Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/02.28.13.08-TDI

ESTUDO DE REQUISITOS E ESPECIFICAÇÕES PARA A TOLERÂNCIA A FALHA SIMPLES DO SISTEMA DE CONTROLE DE ATITUDE DA PLATAFORMA MULTIMISSÃO

Humberto Manelli Neto

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 01 de abril de 2011

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/3996485>

INPE
São José dos Campos
2011

PUBLICADO POR :

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/02.28.13.08-TDI

ESTUDO DE REQUISITOS E ESPECIFICAÇÕES PARA A TOLERÂNCIA A FALHA SIMPLES DO SISTEMA DE CONTROLE DE ATITUDE DA PLATAFORMA MULTIMISSÃO

Humberto Manelli Neto

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 01 de abril de 2011

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/3996485>

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

Manelli Neto, Humberto.

N384e Estudo de requisitos e especificações para a tolerância a falha simples do sistema de controle de atitude da plataforma multimissão / Humberto Manelli Neto. – São José dos Campos : INPE, 2011.

xxvi+181 p. ; (sid.inpe.br/mtc-m19/2011/02.28.13.08-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011.

Orientador : Dr. Marcelo Lopes de Oliveira e Souza.

1. Falhas. 2. Tolerância a falhas . 3. Sistema de controle. 4. Confiabilidade. 5. Disponibilidade. 6. Dependabilidade. I.Título.

CDU 629.7.062.2

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Mestre em

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Mario Cesar Ricci

Presidente / INPE / SJCampos - SP

Dr. Marcelo Lopes de Oliveira e Souza

Orientador(a) / INPE / SJCampos - SP

Dr. Valdemir Carrara

Membro da Banca / INPE / SJCampos - SP

Dr. Fernando José de Oliveira Moreira

Convidado(a) / EMBRAER / SJCampos - SP

Aluno (a): Humberto Manelli Neto

São José dos Campos, 01 de abril de 2011

“- Este nosso rapazinho tem vista curta. Espera aí, Miguilim...

E o senhor tirava os óculos e punha-os em Miguilim, com todo o jeito.

- Olha agora!

Miguilim olhou. Nem podia acreditar! Tudo era uma claridade, tudo novo e lindo e diferente, as coisas, as árvores, as caras das pessoas. Via os grãozinhos de areia, a pele da terra, as pedrinhas menores, as formiguinhas passeando no chão de uma distância. E tonteava.”

João Guimarães Rosa em “Manuelzão e Miguilim”.

A minha mãe pela responsabilidade; a meu pai pelo empreendedorismo; a meus irmãos pelo bom exemplo; e á minha esposa pela paciência e incentivo.

AGRADECIMENTOS

Agradeço aos professores do Curso de Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle pelos conhecimentos e sabedoria, e aos funcionários pelo suporte essencial.

Agradeço em especial ao Professor Marcelo Lopes de Oliveira e Souza. Sem sua insistência em ensinar e instigar os alunos a buscar novas fronteiras este trabalho não teria sido feito. Faço reverência à sua dedicação ao ensino e ao seu modo de trabalho: o aprendizado honesto, simples e verdadeiro.

Agradeço à EMBRAER por disponibilizar horas de trabalho para que eu pudesse assistir às aulas e iniciar este trabalho. Agradeço em especial ao colega de EMBRAER Engenheiro Paulo Donato Allemand Borges pela insistência para que eu iniciasse o programa de Mestrado.

Agradeço aos colegas do INPE Eloy Martins de Oliveira Júnior e Andreza Oliveira Batista pelo companheirismo desde a época do Período de Adaptação.

Agradeço aos colegas do INPE que, com seus trabalhos na área de controle, e sem saber do futuro, deram contribuição fundamental para a realização deste trabalho. Sem querer ser injusto com todos aqueles citados no decorrer do trabalho, agradeço em particular os Engenheiros Marcio Ferraz Gobato, Alexandre Carvalho Leite e Hermínio Duque Lustosa.

RESUMO

Em muitas aplicações de engenharia, a Confiabilidade é uma das mais importantes características. Os aspectos em torno da Confiabilidade desempenham um papel essencial em projetos de aeronaves, espaçonaves, automóveis, sistemas médicos e bancários, etc., evitando perdas de vidas e propriedade. Os sistemas altamente confiáveis são projetados para operarem continuamente mesmo sob ameaça de riscos externos e falhas internas. Não obstante, Confiabilidade não é a única característica almejada por sistemas modernos: Disponibilidade, Integridade, Proteção e Segurança são sempre partes da mesma especificação, com o mesmo nível de importância da Confiabilidade. Este trabalho estuda os requisitos e especificações para a tolerância a falha simples do sistema de controle de atitude da Plataforma MultiMissão. Para cumprir os seus objetivos, o trabalho começou por uma revisão teórica de definições e conceitos. A seguir, foi feita uma revisão histórica de soluções adotadas no passado. Em seguida, os requisitos de um projeto espacial foram selecionados. Começando de um senso mais amplo das definições, o trabalho aplicou os conceitos discutidos em dois estudos de caso. Para solucionar os requisitos levantados, duas soluções de arquitetura foram propostas: duplo-simplex e a triplo-simplex. As duas soluções constituíram-se nos estudos de caso do trabalho. As soluções foram analisadas usando-se Árvores de Falhas, simuladas através de MATLAB/Simulink® e então comparadas. Como conclusão, diferente do que se esperava no começo, a comparação não levou ao completo descarte de uma das soluções, mas sim no entendimento de que ambas são válidas mas para cenários de projeto diferentes.

A STUDY OF REQUIREMENTS AND SPECIFICATIONS FOR SINGLE FAULT-TOLERANCE OF THE MULTI-MISSION PLATFORM ATTITUDE CONTROL SYSTEM

ABSTRACT

On several engineering applications, high reliability is one of the most wanted features. The aspects of Reliability play a key role in designing aircraft, spacecraft, automotive, medical, banking, etc., systems; because it may avoid loss of life, property, or costly recalls. The highly reliable systems are designed to work continuously, even upon external threats and internal failures. Nevertheless, Reliability is not the only requirement for a modern system. Other features as Availability, Integrity, Security and Safety are always part of the same technical requirements, in the same level of importance. This work aimed at studying the requirements and specifications for single fault-tolerance of the attitude control system of the Multi-Mission Platform. To accomplish its intent, the work started by a theoretical review of definitions and concepts. Then, a historical review of adopted solutions was conducted. After that, the requirements of a spacecraft project were selected and/or written. To solve the listed requirements two architecture arrangements were proposed: the triple-simplex and the double-simplex architectures. These two architectures became the two case studies of the work. The case studies were analyzed using Fault Trees, simulated using MATLAB/Simulink©, and then compared. In the end, as conclusion, the comparison between two architectures has shown that none of them could be discarded, in the sense that it could not be useful for any application. Au contraire, the comparison has highlighted the advantages and disadvantages of both architecture; and it has indicated the applications they are more suitable for.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág.

LISTA DE SIGLAS E ABREVIATURAS

AC	Advisory Circular
ACDH	Attitude Control and Data Handling
ANAC	Agência Nacional de Aviação Civil
ARP	Aeronautical Recommended Practices
COM	Command
COTS	Commercial Off The Shelf
DAL	Design Assurance Level
EASA	European Aviation Safety Agency
ESA	European Space Agency
FAA	Federal Aviation Administration
FHA	Failure Hazardous Analysis
IFIP	International Federation for Information Processing
INPE	Instituto Nacional de Pesquisas Espaciais
LEO	Low Earth Orbit
MON	Monitor
MPP	Multi Purpose Platform
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
NASA	National Aeronautics and Space Administration
OBC	Onboard Computer
PID	Proporcional Integral e Derivativo
PMM	Plataforma MultiMissão
SACI	Satélite de Aplicações Científicas
SAE	Engineering Society for Advancing Mobility
SCA	Sistema de Controle de Atitude
SCAO	Sistema de Controle de Atitude e de Órbita
SEU	Single Event Upset
SID	Serviço de Informação e Documentação
SISO	Single Input Single Output
SPG	Serviço de Pós-Graduação
TDI	Teses e Dissertações Internas
VLHL	Vertical Local Horizontal Local

SUMÁRIO

Pág.

1 INTRODUÇÃO

1.1 Motivação e Justificativa

O estudo de Tolerância a Falhas é essencial para sistemas aeroespaciais, seja por motivos econômicos, como evitar a perda ou interrupção dos serviços de um satélite, seja por motivo de Segurança de vidas, como preservar íntegros o *Space Shuttle* e aeronaves de transporte civil. Tolerância a falhas aplica-se, principalmente, a sistemas de alta Confiabilidade, onde a operação livre de falhas é um requisito essencial. Mais do que tolerar falhas, os sistemas de alta Confiabilidade têm que operar de acordo com as especificações e, muitas vezes, sem perda de desempenho ou de funções. Alguns exemplos de sistemas altamente confiáveis são aqueles que provêem Segurança a bancos, sistemas médico-hospitalares, sistemas de guiagem de aviões e trens, e sistemas de controle de usinas nucleares.

O estudo de falhas de sistemas e de sistemas tolerantes a falhas tem produzido inúmeras referências ao longo dos anos. Assim, se pode questionar: qual seria a motivação para mais um trabalho sobre o assunto? A explicação está no fato que a grande maioria dos trabalhos e guias sobre o tema aplica-se somente ao que a Engenharia de Sistemas chamaria de componentes (e.g. principalmente microprocessadores e computadores) relegando a um segundo plano o aspecto sistêmico do tema. Mesmo no campo do desenvolvimento aeronáutico, que é muito prolífero em publicações científicas, a falta de uma referência para o desenvolvimento de sistemas é notável. Foi só em 1996, por exemplo, que, reconhecendo esta deficiência, a autoridade certificadora norte-americana – FAA (do Inglês *Federal Aviation Administration*) – encomendou à SAE uma norma que servisse de guia para o desenvolvimento de sistemas. Foi assim que, em 1996, foi apresentada a norma SAE ARP4754 (1). Desde então ela é o grande guia, e um dos únicos, para o desenvolvimento de sistemas do setor aeroespacial.

Seguindo essa tendência evidenciada acima, este trabalho propõe-se a discutir o problema das falhas e sua tolerância no âmbito de sistemas de controle para projetos aeroespaciais.

1.2 Objetivos do trabalho

O Objetivo deste trabalho é o Estudo de Requisitos e Especificações para Tolerância a Falhas Simples Aplicados a Sistemas de Controle Aeroespaciais. Para tanto, dividiu-se o trabalho em passos, como mostrado a seguir:

a) Revisar a literatura com o objetivo de definir os termos que são usados e o repertório das falhas que é tratado, identificando as suas causas e efeitos. No contexto deste trabalho, são tratadas as falhas simples, com especial atenção às falhas bizantinas e às falhas de modo comum.

b) Selecionar uma missão ou projeto espacial, elicitar e compreender seus requisitos com respeito à tolerância a falhas simples. Se for o caso, incluir normas e requisitos adotados pela indústria e/ou requeridos por autoridades reguladoras.

c) Especificar pelo menos duas soluções de arquitetura tolerantes a falhas simples para tal missão ou projeto espacial. No contexto deste trabalho são revisados os mecanismos de tolerância a falhas utilizados por projetos espaciais descritos na literatura.

d) Modelar e simular tais soluções, a fim de verificar o atendimento dos requisitos de missão ou projeto espacial anteriormente identificado.

e) Comparar e discutir os casos estudados identificando possíveis lições aprendidas de um caso para o outro.

1.3 Organização do trabalho

Para atingir os objetivos propostos, como primeira tarefa é necessário se entender e uniformizar a compreensão dos termos e definições básicas relacionadas a falhas e à Tolerância a Falhas. Do grande elenco de Falhas a que um sistema está submetido, faz-se a listagem daquelas mais relevantes para sistemas Aeroespaciais. O Capítulo 2, seção 2.1, trata de falhas e Confiabilidade de sistemas. Ainda no Capítulo 2, há ainda seções que revisam os trabalhos anteriores desenvolvidos no INPE sobre os temas tratados aqui (seção 2.2), revisão da engenharia de simulação e controle (seção 2.3) e sobre Requisitos de um sistema aeroespacial (seção 2.4).

O Capítulo 3 dedica-se a uma revisão histórica de soluções tolerantes a falhas usadas em projetos do passado e presente. O estudo de soluções já adotadas é fonte inspiradora para a proposição de soluções para os estudos de caso.

O Capítulo 4 apresenta o problema a ser resolvido (requisitos, condições de contorno, hipóteses, etc), as soluções candidatas (descrição detalhada dos mecanismos de tolerância a falhas), a verificação das soluções desenvolvidas e uma comparação entre as soluções desenvolvidas.

O Capítulo 5 se encarrega de resumir as conclusões e considerações finais sobre o trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo destina-se a esclarecer os termos e uniformizar as definições teóricas que serão utilizadas neste trabalho. Além de esclarecer, pretende também localizar os estudos dentro do universo de possibilidades que o tema dispõe, uma vez que o estudo de Tolerância a Falhas pode levar a uma discussão quase que infindável de métodos e particularidades. No decorrer do texto, o leitor identificará a aplicação ao trabalho dos conceitos discutidos nesse capítulo. Assim, como uma estratégia alternativa de leitura, propõe-se ir direto ao Capítulo 3 e, à medida que o leitor precisar de embasamento teórico, pode recorrer ao Capítulo 2, ao seu próprio gosto.

2.1 Introdução à Dependabilidade (*Dependability*)

A definição de Tolerância a Falhas em aplicações computacionais remonta a 1967, por Avižienis (3). Os americanos estavam então envoltos com o desenvolvimento do programa Apollo da NASA. De acordo com Lala et al (4), o desafio era implementar sistemas computacionais no controle de atitude do Módulo de Comando e no veículo lançador, o Saturno V. Entenda que àquela época, os computadores e autômatos não eram tão usuais como nos dias de hoje e, como toda nova tecnologia, carecia de Confiabilidade traduzindo um histórico de experiência em campo. A resposta dada pelo time à incipiência foi o uso de dois métodos distintos: no Módulo de Comando, tomaram-se cuidados na escolha dos componentes a serem usados no sistema computacional, além de se selecionar um sistema simples, sem muitos componentes a falhar. Assim, aplicou-se o conceito de Prevenção a Falhas (do Inglês *Fault Avoidance*) para se aumentar a Confiabilidade do sistema. No veículo lançador a estratégia foi outra. Sem a mesma restrição severa de peso do Módulo de Comando, configuraram-se sistemas redundantes como um meio primário para Tolerância a Falhas (do Inglês *Fault Tolerance*), e assim aumentar a sua Confiabilidade.

Assim, da necessidade do programa espacial nascia o conceito de Tolerância a Falhas em aplicações computacionais, definido da seguinte maneira por seu criador: segundo Avižienis (3), se diz que um sistema é Tolerante a Falhas "se o seu programa pode ser propriamente executado mesmo com a ocorrência de falhas lógicas". Note-se que essa primeira definição de Tolerância a Falhas é fortemente influenciada pela sua aplicação no programa espacial. Mais tarde, quando o assunto foi revisitado, outras definições desvencilhadas do programa espacial foram propostas. Entre elas, uma das definições mais recentes para Tolerância a Falhas foi proposta pela comunidade internacional de computação (materializada pela IFIP, sigla do Inglês *International Federation for Information Processing*). Ela deixa clara a idéia de Tolerância a Falhas como um meio e não um fim (como proposto no começo). Essa definição foi sumarizada por Laprie (5) do seguinte modo: "Tolerância a Falhas constitui um dos meios para se conseguir a Dependabilidade (anglicismo para o termo em Inglês *dependability*[1]) desejada". O próprio Laprie (5) define Dependabilidade como "o grau de confiança depositado em um Sistema que ele executará as funções para as quais foi projetado". Pode-se ter a impressão de que as duas últimas definições justapostas remetam à definição de Avižienis (3). Porém, há diferenças fundamentais entre as duas definições (i.e, Laprie (5) e Avižienis (3)). Primeiramente, Laprie generalizou o foco de atenção: passou de "*software* e seu programa" para "sistema e sua função". Além disso, outro detalhe pode-se perder com a tradução para o Português: Laprie tem o cuidado de usar um termo mais amplo para Confiabilidade: *Dependability* e não *Reliability* como feito por Avižienis (3). A diferença entre os dois termos será clarificada, em tempo, mais à frente.

[1] Como em Português ambas palavras *Reliability* e *Dependability* são traduzidas por Confiabilidade, optou-se por traduzir *Dependability* por Dependabilidade e *Reliability* pelo termo mais conhecido (e gramaticalmente correto), Confiabilidade. O artifício é usado para evitar confusões durante o trabalho.

Assim, antes de especificar a necessidade pela Tolerância a Falhas, é necessário entender a real necessidade de sua implementação, e quais problemas se almeja solucionar. Uma rápida reflexão deveria incluir os seguintes pontos:

a) Para satisfazer os requisitos de missão exige-se de um dado sistema uma Dependabilidade específica;

b) As falhas a que o sistema em questão é submetido impedem que os requisitos de Dependabilidade sejam alcançados;

c) Entre os métodos conhecidos para se aumentar a Dependabilidade, a Tolerância a Falha é uma das soluções possíveis.

Laprie (5) resumiu e estruturou muito bem essa “cadeia de eventos”. Segundo ele, como já foi dito, Tolerância a Falhas é um meio, Dependabilidade é o fim, enquanto as falhas são obstáculos para se ir de um a outro. A Figura 2.1 baseada em seu trabalho ilustra a cadeia de eventos:

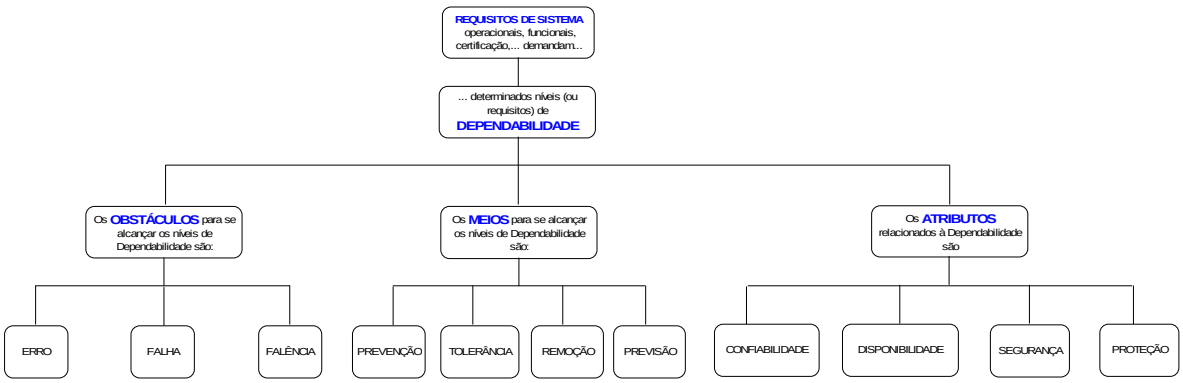

Figura 2.1- Árvore de Dependabilidade. Baseado em Laprie (5).

Note, antes de prosseguir, que foram os requisitos do Sistema que iniciaram a cadeia de eventos que culminou no Sistema Tolerante a Falhas. Não há sentido, pois, falar em Tolerância ou Prevenção a Falhas sem primeiro discutir os requisitos que o Sistema deve atender (e.g. requisito de Missão, Operação, Dependabilidade). Mais adiante, serão listados os requisitos de uma missão aeroespacial e esse assunto será retomado.

Usando a Árvore de Dependabilidade de Laprie como base, a seguir, são vistos brevemente cada um de seus ramos.

2.1.1 Atributos da Dependabilidade

2.1.1.1 Confiabilidade (ou *Reliability*)

O primeiro termo, do Inglês "*Reliability*", que em Português se traduz por Confiabilidade, carrega consigo a idéia numérica de Dependabilidade, ou, como normalmente se encontra nas referências, uma probabilidade de falha ou sucesso associada à função exercida pelo sistema. Apesar dos dois termos serem sinônimos, o termo "*Reliability*" historicamente teria sido muito atrelado ao senso numérico, não deixando claro os dois outros aspectos de "*Dependability*": "*Safety*" e "*Security*". O que Laprie (5) propõe em termos práticos é uma espécie de subterfúgio lingüístico ou uma "mudança de variáveis", como empregado em Álgebra, criando uma variável mais genérica – "*Dependability*" – e reservando "*Reliability*" ao senso quantitativo e histórico da definição. Para fazer a distinção mais clara dos dois termos, sugere-se nesse trabalho traduzir *Reliability* por Confiabilidade.

Segundo a ARP4754 (1), "Confiabilidade é a probabilidade que um item desempenhará uma função requerida sob determinadas condições, sem falhas, por um período específico de tempo". Confiabilidade, assim associado à probabilidade, reflete a expectativa de se ter um componente executando propriamente suas funções durante um intervalo de tempo.

Souza e Carvalho (6) deduzem a expressão matemática dessa probabilidade desde sua interpretação primária e intuitiva até a o seu formato final de uso prático. Por simplicidade, apresenta-se aqui só o resultado final de seu desenvolvimento:

$$R(t) = e^{-\int \lambda(\tau) d\tau} \qquad \textbf{(2.1)}$$

, onde:

R(t): Confiabilidade;

$\lambda(t)$: taxa de falha de um componente, definida a seguir.

Como elucidado por Souza e Carvalho (6), a taxa de falhas de um componente pode ser entendida através da ilustração de um teste para se determinar a sua durabilidade: N_0 unidades de um componente são colocadas em funcionamento em um tempo t_0, e passado um tempo Δt qualquer, N_F unidades haviam falhado. A taxa em que os elementos falharam no período Δt será dada por :

$$\lambda(t) = \frac{N_F}{(N_0 \Delta t)} \tag{2.2}$$

Mas note que a expressão acima, apesar de traduzir *ipsis literis* o significado do “termo taxa de falhas”, é muito dependente do intervalo Δt de medidas, o que dificultaria muito seu uso. Assim, propõe-se a relação abaixo:

$$\lambda(t) = \lim_{\Delta t \to 0} \left(\frac{N_F}{N_0 \Delta t} \right) \tag{2.3}$$

Como apontado por Heidergott (7), a taxa de falha $\lambda(t)$ é comumente encontrada associada à outra grandeza, o MTBF (do Inglês, *Mean Time Between Failures*, ou tempo médio entre falhas):

$$\lambda(t) = \frac{1}{MTBF} \tag{2.4}$$

Tradicionalmente atribui-se à MTBF o tempo entre falhas dado que o sistema esteja operacional. Mais adiante, após a definição de Disponibilidade, este assunto será retomado, discutindo-se o mérito da inclusão do tempo de manutenção no cômputo do MTBF.

Valores de MTBF são encontrados em bancos de dados dos próprios fabricantes de componentes ou em referências históricas aceitas pela comunidade de projeto, como é o caso da norma MIL-HDBK-217F (8), que traz uma grande variedade de valores de MTBF para diversos componentes. Tipicamente os valores de MTBF são apresentados em falhas por hora de operação.

2.1.1.2 Disponibilidade (ou *Availability*)

Disponibilidade e Confiabilidade são dois termos complementares, e muitas vezes, usados erroneamente de forma indistinta. A ARP454 (1) define da seguinte maneira a Disponibilidade: “Disponibilidade é a probabilidade que um item está em um estado operacional em um determinado instante.” Recorrendo à definição de Confiabilidade dada anteriormente pela própria ARP4754 (1) vê-se que elas são muitíssimo parecidas, a menos da sutileza da definição esclarecida por Heidergott (7), “Confiabilidade é a probabilidade condicional que um sistema permanecerá operacional, sem interrupções. Disponibilidade é definida como a probabilidade que um sistema estará acessível em um instante particular”. Ou seja, em termos práticos para o usuário, Disponibilidade é uma medida da prontidão do sistema, enquanto Confiabilidade é uma medida da continuidade de operação.

Heidergott (7) define Disponibilidade em termos matemáticos como a seguir:

$$A = \frac{\mu}{(\mu + \lambda)} \tag{2.5}$$

onde:

$$\mu = \frac{1}{MTTR} \tag{2.6}$$

onde:
MTTR: Tempo médio para reparo/recuperação (do Inglês *Mean Time to Repair/Recover*).

Vê-se que o conceito de Disponibilidade envolve o tempo de reparo. Heidergott (7) ainda lembra que sistema com alta Confiabilidade pode não ter alta Disponibilidade e vice-versa. Explica-se: imagine um sistema que quando em funcionamento raramente deixa de funcionar, apresentando, assim, alta Confiabilidade. Porém, tal sistema ao falhar leva muito tempo para ser reparado e colocado de volta a funcionar, apresentando, assim, baixa Disponibilidade. O mesmo exercício vale na direção contrária.

Antes de se prosseguir, vale uma ressalva interessante apontada por Kopetz (9) sobre a definição de MTBF e MTTR. Ele introduz um novo parâmetro, o MTTF (do Inglês *Mean Time to Failure*), que seria similar ao que foi definido aqui como MTBF. A definição de MTTF, como proposta por Kopetz (9), deixa claro que para fins de cálculo de Confiabilidade não se deve incluir o tempo que o sistema ficou parado em reparo. Então, Kopetz (9) define a taxa de falhas como a seguir:

$$\lambda = \frac{1}{MTTF} \tag{2.7}$$

onde:
MTTF: Tempo médio até a próxima falha;
Kopetz (9) ainda define MTBF como sendo:

$$MTBF = MTTF + MTTR \tag{2.8}$$

A

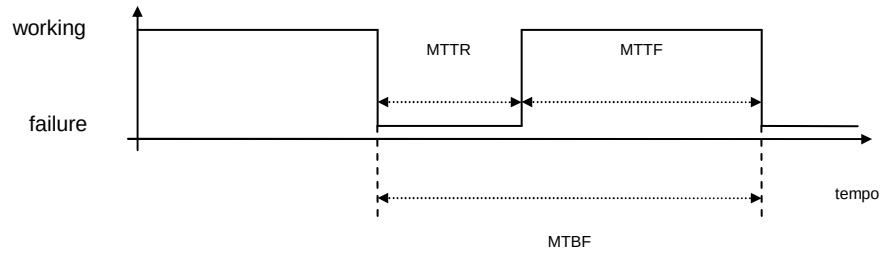

Figura 2.2, extraída de Kopetz (9), resume a explicação dada:

Figura 2.2 – Representação gráfica das definições de MTBF, MTTF e MTTR.
Fonte: Kopetz (9).

2.1.1.3 Proteção (ou *Security*)

Novamente o trabalho se deparara com uma particularidade da tradução dos termos: *Security* e *Safety*, ambos normalmente traduzidos como Segurança. Para se fazer distinção dos dois, propõe-se tratar *Security* como “Proteção” e *Safety* como “Segurança”. Certamente a melhor tradução para *Security* seria Segurança, mas com o sentido de Proteção à informação. Assim, mesmo que se deteriore um pouco a qualidade da tradução, mas em favor do entendimento, Proteção será usada doravante.

Segundo Laprie (5), Proteção significa “prevenção a acessos não autorizados ou manipulação não autorizada de informação”. Dos atributos listados por Laprie (5) em seu trabalho, esse talvez seja o que historicamente tem menos definido as características de um projeto de sistema aeroespacial. Não que houvesse negligência com o aspecto de Proteção dos sistemas embarcados, mas os procedimentos de acesso e manipulação até então usados sempre se mostraram eficientes. Tanto isso é verdade que não há requisitos específicos de certificação de aeronaves civis com respeito a esse aspecto. Nem mesmo a ARP4754 (1), que é um documento atual, sequer menciona Proteção.

Porém, os recentes ataques terroristas, o crescente uso de componentes convencionais – COTS (do Inglês *Commercial of The Shelf*) – em sistemas embarcados e o acesso a redes cada vez mais integradas, têm provocado novas discussões sobre o tratamento da Proteção em sistemas aeroespaciais. O fabricante americano *Boeing* foi surpreendido durante o desenvolvimento do Boeing 787 com a suspeita de que passageiros poderiam acessar o barramento de dados que controla o sistema de Comandos de Vôo da aeronave através do sistema de entretenimento pessoal. A suspeita foi levantada e documentada nada menos que pelo próprio FAA, que emitiu uma *Special Condition*[2] (10) para tratar do assunto. Segundo o documento:

> A Arquitetura do sistema computacional e redes do Boeing modelo 787-8 pode permitir o acesso a sistemas e redes externas, tais como redes sem-fio para operação da linha-aérea e manutenção dos sistemas embarcados, comunicações via-satélite, e-mails, Internet, etc., aparelhos sem-fio ou com fios embarcados podem ter acesso a partes do avião que provêem funções críticas de vôo. Essas novas capacidades de conexão podem resultar em problemas de vulnerabilidade da Proteção aos sistemas críticos do avião (10).

Em resposta à preocupação do FAA a *Boeing* e a fabricante de sistemas *Honeywell*[3] lideraram a escrita de um documento que não só respondeu à preocupação específica que o originou, mas também despontou como um poderoso guia de análise e projeto de redes de dados embarcadas, quanto à

---

[2] *Special Conditions* são documentos emitidos pelo órgão de certificação civil americano – FAA – para os fabricantes aeronáuticos. O *Special Condition* tem a função de cobrir assuntos ainda não tratados pelos requisitos do FAR (do inglês, *Federal Aviation Requirements*) 14 CFR Part 25. Normalmente, os *Special Conditions* são emitidos para tratar de novas tecnologias da indústria.

[3] *Honeywell Co.* é a empresa americana responsável pelo fornecimento do sistemas de Comandos de Vôo do *Boeing 787*, entre outras partes.

sua adequação para uso aeronáutico. No capítulo 9, o documento trata de Proteção a redes embarcadas (11): “Historicamente, Proteção a dados de comunicação nunca foi um problema para eletrônica da aviação comercial.” Além dos motivos levantados no *Special Condition* (10) do FAA, o documento ainda lista o crescente uso de componentes do tipo COTS como um dos fatores que justifica o aumento da sensibilidade da comunidade aeronáutica,

> O aumento do uso de protocolos e tecnologias de rede do tipo COTS – os quais têm as suas próprias fraquezas inerentes – têm o potencial de atrair invasores que sejam familiares com essas fraquezas (10).

Para facilitar a avaliação da rede em questão, o documento sugere um questionário sobre os principais pontos sensíveis para a Proteção dos dados, que requererão mais atenção dos projetistas. O documento ainda guia a análise das respostas, para não deixar a sua avaliação dependente do subjetivismo do auditor.

2.1.1.4 Segurança (ou *Safety*)

A Segurança de sistemas é tema freqüente na literatura aeroespacial, muitas vezes usada como um termo genérico, sem definição padronizada, como se fosse um ideal de projeto.

Segundo a ARP4754 (1), Segurança é “o estado em que o risco é menor do que o risco aceitável. O risco aceitável é definido por um processo ou é por declaração”. Laprie (5) define Segurança da seguinte maneira, “Dependabilidade com respeito a não ocorrência de falências que sejam catastróficas”. Definição essa similar àquela encontrada em Kopetz (9), “Segurança é a Confiabilidade com relação a modos críticos de falhas. Um modo de falha crítico é dito ser maligno em contraste com um modo não crítico que é dito ser benigno”. Note que essas definições delineiam Segurança como um estado crítico, que está além dos níveis de risco toleráveis, que, no pior dos cenários, levaria a um evento catastrófico. Apesar de a idéia ser intuitivamente correta, é muito subjetiva. Diferentes pessoas podem ter diferentes opiniões

sobre o que é “aceitável” ou “crítico”. Talvez, apesar de haver uma idéia de Segurança, não haja uma definição universal para o termo, e cada aplicação deva definir os seus próprios níveis “críticos” e “toleráveis”.

No setor civil aeronáutico, Segurança é um termo muito bem definido pelos requisitos de certificação, em especial pelo requisito FAR 25.1309 (12) e *Advisory Circular* 25.1309[4] (13), que estabelecem o que as autoridades certificadoras chamam de *Safe Design Concept* ou Conceito de Projeto Seguro.

O Conceito de Projeto Seguro (mesmo conceito adotado pelas agências de certificação européia – EASA – e a brasileira – ANAC) estabelece uma relação inversa entre severidade das falhas e sua probabilidade (Confiabilidade), i.e., quanto mais severa a falha, menos provável ela deve ser. Segundo a AC 25.1309 (13), as falhas são divididas em 5 diferentes níveis de severidade,

a) Catastrófico (traduzindo do Inglês, Catastrophic).

b) Perigoso (traduzindo do Inglês, Hazardous);

c) Maior (traduzindo do Inglês, Major);

d) Menor (traduzindo do Inglês, Minor);

e) Sem impacto em Segurança (traduzindo do Inglês, No Safety Effect);

A avaliação da severidade é um processo longo que envolve a identificação das falhas, a caracterização de seus efeitos nos passageiros, no aumento da

[4] *Advisory Circulars* são documentos emitidos pelo FAA com o objetivo de elucidar os requisitos de certificação e, principalmente, quais são os meios aceitáveis de cumprimento a esses requisitos.

carga de trabalho da tripulação e na aeronave e, finalmente, a classificação propriamente dita da severidade. Todo esse trabalho é normalmente capturado em um documento chamado FHA (do Inglês *Failure Hazardous Analysis*). Vale notar que a classificação da severidade é um processo padronizado que tem diretrizes claras de como as falhas devem ser classificadas nos 5 níveis. Para maiores informações sobre o processo de FHA, a norma SAE ARP4761 (14) é um dos melhores guias da atualidade.

A Confiabilidade também é dividida em 3 categorias:

f) Provável (traduzindo do Inglês, *Probable*);

g) Improvável (traduzindo do Inglês, *Improbable*);

h) Extremamente Improvável (traduzindo do Inglês, *Extremely Improbable*).

A AC25.1309 (13) ainda estabelece números para as categorias de Confiabilidade:

a) Falhas Prováveis são aquelas que têm uma (ordem de) probabilidade maior que $1x10^{-5}$ falha/h;

b) Falhas Improváveis são aquelas que têm uma (ordem de) probabilidade menor que $1x10^{-5}$ e maior que $1x10^{-9}$ falha/h;

c) Falhas Extremamente Improváveis são aquelas que têm uma ordem de grandeza menor ou igual à $1x10^{-9}$ falha/h.

No requisito 25.1309 (12) são encontradas as relações admissíveis entre severidade e probabilidade. Seguem algumas das relações mais relevantes do requisito:

> 25.1309 (a)(1): a ocorrência de qualquer condição de falha que possa prevenir a continuidade do vôo e pouso seguro da aeronave deve ser Extremamente Improvável.
> 25.1309 (a)(2): a ocorrência de qualquer outra condição de falha que possa reduzir a capacidade do avião ou a habilidade da tripulação em

> responder a situações adversas de operação deve ser Improvável (12).

Na AC25.1309 (13) também são encontradas algumas relações entre severidade e análises qualitativas (independentes de sua probabilidade):

> 5. (a)(1): Em qualquer sistema ou subsistema, qualquer falha simples durante o vôo (desde liberação do freio na decolagem até a parada no pouso) deve ser suposta independente de sua probabilidade (13).

Assim, com o Conceito de Projeto Seguro, as autoridades certificadoras não se preocupam em ter uma simples definição, mas um projeto que atenda a uma filosofia de Segurança, muito bem definido por números e critérios de avaliação de falhas. Mas o que dizer daqueles eventos a que não se pode associar uma probabilidade, como os *softwares*, por exemplo?

Recentemente, com o aumento do uso de componentes micro-codificados e de *software* as autoridades viram-se obrigadas a repensar o modo de análise e certificação de tais elementos à luz do requisito 25.1309 (12). A própria AC 25.1309 (13) admite sua limitação na cobertura de *softwares*, “Em geral, os meios de cumprimento a requisitos estabelecidos nesta AC não são diretamente aplicáveis a *software*, porque não é possível analisar os erros de *software* em termos de números ou tipos”. Em outras palavras, não se atribui probabilidades a erros de *software* e de *hardware* complexo, por conta da natureza incontável de seus estados de erro.

A saída dessa aparente incompatibilidade foi a criação do conceito de DAL (*Design Assurance Level*, traduzido do Inglês como Nível de Confiança do Sistema) e dos processos de concepção de *software*, a DO178B (15), e de *hardware*, a DO254 (16). O conceito da aplicação de processos é evitar que haja erros durante a concepção dos elementos, atribuindo assim um certo *pedigree* àquele componente. O DAL nada mais é que a dosagem do rigor aplicado na concepção. Assim, quanto mais severa for a falha do componente, mais rigoroso deverá ser o seu processo de concepção. A discussão em torno

dos Erros de Projeto (ou *Design Errors*) é, em si, um trabalho à parte. Manelli et al (17) resumiram a discussão em torno do uso de *software* e de *hardware* micro-codificados em sistemas embarcados.

2.1.1.5 Integridade (ou *Integrity*)

O trabalho de Laprie (5) tem sido uma das mais usadas referências para este trabalho, mas Laprie (5) não elenca a Integridade como um dos seus atributos. Ele menciona no texto que se trata de um pré-requisito para os outros atributos. Integridade, segundo Laprie (5), “é a condição de ser incorruptível, no senso geral do termo”. Já a ARP4754 (1) assim define Integridade: “Atributo de um sistema ou um item indicando que se pode confiar para funcionar corretamente”. Ou seja, Integridade é o atributo que passa ao usuário a credibilidade que a resposta do sistema é crível, está correta.

Comumente, Integridade é associada à probabilidade de mau funcionamento dos sistemas. Note que Integridade e Confiabilidade são termos complementares: Confiabilidade denota a probabilidade de funcionamento enquanto que Integridade denota a correção do resultado.

2.1.2 Obstáculos à Dependabilidade – Falhas (*Faults*), Erros (*Errors*) e Falências (*Failures*)

Uma vez estudados os atributos da Dependabilidade, serão analisados agora os elementos que causam a sua debilitação. Aqui serão também listadas as principais falhas, que, quando conhecidas de antemão, são transformadas em requisitos que definirão barreiras protetoras do sistema. Falhas, Erros e Falências podem ser, na verdade, diferentes estados do mesmo acontecimento só que em diferentes instantes no tempo. Esses termos, apesar de serem usados como sinônimos têm significados distintos. Segundo Anderson et al (2), diz-se de um sistema que houve sua Falência “quando o comportamento do

sistema se desviou daquele esperado por sua especificação". Definição similar encontra-se na ARP4754 (1): "Falência é a inabilidade de um sistema em realizar as suas funções". Falência é o evento perceptível em nível do usuário, pois é quando o sistema não responde mais às suas expectativas. Por exemplo, quando o sistema pára de funcionar ou está funcionando de forma errada, diz-se que houve sua Falência.

Anderson et al (2) definem da seguinte maneira Erro: "é um estado errôneo assumido pelo sistema que constitui um desvio do estado válido". Erro é qualquer estado que o sistema assuma que diverge de sua especificação. Uma seqüência de estados errôneos pode ou não levar à Falência do sistema, mas se pode dizer que toda Falência começou com um Erro. A ARP4754 (1) tem uma visão diferente de Erro: "(1) é uma ocorrência proveniente do resultado de uma ação ou decisão tomada pela operação ou manutenção do sistema; (2) um engano na definição de requisito ou na sua implementação". Por se tratar de um guia de projeto e análise de sistemas, a definição da ARP 4754 (1) está mais voltada a identificar as origens do Erro, enquanto que Anderson et al (2) dão um tom mais genérico, de definição do que seja Erro.

Laprie (5) define Falha como simplesmente "a causa hipotética ou confirmada de um erro ou a causa do erro a ser evitada". Essa última definição de Laprie (5) – causa do erro a ser evitada – resume a motivação em se estudar "Tolerância a Falhas": atuando-se na raiz do problema – ou seja, nas Falhas – pode-se evitar a Falência do sistema.

2.1.2.1 Falhas

Há várias maneiras de se classificar as Falhas e, cada projetista o deve fazer da forma mais conveniente para o seu estudo. Laprie (5) e Kopetz (9) sugerem taxonomias parecidas para a classificação das Falhas. Laprie (5) divide as falhas de acordo com os seguintes critérios:

a) Natureza:

Acidental: falhas criadas por acidente, como seleção errada de tensão, por exemplo.

Intencional: falhas criadas intencionalmente, como vírus de computador, por exemplo.

b) Origem:

Fenômeno de origem:

Físico: falhas causadas por um fenômeno físico adverso, como desgaste, fadiga, pressão, calor, vibração, etc.

Humano: falhas causadas por imperfeições humanas, como manutenções erradas, operação fora do padrão estabelecido, etc.

c) Fronteiras do sistema:

Interno: falhas que ocorreram dentro do sistema.

Externo: falhas que afetaram o sistema, mas ocorreram fora dele, como interferência eletromagnética, vazamento de combustível, vazamento de vapor, etc.

d) Fase de origem:

Falhas de projeto: falhas introduzidas durante a concepção do projeto, como requisitos e codificação, ou introduzidas durante a vida em serviço do sistema, como melhorias e adaptações.

Falhas de operação: falhas introduzidas durante a manipulação do sistema pelo operador.

e) Persistência:

Permanentes: falhas independentes de uma condição específica que perduram por toda a fase de operação.

Temporárias: falhas presentes somente durante uma condição específica e/ ou durante um período limitado de tempo.

A Figura 2.3 resume a taxonomia das Falhas segundo Laprie (5):

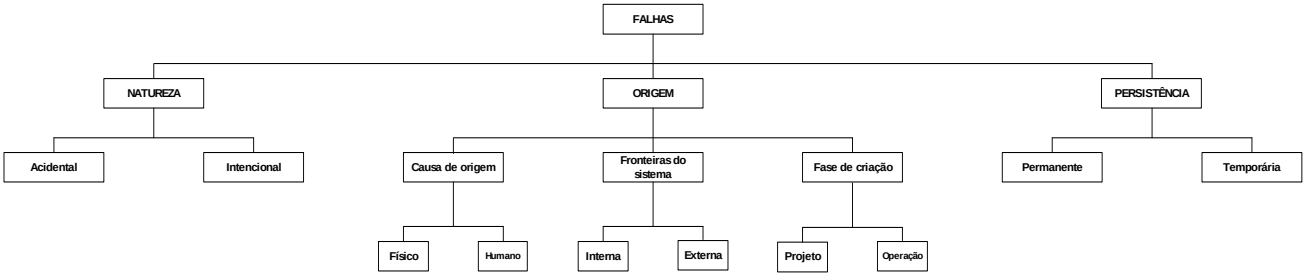

Figura 2.3 - Taxonomia das falhas.

Fonte: Laprie (5).

Logicamente, podem-se reagrupar as classes de acordo com a conveniência da análise e, principalmente, a experiência o projetista. A experiência também é um fator primordial para a seleção da gama de falhas pela qual um sistema deva ser analisado. Certamente o número de falhas possíveis a que um sistema está sujeito é muito grande, porém a vivência e a experiência apontam aquelas mais importantes, que devem ter a análise priorizada. A bibliografia consultada, grande fonte de concentração da experiência, apontou para algumas falhas de interesse, sumarizadas na
Tabela 2.1.

Tabela 2.1 - Falhas que serão estudadas e consideradas para os estudos de caso.

Referência	Nome da Falha	Natureza	Origem	Fronteira	Fase de origem	Persistência
1	Falha Aleatória	Acidental	Físico	Interna	Operação	Permanente ou Temporária
2	Falha de Projeto	Acidental	Humano Físico	Interna	Projeto	Permanente ou Temporária
3	*Single Event Upset*	Acidental	Físico	Interna	Operação	Permanente ou Temporária
4	Falha de Modo Comum	Acidental	Físico	Interna	Operação ou Projeto	Permanente ou Temporária
5	Falha Bizantina	Acidental	Físico	Interna	Operação	Permanente ou Temporária

Nas seções seguintes serão descritas as Falhas apresentadas na Tabela 2.1.

Antecipando a seção de análise dos estudos de caso, vale ressaltar já aqui que, para o propósito deste trabalho será considerada só uma das falhas apresentadas de cada vez. Em outras palavras, serão analisadas só falhas simples. O trabalho não perde em conteúdo, mas ganha em simplicidade, pois a análise de falhas combinadas, apesar de interessante e muito enriquecedora, é muito extensa e, certamente, merece um trabalho por si.

2.1.2.2 Falha Aleatória

O termo Falha Aleatória é usado principalmente para se fazer distinção do termo Falha de Projeto. Ou seja, enquanto Falha de Projeto preocupa-se em discutir os erros de requisitos e implementação, as Falhas Aleatórias são creditadas puramente a fenômenos físicos, como desgaste e fadiga dos

componentes. A esse tipo de falha – diferentemente das Falhas de Projeto – pode-se associar uma probabilidade e assim se calcular a Confiabilidade do sistema. Como o próprio nome sugere, esse tipo de falhas ocorre de forma aleatória no sistema, mas tem seu comportamento bem definido, ou seja, um componente falhado sempre provocará o mesmo efeito no sistema.

Para o propósito deste trabalho, serão impostas as Falhas Aleatórias em componentes chaves das soluções, analisando-se, a seguir, os efeitos no sistema.

2.1.2.3 Falhas de Projeto

Segundo Avižienis (18), “Falhas de Projetos são falhas introduzidas durante o projeto *(bugs* de *software* ou erratas de *hardware*) que permanecem indetectadas durante o processo de desenvolvimento do produto (*software* ou *hardware*) e se manifestam durante a operação do sistema”. Os termos-chave da definição são “falhas introduzidas durante o projeto” e “permanecem indetectadas” até a operação.

O primeiro termo-chave da definição – “falhas introduzidas durante o projeto” – é explicado pelo fato de que durante o desenvolvimento de *software* e *hardware* complexos há várias oportunidades para se inserir falhas no produto: desde requisitos mal-escritos ou mal-interpretados, até falhas introduzidas por ferramentas de síntese e teste. Já o segundo termo-chave – “permanecem indetectadas” – deve-se ao fato de que apesar da grande quantidade de testes realizada em *software* e *hardware* complexos, não é possível se esgotar todos os cenários de testes possíveis (formados pela combinação de todas as portas lógicas e laços de decisão).

Para se ter uma idéia da importância das Falhas de Projeto em sistemas embarcados veja a seguir o relato transcrito por Kopetz (9):

> A Falha do Ariane 501 foi causada pela completa perda das informações de guiagem e atitude 37 segundos após a ignição dos motores principais. Esta perda de informação foi devida a erros de especificação e projeto no software do sistema de referência inercial (9).

Embora *software* e *hardware* complexos estejam em voga ultimamente, Falhas de Projeto podem afetar outros componentes do sistema, como foi o caso do pára-quedas da *probe* do projeto Galileo da NASA. Segundo a própria NASA (19):

> A *probe* atmosférica acionou seu primeiro pára-quedas um minuto atrasada em relação ao planejado, levando à perda das leituras dos dados da alta atmosfera. As investigações concluíram que o acelerômetro que controlava o sistema pirotécnico do pára-quedas foi instalado ao contrário. Ao final, a abertura do pára-quedas foi considerada um caso de sorte (19).

Para os fins deste trabalho, não serão discutidos os métodos de Prevenção a Falhas promovidas pelos processos de desenvolvimento de *software* e *hardware* complexos. Para a análise da sensibilidade das soluções de arquiteturas propostas com relação a Falhas de Projeto, será aplicada a metodologia proposta em Manelli et al (17). Trata-se de uma análise qualitativa que propõe uma falha simples dos elementos de *software* e *hardware* complexos e a análise dos seus efeitos no sistema. O método será detalhado mais adiante quando forem estudados os casos de estudo.

2.1.2.4 Single Event Upset - SEU (ou Perturbação por um Evento Simples)

Single Event Upset (ou simplesmente SEU), segundo Heidergott (7):

> SEU é a designação da interação de partículas energizadas com as estruturas micro-eletrônicas de equipamentos. As partículas causadoras de SEUs no espaço são o resultado de raios cósmicos vindos da galáxia e do Sol, prótons aprisionados e partículas de eventos solares (7).

Ainda segundo Heidergott (7), como resultados dessas interações podem ser observadas “mudanças de estado em semicondutores, ruptura de dielétricos e outros potenciais efeitos destrutivos.”

Essas interações podem ser vistas com mais intensidade fora da atmosfera. Heidergott (7) explica que “As ocorrências mais severas estão no espaço, onde a própria severidade varia, dependendo das condições da órbita (altitude e inclinação)”. Ele lista uma quantidade razoável de projetos que comprovaram a existência na prática de SEUs:

> Clementine, um projeto de curta duração para mapeamento da Lua, experimentou 71 erros por dia em uma memória de imagem de 2.1 Gbit [...] O projeto Cassini, uma missão para a exploração de Saturno, experimentou 280 erros por dia em uma memória de 2.5 Gbit (7).

A incidência de SEUs, apesar de mais intensa fora da atmosfera, não é exclusiva daquele ambiente. Segundo Normand et al (20):

> Componentes de interesse usados em sistemas aviônicos podem ser susceptíveis a SEUs induzidos por nêutrons na atmosfera. Este fato foi demonstrado por registros de vôo e por testes com feixes de nêutrons em dois tipos de componentes, memórias e microcontroladores (20).

Normand et al (20) propuseram um teste em receptores do protocolo ARINC429, um dos mais usados em aviação comercial, simulando-se níveis de nêutrons encontrados em altitudes da atmosfera em que operam os aviões comerciais. Os resultados comprovaram a susceptibilidade desses componentes a SEUs.

Para o propósito deste trabalho, será analisado o efeito que um evento SEU em um micro-computador ou memória causaria no sistema. Espera-se que as arquiteturas propostas possam, ao menos, suportar um evento de SEU.

2.1.2.5 Falhas de Modo Comum

Segundo a ARP4754 (1), “Falhas de Modo Comum são eventos que afetam simultaneamente diferentes partes do sistema que até então eram consideradas independentes”. A análise de Falhas de Modo Comum é particularmente importante em sistemas complexos e altamente integrados, onde são feitas várias hipóteses de independência entre os sistemas. Por exemplo, uma hipótese importante na aviação comercial, que deve ser garantida pela análise de Falhas de Modo Comum, é a independência entre o sistema de fornecimento de energia principal – geradores elétricos acoplados aos motores – e as baterias de emergência. Nenhuma Falha de Modo Comum pode afetar esses dois sistemas ao mesmo tempo.

As Falhas de Projeto podem ser consideradas como Falhas de Modo Comum, quando são originadas por um mesmo agente, por exemplo: uma falha intrínseca no projeto dos microprocessadores de um determinado fabricante provocados por uma falha na produção, ou a síntese errada de software provocada pela falha de compiladores. Outra fonte importante para Falhas de Modo Comum são agentes naturais como calor/frio em excesso, umidade, pressão, etc.

2.1.2.6 Falha Bizantina

A Falha Bizantina foi proposta pela primeira vez por Lamport et al (21) em 1980. O nome se inspirou em um problema fictício de comandantes do exército bizantino que têm que decidir sobre um ataque (ou recuo) de forma autônoma, mas com base nas ordens de um general. Além disso, existe a possibilidade de haver um traidor entre eles. Segundo Lala et al (4):

> A Falha Bizantina é um modelo de falha conservativo que consiste em atribuir um comportamento arbitrário aos componentes falhados. Este tipo de falha pode incluir interrupção e depois reinício de serviços, envio de mensagens conflitantes e, em suma, qualquer

> coisa que um componente falhado pode tentar para corromper o sistema (4).

Em comparação com as Falha Aleatórias, a Falha Bizantina, apesar de ser um modo de falha atribuído a fenômenos físicos, tem um caráter mais imprevisível, onde os componentes não são coerentes ao falhar.

A Falha Bizantina em si e os algoritmos que se propõem a tratá-la são detalhados nos trabalhos de Lamport et al (21). Lamport et al (21) propõem dois algoritmos: os que usam mensagens orais OM(m) e aqueles que usam mensagens escritas SM(m). A letra “m” representa o número de traidores do exército, o que, no nosso caso, será o número de falhas a que o sistema será submetido. O algoritmo de mensagens orais, OM(m), requer, no mínimo, 3m + 1 elementos para tolerar m traidores. Assim, para se tolerar a presença de 1 traidor no exército (ou 1 falha), seriam necessários 4 elementos entre generais e comandantes para que houvesse uma decisão consistente. Usando-se o algoritmo de mensagens escritas SM(m), 3m elementos são suficientes para se tolerar m traidores, pois as ordens do general são assinadas e pode-se identificar a sua marca pessoal.

Para os fins desse projeto, será imposto um “traidor” a cada proposta, em pontos estratégicos das soluções. Espera-se que as propostas sejam imunes ao menos a um “traidor”.

2.1.3 Meios para promover o aumento da Dependabilidade

Recapitulando o “plano de vôo” do trabalho, tomando-se por base a “Árvore de Dependabilidade”, até agora foram definidos e estudados os atributos de Dependabilidade e os seus obstáculos. Nas próximas seções serão definidos e estudados os métodos disponíveis para se promover o aumento (ou se atingir os níveis especificados) de Dependabilidade.

Laprie (5) sugere quatro meios para se promover o aumento da Dependabilidade de sistemas, ou como ele mesmo os denomina, *How to's*:

a) Prevenção a Falhas: Como prevenir a ocorrência de uma falha;

b) Tolerância a Falhas: Como prover um serviço de acordo com as especificações, mesmo na presença de falhas;

c) Remoção de Falhas: Como reduzir a presença de falhas;

d) Prognóstico de Falhas: Como estimar o número atual, a incidência futura e as conseqüências da falhas.

A escolha do método adequado ao projeto nem sempre é excludente. A pergunta, talvez, não seja "Qual método?", mas sim "Quando?". Pode-se e recomenda-se a aplicação dos vários métodos em etapas distintas do projeto, como sugere Laprie (5):

> Todos os "Como Fazer" (*How to's*), na verdade, são metas que não podem ser atingidas integralmente, pois são atividades humanas, e sendo assim, imperfeitas. Somente a aplicação combinada dos métodos pode levar à concepção do melhor sistema possível (5).

A dependência dos métodos pode ser exemplificada como a seguir: não importam as metodologias de prevenção a falhas aplicadas, as falhas existirão. Daí a necessidade pela remoção de falhas. Mas métodos de remoção são, na sua maioria, COTS, que são imperfeitos, daí a necessidade pelo prognóstico de falhas. Porém, a nossa dependência cada vez maior dos sistemas, gera espontaneamente requisitos de tolerância a falhas para os sistemas. Porém,a tolerância a falhas baseia-se em regras construtivas, daí a necessidade de remoção de falhas, prognóstico de falhas, etc".

Apesar da natural integração dos métodos e da tentação em se aprofundar em cada um deles, em favor do planejamento deste trabalho somente Tolerância a Falhas será estudado.

2.1.3.1 Tolerância a Falhas

Anderson et al (2) e Laprie (5) têm definições similares das fases constituintes do processo de Tolerância a Falhas. Segundo Anderson et al (2), as quatro fases de tolerância a falhas são:

a) Detecção de erros: Para se tolerar falhas é preciso primeiro detectá-las. Enquanto a falha não puder ser detectada pelo sistema, a manifestação da falha causará seguidos estados errôneos. Assim, usualmente, o ponto inicial para tolerância a falhas é a detecção de estados errôneos.

b) Confinamento e avaliação dos danos causados: Quando um erro é detectado, devido ao atraso entre a manifestação de uma falha e a percepção do estado errôneo informação inválida pode ter se espalhado pelo sistema. Assim, antes de qualquer tentativa de tratar da falha é necessário avaliar a extensão do dano.

c) Processamento de erros. Segundo Laprie (5), a fase de Processamento ainda se subdivide em: Recuperação e Compensação do erro. Seguindo a fase de confinamento e avaliação do erro, a recuperação levará o sistema de um estado errôneo em um bem definido estado livre de erros.

d) Tratamento da Falha e continuidade de operação: Embora a fase de recuperação possa ter retornado o sistema para um estado livre de erros, faz-se necessário garantir que a falha que causou o estado errôneo não volte a acontecer. A relação entre erro e falha nem sempre é evidente, assim o primeiro aspecto do tratamento de falhas é localizar precisamente o local da falha e isolá-lo do resto do sistema.

Ainda segundo Anderson et al (2), apesar de ser um bom ponto de partida, o desenvolvimento e detalhamento das quatro fases acima devem ser precedidos por discussões do tipo onde é necessário aplicar tolerância a falhas e quanto é

necessário. A discussão da quantidade, nos leva irremediavelmente à discussão sobre Redundância.

2.1.3.2 Redundância

Segundo a ARP4754 (1), Redundância “é a técnica de se prover múltiplas implementações de uma função seja por múltiplos itens ou por múltiplos canais dentro de um item.” E continua, “é a técnica de projeto baseada na hipótese de que um conjunto de falhas com o mesmo efeito de sistema não ocorrerão ao mesmo tempo em dois ou mais elementos independentes”.

O mais interessante dessa definição da ARP4754 (1) é o modo como ela ressalta a importância da independência. A independência dos elementos redundantes é condição fundamental para sua existência. Tão importante quanto prover elementos redundantes é garantir que a hipótese de independência seja verdadeira para o sistema em questão.

Segundo Anderson et al (2), a redundância pode ser classificada em dois tipos, a saber: Estática e Dinâmica. Na Redundância Estática, segundo eles, componentes redundantes dentro de um sistema são organizados de tal maneira que os efeitos de um componente faltoso são mascarados para o sistema.

Em contraste à Redundância Estática, a Redundância Dinâmica é usada somente para detectar a falha e tem que ser suplementada em outro lugar qualquer do sistema para se conseguir a tolerância a falhas. Patton (22) sugere outra classificação baseada em outros aspectos da redundância: Física e Funcional. A Redundância Física coincide com a definição dada anteriormente pela ARP4754 (1), trata-se da implementação de itens de *hardware* com a mesma função e independentes. Já a Redundância Funcional, segundo Patton (22):

> São basicamente as técnicas de processamento sinais que empregam estimativas de estado, de parâmetros, filtros adaptativos, etc […] Usando-se estas técnicas, é possível gerar sinais que podem ser usados em algoritmos de votação de maioria simples assim como sinais vindos de Redundâncias Físicas (22).

Apesar de a redundância ser um fator primordial da tolerância a falhas, o seu uso sem gerenciamento é inútil. Segundo Lala et al (4):

> Redundância também pode complicar substancialmente a tarefa de validação. Na verdade, no princípio era fácil projetar um sistema com redundância que era mais suscetível a falhas que antes. Um fator contribuinte era a diretriz *ad hoc* de se adicionar redundância sob a premissa de que redundância era igual à tolerância a falhas (4).

Segundo Anderson et al (2), “o perigo de se incorporar redundância em um sistema é que a Dependabilidade do sistema como um todo pode reduzir, devido ao aumento do número de componentes.”

Redundâncias podem também representar um ônus extra para os recursos do sistema, como salienta Lala et al (4) “um computador tolerante a falhas pode gastar até 50% do seu potencial de vazão de tarefas (*throughput*) gerenciando redundâncias”. Para dirimir esse problema, Lala e Haper (4) propõem a criação de regiões de contenção de falhas (FCR, do Inglês *Fault Containement Units*) menores e isoladas fisicamente entre si, de tal modo que os recursos do sistema sejam melhores geridos. A idéia de desenvolver partes menores do sistema tolerantes a falhas é corroborada por Anderson et al (2), “é amplamente conhecido entre os projetistas que a redundância de hardware é mais efetiva quando aplicada a componentes do que quando aplicada a sistemas.” O conceito de regiões de contenção de falhas será melhor discutido a seguir.

2.1.3.3 Regiões de Contenção de Falhas

Um conceito muito importante em sistemas Tolerantes a Falhas é o conceito de Regiões de Contenção de Falhas (FCR, do Inglês *Fault Containment Regions*). Segundo Butler (41):

> O principal objetivo de Sistemas Tolerantes a Falha é limitar os efeitos de uma falha e prevenir a propagação dos erros de uma região do sistema para outra. Uma região de Contenção de Falhas é um subsistema que irá operar corretamente independente qualquer falha arbitrária fora de sua região (41).

Regiões de Contenção de Falhas são criadas através do isolamento físico e lógico de partes do sistema, e da proposição de mecanismos que promovam o isolamento. Monitores que detectam falhas e isolam a região do sistema falhado são usualmente empregados. Embora Regiões de Contenção previnam a propagação de Falhas, os estados errôneos, irremediavelmente, são transmitidos a outras regiões, fazendo-se necessário o uso de barreiras de votação (*voting planes*) para o mascaramento dos erros. Segundo Lala et al (4):

> Embora um FCR possa prevenir que uma Falha propague para outras FCR's, os efeitos das Falhas podem propagar através das fronteiras da FCR. Portanto, o sistema deve prover contenção de erros também. O princípio básico é simples: "barreiras de votação" mascaram Falhas em diferentes estágios em um sistema Tolerantes a Falhas (4).

2.1.3.4 Votação

Para se mascarar Falhas é necessário comparar e votar dados de canais ou regiões redundantes. Há duas maneiras distintas para se votar dados: Consenso Exato e Aproximado.

O Consenso Aproximado faz uso de *thresholds* (ou tolerâncias) para se comparar duas grandezas. Segundo Lala et al (4):

> Já que não há uma maneira precisa de se definir *thresholds,* a maioria dos projetistas usa métodos empíricos para satisfazer dois requisitos antagônicos: o *threshold* muito pequeno gera falsos alarmes; o *threshold* muito folgado evita falsos alarmes mas deixará algumas Falhas reais sem detecção. Devido a esse dilema, a aplicação da estratégia de Consenso Aproximado não garante detecção de 100% das Falhas (4).

O Consenso Exato baseia-se no fato de que dentro de certas condições computadores devem gerar os mesmos resultados, *bit*-a-*bit*. Assim, não se faz necessário o uso de *thresholds* já que a comparação é feita por seqüências de binários. Porém, para tornar isso possível é necessário se garantir que:

a) Os estados iniciais dos dois computadores devem ser os mesmos;

b) As entradas dos computadores deve ser a mesma, no mesmo instante de tempo;

c) As operações executadas pelos computadores devem ser as mesmas;

d) Os atrasos entre os dois computadores devem ser limitados, a fim de não se perder o sincronismo entre eles.

2.2 Resumo de alguns trabalhos científicos do INPE sobre e Detecção de Falhas

Os resultados apresentados aqui representam os esforços em conjunto de vários pesquisadores do Laboratório de Simulação e Controle da Divisão de Mecânica Espacial e Controle-DMC do INPE. As atividades que culminaram no presente trabalho começaram com Prudêncio (25).

Prudêncio (25) discutiu o projeto e a simulação em Tempo Real de Sistemas de Controle de Atitude (SCA) de satélites utilizando um computador (um *desktop*

convencional) e o ambiente MATRIXx/SystemBuild®. Este trabalho utilizou como aplicação o SCA do Satélite de Aplicações Científicas (SACI-1). Mostrou o desenvolvimento dos modelos matemáticos do ambiente de operação do satélite e da dinâmica do satélite utilizados.

Gobato (26) discutiu os controles monovariáveis e multivariáveis aplicados a sistemas aeroespaciais fracamente ou fortemente acoplados. Este trabalho adaptou parte dos modelos desenvolvidos por Prudêncio (25) na simulação do SCA da PMM no seu Modo Nominal de Operação utilizando e comparando várias leis de controle monovariáveis e multivariáveis. Também foi necessário desenvolver outro SCAO para este satélite, visto que a PMM possuirá estabilização nos três eixos, enquanto o SACI possuiu estabilização por rotação.

Moreira (27) discutiu a análise, projeto e simulação de um controle discreto para a Plataforma MultiMissão e sua migração para um sistema operacional de Tempo Real. O trabalho aplicou parte dos modelos desenvolvidos por Gobato (26) na simulação do SCAO da PMM no seu Modo Nominal de Operação utilizando um processador e um sistema operacional de Tempo Real.

Leite (28) estudou os modos de falhas em sensores e atuadores da PMM e propôs métodos para a sua detecção. Leite (28) ainda estudou estratégias de detecção de falhas: detecção por observadores e por estimadores de estado.

Lustosa (29) analisou a influência de tipos de barramentos (seriais digitais) em sistemas de controle por rede. Lustosa (29) usou os modelos desenvolvidos por Gobato (26), porém os converteu de Matrixx® para Matlab/Simulink®.

2.3 Fundamentos de Modelagem, Teoria de Controle e Análise de Sistemas Usados no Trabalho

Descrever em um único capítulo todos os conceitos de Modelagem, Teoria de Controle e Análise de Sistemas é uma tarefa muito difícil, para não se dizer

impossível. Certamente o entendimento completo desses assuntos só será possível fazendo-se várias consultas às referências indicadas. Porém, é proveitoso uniformizar conceitos (assim como relembrá-los) que serão usados mais à frente. Assim, com esse espírito, alguns conceitos de Modelagem, Teoria de Controle e Análise de Sistemas são apresentados a seguir.

Diagramas de blocos:

Segundo Ogata (30), “o diagrama de blocos de um sistema é a representação figurativa das funções realizadas pelos componentes e o fluxo de sinais”. A Figura 2.4 traz o diagrama de blocos de um sistema com controle realimentado, com uma entrada e uma saída (SISO, do Inglês *Single Input Single Output*).

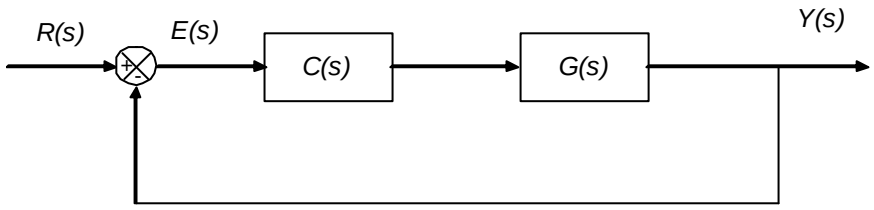

Figura 2.4 – Diagrama de blocos de um sistema com controle realimentado.
Fonte: Ogata (30).

Segue a definição dos elementos da Figura 2.4:

R(s), Referência, é o valor do estado final que o usuário deseja que o seu sistema assuma.

E(s), Erro, é a diferença entre a Referência e o sinal de realimentação.

C(s), Controlador, é o elemento do sistema responsável por conduzir a planta do valor do estado atual até a referência.

G(s), Planta, Segundo Ogata (30), ”é chamado de planta qualquer objeto físico a ser controlado”.

Y(s), Saída, representa o valor corrente do estado da planta.

(s), Transformada de Laplace, indica que a função em questão depende de uma variável no domínio da freqüência, em contraste à dependência de variáveis no domínio do tempo. Segundo Ogata (30), dada uma função *f(t)* dependente de uma variável no domínio do tempo *(t)* é possível encontrar uma

função *F(s)* equivalente só que dependente de uma variável no domínio da freqüência *(s)* através da seguinte expressão:

$$F(s) = \int_0^{\infty} f(t)e^{-st}dt \qquad \textbf{(2.9)}$$

Alternativa e conveniente, a maioria das funções usadas em problemas de controle de sistemas pode ser encontrada em práticas tabelas, fazendo-se rara a recorrência à (2.9).

Controle realimentado:

Segundo Ogata (30), “controle realimentado refere-se a uma operação que, na presença de distúrbios, tende a reduzir a diferença entre a saída de um sistema e uma dada referência através da diferença entre as duas”

Função de transferência de um sistema realimentado:

A função de transferência de um sistema realimentado como o mostrado na Figura 2.4, é relação matemática entre a as Transformadas de Laplace da saída Y(s) e da entrada R(s) com condições iniciais nulas, definida pela seguinte expressão:

$$\frac{Y(s)}{R(s)} = \frac{G(s).C(s)}{1+G(s).C(s)} \qquad \textbf{(2.10)}$$

Sistemas de segunda ordem:

São ditos sistemas de segunda ordem os sistemas cujo grau do polinômio de seu denominador (remeter (2.10) acima) é 2. Ogata (30) apresenta as funções de segunda ordem de uma maneira muito conveniente:

$$\frac{Y(s)}{R(s)} = \frac{w^2{}_n}{s^2 + 2.\xi.w_n.s + w_n{}^2} \qquad \textbf{(2.11)}$$

Onde:

w_n, é a freqüência natural não-amortecida da planta.

ξ, é a taxa de amortecimento empreendida ao sistema pelo controlador e pelas próprias forças do sistema.

Alternativamente, (2.11) pode ser reescrita como:

$$\frac{Y(s)}{R(s)} = \frac{w^2{}_n}{s^2 + 2.\sigma.s + w_n{}^2} \qquad \textbf{(2.12)}$$

Onde, σ, dita atenuação, é definida por:

$$\sigma = 2.\xi.w_n \qquad \textbf{(2.13)}$$

Quando a entrada de um sistema de segunda ordem como o descrito acima é sujeita a uma função de excitação, a resposta do sistema de controle geralmente apresenta oscilações amortecidas antes de alcançar o estado ou regime estacionário. Ao especificar as características de resposta transitória de um sistema de controle, geralmente para uma entrada a degrau, é comum especificar os seguintes indicadores:

a) Tempo de atraso, t_d : é o tempo necessário para a resposta alcançar pela primeira vez a metade do seu valor final;

b) Tempo de subida, t_r : é o tempo necessário para a resposta passar de 10% a 90%, 5% a 95% ou 0% a 100% do seu valor final. Para sistemas de segunda ordem sub-amortecidos, usa-se normalmente o tempo de subida de 0% a 100%. Para sistemas sobreamortecidos, usa-se normalmente o tempo de subida de 10% a 90%;

c) Instante do pico, t_p: é o tempo necessário para a resposta alcançar o primeiro pico de sobre sinal;

d) Sobre-sinal máximo, M_p: o sobre sinal máximo é o máximo valor de pico da curva de resposta medido a partir do seu valor final. Também se usa o máximo sobre-sinal percentual. O sobre-sinal máximo

percentual indica indiretamente a estabilidade relativa do sistema. O sobre-sinal pode ser encontrado pela seguinte relação:

$$M_p = e^{-(\frac{\xi}{\sqrt{1-\xi^2}})*\pi} \quad \textbf{(2.14)}$$

e) Tempo de acomodação, t_s: é o tempo necessário para a curva de resposta alcançar e permanecer dentro de uma faixa em torno do seu valor final, faixa essa de magnitude especificada por uma porcentagem absoluta do valor final (geralmente 5% ou 2%). O tempo de acomodação está relacionado com a constante de tempo do sistema de controle. A escolha de que porcentagem usar no critério de erro pode ser determinada a partir dos objetivos do projeto do sistema em questão.

Veja a Figura 2.5 para identificação gráfica dos elementos descritos acima.

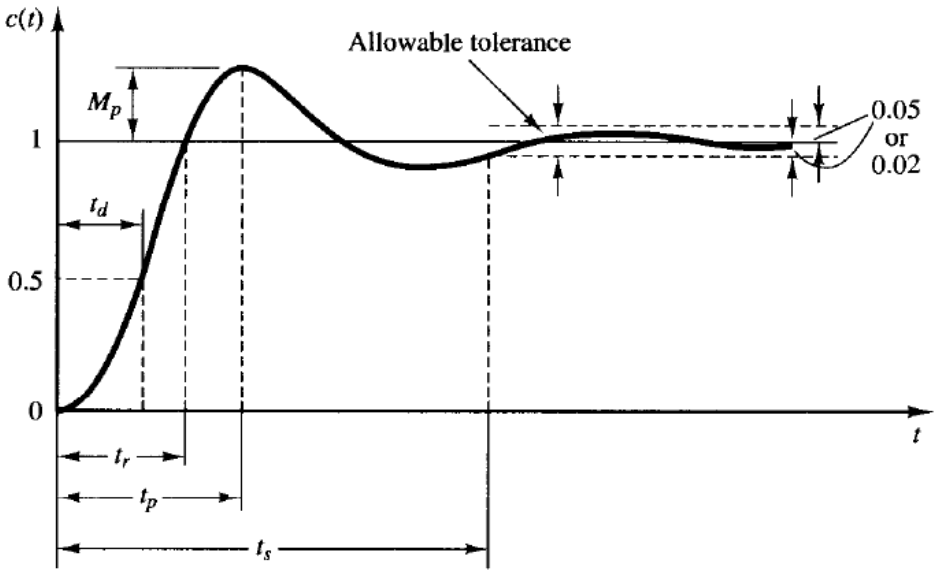

Figura 2.5 – Especificações da resposta transitória a um degrau unitário.
Fonte: Ogata (30).

Controladores PID:

Dado um sistema de segunda ordem com suas características naturais, cabe ao controlador guiar o sistema até o estado requerido pelo seu operador, tanto no que tange ao valor de referência, quanto às características de resposta transitória. O controlador PID (Proporcional, Integral e Derivativo) e suas várias possibilidades podem ser descritos pela seguinte equação:

$$C_{PID}(s) = K_P + sK_D + \frac{1}{K_I s} \qquad \textbf{(2.15)}$$

Onde, os ganhos K_p, K_D e K_I são conhecidos, respectivamente, como ganho proporcional, ganho derivativo e ganho integral. Cada parcela do controlador PID destina-se a controlar uma característica específica do sistema. Segundo Ogata (30):

> A ação proporcional de controle provê uma contribuição que depende do valor instantâneo do erro, diferença entre o valor desejável e o valor real. Um controlador proporcional pode controlar qualquer planta estável, mas oferece performance limitada e erro estacionário não nulo. A ação integral de controle (ou rede "lag") provê uma contribuição que é proporcional ao erro acumulado, implicando em um modo de controle com ação mais lenta. Esse modo, analisado separadamente, tem duas grandes desvantagens: a primeira é o fato do pólo na origem ser altamente prejudicial à estabilidade da malha; a segunda é o fato de dar margem ao efeito de "*wind-up*" (entrada atinge o limite de saturação e continua sendo integrada na malha).
>
> A ação derivativa de controle (ou rede "lead") atua na taxa de mudança do erro, implicando numa ação de controle rápida que desaparece na presença de erros constantes. É muitas vezes chamada de modo preditivo por ser dependente da tendência de variação do erro. A maior limitação do modo derivativo, quando analisado isoladamente, é a geração de ações de controle de amplitude elevada em resposta a erros de freqüência elevada, tais como ruído na medição (30).

2.3.1 Determinação dos ganhos do controlador da PMM

Os modelos da PMM usados nesse trabalho foram baseados no trabalho de Gobato (26). Gobato (26) calculou e comparou ganhos mono e multivariáveis para a PMM: (1) PD e PID (SISO) e (2) alocação de pólos e LQR (MIMO). Para ajuste dos controladores foram supostos como requisitos de projeto as seguintes características transitórias do sistema:

a) $\zeta = 0{,}7$;

Arrazoado: Segundo Gobato (26) é um valor que mantém o compromisso razoável entre velocidade de resposta e sobre-sinal máximo.

b) t_s = 100 s;

Arrazoado: garante uma faixa de tolerância de 2% em aproximadamente quatro vezes a constante de tempo do sistema.

c) Precisão de apontamento < 0,05°;

d) "*Drift*"< 0,001°/s;

e) Determinação de atitude < 0,005° (3 σ);

f) "Off pointing" de até 30º em 180s.

Arrazoado: requisitos da PMM.

2.4 Equações do movimento de um corpo rígido

As equações de movimento de atitude podem ser divididas em dois grupos: as equações cinemáticas de movimento e as equações dinâmicas de movimento. Antes, porém, de se apresentar as equações do movimento, faz-se necessário apresentar-se os referenciais usados no trabalho.

2.4.1 Referenciais

No trabalho desenvolvido por Gobato (26) foram utilizados três referenciais básicos: o primeiro sendo o referencial inercial, o segundo sendo o referencial Vertical Local Horizontal Local (VLHL) e o terceiro sendo aquele fixo no satélite, com origem em seu centro de massa.

Como sugerido por Prudêncio (25), o sistema de coordenadas inercial adotado é o sistema de coordenadas celestial definido relativo ao eixo de rotação da Terra. O Pólo Norte desse sistema (eixo Z) está aproximadamente a 1° da Estrela Polar. Para esse sistema de coordenadas ser completamente definido, deve-se definir o meridiano de referência ou ponto de referência. O ponto no equador terrestre escolhido como referência é o ponto da Eclíptica, ou plano da órbita do Sol ao redor da Terra, que cruza o Equador indo do sul para o norte, conhecido como Equinócio Vernal. Esta é a direção do eixo X , paralela a linha do centro da Terra na direção do Sol no primeiro dia da primavera. O eixo Y é encontrado usando a regra da mão direita, completando o sistema dextrógiro.

O segundo referencial adotado no trabalho de Gobato (26) é o VLHL. Esse é um referencial girante no plano da órbita do satélite cujo sistema de coordenadas tem origem no centro de massa do satélite. O eixo z_o aponta na direção do centro da Terra, eixo y_o aponta na direção normal à órbita e o eixo x_o é obtido pela regra da mão direita, e coincide com a direção do vetor velocidade orbital linear, para uma órbita circular.

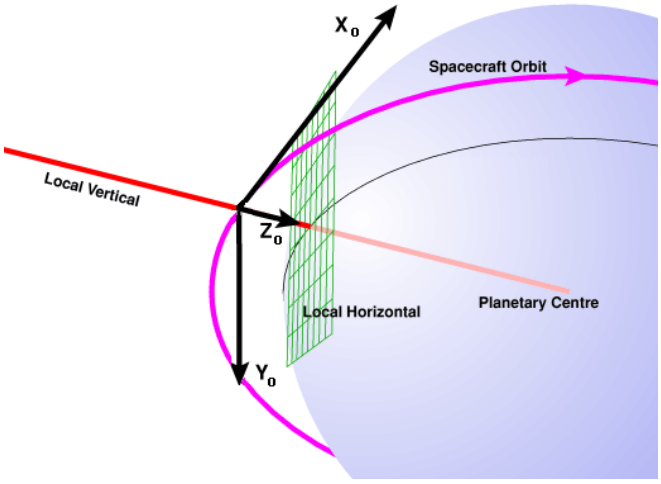

Figura 2.6 – Referencial Vertical Local Horizontal Local (VLHL).
Fonte: Gobato (26).

O terceiro e último referencial adotado no trabalho de Gobato (26) foi o referencial do corpo, ou do satélite, que é um sistema de coordenadas com origem no centro de massa do satélite. Os eixos são escolhidos como sendo coincidentes com os eixos dos momentos principais de inércia. Para estudos de satélite estabilizados em três eixos, Terra-apontado, é prático definir os eixos de "*roll*", "*pitch*" e "*yaw*" como sendo:

a) eixo de "roll" x , nominalmente alinhado com x_o;

b) eixo de "pitch" y , nominalmente alinhado com y_o;

c) eixo de "yaw" z , nominalmente alinhado com z_o.

Onde, por sua vez, em problemas de estabilização de atitude de satélites em três eixos é comum definir:

a) ângulo de "roll" (ϕ) é a integral da velocidade no eixo de rolamento;

b) ângulo de "pitch" (θ) é a integral da velocidade no eixo de arfagem;

c) ângulo de "yaw" (ψ) é a integral da velocidade no eixo de guiagem.

2.4.2 Equações Dinâmicas de Movimento

As equações da dinâmica do movimento de um satélite definem a dependência no tempo das quantidades vetoriais – tais como a velocidade angular – relacionadas com a geometria do movimento frente a perturbações externas – tais como torques perturbadores.

De acordo com Hughes (31), o momento angular de um satélite – considerado nesse trabalho como um corpo rígido – em relação à origem do sistema de coordenadas do corpo coincidente com o centro de massa é:

$$\overrightarrow{H_S} = \int \vec{r} \times (\overrightarrow{\omega_S} \times \vec{r}) dm \qquad \textbf{(2.16)}$$

Onde, $\vec{r}$ é o vetor posição do elemento de massa *dm* em relação à origem e $\overrightarrow{\omega_S}$ o vetor velocidade angular absoluta do satélite.

O desenvolvimento da expressão acima resulta na expressão do momento angular do satélite em termos de suas componentes ortogonais:

$$\overrightarrow{H}_S = h_{Sx}\hat{x} + h_{Sy}\hat{y} + h_{Sz}\hat{z} \qquad \textbf{(2.17)}$$

Onde,

$$h_{Sx} = I_{Sx}\omega_{Sx} - I_{Sxy}\omega_{Sy} - I_{Sxz}\omega_{Sz} \qquad \textbf{(2.18)}$$

$$h_{Sy} = -I_{Sxy}\omega_{Sx} + I_{Sy}\omega_{Sy} - I_{Syz}\omega_{Sz} \qquad \textbf{(2.19)}$$

$$h_{Sx} = -I_{Sxz}\omega_{Sx} - I_{Syz}\omega_{Sy} + I_{Sz}\omega_{Sz} \qquad \textbf{(2.20)}$$

Na forma mais compacta:

$$\overrightarrow{H}_S = I_S \vec{\omega}_S \qquad \textbf{(2.21)}$$

Onde, I_S é definida como a matriz de inércia do corpo.

De acordo com Souza (23), podemos dizer que a taxa de mudança do momento angular, com relação ao seu centro de massa, é igual à resultante dos torques externos em relação à mesma origem. Essa relação define a equação básica da dinâmica de atitude do satélite, equipado com rodas de reação, escrita em coordenadas do referencial inercial como mostrado abaixo:

$$\frac{d\vec{H}}{dt} = \vec{M}_{Ext} \qquad \textbf{(2.22)}$$

onde:

$$\vec{H} = \vec{H}_S + \vec{H}_R \qquad \textbf{(2.23)}$$

Onde, $\vec{H}_R$ representa o momento angular das rodas de reação.

Reescrevendo a equação (2.22) em termos do referencial móvel, que está girando com velocidade angular $\vec{\omega}_S$, temos:

$$\left[\frac{d\vec{H}}{dt}\right] + \vec{\omega}_S \times \vec{H} = \vec{M}_{Ext} \qquad \textbf{(2.24)}$$ [5]

Logo,

$$\dot{\vec{H}}_S + \dot{\vec{H}}_R + \vec{\omega}_S \times (\vec{H}_S + \vec{H}_R) = \vec{M}_{Ext} \qquad \textbf{(2.25)}$$

[5] Os colchetes são usados nessa seção com objetivo de evidenciar que os elementos no seu interior estão em um sistema de referência diferente dos demais elementos da equação.

Mas,

$$\vec{\omega}_R = \vec{\omega}_S + \vec{\omega}_{RS} \tag{2.26}$$

Onde, $\vec{\omega}_{RS}$ é o vetor velocidade angular das rodas de reação em relação ao satélite:

$$\vec{H}_R = \vec{I}_R \vec{\omega}_R = \vec{I}_R \vec{\omega}_S + \vec{I}_R \vec{\omega}_{RS} \tag{2.27}$$

Desta forma,

$$\vec{\dot{H}}_S + \vec{\dot{H}}_R + \vec{\omega}_S \times (\vec{I}_S \vec{\omega}_S + \vec{I}_R \vec{\omega}_S + \vec{I}_R \vec{\omega}_{RS}) = \vec{M}_{Ext} \tag{2.28}$$

$$\vec{\dot{H}}_S + \vec{\dot{H}}_R + \vec{\omega}_S \times [(\vec{I}_R + \vec{I}_S)\vec{\omega}_S + \vec{I}_R \vec{\omega}_{RS}] = \vec{M}_{Ext} \tag{2.29}$$

Observando que o segundo termo $\vec{\dot{H}}_R$ com o sinal trocado opera como o torque de controle gerado pelas rodas de reação sobre o satélite e reordenando-se os termos tem-se finalmente:

$$\vec{\dot{H}}_S + \vec{\omega}_S \times [(\vec{I}_R + \vec{I}_S)\vec{\omega}_S + \vec{I}_R \vec{\omega}_{RS}] = \vec{M}_{Ext} + \vec{\dot{H}}_R \tag{2.30}$$

A equação (2.30) é uma forma de representação da equação de Euler do movimento com relação ao referencial inercial, porém descritas nas coordenadas do satélite, e inclui três fontes de Momento Angular:

a) Momento devido ao produto de $\vec{H}_S = \vec{I}_S \vec{\omega}_S$ para os casos em que $\vec{I}_S$ é não diagonal.

b) Momento devido ao satélite: $\vec{\omega}_S \times [(\vec{I}_R + \vec{I}_S)\vec{\omega}_S]$;

c) Momento devido às rodas de reação: $\vec{\omega}_S \times (\vec{I}_R \vec{\omega}_{RS})$;

O vetor Momento $\overrightarrow{M}_{Ext}$ é a resultante dos torques perturbadores (efeitos ambientais tais como gradiente de gravidade, correntes parasitas, arrasto aerodinâmico e pressão da radiação solar).

2.4.3 Equações Cinemáticas de Movimento

A obtenção das equações cinemáticas do movimento de um satélite, como apresentadas nessa seção, foram desenvolvidas por Gobato (26) e serão resumidas aqui a título de conhecimento.

Formas de descrever orientações de corpos rígidos foram estudadas por Euler, Jacobi, Hamilton, e outros, resultando em uma série de técnicas disponíveis, dentre as quais se podem destacar:

a) Parâmetros Simétricos de Euler (quaternions);

b) Ângulos de Euler;

c) Ângulos do Eixo Equivalente.

Uma boa escolha do sistema de transformação de coordenadas pode simplificar os cálculos matemáticos e prevenir situações como singularidades geométricas ou equações diferenciais cinemáticas não lineares. Para o equacionamento da cinemática do satélite que foi usado como estudo de caso, foi escolhido o método de Ângulos de Euler. Segundo Hughes (31), “Ângulos de Euler são três rotações angulares sucessivas que transformam a referência de um dado sistema cartesiano para outro”.

Considerando três rotações sucessivas ao redor dos eixos de um corpo, pode-se descrever a orientação do referencial U relativo a outro referencial A. Uma seqüência particular de rotações pode ser escolhida e simbolicamente representada por:

$$C_3(\theta_3): W \leftarrow A$$

$$C_2(\theta_2): V \leftarrow W$$

$$C_1(\theta_1): U \leftarrow V$$

onde W e V são dois referenciais intermediários.

Nesse caso cada rotação é descrita por:

$$C_3(\theta_3) = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(2.31)}$$

$$C_2(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \quad \textbf{(2.32)}$$

$$C_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \quad \textbf{(2.33)}$$

Os três ângulos θ_1, θ_2 e θ_3 são conhecidos como Ângulos de Euler.

A seqüência de rotação de A para U no exemplo acima resulta em uma matriz de rotação definida como:

$$C_{UA} \equiv C_1(\theta_1)C_2(\theta_2)C_3(\theta_3) = \begin{bmatrix} c_2c_3 & c_2s_3 & -s_2 \\ s_1s_2c_3 - c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 - s_1c_3 & c_1c_2 \end{bmatrix} \quad \textbf{(2.34)}$$

onde $c_i = \cos(\theta_i)$ e $s_i = \sin(\theta_i)$.

Essa seqüência é definida como seqüência (3-2-1) dos Ângulos de Euler, seguindo a ordem das rotações. Primeiro uma rotação ao redor do terceiro eixo, então uma rotação ao redor do segundo eixo e por fim uma rotação ao redor do primeiro eixo, como mostrado na Figura 2.7 abaixo:

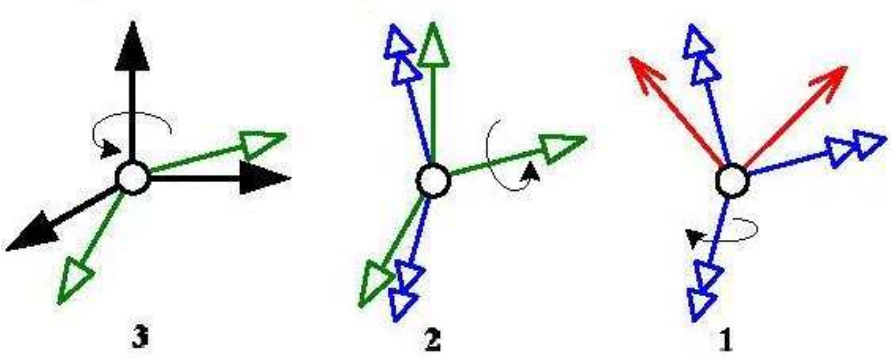

Figura 2.7 – Seqüência de rotação 3-2-1 dos ângulos de Euler.

O estudo da cinemática de um satélite está focado na geometria do movimento, desconsiderando-se os aspectos de massa e força. Essencialmente constitui-se de métodos de cálculo matricial para descrever posições, velocidades e acelerações de corpos rígidos transcritas em diferentes referenciais de coordenadas, acompanhando a evolução da orientação entre os mesmos ao longo do tempo. As equações cinemáticas do movimento são um conjunto de equações diferenciais de primeira ordem que contém o vetor velocidade angular instantâneo $\vec{\omega}$ e especificam a evolução no tempo de parâmetros de atitude (ϕ, θ e ψ).

Considerando o satélite como um corpo rígido em uma órbita circular, define-se o referencial Vertical Local Horizontal Local como aquele cuja origem está no centro de massa do satélite e apresenta o versor descrito no referencial inercial:

$$\hat{V} = \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \end{bmatrix} \tag{2.35}$$

onde:

$\hat{v}_x$ está apontado na direção do movimento orbital;
$\hat{v}_y$ está perpendicular ao plano da órbita;
$\hat{v}_z$ está apontado para o centro da Terra.

A velocidade angular de $\hat{V}$ com relação ao referencial inercial é:

$$\vec{\omega}_{v/I} = -\omega_0 \hat{v}_y \tag{2.36}$$

onde ω_0 é a velocidade orbital.

A velocidade angular do referencial fixo no corpo $\vec{\omega}_S$ é dada por:

$$\vec{\omega}_S = \vec{\omega}_{S/V} + \vec{\omega}_{V/I} = \vec{\omega}_{S/V} - \omega_0 \hat{v}_y \tag{2.37}$$

Para descrever a orientação do referencial fixo no corpo com relação ao referencial Vertical Local Horizontal Local, em termos da seqüência de rotações (3-2-1) dos ângulos de Euler, a seguinte matriz de rotação deve ser usada:

$$C_{VS} \equiv C_1(\theta_\phi)C_2(\theta_\theta)C_3(\theta_\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \tag{2.38}$$

onde:

$c_\phi = \cos(\phi)$ e $s_\phi = \sin(\phi)$;

$c_\theta = \cos(\theta)$ e $s_\theta = \sin(\theta)$;

$c_\psi = \cos(\psi)$ e $s_\psi = \sin(\psi)$.

Ainda, para a seqüência (3-2-1), a velocidade angular do referencial fixo no satélite em relação ao referencial Vertical Local Horizontal Local pode ser representada por:

$$\begin{bmatrix} \omega_{Sx/V} \\ \omega_{Sy/V} \\ \omega_{Sz/V} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \textbf{(2.39)}$$

Como,

$$\vec{\omega}_S = \vec{\omega}_{S/V} + \vec{\omega}_{V/I} = \vec{\omega}_{S/V} - \omega_0 \hat{v}_y \quad \textbf{(2.40)}$$

Tem-se,

$$\begin{bmatrix} \omega_{Sx} \\ \omega_{Sy} \\ \omega_{Sz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \omega_0 \begin{bmatrix} c_\theta s_\psi \\ s_\phi s_\theta s_\psi + c_\phi c_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \end{bmatrix} \quad \textbf{(2.41)}$$

E, finalmente, a equação diferencial cinemática do movimento orbital de um satélite é:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} \omega_{Sx} \\ \omega_{Sy} \\ \omega_{Sz} \end{bmatrix} - \frac{\omega_0}{c_\theta} \begin{bmatrix} s_\psi \\ c_\theta c_\psi \\ s_\theta s_\psi \end{bmatrix} \quad \textbf{(2.42)}$$

2.4.4 Determinação das equações do movimento da PMM

Gobato (26) propôs linearizações para as equações cinemática e dinâmica da PMM, impondo algumas particularidades à simulação final:

a) os modelos são válidos para “pequenos” ângulos (entenda-se por pequeno um ângulo cujo cosseno seja aproximadamente igual á 1);

b) a PMM é modelada como um corpo rígido, sem flexão;

c) exceto pelo torque provocado pelas rodas de reação, os demais torques produzidos por elementos internos da plataforma são nulos;

d) os torques externos de perturbação são conservativos, propõe-se o valor de 1×10^{-4} N.m em cada eixo do satélite;

e) as rodas de reação funcionam em Modo Nominal, ou seja, as três rodas de ““roll””, “pitch” e “yaw”;

f) as rodas de reação funcionam em modo “backup”, ou seja uma das três rodas de ““roll””, “pitch” ou “yaw” falhadas;

Talvez a única limitação para uso dos modelos como proposto por Gobato (26) nesse trabalho seja a simulação de “pequenos” ângulos, cuidado a ser tomado durante os testes de verificação, logo à frente.

2.5 Requisitos de um sistema de controle espacial

Como as próximas seções discutirão em demasia os requisitos de sistemas, faz-se necessário abordar o tema de forma geral antes da discussão prática. A ARP4754 (1) lista as classes de requisitos mais importantes para um sistema:
Os tipos de requisitos detalhados abaixo devem ser considerados nas várias fases do desenvolvimento (i.e, ao nível funcional, de sistemas, componentes e hardware e software):

a) Requisitos de Segurança: requisitos atrelados à definição de Segurança dada anteriormente;

b) Requisitos Funcionais: são combinações de necessidades dos clientes, restrições operacionais, restrições regulatórias e implementações práticas;

c) Requisitos do Cliente: esses requisitos podem incluir aqueles associados com a carga paga almejada pelo cliente, práticas de operação, conceitos de manutenção e funções desejadas;

d) Requisitos Operacionais: esses requisitos definem as interfaces entre a tripulação e sistemas funcionais (Nota do autor: ou no caso espacial, operador em Terra e satélite), pessoal de manutenção e cada sistema do avião, e várias outras pessoas de suporte do avião e funções ou equipamentos relacionados;

e) Requisitos de Desempenho: incluem especificidades das funções tais como: acurácia, fidelidade, espectro, resolução, velocidade e respostas no tempo;

f) Requisitos Instalativos e Físicos: incluem: tamanho, provisão para montagem, potência consumida, calor dissipado, peso, ventilação, restrições ambientais, acesso, manipulação e armazenamento;

g) Requisitos de Manutenabilidade: incluem tarefas de manutenção agendadas ou inesperadas;

h) Requisitos de Interface: incluem características das informações trocadas e dos meios físicos de interface;

i) Requisitos de Certificação: requisitos escritos pelas autoridades regulatórias;

Claramente a lista de requisitos acima foi baseada em sistemas aeronáuticos, assim, algumas considerações são necessárias para o seu aproveitamento no caso espacial.

Requisitos de Segurança: Segurança no caso da ARP4754 baseia-se na definição de Safety da AC 25.1309, que estabelece níveis de Disponibilidade e Integridade para os sistemas. Assim, para o caso espacial, este item será coberto por requisitos de Disponibilidade e Integridade.

Requisitos de Manutenabilidade: certamente não têm o mesmo sentido para os casos espacial e aeronáutico. Para o caso espacial, podem ser interpretado como tarefas periódicas agendadas pela estação de Terra para execução de rotinas de diagnóstico e detecção de falhas.

Requisitos de Certificação: Não são aplicáveis a sistemas espaciais.

Quanto às demais classes de requisitos (i.e. requisitos de interface, funcionais, do cliente, operacionais, de desempenho e instalativos) são aplicáveis a sistemas espaciais com a mesma definição dada pela ARP4754.

Da lista de requisitos sugerida pela ARP4754 – e descrita acima – serão estudados por esse trabalho os seguintes requisitos: de Disponibilidade e de Desempenho.

Faz-se necessário explicar as classes de requisitos escolhidas. Logicamente, apesar de sua inegável importância para o desenvolvimento do projeto, os demais requisitos serão deliberadamente negligenciados por este trabalho, por conta da limitação do seu escopo e duração. O extrato de requisitos proposto é o mínimo necessário para a construção de estratégias de tolerância a falhas, mas não exaure o conjunto de requisitos suficientes para se completar um projeto. Porém, como prova da metodologia proposta, eles são suficientes. Não se pode, por exemplo, propor quantas unidades redundantes se queiram em um sistema espacial para simplesmente se cumprir os requisitos de Disponibilidade. Faz-se necessário cuidar do peso, potência elétrica consumida, calor dissipado, entre outros. Assim, as soluções propostas aqui atenderão os requisitos selecionados, mas em um projeto completo deveriam ser escrutinadas por vários outros requisitos.

3 REVISÃO HISTÓRICA DE SOLUÇÕES TOLERANTES A FALHAS EM PROJETOS ESPACIAIS

Como base para a proposição de uma arquitetura tolerante a falhas, sempre é de muita utilidade estudar as soluções já disponíveis em outros projetos, aprender com seus acertos e erros.

Os projetos da agência espacial americana, NASA, têm normalmente uma vasta bibliografia disponível nos arquivos eletrônicos da agência. Um dos projetos pioneiros da agência, Gemini[6], contava com um computador simplex, com muitos comandos sendo iniciados pelo próprio piloto. Segundo o manual de familiarização da Gemini (32):

> O OBC (*On Board Computer*) provê a capacidade necessária de armazenamento e processamento de dados para guiagem e controle. Um seletor de modos determina o tipo de processamento a ser executado. Um botão de START permite que o piloto inicie certas rotinas sob sua decisão (32).

Àquela época não havia recursos computacionais com a mesma Disponibilidade que se tem hoje em dia. Para se ter uma idéia da tecnologia da época, ainda segundo o manual de familiarização da Gemini (32), “O módulo de reentrada ATM IV é carregado no OBC (o carregamento demora menos de 40 minutos). Os giros e acelerômetros requerem cerca de meia hora para aquecimento e outra meia hora para alinhamento e estabilização”.

---

[6] Projeto da agência espacial americana, NASA, para vôos tripulados em baixas altitudes. Precedeu o projeto Apollo.

O projeto Apollo, que sucedeu o Gemini, aproveitou o legado de seu antecessor e adotou a mesma arquitetura. Segundo Lala et al (4), o lançador Saturno V era controlado por um computador com redundância tripla, enquanto que o módulo lunar era guiado por um computador simplex.

O projeto Galileo[7] contava com duas linhas de processamento que funcionavam continuamente em paralelo. Essas duas linhas contavam ao todo com 6 computadores de bordo (33).

Segundo a sua especificação técnica (34), o satélite GOES I-M[8] contava com dois controladores eletrônicos de atitude e órbita (AOCE 1 e 2), além de um módulo de Segurança, independente dos primeiros, que recebe comandos da estação-Terra ou é acionado pelo AOCE quando uma anomalia é detectada.

A nave Vikings[9] possuía um sistema duplex, totalmente redundante, tanto no *Lander* quanto no *Orbiter*. Segundo Tomayko (35):

> Os processadores eram entrelaçados, assim, em caso de falhas eles poderiam ser renomeados. Os documentos de requisitos gerados pela JPL chamavam esse tipo de redundância de “Tolerância a Falha Simples”, na qual um computador tinha um *backup*, tornando possível a redistribuição de tarefas em caso de falhas. Na prática, os dois conjuntos de computadores eram muito úteis, porque, às vezes, havia

[7] Projeto da agência espacial americana, NASA, não tripulado para a exploração de Júpiter.
[8] GOES (do inglês, *Geostationary Operational Environmental Satellites*) é um projeto da agência americana, NASA, para observação e previsão climáticas.
[9] Vikings é um projeto da agência espacial americana, NASA, não tripulado para a exploração de Marte.

> muito para um único computador fazer. Os requisitos especificavam três modos de operação: 1) Individual, onde cada computador podia trabalhar em eventos diferentes; 2) paralelo, onde os computadores trabalhavam no mesmo evento; e 3) *tandem*, onde os computadores trabalhavam no mesmo evento e as saídas eram votadas (35).

Em contraste à simplicidade dos computadores da Apollo e da Gemini, segundo Hanaway e Moorehead (36), o *Space Shuttle* conta com cinco computadores para controle de atitude, sendo que um deles é usado exclusivamente como *backup* dos outros quatro.

A Tabela 3.1 traz um resumo do histórico da evolução das arquiteturas dos sistemas de controle de atitude e de órbita.

Tabela 3.1 - Histórico de arquiteturas de sistema de controle de atitude e órbita.

Projeto	Arquitetura	Tipo da missão [tripulada/não tripulada]	Ano lançamento
Gemini	Simplex	Tripulada	1965~1966
Apollo	Simplex	Tripulada	1969
Vikings I e II	Duplo-simplex	Não tripulada	1975
Space Shuttle	Quadri-simplex com um módulo *backup*.	Tripulada	1981~1982
Galileo	Duplo-triplex	Não tripulada	1989
GOES I-M	Duplo-simplex com um módulo *backup*	Não tripulada	2001

4 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO

Como já exposto anteriormente, este trabalho propõe-se a estudar o método de tolerância a falha aplicado a sistemas de controle aeroespaciais. Como tolerância a falha é um meio e não um fim em si, antes de se propor a aplicação do método é necessário investigar quais são as características do sistema (requisitos) que se quer modificar, melhorar, promover através do método de tolerância a falhas. O método de tolerância a falha pode ser aplicado a todas as partes do sistema, mas a sua colocação equivocada pode causar um efeito contrário ao desejado, degradando os requisitos em questão. Assim, se faz necessário conhecer bem o sistema a ser usado e sua reação ao método. Portanto, como seqüência de trabalho, propõe-se conhecer um pouco mais do projeto de estudo de caso, elicitar os seus requisitos e falhas aplicáveis – levantando aqueles em que a tolerância a falha pode atuar – e propor pontos específicos do projeto candidatos à aplicação do método de tolerância a falhas.

O projeto da Plataforma MultiMissão (PMM) do INPE foi escolhido como estudo de caso para a missão espacial. A seguir, nas próximas seções, o projeto será detalhado.

4.1 Descrição da PMM

A Plataforma MultiMissão (PMM) é um conceito moderno em arquitetura de satélites e consiste em reunir em uma única plataforma versátil os equipamentos essenciais à operação do satélite, independente de sua órbita e de sua missão específica (definida pelos sensores do módulo de carga útil). Nesta arquitetura existe uma separação física entre a plataforma e o módulo de carga útil, possibilitando que ambos sejam desenvolvidos, construídos e testados separadamente, antes da integração e teste final do satélite. A Figura 4.1 ilustra a Plataforma MultiMissão em sua configuração em órbita.

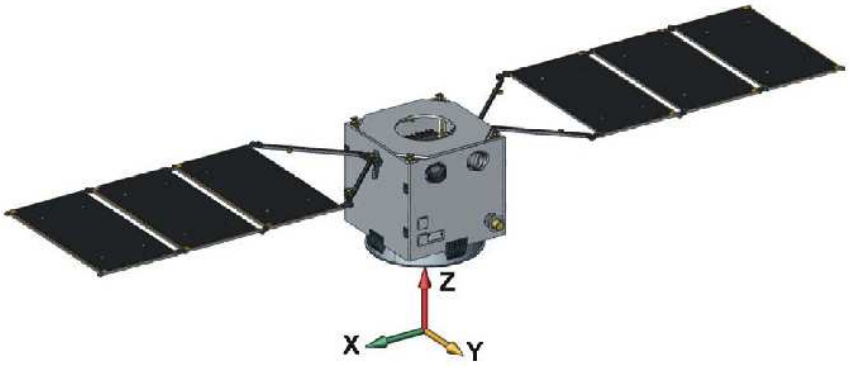

Figura 4.1– PMM mostrada em configuração em órbita.

Fonte: Manual de Especificação técnica PMM (37).

A plataforma tem como finalidade o apoio a diversas atividades de observação terrestre, ciência e comunicação em LEO (do Inglês *Low Earth Orbits*) como ilustrado na Figura 4.2.

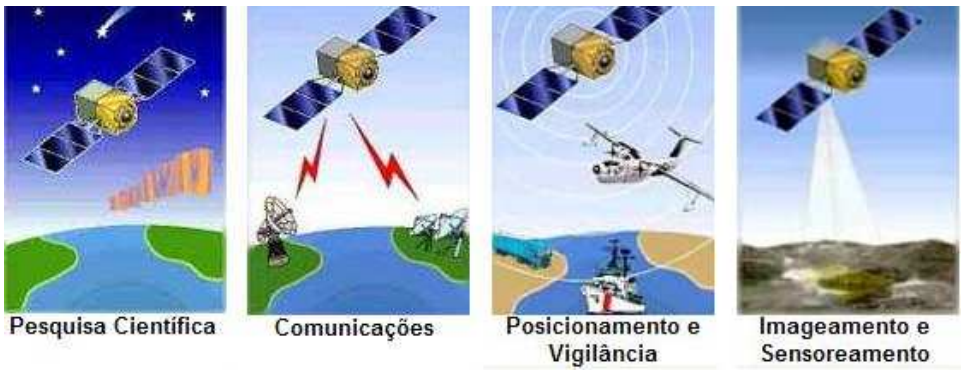

Figura 4.2 – Ilustração de aplicações da PMM.

A PMM é constituída dos seguintes subsistemas:

a) Subsistema Estrutural: provê suporte mecânico para os demais subsistemas, *hardware* e acessórios;

b) Subsistema de Suprimento de Energia Elétrica: converte energia solar incidente em energia elétrica através de células fotovoltaicas, armazenando-a em baterias e suprindo energia para as várias cargas úteis;

c) Subsistema de Controle Térmico: promove distribuição térmica adequada para que os equipamentos embarcados operem dentro dos limites de temperatura especificados;

d) Subsistema de Controle de Atitude e Gerenciamento de Dados: provê controle de atitude e órbita estabilizado em três eixos, permitindo atitudes de apontamento para a Terra, o Sol e Inercial. Esse subsistema também provê processamento de dados e capacidade de armazenamento através do computador de bordo;

e) Subsistema de Propulsão: provê meios de aquisição e manutenção de órbita usando o mono-propelente Hydrazina;

f) Subsistema de Telemetria e Telecomando: provê comunicação entre a plataforma e estações de Terra.

4.2 Descrição dos Subsistemas de Controle de Atitude e Gerenciamento de Dados

O subsistema de Controle de Atitude e Gerenciamento de Dados, também conhecido como "*Attitude Control and Data Handling (ACDH)*", implementa as seguintes funções:

a) Gerenciamento de dados a bordo do satélite;

b) Controle de atitude e órbita, que implementa as seguintes funções:

- Controle de atitude estabilizado em três eixos no Modo Nominal, permitindo apontamento para Terra, "Anti-Terra", Inercial e para o Sol;
- Transição do controle para o Modo de Contingência;
- Controle de atitude no Modo de Contingência: visa a aquisição segura de atitude após a fase de lançamento ou após alguma falha;
- Controle do posicionamento dos painéis solares;

- Determinação a bordo do satélite da posição e velocidade;
- Controle dos propulsores para aquisição de órbita e manutenção de órbita;
- Dessaturação das rodas de reação através das bobinas eletromagnéticas;
- Dessaturação das rodas de reação através dos propulsores.

A arquitetura do subsistema ACDH é baseada em um computador de bordo modular (OBC, sigla que Inglês significa *On board Computer*) que faz interface (aquisição de dados e comando) com o sistema de controle de atitude e outros subsistemas e cargas pagas. Veja Figura 4.3.

Abaixo, são nomeados os componentes do subsistema ACDH:

a) Computador de Bordo ("*On Board Computer (OBC)*")

b) Sensores

- Magnetômetros: cada um dos dois magnetômetros instalados provê medição do campo magnético em três eixos;
- Unidade Inercial: provê velocidade angular nos três eixos ;
- Sensores Solares: o conjunto de oito sensores solares provê informação suficiente para determinação da direção do Sol em três eixos com cobertura total do céu;
- Sensores de Estrelas: cada um dos dois sensores de estrelas provê informação de atitude em três eixos autonomamente;
- GPS: cada uma das duas unidades GPS é composta por um receptor e suas antenas, e provê hora, posicionamento e velocidade do satélite a bordo e autonomamente.

c) Atuadores:

- Bobinas Magnéticas: o conjunto de três bobinas magnéticas prove torque magnético de controle em três eixos;

- Rodas de Reação: o conjunto de quatro rodas de reação prove controle de atitude em três eixos.

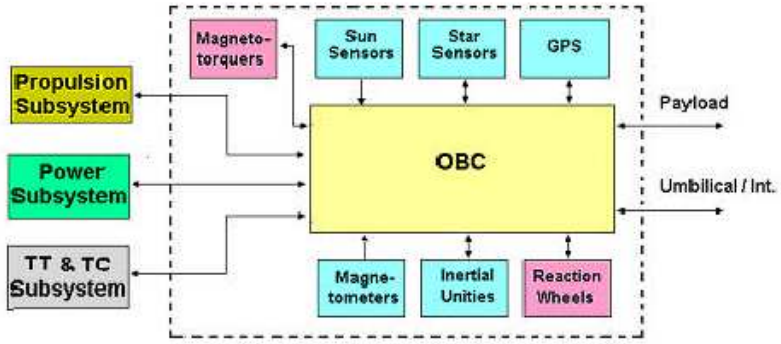

Figura 4.3 – Representação esquemática do OBC e suas várias interfaces.

Fonte: Relatório de requisitos técnicos da PMM (38).

4.2.1 Computador de Bordo

Segundo o relatório de requisitos da PMM (38), o computador de bordo tem as seguintes funções:

a) Processamento de comandos para controle de atitude;

b) Processamento de telecomandos;

c) Processamento de comandos de telemetria;

d) Aquisição de dados;

e) Controle de atuadores;

f) Comunicação serial digital com outros subsistemas;

g) Controle de propulsão;

h) Controle dos *magnetotorquers*;

i) Conversão de tensão DC/DC

4.2.2 Modos de Operação da PMM

O comportamento do satélite pode ser descrito através de seus diferentes Modos de Operação, os quais estão associados com a configuração de seus equipamentos e seu monitoramento. Esses modos são divididos em:

a) Vôo

i. Modo de Inicialização (STM);
ii. Modo de Contingência (COM);
iii. Modo de Navegação fina (FNM);
iv. Modo Nominal (NOM)
v. Modo de Dessaturação das rodas através dos propulsores (WDM);
vi. Modo de Correção de Órbita (OCM);
vii. Modo de Correção de Órbita Backup (OCMB);

b) Solo

i. Modo Desligado (OFM);
ii. Modo de Integração e Testes (ITM);

A

Figura 4.4, abaixo, mostra o diagrama de fluxo mostrando a transição entre os modos descritos acima:

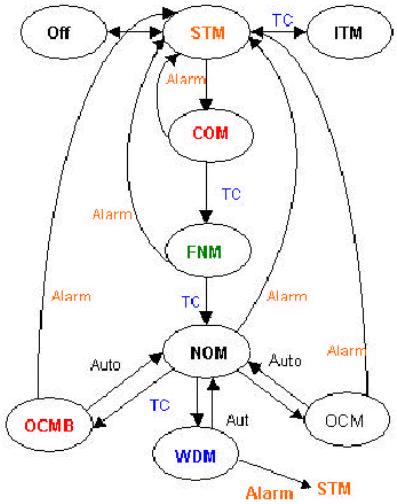

Figura 4.4 – Diagrama de estados que ilustra a transição dos modos da PMM.

Fonte: Relatório de requisitos técnicos da PMM (38).

Em Modo Nominal, a PMM será configurada de forma que sua carga útil cumpra sua missão. A atitude do satélite é mantida em direção ao alvo. Normalmente o satélite permanece nesse modo até que ocorra uma falha e entre no modo STM.

4.2.3 Configuração da PMM usada neste trabalho

Para os fins deste trabalho, será usada uma configuração simplificada da PMM, baseado no trabalho de Lustosa (27), que, por sua vez, fora baseado no trabalho de Gobato (26).

Sensores:

A configuração simplificada deste trabalho da PMM conta com uma unidade giroscópica (enquanto que o projeto real conta com duas unidades) e com um sensor de estrelas (o mesmo número que a plataforma real). Como explica Gobato (26), “Um modelo simplificado dos sensores foi utilizado nesse trabalho. Os mesmos foram modelados por um ganho adicionado de um ruído, ruído este que, para as simulações, foi considerado de magnitude $1x10^{-5}$”.

Os demais sensores da PMM não foram usados neste trabalho. O modelo do giroscópio tem como entradas ω_{Sx}, ω_{Sy} e ω_{Sz}, e, como saída as tensões $V\omega_{Sx}$, $V\omega_{Sy}$ e $V\omega_{Sz}$, proporcionais às velocidades angulares do satélite em cada um dos eixos. O modelo do sensor de estrelas tem como entradas ϕ, θ e ψ, relacionando o referencial do satélite ao referencial VLHL, e como saída as tensões $V\phi$, $V\theta$ e $V\psi$, proporcionais a essas grandezas.

Atuadores:

Como meios de controle da atitude são usadas as rodas de reação, 3 no total, uma para cada eixo, mais uma roda ortogonal, assim como na PMM. Os demais meios de atuação da PMM não foram usados neste trabalho. As rodas de reação foram modeladas como aproximações lineares da curva característica de um servomotor de corrente contínua. O modelo das rodas de reação utilizadas nesse trabalho tem como entradas as tensões V_{Rx_s}, V_{Ry_s}, V_{Rz_s} e V_{Rs_s} geradas pelo controlador; e, como saída, as velocidades angulares das rodas ω_{Rx}, ω_{Ry}, ω_{Rz} e ω_{Rs}; os torques gerados pelas rodas $\dot{h}_{Rx}$, $\dot{h}_{Ry}$, $\dot{h}_{Rz}$ e $\dot{h}_{Rs}$; e os torques totais gerados nos três eixos, acrescidos da parcela de torque gerado pela roda ortogonal, $\dot{h}_{Rx\_t}$, $\dot{h}_{Ry\_t}$ e $\dot{h}_{Rz\_t}$.

Assim, em linhas gerais, o modelo da PMM a ser usado neste trabalho é como apresentado no diagrama genérico da Figura 4.5. Cada um dos blocos (variáveis de controle, contexto e significado) será detalhado mais adiante:

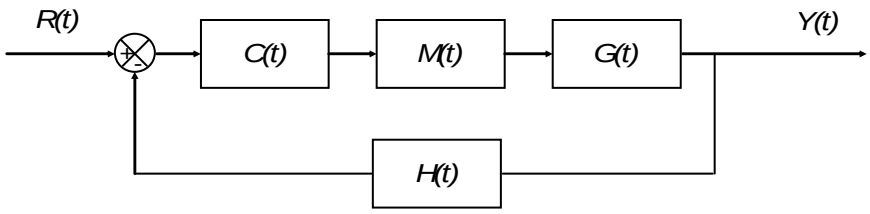

Figura 4.5 – Esquemático da PMM usado neste trabalho.

Depois de desenvolvido em MATLAB o diagrama da Figura 4.5, o modelo da planta (PMM), controlador, sensores e atuadores é como visto na Figura 4.6:

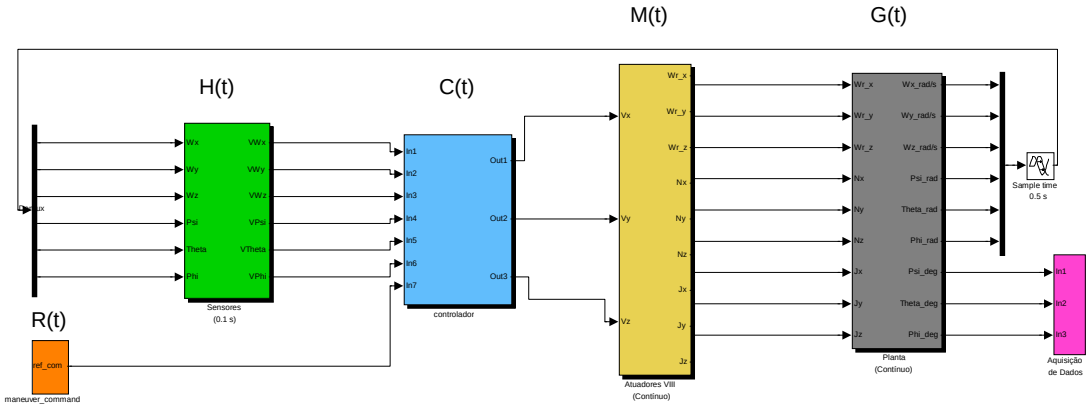

Figura 4.6 – Modelo da PMM, sensores, atuadores e controladores usados no trabalho.

onde:

R(t), bloco de referência: a entrada é dada em termos de um ângulo de referência (em graus). É importante lembrar as limitações do modelo, que por construção, deverá ter como entrada apenas pequenos ângulos. No modelo em MATLAB/Simulink® da Figura 4.6, é o bloco em alaranjado.

- Entrada do bloco: comando (ângulo) em graus;
- Saída do bloco: comando (ângulo) em radianos;

C(t), bloco do controlador: o controlador recebe a diferença entre a referência e o valor atual medido pelos sensores. No modelo em MATLAB® da Figura 4.6, é o bloco em azul.

- Entrada do bloco: erro (ângulos) nos três eixos, *roll, yaw* e *pitch;*
- Saída do bloco: tensões de controle das rodas de reação, V_x, V_y e V_z

M(t), bloco dos atuadores: o controlador age sobre os atuadores, gerando os torques de controle do satélite. No modelo em MATLAB® da Figura 4.6, é o bloco em amarelo.

- Entrada do bloco: tensões de controle;
- Saída do bloco: torques de controle;

G(t), planta: Este bloco recebe os torques calculados no bloco dos atuadores e entrega como saída a posição do satélite, através das equações dinâmicas de Euler.

- Entrada do bloco: torques de controle;
- Saída do bloco: atitude da PMM;

H(t), sensores: Esse bloco é responsável por simular os sensores de atitude da plataforma: sensor de estrelas e giroscópio.

- Entrada do bloco: atitude do satélite;
- Saída do bloco: tensões equivalentes às taxas de variação de posição em cada um dos eixos, e a posição do satélite através do sensor de estrelas.

4.3 Requisitos do estudo de caso – PMM

A seguir, serão apresentados os requisitos do estudo de caso. A referência entre colchetes representa a fonte do requisito, para garantir a rastreabilidade da informação.

Requisitos de Disponibilidade

[MPP-R-1: A822000-DPK-1 (37), Issue D5, pág. 20**]** A Plataforma MultiMissão deve ter Confiabilidade maior que 0,8 (taxa de sucesso) considerando-se uma vida útil de 4 anos (35040 horas).

[MPP-R-2: A822700-SPC-01/04 (39), pág. 38**]** O sistema de controle de atitude e manipulação de dados (ACDH) deve ter apresentar Confiabilidade maior que 0,9462 (taxa de sucesso) considerando-se uma vida útil de 4 anos (35040 horas).

Nota: uma taxa de sucesso de 0,9642 em 35040 horas, corresponde a uma taxa de falha de aproximadamente 1,0 x 10^{-6} falha/hora, que é o resultado do seguinte cálculo:

Taxa de falha = 1 – Taxa de Sucesso

$1 - 0{,}9642 = 0{,}0358$, *taxa de falha em 35040 horas*

$\frac{0{,}0358}{35040} = 1{,}0 \times 10^{-6}$ *, taxa de falha por hora*

[MPP-R-3: A822700-SPC-01/04 (39), pág.38**]** O sistema de controle de atitude deve ser totalmente redundante.

Nota: "falha" nesse requisito e dentro do contexto do trabalho será entendida como o repertório de falhas como definido na seção 2.1.2.1.

[MPP-R-4: A822700-SPC-01/04 (39), pág.38**]** O sistema de controle de atitude deve eliminar todos os pontos de falhas simples.

[MPP-R-5: A822000-DPK-1 (37), Issue D5, pág.20**]** A Plataforma MultiMissão deve ser imune a perdas de operação devido a falhas simples.

[MPP-R-6: A822700-SPC (38), Issue 02/03, pág.11**]** Na presença de falhas, o computador de Bordo deve ter a capacidade de se reconfigurar automaticamente.

Requisitos de Desempenho

[MPP-R-7: A822000-DPK-1 (37), Issue D5, pág. 11 e Gobato (26)**]** Em Modo Nominal, a atitude do satélite deve ser controlada nos três eixos para cumprir com os seguintes requisitos:

a) Erro de determinação de atitude menor que 0,005° (3σ);
b) Taxa de amortecimento $0{,}3 < \xi < 0{,}8$;
c) t_s = 100s (5% do valor final);
d) "*Drift*" menor que 0,001°/s;

Em primeira análise da lista de requisitos apresentados, alguns se destacam por nortearem a solução a ser proposta. Os requisitos MPP-R-4 e 5 demandam a eliminação de falhas simples, enquanto que o requisito MPP-R-3 declara que qualquer que seja a solução aplicada para o atendimento desses requisitos, ela

deve empregar redundância. O requisito MPP-R-3 é uma indicação clara da preferência do cliente por tolerância a falhas como método de aumento de Dependabilidade. Embora por um lado seja questionável a declaração de uma solução como em MPP-R-3, pois elimina todas as outras possíveis soluções (porque não se usar prevenção a falhas, por exemplo?), por outro lado agiliza a escolha da arquitetura final. Além disso, MPP-R-3 pode representar a experiência do cliente que já operou outras soluções anteriormente que não o tenham atendido.

O requisito MPP-R-2 estabelece um rígido valor de Disponibilidade para o sistema: 0,9642 ou 1,0 x 10^{-6} falha/h. Ainda mais se levar-se em conta que um microprocessador convencional tem uma taxa de falha de aproximadamente 1,0 x 10^{-4} por hora (veja explicação no **Apêndice A**). Certamente a redundância será essencial na melhoria desse número.

Antes de se prosseguir vale a pena uma última ressalva. Tanto o requisito MPP-R-4 quanto 5 fazem alusão a “falhas simples”, mas o que quer dizer “falhas simples”? Para este projeto o repertório de falhas simples é aquele apresentado anteriormente, na seção 2.1.2.

4.3.1 Análise quantitativa dos requisitos do estudo de caso – PMM

Como análise preliminar das possíveis soluções para os requisitos apresentados, sugere-se uma avaliação dos valores de Disponibilidade exigidos usando-se a Árvore de Falhas. De acordo com Souza e Carvalho (6):

> O método de análise por Árvore de Falhas usa lógica *Booleana* para mostrar a relação entre o efeito e os modos de falha. As duas portas lógicas mais comuns nessa análise são portas E’s e OU’s. Uma porta E representa uma condição na qual a coexistência de todas as entradas é requerida para se produzir uma saída que representa um evento. Uma porta OU representa a condição na qual uma ou mais entradas são necessárias para a realização de um evento. Esta análise usa probabilidade para avaliar se uma configuração particular ou arquitetura cumprirá com os requisitos impostos (6).

A última frase extraída de Souza e Carvalho (6) resume bem a estratégia, adotada por hora: “Esta análise usa probabilidade para avaliar se uma configuração particular ou arquitetura cumprirá com os requisitos impostos.”

A Figura 4.7 e a Figura 4.8 trazem trechos da Árvore de Falhas cujo evento topo é a perda da PMM. Note que, por mera conveniência a probabilidade do topo e dos ramos é em termos de “falha” e não “sucesso”, como especificado nos documentos da PMM. Porém, basta lembrar que a probabilidade de falha é o complemento da probabilidade de sucesso ($P_{falha} = 1 – P_{sucesso}$). Os MTBFs dos componentes foram extraídos de documentos de requisitos preliminares da PMM ((37), (38) e (39)). As árvores de falhas foram construídas com o *software* CAFTA®, ferramenta que é muito utilizada na indústria aeronáutica. Para maiores detalhes das probabilidades usadas remeter-se ao **Apêndice A** (incluindo a árvore completa). A Figura 4.7 apresenta a Árvore de Falhas cujo evento topo é “Perda da PMM”. As taxas de falhas usadas foram baseadas na especificação da PMM (37), (38) e (39), as taxas de falhas foram baseadas em catálogos e experiências passadas. Para mais detalhes, veja o **Apêndice A**.

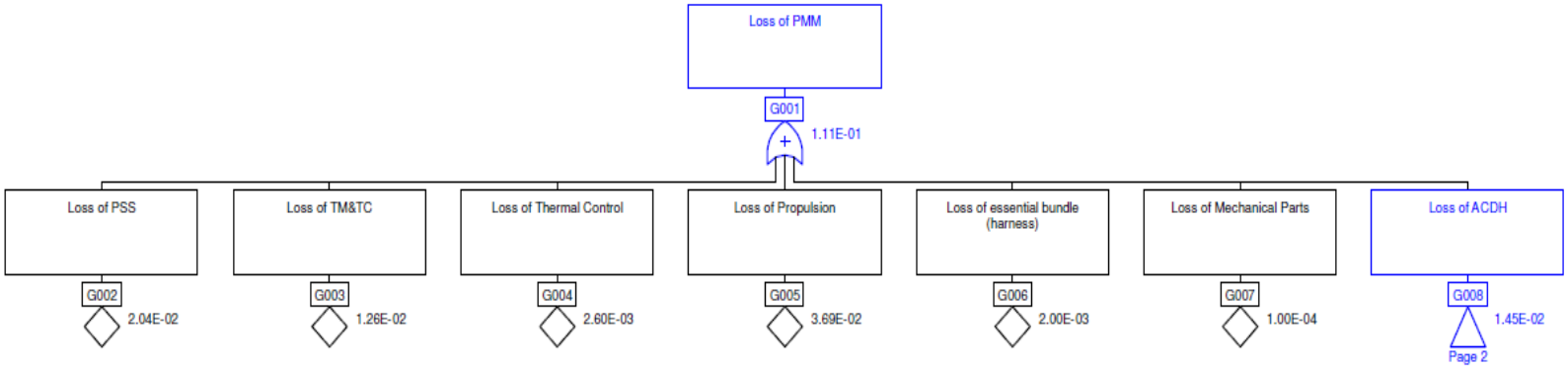

Figura 4.7 – Árvore de Falhas com o evento topo mostrando a probabilidade de perda associada à PMM.

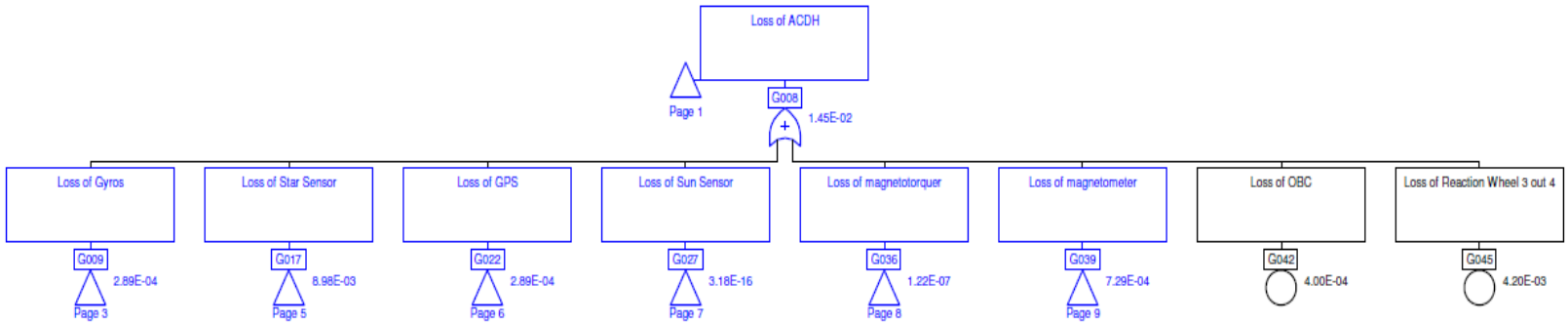

Figura 4.8 – Árvore de Falhas com o evento topo mostrando a probabilidade de perda associada ao sistema de ACDH. Essa árvore é um ramo da árvore mostrada na Figura 4.7.

A construção das árvores de falhas da Figura 4.7 e Figura 4.8 considerou as seguintes hipóteses:

- PMM operando no Modo Nominal;

- Não se levou em consideração nenhum tipo gerenciamento de sinais para substituição de sensores falhados;

- Não se considerou a intervenção da estação em Terra para detectar e mitigar falhas;

Alguns pontos são notórios à primeira análise das árvores apresentadas nas Figura 4.7 e Figura 4.8:

a) A maior probabilidade de falha é associada ao sistema de ACDH, de onde se pode concluir que o sistema de ACDH seria o principal candidato em uma campanha de melhoria da Dependabilidade da PMM, seguido de perto pelo sistema de propulsão;

b) Olhando-se em mais detalhe dentro do sistema de ACDH, se nota que o OBC é o principal fator para limitação da Dependabilidade;

c) Pelos números associados à perda do OBC (da ordem de $1x10^{-2}$ falha/4 anos ou $1x10^{-2}$ falha/35040 horas ou $1x10^{-6}$ falha/h) e levando-se em conta que um computador comercial tem Disponibilidade da ordem de $1x10^{-4}$ falha/h (veja o **Apêndice A** para mais detalhes), conclui-se que essa primeira análise considerou computadores operando em completa redundância; ou seja, no caso da perda de um deles, o outro assume inteiramente as suas funções;

d) Não foi considerado comando errôneo de nenhum dos sistemas envolvidos.

Desenvolvendo-se um pouco melhor o item “d” acima citado, é praxe se atribuir duas falhas a microprocessadores: falha e comando errôneo. O comando errôneo é um erro provocado por uma operação equivocada do microprocessador, que por sua vez pode ser causada por qualquer falha de um

dos processos envolvidos no processamento do dado (erro de agendamento, endereçamento errôneo, perda de tarefa, etc.). Fala-se em erro, pois diferente da falha de projeto, o comando errôneo é um estado passageiro, e não uma falha do projeto do microprocessador. O comando errôneo poderia levar à perda da missão caso, por exemplo, o OBC comandasse os propulsores à toda carga em uma dada direção e sem que houvesse tempo da estação em Terra intervir. A análise do efeito do comando errôneo deve ser considerada logo nos primórdios do projeto e o seu resultado afetará diretamente a arquitetura do sistema. Para os fins didáticos desse trabalho (principalmente considerando-se que, por agora, não há meios de se fazer uma análise do efeito de um comando errôneo do OBC na PMM), considerar-se-á que o comando errôneo leva à perda da missão.

Levando em conta que o comando errôneo por computador seja da ordem de $1x10^{-5}$ falha/h, se forem considerados computadores simples como candidatos à solução, a probabilidade de perda do OBC ficaria limitada à ordem de $1x10^{-5}$ falha/h (a perda de qualquer um dos dois levaria à perda da missão) e se quebraria a premissa original de considerar a perda do OBC da ordem de $1x10^{-6}$ falha/h (item 3 acima). Ao passo que se forem considerados múltiplos computadores que tenham a capacidade de validar uns os resultados de cálculos dos outros, a probabilidade de comando errôneo diminui à razão que se usam mais computadores na validação dos cálculos: $1x10^{-10}$ falha/hora para dois computadores, $1x10^{-15}$ falha/hora para três computadores e assim por diante.

Vale notar aqui que a diminuição da probabilidade do comando errôneo pela adição de mais computadores para comparação pode diminuir a Disponibilidade. Considere, por exemplo, o caso de um computador simples. A sua Disponibilidade seria de $1x10^{-4}$ falha/h e a probabilidade de um comando errôneo seria de $1x10^{-5}$ falha/h. Ao se adicionar um segundo computador para se checar os resultados do primeiro a probabilidade de comando errôneo cai para $1x10^{-10}$ falha/h – uma vez que ambos os computadores devem errar ao

mesmo tempo para o conjunto errar – mas Disponibilidade do conjunto também diminui para a metade, $2x10^{-4}$ falha/h, uma vez que qualquer um dos computadores que falhar levará à perda do conjunto. Assim, não se deve adicionar indiscriminadamente redundâncias para se resolver o problema do comando errôneo sem se cuidar da Disponibilidade.

Ao final dessa discussão sobre comandos errôneos sente-se a falta na lista de requisitos de uma especificação sobre os comandos errôneos. Assim, se propõe o seguinte requisito:

[MPP-R-8] O sistema de controle de atitude e manipulação de dados (ACDH) deve apresentar uma probabilidade de comando errôneo menor que $1x10^{-6}$ falha/h.

Trocando-se em miúdos o requisito MPP-R-8, demanda que o comando errôneo não seja um entrave à Disponibilidade geral do sistema de ACDH, tornando válida a premissa original da Disponibilidade de $1x10^{-6}$ falha/h.

4.4 Proposta de soluções para os requisitos apresentados

Baseado nos requisitos e na análise hora apresentados, delineiam-se a seguir algumas características de alto nível das soluções a serem apresentadas:

a) Para se cumprir com os requisitos MPP-R-2 e 3 e a hipótese “c” levantada na seção 4.3.1, a solução deve ter, no mínimo, dois canais;

b) Para se cumprir os requisitos MPP-R-7 e 8 os dois canais devem se automonitorar, para evitar o comando errôneo e permitir a autodetecção de falhas. Essa característica sugere o uso de soluções do tipo *self checking pairs* onde um canal é responsável pelo

comando direto dos atuadores enquanto o outro monitora os cálculos e saídas do primeiro;

c) Para se satisfazer o item “b” acima apresentado e o requisito MPP-R-4, a solução deve contar com, no mínimo, 3 computadores, para que ao mesmo tempo em que se tenha autoverificação dos comandos, não se permita que uma falha simples cause a perda do sistema. Com apenas dois computadores se monitorando, à primeira falha, como não se sabe qual computador está correto, desligam-se os dois.

A primeira solução sai do próprio item “c” acima, a triplo-simplex, ou seja, três computadores simples interligados entre si para permitir o monitoramento mútuo.

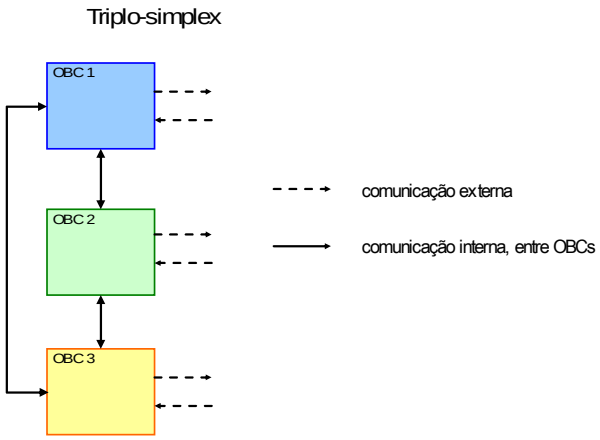

Figura 4.9 – Arquitetura triplo-simplex.

A idéia da arquitetura com três computadores é que somente um esteja em controle por vez, enquanto os outros ficam em *hot standby*. Os três computadores trocam entre si os seus cálculos e, usando um sistema de comparação, decidem a validade dos comandos. Assim, se uma falha ocorrer em qualquer um dos computadores (mesmo que esse esteja em *standby*) provocando a diferença entre as comparações, os outros computadores têm capacidade de detectar o computador falhado e isolá-lo. Note que essa configuração só tolera uma falha desse tipo (desacordo de valores entre computadores), pois a segunda falha nas comparações leva ao desligamento

dos dois computadores remanescentes, pois não se saberia qual entre os dois estaria correto.

Uma segunda solução pode ser derivada da primeira. A idéia de se usar três computadores com capacidade de comando é interessante, pois aproveita ao máximo a capacidade computacional do sistema. Porém, os projetistas teriam que acomodar três módulos computacionais distintos em um espaço diminuto do satélite. Uma maneira de otimizar a solução com três computadores é adotar o mesmo princípio do satélite GOES I-M (veja Tabela 3.1), um sistema duplo simplex mais um módulo de Segurança como *backup*. Nessa opção, que como a anterior ainda conta com três computadores, o duplo simplex é arranjado em um esquema de comando e monitoramento conhecido como *self-checking pairs*. Só a linha de comando do duplo-simplex tem a capacidade de controlar os atuadores, enquanto que ao monitor fica atribuída a função de monitoramento da linha de comando. Comando e monitor podem compartilhar recursos, como placas de I/O (do Inglês *Input and Output*), alimentação elétrica, monitores, etc., fazendo com que seja um pouco mais compacta que a anterior. Com o módulo duplo-simplex no comando, fica atribuído ao módulo de Segurança o papel de *hot-standby*. Qualquer falha detectada no módulo duplo o comando é transferido para o módulo de Segurança. A Figura 4.10 ilustra a solução duplo-simplex mais o módulo de Segurança.

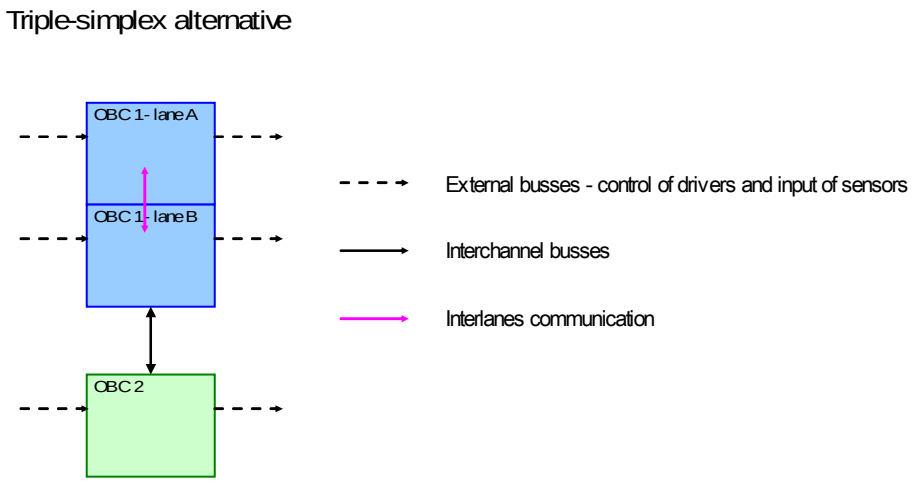

Figura 4.10 – Arquitetura duplo-simplex mais módulo de Segurança.

A seguir, serão mais bem detalhadas as soluções propostas. Discutindo-se com mais profundidade suas entradas, processamento e saída, é possível se detectar falhas de modo comum que poderiam arruinar as propostas. É muito comum em sistemas embarcados, pequenos detalhes de implementação comprometerem a estratégia de redundância adotada em alto nível.

4.4.1 Detalhamento da arquitetura triplo-simplex

4.4.1.1 Entradas de dados (Arquitetura de um votador de dados simples)

As entradas de dados são pontos fundamentais da arquitetura, pois como um dos princípios da arquitetura triplo-simplex é a comparação dos dados de saída, tem-se que reduzir ao máximo possível as diferenças entre os canais, começando pela entrada. Garantindo-se que as entradas dos dados nos canais são próximas, as saídas também o serão, a menos dos atrasos inerentemente diferentes dos três computadores e de seus erros de truncamento. Como proposto por Gobato (26) e já descrito anteriormente, o modelo da PMM usado neste trabalho usa um giroscópio e um sensor de estrelas como sensores de entrada. Porém, como sugerido por Lamport et al (21), e demandado pelos requisitos PMM-R-3, 4, 5 e 6, não é possível identificar um número “m” de falhas (ditos “traidores” no trabalho de Lamport et al (21)) sem ao menos 3 vezes “m” elementos usados para identificação. Isso se levar-se em conta o algoritmo de Mensagens Escritas, como descrito na seção 2.1.2.6. Assim, propõe-se para a entrada da solução triplo-simplex o uso de, no mínimo, 3 sensores diferentes, de tal forma que a falha de um deles possa ser identificada e isolada.

Como alternativa à configuração da PMM proposta por Gobato (26), como segunda fonte de dados da atitude pode-se se usar o segundo módulo giroscópico. Falta ainda a terceira fonte. Para tal, propõe-se usar uma redundância funcional, como proposto por Patton (22), sintetizando-se a tensão nos três eixos (V_x, V_y e V_z) através da saída do sensor de estrelas. O sensor de estrelas tem, como saída, a atitude do satélite: ϕ, θ e ψ já no referencial do satélite, uma vez que o modelo proposto por Gobato (26) já fez a rotação do

referencial VLHL para o referencial do satélite. Assim, $\dot{\theta}$, $\dot{\psi}$ e $\dot{\phi}$ podem ser obtidos através de derivação dos ângulos de atitude ϕ, θ e ψ. Os ângulos de atitude na saída do bloco de sensores (H(t), vide Seção 4.2.3) no modelo da PMM já estão amostrados, o que facilita a proposta de uma operação de derivação:

$$\Delta\theta(k) = \frac{\theta(k) - \theta(k-1)}{\Delta t} \quad \textbf{(4.1)}$$

$$\Delta\phi(k) = \frac{\phi(k) - \phi(k-1)}{\Delta t} \quad \textbf{(4.2)}$$

$$\Delta\psi(k) = \frac{\psi(k) - \psi(k-1)}{\Delta t} \quad \textbf{(4.3)}$$

Onde,

Δ é o operador diferença;

t é o período de amostragem;

k é a amostra no instante atual;

k-1 é a amostra no instante anterior;

Segue ilustrado na Figura 4.11 o esquemático da entrada de dados na proposta triplo-simplex:

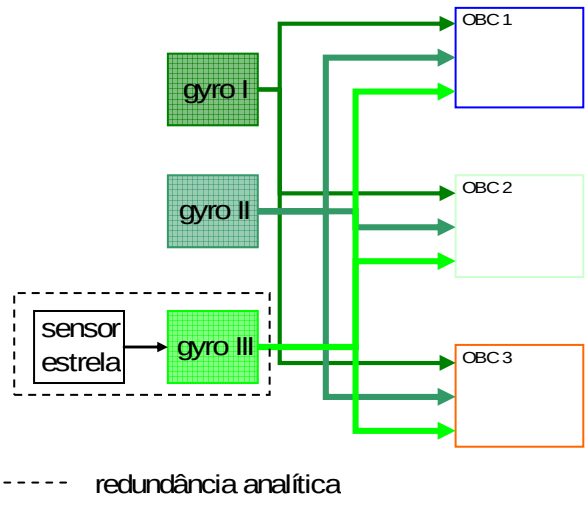

Figura 4.11 – Esquema de entrada de dados da configuração triplo-simplex. Os barramentos entre os módulos foram suprimidos propositadamente para melhor entendimento do esquema de entrada.

A implementação do esquemático da Figura 4.11 em MATLAB/Simulink® é mostrado na Figura 4.12:

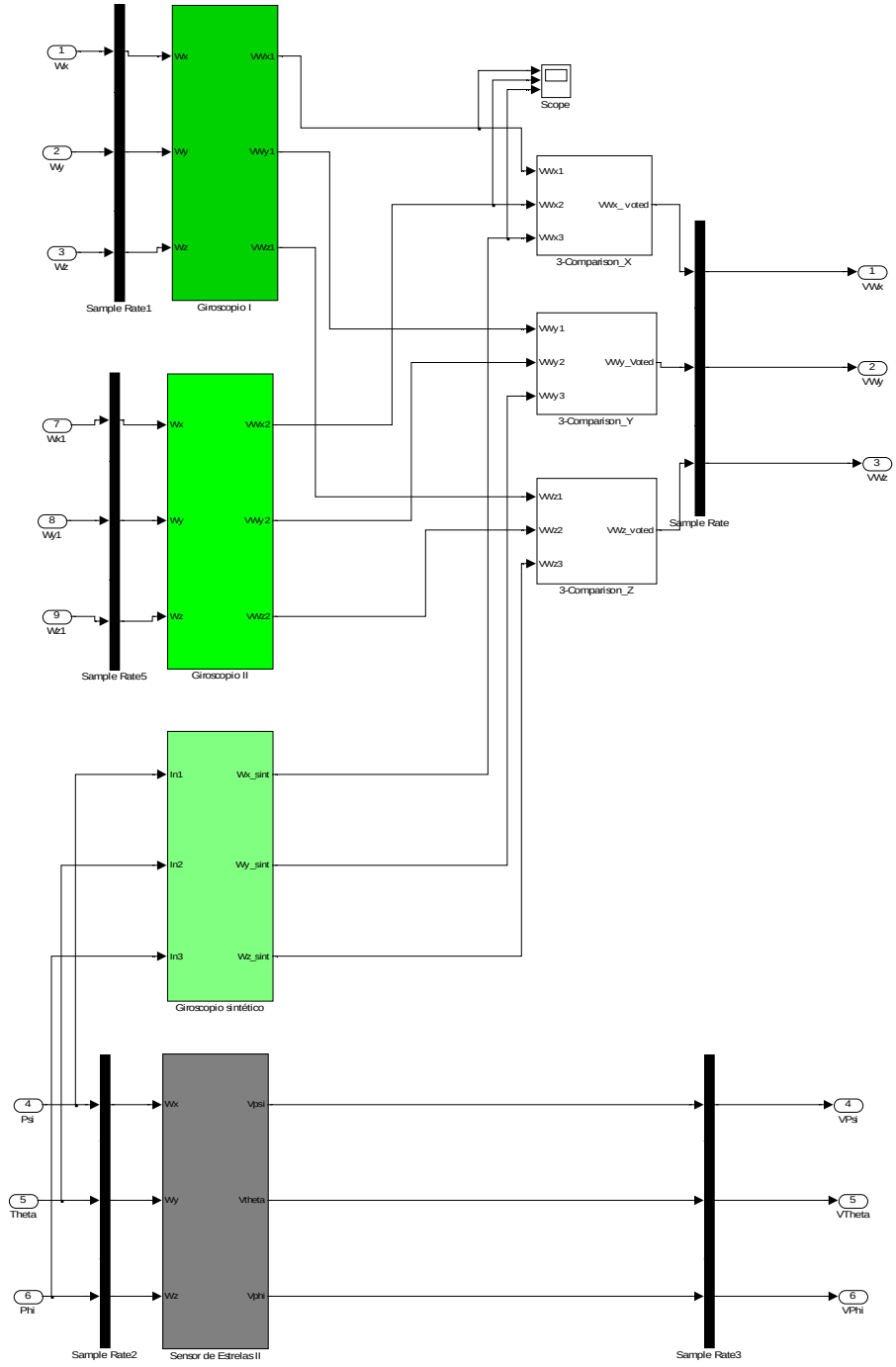

Figura 4.12 – Implementação em MATLAB/Simulink® do esquema de votação das entradas de dados da arquitetura triplo-simplex. Caminho dentro do modelo: \pmm\sensores\.

A Figura 4.13 traz a resposta a degrau da simulação da implementação da Figura 4.12:

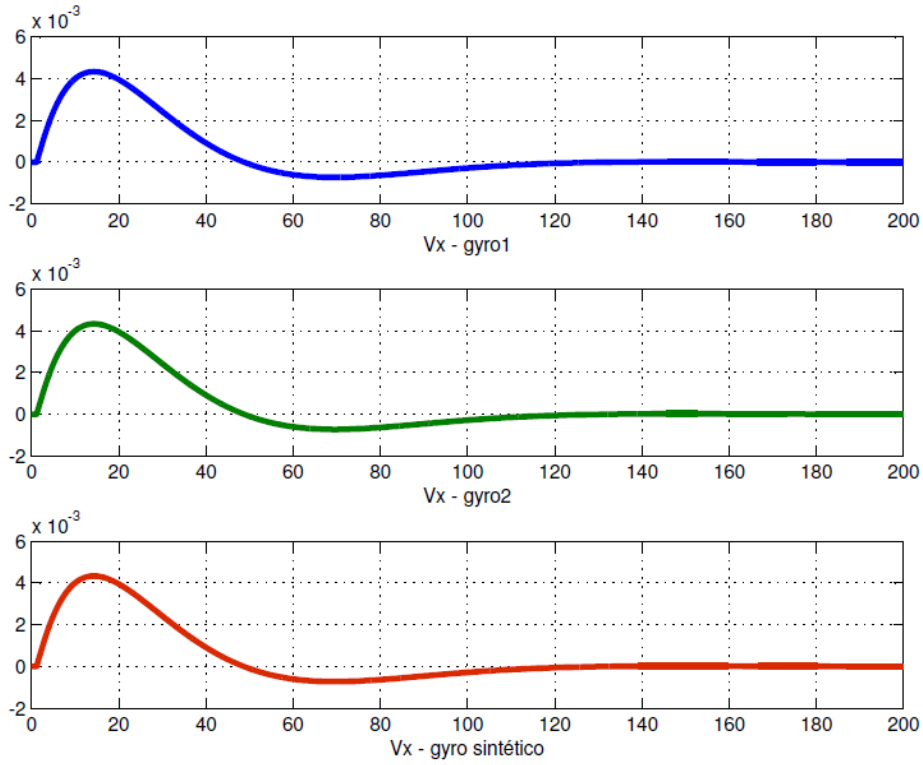

Figura 4.13 – Comparação entre a síntese (entrada degrau de 5º) de $\dot{\theta}$ e a simulação dos sensores físicos.

Depois de adquiridos, já dentro de cada um dos OBC's, os dados de entrada são votados, para se identificar possíveis dados faltosos. A votação pode ser dividida em duas etapas: identificação de falhas e isolamento. A identificação é feita pela comparação das fontes uma a uma. Em seguida, os resultados das comparações individuais são comparados mais uma vez, promovendo o isolamento. Assim, se uma mesma fonte estiver diferente de duas outras fontes, esta fonte diferente será declarada falhada. Segue o algoritmo de comparação de dados na

Figura 4.14 e sua implementação em MATLAB/Simulink® na Figura 4.15:

```
COMEÇO

compare dado1 e dado2
Se comparação 1 e 2 > limite
   Então flag de falha1 = 1
Fim Se

compare dado2 e dado3
Se comparação 2 e 3 > limite
  Então flag de falha2 = 1
Fim Se

compare dado1 e dado2
Se comparação 1 e 3 > limite
  Então flag de falha3 = 1
Fim Se


Se flag de falha1 E flag de falha2 = 1
  Então flag de falha dado2 = 1
Fim Se

Se flag de falha1 E flag de falha3 = 1
  Então flag de falha dado1 = 1
Fim Se
```

Figura 4.14 – Algoritmo de comparação e isolamento de falhas, implementado na entrada das soluções.

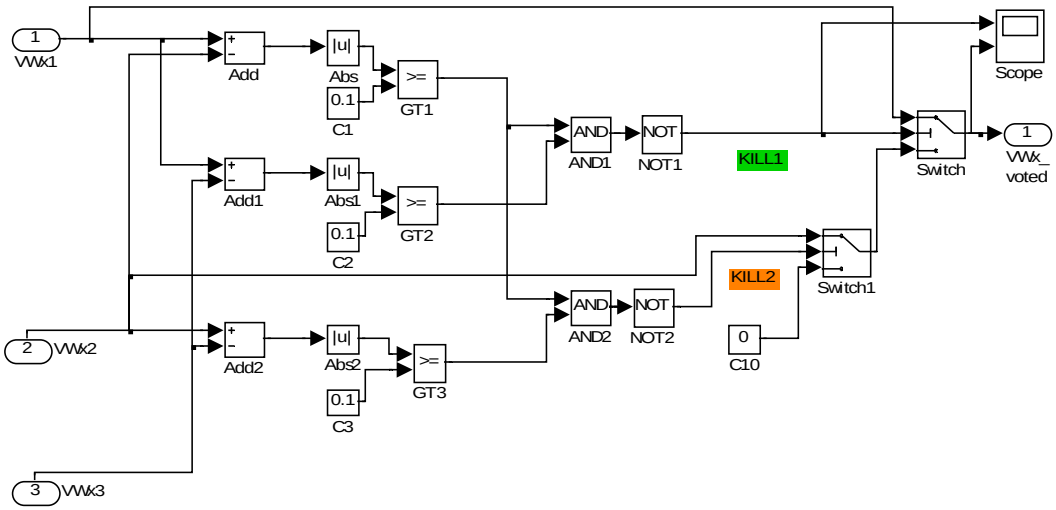

Figura 4.15 – Esquema de votação de dados com identificação e isolamento de falhas.

No primeiro estágio da comparação, para a determinação da falha propriamente dita, comparam-se os dois dados contra um padrão de diferença (ou limite de comparação). Esse padrão é o dito *threshold* do monitor. Qualquer diferença maior que esse valor será considerada uma falha. Mais à frente, quando a solução duplo-simplex estiver sendo detalhada, será mostrado como se determina o valor do *threshold*.

Quando o sistema não tem falhas, a fonte 1 de dados (vinda do giroscópio 1) é preferencialmente escolhida. Poder-se-ia ter selecionado outras das duas fontes como padrão, mas, por simplicidade, escolheu-se a fonte 1. Se a fonte 1 falhar, a 2 será escolhida. A fonte 3 de dados, que é a fonte de dados sintetizados, nunca é usada para cálculos, só para monitoramento. A fonte de dados sintetizados poderia ser usada, desde que provado que as características desse novo dado fazem estimativa tão boa quanto se queira do dado real. A Figura 4.16 traz um teste simples que comprova o funcionamento do votador proposto.

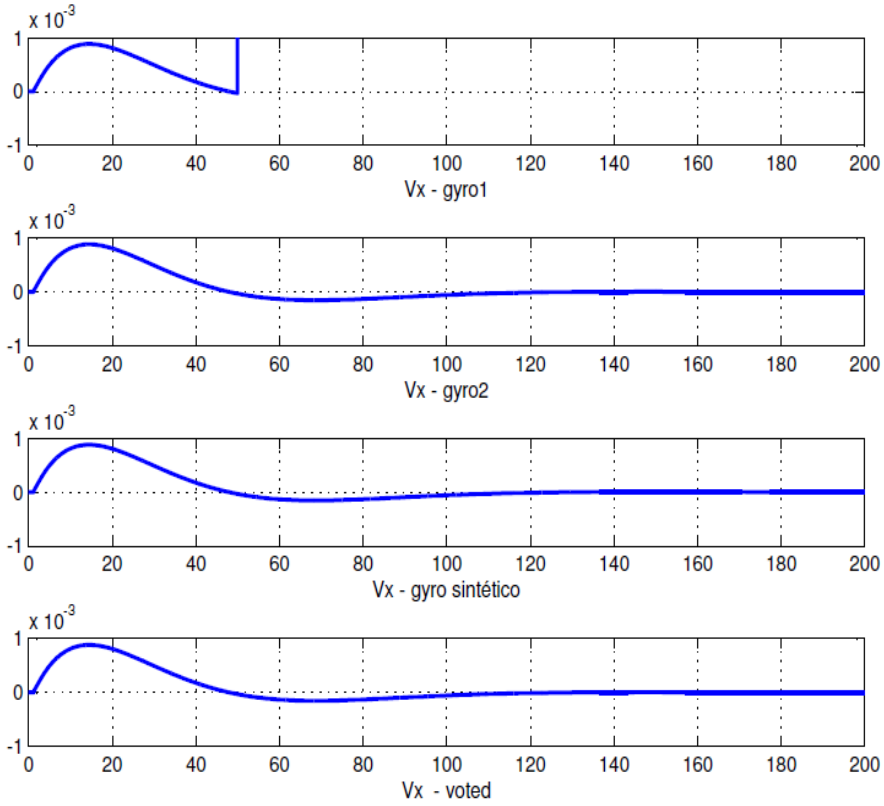

Figura 4.16 – Votador de dados em operação. Falha inserida na fonte 1 de dados.

Uma falha em forma de degrau foi introduzida na fonte 1. Note que se olhando somente a saída do votador, que é o dado que servirá de entrada do controlador – gráfico (d) da Figura 4.16 – a falha lhe foi transparente. Para o controlador que usa essa saída, é como se nada tivesse acontecido. É claro que o comutador adotado, extraído da biblioteca do MATLAB/Simulink®, é instantâneo e não representa um comportamento real.

Há outros algoritmos disponíveis para votação de entrada de dados, como o MVS (do Inglês *Mid-Value Select Voter*), descrito por Krstic et al (40), que pela riqueza do assunto, valeriam um trabalho inteiro por si.

Com o uso do votador de 3 fontes, garante-se que uma falha simples (mesmo nos moldes de uma falha Bizantina) será detectada. Note que o algoritmo como descrito por Lamport et al (21) requer 3m elementos desde que as mensagens sejam escritas, ou seja, que tenham autenticidade garantida. Como os dados do giroscópio e do sensor de estrela são transmitidos via barramento digital

serial, propõe-se a adoção de um *checksum* ou bit de paridade tão comum aos protocolos. Assim, antes do votador usar os dados em qualquer comparação, validar-se-ia a Integridade do dado.

4.4.1.2 Processamento e saída de comandos

Após os dados serem entregues ao controlador, passa-se ao cálculo dos comandos de controle de atitude. O mesmo comando será calculado nos três computadores (OBC1, OBC2 e OBC3). Ao final, esses comandos serão inter cambiados pelos canais e comparados. O processo de cálculo dos comandos não apresenta nenhuma novidade em comparação ao proposto por Gobato (26).

O processo de comparação dos comandos se dá em dois níveis: interno e externo (fórum). Para que a comparação aconteça, cada canal se comunica com os outros por barramentos de comunicação exclusivos, como mostrado na Figura 4.17. Os canais de comunicação são chamados de ICDL:

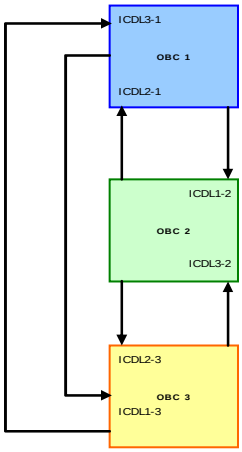

Figura 4.17 – Detalhe da comunicação entre canais da solução triplo-simplex.

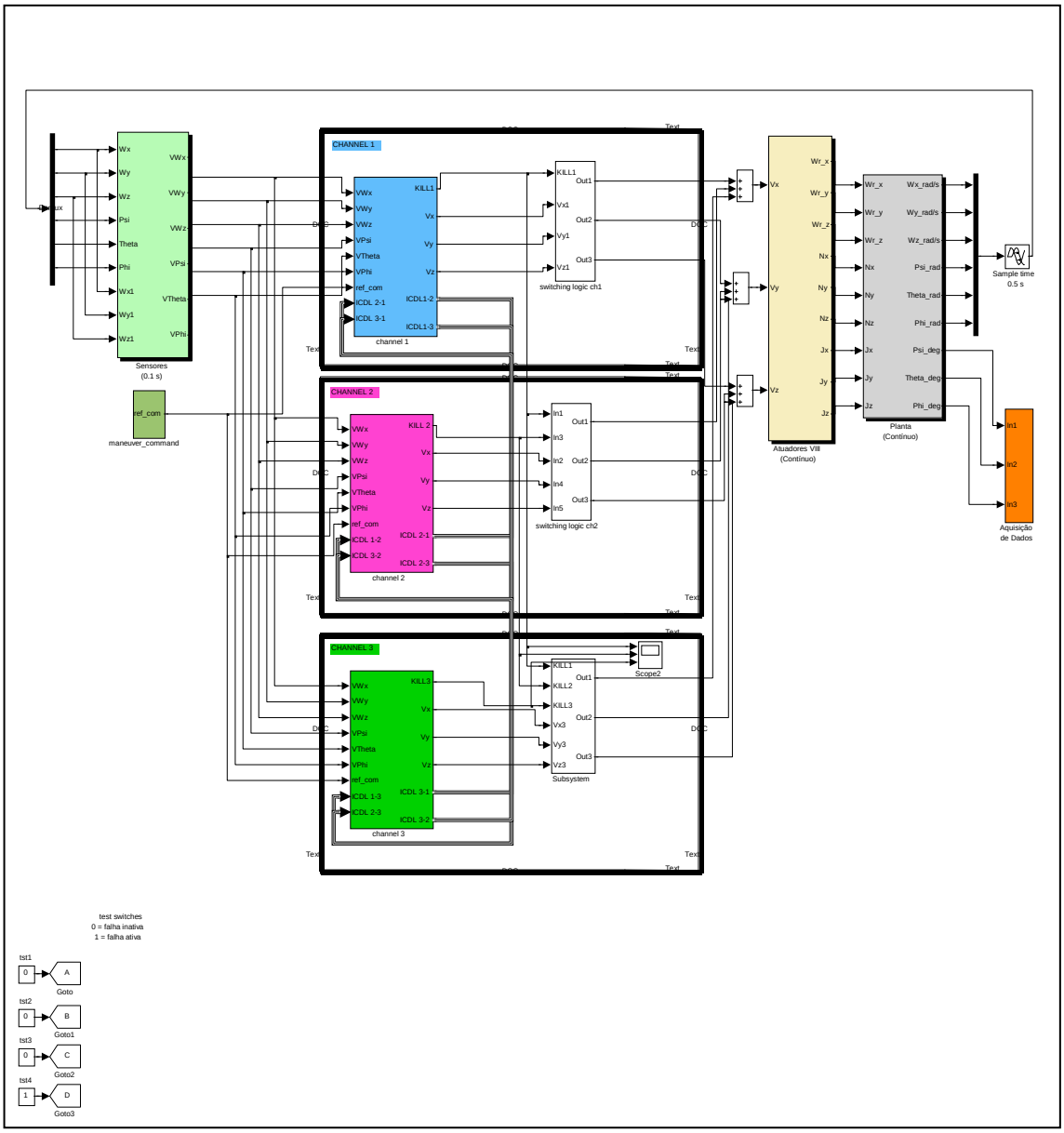

Figura 4.18 – Implementação da arquitetura triplo-simplex em MATLAB/Simulink. Localização no modelo: \pmm\.

A Figura 4.18 apresenta o nível mais alto da implementação em MATLAB/Simulink® da solução triplo-simplex. Cada um dos canais está marcado em cores diferentes no centro da figura. Note à esquerda dos canais, em um bloco verde claro, está o votador de dados apresentado na seção anterior. Os barramentos de dados são as linhas duplas na figura ligando cada um dos canais.

O primeiro nível de comparação de dados é o interno. Uma vez cada canal tenha calculado a sua própria versão do comando, os canais os compartilham

entre si através dos ICDLs (sigla para *Interface Channel Data Link*). Assim, o canal terá sua própria versão do comando e aquela versão calculada pelos outros canais. A comparação interna então confronta o comando calculado pelo canal contra aquele calculado pelos outros canais. Essa comparação usa o mesmo algoritmo para votação de dados de entrada já mostrado na seção anterior (para o caso da votação dos dados de entrada). Se qualquer um dos comandos do canal é declarado falhado (V_x, V_y ou V_z) o canal inteiro é declarado falhado, mesmo que os comandos dos outros eixos estejam bons. Feita a comparação interna, os resultados da comparação de cada canal são compartilhados mais uma vez entre os canais.

Sempre que dois canais concordarem que um terceiro canal está falhado, este será desligado. Assim, nunca uma falha simples determinará o desligamento do canal. Cada canal tem um meio próprio para se autodesligar e um meio alternativo ao barramento de dados (um sinal discreto, por exemplo) para que os outros canais o desliguem (representados por linhas simples que cruzam os canais na Figura 4.18). A razão de não se usar o barramento de comunicação para o desligamento é que se pode advogar que a mesma falha que afetou o processamento pode ter também afetado o desligamento. Assim, usando meios alternativos, garante-se que os outros dois canais têm a capacidade de desligar o terceiro. A Figura 4.19 detalha a implementação da arquitetura interna da solução triplo-simplex.

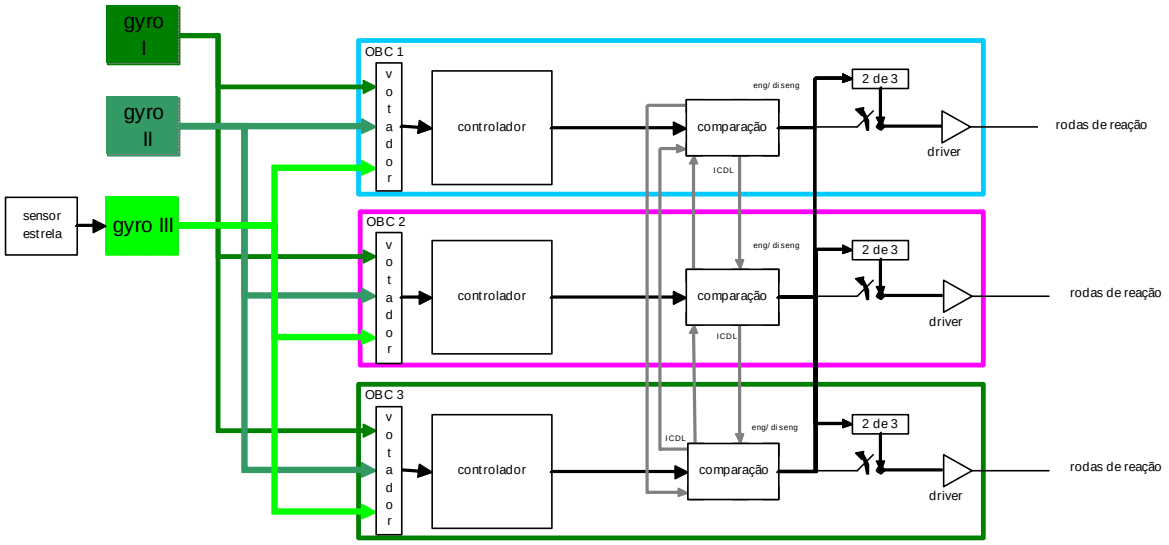

Figura 4.19 – Detalhamento interno da arquitetura da solução triplo-simplex.

A Figura 4.20 apresenta a implementação em MATLAB/Simulink® da comparação interna de comandos, feita dentro de cada um dos canais.

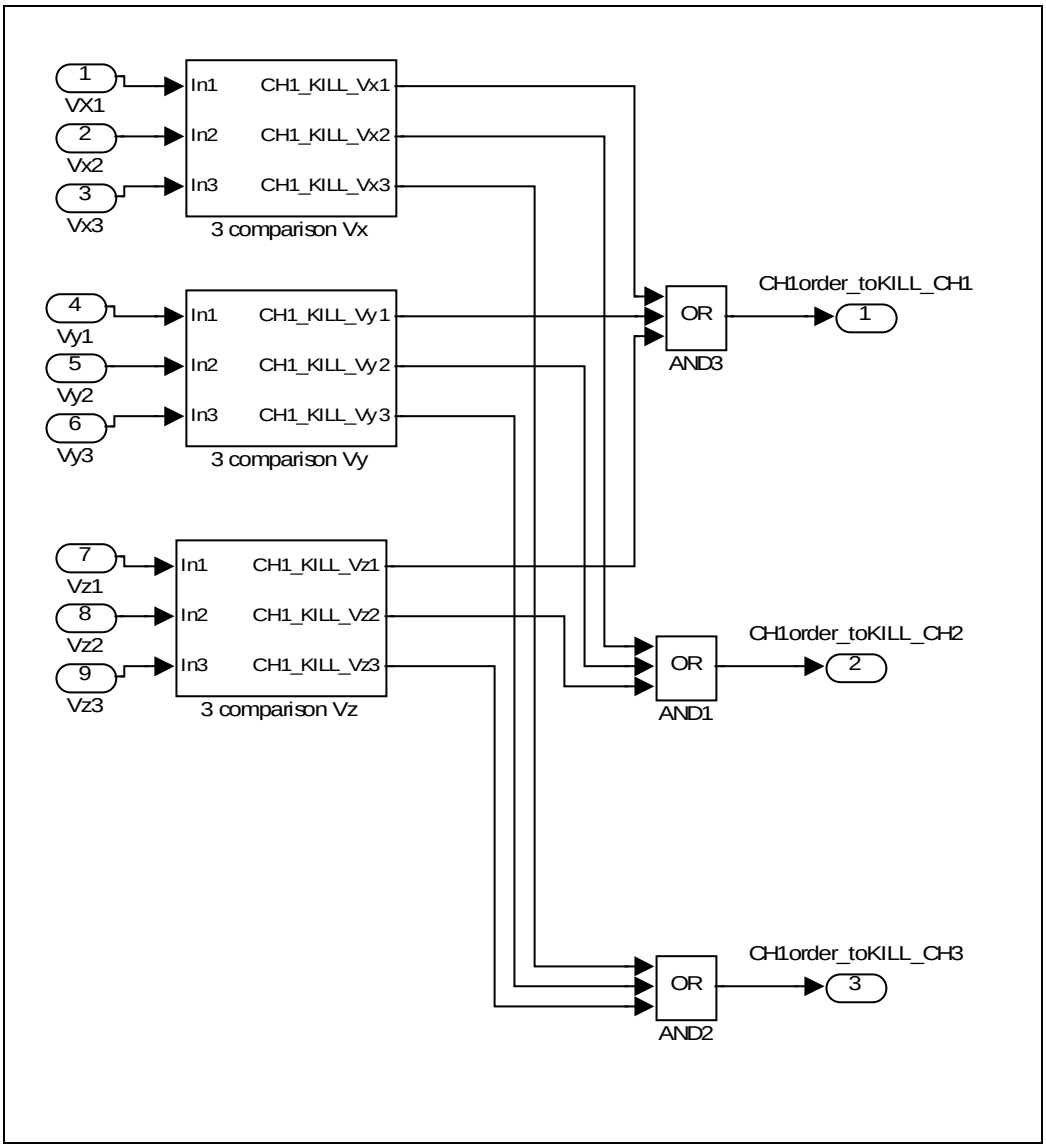

Figura 4.20 – Detalhamento do esquema de detecção de falhas em cada um dos canais. Localização no modelo: \pmm\channel 1\CH1 order to KILL CH1, CH2 and CH3.

A Figura 4.21 apresenta a implementação em MATLAB/Simulink da comparação externa (fórum) entre canais com os resultados da comparação interna de cada canal.

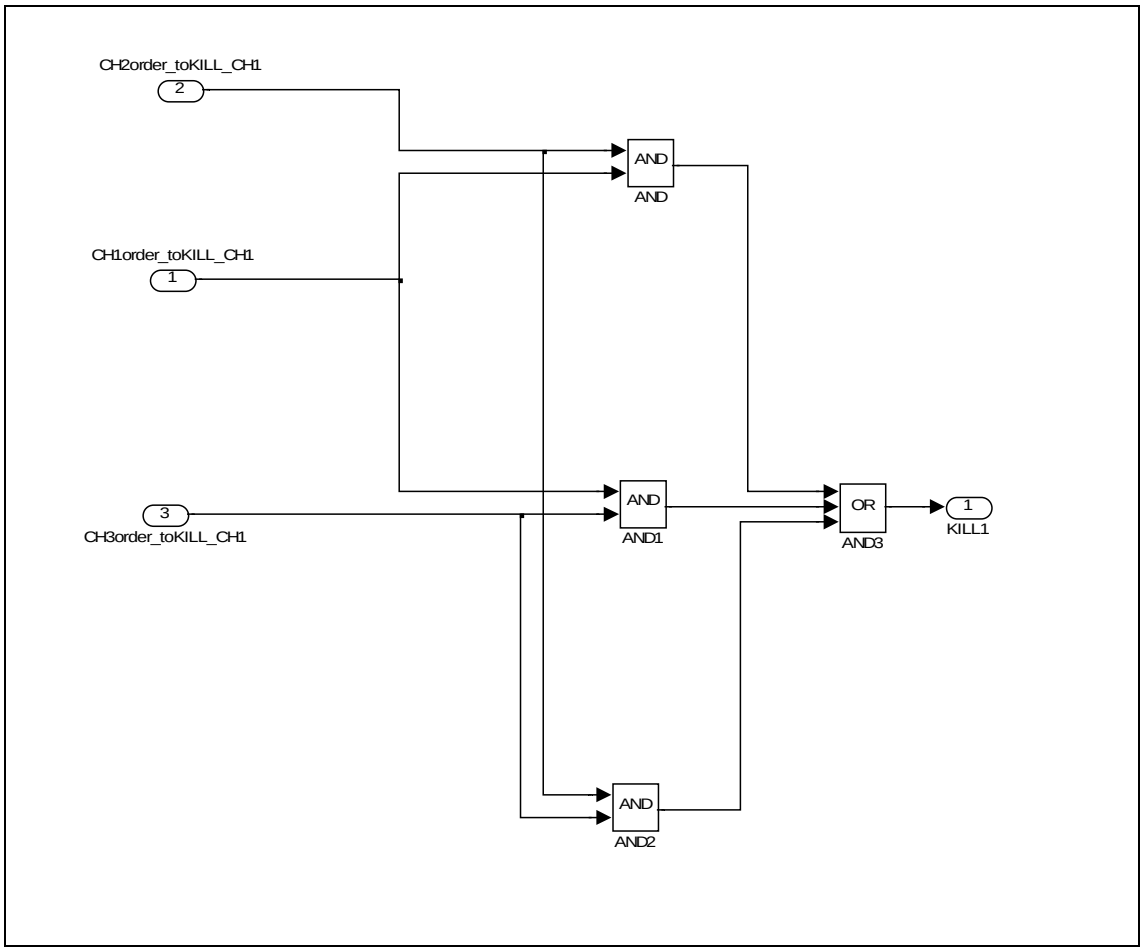

Figura 4.21 – Detalhamento da comparação dos resultados entre os canais. Localização no modelo: \pmm\channel 1\forum to KILL CH1.

Por escolha, o canal 1 estará em controle, a menos que uma falha ocorra e os outros canais o desliguem, passando o controle para outro canal, seguindo a seguinte ordem de precedência: 1, 2 e 3. A ordem de precedência é importante para garantir o determinismo das operações de chaveamento. A Tabela 4.1 resume a ordem de precedência entre os canais 1, 2 e 3 da solução triplo-simplex.

Tabela 4.1– Ordem de precedência para chaveamento entre os canais da solução triplo-simplex.

KILL1[1]	KILL2[1]	KILL3[1]	Canal em comando
0	0	0	Canal 1
0	0	1	Canal 1
0	1	0	Canal 1
0	1	1	Canal 1
1	0	0	Canal 2
1	0	1	Canal 2
1	1	0	Canal 3
1	1	1	Nenhum canal em comando

Nota: 1) Quando o sinal "KILL X" = 0, indica que o canal está são e pode ser indicado para o controle; quando "KILL X" = 1, indica que o canal está inválido e não pode ser indicado para o controle.

Baseado na precedência da Tabela 4.1, e com a ajuda de um Mapa de *Karnaugh* foram definidas lógicas de engajamento em cada um dos canais:

		KILL2 KILL3			
		00	01	10	11
KILL 1	0	1[1]	1	1	1
	1	0	0	0	0

Figura 4.22 – Mapa de Karnaugh para a lógica de engajamento do canal 1 da solução triplo-simplex.

Nota: 1) No Mapa de Karnaugh para o chaveamento: "1" significará a ação de "zerar" os comandos daquele canal; "0" significará que aquele canal deverá ser selecionado para estar em comando.

		KILL1 KILL3			
		00	01	10	11
KILL 2	0	0[1]	0	1	1
	1	0	0	0	0

Figura 4.23 – Mapa de Karnaugh para a lógica de engajamento do canal 2 da solução triplo simplex.

KILL 3	KILL1 KILL2			
	00	01	10	11
0	0^1	0	0	1
1	0	0	0	0

Figura 4.24 – Mapa de Karnaugh para a lógica de engajamento do canal 3 da solução triplo-simplex.

Uma vez que o canal em controle for desligado, outro canal assume o controle em seu lugar. A lógica de chaveamento entre os canais foi feita de modo que, ao perceber o desligamento de um canal, outro já assume em seu lugar.

Uma vez votados os comandos, estes são enviados para as rodas de reação, para a devida atuação. As rodas de reação devem ter a capacidade de receber os comandos dos três canais e usar aquele do canal em controle.

A seqüência (Passos 0, 1 e 2) apresentada a seguir ilustra o funcionamento da lógica de chaveamento da solução. A seqüência retrata as ações desencadeadas por uma falha introduzida no comando V_x do canal 1.

PASSO 0: Antes da falha, o canal 1 está em comando.

PASSO 1: detecção da falha. Os três canais 1, 2 e 3 têm seus comandos comparados, eixo a eixo. Aqui, é introduzida uma falha no comando V_x do canal 1.

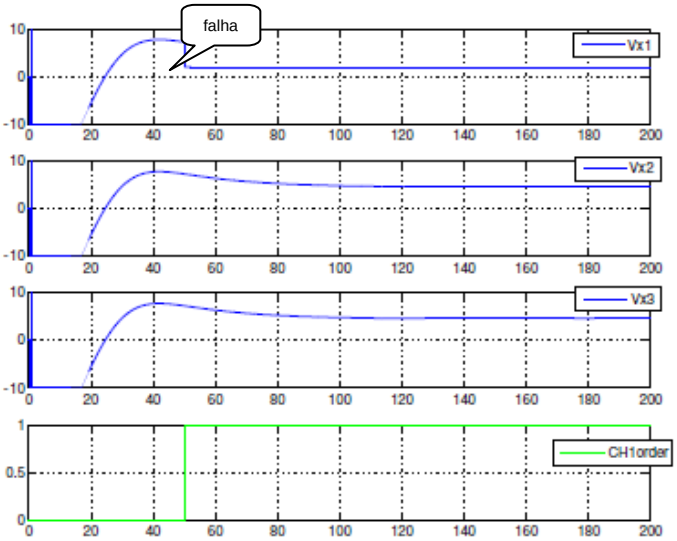

Figura 4.25 – Falha introduzida no comado V_x do canal 1 para demonstrar a operação do chaveamento dos canais. PASSO1, detalhe dos comandos dos três canais que chegam ao canal 1 para a comparação.

Vista do canal 1 (Figura 4.25), a falha é identificada e é emitida uma ordem para se desligar o canal 1 (CH1order_toKILL_CH1).

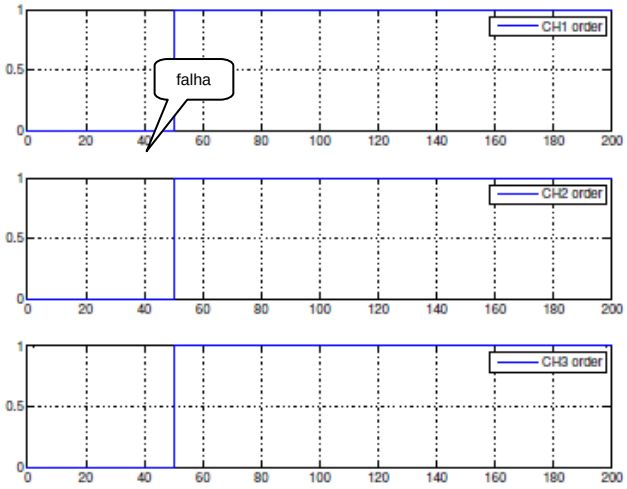

Figura 4.26 – Falha introduzida no comado V_x do canal 1 para demonstrar a operação do chaveamento dos canais. PASSO1, identificação da falha pelos canais 2 e 3 e ordem dos três canais para desligamento do canal 1.

Assim como no canal 1, os canais 2 e 3 também detectam a falha e emitem ordens similares à do canal 1 para o seu desligamento (Figura 4.26).

PASSO 2: Troca de canais. Dado que o canal 1 está falhado, todos os canais emitem ordens para desligar o canal 1. Mesmo que somente dois dos canais emitissem a ordem, ainda assim haveria o chaveamento.

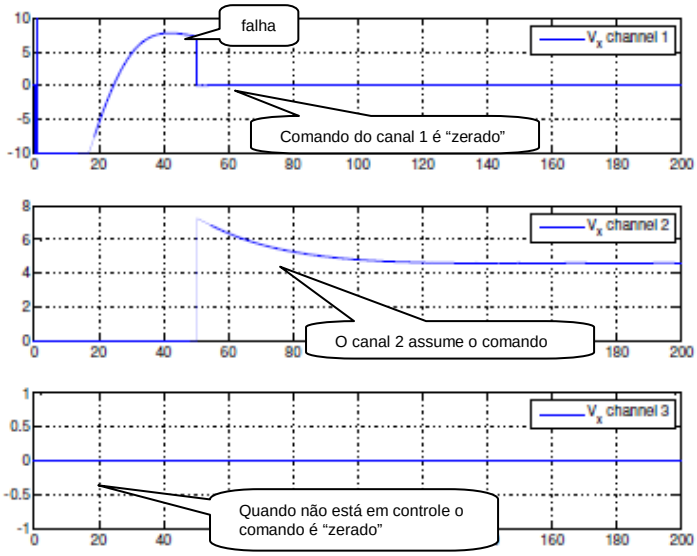

Figura 4.27 – Falha introduzida no comado V_x do canal 1 para demonstrar a operação do chaveamento dos canais. PASSO2, chaveamento de canal: canal 1 para canal 2.

No momento que a falha é detectada os comandos do canal 1 são "zerados" e o canal 2 assume o comando da roda de reação (vide Figura 4.27). Os sinais da Figura 4.27 foram capturados na entrada da roda de reação. Note que, para a roda, só chega o sinal em comando. Por isso o canal 3, por exemplo, permanece "zerado" durante todo o teste.

4.4.2 Detalhamento da arquitetura duplo-simplex mais módulo de Segurança (arquitetura de um monitor típico)

A estratégia de coordenação de canais do duplo-simplex mais módulo de Segurança (duplo-simplex de agora em diante) é um pouco diferente da apresentada anteriormente para o triplo-simplex. O canal duplo sempre estará em comando, até que uma falha seja detectada. Ao invés de se tentar isolar qual dentre os dois computadores do canal está falhado, simplesmente se desliga o conjunto todo e o comando é transferido para o módulo de Segurança. Porém, e se a falha ocorrer no módulo de Segurança? Para que

essa falha não fique latente no módulo de Segurança, propõe-se testar o módulo de Segurança esporadicamente para se verificar a sua sanidade. Esse teste poderia ser iniciado da estação-terra ou agendado para se iniciar periódica e automaticamente.

O módulo de Segurança como proposto nesse trabalho atenderá a todos os requisitos de desempenho apresentados, mas poder-se-ia adotar uma estratégia diferente. O intuito do módulo de Segurança poderia ser, por exemplo, manter o controle da PMM sob características degradadas. Assim, não seria necessário conectá-lo a todos os atuadores ou a todos os sensores. Essa estratégia seria interessante para se racionar a quantidade de *hardware* necessária e, assim, se aliviar peso, consumo elétrico, dissipação de calor, espaço, etc.

4.4.2.1 Entradas de dados

As mesmas considerações sobre entradas de dados para a solução triplo-simplex valem para a duplo-simplex, i.e., o uso dos dois giroscópios e um sensor de estrelas, além da implementação do votador de entradas. A única ressalva, porém, é que o módulo de Segurança não precisa ter todas as conexões do módulo duplo-simplex. Ora, o módulo de Segurança não está no comando em situações sem falhas. Se ocorrer uma falha no módulo de Segurança, seja no processamento, no *driver* de saída ou no sensor de entrada, isto será detectado pelo teste esporádico proposto. Assim, sugere-se conectar somente um dos giroscópios ao módulo de Segurança. A primeira pergunta que pode surgir é “e se esse giroscópio que alimenta o módulo de Segurança e o duplo-simplex falhar? A PMM ficará sem controle?” Como já visto anteriormente, caso ocorra uma falha em um dos sensores, o votador se encarregará de selecionar o outro giroscópio para calcular os comandos.

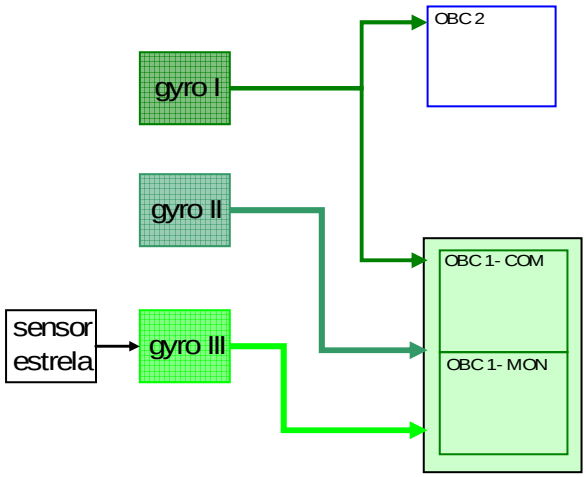

Figura 4.28 – Configuração dos sensores na entrada da solução duplo-simplex.

A Figura 4.28 apresenta a configuração de entrada da solução duplo-simplex. A implementação da solução duplo-simplex em MATLAB/Simulink® é mostrada na Figura 4.29.

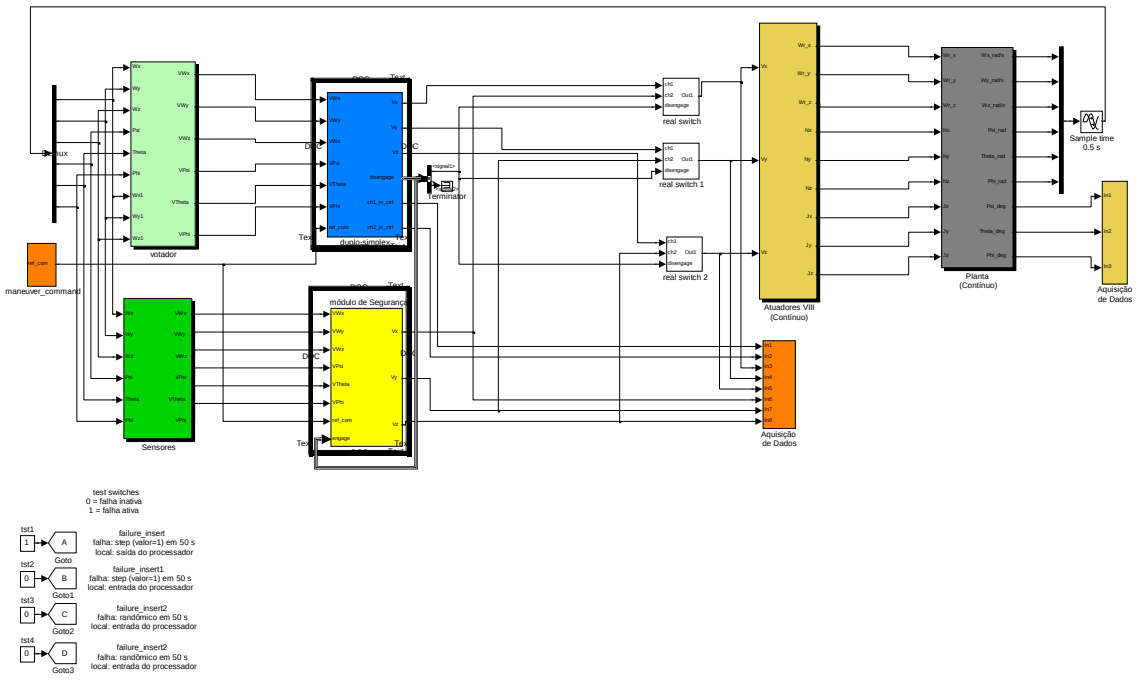

Figura 4.29 – Implementação em MATLAB® da solução duplo-simplex. Localização no modelo: \pmm\.

Note na Figura 4.29, à esquerda do modelo, no bloco verde-claro está implementado o votador de entradas, já apresentado anteriormente. O votador

alimenta somente o canal duplo-simplex, enquanto que o módulo de Segurança recebe os dados somente de um giroscópio. Note também, à direita do canal duplo-simplex (bloco cor laranja) e do módulo de Segurança (bloco amarelo) há três blocos (incolores) que implementam um comutador real para se estudar o efeito da troca de canais. O comutador real será melhor explorado na seção seguinte.

4.4.2.2 Processamento e saída de comandos

Uma vez que os dados estejam disponíveis para o controlador, as duas linhas de comando do canal duplo-simplex passam a calcular os comandos das rodas de reação. As linhas do duplo-simplex operam em uma configuração de comando (COM) e monitoramento (MON). Somente o COM tem o privilégio de comandar as rodas de reação, enquanto que ao MON fica relegada a função de monitoramento. A Figura 4.30 apresenta a configuração interna do canal duplo-simplex.

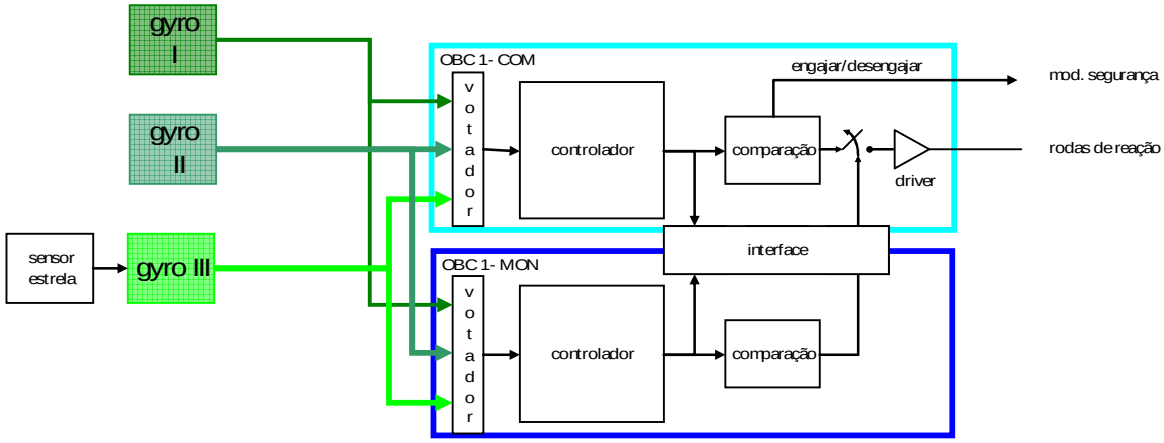

Figura 4.30 – Representação da configuração interna do canal duplo-simplex.

Ambos, COM e MON, têm privilégios de desligar o COM. Ao desligar o canal duplo-simplex, uma ordem de engajamento do módulo de Segurança é emitida. A

Figura 4.31 apresenta a implementação do canal duplo-simplex em MATLAB/Simulink®.

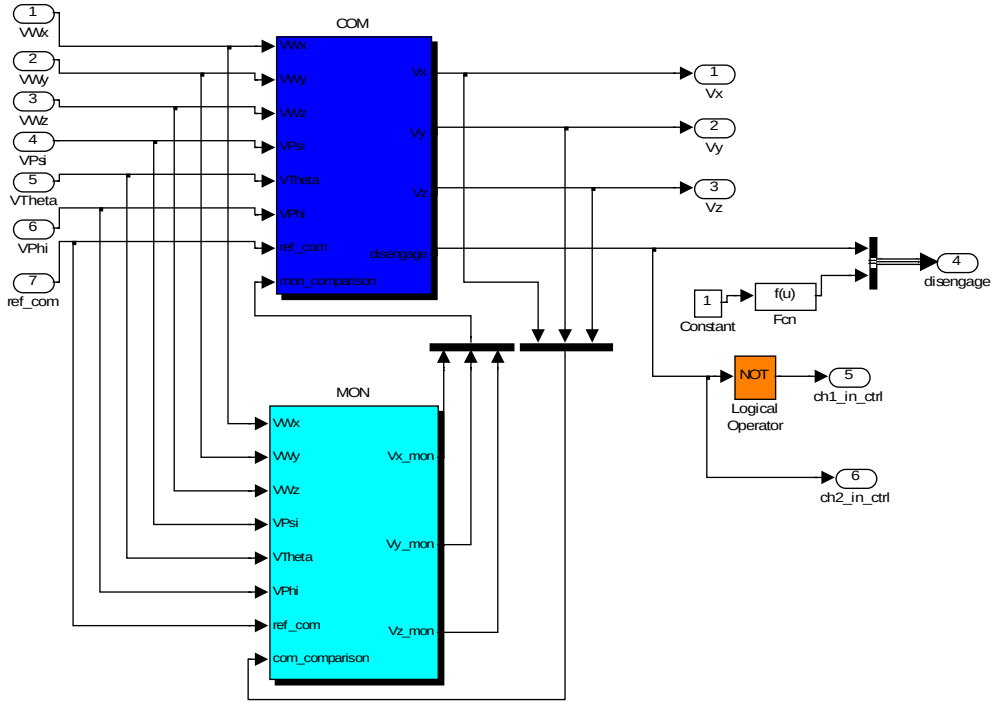

Figura 4.31 – Implementação em MATLAB/Simulink® do canal duplo-simplex. Localização no modelo: \pmm\duplo-simplex.

Ao final dos cálculos de cada uma das linhas, os comandos são trocados, e conferidos separadamente. Se os dois cálculos forem os mesmos (a menos de uma tolerância, ou "*threshold*") o comando é transmitido às rodas através de um barramento de dados. Caso contrário, o canal é desligado e o comando é transferido para o módulo de Segurança.

A comparação entre COM e MON é feita através de um monitor típico que conta com a comparação dos valores calculados (contra um limite de diferença) e de uma persistência no tempo. Se a diferença entre COM e MON for maior que a tolerância (*threshold*), por um tempo maior que a persistência, então declara-se a falha (Figura 4.32).

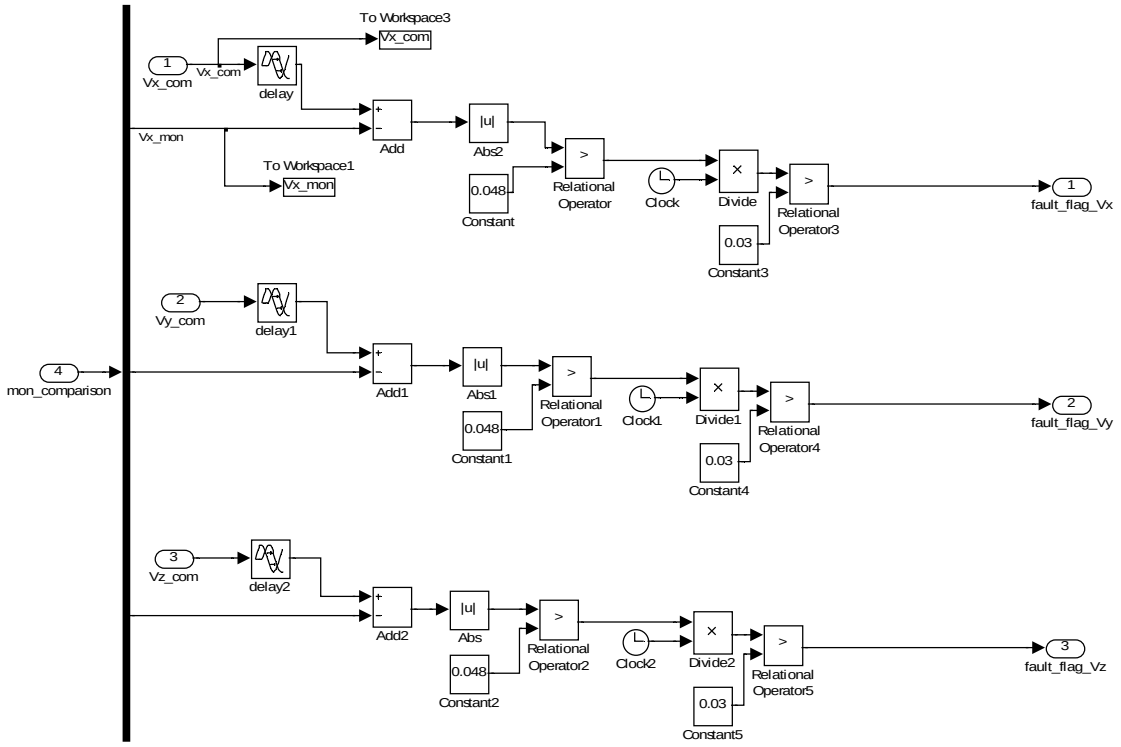

Figura 4.32 – Implementação em MATLAB/Simulink® de um monitor típico. Localização dentro do modelo: \pmm\duplo-simplex\COM\lane to Lane comparison.

A estrutura simples desse monitor pode mascarar a dificuldade em se projetá-lo. O grande dilema do projeto de um monitor é não concebê-lo permissivo demais e nem restritivo demais. Se for muito restritivo, o monitor detectará todas as falhas, porém, em contrapartida, acusará muitos alarmes falsos, que são casos de operação normal (i.e., sem falhas), mas que estão no limite da detecção. Isso é ruim para a operação do sistema, mas bom para a Dependabilidade. Por outro lado, se o monitor for muito permissivo, não detectará algumas falhas; mas terá poucos (ou nenhum) alarmes falsos. Isso é bom para operação, porém ruim para a Dependabilidade. O ponto ótimo é aquele em que o monitor detecte o máximo de falhas com o mínimo de alarmes falsos.

Uma maneira de se buscar esse ponto ideal é o correto ajuste da tolerância (*thresholds*) e persistência. Para tanto, tem se que conhecer as grandezas sendo comparadas.

A seqüência apresentada seguir exemplifica a atuação do monitor com uma falha introduzida na linha COM do canal duplo-simplex.

PASSO 0: Antes da falha a linha COM do canal duplo-simplex está no comando.

PASSO 1: Uma falha é introduzida na linha COM do canal duplo-simplex. A comparação entre COM e MON acusa uma diferença maior que o limite de comparação por um tempo maior que a persistência, caracterizando a falha.
O indicador de falha é acionado. A Figura 4.33 evidencia essa seqüência de ações descritas.

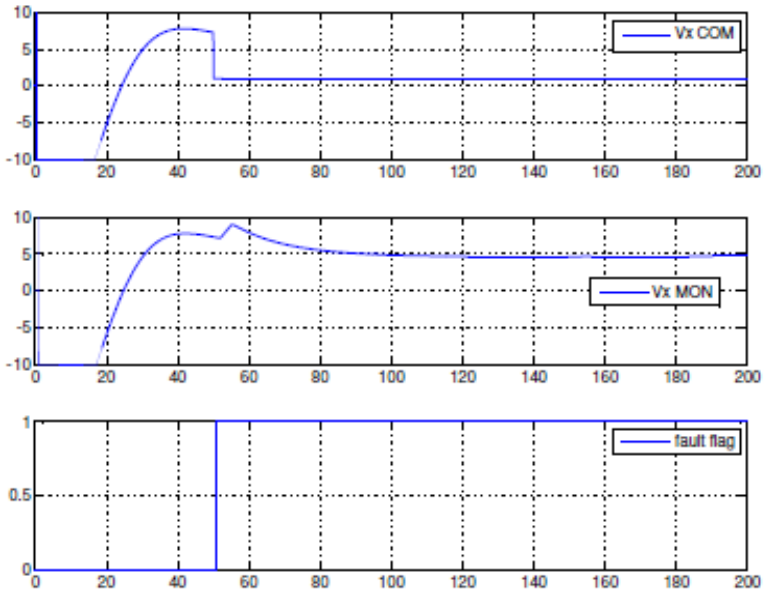

Figura 4.33 – Falha introduzida na linha COM do canal duplo-simplex. PASSO 1: a comparação entre os comandos de COM e MON acusa a falha que é evidenciada pelo indicador.

PASSO 2: Com o indicador de falhas ativo, o sinal do canal duplo-simplex é zerado e uma ordem de “engajamento” é gerada e transmitida para o módulo de Segurança, que assume o controle. A Figura 4.34 evidencia essa seqüência de ações descritas.

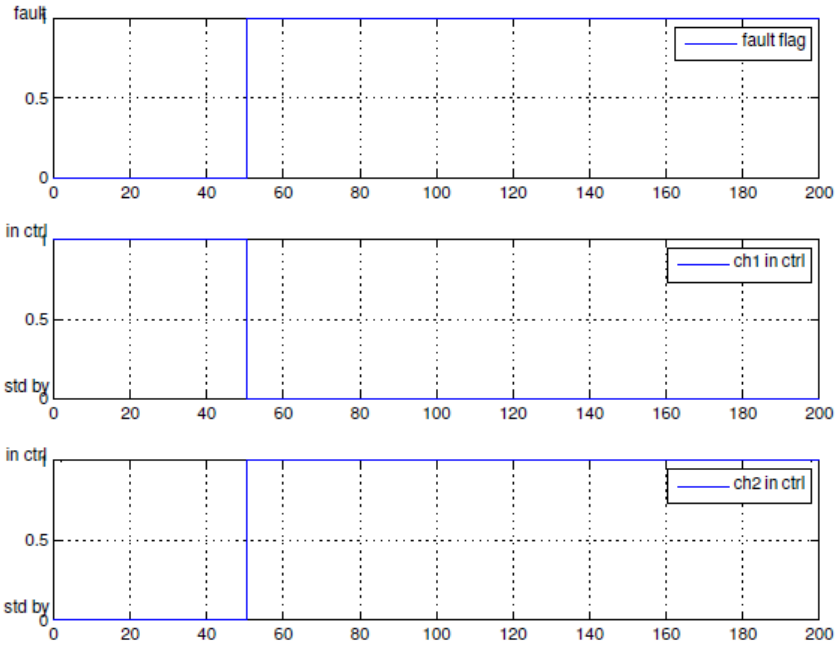

Figura 4.34 - Falha introduzida na linha COM do canal duplo-simplex. PASSO 2: o indicador de falhas gera uma ordem para o desengajamento do canal duplo-simplex e engajamento do módulo de Segurança.

Do ponto de vista das rodas de reação, a falha e o chaveamento serão praticamente transparentes, a menos do tempo de chaveamento, como mostrado na Figura 4.35.

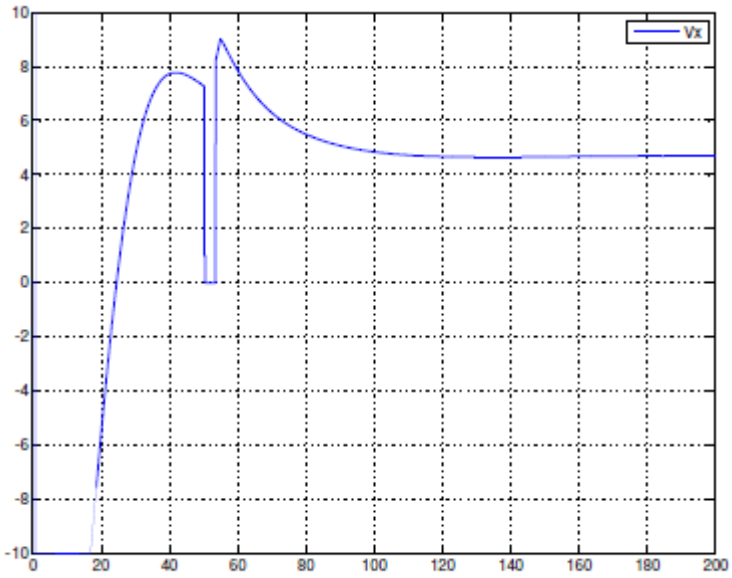

Figura 4.35 - Falha introduzida na linha COM do canal duplo-simplex. PASSO 2: comando de Vx como visto pela roda de reação.

Esse intervalo perceptível no chaveamento foi introduzido artificialmente, pois os elementos da biblioteca do Simulink® fazem a transferência de canais instantaneamente. Porém, na prática, o chaveamento vai incluir atrasos do barramento de comunicação, do próprio *clock* do processador, dos *drivers*, etc.

Para este sistema em estudo o atraso do chaveamento não atrapalhou o sistema em cumprir com os requisitos de desempenho impostos, talvez a inércia do sistema seja muito alta e o efeito do chaveamento desprezível.

4.5 Verificação das propostas apresentadas

Visto como as duas soluções propostas operam, têm-se que analisá-las frente aos requisitos impostos. Como os requisitos são muitos, antes de começar a analisá-los será proveitoso criar uma estratégia de verificação, listando-se os requisitos e o método usado em sua verificação (e.g. teste, análise, etc.). Veja Tabela 4.2.

Tabela 4.2 – Métodos e meios de verificação das soluções apresentadas frente aos requisitos propostos.

	Método de verificação/ Meio de verificação			
Requisito	Triplo-simplex	Duplo-simplex	Tipos de falhas consideradas	Seção com o meio de verificação
MPP-R-1	Análise/ Árvore de Falhas	Análise/ Árvore de Falhas	Aleatórias	4.5.1
MPP-R-2	Análise/ Árvore de Falhas	Análise/ Árvore de Falhas	Aleatórias	4.5.1
MPP-R-3	Análise/ Inspeção da proposta	Análise/ Inspeção da proposta	Aleatórias	4.5.3
MPP-R-4	Análise/ Inspeção da proposta	Análise/ Inspeção da proposta	Aleatórias, SEU, Projeto	4.5.3
MPP-R-5	Análise/ Inspeção da proposta	Análise/ Inspeção da proposta	Aleatórias, SEU, Projeto	4.5.3
MPP-R-6	Análise/ Inspeção da proposta	Análise/ Inspeção da proposta	Aleatórias, SEU, Projeto	4.5.3
MPP-R-7	Teste/ Simulação	Teste/ Simulação	Aleatórias	4.5.2
MPP-R-8	Análise/ Árvore de Falhas	Análise/ Árvore de Falhas	Aleatórias	4.5.1

4.5.1 Verificação das soluções apresentadas – verificação por análise de Árvore de Falhas

Essa seção tem por objetivo verificar a aderência das soluções propostas aos seguintes requisitos (apresentados na seção 4.3 e repetidos aqui por conveniência):

- MPP-R-1;
- MPP-R-2;

- MPP-R-8.

[MPP-R-1] A Plataforma MultiMissão deve apresentar Confiabilidade maior que 0,8 (taxa de sucesso) considerando-se uma vida útil de 4 anos (35040 horas).

As duas soluções foram analisadas com a ajuda de suas respectivas árvores de falhas, onde o evento topo de ambas era a “Perda da PMM” (veja extrato das árvores nas Figura 4.36 e Figura 4.37). As árvores completas com todos os ramos, hipóteses e explicações para as taxas de falhas usadas encontram-se no **Apêndice A**.

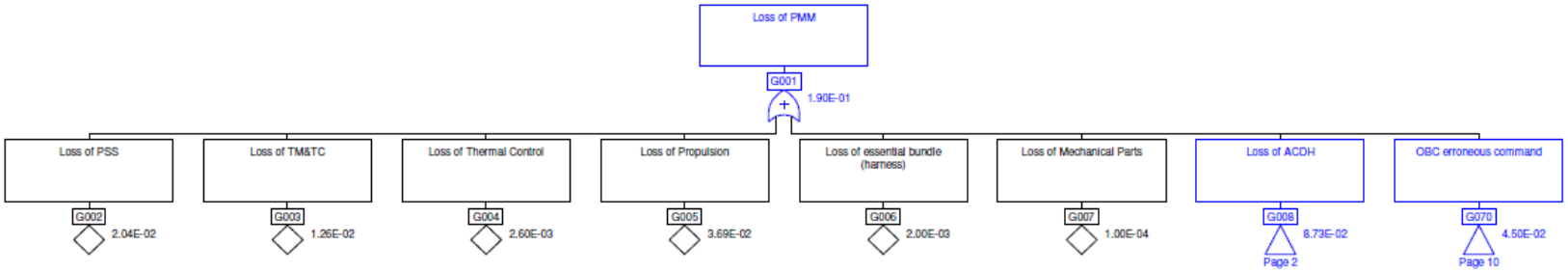

Figura 4.36 – Árvore de Falhas mostrando o cumprimento do requisito MPP-R-1 pela solução duplo-simplex.

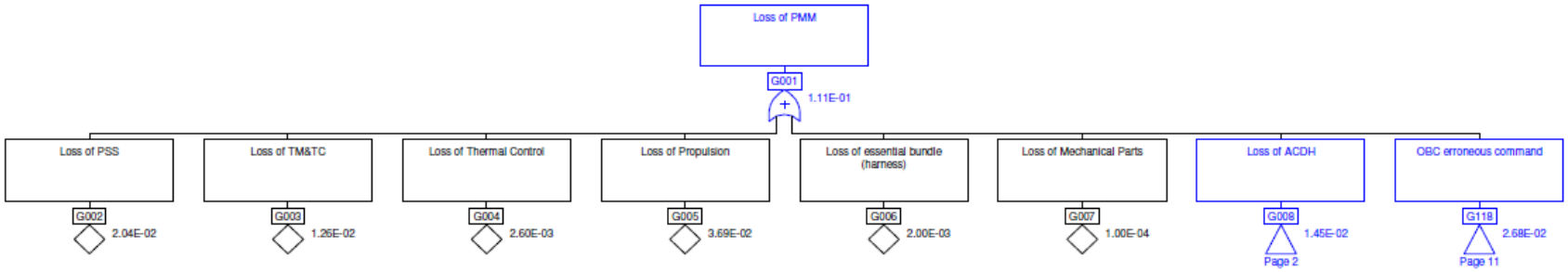

Figura 4.37 - Árvore de falhas mostrando o cumprimento do requisito MPP-R-1 pela solução triplo-simplex.

Como se pode ver pela Figura 4.36, a rigor, a solução duplo-simplex cumpre o requisito MPP-R-1 com uma margem mínima, pois a especificação requer uma probabilidade de sucesso de 0,8, enquanto que a solução só cumpre com 0,81 [10].

É discutível considerar o requisito atendido com tão pouca margem Porém, seria muito desperdício descartar a solução por uma diferença tão pequena. Lembre-se ainda que as taxas de falhas usadas são muitas vezes estimativas (veja **Apêndice A**) e podem ser melhores na realidade. Interessante notar que essa probabilidade agora apresentada (i.e., 0,81) é pior do que a primeira estimativa apresentada na Figura 4.7. Isso é natural, já que, à medida que se detalha uma solução, encontram-se percalços que não haviam sido considerados à *priori*. Além disso, a especificação original não considerava a probabilidade de comando errôneo do OBC como uma das causas de perda da PMM, o que no final de contas, é mais um argumento a favor da insistência na solução duplo-simplex.

Já a solução triplo-simplex cumpre com folgas o requisito PMM-R-1. A probabilidade de sucesso apresentada na Figura 4.37 é de 0,89 [10], já se considerando a contribuição do comando errôneo do OBC para a perda da PMM.

A seguir está a análise para verificação do cumprimento do requisito [MPP-R-2].

---

[10] A probabilidade da árvore de falhas é sempre “falhas”. No caso, a probabilidade de falha é de 1,9E-01, mas a probabilidade de sucesso é de 0,81, pois a taxa de sucesso é o complemento da taxa de falha, i.e. 1 - 0,19 = 0,81.

[MPP-R-2] O sistema de controle de atitude e manipulação de dados (ACDH) deve ter Confiabilidade maior que 0,9462 (taxa de sucesso) considerando-se uma vida útil de 4 anos (35040 horas).

O requisito MPP-R-2 requer uma probabilidade de sucesso de 0,9462. A solução duplo-simplex, de acordo com a Figura 4.36, provê somente 0,9127. Novamente, analisando-se rigorosamente a aderência aos requisitos, a solução não cumpre com a especificação. Porém, pelos argumentos já apresentados, vale insistir na solução como apresentada.

A solução triplo-simplex cumpre com sobras a especificação. De acordo com a Figura 4.37, a probabilidade de sucesso associada ao ACDH é de 0,9855 contra 0,9462 da especificação.

[MPP-R-8] O sistema de controle de atitude e manipulação de dados (ACDH) deve apresentar uma probabilidade de comando errôneo (Integridade) menor que $1x10^{-6}$ falha/h.

A solução duplo-simplex não cumpre com o requisito de Integridade. De acordo com a Figura 4.36, a solução apresenta probabilidade de comando errôneo de $1,28 \times 10^{-6}$ falha/h [11]. Porém, pelos mesmos motivos apresentados acima, não se descartará a solução até a análise de todos os requisitos.

Já a solução triplo-simplex cumpre com folgas o requisito, $4,14 \times 10^{-7}$ falha/h, ou $0,414 \times 10^{-6}$ falha/h, como se pode ver pela Figura 4.37.

[11] O requisito é apresentado em termos de probabilidade de falha/h e a árvore de falhas apresenta as probabilidades em termos do tempo total de operação, ou 35040 horas. Assim, a probabilidade de comando errôneo apresentada na árvore deve ser dividida por 35040 horas, que é o tempo de operação da plataforma.

4.5.2 Verificação das soluções apresentadas – verificação por simulação

Essa seção tem por objetivo verificar a aderência das soluções propostas aos seguintes requisitos (apresentados na seção 4.3 e repetidos aqui por conveniência):

- MPP-R-7.

[MPP-R-7] Em Modo Nominal, a atitude do satélite deve ser controlada nos três eixos para cumprir com os seguintes requisitos:

a) Erro de determinação de atitude menor que 0,005° (3σ);
b) Taxa de amortecimento $0,3 < \xi < 0,8$;
c) t_s = 100s (5% do valor final);
d) "*Drift*" menor que 0,001°/s;

A letra "a)" do requisito MPP-R-7 será verificada durante as simulações, porém, sua determinação depende muito das características física dos componentes mecânicos, eletrônicos e digitais usados na fabricação do sistema, as quais não foram representadas nas simulações.

Uma estratégia para verificação do requisito MPP-R-7 é introduzir falhas no modelo das soluções e observar o seu comportamento. Em um projeto todos os componentes do sistema deveriam ter o seu efeito especificado e verificado contra os requisitos. Assim, é comum a fase de testes para verificação de sistemas complexos tomar tanto tempo, ou até mais, que a fase de projeto. Para os fins deste trabalho, como as funções não foram atribuídas a componentes, sugere-se identificar as principais funções e se introduzir falhas a elas.

Para a solução duplo-simplex, propõe-se 4 pontos de inserção de falhas, identificados com as letras de "A" à "D" na Figura 4.38.

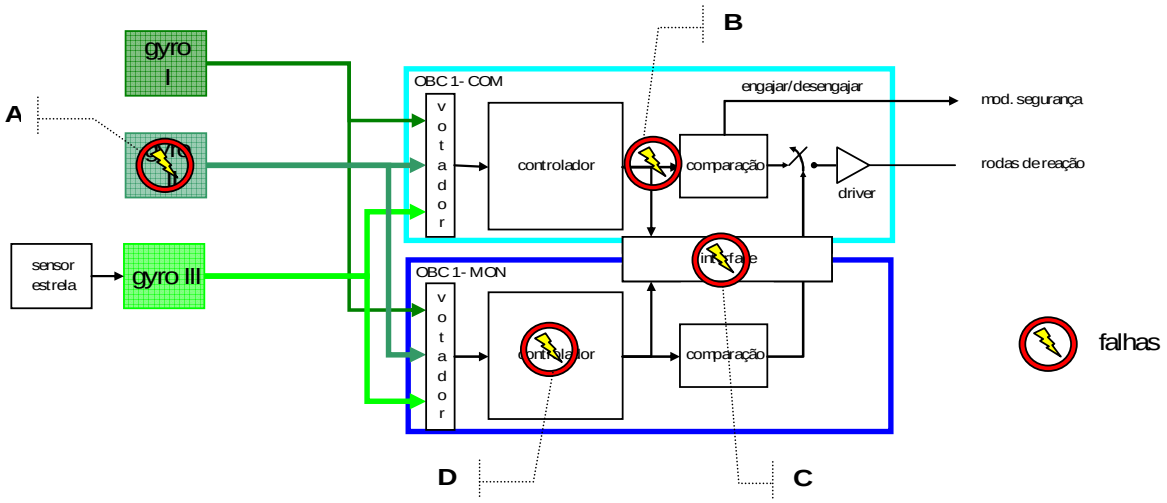

Figura 4.38 – Pontos de inserção de falhas da arquitetura duplo-simplex para a verificação do requisito MPP-R-7.

A Tabela 4.3, resume os pontos em que serão introduzidas as falhas, assim como o tipo de falha. Antes de se começar a seqüência de falhas, um caso sem falhas será apresentado.

Tabela 4.3 – Resumo dos pontos de inserção de falhas para verificação da aderência da solução duplo-simplex ao requisito MPP-R-7.

Falha	Ponto de inserção da falha	Tipo de sinal de falha	Detalhamento da falha
AA	Sem falha	Sem falha.	Manobra de captura de Psi = 1°.
A1	Giroscópio I (\pmm\votador)	Falha de valor constante, baseado em Leite (28).	Degrau de 2° introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1°.
A2	Giroscópio I (\pmm\votador)	Deriva de *offset*, baseado em Leite (28).	Rampa de offset somado ao valor do sinal do giroscópio introduzida a partir de t = 50s, e durante manobra de captura de atitude de Psi = 1°.
A3	Giroscópio I (\pmm\votador)	Falha oscilatória.	Função seno com amplitude de 1° e com freqüência constante de 10Hz introduzida em t = 50s, e durante manobra de captura de atitude de Psi = 1°.
B	Saída do controlador PID (\pmm\duplo-simplex\COM)	Falha de valor constante.	Degrau de 10° introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1°.
C	Entrada do comparador entre COM e MON na linha COM (\pmm\duplo-simplex\COM)	Falha de valor constante.	Degrau de 10° introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1°.
D	Saída do controlador PID (\pmm\duplo-simplex\MON)	Falha intermitente, baseado em Leite (28).	Onda quadrada de 1° e freqüência de 10Hz introduzida em t=50s durante manobra de captura de atitude de Psi = 1°.

As falhas serão introduzidas aos 50s de uma simulação de captura de atitude de Psi de 1°. As escolhas do eixo a ser estimulado e da referência foram aleatórias, mas Gobato (26) estressou várias outras possibilidades em seu trabalho, que comprovaram a fidelidade do modelo para os seus propósitos.

Antes de se inserir as falhas, um caso sem falha foi rodado (veja Figura 4.39).

AA) Caso sem falha

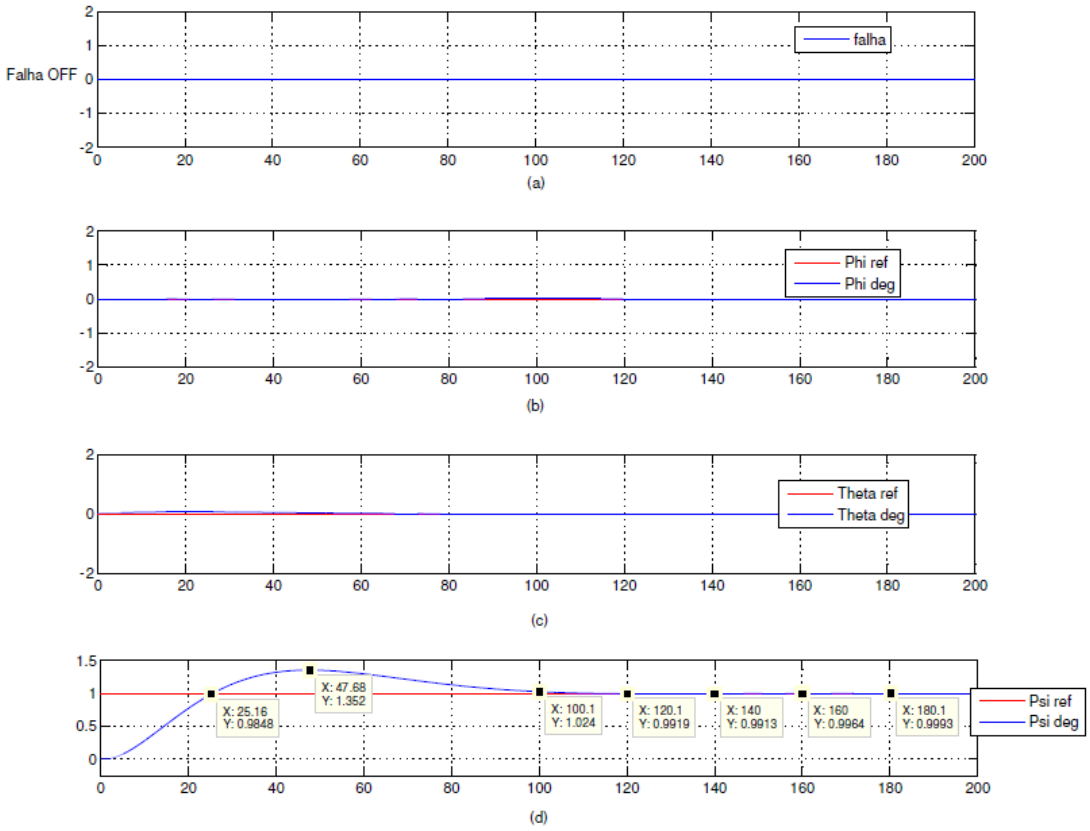

Figura 4.39 – Manobra de captura de Psi de 1° para a solução duplo-simplex, entrada degrau introduzida em t = 0s, sem falhas. (a) falha inserida = 0, (b) Phi referência versus Phi medido, (c) Theta referência versus Theta medido, (d) Psi referência versus Psi medido.

a) Erro de determinação de atitude menor que 0,005°

A Figura 4.39 mostra que o erro entre o sinal de referência e a atitude da PMM é menor que 0,005°.

b) Taxa de amortecimento 0,3 < ξ < 0,8

Segundo (2.3-6),

$$M_p = e^{-(\frac{\xi}{\sqrt{1-\xi^2}})*\pi}$$

, donde vem que:

$$\xi = \frac{\ln M_p}{\sqrt{(\ln M_p)^2 + \pi^2}} \qquad \textbf{(4.4)}$$

A Figura 4.39 mostra que M_p = 0,35, substituindo esse valor em (4.5-1) ξ = 0,32.

c) t_s = 100s

A Figura 4.39 mostra que em t = 100s, Psi = 1,024, ou seja, erro é menor que 5%.

d) “Drift” menor que 0,001º/s;

A Figura 4.39 mostra que a variação da atitude é menor que 0,0006°em 20s.

A1) Falha de valor constante no sensor Giroscópio

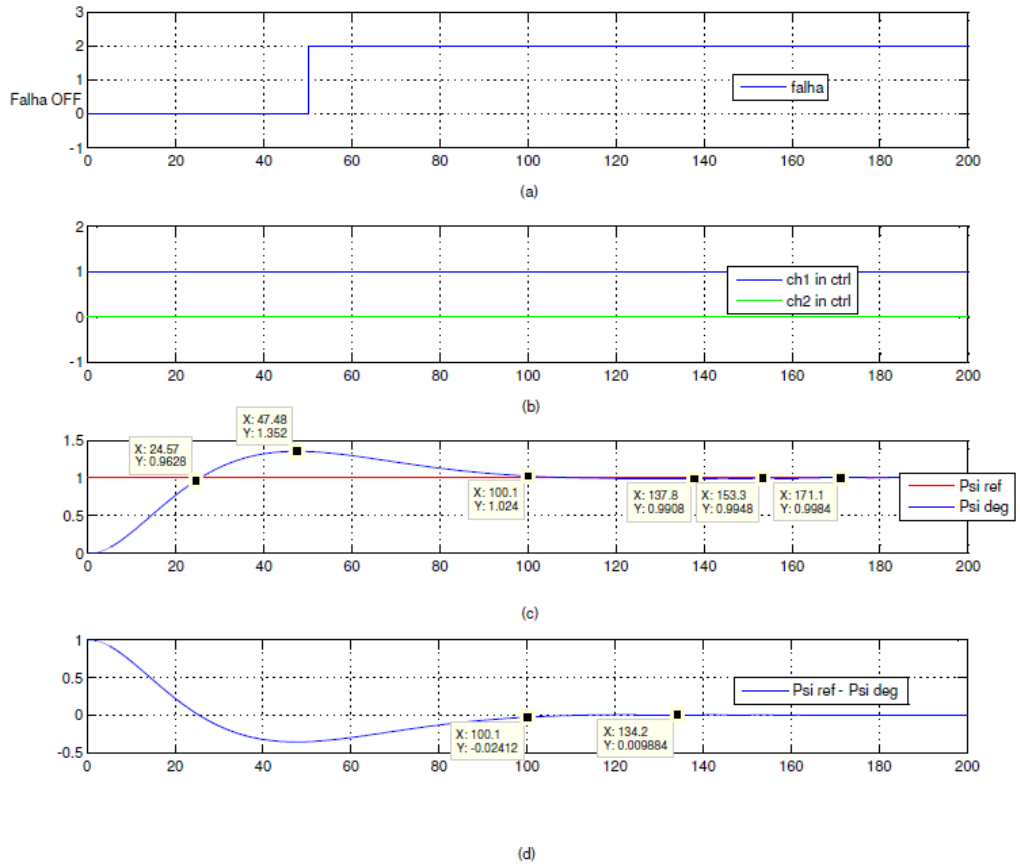

Figura 4.40 – Manobra de captura de Psi de 1 grau para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha introduzida em t=50s. (a) falha constante, (b) canal em controle, (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.40 mostra o resultado da simulação da solução duplo-simplex para a falha de valor constante inserida em um dos sensores giroscópios. Note que por conta do votador de sinais, não houve troca de canais. O canal 1 continuou controlando as rodas de reação, mas usando o segundo giroscópio. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,002°

b) Taxa de amortecimento:
requerido: $0,3 < \xi < 0,8$
medido: $\xi = 0,32$

c) Tempo de acomodação (5%)
requerido: Psi medido (t = 100s) – Psi referência < 5%
medido: Psi medido (t = 100s) – Psi referência < 2,4%

d) “Drift” menor que 0,001°/s
medido: 0,004° em 16s.

A2) Falha em deriva no sensor Giroscópio

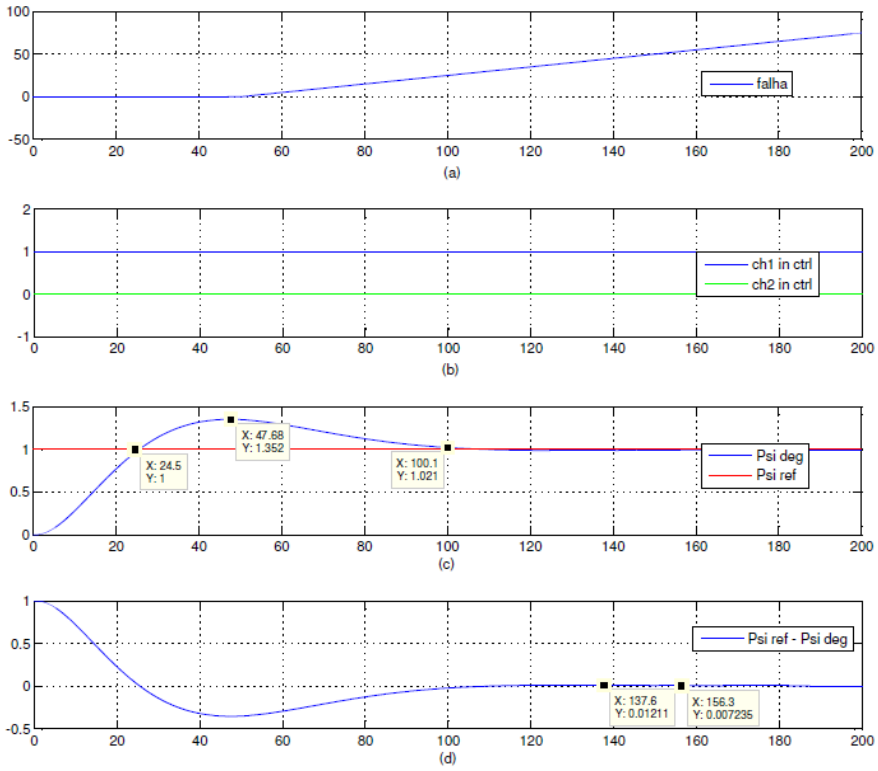

Figura 4.41 – Manobra de captura de Psi de 1° para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha introduzida em t=50s. (a) falha em rampa, (b) canal 1 em controle, (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.41 mostra o resultado da simulação da solução duplo-simplex para uma falha em rampa inserida em um dos sensores giroscópios. Essa falha pretende simular uma deriva no valor do sensor, como sugerido por Leite (28) em seu trabalho. Note que por conta do votador de sinais, não houve troca de canais. O canal 1 continuou controlando as rodas de reação, mas usando o segundo giroscópio. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,007°

b) Taxa de amortecimento:
requerido: $0,3 < \xi < 0,8$
medido: $\xi = 0,32$

c) Tempo de acomodação (5%)

requerido: Psi medido (t = 100s) – Psi referência < 5%

medido: Psi medido (t = 100s) – Psi referência < 2,1%

d) “Drift” menor que 0,001°/s;

medido: 0,002° em 19s;

A3) Falha oscilatória no sensor Giroscópio

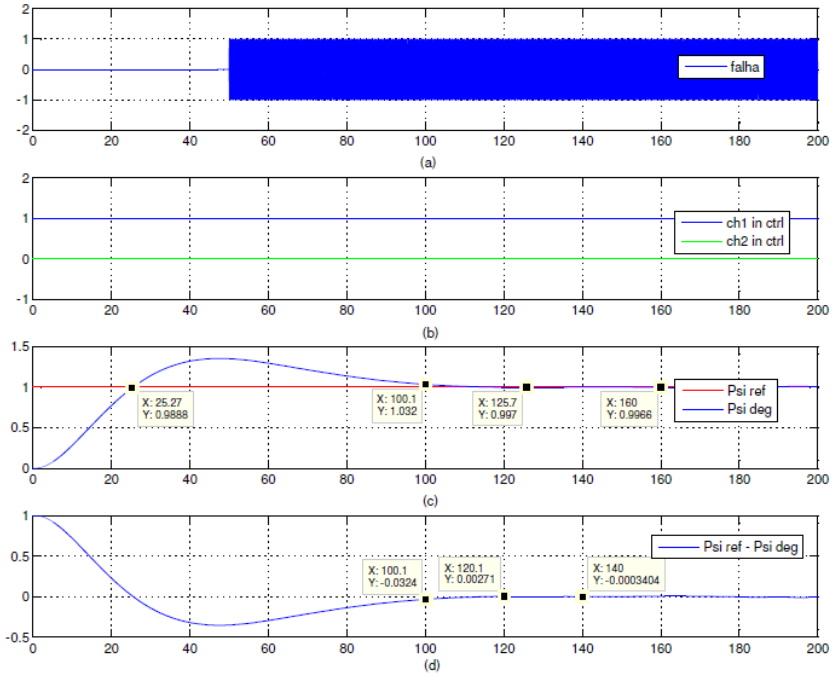

Figura 4.42 - Manobra de captura de Psi de 1° para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha oscilatória (seno 1° amplitude e freqüência de10Hz) introduzida em t=50s (a) falha oscilatória, (b) canal 1 em controle (canal 1 ou canal 2), (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.42 mostra o resultado da simulação da solução duplo-simplex para uma falha oscilatória inserida em um dos sensores giroscópios. Note que por conta do votador de sinais, não houve troca de canais. O canal 1 continuou controlando as rodas de reação, mas usando o segundo giroscópio. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:

requerido: erro Psi < 0,005°

medido: erro Psi < 0,003°

b) Taxa de amortecimento:

requerido: 0,3 < ξ < 0,8

medido: ξ = 0,32

c) Tempo de acomodação (5%)

requerido: Psi medido (t = 100s) – Psi referência < 5%

medido: Psi medido (t = 100s) – Psi referência < 3,2%

d) “Drift” menor que 0,001°/s;

medido: 0,003° em 20s

B) Falha constante no controlador PID

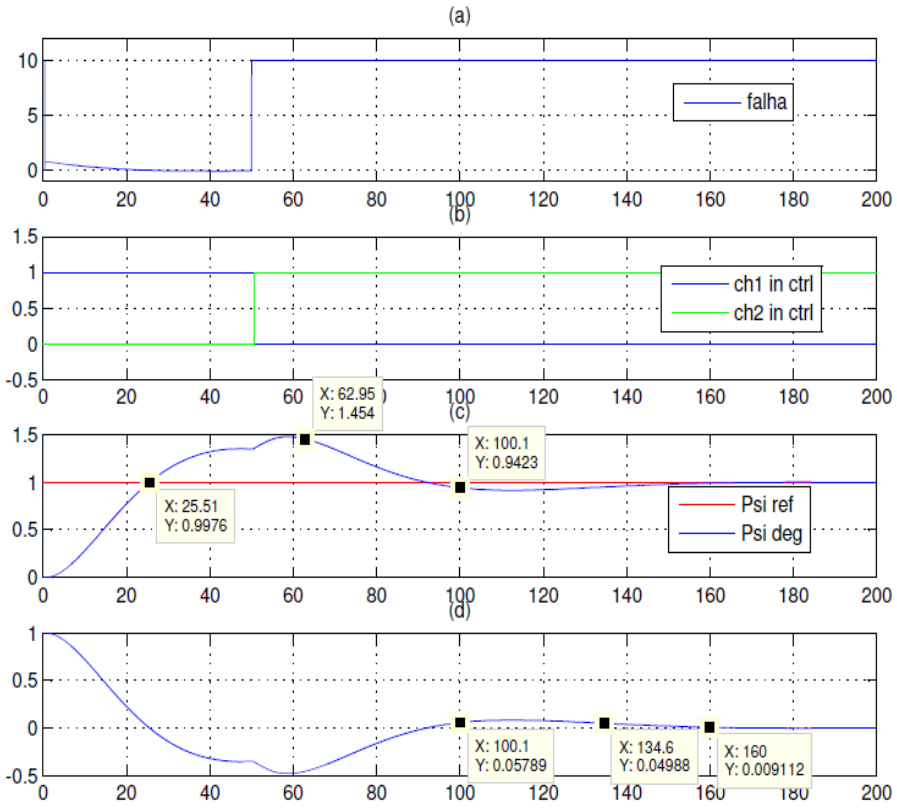

Figura 4.43 – Manobra de captura de Psi de 1 grau para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha constante introduzida após o controlador em t=50s (a) falha (10o amplitude), (b) canal 1 em controle, t<50s, e canal 2 em controle em t>50s , (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.43 mostra o resultado da simulação da solução duplo-simplex para uma falha constante inserida na saída do controlador PID. Esse sinal pretende simular uma falha no computador do OBC levando o seu comando ao máximo do saturador (i.e, 10°). Note que ao ser detectada a falha os canais trocaram de engajamento: o canal estava em controle até t = 50s, depois da inserção da falha o canal 2 assumiu o controle. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,05°

b) Taxa de amortecimento:
requerido: $\xi = 0,3 < \xi < 0,8$
medido: $\xi = 0,23$

c) Tempo de acomodação (5%)
requerido: Psi medido (t = 100s) – Psi referência < 5%
medido: Psi medido (t = 100s) – Psi referência < 5,8%

d) “Drift” menor que 0,001⁰/s;
medido: 0,04° em 16s, ou 0,0025°

Com a inserção da falha, o controlador não cumpre mais com os parâmetros requeridos. O principal causador desse insucesso é um atraso incluído propositadamente no comparador de comandos do COM. A linha COM tem que esperar ao menos 1 ciclo para comparar o seu dado com aquele vindo da linha MON, pois os dados calculados no mesmo instante (com as mesmas entradas) levam ao menos um ciclo para serem transmitidos de uma linha à outra. Com isso, o dado errado produzido na linha COM atua pelo menos por um ciclo antes de ser eliminado, produzindo esse inconveniente. Eliminar esse atraso significa ter que se conviver com constantes alarmes falsos, pois sem ele COM e MON já teriam uma diferença natural provocado pelos cálculos feitos em instantes diferentes no tempo.

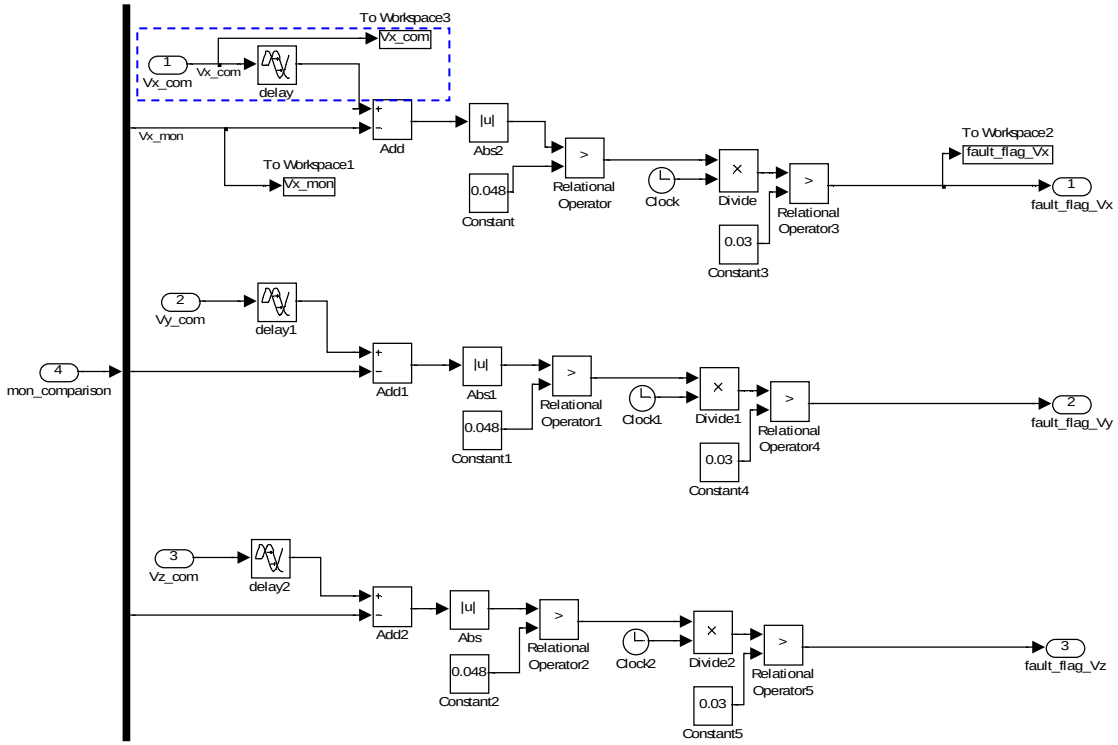

Figura 4.44 – Detalhe do monitor que compara dados de COM e MON. Em destaque (azul tracejado) o atraso incluído propositadamente.

C) Falha constante no sinal de interface entre COM e MON

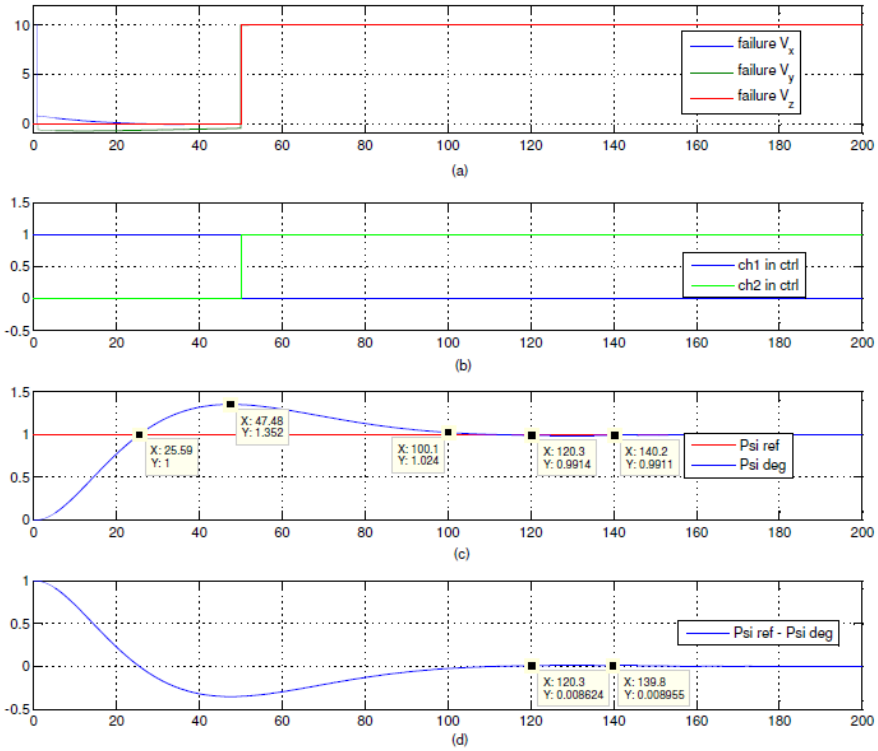

Figura 4.45 – Manobra de captura de Psi de 1° para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha constante introduzida entre a comparação entre COM e MON em t=50s (a) falha (10° amplitude), (b) canal 1 em controle, t<50s, e canal 2 em controle, t>50s , (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.45 mostra o resultado da simulação da solução duplo-simplex para uma falha constante inserida na entrada do comparador entre COM e MON. Esse sinal pretende simular uma falha na interface entre COM e MON. Note que, ao ser detectada a falha, os canais trocaram de engajamento: o canal estava em controle até t = 50s, depois da inserção da falha o canal 2 assumiu o controle. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,008°

b) Taxa de amortecimento:
requerido: $\xi = 0,3 < \xi < 0,8$

medido: $\xi = 0{,}315$

c) Tempo de acomodação (5%)

requerido: Psi medido (t = 100s) – Psi referência < 5%

medido: Psi medido (t = 100s) – Psi referência < 2,4%

Como a falha foi inserida nos dados da linha MON, não houve o mesmo efeito danoso do atraso descrito na falha B. Assim, a solução cumpriu com os requisitos.

D) Falha intermitente no controlador da linha MON

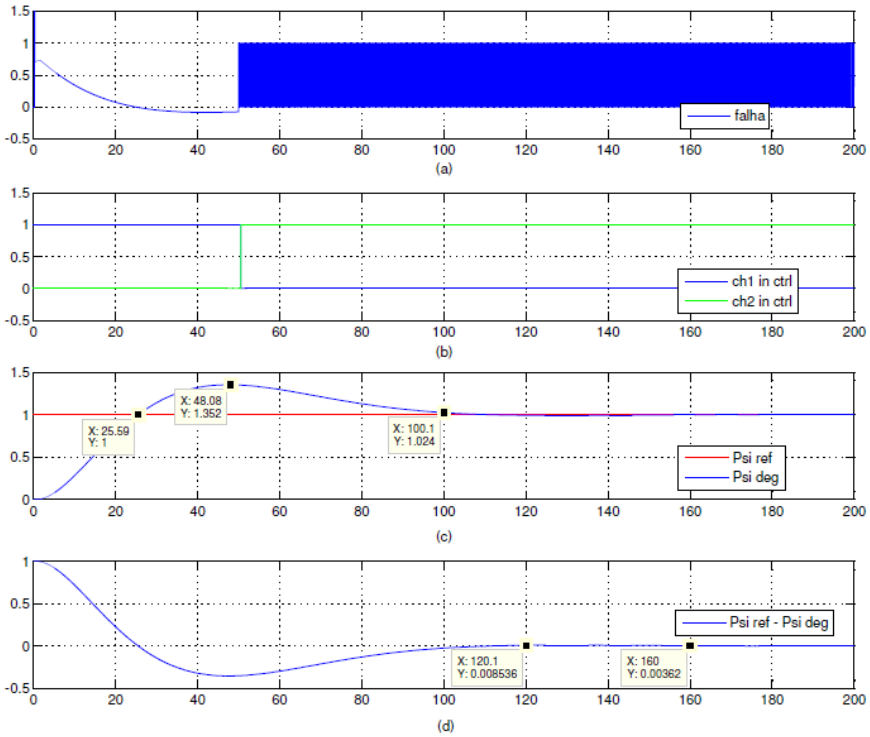

Figura 4.46 – Manobra de captura de Psi de 1° para a solução duplo-simplex, entrada degrau introduzida em t = 0s, falha constante introduzida no controlador da linha MON em t=50s (a) falha (1o amplitude), (b) canal 1 em controle, t<50s, e canal 2 em controle, t>50s , (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.46 mostra o resultado da simulação da solução duplo-simplex para uma falha em forma de uma seqüência de degraus inserida na saída do controlador da linha MON. Essa seqüência de degraus simula uma falha intermitente, que é uma variação freqüente de níveis altos e baixos,

característica, por exemplo, de conectores frouxos, ou conexões sujeitas a vibrações. Note que, ao ser detectada a falha, os canais trocaram de engajamento: o canal 1 estava em controle até t = 50s, depois da inserção da falha o canal 2 assumiu o controle. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,003°

b) Taxa de amortecimento:
requerido: $\xi = 0,3 < \xi < 0,8$
medido: $\xi = 0,315$

c) Tempo de acomodação (5%)
requerido: Psi medido (t = 100s) – Psi referência < 5%
medido: Psi medido (t = 100s) – Psi referência < 2,4%

d) “Drift” menor que 0,001°/s;
medido: 0,005° em 40s, ou 0,000125°/s

O mesmo procedimento de teste da solução duplo-simplex pode ser aplicado à solução triplo-simplex. Como as soluções usam elementos semelhantes (como os votadores de entrada e os monitores) é interessante focar os testes nas diferenças entre as duas soluções. Assim, a Figura 4.47 resume os pontos sugeridos para a solução triplo-simplex.

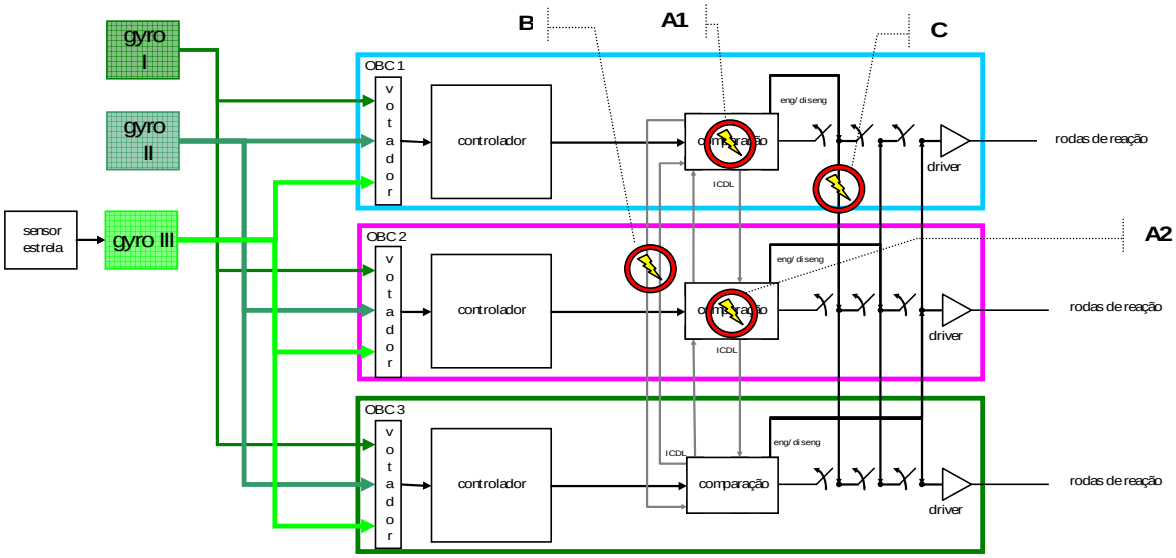

Figura 4.47 - Pontos de inserção de falhas da solução triplo-simplex para a verificação do requisito MPP-R-7.

A Tabela 4.4 resume os pontos de testes que serão estimulados.

Tabela 4.4 – Resumo dos pontos de inserção de falhas para verificação da aderência da solução triplo-simplex ao requisito MPP-R-7.

Falha	Ponto de inserção da falha	Tipo de sinal de falha	Detalhamento da falha
AA	Sem falha	Sem falha	Manobra de captura de Psi = 1^o
A1	Entrada do monitor de comparação (\pmm\channel1)	Falha de valor constante	Degrau de 10^o introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1^o
A2	A1 + Entrada do monitor de comparação (\pmm\channel2)	Falha de valor constante	Trata-se da continuação do caso anterior, A1. Após A1, em t= 100s, inclui-se mais uma falha, só que no canal 2. A segunda falha será um degrau de 1^o introduzido em t = 100s, durante manobra de captura de atitude de Psi = 1^o
B	Saída do controlador no canal 2 (\pmm\channel2)	Falha de valor constante	Degrau de 10^o introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1^o
C	Saída do moniotr no canal 2 (\pmm\channel2)	Falha de valor constante	Degrau de 1^o introduzido em t = 50s, durante manobra de captura de atitude de Psi = 1^o. A falha é introduzida no sinal que sai do canal 2 indicando falsamente que o canal 2 está falhado.

AA) Caso sem falha

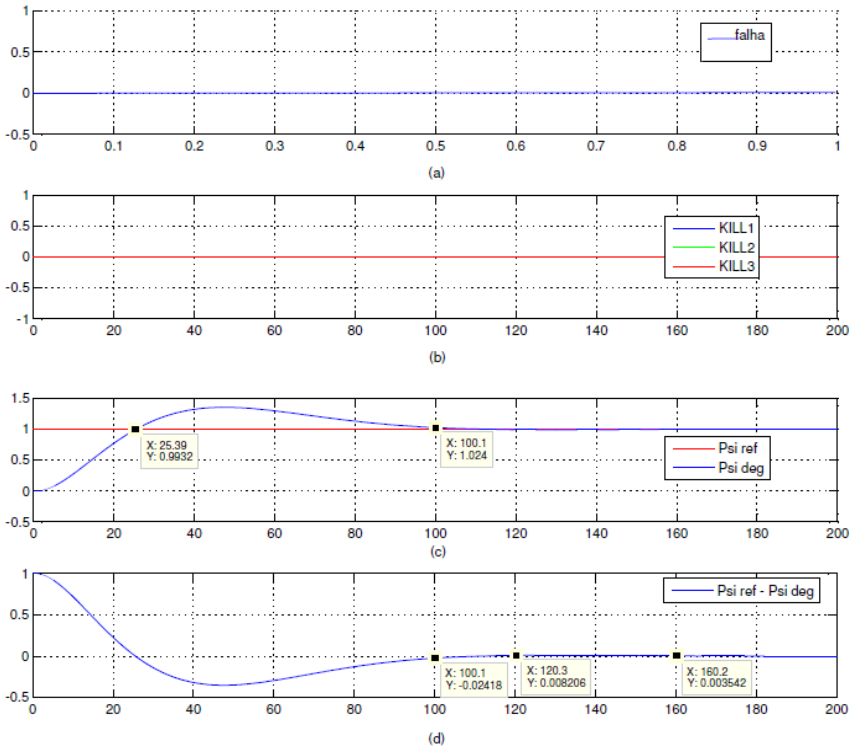

Figura 4.48 - Manobra de captura de Psi de 1° para a solução triplo-simplex, entrada degrau introduzida em t = 0s, sem falhas. (a) falha inserida = 0, (b) KILL1, KILL2 e KILL3 indicam qual canal foi passivado, (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.48 mostra o resultado da simulação da solução triplo-simplex para o caso sem falha. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:
requerido: erro Psi < 0,005°
medido: erro Psi < 0,003°

b) Taxa de amortecimento:
requerido: $\xi = 0,3 < \xi < 0,8$
medido: $\xi = 0,315$

c) Tempo de acomodação (5%)
requerido: Psi medido (t = 100s) – Psi referência < 5%

medido: Psi medido (t = 100s) – Psi referência < 2,4%

d) “Drift” menor que 0,001°/s;

medido: 0,005° em 40s, ou 0,000125°/s

A1) Falha de valor constante na entrada do monitor do canal1

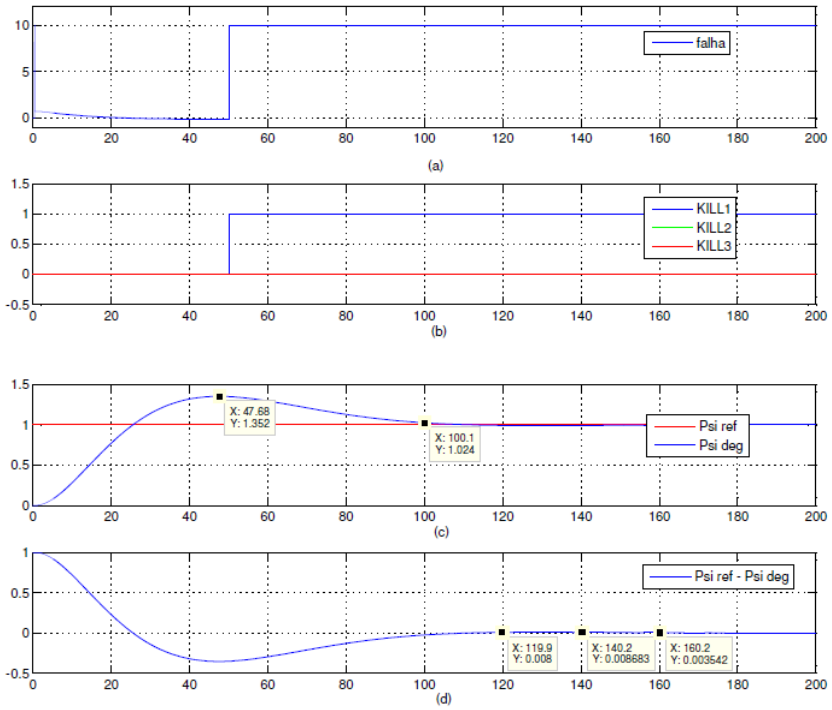

Figura 4.49 - Manobra de captura de Psi de 1° para a solução triplo-simplex, entrada degrau introduzida em t = 0s, falha de valor constante introduzida em t = 50s. (a) falha inserida no canal 1 (amplitude de 10°), (b) sinal referente ao engajamento dos canais (KILL = 1 significa que o canal deve ser desligado), (c) Psi referência e Psi medido, (d) diferença entre Psi referência e Psi medido.

A Figura 4.49 mostra o resultado da simulação da solução triplo-simplex para o caso de falha constante no canal 1. O sinal introduzido na simulação pretende representar o funcionamento faltoso do processador do canal 1. Abaixo, segue o resumo dos parâmetros de interesse para atendimento ao requisito MPP-R-7:

a) erro de determinação de atitude:

requerido: erro Psi < 0,005°

medido: erro Psi < 0,003°

b) Taxa de amortecimento:

requerido: $\xi = 0,3 < \xi < 0,8$

medido: $\xi = 0,315$

c) Tempo de acomodação (5%)

requerido: Psi medido (t = 100s) – Psi referência < 5%

medido: Psi medido (t = 100s) – Psi referência < 2,4%

d)“Drift” menor que 0,001°/s;

medido: 0,005° em 20s ou 0,00025°/s

No quadro (b) da Figura 4.49 o sinal “KILL1” indica o funcionamento correto do monitor de comparação, ou seja, o monitor identificou corretamente que o canal 1 estava falhado.

A Figura 4.50 mostra a atuação do monitor nos três canais. Todos eles detectaram a falha e ordenaram o desligamento do canal 1.

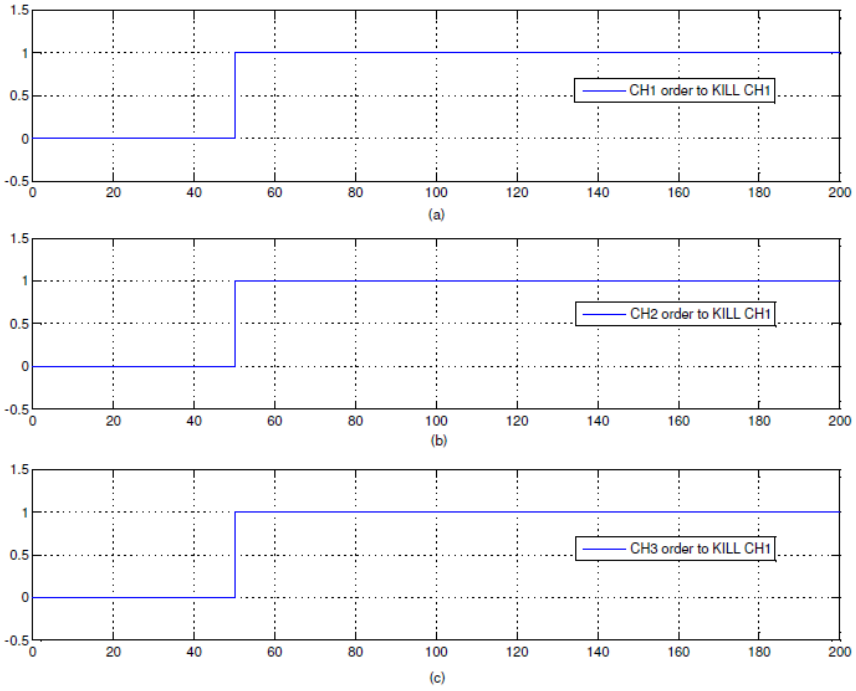

Figura 4.50 – Resultado do monitor de comparação de comandos nos três canais. Os três canais detectaram a falha e anunciaram em uníssono o desligamento do canal 1.

A2) Falha de valor constante na saída do controlador do canal 2

O objetivo desta falha não é mostrar o atendimento ao MPP-R-7 e sim comprovar o funcionamento correto da lógica de desligamento dos canais. A falha inserida no canal 2, após a falha inserida no canal 1, provoca o desligamento dos canais, uma vez que os dois canais remanescentes não conseguem definir entre si qual deles está correto. Assim, como o sistema é projetado para tolerar uma falha, após a segunda falha perde-se o controle da PMM. A Figura 4.51 apresenta o cenário de falhas nos canais e a perda de controle da plataforma.

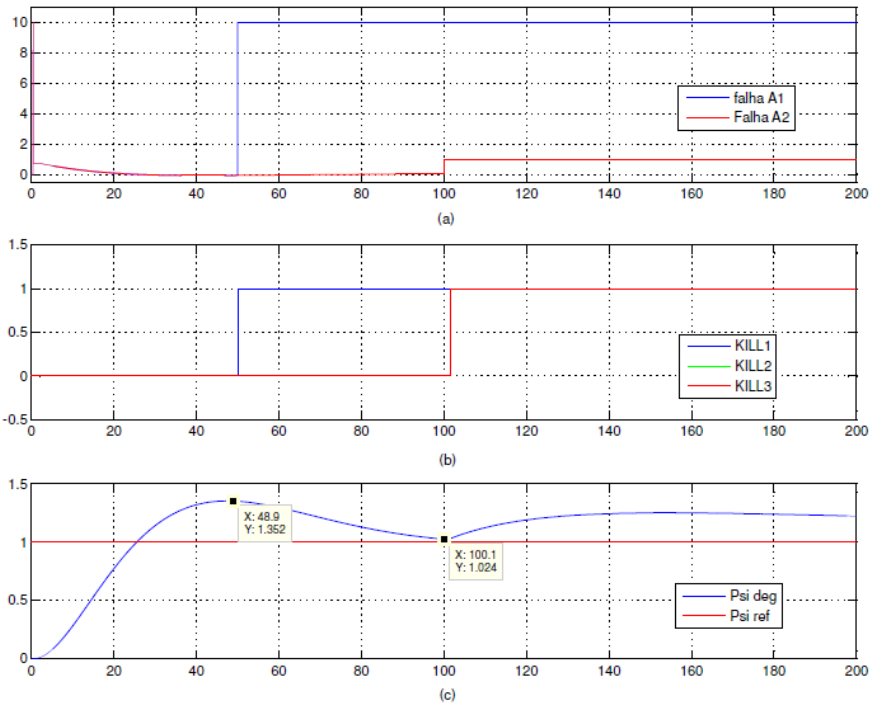

Figura 4.51 – Manobra de captura de Psi de 1° para a solução triplo-simplex, entrada degrau introduzida em t = 0s (a) falhas inseridas no canal 1 (amplitude de 10° em t = 50s) e canal 2 (amplitude de 1° em t = 100s), (b) KILL1, KILL2 e KILL3 são sinais que indicam o engajamento dos respectivos canais, (c) Psi referência versus Psi medido.

B) Falha de valor constante na saída do controlador do canal 2

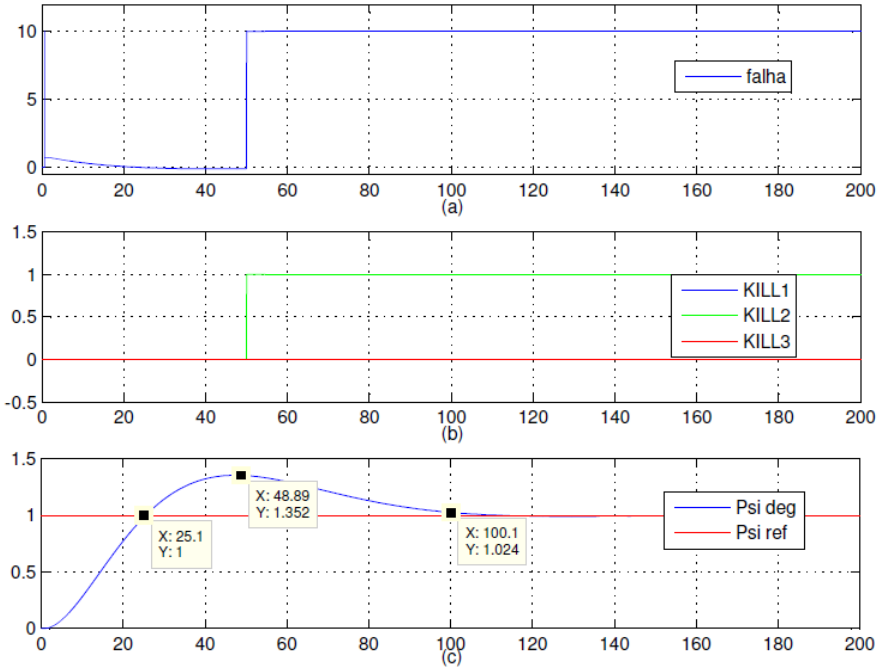

Figura 4.52 - Manobra de captura de Psi de 1° para a solução triplo-simplex, entrada degrau introduzida em t = 0s (a) falhas inseridas no canal 2 (amplitude de 10° em t = 50s), (b) KILL1, KILL2 e KILL3 são sinais que indicam o engajamento dos respectivos canais, (c) Psi referência versus Psi medido.

Assim como a falha anterior, a falha B tem o caráter mais de verificação da eficácia do engajamento e a robustez da solução do que cumprimento com os parâmetros do MPP-R-7. O canal 1 permanece engajado durante o tempo todo e assim cumpre com todos os requisitos de desempenhos impostos por MPP-R-7. O interessante da falha é notar que uma falha de comunicação entre dois canais não levou á perda do comando. O fórum entre os canais concluiu que o problema estava no canal 2 e o desligaram. A falha de comunicação é um dos pontos fundamentais na discussão de falhas Bizantinas, apresentada na seção 2.1.2.6.

C) Falha de valor constante na indicação de falha do canal 2 para o canal 1

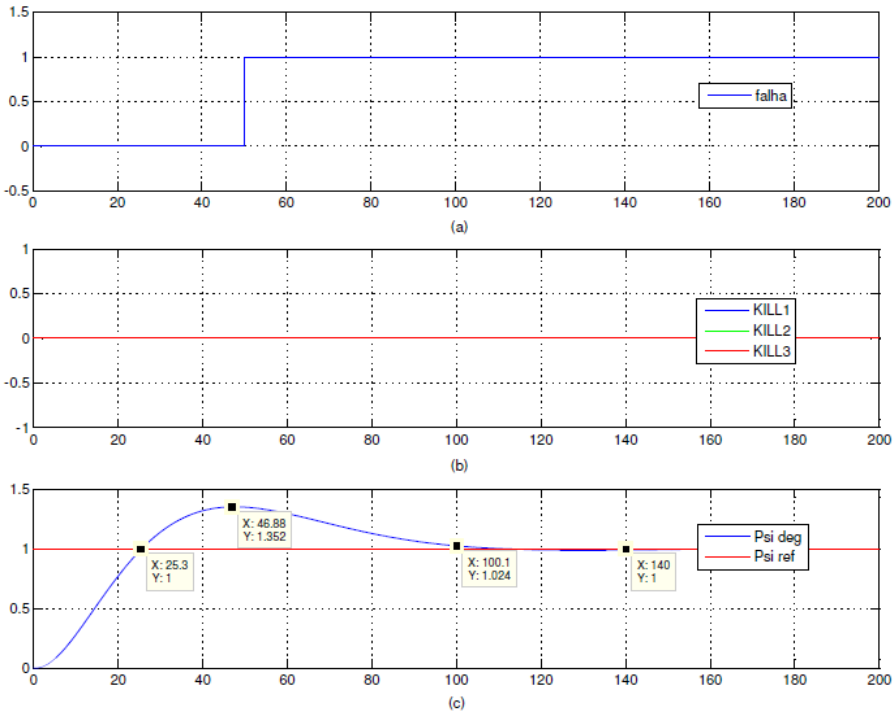

Figura 4.53 - Manobra de captura de Psi de 1° para a solução triplo-simplex, entrada degrau introduzida em t = 0s (a) falhas inseridas no canal 2 (amplitude de 1o em t = 50s), (b) KILL1, KILL2 e KILL3 são sinais que indicam o engajamento dos respectivos canais, (c) Psi referência versus Psi medido.

Assim com as 2 falhas (Casos A3 e B) anteriores, essa falha tem como finalidade testar a robustez da implementação, uma vez que como o canal 1 não perde o engajamento, os parâmetros requeridos em MPP-R-7 serão claramente atendidos. A falha simula uma indicação errônea para o canal 1 de que o canal 2 está falhado. Como só há uma indicação de falha do canal 2, a indicação é considerada espúria e o canal 2 continua operando em *standby* (lembre-se que são necessários no mínimo dois canais indicando a falha para que o canal seja desligado).

4.5.3 Verificação das soluções apresentadas – verificação por inspeção

O objetivo desta seção é mostrar a aderência das soluções propostas aos seguintes requisitos (apresentados na seção 4.3 e repetidos aqui por conveniência):

- MPP-R-3;
- MPP-R-4;
- MPP-R-5;
- MPP-R-6.

[MPP-R-3] O sistema de controle de atitude deve ser totalmente redundante.

[MPP-R-4] O sistema de controle de atitude deve eliminar todos os pontos de falhas simples.

[MPP-R-5] A Plataforma MultiMissão deve ser imune a perda de operação devida à falha simples.

[MPP-R-6] Na presença de falhas, o computador de bordo deve ter a capacidade de se re-configurar automaticamente.

Os requisitos acima, de certa forma, são variações do mesmo tema: a tolerância a falha simples. As seções anteriores cuidaram de mostrar que as soluções propostas atendem esses requisitos para parte do repertório de falhas proposto. Esta seção discutirá quais as falhas que não foram explicitamente incluídas nas seções anteriores e como as soluções propostas toleram uma falha de cada um dos tipos propostos.

O seguinte repertório de falhas foi proposto na seção 2.1.2.1:

a) Falha Aleatória (para mais detalhes veja 2.1.2.2);

b) Falha de projeto (para mais detalhes veja 2.1.2.3);

c) SEU (para mais detalhes veja 2.1.2.4);

d) Falha de modo comum (para mais detalhes veja 2.1.2.5);

e) Falha Bizantina (para mais detalhes veja 2.1.2.6).

A seguir cada uma dessas falhas será discutida:

1. Falha Aleatória:

As falhas Aleatórias foram bastante discutidas nas análises de Arvores de Falhas (seção 4.5.1) e simulação (seção 4.5.2).

2. Falha de projeto:

A falha de projeto pode ser analisada de acordo com a metodologia proposta por Manelli et al (17). A metodologia consiste primeiramente em identificar as funções de cada elemento do sistema. Depois, admitem-se duas classes de falhas para esse elemento: (i) não executar a sua função atribuída e (ii) executar erroneamente a sua função atribuída. Lembre que as falhas de projeto têm sentido para elementos de *software*, *hardware* complexos ou sistemas complexos e/ou altamente integrados.

As soluções duplo-simplex e triplo-simplex têm três elementos de interesse para essa análise: os dois tipos de sensores (sensor de estrela e giroscópios I e II) e o OBC. As rodas serão consideradas elementos simples e não farão parte desta análise (considera-se que é possível testar-se por completo uma roda de reação). Tabela 4.5 resume a análise para falhas de projeto das soluções propostas.

Tabela 4.5 – Aderência das soluções ao requisito sobre falhas de projeto.

Elemento do sistema	Função	Falha de projeto	Resultado da Falha (não considerar a mitigação)	Mitigação
Giroscópio	Fornecer dados de atitude do satélite.	(i) não fornecer dados de atitude; (ii) fornecer dados errôneos.	(i) interrupção do cálculo de comando para controle de atitude do satélite. (ii) cálculo errôneo do comando de controle de atitude.	(i) e (ii) os votadores implementados nas entradas de ambas as soluções garantem que os dados errôneos ou ausência de dados serão detectados, desde que se usem giroscópios dissimilares. Se os giroscópios I e II forem similares, uma falha do projeto do giroscópio será entendida como um comando normal, uma vez que os votadores procuram por maioria de 2 em três dados.
Sensor de estrelas	Fornecer dados de atitude do satélite (como empregado nas soluções).	(i) não fornecer dados de atitude; (ii) fornecer dados errôneos.	(i) e (ii) perda da terceira fonte para votação dos giroscópios. Analisando-se somente a falha simples, não há efeito nenhum para o sistema.	(i) e (ii) não é necessária. O votador de entrada indicará que o sensor de estrelas falhou.
OBC	Calcular os comandos para controle de atitude	(i) não calcular os comandos para controle de atitude;	(i) perda da PMM por falta de comando; (ii) perda da PMM por comando	(i) e (ii) Considerar componentes dissimilares para cada uma das linhas (duplo-simplex e módulo de Segurança) e

Elemento do sistema	Função	Falha de projeto	Resultado da Falha (não considerar a mitigação)	Mitigação
	do satélite.	(ii) calcular erroneamente os comandos de controle de atitude[12].	errôneo.	cada um dos canais (triplo-simplex). Veja análise a seguir.

[12] Note que esse comando errôneo aqui mencionado nada tem a ver com o comando errôneo considerado na análise de árvores de falhas da seção 4.5.1. Os dois têm o mesmo efeito, porém as causas são diferentes. Comandos errôneos por falhas Aleatórias são mesuráveis e previsíveis, ao passo que comando errôneos vindos de falhas genéricas são incomensuráveis e imprevisíveis.

A falha de projeto do OBC merece um pouco mais de discussão.

Caso se esteja analisando a solução duplo-simplex, uma falha que cause a perda da capacidade de cálculo dos comandos levará à perda total de comando da plataforma. Isso é possível se considerarem-se as linhas COM e MON especificadas com os mesmos componentes (principalmente os mesmos microprocessadores). Desse modo, a falha acometerá as duas linhas (COM e MON) do OBC ao mesmo tempo. Assim, nenhuma discrepância será detectada pelos monitores e não haverá chaveamento para o módulo de Segurança. Uma mitigação para esse cenário pode ser feita propondo-se componentes dissimilares para as duas linhas. Assim, se pode advogar que as falhas de projeto de uma linha não acometerão a outra. A falha será detectada e o módulo de Segurança governará a plataforma.

Caso se esteja analisando a solução triplo-simplex, as mesmas considerações acima mencionadas para o duplo-simplex são válidas aqui. A diferença é que se tem que pensar em dissimilaridade para os três canais.

3. SEU:

Há dois níveis em que se pode fazer a análise para SEU: nível de (i) componentes e (ii) sistemas. A análise de componentes se encarregará de verificar o quanto a tecnologia empregada nos componentes eletrônicos é susceptível a SEUs; e, caso seja, propor meios de correção dos erros gerados pelo evento.

No caso de uma análise de sistemas, que é o interesse desse trabalho, admiti-se que houve um SEU e o sistema precisa se re-configurar. Nas duas soluções propostas, qualquer um dos elementos do OBC, sejam os canais da solução triplo-simplex ou as linhas da solução duplo-simplex (e o módulo de Segurança), que sofra uma perturbação advinda de um SEU, os monitores e votadores das outras linhas/canais o detectarão. Isso é válido porque se admite que somente um evento de SEU acometerá um dos elementos por vez. Essa

hipótese é razoável desde que os elementos estejam acondicionados em compartimentos diferentes na espaçonave, garantindo segregação física.

4. Falhas de modo comum

A análise de susceptibilidade das soluções a falhas de modo comum começa levantando-se possíveis fontes de falhas comuns. Duas fontes já foram discutidas no trabalho: falhas de projeto e SEUs. Qualquer fonte de calor extremo (ou frio extremo), alta corrente elétrica (por ser uma fonte emissiva de campos elétricos), fluidos (hidráulicos, refrigerantes, etc.), molas ou cabos com alta energia potencial, pás rotativas, etc., podem ser fontes internas de falhas de modo comum à plataforma. Há ainda fontes externas como SEU ou detritos espaciais. Note que nem sempre se associa probabilidades a falhas de modo comum, pois não é o objetivo da análise prover uma solução usando-se números, ainda mais quando se pode perder a plataforma toda com uma falha simples. O resultado da análise de falhas de modo comum deve ser mitigações para cada um dos riscos identificados.

Para as soluções apresentadas há algumas áreas candidatas a fontes de falhas de modo comum. Para a solução duplo-simplex: a fonte de alimentação, os circuitos de entrada, os circuitos de interface, etc., podem produzir falhas que afetem as duas linhas concomitantemente e assim impedir que uma possível falha seja identificada e o comando seja transferido para o módulo de Segurança.

No caso da solução triplo-simplex, os barramentos de dados interconectam todos os canais, sendo uma possível fonte de falha de modo comum. Uma sobrecorrente em um dos *drivers* de saída pode induzir falhas nos demais, causando o colapso do sistema. Assim, circuitos de proteção á sobrecorrente em cada um dos *drivers* deve ser projetado.

Pela sua própria concepção, as duas soluções foram propostas para serem capazes de tratar de falhas de modo comum. Basta que todas sejam identificadas e os devidos cuidados tomados.

5. Falha Bizantina

As duas soluções foram concebidas para atender ao critério de prevenção a falhas Bizantinas (i.e. 3 x m elementos para mensagens escritas, onde m = número de falhas), como foi descrito na seção 2.1.2.6. A própria seção 4.5.2, que simulou várias falhas tanto internas quanto externas às soluções, pode ser usada como evidência do atendimento das soluções ao cenário de falhas bizantinas.

4.6 Comparação das propostas apresentadas

A verificação das duas soluções ante os requisitos apresentados é um bom ponto de partida para compará-las. A Tabela 4.6 resume os requisitos e como as soluções se comportaram diante deles.

Tabela 4.6 – Resumo do cumprimento dos requisitos propostos.

	Atende ao requisito? [sim/não]	
Requisito	Duplo-simplex	Triplo-simplex
MPP-R-1	Sim	Sim
MPP-R-2	Não	Sim
MPP-R-3	Sim	Sim
MPP-R-4	Sim	Sim
MPP-R-5	Sim	Sim
MPP-R-6	Sim	Sim
MPP-R-7	Não	Sim
MPP-R-8	Não	Sim

Uma análise simplista da Tabela 4.6 poderia levar á conclusão de que das duas soluções apresentadas somente a triplo-simplex satisfaz aos requisitos propostos. Porém, ao se recapitular o resultado dos testes e análises de verificação, a solução duplo-simplex não cumpriu os requisitos por uma pequena diferença. Por exemplo, a Confiabilidade do ACDH com a solução duplo-simplex ficou em 0,91 contra 0,94 do requisito. O desempenho da duplo-simplex também não foi suficiente para cumprir com o requisito MPP-R-7, mas a diferença também foi pequena: ξ = 0,23 contra 0,3 do requisito, erro (t=100s) < 0,058° contra < 0,05° do requisito e Psi medido (t = 100s) – Psi referência < 5,8% contra 5% do requisito. Há de se considerar que os dados usados são de uma fase muito preliminar do projeto, e como se sabe, os dados iniciais sempre são conservativos de maneira a se construir uma solução mais robusta. Assim, é possível que a solução duplo-simplex possa atender os requisitos com a evolução dos dados e detalhes do projeto.

É inegável também que a solução triplo-simplex cumpriu os requisitos com muito mais folgas, mesmo se levando em conta o conservadorismo dos dados preliminares do projeto. Além de seu melhor desempenho em números, a solução triplo-simplex agrada também por permitir que os seus três computadores comandem a plataforma, enquanto que na solução duplo-simplex somente a linha COM e o módulo de Segurança são capazes de gerar comandos. Isso representa um aproveitamento total dos recursos computacionais disponíveis, enquanto na solução duplo-simplex a linha MON não é usada ativamente, i.e., não comanda diretamente a PMM. Outra vantagem da solução triplo-simplex é monitorar as suas falhas continuamente, minimizando falhas latentes, como é o caso do módulo de Segurança, que depende de um teste periódico para garantir o seu funcionamento.

Em contrapartida, a solução triplo-simplex exige que três computadores sejam alocados em espaços próprios e diferentes daqueles dos dois outros computadores, para se prevenir falhas de modo comum. Além disso, para se

garantir a independência dos computadores é necessário que cada um deles tenha a sua própria alimentação e o seu próprio módulo de entrada/saída (I/O). Isso significa mais consumo de energia, mais dissipação de calor, e um maior espaço necessário. Os fios dos barramentos de dados da solução triplo-simplex também podem ser um problema dado o espaço limitado do projeto, além de aumentar a susceptibilidade da solução a efeitos eletromagnéticos.

Por outro lado, o canal duplo-simplex pode compartilhar a alimentação, o módulo de entrada/saída, e só tem um *driver* de saída, que, com mais um canal do módulo de Segurança, somam dois *drivers* na solução contra três da solução triplo-simplex. A solução duplo-simplex é mais compacta que a triplo-simplex, o que pode ser valioso no projeto de um satélite, onde o espaço é um bem precioso.

Outro aspecto importante a ser comparado é a dificuldade para se desenvolver a solução. O tempo empregado no projeto (i.e., projeto conceitual, detalhamento, implementação e testes), às vezes é uma das variáveis mais importantes a ser considerada, dadas as limitações de tempo e custo. A solução triplo-simplex, por conta da integração dos três módulos, pode apresentar maiores dificuldades de desenvolvimento como, por exemplo, àquelas advindas da definição e integração dos barramentos de comunicação, o acerto dos monitores por conta dos atrasos de comunicação, o *drift* natural dos integradores (que, ao longo do tempo, pode provocar o acionamento espúrio dos monitores), etc. Já a solução duplo-simplex é mais simples em sua concepção: o módulo de Segurança será uma versão simplificada da aplicação do módulo duplo-simplex, os atrasos serão menores no módulo duplo-simplex (a transferência de dados pode se dar por meio de compartilhamento de uma memória, por exemplo), a linha MON pode ser uma versão simplificada da linha COM, etc.

A Tabela 4.7 resume a discussão acima em torno das soluções.

Tabela 4.7 – Resumo da comparação das soluções frente os requisitos e outros aspectos.

	Solução que melhor atende o critério	
Aspecto	Duplo-simplex	Triplo-simplex
Atendimento aos requisitos propostos		X
Espaço alocado para solução	X	
Calor dissipado	X	
Energia consumida	X	
Simplicidade do desenvolvimento	X	

Os resultados e comparações apresentados até então podem parecer divergentes para os projetistas, pois ao passo que a solução triplo-simplex atendeu melhor os requisitos, a solução duplo-simplex é mais vantajosa em vários aspectos. Isso é evidência que as duas soluções são adequadas, porém para projetos diferentes. Explica-se: a solução triplo-simplex é mais robusta, é mais autônoma e tem maior poder computacional, porém cobra o seu preço em espaço, dissipação de calor, consumo de energia e dificuldade de desenvolvimento; a solução duplo-simplex é mais simples para se desenvolver, mais compacta e consome e dissipa menos energia. Projetos com um tempo maior de operação e sem restrição de tempo de desenvolvimento encontram na solução triplo-simplex uma forte candidata. Projetos mais curtos (menor tempo de operação) e com menor tempo de desenvolvimento podem considerar a solução duplo-simplex.

5 CONCLUSÕES, RECOMENDAÇÕES E SUGESTÕES PARA TRABALHOS FUTUROS

5.1 Conclusões

Este trabalho se propôs a estudar os requisitos e especificações para a tolerância a falha simples do sistema de controle de atitude da Plataforma MultiMissão.

A revisão bibliográfica apontou um aspecto importantíssimo das soluções tolerantes à falha: elas são meios para se aumentar a Dependabilidade e não o fim do projeto em si. Sistemas tolerantes a falha comparados a sistemas não tolerantes são mais complexos, sua implementação é mais desafiadora, são mais caros e demandam mais tempo. Assim, a sua especificação deve passar primeiro por um severo estudo de Dependabilidade e os requisitos associados a seus atributos: Confiabilidade, Disponibilidade, Segurança, Proteção e Integridade. Tão importante quanto o estudo dos atributos da Dependabilidade é o estudo das falhas associadas ao projeto. A verificação da adequação da solução proposta deve necessariamente cobrir o repertório de falhas definido para o projeto.

O detalhamento do estudo de caso mostrou alguns dos muitos desafios da implementação dos sistemas tolerantes à falha. A síntese dos monitores talvez tenha sido o maior deles. Principal elemento do mecanismo de detecção e isolamento da falha, o monitor é um dos elementos mais desafiadores do projeto. A determinação incorreta do seu elemento de comparação pode representar a manifestação de uma falha incontida ou a interrupção da operação por conta dos indesejáveis falsos alarmes. Durante a síntese da solução todo tempo gasto na determinação dos valores corretos da tolerância e da persistência é bem empregado. Outro ponto importante do desenvolvimento

das soluções é a garantia contínua da independência dos elementos redundantes (ausência de falhas de modo comum).

Deve-se tomar muito cuidado com os atrasos do sistema. Tome como exemplo o caso da solução duplo-simplex. O atraso de um *frame* no chaveamento dos canais levou ao não cumprimento dos requisitos. Outro desafio da comparação entre os canais foi alinhar os comandos calculados com as mesmas entradas para se diminuir as chances de alarmes falsos. Esse alinhamento pode justificar a inserção proposital de atrasos, que, infelizmente, podem levar ao problema de chaveamento da solução duplo-simplex relatado no começo do parágrafo.

A análise constante da aderência das soluções aos requisitos, principalmente àqueles de Disponibilidade, Integridade e Confiabilidade, por meio de árvores de falhas é uma ferramenta poderosíssima na tomada de decisões que afetem a arquitetura das soluções. A Árvore de Falhas não deve ser uma ferramenta de verificação final de aderência aos requisitos, depois que a solução esteja pronta. As árvores de falhas do projeto devem ser mantidas vivas e modificadas de acordo com as evoluções de projeto. Desse modo garante-se o cumprimento dos requisitos durante todo o projeto, ou podem-se identificar ações corretivas o mais cedo possível para alterar o projeto.

A comparação entre as arquiteturas mostrou que não há uma prevalência de uma sobre a outra de forma absoluta. Há uma maior adequação de uma certa solução com relação a outra se considerar-se aplicações específicas. A análise de solução de compromisso ao começo do projeto deve determinar qual a solução preferida para uma determinada aplicação.

5.2 Recomendações e sugestões para trabalhos futuros

Ao longo do processo de desenvolvimento das soluções por várias vezes foi necessário abandonar a discussão de um determinado tema por conta de sua grande extensão e em prol dos objetivos determinados no começo do trabalho. A riqueza de assuntos em torno da Dependabilidade é muito grande. Dentre as várias possibilidades de temas a serem desenvolvidos por trabalhos futuros destacam-se os seguintes:

a) Comparação entre os diversos mecanismos para o aumento da Dependabilidade, em especial entre Prevenção a falhas e Tolerância a falhas.

b) Falhas em sistemas altamente complexos e integrados. Com o aumento constante da integração de sistemas, principalmente promovido pela facilidade de integração digital por meio de barramentos seriais, há um aumento proporcional da dificuldade de se garantir a independência de elementos redundantes no sistema.

c) Uso de componentes comerciais (COTS) em sistemas que demandem alta Dependabilidade. Os prós e os contras em se especificar COTS é um assunto extenso e pouco explorado. De um lado a grande diversidade e Disponibilidade, á baixo custo, de componentes são uma grande vantagem para o projeto. De outro lado a presença inconveniente de erros de projeto podem impedir o uso desses componentes em aplicações de alta Dependabilidade.

d) Métodos de projeto e análise de monitores. Por conta de sua importância vital no projeto, os monitores merecem um trabalho à parte. Não só a sua síntese, mas a verificação de sua eficácia e robustez são temas férteis e de grande importância para o projeto.

e) Métodos para se garantir a sincronia de integradores. Outra grande fonte de falsos alarmes de monitores é a sincronia (ou a falta de) entre os integradores das diversas linhas e módulos. A simples inserção de um gatilho comum não é uma solução aceitável, já que ele se torna um elemento de falha comum entre os canais independentes.

f) Taxa de falhas de componentes. As referências usadas para se determinar as taxas de falhas de componentes são antigas e muitas vezes não acompanharam a evolução das tecnologias.

g) Estudo das diversas normas disponíveis para se aumentar a Dependabilidade do sistema, tais como ARP4745, ARP4761, DO254 e DO178B.

h) Comparação de requisitos e arquiteturas da área espacial com o caso aeronáutico. A pesar de compartilharem muitos aspectos em comum, as áreas espacial e aeronáutico guardam muitas particularidades.

i) Comparação das vantagens e desvantagens em se propor um sistema com reconfiguração automática comparado a um sistema monitorado pela estação em Terra e que utilize redundância a frio.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ENGINEERING SOCIETY FOR ADVANCING MOBILITY, SAE. **SAE ARP4754**. Warrendale - PA, EUA, 1996. 88p.

[2] ANDERSON, T.; LEE, P. A. **Fault tolerance:** principles and practice. 1. ed. EUA: Prentice Hall, 1981. 369p. ISBN 0-13-308254-7.

[3] AVIŽIENIS, A. Design of fault-tolerant computers. In: FALL JOINT COMPUTER CONFERENCE, 1967, New York, EUA. **Proceedings…** New York-NY, EUA: ACM, 1967, v. 31. p. 733–743.

[4] LALA, J. H.; HARPER, R. E. Architectural principles for safety-critical real-time applications. In: IEEE CONFERENCE, EUA, 1994. **Proceedings of IEEE.** 2002, v. 82. p. 25-40. ISSN: 0018-9219.

[5] LAPRIE, J. C. **Dependability:** basic concepts and terminology. Viena, Áustria: Editora Springer, 1992. 265p.

[6] SOUZA, M. L. O.; CARVALHO, T. R. The Fault avoidance and the fault tolerance approaches for increasing the reliability of aerospace and automotive systems. In: SAE BRASIL, 2005, São Paulo, Brasil. **Anais...** Disponível em www.sae.com. Acesso em: 22 mai. 2011.

[7] HEIDERGOTT, W. SEU Tolerant Device, Circuit and Processor Design. In: CONFERENCE ON DESIGN AUTOMATION (DAC), 2005, Anaheim-CA, EUA. **Proceedings....** Disponível em www.acm.org. Acesso em: 22 mai. 2011.

[8] DEPARTAMENT OF DEFENSE, DoD. **MIL-HDBK-217F**. Washington DC, EUA, 1990. 205p.

[9] KOPETZ, H. **Real-time systems:** design priciples for distributed embedded applications**.** EUA: Kluwer Academic Publishers, 2002. xiv+338p. ISBN livro 0-792-39894-7. ISBN eletrônico 0-306-47055-1.

[10] FEDERAL AVIATION ADMINISTRATION – FAA. **Special condition no. 25-357-SC**. EUA, 2008. Disponível em www.faa.gov. Acesso em: 09 jun. 2010.

[11] FEDERAL AVIATION ADMINISTRATION – FAA. **Data network evaluation criteria report, DOT/FAA/AR-09/27.** EUA, 2009. Disponível em www.faa.gov. Acesso em: 09 jun. 2010.

[12] FEDERAL AVIATION ADMINISTRATION – FAA. **FAA requirements, FAR 25.1309.** EUA. Disponível em www.faa.gov. Acesso em: 09 jun. 2010.

[13] FEDERAL AVIATION ADMINISTRATION – FAA. **FAA advisory circular AC 25.1309.** EUA. Disponível em www.faa.gov. Acesso em 09 jun. 2010.

[14] ENGINEERING SOCIETY FOR ADVANCING MOBILITY – SAE. **SAE ARP4761**. Warrendale - PA, EUA, 1996.

[15] RADIO TECHNICAL COMISSION FOR AERONAUTICS – RTCA. **Software considerations in airborne systems an equipment certification, DO178 revision B.** EUA, 1992. 112p.

[16] RADIO TECHNICAL COMISSION FOR AERONAUTICS – RTCA. **Design assurance guidance for airborne electronic hardware.** EUA, 2000. 137p.

[17] MANELLI, H. N.; SOUSA, G. B.; SOUZA, M. L. O; Use of dissimilar hardware architecture to mitigate design errors in a flight control system application. In: SAE BRASIL, 2009. São Paulo, Brasil, 2009. **Anais...** Disponível em www.sae.org. Acesso em 09 jun. 2010.

[18] AVIŽIENIS, A. **Design diversity and the immune system paradigm:** cornestones for information system survivability, Los Angeles-CA, EUA: University of California, 2000. 4p.

[19] SOBECK, C. **Online interview with NASA Probe Engineering.** Disponível no site http://quest.nasa.gov/galileo/webchat/galtrans1.html. Acesso em 20 jun. 2010.

[20] NORMAND, E.; WERT, J. L.; MAJEWSKI, P. P.; OBERG, D. L.;BATHOLET, W. G.;DAVIS, S. K.;SHOGA, M.; WENDER, S. A.; GAVRON, A. single event upset and latchup measurements in avionics devices using the WNR Neutron Beam and a New Neutron-Induced Latchup model. In: IEEE RADIATIONS EFFECTS DATA WORKSHOP. **Proceedings….** EUA, IEEE, 1995, 33p. Disponível em: http://www.boeing.com/assocproducts/radiationlab/publications/. Acesso em 20 jun. 2010.

[21] LAMPORT, L.; SHOSTAK, R.; PEASE, M. The byzantine generals problem. **ACM Transactions on Programming Languages and Systems.** v.4, no. 3. p. 382-401,1982. Disponível em www.acm.org. Acesso em 20 jun. 2010.

[22] PATTON, F. **Fault diagnosis in dynamic systems:** theory and applications**.** New York, NY, EUA: Prentice-Hall, 1989. 602 p.

[23] SOUZA, M. L. O. **Estudo e desenvolvimento de um sistema de controle de atitude ativo em três eixos para satélites artificiais usando**

atuadores pneumáticos a gás frio e volantes de reação. 1980. Dissertação Mestrado em Ciência Espacial – INPE, São José dos Campos, SP, 1980. 147f.

[24] SOUZA, L. C. G. **Controle de atitude de um satélite artificial através da extensão da teoria do regulador linear quadrático.** 1987. Dissertação Mestrado em Ciência Espacial – INPE, São José dos Campos, SP, 1987. 123f.

[25] PRUDENCIO, S. V. **Simulação digital em tempo real de um sistema de controle de atitude magnético autônomo de um satélite.** 1997. Dissertação de Mestrado - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1997. 167 p.

[26] GOBATO, M. F.; **Controles monovariáveis e multivariáveis aplicados a sistemas aeroespaciais fracamente ou fortemente acoplados.** 2006. Dissertação de Mestrado – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006.

[27] MOREIRA, M. L. B. **Projeto e simulação de um controle discreto para a plataforma MultiMissão e sua migração para um sistema operacional de tempo real**. 2006. Dissertação de Mestrado – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006. 181 p.

[28] LEITE, A. C. **Detecção e diagnóstico de falhas em sensores e atuadores da plataforma MultiMissão.** 2007. Dissertação de Mestrado – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2007. xxiii+333 p.

[29] LUSTOSA, H. D. **Influência de tipos de barramentos e suas características de alto nível em sistemas de controle por rede.** 2009. Dissertação de Mestrado – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009. 246 p.

[30] OGATA, K. **Modern control engineering.** 4. ed. EUA: Pearson Education, 2006. 976p. ISBN 81-317-0311-8.

[31] HUGHES, P. C. **Spacecraft attitude dynamics.** Mineola-NY, EUA: Dover Publications INC, 2004. 570 p. ISBN 0-486-43925-9.

[32] NATIONAL AERONAUTICS and SPACE ADMINISTRATION – NASA. **Document control number C- 119162.** EUA, 1965. 291 p. Disponível em http://www.ibiblio.org/apollo/Documents/GeminiManualVol1Sec2.pdf. Acesso em 20 jun. 2010.

[33] WIKIPEDIA. Command and data handling (CDH). http://en.wikipedia.org/wiki/Galileo_(spacecraft)#cite_note-43 . Acesso 21 jun. 2010.

[34] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION – NASA. **GOES I-M Databook.** EUA, 1996. 186p.

[35] TOMAYKO, J. **Computers in spaceflight:** the NASA experience. EUA: National Aeronautics and Space Administration – NASA. Disponível em http://history.nasa.gov/computers/Ch5-6.html. Acesso 21 jun. 2010.

[36] HANAWAY, J. F. MOOREHEAD, R. W. **Space shuttle avionics system.** EUA: National Aeronautics and Space Administration – NASA. 1989. 82p.

[37] INSTITUTO NACIONAL DE PESQUISA ESPACIAIS – INPE. **Multi-Mission Platform Data Package for System Requirements Review.** São José dos Campos-SP, Brasil, 2001. 383p. (Relatório número A822000-DPK-01/D5a)

[38] INSTITUTO NACIONAL DE PESQUISA ESPACIAIS – INPE. **Multi-mission platform attitude control and data handling (ACDH) subsystem equipment specification**.. São José dos Campos-SP, Brasil, 2001. 35p. (Relatório número A822700-SPC-02/03).

[39] INSTITUTO NACIONAL DE PESQUISA ESPACIAIS – INPE. **Multi-mission platform attitude control and data handling (ACDH) subsystem specification.** São José dos Campos-SP, Brasil, 2001. 45p. (Relatório número A822700-SPC-01/04).

[40] KRSTIC, M. D.; STOJCEV, M. K.; DJORDJEVIC, G. L; ANDREJIC, I. D. A **Mid-value select voter.** Servia e Montenegro: Faculdade de Engenharia Eletrônica, Universidade de Nis. 7p.

[41] BUTLER, R. W. **A** primer **on architectural level fault tolerance.** Virginia, EUA: Langley Research Center, 2008. 53p.

[42] INSTITUTO NACIONAL DE PESQUISA ESPACIAIS – INPE. **Manual para elaboração, formatação e submissão de teses, dissertações e outras publicações do INPE.** São José dos Campos, Brasil, 2010. xv + 97p.

GLOSSÁRIO

Canal – no contexto deste trabalho é chamado de canal todo elemento independente de uma redundância. Assim, por exemplo, na solução duplo-simplex mais o módulo de Segurança, diz-se que há dois canais, o duplo-simplex e o módulo de Segurança. Já na solução triplo-simplex são três os canais.

Confiabilidade - Confiabilidade é a probabilidade condicional que um sistema permanecerá operacional, sem interrupções. Confiabilidade é uma medida da continuidade de operação.

Dependabilidade – É uma característica dos sistemas que expressa o seu funcionamento correto de acordo com a especificação do usuário. A Dependabilidade de um sistema pode ser determinada através de seus atributos: Disponibilidade, Confiabilidade, Segurança, Proteção e Integridade.

Disponibilidade - Disponibilidade é a probabilidade que um sistema estará acessível em um instante particular. Disponibilidade é uma medida da prontidão do sistema.

Erro - É qualquer estado que o sistema assuma que diverge de sua especificação. Uma seqüência de estados errôneos pode ou não levar à Falência do sistema, mas se pode dizer que toda Falência começou com um Erro.

Falência - Falência é a inabilidade de um sistema em realizar as suas funções. Falência é o evento perceptível em nível do usuário, pois é quando o sistema não responde mais às suas expectativas. Por exemplo, quando o sistema pára de funcionar ou está funcionando de forma errada, diz-se que houve sua Falência.

Falha – É a causa hipotética ou confirmada de um erro ou a causa do erro a ser evitada.

***Hardware* complexo** – Diz-se que um *hardware* é complexo quando não há um conjunto finito de testes que possa garantir a verificação completa daquele componente. Finito aqui pode ser entendido como passível de ser executado durante o tempo estipulado do projeto.

Integridade - é o atributo da Dependabilidade que passa ao usuário a credibilidade que a resposta do sistema é crível, está correta.

Limite de Comparação – É um dos parâmetros do monitor, junto com Limite de Comparação. Trata-se da máxima diferença aceitável pelo monitor entre o valor normal e o medido.

Linha – Denominação usada na solução duplo-simplex para identificar cada um dos elementos do canal. Na solução duplo-simplex há duas linhas, COM e MON.

Persistência – É um dos parâmetros do monitor, junto com Limite de Comparação. É o tempo entre a grandeza monitorada ter excedido o Limite de Comparação e o acionamento do monitor. Pode ser entendido também como o tempo máximo que o monitor tolera que a grandeza monitorada esteja acima do Limite de Comparação.

Tolerância a Falha – Constitui um dos meios de melhoria da Dependabilidade do Sistema.

Simplex – É um adjetivo que indica a unidade. Na solução duplo-simplex, por exemplo, há duas unidades que compõe o módulo, a linha COM e a linha MON. Na solução triplo-simplex há três unidades.

Verificação – No contexto do trabalho é processo pelo qual se comprova o funcionamento do sistema de acordo com um determinado requisito. O processo de verificação pode se dar através de um teste, uma análise ou uma simulação.

APÊNDICE A - Árvore de Falhas e taxas de falhas usadas no trabalho

A.1 Árvore de Falhas – baseada na especificação da PMM

A Árvore de Falhas apresentada na seção 4.3.1 foi baseada nas informações obtidas no documento de especificação da PMM (37), (38) e (39). Repete-se aqui (veja Figura A.1), por conveniência, as tabelas com taxas de falhas (ou sucesso) e probabilidades de falhas (ou sucesso) extraídas do documento e usadas como referência.

Equipment/Subsystem	Redcy	Type	Fits	Probability	Reliability
POWER					
String (40)			40,00	0,9986	
11 Circuit	7/9				1,0000
1 Circuits	8/9				0,9999
Total Sag					0,9999
SADA			100,00	0,9965	0,9930
PCDU	1/2	HOT	3500,00	0,8846	0,9867
BATERY CELL			24,00	0,9992	
BATERRY(21/22)					0,9998
Total PSS					0,9796
TMTC					
antenna(2)	NO		96,00	0,9966	0,9933
cables	NO		16,20	0,9994	0,9989
hybrid	NO		16,90	0,9994	0,9994
Transponder	1/2	Stby	2701,04	0,9097	0,9958
Total TM&TC					0,9874
ACDH					
gyro (1/2)	1/2	Hot	490,00	0,9830	0,9997
R. wheel			775,00	0,9732	
Reaction wheel (3/4)			775,00	0,9732	0,9958
Star sensor (2)	NO			0,9955	0,9910
GPS	1/2	Hot	483,00	0,9832	0,9997
Sun sensor (6/8)			44,00	0,9985	1,0000
Magnetotorquer			10,00	0,9996	
Magnetotorquer	1/2	Hot		1,0000	
Magnetotorquer(6)					1,0000
Magnetometer	1/2	Hot	780,00	0,9730	0,9993
OBC					0,9600
Total ACDH					0,9462

Equipment/Subsystem	Redcy	Type	Fits	Probability	Reliability
Structure					0,9999
Thermal Control					
Heaters			10,00	0,9996	1,0000
Thermistor			15,00	0,9995	0,9974
Total Thermal Control					0,9974
Propulsion					
Thruster Unit	NO		150,00	0,9948	0,9948
4 thrusters					0,9792
Transducer	NO		250,00	0,9913	0,9913
Valve	NO		109,00	0,9962	0,9962
Thermocouples	NO		74,30	0,9974	0,9974
Tank	NO		40,00	0,9986	0,9986
Total propulsion					0,9631
HD&Dep. Mech					0,9999
Harness					0,9980
Total MMP					0,8772

Figura A.1– Taxas de falhas, redundância e probabilidades associadas com a PMM usadas durante o trabalho. Fonte: Documento de requisitos da PMM (38).

A Árvore de Falhas foi calculada em termos de probabilidade de falha, assim todas as probabilidades de sucesso foram transformadas em probabilidades de falha (sucesso = 1 – falha). As árvores de falha deste trabalho têm como entrada taxas de falha na unidade falha/h, por uso da convenção aeronáutica. Assim, as probabilidades apresentadas na Figura A.1 foram divididas por 35040 horas, que correspondem a 4 anos de operação contínua.

A árvore completa é apresentada a seguir (desde a Figura A-2 até a A-10). Note que há um evento na árvore marcado em verde. Este evento indica uma possibilidade de melhora na árvore, já que originalmente não se foi considerado uma combinação das perdas das rodas de reação, o que pode reduzir um pouco a probabilidade de perda. Porém, para fins deste trabalho essa melhoria não foi considerada. Trabalhou-se com a probabilidade com apresentada no relatório da PMM (38). Note ainda que haja outro evento na árvore que traz a probabilidade de perda das rodas de reação como indicado na especificação da PMM (38).

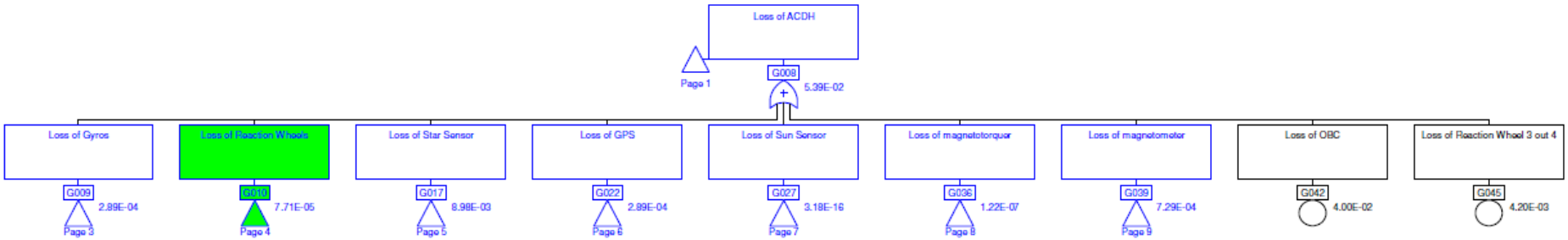

Figura A.1 – Árvore de Falhas completa da perda da PMM – Página 1 de 9.

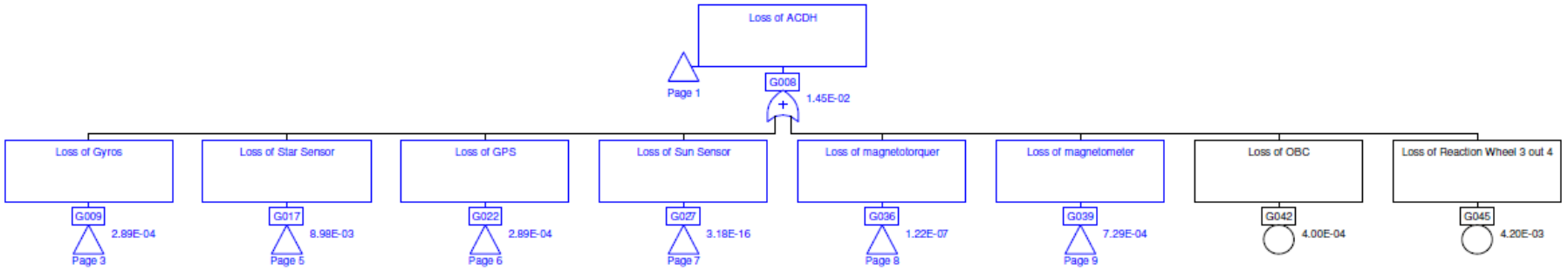

Figura A.2 – Árvore de Falhas completa da perda da PMM – Página 2 de 9.

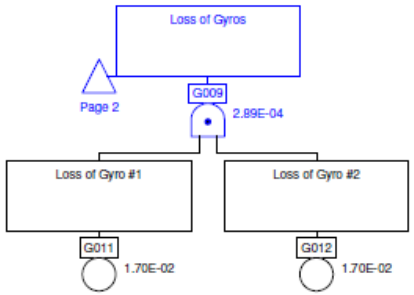

Figura A.3 – Árvore de Falhas completa da perda da PMM – Página 3 de 9.

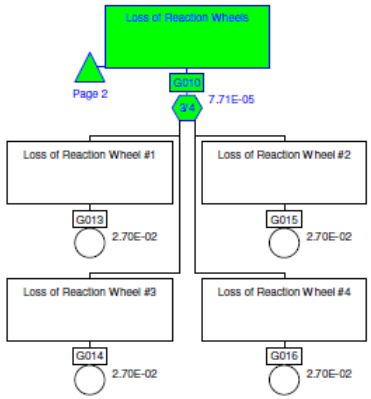

Figura A.4 – Árvore de Falhas completa da perda da PMM – Página 4 de 9.

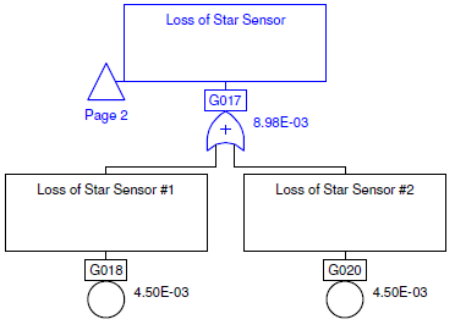

Figura A.5 – Árvore de Falhas completa da perda da PMM – Página 5 de 9.

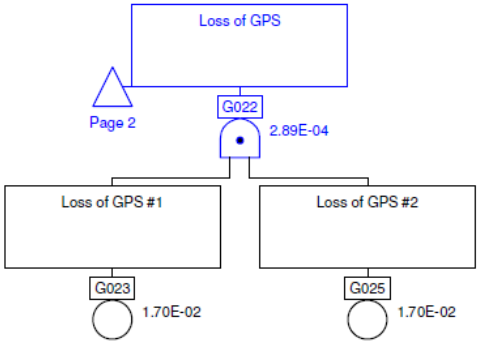

Figura A.6 – Árvore de Falhas completa da perda da PMM – Página 6 de 9.

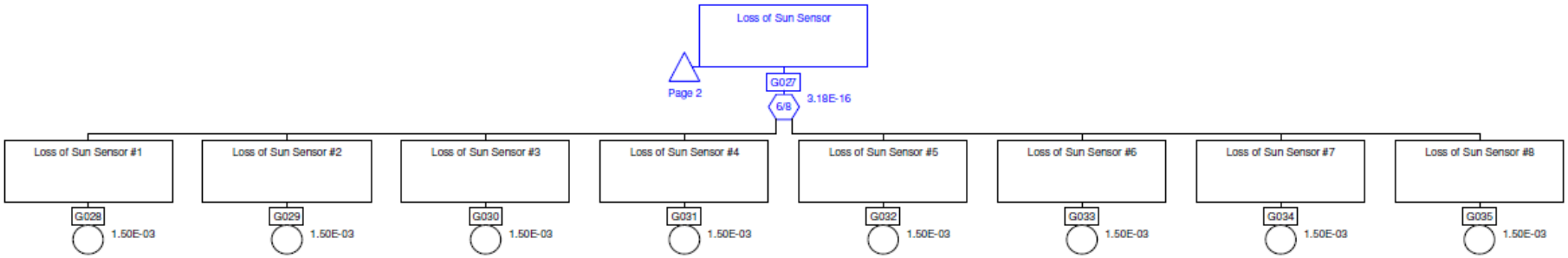

Figura A.7 – Árvore de Falhas completa da perda da PMM – Página 7 de 9.

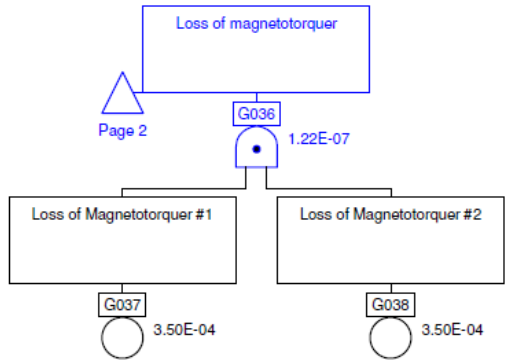

Figura A.8 – Árvore de Falhas completa da perda da PMM – Página 8 de 9.

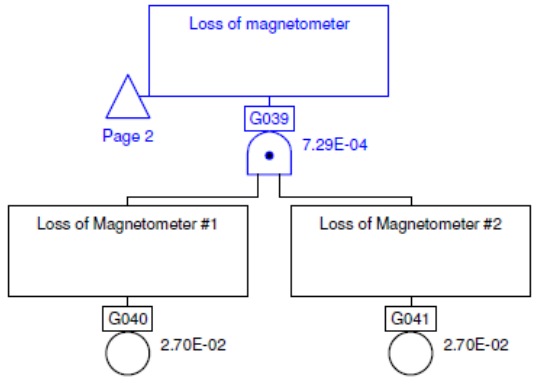

Figura A.9 – Árvore de Falhas completa da perda da PMM – Página 9 de 9.

A.2 Árvore de Falhas – solução duplo-simplex

A Árvore de Falhass para a solução duplo-simplex foi baseada na árvore básica para a PMM, apresentada na seção A.1. Esta árvore, porém, considera a arquitetura proposta para o OBC para cálculo da probabilidade de perda da PMM. Algumas considerações foram feitas para se atribuir probabilidades á perda das linhas COM e MON e do módulo de Segurança.

Com a ajuda da MIL-HDBK-217F (8) calculou-se a Confiabilidade de um micro-processador. O micro-processador foi escolhido por tratar-se do componente mais complexo da solução e, portanto, um dos componentes mais susceptíveis a falhas. Considerou-se que todos os outros componentes da solução seriam mais simples e com taxas de Confiabilidade superiores. Assim, a perda do micro-processador determina a Confiabilidade da solução. Segundo a MIL-HDBK-217F (8) a seguinte equação determina a probabilidade de falha de um micro-processador:

$$\lambda_p = \left(C_1\pi_T + C_2\pi_E\right)\pi_Q\pi_L \ \ 10^{-6} \text{ Falha/h} \qquad \textbf{(A.1)}$$

onde,

C1 = 0,56

C2 = 1,94 x 10-6

π_E = 0,5

π_Q = 2

π_L = 1

$$\pi_T = e^{\left(\frac{-E_a}{8{,}617x10^{-5}}\left(\frac{1}{T_J+273}x\frac{1}{298}\right)\right)} \qquad \textbf{(A.2)}$$

onde,

$$TJ = TC + P\theta_{JC} \qquad \textbf{(A.3)}$$

onde,

TC: é temperatura do envoltório do componente, e foi suposta estar a 150℃;

θ_{JC}: é a resistência térmica entre o envoltório e a junção, e foi suposto ser 0,1;

Ea = 0,4 ;

Com esses dados a taxa de falha do micro-processador foi de aproximadamente 12 x 10^{-6} falha/h, ou 1 x 10^{-5} falha/h. Multiplicando-se a taxa pela exposição de 35040, têm-se aproximadamente uma probabilidade de falha 3 x 10^{-1} falha/h.

Considerou-se ainda que a linha MON e o módulo de Segurança não precisam ser tão complexos quanto a linha COM, assim, suas probabilidades de falha devem ser maiores. Considerou-se para o módulo de Segurança e para a linha MON probabilidades de falha de 1,5 x 10^{-1}, duas vezes menor que a probabilidade da linha COM.

A árvore da solução duplo-simplex traz também a probabilidade de comando errôneo do OBC. Foi atribuída uma probabilidade de 1 x 10^{-6} falha/h comando errôneo a um micro-processador. A Árvore de Falhas da perda da PMM usando-se a solução duplo-simplex é apresentada a seguir (desde a Figura A.11 até a Figura A.20).

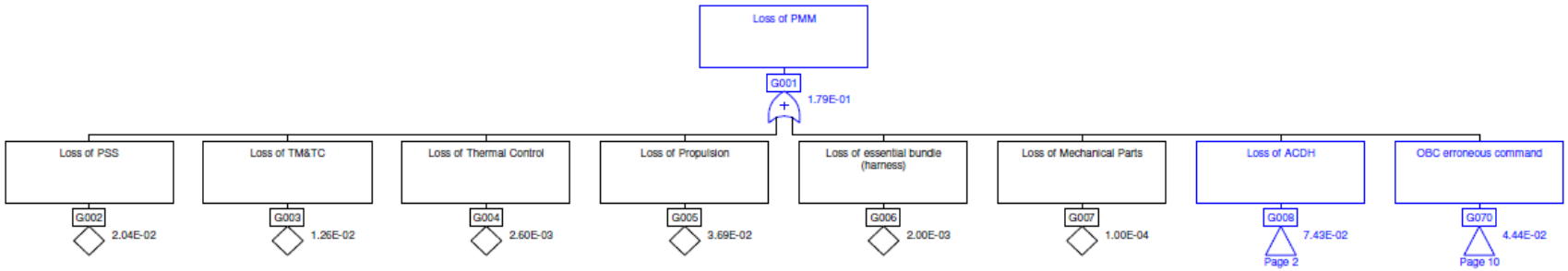

Figura A.11 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 1 de 10.

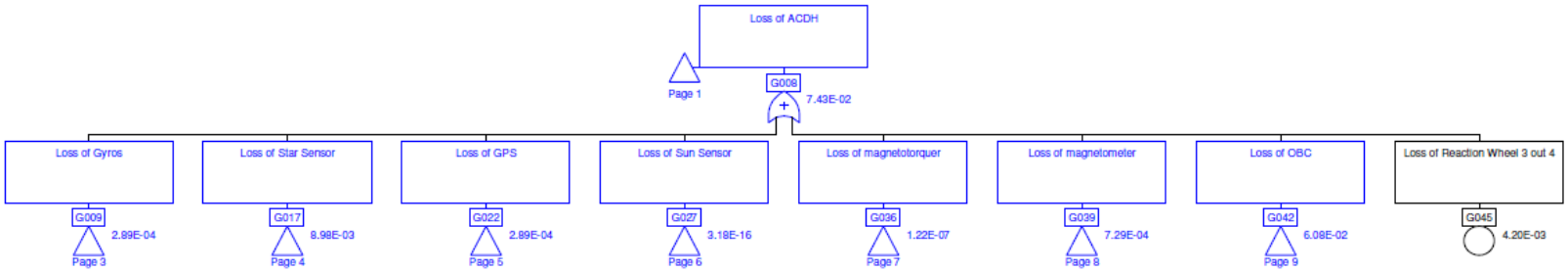

Figura A.12 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 2 de 10.

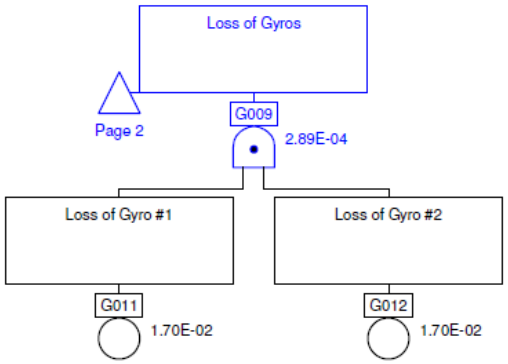

Figura A.13 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 3 de 10.

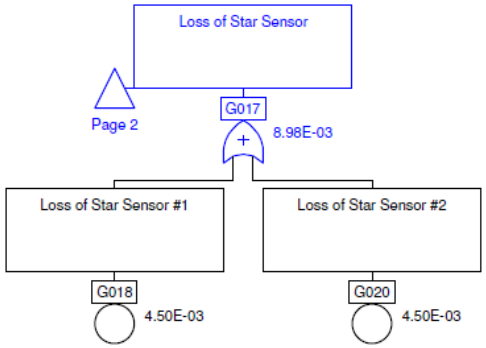

Figura A.14 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 4 de 10.

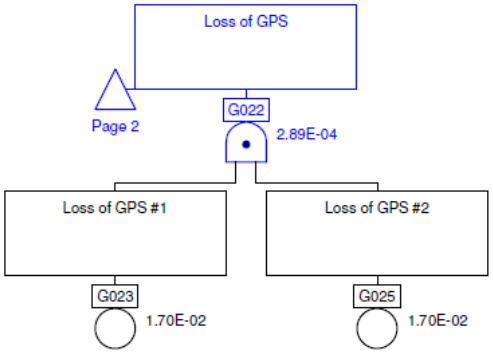

Figura A.15 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 5 de 10.

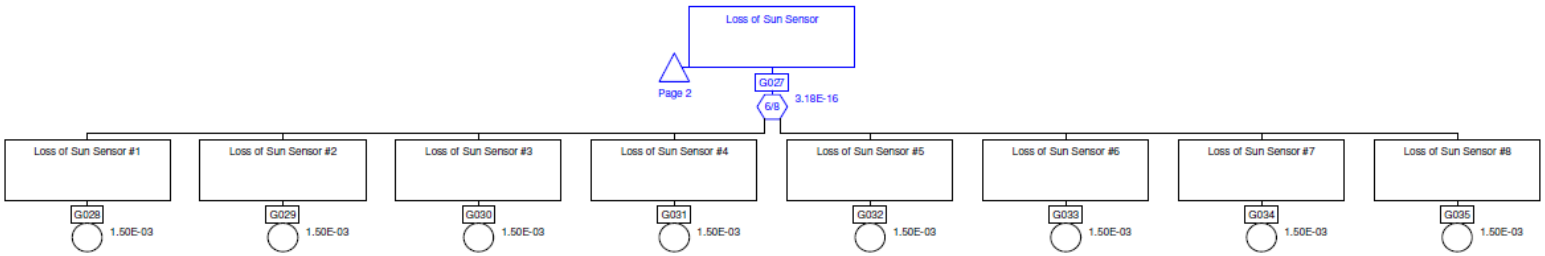

Figura A.16 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 6 de 10.

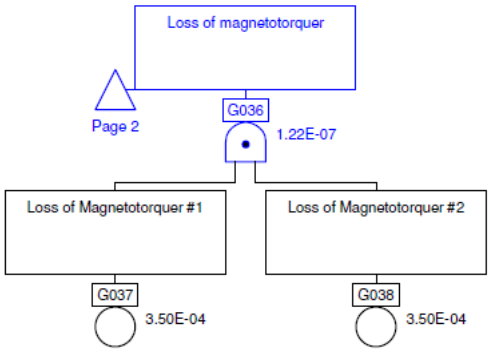

Figura A.17 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 7 de 10.

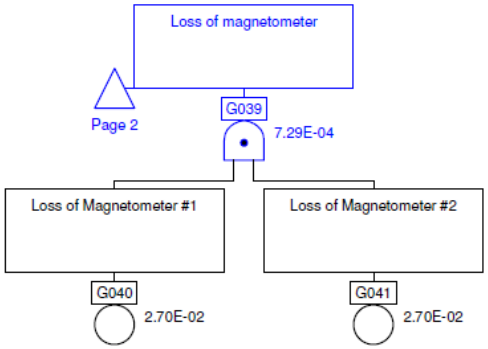

Figura A.18 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 8 de 10.

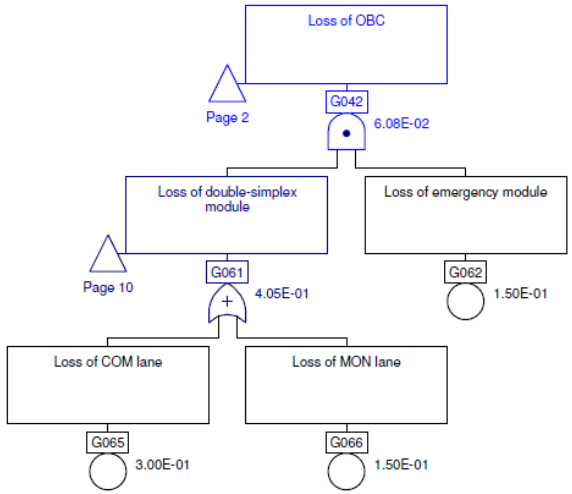

Figura A.19 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 9 de 10.

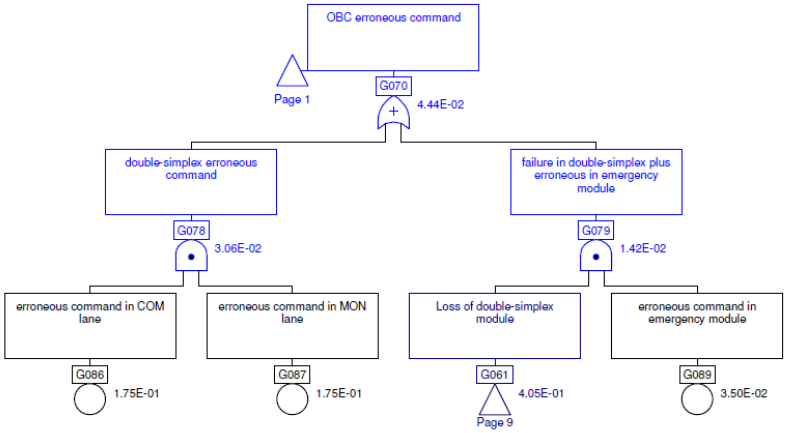

Figura A.2010 - Árvore de Falhas da perda da PMM considerando-se a solução duplo-simplex– Página 10 de 10.

A.3 Árvore de Falhas – solução triplo-simplex

A solução triplo-simplex usou as mesmas considerações da solução duplo-simplex já apresentadas na seção anterior. A Árvore de Falhas da perda da PMM usando-se a solução triplo-simplex é apresentada a seguir.

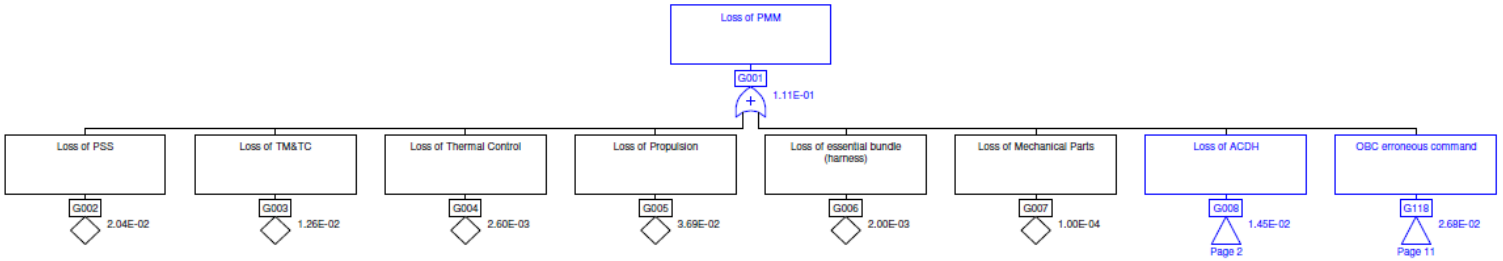

Figura A.11 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 1 de 14.

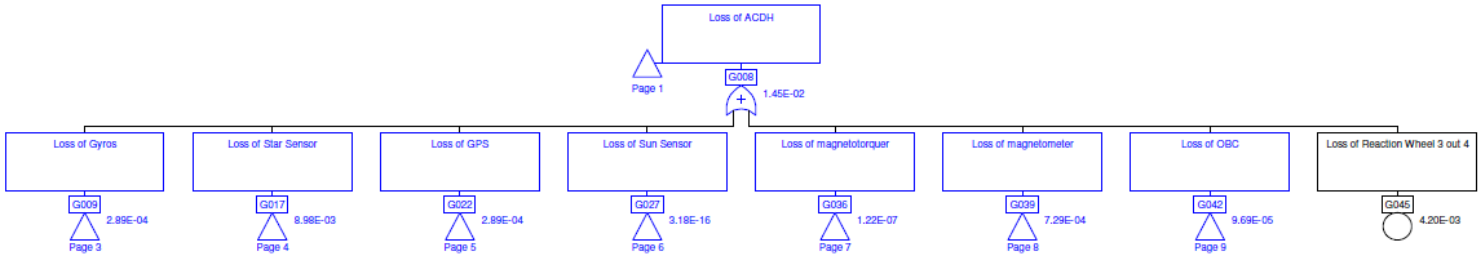

Figura A.12 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 2 de 14.

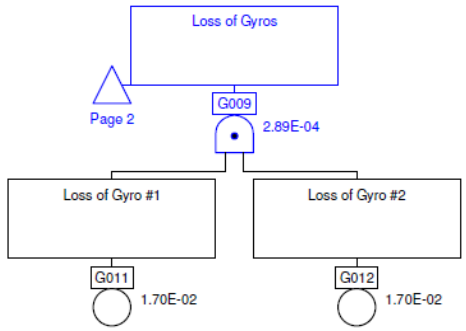

Figura A.13 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 3 de 14.

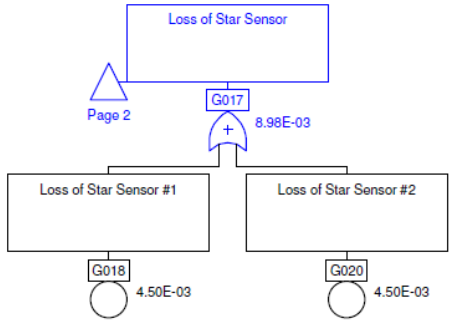

Figura A.14 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 4 de 14.

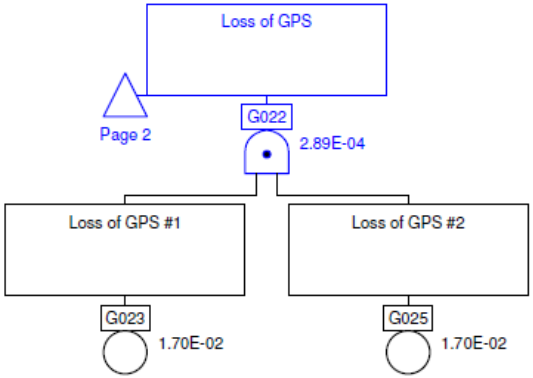

Figura A.15 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 5 de 14.

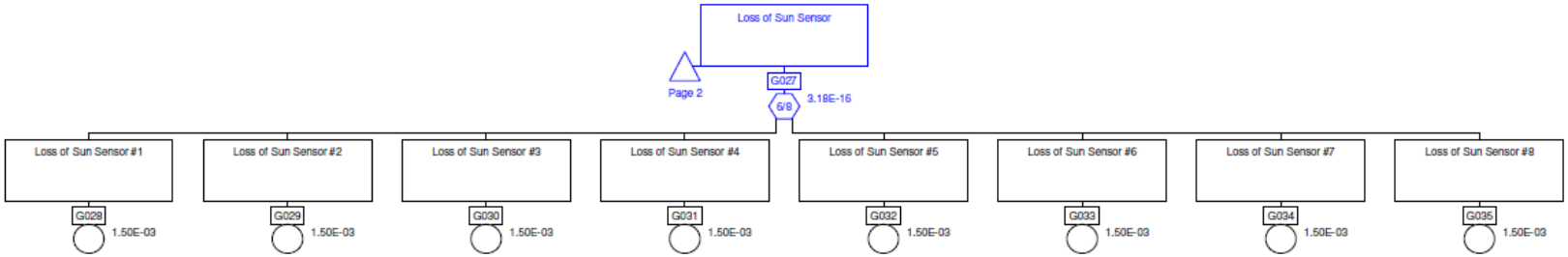

Figura A.16 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 6 de 14.

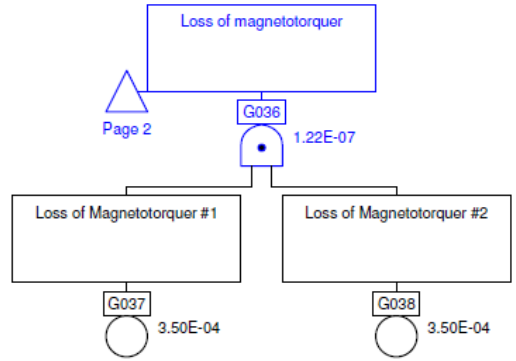

Figura A.17 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 7 de 14.

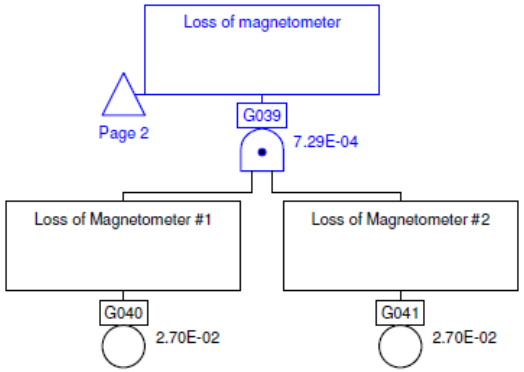

Figura A.18 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 8 de 14.

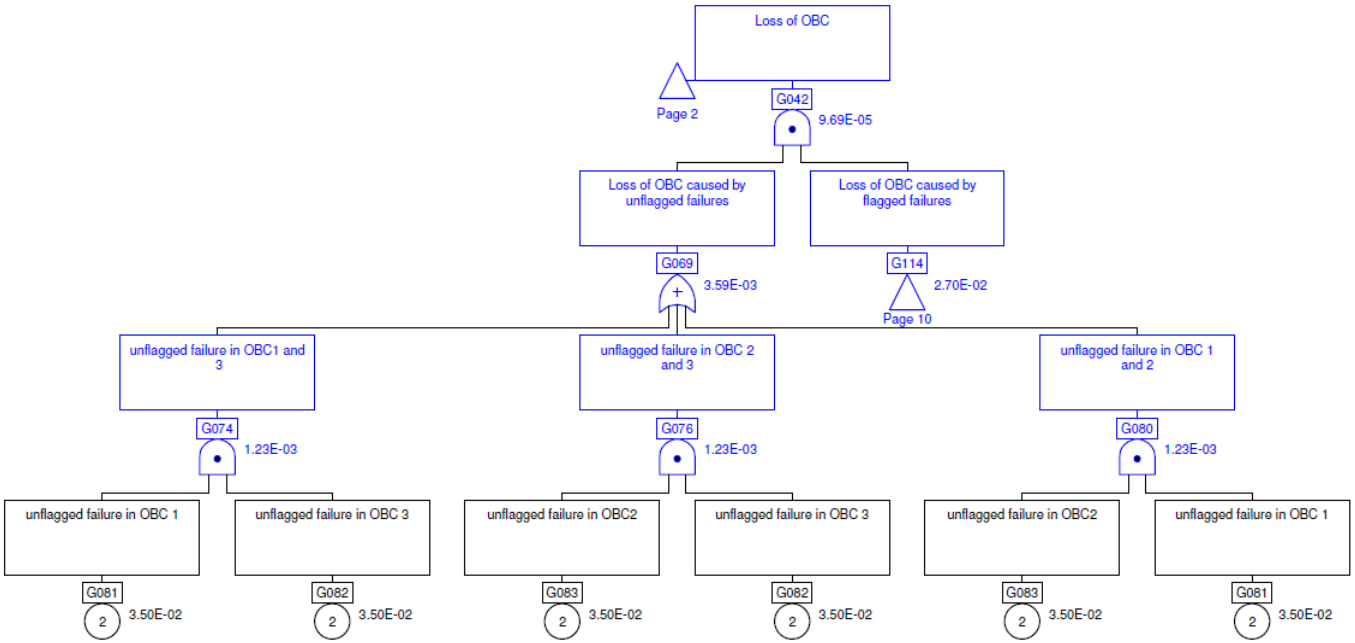

Figura A.19 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 9 de 14.

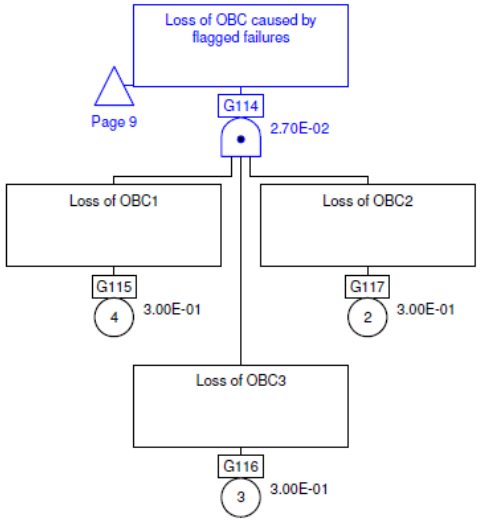

Figura A.20 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 10 de 14.

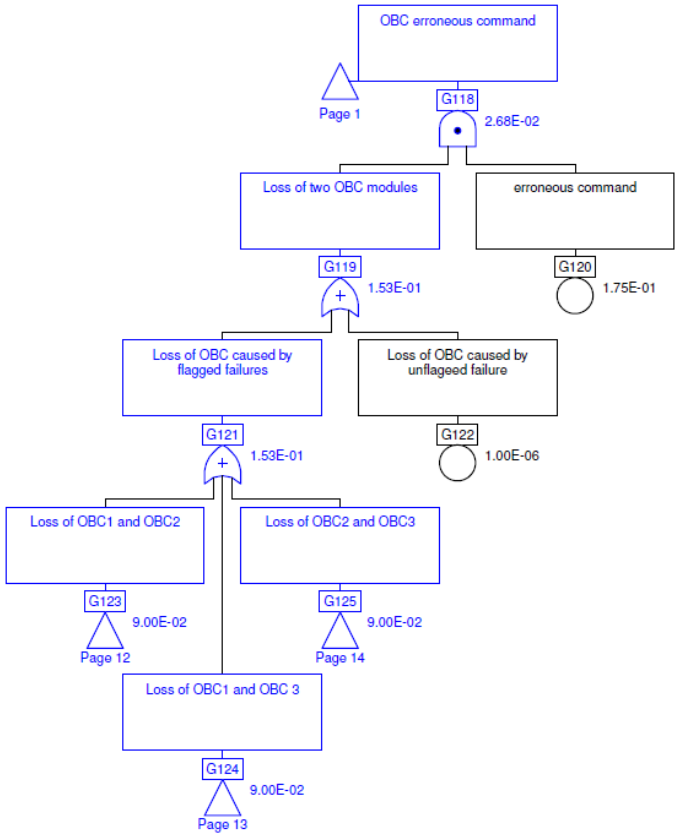

Figura A.21 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 11 de 14.

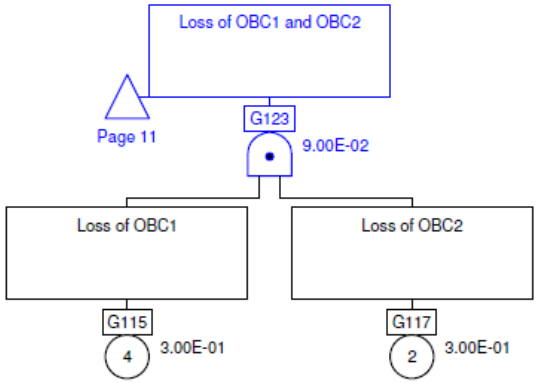

Figura A.22 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 12 de 14.

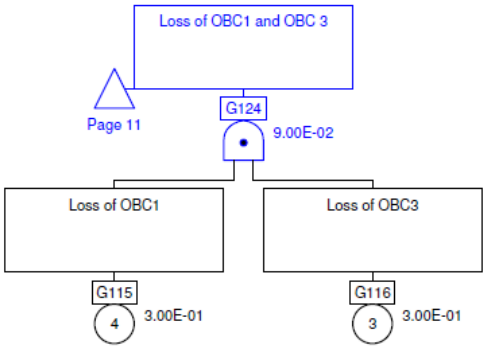

Figura A.23 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 13 de 14.

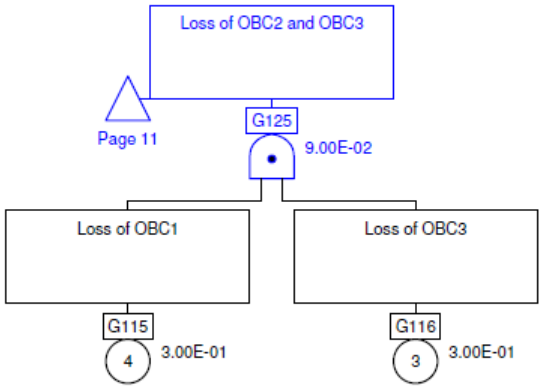

Figura A.24 - Árvore de Falhas da perda da PMM considerando-se a solução triplo-simplex– Página 14 de 14.