João Batista Camargo Júnior

Metodologia de Análise de Risco em Sistemas Computacionais de Aplicação Crítica

Tese apresentada à Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a obtenção do título de Livre Docente no Departamento de Engenharia de Computação e Sistemas Digitais

Área: Sistemas Computacionais de Aplicação

São Paulo
2002

Morre lentamente quem se transforma em escravo do hábito, repetindo todos os dias os mesmos trajetos, quem não muda de marca, não arrisca vestir uma cor nova e não fala com quem não conhece.

Morre lentamente quem faz da televisão seu guru.

Morre lentamente quem evita uma paixão, quem prefere o negro ao invés do branco e os pingos nos iis a um redemoinho de emoções, exatamente a que resgata o brilho nos olhos, o sorriso nos lábios e coração aos tropeços.

Morre lentamente quem não vira a mesa quando está infeliz no trabalho, quem não arrisca o certo pelo incerto, para ir atrás de um sonho.

Morre lentamente quem não se permite, pelo menos uma vez na vida, ouvir conselhos sensatos.

Morre lentamente quem não viaja, não lê, quem não ouve música, quem não encontra graça em si mesmo.

Morre lentamente quem passa os dias queixando-se da sua má sorte, ou da chuva incessante.

Morre lentamente quem destrói seu amor próprio, quem não se deixa ajudar.

Morre lentamente quem abandona um projeto antes de iniciá-lo, nunca pergunta sobre um assunto que desconhece e nem responde quando lhe perguntam sobre algo que sabe.

Evitemos a morte em suaves porções, recordando sempre que estar vivo exige um esforço muito maior que o simples ar que respiramos.

Somente com infinita paciência conseguiremos a verdadeira felicidade.

PABLO NERUDA

Aos meus sobrinhos: Karina, Kaique, Matheus, Debora, Vitor, Flávia e Luiza

AGRADECIMENTOS

A Deus, por permitir que este trabalho se realizasse a contento.

Aos meus queridos amigos do Grupo de Análise de Segurança – GAS, em especial aos professores Jorge Rady de Almeida Júnior, Selma Shin Shimizu Melnikoff e Paulo Sérgio Cugnasca, pelo apoio e incentivo.

Ao professor Moacyr Martucci Júnior pelo constante apoio à realização deste trabalho.

Aos colegas da Escola Politécnica, que de alguma forma tenham contribuído para a realização deste trabalho.

Ao amigo Etor Cella pela constante orientação e apoio.

Aos meus amigos, que sempre me incentivaram na realização deste trabalho.

A todos os meus familiares, pela compreensão, paciência e incentivo à elaboração desse trabalho.

Metodologia de Análise de Risco em Sistemas Computacionais de Aplicação Crítica

LISTA DE FIGURAS

Figura 4.21 - Aplicação das Regras na Ferramenta MATHLAB para o Exemplo Apresentado

LISTA DE TABELAS

METODOLOGIA DE ANÁLISE DE RISCO EM SISTEMAS COMPUTACIONAIS DE APLICAÇÃO CRÍTICA

1 INTRODUÇÃO

Este trabalho de Livre Docência tem, como objetivo, apresentar uma Metodologia de Análise de Risco aplicada a sistemas computacionais críticos quanto à segurança, ou seja, sistemas que podem provocar perdas ou danos a vidas humanas, prejuízos ao meio ambiente ou grandes perdas materiais. Este tema está englobado em diversos projetos de pesquisa em andamento, sob orientação do docente, dentro do Grupo de Pesquisa que coordena, Grupo de Análise de Segurança – GAS.

Este trabalho está organizado em 6 (seis) capítulos.

O presente capítulo tem o objetivo de apresentar os aspectos motivadores, um breve histórico da área de segurança e sua complexidade envolvida.

O capítulo 2 apresenta conceitos relevantes em sistemas de segurança, destacando o papel do ser humano nesses sistemas e apresentando, também, uma sugestão de padronização de terminologia discutida e utilizada no GAS.

O capítulo 3 apresenta uma Metodologia de Análise de Risco, envolvendo as etapas de Definição e Descrição do Sistema, Análise de Perigo ("Hazard"), Qualificação do Risco Residual, Redução da Severidade dos Acidentes e Realimentação e Avaliação da Experiência Operacional. A etapa de Análise de Perigo é apresentada em mais detalhes, por constituir-se numa etapa fundamental dentro do processo de Análise de Risco. O conjunto do Gerenciamento da Segurança com a Análise de Risco constitui-se num processo mais amplo, denominado Análise de Segurança. Neste capítulo é também realizada uma breve apresentação dos aspectos gerenciais da segurança.

No capítulo 4 são apresentados dois estudos de casos. A primeira aplicação refere-se a um sistema metroviário, onde são destacados diversos resultados já obtidos ao longo da aplicação da metodologia de análise de risco. A segunda aplicação diz respeito ao sistema de controle de tráfego aéreo. Esta atividade está em processo de iniciação e de definição das pesquisas a serem realizadas pelo GAS. Neste sentido, o objetivo desse item é o de apresentar a forma com que a metodologia de análise de risco pretende ser aplicada a uma outra área de aplicação, além de serem apresentados alguns resultados preliminares das pesquisas em andamento.

No capítulo 5 são apresentadas as conclusões e considerações finais desta tese, além de serem apontadas futuras atividades de pesquisa que se mostram promissoras na área de análise de risco.

No capítulo 6 são apresentadas as referências bibliográficas.

1.1 Motivação

Ao longo dos últimos 15 anos o autor desta tese vem se dedicando a trabalhos de pesquisa que se enquadram na área de confiabilidade e segurança de sistemas computacionais.

Dentro desta linha de pesquisa, os diversos projetos de pesquisa e extensão, em andamento, estão relacionados com sistemas críticos, ou seja, sistemas em que uma falha pode provocar perdas de vidas humanas, danos ao meio ambiente ou grandes perdas materiais.

Um grande desafio em relação aos sistemas críticos, que envolvem técnicas computacionais, está na pesquisa de metodologias, métodos e ferramentas a serem utilizadas na avaliação da segurança desses sistemas, permitindo sua relação com os níveis de risco aceitáveis pela sociedade. Através das pesquisas realizadas, dentro do GAS, conclui-se que há grandes discussões com relação à padronização de metodologias a serem aplicadas aos sistemas críticos, discussões sobre avaliações quantitativas e qualitativas, além de debates sobre quais métodos de análise de perigo devem ser usados em função de uma maior criticidade dos sistemas sendo avaliados.

1.2 Histórico e Complexidade do Problema

Sistemas computacionais têm sido utilizados praticamente em diversos sistemas atualmente projetados, muitas vezes substituindo, ou então controlando intertravamentos vitais, até então projetados e implementados exclusivamente por intermédio de circuitos eletro-mecânicos, válvulas, relês mecânicos entre outros.

Um computador pode desempenhar diversos papéis dentro de um sistema crítico quanto à segurança. Ele pode, por exemplo, ser utilizado simplesmente fornecendo informações sobre o processo a um controlador humano. Em um segundo estágio, o computador pode fornecer algum tipo de interpretação sobre os dados coletados, ou finalmente comandar todo o processo, sem o auxílio de um operador. Pode-se imaginar que o primeiro caso seja o mais seguro, o que nem sempre é verdade. Se o computador apresentar alguma informação de forma incorreta, irá induzir o operador a agir incorretamente.

O motivo pelo qual a utilização de computadores é cada vez maior em sistemas dos mais diversos tipos é que eles propiciam capacidade de controle, velocidade de resposta e desempenho amplamente superiores a qualquer outro meio até então utilizado.

Outro aspecto valorizado nos computadores é a flexibilidade em se realizar alterações no software. Realmente, a alteração em si é extremamente simples de ser efetivada. No entanto, uma mudança no software pode introduzir erros não previsíveis, sendo, portanto, necessária uma avaliação completa do software modificado.

Também no que se refere à confiabilidade obtida com a utilização de computadores, pode-se pensar que seja maior do que àquela obtida com o uso de componentes convencionais. No entanto, os modos de falha do hardware que compõe um computador são extremamente complexos e difíceis de se prever.

Há ainda o componente software que, embora não apresente desgaste, pode conter erros ou alguma combinação inadequada com falhas de hardware, cujas conseqüências nem sempre são imediatamente compreendidas ou avaliadas. Não há atualmente uma metodologia amplamente aceita que avalie a segurança de um software, de modo a se poder compará-la com a de um hardware com características equivalentes. Uma possibilidade seria produzir um software completamente livre de falhas, o que até hoje tem se mostrado uma tarefa de enorme dificuldade, principalmente considerando-se o tamanho e a complexidade dos programas de controle de sistemas existentes. Há ainda a saída de se testar exaustivamente um software de maneira a se verificar seu comportamento em todos os caminhos possíveis, o que é inviável na prática, devido ao enorme número de estados que um software, mesmo não muito sofisticado, pode assumir.

Também podem ser citados os métodos formais, que exigem que as especificações sejam feitas através de linguagens formais, tais como Z e VDM. Um problema com esta abordagem é que muitas das falhas devidas ao software não têm como origem desvios em relação à sua especificação, mas esta última pode conter omissões, incompletezas ou erros.

Vale ressaltar também que a propriedade de segurança não é uma propriedade inerente ao software considerado isoladamente, mas fundamentalmente do sistema e do ambiente no qual ele é utilizado. Dessa forma, a reutilização de um software considerado seguro em um sistema não significa, necessariamente, que será segura em outro sistema e ambiente.

Muito esforço tem sido investido, envolvendo tempo e recursos na busca de um software perfeito, enquanto que pouco tem sido investido em outra direção, que seria a de se obter sistemas robustos e seguros mesmo na presença de erros no software ou qualquer outro tipo de erro. Nesse sentido, o desenvolvimento de metodologias para avaliação da segurança e da robustez de um sistema constituem-se em promissores campos de pesquisa na área de sistemas computacionais críticos.

2 SEGURANÇA

Este capítulo tem como objetivo a apresentação dos principais conceitos na área de segurança, as causas fundamentais dos acidentes, envolvendo aspectos gerenciais, técnicos e humanos, além da importância da cultura de segurança como uma atividade presente em todo ciclo de vida de um sistema crítico. É apresentada também uma sugestão de padronização de terminologia discutida e utilizada dentro do GAS.

2.1 Aspectos Conceituais

O conceito de segurança, dentro deste trabalho, pode ser definido como a probabilidade de um sistema desempenhar, num determinado período de tempo, suas funções ou descontinuá-las sem causar mortes, danos à saúde, destruição de propriedades, perda de missão ou danos ao meio ambiente. Neste sentido, o termo segurança, neste trabalho, corresponde ao termo inglês "safety".

Para que se possa garantir a segurança de um sistema, devem ser utilizadas técnicas que permitam prevenir os acidentes previsíveis, bem como minimizar as conseqüências daqueles imprevisíveis. Em um acidente, são consideradas as perdas em geral, tais como destruição de propriedade, cancelamento de missões ou danos causados ao ambiente, além de ferimentos ou morte causados em seres humanos. O desenvolvimento desses sistemas considerados críticos é, normalmente, controlado por regulamentação governamental, que estabelece critérios de certificação de sistemas em cada área de aplicação.

Quando uma perda é considerada como grave ou de vulto, geralmente justifica os esforços e os recursos a serem investidos para sua prevenção. O valor a ser investido é considerado válido ou justificável, considerando-se tanto os aspectos técnicos, quanto outros fatores, tais como sociais, psicológicos, políticos e econômicos.

As atividades relativas à segurança devem se iniciar quando o sistema começa a ser concebido e ter seqüência no projeto, produção, teste e operação do sistema. A segurança deve ser considerada como um todo, ou seja, não é suficiente que se assegure apenas a correção de partes ou de sub-sistemas de um sistema maior.

Desta forma, é de fundamental importância ter-se uma atividade de garantia da segurança atuante em todas as fases do ciclo de vida de um sistema, bem como os problemas, apontados durante a análise de risco, devem ser seriamente considerados, nunca se desprezando qualquer indício ou suspeita que possa vir a provocar um

acidente. Daí a importância em se adotar e desenvolver metodologias e métodos que cada vez mais assegurem a qualidade dos trabalhos desenvolvidos na garantia da segurança de sistemas, de forma que se obtenham sistemas de funcionamento robusto.

No contexto dos sistemas críticos quanto à segurança, os estados inseguros correspondem àqueles estados também denominados perigosos, onde o sistema está exposto à ocorrência de um acidente. Neste trabalho é adotado o termo "estado perigoso". Um projeto que tenha de contemplar aspectos referentes à segurança de um sistema deve, em primeiro lugar, buscar a eliminação de estados perigosos. Se isto não for possível, deve ser alcançado o controle desses estados perigosos preferencialmente por meio de dispositivos passivos, baseados em processos físicos, tais como a força da gravidade. Novamente, se não houver possibilidade desse controle, deve ser buscada a redução de eventuais danos causados pela ocorrência do estado perigoso. A segurança de um sistema deve ser eficaz, evitando ou minimizando a ocorrência de acidentes, além de ter um custo compatível com o sistema considerado como um todo. Sua validade só é comprovada se acidentes forem realmente prevenidos ou evitados.

Algumas vezes, a segurança de um sistema atua como uma restrição aos projetos, pois alguns de seus requisitos podem entrar em conflito com aspectos operacionais e de desempenho, bem como podem ocasionar aumentos nos custos envolvidos.

Um acidente pode acontecer se houver alguma falha ou entrada imprópria não prevista ou não coberta pelos dispositivos que deveriam garantir a segurança de um sistema. Outra causa da ocorrência de acidentes deve-se ao fator humano, ou seja, uma falha de operação por parte de um ser humano, que ocasione uma condição que possa levar a um acidente, condição esta não prevista ou não coberta pelo projeto e pela implementação do sistema.

Desta forma, as causas que justificam a ocorrência de um acidente podem ter origem em deficiências na cultura de segurança das instituições, em falhas na estrutura organizacional dessas mesmas instituições ou ainda em atividades técnicas superficiais ou ineficientes, relativas à segurança.

Cada uma dessas formas é a seguir discutida.

a) Aspectos relacionados com as deficiências na cultura de segurança das organizações.

Requisitos desejáveis em um sistema podem ser conflitantes entre si, sendo necessário, portanto, estabelecer compromissos para o desenvolvimento do projeto. Neste aspecto, pode ocorrer que a melhor ou mais avançada tecnologia seja invalidada por decisões incorretas. Pode-se citar, como exemplo, o conflito entre os fatores disponibilidade e segurança. Muitas vezes as pressões existentes podem forçar o comprometimento da segurança em função da obtenção de maiores níveis de disponibilidade. Outro campo de pesquisa e discussão que vem ganhando destaque refere-se ao confronto entre os conceitos de "Safety" e "Security", ambos denominados "segurança" na língua portuguesa. Com a tendência crescente da utilização de sistemas integrados via redes de computadores em aplicações críticas de segurança, pode-se haver conflitos entre requisitos de "security" e requisitos de "safety". Se os requisitos de "safety" e "security" forem definidos isoladamente, há o perigo que incongruências não reconhecidas e discutidas, e portanto não resolvidas, possam ocorrer, comprometendo a segurança final do sistema. Talvez a forma de integração mais apropriada seja a harmonização entre os processos de verificação de cada um desses aspectos, permitindo que os conflitos entre ambos sejam reconhecidos e resolvidos antecipadamente. [Eames 99]

Outro aspecto relacionado com a cultura de segurança refere-se à complacência e autoconfiança. As pessoas normalmente desenvolvem uma mentalidade sobre a infalibilidade do equipamento a que estão acostumadas a lidar, fruto de repetidas afirmações sobre a garantia da tecnologia utilizada para aumentar a segurança do sistema. Acredita-se que um acidente não possa ocorrer, pois não seria possível que houvesse a ocorrência de tantas condições adversas simultaneamente, o que nem sempre é verdade. Os piores acidentes ocorrem quando não se espera que possam vir a acontecer, pois geralmente se gera uma acomodação por parte das pessoas. Ao contrário, quando se acredita na possibilidade de ocorrência de um acidente, são tomadas providências para preveni-lo ou minimizar seus efeitos.

As técnicas de redundância e diversidade de projetos, utilizadas em alguns sistemas para prevenir acidentes e aumentar a segurança, também não devem ser encaradas como a solução ótima para todos os casos. Muitas vezes, há as falhas de modo comum, que podem afetar todos os canais ou parâmetros ao mesmo tempo.

Normalmente as condições perigosas mais evidentes são as que recebem maior atenção e são, por conseguinte, controladas, enquanto que aquelas condições com menor

probabilidade de ocorrência são desprezadas. É comum se verificar que, após a ocorrência de um acidente, sua causa era um evento conhecido, que foi desprezado, considerando sua ocorrência como improvável.

Outro fator importante a ser considerado no que diz respeito à complacência assumida é o caso em que um sistema opera por longos períodos de tempo sem falhas. Neste caso, acredita-se que o sistema não irá falhar nunca mais, o que não é verdade. De certo modo, o risco da ocorrência de um acidente pode até aumentar, devido a alterações no ambiente em que o sistema estiver inserido.

Também não podem ser ignorados sinais de alarme, pois os acidentes são freqüentemente precedidos por alertas, ou por uma série de ocorrências menores, geralmente ignoradas, pois não se acredita que algo mais grave ainda possa vir a acontecer.

b) Aspectos relacionados com problemas na estrutura organizacional das instituições.

A busca pela segurança deve vir desde os mais altos escalões de uma organização, difundindo-se em todos os setores da instituição. Agências governamentais e grupos de usuários podem tentar fazer com que a segurança seja mais seriamente considerada, o que só se tornará mais eficaz se houver uma conscientização, por parte da sociedade em geral, a respeito da importância fundamental das atividades que garantam a segurança dos sistemas considerados mais críticos.

De particular interesse são os sistemas compostos por diversos sub-sistemas, onde é necessário que haja um órgão centralizador de todos os grupos envolvidos na desenvolvimento do sistema, de forma que se atribua qual, ou quais grupos devem se preocupar com atributos de segurança, principalmente nas interfaces entre os sub-sistemas.

Outro ponto a ser considerado com relação ao grupo responsável pela segurança, é que deve ser independente, no que diz respeito às equipes de projeto. De preferência é recomendável que tal grupo seja vinculado a entidades completamente independentes daquelas que estiverem realizando o projeto.

Devem existir canais de comunicação adequados, entre os diversos grupos envolvidos com o sistema, de forma tal que as metas desejáveis possam ser transmitidas dos níveis hierárquicos mais altos para os mais baixos, e no sentido contrário, permitindo a avaliação sobre a evolução do projeto.

c) Problemas decorridos de atividades técnicas superficiais ou ineficientes quanto à segurança.

As condições perigosas devem ser cuidadosamente analisadas, com registros que justifiquem e suportem cada decisão de projeto, bem como os compromissos assumidos entre demais fatores e segurança.

Muitas vezes, esforços no sentido de garantir a segurança não são eficazes, pois são eliminadas as causas específicas de acidentes, mas não as causas básicas. Outro ponto que ocorre é que o projeto pode ser baseado em falsas suposições, como por exemplo, independência entre os eventos. Pode acontecer ainda de que eventuais modificações para aumentar a segurança acabem por ocasionar efeito contrário, ou seja, incrementar o número de estados perigosos devido ao aumento da complexidade do sistema. Pode ocorrer que a colocação de um dispositivo de segurança instalado para corrigir determinada condição, seja utilizado para justificar redução em margens de segurança em outros aspectos.

Outro fator muito importante é a realimentação com as informações sobre os acidentes ocorridos em outros sistemas, similares ou não, ao que estiver sendo desenvolvido, pois essas informações podem e devem ser efetivamente utilizadas na prevenção de novos acidentes.

2.2 Terminologia

Neste item é apresentada uma padronização dos termos mais importantes na área de segurança e que foram resultados de diversas discussões dentro do GAS.

Na língua inglesa existem três conceitos básicos plenamente aceitos: "Fault", "Error" e "Failure". [Johnson 89]

O termo "Fault" corresponde a um problema interno de um componente, seja de hardware ou software. No que diz respeito ao hardware, o termo pode corresponder a um problema na especificação, na implementação, desgaste do componente ou a distúrbios externos. Já com relação ao software são considerados apenas os dois primeiros aspectos, especificação e implementação. O termo "Fault" corresponde à causa hipotética do "Error".

O termo "Error" corresponde à contaminação da informação interna do sistema, podendo levar o sistema a ocorrência de "Failure", ou seja mau-funcionamento.

O termo "Failure" corresponde, desta forma, à não conformidade do serviço prestado em relação à sua especificação. Eqüivale, desta forma, ao término da capacidade do

sistema em executar sua função requerida ou à sua incapacidade de executá-la dentro limite especificado.

Em Portugal os termos "Fault", "Error" e "Failure" são traduzidos por "Falha", "Erro" e "Avaria". Este último termo não é usual no Brasil. Dentro da Sociedade Brasileira de Computação – SBC existe uma forte tendência de se adotar a tradução "Falha", "Erro" e "Defeito". De acordo com a norma ABNT NBR9126 são utilizados os termos "Defeito" para "Fault" e "Falha" para "Failure". [NBR 9126]

Pode-se concluir que, na língua portuguesa, ainda existe muita discussão e pouca padronização no que se refere a estes termos. Neste sentido, pretende-se adotar, neste trabalho, uma terminologia em função de alguns aspectos a seguir apresentados. Esta terminologia tem como objetivo estabelecer uma maior clareza na apresentação das idéias.

Com relação ao termo "Fault" será adotada a tradução "Falha". Este termo tem sido consagrado em diversas expressões em português como, por exemplo, Árvore de Falhas ("Fault Tree") e Detecção de Falhas ("Fault Detection"). No contexto do software, o termo é adotado no sentido de *um passo, processo ou definições de dados incorretos em um programa de computador*. Outras alternativas de tradução podem ser "defeito", "desvio", "incorreções" ou ainda "bug" no caso de módulos de software.

O termo "Error" será traduzido como "Erro".

O termo "Failure" será traduzido como "Disfunção". A justificativa para tal padronização está relacionada com a precisão do seu significado e também pela sua consistência de uso na área médica. Alternativas para este termo seriam "Avaria" e "Mau Funcionamento". Não é aconselhável o uso do termo "Defeito" neste caso por considerá-lo como sendo intrínseco ao sistema.

2.3 Erro Humano

Muitos fatores podem levar o ser humano a agir de forma incorreta nos sistemas críticos quanto à segurança. Neste aspecto está sendo feita referência especial aos Operadores. Podem ser citados, como exemplo, o fornecimento ao operador de dados incompletos, incorretos, complexos ou excessivos. Outra crença negativa é que os operadores podem superar qualquer emergência, forçando-os a intervir em situações limites.

Além destes aspectos, muitas vezes o operador é responsabilizado por acidentes como forma de negligenciar ou até mesmo esconder erros cometidos por projetistas e

gerentes. Na grande maioria das vezes, as ações positivas dos operadores raramente são destacadas, sendo registradas apenas as ações negativas.

Tendo em mente todas essas dificuldades, é apresentado a seguir uma breve discussão sobre a necessidade dos operadores nos sistemas automáticos, os modelos cognitivos da tarefa humana e os possíveis papéis dos operadores nestes sistemas.[Leveson 95]

a) A Necessidade do Operador em Sistemas Críticos quanto à Segurança

Uma das grandes questões que surge se refere à eliminação ou não do operador dos sistemas críticos, substituindo-os por computadores. Diversas razões corroboram para que o operador não seja eliminado desses sistemas. São discutidos a seguir alguns destes aspectos.

Pode-se afirmar que é extremamente difícil antecipar todas as condições e todas as interações não desejadas entre os componentes, que podem ocorrer no ambiente de um sistema. A presença do operador nestes casos pode diminuir o risco de um acidente.

Outro aspecto importante refere-se aos eventuais erros existentes num determinado sistema provocados por erros de seus projetistas. Embora os projetistas trabalhem sob condições de menor pressão que os operadores, eles também cometem erros. Os erros relacionados com os projetistas estão focados em aspectos como, dificuldade em alocar probabilidades em eventos raros, não consideração de efeitos colaterais, não atenção para medidas contingênciais, controle da complexidade do sistema, concentrando-se apenas em alguns aspectos do problema, capacidade limitada de compreender relações complexas, além de dificuldades em se ter uma visão ampla do sistema, especialmente no que diz respeito aos sistemas críticos computacionais.

Desta forma, é importante a participação de operadores nestes sistemas. A questão que se deve discutir é qual deve ser o seu papel nestes sistemas

b) O papel do ser humano nos Sistemas Críticos quanto à Segurança

A automação de sistemas provoca o reposicionamento do ser humano em novos níveis de complexidade, com um maior nível de controle e supervisão, refletindo, desta forma, maiores níveis de tomada de decisões. Neste sentido, aumenta-se o nível de centralização, tornando a base de decisão extremamente mais complexa. Desta forma, o estudo da confiabilidade humana e da ergonomia passam a ter um papel fundamental dentro do estudo da segurança dos sistemas computacionais. Nessa relação do operador com os sistemas computacionais o ser-humano pode assumir basicamente três papéis: Monitor, Back-Up ou Parceiro.

A experiência demonstra que o ser humano apresenta um desempenho fraco como monitor em sistemas automatizados. Nesta situação o operador está extremamente dependente de informações fornecidas pelo sistema, que podem ser disponibilizadas em grande quantidade e de forma não adequada. Vale ressaltar, nessa situação, que as tarefas, que requerem baixa atividade do operador, podem implicar numa baixa vigilância de sua parte, podendo conduzi-lo a atitudes de superconfiaça ou complacência em relação ao sistema automatizado.

Quando o operador desempenha o papel de Back-up, o projeto do sistema de automação pode tornar-se de difícil gerenciamento durante uma situação emergência ou em sistemas complexos. Nestas situações o operador pode ter muito pouco tempo para decidir, além do sistema oferecer diversas opções de escolha. Adiciona-se a isto, o fato do operador estar inativo por longos períodos, dificultando ainda mais sua tomada de decisão.

Já quando o operador tem o papel de Parceiro dentro do sistema crítico, ele irá realizar tarefas que não puderam ser automatizadas por algum motivo ou intencionalmente alocadas a ele. Neste modo de trabalho, podem surgir algumas vantagens e desvantagens que devem ser avaliadas em cada caso. Podem surgir tarefas extremamente complexas e com muita arbitrariedade na tomada de decisão. Por outro lado, o operador pode agir com maior criatividade e utilizando seus conhecimentos a respeito do sistema, se sentindo, desta forma, realmente integrado no processo de automação.

Nesse sentido, o foco, com relação ao Operador, dentro um projeto de um sistema crítico quanto à segurança, deve ser sua Operação e não sua Função. O Operador deve ser envolvido num processo de tomada de decisões do projeto e da sua análise de risco. Quando o operador atua como parceiro, ele se sente realmente integrado e motivado para atuar no âmbito da responsabilidade deste tipo de sistema.

3 METODOLOGIA DE ANÁLISE DE RISCO PROPOSTA

Neste capítulo é apresentada uma Metodologia de Análise de Risco para sistemas críticos quanto à segurança. Esta metodologia foi desenvolvida através da sinergia de dois aspectos fundamentais: a pesquisa acadêmica e a identificação de necessidades através de experiências práticas em análise de risco de sistemas críticos na área de transporte metroviário e aeroviário. No entanto, esta metodologia pode ser aplicada a outros sistemas críticos quanto à segurança através da avaliação e a adequação às características das outras áreas de aplicação. Trata-se de uma via de duas mãos. Tanto a pesquisa acadêmica abre novas possibilidades para a resolução de diversos problemas práticos, como a experiência auxilia em prover um maior direcionamento nas pesquisas acadêmicas sendo realizadas.

A metodologia, aqui apresentada, é parte integrante de um processo mais amplo denominado Análise de Segurança, constituído pelo Gerenciamento da Segurança e pela Análise de Risco, propriamente dita. O aspecto de gerenciamento da segurança não é o foco principal desta pesquisa, sendo apenas apresentada uma breve descrição no item 3.1.

Dentro da metodologia de Análise de Risco, apresentada no item 3.2., são destacadas as seguintes etapas:

- Definição e descrição do sistema, suas interfaces e demais informações necessárias para a Análise de Risco;
- Realização do processo de Análise de Perigo;
- Qualificação do Risco Residual;
- Avaliação da severidade dos acidentes relacionados com o estado perigoso; e
- Realimentação e Avaliação da experiência operacional.

No que se refere ao processo de Análise de Perigo são apresentadas, no item 3.3., as seguintes etapas:

- Análise Preliminar de Perigo;
- Análise de Perigo do Sistema;
- Análise de Perigo do Subsistema;
- Análise de Perigo da Operação e Suporte; e
- Análise Final de Perigo

Finalizando são apresentados alguns métodos que podem ser utilizados durante o processo de Análise de Perigo.

3.1 Gerenciamento da Segurança

O primeiro aspecto em um sistema de segurança corresponde ao gerenciamento da segurança. Nesse sentido, a discussão sobre as questões de segurança deve criar uma atmosfera de cooperação e não acusação entre os diversos grupos técnicos. Este deve ser um cuidado fundamental para o bom andamento dos trabalhos.

Outro ponto importante refere-se à definição de políticas de segurança que definam claramente a relação entre a Segurança ("Safety") e outros objetivos do sistema sendo avaliado, tais como, Confiabilidade, Disponibilidade, Segurança ("Security") e Custos, entre outros.

Para que estes aspectos sejam efetivamente implementados torna-se necessária a existência de um Grupo de Segurança responsável por formular, difundir e implantar a política de segurança no âmbito do projeto do sistema crítico, documentar o rastreamento dos perigos e suas relações, adaptar e desenvolver normas específicas, realizar a análise de perigo, planejar e monitorar as tarefas dos testes de segurança, participar de revisões do programa, realizar intercâmbios com outros grupos de segurança e investigar e analisar acidentes. Evidentemente este grupo deve ser tão mais independente quanto maior for o nível de risco envolvido no sistema computacional sendo implantado ou avaliado. A política de segurança estará refletida através de um Plano Global de Segurança, destacando, dentro deste, um Plano Específico para a Segurança do Software, em função de sua grande complexidade. Este Plano Global de Segurança deve destacar a organização da segurança do sistema, o cronograma do programa de segurança, os critérios de segurança além das atividades que devem ser realizadas ao longo do desenvolvimento do sistema visando a comprovação da segurança. [Hall 97]

Em função das grandes responsabilidades envolvidas nas atividades deste Grupo de Segurança, existe uma tendência mundial em se exigir uma maior qualificação dos profissionais participantes dessas atividades. Este ponto não elimina de maneira nenhuma a responsabilidade de todas as demais equipes envolvidas no projeto, implantação, operação e manutenção do sistema de segurança. [Renn 98] Vale ressaltar que o atributo segurança é conseqüência de uma cooperação decorrente de um programa global de segurança. Como este Grupo de Segurança deverá manter intercâmbios com

as demais equipes envolvidas no sistema de segurança, canais de comunicação devem ser estabelecidos permitindo, desta forma, uma maior rapidez e eficácia.

3.2 Análise de Risco

A Análise de Risco avalia a importância relativa do perigo e permite avaliar sua aceitabilidade ou não. Na realidade, em diversos momentos de nossas vidas, estão sendo avaliados subjetivamente os riscos que podemos estar correndo.[Bohnenblust 98] No sentido de uma melhor compreensão da natureza do risco é extremamente útil considerar a relação entre Perigos e Acidentes, que resultam em um dano à pessoa humana ou ao meio ambiente. Considera-se que o perigo representa uma situação de potencial acidente.

O Risco associado a um perigo é determinado pela combinação de dois fatores: a freqüência ou probabilidade da ocorrência de um acidente e a severidade da conseqüência envolvida no acidente. O Risco pode ser avaliado qualitativamente e quantitativamente. As escalas quantitativas podem variar de aplicação para aplicação, apesar do esforço de algumas normas internacionais em padronizar essa escala. [Storey 96] [Ladkin 01]

Um determinado Risco não é aceitável, se existir um determinado perigo com alta probabilidade de ocorrência e conseqüências desastrosas. No entanto, poder-se-ia aceitar um risco, em relação a um determinado perigo, com baixíssima probabilidade, apesar de apresentar conseqüências altamente danosas. O nível de aceitabilidade do risco é determinado pelo benefício associado ao Risco, e pelo esforço requerido em diminui-lo.

A redução necessária de risco deve ser alcançada para se manter o risco tolerável em uma determinada situação. O conceito de redução necessária de risco é de fundamental importância na avaliação de sua aceitabilidade. [DD ENV 50129:1999]

A norma IEC 61508-1 define níveis de integridade de segurança SIL – "Safety Integrity Level" para sistemas que operam em regime de baixa demanda e sistemas que operam em regime de alta demanda ou de modo contínuo. [IEC 61508]

Um sistema opera em regime de baixa demanda, se a freqüência com a qual ele for solicitado não for maior que uma vez por ano e não for maior que duas vezes a freqüência com a qual ele é verificado, ou seja, sofre um processo de manutenção preventiva. Caso contrário, considera-se que o sistema tem um regime de operação de alta demanda ou de modo contínuo.

Para sistemas que operam em baixa demanda, são definidos os níveis de integridade de segurança em termos de valores limites da probabilidade média de falha ao executar a função para a qual foi projetado na demanda. Para sistemas que operam em regime de alta demanda os níveis estão em termos de valores limites para a probabilidade de ocorrência de falhas inseguras por hora.

A tabela 3.1 apresenta os 4 (quatro) níveis SIL existentes para os dois tipos de modo de operação citados.

SIL	Baixa Demanda (falha insegura por demanda)	Alta Demanda (falha insegura por hora)
4	$10^{-5} \leq \lambda \leq 10^{-4}$	$10^{-9} \leq \lambda \leq 10^{-8}$
3	$10^{-4} \leq \lambda \leq 10^{-3}$	$10^{-8} \leq \lambda \leq 10^{-7}$
2	$10^{-3} \leq \lambda \leq 10^{-2}$	$10^{-7} \leq \lambda \leq 10^{-6}$
1	$10^{-2} \leq \lambda \leq 10^{-1}$	$10^{-6} \leq \lambda \leq 10^{-5}$

Tabela 3.1 – Níveis de Integridade de Segurança

Um dos métodos de determinação do risco aceitável é denominado ALARP - "As Low As Reasonable Practible". [Melchers 01] A aplicação do princípio ALARP significa tentar reduzir o nível de risco de um sistema a valores tão baixos quanto a relação entre o ganho e o investimento for aceitável. Esta técnica é usada principalmente no Reino Unido.

Em uma aplicação, os níveis de risco são classificados da seguinte forma:

a) O Risco é tão grande que não deve ser tolerado; ou
b) O Risco é ou tornou-se tão pequeno, tornando-se insignificante; ou
c) O Risco está entre os dois estados especificados nos itens **a** e **b**, tendo sido reduzido ao mais baixo nível praticável, tendo em vista os benefícios resultantes de sua aceitação e os custos de qualquer redução adicional.

Com respeito ao item **c**, o princípio ALARP requer que qualquer risco deva ser reduzido, o quanto for razoavelmente praticável, para um nível tão baixo quanto razoavelmente aceitável.

As três regiões são mostradas na figura 3.1.

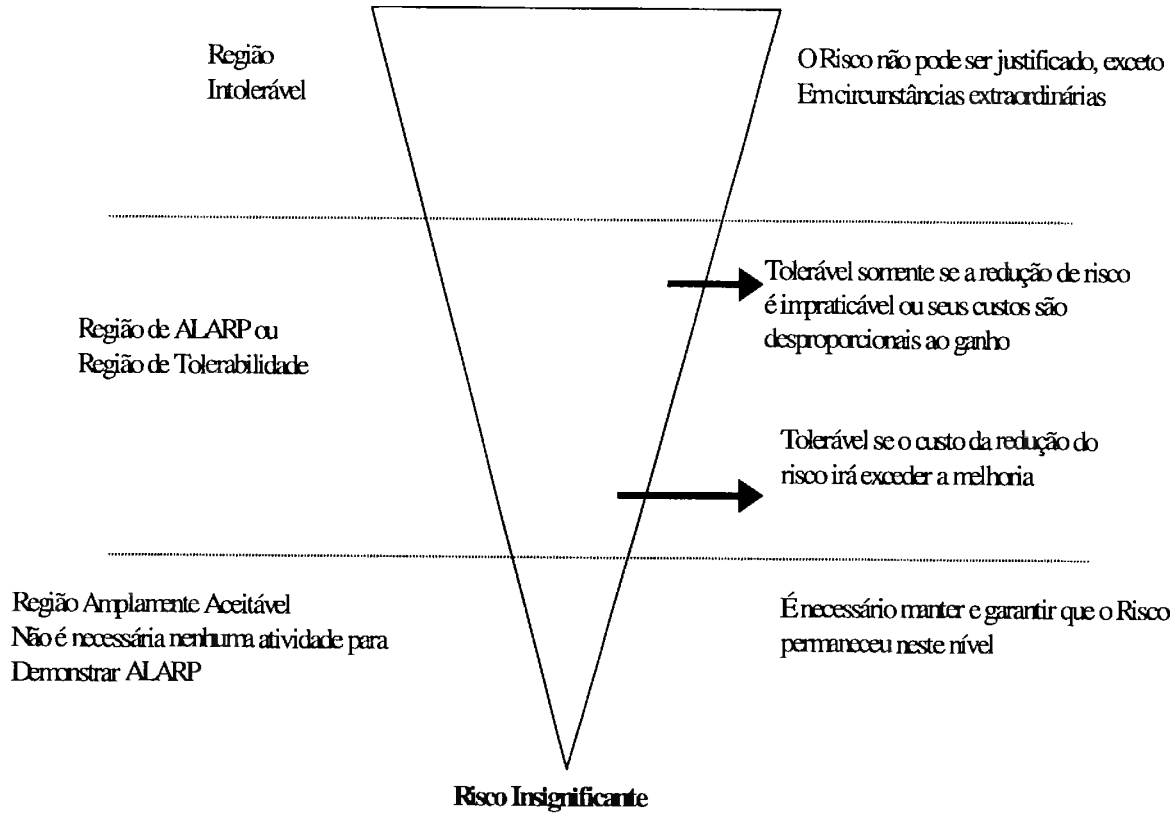


Figura 3.1 – Regiões de Risco

Acima de um certo nível, um Risco é considerado intolerável e não pode ser justificado em

qualquer circunstância ordinária. Neste caso, ele está situado na região intolerável.

Abaixo desse nível, há a região de tolerabilidade, onde se permite que uma atividade ocorra com seus Riscos associados, que são tão baixos quanto o razoavelmente praticável. Ser tolerável significa conviver com Riscos, obtendo os benefícios do funcionamento do sistema. Ao mesmo tempo, espera-se que este nível de Risco seja mantido sob constante acompanhamento, sendo reduzido como e quando isto possa ser feito.

Uma avaliação de custo e benefício é exigida explícita ou implicitamente de forma a se avaliar o custo e a necessidade de melhorias, ou ainda para se avaliar a necessidade de medidas de segurança adicionais.

Um trabalho fundamental refere-se à determinação dos níveis SIL. De acordo com a norma IEC 61508-5, os níveis SIL dependem dos riscos aos quais a aplicação está sujeita. Estes riscos devem levar em conta as seguintes conseqüências:

- Perdas de Vidas Humanas ou de outras vidas (animais);
- Ferimentos ou doenças em pessoas;

- Poluição Ambiental; e
- Perdas ou danos à propriedade.

A norma apresenta dois métodos para a determinação do nível SIL, um quantitativo, ainda não desenvolvido completamente e outro qualitativo. No item 3.3.3.9 deste capítulo é apresentado um método de avaliação quantitativa da segurança de aplicações microprocessadas, que pode fornecer subsídios para a verificação do nível SIL exigido.

A avaliação qualitativa é realizada através de um gráfico de risco, apresentado na figura 3.2, que leva em consideração os seguintes parâmetros:

- Conseqüência do evento perigoso (**Ci**)
- Freqüência e tempo de exposição ao perigo (**Fi**)
- A possibilidade de evitar o evento perigoso (**Pi**)
- A probabilidade de ocorrência indesejável, como um acidente (**Wi**)

Nessa figura os parâmetros ***a, b, c, d, e, f, g, h*** estão relacionados com o Nível de Integridade de Segurança requerido.

			W3	W2	W1
C1			**a**	**-**	**-**
C2	**F1**	**P1**	**b**	**a**	**-**
		P2	**c**	**b**	**a**
	F2	**P1**	**d**	**c**	**b**
		P2	**e**	**d**	**c**
C3	**F1**		**f**	**e**	**d**
	F2		**g**	**f**	**e**
C4			**h**	**g**	**f**

Figura 3.2 – Gráfico de Risco

A tabela 3.2. apresenta a classificação dos níveis desses parâmetros:

Parâmetro de Risco	Classificação
Conseqüência (C)	C1 – Perdas Pequenas C2 – Perdas sérias e permanentes para uma ou mais pessoas, ou morte de uma pessoa C3 – Morte de várias pessoas C4 – Muitas pessoas assassinadas
Freqüência e tempo de exposição ao perigo (F)	F1 – Exposição de rara a freqüente F2 – Exposição de freqüente a permanente
Possibilidade de evitar o evento perigoso (P)	P1 – Possível sob certas condições P2 – Quase possível
Probabilidade de ocorrência de eventos indesejáveis (W)	W1 – Probabilidade muito pequena de que as ocorrências indesejáveis acontecerão W2 – Probabilidade pequena de que as ocorrências indesejáveis acontecerão W3 – Probabilidade relativamente alta de que as ocorrências indesejáveis acontecerão

Tabela 3.2 – Classificação dos Níveis do Gráfico de Risco

Uma vez classificados, através da tabela anterior, os níveis dos parâmetros ***C, P, F e W***, é possível determinar, a partir do Gráfico de Risco, um parâmetro que define a redução mínima do risco necessário para a aplicação.

A tabela 3.3. associa o Gráfico de Risco com o Nível de Integridade de Segurança – SIL requerido para a aplicação.

Redução para um Risco Mínimo	Nível de Integridade de Segurança
-	Sem requisito de segurança
a	Sem requisitos especiais de segurança
b,c	1
d	2
e,f	3
g	4
h	Sistemas de proteção não são suficientes

Tabela 3.3. – Estabelecimento do Nível de Integridade de Segurança

Para um melhor entendimento do método de determinação do nível SIL, é apresentado uma aplicação a um sistema metroviário. Num sistema metroviário, a conseqüência de um evento perigoso pode apresentar características catastróficas (nível **C4**), devendo apresentar uma probabilidade pequena de ocorrência (nível **W2**). Desta forma o parâmetro selecionado através da figura 3.2 corresponde à letra **g**. Através da tabela 3.3 pode, portanto, estabelecer o SIL nível 4 para um sistema metroviário.

Após a determinação dos Níveis de Integridade de Segurança, pode-se iniciar o processo de Análise de Risco. [Leveson 95] [Storey 96] [NASA 96],[BS EN 50126:1999] Para tal são necessárias as seguintes etapas:

- Definição e Descrição do Sistema, Interfaces e demais informações necessárias para a Análise de Risco
- Realização do processo de **Análise de Perigo**
- Qualificação do Risco Residual
- Caso o Nível de Risco residual não seja aceitável, redução da severidade dos acidentes relacionados com o estado perigoso, ou da probabilidade de sua ocorrência
- Realimentação e Avaliação da experiência operacional

De forma a exemplificar os conceitos apresentados, considere-se um sistema de controle de uma cancela numa passagem de nível. Dentro do processo de Análise de Risco padroniza-se como Análise de Perigo o processo de determinação da probabilidade de se atingir um estado perigoso. Neste sistema o estado perigoso corresponde ao sistema falhar de forma a não abaixar a cancela quando da aproximação de um trem. Na realidade, se este estado perigoso for alcançado, não obrigatoriamente ocorrerá um acidente. Outros fatores deverão ser considerados. A partir deste estado perigoso, pode-

se determinar a probabilidade da ocorrência de um acidente e sua severidade. Se o risco residual não for aceitável deve-se então trabalhar no sentido de diminuí-lo, seja através da diminuição de sua probabilidade ou diminuição da severidade envolvida.

A Análise de Risco pode ser exigida através de um processo denominado de Certificação. [Sotrey 96]

Trata-se do processo de emitir um Certificado indicando conformidade com uma norma, um conjunto de recomendações ou algum documento similar. Qualquer Organização ou Indivíduo pode emitir um Certificado, e sua importância irá variar muito com a natureza do objeto certificado. Em alguns casos, pode-se exigir um Certificado devido às exigências legais. Nestes casos o Certificado tem o papel de uma Licença de uma Autoridade Regulamentada. Esta necessidade para sistemas críticos quanto à segurança varia muito conforme o país.

Em áreas não cobertas por exigências legais, o Certificado pode ter, por exemplo, uma importância comercial. Muitas indústrias possuem uma Autoridade Reguladora que governa todos os projetos dentro de um determinado setor.

Por exemplo, no setor de aviação civil do Reino Unido, a entidade certificadora é a "Civil Aviation Authority - CAA" e dos Estados Unidos, a "Federal Aviation Authority – FAA".

A Certificação pode ser aplicada a Organizações e Indivíduos, a Ferramentas e Métodos e a Sistemas e Produtos.

Com o objetivo de obter certificação, o projetista de um produto crítico deve provar, ao Órgão Certificador, a segurança de seu produto. O projetista deve ser capaz de mostrar que todos os perigos foram identificados e tratados adequadamente e que a integridade do sistema é apropriada para aquela aplicação, ou seja, atende aos níveis de integridade de segurança exigidos. O trabalho envolvido num processo de Certificação é bastante grande e requer um planejamento cuidadoso.

A Certificação pode ser realizada sobre Organizações e Indivíduos, sobre Ferramentas e Métodos e sobre Sistemas e Produtos, esta última forma a mais conhecida. É apresentada, a seguir, uma breve descrição destes tipos de Certificação.

a) Certificação de Organizações e Indivíduos

Esta Certificação tem como objetivo estabelecer a competência de uma organização em uma área específica de atividade. Existe um paralelismo com a garantia de qualidade através da ISO9000.

A certificação pode ser aplicada também a indivíduos. Estes devem ser certificados com o objetivo de serem autorizados a desempenhar uma determinada profissão.

Poucas indústrias e organizações, possuem alguma forma de certificação de profissionais para trabalhar com o desenvolvimento, teste e avaliação de sistemas críticos quanto à segurança. Como exemplo, pode-se citar que a FAA delega a Certificação de Profissionais para as DER's - "Designated Engineering Representatives". A Boeing e a Mc Donnell Douglas possuem 90 a 95% de suas atividades certificadas pelas DER's. Desta forma, a Certificação pode também ser aplicada a operadores de sistemas críticos quanto à segurança.

b) Certificação de Ferramentas e Métodos

As ferramentas e os métodos de desenvolvimento utilizados na produção de sistemas críticos de segurança desempenham um papel fundamental na obtenção destes sistemas.

No caso do software, um maior nível de qualidade do processo é exigido. Pode-se citar, como exemplo, a exigência de um maior nível CMM, correspondente a um maior de segurança SIL. [Myerson 96]

c) Certificação de Sistemas ou Produtos

Este tipo de certificação é realizado devido a exigências legais ou por motivos de mercado.

Na área médica, os sistemas eletrônicos possuem certificação voluntária no Reino Unido e certificação obrigatória nos EUA e na Alemanha. Na área da aviação civil e na área nuclear a certificação é sempre obrigatória em todos os países.

A certificação pode ser aplicada ao Sistema Completo ou a Componentes Individuais.

Embora a fase de certificação de um sistema se realize no final de seu desenvolvimento, o planejamento deste trabalho deve ser realizado no início do ciclo de vida, da mesma forma que as atividades de verificação e validação. Nestes casos, o projetista precisa preparar um Plano de Certificação para ser aprovado pela Autoridade Reguladora.

Este Plano de Certificação deve conter detalhes do sistema proposto, os métodos de desenvolvimento a serem utilizados e a documentação a ser fornecida. Quando uma norma em especial é adotada, o plano deve indicar os métodos/técnicas adotados para

atender à norma. Aqueles aspectos em que a norma não é atendida devem ser amplamente justificados.

A submissão do Plano de Certificação será continuada por um debate entre o Projetista e o Órgão Regulador, com o objetivo de resolver quaisquer desacordos ou mal entendimentos.

Se tudo ocorrer bem, no final deste processo o Projetista recebe um "De Acordo" do Órgão Regulador para o desenvolvimento proposto. Caso contrário, mudanças devem ser realizadas no projeto proposto e gerado um novo Plano de Certificação.

Se, durante o projeto, ocorrerem mudanças que podem influenciar o Plano de Certificação, deve haver uma nova avaliação deste plano. Este processo é válido durante todo o Ciclo de Vida do Sistema.

À medida que o projeto se desenvolve, o Projetista deve fornecer, ao Órgão Regulador, toda a documentação adequada, conforme o Plano de Certificação. Em grandes projetos, a documentação é muito vasta, representando um grande investimento em tempo e esforço.

Uma grande porcentagem desta documentação refere-se ao Plano de Segurança que detalha o tratamento das tarefas de segurança através do processo de desenvolvimento. Através de todo o material fornecido, o Órgão Regulador irá emitir uma série de revisões sobre o mesmo. Se o Órgão Regulador aceitar toda essa documentação, é emitido um Certificado ou uma Licença. Em alguns casos, esta Certificação pode ser condicional, através de certas restrições operacionais.

3.3 Análise de Perigo

Neste item são apresentados os detalhes que devem nortear um trabalho de Análise de Perigo. O escopo da aplicação de um processo de Análise de Perigo é basicamente constituído por dois objetivos.

O primeiro refere-se ao desenvolvimento de novos sistemas. Neste aspecto a Análise de Perigo procura identificar e avaliar potenciais perigos, além de eliminá-los ou controlá-los.

O segundo escopo refere-se à Análise de Perigo de sistemas existentes. Neste caso o trabalho tem como meta identificar e avaliar perigos, visando quantificar os seus níveis de segurança, formular políticas de segurança, treinar profissionais e aumentar a motivação em se atingir uma operação segura e eficiente.

A Análise de Perigo deve ser um processo contínuo e interativo, se estendendo por todo o ciclo de vida de um sistema.

Ao longo de cada uma das etapas de Análise de Perigo, podem ser utilizados diversos métodos, sendo apresentados, no item 3.4., parte significativa e representativa deles. Alguns métodos são aplicáveis ao sistema, outros aplicáveis aos módulos de hardware, enquanto outros se adaptam mais aos módulos de software. A melhor aplicação é discutida na apresentação específica de cada método.

O processo de Análise de Perigo pode ser dividido em cinco etapas:

- Análise Preliminar de Perigo
- Análise de Perigo do Sistema
- Análise de Perigo do Subsistema
- Análise de Perigo da Operação e Suporte
- Análise Final de Perigo

A seguir são descritas cada uma destas etapas.

3.3.1 Análise Preliminar de Perigo - Preliminary Hazard Analysis (PHA)

Os objetivos desta etapa são:

- Determinar quais perigos podem existir durante a operação do sistema e sua magnitude relativa;
- Desenvolver recomendações, especificações e critérios para serem seguidos no projeto do sistema. Neste sentido pode ser estabelecido um conjunto de Requisitos Gerais de Segurança – RGS do Sistema;
- Ações iniciais para o controle de um perigo em particular;
- Identificação de Responsabilidades Técnicas e Gerenciais para a Ação e Aceitação de Riscos, como também para Avaliação do Controle sobre os Perigos Indentificados;
- Determinação da magnitude e complexidade dos problemas de segurança.

3.3.2 Análise de Perigo do Sistema - System Hazard Analysis (SHA)

A Análise de Perigo do Sistema pode ter seu início na Revisão Preliminar de Projeto, devendo se estender ao longo de todo o ciclo de vida de um projeto. O principal objetivo desta atividade é recomendar mudanças além de controlar e avaliar o atendimento aos Requisitos Gerais de Segurança – RGS, tanto em operação normal como em operação degradada do sistema, levando evidentemente em consideração a presença de falhas. Nesta etapa, os componentes envolvidos são os diversos subsistemas, incluindo a interface homem-máquina. Como no início deste tipo de análise já se tem uma visão preliminar da arquitetura do sistema, deve-se iniciar o estudo da interação entre os diversos subsistemas constituintes e como suas interações podem afetar a segurança do sistema. Em função desse trabalho podem ser determinados os Requisitos Gerais de Segurança relativos aos subsistemas envolvidos.

3.3.3 Análise de Perigo dos Subsistemas – Subsystem Hazard Analysis (SSHA)

Esta etapa de análise deve ter início a partir da existência de um projeto mais detalhado relativo aos subsistemas. Ela apresenta os mesmos objetivos da análise anterior, só que focada em cada subsistema, de forma mais detalhada. Evidentemente que os outros subsistemas envolvidos, implementados através de outras tecnologias, deverão sofrer um processo de análise com técnicas específicas e desenvolvidas para este fim.

Tendo em mente os sistemas computacionais, objeto deste trabalho de pesquisa, a etapa de análise de perigo dos subsistemas pode ser dividida em: Análise de Perigo do Hardware e Análise de Perigo do Software.

3.3.3.1 Análise de Perigo do Hardware

A Análise de Perigo do Hardware pode ser subdividida em Análise de Perigo do Hardware Fail-Safe, Análise de Perigo do Hardware Redundante, Determinação dos Meios de Detecção e Recuperação de Falhas Implementados por Hardware e Análise dos Aspectos Construtivos do Hardware.

Um hardware é considerado redundante quando diversas réplicas deste módulo de hardware são utilizadas no sistema visando o atendimento a requisitos não funcionais como, por exemplo, segurança e confiabilidade. Já um hardware é considerado "fail safe" quando, na presença de qualquer falha, simples ou múltipla, sempre é atingido um estado seguro. A exigência de falhas simples ou múltiplas está intimamente ligado ao grau de tolerância de falhas considerado no conceito "fail safe" de um determinado

projeto. Pode-se dizer que em sistemas metroviários, o conceito "fail safe" geralmente trabalha com falhas simples. Já na aplicação aeroviária, o conceito "fail safe" lida normalmente com falhas duplas. A diferença básica entre a Análise de Perigo de um Hardware Redundante e de um Hardware Fail-Safe é que, no primeiro, deve haver também uma análise de independência entre os canais redundantes, procurando identificar as falhas de causa comum.

De acordo com a norma [IEC 61508], algumas considerações são feitas com relação ao aspecto de independência dos canais redundantes:

"*Os sistemas redundantes podem ser tratados como independentes, ou seja, não apresentam falhas de modo comum, desde que:*

- *Sejam funcionalmente diversos para atingir o mesmo resultado;*
- *Sejam baseados em tecnologias diversas;*
- *Não compartilhem partes ou serviços cuja falha possa resultar numa situação perigosa;*
- *Sejam projetados de forma que o modo de falha predominante da parte comum do sistema de suporte (energia) seja na direção segura;*
- *Não devam compartilhar procedimentos operacionais ou de manutenção e testes comuns; e*
- *Estejam fisicamente separados de maneira que falhas externas não afetem os sistemas redundantes e as facilidades externas de diminuição de risco.*

Se todas as exigências anteriores não puderem ser atendidas, então os sistemas relacionados com a segurança não devem ser considerados como independentes do ponto de vista da Alocação da Integridade de Segurança, a menos que uma análise tenha sido realizada, mostrando que a probabilidade da falha dependente (de modo comum) seja suficientemente baixa em comparação com o Requisito de Integridade de Segurança desejado."

De acordo com a norma [EN 50128] são feitas algumas considerações com relação às Falhas de Causa Comum:

"*Algumas falhas podem ser comuns em mais de um componente redundante. Por exemplo, se um sistema computacional é instalado em uma única sala, problemas no ar condicionado podem reduzir os benefícios da redundância. O mesmo pode-se dizer para outros eventos externos como: fogo, inundação, interferência eletromagnética, acidentes aéreos e terremotos. O sistema computacional pode também ser afetado por incidentes relacionados com sua operação e manutenção. É essencial, portanto, que sejam estabelecidos procedimentos adequados de operação e manutenção, além de serem bem documentados. O treinamento da equipe de operação e manutenção é também essencial*"

a) Análise de Perigo do Hardware "Fail-Safe"

A Análise de Perigo do Hardware "Fail-Safe" é constituída pelas seguintes atividades:

- **Descrição Funcional e Análise do Módulo em Operação Normal**: esta atividade tem como função primordial descrever e entender todos os aspectos funcionais envolvidos na implementação do módulo em questão. No processo de análise são utilizadas técnicas de simulação com o apoio de ferramentas apropriadas, além de discussões sobre a funcionalidade do módulo entre os profissionais da equipe.
- **Detalhamento dos Requisitos Gerais de Segurança:** esta atividade tem a meta de determinar os requisitos de segurança específicos para determinados blocos de hardware. O objetivo nesta atividade é fornecer subsídios para as próximas atividades, visando a identificação da presença de estados perigosos.
- **Análise Crítica dos Efeitos dos Modos de Falhas – FMECA**: esta atividade tem como finalidade analisar todos os efeitos locais e no sistema, de cada um dos modos de falhas dos diversos componentes existentes neste módulo. No caso de falhas não detectáveis, devem ser avaliadas as combinações com outras falhas possíveis ou entradas impróprias, no sentido de se avaliar qualquer possibilidade de se atingir um estado perigoso. [Beerthuizen 01] São também realizadas simulações na presença de falhas, tanto operacionais como do próprio hardware.
- **Análise de Entradas Impróprias**: nesta atividade são avaliadas as conseqüências envolvidas quando da ocorrência de entradas impróprias ao módulo, seja através de sinais de entrada errados, não de acordo com a especificação, seja através de

variações na alimentação do módulo, considerando os aspectos de diminuição, aumento e oscilação do nível de alimentação.

b) Análise de Perigo do Hardware Redundante

A Análise de Perigo do Hardware Redundante é constituída pelas mesmas atividades da Análise de Perigo do Hardware "Fail-Safe", acrescentando-se a Análise de Independência dos Canais Redundantes. O grande objetivo desta análise de independência é a determinação de focos de Falhas de Causa Comum. Na realidade este tipo de análise pode ser extremamente rígido conforme o tipo de aplicação envolvido e as recomendações apresentadas anteriormente.

c) Determinação dos Meios de Detecção e Recuperação de Falhas Implementados por Hardware.

Esses meios de detecção irão influenciar na determinação do Fator de Cobertura de falhas do sistema, e por conseqüência interferir na avaliação do grau de segurança avaliado. Esse meios de detecção são determinados a partir da análise do hardware redundante e "fail-safe".

d) Análise dos Aspectos Construtivos do Hardware

Esta atividade procura identificar possíveis focos de falhas que possam levar o sistema a uma condição perigosa.

3.3.3.2 Análise de Perigo do Software

Um dos grandes desafios refere-se à Análise de Perigo do Software. Esta etapa pode ser subdividida em Descrição Funcional do Software, Detalhamento dos Requisitos Gerais de Segurança, Elaboração da Descrição Funcional das Rotinas a partir do Código Fonte, Elaboração da Descrição das Variáveis Globais, Identificação dos Meios de Detecção e Recuperação de Falhas Implementados por Software, Inspeção Formal do Código Fonte, Reuniões Formais de Análise das Rotinas do Software, e Elaboração de Casos de Testes e Simulações.

a) Descrição Funcional do Software.

Nesta atividade é elaborada uma descrição funcional do software visando identificar a arquitetura utilizada e os atributos funcionais dos módulos envolvidos. Esta atividade tem importância fundamental no sentido de se fornecer uma visão funcional ampla do software.

b) Detalhamento dos Requisitos Gerais de Segurança

Nesta atividade são gerados os requisitos de segurança específicos para o software a partir dos requisitos gerais de segurança, procurando fornecer subsídios para as próximas atividades, visando a identificação da presença de estados perigosos.

c) Elaboração da Descrição Funcional das Rotinas a partir do Código Fonte.

Nesta atividade devem ser incluídas descrições textuais, diagramas de fluxo de dados (contexto e detalhados), diagramas estruturados NS, redes de Petri quando aplicáveis, em especial na representação de eventos concorrentes e sincronizados.

d) Elaboração da Descrição das Variáveis Globais

Nesta atividade são elaboradas a descrição funcional das variáveis globais, seu tipo e tamanho, no caso de vetores e matrizes, e sua relação de dependência em relação às rotinas, como ações de Leitura ou Escrita na variável em questão.

e) Identificação dos Meios de Detecção e Recuperação de Falhas Implementados por Software.

Esses meios de detecção irão influenciar na determinação do Fator de Cobertura de falhas do sistema, e por conseqüência interferir na avaliação do grau de segurança avaliado. Esses meios de detecção são selecionados através de uma classificação das rotinas de software do sistema computacional.

f) Inspeção Formal do Código Fonte

Esta atividade é realizada através da aplicação de uma Lista de Verificações, "Checklist", desenvolvido especialmente para uma linguagem sendo utilizada. Vale ressaltar que este tipo de análise não exige um conhecimento prévio da funcionalidade do sistema em análise, podendo ser realizado por especialistas em software sem conhecimento da aplicação prática.

A não observância de qualquer dos itens contidos nesta Lista de Verificações pode provocar a realização de processamento não correto, ou mesmo não previsto nas especificações do sistema sob análise. A verificação dos pontos apresentados na Lista de Verificações constitui-se já em um forte indício de que o código verificado tem possibilidade de atender aos requisitos mínimos para utilização em aplicações críticas.

Embora algumas linguagens, como por exemplo C, não tenham sua utilização recomendada, inclusive por normas internacionais, sua aplicação tem sido grande em sistemas críticos de segurança. Um dos motivos que levam a essa utilização é o fato de que há grande difusão e conhecimento no meio técnico dessas linguagens não totalmente recomendadas para aplicações críticas. Especialmente nestes caso é

extremamente útil a aplicação da inspeção formal, tendo como referência uma Lista de Verificação. [Lawrence 00]

Outro motivo é que a linguagem "assembly", que seria a mais recomendada pelo fato de se lidar quase que diretamente com o hardware do processador, tem a grande desvantagem de tornar os programas muito complexos, à medida em que as funções implementadas têm a sua complexidade aumentada, exigindo grande habilidade do programador nesta linguagem.

Além disso, o corpo técnico das organizações não tem, em geral, cultura apropriada ao projeto de sistemas críticos de segurança. Dentro desse quadro, a preocupação com a Inspeção Formal do código fonte é fornecer subsídios para uma utilização, das diversas linguagens, voltada para sistemas críticos.

g) Reuniões Formais de Análise das Rotinas do Software

Nesta atividade, cada uma das rotinas do software é avaliada em uma reunião formal de análise. Esta reunião tem a participação de um Moderador, um Relator, um Apresentador da Rotina e demais profissionais relacionados com a análise do software e do hardware do sistema. Cada uma dessas reuniões deve ter duração máxima de 2 (duas) horas, procurando manter ativo o senso crítico da equipe em relação aos requisitos de segurança, aspecto fundamental durante este tipo de atividade.

Durante essas reuniões, as rotinas são avaliadas e são realizadas simulações, conforme as necessidade envolvidas. As avaliações são realizados de acordo com critérios a seguir apresentados. Em alguns casos são necessários esclarecimentos junto ao operador do sistema sendo avaliado.

h) Elaboração de Critérios de Avaliação e Simulações

Os critérios básicos na planejamento destas avaliações e simulações são: cobertura lógica, verificação de valores inválidos e de fronteira, verificação de caminhos independentes e "error guessing". A técnica "error-guessing" é extremamente útil quando a equipe de análise apresenta vasta experiência com o sistema sendo analisado. Evidentemente nesta análise são incluídas as verificações dos "time-outs" críticos envolvidos na operação do sistema, sempre considerando o pior caso. Diversos métodos podem ser utilizados nesta atividade, podendo destacar, como exemplo, os métodos de árvore de falhas, árvore de eventos, redes de Petri, statecharts, análise de completeza e autômatos híbridos.

3.3.4 Análise de Perigo da Operação e Suporte:Operating Support Hazard Analysis - OSHA

Nesta etapa são identificados os perigos e os procedimentos de redução de risco durante a operação e manutenção. Em especial são analisados os perigos criados através da interface homem máquina.

Nesta etapa são avaliados os procedimentos operacionais visando identificar seqüências operacionais que possam levar o sistema a um estado perigoso e, portanto, devem ser evitadas ou pelo menos minimizada a sua possibilidade do ocorrência.

3.3.5 Análise Final do Perigo

A Análise Final do Perigo do sistema constitui-se na determinação do grau de segurança do sistema sendo analisado. Trata-se de uma integração entre a Avaliação Qualitativa e a Avaliação Quantitativa.

Na Avaliação Qualitativa, são apresentados os problemas encontrados no Software, no Hardware, nos Procedimentos Operacionais e no nível de sistema e subsistema.

Com relação ao software são apresentados os resultados de acordo com a seguinte classificação: problemas potencialmente perigosos, problemas que afetam a disponibilidade do sistema, problemas que afetam a manutenção do sistema, problemas relacionados com aspectos metodológicos além de aspectos estruturais como comentários errados, código não executado.

Com relação ao hardware, são apresentadas as conclusões da análise destacando as falhas ou entradas impróprias que podem levar o sistema a situações perigosas. São incluídas, nesta análise, as falhas no software decorrentes de falhas oriundas do hardware.

No que diz respeito aos Aspectos Operacionais são apresentadas as seqüências operacionais, considerando também as falhas no software e hardware, que podem levar o sistema a alguma condição perigosa.

São também avaliados os aspectos de manutenção corretiva no requisito de qualidade do alarme fornecido. Com relação aos aspectos de manutenção preventiva é avaliada a cobertura de falha dos testes realizados.

Neste momento, no nível de sistema e subsistema, em função de um maior conhecimento detalhado do projeto sendo avaliado, podem ser realizadas as simulações com o intuito de se verificar o atendimento aos Requisitos Gerais de Segurança. Evidentemente que se torna impraticável a modelagem de todo um sistema em função de sua complexidade. Neste sentido, este tipo de avaliação pode ser conduzido sobre

partes do sistema, escolhidas em função da criticidade envolvida com a segurança. Neste tipo de avaliação também podem ser utilizados os métodos como Verificações Formais, através de Model Checking, Statecharts, entre outros. Outra possibilidade de expansão dessa avaliação qualitativa é incluir, em seu método, algum aspecto de avaliação quantitativa, através do levantamento de probabilidades dos eventos envolvidos.

Na Avaliação Quantitativa calcula-se o grau de segurança do sistema representado através do valor do seu MTTUF – "Mean Time to Unsafe Failure", ou seja, Tempo Médio entre Disfunções Inseguras. Vale ressaltar que o Fator de Cobertura de Falhas, utilizado nesta avaliação, é decorrente da composição dos meios de detecção e recuperação de falhas implementados por software e por hardware, e já determinados nas suas respectivas análises.

3.4 Métodos para Realizar Análise de Perigo

É apresentado, neste item, um conjunto de métodos que podem ser utilizados durante o processo de Análise de Perigo. É importante destacar que estes métodos devem ser utilizados em conjunto e serem avaliados por especialistas. Vale dizer também que a melhor cobertura obtida na Análise de Perigo é conseqüência da combinação dos diversos métodos. Evidentemente existem diversas interseções lógicas entre os métodos. No entanto, a complementariedade entre eles é que garante uma maior cobertura no processo de Análise de Perigo. [Profit 95]

Estes métodos foram escolhidos em função de pesquisas realizadas pelo GAS, além de experiências já adquiridas pelo autor. A relação dos métodos não pretende ser completa, tendo a finalidade de dar uma visão ampla de diferentes métodos em diferentes fases do ciclo de vida do sistema. Foram também escolhidos alguns métodos, que embora ainda não tão detalhados, apresentam um futuro bastante promissor dentro do escopo das pesquisas aqui apresentadas.

3.4.1 Lista de Verificação

Esta técnica é derivada, normalmente, de normas, práticas de boa engenharia e da experiência adquirida através de diversos projetos. Esta lista de verificação pode ser uma referência para reflexão sobre o sistema que estiver sendo desenvolvido ou avaliado, devendo ser utilizada ao longo de todo o ciclo de vida do sistema.

Dentro do escopo da Análise de Perigo esta técnica pode ser utilizada na obtenção de maiores detalhes sobre os possíveis perigos, como também, na orientação de decisões de projeto, procurando, desta forma, diminuir os riscos envolvidos.

3.4.2 Árvore de Falhas

A técnica de Árvore de Falhas tem como objetivo fundamental estudar, em detalhes, a forma com que os perigos podem ser alcançados, através da combinação lógica de eventos primários. Neste sentido, a Árvore de Falhas não avalia a possibilidade de um novo perigo, mas estuda em detalhes os perigos já identificados.

O topo de uma Árvore de Falha constitui-se num perigo, ou até mesmo, em um acidente. Desta forma, a partir de um determinado Requisito Geral ou Específico de Segurança, adota-se como *Topo* desta árvore o evento da negação do respectivo requisito. Assim sendo, essa técnica utiliza-se de uma filosofia top-down de detalhamento.

Na construção de Árvores de Falhas são utilizados os conectivos lógicos, na maioria das vezes OR e AND para manter a simplicidade do seu entendimento. A Árvore de Falhas é detalhada até se atingir um evento primário, ou básico, ou se chegar a um determinado evento que deverá ser detalhado posteriormente. Após a construção de uma Árvore de Falhas, ela pode sofrer dois processos de avaliação: qualitativo e quantitativo. [Kececioglu 91]

A avaliação qualitativa tem como finalidade representar a ocorrência do perigo através de uma forma lógica equivalente, mostrando a combinação dos eventos básicos que podem causar o evento de topo. Pode-se, através desta avaliação, determinar a criticidade dos eventos envolvidos, podendo inclusive influenciar em decisões de projeto ao longo do desenvolvimento.

A avaliação quantitativa tem como meta avaliar a probabilidade de ocorrência do evento de topo em função das probabilidades de ocorrência dos eventos básicos. Neste sentido é de fundamental importância verificar a independência dos eventos básicos envolvidos.

A título de ilustração é apresentado na figura 3.3 um exemplo simples de Árvore de Falhas com suas respectivas avaliações qualitativa e quantitativa.

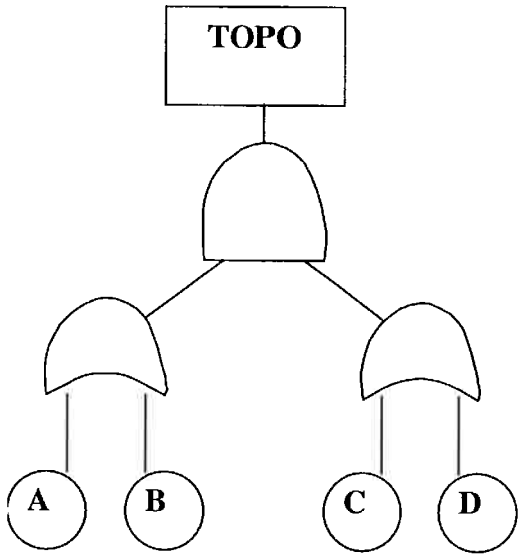


Figura 3.3. – Árvore de Falhas

Avaliação Qualitativa:

TOPO = (A + B) . (C + D)

A C + A D + B C + B D

Avaliação Quantitativa:

P (TOPO) = P (A) . P (C) + P (A) . P (D) + P (B) . P (C) + P (B) . P (D)
- P (A) . P (C) . P (D) - P (B) . P (C) . P (D) - P (A) . P (B) . P (C)
- P (A) . P (B) . P (D) + P (A) . P (B) . P (C) . P (D)

onde P(X) indica a probabilidade da ocorrência do evento X.

3.4.3 Árvore de Eventos

O método de Árvore de Eventos tem como principal meta avaliar quais eventos finais podem acontecer como conseqüência da combinação da ocorrência de eventos iniciais. Estes eventos iniciais podem se constituir em uma determinada falha de componente do sistema ou ocorrência de eventos externos. Para cada evento inicial devem existir, no mínimo, duas ramificações, uma com seu sucesso e outra com seu fracasso. Nesse sentido, este método tem como filosofia de detalhamento a técnica bottom-up, contrário ao método de Árvore de Falhas e, portanto, tem uma função complementar. Como para sua aplicação são necessárias maiores informações sobre eventos iniciais, este tipo de método é mais adequado de ser aplicado após o projeto detalhado. Na figura 3.4. é apresentada uma ilustração simplificada de aplicação deste método. Foram omitidas, nas probabilidades dos eventos, os termos (1 – Px), para efeito de simplificação.

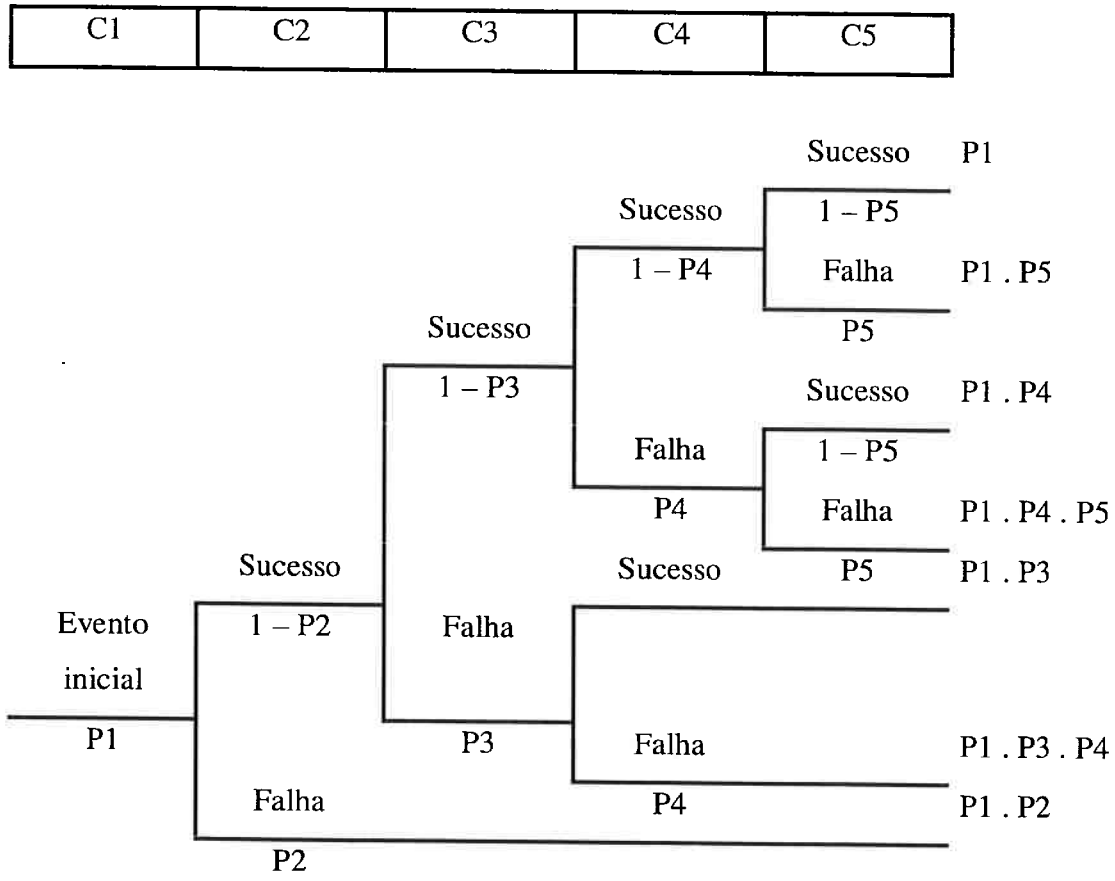


Figura 3.4 – Árvore de Eventos

3.4.4 Análise dos Efeitos dos Modos de Falhas - FMEA - Failure Modes and Effects Analysis

O método FMEA, tem como objetivo, identificar os efeitos dos modos de falha de um determinado componente do sistema. Este método pode ser aplicado sobre os níveis de sistema, subsistema ou módulos, como uma placa de hardware ou uma rotina de software. Quando é aplicado sobre o nível de sistema, os componentes a serem avaliados são os seus subsistemas constituintes. Quando o método é aplicado ao nível de subsistema os componentes são seus grandes módulos, sejam eles hardware ou software. Por outro lado, quando o método é aplicado sobre módulos específicos, como por exemplo uma placa de hardware, os componentes referem-se aos seus elementos básicos, como um circuito integrado, um capacitor, um resistor, uma memória, entre outros. Esta última forma de aplicação tem sido a mais comumente utilizada. A aplicação do FMEA sobre módulos de software, apesar de possível, tem sido pouco utilizada, devido à grande complexidade de sua aplicação em relação aos benefícios obtidos na qualidade do projeto. Na realidade, outros métodos são apresentados neste trabalho que melhor se aplicam à avaliação de módulos de software.

A tabela 3.4 apresenta um exemplo estrutural de uma tabela de FMEA que pode ser utilizada. Para cada componente considerado existe uma coluna que representa a sua probabilidade de falha. Em seguida existe outra coluna onde são detalhados todos os modos de falha possíveis de cada um dos componentes. Para cada um destes modos de falha são calculadas as probabilidades de sua ocorrência, seus efeitos locais e no sistema.

Componentes	Probabilidade de falha	Modos de falha	% por modo de falha	Efeitos Locais	Efeitos No Sistema	
					Critico	Não Crítico
A	1×10^{-3}	Aberto	90	Diminui Tensão		X
		Curto	5	Aumenta Tensão	5×10^{-5}	
		Outros	5	Aumenta Tensão	5×10^{-5}	

Tabela 3.4.- Tabela de FMEA

3.4.5 Análise Crítica dos Efeitos dos Modos de Falhas - FMECA - Failure Modes, Effects, and Criticality Analysis

O método FMECA constitui-se na adição, ao método FMEA, de uma análise mais detalhada da criticidade da falha. Este estudo mais detalhado pode envolver a determinação dos meios de controle da falha ou até mesmo a necessidade de projeto de um controle adicional visando a diminuição do risco envolvido.

3.4.6 Análise de Operação e Perigo - Hazard and Operability Analysis - HAZOP

Para se aplicar o método HAZOP não há a necessidade da identificação prévia dos perigos existentes. Por esta razão, é extremamente importante realizar um HAZOP preliminar para se determinar os Requisitos Gerais de Segurança. No que diz respeito à Análise de Perigo, o método HAZOP é confundido algumas vezes com o método FMEA. Quando o HAZOP é realizado, há sempre algum elemento de FMEA fazendo parte do método. O HAZOP constitui-se num exercício, realizado por uma equipe, para identificar as causas e as conseqüências dos perigos, enquanto que o FMEA examina apenas as conseqüências das falhas nos componentes e pode ser realizado por apenas

um profissional. O ponto inicial da aplicação do HAZOP é a avaliação de possíveis desvios da intenção original do projeto.

O HAZOP identifica o perigo enquanto o FMEA verifica se um determinado componente pode falhar e alcançar aquele perigo. O uso do HAZOP antes do FMEA pode resultar na realização de um FMEA mais focalizado e eficiente. [Redmill 99]

O aumento do conhecimento das condições de projeto, operacionais e ambientais provoca alguns questionamentos, como por exemplo:

- O projeto resolve adequadamente os perigos identificados previamente ?
- Algum novo perigo foi introduzido ?
- O conhecimento crescente do sistema permitiu que novos perigos fossem identificados?
- Quais informações adicionais podem ser obtidas sobre as causas potenciais e conseqüências dos perigos identificados anteriormente ?

O objetivo do HAZOP é, portanto, rastrear os perigos em potencial sobre o ambiente do sistema e determinar os Requisitos Gerais de Segurança, além de construir estratégias que possam ser utilizadas para implementar os requisitos e evitar a possível existência de um determinado perigo.

É o ambiente em que o sistema está mergulhado que irá determinar os perigos em potencial. Neste sentido, o software, por si só, não se constitui num perigo mas pode contribuir para a sua ocorrência.

A aplicação de HAZOP em Sistemas Eletrônicos Programáveis é uma inovação recente. Há pouco consenso entre os especialistas em engenharia de software e segurança em como analisar um sistema computacional de aplicação crítica quanto á segurança. O software é freqüentemente pouco considerado durante a análise dos sistemas. No entanto, tal aspecto é inaceitável quando o software desempenha funções de segurança. Neste sentido, a aplicação do HAZOP sobre especificações de software é bastante eficaz do ponto de vista de segurança, procurando identificar caminhos perigosos não facilmente localizáveis. [Lawrence 97] Desta forma, apontam-se duas questões:

- Se o software operar de acordo com sua especificação, qual o efeito sobre os perigos do sistema ?
- Se o software operar em desacordo com sua especificação, qual o efeito sobre os perigos do sistema ?

Neste sentido algumas questões devem ser consideradas quando o software apresenta um papel primordial na segurança do sistema:

- O software responde inadvertidamente a um estímulo;
- O software falha em responder quando requerido;
- O software responde fora da seqüência especificada; e
- O software responde de forma não planejada quando combinado com outras ações.

Dentro desta filosofia, há basicamente pelo menos quatro impactos potenciais do software com relação aos perigos já identificados:

- O software pode comprometer a segurança do sistema: falha na operação do software tem o potencial de criar condições de perigo que devem ser removidas ou amenizadas através de outros sistemas;
- O software pode ser responsável em prevenir que perigos evoluam para acidentes;
- O software pode ser utilizado para transpor o sistema de um estado perigoso para um estado não perigoso; e
- O software pode ser utilizado para amenizar as conseqüências de um acidente.

Outros aspectos importantes relacionados com o software, como por exemplo, ferramentas, linguagens, técnicas de codificação devem ser examinadas de forma a avaliar o seu potencial de introduzir falhas de causa comum a todos os módulos redundantes.

Desta forma, existem no HAZOP as *Guide Words* cujo objetivo é focalizar o estudo e maximizar as chances de identificar algum perigo.

Na realidade cada *Guide Word* pode ter mais de uma interpretação no contexto da representação do projeto. O objetivo destas palavras é focalizar melhor a análise permitindo, desta forma, maximizar as chances de identificação dos perigos. São aplicadas sobre os sinais de dados e controle.

As *Guide Words* consideradas são [Redmill 97]:

NO : nenhum dado ou controle é fornecido;

MORE: Os dados são fornecidos a uma taxa superior do que a esperada, ou mais dados são fornecidos;

LESS: Os dados são fornecidos a uma taxa menor do que a esperada, ou menos dados são fornecidos;

AS WELL AS: Os objetivos foram alcançados, mas com resultados adicionais;

PART OF: Os dados ou controle estão incompletos;

REVERSE: Ocorre fluxo reverso;

OTHER THAN: Os dados ou controle estão completos, mas incorretos;

EARLY: O sinal chega mais cedo que o esperado;

LATE: O sinal chega mais tarde que o esperado;

BEFORE: O sinal chega antes, dentro de uma seqüência de eventos esperados; e

AFTER: O sinal chega depois, dentro de uma seqüência de eventos esperados.

3.4.7 Avaliação de Completeza de Especificações

Este método foi desenvolvido na tese de doutorado do autor, sendo apresentado, nesta seção, um breve resumo do trabalho de pesquisa, com o intuito de destacar este tipo de análise como um método importante dentro de um processo de Análise de Perigo.

Para uma especificação de requisitos, seja de software, hardware ou sistema, ser considerada robusta, são necessários certos atributos fundamentais: não ser ambígua, ser completa, verificável, consistente, modificável, rastreável e utilizável durante as fases de operação e manutenção.[IEEE - Std 830-1993]

A segurança é definida como uma probabilidade de o sistema não atingir um estado perigoso, que pode afetar vidas humanas e bens materiais. De acordo com alguns pesquisadores dessa área, a análise probabilística da segurança, ou seja, a análise quantitativa, não é muito adequada para assegurar ou avaliar a segurança de um sistema. [LEVESON 83], [HAMLET 92], [BUTLER 93] Esse grupo adota a postura de garantir ou melhorar a segurança através de métodos específicos de projeto.

Em função também da experiência adquirida por diversos engenheiros de segurança de sistemas, concluiu-se que a grande causa de problemas de segurança em sistemas críticos é a existência de uma especificação inadequada de requisitos, ou seja, falta de robustez, além dos erros na aplicação de técnicas de engenharia.[LEVESON 86], [RUSHBY 94], [SHELDON 92]

Pela experiência profissional do autor, também adquirida através de diversos trabalhos de análise de risco de sistemas críticos, especificamente metro-ferroviários, é possível concluir que grande parte das possíveis falhas inseguras ocorre devido à falta de especificação com relação ao universo da falha em questão, ou a possíveis mudanças no ambiente operacional.

Em razão dessas observações, pode-se dizer que uma melhoria na análise da completeza de uma especificação deverá aumentar muito a qualidade das mesmas em diversos

b) Critérios para os Estados

- **Critério 2.1**: O Sistema deve iniciar e finalizar num estado seguro. O sistema de segurança deve estar apto a funcionar na iniciação e na finalização do sistema, incluindo também, nesse aspecto, a reiniciação após um período fora de operação.

Com relação às condições de iniciação, existem duas situações básicas: iniciação após um processo completo de desligamento e após um desligamento temporário, as quais caracterizam o que se denomina de iniciação a "frio" e a "quente", respectivamente. Em ambos os casos, muitos acidentes acontecem em função de um processamento não adequado.

- **Critério 2.2**: O comportamento do sistema com relação às entradas antes da iniciação, após o desligamento e durante o desligamento, deve ser especificado, seja detalhando a forma de utilização dessas entradas, ou podendo ignorá-las de maneira a não comprometer a segurança do sistema.

A utilização dos modos de operação auxilia na simplificação da descrição da operação do sistema, permitindo uma visão de alto nível do fluxo de controle do mesmo. Em sistemas de segurança crítica, os modos de operação estão associados a estados operacionais que são subdivididos em estados seguros e estados perigosos. Os estados perigosos podem ainda ser classificados em estados de alto risco e estados de baixo risco, conforme a probabilidade de serem levados a estados perigosos. Dessa forma, as atitudes a serem tomadas quando se atinge um estado perigoso podem diferir dependendo do modo de operação em que se encontra o sistema.

- **Critério 2.3**: Os caminhos a partir de um estado seguro devem ser especificados. Além disso, deve ser minimizado o tempo de permanência num estado seguro de funções reduzidas, ou seja, um estado em que a degradação é caracterizada pela não habilitação de algumas funções. Deve também ser minimizado o tempo de permanência num estado perigoso.
- **Critério 2.4**: Falhas no sistema de segurança devem provocar a transição do sistema para um estado seguro degradado, com o desligamento das funções que envolvem risco.
- **Critério 2.5**: A falha no sistema deve provocar a transição do mesmo para um estado seguro.

aspectos, especialmente no que diz respeito à segurança. Dentro desse enfoque, desenvolveu-se um método de avaliação da completeza de uma especificação de um sistema crítico em relação ao ambiente de aplicação.

Os critérios de completeza são classificados em função de suas características sobre o modelo de transição de estados e devem ser aplicados conforme o nível de detalhe em que se analisa a especificação, ou seja, nos níveis de sistema, subsistema, módulo ou componente.

Tendo como referência básica o trabalho de [JAFFE 91], e adaptando-se aos aspectos propostos neste trabalho, criaram-se seis categorias básicas de Critérios para verificar a completeza:

- Variáveis de Entrada/Saída;
- Estados;
- Predicados de Entrada;
- Predicados de Saída;
- Relação Entrada/Saída; e
- Transições.

a) Critérios para as Variáveis de Entrada/Saída

- **Critério 1.1**: As variáveis de entrada/saída devem ser utilizadas, caso contrário não haveria a necessidade de existirem. Uma variável de entrada qualquer deve ser utilizada em algum predicado de entrada de uma transição. Da mesma forma, uma variável de saída deve ser utilizada em algum predicado de saída de uma transição. Se isso não acontecer, ou a variável de entrada/saída não deve fazer parte do sistema, ou então existe uma omissão na especificação.
- **Critério 1.2**: Seja **v** uma variável de entrada ou saída. A **validade(v)** reflete a sua validade ou consistência. Esse atributo pode ser usado, por exemplo, para especificar valores aceitáveis, paridade, precisão das variáveis. Quando do não atendimento a essas verificações, uma resposta adequada deve ser declarada, ou, pelo menos, ser armazenada num arquivo histórico, para posterior análise "off-line", conforme for mais conveniente.

c) Critérios para os Predicados de Entrada

Para que o sistema seja robusto, é necessário também que haja um caminho previsto para todas as entradas em cada estado. Outro aspecto importante, já comentado anteriormente, é a temporização na especificação de sistema críticos, em função do instante da ação. Dessa forma, os critérios para os predicados de entrada estarão organizados na seguinte classificação: aspectos Lógicos, de Temporização e de Capacidade, que são explicados nos seus respectivos itens.

c.1) Aspectos Lógicos

Os aspectos lógicos relacionam-se com a consistência lógica desses predicados dentro do modelo de transição de estados adotado. Assim, enquadram-se dentro dessa categoria os seguinte critérios:

- **Critério 3.1**: Cada estado deve ter uma transição definida para cada entrada possível.

- **Critério 3.2**: O "OR" lógico dos predicados de entrada nas transições de saída de qualquer estado deve constituir uma tautologia; caso contrário, constata-se a presença de alguma situação não prevista no sistema.
- **Critério 3.3**: Para haver um comportamento determinístico do sistema, é necessário que, num determinado instante, seja verdadeiro apenas um predicado de entrada, correspondente a uma determinada transição a partir do estado corrente do sistema.

c.2) Aspectos de Temporização

Os aspectos de temporização englobam as condições relacionadas com os intervalos válidos para as entradas, bem como para a ausência desses sinais. Assim, dentro dessa categoria, enquadram-se os seguintes critérios:

- **Critério 3.4**: Em cada estado o sistema deve ter um comportamento bem definido, ou seja, deve executar alguma transição no caso de não haver entradas por um período de tempo denominado "Time Out". Esse aspecto pode ser também utilizado na reiniciação do sistema quando do não recebimento de determinadas entradas.

- **Critério 3.5**: Todos os eventos de entrada devem ser totalmente limitados no tempo, através da especificação de um tempo mínimo e de um tempo máximo, para sua coerência.

Deve ser especificada uma saída para a ausência do sinal por um período de tempo em relação a algum evento observável. Assim sendo, deve ser estabelecido um tempo específico a partir do qual se inicia a temporização correspondente à ausência de entradas.

- **Critério 3.6**: Para um estado qualquer, deve existir uma transição de saída, cujo predicado de entrada envolva a não existência de uma determinada entrada durante um intervalo específico em relação a algum evento observável.

c.3) Aspectos de Capacidade

Embora as entradas fornecidas ao sistema pelo operador ou por um outro sistema dificilmente causem alguma sobrecarga no sistema avaliado, outros problemas, devidos a mal funcionamento, podem causar esse excesso, ultrapassando o limite de capacidade. A robustez, nesse caso, requer que se especifique a forma de se manipularem entradas excessivas e determina um limite de capacidade para tais entradas, como meio de se detectarem possíveis problemas externos e internos ao sistema em questão. Isso implica em se determinar o número máximo de entradas dentro de um certo intervalo de tempo. Para os casos em que a capacidade é excedida, deve haver alguma especificação da maneira como o sistema poderá falhar.

Software e hardware requerem que uma declaração seja feita sobre o número máximo de entradas **N** por um período de tempo de duração **T.**

A capacidade de entrada **Ce** depende do processo controlado e dos sensores que enviam informações sobre o processo. Num sistema de controle, essa capacidade de entrada subdivide seus estados em: estados normais e estados de sobrecarga.

Dessa forma, enquadram-se dentro dessa categoria os seguintes critérios:

- **Critério 3.7**: Uma variável de entrada no sistema requer uma declaração de capacidade.
- **Critério 3.8**: Declarações de capacidade mínima e máxima devem ser especificadas para cada variável de entrada, cuja taxa de chegada não seja limitada por outro tipo de evento.

- **Critério 3.9**: Deve ser requerida uma verificação das taxas mínima e máxima de chegada de eventos para cada caminho de comunicação fisicamente distinto. O sistema deve ser capaz de supervisionar o ambiente de comunicação.
- **Critério 3.10**: As respostas às entradas que excedem a capacidade do sistema devem ser especificadas. Essas respostas devem se enquadrar em alguma das classes detalhadas a seguir:

– Requisitos para gerar mensagens de advertência, ou seja, avisos aos operadores do sistema para adotarem atitudes operacionais que amenizem as conseqüências da sobrecarga.

– Requisitos para gerar sinais controle para sistemas externos visando a diminuição de capacidade, "slowdown". Nesse caso, a sobrecarga pode ser limitada através de um controle indireto dos sistemas externos. É fundamental a interface de comunicação entre os sistemas envolvidos.

– Requisitos para bloquear os canais em sobrecarga, "lockout", havendo, dessa forma, uma degradação do sistema quanto ao tratamento das informações bloqueadas.

– Requisitos para reduzir a precisão, o tempo de resposta, ou outra característica que permita ao sistema processar uma capacidade maior com relação às entradas. Nesse aspecto, o sistema passa para um estado de degradação, diminuindo a qualidade do processamento das informações de entrada.

– Requisitos para reduzir a funcionalidade do sistema ou, em casos extremos, solicitar o desligamento do controle ou, até mesmo, do processo, quando necessário.

- **Critério 3.11**: Se a degradação ou um processo de reconfiguração é utilizado como meio de atender a uma capacidade excessiva, deve ser especificado um atraso referente à histerese para que o sistema possa retornar ao processamento com carga normal. Esse atraso devido à histerese tem, por objetivo, evitar o chaveamento indevido entre carga normal e sobrecarga, também denominado "efeito ping-pong".

d) Critérios para os Predicados de Saída

Para que uma especificação seja robusta em relação aos predicados de saída, deve-se verificar os limites de tempo inferior e superior das saídas. Esses requisitos estão

classificados de acordo com a seguinte estrutura: capacidade do ambiente, envelhecimento da informação e latência.

d.1) Capacidade do Ambiente

Define-se a capacidade do ambiente como sendo a taxa máxima para a qual os atuadores podem aceitar e reagir aos dados produzidos pelo controlador. Esse valor-limite é afetado pelos seguintes elementos:

- limitação de capacidade dos próprios atuadores;
- limitações no comportamento do processo controlado; e
- considerações de segurança.

Dentro dessa filosofia, enquadram-se os seguintes critérios:

- **Critério 4.1**: A capacidade do ambiente deve corresponder à capacidade máxima de saída do sistema. Se a capacidade máxima de saída do sistema for excedida, é necessária a realização de alguma ação especial, visando normalizar a taxa de saída.

d.2) Envelhecimento da Informação

Outro aspecto importante envolve a obsolescência das informações. As decisões de controle devem ser baseadas em dados atualizados do estado do sistema.

- **Critério 4.2**: Todas as entradas utilizadas na especificação de predicados de saída devem ser adequadamente limitadas no tempo, para garantir, dessa forma, o seu não envelhecimento.
- **Critério 4.3**: A seqüência de ações perigosas deve ser limitada no tempo, após o qual o sistema deve requerer o cancelamento automático dessas ações e informar ao operador.

d.3) Latência da Informação

Dado que um sistema de controle não é arbitrariamente rápido, há um intervalo de tempo durante o qual o recebimento de uma nova informação não pode mudar a variável de controle de saída, mesmo que chegue antes dessa geração. Esse intervalo de tempo é influenciado pelo hardware e pelo software do sistema de controle, podendo ser bastante pequeno, mas nunca reduzido a zero. A escolha do sistema operacional a ser utilizado, a lógica de interrupção, a prioridade no escalonamento de tarefas e os diversos parâmetros

de projeto serão influenciados pelo valor máximo permitido para esse intervalo de latência.

- **Critério 4.4**: Um fator de latência deve ser incluído na especificação quando uma saída não restritiva é disparada pela ausência ou pela chegada de uma entrada específica, para a qual o limite superior desse intervalo de tempo não é um evento observável simples, em função de prováveis interferências ou da própria dinamicidade do sistema de controle.
- **Critério 4.5**: Ações contingentes podem ser necessárias na especificação, para tratar os eventos de entrada que ocorrem dentro do período de latência.
- **Critério 4.6**: Um fator de latência deve ser especificado para mudanças ocorridas na interface homem máquina, quando usadas para decisões críticas. Da mesma forma, as ações contingentes devem ser especificadas quando da mudança das informações na interface homem máquina durante o período de latência. Esse critério está relacionado com o tempo de arrependimento do operador.
- **Critério 4.7**: Um período de histerese deve ser especificado, correspondente ao atraso da ação do operador na interpretação das informações a ele apresentadas. Devem ser especificadas as ações caso as informações apresentadas, ao operador, mudem durante esse período de histerese.

e) Critérios para a Relação Entrada/Saída

Existem requisitos que estão relacionados com Entrada/Saída, podendo ser classificados de acordo com os seguintes aspectos:

- Capacidade de resposta; e
- Espontaneidade.

A Capacidade de resposta e a Espontaneidade lidam com o comportamento do processo e, como ele, reage às saídas produzidas pelo sistema de controle. Esses parâmetros são supervisionados pela realimentação.

- **Critério 5.1**: Deve haver variáveis supervisionadas que detectem o efeito das variáveis de controle. As informações sobre a característica do processo devem ser utilizadas como características preditivas do comportamento esperado do sistema.

- **Critério 5.2**: Para cada variável de controle que possui uma variável de supervisão de comportamento esperado, devem ser avaliadas as condições para resposta muito demorada ou muito rápida.
- **Critério 5.3**: O recebimento espontâneo de uma entrada, apenas esperada em resposta a alguma saída anterior do sistema, deve ser detectada e respondida como uma situação anormal, realizando alguma ação contingente.

f) Critérios para as Transições

Em particular, os requisitos relacionados com as transições servem para garantir que certos estados sejam alcançáveis.

As características básicas consideradas são Alcance Básico, Comportamento Recorrente, Reversibilidade e Alcance de Estados Seguros.

f.1)Alcance Básico

- **Critério 6.1**: Todos os estados devem ser alcançáveis a partir de um estado inicial. Caso um estado não seja alcançável, há duas possibilidades:
  - o estado não tem função e pode ser eliminado da especificação; e
  - o estado deve ser alcançável e a especificação precisa ser modificada.

f.2)Comportamento Recorrente

- **Critério 6.2**: O comportamento recorrente desejável deve ser parte de um ciclo, ou seja, deve ser possível atingir um estado, de forma coerente, a partir dele mesmo e através de um caminho não vazio, passando por outros estados. Esse critério passa a ter um papel fundamental se o estado em questão for de importância para a segurança do sistema.

f.3) Reversibilidade

A reversibilidade está relacionada com a capacidade de os comandos sobre os atuadores serem cancelados ou revertidos por algum outro comando ou combinação.

- **Critério 6.3**: A reversibilidade de um procedimento de operação no sistema, por outro procedimento, requer a existência de caminhos possíveis entre os estados atingidos e revertidos.

f.4) Alcance dos Estados Seguros

Nesse item, são avaliados os critérios específicos relacionados com os estados seguros e as ações esperadas caso os mesmos, por algum motivo, não possam ser alcançados.

- **Critério 6.4**: Não deve haver caminho coerente para um estado perigoso.

Dado que o sistema atingiu um estado perigoso, seja devido a falhas de componentes ou do sistema computacional, erro humano ou distúrbio externo, entre outros, o sistema de controle deverá conduzir o processo para um estado seguro. A decisão da ação adequada a ser tomada deverá depender das condições ambientais em que o processo se localiza no momento.

- **Critério 6.5**: Todo caminho a partir de um estado perigoso deve conduzir a um estado seguro.

Pode não ser possível construir um sistema intrinsicamente seguro, ou seja, "fail-safe". Nesse caso, o sistema deverá transformar o estado perigoso em um estado de mínimo risco aceitável, aspecto relacionado com o MTTUF do sistema.

- **Critério 6.6**: Se um estado seguro não puder ser alcançado a partir de um estado perigoso, todos os caminhos a partir desse estado perigoso devem levar o sistema a estados de mínimo risco aceitável.

3.4.8 Métodos Semi Formais

De acordo coma norma IEC 61508, os Métodos Semi Formais são aqueles que fornecem meios de se desenvolver uma descrição do sistema em algum nível de seu desenvolvimento. A descrição pode, em alguns casos, ser analisada automaticamente ou com animação, mostrando vários aspectos do comportamento do sistema. O aspecto de animação pode fornecer informações adicionais, mostrando que o sistema atende os requisitos especificados. Os métodos apresentados aqui são Diagrama de Transição de Estados, Redes de Petri e Statecharts.

a) Diagrama de Transição de Estados

O Diagrama de Transição de Estados é especialmente apropriado para sistemas seqüenciais. A figura 3.5 apresenta um exemplo de formalização deste método, onde **Ci↑** representa o evento condição da ocorrência da transição e **Si↑** representa o evento saída gerado na ocorrência da transição.

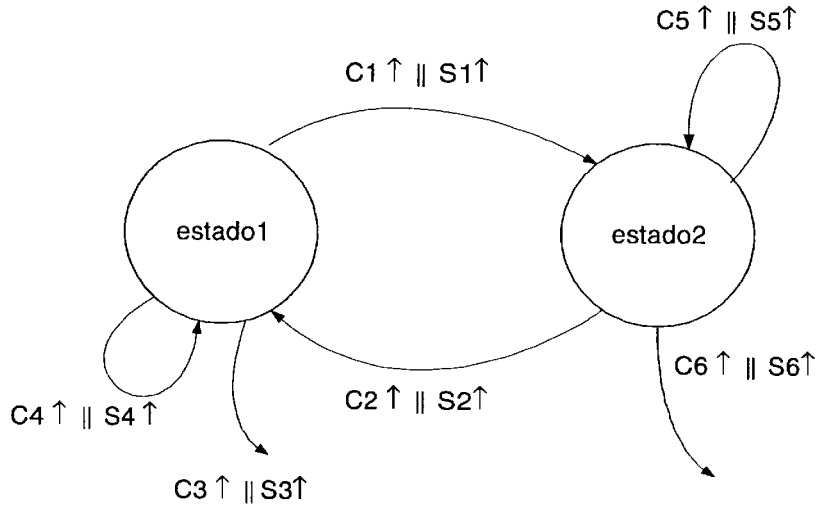


Figura 3.5 – Diagrama de Transição de Estados

b) Redes de Petri

As Redes de Petri são especialmente apropriadas para representar sistemas com alto grau de concorrência. Os elementos de uma Rede de Petri são:

NÓ

TRANSIÇÃO

CAMINHOS.ARCOS

MARCAS - TOKEN

As Regras básicas de sua formação são:

- Uma transição é permitida quando todos os estados que possuem arcos orientados para a transição, contêm marcas;

- Quando uma transição é permitida, ela é disparada, retirando-se uma marca de cada um dos estados de entrada da transição e colocando-se uma marca em cada um dos estados para os quais existe um arco orientado que sai da transição;
- Um estado qualquer nunca pode estar ligado por outro arco orientado a outro estado; e
- Uma transição qualquer nunca pode estar ligada por um arco orientado a outra transição.

Na figura 3.6 é apresentado um modelo de um cruzamento de uma ferrovia com uma rodovia através de Redes de Petri. Através de sua execução, pode ser gerado o Grafo de Alcançabilidade, conforme apresentado na figura 3.7. Através deste Grafo de Alcançabilidade podem ser identificados os estados perigosos (X X P3 X X P11 X X) e avaliados métodos de controle destes perigos.

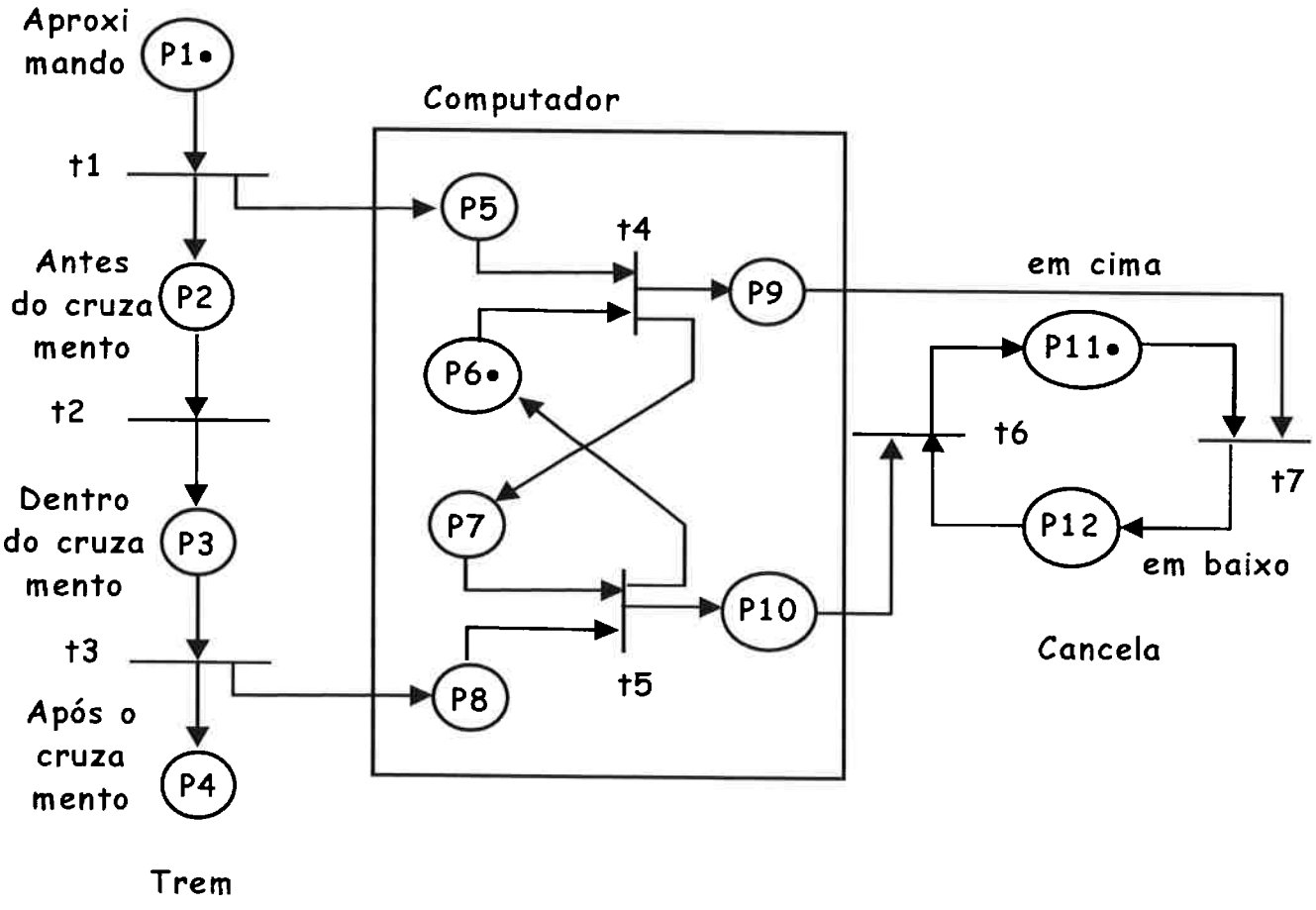


Figura 3.6 – Modelo de Cruzamento através de Rede de Petri

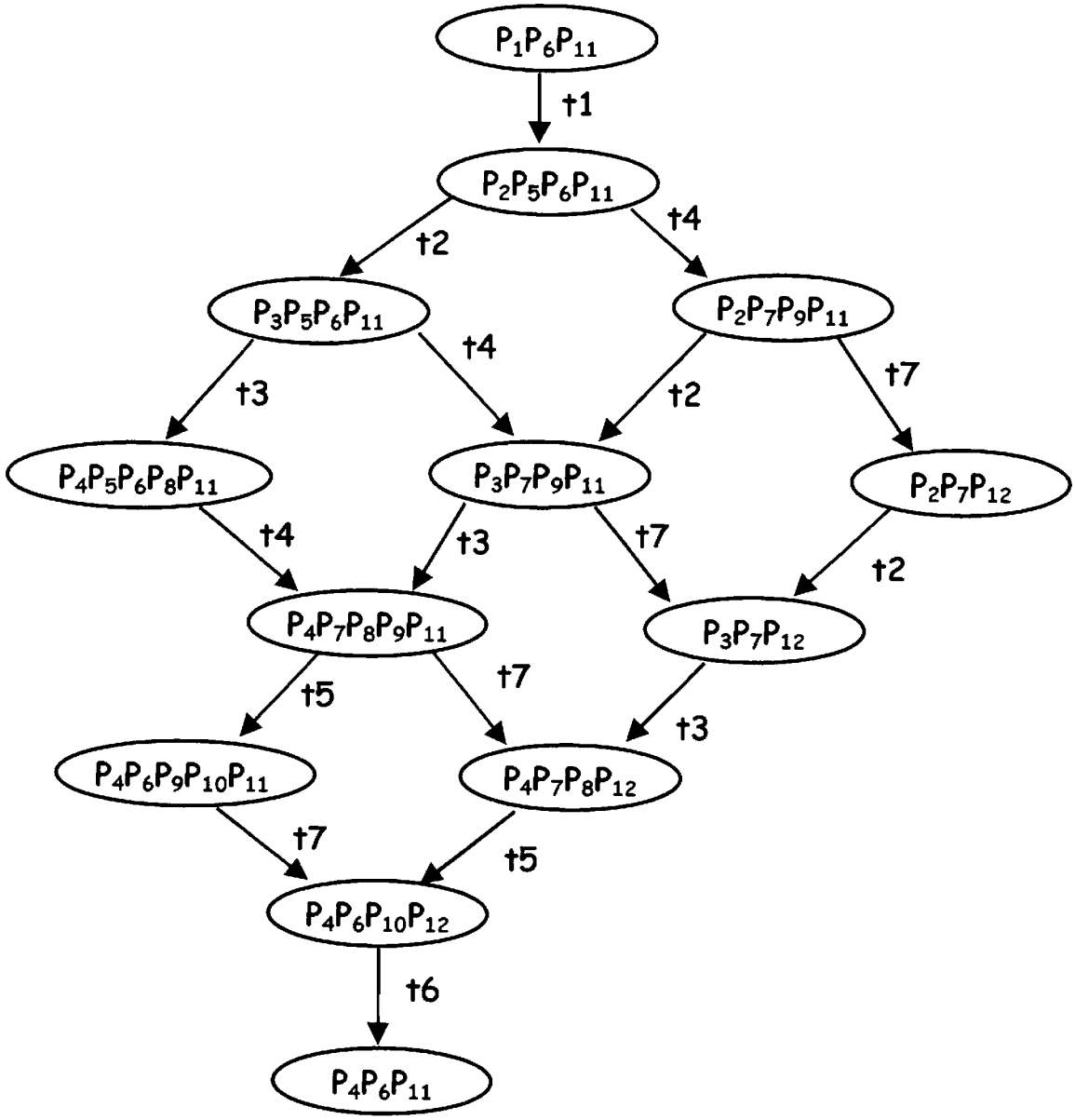


Figura 3.7 – Grafo de Alcançabilidade do Cruzamento através de Rede de Petri

c) Statecharts

Este método engloba as vantagens do Diagrama de Transição de Estados com a Rede de Petri, permitindo, desta forma, representar o comportamento seqüencial e concorrente. Neste formalismo as bolhas AND correspondem às linhas tracejadas enquanto que as bolhas XOR correspondem às linhas cheias.

Se uma determinada bolha estiver ativa (for o estado corrente) e tiver sido decomposta em outras bolhas XOR, então apenas uma dessas bolhas XOR (bolhas descendentes) será ativada.

Se por outro lado, ela tiver sido decomposta em bolhas AND, todas essas bolhas descendentes (bolha AND) serão ativadas quando seu antecessor for ativado. Desta

maneira, as bolhas AND são um meio de descrever o comportamento paralelo de forma explícita.

Uma transição acontece na dependência de condições restritivas e na ocorrência de eventos específicos, sendo representada por meio de arcos direcionados. Os eventos e condições restritivas que disparam uma transição, bem como as ações específicas a serem realizadas quando de uma ocorrência, são representados como atributos do arco correspondente. Ações associadas às transições de estado podem ser utilizadas para gerar novos eventos, os quais podem ter efeito sobre outras transições dos componentes concorrentes. A comunicação entre os componentes concorrentes pode ser estabelecida pela geração de eventos internos, ou seja, gerados diretamente pelos processos representados pelas Bolhas, e portanto internamente ao Statechart.

Além das ações, um atributo de história pode ser associado às bolhas. Atributos de história levam ao processo de ativação de bolhas. Sempre que uma bolha com o atributo de história for ativada, verifica-se em seu Statechart de decomposição quais foram as bolhas que estavam ativas na última vez em que se realizou a simulação de tal detalhamento, com a finalidade de que se ativem tais bolhas. Se não houver uma história nos descendentes ou se for a primeira vez em que a bolha com o atributo de história foi ativada, então seus descendentes "default', se existirem, é que serão ativados.

Como exemplo de aplicação é apresentado na figura 3.8 a modelagem do sistema de cancela, anteriormente formalizado através de Rede de Petri.

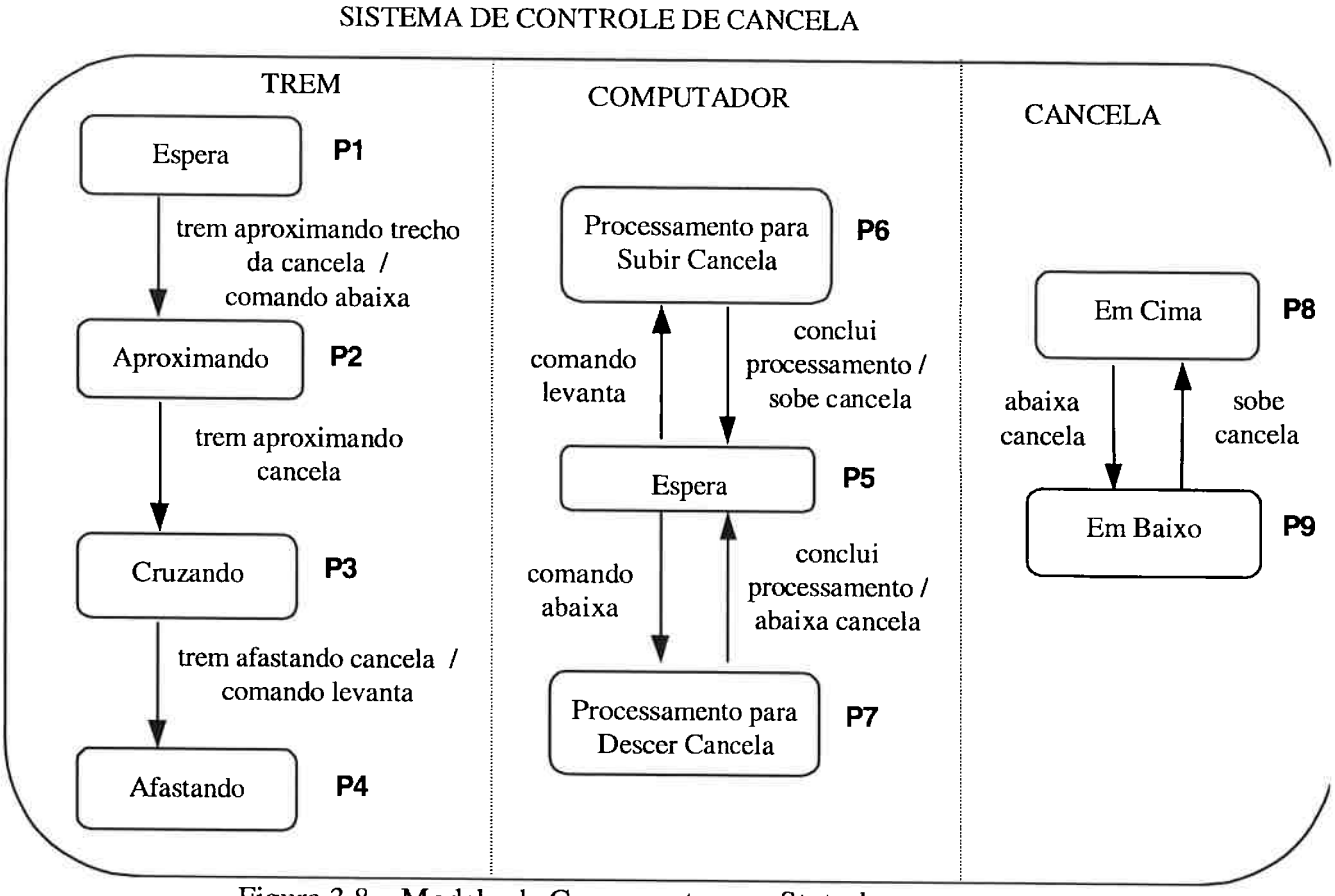


Figura 3.8 – Modelo de Cruzamento com Statechart

Na figura 3.9 é apresentado o Grafo de Alcançabilidade correspondente a esse novo modelo, onde o estado (P3 P7 P8) corresponde a um estado perigoso.

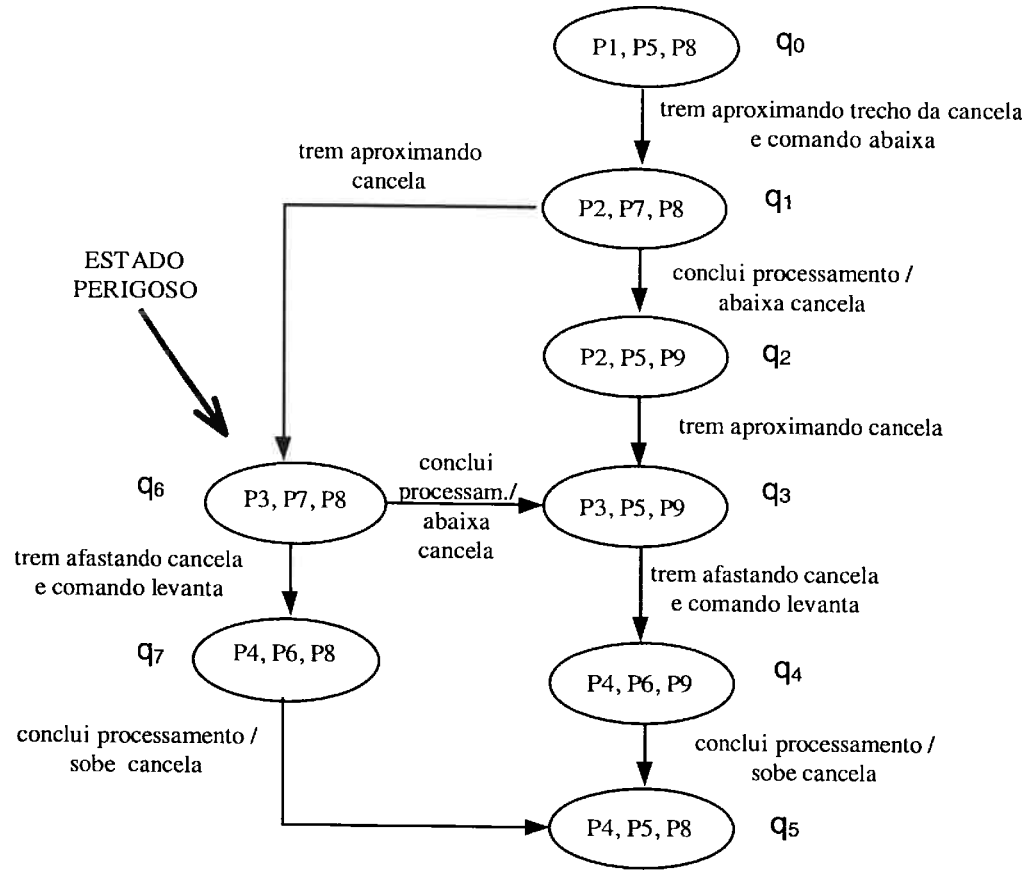


Figura 3.9 – Grafo de Alcançabilidade do Cruzamento Modelado com Statechart

3.4.9 Métodos Formais

De acordo coma norma [IEC 61508] um Método Formal é aquele que fornece meios de desenvolver uma descrição do sistema em algum nível de seu desenvolvimento. Esta descrição resultante apresenta uma forma matemática e pode ser submetida a análise matemática com o intuito de detectar várias classes de inconsistências ou incorreções.

Com o aumento da complexidade do software e do hardware, aumenta-se a probabilidade da presença de erros ainda não detectados. Especial atenção deve ser dada quando esses erros fazem parte de um sistema crítico quanto a segurança.

Um dos grandes objetivos da engenharia de software é permitir aos projetistas a construção de sistemas com maior segurança e confiabilidade, apesar da grande complexidade. Um dos meios de se atingir esse objetivo são os Métodos Formais.[Broomfield 97]

O Uso dos Métodos Formais não garante a correção *a priori*. Entretanto, eles podem aumentar bastante o entendimento do sistema, revelando inconsistências, ambigüidades ou incompletezas, que poderiam continuar não detectáveis.

As duas grandes abordagens dentro deste campo são: ESPECIFICAÇÃO E VERIFICAÇÃO. [Clarke]

A Especificação corresponde ao processo de descrever o sistema e suas propriedades. A especificação formal utiliza-se de linguagem com sintática e semântica matematicamente definidas. As características formalizadas podem incluir Comportamento Funcional, Temporal, Características de Desempenho ou Estrutura Interna.

Uma linha atual de trabalho corresponde à integração de diferentes linguagens de especificação, cada uma permitindo a manipulação de aspectos diferentes do sistema, enquanto outras manipulam aspectos de desempenho, limitações de tempo real, políticas de segurança e projeto de arquitetura.

Como exemplos de formalismos de especificação podem ser citados:

- Z: usado na especificação do comportamento de sistemas seqüenciais;
- CSP, CCS, Temporal Logic, I/O Automata: utilizados na especificação do comportamento de sistemas concorrentes.

Já o processo formal de Verificação apresenta duas abordagens: Verificação de Modelos , "Model Checking" [Henzinger], [Atlee], [Henziger 1] e Prova de Teorema. O processo de Verificação caminha além da Especificação, pois ela é utilizada para analisar um sistema sob o enfoque de propriedades desejadas.

A Verificação de Modelos corresponde ao método que constrói um modelo finito de um sistema e verifica que certas propriedades desejadas são atendidas pelo modelo. Na realidade, a verificação é realizada como uma pesquisa exaustiva no espaço de estados com término garantido em função de se tratar de um modelo finito. Este método teve seu início na verificação de hardware e protocolos e a tendência atual é sua aplicação na análise da especificação de software. Em contrapartida à Prova de Teorema, a Verificação de Modelos pode ser automática e mais rápida. Pode ser usada para verificar a especificação por etapas, auxiliando em sistemas que ainda não foram totalmente especificados.

A principal desvantagem da Verificação de Modelos corresponde ao problema do número muito grande de estados. As ferramentas para Verificação de Modelos atuais têm uma expectativa de manipular sistemas de 100 a 200 varáveis de estado, gerando uma quantidade enorme de estados alcançáveis.

Há duas abordagens utilizadas para a Verificação de Modelos: A Verificação de Modelos Temporais, "Temporal Model Checking" e o Autômato, "Automaton".

Na abordagem do temporal, a especificação é expressa numa lógica temporal e os sistemas são modelados como máquinas de transição de estados finitos. Um procedimento eficiente é a utilização para a verificação se o modelo de transição de estados finitos corresponde a um modelo para a especificação. Neste tipo de representação se enquadram os Autômatos Híbridos, descritos no item 3.4.9.1.

Por outro lado, na abordagem por Automato, o autômato do sistema/subsistema é comparado com a especificação para verificar se seu comportamento está em harmonia com a especificação.

A Prova de Teorema, por outro lado, é uma técnica onde tanto o sistema como suas propriedades desejadas são expressas através de fórmulas em alguma lógica matemática. São definidos um conjunto de axiomas e um conjunto de regras de inferência. A Prova de Teorema é um processo de provar uma propriedade a partir dos axiomas do sistema. Essas provas podem ser auxiliadas por ferramentas computacionais denominadas Provadores de Teorema Semi-Automáticos, "Machine-Assisted Theorem Proving". Como estas ferramentas requerem interação com o ser-humano, seu processo é lento e sujeito a erros. Por outro lado, o projetista ganha uma visão privilegiada do sistema e de suas propriedades. Esse provadores têm sido utilizados na verificação matemática de propriedades de sistemas críticos, tanto em projeto de hardware como de software. Em contraste com o método de Verificação de Modelos, a Prova de Teorema pode manipular com espaços infinitos de estados. A técnica corresponde à indução estrutural para provar aspectos sobre um domínio infinito.

Todos os métodos formais, apresentados a seguir, podem gerar todos os caminhos possíveis a partir de um determinado estado do sistema, além de determinar se alguns dos estados alcançáveis são ou não perigosos. Em função do nível de abstração técnico do método, ele pode ser mais aplicável ao sistema, onde o nível de abstração exigido é maior ou em partes do sistema mais críticas. Em cada método será apresentada uma breve discussão sobre sua aplicabilidade mais adequada. [Bowen 96].

3.4.9.1. Autômatos Híbridos

Sistemas Híbridos são sistemas cujo comportamento apresenta uma combinação de características discretas e contínuas. Autômatos Híbridos são autônomos finitos cujos estados descrevem o comportamento dinâmico do sistema modelado. Transições entre estados caracterizam uma mudança de perfil dinâmico e são sinalizadas por eventos , ou mensagens, recebidos de outros componentes ou do ambiente externo. [Alur]

Os vários componentes que fazem parte do sistema são modelados por autônomos independentes, sendo o comportamento do sistema como um todo obtido do produto dos autônomos componentes.

Para efeito de um melhor entendimento e com o intuito de apresentar o formalismo envolvido é apresentado, a seguir, um exemplo de aplicação deste método no controle de temperatura de um aquecedor, conforme mostra a figura 3.10, onde a variável **x** representa a temperatura.

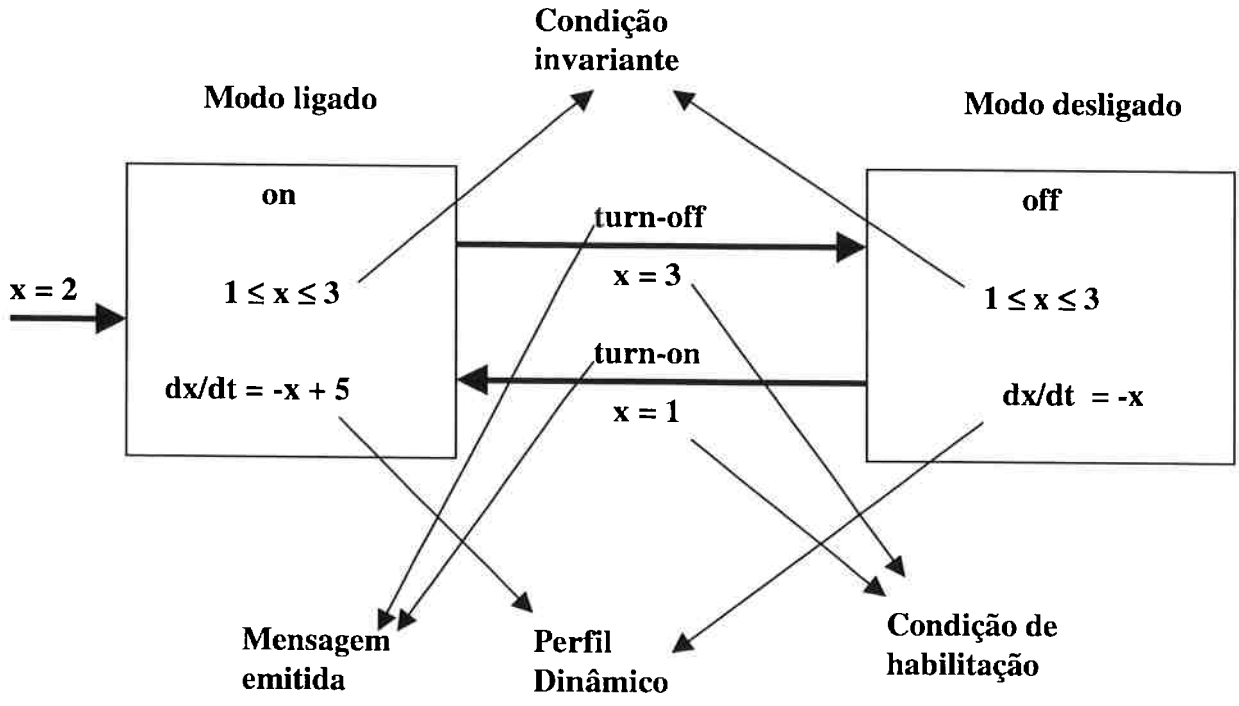


Figura 3.10 – Modelagem por Autômato Híbrido de um Aquecedor

O sistema só pode se manter num determinado modo de operação enquanto a *Condição Invariante* daquele modo for satisfeita. No exemplo, a mudança de modo deve ocorrer antes da temperatura deixar o intervalo operação de entre 1 e 3 graus ($1 \leq x \leq 3$)

Um autômato híbrido **A** é formalizado da seguinte maneira:

$$\mathbf{A = (X,V,flow,init,inv,E,jump,\Sigma,syn)}$$

X : {x1,x2,....,xn}, conjunto finito de variáveis. No exemplo X = {x}

V : Conjunto finito de modos de operação. No exemplo V = {on, off}

flow : condições de atividade contínua. No exemplo

flow(on): dx/dt = -x + 5; flow(off) : dx/dt = -x

A especificação de uma atividade contínua pode envolver não determinismo como, por exemplo, dx/dt $\in$ [2,4].

inv : condições invariantes. No exemplo inv(on) = inv(off) : $1 \leq x \leq 3$

init : condições iniciais, sendo que o autômato **A** pode iniciar no modo **v** quando a condição inicial init(v) for verdadeira. No exemplo init(on): x = 2; init(off): false.

E : chaves de controle ou transições. No exemplo E = {(on,off);(off,on)}

jump : condição de mudança de fase. Trata-se de um predicado, referente a uma transição, que atua sobre as variáveis em $X \cup X'$, onde as variáveis em X' referem-se aos valores atribuídos às variáveis em X após a mudança do modo de operação. No exemplo:

jump((on,off)) : $x = 3 \ \Lambda \ x' = x$

jump((off,on)) : $x = 1 \ \Lambda \ x' = x$

Σ : conjunto de eventos. No exemplo Σ = {turn-on, turn-off}

syn : associa um evento em Σ a cada transição em **E**. No exemplo:

syn((on,off)) = turn-off; syn((off,on)) = turn-on

Quando se modela um sistema através de autômatos híbridos, os Requisitos Gerais de Segurança só são considerados atendidos pelo sistema se e somente se dentre todas as configurações alcançáveis não existirem configurações perigosas. Um Requisito Geral de Segurança é normalmente especificado pela descrição de combinações de valores desejáveis e, em especial, de combinações de valores indesejáveis das variáveis do sistema, conforme já apresentado na aplicação através de métodos com Redes de Petri e Statecharts. No caso específico de um autômato híbrido, o Requisito Geral de Segurança pode ser especificado através de asserções relativas aos diversos modos de operação. Outra forma de se especificar um Requisito Geral de Segurança á através da criação de um predicado *Hazard* sobre as configurações do sistema. Esse predicado descreve as condições que violam a segurança do sistema. Desta forma, o sistema modelado é considerado seguro se a região *Hazard* das configurações do sistema não incluir nenhuma configuração alcançável do sistema em questão. O Requisito Geral de

Segurança do exemplo do aquecedor seria desligar o aquecedor quando atingi-se uma determinada temperatura máxima de perigo.

3.4.10 Avaliação Quantitativa da Segurança de Aplicações Microprocessadas

A avaliação quantitativa é o último passo de um processo de Análise de Perigo, dentro da etapa Análise Final de Perigo. No método aqui apresentado são feitas algumas considerações considerando o sistema microprocessado triplicado TMR – "Triple Modular Redundancy". [Camargo 01] Geralmente nestes sistemas de segurança crítica, são utilizadas técnicas de redundância em combinação com técnicas de blocos "fail-safe", que normalmente se constituem no gargalo destes sistemas como, por exemplo, comparadores e votadores, no caso do sistema TMR. Nos sistemas microprocessados não é recomendável realizar uma análise de risco sem considerar a presença do software, em especial quando a segurança é dependente do software. Neste trabalho a função do bloco microprocessado é dividida em módulos funcionais, de diagnóstico e de reconfiguração. A avaliação das falhas inseguras está relacionada com módulos funcionais. Os módulos de diagnóstico irão influenciar o fator de cobertura de falhas e os módulos de reconfiguração influenciam na modelagem final do sistema.

As taxas de falhas do hardware e do software podem agrupadas visando obter a taxa de falhas global do sistema. Para esta decisão foram adotadas algumas hipóteses [Iyer 85]: Os módulos funcionais estão logicamente em série num determinado módulo do sistema;

- Quando um módulo funcional falha, ele não influencia na falha de um outro bloco funcional. Um subconjunto de erros de software tem grande proximidade com erros no hardware. Um erro pode causar uma falha num bloco único. Tais erros são denominados erros de software relacionados com o hardware. [Houtermans 98] Neste trabalho erros deste tipo são classificados como erros de hardware. São considerados erros de software aqueles que não apresentam conexão com falhas no hardware;
- Erros de projeto do hardware não são considerados pois o projeto do hardware microprocessado já apresenta grande maturidade;
- As taxas de falhas são expressas em relação a um tempo absoluto, e função de uma referência comum de tempo; e
- Deve-se ter o cuidado de não contar duas vezes a mesma falha, que poderia acontecer no caso de módulos funcionais em série.

A taxa de falhas de um canal simples pode ser assumida como a soma da taxa de falhas seguras e da taxa de falhas inseguras. [Houtermans 98]

$$\lambda_{singleboard} = \lambda_{Ssingleboard} + \lambda_{Usingleboard}$$

A falha segura representa uma falha que respeita os Requisitos Gerais de Segurança, caso contrário, ela será considerada falha insegura.

A avaliação da taxa de falhas de um canal microprocessado deve se basear em probabilidades de eventos bem definidos. Considerando a existência de erros de projeto de software, a ocorrência destes erros no canal simples, corresponde a um evento bem definido. [Garrett 99] Neste caso, a taxa de falha do software deve ser igual à taxa de ocorrência das condições associadas àquele perigo. Por outro lado, a taxa de falhas do hardware corresponderá às falhas decorrentes do desgaste do hardware envolvido. Neste método não foram consideradas falhas transientes. As falhas transientes devem ser avaliadas através de métodos qualitativos apresentados nesta tese. Feitas estas considerações pode-se dizer que:

$$\lambda_{singleboard} = \lambda_{HW} + \lambda_{SW}$$
$$\lambda_{singleboard} = (\lambda_{SHW} + \lambda_{UHW}) + (\lambda_{SSW} + \lambda_{USW})$$

onde λ_{HW} corresponde à taxa de falhas do hardware, λ_{SW} à taxa de falha do software, λ_{SHW} à taxa de falha segura do hardware, λ_{UHW} à taxa de falha insegura do hardware, λ_{SSW} à taxa de falha segura do software e λ_{USW} à taxa de falha insegura do software.

Um grande problema reside na determinação da porcentagem da taxa de falha do canal simples $\lambda_{singleboard}$ que corresponde às falhas seguras e inseguras. Esta decisão é altamente dependente da aplicação e, desta forma, é altamente dependente do software utilizado. [Bastt 98] Além deste aspecto, esta taxa de falhas inseguras pode ser dividida em duas categorias: detectável λ_{UD} e não detectável λ_{UND}. [Laprie 90]

A taxa de falhas inseguras do canal simples pode ser então avaliada como:

$$\lambda_{Usingleboard} = \lambda_{UHW} + \lambda_{USW}$$
$$\lambda_{Usingleboard} = (\lambda_{UDHW} + \lambda_{UNDHW}) + (\lambda_{UDSW} + \lambda_{UNDSW})$$

onde λ_{UDHW} corresponde à taxa de falhas inseguras detectáveis do hardware, λ_{UNDHW} corresponde à taxa de falhas inseguras não detectáveis do hardware, λ_{UDSW} corresponde à taxa de falhas inseguras detectáveis do software e λ_{UNDSW} à taxa de falhas inseguras não detectável do software. Pesquisas futuras devem ser realizadas na estimativa de λ_{UNDSW} e λ_{USW} já que existem estudos em como estimar λ_{SW}. [Wright 94] [Schneidewind 97]

Considerando cada módulo funcional de hardware, a taxa de falhas pode ser avaliada através do modelo combinatório série, de acordo com a seguinte equação:

$$\lambda_{MODULE} = \sum \lambda_{COMPONENT}$$

Para os módulos funcionais de hardware com circuitos digitais de baixa complexidade, a taxa de falhas dos componentes pode obtida, por exemplo, de normas internacionais como a do Departamento de Defesa Norte Americano – MIL-HDBK-217. [Military Handbook] Os modos de falhas destes componentes são bem conhecidos e através da técnica FMECA podem ser determinados os modos de falhas perigosos detectáveis e não detectáveis. Assim a avaliação destas taxas de falhas pode ser realizada da seguinte forma:

$$\lambda_{UDCOMP} = \frac{\lambda_{COMPONENT}}{\sum cfm} * \sum udfm$$

$$\lambda_{UNDCOMP} = \frac{\lambda_{COMPONENT}}{\sum cfm} * \sum uufm$$

onde Σcfm corresponde ao número total de modos de falhas do componente, Σudfm ao número de modos de falhas perigosos detectáveis, Σuufm ao número de modos de falhas perigosos não detectáveis, todos determinados através do FMECA. O valore λ_{UDCOMP} corresponde à taxa de falhas insegura detectável do componente e $\lambda_{UNDCOMP}$ à taxa de falha insegura não detectável do componente.

Entretanto, é extremamente complexo analisar o modo de falhas de componentes de alta escala de integração (LSI/VLSI) e seu efeito num sistema crítico. Não é possível, nestes casos, realizar um FMECA dos blocos funcionais nem determinar as taxas de falhas inseguras. Nestes casos, são propostas três faixas de falhas inseguras:

- $\lambda_U = 0.1\ \% \ . \ \lambda_{MODULE}$ (menos pessimista)
- $\lambda_U = 1.0\ \% \ . \ \lambda_{MODULE}$
- $\lambda_U = 10.0\ \% \ . \ \lambda_{MODULE}$ (mais pessimista)

Outro aspecto importante com relação aos módulos funcionais de hardware com componentes LSI/VLSI é a determinação de falhas inseguras não detectáveis. Este tipo de falha num canal simples permanece no sistema e, no caso de utilização de técnicas redundantes, uma falha compensatória em outro canal pode conduzir o sistema a uma condição insegura. Os valores da taxa de falhas inseguras detectáveis e não detectáveis do sistema irão influenciar o valor final do MTTUF – "Mean Time to Unsafe Failure" do sistema de acordo com a seguinte expressão:

$$MTTUF = \int_0^\infty S(t).dt$$

onde S(t) corresponde à probabilidade do sistema permanecer num estado seguro.
É apresentado a seguir um método de avaliação da taxa de falha insegura do canal simples.

$$\lambda_{UDHW} = (A_{SW} * C_{SW} + A_{HW} * C_{HW}) * \lambda_{UHW}$$

onde:

- A_{SW} - Disponibilidade dos recursos de hardware necessários para que o software, que detecta falhas no hardware, seja executado;
- C_{SW} - Diagnóstico implementado por software, que determina o fator de cobertura do software; [Leveson 90]
- A_{HW} - Disponibilidade de outros recursos de hardware, necessários para detectar falhas, quando o software não está sendo executado;
- C_{HW} - Diagnóstico implementado através destes outros recursos de hardware, que determina o fator de cobertura do hardware.

3.4.11 Injeção de Falhas

O conceito essencial fundamentando a aplicação do método de Injeção de Falhas num sistema computacional está no processo computacional, que realiza transições sobre diversas entradas, passando através de estados intermediários, e gerando saídas. A Injeção de Falhas corresponde à ação de adicionar uma nova transição ou retirar uma existente ao processo e observar seus efeitos na saída do sistema. Neste aspecto é

fundamental realizar esta atividade num ambiente real de operação. Através deste método é possível verificar o efeito da falha em todo sistema e não apenas localmente. [Voas 98]

A Injeção de Falhas pode ser aplicável às entradas, aos diversos estados do sistema, às suas informações, com o propósito de verificar o que ocorre com suas respectivas saídas. Desta forma, podem ser inseridas falhas de hardware ou software, cuja simulação do comportamento pode ser inserido através de técnicas de software, evitando assim ações destrutivas no teste do sistema. Como o método de Injeção de Injeção de Falhas é aqui utilizado dentro do processo de Análise de Perigo, seu enfoque deve ser o de avaliar o efeito na saída, verificando o atendimento ou não aos Requisitos Gerais de Segurança. Este método pode inclusive ser utilizado para auxiliar na pesquisa dos valores das taxas de falhas inseguras relacionados com os sistemas computacionais e que já foi discutido um método apresentado no item anterior. Desta forma, a Injeção de Falhas pode auxiliar em fornecer dados mais confiáveis a serem utilizados num processo de Avaliação Quantitativa da segurança de sistemas microprocessados.

4 ESTUDO DE CASO

Neste capítulo são apresentados dois estudos de caso em que são aplicadas algumas das filosofias de Análise de Risco, discutidas no capítulo anterior.

O primeiro trabalho refere-se às atividades de Análise de Risco de Sistemas de Sinalização e Controle Metroviário, tendo como referências projetos de pesquisa na área além de trabalhos de Análise de Perigo já realizados em diversos sistemas metroviários, entre eles a Companhia do Metropolitano de São Paulo – METRÔ.

O segundo trabalho envolve o planejamento da Análise da Segurança do Sistema CNS/ATM – "Communication, Navigation and Surveillance/ Air Traffic Managment", que se constitui no futuro sistema de navegação aéreo a nível internacional, ou seja, FANS – "Future Air Navigation System". [Galotti 99] Este segundo trabalho tem como referência o intercâmbio em pesquisa com o Instituto de Proteção ao Vôo – IPV, pertencente à Diretoria de Eletrônica e Proteção ao Vôo do Ministério da Aeronáutica.

4.1 Sistema de Sinalização e Controle Metroviário

Neste estudo de caso é apresentada uma breve descrição de um sistema de sinalização e controle metroviário seguido da aplicação da Metodologia de Análise de Risco proposta no capítulo anterior. Nesta aplicação são destacados alguns importantes resultados já alcançados ao longo desta pesquisa.

4.1.1 Descrição Funcional do Sistema de Sinalização e Controle Metroviário

De forma a permitir o controle de movimentação das composições num sistema metroviário é necessário um sistema de sinalização e controle que tenha condições de realizar suas funções básicas atendendo os Requisitos Gerais de Segurança, Disponibilidade e Confiabilidade exigidos.

Um Sistema de Sinalização e Controle Metroviário típico constitui-se num conjunto de equipamentos utilizados com a finalidade básica do controle de tráfego de trens. Sua arquitetura básica é apresentada na figura 4.1. [Accurso 99]

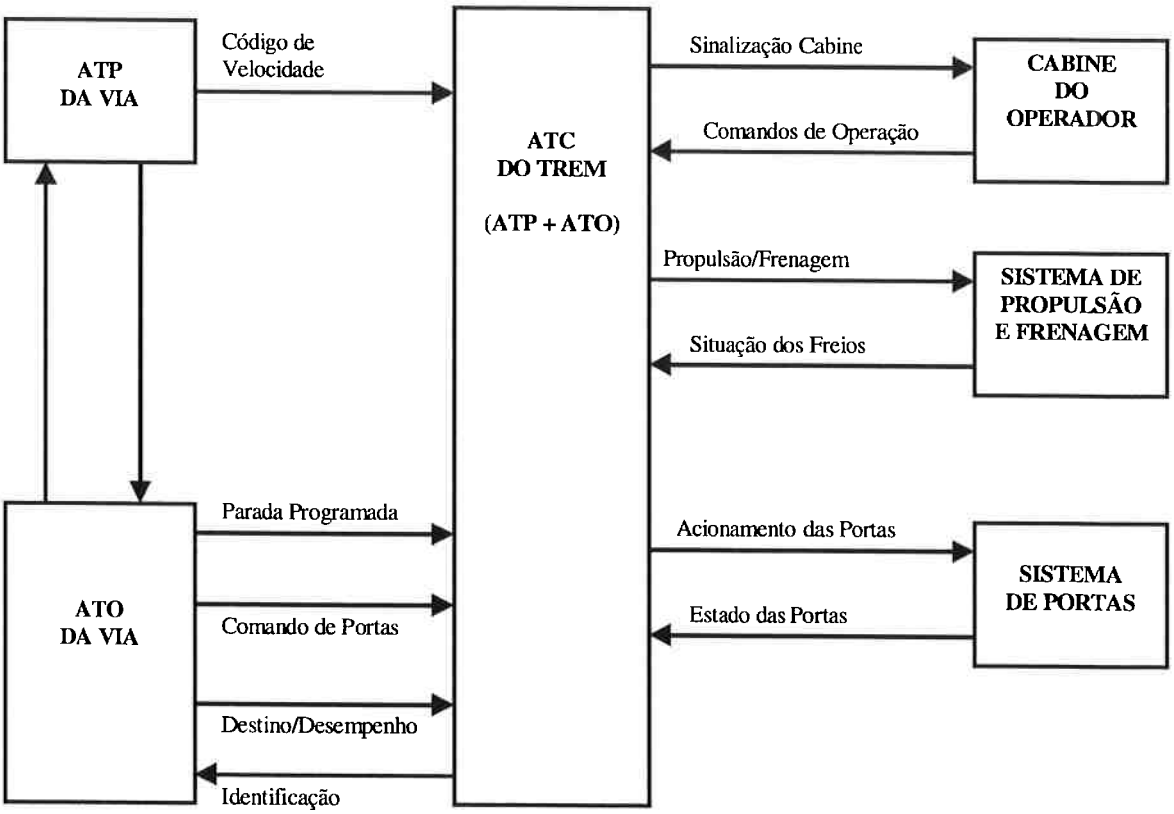


Figura 4.1 – Arquitetura Básica de um Sistema de Sinalização Metroviário

O subsistema ATO – "Automatica Train Operation" é responsável pela operação automática da movimentação dos trens, como sua parada programada e partida das estações, comando de abertura e fechamento de portas e controle de velocidade e identificação dos trens. Este subsistema é composto por equipamentos instalados nas estações e a bordo dos trens.

O subsistema ATP – "Automatic Train Protection" é responsável pela proteção da movimentação dos trens e é composto por equipamentos instalados nas estações mestras, à margem da via e à bordo dos trens.

O sistema ATC – "Automatic Train Control" a bordo dos trens se comunica também com a Cabine do Operador, com o Sistema de Propulsão e Frenagem e com o Sistema de Portas.

Atualmente muitos destes subsistema já estão implantados com tecnologia microprocessada.

4.1.2 Alguns Resultados da Aplicação da Metodologia de Análise de Risco

Os resultados apresentados neste item referem-se à aplicação da metodologia e de diversos métodos de análise, descritos no capítulo anterior, no escopo de um sistema de sinalização e controle metroviário. Alguns dos métodos têm sido aprimorados em função de novas necessidades decorrentes de novas tecnologias e novos requisitos.

Com relação aos aspectos gerenciais, comentados no item 3.1, pode-se afirmar que o GAS constitui-se num grupo independente que realiza atividades de pesquisa em análise de risco de sistema críticos, em especial sistemas de sinalização metroviários. As atividades neste grupo concentram-se fundamentalmente na avaliação do nível de segurança presente nos sistemas de sinalização e controle metroviários.

O trabalho apresentado refere-se a uma atividade de **Análise de Perigo**, pois é avaliada a probabilidade da ocorrência de um estado perigoso. Assim sendo, não são avaliadas as probabilidades de acidentes nem a severidade das conseqüências envolvidas.

4.1.2.1 Análise Preliminar de Perigo

O primeiro trabalho desenvolvido foi o estabelecimento de um conjunto de Requisitos Gerais de Segurança - RGS do Sistema de Sinalização e Controle. Para este objetivo foram definidos os **Conceitos Gerais de Segurança**, ou seja, foram determinados quais perigos devem ser evitados durante a operação do sistema.

Para um melhor entendimento dos resultados apresentados a seguir, foi desenvolvido um Glossário Técnico, apresentado no Apêndice A desta tese.

Do ponto de vista de segurança de um sistema metroviário, há três conceitos gerais de segurança:

- evitar colisões;
- evitar descarrilamentos; e
- evitar atropelamentos (garantir a segurança para a equipe de manutenção).

Neste sentido, o sistema deve executar, de forma segura, suas funções de controle de movimentação dos trens, através das funções de:

- Proteção dos Aparelhos de Mudança de Via - AMV;
- Controle de Tráfego; e

- Seleção dos Códigos de Velocidade.

Uma das técnicas utilizadas para a determinação dos Requisitos Gerais de Segurança é o HAZOP. É apresentado, a seguir, um exemplo ilustrativo de sua aplicação em um Sistema de Sinalização e Controle Metroviário.

Em seguida são apresentados os Requisitos Gerais de Segurança que foram desenvolvidos tendo em mente um sistema ATP de via, já sendo um mapeamento do Sistema de Sinalização num de seus subsistemas. Pode-se afirmar, desta forma, que esta etapa já se constitui também numa Análise de Perigo do Sistema.

a) Aplicação do HAZOP a um Sistema Metroviário

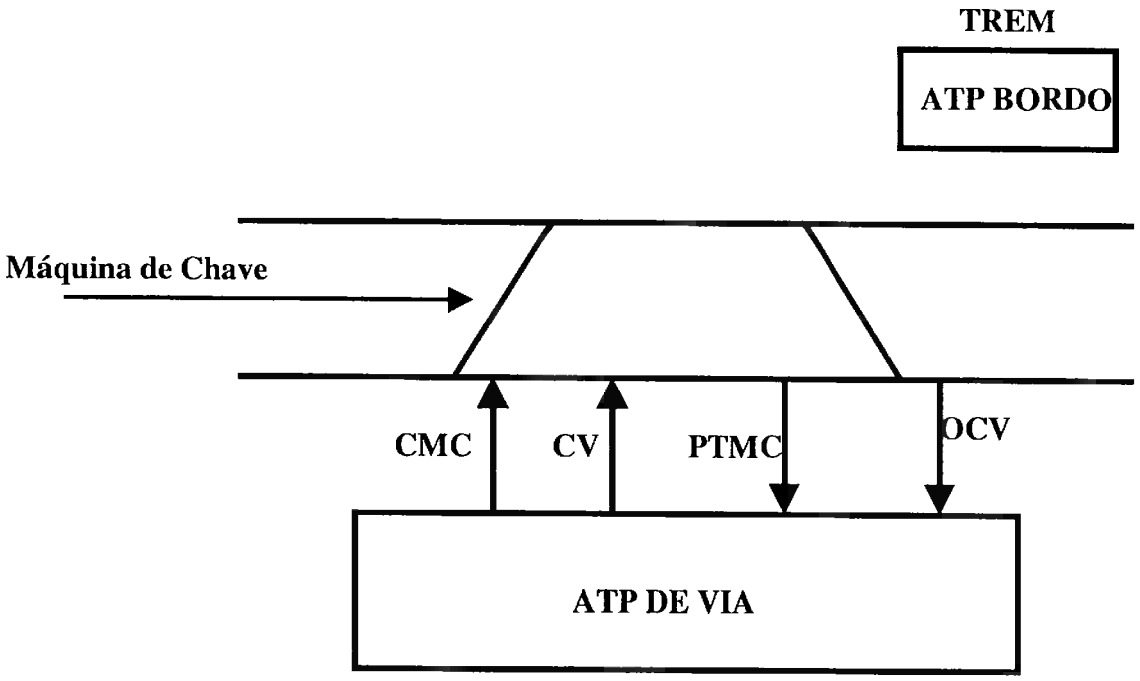
O sistema considerado neste exemplo refere-se a uma planta simplificada de um sistema de sinalização, conforme apresentado na figura 4.2 [Camargo 99]

O ATP de Via envia à Via os comandos de máquina de chave e os comandos de velocidade.

O ATP de Via recebe da Via as indicações de ocupação de circuito de via e posicionamento e travamento da máquina de chave.

Neste exemplo ilustrativo são considerados dois estados perigosos: Colisão e Descarrilamento. Cada *Guide Word* é aplicada a este modelo simplificado, com o objetivo de obter algumas informações que possam auxiliar no projeto de sistemas de sinalização, como por exemplo, através da determinação de Requisitos Gerais de Segurança. [Hebbron 97]

Para cada *Guide Word* é realizada sua aplicação aos sinais envolvidos e avaliada a sua relação com algum estado perigoso.



CMC – Comando de Máquina de Chave
CV – Comando de Código de Velocidade
PTMC – Posicionamento e Travamento da Máquina de Chave
OCV – Ocupação de Circuito de Via

Figura 4.2. – Planta Simplificada de um Sistema de Sinalização Metroviário

Guide Word **NO**

- Comando de Máquina de Chave não foi enviado. A Máquina de Chave não será posicionada corretamente. Este evento será detectado pelo sinal de Posicionamento e Travamento de Máquina de Chave. Não há perigo.
- Comando de Velocidade não foi enviado. O ATP de Bordo decodifica este evento como velocidade zero. Não há perigo.
- O Posicionamento e Travamento da Máquina de Chave não foi recebido. O ATP de Via deve detectar tal situação, caso contrário um estado perigoso pode existir como um descarrilamento e/ou colisão.
- A Ocupação do Circuito de Via não foi recebida. O ATP de Via deve detectar tal situação, caso contrário, um estado perigoso pode existir como uma colisão.

Guide Word **MORE/LESS**

- O Comando de Máquina de Chave é enviado a uma taxa maior/menor do que a especificada. Este comando pode não ser entendido. Este evento será detectado pelo sinal de Posicionamento e Travamento de Máquina de Chave. Não há perigo.
- O sinal Código de Velocidade é enviado a uma taxa maior/menor do que a especificada. O ATP de Bordo pode decodificar tal sinal com um código de velocidade diferente. Os sistemas ATP de Bordo e de Via devem detectar tal situação, caso contrário, um estado perigoso pode existir podendo conduzir a uma colisão.
- O sinal de Ocupação de Circuito de Via é recebido a uma taxa maior/menor do que a especificada. O sistema ATP de Via deve detectar tal evento, caso contrário, pode existir uma falsa desocupação, podendo conduzir a uma colisão.
- O sinal de Posicionamento e Travamento de Máquina de Chave é recebido a uma taxa maior/menor do que a especificada. O sistema ATP de Via deve detectar tal situação, caso contrário uma posição incorreta de uma máquina de chave pode ser entendida, podendo causar um descarrilamento e colisão.

Guide Word **AS WELL AS**

- Todo o funcionamento especificado foi realizado mas efeitos adicionais foram obtidos. Estes efeitos não devem interferir no correto funcionamento dos sistema ATP de Via e de Bordo.

Guide Word **PART OF**

- Parte do sinal de Comanda de Máquina de Chave foi enviado. Este evento será detectado pelo sinal de Posicionamento e Travamento de Máquina de Chave. Não há perigo.
- Parte do sinal de Comando de Velocidade foi enviado. O sistema ATP de Bordo pode decodificar este sinal como um código de velocidade diferente. O sistema ATP de Via e de Bordo devem detectar tal situação, caso contrário um estado perigoso pode ser alcançável com uma colisão.

- Parte do sinal Ocupação de Circuito de Via foi recebido. O sistema ATP de Via deve detectar tal situação, caso contrário pode existir uma falsa desocupação, conduzindo a uma estado perigoso como uma colisão.

- Parte do sinal de Posicionamento e Travamento de Máquina de Chave foi recebido. O sistema ATP de Via deve detectar tal situação, caso contrário uma posição incorreta da máquina de chave pode ser entendida, causando a existência de um estado perigoso, como o descarrilamento.

Guide Word **REVERSE**

- Não aplicado aos sinais em questão

Guide Word **OTHER THAN**

- O sinal de Comando de Máquina de Chave é enviado incorretamente. O sistema ATP de Via deve detectar tal evento através do sinal de Posicionamento e Travamento de Máquina de Chave. Não há perigo.

- O sinal de Código de Velocidade é enviado incorretamente. Os sistema ATP de Via e de Bordo devem detectar tal situação, caso contrário, pode ser interpretado um código de velocidade incorreto, conduzindo a um estado perigoso de colisão.

- O sinal de Ocupação de Circuito de Via é recebido incorretamente. O sistema ATP de Via deve detectar tal situação, caso contrário pode existir um estado perigoso como uma falsa desocupação, podendo levar a uma colisão.

- O sinal de Posicionamento e Travamento de Máquina de Chave foi recebido incorretamente. O sistema ATP de Via deve detectar tal evento, caso contrário uma posição incorreta de máquina de chave pode ser entendida, causando a existência de um estado perigoso e podendo levar a um descarrilamento e colisão.

Guide Word **EARLY/BEFORE**

- O sinal Comando de Máquina de Chave é enviado muito antecipadamente ou antes da seqüência esperada. O sistema ATP de via deve detectar tal situação, caso contrário pode ocorrer um descarrilamento.

- O sinal de Comando de Velocidade é enviado muito antecipadamente ou antes da seqüência esperada. O sistema ATP de Via deve detectar tal evento, caso contrário

pode ocorrer um descarrilamento e/ou colisão. Um caso típico refere-se à partida prematura do trem da estação.

- O sinal de Ocupação do Circuito de Via é recebido muito antecipadamente ou antes da seqüência esperada. O sistema ATP de Via deve detectar tal evento, caso contrário pode existir uma falsa desocupação, conduzindo a uma colisão.
- O sinal de Posicionamento e Travamento de Máquina de Chave é recebido muito antecipadamente ou antes da seqüência esperada. O sistema ATP de Via deve detectar tal situação, caso contrário uma posição incorreta da máquina de chave pode ser alcançada, causando um descarrilamento e/ou colisão.

Guide Word **LATE/AFTER**

- O sinal de Comando de Máquina de Chave é enviado muito tardiamente ou após uma seqüência esperada. A máquina de chave não estará posicionada corretamente. Este evento será detectado pelo sinal de Posicionamento e Travamento de Máquina de Chave. Não há perigo.
- O sinal de Código de Velocidade é enviado muito tardiamente ou após uma seqüência esperada. O sistema ATP de Via deve detectar e evitar tal evento, caso contrário uma colisão pode correr, em especial quando do decréscimo de velocidade.
- O sinal de Ocupação de Circuito de Via é recebido muito tardiamente ou após uma seqüência esperada. O sistema ATP de Via deve detectar e evitar tal evento, caso contrário uma falsa desocupação pode existir e conduzir o sistema a uma colisão.
- O sinal de Posicionamento e Travamento de Máquina de Chave é recebido muito tardiamente ou após uma seqüência esperada. O sistema ATP de Via deve considerar o valor do sinal de Posicionamento e Travamento de Máquina de Chave e tomar as decisões de segurança.

b) Requisitos Gerais de Segurança do Subsistema ATP de Via

RGS1: Só poderá haver alinhamento de uma rota em condições normais do sistema se não houver tráfego estabelecido no sentido oposto ao bloqueio de saída desta rota.

RGS2: Só poderá haver alinhamento de rota em condições normais do sistema se todos os circuitos de via pertencentes à rota (no interior de uma região de AMVs e adjacentes) estiverem desocupados.

RGS3: Só poderá haver alinhamento de rota em condições normais do sistema se o número dos circuitos de via desocupados, imediatamente posteriores ao bloqueio de saída da rota, forem suficientes para permitir a parada do trem antes do próximo bloqueio.

RGS4: Só poderá haver alinhamento de rota se não houver outra rota conflitante com a primeira.

RGS5: Só poderá haver alinhamento de rota de chamada se não existir qualquer outra rota de chamada em sentido oposto.

RGS6: Só poderá haver abertura de um bloqueio como entrada de uma rota de chamada após um intervalo de tempo especificado em 60 segundos. Este tempo deve ser contado a partir da imposição de código de velocidade de 0 km/h aos circuitos de via da rota. A finalidade deste tempo é permitir ao trem, que eventualmente esteja trafegando em sentido oposto, parar antes da abertura do bloqueio para a rota de chamada.

RGS7: Um bloqueio só pode ser aberto se todas as máquinas de chave envolvidas estiverem eletricamente travadas em posições que definam uma rota prevista no intertravamento.

RGS8: Se houver proibição de um bloqueio como saída de uma rota por qualquer uma das salas operantes, não poderá haver alinhamento de rota que utilize este bloqueio como saída.

RGS9: Se houver proibição de um bloqueio como entrada de uma rota por qualquer uma das salas operantes, não poderá haver alinhamento de rota que utilize este bloqueio como entrada.

RGS10: Um bloqueio deve permanecer fechado enquanto houver pelo menos uma requisição de proibição de abertura por qualquer uma das salas operantes.

RGS11: Quando um bloqueio é fechado ou colocado no modo automático, a rota que tiver sido alinhada pela abertura deste bloqueio deve entrar em processo de cancelamento por tempo.

RGS12: Um sinaleiro referente ao bloqueio de entrada de uma rota só pode apresentar o aspecto amarelo se esta rota estiver alinhada e liberada em condições normais de operação do sistema.

RGS13: Um sinaleiro referente ao bloqueio de entrada de uma rota de chamada só pode apresentar aspecto vermelho piscante se esta rota estiver alinhada e requisitada como tal.

RGS14: Só poderá haver cancelamento de uma rota alinhada, ou por desocupação seqüencial (cancelamento automático), ou por cancelamento requisitado pelo operador.

RGS15: O cancelamento de uma rota pelo operador não deverá ser efetivado se algum dos circuitos de via pertencentes à rota alinhada já tiverem sido ocupados por um trem.

RGS16: O cancelamento de uma rota pelo operador só deve ser efetivado após uma temporização suficiente para parar o trem antes que ele ultrapasse o bloqueio de entrada. Esta temporização está especificada como sendo de 60 segundos. Se o trem não estiver a uma distância do bloqueio que permita a sua parada, o trem irá ocupar os circuitos de via pertencentes à rota e o RGS15 deverá ser garantido.

RGS17: No cancelamento de uma rota por desocupação seqüencial, cada circuito de via só deverá deixar de fazer parte da rota após ter sido desocupado pelo trem.

RGS18: Só poderá haver destravamento de uma máquina de chave se esta não pertencer a nenhuma rota e o circuito de via da máquina de chave estiver desocupado.

RGS19: O perfil seguro de velocidade que se segue a uma ocupação deve ser respeitado pelo intertravamento.

RGS20: O perfil seguro de velocidade que antecede a um bloqueio fechado deve ser respeitado pelo intertravamento.

RGS21: No alinhamento de rota, a seleção de código de velocidade deverá obedecer ao sentido de tráfego estabelecido e ao perfil de velocidade imposto pelas condições da via.

RGS22: O trecho correspondente ao fim de via deverá estar com código de velocidade de 0 km/h e com sinalização apropriada.

RGS23: Só poderá haver geração de código de velocidade superior a 0 km/h para uma rota já alinhada em condições normais do sistema.

RGS24: Quando o ATP de via receber um comando de restrição de código de velocidade por qualquer uma das salas operantes, ele deverá impor um limite máximo de velocidade especificado em 44 km/h, em todos os circuitos de via envolvidos que estejam com um valor de código de velocidade acima deste limite.

RGS25: Na ocorrência de violação de bloqueio, deverá ser imposto código de velocidade de 0 km/h nos circuitos de via pertencentes a região de AMVs invadida, com o fechamento imediato de todos os bloqueios abertos desta região.

RGS26: Só poderá haver efetivação da inversão de sentido de tráfego se o bloqueio para o qual algum trem se dirigirá não estiver sendo utilizado como saída de uma outra rota.

RGS27: A realização de manutenções em uma região de AMVs só pode ocorrer se não houver nenhuma rota alinhada na região. Rotas que estejam alinhadas devem entrar em processo de cancelamento. Os códigos de velocidade nos trechos da região em manutenção devem ser 0 km/h e devem ser gerados perfis de frenagem nos trechos que antecedem os bloqueios de acesso à região.

RGS28: Uma vez que uma região de AMVs esteja em manutenção, nenhuma rota pode ser alinhada na respectiva região.

RGS29: A permanência de pessoas e a movimentação de veículos dentro de uma região de AMVs em manutenção devem estar regulamentados por procedimentos operacionais.

RGS30: Quando o trem estiver em operação manual, a distância segura entre dois trens deve ser garantida por procedimentos operacionais.

RGS31: Os módulos redundantes devem ser independentes entre si no que se refere a parte envolvida com a segurança.

RGS32: A votação dos módulos redundantes deve ser feita através de circuitos implementados com técnicas "fail-safe".

RGS34: A comunicação dos dados vitais entre os diversos módulos do sistema de sinalização deve ser feita de realizada segura.

RGS35: A troca de sinais vitais entre o Sistema de Sinalização e os dispositivos de via deve ser feita de forma segura.

RGS36: Uma vez requisitada, pelo operador, uma restrição de circulação na via, o sistema deve garantir a sua efetivação.

RGS37: O sistema deve garantir a manutenção de restrições de circulação impostas a via, até a emissão de requisições, por parte do operador, que as removam.

4.1.2.2 Análise de Perigo dos Subsistemas

De posse dos Requisitos Gerais de Segurança passou-se à etapa de Análise de Perigo dos Subsistemas Envolvidos. Nesta etapa as sub-atividades já definidas no capítulo 3 foram mais detalhadas, sendo descritas a seguir.

Neste item são apresentados alguns resultados desta etapa.

a) Análise de Perigo do Hardware - Modos de Falhas considerados no FMECA

Alguns modos de falhas de componentes do hardware considerados nesta análise são:

- Circuitos Integrados: entradas e saídas presas em nível "1" ou "0".
- Resistores: Aberto/Curto/Aumento e Diminuição da Resistência
- Capacitores: Aberto/Curto/Fuga
- Diodos: Aberto/Curto/Fuga
- Transistores: Circuito Aberto Coletro-Emissor, Base-Emissor/ Curto Coletor-Emissor, Base-Emissor/Alteração no Fator de Amplificação β.
- Transformadores: Primário Aberto/Secundário Aberto/Curto entre Espiras no Primário ou Secundário/Curto do Primário ou Secundário para carcaça.

Com relação aos módulos microprocessados, é praticamente impossível se realizar um FMECA dos seus componentes integrantes, pois torna-se extremamente complexo definir os efeitos no sistema dos resultados dos inúmeros modos de falhas possíveis. Nestes casos foi adotado o método de avaliação quantitativa apresentado no item 3.4.10 desta tese. Como este método depende fortemente do tipo de software implementado,

sua aplicação será apresentada no item 4.1.2.3, referente à etapa de Análise Final de Perigo – Avaliação Quantitativa.

b) Análise de Perigo do Software – Inspeção Formal

Com no caso específico do METRÔ, as linguagens em questão são "C" e "Assembly", a Lista de Verificação desenvolvida leva em consideração características das respectivas linguagens e de seus compiladores utilizados. É apresentado, a seguir, um resumo desta Lista de Verificação com exemplos ilustrativos em alguns casos. [Rady 00] É utilizada nesta lista princípios de programação defensiva. [Yu 98]

Cada tópico apresentado é composto pela sua explicação e por exemplos, quando assim se fizer necessário.

b.1) Procedimento de Retorno de Rotinas: consiste em verificar se as rotinas do software têm um procedimento normal de retorno, ao final do qual contêm o código necessário para que, quando ocorrer o retorno à rotina acionadora, o processamento possa prosseguir sem nenhuma perda, ou seja, a continuidade seja assegurada.

Se existirem rotinas com procedimento de retorno incorreto, as demais rotinas existentes no software, inclusive aquelas diretamente ligadas com a segurança do sistema podem ter seu processamento afetado, realizando ações de maneira incorreta.

A verificação da correta terminação de rotinas é realizada por intermédio da inspeção dos seguintes itens:

- Se as rotinas do tipo função retornam um valor em qualquer dos caminhos que conduzam à sua finalização;

Exemplo:

```
Int a (int b)
{
 if (b == 0) {
    d = 1;
    return 0;
  } else {
       if (b == 1)
          d = 3;
       else
          return 1;
      }
}
```

No exemplo, no caso em que b = 1, o valor retornado pela função é indefinido.

- Caso uma rotina em assembly receba parâmetros por intermédio da pilha deve, necessariamente, limpar a pilha ao final da execução, ou seja, retirar todos os parâmetros passados através da pilha. Tal procedimento é indispensável para o retorno à rotina acionadora;

- Se as rotinas acionadas por interrupção de hardware fornecem o comando de reconhecimento de interrupção aos respectivos controladores de interrupção (EOI – *End Of Interrupt*). No caso da rotina acionada não fornecer este comando, o controlador de interrupções poderá manter a interrupção como pendente e deixar de reconhecer interrupções de menor prioridade;

- Se, na linguagem assembly, a rotina de interrupção se encerra com a instrução IRET.

b.2) Rotinas de Tratamento de Interrupção: consiste em verificar se as rotinas do software de tratamento de interrupção respeitam algumas regras primordiais no que se refere ao controle das interrupções propriamente dito.

Se houver o desrespeito a tais regras, pode haver utilização incorreta variáveis ou parâmetros, possivelmente resultando em processamento errôneo.

A verificação deste tópico é realizada por intermédio da inspeção dos seguintes itens:

- Se, nas linguagens PL/M ou C, a rotina de tratamento de interrupção foi declarada com o atributo *Interrupt*, o que garante que o compilador gerará o código apropriado para o tratamento de interrupções (salvamento do estado do processador, tratamento da pilha e instrução de retorno);

- Se, na linguagem assembly, a rotina de tratamento de interrupção salva os registradores por ela modificados, restaurando-os ao final. Caso estes registradores não sejam resguardados, a rotina acionadora vai empregar valores indefinidos em seu processamento. Neste caso o próprio programador deve se encarregar de implementar todos o tratamento para as rotinas de interrupção;

- O tempo máximo de execução do serviço de interrupção deve ser limitado, pois durante a execução de tal serviço as demais interruções podem não ser atendidas, incluindo-se aí o próprio serviço desta interrupção. Por exemplo, uma conseqüência

de uma possível demora é a não atualização de variáveis vitais à segurança do sistema.

b.3) Controle de Laços Repetitivos: consiste em verificar se os laços existentes nas rotinas do software terminam, ou seja, conseguem atingir a instrução de saída do respectivo laço, a menos dos laços que, intencionalmente, nunca terminem, colocando o software em um ciclo repetitivo. Também consiste em verificar se o laço deixa de ser executado por erro do controle do laço.

No caso de existir um laço, que por erro de programação nunca termine, pode provocar o não acionamento das demais rotinas existentes no software, inclusive aquelas diretamente ligadas com a segurança do sistema.

No caso da não execução do laço, os comandos em seu interior não são executados, inclusive instruções relativas à segurança.

A verificação deste tópico é realizada por intermédio da inspeção dos seguintes itens:

- Se as variáveis de controle de laços são utilizadas e atualizadas de maneira correta. É claro que o uso ou a modificação de valores realizados de maneira inadequada, pode resultar em alterações que levem à não terminação ou não execução de um determinado laço;

- Se o tipo das variáveis utilizadas no laços de controle é coerente com a forma de sua utilização.

Exemplos:

```
void a (int b)
{
 char i
 static c[256]
 for (i = b; i > 0; i--)
    c[i] = 0;
}
```

O laço nunca é executado quando o valor passado como parâmetro é maior do que 127, pois o valor em byte com sinal passa a ser negativo, condição que encerra o laço.

```
int a (int b)
{
 char i
```

```
for (i = 0; i < b; i++)
    c[i] = 0;
}
```

O laço nunca termina ao ter recebido um valor maior do que 127 como parâmetro, já que o contador jamais atingirá o limite final de contagem.

b.4) Testes de Entrada e Saída: consiste em verificar se há testes de entrada e saída das rotinas, especialmente aquelas cuja reentrância deva ser evitada.

Na eventualidade de ocorrerem desvios indevidos a partir do meio de uma rotina, ou para o meio de uma rotina, a existência dos testes de entrada e saída possibilita a detecção deste tipo de falha, impedindo a realização de ações não desejadas.

A verificação deste tópico é realizada por intermédio da inspeção dos seguintes itens:

- Se há o teste de entrada da rotina, ou seja, se há variáveis de controle que possibilitem que, ao ser iniciada uma rotina, esteja claro que a mesma chegou ao seu fim em uma execução anterior;
- Se há o teste de saída da rotina, ou seja, se há variáveis de controle que possibilitem que, ao se terminar uma rotina, esteja claro que a mesma foi executada a partir de seu início.

b.5) Controle de Fluxo de Programas: consiste em verificar se o controle do fluxo das rotinas está sendo executado corretamente.

A má utilização de estruturas de controle pode levar à execução de processamentos não previstos, tendo a possibilidade de causar situações que coloquem em risco a segurança do sistema.

A verificação do controle de fluxo das rotinas é realizada por intermédio da inspeção dos seguintes itens:

- Verificação das estruturas de controle por intermédio de blocos seletores:

  Nos códigos em linguagem PL/M, se antes de cada bloco *DO-CASE* existe um teste para verificar se o parâmetro utilizado está dentro do intervalo estipulado para o bloco.

  Este teste é utilizado para que o bloco *DO-CASE* não seja executado se o parâmetro estiver fora do intervalo estipulado, pois no caso do parâmetro ser maior do que o

número de sentenças do bloco o efeito do *DO-CASE* é indeterminado, dependendo do compilador utilizado (McCrachen, 1978).

Nos códigos em linguagem C, se na utilização do comando *switch* existe a opção *default*, possibilitando a detecção de eventuais situações de falha, caso nenhuma das opções (*case*) se refira ao valor em teste.

Também nos códigos em linguagem C, é recomendável que em todas as opções de valores (*case*) presentes no caso do comando *switch*, haja a terminação com a palavra chave *break* ou equivalente. Em sistemas críticos, todas as situações devem ser previstas, ou seja, o sistema deve ser determinístico.

Exemplo:

```
Switch (a)
{
 case 0:
    b = 1;
    break;
 case 1:
    b = 4;
    break;
 case 2:
    b = 6;
 case 3:
    b = 7;
    break;
}
```

O caso quando a = 2 está invalidado pela ausência da palavra chave *break*, o que vai ocasionar o processamento associado ao caso a = 3. Além disso, outros possíveis valores da variável de seleção – que podem indicar situações de falha – são ignorados.

- Se as rotinas aplicam os comandos de habilitação de interrupções, correspondentes aos comandos de desabilitação aplicados. Este procedimento é particularmente importante nos seguinte casos:
  - Necessidade de execução de regiões críticas do programa em termos de tempo de latência;
  - Acesso a variáveis alteradas por serviços de interrupção

Um comando de desabilitação de interrupções inibe o atendimento às interrupções, e enquanto não for executado um comando de habilitação, nenhuma outra interrupção poderá ser atendida.

Caso alguma rotina não apresente um comando de habilitação correspondente a um comando de desabilitação anteriormente executado, rotinas de tratamento de interrupção poderão deixar de ser acionadas. Enquanto o comando de habilitação não é executado, nenhuma interrupção pode ser atendida. Por este motivo, essa região desabilitada não pode ser muito extensa.

Exemplo:

```
Interrupt void insere (void)
{
 char novo
 novo = inp(0xf3);
 fila[pos] = novo;
 pos++;
}

char remove (void)
{
 char ret
 if (pos == 0) return -1;
 ret = fila[pos];
 pos++;
 return ret;
}
```

No exemplo, a rotina insere( ), acionada por interrupção, acrescenta caracteres lidos do hardware em uma fila, que são removidos pela rotina remove( ). Como remove ( ) não desabilita as interrupções, a ocorrência de insere ( ) durante a execução de remove ( ) pode produzir resultados inconsistentes.

- Há duas formas de tratamento das habilitações e desabilitações de interrupções na linguagem assembly:

  → Comando de desabilitação global, seguido pelo código da rotina, e pelo correspondente comando de habilitação global das interrupções;

  → Salvamento de *flags*, seguido pelo comando de desabilitação de interrupções, seguido pelo código da rotina e pela recuperação dos *flags*.

O segundo caso é preferível ao primeiro, pois nele restaura-se o estado anterior das interrupções, ou seja, as interrupções só serão habilitadas se já estavam habilitadas antes do comando de desabilitação.

As ocorrências dessas estruturas no código devem ser analisadas quanto à sua correção, em cada caso.

- Se, no caso da linguagem Assembly, há os comandos de desempilhamento correspondentes aos comandos de empilhamento aplicados.

  Caso não haja um comando de desempilhamento referente a um comando de empilhamento, o ponteiro do topo da pilha é alterado em relação a seu valor esperado.

- Se, no caso da linguagem Assembly, todos os operandos estão referenciados corretamente no que se refere ao seu registrador de segmento.

  As referências aos operandos que não estiverem no segmento de dados devem ser sempre precedidas pela identificação do registrador de segmento em que se encontrarem os operandos.

  A não observância desta regra pode levar o processador a atualizar áreas de memória não previstas e, desta forma, alterar ou utilizar outras variáveis, eventualmente vitais.

b.6) Código Fonte não Utilizado: consiste em verificar se há trechos de código fonte que tenham servido ao propósito de desenvolvimento do programa, tais como depuração ou versões anteriores, e que não estejam mais sendo utilizados, estando presentes no código fonte, mesmo na forma de comentários.

A presença em si desse tipo de comentário no programa, não ocasiona nenhuma interferência direta. Sua periculosidade está no fato de que em possíveis manutenções a serem realizadas no programa, tais comentários possam induzir a erros.

b.7) Utilização de Variáveis/Constantes: consiste em verificar se a forma de utilização das variáveis e constantes está sendo feita de maneira apropriada.

A utilização incorreta de variáveis e constantes pode levar à execução de processamentos não previstos

A verificação da utilização das variáveis e constantes nas rotinas é realizada por intermédio da inspeção dos seguintes itens:

- Se for utilizado vetor no interior de um laço, cujo índice é alterado no laço, deve-se verificar a coerência entre o controle de terminação e a dimensão do vetor.
- Todas as variáveis que são alteradas em rotinas acionadas por interrupção devem ter os seus acessos fora dessas rotinas, protegidos pelos comandos de habilitação e desabilitação de interrupções
- Uma constante só deve ser definida uma única vez, para cada linguagem utilizada, em um arquivo de declarações, e não em diversos arquivos. O arquivo de declarações deve ser incluído nos diversos módulos do programa. Tal procedimento tem como finalidade evitar que no caso de ser necessária alteração no valor de constantes em mais de uma posição de código fonte, leve a condições inconsistentes.
- Os tipos de variáveis presentes na declaração externa devem ser coerentes com a declaração estática.

  Exemplo:

  Declaração externa:

  ```
  extern char *test
  ```

  Declaração estática:

  ```
  int test [80]
  ```

- Todas as variáveis utilizadas no programa devem ser iniciadas de maneira correta.

  A não iniciação, ou a iniciação incorreta de uma variável pode ocasionar uma situação não prevista no processamento do código, notadamente na primeira execução de determinados trechos de rotina.

b.8) Comentários do Código Fonte: os comentários presentes no código fonte têm como finalidade melhorar a compreensão do significado do código, tornando mais simples a tarefa de manutenção do programa.

Tendo em vista esta importante função, os comentários presentes no código fonte não devem, de forma alguma, levar a conclusões incorretas, nem conter ambigüidades, ou seja, levar a dúvidas na interpretação do código.

Assim como no caso de código não utilizado e mantido no programa no interior de seções comentadas, a presença de comentário errado ou ambíguo no programa não ocasiona nenhuma interferência direta. Novamente, o problema está no fato de que em possíveis manutenções a serem realizadas no programa, o programador seja induzido a erros.

b.9) Legibilidade de Código: a legibilidade tem um papel fundamental na manutenção de um código de um programa, visto que um código de difícil leitura, torna muito mais penoso o trabalho de entendimento, e por conseguinte induz a erros. Tais erros podem ocorrer não apenas na manutenção do programa, mas também na própria codificação inicial.

Alguns tópicos a serem verificados no que se refere à legibilidade de código são:

- Devem ser evitadas abreviaturas em nomes de constantes, variáveis e rotinas, bem como nomes muito extensos que dificultem a leitura;
- Deve ser utilizado um mecanismo para delimitação de palavras, como por exemplo o uso de caracteres separadores;
- Os nomes devem seguir a critérios coerentes e consistentes (por exemplo, os nomes das rotinas devem ser formas verbais sempre no mesmo tempo, mesma pessoa e mesmo idioma);
- Deve-se utilizar prefixos ou sufixos nos símbolos funcionalmente relacionados, como por exemplo ponteiros;

  Exemplo:

```
BYTE Modo
BYTE * p-Modo
```

- Constantes e variáveis devem ser representados de forma distinta, como por exemplo, nomes compostos por letras maiúsculas podem ser reservados às constantes, e por letras minúsculas às rotinas e variáveis, quando a linguagem assim o permitir;
- Outros prefixos e sufixos devem ser consistentemente utilizados (por exemplo, MAX e MIN para máximo e mínimo);

- A indentação deve ser utilizada criteriosamente (por exemplo, o comandos DO e END do PL/M devem ser alinhados);
- Parênteses devem ser utilizados sempre que contribuírem para a clareza das expressões, mesmo que sejam semanticamente desnecessários;

  Exemplo:

  A + (B * C) / D

  A + B * C / D
- Evitar estruturas demasiadamente complexas, quer devido à excessiva quantidade de operações, quer à excessiva profundidade de aninhamento.

b.10) Diretivas de Compilação: o uso indiscriminado e em grande número de diretivas de compilação no código fonte, podem vir a constituir outra causa de erros em futuras manutenções. A alteração direta ou indireta do valor de controle de uma diretiva de compilação pode ocasionar a não compilação ou compilação errônea de partes do código. Erros deste tipo só podem ser identificados através da análise da saída do compilador.

b.11) Otimização de Código: o uso de otimizações no processo de compilação pode provocar a geração de código objeto com comportamento diferente do esperado, uma vez que as linguagens de alto nível apresentam certos aspectos ambíguos, podendo ser interpretada de forma diferente pelo compilador, dependendo da otimização selecionada.

A lista de inspeção apresentada não tem a pretensão de ser exaustiva, abordando os principais aspectos a serem verificados quando do trabalho de inspeção de código, principalmente aqueles utilizados em sistemas críticos quanto à segurança.
A lista apresentada deve ser complementada, não apenas à medida em que novas linguagens venham a ser utilizadas, mas também novas características de programação venham a ser utilizadas nos sistemas críticos.

c) Análise de Perigo do Software – Elaboração de Simulações com Autômatos Híbridos

Como objetivo de ilustração, é apresentado, a seguir, um experimento da aplicação dos Autômatos Híbridos numa malha metroviária simplificada. [Moura 00] [Bonifácio 99]
O modelo executado refere-se à modelagem da dinâmica de aproximação de um trem a

uma região de AMV. [Haxthausen 00] Os Requisitos Gerais de Segurança correspondem aproximadamente aos RGS2, RGS4 e RGS18, onde o AMV deve estar posicionado corretamente e travado. A simulação realizada procura verificar até que ponto os Requisitos Gerais de Segurança podem ser relaxados de forma que o sistema ainda permaneça seguro.

A configuração da via adotada é apresentada na figura 4.3 com os possíveis sentidos de movimento.

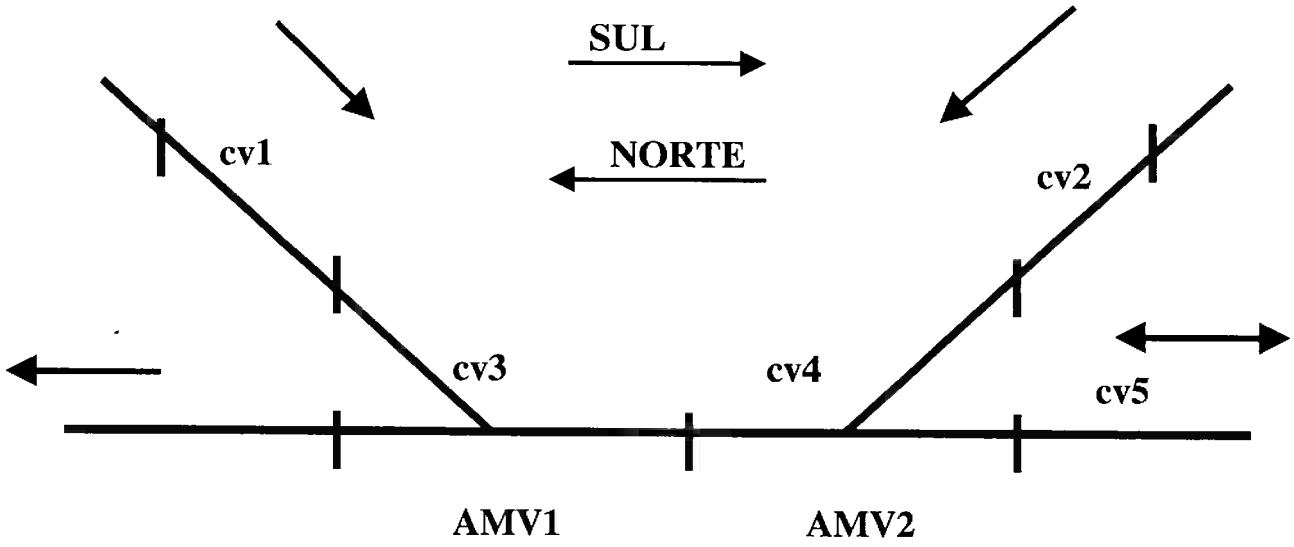


Figura 4.3 – Configuração Simplificada da Via

Para esta simulação foi adotado um modelo de operação hipotético visando verificar o grau de relaxamento dos RGSs. Neste exemplo, um trem pode percorrer os circuitos de via **cv1**, **cv3**, **cv4** e **cv5** (sentido Sul), inverter seu sentido de movimento e voltar através dos circuitos de via **cv5**, **cv4** e **cv3** (sentido Norte). Um outro trem pode realizar o percurso através dos circuitos de via **cv2**, **cv4** e **cv3** (sentido Norte). Em qualquer uma das possíveis rotas, os circuitos de via posteriores ao trem devem estar desocupados e os respectivos AMVs posicionados na posição correta e travados quando o trem ocupar o seu respectivo circuito de via. O tempo de movimentação da máquina de chave pertencente ao AMV foi adotado como sendo de 15 segundos com uma tolerância de até 20%. Considerou-se também que os circuitos de via têm tamanho de 200 metros e foi adotado como Requisito Específico de Segurança que, em havendo ocupação em torno de 20 metros do AMV, ele já deve estar posicionado corretamente e travado. Para efeito de simulação adotou-se a velocidade do trem como sendo de 7 a 9 metros por segundo.

Através das simulações executadas observou-se não haver nenhuma situação perigosa, de acordo com os critérios adotados.

Na realidade esta aplicação ocorreu num sistema bastante simplificado. No entanto, vale ressaltar que este tipo de método é extremamente adequado para análises pontuais, onde as interfaces do problema a ser estudado são bem definidas. A ferramenta utilizada é denominada Hytech.

4.1.2.3 Análise Final do Perigo - Avaliação Quantitativa da Segurança de Aplicações Microprocessadas

A seguir são apresentadas as premissas adotadas e os resultados obtidos considerando-se valores hipotéticos para as taxas de falhas envolvidas além de algumas simplificações na arquitetura utilizada do ATP de via.

A arquitetura do ATP de via considerado constitui-se numa arquitetura TMR – Triple Modular Redundancy, com votação majoritária, conforme apresentado na figura 4.4.

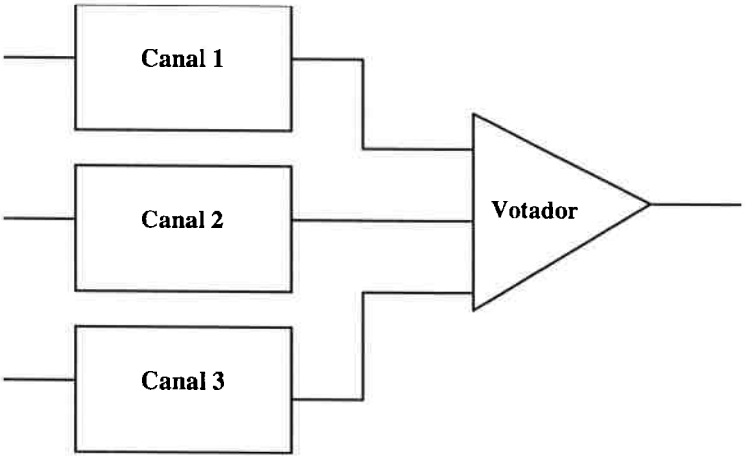


Figura 4.4 - Arquitetura TMR do Sistema ATP de Via

a) Descrição do Hardware

Cada canal redundante do sistema TMR é constituído basicamente pelos seguintes módulos, conforme apresentado na figura 4.5.

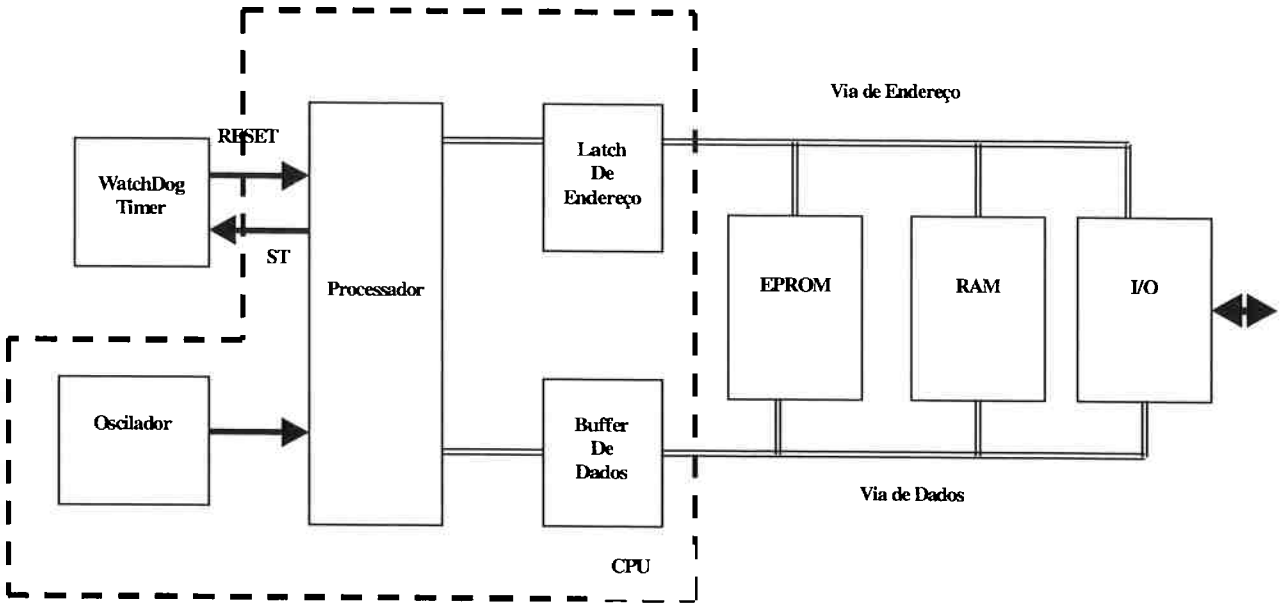


Figura 4.5. - Diagrama em Blocos do Hardware Microprocessado

O processador deve executar as instruções na memória de programa. Através desse programa, o processador pode manipular variáveis na memória de dados, ativar e desativar sinais de E/S e operar o circuito de watchdog. Alguns módulos são altamente dependententes. Durante a execução do software estes módulos trocam endereços, sinais de dados e de controle, tornando, desta forma, sua análise individual extremamente complexa. Por esta razão, o Processador, o Oscilador, o Latch de Endereço e o Buffer de Dados, são todos considerados como um único módulo funcional, denominado de UCP – Unidade Central de Processamento. A função do WatchDog é monitorar a execução do software e o fornecimento de energia. Este circuito pode aumentar bastante a segurança final do sistema. O processador deve ativar continuamente a saída ST num intervalo de tempo pré-determinado. Se uma determinada falha mantiver o sinal ST num valor estável, o circuito do WatchDog irá enviar um RESET ao processador, levando-o a um estado conhecido e seguro.

b) Considerações de Segurança

Para cada um dos módulos, UCP, WatchDog, Memória de Programa, Memória de Dados e E/S, são avaliados os parâmetros A_{SW}, C_{SW}, A_{HW} and C_{HW}, conforme item 3.4.9, ou a aplicação da técnica FMECA, quando possível. Neste estudo de caso, as falhas perigosas do software que não têm relação com o hardware não foram consideradas, devendo serem avaliadas através de técnicas qualitativas e não quantitativas. A seguir é apresentada a avaliação quantitativa para cada um dos módulos envolvidos.

CPU – Não é possível haver detecção de falhas em todos os blocos deste componente devido às limitações dos meios de detecção de falhas. Além deste aspecto, falhas na CPU podem afetar a temporização geral do sistema tornando imprevisível a conseqüência de tais eventos. Neste estudo de caso foi considerado apenas o WatchDog como meio de detecção de falhas neste módulo.

O Requisito Específico de Segurança da CPU é que uma falha não deve mudar a execução do programa de forma perigosa.

Tendo como referência as considerações anteriores pode-se afirmar que a probabilidade do software detectar falhas na CPU é considerada zero, já que a CPU é um recurso fundamental para a própria execução do software. ($A_{SW} = 0$). Trata-se de uma assertiva pessimista, conforme requerido num trabalho de análise de perigo. Desta forma, o fator de cobertura do hardware (C_{HW}) irá depender da disponibilidade do circuito WatchDog (A_{HW}).

Circuito de WatchDog – O objetivo deste circuito é detectar falhas no sistema e providenciar a reconfiguração necessária. Na eventualidade de uma falha no circuito de WatchDog que cause a não ativação do sinal RESET, algumas falhas do sistema não serão detectadas.

O Requisito Específico de Segurança para o Circuito de WatchDog é que falhas internas neste circuito não devem prevenir a ativação do sinal RESET após da ocorrência do time-out do sinal ST.

Normalmente o circuito de WatchDog não é um circuito complexo e, desta forma, é possível realizar o seu FMECA, além de avaliar as taxas de falhas inseguras detectáveis e não detectáveis.

Memória de Programa – O problema aqui corresponde à ocorrência de falhas que possam afetar o conteúdo das posições acessadas, mudando qualquer instrução de código, dados ou endereços de variáveis na memória de dados. Esta falha pode causar a

execução do software em uma seqüência incorreta ou o processamento de funções com dados errados, que pode ser perigoso.

O Requisito Específico de Segurança para a Memória de Programa é que qualquer falha não deve mudar a informação armazenada de forma perigosa.

Um mecanismo de detecção de falha implementado por software é o Teste de Memória de Programa. A função deste teste é garantir a integridade da memória. Quando uma falha é detectada por meio deste teste (C_{SW}), o processamento deve ser paralisado, conduzindo o sistema para um estado restritivo ou seguro.

Entretanto, como as instruções para a execução deste teste estão localizadas na própria memória de programa, as posições da memória que contêm estas instruções devem estar em perfeitas condições. As falhas na memória de programa devem estar então localizadas fora da região onde está o programa de teste. Desta forma, é possível determinar a porcentagem da memória de programa que não contém estas instruções de código. A disponibilidade da UCP é também necessária para que o teste de memória de programa seja executado. No entanto, falhas na CPU já foram consideradas na análise do bloco UCP. Então valor de A_{SW} irá depender da porcentagem da memória de programa que não contém as instruções desta rotina de teste.

Qualquer falha que ocorra na região de memória que contenha estas instruções será considerada como não detectável. Nestes casos, a detecção de falhas irá depender do fator de cobertura do circuito de WatchDog (C_{HW}). A disponibilidade do hardware necessário, quando o teste de memória de programa não é executado, irá corresponder à porcentagem da memória de programa que contém os códigos de instrução e da disponibilidade do circuito WatchDog (A_{HW}).

Memória de Dados – Qualquer falha no conteúdo desta memória pode fazer com que o software processe informações erradas armazenadas na memória. Esta situação pode conduzir o sistema a um estado perigoso.

O Requisito Específico de Segurança da Memória de Dados é que qualquer falha nessa memória não deve modificar seu conteúdo de forma a propiciar a ocorrência de um estado perigoso.

Há alguns tipos de teste que são implementados para detectar este tipo de falha. Estes testes são Teste Dinâmico de Memória, Teste Completo de Memória, e Teste de Endereçamento de Memória. O Teste Dinâmico de Memória é executado periodicamente enquanto o Teste Completo de Memória é executado durante a iniciação do sistema. O Teste Dinâmico verifica algumas posições de memória em cada interação.

O objetivo de ambos os testes é detectar falhas na memória de dados durante uma operação de escrita e leitura. O Time-out para se detectar as falhas pelo Teste Completo está relacionado com o tempo médio entre duas iniciações.

O Teste de Endereçamento de Memória tem por objetivo detectar falhas no endereçamento da memória de dados e também endereçamento da memória de programa, pois ambas dividem as mesmas linhas de endereço.

Falhas na memória de dados podem ser detectadas através dos testes dinâmicos e completo da memória (C_{SW}) desde que haja disponibilidade do hardware necessário (A_{SW}), ou seja, UCP e Memória de Programa. É assumido que nenhum outro recurso possa detectar falhas na memória de dados, fazendo com que C_{HW} seja igual a zero. Entretanto, deve ser observado que falhas na UCP ou na Memória de Programa já foram considerados, respectivamente, nos blocos UCP e Memória de Programa. Neste caso, o valor de A_{SW} é considerado igual a um.

Interface E/S – Qualquer falha na interface E/S deve, a princípio, ser detectada por testes de software, desde que o hardware necessário (UCP e Memória de Programa) estejam disponíveis (A_{SW}) Da mesma forma que no caso anterior este valor será considerado igual a um. O Fator de Cobertura implementado por software (C_{SW}) irá depender da completude da implementação deste teste. É assumido que nenhum outro recurso pode detectar falhas na Interface E/S, fazendo com que C_{HW} seja igual a zero.

A Tabela 4.1 apresenta os resultados anteriores de forma resumida.

Faults in	$\lambda_{UDHW} = (A_{SW} * C_{SW} + A_{HW} * C_{HW}) * \lambda_{UHW}$			
	A_{SW}	C_{SW}	A_{HW}	C_{HW}
CPU	0	Não importa	Disponibilidade do WDT	Fator de Cobertura de falhas no WDT
WDT	Avaliado através do FMECA			
Memória de Programa (MP)	Porcentagem da MP que não contém instruções de código	Fator de Cobertura de Falha do Teste de Memória de Programa	Disponibilidade do WDT e Porcentagem da MP que contém instruções de código	Fator de Cobertura de falhas do WDT
Memória de Dados (MD)	1	Fator de Cobertura de Falhas do teste de Memória de Dados	Não importa	0
I/O	1	Fator de Cobertura de Falha do Teste de I/O	Não importa	0

Tabela 4.1 – Avaliação dos Parâmetros Constituintes de λ_{UDHW}

Aplicando esta metodologia ao sistema ATP de via e, considerando dados de taxas de falhas obtidos através da norma MILHDBK 217, chegou-se a valores de MTTUF conforme apresentado na Tabela 4.2. Foram considerados N subsistemas TMR em série.

MTTUF - Anos			
Número de Subsistemas	**λu = 0.1 % λmodule**	**λu = 1.0 % λmodule**	**λu = 10 % λmodule**
1	1.098.770	1.022.700	587.009
2	549.385	511.350	293.504
3	366.256	340.900	195.669
4	274.692	256.676	146.752
5	219.754	204.540	117.401

Table 4.2 – Resultados do MTTUF em Função do Número de Subsistemas TMR

4.1.2.4 Conclusões

Deste trabalho de aplicação da metodologia de análise de risco em sistemas metroviários pode-se concluir que as diversas etapas desta metodologia fornecem uma referência aos analistas, fazendo com que eles mantenham o foco e a precisão de análise conforme o nível do módulo sendo avaliado. Vale ressaltar também que a problemática envolvida com a segurança em sistemas computacionais requer um vasto conhecimento em diversas áreas, procurando integrar diversos métodos de análise de perigo dentro de uma metodologia de análise. A combinação de métodos formais, semi-formais e informais é de fundamental importância em função da complementariedade de suas aplicações.

4.2 Sistema de Controle e Supervisão de Tráfego Aéreo - CNS/ATM

Este estudo da segurança do sistema CNS/ATM constitui-se num trabalho de pesquisa em processo de iniciação. Nesta etapa inicial há a participação de pesquisadores do GAS da EPUSP e de profissionais do IPV – Instituto de Proteção ao Vôo. Este projeto de pesquisa conta também com o apoio da Embraer e da TAM. Neste sentido, o objetivo deste item é apresentar um plano de aplicação da metodologia de análise de risco, apresentada no capítulo 3, no sistema CNS/ATM, além de alguns resultados preliminares de alguns experimentos já realizados.

4.2.1 Descrição Funcional do Sistema CNS/ATM

Por definição o sistema CNS/ATM é "a aplicação de sistemas de comunicação, navegação e vigilância (CNS – Communication, Navigations, Surveillance), empregando tecnologias digitais avançadas ou tecnologia de satélite junto a outros níveis de automação, oferecendo suporte a um sistema de gerenciamento de tráfego aéreo (ATM – "Air Traffic Management") global sem barreiras. [CANSO 99]

A aplicação do conceito CNS/ATM, cujo objetivo maior é a segurança da aviação, prevê uma comunicação de dados precisa entre o piloto do avião e o controlador de tráfego aéreo. O meio de comunicação mais comum existente atualmente é a comunicação via rádio - freqüência VHF e HF, que possue algumas limitações, como por exemplo, o fato das ondas de rádio viajarem em linha reta e não acompanharem a curvatura da terra. Para grandes distâncias, essa tecnologia é ineficaz, razão pela qual já se tem feito uso de satélites para compensar as perdas.

Os conceitos envolvidos no sistema CNS são definidos como: **Comunicação** é a troca de informação de voz e dados entre o avião e os controladores de tráfego aéreo (ATC – "Air Traffic Control") ou Centros de Informação de Vôo; **Navegação** apresenta a localização do avião através de aparelhos que conseguem determinar sua posição e direcioná-lo ao seu destino; **Vigilância** apresenta a localização do avião para os controladores de tráfego aéreo, incluindo a comunicação de informações de navegação do avião para os ATCs, garantindo que as distâncias seguras entre aeronaves possam ser respeitadas.

Atualmente, em várias regiões do mundo, a menor distância aérea entre dois pontos ainda se caracteriza por um conjunto de trechos que são supervisionados por equipamentos de auxílios à rádio - navegação. As separações entre as aeronaves que sobrevoam os mares são significativamente maiores do que quando sobrevoam regiões dotadas de radares e outros serviços de apoio. A voz, com todos seus sotaques e limitações, ainda é o principal meio de comunicação entre as aeronaves e os ATCs em solo.

Em função destes problemas e do congestionamento de algumas áreas com a taxa crescente da demanda do transporte aéreo, as autoridades, responsáveis pela manutenção e adequação do sistema de controle do tráfego aéreo mundial resolveram implantar um novo conceito. Com o uso de uma constelação de satélites se viabilizará a plena expansão do transporte aéreo no novo século, atingindo-se níveis de segurança e de operacionalidade que jamais seriam obtidos com os convencionais sistemas de controle de tráfego.

O Sistema CNS/ATM constitui-se numa solução global que virá a substituir os sistemas unicamente regionais. Da etapa inicial do sistema CNS/ATM participam os sistemas GPS (*Global Positioning System*) dos EUA [Hurn 89] [Hurn 93], e Glonass (*Global Navigation Satellite System*) da Rússia. [Machado 2000] Ambos os sistemas

compreendem grandes redes de satélites com capacidade para estabelecer a posição exata de objetos e até de pessoas na superfície da Terra, no mar ou no espaço aéreo.
Entre os objetivos pretendidos pelo novo sistema, pode-se destacar:

- ligações mais diretas e eficientes entre solo e sistemas embarcados;
- melhor processamento e transferência de dados entre operadores, aeronaves e fornecedores do ATM, bem como a melhoria da tratamento e da transferência de informações, principalmente, fazendo uso de técnicas de *data-link*;
- reduções do congestionamento dos canais de comunicação e dos erros devido às más interpretações de comunicações, aumentando os níveis de segurança;
- ampliação da vigilância das aeronaves e o aperfeiçoamento dos níveis de precisão da navegação, com o uso do sistema de navegação global através do auxílio de satélites;
- em função do melhor nível de segurança, aumentar a capacidade do espaço aéreo e criar um sistema confiável de comunicações e de navegação, que seja homogêneo em todo o planeta;
- integrar indústria e companhias aéreas;
- facilidades na interoperacionalidade, com mínimos requisitos de aviônicos, bem como redução dos custos e melhoria da eficiência com a padronização da rede de telecomunicações aeronáuticas.

O tráfego aéreo, dentro do novo conceito, tem como objetivo proporcionar maiores níveis de segurança, redução de esperas e aumento da capacidade de aeroportos e do próprio espaço aéreo. As operações oceânicas podem tornar-se mais flexíveis, dando oportunidade à escolha de trajetórias mais convenientes. O aperfeiçoamento do controle tático propicia melhores condições para o estabelecimento de fluxos convergentes de tráfego para os aeroportos. Desta forma, conflitos tendem a ser minimizados. Até a maior flexibilidade nas operações de aproximação permitirá procedimentos que causem menor incômodo sonoro às populações adjacentes aos aeroportos com grande movimento.
O Brasil possui uma política para implantação do CNS/ATM no país. O documento data de 1994 e foi elaborado pelo Ministério da Aeronáutica, que participou através de um Comitê Especial, de diversas reuniões da ICAO - "International Civil Aviation

Organization", auxiliando na elaboração da concepção global do CNS/ATM. Dentre os pontos abordados nesse documento existem duas questões importantes: soberania (e.g., limites dos espaços aéreos) e responsabilidade civil (e.g., responsabilidades dos países provedores e usuários dos sistemas globais).

A estratégia de implantação dessa tecnologia no país pretende seguir Normas e Métodos recomendados pela ICAO, destacando-se que existe a intenção de participar do provimento de meios adicionais de Sistemas Globais de Navegação, visando assegurar a segurança e a confiabilidade de sua operação no país.

- A transição esperada para o sistema CNS/ATM será uma das maiores já realizadas pela comunidade da aviação, não somente por causa da sua imensa escala, mas porque isso afetará definitivamente a forma como todas as empresas relacionadas à aviação irão oferecer serviços de tráfego aéreo.

4.2.2 Planejamento da Aplicação da Metodologia de Análise de Risco

O processo de Análise de Risco que está sendo adotado neste sistema tem como referência a metodologia apresentada no capítulo 3 desta tese.

A Metodologia de Trabalho é composta basicamente por duas grandes atividades. A primeira tem como objetivo uniformizar o conhecimento entre as equipes participantes, tendo em vista a multi - disciplinariedade envolvida neste projeto. A equipe envolvida neste projeto de pesquisa é composta por vários especialistas do IPV além de 5 professores do GAS com seus respectivos orientados de mestrado e doutorado. A segunda atividade aborda o estudo da segurança da implantação do CNS/ATM no espaço aéreo brasileiro.

Cada uma destas grandes atividades é subdividida em diversas etapas, com o intuito de se fazer um maior detalhamento da metodologia proposta.

Desta forma, este item tem o objetivo de apresentar o Plano de Trabalho, visando a Análise de Risco dos sistemas envolvidos no CNS/ATM, apresentando também alguns resultados inicias, bem como as perspectivas de futuras pesquisas na área.

Todas as etapas envolvidas nesta metodologia englobam aspectos de pesquisa de ponta, seja no desenvolvimento de novas técnicas de avaliação, seja na aplicação das mesmas na implantação de um novo sistema de gerenciamento e controle do tráfego aéreo, onde há a predominância de técnicas digitais avançadas em conjunto com a tecnologia de satélites.

As atividades deste Plano de Trabalho são:

- **Atividade 1 – Uniformização do Conhecimento entre as Entidades Participantes**

Esta atividade é composta basicamente por cinco etapas:

- Pesquisa Bibliográfica sobre o Sistema CNS/ATM
- Estudo Detalhado do Estágio Atual do Sistema CNS/ATM
- Estabelecimento de Convênios Internacionais/Nacionais com Universidades/Institutos de Pesquisa
- Estudo de Métodos a serem utilizadas no Processo de Análise de Risco
- Participação em Congressos Nacionais/Internacionais a respeito do sistema CNS/ATM

- **Etapa 1: Pesquisa Bibliográfica sobre o Sistema CNS/ATM**

  Esta etapa tem o objetivo básico de manter atualizados os profissionais integrantes deste Plano de Trabalho, no que se refere ao material bibliográfico relacionado com a especificação e desenvolvimento do sistema CNS/ATM. Dentre esta bibliografia pode-se citar artigos, livros bem como normas/recomendações elaboradas por órgãos internacionais como ICAO – "International Civil Aviation Organization", FAA- Federal Association Aviation", NASA – "National Aeronautics and Space Administration" , IATA - "International Air Transport Association", IEEE – "Institute of Electrical and Electronics Engineers", IEC – "International Electrotechnical Commision".

  Esta etapa está em curso já há um ano, tendo sido realizadas duas reuniões globais no IPV em São José dos Campos e uma reunião setorial no GAS em São Paulo.

- **Etapa 2: Estudo Detalhado do Estágio Atual do Sistema CNS/ATM no Brasil e no Exterior**

  Nesta etapa será realizado o estudo detalhado dos aspectos funcionais e operacionais do sistema CNS/ATM, com a finalidade de obter as informações necessárias para a realização da Análise de Risco do Sistema CNS/ATM.

  Esta etapa também está em curso, sendo discutidos dentro da equipe a junto ao IPV os aspectos funcionais e operacionais do CNS/ATM a serem implantados no Brasil.

- **Etapa 3: Estabelecimento de Convênios Internacionais/Nacionais com Universidades/Institutos de Pesquisa**

  Ao longo desta etapa realizar-se-ão convênios internacionais/nacionais com Universidades e Instituições de Pesquisa envolvidas com a pesquisa de segurança

em sistemas críticos. Através destes convênios poderão ser realizadas palestras convidadas, cursos de aperfeiçoamento, bem como visitas técnicas. Ainda no âmbito destes convênios, poderá haver a participação de profissionais convidados a participarem do Estudo da Segurança do sistema CNS/ATM no Brasil.

Esta etapa ainda não está em implantação, devendo ser iniciada a medida que a pesquisa na área de segurança do sistema CNS/ATM estiver avançando.

- **Etapa 4: Estudo de Técnicas a serem Utilizadas no Processo de Análise de Risco.**

  Esta etapa tem como objetivo manter os pesquisadores, de todas as entidades envolvidas neste Plano de Trabalho, atualizados com relação às técnicas mais recentes que possam vir a ser utilizadas no Processo de Análise de Risco.

- **Etapa 5: Participação em Congressos Nacionais/Internacionais a respeito do Sistema CNS/ATM**

  Esta etapa visa proporcionar uma maior atualização dos pesquisadores sobre os conceitos envolvidos na área de segurança, bem como o acompanhamento dos estudos do sistema CNS/ATM realizado através de outras entidades nacionais/internacionais. Outro objetivo é manter uma interação constante dos pesquisadores com os profissionais envolvidos no desenvolvimento e implantação do sistema CNS/ATM nos demais países.

- **Atividade 2 – Estudo da Segurança da Implantação do CNS/ATM no Brasil**

Esta atividade engloba duas grandes etapas, o Gerenciamento do Risco e a Análise do Risco propriamente dito.

- **Etapa 1: Gerenciamento do Risco na Implantação do Sistema CNS/ATM no Brasil**

  Estudo da Metodologia de Gerenciamento de Risco e aplicação dessa metodologia ao sistema CNS/ATM.

  O principal aspecto é o comprometimento com a segurança por parte do gerenciamento. As discussões sobre segurança devem criar uma atmosfera de cooperação e não acusação.

  As sub-etapas dentro deste processo de Gerenciamento de Risco são:

Etapa 1.1. Definição de Políticas de Segurança e Objetivos

As Políticas de Segurança definem o relacionamento entre segurança e os demais objetivos, as vezes conflitantes, decidindo o que deve ser feito em situações especificas.

Etapa 1.2. Estabelecimento de Responsabilidades, Autoridade e Escopo de Atividades

Devido a existência de conflitos entre Segurança e outros objetivos, a independência do Grupo de Segurança é importante. A responsabilidade de se atingir a Segurança deve ser separada da responsabilidade de se atingir outros objetivos.

Etapa 1.3. Estabelecimento de Canais de Comunicação

As informações devem chegar rapidamente nas pessoas certas. É fundamental estabelecer uma comunicação eficiente entre o grupo de desenvolvimento do software de segurança e os demais grupos envolvidos com a segurança do sistema.

Etapa 1.4. Definição das Obrigações, Responsabilidades e Características da Organização da Segurança do Sistema

Nesta etapa podem ser abordados os seguintes pontos, com relação à Organização da Segurança:

- Ser de nível alto e independente;
- Formular e Implementar a Política de Segurança;
- Documentar o rastreamento dos Perigos e suas resoluções;
- Adaptar e Desenvolver Normas/Recomendações;
- Conduzir/ Participar na Análise de Risco (Etapa 2);
- Planejar e monitorar os testes das tarefas de segurança;
- Participar em Revisões do Programa;
- Manter Intercâmbio com outros Grupos de Segurança;
- Investigar e Análisar Acidentes.
- Qualificar Profissionais para trabalharem em Análise de Risco
- Definir responsabilidades no caso de subcontratações
- Estabelecer Grupos de Trabalho

- **Etapa 2: Análise de Risco da Implantação do Sistema CNS/ATM no Brasil**

Dentro desta etapa da destacam-se dois aspectos fundamentais: as metas a serem atingidas através desta análise e as atividades constituintes deste processo de Análise de Risco.

O objetivo principal é o desenvolvimento acadêmico de pesquisa na área de Confiabilidade e Segurança aplicada ao sistema CNS/ATM "Communication, Navigation, and Surveillance/ Air Traffic Management".

Os resultados deste Projeto deverão ser disponibilizados à Diretoria de Eletrônica e Proteção ao Vôo, do Comando da Aeronáutica, que poderá, utilizá-los na determinação de parâmetros na área de Confiabilidade e Segurança para a implantação do CNS/ATM no Brasil.

- **Etapa 2.1 Metas do Processo de Análise de Risco da Implantação do Sistema CNS/ATM no Brasil**

O processo de implantação do Sistema CNS/ATM no Brasil prevê a substituição gradativa dos diversos sistemas atuais de controle, tanto em terra, quanto embarcados. Para cada etapa desse processo de implantação, os sistemas que substituirão os atuais serão alvo de um trabalho de Análise de Risco.

As principais metas deste trabalho de Análise de Risco são:

- Identificação dos Estados Perigosos, ou seja, aqueles em que o sistema está exposto a um acidente.
- Determinação dos possíveis efeitos decorrentes dos estados perigosos existentes no sistema
- Avaliação das causas fundamentais relacionadas com os estados perigosos compreendendo:
  - Determinação de como o estado perigoso possa ser alcançado
  - Determinação da inter-relação entre as diversas causas dos estados perigosos
  - Avaliação da influência do operador do sistema na segurança, através do estudo da confiabilidade humana.
- Identificação de Recomendações/Especificações/Critérios de Segurança que irão auxiliar no projeto, na determinação de dispositivos de segurança e na

identificação de procedimentos que eliminam ou, pelo menos, minimizam os efeitos decorrentes do alcance de um estado perigoso

- Determinação dos Requisitos Gerais de Segurança – RGS, dos Níveis de Risco Aceitáveis, bem como a identificação dos Níveis de Integridade de cada subsistema integrante do sistema CNS/ATM. Os Níveis de Integridade determinam o grau de responsabilidade de cada subsistema em relação aos aspectos de segurança. Quanto menor o Risco Aceitável, maior o Nível de Integridade associado.
- O Estabelecimento de uma proposta de **Plano Global de Segurança** para, a critério e ao arbítrio da DEPV, auxiliar na implantação e certificação do Sistema CNS/ATM no Brasil.

  Este **Plano Global de Segurança** deve contemplar:

  - Estabelecimento de Critérios Gerais de Segurança
  - Identificação dos Estados Perigosos associados ao sistema CNS/ATM, bem como aos seus subsistemas, além da identificação dos Níveis de Segurança aceitáveis.
  - Estabelecimento de Requisitos de Controle sobre os estados perigosos que não podem ser eliminados, visando proteger as pessoas, os equipamentos e a propriedade.
  - Estabelecimento de um Nível de Máximo Risco Aceitável quando da utilização de novas tecnologias.
  - Diminuição na quantidade de correções realizadas durante a Operação do sistema CNS/ATM.
  - Estabelecimento de uma Metodologia de Certificação do Sistema CNS/ATM, bem como de seus equipamentos envolvidos, tendo como referência normas nacionais e internacionais aplicáveis à certificação.
- **Etapa 2.2. Atividades do Processo de Análise de Risco da Implantação do Sistema CNS/ATM no Brasil**

  Para que as metas apresentadas anteriormente possam ser alcançadas, são necessárias diversas atividades de trabalho, que compõem o Processo de Análise de Risco.

  A seguir estão listadas estas atividades:

  - Definição do Escopo da Análise de Risco

- Definição e Descrição do Sistema, suas interfaces e demais informações necessárias para o trabalho de Análise de Risco
- Identificação dos Estados Perigosos através da Análise Preliminar de Perigo, envolvendo as seguintes sub-atividades:
  - Determinação de quais estados perigosos possam existir durante a operação do sistema, bem como sua magnitude relativa
  - Elaboração de recomendações, especificações e critérios que possam ser adotados como referência ao longo do desenvolvimento do sistema em questão
  - Determinação das ações iniciais visando o controle de determinados estados perigosos
  - Determinação da complexidade dos aspectos de segurança envolvidos no sistema
  - Identificação das responsabilidades técnicas e gerenciais ao longo de um processo de Certificação de Segurança
- Identificação das Causas Fundamentais dos Estados Perigosos
- Investigação das causas de acidentes já ocorridos através do levantamento de dados históricos de sistemas similares
- Classificação Qualitativa dos Estados Perigosos baseada nos seus efeitos, além de uma Classificação Quantitativa, através da associação de probabilidade de ocorrência
- Identificação de medidas preventivas e corretivas com relação aos Estados Perigosos, além da determinação de critérios gerais de projeto

- **Etapa 3: Orientação de Teses de Doutorado, Dissertações de Mestrado e Projetos de Iniciação Científica**

  Em paralelo a todos os trabalhos que fazem parte do Estudo da Segurança da implantação do Sistema CNS/ATM no Brasil, prevê-se o desenvolvimento de teses de doutorado, dissertações de mestrado e projetos de iniciação científica.

  As diversas áreas em que estes trabalhos serão elaborados, foram determinadas em função das principais necessidades de pesquisa com relação ao Estudo da Segurança da Implantação do Sistema CNS/ATM no Brasil e das linhas de pesquisa atendidas pelo GAS.

4.2.3 Resultados Preliminares

Dentro desta filosofia de trabalho, dois trabalhos iniciais já foram realizados tendo como meta obter uma maior sensibilidade dos problemas de segurança envolvidos na área da aviação. [Walters 00] O primeiro trabalho diz respeito a uma aplicação preliminar dos autômatos híbridos na análise e verificação de aspectos de segurança de um sistema de gerenciamento de tráfego aéreo. [Moura 00] [Bonifácio 99] O segundo trabalho refere-se ao estudo da influência da precisão do posicionamento das aeronaves nos níveis de segurança do controle de tráfego aéreo. [Naufal 01] Para o adequado entendimento das aplicações torna-se fundamental uma breve explicação sobre o sistema de controle de tráfego aéreo.

O gerenciamento de tráfego aéreo denominado ATM (*Air Traffic Management*) constitui-se no gerenciamento, bem como no controle de aeronaves em um determinado espaço aéreo. [Boeing 97] Para tanto, o espaço aéreo é sub-dividido entre Centros de Controle de Tráfego Aéreo, denominados ATC (*Air Traffic Control*). O ATC é responsável por toda a coordenação e resolução de conflitos no espaço aéreo, considerando-se os diversos níveis de complexidade. O julgamento humano constituí-se em uma parcela fundamental do ATC. Este fato torna-se ainda mais grave se for observado o aumento de tráfego e a sobrecarga de trabalho humano dentro dos ATCs.

Atualmente toda a comunicação entre os aviões e os controles em terra é feita por voz. Transmissores VHF conectam a aeronave ao ATC. Rádios de alta freqüência HF são utilizados para comunicações em áreas que excedam a cobertura por VHF.

A comunicação terra-ar é realizada apenas por voz, o que torna a comunicação deficiente e limitada, principalmente em função da velocidade e volume de informações a serem transmitidas.

Outro fato importante a citar é que a taxa de crescimento do tráfego aéreo mundial nos próximos 15 anos deverá se situar entre 3% a 5% ao ano. O sistema atual do ATM não está preparado para processar este volume crescente de tráfego, o que pode colocar a segurança do sistema aéreo em níveis não aceitáveis pela sociedade.

No CNS/ATM será utilizada comunicação via *Datalink* ADS-B (*Automatic Dependence Surveillance-Broadcast*) e a comunicação via satélite GPS (*Global Positioning Systems*) como formas de se intensificar e agilizar a comunicação padrão atualmente utilizada (vocal). A comunicação via ADS-B transmitirá mensagens periódicas contendo informações pertinentes à navegação aérea, tais como posicionamento, identificação,

latitude, altitude, velocidade em relação ao solo, entre outros parâmetros das aeronaves. Estes parâmetros são fundamentais em muitas funções, inclusive na Detecção de Conflitos entre aeronaves.

A troca de mensagens ADS-B pode ocorrer entre aeronaves ou entre aeronaves e ATCs. O período de comunicação ocorre entre 0,5 e 14 segundos e apresenta um alcance de 50 a 120 milhas náuticas.

O futuro ATM deverá permitir a implementação do conceito de *Free Flight*, que fornecerá maior flexibilidade na operação de sistemas aéreos e uma melhor estratégia no gerenciamento do espaço aéreo por parte de pilotos, ATCs e companhias aéreas. [Bartolomeu 2000] O principal benefício será a adoção de rotas mais econômicas através da maior eficiência e colaboração entre as companhias aéreas, resultando em uma economia de centenas de milhões de dólares.

Tanto a predição como a resolução de conflitos das aeronaves utilizam um sistema denominado TCAS– "Traffic Alert and Collision Avoidance System". Este sistema é utilizado em grande parte das aeronaves comerciais em seus Sistemas de Gerenciamento de Vôo, denominados FMS – "Flight Managment System". O TCAS utiliza-se de um algoritmo para predição de resolução de conflitos. Tais algoritmos de detecção e resolução de conflitos têm como objetivo permitir a manobra da aeronave em condições de alerta e, principalmente, em condições de conflito, de forma a manter assegurada a separação entre as aeronaves, notadamente se for observada a presença de incertezas nas ações e trajetórias das aeronaves devido à flexibilidade fornecida pelo conceito de "free flight". Uma estratégia de detecção e resolução de conflitos utilizada no TCAS é apresentada na figura 4.6.

Este diagrama apresenta os seguintes módulos:

- Interface de Comunicação: estabelece a comunicação entre aeronaves em conflito e entre aeronaves e o ATC. Coleta todas as informações necessárias para a correta execução do algoritmo de detecção e resolução de conflito.
- Resolução de Conflito: possui o algoritmo de detecção e resolução de conflito. Para tanto utiliza como principais informações a posição, a velocidade das aeronaves em alerta ou conflito e a posição angular entre as aeronaves
- Regulador: atua na aerodinâmica da aeronave de forma a evitar a colisão entre as aeronaves e garantindo a segurança.

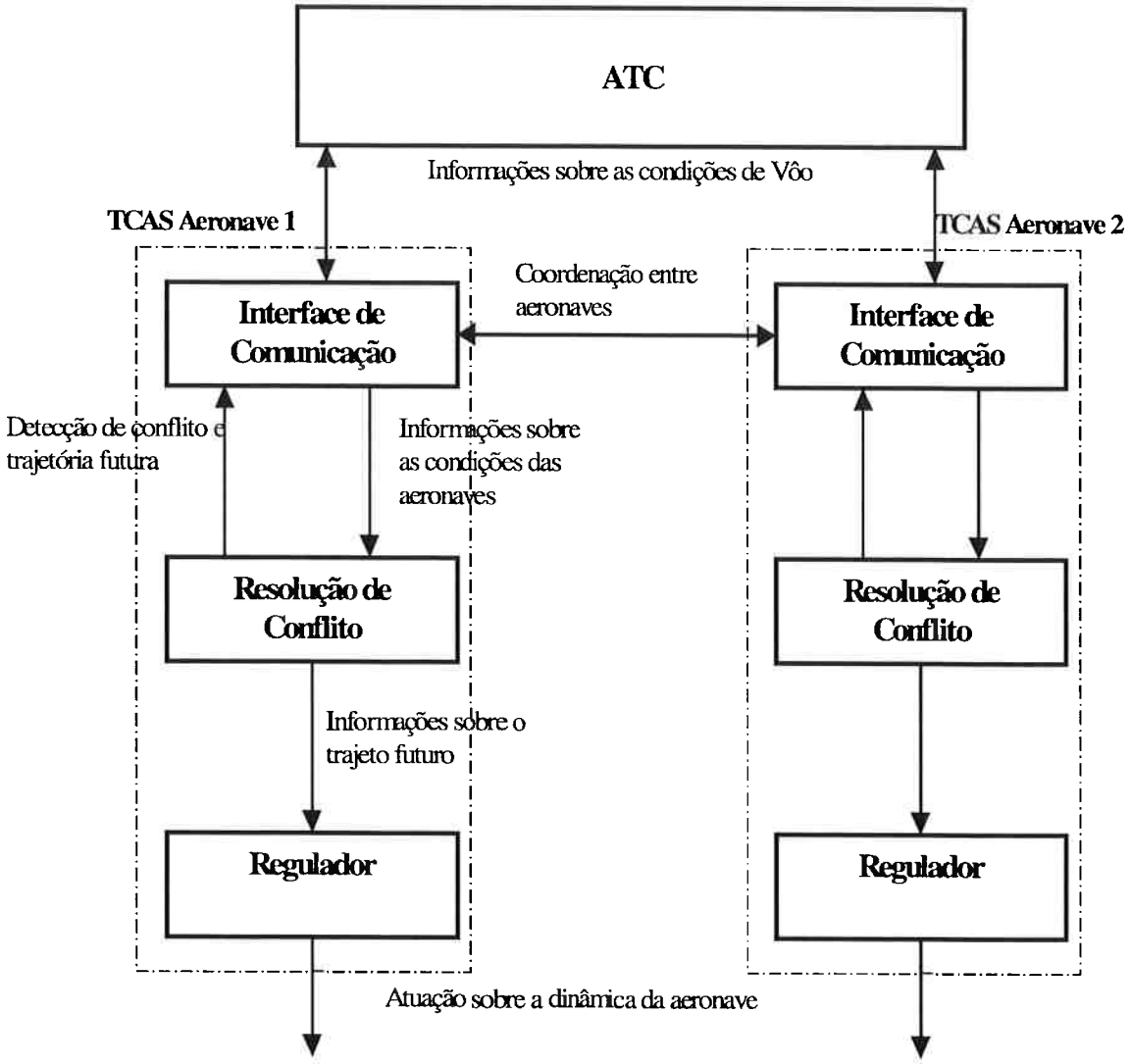


Figura 4.6 - Estratégia de detecção e resolução de conflitos utilizada no TCAS

4.2.3.1 Autômatos Híbridos na Verificação da Segurança no Tráfego Aéreo

Este trabalho descreve uma aplicação preliminar dos autômatos híbridos na análise e verificação de aspectos de segurança de um sistema de gerenciamento de tráfego aéreo. A figura 4.7 descreve a situação de duas aeronaves percorrendo o mesmo espaço aéreo.

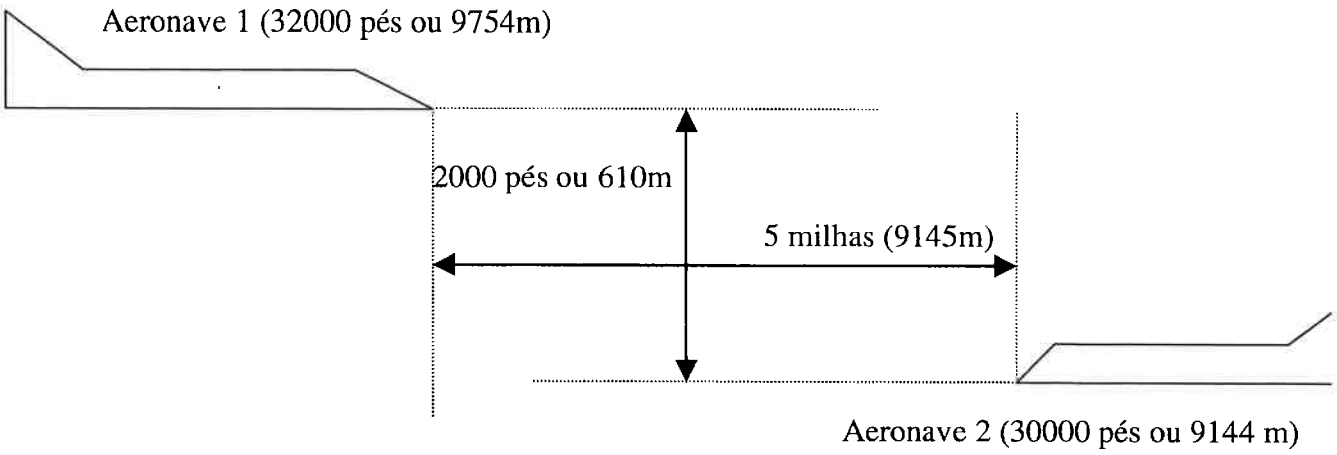


Figura 4.7 - Situação de duas aeronaves percorrendo o mesmo espaço aéreo.

Quando um avião deseja mudar sua rota deve respeitar regras de segurança. O controlador de tráfego pode enviar mensagem ao avião para aumentar a derivada de descida ou reduzir sua velocidade. No modelo aqui considerado, a aeronave 1 pode aumentar sua velocidade de descida ou arremeter, dependendo da sua distância em relação à aeronave 2. A aeronave 2 pode reduzir sua velocidade para que a aeronave 1 possa cruzar sua rota a uma distância segura. Os padrões de segurança das aeronaves variam de acordo com as circunstâncias. Acima de 29.000 pés, quando as aeronaves estão a uma velocidade alta, o exigido é 5 (cinco) milhas de separação horizontal e 2000 pés de separação vertical. Abaixo de 29.000 pés a separação vertical é reduzida para 1.000 pés e a separação horizontal permanece em 5 (cinco) milhas.

No processo de aterrisagem e decolagem o valor de separação horizontal cai para 3 (três) milhas e a separação vertical é de 1.000 pés. O valor padrão para taxa de subida e descida é de 10.000 pés por minuto. Em situações de emergência esta velocidade pode chegar a 12.000 pés por minuto. Adotou-se no exemplo em questão que as aeronaves estão a uma velocidade de 550 milhas por hora, em cruzeiro. Em situações críticas esta velocidade pode ser reduzida a 490 milhas por hora, conforme dados reais para a aeronaves Boeing 707 e Boeing 747.

No exemplo em questão, para efeito de simplificação, o protocolo adotado de funcionamento do TCAS foi:

- Quando a distância horizontal entre as aeronaves chega a 6.000 metros e se a separação vertical é maior que 300 metros, o TCAS força a aeronave 1 a subir a 50 metros por segundo, ou 10.000 pés por minuto.

- A aeronave 1 estando em movimento de descida e a distância horizontal entre as aeronaves atingir 6.000 metros, além da separação vertical ser de 300 metros, o TCAS pode agir reduzindo a velocidade da aeronave 2. No entanto, quando a distância horizontal entre as aeronaves é de 5.000 metros e a separação vertical 200 metros, o TCAS pode agir enviando um comando de subida para a aeronave 1.

Através dos autômatos híbridos criados foram simulados as seguintes situações:

a) Aeronave 1 em movimento descendente

Nesta simulação a altura atingida pela aeronave 1 foi de 8.821 metros. Como a aeronave 2 está a uma altitude de 9.144 metros, além da altitude estar abaixo de 29.000 pés, a separação vertical exigida é de 1.000 pés. Como a separação obtida pela simulação é de 1060 pés, o sistema encontra-se num valor de segurança aceitável.

b) Aeronave 1 em movimento descendente com redução de velocidade da aeronave 2.

Nesta simulação a altura atingida pela aeronave 1 é de 8.785 metros, sendo a margem de segurança um pouco maior que na situação anterior.

c) Aeronave 1 em movimento descendente com comando de subida pelo TCAS e redução de velocidade da aeronave 2.

A simulação mostrou que quando a separação horizontal é de 4.820 metros, a altitude alcançada pela aeronave 1 no ponto de cruzamento é de 9.772 metros, superior aos 2.000 pés de diferença entre as aeronaves.

4.2.3.2 Influência do Posicionamento das Aeronaves nos Níveis de Segurança

Este trabalho tem como enfoque a avaliação da relação entre os níveis de segurança do controle de tráfego aéreo e a periodicidade da atualização do posicionamento das aeronaves, tendo como meta a avaliação de seu impacto sobre os níveis de segurança desejáveis. Evidentemente diversos parâmetros influenciam nesta análise, podendo, inclusive, alguns deles apresentarem características com um certo grau de imprecisão. Neste sentido foi realizado, como primeira etapa, um levantamento do grau de influência da precisão do posicionamento e da velocidade relativa entre as aeronaves na detecção de rotas conflitantes. Como conseqüência, esta detecção de conflito leva à determinação do tipo de manobra de desvio a ser executada pelas aeronaves.

No tema do segundo trabalho, no contexto do *Free Flight*, cada aeronave deve apresentar uma determinada separação, de forma a prevenir a possibilidade de colisão entre aeronaves. Para tanto, cria-se dois cilindros virtuais, definidos como Zona

Protegida (*Protected Zone*) e Zona de Alerta (*Alert Zone*). A figura 4.8 ilustra tal conceito.

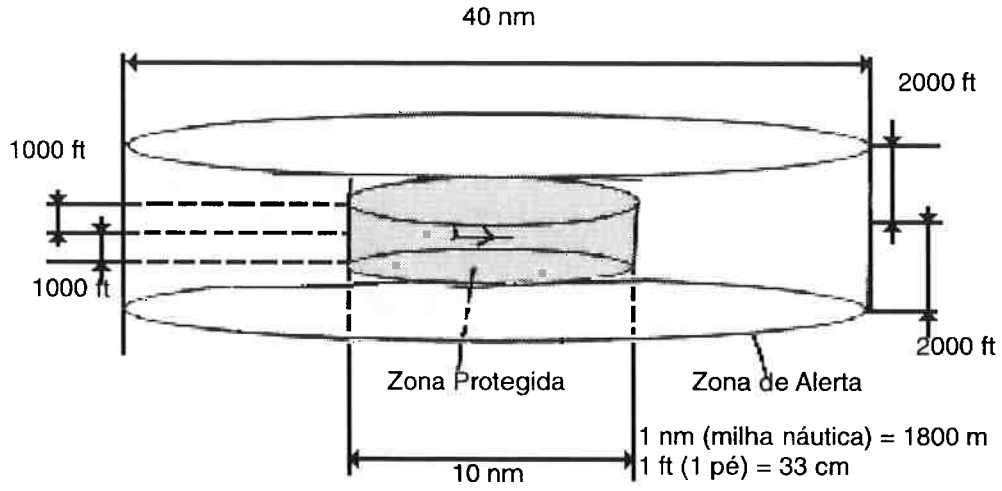


Figura 4.8 - Zona Protegida e Zona de Alerta

A Zona Protegida apresenta um cilindro com um diâmetro de 10 milhas náuticas, separação superior de 1000 pés e separação inferior também de 1000 pés. A Zona de Alerta também apresenta a forma de um cilindro, com um diâmetro de 40 milhas náuticas, 2000 pés de separação superior e 2000 pés de separação inferior entre aeronaves comerciais, conforme mostra a figura anterior.

A Detecção de Conflito ocorre quando uma aeronave ou o ATC detecta a presença de uma ou mais aeronaves em sua Zona Protegida. Quando este fato ocorre existe um risco eminente de colisão. Quando uma ou mais aeronaves são detectadas em sua Zona de Alerta existe um potencial risco de conflito. A figura 4.9 apresenta estas situações.

Na verdade ocorre uma sobreposição entre as zonas (zona de alerta e protegida) das aeronaves. O tamanho da Zona de Alerta depende de vários fatores, incluindo velocidade do ar, precisão na velocidade das aeronaves, altitude, precisão da sensibilidade dos equipamentos do ATC e a bordo da aeronave, situação de tráfego, tempo de comunicação entre ATC e aeronave, e entre aeronaves (quando ocorre a invasão da Zona de Alerta e da Zona Protegida). A Zona de Alerta pode ser subdividida em duas ou mais zonas, de forma a melhorar precisão de resolução de um conflito.

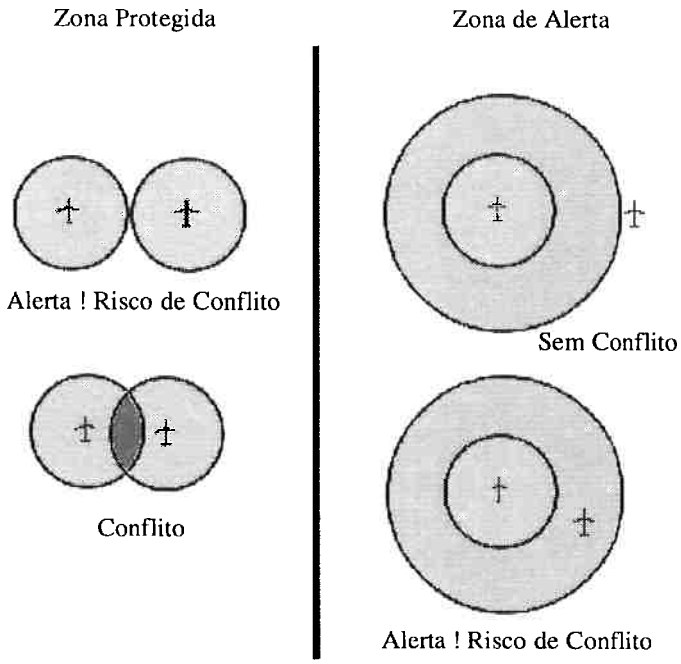


Figura 4.9 - Detecção de Conflito na Zona Protegida e na Zona de Alerta

Para a predição e resolução de conflito são considerados três níveis diferentes no processo ATM (Air Traffic Management):

- Longo alcance: a predição normalmente é realizada por um ATC em um período de dezenas de minutos. A resolução do conflito é solucionada pela aeronave em concordância com o ATC destino. O processo envolve a recomposição de planos de vôo e o replanejamento de linhas aéreas para garantir que a capacidade do aeroporto destino não seja excedida.
- Médio alcance: a predição normalmente pode ser realizada por algum ATC ou pela aeronave em um período de minutos. A resolução de conflito também é executada pela aeronave em concordância com o ATC destino. O plano de vôo planejado é alterado em tempo real é alterado e informado ao ATC, de forma a assegurar a separação adequada entre as aeronaves.
- Curto alcance: a predição normalmente é executada pela aeronave, em um período de segundos e a resolução é executada pela aeronave. Normalmente esta é uma solução de último recurso, devendo, portanto, ser evitada.

Para a predição de longo alcance utiliza-se o GPS e a comunicação ocorre entre os ATCs e as aeronaves. Para a predição de médio e curto alcance utiliza-se tanto o GPS quanto o ADS-B. A comunicação ocorrerá entre os ATCs e as aeronaves em conflito, bem como entre as próprias aeronaves em conflito.

Tanto a predição quanto a resolução de conflitos das aeronaves utilizam um Sistema de Alerta e Anti-Colisão denominado de TCAS, já explicado anteriormente. O TCAS utiliza-se de um algoritmo para a predição e resolução de conflitos.

Tais algoritmos de Detecção e Resolução de Conflitos têm como objetivo permitir a manobra da aeronave em condições de alerta (invasão de sua Zona de Alerta) e, principalmente, em condições de conflito (invasão de sua Zona Protegida), de forma a manter assegurada a separação entre as aeronaves, notadamente se for observada a presença de incertezas nas ações e trajetórias das aeronaves devido à flexibilidade fornecida pelo conceito de *Free Flight*.

Quando as Zonas de Alerta e principalmente as Zonas Protegidas se sobrepõem, as aeronaves trocam informações intensamente de forma a prever e resolver conflitos. Baseado nestas informações, os algoritmos de Detecção e Resolução de Conflito são executados, podendo aconselhar o piloto a executar a manobra passo-a-passo, ou então, de forma automática, através do piloto automático.

Informações como a velocidade relativa, distância relativa e velocidade angular entre as aeronaves são fundamentais para a operação de manobra das aeronaves. As figuras 4.10 e 4.11 apresentam os conceitos de distância relativa e velocidade angular entre aeronaves.

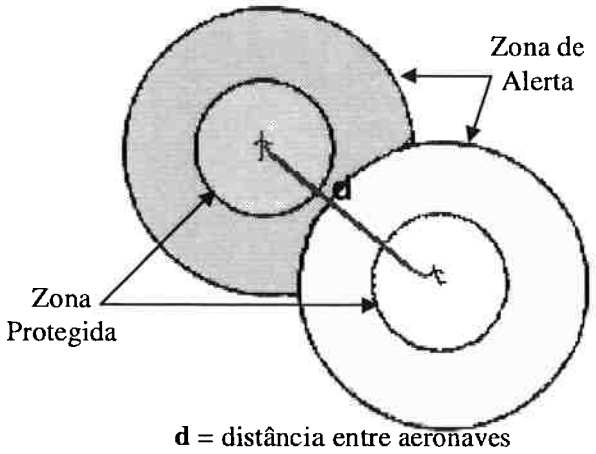


Figura 4.10 - Distância relativa entre aeronaves

Para a distância relativa entre aeronaves tem-se:

1) Se distância relativa entre duas aeronaves for maior que a distância de Zona de Alerta, a separação de segurança entre aeronaves está garantida.

2) Se distância relativa entre duas aeronaves for menor que a distância de Zona de Alerta, e maior que a Zona Protegida, assume-se um estado de alerta entre as duas aeronaves, conforme apresentado na figura 4.10.

3) Se distância relativa entre duas aeronaves for menor que a distância de Zona Protegida, assume-se um estado de conflito com elevado risco de colisão.

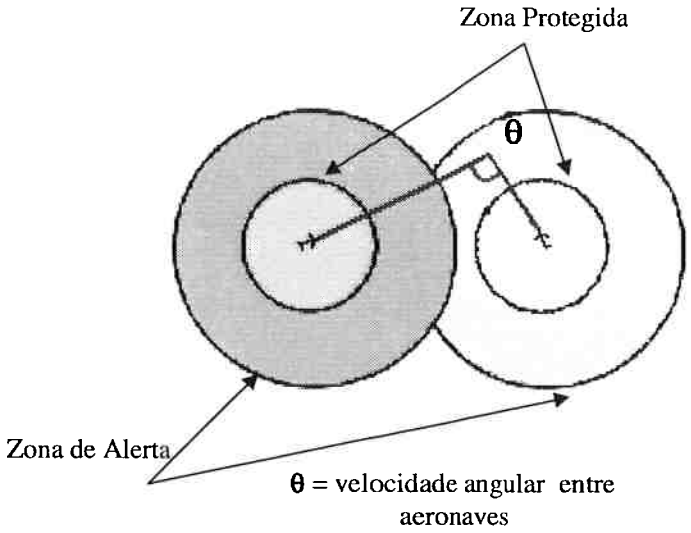


Figura 4.11 - Velocidade angular entre aeronaves

A velocidade angular é determinada no ponto de cruzamento entre as trajetórias futuras de duas aeronaves, ou seja, no ponto de colisão entre as aeronaves.

Um estado seguro de operação para uma aeronave implica que:

1) Não há presença de qualquer outra aeronaves em sua Zona de Alerta e Zona Protegida;e

2) A velocidade angular com qualquer aeronave em um dado espaço aéreo é igual a 0°.

As aeronaves militares apresentam maior dinâmica para realização de manobras mais bruscas e arrojadas. O mesmo não se pode afirmar para as aeronaves comerciais que apresentam maior movimento inercial e, por conseqüência, maior dificuldade em realizar manobras mais arriscadas. Além disso os passageiros de um aeronave comercial não estão preparados para tal tipo de manobra.

A proposta de *Roundabout* propõe que a manobra em um estado de alerta, seja realizada de forma suave, através de um movimento circular e velocidade constante, mantendo-se a Zona Protegida de cada aeronave, conforme mostrada na figura 4.12.

Observe que durante a manobra de Rounbabout as Zonas Protegidas das aeronaves não são invadidas por outras aeronaves, garantindo a segurança na resolução do conflito. Os

centros das circunferências executadas por ambas as aeronaves são iguais de forma a garantir que as Zonas Protegidas das aeronaves sejam invadidas.

A aeronave que estiver executando sua trajetória na circunferência mais interna irá sair de seu desvio, desde que não invada a Zona Protegida da aeronave que estiver executando a trajetória da circunferência mais externa. Se existir a possibilidade de invasão, a aeronave na trajetória da circunferência mais interna irá executar uma volta completa em seu desvio circular, até que possa novamente retomar à sua rota original.

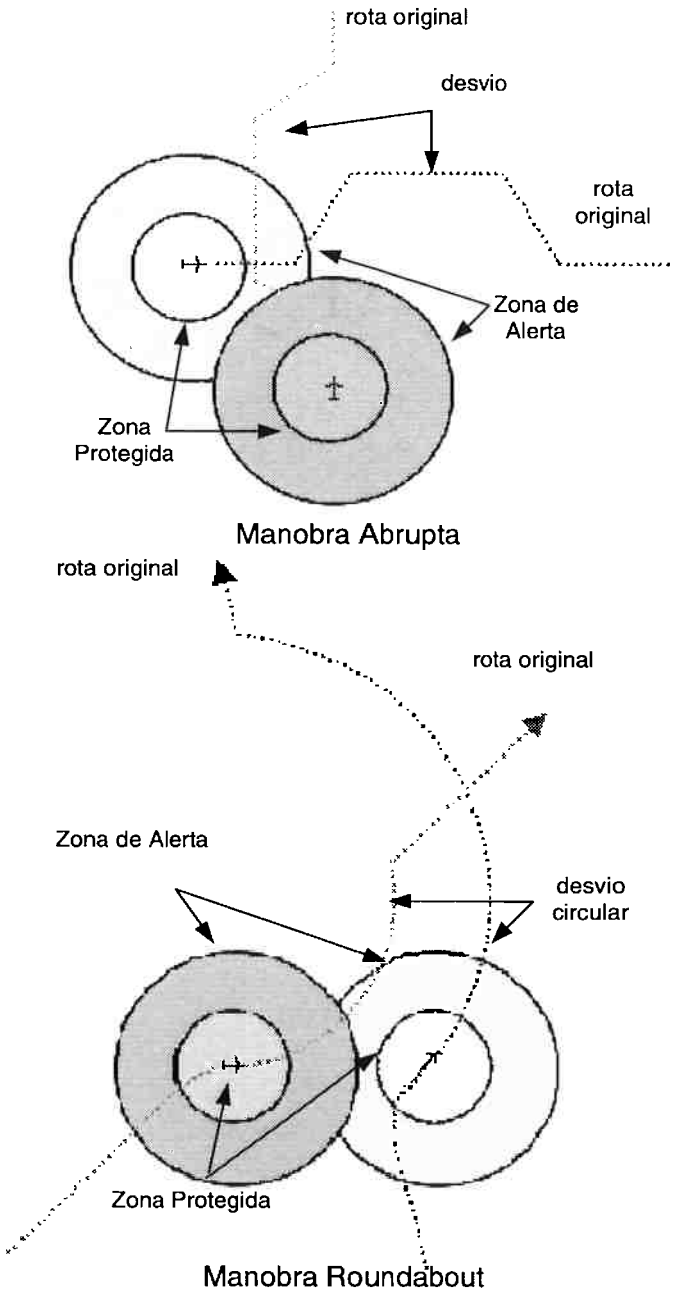


Figura 4.12 - Manobra Abrupta e Manobra Proposta no método *Rounbabout*

Em função da velocidade angular entre as aeronaves e a manobra de *Roundabout*, existem algumas situações de conflito analisadas a seguir.

Existem três casos em função da velocidade angular a serem analisados na existência de duas aeronaves com sobreposição das suas Zonas de Alerta ou Protegida:

1) Velocidade angular igual a 0°

Para velocidade angular igual a 0° ambas as aeronaves estão seguindo na mesma trajetória, ou muito próximas e paralelas no mesmo sentido, conforme apresentado na figura 4.13.

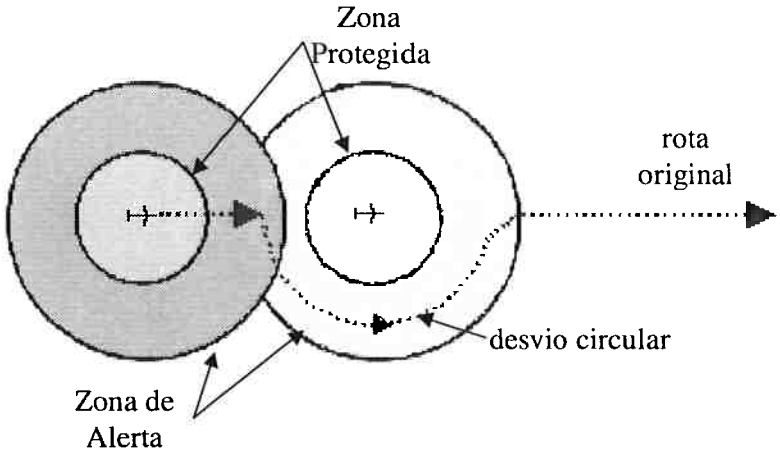


Figura 4.13 - Manobra de *Roundbout* com Velocidade Angular de 0°

Neste caso, a aeronave de maior velocidade desvia da aeronave de menor velocidade em uma manobra circular. Após a ultrapassagem retorna a sua rota original, preservando a Zona Protegida de ambas as aeronaves.

2) Velocidade angular igual a 180°

Para velocidade angular igual a 180°, ambas as aeronaves estão seguindo na mesma trajetória, ou muito próximas e paralelas, mas seguindo sentidos opostos, conforme apresentado na figura 4.14

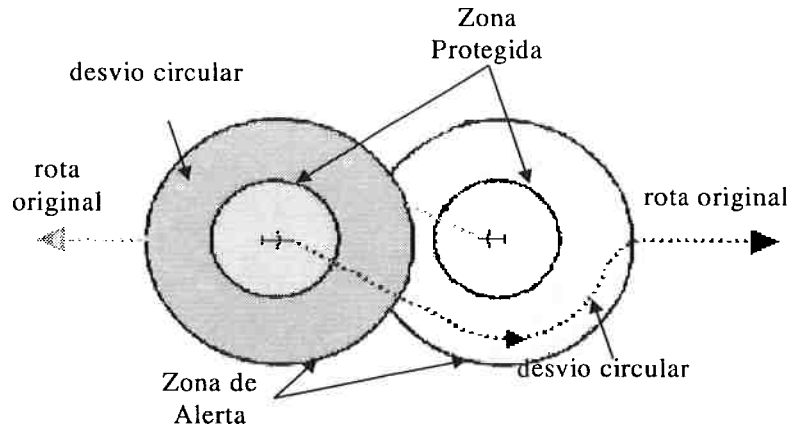


Figura 4.14 Manobra de *Roundbout* com Velocidade Angular de 180°

Como as aeronaves ocupam a mesma trajetória, mas em sentidos opostos, ambas as aeronaves devem realizar um desvio circular, para posteriormente retornarem às suas rotas originais, sempre preservando a Zona Protegida de ambas as aeronaves.

3) Velocidade angular diferente de 0° e 180°

Para velocidades angulares diferentes de 0° e 180°, a composição das manobras das aeronaves é similar à apresentada na figura 8, em que é definida a manobra de *Roundbout*.

Para que se possa identificar um conflito, além da distância entre as aeronaves, deve-se analisar a velocidade relativa entre as mesmas. A velocidade relativa permite identificar:

- Se as aeronaves estão se aproximando ou se afastando mutuamente; e
- O quão crítica pode ser uma aproximação mútua, ou seja, quanto maior a velocidade de aproximação, mais crítico é o estado de conflito e, portanto, mais acentuada deve ser a manobra de Roundbout.

A velocidade relativa entre aeronaves é calculada ponto-a-ponto, em função da variação da distância relativa entre as aeronaves, ou seja:

velocidade relativa = $\Delta d/\Delta t$

onde d = distância entre aeronaves e t = tempo percorrido

Se Δd for < 0, então as aeronaves estão se aproximando.

Se Δd for > 0, então as aeronaves estão se afastando.

A Modelagem para a Detecção e Resolução de Conflito baseia-se na alteração do módulo de TCAS apresentado na figura 4.6. Para tanto, este módulo é alterado de forma

a suportar a modelagem utilizando a lógica *fuzzy*. A figura 4.15 apresenta o módulo de TCAS adaptado para a lógica *fuzzy*:

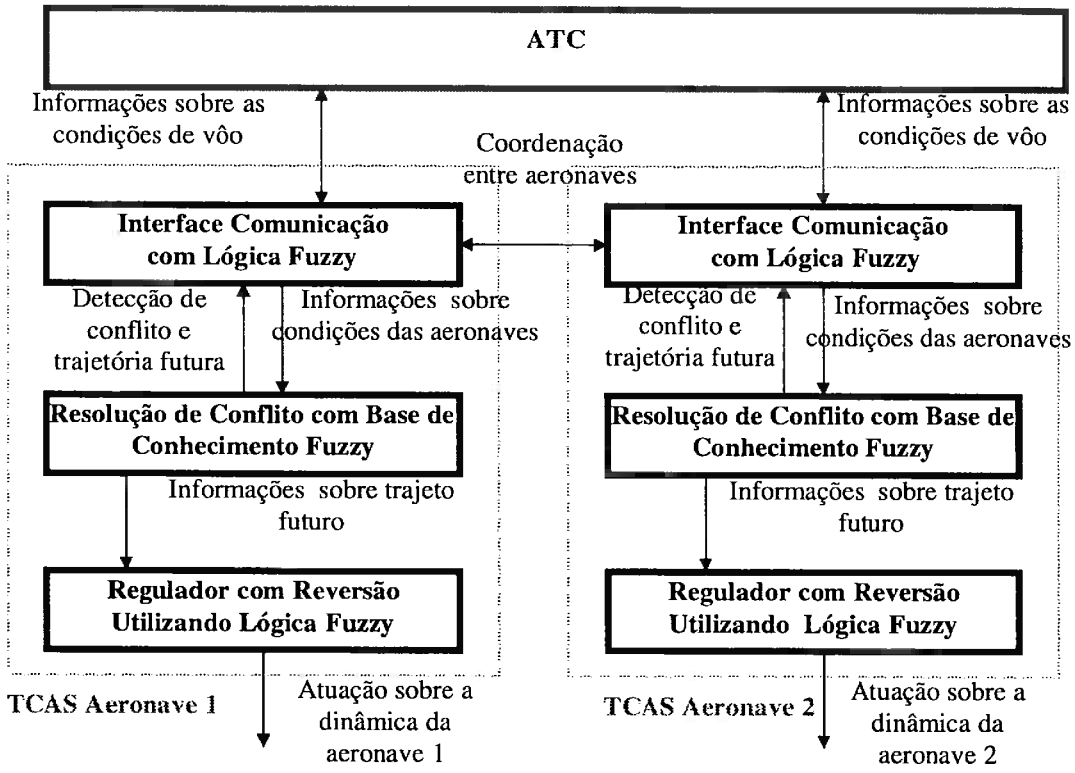


Figura 4.15 - Módulo do TCAS Utilizando Lógica *Fuzzy*

- Interface de Comunicação com Aplicação de Lógica *Fuzzy*: mapeia os valores lidos em um intervalo do universo [0,1];
- Resolução de Conflito e Base de Conhecimento *Fuzzy*: possui o modelo do TCAS, apresentando uma base de dados e uma base de regras lingüísticas *fuzzy*, que através de inferência *fuzzy*, tomam as decisões na Detecção e Resolução de Conflito.
- Regulador com Reversão Utilizando Lógica *Fuzzy*: produz uma ação de controle na aerodinâmica da aeronave (*crisp*) a partir da ação de inferência *fuzzy*.

Um importante modelo a ser adotado utilizando lógica *fuzzy* é a adaptação dos conceitos de Zona Protegida e Zona de Alerta. Utilizando a lógica *fuzzy* estes cilindros, referentes à Zona Protegida e à Zona de Alerta correspondem a cilindros *fuzzy*, ou seja, a maior probabilidade de encontrar a aeronave está no centro do Zona Protegida, diminuindo ao se aproximar das bordas deste cilindro. A mesma analogia vale para a Zona de Alerta.

Esta adaptação torna-se coerente, pois a Zona Protegida e Zona de Alerta são áreas de proteção da aeronave. Devido à incerteza da posição instantânea da aeronave, diversos fatores tais como a precisão de equipamentos, a presença de interferência de ruído

branco em radares, atrasos em transmissões via satélite e erros humanos interferem diretamente na localização exata da aeronave.

A figura 4.16 apresenta como seriam a Zona Protegida e a Zona de Alerta representadas através de cilindros *fuzzy*.

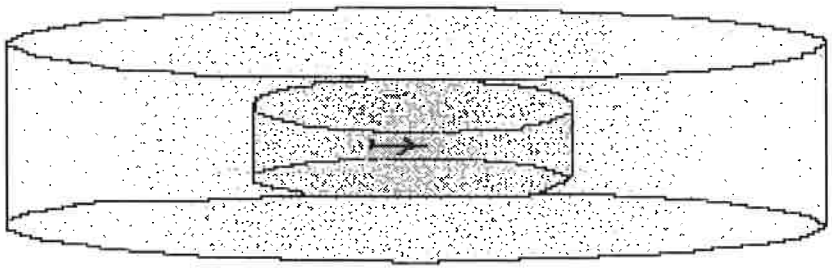


Figura 4.16 - Modelagem da Zona Protegida e Zona de Alerta pela lógica *fuzzy*

As variáveis lingüísticas de entrada são a distância e a velocidade relativa entre as aeronaves. A variável lingüística de saída é o ângulo de desvio da aeronave durante a manobra de desvio baseada na manobra de *Roundbaunt*.

A figura 4.17 apresenta os conjuntos *fuzzy* da variável lingüística **Distância entre Aeronaves (DA)**. Esta variável representa as distâncias entre as aeronaves segundo as Zona Protegida e de Alerta.

Para a variável lingüística: **Distância entre Aeronaves**, o universo de discurso varia de 0 a 60 milhas náuticas, tendo como valores lingüísticos: Crítico (C), Alerta Crítico (AC), Alerta (A) e Normal (N).

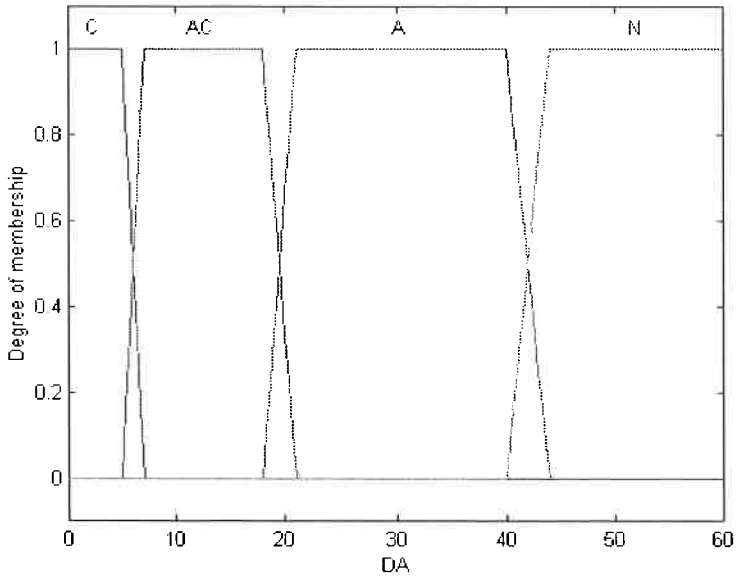


Figura 4.17 - Função de Pertinência da Variável lingüística **Distância entre Aeronaves (DA)**

A figura 4.17 apresenta os conjuntos *fuzzy* da variável lingüistica **Velocidade Relativa (VR)**. Esta variável representa a velocidade entre as aeronaves variando de -2400 Km/h a 90 Km/h (transporte comercial e civil).

Para a variável lingüística **Velocidade Relativa**, o universo de discurso varia de -2400 Km/h a 90 Km/h, tendo como valores lingüísticos: AProximação Lenta (APL), AProximação Rápida (APR) e Sem Aproximação (AS).

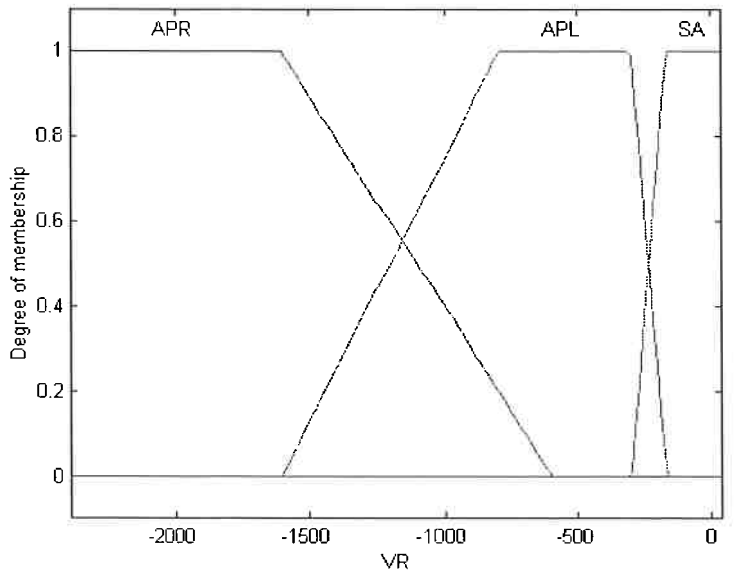


Figura 4.18 - Função de Pertinência da Variável Lingüística **Velocidade Relativa (VR)**

A velocidade angular entre as aeronaves é utilizada como uma informação adicional, permitindo determinar se:

- Ambas as aeronaves devem executar um desvio circular, quando a velocidade angular for diferente de 0°.
- Se apenas uma aeronave deve executar a um desvio circular, quando a velocidade angular for igual a 0°.

Desta forma, esta variável permite coordenar o tipo de manobra a ser executada em cada aeronave durante a resolução de conflito.

Outra variável de entrada foi utilizada para determinar, durante a manobra circular, em que ponto a aeronave deve retornar à sua trajetória original.

Foi utilizada uma única variável de saída. A figura 4.19 apresenta os conjuntos *fuzzy* da variável lingüística **Manobra de Desvio (MD)**. Esta variável representa o ângulo de desvio executado durante a manobra de *Roundbaut* variando de 0° a 90°.

Para a variável lingüística: **Manobra de Desvio**, o universo de discurso varia de -10° a 90°, tendo como valores lingüísticos: Sem Desvio (SD), Desvio Circular Lento (DCL), Desvio Circular Rápido (DCR) e Desvio Circular Muito Rápido (DCMR).

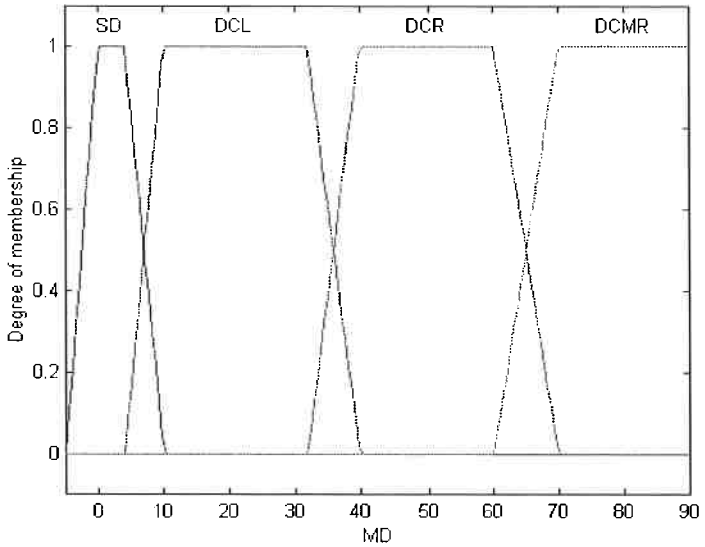


Figura 4.19 - Função de Pertinência da Variável Lingüística **Manobra de Desvio (MD)** de Roundbout

O relacionamento entre as diversas variáveis lingüísticas e seus respectivos valores estão resumidos na tabela 4.3. Este conjunto de variáveis lingüísticas foi extraída após o refinamento de um conjunto inicial de 48 regras. As regras excluídas apresentaram

pouca influência nas variável de saída durante as simulações, ou então correspondiam a situações que não ocorrem na prática.

ENTRADAS		SAÍDAS
DA	**VR**	**MD**
C	APL	DCR
C	APR	DCMR
C	AS	DCL
AC	APL	DCL
AC	APR	DCR
AC	AS	DCL
A	APL	DCL
A	APR	DCR
A	SA	DCL
N	SA	SD

Tabela 4.3 - Base de Dados com o Conhecimento Especialista

A partir da tabela 4.3 pode-se obter as seguintes regras de inferência no formato **IF - THEN**:

1. IF (DA is C) AND (VR is APL) THEN (MD is DCR)
2. IF (DA is C) AND (VR is APR) THEN (MD is DCMR)
3. IF (DA is C) AND (VR is SA) THEN (MD is DCL)
4. IF (DA is AC) AND (VR is APL) THEN (MD is DCL)
5. IF (DA is AC) AND (VR is APR) THEN (MD is DCR)
6. IF (DA is AC) AND (VR is SA) THEN (MD is DCL)
7. IF (DA is A) AND (VR is APL) THEN (MD is DCL)
8. IF (DA is C) AND (VR is APR) THEN (MD is DCR)
9. IF (DA is A) AND (VR is SA) THEN (MD is DCL)
10. IF (DA is N) AND (VR is SA) THEN (MD is SD)

As simulações foram realizadas utilizando-se a ferramenta MATHLAB versão 5.3. A figura 4.20 apresenta o sistema de inferência *fuzzy* obtido a partir desta ferramenta.

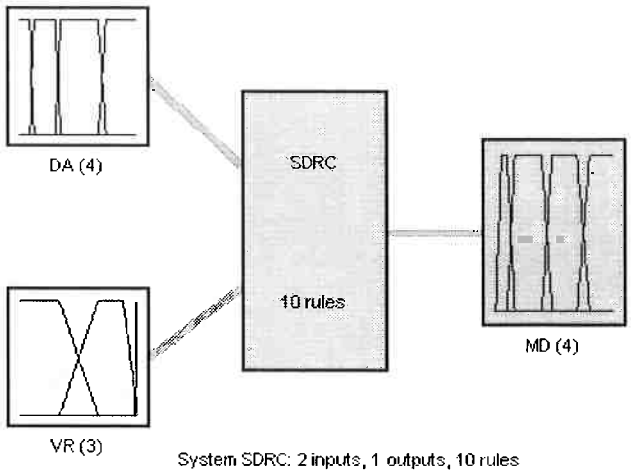


Figura 4.20 - Sistema de Inferência *Fuzzy* a Partir da Ferramenta MATHLAB.

No primeiro ciclo de simulação executa-se a transformação para lógica *fuzzy*, onde é realizada a classificação das variáveis de entrada nas variáveis lingüisticas. Supondo-se as seguintes variáveis de entrada:

Entradas:

Distância entre Aeronaves = 6 milhas náuticas ⇒ Crítico com grau de pertinência igual a 0.5, e Alerta Crítico com grau de pertinência igual a 0.5.

Velocidade Relativa = -1100 Km/h ⇒ Aproximação Lenta com grau de pertinência igual a 0.5, e Aproximação Rápida com grau de pertinência igual a 0.5.

Após realizar a classificação, verifica-se quais as regras na base de conhecimento são aplicáveis, podendo-se obter as seguintes regras selecionadas:

1. IF (DA is C) AND (VR is APL) THEN (MD is DCR)
2. IF (DA is C) AND (VR is APR) THEN (MD is DCMR)
4. IF (DA is AC) AND (VR is APL) THEN (MD is DCL)
5. IF (DA is AC) AND (VR is APR) THEN (MD is DCR)

Para os antecedentes de cada regra relacionados pelo conectivo **AND,** aplica-se o operador interseção (operador mínimo), no qual é obtido o valor mínimo entre seus antecedentes. A ação final de controle é obtida a partir da união (operador máximo) das contribuições de cada regra. Como saída do sistema é realizada a tem-se o ângulo de Manobra de Desvio na forma fuzzy. Um valor numérico (*crisp*) pode ser obtido utilizando-se o Método do Centro de Área (CDA) como método de *defuzzificação*, gerando-se para a variável Manobra de Desvio o valor:

A saída obtida especificou uma Manobra de Desvio correspondente a 48°.

A figura 4.21 apresenta o resultado de simulação a partir da ferramenta MATHLAB.

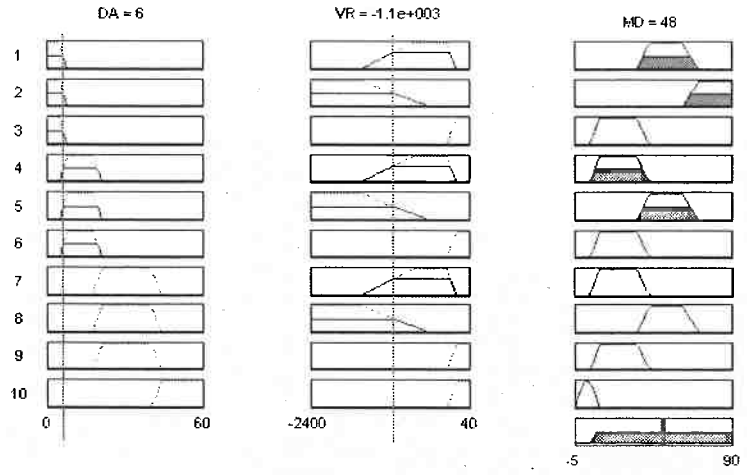


Figura 4.21 - Aplicação das Regras na Ferramenta MATHLAB para o Exemplo Apresentado

A tecnologia CNS/ATM está sendo implantada em todo o mundo, e uma das principais aplicações está relacionada com a Detecção e Resolução de Conflitos entre Aeronaves, fundamental para a segurança do tráfego aéreo. Este tipo de aplicação apresenta características de uma aplicação *fuzzy*, tais como: a aplicação depender da habilidade de um especialista (piloto da aeronave e operador de vôo no ATC); a existência de um certo grau de imprecisão na localização da aeronave; e o fato do controle apresentar como principal foco a sua forma de controle, mais do que a sua precisão.

No modelo simulado pode-se observar que as variáveis de entrada apresentaram grande coerência na mudança da variável de saída, ou seja, refletem de forma aderente o raciocínio do especialista (piloto de uma aeronave) na ação da Detecção e Resolução de Conflito entre Aeronaves.

A simulação do processo de Detecção e Resolução de Conflito entre Aeronaves modelado aplicou-se, principalmente, na fase de aproximação e Manobra de Desvio entre aeronaves. A velocidade angular e a posição de retorno para a rota original durante a manobra de *Roundbout* foram tratadas como variáveis auxiliares. Estas variáveis poderiam também ser modelado pela metodologia *fuzzy*, mas a modelagem tornaria-se mais complexa, retirando a simplicidade do modelagem da lógica *fuzzy* apresentada, além de que seriam necessários maiores informações de bordo das aeronaves, bem como o plano de viajem das mesmas.

Para se manter os mesmos níveis de segurança relacionados com um maior período de atualização do posicionamento das aeronaves, deve-se aumentar as larguras dos

patamares relativos aos valores lingüísticos da distância entre as aeronaves, ou seja: Crítico, Alerta Crítico, Alerta e Normal. Evidentemente, esta mudança irá afetar a determinação da trajetória de manobra de desvio de colisão entre aeronaves.

4.2.4 Conclusões

O estudo da segurança do sistema CNS/ATM é um projeto de pesquisa bastante recente. Pode-se afirmar que no momento a equipe do GAS encontra-se em processo de entendimento da aplicação aviação. Neste sentido os dois trabalhos preliminares realizados estão auxiliando bastante numa maior compreensão das funcionalidades deste novo sistema, além de começar a fornecer subsídios para uma Análise de Perigo. Como nesta aplicação o conhecimento aplicado ainda é bastante primário, torna-se mais difícil a aplicação da metodologia de análise de risco proposta. Neste sentido, a aplicação de uma metodologia de análise de risco é extremamente mais complexa quanto menor for o conhecimento profundo da aplicação. Vale destacar com isso que os trabalhos iniciais sendo realizados podem ser considerados como fundamentais no processo de aprendizado das funcionalidades do sistema de controle de tráfego aéreo.

5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

O Estabelecimento de uma Metodologia de Análise de Risco para sistemas críticos é de fundamental importância no sentido de se atingir melhores níveis de integridade de segurança, em especial quando se trata de sistemas computacionais. Antes mesmo de se escolherem quais métodos/técnicas devem ser aplicados, a metodologia com que um trabalho desse tipo deva ser conduzido tem uma importância fundamental. Caso não haja uma visão ampla do conceito de segurança ("safety") provocando uma saudável discussão tanto a nível gerencial como a nível puramente técnico, pode-se atingir resultados completamente insatisfatórios, onde pesquisas são realizadas em áreas específicas mas não se integram a um objetivo comum.

Esse foi o objetivo desta tese de Livre Docência, prover uma visão de uma Metodologia de Análise de Risco e integrá-la a métodos/técnicas já existentes, além de outros em plena pesquisa. Desta forma, se aprofundou-se em alguns métodos sem, no entanto, deixar de apresentar sua integração ao longo da aplicação da metodologia de análise de risco.

Outro aspecto importante a se destacar refere-se às grandes discussões envolvendo terminologia. Neste trabalho procurou-se adotar uma padronização, sem com isso afirmar que esta seja a melhor. Sem esse trabalho de padronização, que na realidade foi se realizando ao longo da apresentação desta pesquisa, dificilmente se chegaria a um método de análise de risco claro e objetivo.

Já com relação aos sistemas computacionais em particular, as dificuldades em se estabelecer valores quantitativos com relação aos Níveis de Segurança são enormes. Vale ressaltar que, conforme exposto ao longo deste trabalho, a avaliação qualitativa desempenha um papel fundamental nestes sistemas, em especial ao próprio desenvolvimento do software. No entanto, não se deve renegar, nem desprezar qualquer esforço no sentido de se alcançar métodos de avaliação quantitativos mais eficazes. Estes métodos podem e devem ser bastante úteis na avaliação comparativa entre níveis de segurança de sistemas diferentes e não tanto como indicadores de valores absolutos de segurança.

Outro aspecto fundamental que se pode concluir é que a relação entre os meios acadêmicos e os meios industriais é de fundamental importância nestes trabalhos de pesquisa de verificação da segurança. Sem a colocação clara por parte da indústria de suas necessidades fica mais complexo o direcionamento das pesquisas, no sentido de se

obterem melhores resultados para a sociedade. Vale ressaltar que devido a análise de risco se tratar de um tema de pesquisa bastante "árido e abstrato", observa-se claramente a necessidade primordial de esclarecimentos técnico-acadêmicos para a comunidade industrial, no sentido de se deixarem bastante claros os conceitos, os métodos e os objetivos a serem alcançados. Por outro lado é fundamental o conhecimento da aplicação por parte dos pesquisadores acadêmicos, caso contrário a eficácia do trabalho de análise pode ficar comprometido.

Enfim, todo esse trabalho de pesquisa forma a base para a constituição de uma Cultura de Segurança, sem a qual será praticamente impossível a melhoria dos níveis de integridade de segurança.

Em função do avanço crescente da tecnologia, em especial dos sistemas computacionais com aplicação crítica quanto à segurança, é extremamente importante e fundamental a contínua pesquisa no sentido de se melhor aprimorar as metodologias de análise de risco, além de uma pesquisa bastante profunda em métodos/técnicas de apoio, em especial aqueles métodos aplicáveis a nível de desenvolvimentos de projetos de software. Vale dizer que os próprios métodos/técnicas já existentes devem continuar a serem aprimorados visando uma melhor adequação às novas tecnologias.

Nesse sentido algumas linhas de pesquisa que merecem maior destaque são apontadas a seguir.

Na avaliação do ser humano em sistemas críticos

Estudo de modelos e métodos visando uma melhorar integração do ser humano nos sistemas críticos, além de modelos de avaliação do desempenho humano no que diz respeito à segurança. Trata-se de pesquisas que envolvem outras áreas do conhecimento, desde a ergonomia até a psicologia.

Na elaboração de Listas de Inspeção

Deve-se aprimorar os padrões a serem adotados nos diversos níveis do projeto, desde a especificação até a implementação propriamente dita.

Método HAZOP em projetos de software

Pesquisa no aprimoramento do método HAZOP quando aplicado a módulos de software, visando uma maior clareza e eficácia em sua utilização.

Método de Injeção de Falhas

Estudo de formas de utilização desse método no sentido de auxiliar na determinação de taxas de falhas inseguras de sistemas microprocessados, levando em consideração o tipo de programação defensiva utilizado.

Métodos de Avaliação da Segurança

Maior pesquisa em meios de integração dos métodos de Avaliação Qualitativa e dos métodos de Avaliação Quantitativa da segurança, tornando, desta forma, mais objetivas as recomendações de projeto e prováveis reprojetos, em especial quando se trata de sistemas computacionais. Outro campo importante de pesquisa está no aprimoramento de métodos de simulação com o intuito de se obter resultados mais eficazes no trabalho de Análise de Risco. Podem ser citados aqui métodos que se utilizam de lógica Fuzzy, Redes Neurais, Algoritmos Genéticos, Autômatos Híbridos, Statecharts, entre outros.

Métodos formais

O progresso nesta área depende fortemente de pesquisas de novos métodos e ferramentas, da integração de diferentes métodos, e da transferência efetiva de tecnologia para a indústria.

No campo de Conceitos Fundamentais

Nesta área diversas pesquisas devem ser feitas contemplando a composição de conceitos (como combinar métodos, especificações, modelos, matemática discreta e contínua, e provas), a decomposição de conceitos (métodos para decompor uma propriedade global em propriedades locais, cuja verificação é mais simples), a abstração de conceitos (identificar diferentes tipos de abstração para se adequar às características dos sistemas reais), a teoria para modelos reusáveis (maior eficiência no seu uso) e algoritmos e estrutura de dados mais eficientes e concisos.

Métodos e Ferramentas

As pesquisas sob este enfoque devem contemplar métodos e ferramentas cuja aplicação dêem um retorno mais rápido, que apresentam um ganho e esforço incremental, que possa ser aplicada ao longo do ciclo de vida do sistema, facilmente integrada com demais métodos/ferramentas (por exemplo, compiladores, simuladores, etc...), de fácil utilização e aprendizado, orientada para detectar erros, análise focada para a segurança e permitir um desenvolvimento evolucionário em diversos níveis do sistema

Integração dos Métodos

Dado que nenhum método é adequado para descrever e analisar todos os aspectos de um sistema complexo, uma saída interessante é a utilização da combinação de diferentes métodos. Neste sentido, um dos futuros promissores é a integração do Model Checking com a Prova de Teorema. Pode-se, por exemplo, utilizar o Model Checking como um procedimento de decisão sendo que a Prova de Teorema pode ser utilizadas na verificação das propriedades temporais em espaços infinitos, através da técnica de indução.

Outro futuro promissor é a integração com o processo de desenvolvimento de sistema. Os métodos formais podem complementar os métodos menos formais ao longo do processo de desenvolvimento, em especial na especificação e verificação.

Educação e Transferência Tecnológica:

O Processo educacional é fundamental para o sucesso dos métodos formais. Educação no âmbito de especialização/pós-graduação como também em cursos de graduação. Ainda neste aspecto vale ressaltar também a importância de uma melhor comunicação entre o mundo acadêmico e industrial, visando uma efetiva transferência de conhecimento da academia para a indústria, como também uma melhor conhecimento das novas tecnologias por parte do mundo acadêmico permitindo, desta forma, um maior direcionamento nas pesquisas aplicadas.

Concluindo este trabalho, vale reforçar que o retorno para a sociedade, no sentido de se obterem sistemas com maiores níveis de segurança estará fortemente dependente de uma cultura de segurança. Esta cultura de segurança estará dependente de diversos fatores, sociais, políticos, gerenciais e técnicos. Este trabalho teve seu grande enfoque nos aspectos técnicos, em especial sobre uma Metodologia de Análise de Risco e alguns de seus métodos. A inovação tecnológica cada vez maior dos novos sistemas, em especial atenção, os sistemas metroviários e aeroviários, irá exigir cada vez mais um maior aprofundamento de pesquisas direcionadas para a segurança. Que esta necessidade motive um maior número de pesquisadores a desenvolverem trabalhos nesta importante área do conhecimento.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ARTIGOS

[Accurso 99] Accurso, A.; Timóteo, C. A. F.; Freitas, J. H. Z.; Borloni, R. N. Operar com Segurança. Pgs 94-98. Revista Engenharia. Operação Metrô de São Paulo. 1999.

[Alur] Alur, R. Henzinger, T.A., Ho, P.H. Automatic Symbolic Verification of Embedded Systems. 37 pgs. Bell Labs, Lucent Technologies, Department of Computer Science, Cornell University, University of California at Berkley.

[Atlee], Atlee, J. ChechiK, Gannon, J. Using Model Checking to Analyze Requirements and Designs. 23 pgs. University of Maryland, USA, and University of Waterloo, Canada.

[Bartolomeu 2000] Bartolomeu, C. CNS/ATM no Mundo. Pgs28-36. Revista Aeroespaço. Agosto 2000. Edição Especial. CNS/ATM

[Bastt 98] Bastt,W.;Bock,H.W; German Qualification and Assessment of Digital IEC Systems important to safety; Reliability Engineering and System Safety, 59, 1998, 163-170. Elsevier Science Limited

[Beerthuizen 01] Beerthuizen, P. G.; Kruidhof, W. System and software safety analysis for the ERA control computer. Reliability Engineering and System Safety, 71, 2001, 285-297. Elsevier Science Limited

[Boeing 97] Air Traffic Managment Concept Baseline Definition. Prepared by Boeing Commercial Airplane Group. Nextor Report #RR-97-3. October 31, 1997.

[Bohnenblust 98] Bohnenblust, H. Slovic, P. Integrating technical analysis and public values in risk-baesd decision making. Reliability Engineering and System Safety 59. 1998. Pgs 151-159. Elsevier Science Limited.

[Bonifácio 99] Bonifácio, A.L.; Moura, A. M.; Camargo Jr., J.B.; Almeida Jr.,J.R. Análise, Verificação e Síntese de Segmentos de Via de uma Malha Metroviária.

Relatório Técnico IC-99-45. 48 páginas. Agosto 1999. Instituto de Computação. UNICAMP.

[Bowen 96] Bowen, J.P.; Butler, R.W.; Dill,D.; Glass, R. L.; Gries, D.; Hall, A.; Hinchey, M.G.; Holloway, C.M.; Jackson, D.; Jones, C.; Lutz, M.J.; Parnas, D.L.; Rushby, J.; Wing, J.; Zave, P. An Invitation to Formal Methods. IEEE Computer. Pgs 16-30. April 1996.

[Broomfield 97] Broomfield, E.J.; Chung, P.W.H. Safety assessment and the software requirements specification. Reliability Engineering and System Safety 55. 1997. Pgs 295-309. Elsevier Science Limited.

[Bonifácio 99]Bonifácio, A.L; Moura, A.V; Camargo Jr.,J.B; Almeida Jr.,J.R. Análise e Verificação de Segmentos de Via de uma Malha Metroviária. II Workshop on Formal Methods, pags 13-22, realizado de 12 a 13 de outubro de 1999 em Florianópolis, SC, Brasil.

[Butler 93] BUTLER, R. W.;FINELLI, G. B. The infeasibility of quantifying the reliability of life-critical real-time software. IEEE Transactions on Software Engineering, v.19, n.1, p.3-12, Jan. 1993.

[CANSO 99] Demystifying CNS/ATM. CANSO CNS/ATM WORKING GROUP. Final Version June 1999.

[Camargo 01] Camargo Jr., J.B., Canzian, E., Almeida Jr., J.R., Paz, S.M., Basseto, B.A., Quantitative Analysis Methodology in Safety-Critical Microprocessor Applications, Reliability Engineering & System Safety 74 – 2001. Pgs 53-62. Elsevier Science Limited .

[Camargo 99]Camargo JR.,J.B; Almeida JR., J. R. Applying HAZOP to a Subway Signaling System. Artigo aceito para ser publicado na Conferência "ISSC'99 – 17th International System Safety Conference – System Safety at the Dawn of a New Millenium", a ser realizada em Orlando, Florida, USA, de 16 a 21 de agosto de 1999.

[Clarke] Clarke, E.M.; Wing, J.M. Formal Methods: State of the Art and Future Directions. 22 pgs. Carnegie Mellon University.

[Eames 99] Eames, P. D.; Moffett, J. The Integration of Safety and Security Requirements. In: Safecomp 1999, Toulouse, France, September 1999.

[Garrett 99] Garrett, C.; Apostolakis, G. Context in the Risk Assessment of Digital Systems; Risk Analysis, vol 19, No.1, 1999.

[Hamlet 92] Hamlet,D. Are we testing for true reliability?. IEEE Software, v.9, n.4, p.21-7, July 1992.

[Haxthausen 00] Hazthausen, A.E.; Peleska,J. Formal Development and Verification of a Distributed Railway Control System. IEEE Transaction on Software Engineering, Vol 26, NO 8, pgs 687-701, August 2000.

[Hebbron 97] Hebbron, B.D.; Fenelon, P. The applicationof hazard and operability studies to real time structured requirements models. Reliability Engineering and System Safety 55. 1997. Pgs 311-325. Elsevier Science Limited.

[Henzinger] Henziger, T.A; Ho, P.H.; Wong-Toi, H. HyTech: A Model Checker for Hybrid Systems. 23 pgs. Research suppoerted by ONR YIP, NSF CAREER, AFOSR, ARO MURI, ARPA and SRC.

[Henzinger 1] Henzinger, T.A. The Theory of Hybrid Automata. 24 pgs. Electrical Engineering and Computer Sciences. University of California at Berkeley.

[Houtermans 98]Houtermans, M.J.M.;Rouvroye,J.L.; The Influence of Design Parameters on the Performance of safety -–Related Systems; Eindhoven University of Technology – TUV Product Services Inc., 1998.

[Iyer 85] Iyer, R. K.; Velardi, P. Hardware-Related Software Errors: Measurement and Analysis. IEEE Transactions on Software Engineering, Vol SE-11, No.2, February 1985.

[Jaffe 91] Jaffe,M. S.;Leveson,N. G.;Heimdahl,M. P. E. M.;Bonnie E. Software requirements analysis for real time process control systems. IEEE Transactions on Software Engineering, v.17, n.3, p.241-58, Mar. 1991.

[Ladkin 01] Ladkin, P.B. Na Example of Everyday Risk Assessment. Faculty of Technology. University of Bielefeld. 12 pgs.Februaray 2001.

[Laprie 90] Laprie, Jean C.; Arlat, J.; Beouner,C.; Anoun, K.; Definition and Analysis of Hardware and Software Fault Tolerant Architectures; LAAS – CNRS, Computer – July 1990 – 39-51.

[Lawrence 97] Lawrence, J.D.; Gallagher, J.M. A proposal for performing software safety hazard analysis. Reliability Engineering and System Safety 55. 1997. Pg 267-282. 1997 Elsiver Science Limited.

[Lawrence 00] Lawrence, J.D. Software qulification in safety applications. Reliability Engineering and System Safety 70. 2000. Pg 167-184. 2000 Elsiver Science Limited.

[Leveson 90]Leveson, N.G.;Cha,S.S.; Knight,J.C.; Shimeall,T.J.; The Use of Self Checks and Voting in Software Error Detection: An Empirical Study; IEEE Transactions on Software Engineering, Vol 16., No 4, April 1990.

[Leveson 86] Leveson,N. G. Software safety:why,what, and how. Computing Surveys, v.18, n.2, p.25-163, June 1986.

[Leveson 83] Leveson,N. G.;Harvey,P. Analyzing software safety. IEEE Transactions on Software Engineering, v.9, n.5, p.569-79, Sept. 1983.

[Machado 2000] Machado, W.C.C. O GNSS Transitório Brasileiro. Pgs14-22. Revista Aeroespaço. Agosto 2000. Edição Especial. CNS/ATM

[Melchers 01] Melchers, R.E. On the ALARP approach to risk managment. Reliability Engineering and System Safety 71, 2001, pgs 201-208. Elsevier Science Limited.

[Moura 00] MOURA,A.; BONIFÁCIO,A.; CAMARGO Jr.,J.B.; RADY Jr..J.R. Formal Parameter Synthesis for Track Segments of a Subway Mesh. 7th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, a ser realizada de 3 a 7 de abril de 2000 em Napier University, Edinburh, Scotland, UK.

[Naufal 01] NAUFAL JR, J.K.; CAMARGO JR, J.B.; ALMEIDA JR., J.R.; CUGNASCA, P.S. Avaliação da Influência da Precisão do Posicionamento das Aeronaves nos Níveis de Segurança do Controle de Tráfego Aéreo. In: Congresso SAE BRASIL, São Paulo – SP, Brasil, 19/11 a 22/11/2001. Artigo n. 230.

[Rady 00] RADY A.,J.R.; CAMARGO Jr.,J.B; BASSETO, B.A.; CUNHA, R.S.; PAZ,S.M. An Inspection List for Critical Software Analysis. In: PSAM 5 -International Conference on Probabilistic Safety Assessment and Management, Osaka, Japan, de 27/11 a 01/12/2000.

[Redmill 97] Redmill, F.; Chudleigh, M.F.; Catmur, J.R. Principles underlying a guideline for applying HAZOP to programmable electronic systems. Reliability Engineering and System Safety 55. 1997. Pgs 283-293. Elsevier Science Limited.

[Renn 98] Renn, O. The role of risk perception for risk managment. Reliability Engineering and System Safety 59. 1998. Pgs 49-62. Elsevier Science Limited

[Rushby 94] RUSHBY,J. Critical system properties: survey and taxonomy. Reliability Engineering & System Safety, v.43, n.2, p.189-219, aug. 1994.

[Sheldon92]Sheldon,F. T.;Kavi,K. M.;Tausworthe,R.C.;Yu,J.;Brettschneider,R.;Everett, W. W. Reliability measurement: from theory to practice. IEEE Software, v.9, n.4, p.13-20, July 1992.

[Schneidewind 97] Schneidewind, N. F. Reliability Modeling for Safety-Critical Software, IEEE Transaction on Reliability, Vol.46, NO.1, 1997 March.

[Yu 98] Yu,W.D. A Software Fault Prevention Approach in Coding and Root Cause Analysis. Bell Labs Technical Journal. Pg 3-21.April-June 1998.

[Wright 94] Wright, D.; Cai, K. Representing Uncertainty for Safety Critical Systems. Draft PDCS2 Project Deliverable, City University, Northampton Square, London EC1V 0HB, May 5, 1994.

LIVROS

[Friedman 95] Friedman, M. A; Voas, J. M.; Software Assessment: Reliability, Safety, Testability. John Wiley & Sons, Inc. 1995.

[Galotti 99] Galotti Jr., V.P. The Future Air Navigation System (FANS). Ashgate. ISBN 0 291 39833 2. 1999.

[Hall 97] Hall, E.M. Managing Risk – Methods for Software Systems Development. SEI Series in Software Engineering. Addison-Wesley. ISBN 0-201-25592-8.1997.

[Hurn 89] Hurn, J. GPS – A Guide to the Next Utility. Trimble. 1989.

[Hurn 93] Hurn, J. GPS – Differential GPS. Trimble. 1993.

[Johnson 89] Johnson, B. W. Design and Analysis of Fault Toerant Digital Systems. University of Virginia, Addison- Wesley Publishing Company. 1989.

[Kececioglu 91] Kececioglu, D. Reliability Engineering Handbook, Volume 2. Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0 13 772302 4, 1991.

[Leveson 95] Leveson, N. G. Safeware – System Safety and Computers. University of Washington. Addison-Wesley Publishing Company. 1995.

[Myerson 96] Myerson, M. Risk Managment Processes for Software Engineering Models. Artech House. ISBN 0-89006-653-3.

[Profit 95] Profit, R. Systematic Safety Managment in the Air Traffic Services. Euromoney Publications. ISBN 1 85564 470 3. 1995.

[Redmill 99] Redmill, F.; Chudleigh, M.; Catmur, J. System Safety: HAZOP and Software HAZOP, John Wiley & Sons, ISBN 0 471 98280 6, 1999.

[Storey 96] Storey, N. Safety-Critical Computer Systems. Assison-Wesley Publishing Company. 1996.

[Voas 98] Voas, J.M. Software Fault Injection. John Wiley & Sons, ISBN 0-471-18381-4. 1998.

[Walters 00] Walters, J.M.; Sumwalt, R.L. Aircraft Accident Analysis: Final Reports. 2000. Mc Graw Hill. ISBN 0-07-135149-3.

NORMAS

[DD ENV 50129:1999] Railway applications – The specification and demonstration of Reliability, Availability, maintainability and Safety (RAMS). EN 50126. CENELEC - Eurpopean Committee for Electrotechnical Standardization. September 1999.

[DD ENV 50129:1999] Railway applications Safety related electronic systems for signalling. – European Presatandard ENV 50129. CENELEC – Eurpopean Committee for Electrotechnical Standardization. May 1998

[EN 50128]Railway applications – Software for railway control and protection systems. CENELEC – Eurpopean Committee for Electrotechnical Standardization. July 1998.

[IEC 61508] Functional Safety Electrical/Electronic/Programmable Eletronic Safey-elated Systems. International Electrotechnical Commission. IEC 61508. 1997.

[Military Handbook] Military Handbook – Reliability Prediction of Electronic Equipament -Department of Defense, Washington DC, 1990.

[NASA 96] NASA Software Safety Standard, NSS. 1740.13. 1996.

[NBR 9126] Tecnologia de Informação – Avaliação de Produto de Software – Características de Qualidade e Diretrizes para o seu Uso. 1994.

APÊNDICE A

Glossário dos termos utilizados nos Requisitos Gerais de Segurança

Neste item é feita uma descrição dos termos utilizados nos requisitos gerais de segurança, apresentados no item seguinte.

ALINHAMENTO DE ROTA – processo de definição uma nova rota, com a posterior abertura de um bloqueio, e a conseqüente liberação do acesso a uma região de AMVs

AMV (APARELHO DE MUDANÇA DE VIA) – dispositivo mecânico utilizado para a transferência de um trem de uma via a outra

BLOQUEIO – entidade lógica que permite ou inibe a entrada e/ou a saída de um trem em ou de uma região de AMVs

Pode ser:

- de entrada, quando controla a entrada de um trem em uma região de AMVs
- de saída, quando controla a saída de um trem de uma região de AMVs

Pode estar:

- aberto para rota normal, quando permite rota normal
- aberto para rota de chamada, quando permite rota de chamada
- fechado, quando não permite nenhuma rota

Pode estar:

- proibido como entrada, quando não pode ser usado como bloqueio de entrada
- proibido como saída, quando não pode ser usado como bloqueio de saída
- não proibido como entrada, quando pode ser usado como bloqueio de entrada
- não proibido como saída, quando pode ser usado como bloqueio de saída

CANCELAMENTO DE ROTA ALINHADA – processo para desfazer o alinhamento de uma rota, com o posterior fechamento do bloqueio previamente estabelecido como de entrada da rota

CIRCUITO DE VIA – segmento de via onde é possível a detecção da presença ou da ausência de um trem

Pode estar:

- ocupado, quando um trem é detectado
- desocupado, quando um trem não é detectado

CÓDIGO DE VELOCIDADE DE CIRCUITO DE VIA – representação da velocidade máxima permitida para um trem percorrer um circuito de via

DESOCUPAÇÃO SEQÜENCIAL DE CIRCUITOS DE VIA – seqüência de ocupações e desocupações de circuitos de via, que evidencia a passagem de um trem por uma rota

DESTRAVAMENTO DE MÁQUINA DE CHAVE – processo de liberação da movimentação de uma máquina de chave

MÁQUINA DE CHAVE – equipamento de via que permite a movimentação de um AMV para a mudança da direção do movimento do trem

Pode estar:

- normal, travada, quando não altera a direção do movimento original do trem, e não pode ser movimentada
- normal, destravada, quando não altera a direção do movimento original do trem, mas pode ser movimentada
- em reverso, travada, quando altera a direção do movimento original do trem, e não pode ser movimentada
- em reverso, destravada, quando altera a direção do movimento original do trem, mas pode ser movimentada
- fora de correspondência, em qualquer outra situação

MODO OPERACIONAL DE REGIÃO DE AMVS – modo pelo qual uma região de AMVs pode ser controlada pelo sistema de sinalização

Pode ser:

- central
- local
- automático
- em manutenção

PERFIL SEGURO DE VELOCIDADE DE VIA – seqüência de códigos de velocidade ao longo de uma via, que garante a movimentação segura dos trens

REGIÃO DE AMVS – conjunto de AMVs, máquinas de chave, circuitos de via e bloqueios associados, que são utilizados para o alinhamento de uma rota

ROTA – seqüência de circuitos de via dentro de uma região de AMVs, compreendidos entre um bloqueio de entrada e um bloqueio de saída, a serem percorridos por um trem

Pode ser:

- normal, quando utilizada para atravessar uma região de AMVs
- de chamada, quando utilizada em condições especiais (por exemplo, em manobras de engate e desengate, ou em ocorrência de falsa ocupação, ou ainda quando há tráfego em sentido conflitante)

Pode estar:

- alinhada, quando um alinhamento solicitado já foi efetivado
- em alinhamento, quando um alinhamento solicitado ainda não foi efetivado

ROTA CONFLITANTE – rota que exige que pelo menos uma máquina de chave assuma posição diferente da posição exigida por outra rota já alinhada ou em alinhamento

ROTA OPOSTA – rota com sentido de tráfego oposto ao sentido de tráfego de outra rota já alinhada ou em alinhamento

SENTIDO DE TRÁFEGO DE ROTA – sentido do tráfego de trens, estabelecido nos circuitos de via de uma região de AMVs, e definido pelos bloqueios de entrada e de saída de uma rota

SENTIDO DE TRÁFEGO DE TRECHO DE VIA – sentido do tráfego de trens, estabelecido para um trecho de via

Pode ser:

- normal, quando coincide com o sentido normal de tráfego da via
- reverso, quando diverge do sentido normal de tráfego da via

SENTIDO NORMAL DE TRÁFEGO DE VIA – sentido do tráfego de trens, previamente definido para todos os trechos de uma via

Pode ser (no caso da linha 2 do Metrô):

- de leste para oeste (na via 1)
- de oeste para leste (na via 2)

SINALEIRO – equipamento de via que sinaliza o estado de um bloqueio de entrada

Pode assumir o aspecto:

- amarelo, quando o bloqueio está aberto para rota normal
- vermelho intermitente, quando o bloqueio está aberto para rota de chamada
- vermelho, quando o bloqueio está fechado
- apagado, quando o bloqueio está fechado

TRAVAMENTO DE MÁQUINA DE CHAVE – processo de inibição da movimentação de uma máquina de chave

TRECHO DE VIA ENTRE REGIÕES DE AMVs – conjunto dos circuitos de via compreendidos entre duas regiões de AMVs

VIOLAÇÃO DE BLOQUEIO – invasão (ou iminência de invasão) de um trem em uma região de AMVs através de um bloqueio fechado