



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/08.22.10.40-TDI

**UM MÉTODO AUTOMÁTICO PARA DESENVOLVER
ARQUITETURAS FUNCIONAIS E FÍSICAS DE
SISTEMAS DE CONTROLE POR OTIMIZAÇÃO
MULTI-OBJETIVO BASEADA EM MODELOS,
ATRIBUTOS E MÉTRICAS SISTÊMICAS**

Francisco Carlos de Amorim Terceiro

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Contrôles, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 26 de agosto de 2013.

URL do documento original:

<http://urlib.net/8JMKD3MGP7W/3EMDHQ8>

INPE
São José dos Campos
2013

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

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/08.22.10.40-TDI

**UM MÉTODO AUTOMÁTICO PARA DESENVOLVER
ARQUITETURAS FUNCIONAIS E FÍSICAS DE
SISTEMAS DE CONTROLE POR OTIMIZAÇÃO
MULTI-OBJETIVO BASEADA EM MODELOS,
ATRIBUTOS E MÉTRICAS SISTÊMICAS**

Francisco Carlos de Amorim Terceiro

Tese de Doutorado 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 26 de agosto de 2013.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3EMDHQ8>>

INPE
São José dos Campos
2013

Dados Internacionais de Catalogação na Publicação (CIP)

Amorim Terceiro, Francisco Carlos.

Am68u Um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multi-objetivo baseada em modelos, atributos e métricas sistêmicas / Francisco Carlos de Amorim Terceiro. – São José dos Campos : INPE, 2013. xxx + 166 p. ; (sid.inpe.br/mtc-m19/2013/08.22.10.40-TDI)

Tese (Doutorado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Contrôles) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2013.

Orientador : Dr. Marcelo Lopes de Oliveira e Souza.

1. sistemas de controle. 2. controle. 3. arquitetura. 4. otimização. 5. multi-objetivo. 6. engenharia de sistemas. I.Título.

CDU 629.7:004.415



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

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

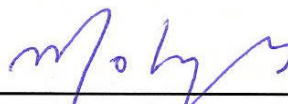
**Engenharia e Tecnologia Espaciais/Mecânica
Espacial e Controle**

Dr. Petrônio Noronha de Souza



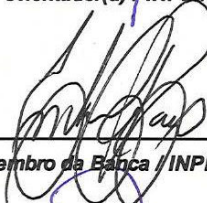
Presidente / INPE / SJCampos - SP

Dr. Marcelo Lopes de Oliveira e Souza



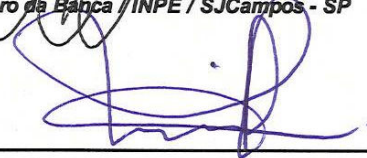
Orientador(a) / INPE / SJCampos - SP

Dr. Evandro Marconi Rocco



Membro da Banca / INPE / SJCampos - SP

Dr. Paulo Giacomio Milani



Membro da Banca / INPE / SJCampos - SP

Dr. Alexandre Carvalho Leite



Convidado(a) / IFF / Rio de Janeiro - RJ

Dr. Henrique Mohallen Paiva



Convidado(a) / MECTRON / São José dos Campos - SP

Este trabalho foi aprovado por:

() maioria simples

unanimidade

Aluno (a): **Francisco Carlos de Amorim Terceiro**

São José dos Campos, 26 de Agosto de 2013

*“Se não podes entender, crê para que entendas.
A fé precede, o intelecto segue.”*
—Santo Agostinho

*À minha esposa Patrícia Helena,
a meus pais Júnior e Luzia,
e à toda minha família...*

AGRADECIMENTOS

Agradeço imensamente à minha amada esposa Patrícia Helena, pelo seu amor, compreensão e ajuda durante toda esta longa jornada. O seu apoio e incentivo foram fundamentais para a conclusão deste trabalho; divido com ela todo o prazer desta conquista.

Agradeço ao Professor Dr. Marcelo Lopes por sua incansável disposição e iluminação em momentos difíceis. É um enorme prazer ser liderado pela sua pessoa, o qual considero um grande Mestre em aspectos profissionais e pessoais. Agradeço muito pela sua amizade e já o considero parte da minha família.

Agradeço aos meus pais, Junior e Luzia, pelo suporte que me deram para eu pudesse seguir o meu caminho. Agradeço aos meus irmãos Aryanna, Pedro Henrique e Paulo Davi e também aos meus sobrinhos Maria Letícia e João Pedro pela compreensão de tanto tempo de ausência e tantos quilômetros de distância de minha parte; o que estendo a toda minha família.

Agradeço enormemente a todos meus amigos que contribuíram de uma forma ou de outra para este trabalho. Em especial agradeço aos meus contemporâneos Alexandre Carvalho Leite, Paula Cristiane, Jairo Amaral e Jairo Siqueira.

Agradeço aos membros da banca: Petrônio Noronha de Souza, Evandro Marconi Rocco, Paulo Giácomo Milani, Alexandre Carvalho Leite e Henrique Mohallen Paiva, não somente pelos comentários e correções deste trabalho, mas também pelo vosso apoio em diversos outros momentos que contribuíram com a conclusão deste doutorado.

Agradeço ao INPE pela infraestrutura dos cursos de pós-graduação e ao meu País que pôde me oferecer a oportunidade de realizar um mestrado e doutorado na área de Engenharia e Tecnologia Espaciais.

Agradeço também a todos que, de alguma forma, contribuíram para minha formação profissional e pessoal.

Meus sinceros agradecimentos.

RESUMO

O conceito de arquiteturas para um sistema está diretamente conectado às suas funções e à sua materialização física. A arquitetura funcional define o que ele é capaz de fazer e a arquitetura física define como ele realiza sua missão. Atualmente, a elaboração de arquiteturas de sistemas é realizada por uma equipe multidisciplinar e esta tem que levar em consideração uma série de atributos de diversos domínios do conhecimento. Entretanto, essa elaboração pode ser considerada como uma atividade em que a Engenharia de Sistemas encontra a Arte. Em alguns domínios específicos, os métodos sobre o desenvolvimento de arquiteturas de sistemas já estão muito bem formalizados. Entretanto, quando analisamos a elaboração de arquiteturas de sistemas multidomínios, como os sistemas de controle, percebe-se que ainda há muito espaço para racionalização. Esta tese propõe um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multiobjetivo baseada em modelos, atributos e métricas sistêmicas. Para isso, a presente tese contém uma revisão da literatura sobre os métodos existentes de elaboração e seleção de arquiteturas de sistemas, modelos e métricas de atributos de sistemas de controle, e otimização multiobjetivo. Em seguida, a Tese apresenta a formulação do problema proposto, descreve três investigações (I1, I2 e I3) e as abordagens para sua solução. Na sequência, a tese descreve o ambiente e modelos desenvolvidos para elaborar diversas alternativas de arquiteturas tanto para sistemas estáticos como para sistemas dinâmicos. Resultados são apresentados com as principais arquiteturas de destaque, onde cada arquitetura é avaliada por meio das métricas escolhidas. Então, por meio de otimização multiobjetivo e pelo Critério da Menor Perda, uma arquitetura/solução é selecionada utilizando levando em consideração o equilíbrio entre todas as métricas da arquitetura. Esta se compara muito favoravelmente com os poucos resultados similares encontrados na literatura consultada. Além disto, e da inovação do método de geração das arquiteturas, a utilização do Critério de Menor Perda para chegar racionalmente a uma solução que une equilibradamente os requisitos tanto da Engenharia de Controle como da Engenharia de Sistemas são conquistas inovadoras desta Tese.

AN AUTOMATIC METHOD FOR DEVELOPING FUNCTIONAL AND PHYSICAL ARCHITECTURES OF CONTROL SYSTEMS BY MULTI-OBJECTIVE OPTIMIZATION BASED ON SYSTEMS MODELS, ATTRIBUTES AND METRICS

ABSTRACT

The concept of architecture for a system is directly connected to its functions and to its physical embodiment. The functional architecture defines what it can do and the physical architecture defines how it accomplishes its mission. Nowadays, a multidisciplinary team carries out the development of system architectures and they have to take into account a number of attributes and domains of knowledge. However, this development can be seen as an activity in which the Systems Engineering meets the Art. In some specific areas, the methods of the development of system architectures are already well formalized. However, when we analyze the development of architectures of multidomain systems, such as control systems, it is clear that there still is much room for rationalization. This thesis proposes an automatic method for developing functional and physical architectures of control systems by multi-objective optimization based on systems models, attributes and metrics. To do that, this thesis contains a literature review of existing methods of preparation and selection of system architectures, models and metrics of attributes of control systems, and multi-objective optimization. Then, the thesis presents the formulation of the proposed problem, describes three investigations (I1, I2 and I3) and approaches to their solutions. Further, the thesis describes the environment and models developed to prepare several alternative systems architectures for both static and dynamic systems. Results are presented with the main architectures, where each architecture is evaluated by means of the metrics chosen. Then, by means of multi-objective optimization and the Smallest Loss Criterion, an architecture / solution is selected by taking into account the balance between all metrics. Besides this, and the innovation of the method of generation of architectures, the use of the Smallest Loss Criterion to arrive at a solution that rationally balances the requirements of both Control Engineering and Systems Engineering are new achievements of this thesis.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 2-1: Exemplo de um ciclo de vida de uma missão espacial. Fonte: Souza (2008).	6
Figura 2-2: Processo de elaboração de uma arquitetura.	11
Figura 2-3: Evolução das diferentes arquiteturas presentes no desenvolvimento de um sistema.	13
Figura 2-4: Exemplo de um diagrama de blocos de um sistema de controle.	15
Figura 2-5: Exemplo de um <i>Reliability Block Diagram</i> (RBD). Fonte: Reliasoft (2010)	16
Figura 2-6: Exemplo de <i>System Breakdown Structure</i> de um satélite como a PMM.	17
Figura 2-7: Associação das Pontes de Königsberg com o respectivo grafo. Fonte: Cardoso (2011)	18
Figura 2-8: Ilustrações Originais do Artigo de Euler sobre as Pontes de Königsberg. Fonte: Euler (1741)	19
Figura 2-9: Ilustrações Originais dos Artigos de Arthur Cayley sobre Árvores. Fonte: Cayley (1857) e Cayley (1859).....	20
Figura 2-10: Resposta em frequência de sistemas de controle e suas margens de ganho e fase. Fonte: Takahashi, Rabins and Auslander (1970).	23
Figura 2-11: Resposta transitória típica da variável de saída de um sistema de 2ª. ordem subamortecido a uma entrada degrau. Fonte: Ogata (1993).	24
Figura 2-12: Exemplo de um diagrama que ilustra a sintaxe da OPN. Fonte: Simmons, Koo and Crawley (2005).	35
Figura 2-13: Modelo do sistema de controle proposto em (KOZA; KEANE; et al., 1999).	36
Figura 2-14: Estrutura do controlador obtida por (KOZA, KEANE, et al. 1999).	37

Figura 2-15: Comparação de resposta temporal a uma entrada degrau dos sistemas de controle (KOZA; KEANE et al., 1999) e (DORF; BISHOP, 1998).....	38
Figura 2-16: Diagrama de Blocos de uma Planta e seu Controlador PID. Fonte: Keane, Yu e Koza (2000).	38
Figura 2-17: Árvore que modela a Planta e o Controlador PID da Figura 2-16. Fonte: Keane, Yu e Koza (2000).	39
Figura 3-1: Sistema de controle da Investigação I1. Fonte: INPE (2001).	42
Figura 3-2: Sistema de controle da Investigação I2. Fonte: Koza, Keane et al. (1999).	44
Figura 3-3: Sistema de controle da Investigação I3. Fonte: Dorf e Bishop (2008).....	46
Figura 4-1: Árvore descrita pelo vetor $F = [0, 1, 2, 2, 4, 4, 4, 1, 8, 8, 10, 10]$	53
Figura 4-2: Matriz de Compatibilidade M, utilizada neste trabalho.....	55
Figura 4-3: Fluxograma simplificado do gerador de arquiteturas de sensores/atuadores.....	57
Figura 5-1: Fluxograma simplificado do gerador de arquiteturas de controladores.....	70
Figura 7.1: Espaço de Soluções com todas as 10.000 Soluções Geradas.	80
Figura 7.2: Resultados da Métrica de Custo.	81
Figura 7-3: Árvores: 1º. Menor Custo (acima) e 2º Menor Custo (abaixo).....	82
Figura 7-4: Árvores: 4º Menor Custo (acima) e 5º Menor Custo (abaixo).	83
Figura 7-5: Árvores: 1º. Maior Custo (acima) e 2º Maior Custo (abaixo).	84
Figura 7-6: Resultados da Métrica de Complexidade.....	85
Figura 7-7: Árvores: 1ª. Menor Complexidade (acima) e 2ª. Menor Complexidade (abaixo).....	86
Figura 7-8: Árvores: 4ª. Menor Complexidade (acima) e 5ª. Menor Complexidade (abaixo).....	87

Figura 7-9: Árvores: 1ª. Maior Complexidade (acima) e 2ª. Maior Complexidade (abaixo).....	88
Figura 7-10: Resultados da Métrica de Confiabilidade.	89
Figura 7-11: Árvores: 1ª. Menor Confiabilidade (acima) e 3ª. Maior Confiabilidade (abaixo).	90
Figura 7-12: Árvores: 1ª. Maior Confiabilidade (acima) e 2ª. Maior Confiabilidade (abaixo).	91
Figura 7-13: Árvores: 5ª. Maior Confiabilidade (acima) e 2ª. Maior Confiabilidade (abaixo).	92
Figura 7-14: Detalhe do Espaço de Soluções e a Fronteira de Pareto.	93
Figura 7-15: Soluções Não Dominadas: Árvore 782 (acima) e Árvore 5404 (abaixo).	94
Figura 7-16: Soluções Não Dominadas: Árvore 3739 (acima) e Árvore 3458 (abaixo).	95
Figura 7-17: Detalhe da Solução mais Próxima do Centro de Massa (Baricentro).	96
Figura 7-18: Árvores: Na Fronteira de Pareto 1ª. Mais Próxima do Baricentro (acima) e 2ª. Mais Próxima (abaixo).....	97
Figura 7-19: Árvores: Na Fronteira de Pareto 1ª. Mais Distante do Baricentro (acima) e 2ª. Mais Distante (abaixo).....	98
Figura 8-1: Estrutura do Sistema de Controle da Investigação I2.	99
Figura 8-2: Espaço de Soluções com todas as 70 soluções aceitáveis.....	101
Figura 8-3: Resultados da Métrica de Tempo de Subida.	102
Figura 8-4: Controlador 67 – 1º. Menor Tempo de Subida.	103
Figura 8-5: Controlador 94 - 2º Menor Tempo de Subida.....	103
Figura 8-6: Controlador 4 – 1º. Maior Tempo de Subida.	104
Figura 8-7: Controlador 20 - 2º Maior Tempo de Subida.	104
Figura 8-8: Resultados da Métrica de Custo.....	105

Figura 8-9: Controlador 4 – 1º. Menor Custo.	106
Figura 8-10: Controlador 70 - 2º Menor Custo.	106
Figura 8-11: Controlador 11 - 3º Menor Custo.	107
Figura 8-12: Controlador 41 – 1º. Maior Custo.	107
Figura 8-13: Controlador 95 - 2º Maior Custo.	108
Figura 8-14: Resultados da Métrica de Complexidade.	109
Figura 8-15: Controlador 4 – 1ª. Menor Complexidade.	110
Figura 8-16: Controlador 60 – 2ª. Menor Complexidade.	110
Figura 8-17: Controlador 2 – 4ª. Menor Complexidade.	111
Figura 8-18: Controlador 75 – 1ª. Maior Complexidade.	111
Figura 8-19: Controlador 65 – 2ª. Maior Complexidade.	112
Figura 8-20: Espaço de Soluções e a Fronteira de Pareto.	113
Figura 8-21: Controlador 12 - Solução Não Dominada.	114
Figura 8-22: Controlador 37 - Solução Não Dominada.	114
Figura 8-23: Controlador 66 - Solução Não Dominada.	115
Figura 8-24: Controlador 72 - Solução Não Dominada.	115
Figura 8-25: Espaço de Soluções e Critério de Menor Perda.	116
Figura 8-26: Solução de Menor Perda.	117
Figura 8-27: Resposta a uma entrada a Degrau Unitário da Solução de Menor Perda.	118
Figura 8-28: Diagramas de Bode da Solução de Menor Perda.	118
Figura 8-29: Resposta ao Degrau Unitário da Solução de Menor Tempo de Subida.	121
Figura 8-30: Diagramas de Bode da Solução de Menor Tempo de Subida.	121
Figura 8-31: Comparação da Arquitetura de Menor Tempo de Subida com a Literatura.	122
Figura 9-1: Espaço de Soluções com todas as 42 Soluções Aceitáveis.	124

Figura 9-2: Resultados da Métrica de Tempo de Acomodação.....	125
Figura 9-3: Controlador 89 – 1º. Menor Tempo de Acomodação.....	126
Figura 9-4: Controlador 45 - 2º Menor Tempo de Acomodação.....	126
Figura 9-5: Controlador 57 – 1º. Maior Tempo de Acomodação.....	127
Figura 9-6: Controlador 19 - 2º Maior Tempo de Acomodação.....	127
Figura 9-7: Resultados da Métrica de Custo.....	128
Figura 9-8: Controlador 8 – 1º. Menor Custo.....	129
Figura 9-9: Controlador 16 - 2º Menor Custo.....	129
Figura 9-10: Controlador 46 – 1º. Maior Custo.....	130
Figura 9-11: Controlador 57 - 2º Maior Custo.....	130
Figura 9-12: Resultados da Métrica de Complexidade.....	131
Figura 9-13: Controlador 88 – 1ª. Menor Complexidade.....	132
Figura 9-14: Controlador 93 – 2ª. Menor Complexidade.....	132
Figura 9-15: Controlador 44 – 3ª. Maior Complexidade.....	133
Figura 9-16: Controlador 65 – 4ª. Maior Complexidade.....	133
Figura 9-17: Espaço de Soluções e a Fronteira de Pareto.....	134
Figura 9-18: Controlador 7 - Solução Não Dominada.....	135
Figura 9-19: Controlador 31 - Solução Não Dominada.....	135
Figura 9-20: Controlador 66 - Solução Não Dominada.....	136
Figura 9-21: Controlador 72 - Solução Não Dominada.....	136
Figura 9-22: Espaço de Soluções e Critério de Menor Perda.....	137
Figura 9-23: Solução de Menor Perda.....	138
Figura 9-24: Resposta a uma Entrada a Degrau Unitário da Solução de Menor Perda.....	139
Figura 9-25: Diagramas de Bode da Solução de Menor Perda.....	139
Figura 9-26: Resposta ao Degrau Unitário da Solução de Menor Tempo de Acomodação (Controlador 89).....	141

Figura 9-27: Diagramas de Bode da Solução de Menor Tempo de Acomodação (Controlador 89).	141
Figura 9-28: Comparação de entre as Arquiteturas de Menor Tempo de Acomodação (Controlador 89) e Menor Perda (Controlador 10), com a Literatura.....	142
Figura 10-1: Fluxograma do Método AGORA.....	144

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 4-1: Blocos utilizados nas arquiteturas de sensores/atuadores.....	52
Tabela 4-2: Valores de custos utilizados para cada categoria de bloco.....	61
Tabela 4-3: Valores de custos utilizados para cada nível de confiabilidade.....	61
Tabela 4-4: Níveis de confiabilidade e respectivos valores utilizado.....	61
Tabela 5-1: Blocos utilizados nas arquiteturas de controladores.....	67
Tabela 5-2: Níveis de confiabilidade e respectivos valores utilizados.....	73
Tabela 5-3: Valores de utilizados para cálculo do custo(P).....	73
Tabela 5-4: Valores de complexidade individual de cada bloco.....	74
Tabela 8-1: Principais diferenças entre o trabalho (KOZA; KEANE et al., 1999) e este.....	119
Tabela 8-2: Comparativo Entre as Funções de Transferência dos Pré-Filtros Utilizados.....	120
Tabela 8-3: Comparativo Entre as Funções de Transferência dos Controladores.....	120
Tabela 8-4: Comparativo entre os Tempos de Subida.....	122
Tabela 9-1: Comparativo Entre as Funções de Transferência dos Controladores.....	140

LISTA DE ABREVIATURAS E SIGLAS

ACDH	-	<i>Attitude Control and Data Handling</i>
AGORA	-	Ambiente Automático de Geração Otimizada, Orientada e Randômica de Arquiteturas.
AOC-AS	-	<i>Attitude and Orbit Control Application Software</i>
CDR	-	<i>Critical Design Review.</i>
COTS	-	<i>Commercial Off-The-Shelf</i>
LQR	-	<i>Linear Quadratic Regulator</i>
MOE	-	<i>Measures of Effectiveness.</i>
OBC	-	<i>On Board Computer</i>
OBDH-AS	-	<i>On Board Data Handling Application Software</i>
PDR	-	<i>Preliminary Design Review.</i>
PID	-	Proporcional, Integral e Derivativo
PMM	-	Plataforma Multi-Missão
TT&C	-	<i>Telemetry, Tracking & Command</i>
VVA	-	Verificação, Validação, Aceitação/Certificação

SUMÁRIO

1 INTRODUÇÃO	1
1.1 Contexto	1
1.2 Motivações	2
1.3 Objetivo	3
1.4 Originalidade, Generalidade e Utilidade	3
1.5 Organização deste Trabalho	4
2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA.....	5
2.1 Ciclo de Vida de um Sistema	5
2.1.1 Etapa de Concepção/Requisitos/Especificações	6
2.1.2 Etapa de Projeto/Elaboração/Otimização	6
2.1.3 Etapa de Construção/Obtenção de Partes/Integração/Testes (VVA).....	7
2.1.4 Etapa de Operação/Manutenção/Descarte	7
2.2 Uma Taxonomia de Arquiteturas	8
2.2.1 Definição de Arquiteturas	8
2.2.2 Onde Estão Presentes	9
2.2.3 Categorias de Arquiteturas.....	9
2.3 Elaboração e Evolução das Arquiteturas.....	10
2.3.1 Elaboração de Uma Arquitetura.....	11
2.3.2 Evolução ao Longo do Ciclo de Vida.....	12
2.4 Linguagens de Modelagem.....	14
2.4.1 Diagrama de Funções.....	14
2.4.2 Diagramas de Componentes	15
2.4.3 Grafos e Árvores	17
2.5 Atributos e Medidas	21
2.5.1 Atributos da Engenharia de Controle.....	21
2.5.2 Atributos da Engenharia de Sistemas	26

2.5.3 Métricas.....	30
2.6 Abordagens Multiobjetivo.....	31
2.6.1 Otimização Multiobjetivo.....	31
2.6.2 Object Process Network (OPN).....	34
2.7 Geração Automática de Controladores.....	35
3 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO.....	41
3.1 Primeira Investigação (I1).....	42
3.2 Segunda Investigação (I2).....	43
3.3 Terceira Investigação (I3).....	45
4 PRIMEIRA INVESTIGAÇÃO (I1): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE SENSORES E ATUADORES.....	49
4.1 Elementos de uma Arquitetura de Sensores e Atuadores.....	49
4.1.1 Premissas.....	49
4.1.2 Definição dos Blocos.....	52
4.2 Modelo de uma árvore.....	53
4.3 Matriz de Compatibilidade.....	54
4.4 Método de Geração de Arquiteturas.....	55
4.4.1 Condições Iniciais.....	55
4.4.2 Evolução da Árvore.....	56
4.4.3 Operação de poda de árvore.....	59
4.4.4 Critério de parada.....	59
4.5 Método de Avaliação e Seleção de Arquiteturas.....	60
4.5.1 Métricas.....	60
4.5.2 Análise Mono-Objetivo.....	63
4.5.3 Análise Multiobjetivo.....	63
5 SEGUNDA INVESTIGAÇÃO (I2): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE CONTROLADORES.....	65

5.1	Elementos de uma Arquitetura de Controladores.....	65
5.1.1	Premissas.....	65
5.1.2	Definição dos Blocos	67
5.2	Modelo de um Controlador	68
5.3	Matriz de Compatibilidade	68
5.4	Método de Geração de Arquiteturas.....	68
5.4.1	Condições Iniciais.....	69
5.4.2	Evolução da Arquitetura	69
5.4.3	Critério de Parada	71
5.5	Ajustes dos Parâmetros do Controlador	71
5.5.1	Restrições Consideradas	71
5.6	Método de Avaliação e Seleção de Arquiteturas.....	71
5.6.1	Métricas.....	72
5.6.2	Análise Mono-Objetivo	74
5.6.3	Análise Multiobjetivo	75
6	TERCEIRA INVESTIGAÇÃO (I3): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE CONTROLADORES.....	77
7	RESULTADOS DA PRIMEIRA INVESTIGAÇÃO (I1).....	79
7.1	Descrição	79
7.2	Resultados Gerais	79
7.3	Métrica de Custo.....	81
7.3.1	Menor Custo.....	82
7.3.2	Maior Custo	84
7.4	Métrica de Complexidade.....	85
7.4.1	Menor Complexidade.....	86
7.4.2	Maior Complexidade	88
7.5	Métrica de Confiabilidade	89

7.5.1 Menor Confiabilidade	90
7.5.2 Maior Confiabilidade	91
7.6 Fronteira de Pareto	93
7.7 Seleção pelo Critério da Menor Perda.....	96
7.7.1 Entre as Soluções na Fronteira de Pareto, as Mais Próximas do Baricentro	97
7.7.2 Entre as Soluções na Fronteira de Pareto, as Mais Distantes do Baricentro.....	98
8 RESULTADOS DA SEGUNDA INVESTIGAÇÃO (I2).....	99
8.1 Descrição	99
8.2 Resultados Gerais	100
8.3 Métrica Tempo de Subida	101
8.3.1 Menor Tempo de Subida.....	102
8.3.2 Maior Tempo de Subida.....	103
8.4 Métrica Custo	105
8.4.1 Menor Custo.....	105
8.4.2 Maior Custo	107
8.5 Métrica Complexidade	108
8.5.1 Menor Complexidade.....	109
8.5.2 Maior Complexidade	111
8.6 Fronteira de Pareto	112
8.7 Seleção pelo Critério da Menor Perda.....	116
8.8 Comparação com Resultados da Literatura.....	119
9 RESULTADOS DA TERCEIRA INVESTIGAÇÃO (I3).....	123
9.1 Descrição	123
9.2 Resultados Gerais	123
9.3 Métrica Tempo de Acomodação	124
9.3.1 Menor Tempo de Acomodação.....	125
9.3.2 Maior Tempo de Acomodação.....	126
9.4 Métrica Custo	128

9.4.1 Menor Custo.....	128
9.4.2 Maior Custo	129
9.5 Métrica Complexidade	130
9.5.1 Menor Complexidade.....	131
9.5.2 Maior Complexidade	133
9.6 Fronteira de Pareto	134
9.7 Seleção pelo Critério da Menor Perda.....	137
9.8 Comparação com Resultados da Literatura.....	140
10 UM MÉTODO AUTOMÁTICO PARA DESENVOLVER ARQUITETURAS FUNCIONAIS E FÍSICAS DE SISTEMAS DE CONTROLE.....	143
10.1 Descrição do Método AGORA	143
10.1.1 Fase de Formação do Ambiente	145
10.1.2 Fase de Geração de Arquiteturas	148
10.1.3 Fase de Ajuste de Parâmetros	148
10.1.4 Fase de Seleção de uma Arquitetura.....	149
10.2 Denominação do Método AGORA	149
11 CONCLUSÕES	151
11.1 Aspectos Gerais	151
11.2 Arquiteturas para Sistemas Estáticos - Primeira Investigação (I1)	152
11.2.1 Sugestões para Trabalhos Futuros	154
11.3 Arquiteturas para Sistemas Dinâmicos - Segunda Investigação (I2) e Terceira Investigação (I3).	155
11.3.1 Sugestões para Trabalhos Futuros	157
11.4 Conclusões finais.....	158
12 REFERÊNCIAS BIBLIOGRÁFICAS	159

1 INTRODUÇÃO

A crescente complexidade dos problemas enfrentados pelas Engenharias é o principal motivo que impulsiona os métodos de desenvolvimento de **sistemas de engenharia** (alvo/fim) a evoluir, fazendo com que se busquem melhores maneiras para solucionar os problemas atuais e também aqueles que estão por vir. Os **sistemas de controle** (instrumento/meio) seguem essa mesma tendência de aumento de complexidade, sendo também influenciados pelo aumento da integração dos componentes que viabiliza o aumento da capacidade de processamento dos computadores digitais, dos fluxos das linhas de comunicação e a alta flexibilidade oferecida pelo software.

1.1 Contexto

O desenvolvimento de um sistema pode ser associado ao ato de solucionar um problema. Sendo assim, a materialização de um sistema específico o torna uma solução específica particular para o dado problema. Entretanto, para um mesmo problema podem existir inúmeras soluções possíveis. Cada **sistema-solução**, como qualquer outro sistema, é composto por vários **componentes** que trabalham em conjunto para exercer as funções desejadas. Os **componentes, seus atributos e relacionamentos**, bem como suas qualidades, quantidades e organização determinam a **arquitetura do sistema**. A arquitetura engloba todas as funções/propriedades/comportamentos (**arquitetura funcional**) e todas as estruturas/interfaces/conexões (**arquitetura física**) que compõem o sistema. Estas admitem medidas por diferentes **métricas** (“Measures of Effectiveness-MOEs”). Alterações nos componentes, seus atributos e relacionamentos bem como diferentes qualidades, quantidades e organização irão resultar em diferentes arquiteturas funcionais e arquiteturas físicas do sistema.

A atividade de definição da arquitetura de um sistema é comumente feita de maneira subjetiva, baseada no conhecimento prévio da equipe de profissionais que conduzem essa atividade. É desejável o uso de técnicas objetivas que auxiliem na elaboração de arquiteturas de sistemas e na seleção de arquiteturas propostas. A procura por arquiteturas de sistemas mais eficazes deve ser realizada ainda nas fases iniciais do

desenvolvimento do sistema de controle. Dessa maneira reduzem-se o custo e o tempo de mudanças. Por esse motivo, a **Engenharia de Sistemas Baseada em Modelos**, é útil e permite que as análises e as avaliações sejam realizadas antes da construção do sistema. Estas usam medidas feitas por diferentes **métricas** (MOEs).

Um diferencial deste trabalho é o foco nas arquiteturas dos sistemas de controle projetadas pela **Engenharia de Controle**. Em geral esse assunto é tratado como uma consequência de decisões tomadas durante a execução de um projeto. Diversas decisões que afetam a arquitetura do sistema são tomadas visando a resolução de problemas locais de domínio específico. Algumas decisões são explicitamente sobre a arquitetura do sistema de controle, outras não são tão aparentes, mas também definem a arquitetura. Neste trabalho pretende-se posicionar a elaboração das arquiteturas no foco da pesquisa.

A Engenharia de Sistemas e a Engenharia de Controle estão conectadas e devem trabalhar em conjunto, apesar desta relação, elas parecem estar se distanciando, sugerindo muitas vezes haver uma dificuldade de diálogo entre as duas. Este trabalho também possui o objetivo de construir pontes entre essas disciplinas para tornar mais fácil a necessária relação entre as duas em um projeto. Da mesma maneira que também busca construir pontes entre as etapas de projeto de um sistema e de sua implementação.

1.2 Motivações

O desenvolvimento de satélites artificiais é uma atividade que necessita ser realizada como extremo rigor e objetividade, devido à imensa complexidade presente neste tipo de desenvolvimento. Decisões arquiteturais, aparentemente simples, nas etapas iniciais de concepção terão grande impacto por todo o ciclo de vida do veículo. Essas decisões irão determinar o quão complexo será o projeto do sistema, a dificuldade de fabricação e até seu custo de operação. O mesmo se aplica aos sistemas de controle desses veículos. É comum que as arquiteturas dos sistemas sejam determinadas como consequências das decisões de projeto de domínios específicos e não como uma preocupação prioritária. Também é comum que essas decisões específicas sobre a arquitetura sejam tomadas com subjetividade.

Todo o esforço dispendido na busca por uma arquitetura eficaz é muito bem recompensado. As decisões corretas podem promover enormes benefícios que vão desde a redução do tempo de desenvolvimento, o aumento da eficiência na operação e até a redução do custo do projeto. Apesar de já existirem técnicas e metodologias para elaboração de arquiteturas de sistemas, ainda há um enorme espaço para novas contribuições neste ramo de pesquisa. Contribuições feitas nesse ramo de pesquisa possuem grande potencial para serem extrapoladas para outros domínios de conhecimento. Todos esses fatos motivaram a escolha do objeto de estudo deste trabalho.

1.3 Objetivo

Este trabalho propõe um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multiobjetivo baseada em modelos, atributos e métricas sistêmicas.

1.4 Originalidade, Generalidade e Utilidade

Um estudo de doutorado deve possuir três características fundamentais, sejam elas: originalidade, generalidade e utilidade. Essas características são postas como metas na busca do objetivo deste trabalho e guiaram as escolhas para definição das contribuições no aprimoramento das técnicas de elaboração de arquiteturas de sistemas de controle.

A originalidade vem da proposta de um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multiobjetivo baseada em modelos, atributos e métricas sistêmicas. Desde o presente momento, observa-se o grande potencial para novas contribuições neste ramo de pesquisa, pois é uma área de pesquisa jovem que apresenta várias oportunidades.

A generalidade é inerente ao estudo de sistemas de controle e também de arquiteturas e já está presente na motivação deste trabalho e nos seus objetivos, assim como o uso de técnicas de otimização multiobjetivo, que reforça ainda mais essa característica.

É desejável que os resultados desta pesquisa sejam de utilidade direta para equipes de projeto envolvidas no desenvolvimento de sistemas de controle. Espera-se que as contribuições deste trabalho possam ajudar a lidar com a crescente complexidade dos sistemas que desejamos construir.

1.5 Organização deste Trabalho

Após essa introdução descrita no presente capítulo, o trabalho se desenvolve com a estrutura descrita a seguir. O Capítulo 2 apresenta os conceitos básicos e uma revisão da literatura sobre o assunto estudado neste trabalho. O Capítulo 3 apresenta a formulação do problema em estudo e as abordagens para sua solução, e descreve três investigações que são realizadas neste trabalho. O Capítulo 4 apresenta uma descrição detalhada da elaboração e seleção de arquiteturas de sensores/atuadores. Os Capítulos 5 e 6 apresentam uma descrição detalhada da elaboração e seleção de arquiteturas de controladores. Os Capítulos 7 a 9 apresentam os resultados das três investigações que são realizadas neste trabalho. O Capítulo 10 descreve o método geral que engloba as elaborações e seleções de arquiteturas dos capítulos anteriores. O Capítulo 11 faz uma análise dos resultados alcançados neste trabalho.

2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA

Este capítulo estabelece os conceitos básicos e a revisão da literatura necessários para estabelecer uma base de conhecimento comum para o entendimento deste trabalho. Também fornece uma visão panorâmica sobre o estudo de arquiteturas de sistemas, para que, dessa forma, possa-se localizar aonde iremos focamos a pesquisa dentre todo o universo que abrange esse estudo. O conteúdo aqui presente serve como referência para todo o desenvolvimento da pesquisa.

2.1 Ciclo de Vida de um Sistema

O termo “**ciclo de vida de um sistema**” remete a uma analogia com o ciclo de vida de um ser vivo. Talvez devido a esse fato, é que vários autores o adotem. É comum encontrar a expressão “ciclo de vida de um sistema” sendo utilizada em vários contextos, embora nem sempre se explicita qual o seu escopo. Pode-se ter, por exemplo, ciclo de vida: de produto, de desenvolvimento, de projeto, da missão, etc. Este trabalho usa o conceito mais amplo, que se inicia na concepção do sistema, passa pelo projeto, construção, operação e finaliza com o descarte.

Em especial, o ciclo de vida de uma missão espacial pode ser dividido em diversas fases, cada fase com um ou mais objetivos específicos. Essa divisão das fases e a terminologia usada podem variar de acordo com a instituição responsável pela execução do programa de desenvolvimento. A Figura 2-1 ilustra um exemplo de fases do ciclo de vida de uma missão espacial. Elas se aplicam a cada sistema integrante.

Essa descrição do ciclo de vida é essencial para entendermos as diferentes etapas que um sistema atravessa durante sua vida. Cada etapa contribui diferentemente para a evolução do sistema e da sua arquitetura. As arquiteturas de um sistema também evoluem ao longo do tempo juntamente à evolução do próprio sistema. Neste trabalho, o ciclo de vida é representado em quatro grandes etapas macroscópicas: concepção/requisitos/especificações, projeto/elaboração/otimização, construção/obtenção de partes/integração/testes(VVA) e operação/manutenção /descarte.

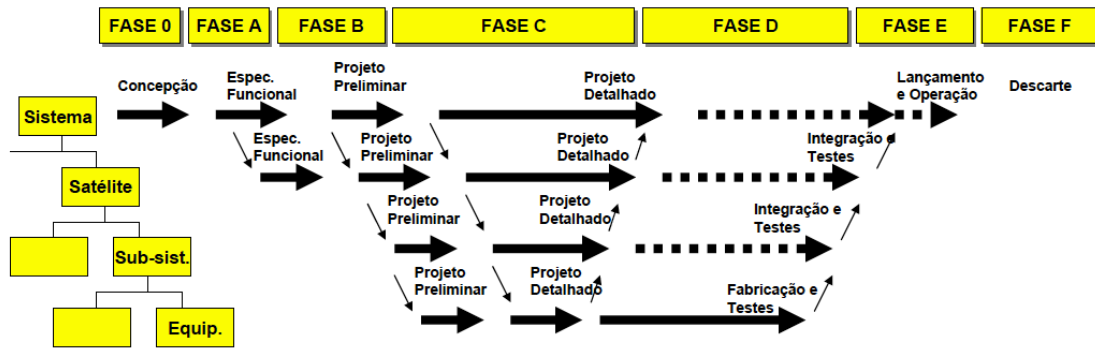


Figura 2-1: Exemplo de um ciclo de vida de uma missão espacial.
Fonte: Souza (2008).

2.1.1 Etapa de Concepção/Requisitos/Especificações

Durante a concepção, a missão, seus objetivos e ambientes, e os interessados, suas necessidades e interesses, são traduzidos em requisitos que são atendidos (total ou parcialmente) por especificações (funcionais, físicas, de interfaces etc.) do sistema.

2.1.2 Etapa de Projeto/Elaboração/Otimização

Durante o projeto de uma missão espacial, também são projetadas/elaboradas as arquiteturas funcional e física do sistema. A etapa de projeto é o momento em que a equipe de projeto é mais livre para criar, avaliar e corrigir as arquiteturas do sistema. Nessa etapa, o sistema ainda não foi construído, sendo, portanto, o intervalo de tempo ideal para se fazer avaliações, correções e tentativas de melhorias na arquitetura do sistema, com o menor custo associado. Após uma 1ª. proposta de arquitetura, pode haver alguma otimização.

Com o uso de modelagem e simulação, diferentes alternativas de arquiteturas podem e devem ser propostas e avaliadas com o objetivo de selecionar aquelas que satisfaçam melhor as especificações desejadas. Após isto, pode-se também otimizar as arquiteturas selecionadas segundo uma ou mais métricas; e, por fim, escolher uma para construção.

Ao término da etapa de projeto devem ser estabelecidas uma arquitetura funcional, uma arquitetura física e a alocação das funções da primeira para os componentes e subsistemas da segunda, dentre os elementos de forma do sistema que está sendo

projetado. A arquitetura física será utilizada para a construção do sistema, na etapa seguinte.

2.1.3 Etapa de Construção/Obtenção de Partes/Integração/Testes (VVA)

Durante a etapa de construção o sistema será materializado. A partir do início da construção do sistema, passamos a nos deparar com entidades de duas naturezas distintas: o modelo e o sistema real. As arquiteturas funcional e física são modelos do sistema.

Apesar do esforço do projetista em modelar o sistema na etapa de projeto, é comum que o mundo real o surpreenda. O resultado é que o mundo real sempre será mais complexo que o seu modelo, seja por falta de recursos, por falta de tempo ou por uma decisão de projeto.

Essa etapa é importante, pois é quando se elabora a ponte entre os modelos (arquiteturas) e o sistema como ele é. Tal ponte deve funcionar nos dois sentidos: as arquiteturas devem fornecer informações para que o sistema seja construído; e eventuais correções ou características não modeladas do sistema em construção devem ser realimentadas para os modelos (“as designed”), fidelizando-os ao que foi construído (“as built”). Por fim, o sistema deverá ser verificado, validado, aceito e até certificado (“VVA”).

2.1.4 Etapa de Operação/Manutenção/Descarte

Após a construção do sistema, o mesmo é posto em operação. Durante a sua operação esse sistema irá inevitavelmente evoluir e alterar suas características devido a fatores internos e externos. Essas alterações podem ser indesejadas ou desejadas, inesperadas ou esperadas e também podem ser alterações naturais ou forçadas.

A evolução do sistema durante a etapa de operação deve ser planejada para qualquer sistema que exerça funções críticas. É necessário prover meios para que o sistema evolua e permaneça executando as funções para o qual foi projetado. Alterações que por ventura surjam podem ser simples alterações naturais nos parâmetros do sistema ou até uma reconfiguração forçada dos elementos de forma (arquitetura física).

2.2 Uma Taxonomia de Arquiteturas

2.2.1 Definição de Arquiteturas

Existem várias definições para o conceito de arquitetura. Seguem algumas traduções de definições no contexto da Engenharia de Sistemas. Segundo elas, uma arquitetura é:

1. Uma descrição abstrata das entidades de um sistema e as relações entre essas entidades. (CRAWLEY et al., 2004);
2. A estrutura, arranjo ou configuração dos elementos de um sistema e suas relações internas necessárias para satisfazer restrições e requisitos. (ROSS, 2003);
3. O arranjo dos elementos funcionais em blocos físicos. (ULRICH; EPPINGER, 2000);
4. A materialização do conceito, a alocação de funções físicas/informacionais aos elementos de forma, e a definição de interfaces entre os elementos e com o contexto em seu entorno. (CRAWLEY, 2007).

Neste trabalho, utilizaremos a definição 4 devido ao fato de ser uma definição mais abrangente, que engloba o aspecto da funcionalidade do sistema e não somente da sua forma. Entende-se por **função** como sendo a resposta para a pergunta: “o que o sistema faz?”; e por **forma** como sendo a resposta para a pergunta: “como o sistema é?”.

A palavra “arquitetura”, assim como “sistema”, é uma palavra de grande abrangência, o que torna seu uso muito frequente e, às vezes, pode causar confusões. Quando se deseja ser específico, é uma boa prática sempre usar a palavra “**arquitetura**” associada ao **tipo** de arquitetura e a qual **sistema** a arquitetura pertence. Ou seja, ao invés de mencionar “a arquitetura do sistema” é mais prudente utilizar “a arquitetura <tipo da arquitetura> do <sistema>”.

2.2.2 Onde Estão Presentes

2.2.2.1 Sistemas Naturais ou Artificiais

As arquiteturas estão presentes em sistemas naturais (não projetados pelo homem) e artificiais (projetados pelo homem). Apesar de este trabalho focar os sistemas projetados pelo homem, é importante ressaltar a existência de arquiteturas em sistemas naturais, pois esses servem como exemplos de arquiteturas bem sucedidas e eficientes e foram utilizadas como inspiração no desenvolvimento do trabalho.

2.2.2.2 Sistemas Físicos e Não-Físicos

Resumindo Aristóteles (Estagira, 384 a.C. — Atenas, 322 a.C.) e outros, entende-se por sistemas físicos/não informacionais/não cibernéticos/concretos aqueles que possuem uma existência no universo objetivo/material, ou seja, compõem-se de matéria, energia, etc. Devido a essa existência, tais sistemas estão sujeitos às restrições impostas pela Natureza e estão subordinados às leis da Física, Química, etc. Exemplos desses sistemas são qualquer objeto dotado de matéria, energia, etc., como: galáxias, estrelas, planetas, veículos, computadores, sensores, atuadores etc.

Os sistemas não físicos/informacionais/cibernéticos/abstratos são aqueles que só existem no universo subjetivo/ideal, ou seja, compõem-se de ideias, relações, etc. Devido a esse fato o sistema é liberado de algumas restrições e se torna menos suscetível a restrições físicas (CRAWLEY et al., 2004). Porém o mundo físico ainda pode limitar algumas características de um sistema não físico. Por exemplo, um software (não físico) tem limitações para representar uma dízima periódica devido às limitações da eletrônica de hardware (físico). Exemplos de outros sistemas não físicos além de software são: modelos de sistemas de controle, organizações e protocolos de comunicação, etc.

2.2.3 Categorias de Arquiteturas

Faz-se necessário a distinção das arquiteturas em arquiteturas funcionais e arquiteturas físicas.

2.2.3.1 Arquitetura Funcional

A arquitetura funcional é a organização dos elementos funcionais (ou funções) necessários para atingir os objetivos e as especificações de um sistema. Essa arquitetura é construída como um modelo abstrato e imaterial, porém muito útil durante o desenvolvimento de sistemas. Em geral, a arquitetura funcional é projetada nas fases iniciais de desenvolvimento e, em seguida, seus elementos funcionais são alocados para os elementos da arquitetura física.

Os sistemas de controle são comumente modelados por meio de diagrama de blocos, que é uma maneira de representar o comportamento funcional de um sistema de controle. Portanto, pode-se afirmar que, neste caso, o diagrama de blocos é um exemplo de uma arquitetura funcional.

2.2.3.2 Arquitetura Física

A arquitetura física (ou estrutural) é a organização dos elementos de forma de um sistema. São os elementos da arquitetura física que são efetivamente construídos para que realizem as funções especificadas na arquitetura funcional. Apesar de um software ser um sistema não físico (imaterial), possui elementos de forma e com isso também possui uma arquitetura física (estrutural). A organização dos elementos de código-fonte e de documentação que compõem a estrutura de um software também será categorizada como arquitetura física.

Assim como a arquitetura funcional, a arquitetura física também é um modelo. Porém, esse modelo tenta se aproximar da construção física do sistema. Desenhos de projeto e diagramas de construção são exemplos de arquiteturas físicas.

2.3 Elaboração e Evolução das Arquiteturas

Durante a evolução de um sistema de controle ao longo do seu ciclo de vida, as arquiteturas desse sistema também evoluem. Esse processo de evolução pode caminhar de diversos modos. Nesta seção do trabalho descrevemos um modo de evolução que acreditamos ser aplicável a uma grande gama de sistemas.

2.3.1 Elaboração de Uma Arquitetura

Inicialmente, faz-se necessário compreender o processo de elaboração de um modelo de uma arquitetura de um sistema. O processo de elaboração é, na maioria das vezes, um processo iterativo que, em geral, atravessa as seguintes fases: criação, avaliação e correção. A Figura 2-2 ilustra o processo de elaboração de uma arquitetura proposta neste trabalho.



Figura 2-2: Processo de elaboração de uma arquitetura.

A criação é a fase em que a equipe de projeto propõe uma arquitetura. Em sistemas de controle há formas canônicas para a arquitetura funcional tanto para Controle Clássico, como para Controle Moderno. Entretanto, para outros tipos de arquiteturas físicas dificilmente tem-se uma forma canônica disponível. Nesses casos a criação tem que surgir da equipe de projeto.

A fase de avaliação serve para analisar a arquitetura recém-criada. Tal análise (RELIASOFT, 2010) pode ser feita de forma subjetiva e intuitiva ou por meio de métodos e técnicas objetivas e descritivas. Um processo comum de avaliação é a realização de revisões de projeto, momento em que outros profissionais experientes podem analisar uma arquitetura proposta. Esse processo identifica falhas e potenciais problemas na arquitetura avaliada, os quais podem ser corrigidos na fase seguinte.

A fase de correção/seleção é o momento para, se necessário, se realizarem ajustes em uma arquitetura proposta e avaliada. Muitas vezes, pequenos ajustes são suficientes para a finalização da arquitetura sem a necessidade de se reprojeter (fase de criação). Então, após essa fase de correção, o processo pode ser finalizado com uma arquitetura elaborada; pode voltar para a avaliação; ou até mesmo voltar para a fase de criação.

Ainda há a possibilidade da elaboração de diversos modelos concorrentes seguindo o processo descrito anteriormente. Então, acontece uma seleção entre essas arquiteturas candidatas buscando otimizar alguma(s) métrica(s). Esse processo é normalmente usado em grandes projetos onde há recursos para financiar equipes distintas que concebem arquiteturas diferentes para atacar um mesmo problema. Então, ao final, um projeto é escolhido para ser implementado. Atualmente, com as ferramentas computacionais disponíveis, essa técnica pode ser usada até mesmo em projetos menores e com poucos recursos financeiros. Há também ferramentas de software para geração automática de modelos de arquiteturas que podem ser usadas nesse processo de seleção. Essa técnica será descrita a seguir.

2.3.2 Evolução ao Longo do Ciclo de Vida

Essa seção tem o objetivo de melhorar o entendimento das relações entre as diversas arquiteturas elaboradas para desenvolver um sistema. A Figura 2-3 traz um panorama das arquiteturas presentes nas diferentes fases do ciclo de vida de um sistema de controle. A mesma ilustração fornece também uma ideia de como caminha o fluxo de informações entre essas arquiteturas, indicado pelas setas numeradas. O sentido da seta indica o sentido do fluxo de informação predominante; ou seja, apesar do fluxo mais significativo ser o indicado, também pode haver fluxo de informação no sentido contrário.

Pressupondo uma Etapa de Concepção, a jornada tem início na Etapa de Projeto com a elaboração dos modelos das arquiteturas funcionais do sistema, os quais definem o seu comportamento. Em seguida, passa-se para a elaboração dos modelos das arquiteturas físicas do sistema que definem como o sistema será construído. O conjunto dos modelos de arquiteturas será utilizado para compor o projeto do produto a ser construído. A Etapa de Projeto finaliza com a aprovação desse projeto em eventos como o *Preliminary Design Review* (PDR) e no *Critical Design Review* (CDR).

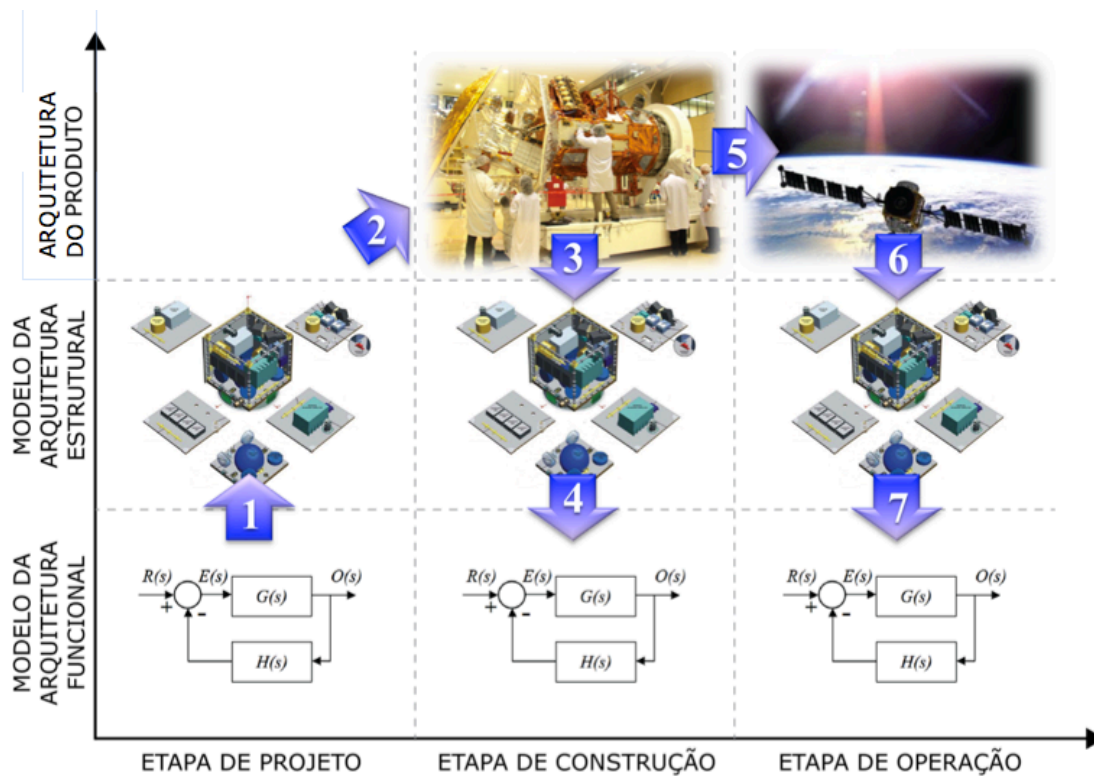


Figura 2-3: Evolução das diferentes arquiteturas presentes no desenvolvimento de um sistema.

A Etapa de Construção se inicia com a materialização dos modelos em um produto. Durante a fabricação do produto surgem divergências entre os modelos e o produto. Uma vez avaliadas tais divergências, toma-se uma das duas decisões: 1) o produto deve ser corrigido de acordo com os modelos; ou 2) os modelos devem incorporar as alterações do produto. É importante que as alterações sejam realimentadas nos modelos, pois assim pode-se garantir a fidelidade do modelo com o produto como construído (“as built”). Essa etapa finaliza-se com a verificação, validação e até certificação do produto construído.

Por fim, a Etapa de Operação, que se inicia quando o produto é posto em operação. Durante a operação o produto sofrerá alterações que podem ser provenientes do ambiente externo ou até mesmo de interação interna de seus componentes. Assim como dito anteriormente, essas alterações podem até ser planejadas, como uma possível reconfiguração de algum subsistema. Mais uma vez essas alterações devem ser realimentadas nos modelos das arquiteturas. Essa etapa finaliza-se com o descarte ou a morte do sistema.

2.4 Linguagens de Modelagem

Há inúmeras linguagens de modelagem, sendo cada uma delas mais adequada para um determinado propósito. Aqui serão descritas algumas linguagens de modelagem candidatas a serem utilizadas no trabalho proposto. As mais comuns são aquelas que servem para a modelagem de uma arquitetura particular, ou seja, cada alternativa de arquitetura será um modelo diferente para o sistema de interesse. Atualmente há opções de modelagem em um nível de abstração superior, ou seja, metalinguagens, que são utilizadas para modelar meta-arquiteturas que por sua vez instanciam várias opções de arquiteturas. Diante de tal diversidade focaremos nas linguagens e seus diagramas que serão úteis para a extração de métricas do sistema de interesse, a saber:

2.4.1 Diagrama de Funções

Os diagramas de funções modelam como será a arquitetura funcional, ou seja, como o sistema operará. Os diagramas de funções são na verdade uma categoria de diagrama. Existem várias representações diferentes que podem ser considerados diagramas de funções como: diagramas de blocos, diagramas de fluxo de sinais, fluxogramas, máquinas de estado, cartas de estado, etc. entre outros tipos de desenhos funcionais.

Na **Engenharia de Controle** a linguagem de modelagem mais difundida é o **Diagrama de Blocos de Funções de Transferência**. Ele é uma interconexão de símbolos representando determinadas operações matemáticas de forma que o diagrama como um todo obedece ao modelo matemático do sistema (CLOSE; FREDERICK 1995). Deste modo, o diagrama de blocos (observar a Figura 2-4) é utilizado para analisar o comportamento dinâmico de um sistema, com as linhas representando as variáveis temporais e os blocos representando as funções que alteram as variáveis. As linhas indicam um fluxo de sinal entre os blocos e os seus sentidos indicam a direção da causalidade conforme visto em Takahashi, Rabins e Auslander (1970).

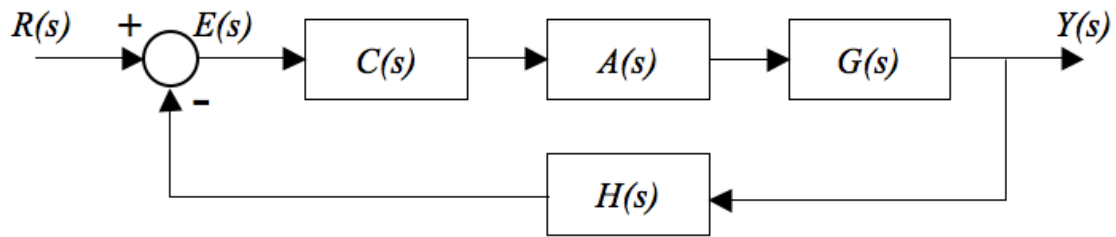


Figura 2-4: Exemplo de um diagrama de blocos de um sistema de controle.

Esse tipo de representação é utilizado para modelar a arquitetura funcional do sistema; ou seja, com esta linguagem de modelagem é possível definir como o sistema irá se comportar, mas não necessariamente determina como ele será implementado. Ou seja, quando se define que o controlador $C(s)$ da Figura 2-4 será o seguinte PID:

$$C(s) = K_P + \frac{1}{K_i \cdot s} + K_D \cdot s \quad (2.1)$$

determinamos o comportamento da relação entrada-saída do sistema de controle em questão. Entretanto, este mesmo controlador $C(s)$ pode ser implementado de diversas maneiras e.g.: capacitores, resistores e amplificadores operacionais; um único circuito integrado de PID; microprocessador; DSP, etc. Cada uma dessas opções pode ser construída com componentes que possuam confiabilidades diferentes; conversores A/D com resolução e períodos de amostragem diferentes; ou softwares com diferentes arquiteturas. Devido a esses fatos, faz-se necessário uma linguagem para modelarmos a arquitetura física do sistema de interesse.

2.4.2 Diagramas de Componentes

Em contraste com os diagramas de funções, os diagramas de componentes modelam como será a arquitetura física, ou seja, como o sistema será implementado. Esse tipo de modelo é importante, pois existem inúmeras maneiras de se implementar um sistema para desempenhar a mesma função. A partir desse tipo de modelo é possível extrair várias métricas do sistema de interesse. Os diagramas de componentes modelam as conexões físicas entre os componentes, podendo incluir também a maneira como eles estão posicionados.

Os diagramas de componentes são na verdade uma categoria de diagrama. Existem várias representações diferentes que podem ser considerados diagramas de componentes como: diagramas de blocos de confiabilidade, esquemas elétricos, desenhos mecânicos, etc., entre outros tipos de desenhos esquemáticos.

Na Engenharia de Confiabilidade, o **Diagrama de Blocos de Confiabilidade** (*Reliability Block Diagrams-RBDs*) pode ser projetado de forma a ser considerado um diagrama de componentes, conforme mostra a Figura 2-5.

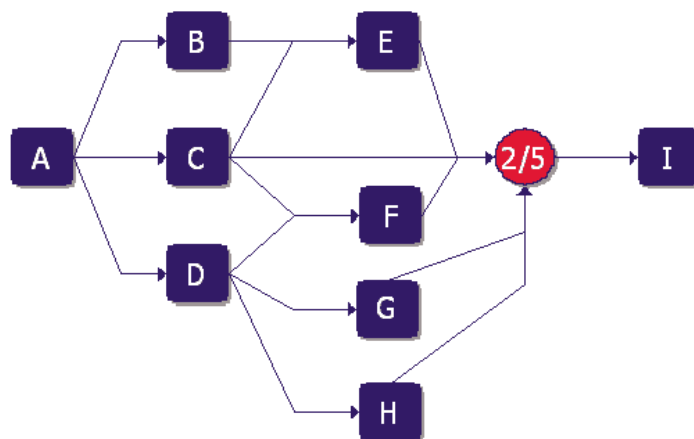


Figura 2-5: Exemplo de um *Reliability Block Diagram* (RBD).
Fonte: Reliasoft (2010)

Na Engenharia de Sistemas, o diagrama chamado de *System Breakdown Structure* é um diagrama que apresenta a estrutura hierárquica de um determinado sistema, ver a Figura 2-6. Esse tipo de diagrama também será considerado como um diagrama de componentes.

Ainda na **Engenharia de Sistemas**, os **Diagramas de Contexto** podem representar diferentes facetas de um sistema e diferentes níveis de detalhamento desse sistema. Além de representar itens materiais esses diagramas podem representar estruturas de software.

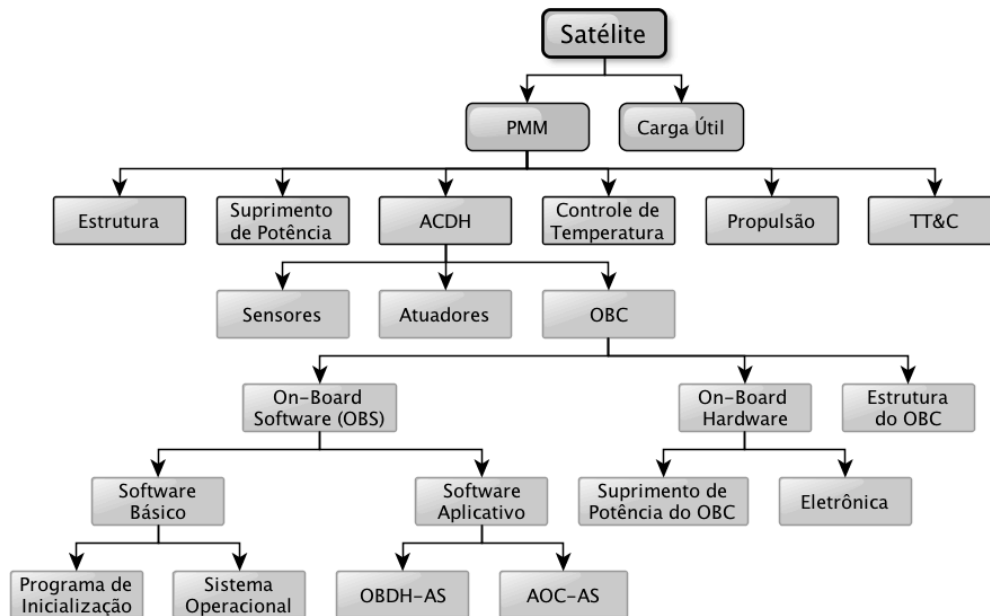


Figura 2-6: Exemplo de *System Breakdown Structure* de um satélite como a PMM.

2.4.3 Grafos e Árvores

Conforme descrito em Cardoso (2011), a origem da **Teoria dos Grafos** é, em geral, associada ao problema das pontes de Königsberg (cidade da Prússia que agora se designa por Kaliningrad). Parte desta cidade se localizava em duas ilhas do rio Pregel as quais estavam ligadas às margens e uma à outra através de 7 pontes, conforme ilustrado na Figura 2-7. Consta que os habitantes de Königsberg não conseguiam encontrar uma rota (com partida e chegada a um mesmo lugar) que lhes permitisse atravessar apenas uma vez cada uma das pontes. O matemático Leonhard Euler resolveu este problema em 1735, indicando a impossibilidade da existência de tal percurso. Para tanto, Euler fez uso de um **grafo** para modelar o problema. O grafo também está ilustrado na Figura 2-7. Seu trabalho foi originalmente publicado sob o título de "Solutio Problematis ad Geometriam Situs Pertinentis" (em tradução livre: Solução do Problema Relacionado com a Geometria da Posição) (EULER, 1741). As ilustrações originais do artigo de Euler podem ser vistas na Figura 2-8.

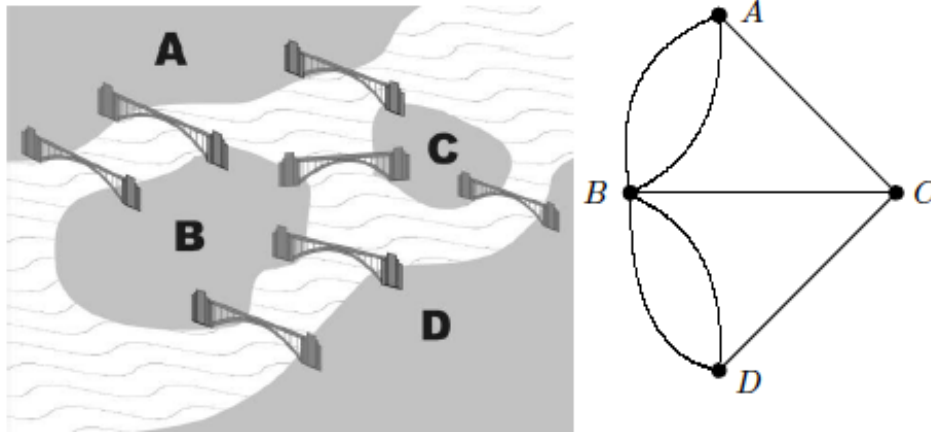


Figura 2-7: Associação das Pontes de Königsberg com o respectivo grafo.
 Fonte: Cardoso (2011)

Em uma definição moderna, um grafo $G = G(V,E)$ é uma estrutura entre V e E , sendo V um conjunto discreto finito e não vazio, e E uma relação binária sobre V . Os elementos de V são representados por pontos. O par ordenado $(v,w) \in E$ (ou simplesmente vw), onde $v,w \in V$, é representado por uma linha ligando v a w . Os elementos do conjunto E são denominados de arestas, linhas ou arcos do grafo. Os elementos do conjunto V são denominados de vértices, pontos ou nós do grafo (RABUSKE, 1992 citado por OSTROSKI, 2009).

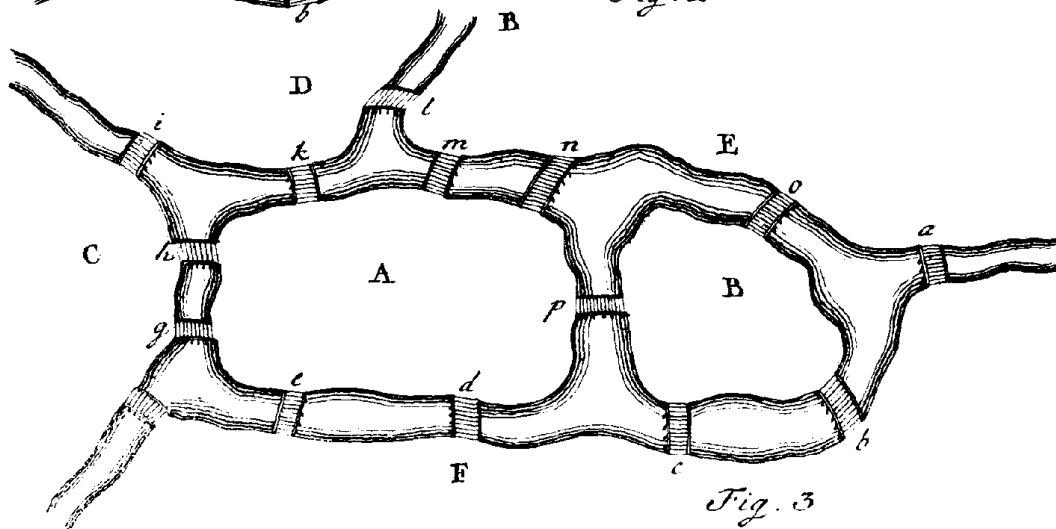
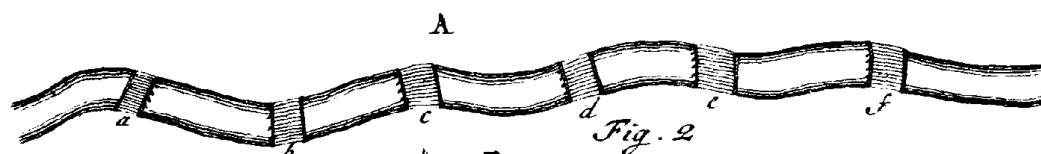
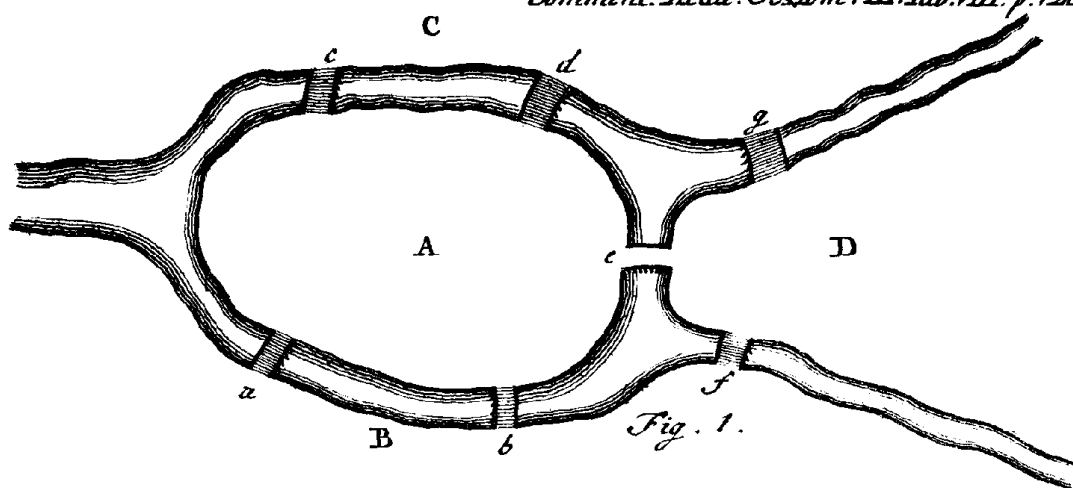


Figura 2-8: Ilustrações Originais do Artigo de Euler sobre as Pontes de Königsberg.
Fonte: Euler (1741)

Entre os grafos há uma espécie particular chamada de **árvore**. De acordo com Lodder (2012), o termo árvore foi cunhado em 1857 por Arthur Cayley para descrever a ramificação lógica que ocorre com a iteração do processo da diferenciação parcial. A primeira publicação do autor sobre esse assunto é intitulada "On the theory of the analytical forms called trees" (CAYLEY, 1857). O próprio autor encontrou outras aplicações para este modelo, como afirma Ostroski (2009), Arthur Cayley utilizou a idéia de árvores para enumerar isômeros dos hidrocarbonetos alifáticos saturados,

para a Química Orgânica. Ilustrações originais dos artigos sobre árvore do autor podem ser vistas na Figura 2-9.

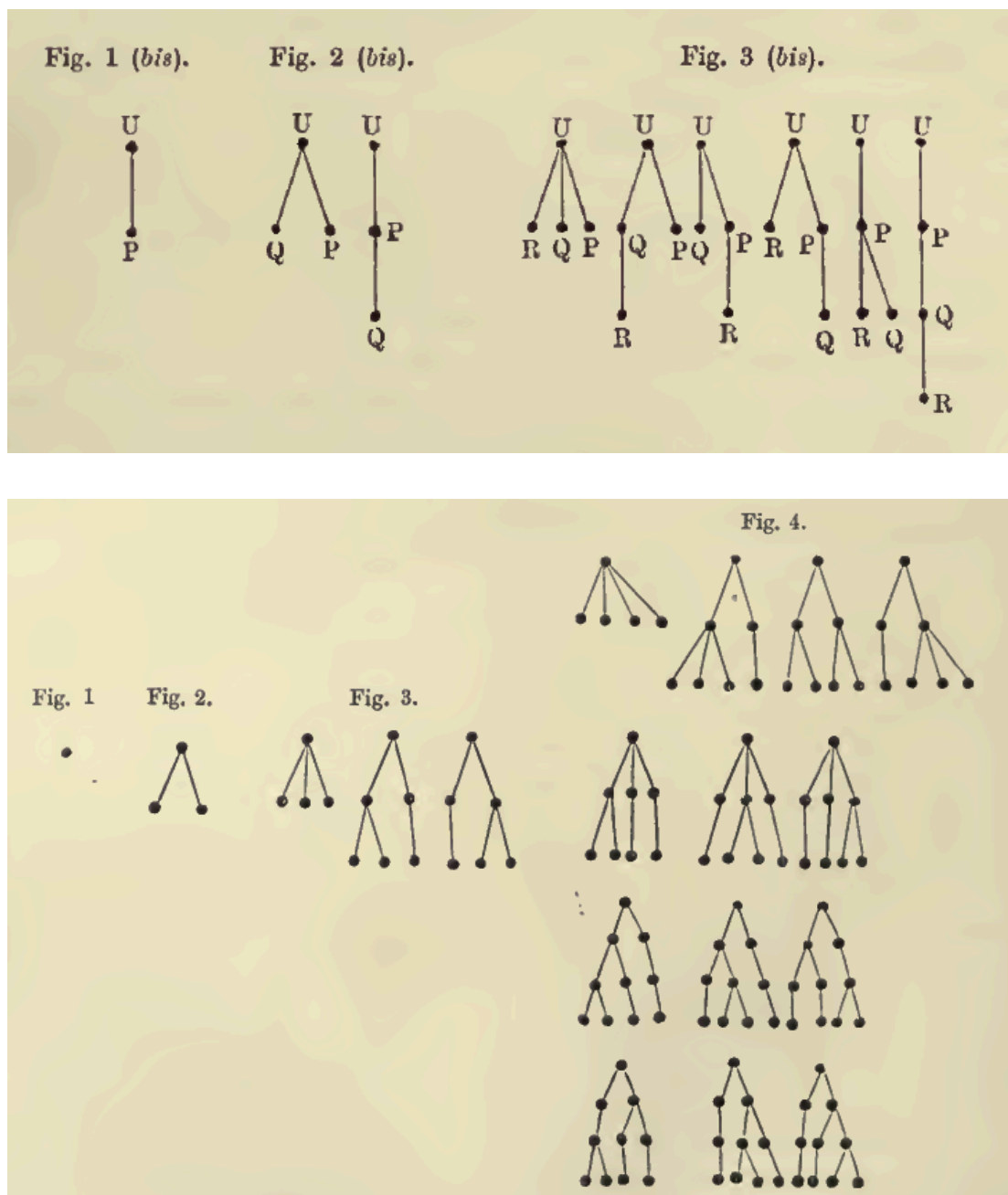


Figura 2-9: Ilustrações Originais dos Artigos de Arthur Cayley sobre Árvores.
 Fonte: Cayley (1857) e Cayley (1859)

Dada a definição anterior de um grafo, uma árvore pode ser definida simplesmente como um grafo sem ciclos, este fato faz da árvore a estrutura mais "econômica" de conexão entre seus vértices (JURKIEWICZ, 2009).

2.5 Atributos e Medidas

Durante o processo de projeto de sistemas nos deparamos com a necessidade de avaliar e comparar esse sistema com soluções alternativas para o mesmo problema. Neste processo temos que simplificar a complexidade inerente deste sistema para conseguirmos avaliar algumas de suas características individualmente. Para tanto, definimos alguns atributos que nos fornecem informações relevantes sobre o sistema de interesse.

Uma vez escolhidos os atributos que irão caracterizar o sistema em estudo, o passo seguinte é medi-los para que possam ser avaliados. De acordo com Eusgeld, Freiling e Reussner (2010): “É importante ressaltar a diferença entre o atributo e sua medida. Por exemplo: a complexidade de um software pode ser medida de várias maneiras. Entretanto, a diferença entre o atributo e sua medida se confundem, pois medidas também são usadas para definir atributos”.

2.5.1 Atributos da Engenharia de Controle

A Engenharia de Controle possui uma riqueza de atributos bem estabelecidos e muitas vezes matematicamente definidos. Esses atributos são medidos ou testados já no início da etapa de projeto a partir dos modelos da arquitetura funcional do sistema de controle. Serão descritos a seguir os atributos mais tradicionais utilizados na caracterização de um sistema de controle.

2.5.1.1 Estabilidade

Dentre os atributos da Engenharia de Controle, a estabilidade é um dos mais importantes, em geral, é o primeiro a ser analisado. A preocupação com a estabilidade já estava presente no artigo clássico de James Clerk Maxwell, “On Governors” de 1868, considerado o primeiro artigo significativo sobre o mecanismo de realimentação. Vale a pena observar a descrição pelas palavras do autor (MAXWELL, 1868):

“Será observado que o movimento da máquina com o seu governador consiste de um movimento geralmente uniforme, combinado com uma

perturbação que pode ser expressa como a soma de várias componentes de movimento. Esses componentes podem ser de quatro tipos diferentes:

- (1) A perturbação pode aumentar continuamente.
- (2) Ela pode diminuir continuamente.
- (3) Ela pode ser uma oscilação de amplitude continuamente crescente.
- (4) Ela pode ser uma oscilação de amplitude continuamente decrescente.

O primeiro e o terceiro caso são evidentemente inconsistentes com a estabilidade do movimento; e o segundo e o quarto somente são admissíveis em um bom governador. Essa condição é matematicamente equivalente à condição de que todas as raízes possíveis, e todas as partes possíveis das raízes impossíveis, de uma certa equação devam ser negativas.”

A partir dessa citação pode-se observar que James Clerk Maxwell observa a semelhança entre o comportamento do mecanismo e o comportamento de equações do movimento, resultando em um movimento amortecido quando as raízes da equação do movimento possuem a parte real (possível) negativa. A partir dessa observação, o autor modela matematicamente o governador. A “certa equação” mencionada pelo autor é conhecida como equação característica e pode ser obtida pelo denominador da função de transferência de sistema de malha fechada.

Esse critério de estabilidade ainda é válido hoje, mas pode não ser suficiente para fornecer uma escala métrica para medir essa estabilidade. Algumas das métricas que podem ser utilizadas para medir a estabilidade são a **Margem de Ganho** e a **Margem de Fase** introduzidas por Hendrik Wade Bode em 1947. Esses conceitos estão explicados em Takahashi, Rabins e Auslander (1970), a Margem de Ganho é o quanto de aumento de ganho em decibéis (dB) que a malha aberta de um sistema pode ser submetida antes de se tornar instável em malha fechada. Da mesma maneira, a Margem de Fase é o quanto de atraso de fase em graus (°) que a malha aberta de um sistema pode sofrer antes de se tornar instável em malha fechada. A Figura 2-10 contém ilustrações da resposta em frequência de sistemas de controle assintoticamente estáveis, marginalmente estáveis (em seu limite de estabilidade), e instáveis. A mesma

figura também apresenta como são obtidas as margens de ganho e fase para aqueles sistemas assintoticamente estáveis.

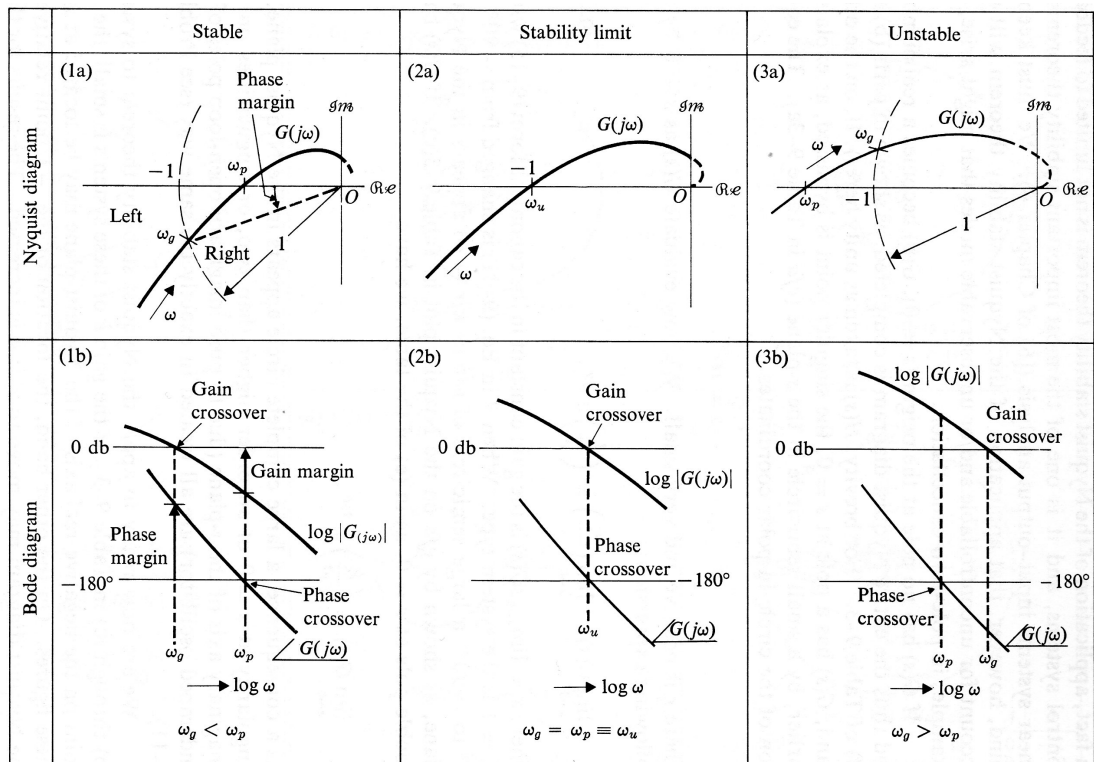


Figura 2-10: Resposta em frequência de sistemas de controle e suas margens de ganho e fase. Fonte: Takahashi, Rabins and Auslander (1970).

Além dessas métricas mencionadas anteriormente existem outras. O estudo da estabilidade evoluiu bastante, sendo considerado um ramo de pesquisa próprio: a Teoria da Estabilidade.

2.5.1.2 Desempenho

Após a confirmação de que o sistema de controle é assintoticamente estável, passamos a analisar outros atributos, como o seu desempenho (ou performance). Para medir o desempenho de uma arquitetura de um sistema de controle existem diversas métricas. Algumas das mais tradicionais são observadas a partir da resposta transitória do sistema a uma entrada degrau.

A Figura 2-11 apresenta a resposta transitória típica da variável de saída de um sistema de segunda ordem subamortecido a uma entrada degrau. Nesta mesma imagem estão ilustradas as métricas mais tradicionais de desempenho, que são:

Tempo de Subida (*Rise Time* - t_r), **Tempo de Acomodação** (*Settling Time* - t_s), e **Porcentagem de Sobresinal** (*% Overshoot*).

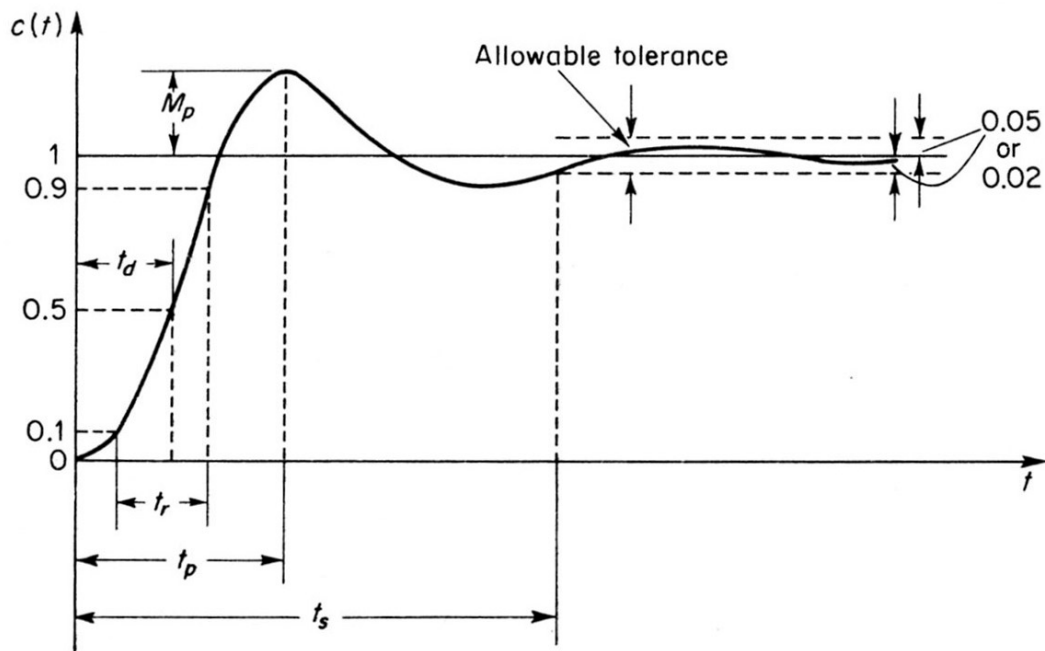


Figura 2-11: Resposta transitória típica da variável de saída de um sistema de 2ª ordem subamortecido a uma entrada degrau.
Fonte: Ogata (1993).

2.5.1.3 Controlabilidade e Observabilidade

Esses conceitos estão sucintamente explicados em (CITRON, 1969) de forma intuitiva. O autor descreve a **Controlabilidade do Estado** de um sistema como a habilidade de garantir que esse sistema, a partir de qualquer estado inicial, pode ser transferido para qualquer estado final desejado em um tempo finito. Do mesmo modo, a **Observabilidade do Estado** é a expressão da habilidade de determinar em um dado tempo o estado do sistema baseado em medidas das saídas do sistema realizadas em um intervalo de tempo finito.

Os atributos de controlabilidade e observabilidade foram introduzidos na Teoria de Controle Moderno por Rudolf Emil Kalman em 1960 nos trabalhos: “Contributions to the Theory of Optimal Control” (KALMAN, 1960); e “On the General Theory of Control Systems.” (KALMAN, 1960). Além das suas definições, o autor estabeleceu condições para testar se um modelo linear, invariante no tempo de um determinado sistema é controlável e/ou observável.

Antes de definir a controlabilidade o autor estabelece o modelo do sistema em estudo, conforme pode se observar em Kalman (1960):

“Nós devemos estudar o sistema representado pelas equações:

$$\frac{dx}{dt} = F(t)x(t) + G(t)u(t) \quad (2.2)$$

$$y(t) = H(t)x(t) \quad (2.3)$$

onde: u é um vetor m , x é um vetor n , y é um vetor p ; $F(t)$, $G(t)$, assim como $H(t)$ são matrizes retangulares contínuas em t , e qualquer uma dessas pode ser singular.

Na visão da motivação física do nosso problema, nós adotamos a seguinte terminologia: as Equações (2.2-3) são a *planta* (ou *modelo*); $x(t)$ é o seu estado; $u(t)$ é a sua *função de controle* ou sua *entrada*; e $y(t)$ é a *saída* da *planta*. A *planta* é constante se F, G, H são constantes. Se $u(t) = 0$ ou $G(t) = 0$ a *planta* é *livre*.”

No mesmo artigo, o comportamento da planta é descrito pela solução das suas equações diferenciais, e estabelece:

“A solução [...] é convenientemente apreciada como o movimento do estado de (2.2); o que leva a notação:

$$x(t) = \phi_u(t; x_0, t_0) \quad (2.4)$$

Lê-se: o movimento de (2.2) iniciando no estado inicial x_0 no tempo t_0 e observado no tempo t , é influenciado pela função de controle fixa $u(t)$ definida no intervalo $[t_0, t]$.”

Para então definir formalmente a controlabilidade como:

“Um estado x é dito ser *controlável no tempo* t_0 se existe uma função de controle $u^1(t)$, dependente de x e t_0 e definido sobre um intervalo fechado finito $[t_0, t]$, tal que $\phi_{u^1}(t; x, t_0) = 0$. Se isso for verdade para cada estado de x , nós dizemos que a planta é *completamente controlável no tempo* t_0 ; se

isto for verdade para cada t_0 , nós dizemos simplesmente que a planta é *completamente controlável*.”

Antes de definir a observabilidade o autor introduz o conceito de dualidade, da seguinte maneira:

“Agora nós buscamos caracterizar a planta de acordo como as propriedades $t^* = -t$ e $F^*(t^*) = F'(t)$, $G^*(t^*) = H'(t)$, e $H^*(t^*) = G'(t)$. Então:

$$\begin{aligned} dx^*/dt^* &= F^*(t^*)x(t)^* + G^*(t^*)u^*(t^*) \\ y^*(t^*) &= H^*(t^*)x^*(t^*) \end{aligned} \quad (2.5)$$

onde x^* , u^* , y^* são vetores n , p , e m respectivamente, é a *planta dual* de (2.2-3).”

Para então definir a observabilidade convenientemente como: “A planta (2.2-3) é *uniformemente completamente observável* se seu dual é uniformemente completamente controlável.”

Os conceitos de controlabilidade e observabilidade são referidos como *duais* porque a controlabilidade envolve a relação entre as variáveis de estado e as entradas deste sistema (o vetor de controle), enquanto a observabilidade envolve a relação entre as variáveis de estado e as saídas do sistema (o vetor de saídas) (CITRON, 1969).

Apesar da importância dos atributos de controlabilidade e observabilidade, ainda não há uma escala métrica para medir suas graduações. Esse fato dificulta o uso desses atributos em métodos objetivos de elaboração de arquiteturas. Entretanto, podem ser utilizados na seleção de arquiteturas, eliminando aquelas que não satisfazem os testes de controlabilidade e observabilidade.

2.5.2 Atributos da Engenharia de Sistemas

A Engenharia de Sistemas está preocupada em auxiliar as pessoas a projetar sistemas corretos e adequados, enquanto a Engenharia de Controle está preocupada em garantir que o sistema irá se comportar da maneira desejada. Portanto, é natural que a Engenharia de Sistemas possua uma outra gama de atributos diferentes daqueles

usados na Engenharia de Controle. Em Amoroso (1999), pode-se encontrar um método de análise que leva em consideração atributos de desempenho, custo e confiabilidade para a especificação de sistema de rodas de reação. Tal trabalho apresenta uma interessante análise de múltiplos atributos, oriundos tanto da Engenharia de Controle como da Engenharia de Sistemas.

A Engenharia de Sistemas é mais jovem que a Engenharia de Controle e é filha parcial desta (GOODE; MACHOL; TEICHMANN, 1957). Devido a esse fato e por tratar problemas mais amplos e complexos, a primeira não alcançou o mesmo nível de maturidade da segunda. Portanto, é natural que os atributos da Engenharia de Sistemas não sejam ainda tão formalmente definidos como aqueles da Engenharia de Controle. Tal fato é visto aqui como uma oportunidade para pesquisa.

Dentre os diversos atributos que podem ser utilizados para caracterizar um sistema, segundo a Engenharia de Sistemas, há um subconjunto particularmente interessante e geral, utilizado para caracterizar a **Dependabilidade** de um sistema. Em Avizienis et al. (2004) pode-se encontrar duas definições de dependabilidade. A primeira enfatiza a necessidade para justificarmos a confiança em um sistema:

“... dependabilidade é a capacidade de fornecer um serviço no qual se pode confiar de modo justificável.”

A segunda definição enfatiza o critério para decidir se é um serviço do qual se pode depender:

“... a dependabilidade de um sistema é capacidade de evitar falhas que são mais frequentes e mais severas do que é aceitável.”

Apesar do conceito de dependabilidade ser interessante e de fácil compreensão, esta é uma maneira bastante abrangente de se caracterizar um sistema. Faz-se necessária uma compartimentação do conceito de dependabilidade em diferentes atributos que pode ser definidos em escopos mais específicos. Em Avizienis et al. (2004), estão definidos os seguintes atributos que compõem o conceito de dependabilidade:

- Disponibilidade: prontidão para o serviço correto.
- Confiabilidade: continuidade do serviço correto.

- Segurança (*Safety*): ausência de consequências catastróficas não intencionais para os usuários e o ambiente.
- Integridade: ausência de alterações impróprias do sistema.
- Manutenibilidade: capacidade de suportar modificações e reparos.
- Confidencialidade: ausência de divulgação não autorizada de informações.

2.5.2.1 Complexidade

Inúmeros trabalhos de diferentes áreas do conhecimento fazem uso do termo **Complexidade**, entre eles podem ser citados a Física, Biologia, Sociologia, Meteorologia, Computação e a Engenharia. O crescente aumento da complexidade permeia a nossa sociedade em todas as suas facetas, sendo esse mesmo fenômeno efeito do avanço tecnológico da humanidade e também a causa de muitos de seus problemas.

O termo complexidade é de difícil definição. De acordo com Lee (2003), muitas maneiras formais e informais de discutir complexidade estão centradas sobre a noção básica de dificuldade e não há um consenso sobre a definição deste termo. Ainda em Lee (2003), percebe-se que a definição de complexidade deve ser cuidadosamente delineada. Por exemplo, a linguagem de modelagem escolhida para descrever o objeto de estudo, a escala ou nível de detalhe e as especificidades ou os aspectos particulares em consideração devem fazer parte da discussão da complexidade. Dado que este trabalho lida com o projeto e construção de sistemas, então a complexidade será analisada aqui sob o ponto de vista da Engenharia. Para tanto, encontra-se a seguinte definição em Lee (2003):

“...complexidade é a propriedade de um sistema que torna-o difícil de ser entendido como um todo através da coleção de conhecimento sobre seus constituintes. Ou em outras palavras, uma das características essenciais de um sistema complexo é seu comportamento emergente (ou coletivo) que não é rapidamente compreensível ou previsível pelas propriedades de seus componentes individuais.”

De acordo com Mcdermid (2000), a complexidade pode ser vista como tendo os seguintes aspectos chave:

- Escala: o número de elementos em um sistema;
- Diversidade: o grau com que o sistema é composto por elementos diferentes;
- Conectividade: a inter-relação entre os elementos do sistema.

Ao analisar a complexidade intrínseca de um objeto, esses aspectos listados anteriormente cobrem as relações intercomponentes; além destes, ainda podem ser levados em consideração os aspectos intracomponentes, como visto em Phukan, Kalava E Prabhu (2005) e Martin (2004). Além da complexidade intrínseca de um objeto de estudo esses trabalhos ainda trazem uma discussão sobre a complexidade extrínseca ou complexidade externa que trata sobre as interfaces entre este objeto e o ambiente que o cerca.

Entre os diversos fatores que conduzem ao aumento da complexidade de um objeto, pode-se destacar o uso de componentes *Comercial-Off-The-Shelf* (COTS). A tendência de usar componentes COTS na engenharia de um sistema introduz nesse uma quantidade de funcionalidades (ou capacidades) além do necessário (MCDERMID, 2000).

É compreensível que temos que conviver com a complexidade dos sistemas e não há uma estratégia única nem efetiva para lidar com esse problema, porém, não dominar a complexidade pode ser a "sentença de morte" do projeto de um sistema (MCDERMID, 2000). É necessário preocupar-se com a complexidade durante todas as fases de desenvolvimento de um sistema. A experiência mostra que devem ser evitadas "soluções-únicas", soluções nas quais engenheiros e gerentes tornam-se presos a uma abordagem particular e não consideram alternativas devidamente (MCDERMID, 2000).

De acordo com Lloyd (2001), para definir uma métrica de complexidade, há três pontos que os pesquisadores frequentemente se questionam quando desejam quantificar um objeto sob estudo. São eles:

1. Quão difícil é para descrevê-lo?
2. Quão difícil é para criá-lo?
3. Qual é o seu grau de organização?

No trabalho, (LLOYD, 2001) encontra-se uma extensa, porém incompleta, lista de métricas sobre complexidade. Essas estão divididas entre as questões listadas anteriormente. Entre elas encontra-se até o valor "custo financeiro" como uma métrica para complexidade com uma medida da dificuldade de criação de um objeto.

O trabalho de Phukan, Kalava e Prabhu (2005) busca definir métricas de complexidade (C) para uma arquitetura de controle de uma manufatura extraíndo conceitos e métricas utilizadas no domínio de software. Neste trabalho o autor utiliza uma métrica que combina a complexidade intercomponentes com a complexidade intracomponente, na seguinte forma:

$$C = \text{Complexidade intracomponente}(\text{Complexidade intercomponente})^2 \quad (2.6)$$

Elevar ao quadrado a complexidade intercomponentes caracteriza o fato que o fluxo de informação contribui para a complexidade física de uma entidade em uma relação não-linear com a complexidade intercomponentes (SHEPPERD, 1993 apud PHUKAN; KALAVA; PRABHU, 2005).

2.5.3 Métricas

De acordo com Eusgeld, Freiling e Reussner (2010): “Qualquer medição é uma forma de abstração: esta reduz a complexidade de um ou de diversos atributos do sistema original em um único símbolo. Os principais propósitos desta forma de abstração são para classificar e comparar sistemas”. As medições estão associadas com a noção de métrica, que será descrita a seguir.

Ainda em Eusgeld, Freiling e Reussner (2010) uma métrica, de forma mais generalizada, pode ser formalizada como uma função M que mapeia um sistema particular do conjunto de sistemas S em um elemento do conjunto V :

$$M : S \mapsto V \quad (2.7)$$

Matematicamente, uma métrica é a abstração do conceito de distância. Formalmente, uma métrica d para um conjunto X é a função que atribui um valor de “distância” (um número real) a um par de elementos de X :

$$d : X \times X \mapsto R^+ \quad (2.8)$$

A função d deve satisfazer a diversas condições para ser considerada uma métrica matemática, i.e. para todo $x, y, z \in X$:

- toda distância é não-negativa, i.e., $d(x, y) \geq 0$,
- a distância é zero para entradas idênticas, i.e., $d(x, x) = 0$,
- a distância é simétrica, i.e., $d(x, y) = d(y, x)$, e
- a desigualdade do triângulo é válida, i.e., $d(x, z) \leq d(x, y) + d(y, z)$.

2.6 Abordagens Multiobjetivo

2.6.1 Otimização Multiobjetivo

A otimização é o processo de busca pela melhor solução possível (solução ótima) para um determinado problema. Não faz sentido falar de uma solução simplesmente ótima, pois uma solução tem que ser ótima segundo algum **critério**. Isto força a necessidade de postular **um problema bem posto** que possua um critério bem estabelecido, o qual se deseja otimizar. Os atributos descritos anteriormente não são suficientes para postular um problema de otimização, mas suas **métricas**, sim, podem e devem ser usadas na formulação desses problemas.

Existem vários métodos e técnicas para a busca de soluções ótimas. Matematicamente a solução ótima é obtida pela maximização ou minimização de uma métrica escolhida, o que nem sempre é solucionável com a matemática analítica atual. Muitas vezes, se recorre a **soluções numéricas** que varrem o espaço do problema em busca de soluções ótimas. Entretanto, essa prática resulta em solução candidatas a ótimas, uma vez que não há garantia da inexistência de uma solução ainda melhor para o problema. Mesmo assim, esses métodos auxiliam e muito a solucionar problemas complexos e de difícil **solução analítica**.

A otimização mencionada até o momento fala somente da busca por soluções baseadas em um critério somente, que é conhecida como **Otimização Mono-Objetivo**. A extrapolação desse conceito para vários critérios (chamados de **funções objetivo**) simultaneamente nos leva à **Otimização Multiobjetivo**. Esse ramo, além de

reutilizar os conceitos de otimização mono-objetivo estabelecidos, definiu uma nova classe de problema: a busca pelo melhor compromisso entre as soluções ótimas.

A busca pela solução ótima para um problema multiobjetivo herda grande parte do conhecimento adquirido para a solução do problema mono-objetivo e expande esse conhecimento com novas técnicas. Os principais avanços foram obtidos por Vilfredo Pareto em 1906 no seu trabalho: “*Manuale di Economia Politica con una Introduzione alla Scienza Sociale*” (PARETO, 1906). A técnica conhecida como *Otimização de Pareto* busca soluções de compromisso. Esse processo é explicado em (SOUSA, 2003):

“[...] uma solução ótima no espaço de funções objetivo é aquela em que a redução no valor de uma destas, não implique em um aumento no valor de nenhuma outra. Uma otimização de Pareto resulta em um conjunto de soluções que atendem ao critério de um compromisso e que define a *Frenteira de Pareto* (ou *Conjunto de Pareto*). Qualquer uma delas poderá ser usada como a solução para o problema e caberá ao projetista escolher a que será implementada na prática.”

Conforme mencionado, a Otimização de Pareto pode fornecer diversas candidatas à solução ótima, mas a escolha da solução a ser implementada fica à mercê da subjetividade da equipe de projeto. Existem tentativas de definição de uma combinação entre as funções objetivo para a escolha da solução a ser implantada. Entretanto, a maioria destas termina por reduzir o problema de volta a um problema mono-objetivo. Ou seja, termina por definir arbitrariamente um critério que combina de maneira mais ou menos justa as funções objetivo.

Segundo Souza (2011): “-Isto nada mais é do que uma manifestação da limitação da razão humana de só saber comparar grandezas escalares”. Ou seja, a escolha de uma única solução da *Frenteira de Pareto* a ser implementada, passa pela incapacidade da razão humana de ordenar grandezas vetoriais, matriciais, tensoriais, etc. Se tomarmos como exemplo os seguintes vetores tridimensionais: $\mathbf{a} = [5 \ 6 \ 7]$, $\mathbf{b} = [4 \ 2 \ 4]$, $\mathbf{c} = [7 \ 8 \ 9]$ e $\mathbf{d} = [2 \ 4 \ 4]$. E se formos solicitados a ordená-los de forma decrescente rapidamente iniciamos com: $\mathbf{c} > \mathbf{a}$. Entretanto como ordenar \mathbf{b} e \mathbf{d} ? A norma desses vetores possui o mesmo valor ($|\mathbf{b}| = |\mathbf{d}|$), mas usar a norma é voltar

para ordenar uma grandeza escalar (mono-objetivo). E, apesar do fato de possuírem a mesma norma, definitivamente $\mathbf{b} \neq \mathbf{d}$. Entendo os vetores \mathbf{b} e \mathbf{d} como duas soluções diferentes na *Fronteira de Pareto*. Qual solução escolher, sem usar a subjetividade da equipe de projeto?

Uma das contribuições significantes para a solução é o Critério da Menor Perda é contribuição original de Rocco (2002) e encontra-se definido no mesmo trabalho. Esse critério que é usado para realizar o compromisso entre as funções objetivo, definindo a solução para o problema como a que produz a menor perda entre todos os objetivos envolvidos. Percebe-se que esta solução não se restringe à simples escolha arbitrária de uma das soluções na *Fronteira de Pareto* nem é uma simples redução do problema para uma versão mono-objetivo. Em vez disso, a escolha parte de soluções candidatas e seleciona aquela mais próxima do baricentro da figura geométrica formada pelas soluções candidatas não dominadas. O estudo detalhado do Critério da Menor Perda pode ser encontrado em Rocco et al. (2002), Rocco et al. (2003), Rocco et al. (2005), Rocco et al. (2005a) e Rocco et al. (2013).

A aplicação do Critério de Menor Perda na otimização de trajetórias espaciais pode ser encontrado em (VENDITTI, 2009; VENDITTI et al., 2010).

De acordo com Liu e Patton (1996) e Liu et al. (2002), no projeto de sistemas de controle é comum haver diversos objetivos a serem considerados. Os objetivos são as vezes conflitantes e não há como conseguir o melhor de todos simultaneamente. Então, tem que haver inevitavelmente um compromisso entre os objetivos considerados. Estas referências tratam somente de aspectos funcionais do controle multivariável e fazem uso de algoritmos genéticos para otimização, tais fatos, as tornam não aplicáveis totalmente ao propósito deste trabalho.

O uso de técnicas de otimização, tanto mono-objetivo como multiobjetivo, foram necessário durante a realização deste trabalho. Como descrito anteriormente, a equipe de projeto de sistema tem que lidar com diversos atributos e suas métricas. É natural o surgimento de problemas que se encaixem sob a forma de um problema de otimização multiobjetivo. O conhecimento de técnicas de otimização foram essencial para o surgimento da contribuição almejada no presente trabalho. Para isso necessita-se gerar alternativas por métodos como a OPN descrita a seguir.

2.6.2 Object Process Network (OPN)

A *Object Process Network* (OPN) é uma metalinguagem executável de domínio-neutro, projetada para representar, gerar e manipular modelos de simulação. (KOO, 2005). A metalinguagem OPN é utilizada para modelar o espaço de opções para as diversas arquiteturas possíveis que atendem a um problema. De acordo com Simona, Pinheiro e Loureiro (2007): -“Modelar o espaço de opções é diferente de modelar o sistema de interesse. Tradicionalmente as ferramentas de modelagem permitem especificar uma única solução, quando deveria se considerar um conjunto completo de arquiteturas factíveis. Em contrapartida, a OPN é capaz de auxiliar os arquitetos para avaliar essas possíveis configurações. Mas, como uma ferramenta de suporte a decisão, ela não oferece o poder descritivo que a OPM, SA e SysML possuem”.

De acordo com Bounova et al. (2005), a OPN representa o sistema em termos de uma rede de objetos e processos. Os objetos em um modelo OPN armazenam os estados intermediários da execução do modelo. Os processos em um modelo OPN armazenam as regras de transformação que mudam os estados de um modelo em execução. Como uma metalinguagem, a OPN permite que os usuários especifiquem o espaço de possibilidades dos modelos usando a sintaxe e a semântica formal da OPN. Algoritmos de simulação pragmáticos, como avaliadores de expressões simbólicas e de cálculo numérico, também são embarcados no ambiente de execução da OPN. Essas características permitem a realização de tarefas mecânicas de criação de modelos assim como as tarefas de computação de domínio-específico.

A Figura 2-12 apresenta um diagrama que ilustra a sintaxe da OPN (SIMMONS; KOO; CRAWLEY, 2005). Os processos são representados por elipses e os objetos por retângulos. As setas de conexão (relações) representam as pré e pós-condições. A estrela representa um *token*, que é uma estrutura de dados abstrata que representa um evento em execução e é o portador da evolução do modelo (CONNE; KOO, 2006). Os *tokens* são unidades de execução e armazenamento dos estados do modelo.

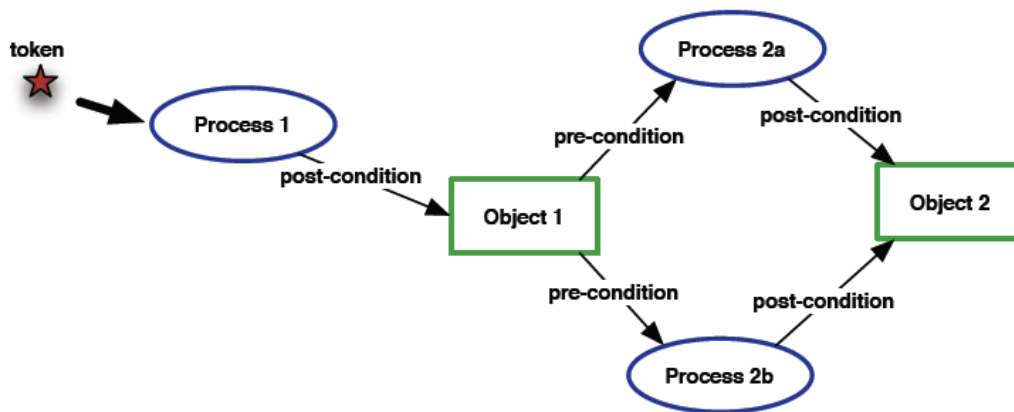


Figura 2-12: Exemplo de um diagrama que ilustra a sintaxe da OPN.
 Fonte: Simmons, Koo and Crawley (2005).

Em Simmons, Koo e Crawley (2005) há a descrição de um interessante uso da metalinguagem OPN. Essa metalinguagem foi utilizada para explorar o espaço de solução de arquiteturas de missões para exploração espacial. Foram geradas automaticamente diferentes arquiteturas físicas que atendem todas as etapas operacionais necessárias para a viagem espacial entre a Terra-Lua e Terra-Marte. Um modelo para simular as viagens dos veículos espaciais foi implementado em MATLAB que recebe como entrada as arquiteturas geradas. As arquiteturas foram avaliadas em termos de uma métrica de custo, a Massa Inicial para Órbita Terrestre Baixa (*Initial Mass to Low Earth Orbit - IMLEO*).

Apesar de o exemplo descrito não apresentar essa técnica como uma abordagem completa de otimização multiobjetivo, ela poderá perfeitamente compor uma parte importante desta abordagem, que é a geração automática de opções para a posterior seleção por um método de otimização. Dessa maneira a OPN poderá ser muito útil, automatizando o processo de criação de arquiteturas de sistemas de controle.

2.7 Geração Automática de Controladores

As principais fontes de referência para a elaboração e seleção de arquiteturas de controle procedem de um grupo de pesquisa de algoritmos genéticos centrados na

Universidade de Stanford, cujos trabalhos mais relacionados com o tema de interesse estão citados abaixo:

- a) (KOZA, KEANE, et al. 1999),
- b) (KOZA, KEANE e BENNETT III, et al. 1999),
- c) (KOZA, KEANE, YU, et al. 2000),
- d) (KOZA, KEANE, MYDLOWEC, et al. 2000),
- e) (KEANE, YU e KOZA, 2000),
- f) (KEANE, KOZA e STREETER 2002), e
- g) (KOZA, STREETER e KEANE 2003).

Todos os trabalhos são consistentes no método utilizado, e para compreensão destes trabalhos será descrito em mais detalhes o trabalho de Koza e Keane; et al. (1999). Esse propõe a geração automática de um controlador para um problema posto em Dorf e Bishop (1998), pagina 707, criando, a topologia e os valores dos parâmetros, de um controlador para a planta com a função de transferência:

$$G(s) = \frac{K}{(1+\tau s)^2}, \quad (2.9)$$

com os ganhos $K = 1$ e 2 , e a constante de tempo $\tau = 0.5$ e 1 , de forma que a planta alcance o valor de referência no mínimo tempo e que o *overshoot* a uma resposta a uma entrada degrau seja menor que 2%. A solução foi comparada com uma solução contida em Dorf e Bishop (1998). Além destas restrições o autor ainda adiciona outras restrições que também são satisfeitas pela solução (DORF; BISHOP, 1998). O modelo geral do sistema pode ser visto na Figura 2-13.

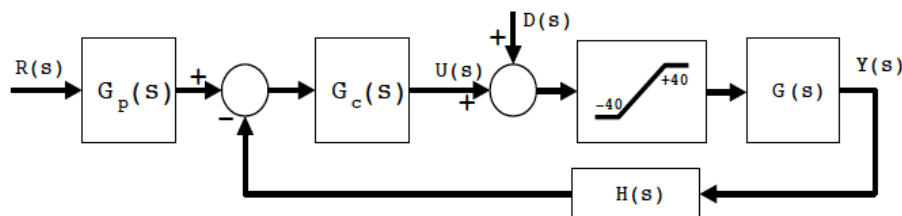


Figura 2-13: Modelo do sistema de controle proposto em Koza e Keane et al. (1999).

Para criar soluções o autor utiliza o método de algoritmos genéticos e, como passos iniciais, faz-se necessário criar um repertório de funções e um repertório de terminais (ex. REFERÊNCIA, SAIDA_DA_PLANTA, SAIDA_DO_CONTROLADOR, etc.).

O autor descreve sucintamente passos para a aplicação do método de algoritmos genéticos e apresenta os resultados descritos abaixo sendo G_p o pré-filtro e G_c o controlador para o modelo do sistema descrito anteriormente:

$$G_{p32}(s) = \frac{1(1+.126s)(1+.203s)}{(1+.038s)(1+.052s)(1+.084s)(1+.16s)(1+.17s)} \quad (2.10)$$

$$G_{c32}(s) = \frac{7487(1+.03851s)(1+.05146s)(1+.08375s)}{s} \quad (2.11)$$

Para haver uma referência, os resultados são comparados com a solução de Dorf e Bishop (1998):

$$G_{p-dorf}(s) = \frac{42.67}{42.67 + 11.38s + s^2} \quad (2.12)$$

$$G_{c-dorf}(s) = \frac{12(42.67 + 11.38s + s^2)}{s} \quad (2.13)$$

A estrutura do controlador obtida por meio de algoritmos genéticos pode ser vista na Figura 2-14.

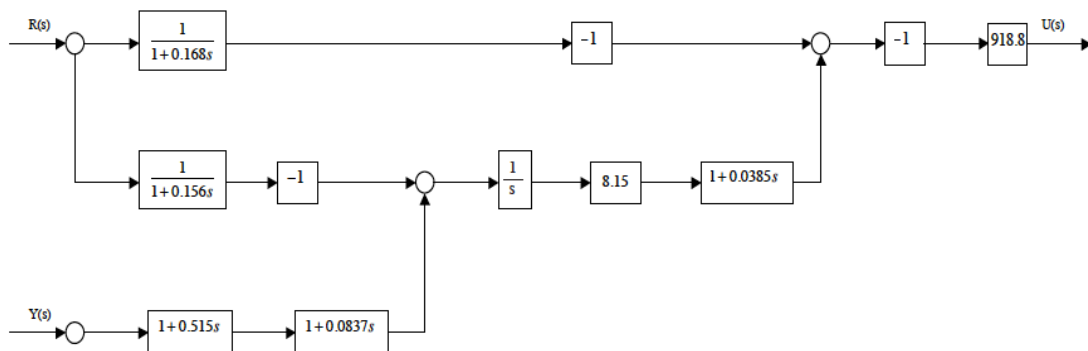


Figura 2-14: Estrutura do controlador obtida por Koza e Keane et al. (1999).

A Figura 2-15 apresenta as respostas no domínio de tempo a uma entrada degrau para as duas soluções Koza e Keane et al. (1999), curva indicada pela legenda *GP* e Dorf e Bishop (1998), curva indicada pela legenda *Textbook*.

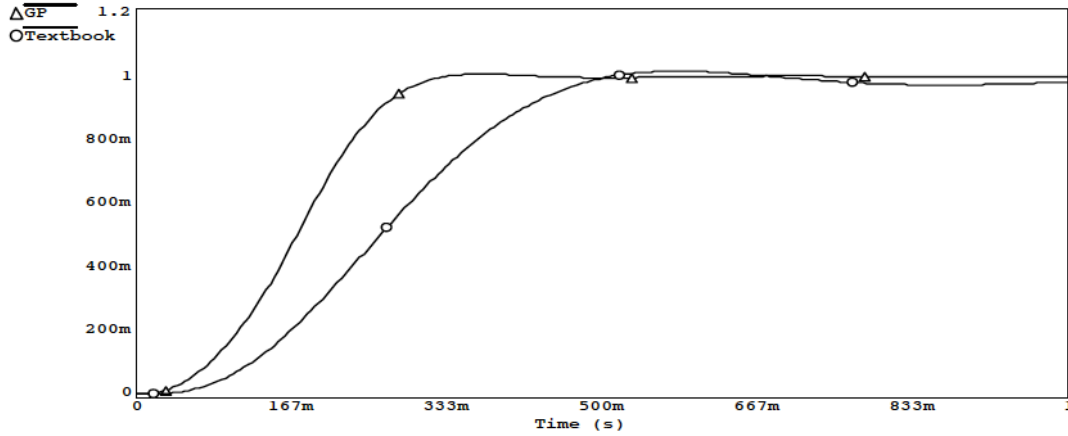


Figura 2-15: Comparação de resposta temporal a uma entrada degrau dos sistemas de controle
 Fonte: Koza e Keane et al. (1999) e Dorf e Bishop (1998).

Em Keane, Yu e Koza (2000), pode-se observar a maneira pela qual uma planta é modelada nos trabalhos do grupo de pesquisa. A Figura 2-16 contém o diagrama de blocos de uma planta e um controlador PID tradicional. Esse controlador é modelado na Figura 2-17 por meio de uma grafo do tipo árvore.

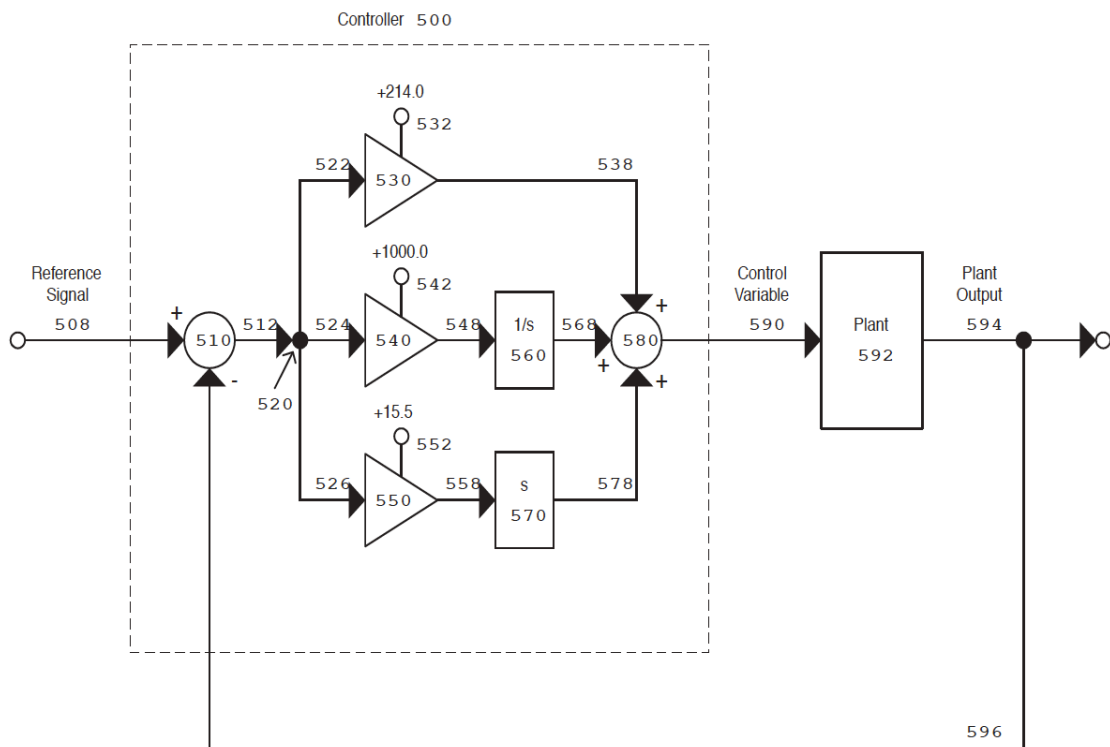


Figura 2-16: Diagrama de Blocos de uma Planta e seu Controlador PID.
 Fonte: Keane, Yu e Koza (2000).

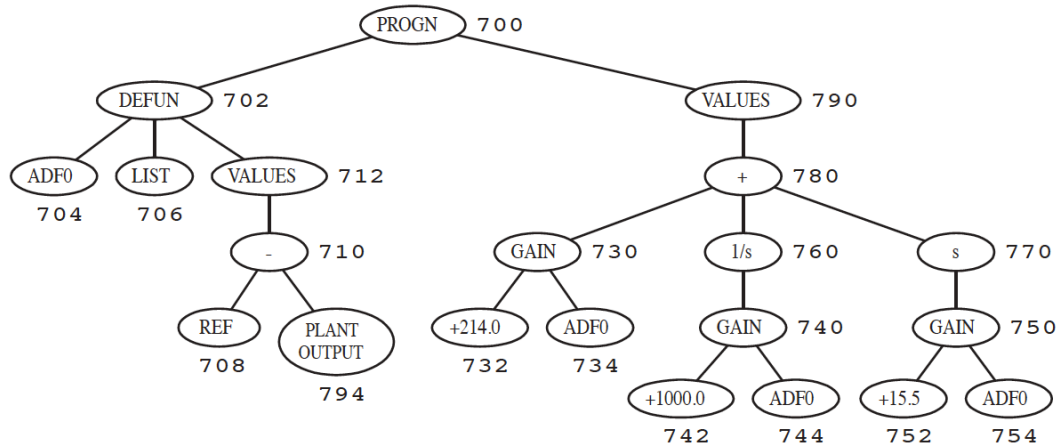


Figura 2-17: Árvore que modela a Planta e o Controlador PID da Figura 2-16.
 Fonte: Keane, Yu e Koza (2000).

Outros artigos deste grupo de pesquisa, citados anteriormente, produzem através do mesmo método, controladores para outras tipos plantas, produzindo resultados com estruturas diferentes para cada tipo de planta a ser controlada. Entretanto, em nenhum dos trabalhos revisados percebe-se: 1) a inclusão de métricas sistêmicas em suas análises; 2) o uso de métodos multiobjetivos; e 3) a análise da forma da solução, como faremos neste trabalho.

O valor deste tema de pesquisa é em parte demonstrado pelas patentes que aquele grupo de pesquisas originou. Aquele grupo de pesquisa patenteou tantos os métodos utilizados para gerar as soluções como as próprias soluções geradas. Como evidência podem ser citadas algumas patentes:

- a) (KOZA, BENNETT III e STIFFELMAN, 2013),
- b) (KEANE, KOZA et al 2005),
- c) (KOZA et al 2006), e
- d) (KOZA et al 2002).

3 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO

O trabalho proposto neste documento tem como objetivo a elaboração e seleção de arquiteturas de sistemas de controle por otimização multiobjetivo baseada em modelos e métricas sistêmicas. Para atingir este objetivo o trabalho adota as seguintes metas:

1. Revisão da literatura sobre métodos de elaboração e seleção de arquiteturas de sistemas, modelos e medidas de atributos de sistemas de controle, e otimização multiobjetivo (já feita no Capítulo 2);
2. Seleção dentre os métodos, arquiteturas, modelos e métricas existentes, aquelas que serão utilizadas neste trabalho.
3. Investigação e busca de melhorias nos métodos, arquiteturas, modelos e métricas, partindo dos conhecimentos existentes na literatura, como por exemplo:
 - a. Métodos multiobjetivo: Otimização de Pareto (1906), sua geração de opções com a *Object Process Network* (OPN) (KOO, 2005), sua escolha dentre soluções não dominadas apresentada em Rocco (2002) etc;
 - b. Arquiteturas: Plantas decompostas segundo os fatores normalizados por Bode, sensores, atuadores, controladores segundo as ações P, I, D, ou segundo a alocação de polos (OGATA, 1993), etc.
 - c. Modelos: Diagrama de Blocos (TAKAHASHI; RABINS; AUSLANDER, 1970), Diagramas de Componentes como o Diagrama de Blocos de Confiabilidade (DODSON; NOLAN, 1999), etc;
 - d. Métricas: Estabilidade (margem de ganho e margem de fase) (TAKAHASHI; RABINS; AUSLANDER, 1970), Desempenho (tempo de subida, tempo de acomodação, sobressinal), Custo e Confiabilidade (AMOROSO, 1999) etc.

O estudo é organizado por meio das três investigações, descritas a seguir.

3.1 Primeira Investigação (I1)

Nesta investigação, a pesquisa foca as arquiteturas de componentes na fase de implementação de um conjunto de sensores/atuadores. Usamos como contexto o sistema de Controle de Atitude e Manipulação de Dados (*Attitude Control and Data Handling – ACDH*) da Plataforma Multi-Missão (PMM), Figura 3-1. Neste sistema há a necessidade de medir diversas grandezas física. Tais medições são realizadas por transdutores cujos sinais medidos devem chegar até um controlador e, durante este trajeto, o sinal passa por diversos processamentos. É comum haver diversos sensores para uma mesma grandeza. Por exemplo, numa plataforma como a PMM deve haver dezenas de sensores de temperatura. A ligação entre os sensores e todos os componentes pelos quais os sinais captados devem transitar até à chegada a um Controlador formam uma arquitetura de sensores.

Após a contextualização anterior, o problema a ser estudado nesta investigação é: dada uma quantidade n de sensores, quais as arquiteturas que melhor cumprem a ligação entre a grandeza a ser medida e o controlador, levando em consideração a sua estrutura e suas propriedades? Outra questão de interesse é: oferecidas duas arquiteturas A1, A2, que atendam os requisitos de projeto, como escolher entre elas de maneira racional?

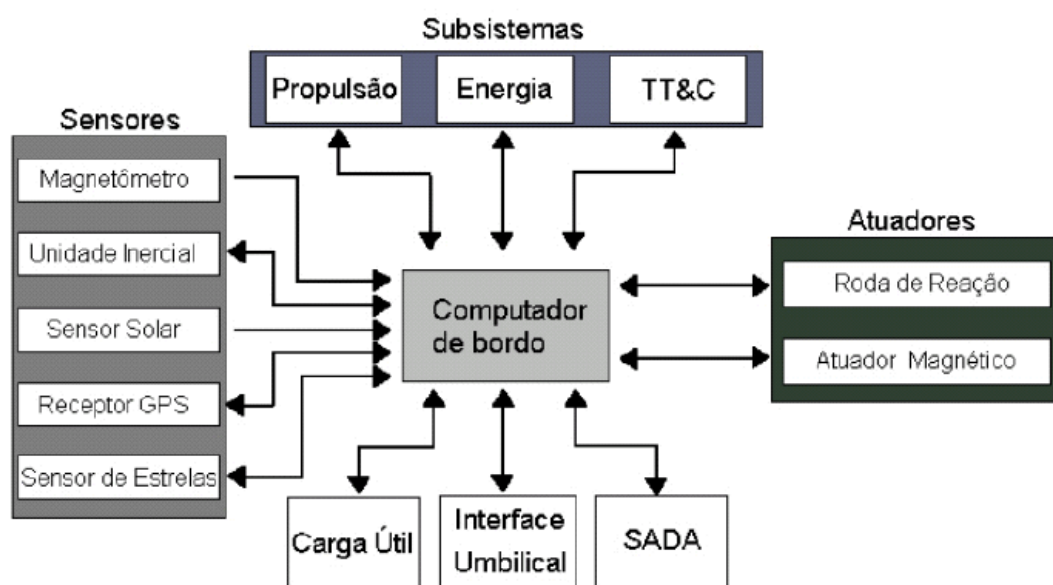


Figura 3-1: Sistema de controle da Investigação I1.

Fonte: INPE (2001).

Este estudo pode envolver uma infinidade de variáveis que podem ser incluídas nas análises. Entretanto, é sábio delimitar um escopo para tornar viável a busca por uma solução. Este escopo será delimitado na seção que descreve a o desenvolvimento da solução. Incluídos neste escopo, estão conhecimentos prévios do funcionamento de uma estrutura de sensores, i.e., quais blocos podem ser utilizados, como eles podem se conectar, critérios para finalização da busca, etc. Este conhecimento prévio torna o gerador mais eficiente uma vez que ele não fará simplesmente buscas aleatórias.

Para tanto, serão realizados os seguintes passos:

- 1) Gerar arquiteturas de sensores/atuadores;
- 2) Utilizar métodos multiobjetivo para buscar arquiteturas de sensores/atuadores candidatas a ótima, fazendo uso de pelo menos três métricas como função objetivo. Serão escolhidas métricas para os seguintes atributos: custo, confiabilidade e complexidade;
- 3) Selecionar uma arquitetura dentre as candidatas à ótima (soluções não dominadas).

3.2 Segunda Investigação (I2)

Nesta investigação, a pesquisa foca arquiteturas de componentes controladores. O estudo de sistemas de controle é um campo já muito explorado, bem formalizado e documentado. No qual, vários esquemas de controladores já são consagrados, como por exemplo os controladores PID ou LQR. Mesmo com todo este histórico ainda há necessidade para desenvolver ferramentas na busca de soluções para controladores, como visto em (KOZA; KEANE; et al., 1999), ainda mais quando se adiciona ao problema a estrutura física da implementação da solução, pois é na implementação que o controlador se materializa para realizar o seu papel.

Na teoria de controle, em geral projetos são feitos sem as preocupações de como se dará a sua implementação física e como os atributos físicos do sistema serão afetados. Uma arquitetura eficaz é capaz de trazer intrinsecamente em sua estrutura propriedades desejáveis ao sistema. Utilizando-se do mesmo problema posto em Dorf

e Bishop (1998), página 707, as arquiteturas geradas nesta investigação terão que controlar a planta da Equação 3.1.

$$G(s) = \frac{K}{(1+\tau s)^2} \quad (3.1)$$

Os ganhos $K = 1$, e a constante de tempo $\tau = 1$ s, de forma que a planta alcance o valor de referência no mínimo tempo e que o sobressinal (*overshoot*) a uma resposta a uma entrada degrau seja menor que 2%. A resposta em frequência do sistema em malha fechada deve permanecer abaixo de um filtro passa-baixa com frequência de corte a 100 Hz e queda de 40dB por década. No problema proposto, a saída do controlador é submetido a um bloco de saturação que limita o sinal de controle entre -40V e +40V (Figura 3-2).

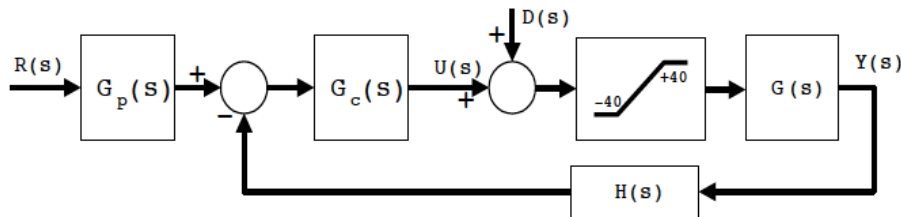


Figura 3-2: Sistema de controle da Investigação I2.
Fonte: Koza, Keane et al. (1999).

Dada a contextualização anterior, o problema a ser estudado nesta investigação é: a elaboração e seleção de arquiteturas de controladores por otimização multiobjetivo baseada em modelos e métricas sistêmicas.

Além da elaboração e seleção, pretende-se analisar como escolher entre elas de maneira racional entre duas arquiteturas A1, A2, que atendam os requisitos de projeto.

Assim com anteriormente também é necessário definir um escopo para a busca de soluções e esta incluirá diversos conhecimentos prévios do funcionamento de um controlador e sua planta.

O problema proposto nesta investigação possui maior complexidade que o da investigação I1. Com isso, as ferramentas desenvolvidas em I1 serão essenciais para solução deste problema, que também contará com novos desenvolvimentos. Alguns dos desafios desta investigação podem ser listados a seguir:

- Introdução de blocos com dinâmica contínua;
- Métricas de Atributos da Engenharia de Controle; e
- Dois universos de buscas, o universo das estruturas e o universo dos parâmetros da estrutura.

Para tanto, serão realizados os seguintes passos:

- 1) Gerar arquiteturas de controladores baseados em blocos P, I, D e suas combinações;
- 2) Adicionar outros tipos de blocos como polos e/ou zeros na geração de arquiteturas;
- 3) Utilizar métodos multiobjetivo para buscar arquiteturas de controladores candidatas a ótima, fazendo uso de pelo menos três métricas como função objetivo. Serão escolhidas métricas para os seguintes atributos: estrutura ou desempenho, custo e confiabilidade;
- 4) Selecionar uma arquitetura dentre as candidatas a ótima (soluções não dominadas).

3.3 Terceira Investigação (I3)

Assim como na investigação I2, esta etapa focará a pesquisa em arquiteturas de componentes controladores, sendo que a planta a ser controlada, descrita abaixo, é de segunda ordem com comportamento oscilatório. Utilizando-se de um problema posto em Dorf e Bishop (2008), as arquiteturas geradas nesta investigação terão que controlar o sistema ilustrado na Figura 3-3. Este é uma versão simplificada do sistema de controle de taxa de atitude de aeronave, voando a velocidade de Mach 4 e altitude de 100.000 pés. Os parâmetros desta planta são $K_1 = 1$, $\omega_a = 1.1594$, $\tau_a = 1$ e $\zeta = 0.69$.

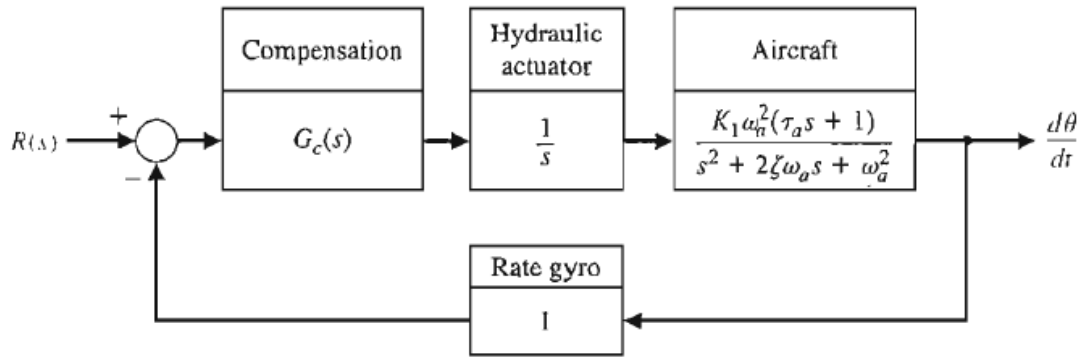


Figura 3-3: Sistema de controle da Investigação I3.
 Fonte: Dorf e Bishop (2008)

O problema pede para encontrar um compensador e que a sobressinal (*overshoot*) a uma resposta a uma entrada degrau seja menor que 5% e que o tempo de acomodação (*settling time*), com o critério de 2%, ocorra em menos de 5 s.

Dada a contextualização anterior, o problema a ser estudado nesta investigação é: a elaboração e seleção de arquiteturas de controladores por otimização multiobjetivo baseada em modelos e métricas sistêmicas.

Além da elaboração e seleção, pretende-se analisar como escolher entre elas de maneira racional entre duas arquiteturas A1, A2, que atendam os requisitos de projeto.

Esta investigação I3 utiliza o mesmo método elaborado para investigação I2.

Além da elaboração e seleção, pretende-se analisar quais os aspectos da forma das arquiteturas dos controladores que as tornam melhores. Outra questão pretende-se analisar é: oferecida duas arquiteturas A1, A2, que atendam os requisitos de projeto, como escolher entre elas de maneira racional?

Mesmo com os precedentes da investigação anterior, a investigação I3 ainda traz mais um desafio, uma vez que as características oscilatórias da planta demandam uma análise mais complexa dos resultados, que podem incluir mais um conjunto de métricas de desempenho.

Para tanto, serão realizados os seguintes passos:

- 1) Gerar arquiteturas de controladores baseados em blocos P, I, D e suas combinações;
- 2) Utilizar métodos multiobjetivo para buscar arquiteturas de controladores candidatas a ótima, fazendo uso de pelo menos três métricas como função objetivo. Serão escolhidas métricas para os seguintes atributos: estrutura ou desempenho, custo e confiabilidade;
- 3) Selecionar uma arquitetura dentre as candidatas à ótima (soluções não dominadas).

4 PRIMEIRA INVESTIGAÇÃO (I1): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE SENSORES E ATUADORES

Para este trabalho foi desenvolvida um ambiente para geração e avaliação de arquiteturas de sensores/atuadores. O desenvolvimento foi realizado em linguagem de programação do MATLAB®. O método implementado pelo ambiente prevê a geração de arquiteturas físicas, tanto de sensores como atuadores.

A investigação I1 especificada no Capítulo 0 estabelece que a avaliação de arquiteturas de sensores utilizará métricas para os seguintes atributos de engenharia de sistemas: custo, confiabilidade e complexidade.

4.1 Elementos de uma Arquitetura de Sensores e Atuadores

A primeira etapa para a geração da arquitetura é estabelecer quais são os **blocos** (ou componentes) que irão compô-la. Estes **blocos** são considerados as menores unidades de construção da arquitetura, ou seja, não serão divididos em sub-blocos. Além dos tipos de **blocos**, faz-se necessário estabelecer as propriedades de cada um destes **blocos** e a gama de opções disponíveis para cada propriedade. Com isto, tem-se o espaço de busca de **blocos** para o processo de geração da arquitetura que será descrito na próxima seção.

4.1.1 Premissas

Devido à indisponibilidade na literatura sobre o tema pesquisado, foi necessário estabelecer algumas premissas para a evolução do trabalho. Essas premissas são detalhadas a seguir.

4.1.1.1 Estrutura do Modelo por Meio de Árvores

As arquiteturas neste trabalho serão modeladas fazendo o uso da teoria dos **grafos**. Nos casos específicos das arquiteturas de sensores e controladores, estas serão

modeladas pelos grafos do tipo **árvore**. Neste modelo, os vértices representarão os **blocos** da estrutura e as arestas representarão conexões entre os blocos. O uso de uma **árvore** como modelo herda as seguintes propriedades: a não ocorrência de ciclos (ou realimentação para a teoria de controle), e que cada vértice somente possui um vértice pai.

4.1.1.2 Blocos como Componentes COTS

Cada **bloco** possui um número de **portas** de entrada igual ou superior às conexões de entrada. Este modelo permite a utilização de componentes com **portas** excedentes. Essa característica modela o fato que durante a construção de um sistema se utilizam os componentes que estão disponíveis no mercado e nem sempre estes estão perfeitamente aderentes às necessidades de um projeto. Por exemplo, caso os conversores Analógico para Digital (A/D) disponíveis para implementar o sistema somente possuam **portas** de entrada pares, e caso o projeto necessite de um **bloco** para converter três sinais analógicos em digitais, o menor **bloco** disponível para este fim possui quatro **portas** de entrada, e uma porta ficará inutilizada.

4.1.1.3 Portas e Blocos Multicanais

Julgou-se necessário utilizar mais um conceito além das **portas** de cada **bloco** para propagação dos sinais entre as folhas das árvores e sua raiz. Ao contemplar uma arquitetura de sensores modelada por uma árvore, é facilmente perceptível que cada elemento sensor (transdutor), representado por uma folha da árvore, deve transportar seu sinal até a raiz desta estrutura. Porém, o número de conexões entre os **blocos** da árvore diminui até chegar à raiz, no entanto, o número de sinais permanece inalterado. Com isso, conclui-se que cada conexão entre **blocos** pode transportar um ou mais sinais e que cada **bloco** deve ser capaz de processar um número de sinais igual ou superior ao seu número de conexões (**portas**). Foi denominada de **canal** de processamento, ou simplesmente **canal**, a capacidade de um **bloco** de processar um sinal. Portanto, o número de **canais** de um **bloco** deve ser igual ou superior à quantidade de sinais que este deve processar. Da mesma forma que o modelo permite a utilização de **portas** excedentes, também permite a utilização de **blocos** com **canais** excedentes.

4.1.1.4 Ausência de Superblocos

As arquiteturas serão compostas somente por **blocos** e não por superblocos. Superblocos são uma abstração para agrupar em uma única entidade um conjunto de blocos, mantendo as propriedades e as interfaces deste conjunto de blocos. Como o objetivo deste trabalho é estudar a estrutura dos sistemas, é mais útil analisar sua estrutura completa sem simplificações.

4.1.1.5 Geração Orientada e Randômica

A geração da arquitetura faz uso de recursos aleatórios presentes nas operações de evolução da árvore. As variáveis aleatórias são utilizadas na escolha de um componente a ser adicionado à estrutura e na confirmação de uma conexão viável entre dois componentes. A aleatoriedade é desejável para geração de alternativas baseadas nas mesmas leis de formação, as quais serão utilizadas para a análise e a comparação a posteriori.

O uso indisciplinado de variáveis aleatórias aumenta o espaço de busca e faz com que soluções não viáveis sejam criadas, o que torna a busca mais custosa e demorada. Neste trabalho o gerador incorpora conhecimentos prévios de como deve ser a estrutura da árvore o que limita o espaço de busca e aumenta a eficiência do gerador.

Podem-se explicitar pelo menos dois aspectos que guiam o gerador, são eles: a compatibilidade entre os blocos, e o encadeamento de processos necessários para conclusão da árvore. A compatibilidade determina se os tipos de dois componentes estão aptos à conexão.

O encadeamento de processos é necessário para transportar e transformar o sinal do elemento sensor até o ponto deste sinal pronto para ser utilizado pelo controlador. Neste trabalho, um sinal "pronto" significa que este foi transportado do ponto de medição até o local do controlador e que o sinal foi convertido de analógico para digital. Para tanto, o sinal deve passar por uma sequência lógica de processos realizados por cada componente. Esta sequência é em parte determinada por uma matriz de compatibilidade, descrita posteriormente, e de estados dos sinais armazenados nos atributos de cada componente da árvore.

4.1.2 Definição dos Blocos

O processo de geração de uma arquitetura que será descrito à frente necessita de um repositório de blocos que podem ser utilizados na construção desta arquitetura. O trabalho (KOZA; KEANE et al., 1999) também faz uso similar de um repertório de funções. A ideia de usar um repositório é a representação da política de fazer uso de componentes "off-the-shelf". A descrição dos tipos de blocos utilizados, seus propósitos e suas propriedades estão descritos na Tabela 4-1. Para este trabalho foi desenvolvido um gerador de blocos para popular o repositório utilizado neste trabalho. Esse gerador de blocos gerou um vetor para cada tipo de bloco com todas as possibilidades definidas no campo "Propriedades" da Tabela 4-1.

Tabela 4-1: Blocos utilizados nas arquiteturas de sensores/atuadores

Tipo	Símbolo	Propósito	Propriedades*
Transdutor	T	Converter energia de estímulos físicos em sinais elétricos.	<ul style="list-style-type: none"> portas de entrada = 0 canais = 1 confiabilidade= {automotiva, militar, espacial}
Condicionador de Sinais	SC	Ajustar sinais proveniente de transdutores.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Conversor Analógico/Digital	AD	Converter sinais analógicos em sinais digitais.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Transmissor Analógico	TxA	Transmitir sinais analógicos.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Receptor Analógico	RxA	Receber sinais enviados por transmissores analógicos.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Transmissor Digital	TxD	Transmitir sinais digitais.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Receptor Digital	RxD	Receber sinais enviados por transmissores digitais.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 0 < k < n\}$ canais = $\{2^k 0 < k < n\}$ confiabilidade= {automotiva, militar, espacial}
Multiplexador	M	Multiplexar os múltiplos canais de entrada em um canal de saída.	<ul style="list-style-type: none"> portas de entrada = $\{2^k 1 < k < n\}$ canais = $\{2^k 1 < k < n\}$ confiabilidade= {automotiva, militar, espacial}

Concentrador	C	Agrupar múltiplos canais em um conector	<ul style="list-style-type: none"> • portas de entrada = $\{2^k 1 < k < n\}$ • canais = $\{2^k 1 < k < n\}$ • confiabilidade = {automotiva, militar, espacial}
--------------	---	---	---

* Neste trabalho $n = 4$.

4.2 Modelo de uma árvore

Uma árvore é modelada neste trabalho por dois vetores distintos: um que contém a forma ou estrutura da árvore; e outro que contém os atributos de cada componente ou bloco da árvore.

No vetor de forma, F , cada elemento representa um vértice da árvore e o valor contido neste elemento é um ponteiro (aresta) para seu pai, sendo que um ponteiro para 0 significa a raiz da árvore. Por exemplo: sendo um $F = [0, 1, 2, 2, 4, 4, 4, 1, 8, 8, 10, 10]$, representa a árvore ilustrada na Figura 4-1.

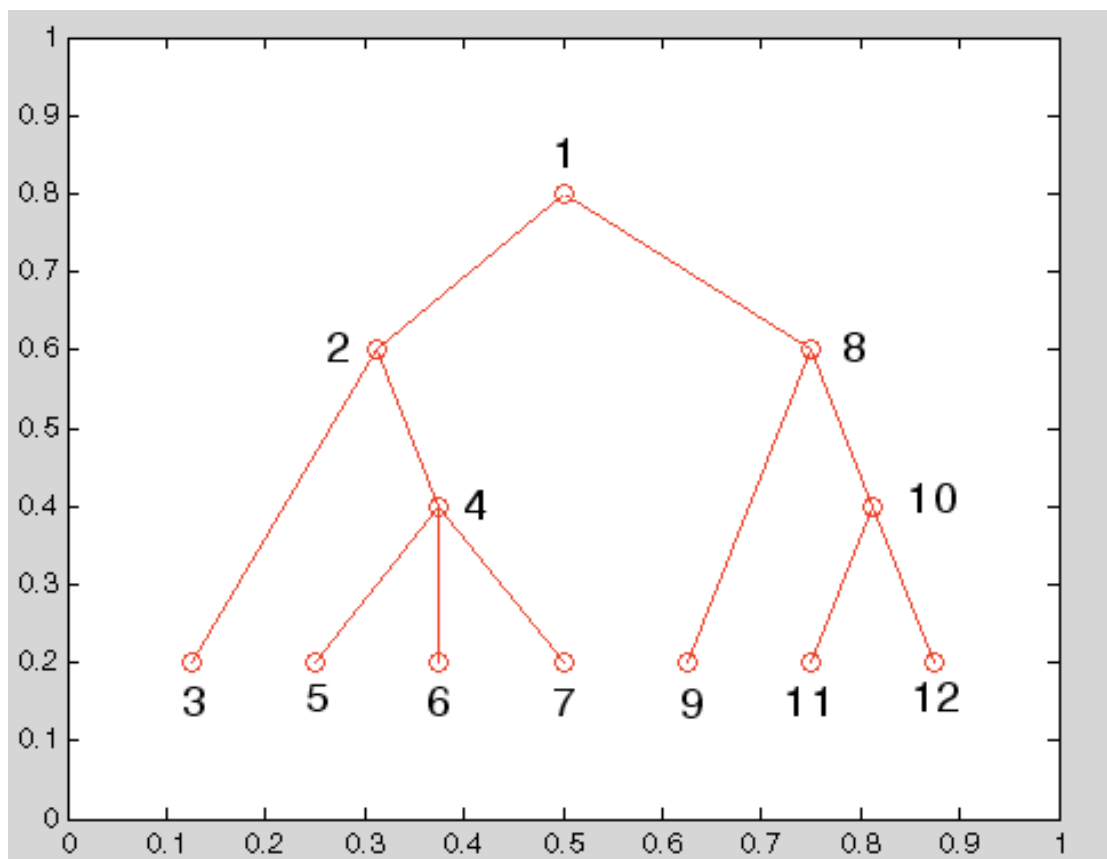


Figura 4-1: Árvore descrita pelo vetor $F = [0, 1, 2, 2, 4, 4, 4, 1, 8, 8, 10, 10]$.

O elementos do vetor de atributos, \mathbf{T} , representam exatamente os mesmos vértices do vetor \mathbf{V} , entretanto cada elemento armazena uma estrutura de dados com todos os atributos que determinam um bloco da arquitetura.

4.3 Matriz de Compatibilidade

Um importante elemento deste método é a Matriz de Compatibilidade \mathbf{M} , que é utilizada principalmente para verificar a compatibilidade entre dois tipos de componentes. Esta matriz é uma das principais influências que moldam a estrutura da árvore gerada, pois nela está incorporada grande parte do conhecimento sobre como deve ser a arquitetura. A matriz \mathbf{M} possui as seguintes utilidades:

- Verificar se um componente do tipo a é compatível com um do tipo b ;
- Dado um tipo de componente, selecionar todos os tipos compatíveis com esse tipo. Essa utilidade é usada para filtrar o espaço de busca quando se deseja adicionar mais um componente à árvore; e
- Determinar parcialmente o encadeamento de elementos na estrutura da árvore.

O sinal que é originado no transdutor deve passar por uma série de transformações, realizadas por processos em cada bloco, até chegar à raiz da árvore. Este encadeamento de processos é parcialmente determinado pela matriz \mathbf{M} .

A Matriz de Compatibilidade utilizada neste trabalho pode ser vista na Figura 4-2.

		Tipo do Componente Filho								
		T	SC	AD	Mux	C	TxA	RxA	TxD	RxD
Tipo do Componente Pai	T	0	0	0	0	0	0	0	0	0
	SC	1	0	0	0	0	0	0	0	0
	AD	0	1	0	1	1	0	1	0	0
	Mux	0	1	0	0	1	0	1	0	0
	C	0	0	1	1	1	0	1	0	1
	TxA	0	1	0	1	1	0	0	0	0
	RxA	0	0	0	0	0	1	0	0	0
	TxD	0	0	1	1	1	0	0	0	0
	RxD	0	0	0	0	0	0	0	1	0

Figura 4-2: Matriz de Compatibilidade M, utilizada neste trabalho.

O uso desta matriz pode ser extrapolado para incorporar outros valores para conexão entre os componentes, modelando um custo diferenciado entre as conexões de componentes ou outro aspecto causador de complexidade, como a diversidade de componentes.

4.4 Método de Geração de Arquiteturas

Após a definição dos elementos que compõem a arquitetura, esta seção descreve o método usado para gerar as arquiteturas de sensores. No método de geração proposto neste trabalho, as conexões tem início nas folhas da árvore e evoluem até sua raiz. A Figura 4-3 ilustra, com um fluxograma simplificado, o algoritmo de geração de árvores desenvolvido neste trabalho. O método descrito nesta seção é aplicado para gerar diversas alternativas de arquiteturas para serem analisadas posteriormente.

4.4.1 Condições Iniciais

Para ser possível iniciar a geração de uma arquitetura é necessário haver um repositório de blocos que podem ser utilizados na sua criação. Conforme descrito anteriormente na seção de Definição de Blocos, foi desenvolvido para este trabalho um gerador de blocos para compor um repositório em substituição a dados reais de um catálogo de componentes.

Além do repositório de blocos, o usuário deve escolher a quantidade de sensores a em uma arquitetura e a quantidade a ser gerada. Este dado é utilizado para inicializar o

vetor de atributos **T** do modelo da árvore com blocos de transdutores. Conforme estabelecido da Tabela 4-1, há três opções para este tipo de bloco, sendo essas dadas pelos três níveis de confiabilidade estabelecidos anteriormente. Então o vetor **T** é inicializado com a quantidade determinada de transdutores, aleatoriamente escolhidos entre as opções disponíveis. O vetor de forma **F** permanece inalterado e será preenchido à medida que a arquitetura evolui.

4.4.2 Evolução da Árvore

Para fazer evoluir a árvore, é feita uma varredura por todos os vértices da árvore e para cada vértice, uma ou mais operações são realizadas. As operações definidas neste trabalho são:

- 1) adição de uma aresta, ou seja, conexão entre dois blocos;
- 2) adição de um vértice, ou seja, adição de um bloco; e
- 3) poda da árvore, operação na qual vértices e suas arestas são eliminados.

Essas operações determinam a estrutura e são informações necessárias para determinar o valor de cada elemento de **F**. Por meio dessas operações se dá o crescimento e até a eventual redução da estrutura da árvore.

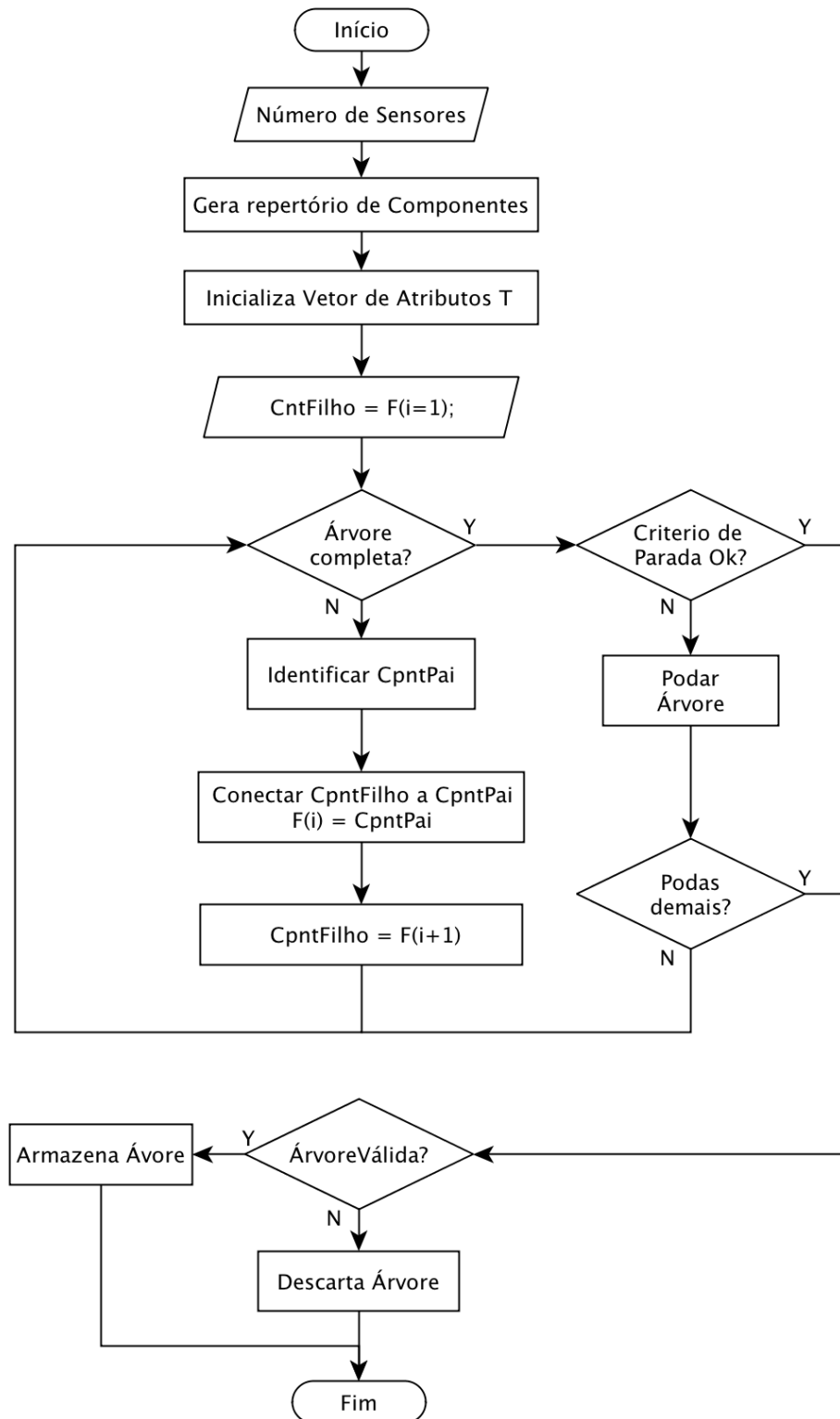


Figura 4-3: Fluxograma simplificado do gerador de arquiteturas de sensores/atuadores.

A varredura dos vértices está representada pelo laço principal do fluxograma da Figura 4-3.

Outro elemento importante para a evolução da árvore é a Matriz de Compatibilidade **M**, descrita anteriormente, que determina a compatibilidade entre dois tipos de componentes e determina parcialmente a sequência de encadeamento de componentes. Esta matriz é um dos principais responsável pela obtenção de árvores viáveis.

4.4.2.1 Identificação do Componente Pai

O passo de identificação de um componente envolve duas alternativas: um componente pai já existente nos vetores **F** e **T**, ou a necessidade de criar um novo componente. Neste passo é que se verifica a necessidade da operação de adição de um vértice. A identificação de um componente pai é feita pelos seguintes passos:

- A varredura seleciona um componente filho;
- Verifica-se no vetor de atributos **T**, qual o tipo desse componente filho;
- Verifica-se se há um componente pai disponível para conexão:
 - havendo um componente pai disponível, verifica-se através da matriz **M** se estes tipos são compatíveis e se o pai ainda possui disponibilidade de portas para conexão; mesmo satisfazendo a todos os critérios de disponibilidade, verifica-se se conexão deve ser realizada através de um sorteio no qual a chance de conexão é proporcional à quantidade de portas disponíveis.
 - não havendo um componente pai disponível, é realizada a operação de criar um novo componente, e este é selecionado aleatoriamente entre as opções viáveis para aquele específico componente filho, para tanto, o espaço de busca é filtrada pela matriz **M**.
 - se for necessário criar um novo componente, então um novo elemento deve ser criado nos vetores **F** e **T**, sendo que esse novo elemento de **T** já será preenchido com todas as informações deste novo componente.
- Ao fim deste passo determina-se quem é o componente pai para o componente filho selecionado.

4.4.2.2 Conectar Componente Filho ao Componente Pai

O passo de conexão entre o componente filho e seu pai é realizado com o resultado do passo de Identificação do Componente Pai, o mesmo componente filho selecionado anteriormente e o componente pai determinado naquele passo.

Esta conexão é feita alimentando o elemento correspondente ao componente filho do vetor **F** com o índice do componente pai.

Diferente do passo anterior onde a operação de adição de um vértice pode ou não acontecer, nesse passo a operação de adição de uma nova aresta sempre acontecerá. E isto mesmo com o último componente que para fins computacionais realiza uma conexão com o índice zero, representando a raiz da árvore.

4.4.3 Operação de poda de árvore

A operação de poda da árvore é eliminar uma quantidade n de vértices e todas as suas arestas. Para tanto, os n últimos elementos de dois vetores **F** e **T** são eliminados e o processo de evolução volta a entrar em ação. Esta operação é utilizada para tentar recuperar uma árvore que foi concluída, porém não atingiu outros critérios para a validade da árvore.

4.4.4 Critério de parada

Concluir da varredura dos componentes filhos não significa necessariamente o encerramento da busca pela estrutura. Sendo o processo de busca aleatório, mesmo que seja orientado, ainda pode gerar arquiteturas que não atendem aos critérios exigidos pelo controlador, que são o sinal ser transportado do local de medição até o local do controlador e que este sinal seja no formato digital.

Então, após a conclusão da varredura, verifica-se se o sinal está pronto para ser utilizado. Caso "sim", a árvore é dita válida e a busca é encerrada. Caso "não", comanda-se uma operação de poda, como descrito anteriormente. A recorrência seguida de um número de podas da árvore também encerra a busca por solução e neste caso a árvore é tida como inválida.

4.5 Método de Avaliação e Seleção de Arquiteturas

Da maneira como a ambiente foi implementada, é possível adicionar facilmente novas métricas para análise das alternativas elaboradas. A seleção racional de uma destas arquiteturas como solução não é uma tarefa simples ou elementar uma vez que se lida com diversos atributos simultaneamente. É conhecido o fato que não é possível conseguir o melhor em todos os atributos simultaneamente, tem que haver um compromisso entre eles. É por isto que a seleção será feita por meio de métodos multiobjetivos descritos a seguir.

Mesmo sabendo que métodos multiobjetivo serão aplicados ainda é útil realizar uma breve análise mono-objetivo para cada um dos atributos em questão: custo, complexidade e confiabilidade.

4.5.1 Métricas

As arquiteturas físicas de sensores e atuadores estudadas neste trabalho devem permitir a definição de valores numéricos referentes as métricas propostas na Primeira Investigação (I1), são elas: custo (P), confiabilidade (R) e complexidade (C). Essas, representam as funções objetivos que se quer otimizar.

4.5.1.1 Custo (P)

O custo (P) de um bloco é definido pelo produto de três valores atribuídos aos seguintes atributos do bloco: seu número de canais, sua categoria, e seu nível de confiabilidade. Ou seja, o custo (P) de um bloco é dado por:

$$P = \text{Custo de Categoria} \cdot \text{Custo de Confiabilidade} \cdot \text{Número de Canais} \quad (4.1)$$

O custo de uma arquitetura é dado pelo somatório do custo de seus blocos.

Os valores da Tabela 4-2 foram arbitrados para as categorias dos componentes e os valores da Tabela 4-3 foram arbitrados para os níveis de confiabilidade.

Tabela 4-2: Valores de custos utilizados para cada categoria de bloco.

Categoria	Custo
T	1
SC	2
C	3
M	4
TxA	5
RxA	5
TxD	6
RxD	6
AD	7

Tabela 4-3: Valores de custos utilizados para cada nível de confiabilidade.

Nível de Confiabilidade	Custo
Automotivo	2
Militar	3
Espacial	4

Esta maneira de calcular o custo é sugerida neste trabalho para substituir a ausência de um catálogo de preços. O custo completo da arquitetura não é influenciado pela forma da árvore, somente pelos atributos de seus componentes e pela quantidade de componentes.

4.5.1.2 Confiabilidade (R)

A Confiabilidade (R) de cada bloco é dada por um valor atribuído ao seu nível de confiabilidade. Os valores atribuídos a cada confiabilidade são ilustrados e não pretendem ser próximos aos valores médios reais destes componentes. Para este trabalho os seguintes valores foram arbitrados para cada nível de confiabilidade:

Tabela 4-4: Níveis de confiabilidade e respectivos valores utilizado.

Nível de Confiabilidade	Confiabilidade
Automotivo	$1 - 10^{-2}$
Militar	$1 - 10^{-3}$
Espacial	$1 - 10^{-4}$

A confiabilidade da arquitetura é dada por um cálculo recursivo realizado para um aglomerado de um bloco pai (B_{Pai}) e seus blocos filhos (B_{Filho}). A confiabilidade (R) de um aglomerado é por:

$$R_{Aglomerado} = R(B_{Pai}) \cdot \left(1 - \prod_{Filho} (1 - R(B_{Filho})) \right) \quad (4.2)$$

Este cálculo é recursivamente aplicado a cada bloco pai até o bloco raiz da arquitetura.

Esta maneira é a maneira tradicional de calcular confiabilidade de uma estrutura quando se leva em consideração os arranjos de componentes em série ou em paralelo. Optou-se por esta maneira devido ao resultado final ser influenciado pela forma da árvore. Entretanto, é possível calcular a confiabilidade da estrutura como se todos os componentes estivessem arranjados em série, representando o fato de que se um componente da estrutura falhar, então toda a estrutura falhará.

4.5.1.3 Complexidade (C)

A complexidade (C) de cada bloco é medida pelo produto entre a quantidade de portas e a quantidade de canais de cada bloco (B). A exceção é para os transdutores que não possuem portas de entrada: a complexidade para esse bloco é suposta igual 1.

$$C(B) = \text{Portas de } B \cdot \text{Canais de } B \quad (4.3)$$

A complexidade da arquitetura é dada por um cálculo recursivo realizado para um aglomerado de um bloco pai (B_{Pai}) e seus blocos filhos (B_{Filho}).

A complexidade de um aglomerado é expressa por:

$$C_{Aglomerado} = \sum_{Filho} C(B_{Pai}) \cdot C(B_{Filho}) \quad (4.4)$$

Este cálculo é recursivamente aplicado desde cada bloco pai até o bloco raiz da arquitetura. A medida da complexidade deve fazer contraposição desejável à medida da confiabilidade.

Esta maneira de calcular complexidade é inspirada na métrica de complexidade apresentada em Phukan, Kalava e Prabhu (2005):

$$C = \text{Complexidade intracomponente}(\text{Complexidade intercomponente})^2 \quad (4.5)$$

Neste trabalho a complexidade de cada bloco (B) é a complexidade intracomponente e o somatório entre o Pai e seus Filhos é a complexidade intercomponentes. Entretanto, no trabalho de Phukan, Kalava e Prabhu (2005) os autores calculam a complexidade local de cada componente independentemente, buscando identificar pontos de estresse de complexidade, desta maneira a estrutura do modelo é somente parcialmente considerada.

Para este trabalho optou-se por fazer um cálculo de complexidade aplicado de maneira cumulativa, similar ao cálculo da confiabilidade que, ao final, produz um valor totalitário representativo da forma como os componentes estão arranjados.

4.5.2 Análise Mono-Objetivo

Essa análise ajuda a compreender de maneira mais isolada a influência da forma da estrutura nos atributos escolhidos. Serão calculadas as métricas para cada uma das árvores geradas e os valores serão armazenados em três vetores distintos: vetor de custo \mathbf{P} , vetor de complexidade \mathbf{C} e vetor de confiabilidade \mathbf{R} . Cada um desses vetores será ordenado de maneira a permitir a rastreabilidade até o índice da árvore que ele representa.

Com os dados calculados em uma análise mono-objetivo é simples determinar quais os melhores resultados individuais para cada métrica, aqueles que: minimizam o custo, minimizam a complexidade e maximizam a confiabilidade. Porém o principal valor desta análise é entender como a estrutura da árvore afeta cada uma das métricas e esta análise é relativa e não absoluta. Na busca pelo entendimento deve-se observar os resultados melhores e os piores. Os resultados serão apresentados no Capítulo 0.

4.5.3 Análise Multiobjetivo

Nesta análise multiobjetivo as métricas serão normalizadas, ou seja:

$$P_n = P / (\text{Máximo de } P) \quad (4.6)$$

$$C_n = C / (\text{Máximo de } C) \quad (4.7)$$

$$R_n = R / (\text{Máximo de } R) \quad (4.8)$$

Dentre todas essas soluções, buscar-se-á aquelas **soluções não dominadas** que compõem a **Fronteira de Pareto**. Estas soluções formam um conjunto de soluções candidatas a ser a solução ótima, pois fazem um compromisso entre as métricas estabelecidas, ou seja, não é mais possível melhorar uma métrica sem prejudicar pelo menos outra.

Apesar das soluções da Fronteira de Pareto serem um compromisso entre as métricas escolhidas, essa fronteira ainda não resolve o problema por completo, pois gera um conjunto de opções e não uma solução. Supondo que o resultado do trabalho é escolher qual solução será implementada, ainda falta determinar essa solução. Para fazer esta escolha de maneira imparcial optou-se por utilizar o método da Menor Perda estabelecido em Rocco (2002), o qual consiste em encontrar o baricentro (neste caso igual ao centro de massa) entre as soluções na Fronteira de Pareto e escolher aquela solução não dominada mais próxima deste centro. Os resultados serão apresentados no Capítulo 0.

5 SEGUNDA INVESTIGAÇÃO (I2): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE CONTROLADORES

Para este trabalho foi desenvolvida um ambiente para geração e avaliação de arquiteturas de controladores. O desenvolvimento foi realizado em linguagem de programação do MATLAB®.

A segunda investigação (I2), especificada no Capítulo 0, estabelece que a avaliação de arquiteturas de controladores combinará atributos da Engenharia de Controle com a Engenharia de Sistemas. Serão utilizadas as seguintes métricas: tempo de subida, custo (da confiabilidade etc.) e complexidade.

5.1 Elementos de uma Arquitetura de Controladores

Assim como na arquitetura de sensores/atuadores, a primeira etapa para a geração da arquitetura é estabelecer quais são os **blocos** (ou componentes) que irão compô-la. Estes blocos são considerados as menores unidades de construção da arquitetura, ou seja, não serão divididos em sub-blocos. Nesta investigação cada bloco possui uma função de transferência. Os **blocos** se combinam segundo a arquitetura gerada de forma a compor a função de transferência da arquitetura gerada.

5.1.1 Premissas

Foi necessário estabelecer algumas premissas para a evolução do trabalho. Essas premissas são detalhadas a seguir.

5.1.1.1 Controlador SISO

Será utilizado um sistema **SISO** (*Single Input and Single Output*) na Investigação I2. Este fato se reflete nas propriedades de cada bloco, ou seja, cada bloco também é um bloco SISO, possuindo somente uma **porta** de entrada, uma **porta** de saída e um único **canal** de processamento. O conceito de múltiplos **canais** utilizados na

arquitetura de sensores se aplica a sistemas **MIMO** (*Multiple Inputs and Multiple Outputs*), que poderão ser objeto de estudos futuros.

5.1.1.2 Estrutura de Controladores

O modelo do arranjo dos blocos de controle utilizará o tradicional diagrama de blocos da Engenharia de Controle, herdando todo o conhecimento embutido nesta forma de modelagem. A estrutura de uma **árvore** não é adequada para modelar uma estrutura de um controlador por não permitir a ocorrência de ciclos. Portanto, a estrutura utilizada nesta investigação se assemelha à forma mais abrangente de **grafo**, permitindo a manifestação de ciclos e oferecendo uma flexibilidade maior para o modelo. Da mesma forma como feito anteriormente, os vértices representarão os **blocos** da estrutura e as arestas representarão as conexões entre os blocos.

Seguindo o modelo SISO, um ciclo somente pode ocorrer através de blocos de operações algébricas, neste trabalho utilizaremos somente o bloco somador. Este irá realizar esta operação em múltiplos sinais e transformá-los em um único sinal antes da entrada de um bloco SISO. A operação de soma ocorrerá de forma transparente no arranjo dos blocos, ou seja, o somador não faz parte do repertório de blocos ele será uma consequência do arranjo dos blocos.

5.1.1.3 Ausência de Superblocos

Ainda para os controladores, as arquiteturas serão compostas somente por **blocos** e não por superblocos. Superblocos são uma abstração para agrupar em uma única entidade um conjunto de blocos, mantendo as propriedades e as interfaces deste conjunto de blocos. Como o objetivo deste trabalho é estudar a estrutura dos sistemas, é mais útil analisar sua estrutura completa sem simplificações.

5.1.1.4 Geração Orientada e Randômica

Da mesma forma como foi feito na Investigação I1, a geração da arquitetura nesta Investigação (I2) faz uso de recursos aleatórios para as operações de evolução da estrutura do controlador. As variáveis aleatórias ainda são utilizadas na escolha de um componente a ser adicionado à estrutura e na confirmação de uma conexão viável

entre dois componentes. O gerador incorpora conhecimentos prévios de como deve ser a estrutura do controlador, o que limita o espaço de busca e aumenta a eficiência do gerador.

5.1.2 Definição dos Blocos

A definição de um repertório de blocos, como feito no trabalho de Koza e Keane et al. (1999) e na Investigação II, é essencial para a geração da arquitetura. Cada bloco possui uma quantidade de parâmetros que podem ser modificados na etapa de ajuste da arquitetura, para que o controlador resultante cumpra seu objetivo. O método aqui descrito foi concebido para utilizar todos os tipos de blocos descritos na Tabela 5-1, porém não se limita a eles.

Tabela 5-1: Blocos utilizados nas arquiteturas de controladores

Tipo	Símbolo	Propriedades
Ganho	G	<ul style="list-style-type: none"> • número de parâmetros=1 • confiabilidade = {automotiva, militar, espacial}
Integrador	I	<ul style="list-style-type: none"> • número de parâmetros=0 • confiabilidade = {automotiva, militar, espacial}
Derivador	D	<ul style="list-style-type: none"> • número de parâmetros=0 • confiabilidade = {automotiva, militar, espacial}
Polo Real	P	<ul style="list-style-type: none"> • número de parâmetros=1 • confiabilidade = {automotiva, militar, espacial}
Zero Real	Z	<ul style="list-style-type: none"> • número de parâmetros=1 • confiabilidade = {automotiva, militar, espacial}
Polo Complexo	Pc	<ul style="list-style-type: none"> • número de parâmetros=2 • confiabilidade = {automotiva, militar, espacial}
Zero Complexo	Zc	<ul style="list-style-type: none"> • número de parâmetros=2 • confiabilidade = {automotiva, militar, espacial}

A adição de qualquer bloco com uma função de transferência pode ser facilmente realizada. Da mesma maneira, pode-se escolher gerar arquiteturas que utilizem somente um subconjunto do repertório criado. Essa escolha é realizada para a Matriz de Compatibilidade \mathbf{M} , já explicada no Capítulo 4.

5.2 Modelo de um Controlador

A árvore da arquitetura de sensores foi modelada por dois vetores: um que armazena os atributos de cada bloco, chamado de vetor de atributos \mathbf{T} ; e outro chamado de vetor de forma \mathbf{F} , que contém a ligação entre os blocos, determinando o arranjo da árvore.

Para a arquitetura dos controladores o vetor de atributos \mathbf{T} permanece o mesmo, porém o vetor de forma \mathbf{F} é substituído por uma Matriz de Incidência \mathbf{M}_I . Os índices das linhas de \mathbf{M}_I representam o bloco de origem da conexão e os índices da coluna representam os blocos de destino da conexão. Os elementos da matriz são binários: 0 indica a ausência e 1 indica a presença de uma conexão.

5.3 Matriz de Compatibilidade

A mesma Matriz de Compatibilidade \mathbf{M} utilizada na Investigação II também está presente na arquitetura de controladores. Esta matriz é umas das principais influências que moldam o arranjo da estrutura gerada, pois nela está incorporada grande parte do conhecimento sobre como deve ser a arquitetura. A matriz \mathbf{M} possui as seguintes utilidades:

- Verificar se um componente do tipo a é compatível com um do tipo b ;
- Dado um tipo de componente, selecionar todos os tipos compatíveis com esse tipo. Essa utilidade é usada para filtrar o espaço de busca quando se deseja adicionar mais um componente à árvore.

5.4 Método de Geração de Arquiteturas

Após a definição dos elementos que compõem a arquitetura, esta seção descreve o método usado para gerar as arquiteturas de controladores. A Figura 5-1 ilustra, com

um fluxograma simplificado, o algoritmo de geração de arquiteturas de controladores desenvolvido neste trabalho. O método descrito nesta seção é aplicado para gerar diversas alternativas de arquiteturas para serem analisadas posteriormente.

Uma importante diferença entre a geração das arquiteturas de sensores e de controladores é o nível de prontidão das estruturas. Isto quer dizer que no caso dos sensores a estrutura precisa evoluir até fazer o sinal captado nos transdutores chegar até a raiz da árvore para ser considerada pronta. No caso das estruturas de controladores, desde o controlador mais básico, somente um bloco de ganho, este já pode ser útil. Com isso, a cada passo da evolução da arquitetura pode-se obter um controlador válido.

5.4.1 Condições Iniciais

Para ser possível iniciar a geração de uma arquitetura, é necessário haver um repositório de blocos que podem ser utilizados na sua criação. Além do repositório de blocos, o usuário deve escolher a quantidade máxima de blocos em um controlador assim como a quantidade de controladores a serem gerados.

O primeiro componente do vetor **T** será um ganho. Todos os blocos que necessitam de parâmetros, ao serem criados, usam o valor 1 para todos seus parâmetros.

5.4.2 Evolução da Arquitetura

Para fazer evoluir a arquitetura, é feita uma varredura por todos os blocos da estrutura; e, para cada bloco, uma das duas operações é escolhida aleatoriamente:

- 1) adição de um bloco em série;
- 2) criação de um novo ramo em paralelo, iniciado por um bloco de ganho.

Essas duas simples operações são responsáveis pela geração de todas as estruturas apresentadas neste trabalho. Diversas outras maneiras de crescimento da estrutura podem ser objetos de estudos futuros.

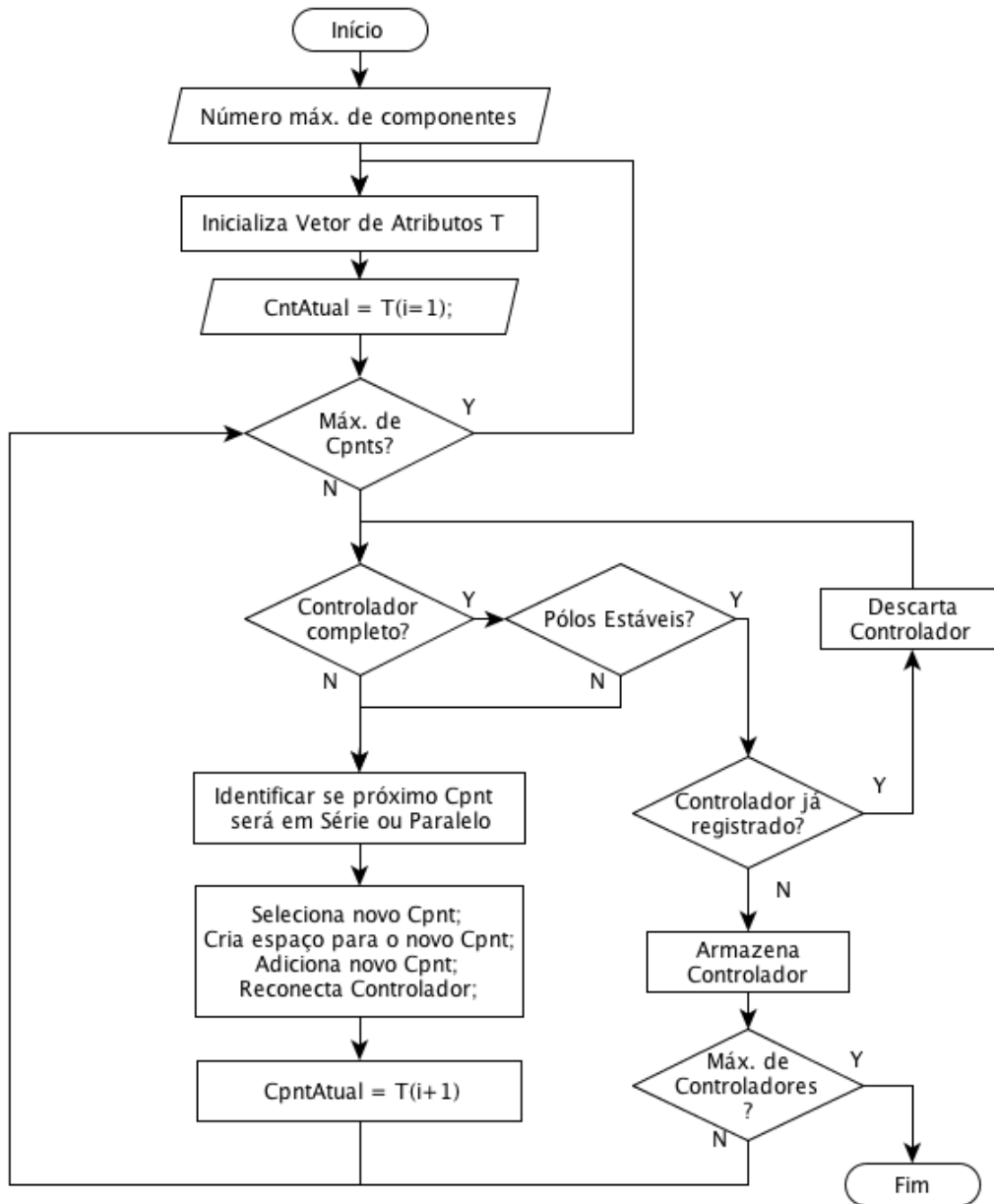


Figura 5-1: Fluxograma simplificado do gerador de arquiteturas de controladores.

As operações são determinantes na definição do arranjo dos elementos do controlador. Com essas operações nas estruturas resultantes os blocos ficarão dispostos em um arranjo matricial, onde cada coluna representará um componente em série e cada linha será um ramo em paralelo que, ao final, será somado para compor a saída do controlador.

5.4.3 Critério de Parada

Seguindo o fluxograma da Figura 5-1 percebemos que o critério de parada deste método de geração é a quantidade de controladores válidos.

5.5 Ajustes dos Parâmetros do Controlador

Cada controlador ainda necessita ter seus ganhos ajustados para atingir os objetivos desejados para aquele controlador. Dado a diversidade de controladores e o automatismo necessário, serão utilizados métodos numéricos de busca para encontrar os ganhos de cada estrutura.

Os métodos de busca precisam ser configurados com uma função objetivo e com as restrições de acordo com cada problema a ser solucionado. Foram utilizadas duas funções de busca do MATLAB® para ajustar os parâmetros dos controladores, são eles: *simulannealbnd* e *fminsearch*.

5.5.1 Restrições Consideradas

As restrições para o método de busca foram incluídas como penalidades na função objetivo, além da restrição imposta que o sobressinal não fosse superior a 2%, a saída do controlador deveria se submeter a um saturador de -40V a +40V. Ao invés de incluir um bloco de saturação escolheu-se penalizar também esse sinal quando este sair da faixa de operação determinada. Outras restrições, mesmo que não impostas pelo problema também foram consideradas, como valor de acomodação mínimo e máximo para evitar soluções com grande erro de *offset*.

5.6 Método de Avaliação e Seleção de Arquiteturas

Da maneira como o ambiente foi implementada, é possível adicionar facilmente novas métricas para análise das alternativas elaboradas. A seleção racional de uma destas arquiteturas como solução não é uma tarefa simples ou elementar uma vez que se lida com diversos atributos simultaneamente. É conhecido o fato de não ser possível conseguir o melhor em todos os atributos simultaneamente, tem que haver um

compromisso entre eles. É por isto que a seleção será feita por meio de métodos multiobjetivos descritos a seguir.

Mesmo sabendo que métodos multiobjetivos serão aplicados, ainda é útil realizar uma breve análise mono-objetivo para cada um dos atributos em questão: custo, complexidade e tempo de subida.

5.6.1 Métricas

As arquiteturas físicas de controladores estudadas neste trabalho devem admitir as métricas propostas na Investigação I2. São elas: custo (P), complexidade (C) e tempo de subida (Ts).

O uso da métrica de Confiabilidade (R) utilizada na Investigação I1 conseguia ser afetada pelo arranjo dos blocos daquela estrutura, representando caminhos paralelos como caminhos redundantes. Na estrutura de um controlador SISO não se pode considerar que haja caminhos redundantes uma vez que, se um dos blocos de um dos caminhos do controlador falhar, todo o controlador falhará, pois o controlador resultante não será mais o mesmo e não há garantia que este continuaria com o desempenho desejado ou até mesmo que continuaria a exercer a função necessária. Assim sendo, a confiabilidade deveria ser modelada com todos os componentes em série, fazendo com que a confiabilidade somente fosse afetada pelo valor individual da confiabilidade de cada bloco e não mais pelo seu arranjo.

Mesmo assim, a Confiabilidade (R) está sendo considerada indiretamente através da métrica de Custo (P), descrita a seguir.

5.6.1.1 Custo (P)

O custo (P) do controlador nesta investigação irá se basear na confiabilidade de cada bloco. Mesmo que a confiabilidade (R) não seja explicitamente considerada na avaliação multiobjetivo, este atributo será considerado indiretamente por meio do custo.

A Confiabilidade (R) de cada bloco é dada por um valor atribuído ao seu nível de confiabilidade. Os valores atribuídos a cada confiabilidade são ilustrados e não

pretendem ser próximos aos valores médios reais destes componentes. Para este trabalho os seguintes valores foram arbitrados para cada nível de confiabilidade:

Tabela 5-2: Níveis de confiabilidade e respectivos valores utilizados.

Nível de Confiabilidade	Confiabilidade
Automotivo	$1 - 10^{-2}$
Militar	$1 - 10^{-3}$
Espacial	$1 - 10^{-4}$

A equação a seguir foi utilizada para o cálculo do custo (P) de cada bloco:

$$P_i(R_i) = e^{(1-f) \cdot \frac{R_i - R_{min}}{R_{max} - R_i}} \quad (5.1)$$

onde f representa a viabilidade de aprimoramento da confiabilidade do bloco.

Os valores utilizados nesta investigação estão descritos na Tabela 5-3.

Tabela 5-3: Valores de utilizados para cálculo do custo(P).

Variável	Valor
f	0.1
R_{min}	$1 - 10^{-2}$
R_{max}	$1 - 10^{-10}$

O custo de uma arquitetura é dado pelo somatório do custo de seus blocos. Esta maneira de calcular o custo é sugerida neste trabalho para substituir a ausência de um catálogo de preços. O custo completo da arquitetura não é influenciado pela forma da árvore, somente pelos atributos de seus componentes e pela quantidade de componentes.

5.6.1.2 Complexidade (C)

A complexidade (C) individual de cada bloco da arquitetura de sensores era feita pelo produto entre a quantidade de portas e a quantidade de canais de cada bloco (B). Os sistemas considerados nesta investigação são do tipo SISO, então, decidiu-se estipular

um valor de complexidade individual de cada bloco, que podem ser vistos na Tabela 5-4.

Tabela 5-4: Valores de complexidade individual de cada bloco.

Categoria	Complexidade Individual
G	1
I	2
D	3
P	4
Z	5
Pc	6
Zc	7

A complexidade da arquitetura é dada pelo arranjo dos blocos. Assim como no modelo de diagrama de blocos, elementos em série representam um produto e elementos em paralelo representam uma soma.

5.6.1.3 Tempo de Subida (Ts)

O Tempo de Subida utilizado neste trabalho foi o período de tempo necessário para a resposta de o sistema ir de 10% a 90% do seu valor final.

5.6.2 Análise Mono-Objetivo

Essa análise ajuda a compreender de maneira mais isolada a influência da forma da estrutura nos atributos escolhidos. Serão calculadas as métricas para cada uma das arquiteturas geradas e os valores serão armazenados em três vetores distintos: vetor de custo **P**, vetor de complexidade **C** e vetor de tempo de subida **Ts**. Cada um desses vetores será ordenado de maneira a permitir a rastreabilidade até ao índice da árvore que ele representa.

Com os dados calculados em uma análise mono-objetivo é simples determinar quais os melhores resultados individuais para cada métrica, aquele que: minimiza o custo, minimiza a complexidade e minimiza o tempo de subida. Porém o principal valor desta análise é entender como a estrutura da árvore afeta cada uma das métricas e esta

análise é relativa e não absoluta. Na busca pelo entendimento deve-se observar os resultados melhores e os piores. Os resultados serão apresentados no Capítulo 8.

5.6.3 Análise Multiobjetivo

A análise multiobjetivo será realizada de maneira similar a Investigação I1. A única alteração será a métrica de desempenho, tempo de subida que substitui a métrica de confiabilidade. Nesta análise multiobjetivo as seguintes métricas serão normalizadas:

$$P_n = P / (\text{Máximo de } P) \quad (5.2)$$

$$C_n = C / (\text{Máximo de } C) \quad (5.3)$$

$$T_{s\ n} = T_s / (\text{Máximo de } T_s) \quad (5.4)$$

Dentre todas essas soluções, buscar-se-ão aquelas **soluções não dominadas** que compõem a **Fronteira de Pareto**. Estas soluções formam um conjunto de soluções candidatas a ser a solução ótima, pois que fazem um compromisso entre as métricas estabelecidas, ou seja, não é mais possível melhorar uma métrica sem prejudicar pelo menos uma outra.

Da mesma maneira como foi feito na investigação (I2), o critério de seleção de uma arquitetura será o critério da Menor Perda e os resultados serão apresentados no Capítulo 8.

6 TERCEIRA INVESTIGAÇÃO (I3): ELABORAÇÃO E SELEÇÃO DE ARQUITETURAS DE CONTROLADORES

A terceira investigação (I3) especificada no Capítulo 0 estabelece que a avaliação de arquiteturas de controladores combinará atributos da Engenharia de Controle com a Engenharia de Sistemas. Serão utilizadas as seguintes métricas: tempo de acomodação, confiabilidade e complexidade.

O método utilizado nesta investigação é o mesmo utilizado na investigação (I2). A diferença está nos arquivos de configuração que ao invés de ter a planta da investigação (I2) possuem a planta da investigação I3 e também as suas restrições.

Os resultados desta investigação serão apresentados no Capítulo 9.

7 RESULTADOS DA PRIMEIRA INVESTIGAÇÃO (II)

7.1 Descrição

Este capítulo apresenta os resultados da geração de arquiteturas de sensores, de acordo com o método descrito anteriormente. Estes dados são resultados da geração de 10.000 arquiteturas viáveis.

Os gráficos que contêm uma árvore possuem uma indicação em cada vértice do tipo de componente com iniciais maiúsculas (ex: T, AD, ARx) e seu nível de confiabilidade em letras minúsculas subscritas, *a* para automotivo, *m* para militar e *s* para espacial.

7.2 Resultados Gerais

A Figura 7.1 traz uma visão ampla das métricas obtidas para todas as 10.000 árvores geradas em uma execução do algoritmo de elaboração de arquiteturas de sensores.

Percebe-se que o eixo de custo (Figura 7.2) é aquele com distribuição mais uniforme entre as métricas, pois essa métrica não é influenciada pela forma da arquitetura, mas somente pela quantidade e custo dos componentes.

Também, percebe-se que os resultados da métrica de complexidade (Figura 7-6) concentraram-se próximo ao seu limite inferior com exceção de alguns valores bem altos quando comparados com a sua média. Está métrica é bem influenciada pelo resultado a forma da estrutura.

Pode-se observar que os resultados da métrica de confiabilidade (Figura 7-10) estão bem concentrados no seu limite superior. Está métrica também é bastante influenciada pelo resultado a forma da estrutura.

Por não haver uma referência na literatura para comparação, as arquiteturas e os resultados serão analisados de maneira relativa entre eles, ou seja, serão comparados os máximos e mínimos para cada medida. Essa maneira se mostrou útil para o entendimento dos resultados.

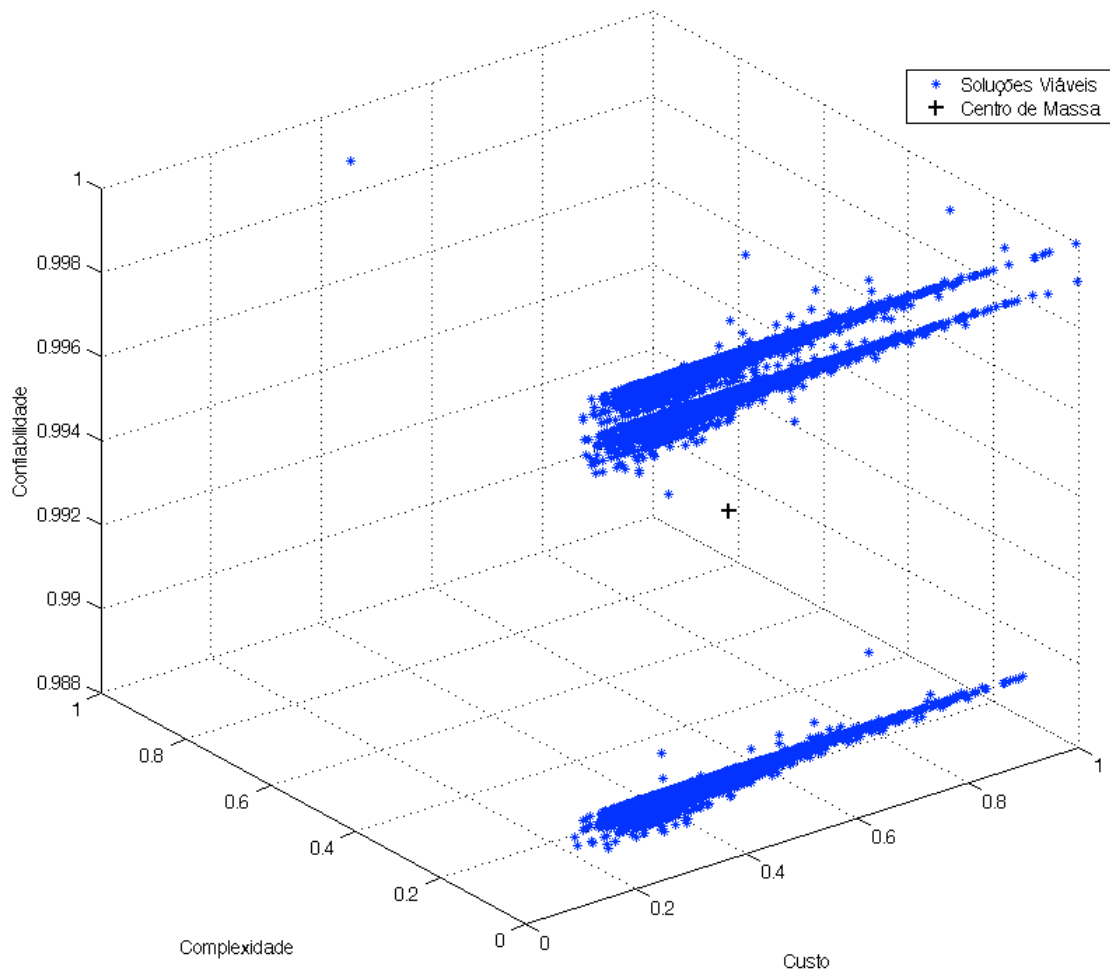


Figura 7.1: Espaço de Soluções com todas as 10.000 Soluções Geradas.

7.3 Métrica de Custo

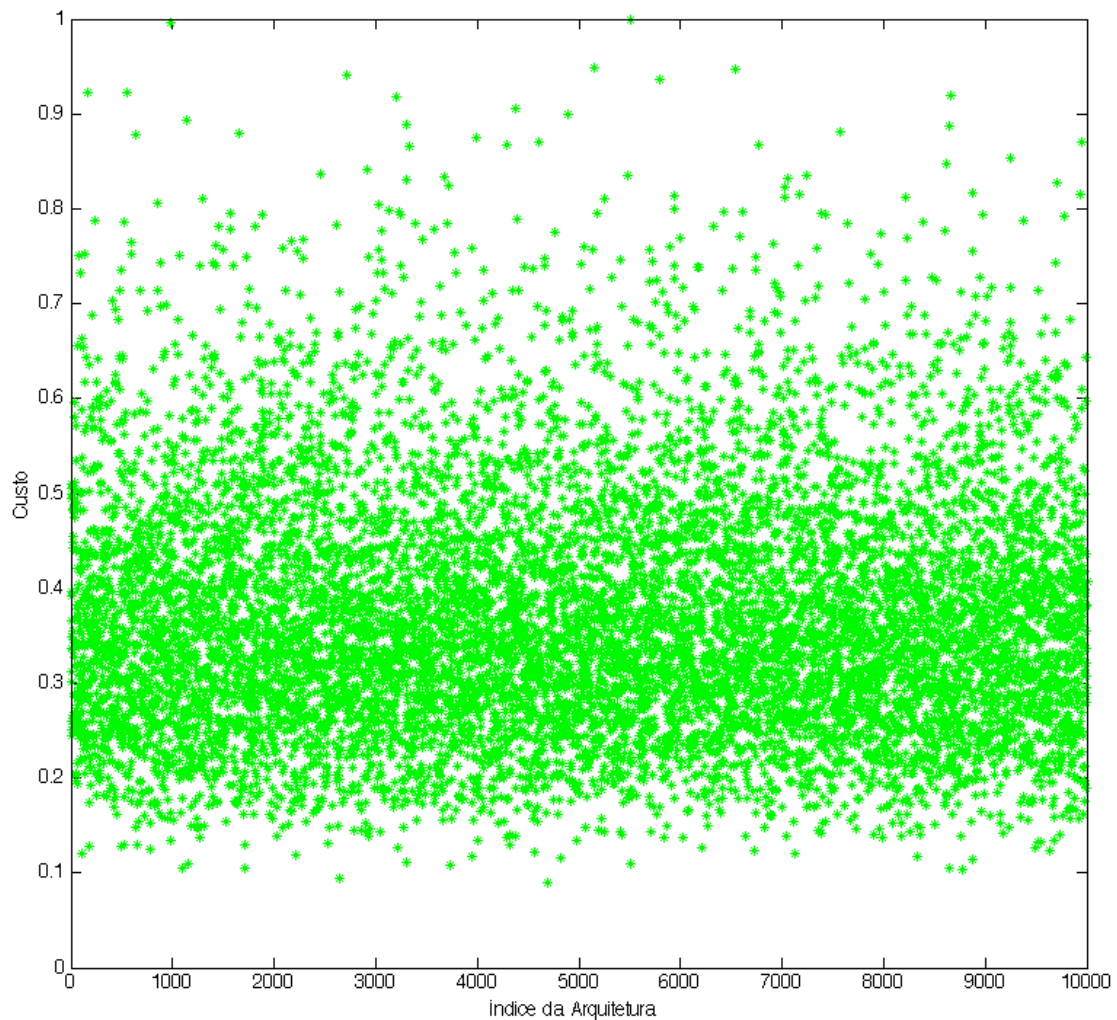


Figura 7.2: Resultados da Métrica de Custo.

A Figura 7-3 e a Figura 7-4 desta seção ilustram as arquiteturas com os menores valores da métrica de custo. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

Pela forma das árvores, aquelas com menor custo, são aquelas com menor quantidade de componentes, menor altura, menor razão entre quantidade de componentes e altura.

7.3.1 Menor Custo

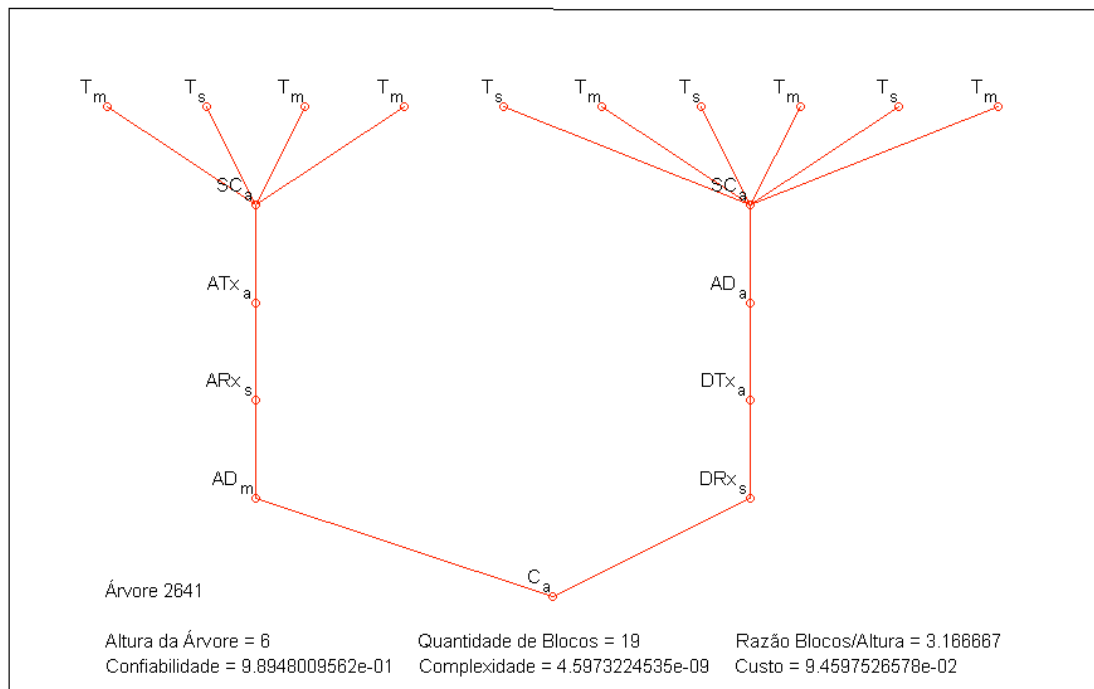
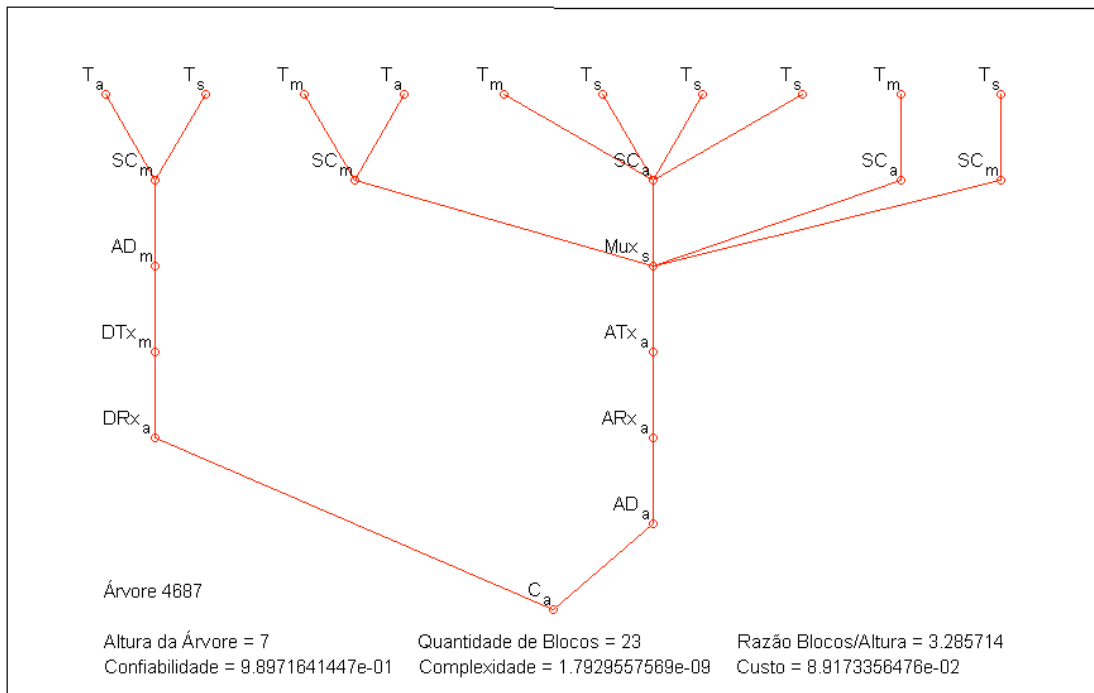


Figura 7-3: Árvores: 1º. Menor Custo (acima) e 2º Menor Custo (abaixo).

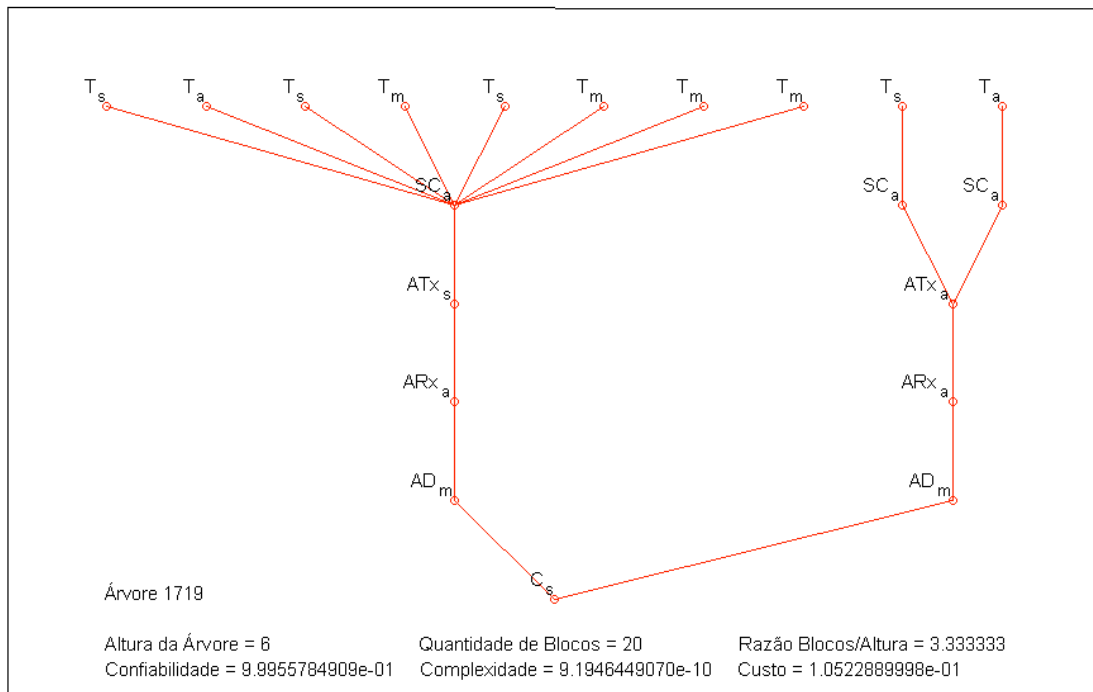
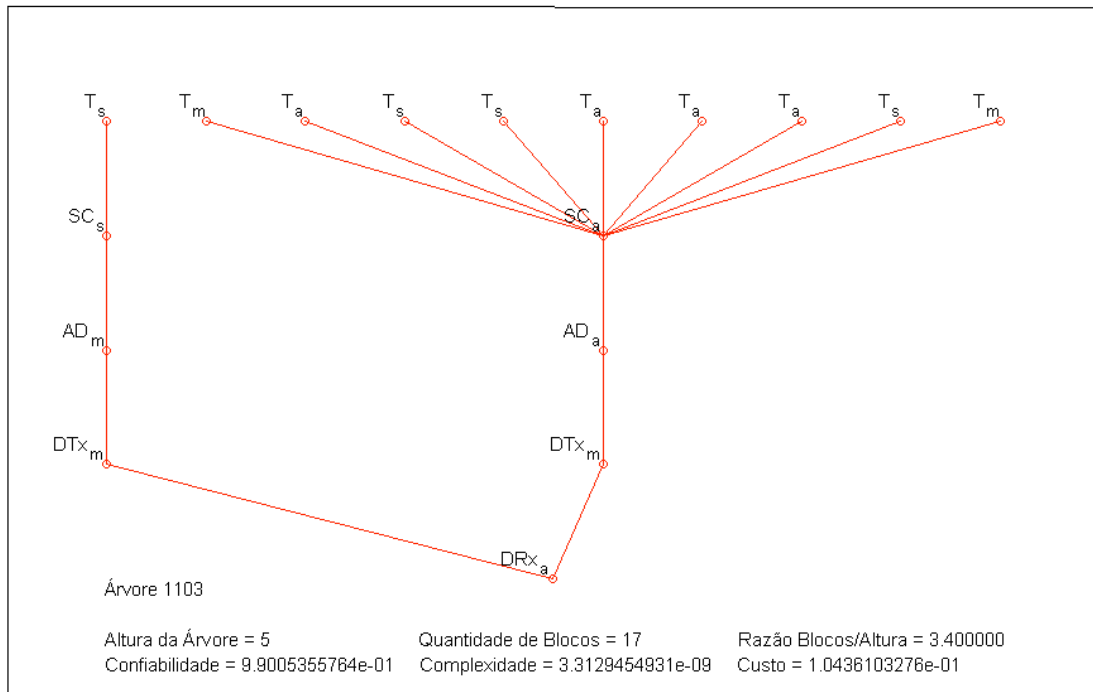


Figura 7-4: Árvores: 4º Menor Custo (acima) e 5º Menor Custo (abaixo).

7.3.2 Maior Custo

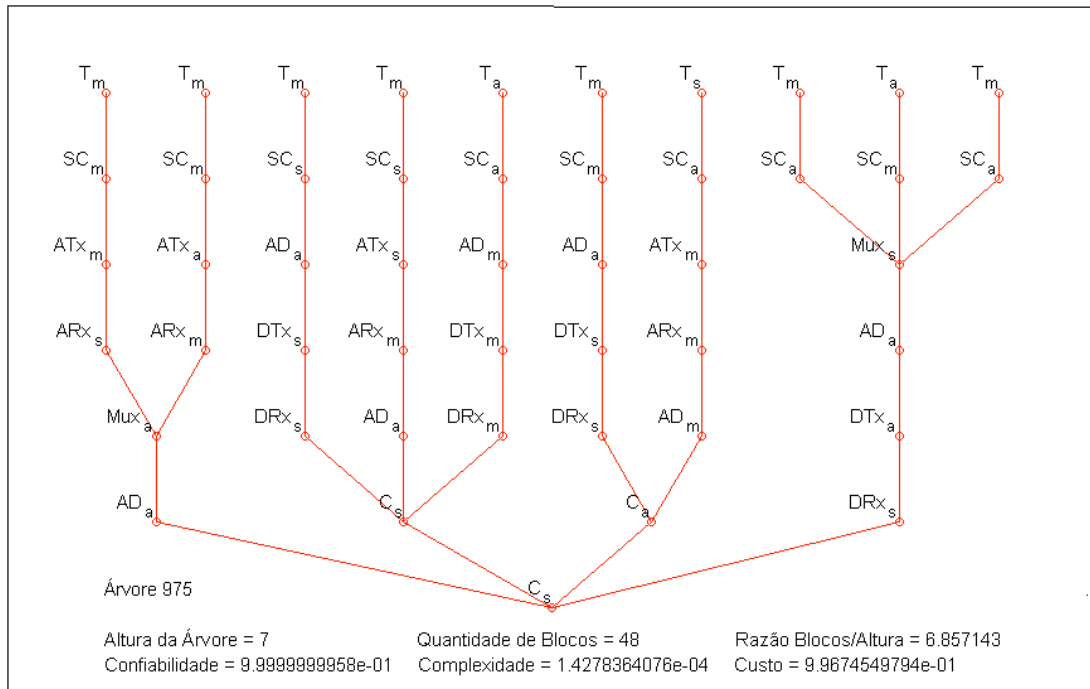
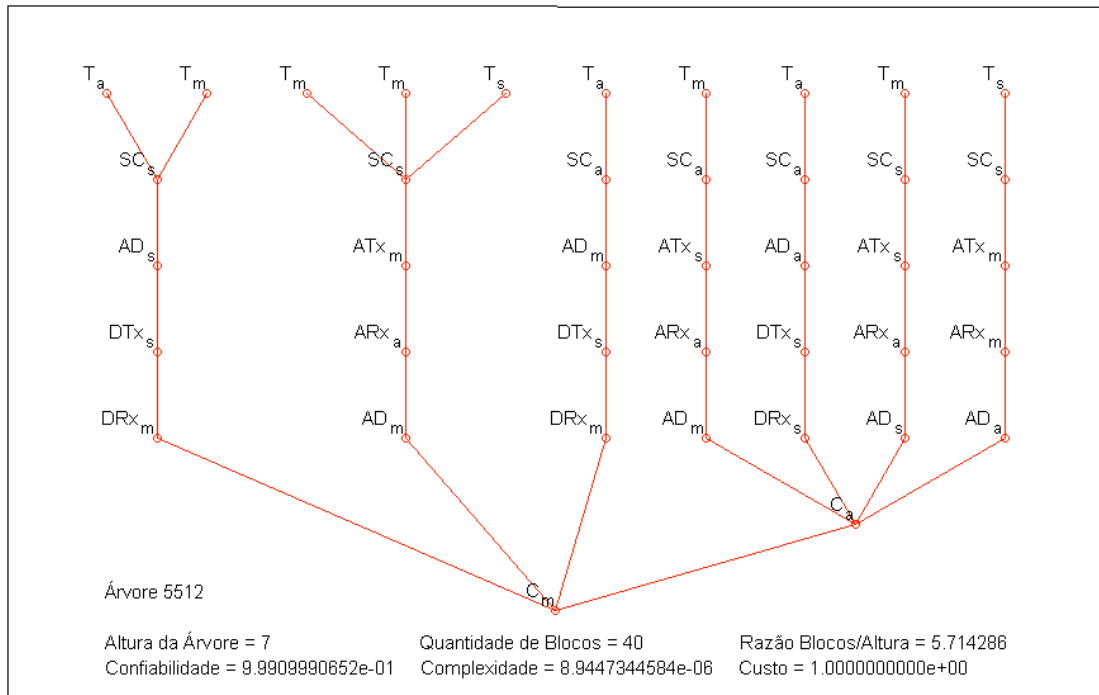


Figura 7-5: Árvore: 1º. Maior Custo (acima) e 2º Maior Custo (abaixo).

7.4 Métrica de Complexidade

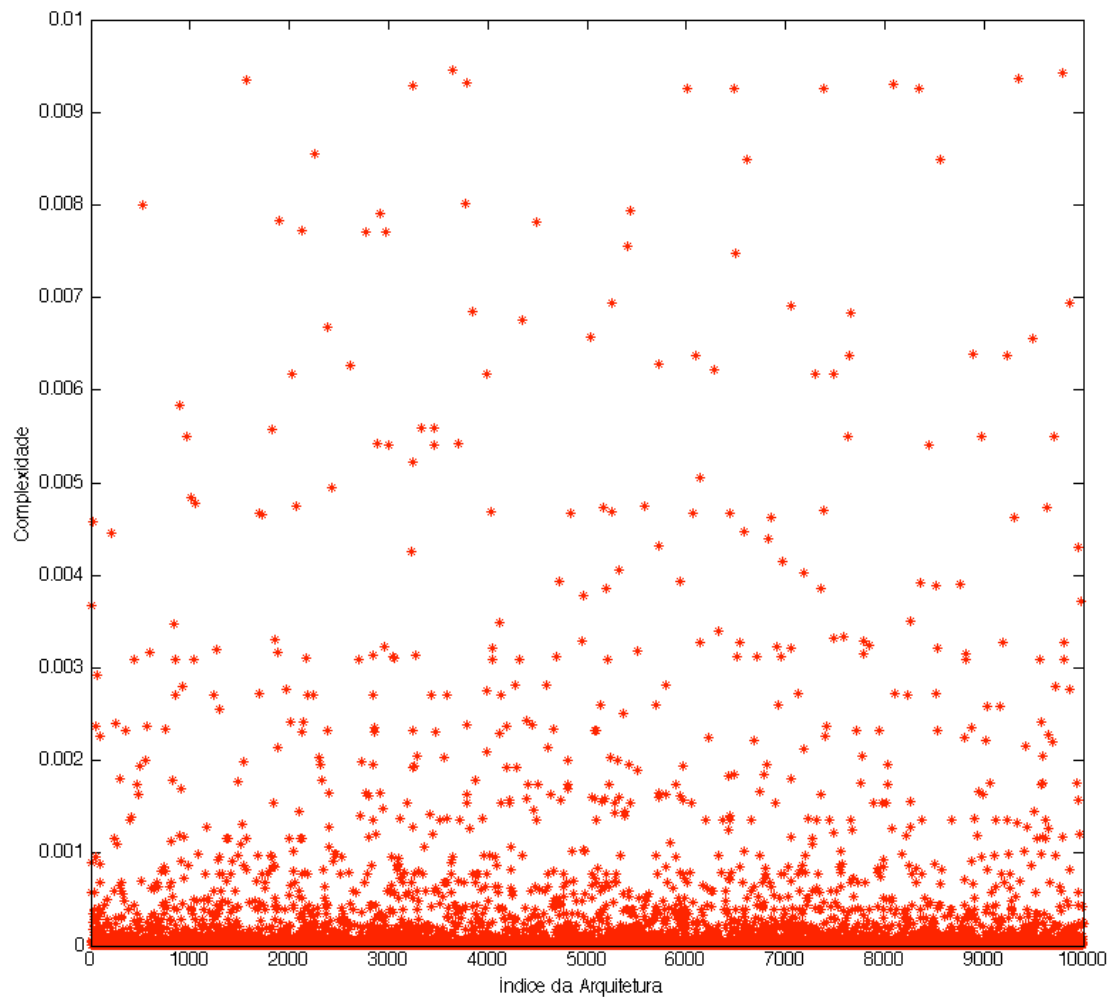


Figura 7-6: Resultados da Métrica de Complexidade.

A Figura 7-7 e a Figura 7-8 desta seção ilustram as arquiteturas com os menores valores da métrica de complexidade. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

Pela forma das árvores, aquelas com menor complexidade, são aquelas com menor altura, baixo valor de razão entre blocos/altura e com menos ramificações.

7.4.1 Menor Complexidade

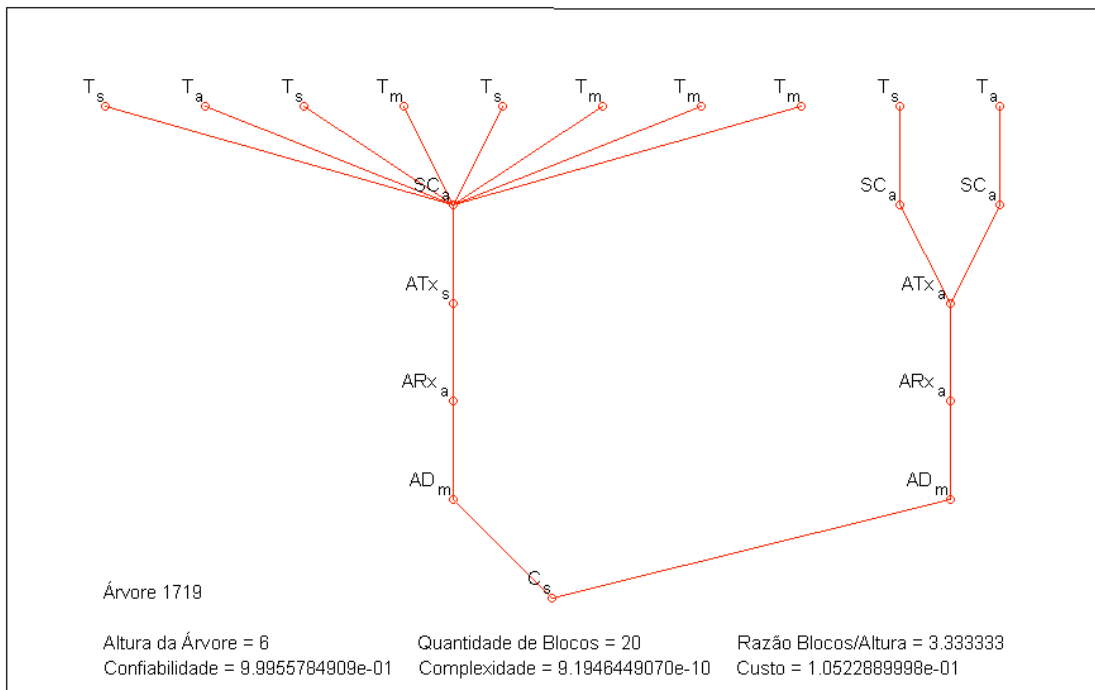
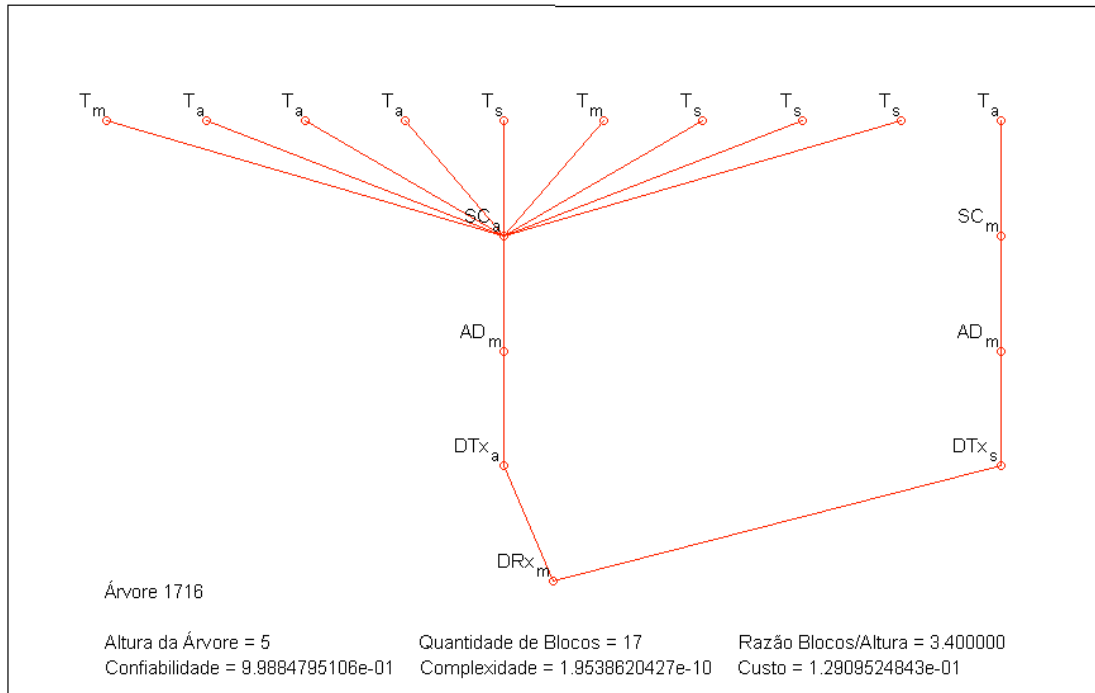


Figura 7-7: Árvores: 1ª. Menor Complexidade (acima) e 2ª. Menor Complexidade (abaixo).

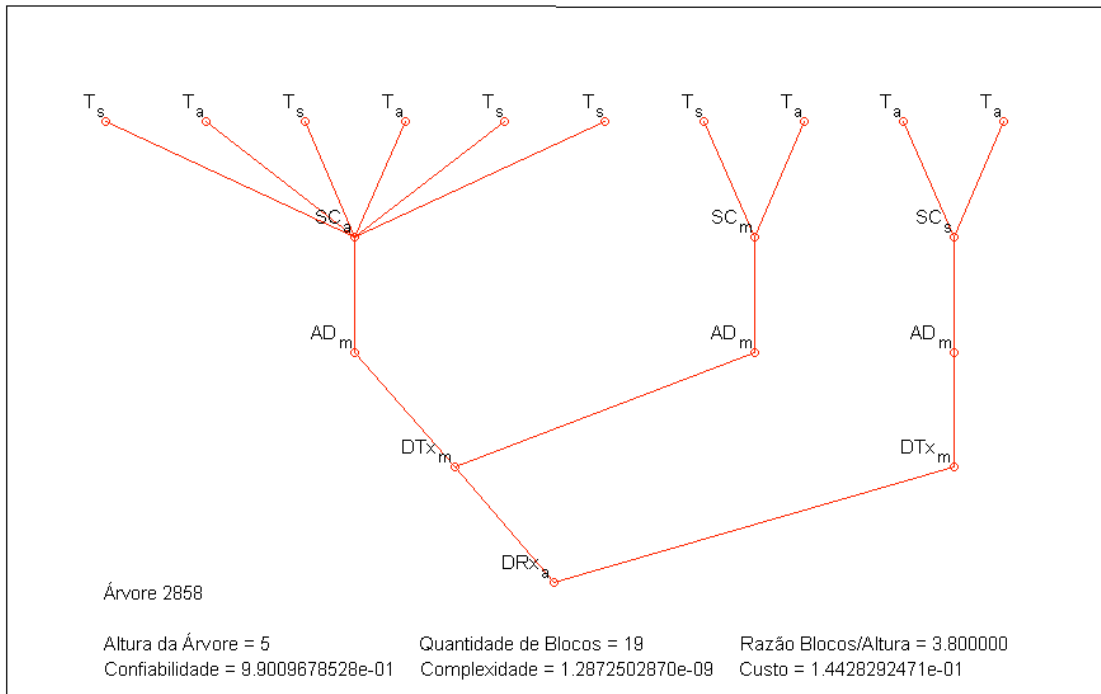
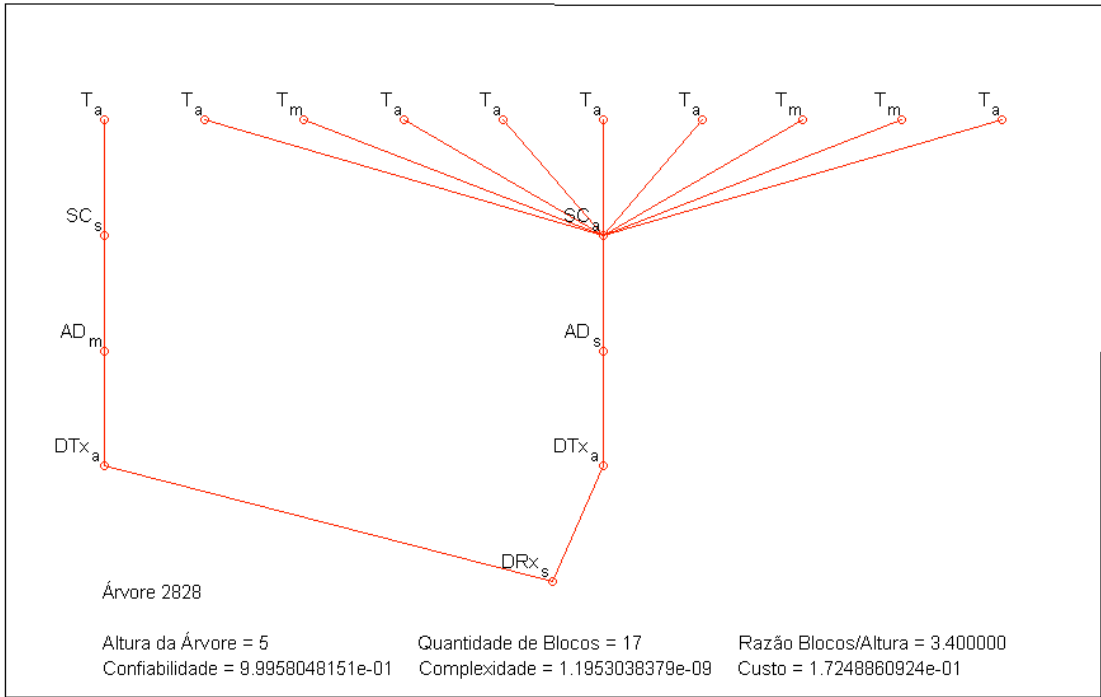


Figura 7-8: Árvores: 4ª. Menor Complexidade (acima) e 5ª. Menor Complexidade (abaixo).

7.4.2 Maior Complexidade

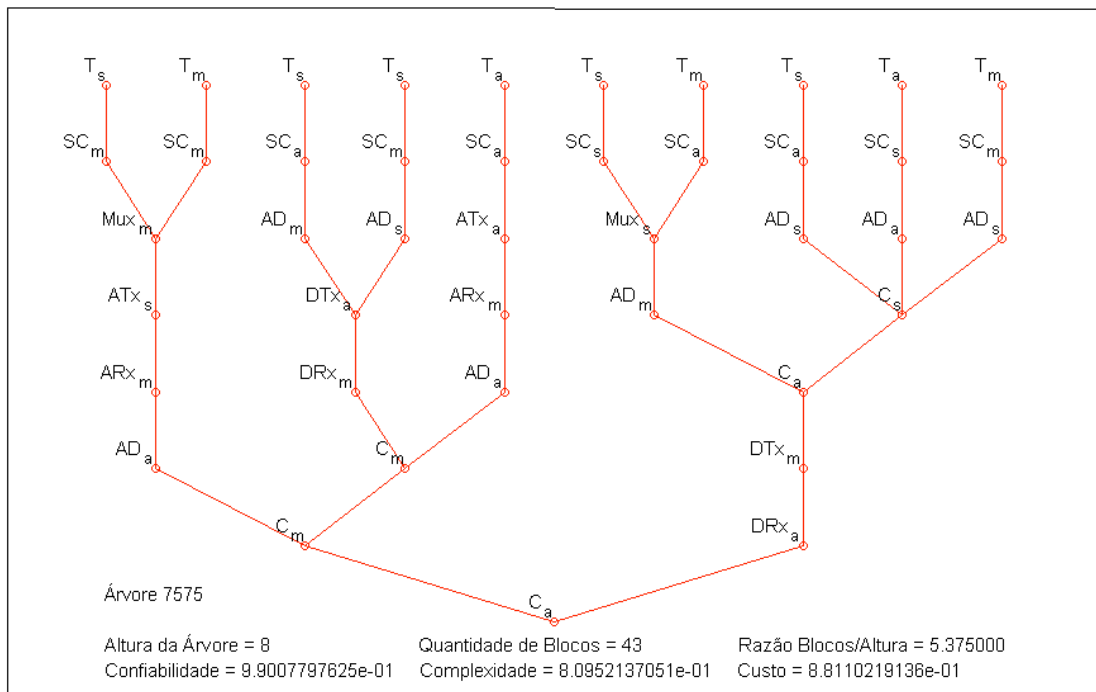
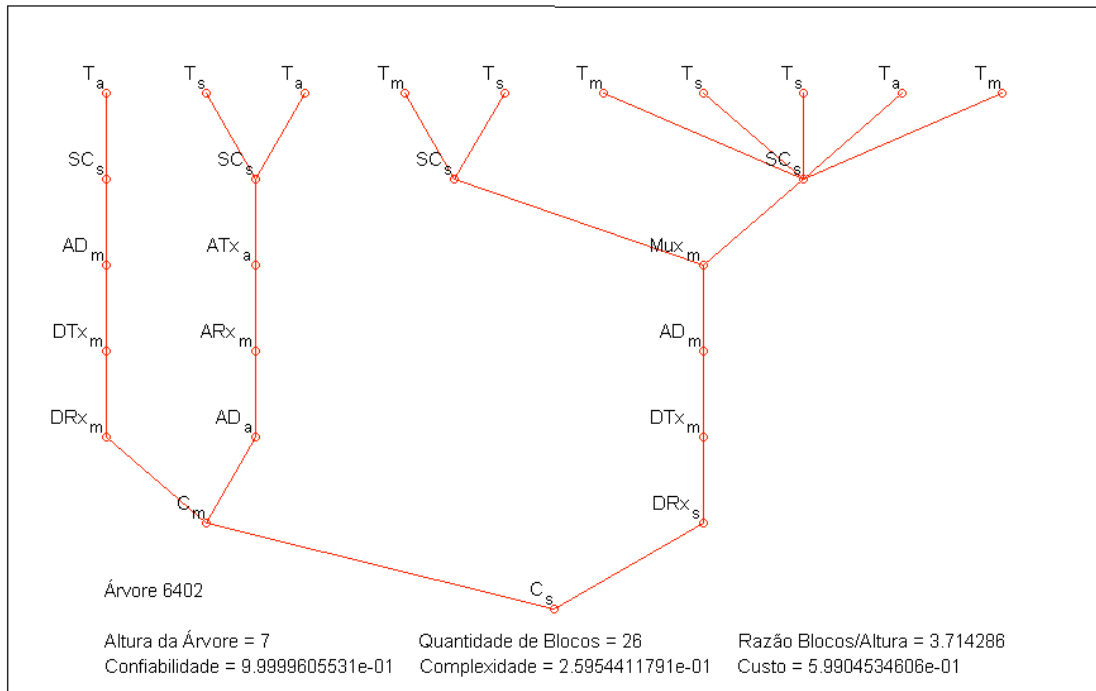


Figura 7-9: Árvores: 1ª. Maior Complexidade (acima) e 2ª. Maior Complexidade (abaixo).

7.5 Métrica de Confiabilidade

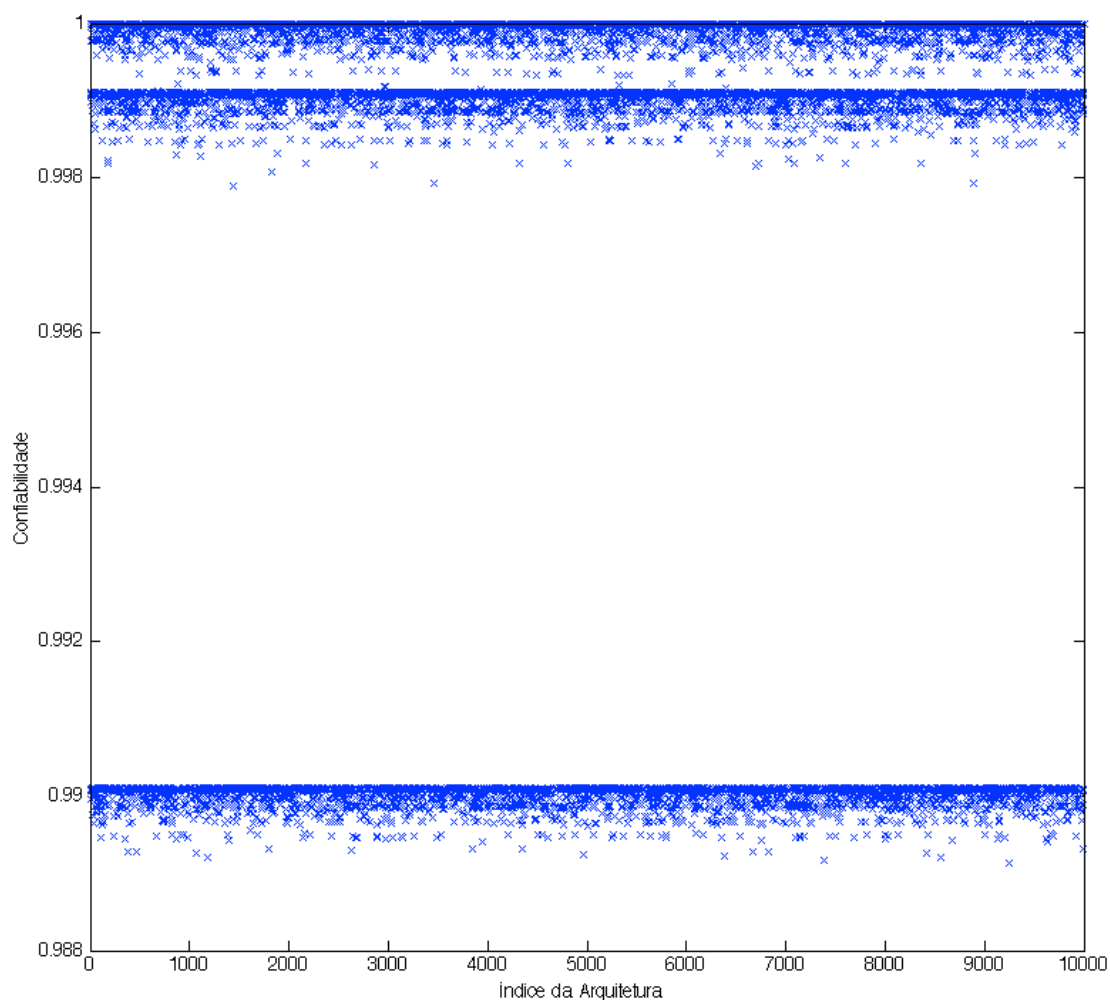


Figura 7-10: Resultados da Métrica de Confiabilidade.

A Figura 7-12 e a Figura 7-13 desta seção ilustram as arquiteturas com os maiores valores para a métrica de confiabilidade. Para efeito de comparação, também serão apresentadas algumas figuras de arquiteturas com os menores valores para a mesma métrica.

Pela forma das árvores, aquelas com maior confiabilidade, são aquelas com maior razão entre blocos/altura, com mais ramificações (caminhos redundantes), e com blocos de maior confiabilidade utilizados próximos à raiz da árvore.

7.5.1 Menor Confiabilidade

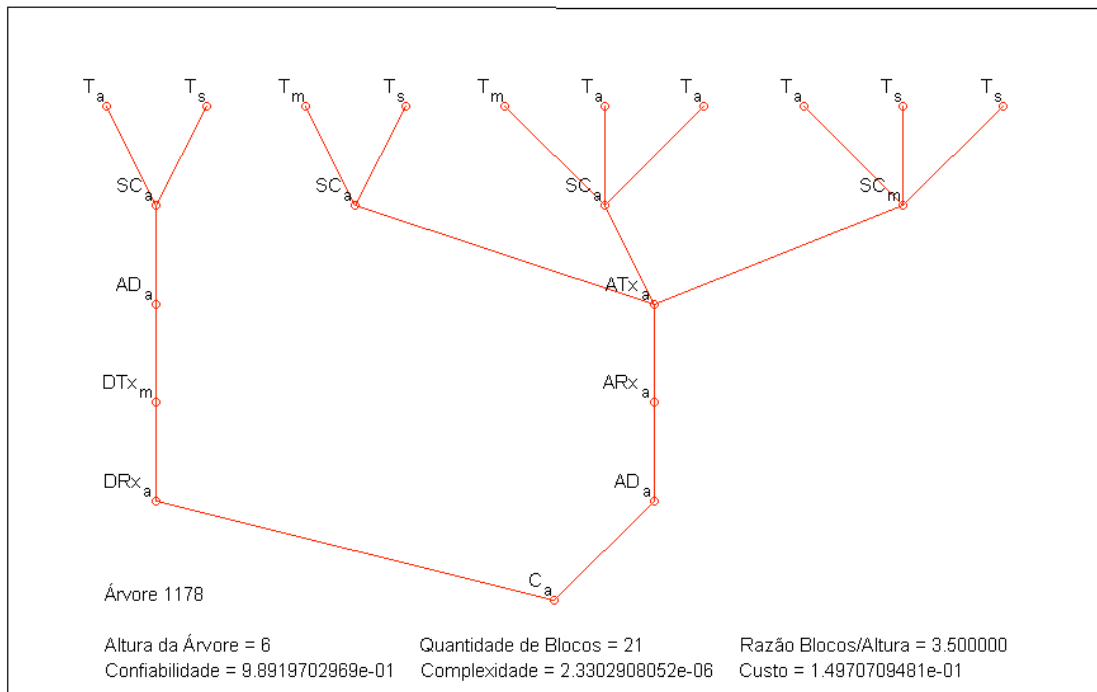
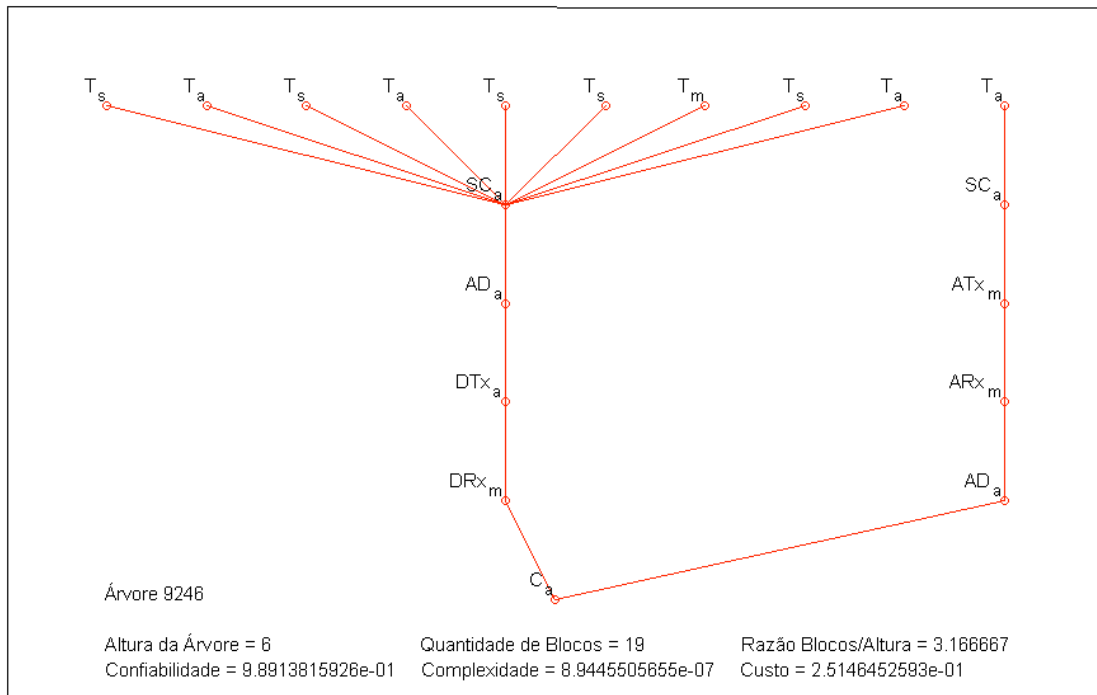


Figura 7-11: Árvores: 1ª. Menor Confiabilidade (acima) e 3ª. Maior Confiabilidade (abaixo).

7.5.2 Maior Confiabilidade

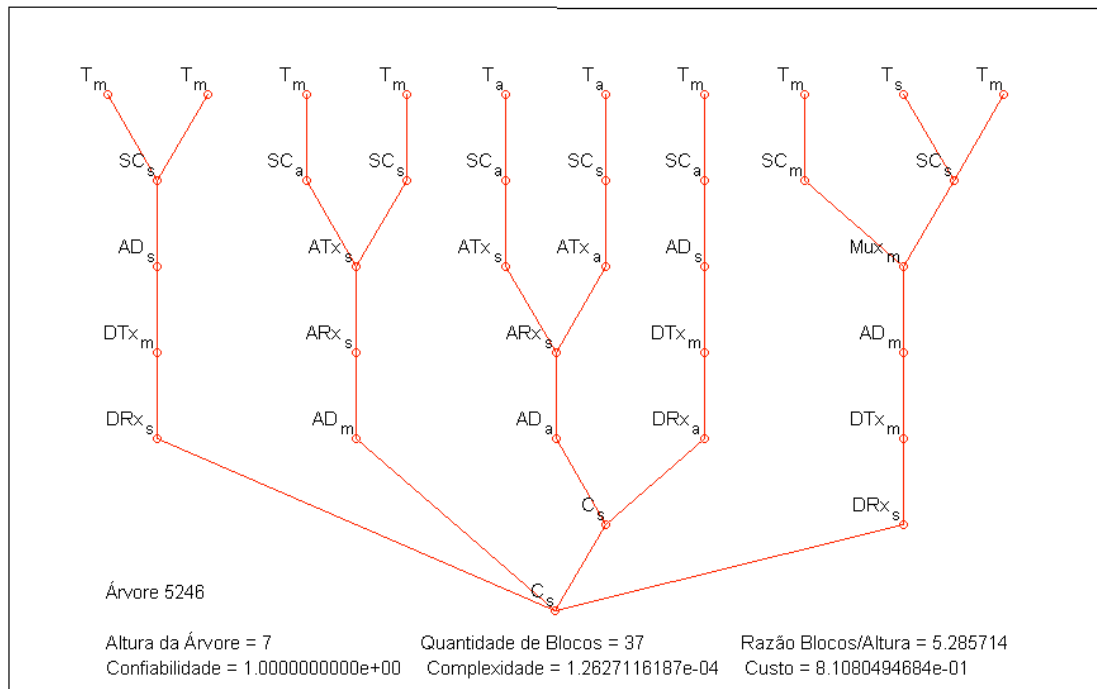
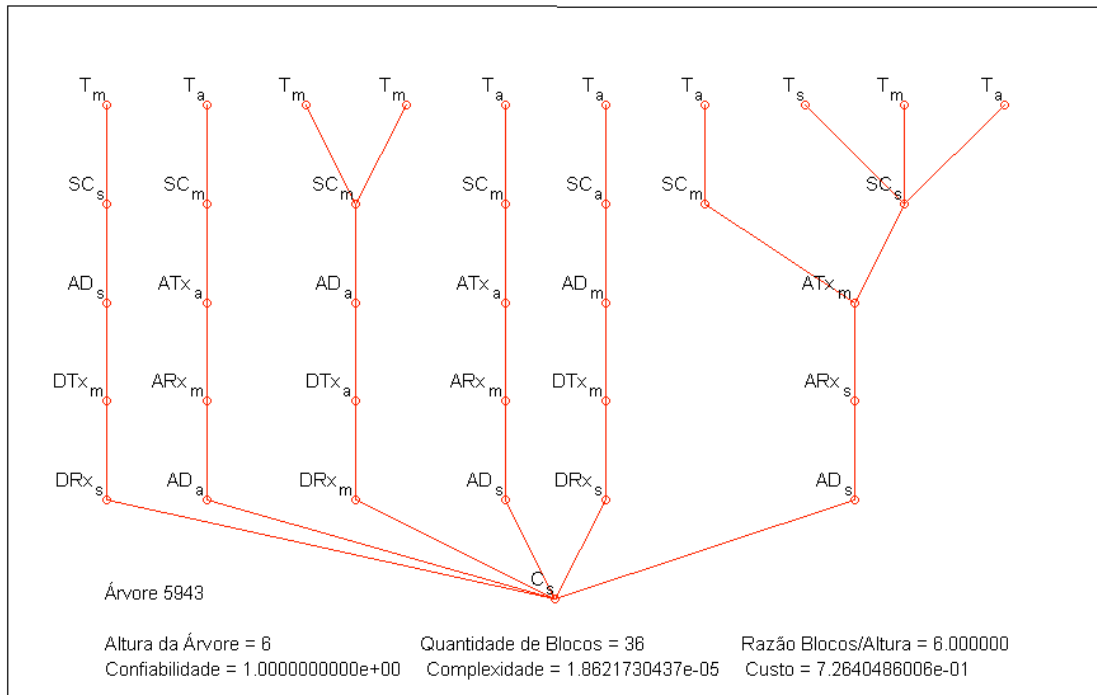


Figura 7-12: Árvores: 1ª. Maior Confiabilidade (acima) e 2ª. Maior Confiabilidade (abaixo).

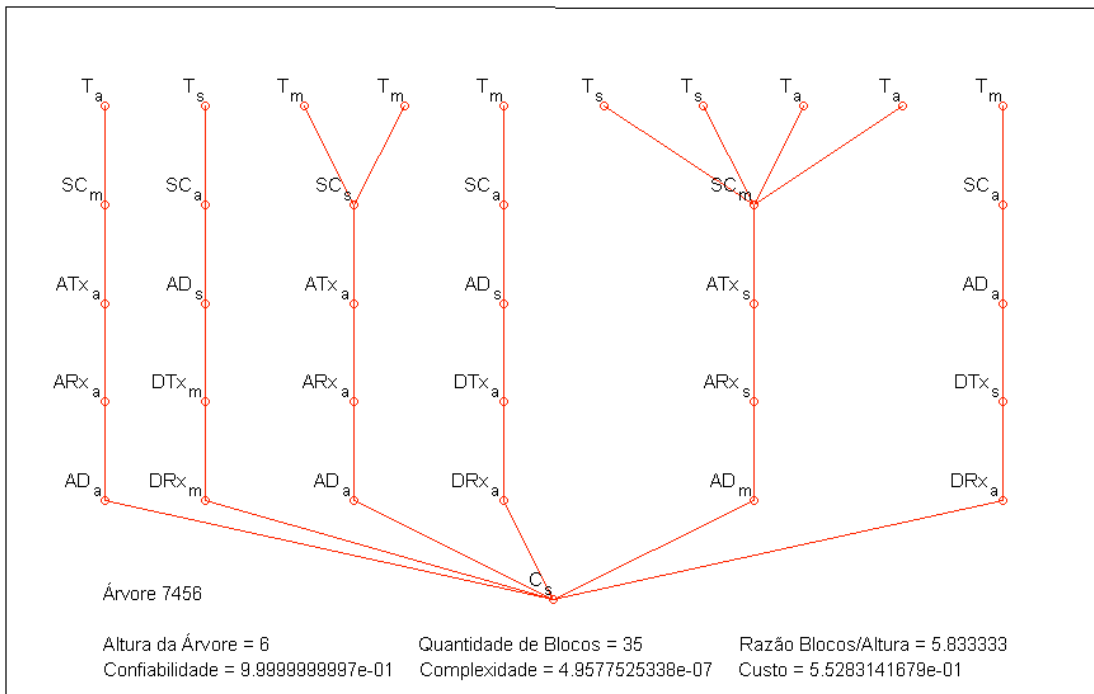
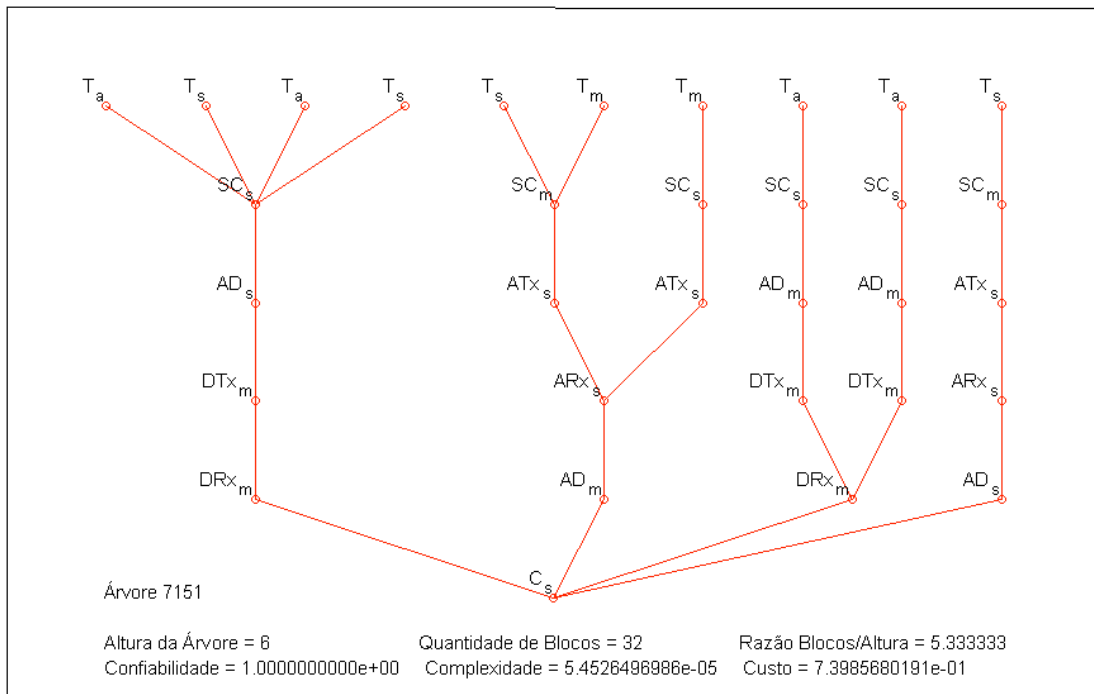


Figura 7-13: Árvores: 5ª. Maior Confiabilidade (acima) e 2ª. Maior Confiabilidade (abaixo).

7.6 Fronteira de Pareto

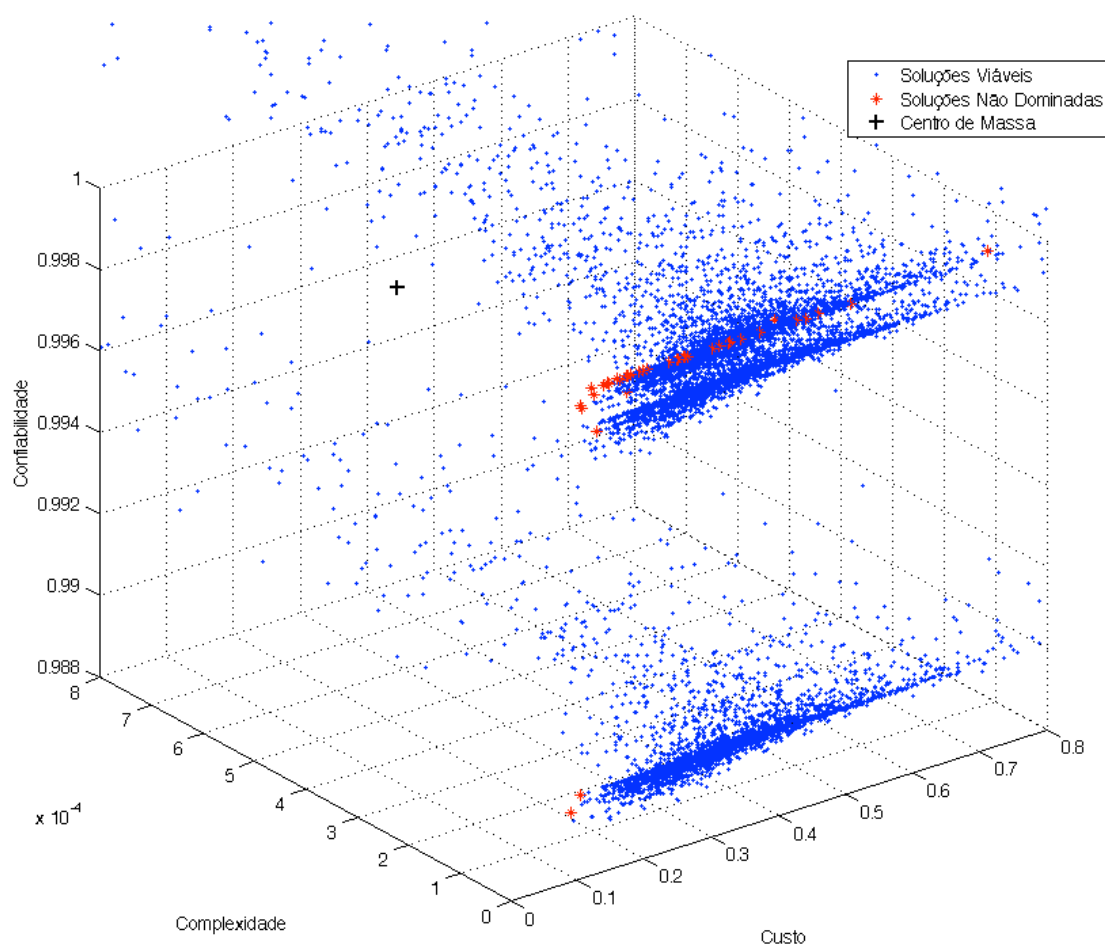


Figura 7-14: Detalhe do Espaço de Soluções e a Fronteira de Pareto.

Dentre as 10.000 soluções viáveis, há 42 soluções não dominadas candidatas a ótimo que formam a Fronteira de Pareto (Figura 7-14). Entre essas algumas são ilustradas pelas Figura 7-15 a Figura 7-16 que seguem ao longo desta seção.

Dentre essas, é possível encontrar soluções que privilegiam a confiabilidade em detrimento das outras métricas; e o mesmo acontece nas outras dimensões. Buscando um equilíbrio entre essas, será usado o critério de equilíbrio revisto na revisão da literatura, e descrito na próxima seção.

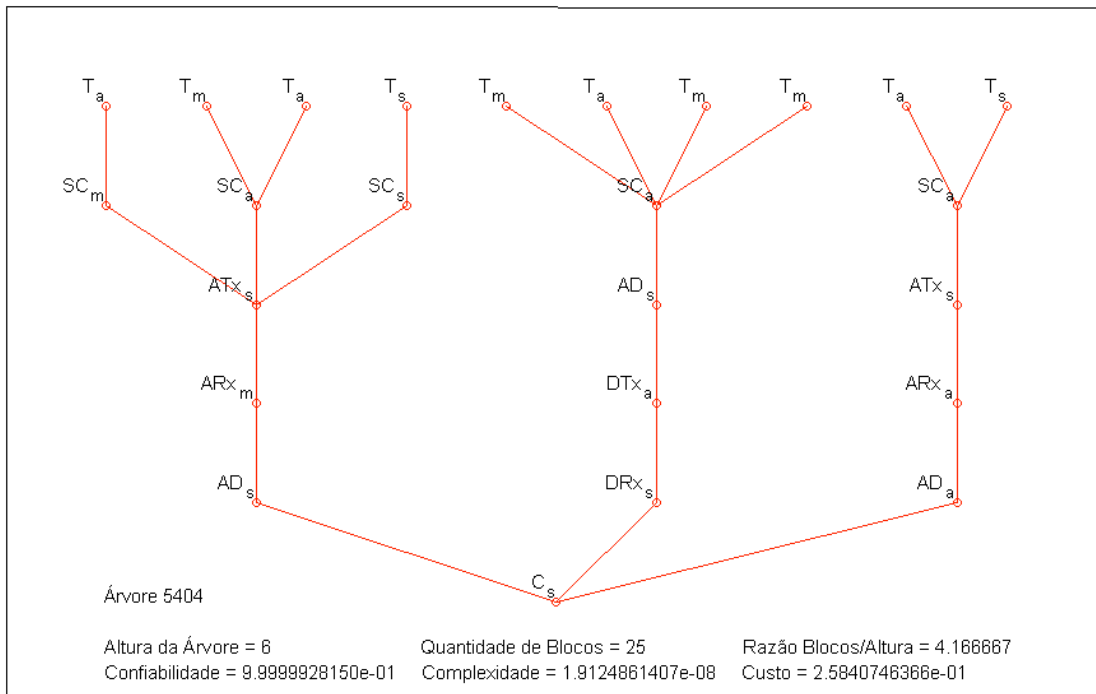
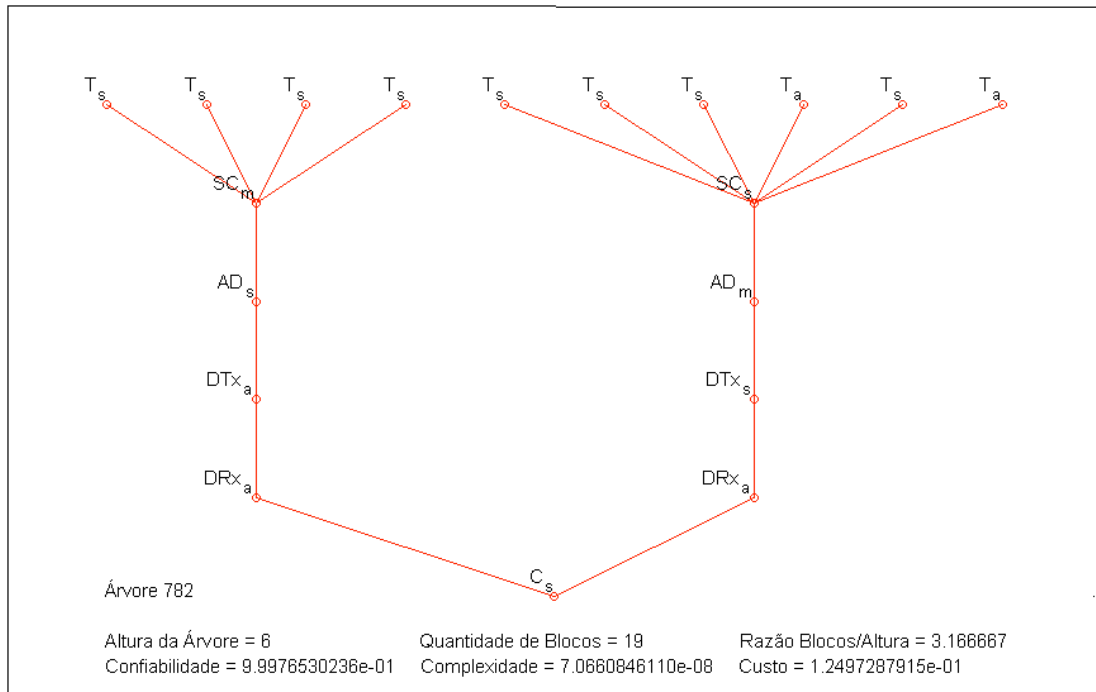


Figura 7-15: Soluções Não Dominadas: Árvore 782 (acima) e Árvore 5404 (abaixo).

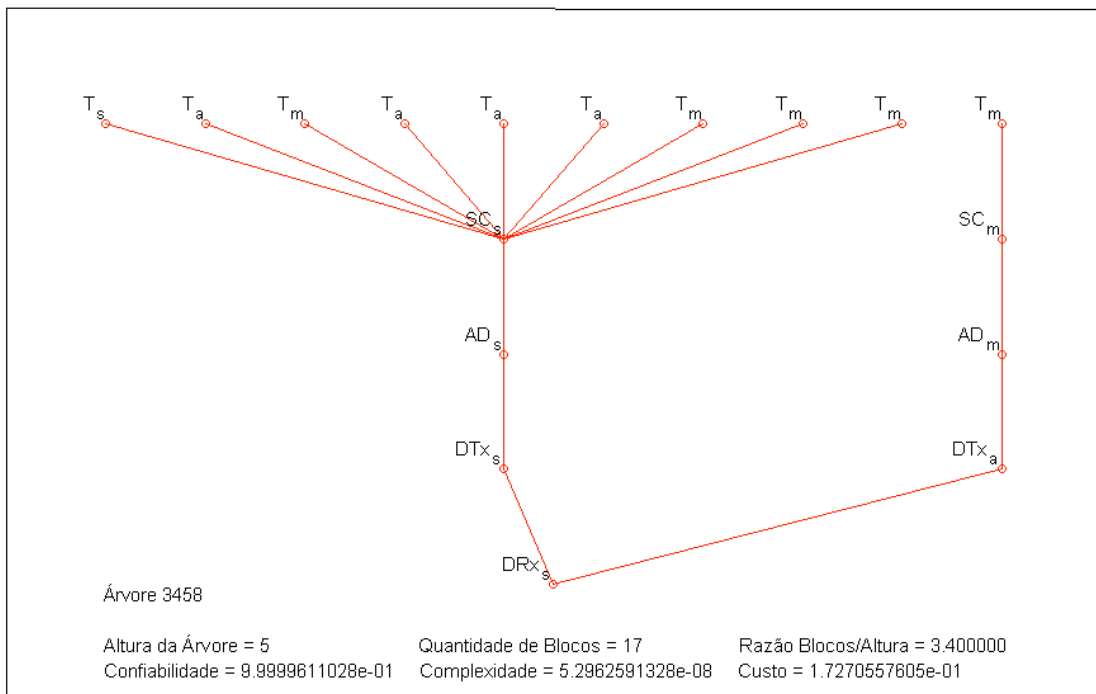
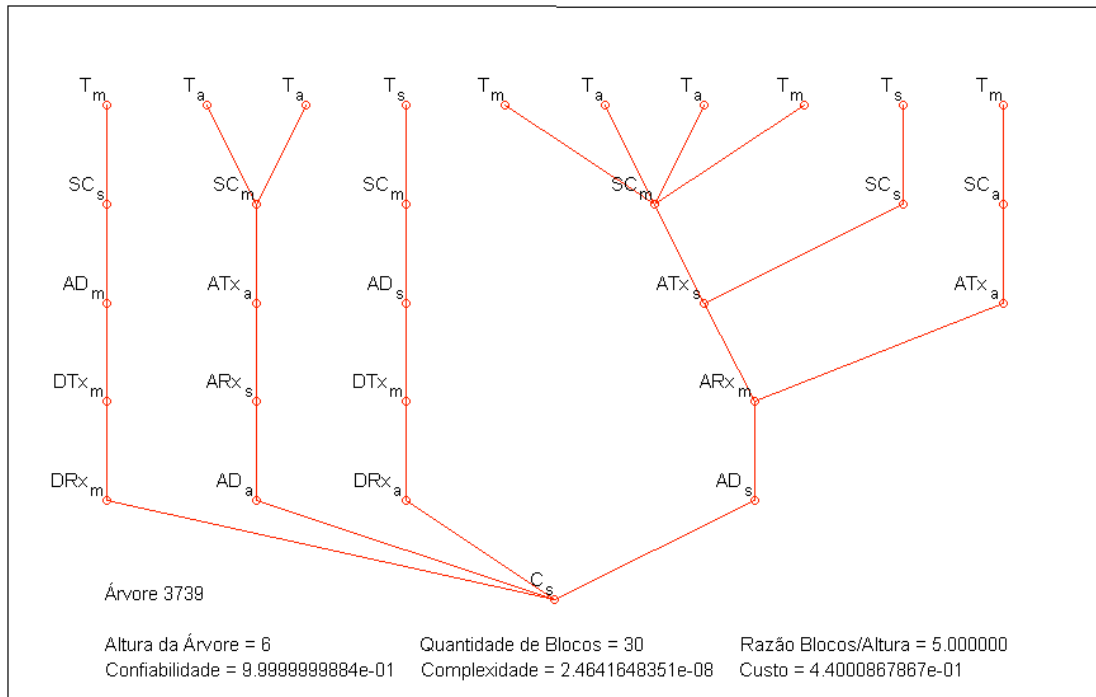


Figura 7-16: Soluções Não Dominadas: Árvore 3739 (acima) e Árvore 3458 (abaixo).

7.7 Seleção pelo Critério da Menor Perda

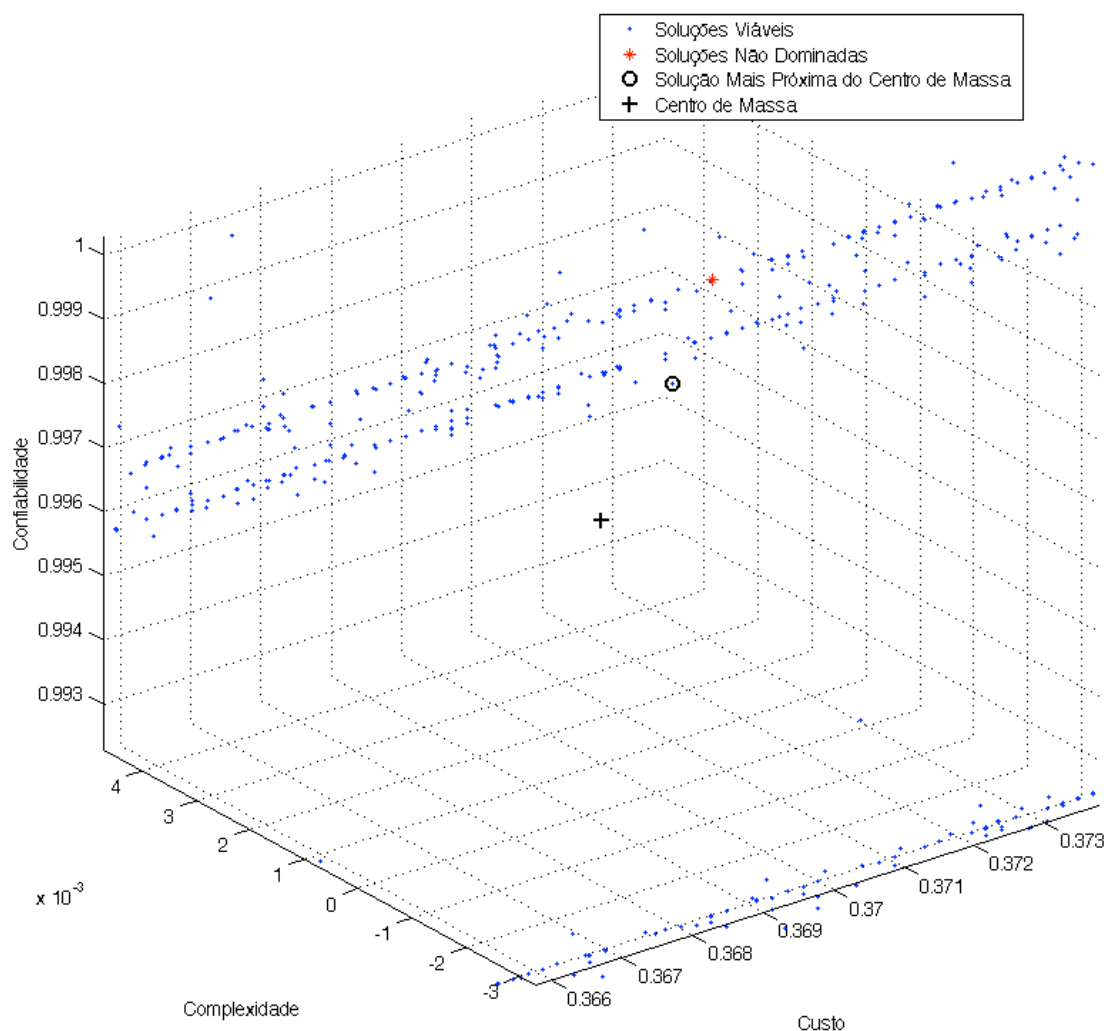


Figura 7-17: Detalhe da Solução mais Próxima do Centro de Massa (Baricentro).
O baricentro em um espaço normalizado foi (Figura 7-17):

Custo = 0.3697551, Complexidade = 0.0007027 e Confiabilidade = 0.9962870.

A distância entre o baricentro e a solução na Fronteira de Pareto mais próxima é de 0.0039197 (Árvore 9725); e, na sequência, a segunda mais próxima possui uma distância de 0.0276410 (Árvore 8825), ambas na Figura 7-18. A distância entre do baricentro e a soluções na Fronteira de Pareto mais distante é de 0.3566698 (Árvore 5943); e, na sequência, a segunda mais próxima possui uma distância de 0.2806595 (Árvore 4687), ambas na Figura 7-19.

7.7.1 Entre as Soluções na Fronteira de Pareto, as Mais Próximas do Baricentro

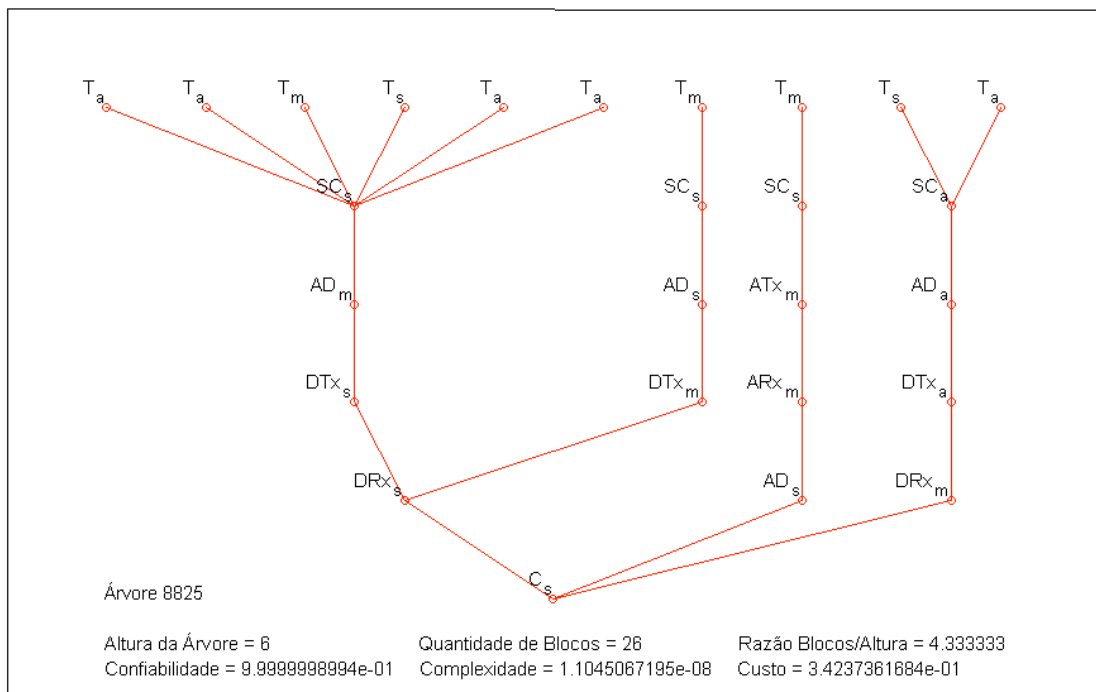
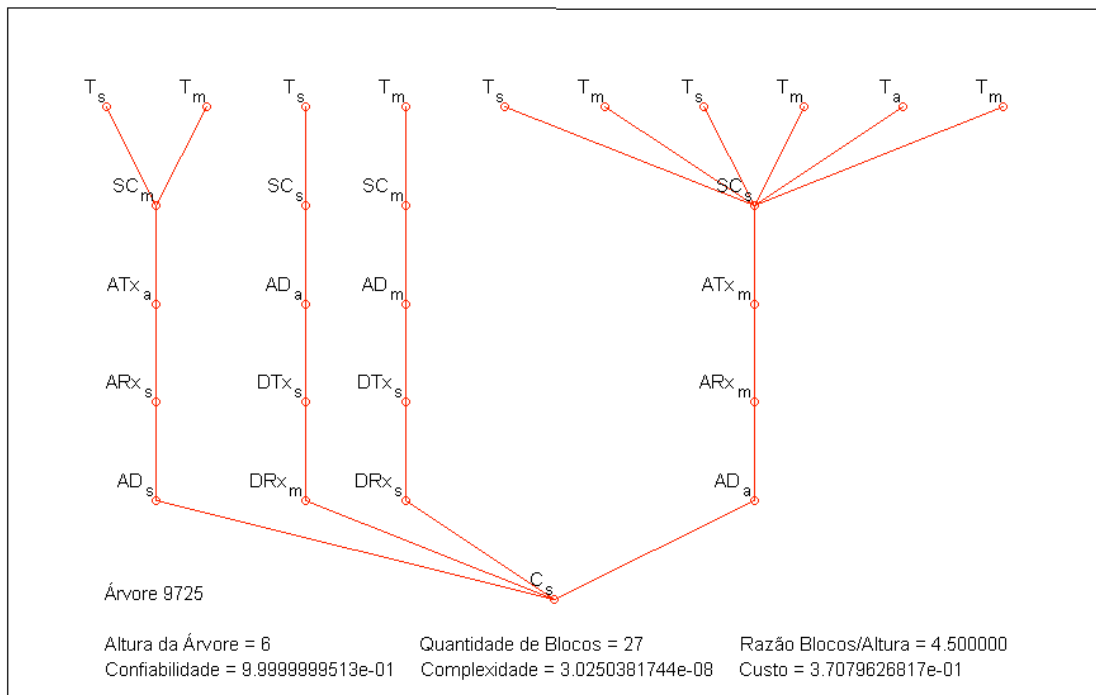


Figura 7-18: Árvores: Na Fronteira de Pareto 1ª. Mais Próxima do Baricentro (acima) e 2ª. Mais Próxima (abaixo).

7.7.2 Entre as Soluções na Fronteira de Pareto, as Mais Distantes do Baricentro

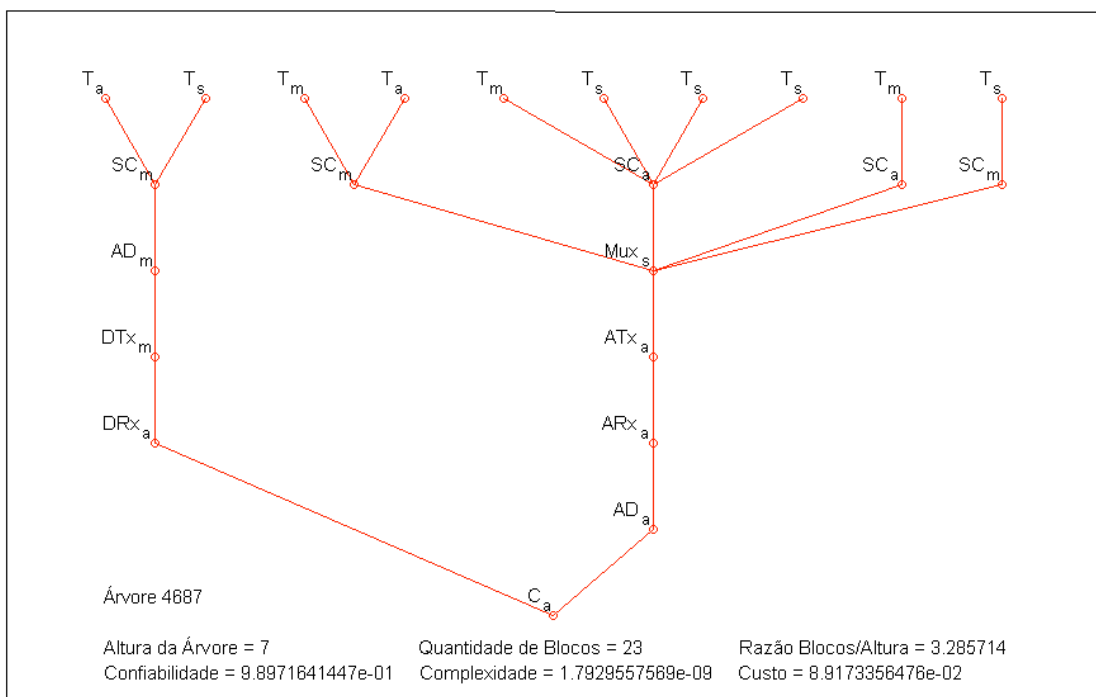
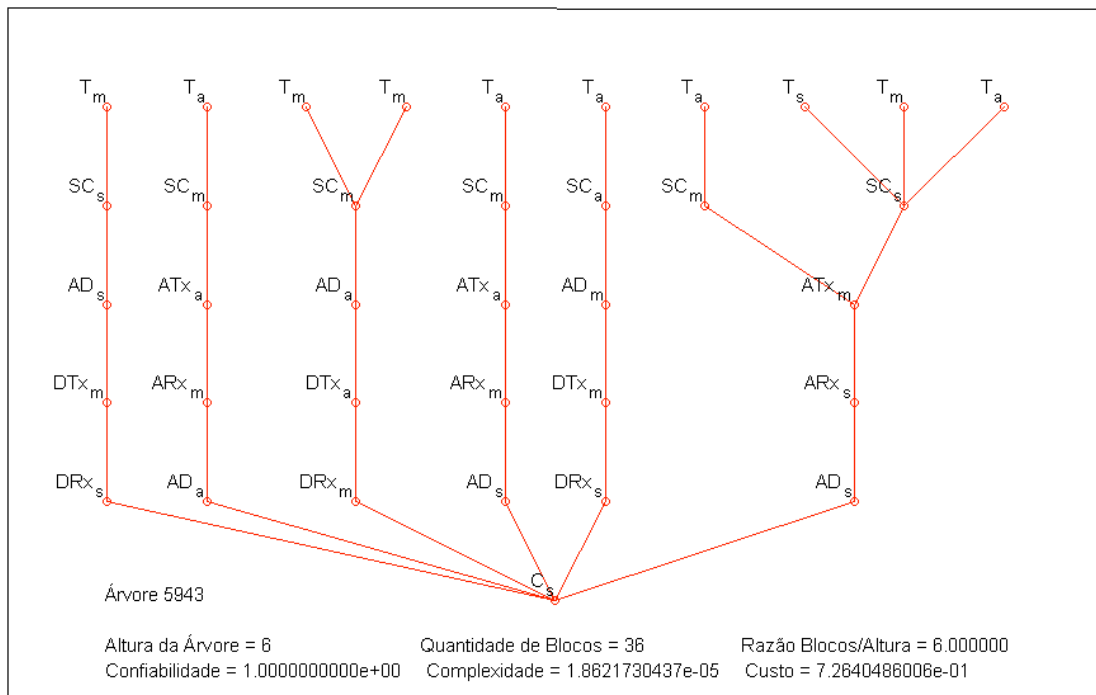


Figura 7-19: Árvores: Na Fronteira de Pareto 1ª. Mais Distante do Baricentro (acima) e 2ª. Mais Distante (abaixo).

8 RESULTADOS DA SEGUNDA INVESTIGAÇÃO (I2)

8.1 Descrição

Este capítulo apresenta os resultados da geração de arquiteturas de controladores de acordo com o método descrito no Capítulo 0. Estes dados são resultados da geração de 100 arquiteturas que após o ajuste do controlador resultaram em 70 arquiteturas de controladores estáveis que não violam nenhuma das restrições impostas pelo problema da Investigação I2.

Nos gráficos que ilustram diagrama de blocos, abaixo de cada bloco há uma indicação do número identificador do bloco seguido do seu nível de confiabilidade, a para automotivo, m para militar e s para espacial.

Esses resultados foram gerados utilizando somente os seguintes tipos de blocos: Proporcional, Integral, Derivativo e um Pólo Real.

A Figura 8-1 ilustra a estrutura utilizada para gerar os resultados deste capítulo. O bloco de saturação após a saída do controlador não se faz necessário, pois os controladores gerados não ultrapassam os limites impostos pelo problema, que são de -40V a +40V.

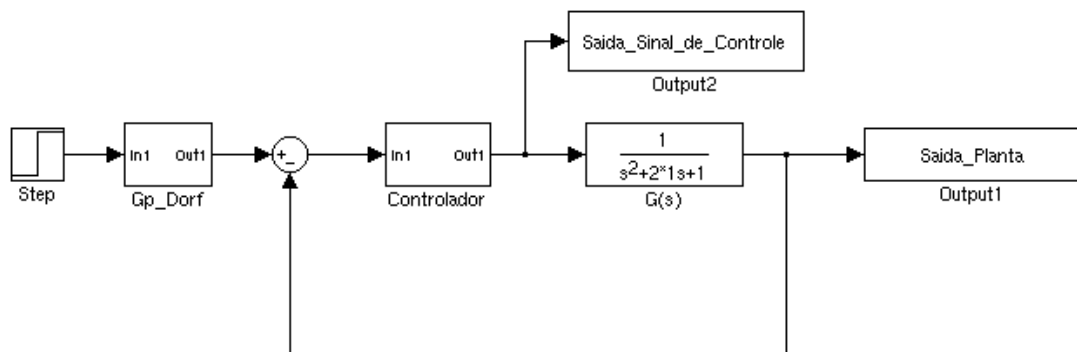


Figura 8-1: Estrutura do Sistema de Controle da Investigação I2.

Para estes resultados foi utilizado o mesmo Pré-Filtro do trabalho do (DORF;

BISHOP, 1998), $G_p = \frac{42.67}{s^2 + 11.38s + 42.67}$.

8.2 Resultados Gerais

A Figura 8-2 traz uma visão ampla das métricas obtidas para todas os 70 controladores gerados viáveis em uma execução do algoritmo de elaboração de controladores.

A quantidade de arquiteturas geradas para os controladores é 100 vezes menor do que aquelas geradas para os sensores. Essa redução se deve ao fato do tempo necessário para ajustar dos parâmetros de cada controlador, tarefa mandatória, uma vez que agora trata-se de um sistema dinâmico.

O ajuste das 100 arquiteturas geradas foi efetuado em um computador com processador de 1.86 GHz Intel Core 2 Duo, com o Sistema Operacional Mac OS X 10.7.5. Cada arquitetura levou em média 4'13" para ser ajustada, totalizando em aproximadamente 7h para os 100 controladores.

As arquiteturas e os resultados serão analisados de maneira relativa entre eles, ou seja, serão comparados os máximos e mínimos para cada medida. Essa maneira se mostrou útil para o entendimento dos resultados.

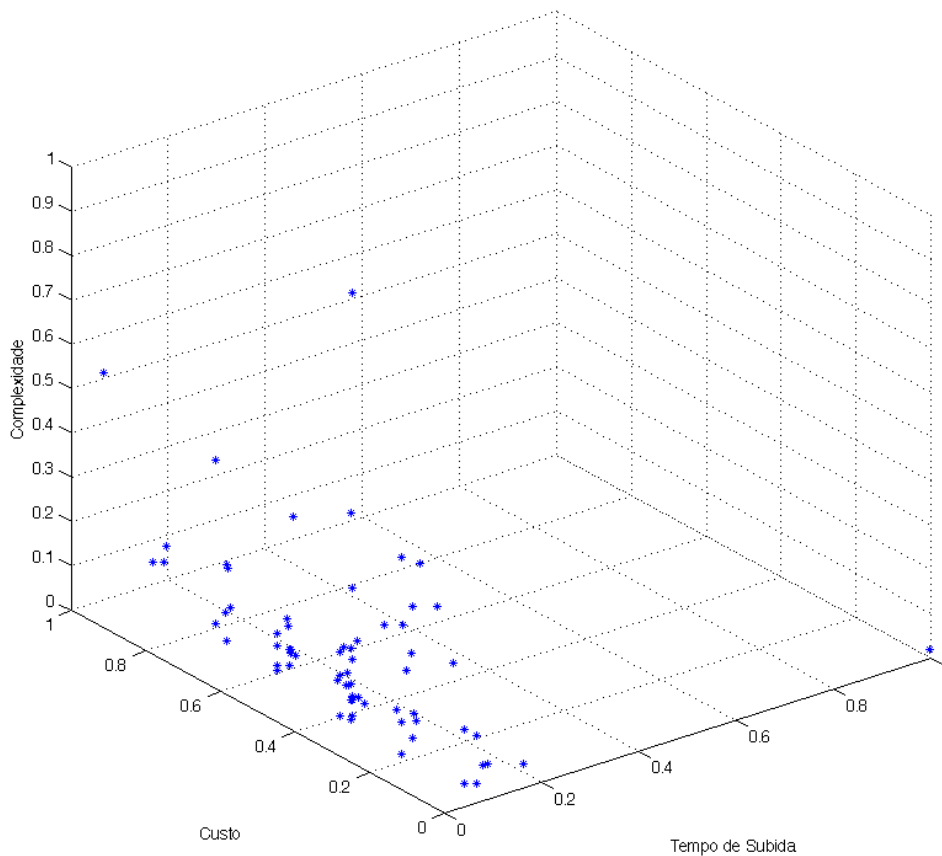


Figura 8-2: Espaço de Soluções com todas as 70 soluções aceitáveis.

8.3 Métrica Tempo de Subida

Esta seção apresenta os resultados relacionados à métrica de Tempo de Subida. O resultado desta métrica para todos os controladores pode ser visto na Figura 8-3, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 8-4 e a Figura 8-5 desta seção ilustram os controladores com os menores valores da métrica de Tempo de Subida. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

O Tempo de Subida utilizado neste trabalho foi o período de tempo necessário para a resposta de o sistema ir de 10% a 90% do seu valor final.

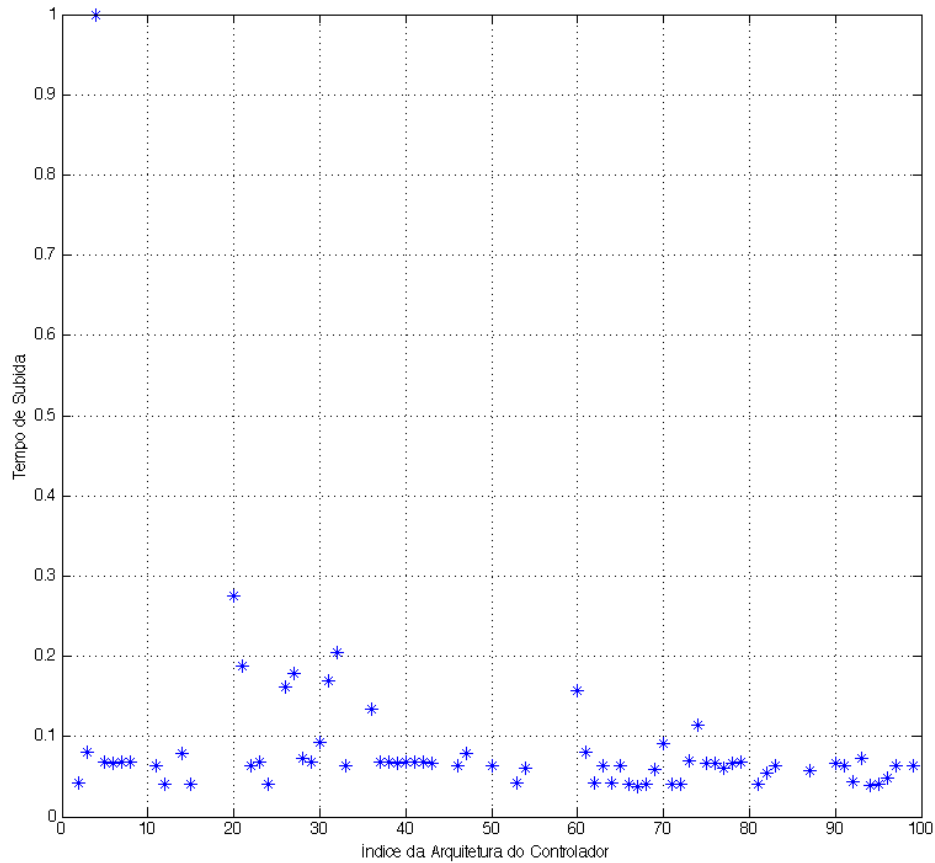


Figura 8-3: Resultados da Métrica de Tempo de Subida.

8.3.1 Menor Tempo de Subida

Esta seção apresenta os controladores com os menores tempos de subida. O controlador da Figura 8-4 obteve um Tempo de Subida de 0.23471s, Custo de 39868.05 e Complexidade 15. O controlador da Figura 8-5 obteve um Tempo de Subida de 0.24916s, Custo de 59796.16 e Complexidade 15.

Pode se observar que estes controladores possuem arquiteturas similares, um PID com um polo após a saída do bloco de derivação.

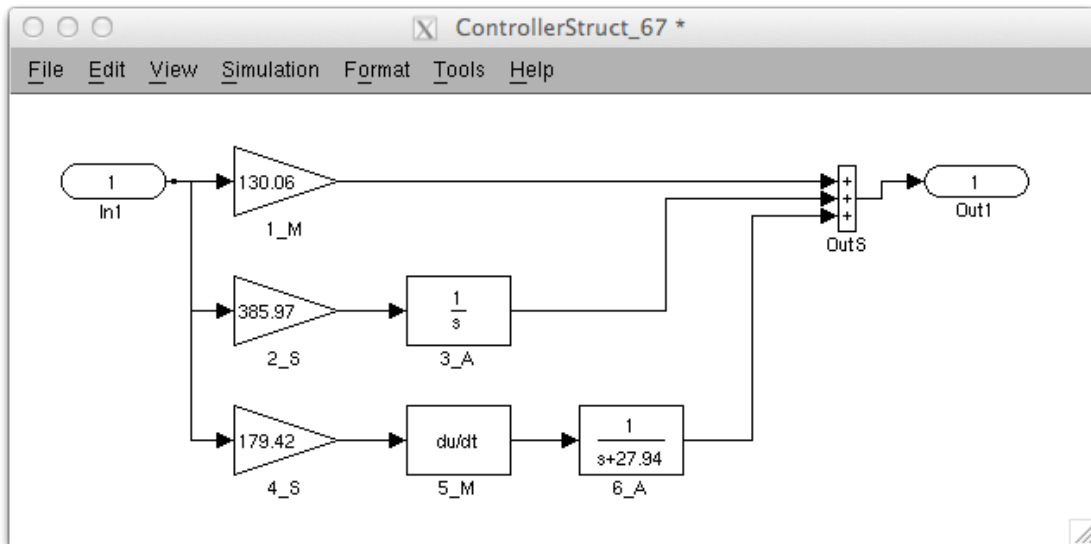


Figura 8-4: Controlador 67 – 1º. Menor Tempo de Subida.

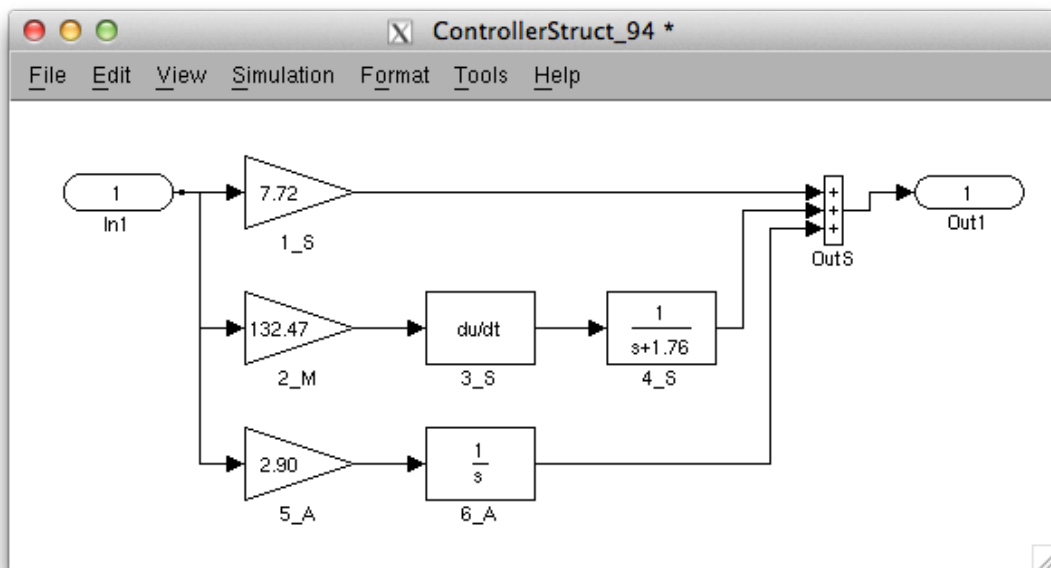


Figura 8-5: Controlador 94 - 2º Menor Tempo de Subida.

8.3.2 Maior Tempo de Subida

Esta seção apresenta os controladores que obtiveram os maiores tempos de subida dentre as soluções otimizadas. O controlador da Figura 8-6 obteve um Tempo de Subida de 6.3161s, Custo de 4.91 e Complexidade 2. O controlador da Figura 8-7 obteve um Tempo de Subida de 1.7409s, Custo de 39868.05 e Complexidade 10.

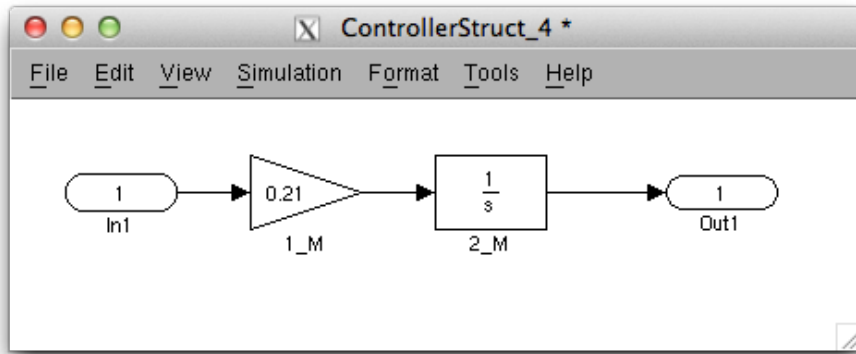


Figura 8-6: Controlador 4 – 1°. Maior Tempo de Subida.

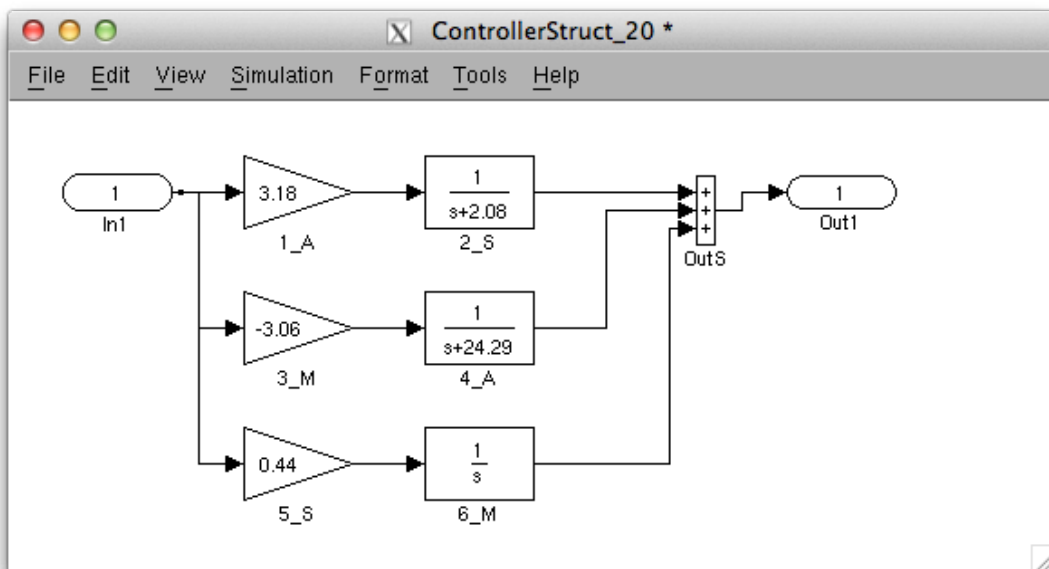


Figura 8-7: Controlador 20 - 2° Maior Tempo de Subida.

8.4 Métrica Custo

Esta seção apresenta os resultados relacionados à métrica de Custo. O resultado desta métrica para todos os controladores pode ser visto na Figura 8-8, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 8-10 e a Figura 8-11 desta seção ilustram os controladores com os menores valores da métrica de Custo. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

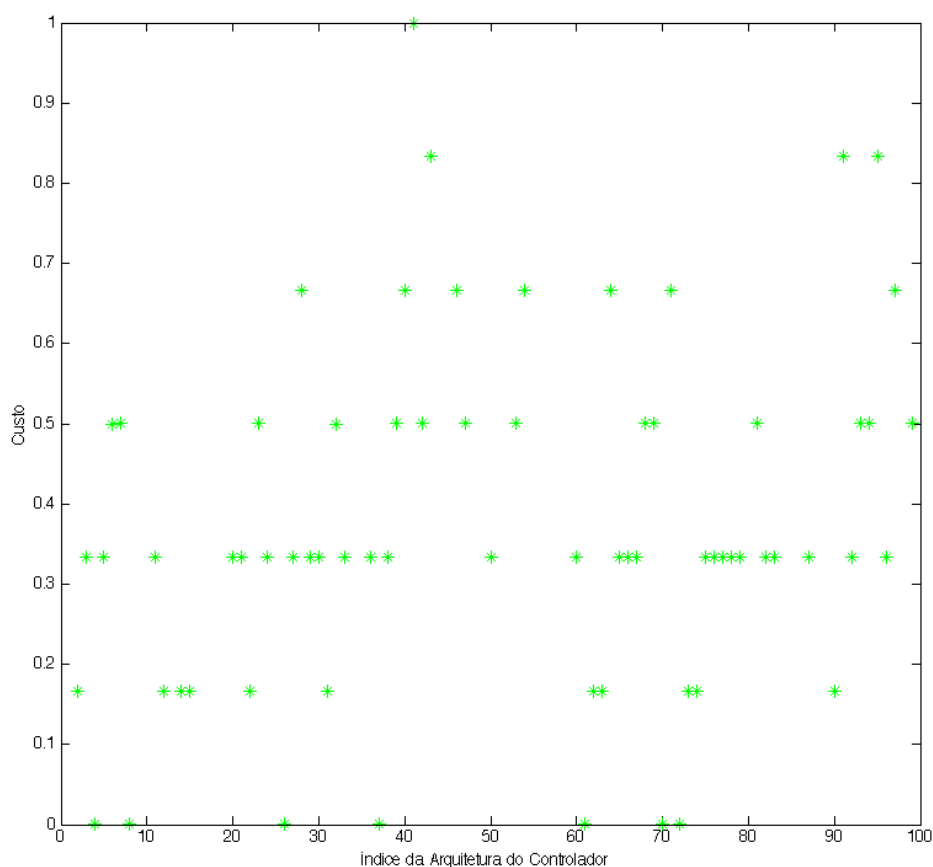


Figura 8-8: Resultados da Métrica de Custo.

8.4.1 Menor Custo

Esta seção apresenta os controladores que obtiveram os menores custos. O controlador 4 apresentado na Figura 8-6 é a estrutura que possui o menor custo, mesmo assim apresentaremos a seguir duas outras alternativas de baixo custo. O

controlador da Figura 8-10 obteve um Tempo de Subida de 0.57487s, Custo de 5.4596 e Complexidade 2. O controlador da Figura 8-11 obteve um Tempo de Subida de 0.40346 s, Custo de 39863.59 e Complexidade 4.

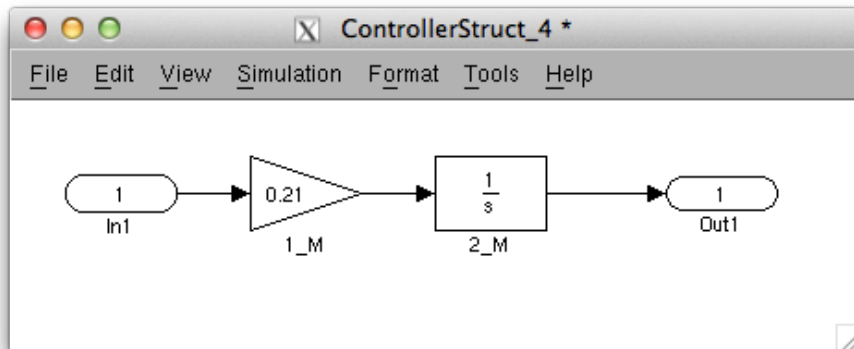


Figura 8-9: Controlador 4 – 1°. Menor Custo.

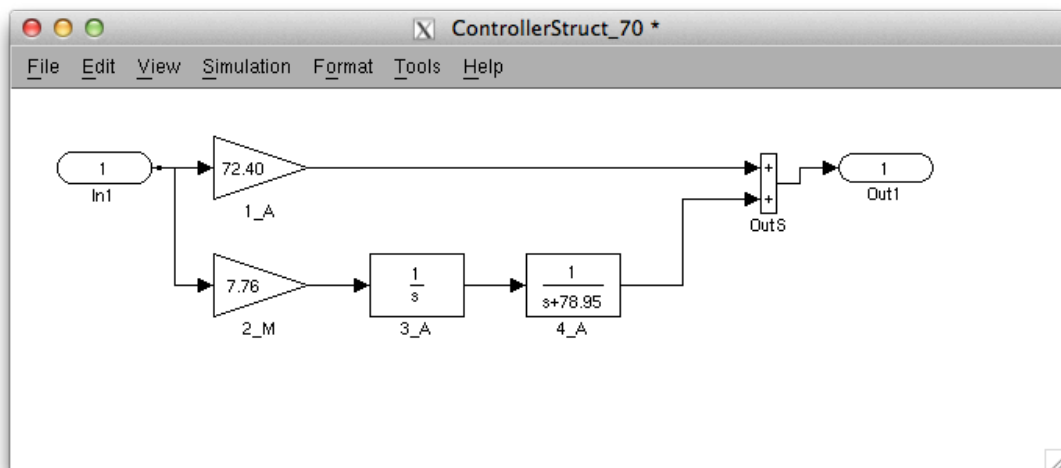


Figura 8-10: Controlador 70 - 2° Menor Custo.

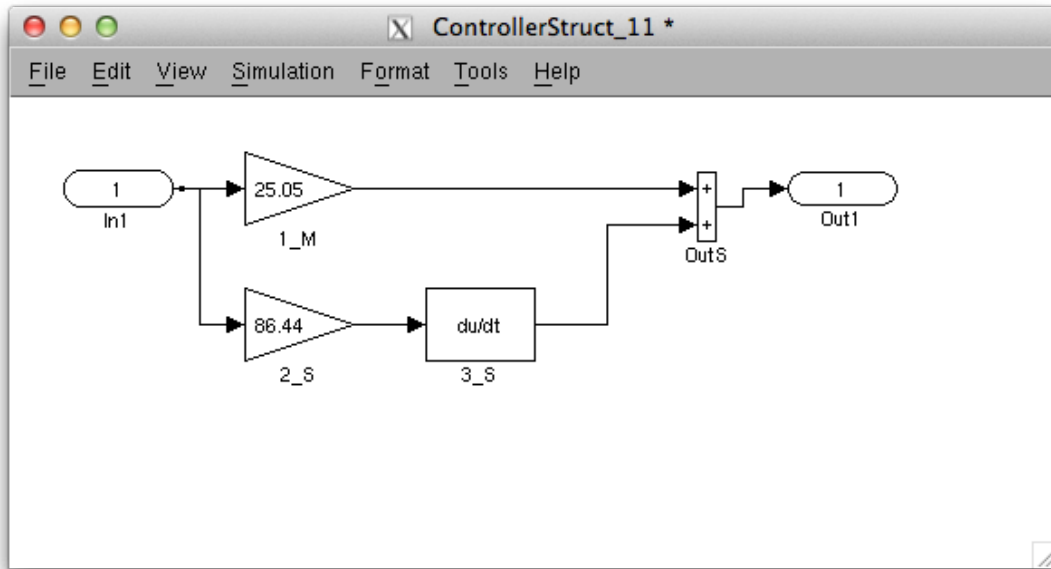


Figura 8-11: Controlador 11 - 3º Menor Custo.

8.4.2 Maior Custo

Esta seção apresenta os controladores que obtiveram os maiores custos. O controlador da Figura 8-12 obteve um Tempo de Subida de 0.42921s, Custo de 119590.32 e Complexidade 57. O controlador da Figura 8-13 obteve um Tempo de Subida de 0.25273 s, Custo de 99660.21 e Complexidade 19.

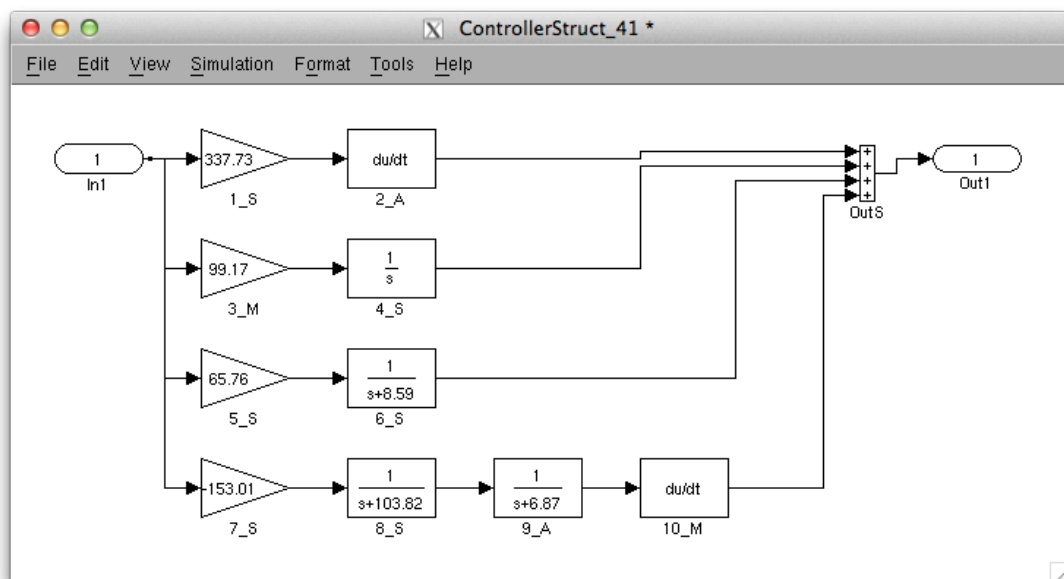


Figura 8-12: Controlador 41 – 1º. Maior Custo.

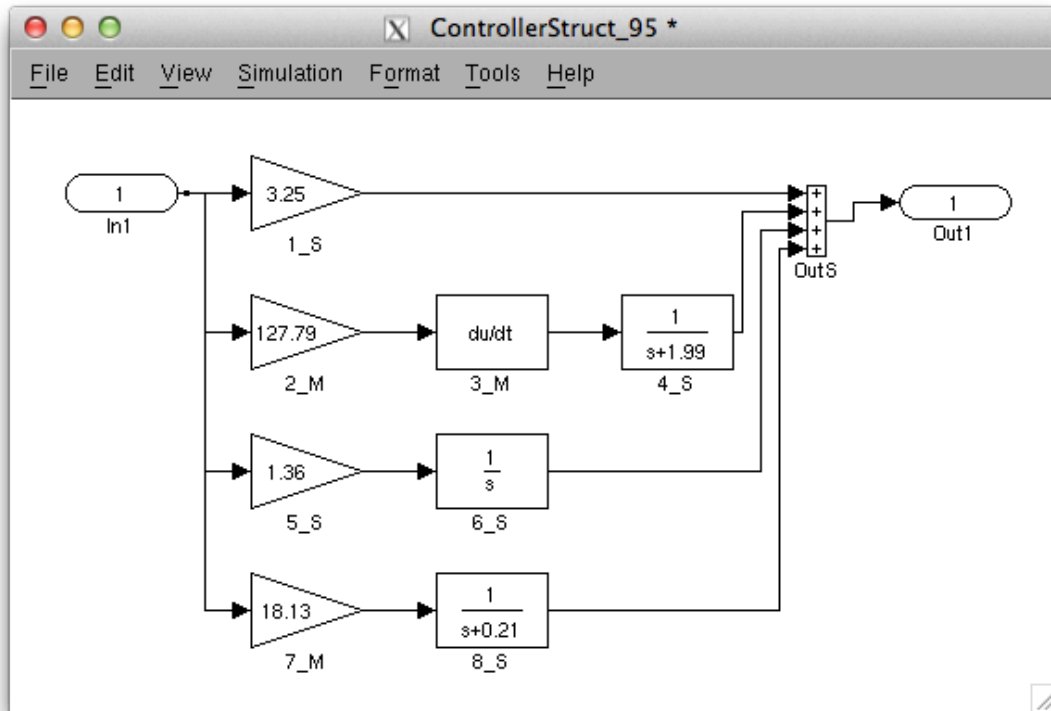


Figura 8-13: Controlador 95 - 2º Maior Custo.

8.5 Métrica Complexidade

Esta seção apresenta os resultados relacionados à métrica de Complexidade. O resultado desta métrica para todos os controladores pode ser visto na Figura 8-14, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 8-16 e a Figura 8-17 desta seção ilustram os controladores com os menores valores da métrica de Complexidade. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

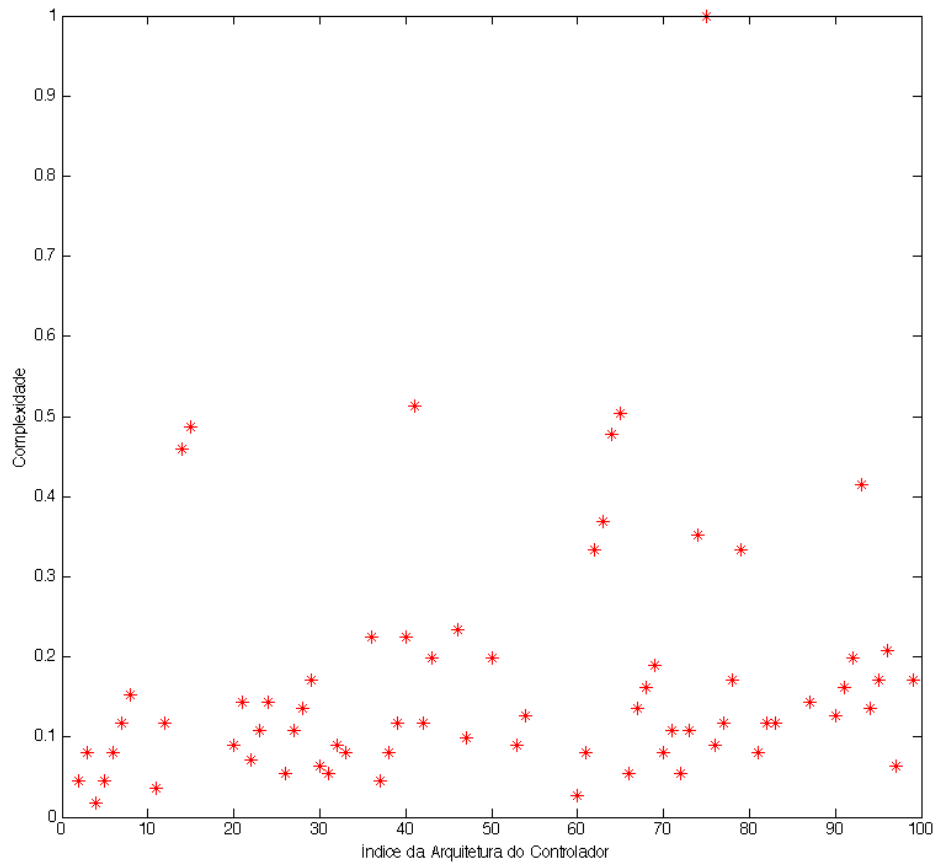


Figura 8-14: Resultados da Métrica de Complexidade.

8.5.1 Menor Complexidade

Esta seção apresenta os controladores que obtiveram as menores complexidades. O controlador 4 apresentado na Figura 8-6 além de possuir o menor custo, também é a estrutura de menor complexidade. Apresentaremos a seguir duas outras alternativas de baixa complexidade. O controlador da Figura 8-16 obteve um Tempo de Subida de 0.99565s, Custo de 39862.13 e Complexidade 3. O controlador da Figura 8-17 obteve um Tempo de Subida de 0.26201 s, Custo de 19935.49 e Complexidade 5.

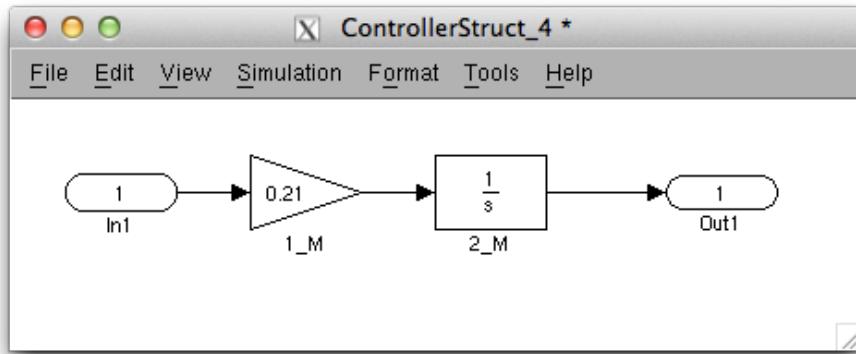


Figura 8-15: Controlador 4 – 1ª. Menor Complexidade.

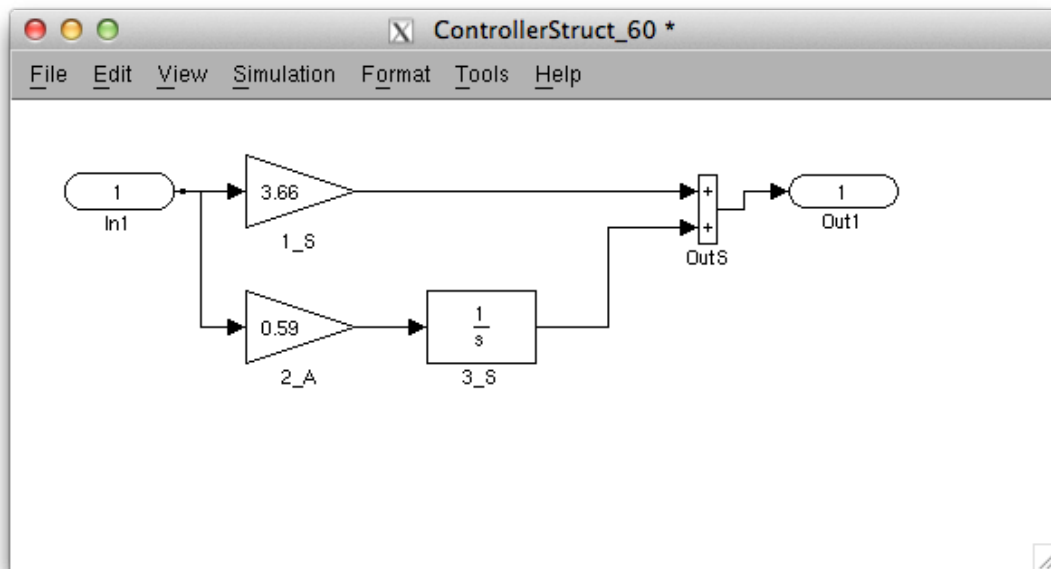


Figura 8-16: Controlador 60 – 2ª. Menor Complexidade.

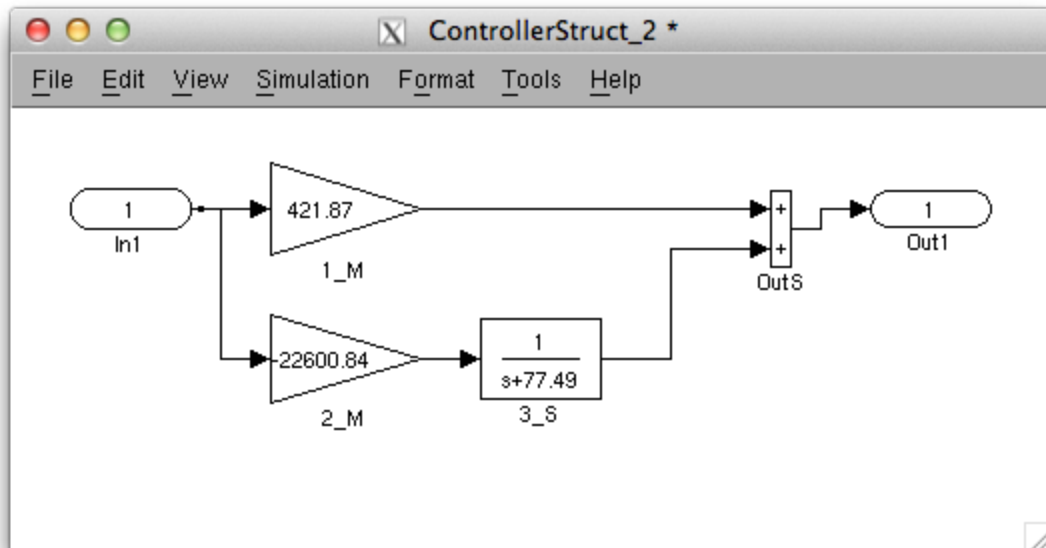


Figura 8-17: Controlador 2 – 4ª. Menor Complexidade.

8.5.2 Maior Complexidade

Esta seção apresenta os controladores que obtiveram as maiores complexidades. O controlador da Figura 8-18 obteve um Tempo de Subida de 0.42257 s, Custo de 39870.05 e Complexidade 111. O controlador da Figura 8-19 obteve um Tempo de Subida de 0.40034 s, Custo de 39874.97 e Complexidade 56.

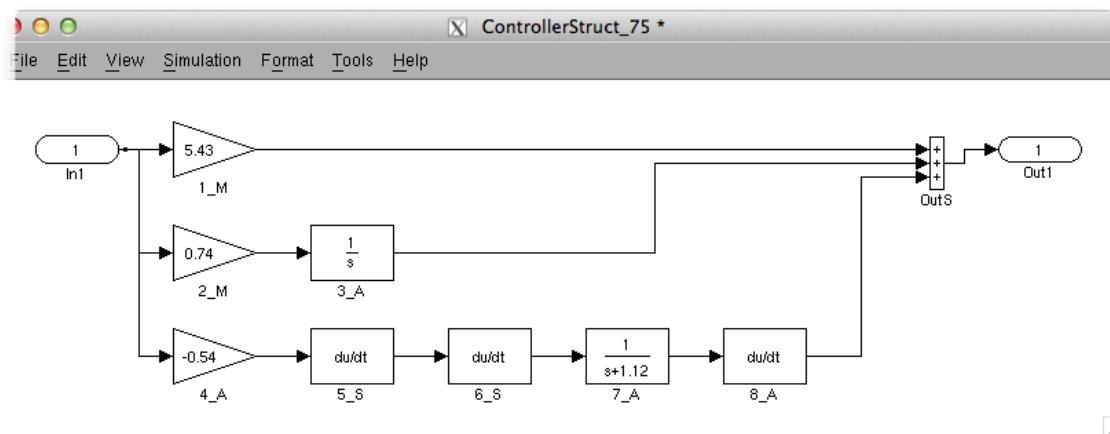


Figura 8-18: Controlador 75 – 1ª. Maior Complexidade.

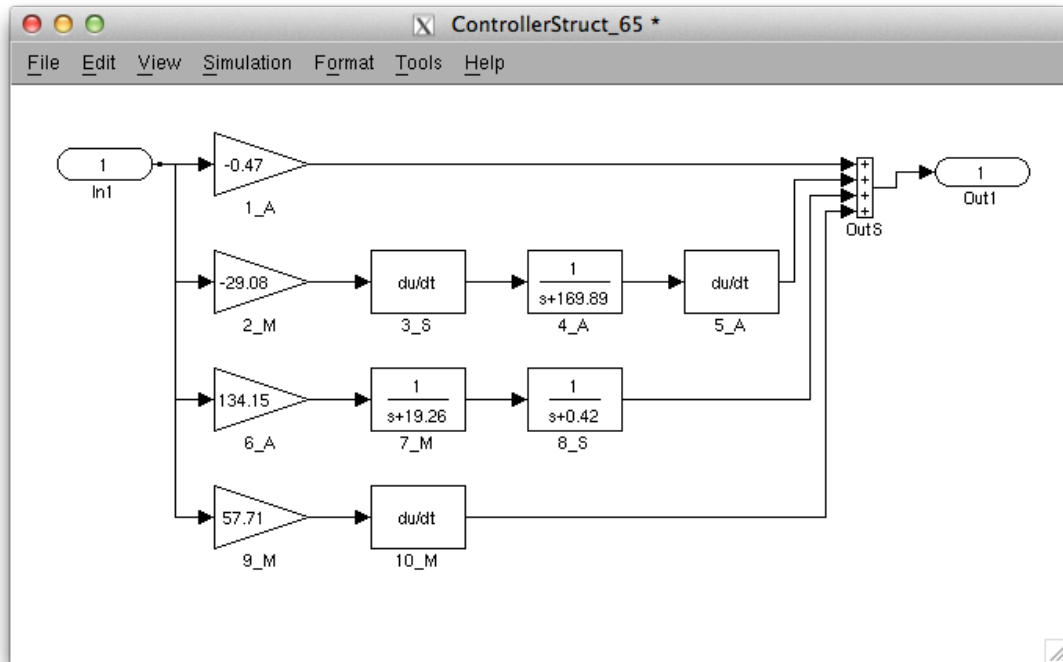


Figura 8-19: Controlador 65 – 2ª. Maior Complexidade.

8.6 Fronteira de Pareto

Dentre as 70 soluções viáveis, há 10 soluções não dominadas candidatas a ótimo que formam a Fronteira de Pareto, Figura 8-20. São os seguintes controladores: 2, 4, 11, 12, 37, 60, 66, 67, 70 e 72. Deste conjunto, vários já foram apresentados anteriormente neste capítulo. Serão exibidos a seguir aqueles ainda não vistos.

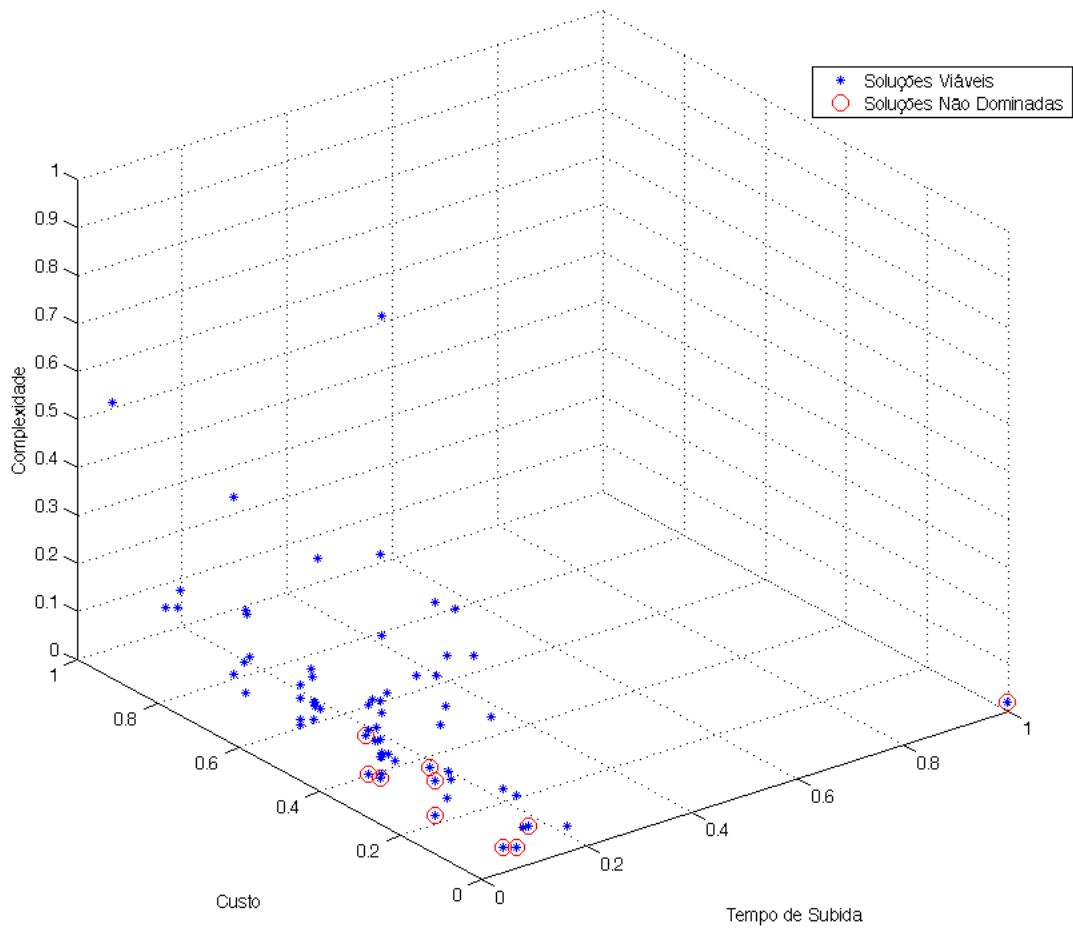


Figura 8-20: Espaço de Soluções e a Fronteira de Pareto.

O controlador da Figura 8-21 obteve um Tempo de Subida de 0.25414 s, Custo de 19936.49 e Complexidade 13. O controlador da Figura 8-22 obteve um Tempo de Subida de 0.42791 s, Custo de 6.92 e Complexidade 5.

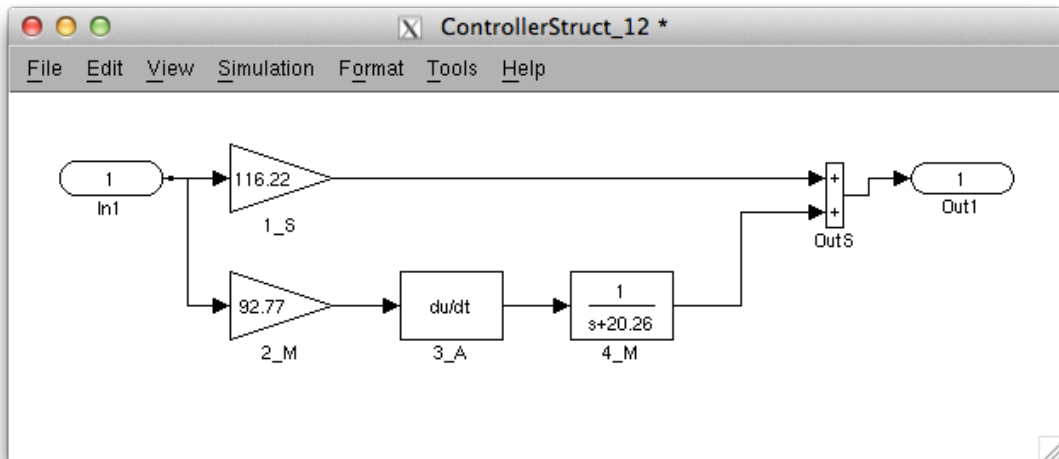


Figura 8-21: Controlador 12 - Solução Não Dominada.

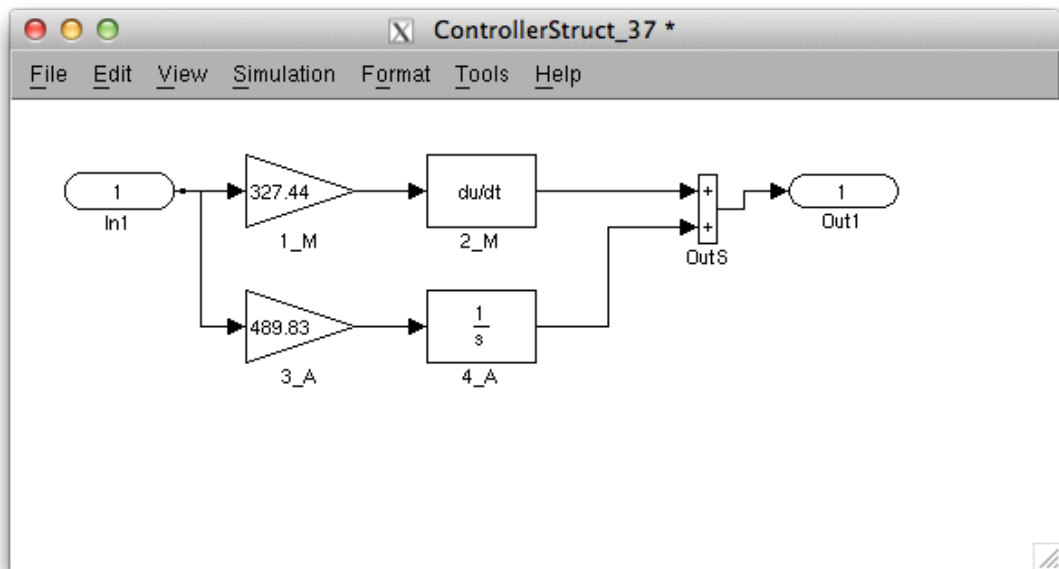


Figura 8-22: Controlador 37 - Solução Não Dominada.

O controlador da Figura 8-23 obteve um Tempo de Subida de 0.25923 s, Custo de 39867.05 e Complexidade 6. O controlador da Figura 8-24 obteve um Tempo de Subida de 0.2608 s, Custo de 10.8384 e Complexidade 6.

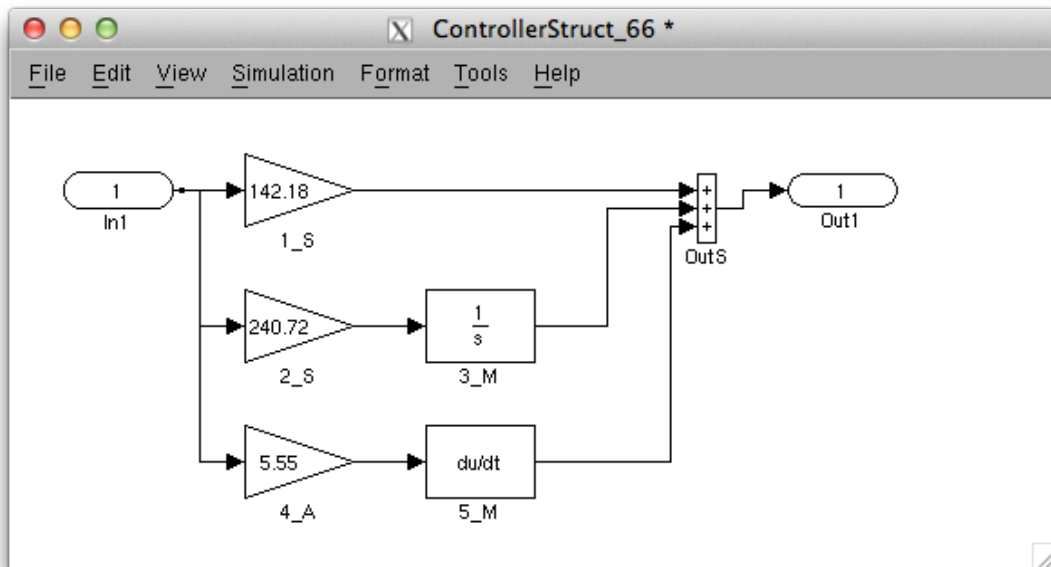


Figura 8-23: Controlador 66 - Solução Não Dominada.

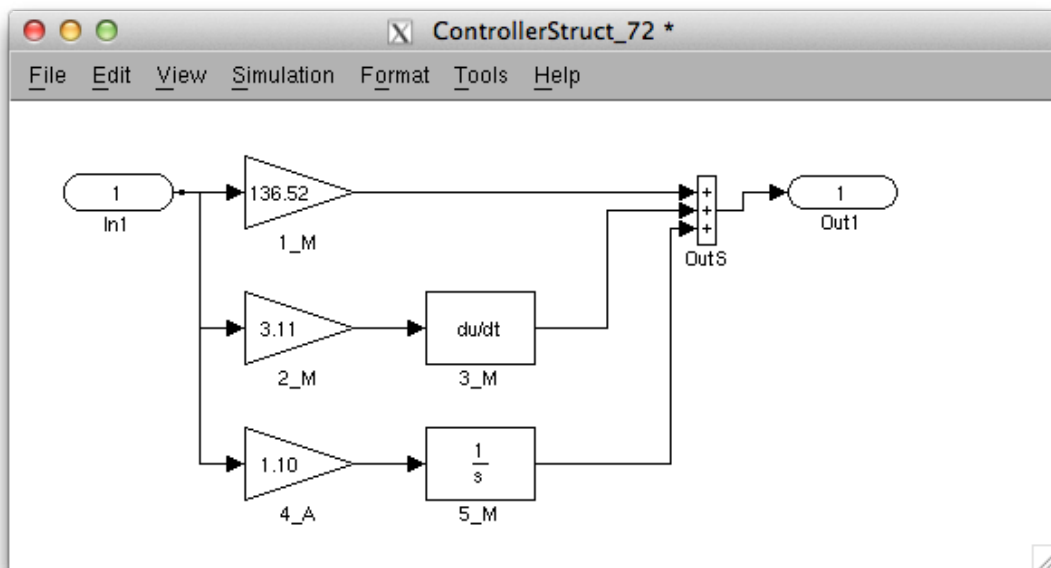


Figura 8-24: Controlador 72 - Solução Não Dominada.

8.7 Seleção pelo Critério da Menor Perda

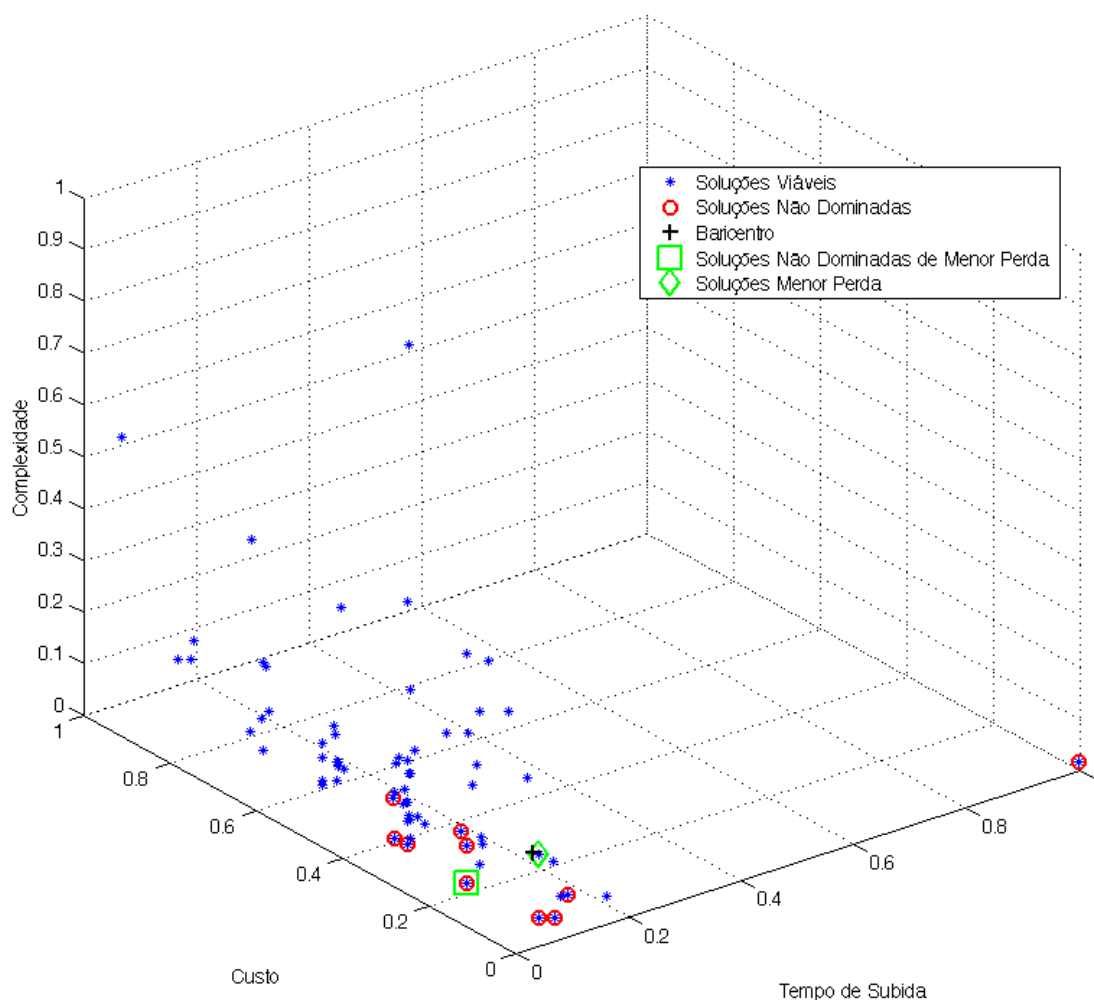


Figura 8-25: Espaço de Soluções e Critério de Menor Perda.

O baricentro em um espaço normalizado foi (Figura 8-25):

Tempo de Subida = 0.1581, Custo = 0.1667 e Confiabilidade = 0.0613.

A distância entre o baricentro e a solução na Fronteira de Pareto mais próxima é de 0.1178 (Controlador 2, Figura 8-17). A solução de Menor Perda é o Controlador 31 (Figura 8-26), com a menor distância entre sua posição e o baricentro no espaço de soluções, de 0.0128. A solução de Menor Perda obteve um Tempo de Subida de 1.066s, Custo de 19936.49 e Complexidade 6, sua função de transferência é:

$$C(s)_{31} = \frac{72.34 s + 13.81}{s^2 + 23.05 s} \quad (8.1)$$

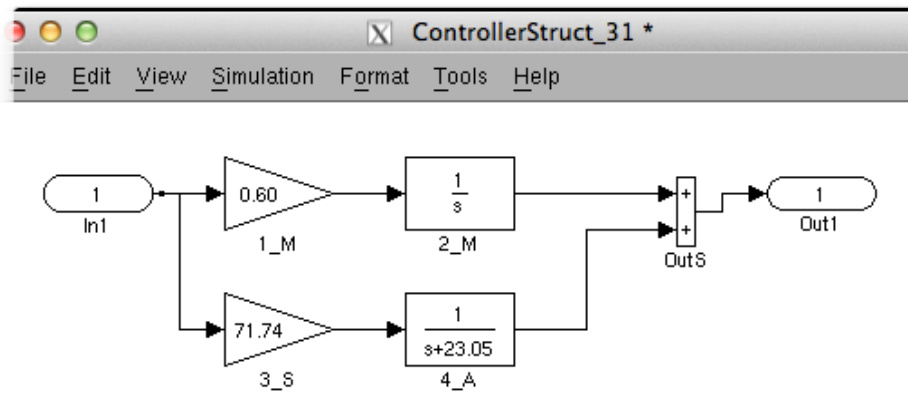


Figura 8-26: Solução de Menor Perda.

A Figura **8-27** apresenta a resposta a uma entrada a degrau unitário utilizando o Controlador 31. Note-se que: a variável de saída, gráfico superior, não ultrapassa os 2% de sobressinal; e o sinal de controle, gráfico inferior, não viola os limites de -40V e +40V. A Figura **8-28** apresenta a resposta em frequência, por meio de diagramas de Bode. Percebe-se que estes não violam as restrições impostas, representadas pelas linhas tracejadas.

Pode-se observar que a resposta do sistema atende às restrições impostas e apresenta uma solução equilibrada no ponto de vista de Tempo de Subida, Custo e Complexidade.

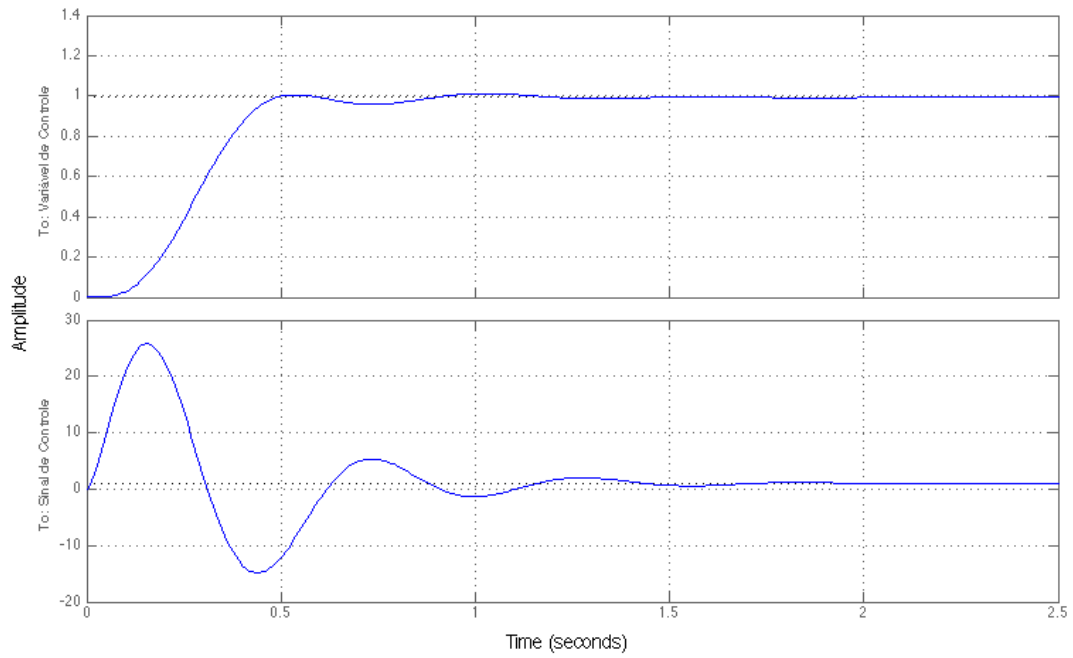


Figura 8-27: Resposta a uma entrada a Degrau Unitário da Solução de Menor Perda.

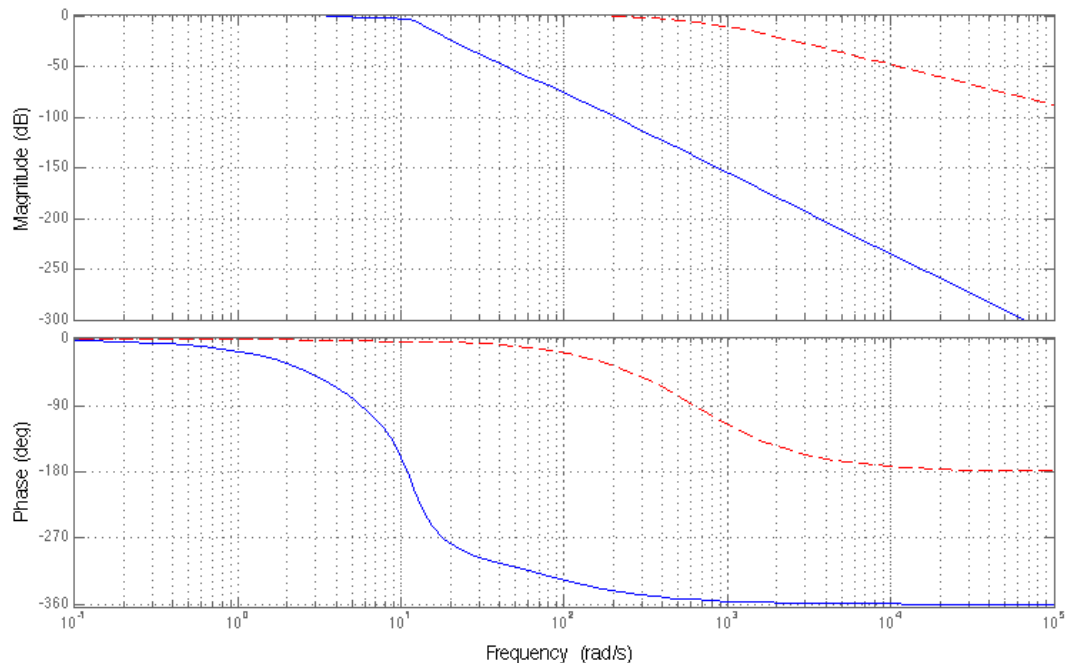


Figura 8-28: Diagramas de Bode da Solução de Menor Perda.

8.8 Comparação com Resultados da Literatura

Utilizaremos os trabalhos de (KOZA; KEANE et al., 1999) e a referência oferecida no trabalho de (DORF; BISHOP, 1998), já mencionados anteriormente, para elaborar uma comparação de resultados. Esse trabalho foi escolhido por realizar uma busca por controladores utilizando modificações na sua arquitetura; e por oferecer uma referência para comparação de resultados. Dentre todos os trabalhos revisados, o artigo de (KOZA; KEANE et al., 1999) é aquele que mais se assemelha à busca realizada nesta tese.

Mesmo com a semelhança mencionada anteriormente, vale a pena ressaltar os principais aspectos diferentes entre os dois trabalhos, descritas na Tabela 8-1.

Como os trabalhos (KOZA; KEANE et al., 1999) e (DORF; BISHOP, 1998) fazem uma busca pela resposta com o menor tempo de subida, será utilizado nessa comparação o Controlador 67 (Figura 8-4), que é a estrutura que obteve o menor tempo de subida.

Tabela 8-1: Principais diferenças entre o trabalho de Koza e Keane et al. (1999) e este.

Aspectos	Trabalho (KOZA; KEANE et al., 1999)	Este Trabalho
Objetivo	Minimizar Tempo de Subida.	Encontrar um solução equilibrada entre métricas de Engenharia de Controle e métricas de Engenharia de Sistemas.
Bloco de Saturação	Utiliza um bloco de saturação antes da entrada da planta.	Não gera sinais de controle maiores que os limites impostos.
Causalidade	Controlador não causal, numerador de 3° ordem e denominador de 1° ordem.	Todos os controladores gerados pelo método são causais.
Processamento	Resultado de 44.5 horas em um sistema computacional paralelo de 66 núcleos de 533MHz cada, sistema operacional Linux.	Resultado de 7 horas em um processador 1.86 GHz Intel Core 2 Duo, com o Sistema Operacional Mac OS X 10.7.5.
Quantidade de Estruturas	Foram processadas 66.000 estruturas.	Foram processadas 100 estruturas.

Tabela 8-2 apresenta os pré-filtros utilizados nos resultados apresentados. Optou-se por utilizar um dos filtros já especificados, neste caso o mesmo utilizado na solução

de (DORF; BISHOP, 1998). A **Tabela 8-3** apresenta as funções de transferência das soluções dos três trabalhos. Dentre todas as soluções, a única solução causal é a solução gerada neste trabalho.

Tabela 8-2: Comparativo entre as funções de transferência dos pré-filtros utilizados.

Trabalho	Pré-Filtro Utilizado
(KOZA; KEANE et al., 1999)	$\frac{0.0256 s^2 + 0.329 s + 1}{0.0000451 s^5 + 0.00207 s^4 + 0.034 s^3 + 0.253 s^2 + 0.846 s + 1}$
(DORF; BISHOP, 1998)	$\frac{42.67}{s^2 + 11.38 s + 42.67}$
Este trabalho	$\frac{42.67}{s^2 + 11.38 s + 42.67}$

Tabela 8-3: Comparativo Entre as Funções de Transferência dos Controladores.

Trabalho	Controlador
(KOZA; KEANE et al., 1999)	$\frac{1.243 s^3 + 71.252 s^2 + 1.3e3 s + 7487}{s}$
(DORF; BISHOP, 1998)	$\frac{12(s^2 + 11.38 s + 42.67)}{s}$
Este trabalho	$\frac{130.1s^3 + 3.135e5 s^2 + 4.031e6 s + 1.079e7}{s^3 + 1028 s^2 + 2.794e4 s}$

A Figura **8-29** apresenta a resposta temporal do sistema a uma entrada a degrau unitário. Percebe-se que: a variável de saída, gráfico superior, não ultrapassa os 2% de sobressinal; e o sinal de controle, gráfico inferior, não viola os limites de -40V e +40V. A Figura **8-30** apresentam a resposta em frequência do sistema e esta não viola as restrições impostas, representadas pelas linhas tracejadas.

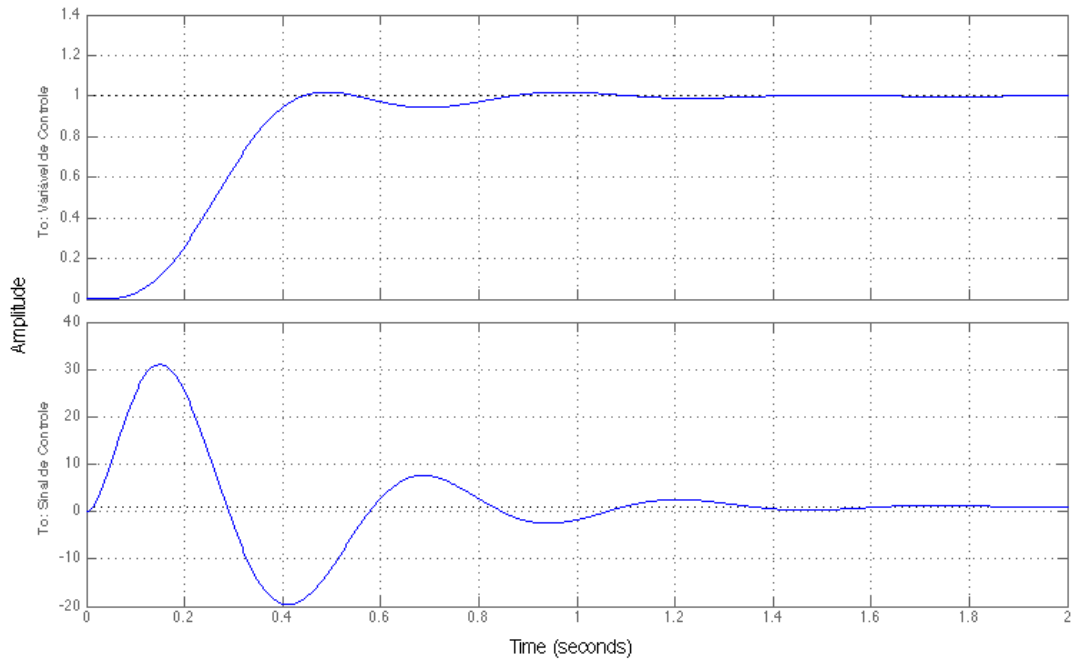


Figura 8-29: Resposta ao degrau unitário da solução de menor tempo de subida.

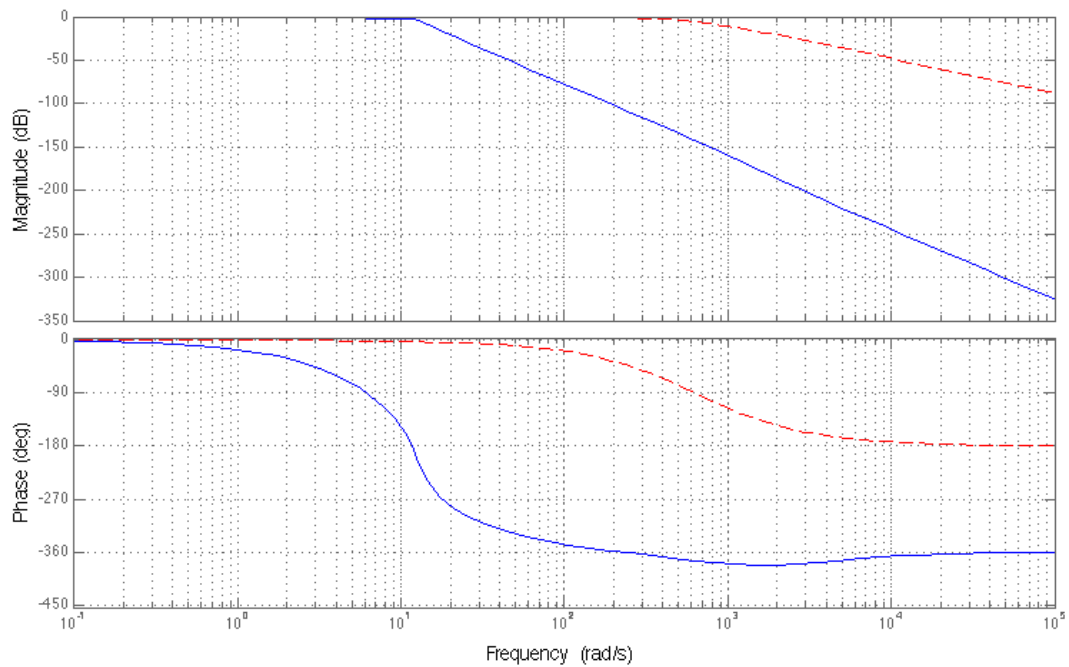


Figura 8-30: Diagramas de bode da solução de menor tempo de subida.

A comparação entre as respostas a uma entrada a degrau unitário dos três trabalhos pode ser vista na Figura **8-31**.

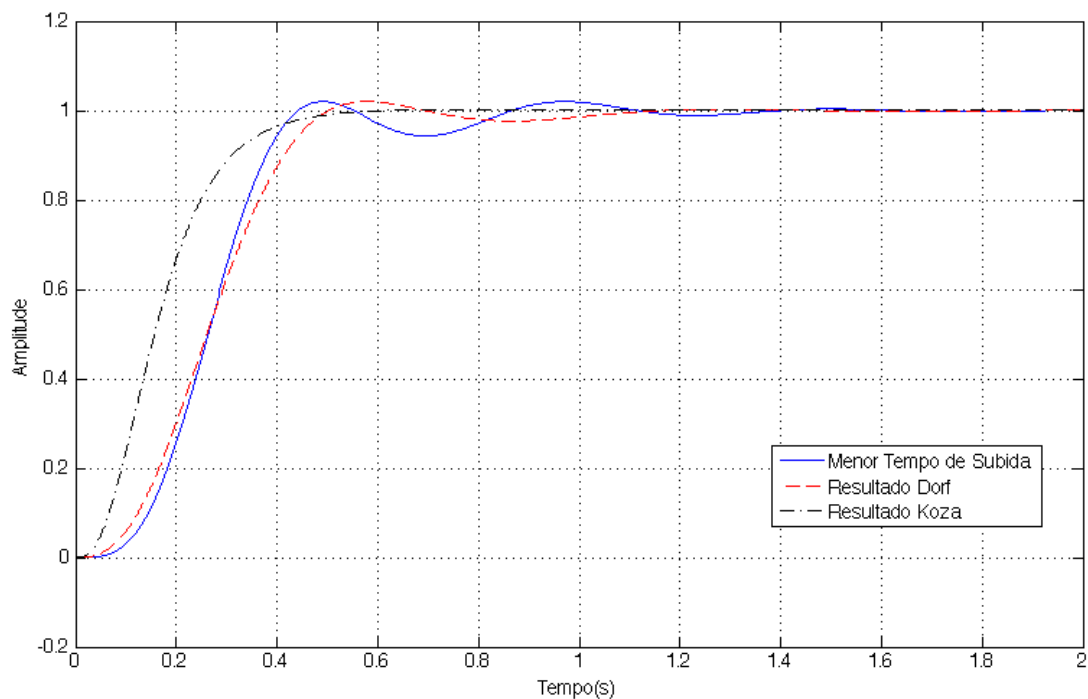


Figura 8-31: Comparação da arquitetura de menor tempo de subida com a literatura.

O trabalho de Koza e Keane et al. (1999) afirma que seu tempo de subida foi de 296ms e o tempo de subida de Dorf e Bishop (1998) foi de 465ms. Ao calcular novamente estes valores utilizando as funções de transferência informadas utilizando a máxima precisão disponível os resultados encontrados foram diferentes dos informados e estes podem ser encontrados na Tabela 8-4. Nesta tabela encontram-se os resultados dos tempos de subida utilizando diversos critérios para seu cálculo. É importante notar que todas as otimizações realizadas neste trabalho levaram em consideração o critério de 10% a 90% do valor acomodação.

Tabela 8-4: Comparativo entre os tempos de subida.

Critério do Tempo de Subida	Tempo de Subida		
	Trabalho (DORF; BISHOP, 1998)	Trabalho (KOZA; KEANE et al., 1999)	Este Trabalho
10% a 90%	289ms	244ms	235ms*
0% a 97%	466ms	415ms*	415ms*
0% a 88%	403ms	296ms*	371ms

* Menor valor

9 RESULTADOS DA TERCEIRA INVESTIGAÇÃO (I3)

9.1 Descrição

Este capítulo apresenta os resultados da geração de arquiteturas de controladores, de acordo com o método descrito no Capítulo 5 e 0. Estes dados são resultados da geração de 100 arquiteturas que após o ajuste do controlador resultaram em 42 arquiteturas de controladores estáveis que não violam nenhuma das restrições impostas pelo problema da Investigação I3.

Nos gráficos que ilustram diagrama de blocos, abaixo de cada bloco há uma indicação do número identificador do bloco seguido do seu nível de confiabilidade, *a* para automotivo, *m* para militar e *s* para espacial.

Esses resultados foram gerados utilizando somente os seguintes tipos de blocos: Proporcional, Integral, Derivativo e um Polo Real.

9.2 Resultados Gerais

A Figura 9-1 traz uma visão ampla das métricas obtidas para todos os 42 controladores gerados viáveis em uma execução do algoritmo de elaboração de controladores.

O ajuste das 100 arquiteturas geradas foi efetuado em um computador com processador de 1.86 GHz Intel Core 2 Duo, com o Sistema Operacional Mac OS X 10.7.5. Cada arquitetura levou em média 5 minutos para ser ajustada, totalizando em aproximada 8h21' para os 100 controladores.

As arquiteturas e os resultados serão analisados de maneira relativa entre eles, ou seja, serão comparados os máximos e mínimos para cada medida. Essa maneira se mostrou útil para o entendimento dos resultados.

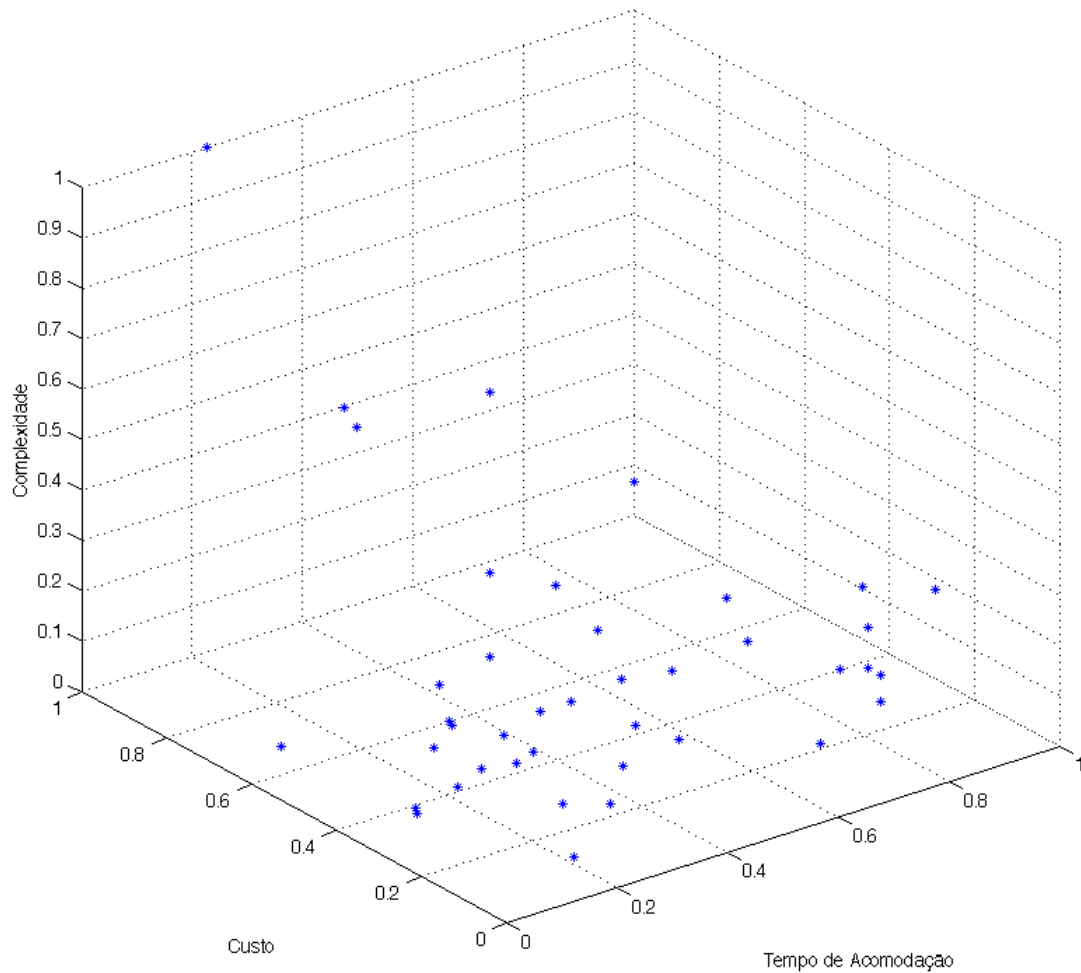


Figura 9-1: Espaço de soluções com todas as 42 soluções aceitáveis.

9.3 Métrica Tempo de Acomodação

Esta seção apresenta os resultados relacionados à métrica de tempo de acomodação. O resultado desta métrica para todos os controladores pode ser visto na Figura 9-2, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 9-3 e a Figura 9-4 desta seção ilustram os controladores com os menores valores da métrica de Tempo de Acomodação. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

O Tempo de Acomodação utilizado neste trabalho foi o período de tempo necessário para a resposta atingir e permanecer dentro de uma faixa de 2% do valor final.

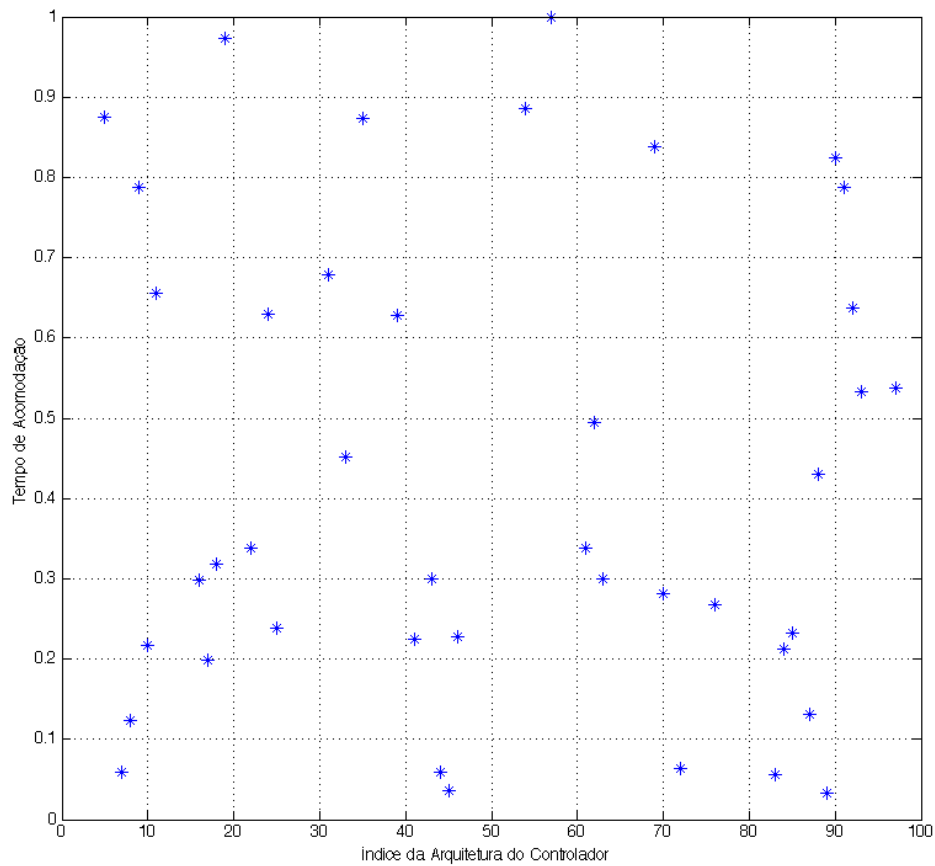


Figura 9-2: Resultados da métrica de tempo de acomodação.

9.3.1 Menor Tempo de Acomodação

Esta seção apresenta os controladores com os menores tempos de acomodação. O controlador da Figura 9-3 obteve um Tempo de Acomodação de 0.16078s, Custo de 79727.19 e Complexidade 15. O controlador da Figura 9-4 obteve um tempo de acomodação de 0.17779s, Custo de 59800.62 e Complexidade 157.

Pode se observar que estes controladores possuem arquiteturas similares, um PID com um polo após a saída do bloco de derivação.

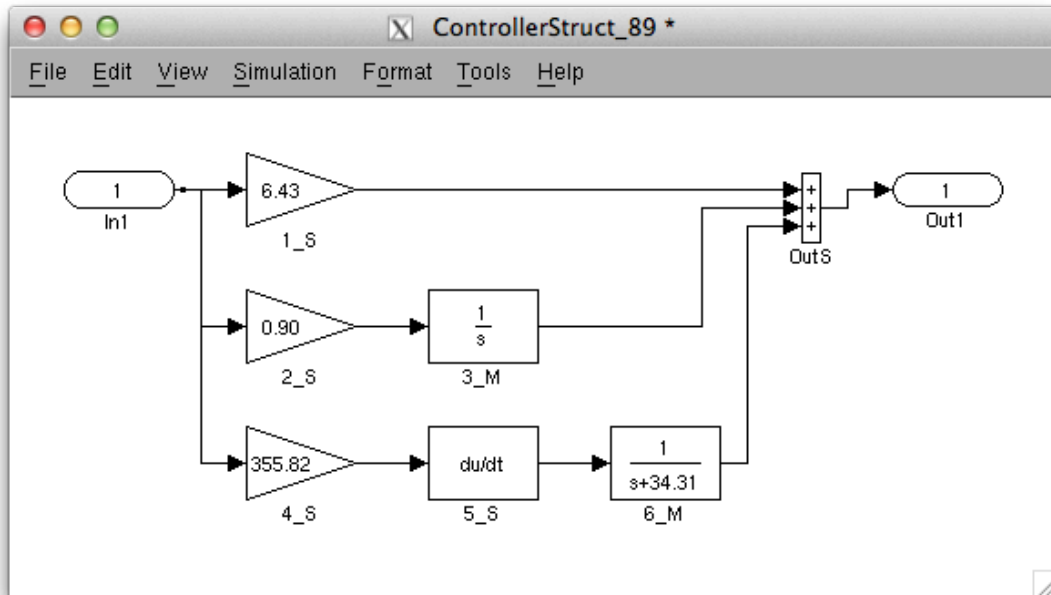


Figura 9-3: Controlador 89 – 1º menor tempo de acomodação.

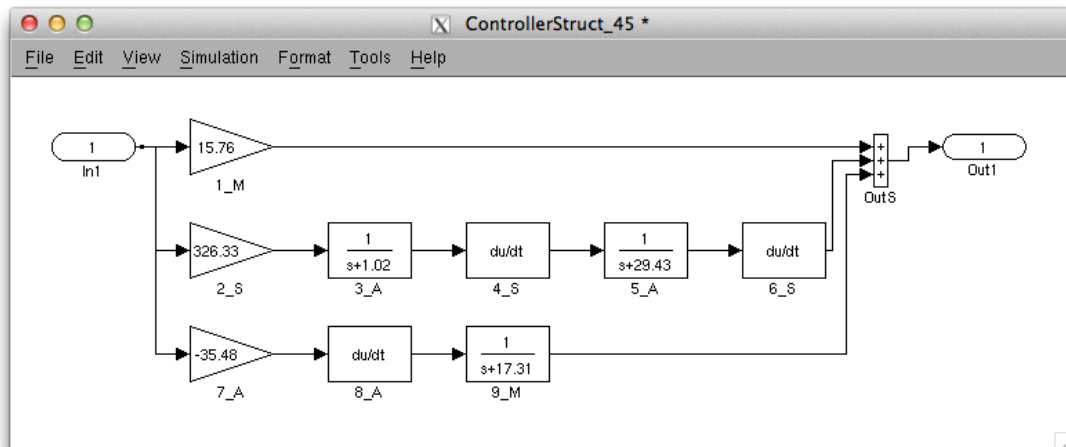


Figura 9-4: Controlador 45 - 2º menor tempo de acomodação.

9.3.2 Maior Tempo de Acomodação

Esta seção apresenta os controladores que obtiveram os maiores tempos de acomodação dentre as soluções otimizadas. O controlador da Figura 9-5 obteve um Tempo de Acomodação de 4.9605 s, Custo de 139513.97 e Complexidade 13. O controlador da Figura 9-6 obteve um Tempo de Acomodação de 4.826 s, Custo de 59799.62 e Complexidade 25.

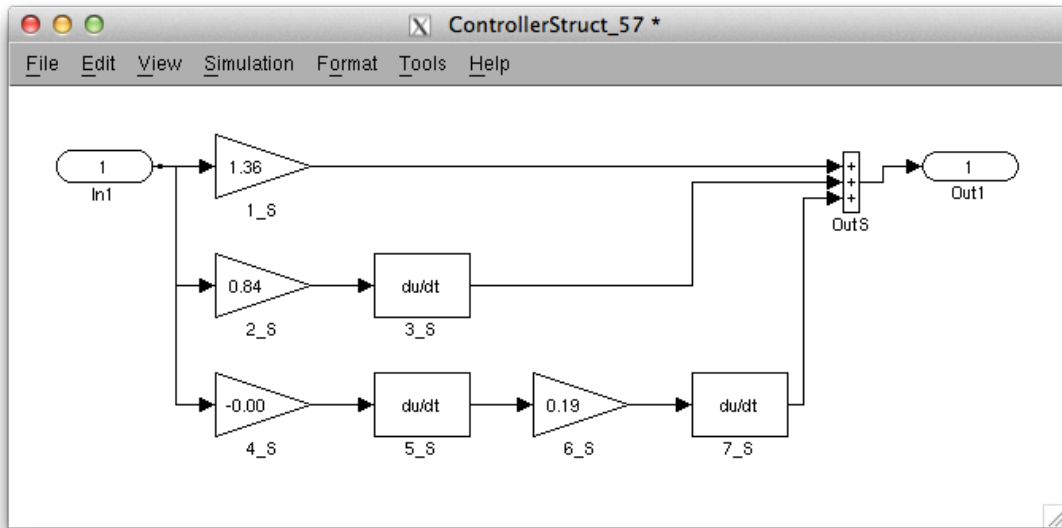


Figura 9-5: Controlador 57 – 1º maior tempo de acomodação.

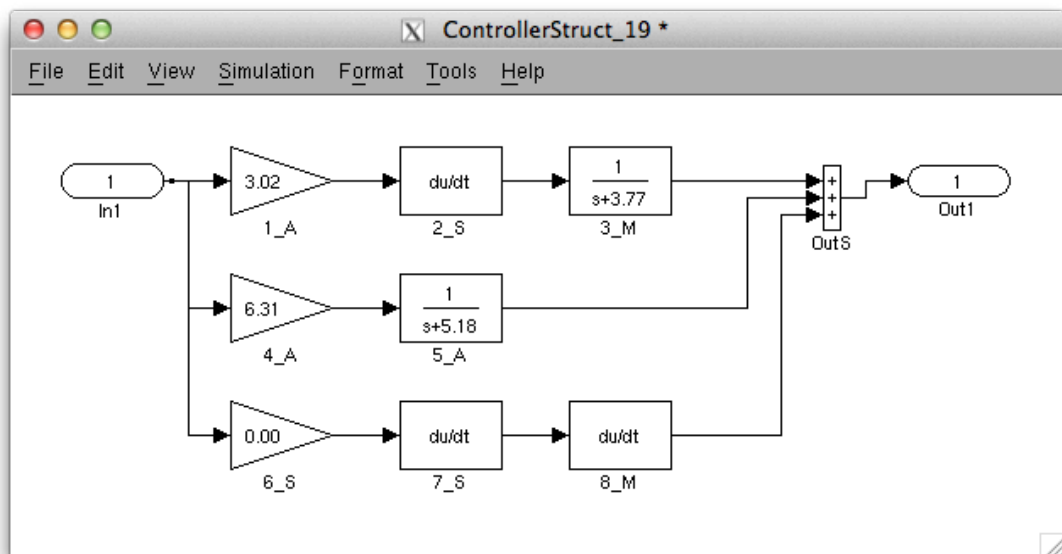


Figura 9-6: Controlador 19 - 2º maior tempo de acomodação.

9.4 Métrica Custo

Esta seção apresenta os resultados relacionados à métrica de Custo. O resultado desta métrica para todos os controladores pode ser visto na Figura 9-7, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 9-8 e a Figura 9-9 desta seção ilustram os controladores com os menores valores para a métrica de Custo. Para efeito de comparação, também serão apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

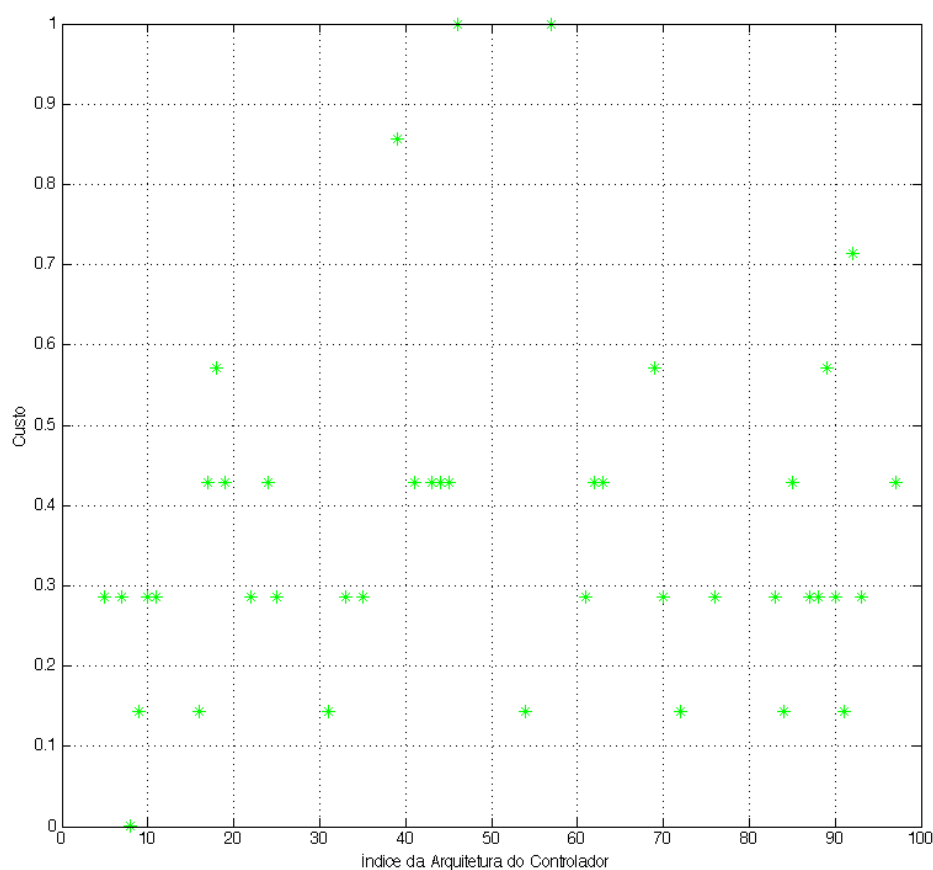


Figura 9-7: Resultados da métrica de custo.

9.4.1 Menor Custo

Esta seção apresenta os controladores que obtiveram os menores custos. O controlador da Figura 9-8 obteve um Tempo de Acomodação de 0,6107 s, Custo de

11.84 e Complexidade 17. O controlador da Figura 9-9 obteve um Tempo de Acomodação de 1.4808 s, Custo de 19936.49 e Complexidade 13.

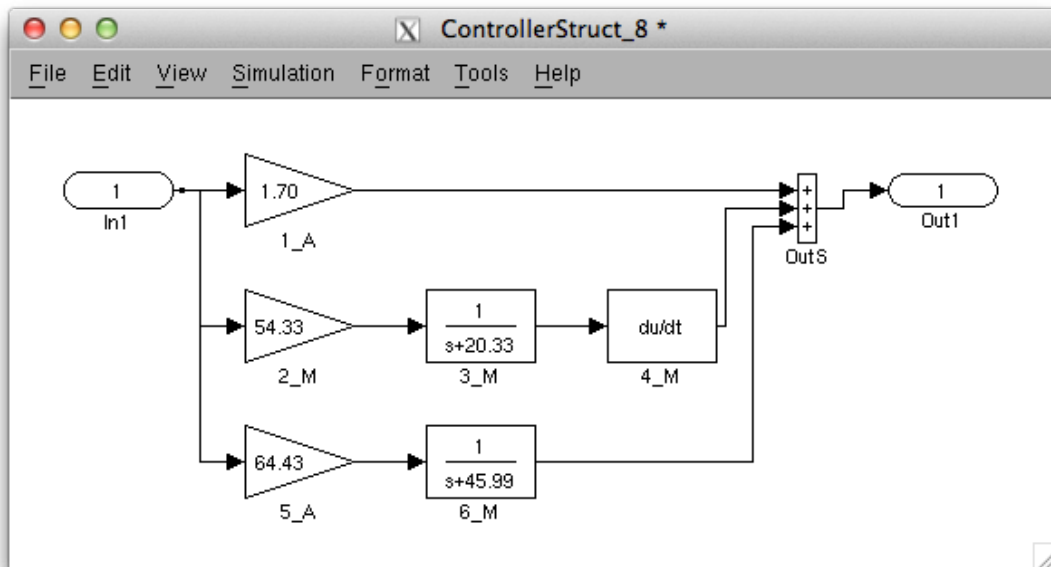


Figura 9-8: Controlador 8 – 1º menor custo.

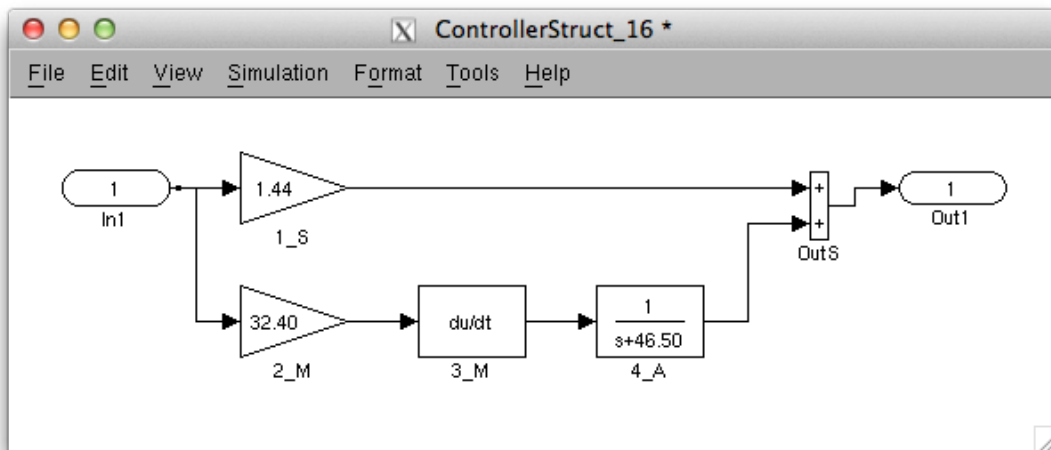


Figura 9-9: Controlador 16 - 2º menor custo.

9.4.2 Maior Custo

Esta seção apresenta os controladores que obtiveram os maiores custos. O controlador da Figura 9-10 obteve um Tempo de Acomodação de 1.1276 s, Custo de 139519.89 e Complexidade 193. O controlador da Figura 9-11 obteve um Tempo de Acomodação de 4.9605 s, Custo de 139513.97 e Complexidade 13.

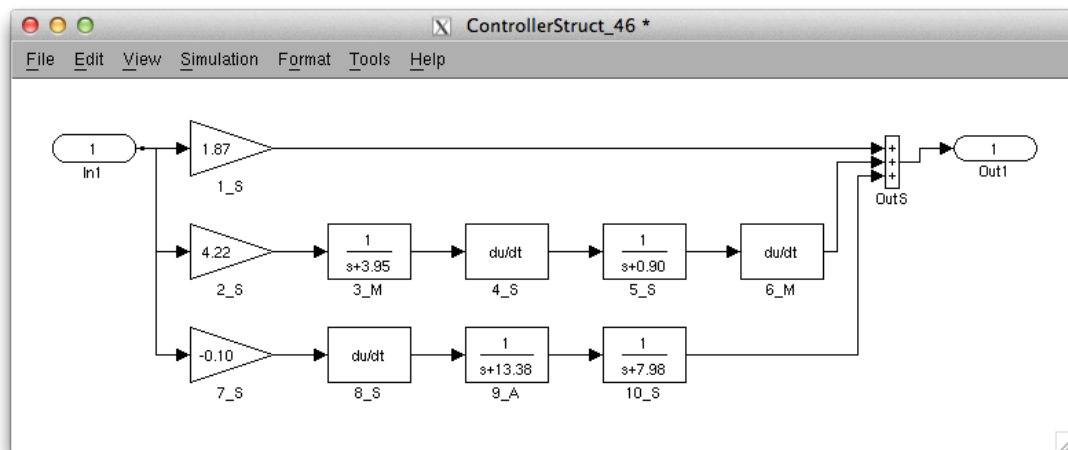


Figura 9-10: Controlador 46 – 1º maior custo.

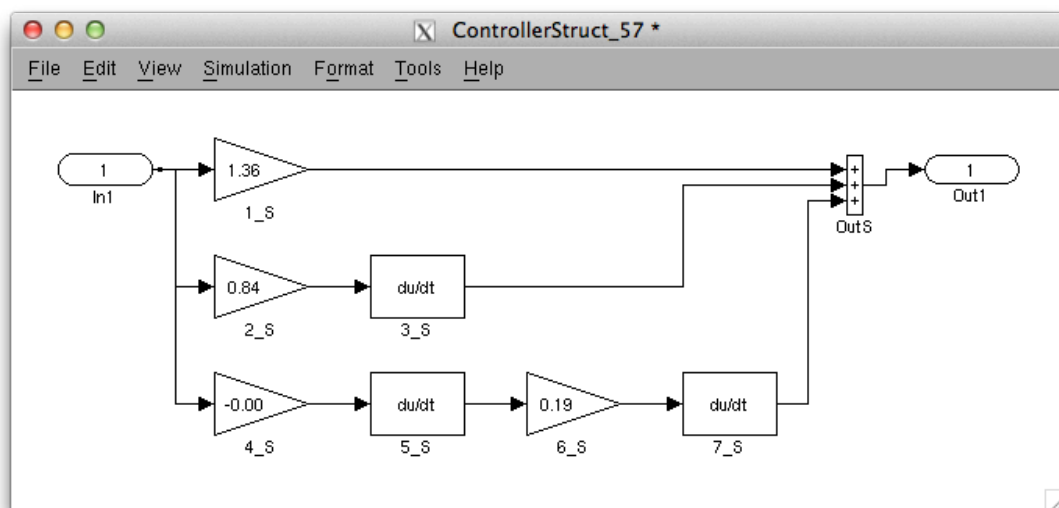


Figura 9-11: Controlador 57 - 2º maior custo.

9.5 Métrica Complexidade

Esta seção apresenta os resultados relacionados à métrica de Complexidade. O resultado desta métrica para todos os controladores pode ser visto na Figura 9-12, onde seus valores estão normalizados pelo valor máximo obtido.

A Figura 9-13 e a Figura 9-14 desta seção ilustram os controladores com os menores valores para a métrica de Complexidade. Para efeito de comparação, também serão

apresentadas algumas arquiteturas com os maiores valores para a mesma métrica.

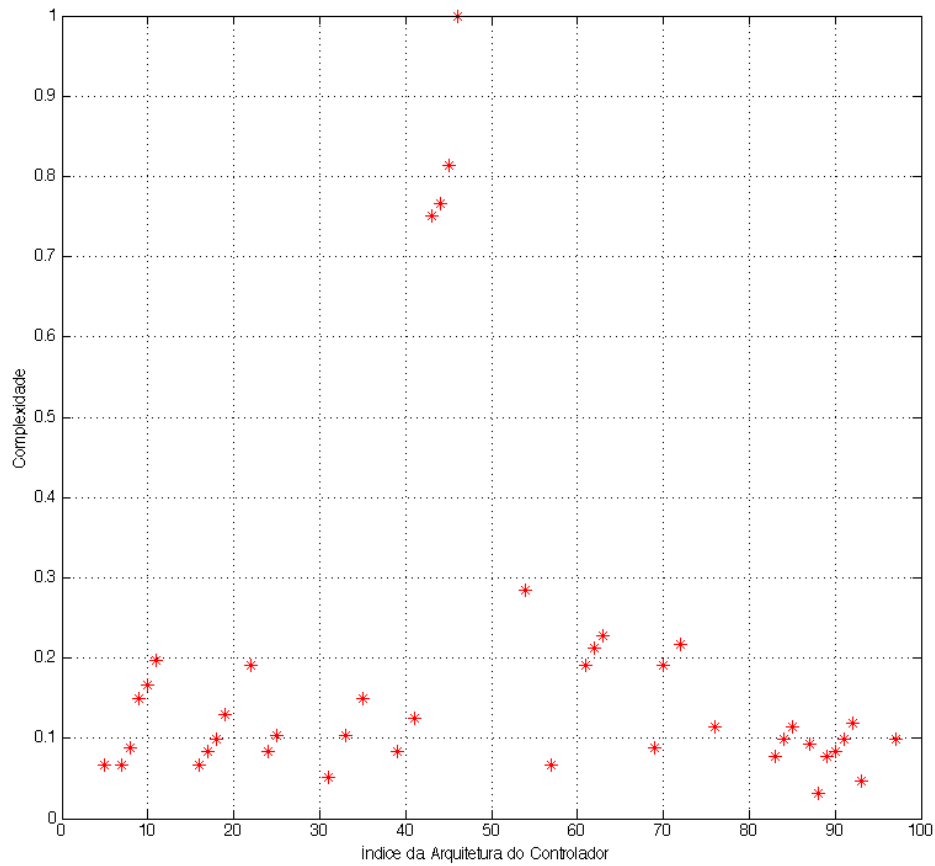


Figura 9-12: Resultados da métrica de complexidade.

9.5.1 Menor Complexidade

Esta seção apresenta os controladores que obtiveram as menores complexidades. O controlador da Figura 9-13 obteve um Tempo de Acomodação de 2.1369 s, Custo de 39865.59 e Complexidade 6. O controlador da Figura 9-14 obteve um Tempo de Acomodação de 2.6443 s, Custo de 39870.51 e Complexidade 9.

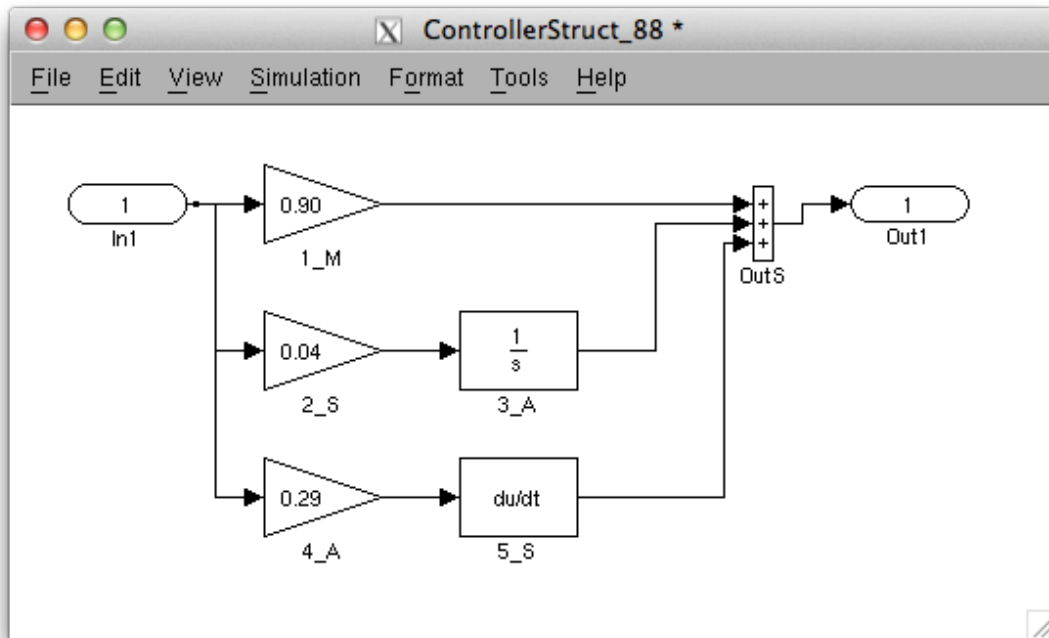


Figura 9-13: Controlador 88 – 1ª menor complexidade.

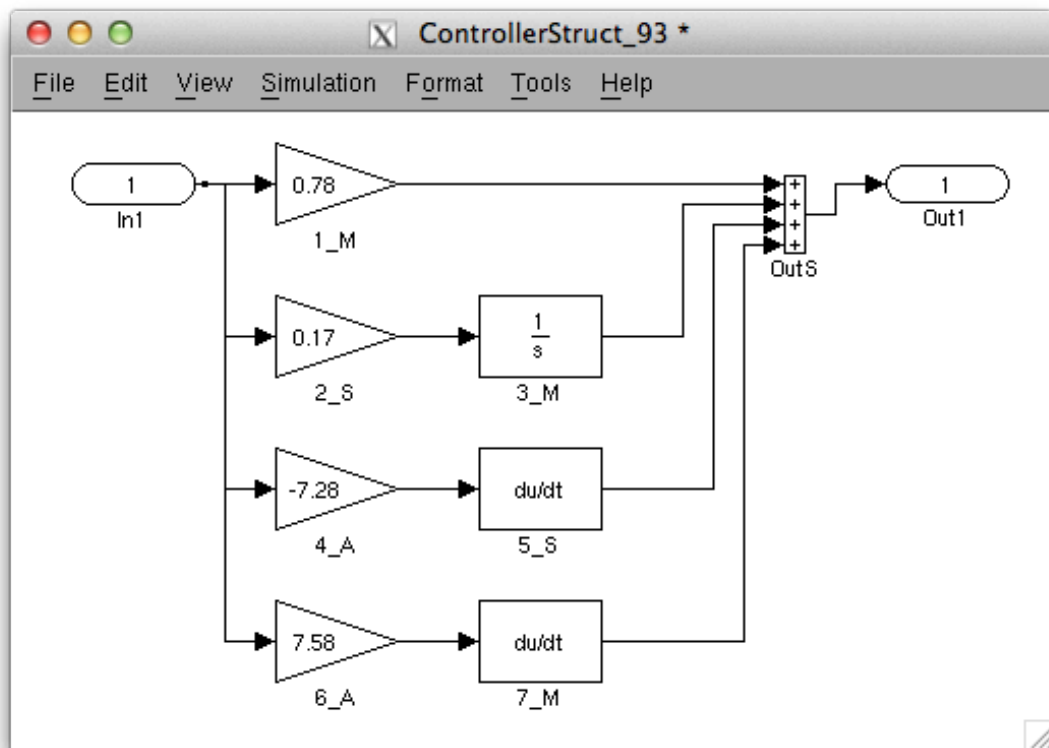


Figura 9-14: Controlador 93 – 2ª menor complexidade.

9.5.2 Maior Complexidade

Esta seção apresenta os controladores que obtiveram as maiores complexidades. O controlador 46 apresentado na Figura 9-10 também é a estrutura que possui a maior complexidade, mesmo assim apresentaremos a seguir 2 outras alternativas de alta complexidade. O controlador da Figura 9-15 obteve um Tempo de Acomodação de 0.29412 s, Custo de 59799.62 e Complexidade 148. O controlador da Figura 9-16 obteve um Tempo de Acomodação de 1.4843 s, Custo de 59796.16 e Complexidade 145.

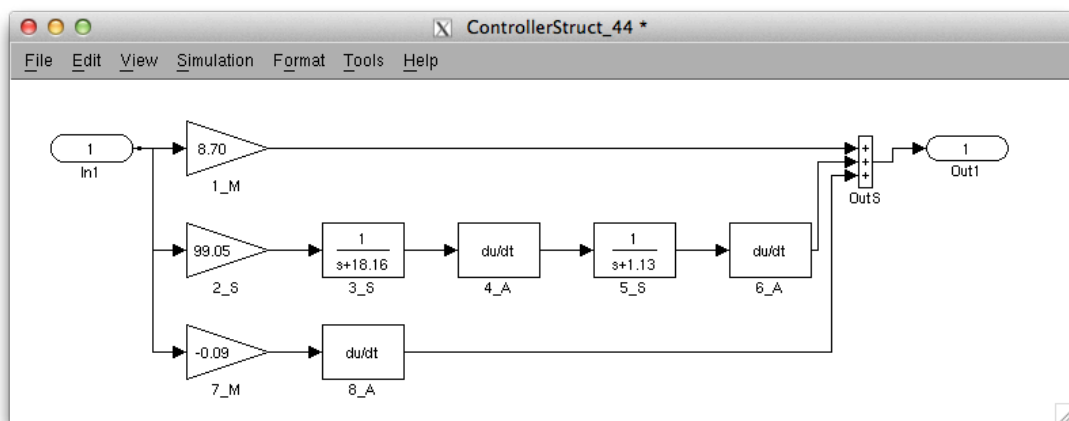


Figura 9-15: Controlador 44 – 3ª maior complexidade.

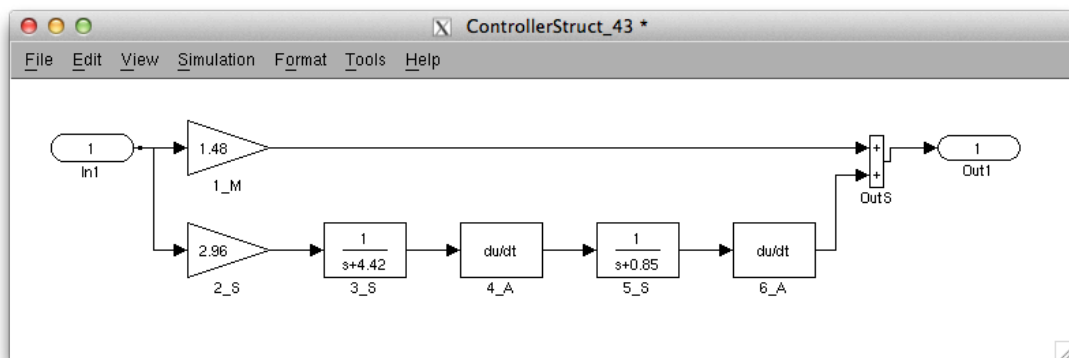


Figura 9-16: Controlador 65 – 4ª maior complexidade.

9.6 Fronteira de Pareto

Dentre as 42 soluções viáveis, há 9 soluções não dominadas candidatas a ótimo que formam a Fronteira de Pareto, Figura 9-17. São os seguintes controladores: 7, 8, 16, 31, 45, 72, 83, 88 e 89. Deste conjunto, vários já foram apresentados anteriormente neste capítulo. Serão exibidos a seguir aqueles ainda não vistos.

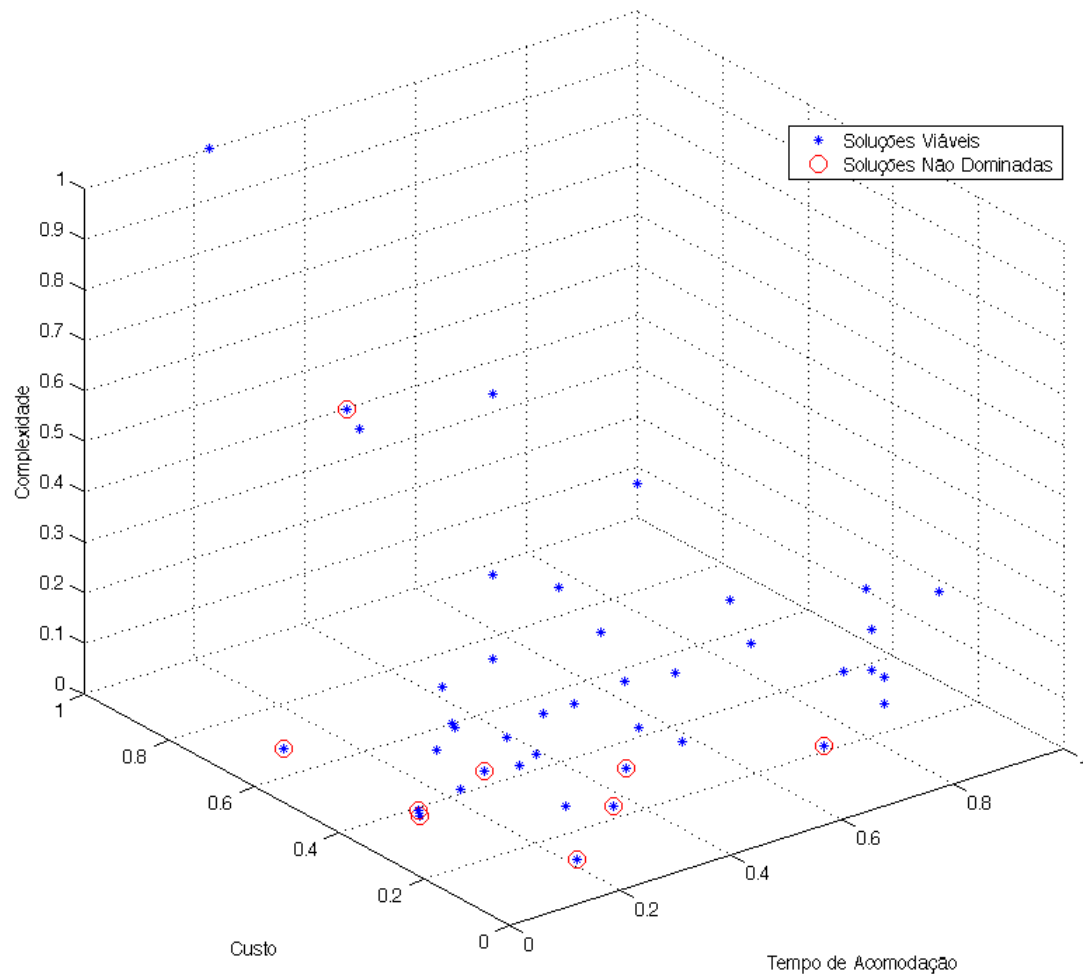


Figura 9-17: Espaço de soluções e a fronteira de Pareto.

O controlador da Figura 9-18 obteve um Tempo de Acomodação de 0.28928 s, Custo de 39863.13 e Complexidade 13. O controlador da Figura 9-19 obteve um Tempo de Acomodação de 3.3638 s, Custo de 19941.41 e Complexidade 10.

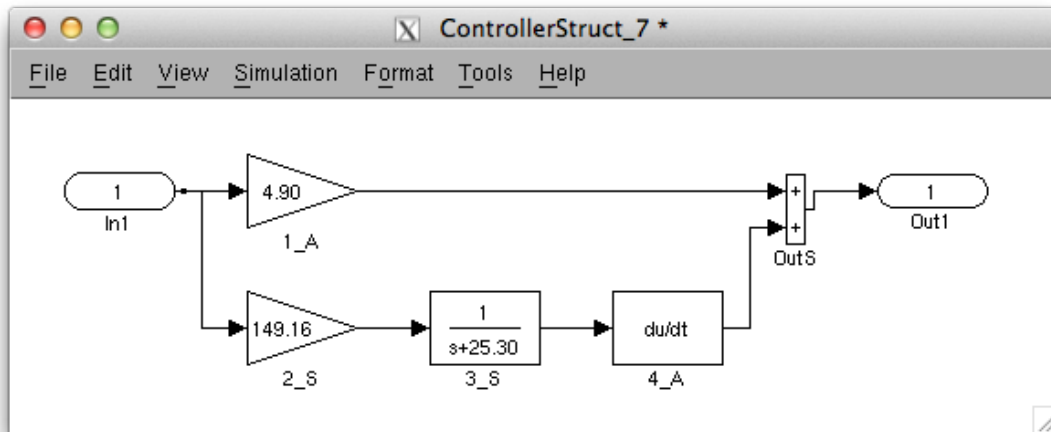


Figura 9-18: Controlador 7 - solução não dominada

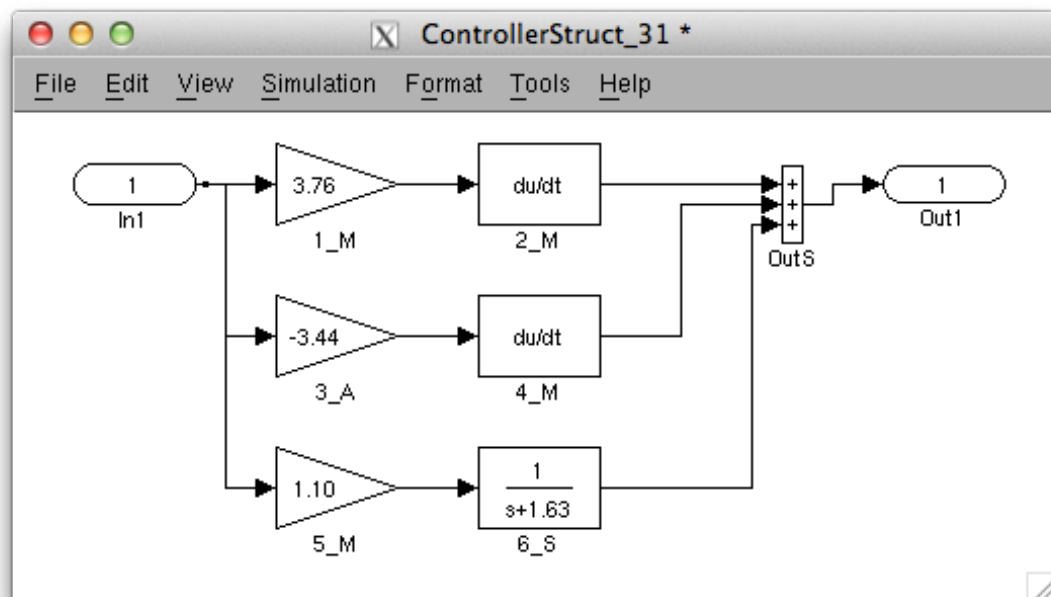


Figura 9-19: Controlador 31 - solução não dominada.

O controlador da Figura 9-20 obteve um Tempo de Acomodação de 0.27942 s, Custo de 39869.51 e Complexidade 15. O controlador da Figura 9-21 obteve um Tempo de Acomodação de 0.31874 s, Custo de 19945.40 e Complexidade 42.

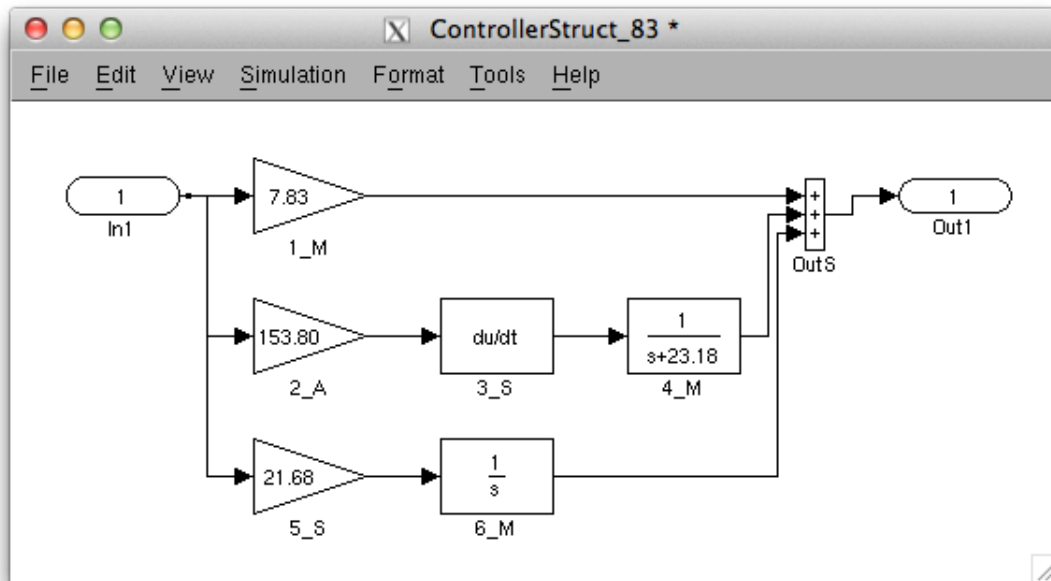


Figura 9-20: Controlador 66 - solução não dominada.

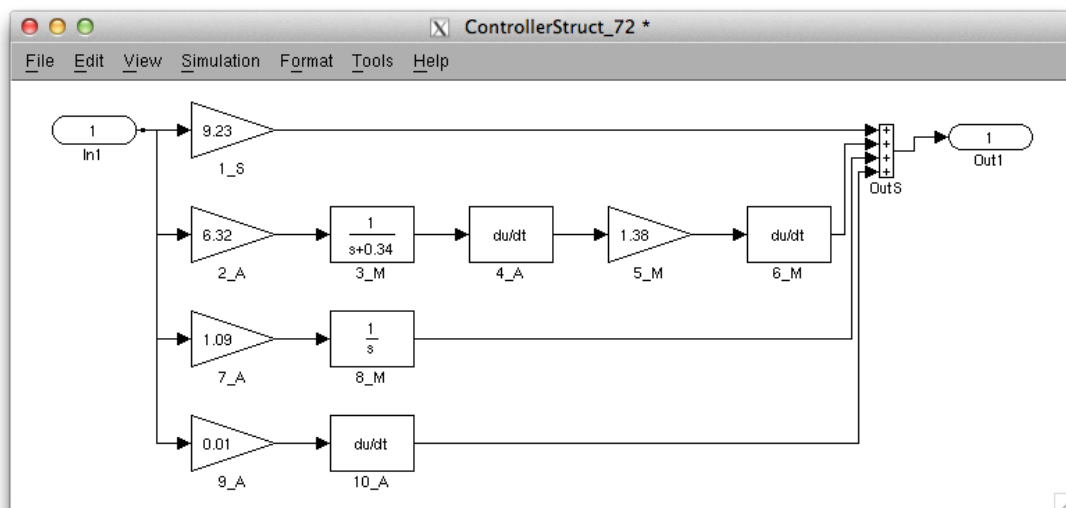


Figura 9-21: Controlador 72 - solução não dominada.

9.7 Seleção pelo Critério da Menor Perda

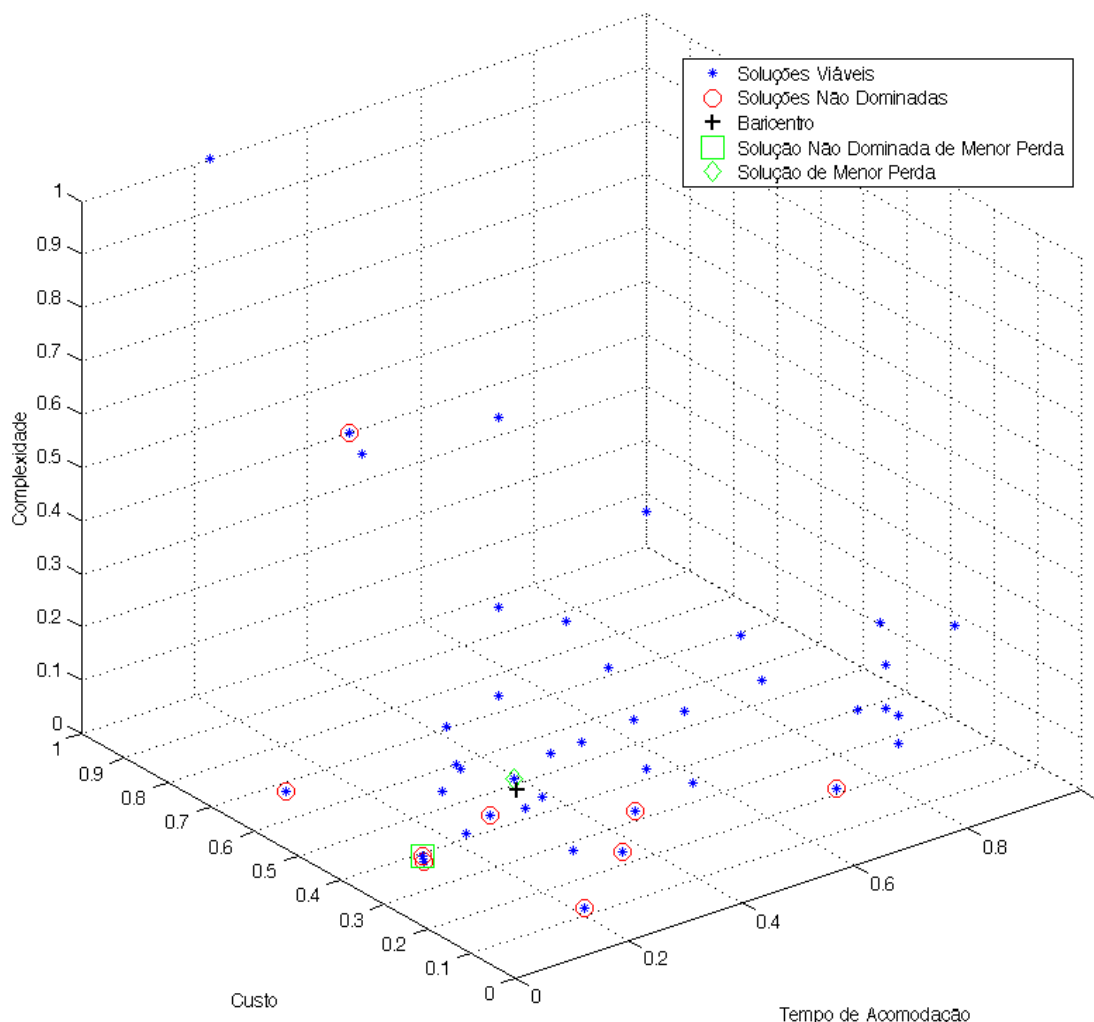


Figura 9-22: Espaço de soluções e critério de menor perda.

O baricentro em um espaço normalizado foi (Figura 9-22):

Tempo de Acomodação = 0.1975, Custo = 0.2540 e Confiabilidade = 0.1658.

A distância entre o baricentro e a solução na Fronteira de Pareto mais próxima é de 0.1694 (Controlador 83, Figura 9-12). A solução de Menor Perda é o Controlador 10 (Figura 9-23), com a menor distância entre sua posição e o baricentro no espaço de soluções, de 0.0369. A solução de Menor Perda obteve um Tempo de Acomodação de 1.0734 s, Custo de 39871.05 e Complexidade 32, sua função de transferência é:

$$\frac{134.5 s^5 + 1.427e5 s^4 + 8.274e6 s^3 + 3.386e7 s^2 + 4.53e7 s + 2.652e7}{s^5 + 2008 s^4 + 1.017e6 s^3 + 8.337e6 s^2 + 1.991e7 s + 1.407e7}$$

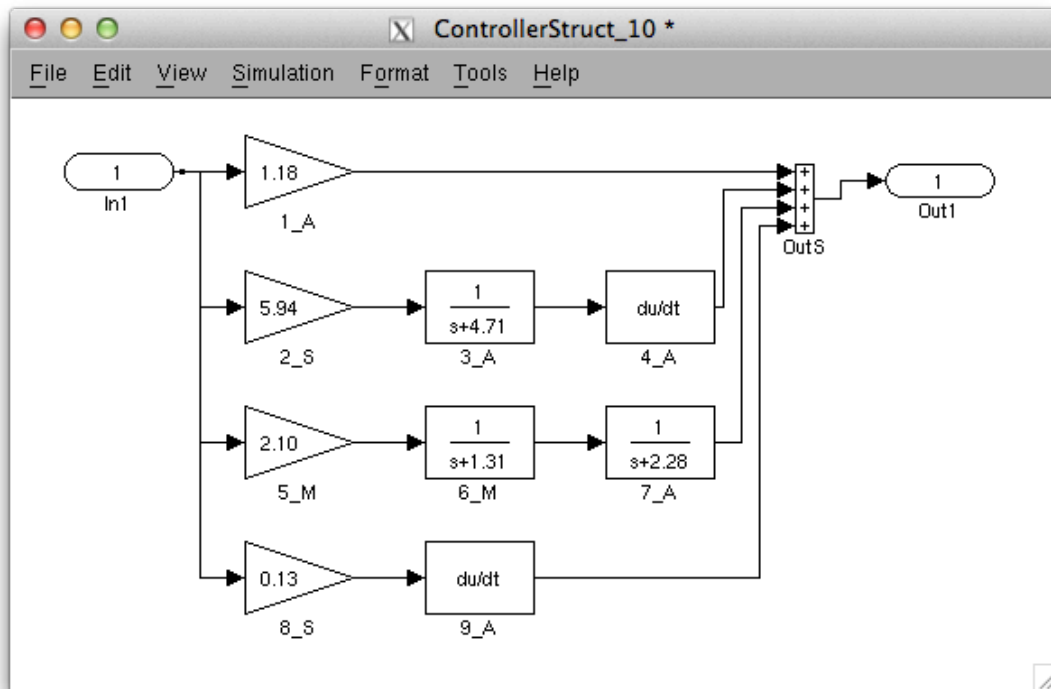


Figura 9-23: Solução de menor perda.

A Figura 9-24 apresenta a resposta a uma entrada a degrau unitário utilizando o Controlador 10. Note-se que: a variável de saída, gráfico superior, não ultrapassa os 5% de sobressinal; e o sinal de controle, gráfico inferior, não viola os limites de -320 e +320. A Figura 9-25 apresenta a resposta em frequência, por meio de diagramas de Bode.

Pode-se observar que a resposta do sistema atende às restrições impostas e apresenta uma solução equilibrada do ponto de vista de Tempo de Acomodação, Custo e Complexidade.

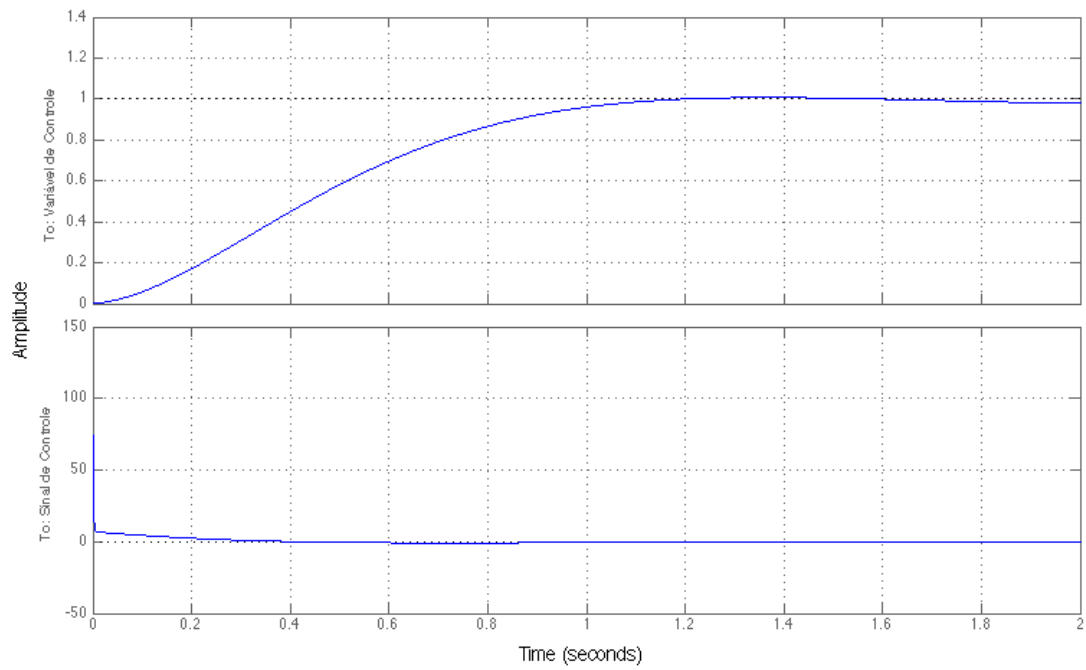


Figura 9-24: Resposta a uma entrada a degrau unitário da solução de menor perda.

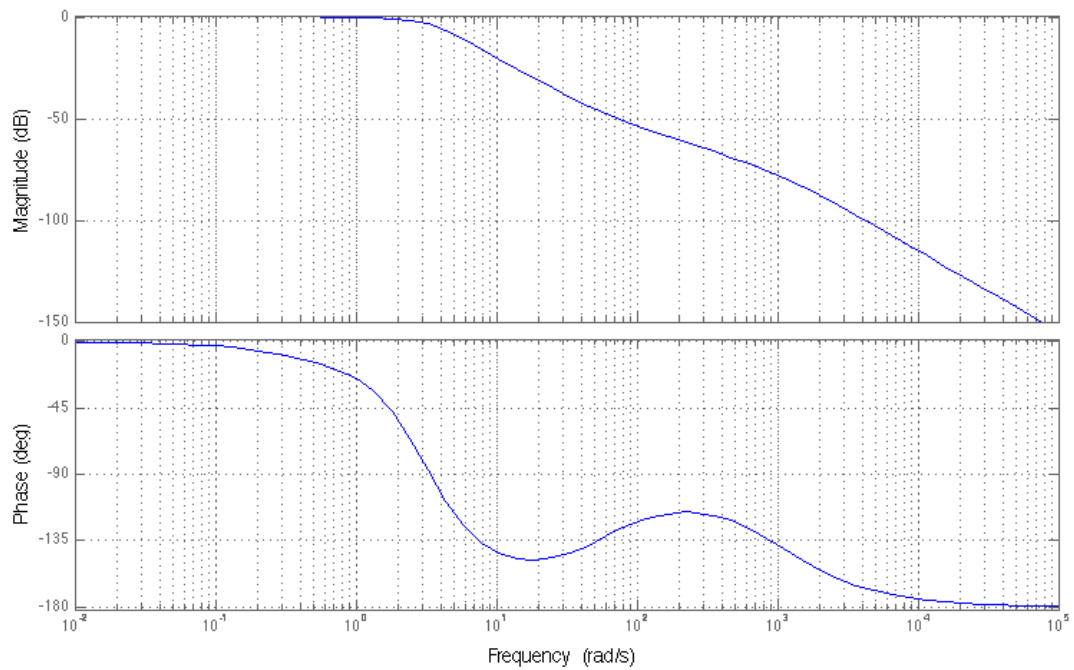


Figura 9-25: Diagramas de bode da solução de menor perda.

9.8 Comparação com Resultados da Literatura

Utilizaremos o problema proposto em (DORF; BISHOP, 2008), já mencionado anteriormente, para elaborar uma comparação de resultados. Esse trabalho foi escolhido por possuir uma referência para uma planta de segunda ordem e considerando a função de transferência do atuador o sistema a ser controlado passa a ser de terceira ordem, gerando assim um desafio diferente para o método desenvolvido. Outra característica deste problema é que este não busca simplesmente minimizar uma métrica de performance mas aceita soluções com tempo de acomodação menor que 5 s. Este tipo de problema se adequa melhor para uma comparação com a solução de Menor Perda encontrada neste trabalho.

Serão utilizados nessa comparação o Controlador 89 (Figura 9-3), estrutura que obteve o menor Tempo de Acomodação e o Controlador 10 (Figura 9-23) estrutura de Menor Perda.

A função de transferência dos controladores utilizados no trabalho do (DORF; BISHOP, 2008) e aquelas geradas neste trabalho podem ser vistas a seguir na Tabela 9-1.

Tabela 9-1: Comparativo entre as funções de transferência dos controladores.

Trabalho	Controlador
(DORF; BISHOP, 2008)	$\frac{320 (s + 1.1)}{s + 100}$
Menor Tempo de Acomodação	$\frac{6.435 s^3 + 3.625e5 s^2 + 2.217e5 s + 3.083e4}{s^3 + 1034 s^2 + 3.431e4 s}$
Menor Perda	$\frac{134.5 s^5 + 1.427e5 s^4 + 8.274e6 s^3 + 3.386e7 s^2 + 4.53e7 s + 2.652e7}{s^5 + 2008 s^4 + 1.017e6 s^3 + 8.337e6 s^2 + 1.991e7 s + 1.407e7}$

A Figura 9-26 apresenta a resposta temporal do sistema a uma entrada a degrau unitário. Percebe-se que: a variável de saída, gráfico superior, não ultrapassa os 5% de sobressinal; e o sinal de controle, gráfico inferior, não viola os limites de -320 e +320. A Figura 9-27 apresenta a resposta em frequência do sistema.

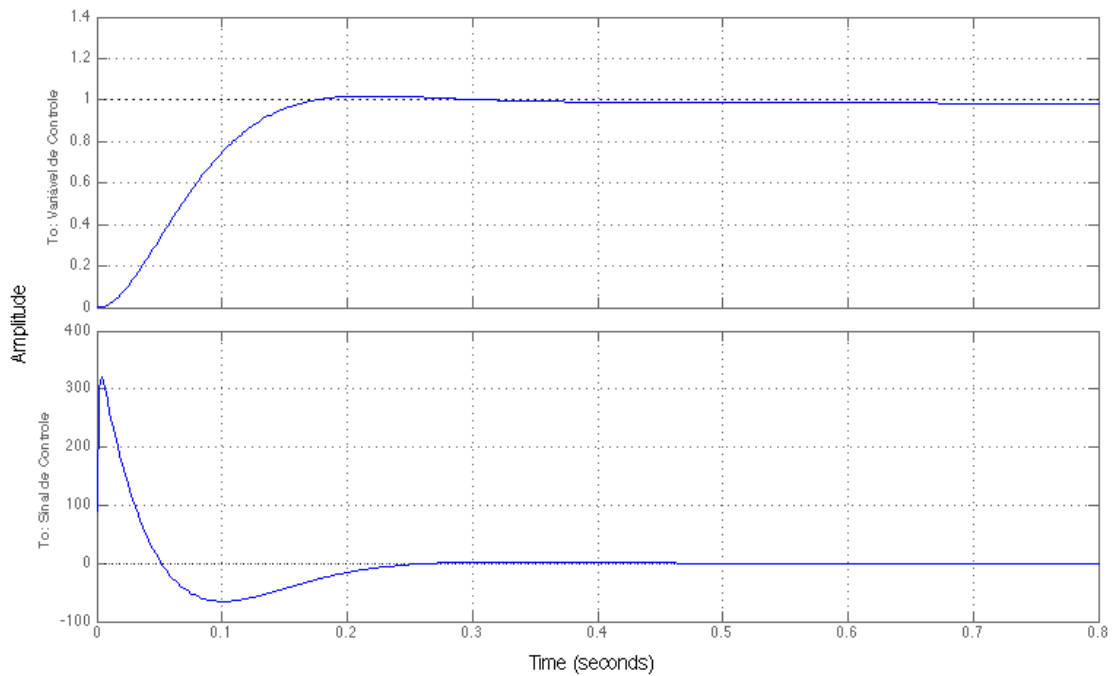


Figura 9-26: Resposta ao degrau unitário da solução de menor tempo de acomodação (controlador 89).

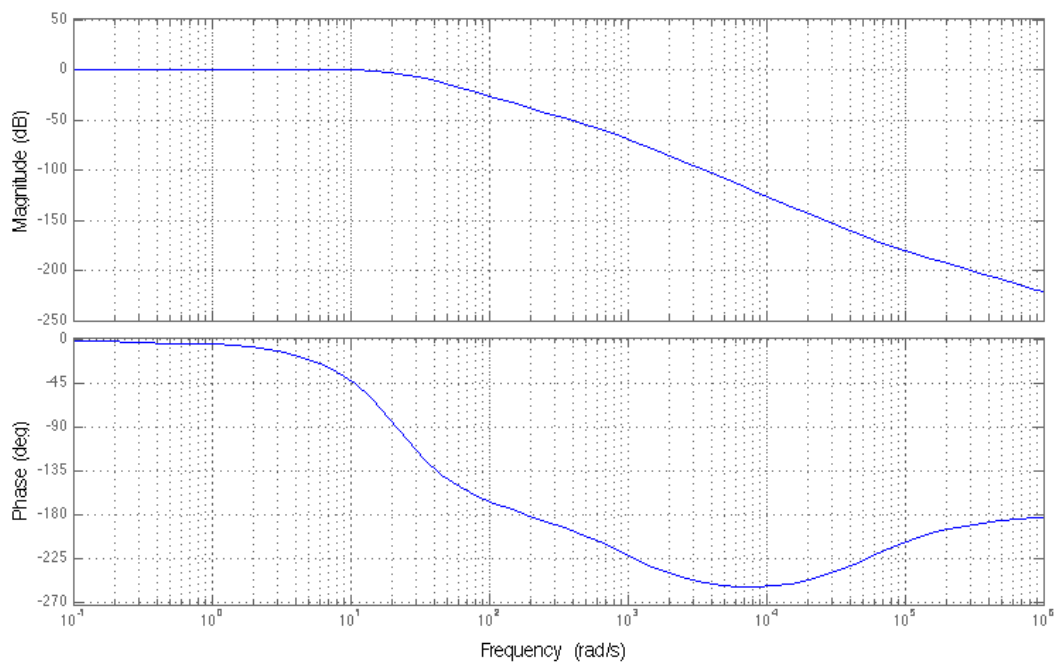


Figura 9-27: Diagramas de bode da solução de menor tempo de acomodação (controlador 89).

A comparação entre as respostas a uma entrada a degrau unitário dos três trabalhos pode ser vista na Figura 9-28. Enquanto a solução apresentada em (DORF; BISHOP, 2008) obteve o Tempo de Acomodação de 0.64638 s, o Controlador 89 foi capaz de atingir 0.16078 s, e o Controlador 10 de Menor Perda alcançou em 1.0734 s.

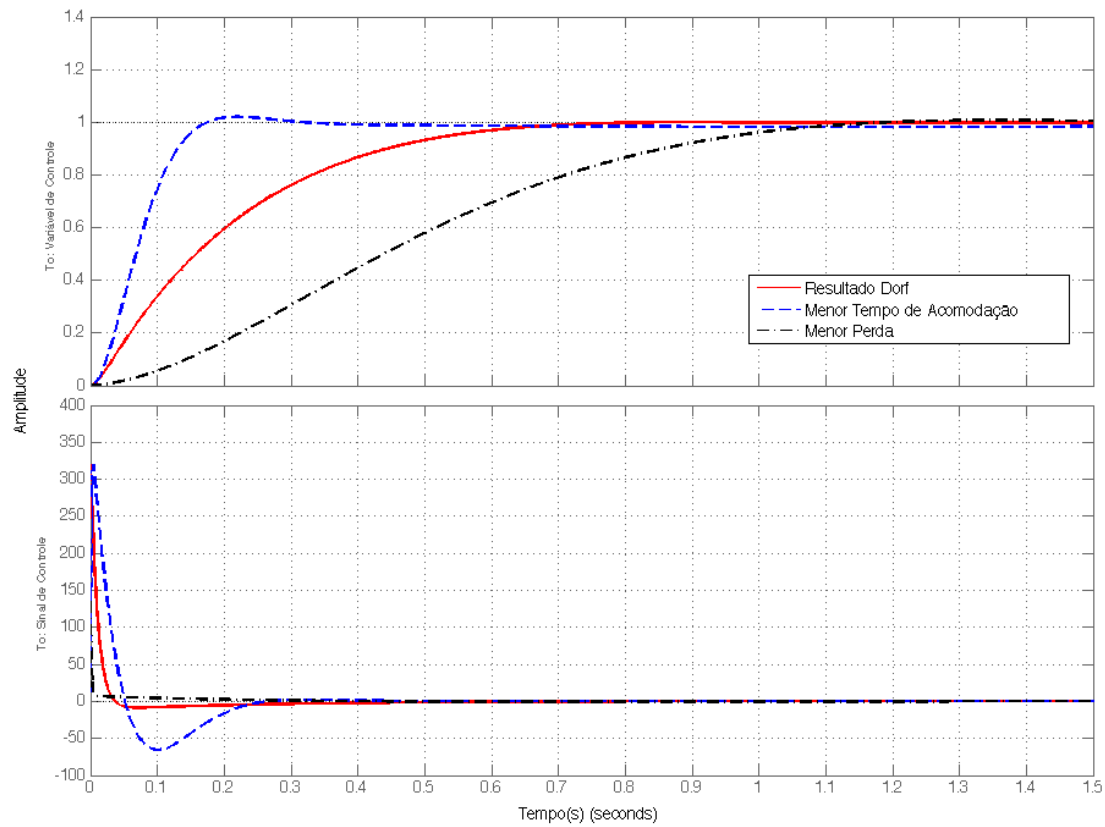


Figura 9-28: Comparação de entre as arquiteturas de menor tempo de acomodação (controlador 89) e menor perda (controlador 10), com a literatura.

Mesmo o Controlador 10 sendo possuidor de um Tempo de Acomodação maior, este ainda atende o requisito de fazê-lo em um intervalo de tempo menor que 5 s. Mesmo o Controlador 10 sendo mais complexo que o Controlador 89, este ainda é menos custoso. Este exemplo demonstra o quão complexo é realizar a escolha de uma arquitetura de controle quando é necessário levar diferentes aspectos em consideração.

10 UM MÉTODO AUTOMÁTICO PARA DESENVOLVER ARQUITETURAS FUNCIONAIS E FÍSICAS DE SISTEMAS DE CONTROLE

Este capítulo organiza o conhecimento construído até o momento e apresenta um método automático para a geração de arquiteturas funcionais e físicas de sistemas de controle, o qual foi induzido a partir dos métodos aplicados nas Investigações I1, I2 e I3. O método será implementado pelo "**Ambiente Automático de Geração Otimizada, Orientada e Randômica de Arquiteturas**" (AGORA), sendo uma generalização, o mesmo engloba os métodos descritos anteriormente como aplicações particulares deste.

10.1 Descrição do Método AGORA

O método AGORA desenvolvido neste trabalho faz uso de recursos aleatórios na geração de arquiteturas; entretanto incorpora também conhecimentos de projeto que restringem a busca aleatória à medida que essa arquitetura evolui. A Figura 10-1 ilustra o fluxograma do método/ambiente/ algoritmo AGORA.

Para fins didáticos, o método será separado nas seguintes fases:

1. Formação do Ambiente;
2. Geração de Arquiteturas;
3. Ajuste de Parâmetros; e
4. Seleção de Arquitetura.

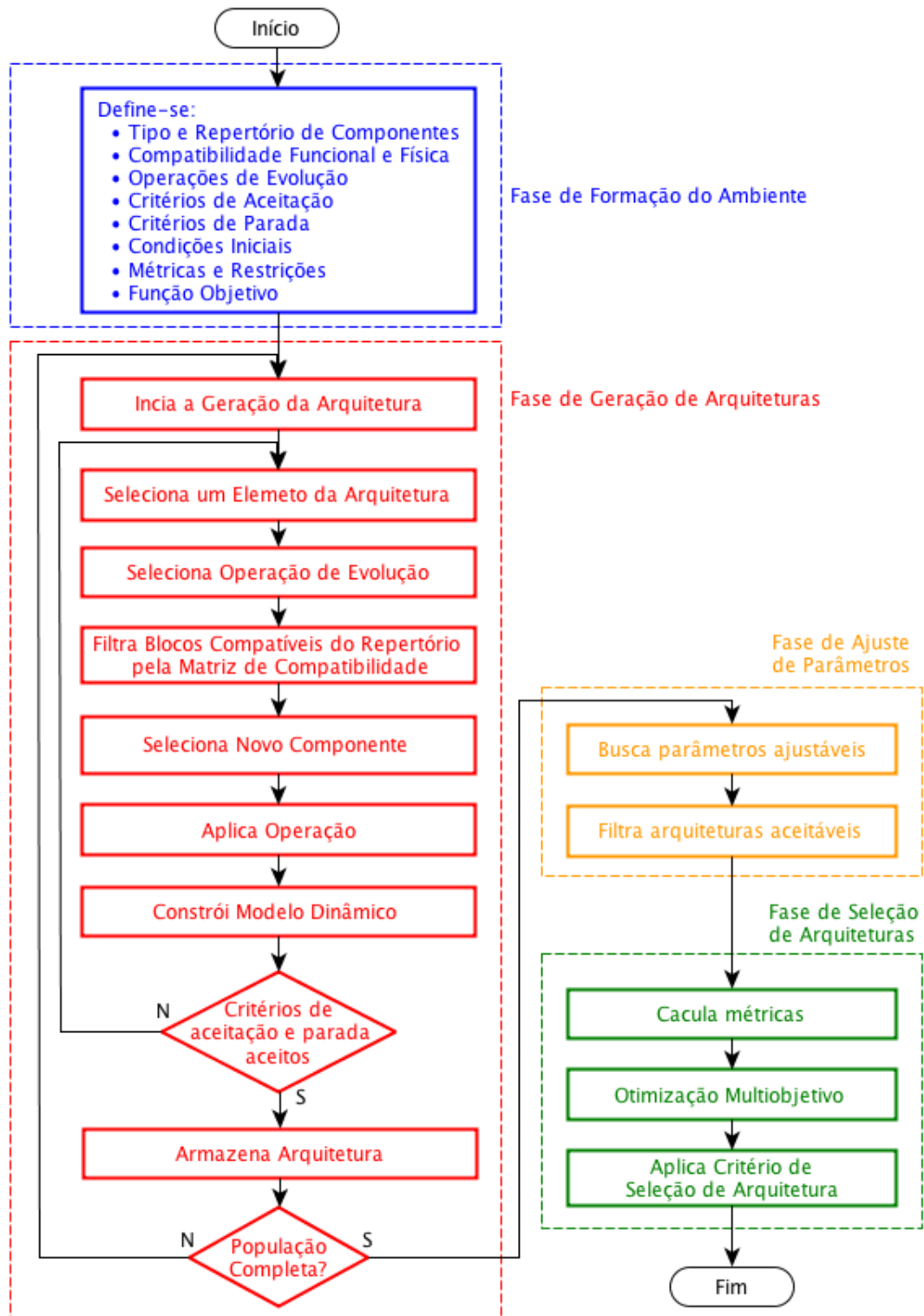


Figura 10-1: Fluxograma do método AGORA.

10.1.1 Fase de Formação do Ambiente

Esta é uma fase preparatória que molda o **espaço de soluções** do problema. Isto concede ao método AGORA uma enorme flexibilidade. Desta forma, pode-se ajustar o ambiente aos recursos disponíveis para um determinado projeto. Pode-se ainda modificar o ambiente para solucionar uma enorme gama de problemas.

Nesta fase, é criado um ambiente necessário para que as arquiteturas sejam geradas, ou seja, são definidos todos os elementos necessários à geração de arquiteturas, que serão descritos em detalhe a seguir.

10.1.1.1 Tipo e Repertório de Componentes e seus Modelos

Inicialmente define-se o tipo de componentes que poderão compor uma arquitetura, onde cada tipo de componente realiza uma função. Também definem-se quais os atributos que estes componentes possuem e seus parâmetros ajustáveis. Então, para cada tipo definido anteriormente, são instanciados diversos componentes alterando os seus atributos, e gerando seus modelos.

O conjunto destes componentes instanciados já com valores de seus atributos é chamado de **repertório de componentes** e definem os objetos que compõem parte do ambiente de geração de arquiteturas.

10.1.1.2 Compatibilidade Funcional e Física

Uma vez definidos os tipos e repertório de componentes, definem-se as compatibilidades funcional e física entre esses componentes. A compatibilidade funcional define se um determinado componente pode ser conectar a outro, levando em consideração seus tipos e o sequenciamento entre eles. Ainda há a compatibilidade física, que verifica se os outros atributos dos componentes são compatíveis, por exemplo: a quantidade de canais a serem processados, se o componente é analógico ou digital, etc.

A definição das compatibilidades funcional e física é de fundamental importância para a eficiência do método. Desta maneira é possível filtrar conexões inviáveis *a priori*, o que reduz bastante o tempo de processamento.

A compatibilidade funcional faz uso de uma Matriz de Compatibilidade **M** já mencionada nos capítulos anteriores.

10.1.1.3 Operações de Evolução da Arquitetura

A definição de quais **operações de evolução da arquitetura** poderão ser utilizadas é também parte da formação do ambiente de geração de arquiteturas. Além das operações, é necessário definir a álgebra de composição dos blocos que determina a dinâmica da arquitetura.

Neste trabalho foram utilizadas as seguintes operações listadas a seguir:

- Adição de um vértice pai;
- Adição de um vértice filho;
- Adição de um vértice em série;
- Adição de um vértice em paralelo;
- Adição de uma aresta vertical;
- Adição de uma aresta horizontal;
- Poda de uma aresta;
- Poda de um vértice;
- Poda de um ramo; e
- Operações compostas a partir de operações simples.

Além destas operações pode-se também implementar outras operações como:

- Adição de um ramo de “feedback”;
- Adição de um ramo de “feedforward”; e
- Outras operações compostas.

10.1.1.4 Critérios de Aceitação e de Parada

Os **critérios de aceitação** de uma arquitetura e os **critérios de parada** do processo de geração também são definidos na fase de formação do ambiente. Estes podem ser dos mais diversos tipos, desde a quantidade de componentes ou até que a arquitetura estabilize uma planta. Também está incluída nos critérios de parada o tamanho da população de arquiteturas que se deseja gerar.

10.1.1.5 Condições Iniciais

Em paralelo com os critérios, se faz necessário definir condições iniciais para o processo de geração, tais como: qual a arquitetura inicial ou qual o valor inicial de parâmetros ajustáveis de componentes.

10.1.1.6 Atributos e Métricas

Já na fase de formação do ambiente, é preciso definir quais os atributos serão utilizados para avaliar e selecionar as arquiteturas geradas e como suas métricas serão calculadas. Esta etapa é de fundamental importância pois é a partir destas métricas que as arquiteturas serão avaliadas e selecionadas, determinando a arquitetura resultante.

10.1.1.7 Restrições

Restrições do problema a ser solucionado também devem ser definidas na formação do ambiente, incluindo restrições do tipo: valor limite para o sobressinal de uma planta, valor máximo de saída de um atuador etc.

10.1.1.8 Função Mono-objetivo Individual

Finalmente conclui-se a formação do ambiente com a definição da função objetivo individual para o sistema de controle. É importante ressaltar que esta é uma função mono-objetivo é utilizada para ajustar os parâmetros de cada arquitetura e não como

função objetivo do método AGORA, pois o método geral utiliza uma função multiobjetivo.

10.1.2 Fase de Geração de Arquiteturas

Com todo o ambiente de geração criado é possível seguir para a fase de geração de arquiteturas. Esta fase começa com uma arquitetura inicial que foi definida nas condições iniciais do problema e então tem início uma varredura pelos componentes da arquitetura, selecionando um componente por vez.

Com um componente selecionado é feita uma escolha aleatória da operação de evolução da arquitetura a ser aplicada naquele componente. Havendo a necessidade de adição de um novo componente filtram-se, dentre todos os tipos de componentes disponíveis, somente aqueles que são funcionalmente compatíveis; e, a partir destes, é feita uma escolha aleatória do tipo. Filtram-se então, no repertório de componentes, pelo tipo escolhido, aqueles que são fisicamente compatíveis; e, mais uma vez, é feita uma escolha aleatória, selecionando-se enfim o componente a ser adicionado. A operação é então aplicada à arquitetura. O uso de filtros com conhecimentos para orientar *a priori* as escolhas aleatórias na geração de uma arquitetura é que dá a parte do nome "Ambiente Automático de Geração Otimizada, Orientada e Randômica de Arquiteturas" ao método AGORA.

Caso esta arquitetura possua comportamento dinâmico seu modelo é reconstruído após a aplicação da operação.

Se os critérios de aceitação forem atingidos, então esta arquitetura será armazenada junto à população de arquiteturas. Se um dos critérios de parada for atingido sem os critérios de aceitação então esta arquitetura será descartada.

O processo segue até que se obtenha a população do tamanho desejado.

10.1.3 Fase de Ajuste de Parâmetros

Esta fase se aplica somente a arquiteturas que possuem dinâmicas. Uma vez que toda a população foi gerada então é necessário encontrar os valores para os

parâmetros de cada arquitetura. Diversos métodos e ferramentas numéricas podem ser utilizados.

Nesta etapa, faz-se uso das restrições e das funções mono-objetivo individual definidas na fase de formação do ambiente. As restrições são utilizadas para compor uma função mono-objetivo penalizando aqueles conjuntos de parâmetros que violam as restrições impostas.

Ainda nesta fase, após todos os ajustes dos parâmetros filtram-se aquelas arquiteturas aceitáveis que não violam nenhuma das restrições impostas.

10.1.4 Fase de Seleção de uma Arquitetura

Esta é a última fase do método. Nesta fase calculam-se as métricas, definidas na fase de formação do ambiente, para cada uma das arquiteturas aceitáveis. Essas métricas são utilizadas como entradas em uma otimização multiobjetivo que resultará em um conjunto de soluções candidatas a ótima.

Com o resultado obtido na etapa anterior, aplica-se um critério de seleção de uma arquitetura. Neste trabalho aplicamos o Critério de Menor Perda já mencionado anteriormente, porém outros critérios podem ser utilizados. Finalmente conclui-se o método proposto e, como seu resultado, obtém-se uma única arquitetura a ser implementada.

10.2 Denominação do Método AGORA

Fazendo uso de uma certa liberdade de pensamento, o nome AGORA, escolhido para o método, também faz referência a algo além da Ambiente Automático de Geração Otimizada, Orientada e Randômica de Arquiteturas.

O uso de várias métricas reunidas na otimização multiobjetivo e a ideia de ter que chegar a um consenso de compromisso entre essas métricas para escolher de uma solução única, faz lembrar a *Ágora* dos gregos.

A *Ágora* era um lugar de reunião nas cidades da Grécia Antiga, onde seu povo se reunia para discutir sobre os mais diversos temas. De todas, talvez a mais famosa seja

a Antiga *Ágora* de Atenas, tido como lugar de nascimento da democracia. A ela, nossa modesta homenagem através da escolha do título do método proposto.

11 CONCLUSÕES

Este trabalho apresentou a pesquisa pertinente e a proposta de um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multiobjetivo baseada em modelos, atributos e métricas sistêmicas. O método desenvolvido foi implementado e pode ser aplicado a dois tipos de sistemas: estáticos, aqueles da Investigação I1; e dinâmicos, aqueles das Investigações I2 e I3. Apesar de o método seguir os mesmos conceitos para a aplicação em ambos os sistemas, cada um possui suas particularidades e, por isso, serão analisados separadamente.

11.1 Aspectos Gerais

Durante a Revisão da Literatura, observou-se que, mesmo havendo uma grande gama de trabalhos de referência em diversas disciplinas do conhecimento, há poucas publicações com pesquisas em que se caminham na direção dos objetivos deste trabalho. O principal grupo de pesquisas sobre este assunto é um grupo de Algoritmos Genéticos centrados na Universidade de Stanford (KOZA; KEANE et al., 1999). E, com base nas pesquisas deste grupo, percebe-se que a busca por arquiteturas de controladores é atual e possui grande valor.

Durante a Análise e Seleção de Métodos, Arquiteturas, Modelos e Métricas Existentes aplicáveis aos sistemas de controle, percebeu-se que há uma distância não desprezível entre o universo da Engenharia de Controle e da Engenharia de Sistemas. O seguinte exemplo pode elucidar essa afirmação: uma vez especificado e projetado um sistema de controle, no domínio da Engenharia de Controle, esse sistema para materializar-se deve ser construído (ou implementado). Essa construção em geral é responsabilidade da Engenharia de Sistemas que, por sua vez, faz um novo projeto levando em considerações os atributos sistêmicos, como confiabilidade, por exemplo. O presente trabalho tem se mostrado útil para reduzir a distância entre esses dois domínios, levando em consideração aspectos de ambos para a elaboração de arquiteturas.

Sobre a Escolha dos Atributos e Construção das Métricas, os atributos foram escolhidos, as métricas foram definidas e calculadas para efeito de demonstração do método proposto. O trabalho realizado não se limita aos atributos escolhidos, nem às métricas definidas. Esta flexibilidade é mantida pois cada problema pode exigir um conjunto de atributos e métricas próprio. A escolha dos atributos utilizados em cada problema é essencial para a obtenção de soluções úteis e fiéis a realidade. Tão importante quanto à escolha desses é a construção das métricas.

11.2 Arquiteturas para Sistemas Estáticos - Primeira Investigação (I1)

Durante a Investigação pelas Melhorias nos Métodos, Arquiteturas, Modelos e/ou Métricas, um modelo e um ambiente foram desenvolvidos, como descrito no Capítulo 0. O modelo leva em consideração atributos sistêmicos; trata a forma da arquitetura de sensores como uma árvore (grafo); procura ser o mais próximo da realidade fazendo uso de componentes "COTS", e incorpora o conceito de canais além de portas de conexão. O ambiente criado é capaz de elaborar, por meio de um processo de otimização, orientado e randômico, gerar e selecionar arquiteturas de sensores fazendo uso de métodos multiobjetivos.

Durante os Resultados da Investigação I1 para as árvores de sensores geradas, estes se mostraram satisfatórios e capazes de fornecer algumas percepções sobre a forma dessas arquiteturas. Essas arquiteturas foram analisadas pelas métricas que são influenciadas pela forma. A maneira como a confiabilidade e a complexidade foram medidas neste trabalho torna possível fazer essas análises. Outro aspecto de análise é a importância da comparação entre os melhores e piores resultados de um determinado atributo. Com esse tipo de análise relativa pode-se observar para uma determinada métrica quais características da árvore são desejadas e sua antítese, para entender o que não se deseja.

Observando a Confiabilidade percebe-se: que com estruturas que possuem mais ramificações (caminhos redundantes), obtém-se uma maior confiabilidade. Observa-se também que a confiabilidade melhora quando os componentes com níveis mais altos de confiabilidade ficam mais próximos da raiz da árvore. Cientes que os valores

de confiabilidade dos componentes usados são representativos mas não reais, os resultados das árvores sugerem que a forma desta estrutura possui uma grande influência no resultado. Ou seja, até que ponto vale a pena comprar poucos componentes de altíssima confiabilidade (**Abordagem de Prevenção de Falhas**), em vez de usar uma estrutura com mais redundâncias e componentes de menor custo (**Abordagem de Tolerância a Falhas**). Outro aspecto interessante é: se o projeto não dispuser de recursos para todos os componentes com mais alto nível de confiabilidade, os resultados sugerem que vale a pena investir naqueles componentes mais próximos da raiz da árvore.

Observando a Complexidade, da forma como foi medida neste trabalho, vê-se que ela possui um comportamento exatamente oposto ao da confiabilidade, ou seja, para minimizar a complexidade das estruturas, uma quantidade menor de ramificações deve ocorrer. E estas duas devem atuar como forças conflitantes para evitar o viesamento das soluções propostas. Encontrar uma métrica indicativa deste equilíbrio entre a confiabilidade e a complexidade pode ser útil na escolha deste tipo de estrutura.

Observando a Análise Multiobjetivo, vê-se que as soluções candidatas a ótimo localizadas na Fronteira de Pareto possuem diferentes características: umas privilegiam a complexidade em detrimento de outras métricas; e outras fazem o mesmo com as outras métricas, como a confiabilidade. Entre essas candidatas, a seleção de uma solução por meio do Critério de Menor Perda proposto em (ROCCO 2002), demonstrou ser uma solução equilibrada das métricas escolhidas.

Observando a Descrição da Ferramenta de Geração de Arquiteturas e do Modelo por Árvores, percebe-se que estes lidam com uma grande quantidade de variáveis, o que se mostrou ser uma força do método, mas também uma dificuldade quando se deseja isolar com clareza a influência da estrutura ou dos atributos dos componentes nas métricas de uma arquitetura resultante.

11.2.1 Sugestões para Trabalhos Futuros

O método e ambiente desenvolvidos para a geração e avaliação de arquiteturas para sistemas estáticos ainda possui enorme potencial de exploração. Para tanto sugerimos os seguintes trabalhos futuros:

- Utilizar esse ambiente com um catálogo real de componentes para um projeto existente.
- Desenvolver outras maneiras de evolução de arquiteturas além daquelas já descritas no Capítulo 4.
- Utilizar o potencial de geração de diversas alternativas deste ambiente para buscar e avaliar novas métricas para outros atributos como, por exemplo, para manutenibilidade e reconfigurabilidade.
- Incluir métricas para as arestas da árvore, dessa forma será possível avaliar, por exemplo:
 - distâncias entre vértices;
 - diferentes tipos de arestas, com custos e/ou confiabilidade distintos.
- Estender para geração de arquiteturas de outros domínios do conhecimento, como por exemplo:
 - no projeto de cabeamento de aeronaves, para coleta e distribuição de dados;
 - no projeto de campos de petróleo, onde se deseja produzir petróleo e gás de diversos poços distintos até uma unidade de produção.
- Definir um conjunto de problemas típicos e elaborar um receituário de arquiteturas típicas que melhor atendam cada um desses problemas.

11.3 Arquiteturas para Sistemas Dinâmicos - Segunda Investigação (I2) e Terceira Investigação (I3).

Durante a Investigação pelas Melhorias nos Métodos, Arquiteturas, Modelos e/ou Métricas, as investigações I2 e I3 foram realizadas e, como resultado delas, um modelo e um ambiente foram desenvolvidos, como descritos nos Capítulos 5 e 0. O modelo leva em consideração atributos de desempenho e atributos sistêmicos; trata a forma da arquitetura de controlador como um grafo; procura ser o mais próximo da realidade fazendo uso de componentes "COTS". O ambiente criado é capaz de elaborar, por meio de um processo randômico e orientado, e selecionar arquiteturas de controladores fazendo uso de métodos multiobjetivos.

Sobre a Ferramenta de Geração de Arquiteturas de Controladores, o ambiente desenvolvido neste trabalho lida com problemas em um nível de abstração superior à maneira tradicional de ajustes de parâmetros de um controlador. Após a configuração do ambiente para solução de um problema de controle, cada estrutura gerada determina um universo de busca próprio, de dimensão igual à quantidade de parâmetros ajustáveis para aquela arquitetura. Dessa maneira, pode realizar uma busca de solução muito mais ampla do que permanecendo com uma estrutura única como um controlador PID por exemplo.

Comparando a Ferramenta de Geração de Arquiteturas de Controladores com a Literatura, percebe-se que o método utilizado para gerar arquiteturas foi superior ao método descrito na literatura (KOZA; KEANE et al., 1999), para o problema proposto no mesmo trabalho. Mesmo não sendo esse o objetivo principal do método, este foi capaz de igualar o tempo de subida ou até mesmo reduzir este tempo, dependendo do critério de tempo de subida escolhido, e isto foi realizado mesmo considerando os seguintes aspectos:

- utilizando uma quantidade menor de arquiteturas;
- utilizando menor poder computacional;
- processando em uma menor quantidade de tempo;
- gerando sempre controladores causais; e

- sem a necessidade de incluir um bloco de saturação que introduz uma não linearidade no sistema sob análise.

Observando as Métricas de Desempenho nas Investigações I2 e I3, as arquiteturas que atingiram o menor tempo de subida (I2) e o menor tempo de acomodação (I3) tem a mesma estrutura, um controlador PID com um polo adicional na saída do bloco derivativo.

Observando as Métricas Sistêmicas nas Investigações I2 e I3, da forma como foram medidas nestas investigações I2 e I3, vê-se que essas métricas, mesmo calculadas de maneira elementar, são capazes de fornecer percepções úteis sobre o sistema de estudo. Ao observarmos as estruturas de menor complexidade nas investigações I2 e I3, percebe-se que, para controlar uma planta de maior ordem (I3), as arquiteturas resultantes também foram de maior complexidade.

Observando a Análise Multiobjetivo, vê-se que as soluções candidatas a ótimo localizadas na Fronteira de Pareto possuem diferentes características: umas privilegiam a complexidade em detrimento de outras métricas; e outras fazem o mesmo com as outras métricas, como as métricas de desempenho. Entre essas candidatas, a seleção de uma solução por meio do Critério da Menor Perda proposto em (ROCCO, 2002), demonstrou ser uma solução equilibrada para as métricas escolhidas.

Analisando os Resultados do Critério de Menor Perda, as soluções escolhidas pelo critério da Menor Perda, tanto na investigação I2 como na investigação I3 não são os controladores com desempenho superior àqueles da literatura. Apesar de parecer intuitivamente inaceitável do ponto de vista da Engenharia de Controle, são estas as soluções que melhor atendem ao problema quando analisado de uma visão multifacetada, conseguindo escolher uma solução balanceada dentre um universo de soluções aceitáveis.

11.3.1 Sugestões para Trabalhos Futuros

O método e ambiente desenvolvidos para a geração e avaliação de arquiteturas para sistemas dinâmicos ainda possui enorme potencial de exploração, para tanto sugerimos os seguintes trabalhos futuros:

- Utilizar esse ambiente para um projeto existente, com um catálogo real de componentes, incluindo os limites reais de parâmetros ajustáveis.
- Desenvolver outras maneiras de evolução de arquiteturas além daquelas já descritas no Capítulo 5 e 6.
- Explorar melhor o método e o ambiente descritos e compará-los mais profundamente com outros métodos disponíveis na literatura.
- Utilizar o potencial de geração de diversas alternativas deste ambiente para buscar e avaliar novas métricas para outros atributos, como, por exemplo, para manutenibilidade e reconfigurabilidade.
- Incluir métricas para as arestas da estrutura. Dessa forma será possível avaliar, por exemplo, atrasos na comunicação entre blocos.
- Definir um receituário de arquiteturas típicas de controladores para diversos tipos de problemas e plantas a serem controladas.
- Criar um método que ao invés de gerar múltiplas alternativas evolua uma arquitetura até convergir para a solução do problema.
- Estender a geração de estruturas para outras áreas da engenharia de controle, como por exemplo:
 - para sistemas MIMO (*Multiple Input and Multiple Output*);
 - para o Controle Moderno;
 - para o Controle Digital.

11.4 Conclusões finais

O método e o ambiente descritos na presente Tese são úteis para solucionar uma ampla gama de problemas e podem ser aplicados em diversos domínios do conhecimento. Pôde-se observar que a arquitetura do sistema altera profundamente suas capacidades funcionais, em um nível que os parâmetros de seus componentes jamais poderiam alcançar. Desta forma a presente Tese cumpre com a **generalidade e utilidade** almejadas.

O método de geração das arquiteturas tanto para sistemas estáticos como sistemas dinâmicos são inovações criadas durante o desenvolvimento desse trabalho. No problema proposta na investigação I2, o qual foi solucionado por outro método comparável com este, o método deste trabalho se mostrou superior em alguns aspectos. Além disto, e da inovação do método de geração das arquiteturas, a utilização do Critério da Menor Perda para chegar racionalmente a uma solução que une equilibradamente os requisitos tanto da Engenharia de Controle como da Engenharia de Sistemas são conquistas inovadoras desta Tese. Esta solução representa a descoberta de uma ponte entre essas duas disciplinas, que de agora em diante pode ser utilizada com maior frequência. Com isso, a presente Tese cumpre com a **originalidade** aspirada.

Espera-se que resultados obtidos nesta Tese sejam ainda ínfimos quando se vislumbra o potencial que pode ser alcançado com a continuidade da pesquisa sobre o tema.

12 REFERÊNCIAS BIBLIOGRÁFICAS

AVIZIENIS, A.; LAPRIE, J. C.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. **IEEE Transactions On Dependable and Secure Computing**, v. 1, n. 1, 2004.

ALEXANDERSON, G. L. About the cover: Euler and Konigsberg's Bridges: a historical view. **Bulletin of the American Mathematical Society**. v. 43. n. 4. 2006. 567.

AMOROSO, A. L. **Um metodo de analise e especificacao de sistmas com requisitos de desempenho, custo e confiabilidade, aplicado a rodas de reacao**. 1999. 131 p. (INPE-7517-TDI/730). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, Sao Jose dos Campos, 1999. Disponível em: <<http://urlib.net/sid.inpe.br/deise/2001/04.24.10.59>>. Acesso em: 26 ago. 2013

ASTRÖM, K. J.; HÄGGLUND, T. **PID controllers: theory, design, and tuning**. 2. ed. Instrument Society of America, 1995.

BARANGER, M. **Chaos, complexity, and entropy**. Cambridge: New England Complex Systems Institute, 2000.

BOUNOVA, G. A., AHN, J.; HOFSTETTER, W.; WOOSTER, P.; HASSAN, R.; WECK, O. L. **Selection and technology evaluation of moon/mars transportation architectures**. Long Beach, CA: American Institute of Aeronautics and Astronautics, 2005. 10.

CAYLEY, A. On the theory of the analytical forms called trees. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**. v. 13. n. 85.p, 172-176, 1857.

CAYLEY, A. On the analytical forms called trees. Second Part. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**. v. 18. n. 121, p. 374-378, 1859.

- CARDOSO, D. M. **Teoria dos grafos e aplicações**. Aveiro: Departamento de Matemática da Universidade de Aveiro, 2011.
- CITRON, S. J. **Elements of optimal control**. Holt: Rinehart and Winston, Inc., 1969.
- CLOSE, C. M.; FREDERICK, D. K. **Modeling and analysis of dynamic systems**. 2. ed. John Wiley & Sons, Inc, 1995.
- CONNE, J.; KOO, B. **OPN usage tutorial version 1.4**. 2006. Disponível em: [http://www.jconne.com/nss-folder/opn1public/OPN Usage Tutorial Version 1.4.pdf](http://www.jconne.com/nss-folder/opn1public/OPN%20Usage%20Tutorial%20Version%201.4.pdf). Acesso em: 26 ago 2013.
- CRAWLEY, E. F. **Introduction to system architecture**, 2007. (notas).
- CRAWLEY, E. F. et al. **The influence of architecture in engineering systems**. Cambridge, MA: MIT ESD, Março de 2004. Engineering Systems Monograph of the Engineering Systems Symposium
- DODSON, B.; NOLAN, N. **Reliability engineering handbook** (Quality and Reliability). 1. ed. CRC Press, 1999.
- DORF, R. C.; BISHOP, R. H. **Modern control systems**. 11 ed. Pearson Education, 2008.
- DORF, R. C.; BISHOP, R. H. **Modern Control Systems**. Addison Wesley, 1998.
- EULER, L. Solutio problematis ad geometriam situs pertinentis. **Commentarii academiae scientiarum Petropolitanae**. v. 8. p. 128-140, 1741.
- EUSGELD, I.; FREILING, F. C.; REUSSNER, R. **Dependability Metrics - Advanced Lectures**. Springer, 2010. ISBN: 978-3-540-68946-1 (Print) 978-3-540-68947-8 (Online)
- GOODE, H. H.; MACHOL, R. E.; TEICHMANN, T. **System engineering: an introduction to the design of large-scale systems**. New York: McGraw-Hill, 1957.
- HOFSTETTER, W, K.; WOOSTER, P, D.; CRAWLEY, E. F. Analysis of human lunar outpost strategies and architectures. **Journal of Spacecraft and Rockets**, v. 46, n. 2, p. 419,2009.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **Multi-Mission Platform Attitude Control and Data Handling (ACDH) subsystem specification**. São José dos Campos, 2001.

JURKIEWICZ, S. **Grafos—uma introdução**. UFRGS, 2009. Apostila

KALMAN, R. E. Contributions to the theory of optimal control. **Boletín de la Sociedad Matematica Mexicana**, v.5, p. 102-119, 1960.

_____. On the general theory of control systems. In: FIRST INTERNATIONAL CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL (IFAC), 1960, London. **Proceedings...** London : Butterworths, 1960. p. 481-492.

KEANE, M. A. et al. **Apparatus for improved general-purpose PID and non-PID controllers**. Patente U.S. Patent n. 6,847,851. 25 de jan de 2005.

KEANE, M. A.; YU, J.; KOZA, J. R. Automatic synthesis of both the topology and tuning of a common parameterized controller for two families of plants using genetic programming. In: GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE (GECCO-2000), 2000, Nevada, USA. **Proceedings...** Nevada, 2000. p. 496-504.

KEANE, M. A.; KOZA, J. R.; STREETER, M J. Automatic synthesis using genetic programming of an improved general-purpose controller for industrially representative plants. In: NASA/DoD CONFERENCE ON EVOLVABLE HARDWARE, 2002. IEEE, 2002, Washington. **Proceedings...** Washington: IEEE, 2002. p. 113-122.

KOZA, J. R. et al. **Method and apparatus for automated design of complex structures using genetic programming**. Patente U.S. Patent n. 6,360,191. 19 de mar de 2002.

KOZA, J. R. et al. **Method and apparatus for automatic synthesis controllers**. Patente U.S. Patent n. 7,117,186. 3 de out de 2006.

KOZA, J. R.; BENNETT III, F. H.; STIFFELMAN, O. **Method and apparatus for designing structures**. Patente U.S. Patent n. 8,356,000. 15 de Jan de 2013.

KOZA, J. R.; KEANE, M.A.; BENNETT III, F. H.; YU, J.; MYDLOWEC, W.; STIFFELMAN, O. Automatic creation of both the topology and parameters for a robust controller by means of genetic programming .In: INTELLIGENT CONTROL/INTELLIGENT SYSTEMS AND SEMIOTICS, 1999/ IEEE INTERNATIONAL SYMPOSIUM ON DECISION CONTROL, 1999, Piscataway, NJ. **Proceedings...** Piscataway, NJ: IEEE . IEEE, 1999. 344-352.

KOZA, J.; KEANE, M. A.; MYDLOWEC, W.; BENNETT III, F. H.; YU, J. Automatic synthesis of both the control law and parameters for a controller for a three-lag plant with five-second delay using genetic programming and simulation techniques. In: AMERICAN CONTROL CONFERENCE, 2000, Chicago. **Proceedings...** Chicago: IEEE, 2000. p. 453-459.

KOZA, J. R.; KEANE, M. A.; BENNETT III, F. H.; YU, J.; MYDLOWEC, W.; STIFFELMAN, O. Automatic synthesis of both the topology and parameters for a robust controller for a nonminimal phase plant and a three-lag plant by means of genetic programming. In: Decision and Control, 1999, Phoenix, Arizona, USA. **Proceedings...** Phoenix, Arizona, USA, IEEE, 1999. p. 5292-5300.

KOZA, J. R.; KEANE, M. A.; YU, J.; BENNETT III, F. H.; MYDLOWEC, W. Automatic creation of human-competitive programs and controllers by means of genetic programming. **Genetic Programming and Evolvable Machines** v.1, n.1/2, p.121-164, 2000.

KOZA, J. R.; STREETER, M. J.; KEANE, M. A. Automated synthesis by means of genetic programming of human-competitive designs employing reuse, hierarchies, modularities, development, and parameterized topologies. In: AAAI SPRING SYMPOSIUM: COMPUTATIONAL SYNTHESIS, 2003, Palo Alto, California. **Proceedings...** Palo Alto: AAAI, 2003. 138-145.

KOO, H. Y. B. **A Meta-language for system architecting**.2005. Thesis (Doctorate in Engineering). Cambridge, MA: MIT, 2005.

LAND, R. A brief survey of software architecture. Västerås, Sweden: Mälardalen University/Department of Computer Engineering, Mälardalen Real-Time Research Center (MRTC) , 2002.

LEE, T. **Complexity theory in axiomatic design**. 2003. Thesis (Doctorate in Engineering) - Massachusetts Institute of Technology, 2003.

LIU, G. P.; R. J. PATTON. Robust control design using eigenstructure assignment and multi-objective optimization. **International Journal of Systems Science** v. 27, n. 9, 1996.

LIU, G. P.; YANG, J-B.; WHIDBORNE, J. F. **Multiobjective optimisation & control**. Edição: Primeira. Research Studies Press Ltd., 2002. Engineering Systems Modelling and Control Series.

LLOYD, S. Measures of complexity: a nonexhaustive list. **Control Systems Magazine**. n. 21. IEEE, p. 7-8, 2001..

LODDER, J. Historical projects in discrete mathematics. In: HISTORY AND PEDAGOGY OF MATHEMATICS, 2012, Daejeon, Korea. **Oral Presentation**, 2012.

MAXWELL, J. C. On governors. **Proceedings of the Royal Society**. v. 16, p. 270-283, 1868.

MARTIN, P-A, J.Y. **A framework for quantifying complexity and understanding its sources**: application to tow large-scale systems. Thesis (Technology and Policy Program)- Massachusetts Institute of Technology (MIT), 2004.

MCDERMID, J. A. Complexity: concept, causes and control. In: IEEE INTERNATIONAL CONFERENCE ON ENGINEERING OF COMPLEX COMPUTER SYSTEMS (ICECCS), 2000, Florence, Italy. **Proceedings...** Florence, Italy, IEEE, 2000. p.2-9.

OSTROSKI, A.; MENONCINI, L. Aplicações práticas da teoria dos grafos. In: SYNERGISMUS SCYENTIFCA UTFPR - 130 ENCONTRO REGIONAL DE

MATEMÁTICA APLICADA E COMPUTACIONAL - XIII ERMAC, 2009, Pato Branco. **Anais...** Pto Branco, 2009.

PARETO, V. **Manuale di economia politica com uma Introduzione alla Scienza Sociale**. Milan: Società Editrice Libreria, 1906.

PHUKAN, A.; KALAVA, M.; PRABHU, V. Complexity metrics for manufacturing control architectures based on software and information flow. **Computers & Industrial Engineering**. v. 49. n. 1, 2005. p. 1-20.

RABUSKE, M.A. **Introdução à teoria dos grafos**. Florianópolis, SC: UFSC, 1992.

RELIASOFT. **RBD Configurations** - k-out-of-n Nodes. ReliaSoft. 01 de 11 de 2010. Disponível em: http://blocksim.reliasoft.com/figs/rbd_type3.htm. Acesso em: 01 de 11 de 2010.

ROCCO, E. M. **Manutenção orbital de constelações simétricas de satélites utilizando manobras impulsivas ótimas com vínculo de tempo**. 2002. Tese (Doutorado em Mecânica Espacial e Controle) -Instituto Nacional de Pesquisas Espaciais: São José Campos.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Station keeping of satellite constellations with time constraint using optimal bi-impulsive maneuvers. In: WINTER, O. C.; PRADO, A. F. B. A. (eds.). **Advances in space dynamics 3 applications in astronautics**. São José dos Campos, SP: Instituto Nacional de Pesquisas Espaciais, 2002.

_____. Multi-objective optimization applied to satellite constellations I: Formulation of the Smallest Loss Criterion. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 54., 2003, Bremen, Alemanha. **Proceedings...** Bremen, 2003.

_____. Further applications of the smallest loss criterion in the multi-objective optimization of a satellite constellations. In: INTERNATIONAL ASTRONAUTICAL CONGRESS (IAC 2005), 56., 2005, Fukuoka, Japão. **Proceedings...** Fukuoka, 2005.

_____. Multi-objective optimization applied to satellite constellations II: initial applications of the smallest loss criterion. In: IWSCFF INTERNATIONAL

WORKSHOP ON SATELLITE CONSTELLATIONS AND FORMATION FLYING, 4. São José dos Campos, Brasil. **Proceedings...** São José dos Campos, 2005. p. 123-132. Papel. (INPE-13374-PRE/8589). Disponível em: <<http://urlib.net/sid.inpe.br/iris@1916/2005/10.06.13.14>>. Acesso em: 30 out. 2013., 2005a.

_____. Station keeping of constellations using multiobjective strategies.

Mathematical Problems In Engineering, v. 2013, n. 476451, p. 15, 2013.

doi: <[10.1155/2013/476451](https://doi.org/10.1155/2013/476451)>.

ROSS, A. M. **Multi-attribute tradespace exploration with concurrent design as a value-centric framework for space system architecture and design**. Dissertação (Master of Science in Technology and Policy and Master of Science in Aeronautics and Astronautics) - MIT, Cambridge, MA, 2003.

SIMMONS, W. L.; KOO, B. H. Y.; CRAWLEY, E. F. Architecture generation for Moon-Mars exploration using an executable meta-language. In: AIAA SPACE, 2005, Long Beach, CA. **Proceedings...** American Institute of Aeronautics and Astronautics (AIAA), 2005. 17. (Paper AIAA-2005-6726).

SIMON, F.; PINHEIRO, G.; LOUREIRO, G. Towards Automatic Systems Architecting. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 14. (CE 2007), 2007, São José dos Campos. **Proceedings...** São José dos Campos: INPE, 2007. p. 113-126. CD-ROM; On-line. Disponível em: <<http://urlib.net/dpi.inpe.br/ce@80/2007/04.15.01.43>>. Acesso em: 30 out. 2013.

SHEPPERD, M. **Software engineering metrics I: measures and validations**. Maidenhead, U.K: McGraw-Hill, 1993.

SOUZA, P. N. **Curso introdutório em tecnologia de satélites**. São José dos Campos: INPE, 2008. São José dos Campos: INPE, 2003. (INPE-9605-PUD/126).

SOUSA, F. L. **Otimização extrema generalizada: um novo algoritmo estocástico para o projeto ótimo**. 2002. 142 p. (INPE-9564-TDI/836). Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos

Campos, 2002. Disponível em:

<<http://urlib.net/sid.inpe.br/marciana/2003/03.18.15.39>>. Acesso em: 26 ago. 2013.

TAKAHASHI, Y.; RABINS, M. J.; AUSLANDER, D. M. **Control and dynamic systems**. Addison-Wesley Publishing Company, Inc., 1970.

ULRICH, K. T.; EPPINGER, S. D. **Product design and development**. 2. ed. New York, NY: Irwin/McGraw-Hill, 2000.

VENDITTI, F. C. F. **Otimização multiobjetivo de trajetórias para Plutão**. 2009. Dissertação (Mestrado Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Ampos.

VENDITTI, F. C. F.; ROCCO, E. M.; PRADO, A. F. B. A. ; SUHKANOV, A. Gravity-assisted maneuvers applied in the multi-objective optimization of interplanetary trajectories. **Acta Astronautica** (Elsevier) 67, n. 9 (2010): 1255-1271.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **Transactions of ASME**. p. 759-768, 1942.