUSH é atingido corretamente	`A[] (GC.Waiting && gmp.states == PULL) imply (((V0.Opened || V0.To_Open) && (V1.Opened || V1.To_Open) && (V2.Opened || V2.To_Open) && (V3.Opened || V3.To_Open)) && ((V4.Closed || V4.To_Close) && (V5.Closed || V5.To_Close) && (V6.Closed || V6.To_Close) && (V7.Closed || V7.To_Close)))`	Sim
Verifica se o estado de *offline* é atingido corretamente	`A[] (GC.Waiting && gmp.states == OFF) imply ((V0.Opened) && (V1.Closed || V1.To_Close) && (V5.Opened) && (V4.Closed || V4.To_Close))`	Sim
Verifica se o estágio de *offline* é atingido corretamente	`A[] (GC.Offline) imply ((V0.Opened) && (V1.Closed || V1.To_Close) && (V5.Opened) && (V4.Closed || V4.To_Close))`	Sim
Verifica se quando o modelo de verificação de conclusão de comando das válvulas estiver aguardando novo comando, obrigatoriamente todos os controladores das válvulas devem estar sem processar comandos de mudança de estado	`A[] VR.Esperando imply (VC0.Inicio && VC1.Inicio && VC2.Inicio && VC3.Inicio && VC4.Inicio && VC5.Inicio && VC6.Inicio && VC7.Inicio)`	Sim

comunicação com o UPPAAL TRON durante os testes de “caixa-preta”. O trecho de código que representa o GMP está descrito na Figura 8.12 e pode ser comparado com o modelo apresentado na Figura 8.9:

```
int32_t cnt = 0;
bool vel_updated = false;
int32_t gmp_states = OFF;
wait_chan(gmp_init);
while(!stop) {
  if (!vel_updated) {
    rep_to(upd_valvs_off);
    if (!wait_chan(all_vc_end))
      break;
    if (gmp_states != OFF) {
      if (gmp_states == PUSH)
  rep_to(upd_valvs_push);
      else if (gmp_states == PULL)
  rep_to(upd_valvs_pull);
      if (!wait_chan(all_vc_end))
  break;
    }
  } else
    vel_updated = false;
  rep_to(upd_gmp_ok, 1, &gmp_states);
  waitUntilInput();
  if (stop) break;
  if (input.front() == ins[upd_gmp_push]) {
    gmp_states = PUSH;
  } else if (input.front() == ins[upd_gmp_pull]) {
    gmp_states = PULL;
  } else if (input.front() == ins[upd_gmp_off]) {
    gmp_states = OFF;
  } else if (input.front() == ins[train_starting]) {
    rep_to(run_motor);
    gmp_states = OFF;
  } else if (input.front() == ins[train_stopped]) {
    rep_to(stp_motor);
    gmp_states = OFF;
  } else if (input.front() == ins[upd_train]) {
    input.pop_front();
    rep_to(upd_velocity);
    wait_chan(all_vc_end);
    vel_updated = true;
  }
  if (!input.empty())
    input.pop_front();
}
```

Figura 8.12 Trecho de código C++ representativo do modelo de controle do GMP

Os testes foram realizados com sucesso para uso de relógio virtual com unidade de tempo de 10000 μs e tempo limite de 6000 unidades de tempo, resultando em aproximadamente 35 ciclos executados por teste. Os atrasos foram determinados como aleatórios nos casos em que as mudanças de estado permitem intervalo de tempo variável para execução. Como base comparativa, tem-se que a mudança do estado do motor máximo utiliza, no máximo, 10 unidades de tempo que o tempo máximo para a mudança de estado das válvulas proporcionais é de 20 unidades de tempo.

O relógio de tempo real (*soft real-time*) do UPPAAL TRON utiliza escalonamento do tipo *round-robin*, podendo chegar, no sistema operacional GNU/Linux (utilizado para o

desenvolvimento do trabalho) a um atraso de cerca de 1 ms. Os testes com relógio de tempo real foram executados com sucesso, tendo sido considerados 300 µs de atraso nos sinais de entrada e saída.

Os testes de relógio de tempo real (testes de desempenho) resultaram em alterações importantes nos modelos, principalmente com relação aos estados *urgent*, *committed* que usam invariantes restritas. Estes estados foram modificados para uso de intervalos de tempo, ou seja, as invariantes apresentadas "`t<=D`", como pode ser observado na Figura 8.9, foram inseridas para representar a incapacidade de sistemas reais de receber e enviar uma mensagem em um intervalo de tempo nulo. Observa-se que estes intervalos não são necessários no código fonte, sendo usados somente no modelo para que o UPPAAL TRON não identifique como um erro a impossibilidade de executar transições acima do erro considerado de 300 µs. Testes mais precisos podem ser realizados com o uso de sistemas operacionais *hard real-time*, como o RTAI [DIAPM, 2005] utilizado, como, por exemplo, em Kunz, 2006.

Se a necessidade de uso de um intervalo de tempo superior ao utilizado no modelo for identificado na implementação do IUT, o mesmo atraso deve ser inserido no modelo seguindo as etapas de simulação e verificação formal. Os intervalos de tempo inseridos nos modelos aproximam os mesmos do sistema real, porém dificultam a verificação formal por necessitam de maior capacidade computacional para processar as diferentes possibilidades de tempo de transição entre estados.

8.3 Conformidade do Sistema de Envio e Recebimento de Mensagens GOOSE

O teste de conformidade foi realizado seguindo a arquitetura apresentada na Figura 8.13. Os modelos utilizados seguem a mesma estrutura dos apresentados no Capítulo 5, porém com um emissor e um receptor de mensagens GOOSE. O modelo de barramento, neste caso, não é implementado por utilizar a infra-estrutura da rede *ethernet*. Neste caso, a IUT trata de um conjunto de modelos de autômatos temporizados. Conforme descrito na Figura 8.13, as mensagens de entrada são "`LD_LLN0_GoCB_SendGOOSEMessage`" e "`LD_LLN0_GoCB_SetGoCBValues`" e as mensagens de saída são "`gse_fail`" (sinal de que houve falha no envio) e "`gse_updt`" (sinal de que houve atualização de estado do equipamento observado).

A relação entre sinais de entrada e saída, assim como as variáveis lidas e escritas estão apresentadas na Figura 8.14, indicando os modelos selecionados como IUT.

Para o envio de mensagens, foi implementada em linguagem C++ a formatação do

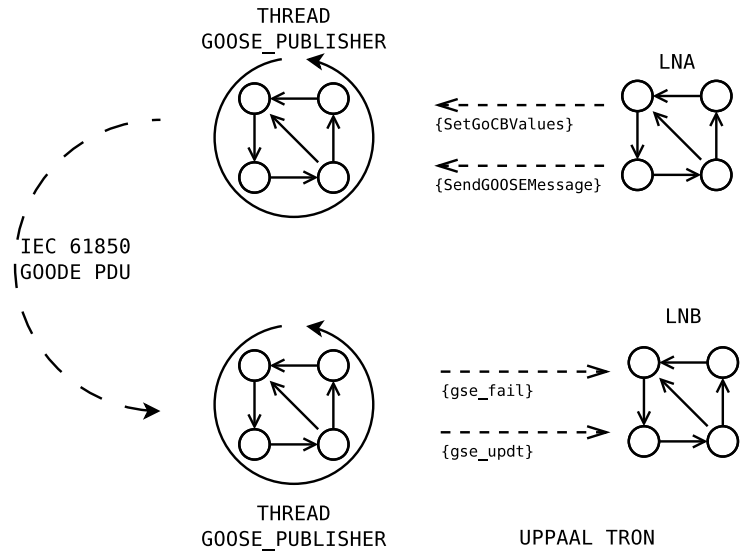


Figura 8.13 Arquitetura de teste do sistema de comunicação IEC 61850 (GOOSE)

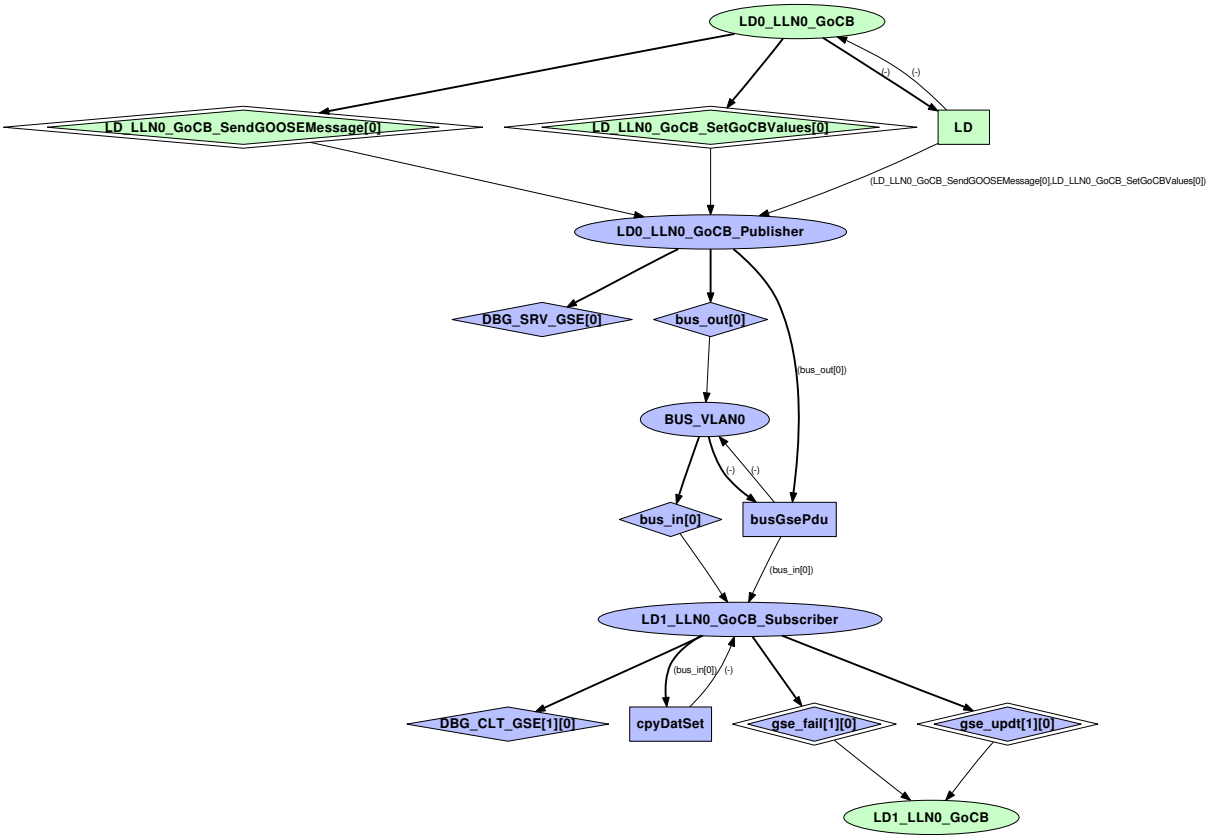


Figura 8.14 Relação entre modelos, sinais e variáveis envolvidos no teste de conformidade

pacote de dados para envio e recebimento das mensagens. Este código apenas monta o pacote, ficando a lógica de controle da comunicação a cargo dos modelos para envio e recebimento de mensagens GOOSE. A estrutura das mensagens foi desenvolvida através da ferramenta de conversão asn1c [asn1c, 2010].

Uma vez que não há atraso no modelo de barramento no teste realizado, por tratar-se

de envio de mensagens pela rede local, o mesmo foi inserido do código da IUT. A seguir, na Figura 8.15, é apresentada a tradução do modelo para envio de dados GOOSE. Este procedimento comunica-se com os dos nós lógicos implementados no UPPAAL através do serviço "SendGOOSEMessage", de acordo com a norma IEC, 2003d.

```
while (!stop_all) {
  if (ft) {
    if (tron_cond_wait(&setGoCBValues_c, &publish_m)==0) {
      if (pData->pGoCB_Class.GoEna == true && !stop_all) {
  gMsg.stNum = 1;
  gMsg.sqNum = 1;
  gMsg.Test = pData->pGoCB_Class.GOOSEData.Value;
  gMsg.timeAllowedtoLive = 2;
  gMsg.GoCBRef = pData->pGoCB_Class.GoCBRef;
  pData->pGoCB_Class.pServiceGoose->SendGOOSEMessage(gMsg.GoCBRef,
    gMsg.stNum, gMsg.sqNum, gMsg.Test, gMsg.timeAllowedtoLive);
  ft = false;
      }
    }
  } else {
    if (pData->pGoCB_Class.GoEna == true) {
      tron_gettime(&timeout);
      timeout += gMsg.timeAllowedtoLive*TUNT;
      int ret = tron_cond_timedwait(&sendGOOSEMessage_c, &publish_m, &timeout);
      if (!stop_all) {
  if (ret==0) {
    gMsg.sqNum    = 0;
    gMsg.timeAllowedtoLive  = 2;
    if (gMsg.stNum<INT32U)
      gMsg.stNum++;
    else
      gMsg.stNum=1;
    pData->pGoCB_Class.pServiceGoose->SendGOOSEMessage(gMsg.GoCBRef,
      gMsg.stNum, gMsg.sqNum, gMsg.Test, gMsg.timeAllowedtoLive);
  } else {
    if (gMsg.timeAllowedtoLive <= TTL_GSE)
      gMsg.timeAllowedtoLive = gMsg.timeAllowedtoLive*2;
    if (gMsg.sqNum>=INT32U)
      gMsg.sqNum=1;
    else
      gMsg.sqNum++;
    pData->pGoCB_Class.pServiceGoose->SendGOOSEMessage(gMsg.GoCBRef,
      gMsg.stNum, gMsg.sqNum, gMsg.Test, gMsg.timeAllowedtoLive);
  }
      }
    }
  }
}
```

Figura 8.15 Código em C++ para o modelo de envio de dados GOOSE

Por sua vez a tradução manual do modelo de recebimento de pacotes GOOSE seguindo especificação IEC, 2003d, está listada abaixo, na Figura 8.16.

Os testes de tempo real, utilizando escalonamento do tipo *round-robin*, foram executados com sucesso, tendo sido considerados 300 μs no atraso nos sinais de entrada e saída. As mensagens foram devidamente identificadas pelo aplicativo MMS Ethereal [NettedAutomation, 2011] que intercepta as mensagens que trafegam na rede registrando seu conteúdo e instante ocorrência. Como pode ser observado através da Figura 8.17, o intervalo de envio e conteúdo da mensagem seguem o comportamento

```
1  while (!stop_all){
2    if (subscriberGoose.GetGoReference(newGo.GoCBRef, newGo.stNum, newGo.sqNum,
3      newGo.Test, newGo.timeAllowedtoLive) == true) {
4      if (newGo.GoCBRef.compare(AppID)) {
5        if (first_time==true) {
6    tron_gettime(&tnew);
7    first_time = false;
8        } else {
9    tron_gettime(&tnew);
10   int64_t result = tnew.tv_sec - told.tv_sec;
11   result *= 1000000L;
12   result += (tnew.tv_nsec - told.tv_nsec)/1000L;
13   result = result/TUNT;
14   if (result>oldGo.timeAllowedtoLive+1)
15     bGseFail = true;
16   else
17     if(newGo.sqNum == 0 && ((newGo.stNum == oldGo.stNum+1) ||
18       (oldGo.stNum == INT32U && newGo.stNum == 1)))
19       bGseUpdt = true;
20     else if((oldGo.stNum != newGo.stNum && newGo.sqNum != 0) ||
21       (oldGo.stNum == newGo.stNum &&
22       ((newGo.sqNum != oldGo.sqNum+1 && oldGo.sqNum != INT32U) ||
23       (newGo.sqNum != 1 && oldGo.sqNum == INT32U))))
24       bGseFail = true;
25       }
26       oldGo = newGo;
27       told = tnew;
28     }
29   }
30 }
```

Figura 8.16 Código em C++ para o modelo de recebimento de pacotes GOOSE

apresentado na Figura 5.13.

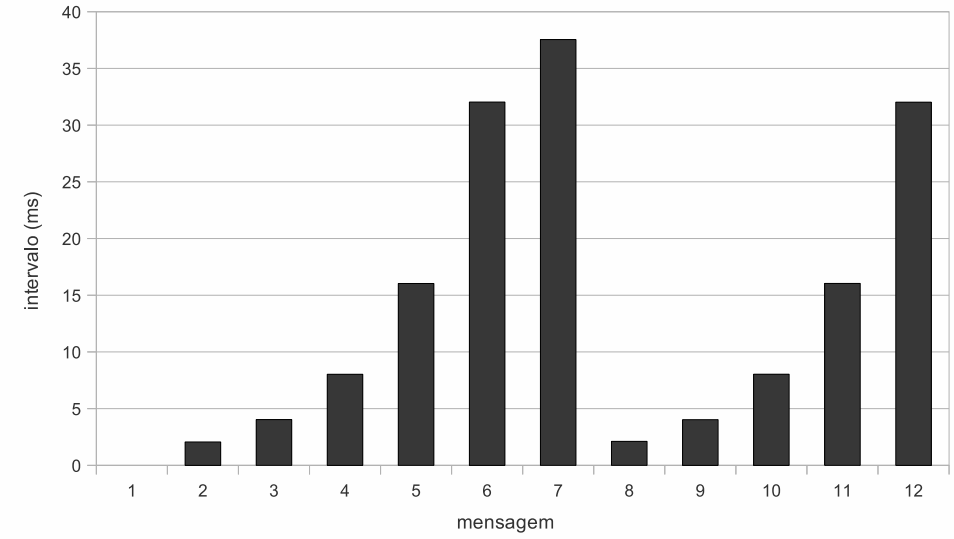


Figura 8.17 Intervalo de tempo entre mensagens GOOSE

Duas das mensagens interceptadas estão apresentadas na Figura 8.18, onde o conteúdo do pacote de dados pode ser observado. Na linha 10 está indicado o identificador do emissor "`G/LLN0$GO$a`", ou seja, o dispositivo lógico "`G`". Através de um serviço da classe "`GO`" do nó lógico "`LLN0`", estão sendo enviandas informações do nó lógico "`A`", pertencente ao mesmo dispositivo lógico. Na linha 11 está indicado o valor de "`timeAllowedtoLive`", que mostra

em quanto tempo o receptor deve receber a próxima mensagem. Observa-se que, no pacote seguinte, este valor duplica até atingir um intervalo constante máximo pré determinado. Ao ocorrer mudança de estado do equipamento, este valor volta para 2 ms.

As linhas 15 e 16 mostram, respectivamente, os valores de "stNum" e "sqNum". Observa-se que "sqNum" é incrementado a cada mensagem enviada e "stNum" é incrementado somente na mudança de estado do equipamento.

```
1  802.1Q Virtual LAN
2  IEC 61850 GOOSE
3   AppID*: 0
4   PDU Length*: 55
5   Reserved1*: 0x0000
6   Reserved2*: 0x0000
7   PDU
8    IEC GOOSE
9    {
10     Control Block Reference: G/LLN0$GO$a
11     Time Allowed to Live (msec): 32
12     DataSetReference:
13     Event Timestamp:
14       2012-03-23 03:45.17,851437
15     StateNumber: 1
16     SequenceNumber: 5
17     Test: FALSE
18     Config Revision*: 0
19     Number Dataset Entries:  0
20     {
21     }
22   }
```

```
1  802.1Q Virtual LAN
2  IEC 61850 GOOSE
3   AppID*: 0
4   PDU Length*: 55
5   Reserved1*: 0x0000
6   Reserved2*: 0x0000
7   PDU
8    IEC GOOSE
9    {
10     Control Block Reference: G/LLN0$GO$a
11     Time Allowed to Live (msec): 64
12     DataSetReference:
13     Event Timestamp:
14       2012-03-23 03:45.18,171776
15     StateNumber:1
16     SequenceNumber:  6
17     Test: FALSE
18     Config Revision: 0
19     Number Dataset Entries:  0
20     {
21     }
22   }
```

Figura 8.18 Conteúdo de pacotes GOOSE

8.4 Etapas de Desenvolvimento

Os testes de conformidade para os atuadores e sensores devem ser realizados sem restrições e em conjunto do uso de geradores aleatórios de comandos. Para os controladores, deve-se primeiramente determinar e avaliar a capacidade dos equipamentos onde os controles do sistema APM serão implementados. Tanto para os equipamentos quanto para os controladores sugere-se as seguintes etapas de desenvolvimento dos testes de conformidade:

1. isolamento de variáveis tem como objetivo avaliar a coerência no acesso a variáveis ou constantes pertencentes à rede de autômatos temporizados.
2. validar o sequenciamento de ações da implementação dos modelos dos controladores através do uso de relógio virtual.
3. validar o desempenho da implementação dos modelos dos controladores através do uso de relógio real.
4. em caso de mudança na rede de autômatos, retornar para as etapas de simulação e verificação formal.

Desta forma acrescenta-se aos itens descritos na Seção 7.2 conforme observado na Figura 8.19.

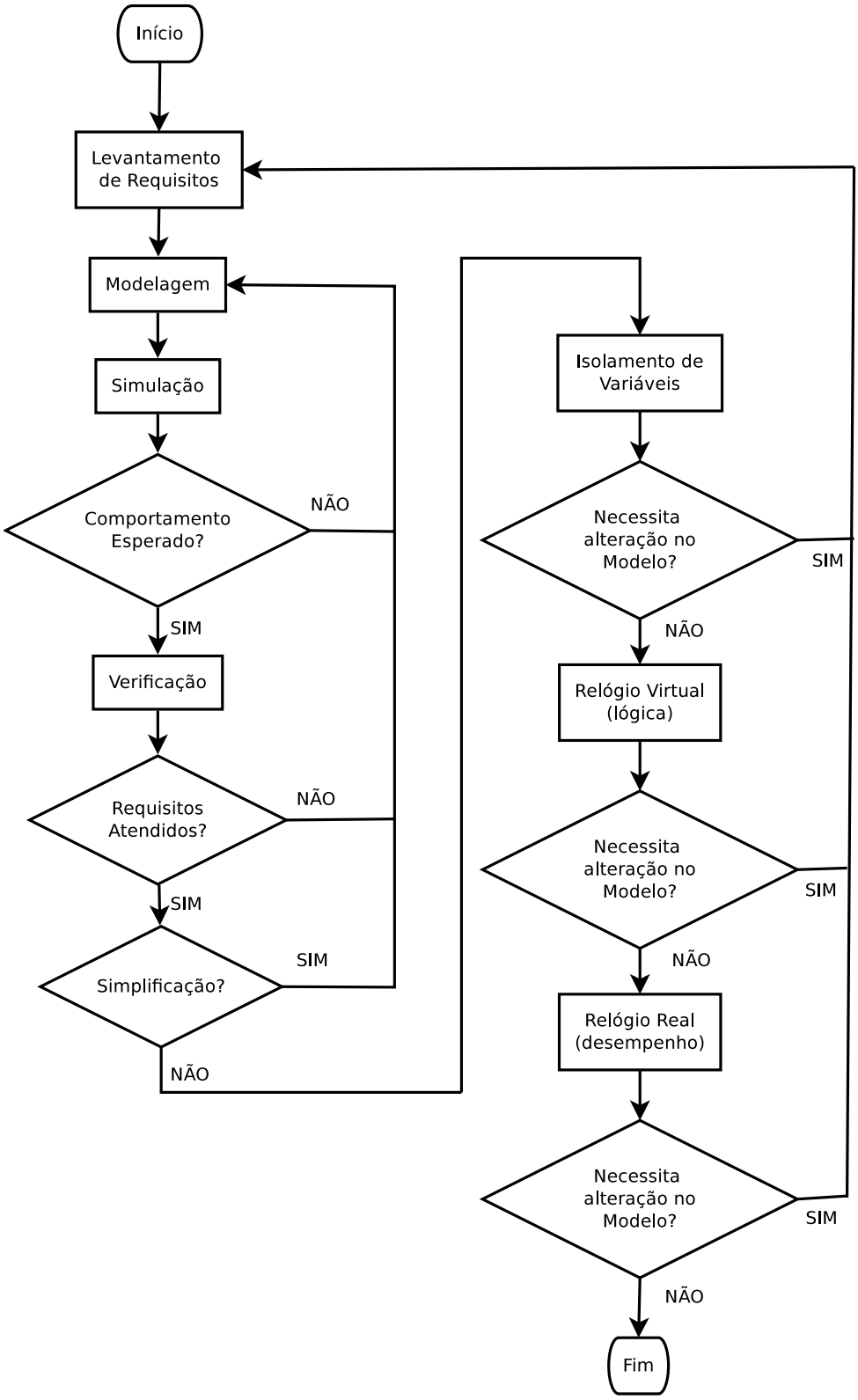


Figura 8.19 Relação entre as etapas de simulação, verificação formal e testes de conformidade

9 CONCLUSÃO

9.1 Resultados Atingidos

O objetivo geral deste trabalho foi a proposição de uma metodologia que garanta o projeto efetivo de controladores para utilização em sistemas de proteção, operação e supervisão de sistemas APM (*Automated People Movers*), utilizando como estudo de caso o Sistema Aeromovel, incluindo os seguintes objetivos específicos:

- Determinação do estudo de caso para aplicação da metodologia;
- Análise das características físicas e de controle do sistema Aeromovel para aplicação da metodologia proposta;
- Utilização da Perspectiva Funcional/Estrutural para aplicação no Sistema Aeromovel;
- Determinação de ferramentas para desenvolvimento e validação das propostas de modelo de comunicação, controle, proteção e equipamentos mecânicos;
- Validação dos modelos de comunicação, controle e proteção para aplicação no Sistema Aeromovel;
- Avaliação da norma IEC 61850 quando aplicada em sistemas APM;
- Ampliação normalização envolvida para aplicação em Sistemas APM (IEC 61850);
- Verificação formal e a simulação em busca das condições de falha no sistema;
- Verificação da viabilidade de aplicação da metodologia proposta para outras tecnologias de sistemas APM.

Assim, para alcançar os objetivos propostos, foi inicialmente realizada uma revisão bibliográfica que abrange a análise de soluções para o controle digital em tempo real, os componentes pneumáticos, a norma IEC 61850, a modelagem de sistemas automáticos, os conceitos relacionados a autômatos temporizados, o sistema Aeromovel de transporte de passageiros e sistemas APM, incluindo uma análise sobre as normas vigentes.

Na pesquisa sobre as principais soluções para a construção e controle mais utilizadas para sistemas APM, identificou-se a necessidade do desenvolvimento de uma metodologia que garanta o projeto efetivo de controladores para utilização em sistemas de proteção, operação e supervisão de sistemas APM (*Automated People Movers*). Também, como resultado da revisão bibliográfica, foi verificada a necessidade de precisão temporal, distribuição do controle e de um

sistema de comunicação (protocolos) e organização (hierarquia e nomenclatura) normatizados para sistemas APM. Este requisito foi resolvido com o uso da norma IEC 61850. Foi proposta uma ampliação da norma IEC 61850, que originalmente é aplicada na geração e distribuição energia, para aplicação em sistemas APM. Além disso, foi definido o uso de autômatos temporizados para modelagem do sistema físico, de comunicação e controle. A ferramenta selecionada para modelagem, simulação e verificação formal foi a UPPAAL e duas de suas extensões: UPPAAL TIGA (algoritmo *on-the-fly*) e UPPAAL TRON (testes de conformidade em tempo real).

Utilizando o sistema Aeromovel como foco de estudo de caso, foi realizado o projeto da estrutura física de comunicação e controle, incluindo os sistemas de interfaceamento dos sensores com a central de processamento, os processos de controle e monitoramento do sistema. A representação desta estrutura foi apresentada a partir da perspectiva funcional/estrutural. Alguns elementos da estrutura de comunicação foram testados em laboratório.

Na sequência, foram desenvolvidos os modelos em autômatos temporizados da estrutura física do grupo moto propulsor do sistema Aeromovel, seguidos da simulação e verificação formal. Os modelos de controle (proteção e operação) do grupo moto propulsor foram modelados e, em conjunto com os modelos da estrutura física, foram simulados e verificados formalmente. Os modelos do sistema Aeromovel foram baseadas em trabalhos acadêmicos, não representando a estrutura utilizada para controle de propulsão do veículo em suas versões comerciais, porém, considerados válidos para análise metodológica. Como resultado desta etapa, foi proposta uma simplificação do modelo dos sistema GMP, tendo em vista os recursos computacionais necessários para verificação formal de sistemas complexos através de autômatos temporizados.

Foi determinada uma estrutura para teste dos veículos, onde pode-se avaliar os sistemas de proteção e operação de dois veículos com duas estações. Estes modelos foram verificados formalmente em conjunto com o modelo simplificado do grupo moto propulsor. Dentro deste escopo, foram também, modelados, simulados e verificados formalmente o sistema de comunicação de tempo real da norma IEC 61850 (serviços GOOSE e SMV). Para verificação formal foram utilizados os testes de conformidade normatizados. Os modelos demonstraram comportar-se de acordo com a especificação.

A validação dos modelos de comunicação IEC 61850 e de operação do veículo e GMP foi realizada demonstrando que a estrutura ampliada da IEC 61850 pode ser aplicada a sistemas APM. Como resultado foram apresentados os nós lógicos para aplicação da norma IEC 61850 em Sistemas APM de acordo com os modelos em autômatos temporizados e as etapas de

desenvolvimento.

Por último, tendo em vista que a simulação e a verificação formal dos autômatos temporizados não dão garantias de que os modelos propostos, de acordo com os requisitos solicitados, serão traduzidos corretamente em uma linguagem de programação que permita o interfaceamento com os dispositivos a serem controlados, foram implementadas em linguagem de programação C++ tanto, a estrutura de comunicação GOOSE, quanto o sistema de controle do conjunto de válvulas do GMP. Os programas foram interfaceados com os modelos da rede de autômatos temporizados, verificando assim a sua conformidade em termos de funcionalidade e desempenho.

Os testes de conformidade, quando na avaliação de desempenho, também auxiliaram na correção dos modelos dos controladores uma vez que algumas operações possíveis nos modelos em autômatos temporizados não são possíveis de serem implementadas em um controlador real, assim indicando a necessidade de correção dos modelos de controladores.

A metodologia para desenvolvimento de controladores demonstrou-se eficiente para sistemas APM principalmente com relação aos seguintes itens:

Econômico. O sistema resultante demonstrou ser economicamente atrativo devido à redução de cabeamento, custo e tempo de instalação.

Tecnológico. Todos os aplicativos utilizados são de livre utilização, evitando, assim, a dependência de tecnologias fechadas.

Confiabilidade. Capacidade de simular e verificar (testes de conformidade) desde a concepção até a aplicação.

Padronização. Utilização de protocolo de comunicação aberto, proporcionando interoperabilidade. Hierarquia de dados e nomenclatura padronizadas.

Depuração. Aumento da capacidade de monitoramento dos sistemas de controle e proteção.

Macro. Infra-estrutura separada da funcionalidade. Capacidade de uma avaliação macroscópica do sistema de atuação, medição e troca de informações e disponibilização de uma visão detalhada do controle.

Flexibilidade. Possibilidade de aplicação em diferentes tecnologias de sistemas APM.

Modularidade. O sistema desenvolvido é modular, podendo ser dividido entre diferentes centrais de processamento e monitoramento.

Conformidade. Auxilia na verificação de conformidade com os requisitos determinados pelas

normas aplicáveis aos Sistemas APM.

Conclui-se que o trabalho atingiu as metas propostas de desenvolver uma metodologia para controle de sistemas APM com segurança. Por ser uma solução flexível, o sistema desenvolvido pode ser utilizado em sistemas APM de outras tecnologias, colaborando com o desenvolvimento e a disseminação de sistemas APM através da diminuição de custos, mantendo a robustez das soluções.

9.2 Sugestões para Trabalhos Futuros

Pesquisas futuras sobre o desenvolvimento de controladores para sistemas APM deverão abordar os seguintes itens:

- Detalhamento e ampliação dos nós lógicos e modelos de autômatos temporizados para os demais componentes dos sistemas ATP, ATO e ATS.
- Implementação dos modelos para aplicação em CLP seguido da aplicação dos testes de conformidade de lógica e desempenho.
- Projeto e construção de bancada de testes integrada aos modelos implementados em CLPs.
- Proposição de novas situações onde seja possível avaliar o comportamento dos controladores com maior quantidade de veículos, incluindo desvios e manutenção.
- Concepção de modelos mais fidedignos da configuração de válvulas utilizados comercialmente no sistema Aeromovel.
- Uso de sistemas híbridos, possibilitando a integração de modelos de sistemas contínuos (ex.: variação de pressão) com sistemas discretos (ex.: sistema de comunicação).
- Implementação das soluções em outras tecnologias de sistemas APM que possibilitem maior proximidade dos veículos na via e aplicação de blocos móveis.
- Análise de confiabilidade, incluindo geração de árvore de falhas.
- Geração automática de testes de conformidade dos sistemas implementados.
- Desenvolvimento de um simulador digital de tempo real com a capacidade de simular diferentes configurações de rede de deslocamento e de comunicação entre os equipamentos de controle e os respectivos sensores e atuadores presentes em sistemas APM. Seguindo as etapas apresentadas no trabalho, pode-se executar a ação de cada elemento do sistema através de um *cluster* onde cada CPU executa uma instância de

código baseado no modelo de autômato temporizado e validado pelo UPPAAL TRON, permitindo que possa trocar componentes simulados por equipamentos reais durante a simulação e, assim, validar individualmente cada componente do sistema antes de sua aplicação.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. **Sistema Movimentador Automático de Pessoas (Sistema APM) - Parte 1: Terminologia, símbolos e abreviaturas**. 2012a.

ABNT. **Sistema Movimentador Automático de Pessoas (Sistema APM) - Parte 2: Ambiente de operação, dependabilidade, segurança e sistema de comunicação de áudio e vídeo**. 2012b.

ABNT. **Sistema Movimentador Automático de Pessoas (Sistema APM) - Parte 3: Requisitos do Sistema de Controle Automático (ATC)**. 2012c.

Aeromovel. **Aeromovel System Technical Specification**. 1999.

Alcatel. **Integration of Wireless Network Technology with Signaling in the Rail Transit Industry**. 2003.

Aldrich, M. **Software Control For an APM System**. 2005.

Alur, R.; Dill, D. L. The theory of timed automata. In: **REX Workshop**. 1991. p. 45–73.

Amendola, A.; Frederiksen, G. S.; Impagliazzo, L.; Mokkapati, C. The copenhagen metro atc system and the use of fault injection experiments. **Automated People Mover Systems**, 2007.

Andrighetto, P. L.; Valdiero, A. C.; Carlotto, L.; Harter, I. I. Desenvolvimento de um manipulador robótico acionado pneumaticamente. **IX Seminário de Automação e Processos**, out. 2005.

Andrighetto, P. L.; Valdiero, A. C.; Vincensi, C. N. Experimental comparisons of the control solutions for pneumatic servo actuators. 2005.

Ansi. **Automated People Mover Standards, Part 2 (21.2-08)**. 2008a.

asn1c. **Open Source ASN.1 Compiler**. janeiro 2010. Disponível em: <`http://lionet.info/asn1c/`>.

Bailey, P. J. **The Dallas / Fort Worth International Airport APM - Skylink**. 2007.

Behrmann, G.; David, A.; Larsen, K. G. A tutorial on uppaal. **4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT'04)**, LNCS 3185, 2004.

Behrmann, G.; David, A.; Larsen, K. G. **A tutorial on Uppaal 4.0**. 2006.

Bengtsson, J.; Yi, W. Timed automata: Semantics, algorithms and tools. **Lecture Notes on Concurrency and Petri Nets**, LNCS 3098, 2004.

Bobrow, J. E.; Mcdonell, B. W. Modeling, identification, and control of a pneumatically actuated force controllable robot. 1988.

Bowman, H.; Faconti, G. P.; Katoen, J.-P.; Latella, D.; Massink, M. Automatic verification of a lip synchronisation algorithm using uppaal. **In Bas Luttik Jan Friso Groote and Jos van Wamel, editors, In Proceedings of the 3rd International Workshop on Formal Methods for Industrial Critical Systems**, 1998. Amsterdam, The Netherlands.

Branco, A. M. Ferrovia versus rodovia. In: **História Viva**. : Duetto, 2008.

Britto, J. F. F. H. **Modelo Computacional do Sistema Aeromóvel de Transportes**. Dissertação (Mestrado) — UFRGS, 2008.

Bucher, R.; Dozio, L.; Mantegazza, P. Rapid control prototyping with scilab/scicos and linux rtai. **Scilab-2004**, 2004.

Campos, J. C.; Machado, J. Pattern-based analysis of automated production systems. **13th IFAC Symposium on Information Control Problems in Manufacturing**, 2009.

Cao, Y.; Niu, R.; Xu, T.; Tang, T.; Mu, J. Wireless test platform of communication based train control (cbtc) system in urban mass transit. In: **Proc. ICVES Vehicular Electronics and Safety IEEE International Conference on**. 2007. p. 1–4.

Cassez, F.; David, A.; Fleury, E.; Larsen, K. G.; Lime, D. Efficient on-the-fly algorithms for the analysis of timed games. **LNCS 3653**, p. 66–80, 2005.

Conley, J. F. **The San Francisco International Airport AirTrain Project**. 2001.

Cruz, F. B. C. **Modelagem e Controle Não-Lineares de um Posicionador Servopneumático Industrial**. Dissertação (Mestrado) — UFSC, Florianópolis, 2003.

D'argenio, P. R.; Katoen, J.-P.; Ruys, T. C.; Tretmans, J. The bounded retransmission protocol must be on time! **In Proceedings of the 3rd International Workshop on Tools and Algorithms for the Construction and Analysis of Systems**, v. 1217, p. 416–431, 1997.

David, A.; Yi, W. Modelling and analysis of a commercial field bus protocol. **In Proceedings of the 12th Euromicro Conference on Real Time Systems**, p. 165–172, 2000. IEEE Computer Society.

Diapm. **Real-Time Application Interface - Department of Aerospace Engineering of Politecnico di Milano**. 2005. Disponível em: <http://www.rtai.org/>. Acesso em: 05/12/2005.

Dill, D. L. Timing assumptions and verification of finite-state concurrent systems. **LNCS 407**, p. 197–212, 1989.

Dozio, L.; Mantegazza, P. Linux real time application interface in low cost high performance motion control. In: ASSOCIAZIONE NAZIONALE ITALIANA PER L'AUTOMAZIONE. **Motion Control 2003**. 2003.

Edge, K. A. The control of fluid power systems - responding to the challenges. **Proc. Inst. Mechanical Engineering**, v. 211, p. 91–110, 1997.

Eidson, J. C. **Measurement,Control, and Communication Using IEEE 1588**. : Springer, 2006.

Freitas, G. A.; Carmo, U. A. Impacto da norma IEC 61850 na padronização dos dados dos sistemas de supervisão e controle da chesf. **XIII ERIAC: Décimo Tercer Encuentro Regional IberoAmericano de Cigré**, 2009.

Freitas, M.; Mira, J. Apm with full service passenger information. **APM**, 2007.

Furtado, S. M. de L. **Análise Comparativa entre o Aeromóvel e outros sistemas de transporte urbanos guiados automáticos em vias exclusivas elevadas**. Dissertação (Mestrado) — Universidade de Brasília, 1994.

Ghosh, A.; Rana, V.; Johnson, B.; Profeta j.a., I. A distributed safety-critical system for real-time train control. In: **Proc. IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation**. 1995. v. 2, p. 760–767.

Giberti, H.; Righettini, P.; Tasora, A. Design and experimental test of a pneumatic translational 3dof parallel manipulator. In: **10th International Workshop on robotics in Alpe-Adria-Danube region**. 2001.

Granosik, G.; Borenstein, J. Minimizing air consumption of pneumatic actuators in mobile robots. 1998.

Gray, M.; Goldfine, A.; Rosenthal, L.; Carnahan, L. **Conformance Testing**. 2000. Disponível em: <http://xml.coverpages.org/conform20000112.html>.

Guoliang, T.; Xuanyin, W. Research on pneumatic-servo calligraphy robot. 2003.

Havelund, K.; Larsen, K. G.; Skou, A. Formal verification of a power controller using the real-time model checker uppaal. **5th International AMAST Workshop on Real-Time and Probabilistic Systems**, 1999.

Havelund, K.; Skou, A.; Larsen, K. G.; Lund, K. Formal modelling and analysis of an audio/video protocol: An industrial case study using uppaal. **In Proceedings of the 18th IEEE Real-Time Systems Symposium**, p. 2–13, 1997.

Henzinger, T. A. The theory of hybrid automata. In: . : IEEE Computer Society Press, 1996. p. 278–292.

Hessel, A.; Larsen, K.; Mikucionis, M.; Nielsen, B.; Pettersson, P.; Skou, A. Testing Real-Time systems using UPPAAL. In: **Formal Methods and Testing**. 2008. p. 77–117. Disponível em: <http://dx.doi.org/10.1007/978-3-540-78917-8_3>.

Hessel, A.; Larsen, K. G.; Mikucionis, M.; Nielsen, B.; Pettersson, P.; Skou, A. Testing real-time systems using uppaal. **Formal Methods and Testing**, 2008.

Hewings, D. Introduction of integrated protection and control to railway electrification systems. In: **Proc. IET 9th International Conference on Developments in Power System Protection DPSP 2008**. 2008. p. 68–73. ISSN 0537-9989.

Hune, T.; Larsen, K. G.; Pettersson, P. Guided synthesis of control programs using uppaal. **Ten H. Lai, editor, Proc. of the IEEE ICDCS International Workshop on Distributed Systems Verification and Validation**, p. E15–E22, 2000. IEEE Computer Society Press.

IEC. **IEC 61850-3 Communication networks and systems in substations - General requirements**. 2002a.

IEC. **IEC 61850-4 Communication networks and systems in substations - System and project management**. 2002b.

IEC. **IEC 61850-1 Communication networks and systems in substations - Introduction and overview**. 2003a.

IEC. **IEC 61850-5 Communication networks and systems in substations - Communication requirements for functions and device models**. 2003b.

IEC. **IEC 61850-7-1 Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Principles and models**. 2003c.

IEC. **IEC 61850-7-2 Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Abstract communication service interface (ACSI)**. 2003d.

IEC. **IEC 61850-7-3 Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Common data classes**. 2003e.

IEC. **IEC 61850-7-4 Communication networks and systems in substations - Basic communication structure for substation and feeder equipment - Compatible logical node classes and data classes**. 2003f.

IEC. **IEC 61850-9-1 Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Sampled values over serial unidirectional multidrop point to point link**. 2003g.

IEC. **IEC 61850-2 Communication networks and systems in substations - Glossary**. 2004a.

IEC. **IEC 61850-6 Communication networks and systems in substations - Configuration description language for communication in electrical substations related to IEDs**. 2004b.

IEC. **IEC 61850-8-1 Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3**. 2004c.

IEC. **IEC 61850-9-2 Communication networks and systems in substations - Specific Communication Service Mapping (SCSM) - Sampled values over ISO/IEC 8802-3**. 2004d.

IEC. **IEC 61850-10 Communication networks and systems in substations - Conformance testing**. 2005.

IEC. **IEC 61850-7-410 Communication networks and systems for power utility automation - Hydroelectric power plants - Communication for monitoring and control**. 2007.

IEEE. **IEEE standard for passenger information system for rail transit vehicles**. 1998.

IEEE. **IEEE Standard for Communications Protocol Aboard Trains**. 1999a.

IEEE. **IEEE standard for the functioning of and interfaces among propulsion, friction brake, and train-borne master control on rail rapid transit vehicles**. 1999b.

IEEE. **IEEE standard for verification of vital functions in processor-based systems used in rail transit control**. 2000.

IEEE. **IEEE Trial-Use Standart for Message Set Template for Intelligent Transortation Systems**. 2000.

IEEE. **IEEE Standart for the Interface Between the Rail Subsystem and the Highway Subsystem at a Highway Rail Intersection**. 2002.

IEEE. **IEEE standard for user interface requirements in communications-based train control (CBTC) systems**. 2003.

IEEE. **IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements**. 2004.

IEEE. **IEEE Recommended Practice for Communications-Based Train Control (CBTC) System Design and Functional Allocations**. 2008.

IEEE. **IEEE Standard for Communications Protocol Aboard Passenger Trains**. 2011.

ISO. **Industrial Automation systems - Manufacturing Message Specification - 9506**. 2003.

Iversen, T. K.; Kristoffersen, K. J.; Larsen, K. G.; Laursen, M.; Madsen, R. G.; Mortensen, S. K.; Pettersson, P.; Thomasen, C. B. Model-checking real-time control programs: Verifying lego mindstorms systems using uppaal. **Proc. of 12th Euromicro Conference on Real-Time Systems**, p. 147–155, 2000. IEEE Computer Society Press.

Kilian, C. T. **Modern Control Technology - Components and Systems**. : Delmar, 2000.

Kunz, G.; Machado, J.; Perondi, E. Modelling and simulation of IEC 61850 requirements applied to an automated people mover's controller. **ICINCO'2011 - 8th International Conference on Informatics in Control, Noordwijkerhout**, 2011.

Kunz, G.; Perondi, E.; Machado, J. A dependable automated people mover system modeled and verified using timed automata: A case study. In: **21st International Congress of Mechanical Engineering, Natal**. 2011.

Kunz, G.; Perondi, E.; Machado, J. Modeling and simulating the controller behavior of an automated people mover using IEC 61850 communication requirements. In: **INDIN'2011 - IEEE 9th International Conference on Industrial Informatics, Lisboa**. 2011.

Kunz, G. O. **Desenvolvimento de um Sistema de Controle em Tempo Real para um Servoposicionador Pneumático**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2006.

Kuun, E. Open standards for cbtc and cctv radio-based communications. In: **APTA Rail Rail Transit Conference Proceedings**. 2004.

Lafraia, J. **Manual de Confiabilidade, Mantenabilidade e Disponibilidade**. Qualitymark, 2006. ISBN 9788573037920. Disponível em: <http://books.google.com.br/books?id=kfIfwNIfnJQC>.

Lages, W. F.; Alt, G. H. Controle em tempo real de sistemas dinâmicos através de redes ip. In: **3o Congresso Internacional de Automação, Sistemas e Instrumentação**. 2003.

Laplante, P. A.; Woolsey, F. C. Ieee 1473: an open-source communications protocol for railway vehicles. **IT Professional**, v. 5, n. 6, p. 12–16, nov. 2003.

Larsen, K.; Mikucionis, M.; Nielsen, B. Online Testing of Real-time Systems using Uppaal: Status and Future Work. In: Brinksma, E.; Grieskamp, W.; Tretmans, J.; Weyuker, E. (Ed.). **Dagstuhl Seminar Proceedings volume 04371: Perspectives of Model-Based Tes ting**. Schloss Dagstuhl, D-66687 Wadern, Germany.: IBFI gem. GmbH, Schloss Dagstuhl, 2004a. Disponível em: <http://www.cs.aau.dk/~marius/tron/Dagstuhl2004.pdf>.

Larsen, K. G.; Mikucionis, M.; Nielsen, B. Testing real-time embedded software using uppaal-tron: an industrial case study. In: **the 5th ACM international conference on Embedded software**. ACM Press New York, NY, USA, 2005. p. 299 – 306. Disponível em: <http://doi.acm.org/10.1145/1086228.1086283>.

Larsen, K. G.; Mikucionis, M.; Nielsen, B. **Uppaal Tron User Manual**. June 2009.

Larsson, L. **Fourteen Industrial Ethernet solutions under the spotlight**. 2005. The Industrial Ethernet Book. Vol. 28. Disponível em: <http://www.iebmedia.com/index.php?id=4811\&parentid=74\&themeid=255\&hft=28\&showdetail=true>.

Latino, F.; Sandoval, D. Quit overspending for servomotion systems. **Machine Design**, p. 93–96, 1996.

Lindahl, M.; Pettersson, P.; Yi, W. Formal design and analysis of a gearbox controller. **Springer International Journal of Software Tools for Technology Transfer (STTT)**, p. 3(3):353–368, 2001.

Lindau, L. A.; Heineck, L. F.; Pedroso, C. A. Performance of an innovative pneumatic transport system and its possibilities of application: the case of congested central urban areas. **III Conferece on Urban Transport in Developing Countries**, 1986.

Liu, S.; Bobrow, J. E. An analysis of a pneumatic servo system and its application to a computer-controlled robot. 1988.

Liu, X.; Smolka, S. A. Simple linear-time algorithm for minimal fixed. **LNCS 1443**, p. 53–66, 1998.

Lnn, H.; Pettersson, P. Formal verification of a tdma protocol startup mechanism. **Proc. of the Pacific Rim Int. Symp. on Fault-Tolerant Systems**, p. 235–242, 1997.

Machado, J.; Seabra, E.; Campos, J.; Soares, F.; C., L. Safe controllers design for industrial automation systems. **Computers & Industrial Engineering**, v. 60, p. 635–653, 2011.

Matos, L. V. de. **Reprojeto de Equipamentos Mecatrônicos com Base na Análise de Confiabilidade de Sistemas**. Dissertação (Mestrado) — UFSC, 2007.

Menezes, P. B. **Linguagens formais e automatos**. : Instituto de Informática da UFRGS, 2008.

Mikucionis, M.; Larsen, K. G.; Nielsen, B. T-uppaal: Online model-based testing of real-time systems: tool demo. In: **the 19th IEEE International Conference on Automated Software Engineering**. Linz, Austria: , 2004. p. 396–397. Disponível em: <http://www.cs.aau.dk/~marius/tron/ASE2004.pdf>.

Mikucionis, M.; Sasnauskaite, E. **On-the-fly Testing Using UPPAAL**. Dissertação (Mestrado) — Department of Computer Science, Aalborg University, Denmark, June 2003. Disponível em: <http://www.cs.aau.dk/~marius/tron/master.pdf>.

Moreno, J. C.; Laloya, E.; Navarro, J. A link-layer slave device design of the mvb-tcn bus (IEC 61375 and ieee 1473-t). **IEEE Transactions on Vehicular Technology**, v. 56 Issue: 6, 2007.

Negri, V. J. D. **Estruturação da Modelagem de Sistemas Automáticos e sua Aplicação a um Banco de Testes para Sistemas Hidráulicos**. Tese (Doutorado) — UFSC, 1996.

Negri, V. J. D. **Introdução aos Sistemas para Automação e Controle Industrial**. 2004.

Nettedautomation. **The IEC Train Communication Network - IEC 61375**. 2011. Disponível em: <http://www.nettedautomation.com/download/MMS-Analyzer-user-guide.pdf>.

Oliveira, K. de V. **Geração Automática de Testes de Conformidade para Programas de Controladores Lógicos Programáveis**. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2009.

Oliveira Ítalo Romani de. **VERIFICAÇÃO DE SEGURANÇA EM CONFLUÊNCIA DE TRAJETÓRIAS DE AERONAVES UTILIZANDO AUTÔMATOS HÍBRIDOS**. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2003.

Paes, F. H. S.; Negri, V. J. D. **Capacitação Industrial para Construção de Sistemas Hidráulicos de Controladores de Turbinas**. 2002.

Pereira, C.; Kunz, G.; Ataide, F.; Freitas, E.; Teodoro, E.; Carvalho, F. Performance evaluation of a java architecture used in embedded real time systems. In: **ETFA**. 2005.

Perondi, E. A. **Controle não-linear em cascata de um servoposicionador pneumático com compensação de atrito**. Tese (Doutorado) — UFSC, Florianópolis, 2002.

Proenca, H. P. M. C. **MARCS - Sistema Multi-Agente para Controlo de Tráfego Ferroviário**. Tese (Doutorado) — Universidade do Porto, 2003.

Pu, J.; Weston, R. H.; Moore, P. R. Digital motion control and profile planning for pneumatic servos. **ASME Journal of Dynamic Systems, Measurement and Control**, v. 114, p. 634–640, dez. 1992.

Rajendran, S.; Bolton, R. W. Position control of a servopneumatic actuator using fuzzy compensation. In: **Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition**. 2003.

Roychoudhury, A. **Embedded Systems and Software Validation**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009. ISBN 0123742307, 9780123742308.

Ruggedcom. Fevereiro 2010. Disponível em: <http://www.ruggedcom.com>.

Sampaio, L. R. **Validação Visual de Programas Ladder Baseada em Modelos**. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2011.

Sarmanho, C. A. C. **Aplicação do Algoritmo de Gain Schedule Baseado na Estimativa de Massa no Controle do Sistema de Freios do Sistema Aeromovel**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2009.

Schafers, C.; Hans, G. IEC 61375-1 and uic 556 - international standards for train communication. **Vehicular Conference Proceedings**, 2000.

Silva, A. de M. **Aplicação de verificação de modelos a programas de CLP: explorando a temporização**. Dissertação (Mestrado) — Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, 2008.

Souto, R. B. **Projeto de Sistemas Automáticos com Modelagem e Controle da Comunicação com o Ambiente Externo**. Dissertação (Mestrado) — UFSC, 2005.

Sullivan, T. Ieee rail transit vehicle interface standards update. **4th International Conference on Communications Based Train Control**, 2001.

Sullivan, T. The ieee 1473-l communications protocol: Experience in rail transit. **On-Board ITS in Rail Transit**, 2002.

Sullivan, T. Open architecture train control. In: **5th International Conference on Communications Based Train Control**. 2003. Washington, DC.

Sun Microsystems. **The Real-Time Java Platform**. jun. 2004.

Tdc. **A Review of State-of-the-Art Train Control Systems Technology**. 1998.

Vieira, A. D. **Análise Teórico-experimental de Servoposicionadores Lineares Pneumáticos**. Tese (Doutorado), 1998.

Virvalo, T. Designing a pneumatic position servo system. **Power International**, p. 141–147, 1989.

Virvalo, T. Modeling and design of a pneumatic position servo system realized with commercial components. 1995.

Vuchic, V. R. **Urban Transit Systems and Technology**. John Wiley & Sons, Inc., 2007. 444–476 p. Disponível em: <http://dx.doi.org/10.1002/9780470168066.ch7>.

Yelloz, G. Urban mass transit goes driverless. In: **11th International Conference on Automated People Movers**. 2007.

APÊNDICE A – ANEXOS

A.1 Autômatos Temporizados em UPPAAL

Para o uso de autômatos temporais a seguinte semântica é utilizada: C é o conjunto de relógios e $B(C)$ é o conjunto de conjunções sobre simples condições da forma $x \bowtie c$ ou $x-y \bowtie c$, onde $x,y \in C, c \in \mathbb{N}$ e $\bowtie \in \{<,\leq,=,\geq,>\}$. Um autômato temporal é um grafo assíncrono finito sobre condições e reinicializações de relógios de valor real não negativo [Behrmann et al., 2004].

Definição 1 (Autômato Temporal). Um autômato temporal é um sêxtuplo (L, l_0, C, A, E, I), onde L é o conjunto de estados, $l_0 \in L$ é o estado inicial, C é o conjunto de relógios, A é o conjunto ações, $E \subseteq L \times A \times B(C) \times 2^C \times L$ é o conjunto de mudanças entre estados com ações, restrições e o conjunto de relógios a serem reinicializados, e $I : L \to B(C)$ as invariantes atribuídas aos estados [Behrmann et al., 2004].

A valoração dos relógios é uma função $u : C \to \mathbb{R}_{\geq 0}$ do conjunto de relógios para valores reais não-negativos, $\mathbb{R}^C$ é o conjunto de todas valorações dos relógios e $u_0(x) = 0$ para todo $x \in C$. Considera-se guardas e invariantes como o conjunto de valoração de relógios, utilizando $u \in I(l)$ para significar que u satisfaz $I(l)$.

Definição 2 (Semântica do Autômato Temporal). Façamos (L, l_0, C, A, E, I) ser um autômato [Behrmann et al., 2004]. A semântica é definida como um sistema de transição $\langle S, s_0, \to \rangle$, onde $S \subseteq L \times \mathbb{R}_{\geq 0}$ é o conjunto de estados, $s_0 = (l_0, u_0)$ é o estado inicial, e $\to \subseteq S \times (\mathbb{R}_{\geq 0} \cup A) \times S$ é a relação de transição como que:

- $(l,u) \xrightarrow{d} (l, u+d)$ se $\forall d' : 0 \leq d' \leq d \Rightarrow u + d' \in I(l)$, e
- $(l,u) \xrightarrow{a} (l', u')$ se existir $e = (l, a, g, r, l') \in E$ s.t. $u \in g$, $u' = [r \mapsto 0]u$, e $u' \in I(l')$, onde para $d \in \mathbb{R}_{\geq 0}, u + d$ mapeia cada relógio x em C para o valor $u(x) + d$, e $[r \mapsto]u$ denota a valoração do relógio que mapeia cada relógio em r para 0 e concorda com u sobre $C \backslash r$.

Autômatos temporais são frequentemente compostos em uma rede de autômatos temporais sobre um conjunto de relógios e ações comuns, consistindo de n autômatos $\mathscr{A}_i = (L_i, l_i^0, C, A, E_i, I_i)$, $1 \leq i \leq n$. Um vetor de estados é o vetor $\bar{l} = (l_1, \ldots, l_n)$. A composição de funções de invariantes em uma função comum sobre o vetor de estados $I(\bar{l}) = \wedge_i I_i(l_i)$. Onde usa-se $\bar{l}[l_i'/l_i]$ para denotar o vetor onde o i-ésimo elemento l_i de $\bar{l}$ é reposto por l_i'.

Definição 3 (Semântica de uma Rede de Autômatos Temporizados). Faça-se $\mathscr{A}_i = (L_i, l_i^0, C, A, E_i, I_i)$ uma rede de n autômatos temporizados [Behrmann et al., 2004]. Seja $\bar{l}_0 = (l_1^0, \ldots, l_n^0)$ o vetor de estados iniciais. A semântica é definida como um sistema de transição $\langle S, s_0, \rightarrow \rangle$, onde $S = (L_1 \times \cdots \times L_n) \times \mathbb{R}^C$ é o conjunto de estados, $s_0 = (\bar{l}_0, u_0)$ é o estado inicial, e $\rightarrow \subseteq S \times S$ é a relação de transição definida por:

- $(\bar{l}, u) \xrightarrow{d} (\bar{l}, u+d)$ se $\forall d' : 0 \leq d' \leq d \Rightarrow u + d' \in I(\bar{l})$.
- $(\bar{l}, u) \xrightarrow{a} (\bar{l}[l_i' \backslash l_i], u')$ se existir $l_i \xrightarrow{\tau g r} l_i'$ s.t. $u \in g$, $u' = [r \mapsto 0]u$, e $u' \in I(\bar{l}[l_i' \backslash l_i])$.
- $(\bar{l}, u) \xrightarrow{a} (\bar{l}[l_j' \backslash l_j, l_i' \backslash l_i], u')$ se existir $l_i \xrightarrow{c?g_i r_i} l_i'$ e $l_j \xrightarrow{c!g_j r_j} l_j'$ s.t. $u \in g_i \wedge g_j$, $u' = [r_i \cup r_j \mapsto 0]u$, e $u' \in I(\bar{l}[l_j' \backslash l_j, l_i' \backslash l_i], u')$.