

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FERNANDO PALMIERI FREIRE

**Projeto de controladores avançados para o piloto automático do
avião Cessna 172/182**

Maringá

2018

FERNANDO PALMIERI FREIRE

**Projeto de controladores avançados para o piloto automático do
avião Cessna 172/182**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Nardênio Almeida
Martins

Maringá
2018

Dados Internacionais de Catalogação-na-Publicação (CIP)
(Biblioteca Central - UEM, Maringá – PR., Brasil)

Freire, Fernando Palmieri

F866p Projeto de controladores avançados para o piloto automático do avião Cessna 172/182/ Fernando Palmieri Freire. -- Maringá, 2018.
162 f., il., figs., tabs.

Orientador: Prof. Dr. Nardênio Almeida Martins.

Dissertação (mestrado) - Universidade Estadual de Maringá, Centro de Tecnologia, Programa de Pós-graduação em Ciência da Computação, 2018.

1. Aviões - Sistema de controle. 2. Automação. 3. Microcontroladores. 4. Piloto automático do Avião Cessna 172/182. 5. Model-in-the-toop. 6. Software-in-the-loop. 7. Hardware-in-the-loop. 8. ISA-PID-EG-Fuzzy. I. Freire, Fernando Palmieri, orient. II. Universidade Estadual de Maringá. Centro de Tecnologia. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ED.629.836

Jane Lessa Monção CRB 1173/97

FOLHA DE APROVAÇÃO

FERNANDO PALMIERI FREIRE

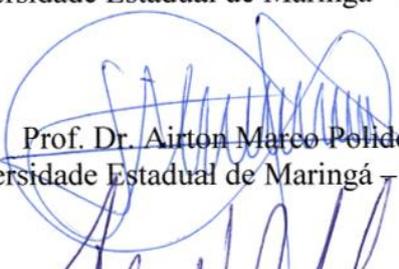
Projeto de controladores avançados para o piloto automático do avião Cessna 172/182

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

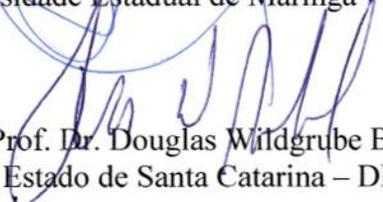
BANCA EXAMINADORA



Prof. Dr. Nardênio Almeida Martins
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Ailton Marco Polidoro
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Douglas Wildgrube Bertol
Universidade do Estado de Santa Catarina – DEE/UEDESC-Joinville

Aprovada em: 16 de fevereiro de 2018.

Local da defesa: Sala 101, Bloco C56, *campus* da Universidade Estadual de Maringá.

DEDICATÓRIA

Dedico esse trabalho à minha família, em especial à minha mãe, Vera Lucia Palmieri, que enquanto em vida sempre me incentivou a estudar.

AGRADECIMENTOS

Agradeço as pessoas que contribuíram na realização deste trabalho. Em especial:

A Mônica, minha esposa, pelo companheirismo, carinho, paciência, dedicação e amor incondicional, sendo um refúgio nos momentos difíceis passados durante esta jornada.

A minha família e amigos pelo apoio e incentivo.

Ao meu orientador Professor Dr. Nardênio Almeida Martins pelo apoio, conselhos, comentários e sugestões no desenvolvimento deste projeto, a quem passei a admirar pelo profissionalismo, honestidade e disciplina.

Aos demais professores das disciplinas que cursei e aos colegas.

E a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

Projeto de controladores avançados para o piloto automático do avião Cessna 172/182

RESUMO

Há uma crescente demanda por sistemas de controle automático para veículos aéreos não tripulados devido ao grande número de aplicações civis e militares. Um sistema de piloto automático em uma aeronave busca melhorar os mecanismos de direção e pilotagem de um avião, e é composto por controladores complexos e sofisticados. Este trabalho contribui com o avanço no desenvolvimento desses sistemas, e têm como objetivo geral projetar um piloto automático com controladores avançados para o avião Cessna 172/182. A técnica de controle clássica, que usa controladores PID, é a mais utilizada por ser eficiente e facilmente desenvolvida, porém possui limitações. Variações não previstas na aeronave e perturbações durante o voo podem não ser compensadas pelo sistema de controle. Por isso o uso de técnicas de controle avançado se justifica para melhorar a eficiência e a confiabilidade do sistema. Neste trabalho também são propostos procedimentos de otimização dos ganhos dos controladores PID projetados a partir de funções de transferência simuladas no software Matlab/Simulink, estabelecendo um sistema *model-in-the-loop*, e então com a integração do software Matlab/Simulink e o simulador de voo X-Plane, formando um sistema *software-in-the-loop*. A implementação dos controladores projetados foi feita em uma placa Arduino, que foi acoplada ao simulador de voo, constituindo um sistema *hardware-in-the-loop*. Neste contexto, resultados de simulações são adquiridos para permitir verificar a efetividade do uso das propostas de otimização feitas neste trabalho. Várias opções de controladores avançados como alternativas ao controlador PID foram usadas, o que permitiu propor o controlador ISA-PID-EG-*Fuzzy*. Os resultados obtidos neste trabalho apontam que o controlador proposto obteve melhores resultados na maioria das comparações feitas e que ainda existem outras possibilidades que devem ser consideradas em futuras implementações.

Palavras-chave: Piloto automático do Avião Cessna 172/182. *Model-in-the-loop*. *Software-in-the-loop*. *Hardware-in-the-loop*. ISA-PID-EG-*Fuzzy*.

Design of advanced controls for the Cessna 172/182 aircraft autopilot

ABSTRACT

There is a growing demand for automatic control systems for unmanned aircraft due to the large number of civilian and military applications. An autopilot system in an aircraft seeks to improve the steering and pilotage mechanisms of an airplane, and is composed of complex and sophisticated controllers. This work contributes to the advancement in the development of these systems, and its general objective is to design an autopilot with advanced controllers for the Cessna 172/182 airplane. The classical control technique, which uses PID controllers, is the most used because it is efficient and easily developed, but it has limitations. Variations not foreseen in the aircraft and disturbances during the flight may not be compensated by the control system. Therefore the use of advanced control techniques is justified to improve the efficiency and reliability of the system. In this work we also propose procedures to optimize the gains of PID controllers designed from simulated transfer functions in the Matlab/Simulink software, establishing a model-in-the-loop system, and then with the integration of Matlab/Simulink and the X-Plane flight simulator, forming a software-in-the-loop system. The implementation of the designed controllers was done on an Arduino board, which was coupled to the flight simulator, constituting a hardware-in-the-loop system. In this context, results of simulations are acquired allowing to verify the effectiveness of the use of the optimization proposals made in this work. Several advanced controller options, as alternatives to the PID controller, were used, which allowed the proposed ISA-PID-EG-*Fuzzy* controller to be proposed. The results obtained in this work indicate that the proposed controller obtained better results in most of the comparisons made and that there are still other possibilities that should be considered in future implementations.

Keywords: Cessna 172/182 Aircraft Autopilot. Model-in-the-loop. Software-in-the-loop. Hardware-in-the-loop. ISA-PID-EG-*Fuzzy*.

LISTA DE FIGURAS

2.1	Malha de controle do ângulo de arfagem	28
2.2	Resposta da malha de controle do ângulo de arfagem	28
2.3	Malha de controle de altitude	29
2.4	Resposta da malha de controle de altitude	30
2.5	Malha de controle do ângulo de rolamento	32
2.6	Resposta da malha de controle do ângulo de rolamento	33
2.7	Malha de controle da taxa de guinada	34
2.8	Resposta da malha de controle da taxa de guinada	34
3.1	Blocos de otimização em conjunto com malha de controle do ângulo de arfagem	46
3.2	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com otimização dos ganhos na plataforma MIL.	47
3.3	Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude com otimização dos ganhos na plataforma MIL.	48
3.4	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento com otimização dos ganhos na plataforma MIL.	49
3.5	Resposta a uma entrada do tipo degrau unitário da malha de con- trole da taxa de guinada com otimização dos ganhos na plataforma MIL.	49
3.6	Plataforma SIL.	51
3.7	Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de arfagem.	52
3.8	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem na plataforma SIL.	52
3.9	Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle de altitude.	53
3.10	Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude na plataforma SIL.	54
3.11	Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de rolamento.	55
3.12	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento na plataforma SIL.	55

3.13	Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de guinada.	56
3.14	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada na plataforma SIL.	57
3.15	Plataforma HIL.	58
3.16	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de arfagem.	59
3.17	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem na plataforma HIL.	60
3.18	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle de altitude.	60
3.19	Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude na plataforma HIL.	61
3.20	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de rolamento.	61
3.21	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento na plataforma HIL.	62
3.22	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de guinada.	62
3.23	Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada na plataforma HIL.	63
3.24	Modelo do Simulink usado na plataforma SIL com malhas de controle para realização dos experimentos.	66
3.25	Resposta da malha de controle do ângulo de arfagem na plataforma SIL.	67
3.26	Resposta da malha de controle de altitude na plataforma SIL.	67
3.27	Resposta da malha de controle do ângulo de rolamento na plataforma SIL.	68
3.28	Resposta da malha de controle do ângulo de guinada na plataforma SIL.	69
3.29	Modelo do Simulink usado na plataforma HIL com o controle de navegação para realização dos experimentos.	72
3.30	Resposta da malha de controle do ângulo de arfagem na plataforma HIL.	73
3.31	Resposta da malha de controle de altitude na plataforma HIL.	73

3.32	Resposta da malha de controle do ângulo de rolamento na plataforma HIL.	74
3.33	Resposta da malha de controle do ângulo de guinada na plataforma HIL.	75
4.1	Resposta controle <i>relay</i> da malha de controle do ângulo de arfagem na plataforma MIL.	80
4.2	Resposta controle <i>relay</i> da malha de controle do ângulo de arfagem na plataforma SIL.	81
4.3	Estrutura do Controlador <i>ISA</i> -PID.	81
4.4	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID na plataforma MIL.	83
4.5	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma SIL.	84
4.6	Estrutura do controlador <i>Fuzzy</i> PID.	86
4.7	Estrutura do controlador <i>Fuzzy</i> PD.	87
4.8	Conjunto <i>fuzzy</i> para a entrada do erro.	87
4.9	Conjunto <i>fuzzy</i> para a entrada da derivada do erro.	88
4.10	Conjunto <i>fuzzy</i> para a saída.	88
4.11	Conjunto <i>fuzzy</i> para as entradas e saída.	89
4.12	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>Fuzzy</i> PID na plataforma MIL.	90
4.13	Controle PID com escalonamento de ganhos <i>fuzzy</i>	90
4.14	Funções de pertinência para $e(k)$ e $\Delta e(k)$	91
4.15	Funções de pertinência para as saídas K'_p e K'_d	92
4.16	Funções de pertinência para α	92
4.17	Funções de pertinência para a entrada do erro.	96
4.18	Funções de pertinência para a entrada da derivada do erro.	96
4.19	Funções de pertinência para a saída K_p	97
4.20	Funções de pertinência para a saída K_i	97
4.21	Funções de pertinência para a saída K_d	98
4.22	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controladores PID-EG- <i>Fuzzy</i> na plataforma MIL.	99

4.23	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador PID-EG- <i>Fuzzy</i> modificado, baseado em Zhao et al. (1993), na plataforma MIL.	100
4.24	Modelo do controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> para a malha de controle do ângulo de arfagem.	101
4.25	Funções de pertinência para a entrada do erro.	101
4.26	Funções de pertinência para a entrada da derivada do erro.	102
4.27	Funções de pertinência para a saída K_p , K_i e K_d	102
4.28	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma MIL.	103
4.29	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma SIL.	105
5.1	Projeto do sistema de controle <i>ISA</i> -PID-EG- <i>Fuzzy</i>	108
5.2	Funções de pertinência para as entradas de erro e derivada do erro na plataforma HIL antes das modificações.	109
5.3	Funções de pertinência para as entradas de erro e derivada do erro na plataforma HIL depois das modificações.	109
5.4	Funções de pertinência para as saídas K_{pr} , K_{ir} e K_{dr} na plataforma HIL.	110
5.5	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de arfagem proposta.	113
5.6	Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL.	114
5.7	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de altitude proposta.	115
5.8	Comparação da resposta ao degrau com distúrbio da malha de controle de altitude com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL.	116
5.9	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de rolamento proposta.	117

5.10	Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de rolamento com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	118
5.11	Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de guinada proposta.	119
5.12	Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de guinada com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	120
5.13	Comparação da resposta na simulação 1 da malha de controle do ângulo de arfagem com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	120
5.14	Comparação da resposta na simulação 1 da malha de controle de altitude com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL. .	121
5.15	Comparação da resposta na simulação 1 da malha de controle do ângulo de rolamento com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	122
5.16	Comparação da resposta na simulação 1 da malha de controle do ângulo de guinada com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	122
5.17	Comparação da resposta na simulação 2 da malha de controle do ângulo de arfagem com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	125
5.18	Comparação da resposta na simulação 2 da malha de controle de altitude com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL. .	125
5.19	Comparação da resposta na simulação 2 da malha de controle do ângulo de rolamento com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	126
5.20	Comparação da resposta na simulação 2 da malha de controle do ângulo de guinada com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	127
5.21	Comparação da resposta na simulação 3 da malha de controle do ângulo de arfagem com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	130
5.22	Comparação da resposta na simulação 3 da malha de controle de altitude com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL. .	130

5.23	Comparação da resposta na simulação 3 da malha de controle do ângulo de rolamento com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	131
5.24	Comparação da resposta na simulação 3 da malha de controle do ângulo de guinada com controlador <i>ISA-PID-EG-Fuzzy</i> na plataforma HIL.	131

LISTA DE TABELAS

2.1	Especificações de projeto para o controle longitudinal	27
2.2	Especificações de projeto para o controle latero-direcional	31
3.1	Comparação de índices de desempenho da otimização na plataforma MIL.	50
3.2	Comparação de índices de desempenho da otimização na plataforma SIL.	58
3.3	Comparação de índices de desempenho da otimização na plataforma HIL.	64
3.4	Ganhos obtidos das simulações.	65
3.5	Respostas da malha de controle de altitude na plataforma SIL. . .	68
3.6	Respostas da malha de controle latero-direcional na plataforma SIL.	70
3.7	Comparação de índices de desempenho da otimização no experimento na plataforma SIL.	71
3.8	Respostas da malha de controle de altitude na plataforma HIL. . .	74
3.9	Respostas da malha de controle latero-direcional na plataforma HIL.	76
3.10	Comparação de índices de desempenho da otimização no experimento na plataforma HIL.	77
4.1	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID na plataforma MIL.	83
4.2	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma SIL.	85
4.3	Regras <i>fuzzy</i> para o controlador.	89
4.4	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>Fuzzy</i> PID na plataforma MIL.	89
4.5	Regras <i>fuzzy</i> para K'_p	93
4.6	Regras <i>fuzzy</i> para K'_d	93
4.7	Regras <i>fuzzy</i> para α	93
4.8	Regras <i>fuzzy</i> para K_p	95
4.9	Regras <i>fuzzy</i> para K_d	95
4.10	Regras <i>fuzzy</i> para K_i	95

4.11	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controladores PID-EG- <i>Fuzzy</i> na plataforma MIL.	99
4.12	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador PID-EG- <i>Fuzzy</i> modificado, baseado em Zhao et al. (1993), na plataforma MIL.	100
4.13	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma MIL.	103
4.14	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma SIL.	105
5.1	Regras <i>fuzzy</i> para K_p e K_i	110
5.2	Regras <i>fuzzy</i> para K_d	110
5.3	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma HIL.	112
5.4	Comparação da resposta ao degrau com distúrbio (D) da malha de controle de altitude na plataforma HIL.	114
5.5	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de rolamento na plataforma HIL.	115
5.6	Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de guinada na plataforma HIL.	116
5.7	Respostas da malha de controle de altitude com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL.	121
5.8	Respostas da malha de controle latero-direcional com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL na simulação 1.	123
5.9	Índices de desempenho da simulação 1 na plataforma HIL	124
5.10	Respostas da malha de controle de altitude com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL na simulação 2.	126
5.11	Respostas da malha de controle latero-direcional com controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na plataforma HIL na simulação 2.	128
5.12	Índices de desempenho da simulação 2 na plataforma HIL.	129
5.13	Índices de desempenho da simulação 3 na plataforma HIL.	132

LISTA DE SIGLAS E ABREVIATURAS

AFCS: Automatic Flight Control System
CLF: Controle baseado em Lógica *Fuzzy*
EG: Escalonamento de Ganhos
ECM: Esforço de Controle Médio
EMQ: Erro Médio Quadrático
 K_{cr} : Ganho crítico
 K_d : Ganho derivativo do sistema
 K_i : Ganho integral do sistema
 K_p : Ganho proporcional do sistema
 K_{rg} : Ganho da taxa do giroscópio
HIL: *Hardware-in-the-loop*
 h : Altitude
ISA: *International Society of Automation*
MIL: *Model-in-the-loop*
PID: Proporcional, Integral, Derivativo
 S_i : Pólo imaginário
SIL: *Software-in-the-loop*
VANT: Veículo Aéreo Não Tripulado
VSC: Variância do Sinal de Controle
 α_r : Ângulo de ataque
 δ_a : Ângulo de deflexão dos *aileron*s
 δ_e : Ângulo de deflexão do profundor
 δ_r : Ângulo de deflexão do leme
 ϕ : Ângulo de rolamento
 ψ : Ângulo de guinada
 θ : Ângulo de arfagem
 ζ : Coeficiente de amortecimento
ZN: Ziegler-Nichols

SUMÁRIO

1	Introdução	18
1.1	Motivação e Justificativa	19
1.2	Formulação do Problema	20
1.3	Soluções Propostas	20
1.4	Objetivos e Contribuições	20
1.5	Organização do Texto	22
2	Fundamentação teórica	23
2.1	Controle Automático	23
2.2	Definições	24
2.3	Controladores PID	25
2.4	Controle de Aeronave	26
2.4.1	Controle Longitudinal	26
2.4.2	Controle Latero-Direcional	31
2.5	Trabalhos Relacionados	35
2.6	Considerações Finais	43
3	Otimização de Ganhos de Sintonia de Controladores PID	44
3.1	Otimização dos Ganhos: Procedimentos e Índices de Desempenho	44
3.2	Otimização com Plataforma MIL a partir das Funções de Transferência	46
3.2.1	Controle Longitudinal	46
3.2.2	Controle Latero-direcional	48
3.3	Otimização com Plataforma SIL	50
3.3.1	Controle Longitudinal	51
3.3.2	Controle Latero-direcional	54
3.4	Otimização com Plataforma HIL	57
3.4.1	Controle Longitudinal	58
3.4.2	Controle Latero-direcional	60
3.5	Experimento	64
3.5.1	Plataforma SIL	64
3.5.2	Plataforma HIL	72
3.6	Considerações	75
4	Controles Avançados	79
4.1	Procedimento de Auto-sintonia do Controlador PID com o Método <i>Relay</i>	79

4.2	Controlador <i>ISA</i> -PID	80
4.2.1	Teste do Controlador <i>ISA</i> -PID na Malha do Ângulo de Arfagem Usando Plataforma <i>MIL</i>	82
4.2.2	Teste do Controlador <i>ISA</i> -PID na Malha do Ângulo de Arfagem Usando Plataforma <i>SIL</i>	83
4.3	Controladores <i>Fuzzy</i> PID	85
4.3.1	Teste do Controlador <i>Fuzzy</i> PID na Malha do Ângulo de Arfagem Usando Plataforma <i>SIL</i>	86
4.4	Controlador PID-EG- <i>Fuzzy</i>	89
4.4.1	Teste do Controlador PID-EG- <i>Fuzzy</i> Usando Plataforma <i>MIL</i>	94
4.5	Controlador PID-EG- <i>Fuzzy</i> integrado com estrutura <i>ISA</i> -PID	100
4.5.1	Teste do Controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> Usando Plataforma <i>MIL</i>	102
4.5.2	Teste do Controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> Usando Plataforma <i>SIL</i>	104
4.6	Considerações	104
5	Controlador <i>ISA</i>-PID-EG-<i>Fuzzy</i> Proposto: Projeto, Resultados e Análises	107
5.1	Controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> Proposto	107
5.1.1	Implementação do Controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> na Plataforma HIL	111
5.1.2	Teste do Controlador <i>ISA</i> -PID-EG- <i>Fuzzy</i> Usando Plataforma HIL	112
5.2	Casos de Simulação	117
5.2.1	Simulação 1	118
5.2.2	Simulação 2	123
5.2.3	Simulação 3	127
5.3	Considerações	128
6	Conclusão	133
6.1	Contribuições	134
6.2	Dificuldades Encontradas	134
6.3	Trabalhos Futuros	134
	REFERÊNCIAS	136
A	Código do Controlador Implementado no Arduino	143

Introdução

Os sistemas de controle automático representam um papel importante no crescimento da aviação, pois contribuem para melhorar os procedimentos de navegação, o gerenciamento de voo e a estabilidade das aeronaves (Nelson, 1998).

A estabilidade de uma aeronave depende dos fatores que a fazem resistir a qualquer mudança em seu vetor de velocidade e direção tornando a manutenção de um voo não acelerado constante. Como o percurso de uma aeronave geralmente é instável e aeronaves sem controles tendem a voar em giro constante, a necessidade de correções nas superfícies de comando são efetuadas continuamente por meio de um piloto humano ou de um piloto automático. O piloto automático, do inglês *Automatic Flight Control System (AFCS)*, tem o propósito de controlar uma aeronave em um voo com trajetória pré-programada sem a intervenção humana. Também tem como função melhorar a precisão e a velocidade de resposta dos comandos aplicados às superfícies de controle e estabilizar a aeronave quando exposta a perturbações (McLean, 1990).

Para que um sistema de piloto automático possa exercer sua função de forma eficiente são necessárias informações de um sistema de navegação que, juntamente com dados disponibilizados por sensores de atitude e direção, fornecem um conjunto de coordenadas que auxiliam no comando das manobras a serem realizadas pela aeronave. Os sistemas de piloto automático podem ser desenvolvidos de várias formas, desde o mais simples, geralmente utilizado em aeronaves de pequeno porte, até os mais sofisticados, empregados em aviões de grande porte.

A primeira apresentação de um piloto automático que obteve sucesso data de junho de 1914, quando o sistema desenvolvido pelos irmãos Sperry foi instalado e testado em uma aeronave (Blakelock, 1991).

A grande maioria dos pilotos automáticos para aeronaves de asa fixa ainda depende de controladores PIDs, (Proporcional-Integral-Derivativo) (Sartori, 2014). Um controlador convencional linear, como o PID, pode ter bom desempenho quando as não-linearidades são bastante suaves ou um processo altamente não-linear opera em uma estreita faixa de condições. Para processos altamente não-lineares, as estratégias de controle linear podem não ser adequadas. Para essas situações, as estratégias de controle não-linear podem fornecer melhorias significativas em relação ao controle PID (Duan et al., 2008).

1.1 Motivação e Justificativa

Desde os primeiros voos realizados com aeronave, muitos pesquisadores vêm trabalhando continuamente no aperfeiçoamento na sua fabricação. Diversos avanços foram realizados em estudos relacionados a aeronaves que resultaram em melhorias aerodinâmicas e de desempenho, proporcionando projetos aeronáuticos capazes de realizar desde voos transcontinentais até espaciais. No Brasil, desde o feito de Santos Dumont, o estudo aeronáutico é impulsionado pelo desejo de aprimorar e enriquecer a história da aviação no país (Rodrigues, 2013).

Hoje em dia cada vez mais sistemas de controle automático são estudados e explorados devido a possibilidade desses sistemas serem usados em Veículos Aéreos Não Tripulados (VANTs), que, nos últimos anos, se popularizou como objeto de pesquisa nos (Turkoglu et al., 2008; Vural e Hajiyev, 2014). Eles possuem diversas aplicações civis e militares, principalmente em tarefas de monitoramento de áreas com finalidades ambientais (incêndios, desmatamentos, mapeamentos), vigilância e monitoramento de regiões urbanas, busca de sobreviventes de naufrágios e desastres naturais, agricultura de precisão, inspeção de linhas de transmissão de energia elétrica, transporte de cargas, análises atmosféricas, reconhecimento de áreas remotas e de risco, armas estratégicas, etc.

Dentro deste contexto, o presente trabalho visa contribuir com as técnicas que auxiliam no desenvolvimento de sistemas de piloto automático na aviação. Em particular, este projeto utiliza a aeronave Cessna 182 por assemelhar-se ao aeromodelo disponível em laboratório para testes futuros e por assemelhar-se ao modelo Cessna 172 disponível no simulador de voo X-Plane, dando continuidade ao trabalho apresentado em Splendor et al. (2015).

1.2 Formulação do Problema

Uma desvantagem da abordagem clássica, utilizando controladores PID simples, é a incapacidade de lidar com algumas condições de voo que podem ser necessárias na maioria das missões não triviais. Outra desvantagem é que ela não garante robustez suficiente quanto à extensão das incertezas paramétricas do modelo que podem ocorrer. Com isso existe a necessidade de desenvolver leis de controle não-lineares que busquem garantir o desempenho satisfatório nas mais diversas condições de voo e também na presença de incertezas.

1.3 Soluções Propostas

A revisão consistiu em um levantamento bibliográfico sobre pilotos automáticos para o Cessna 182 e modelos similares, pilotos automáticos usando controle PID, *fuzzy*, *fuzzy* tipo PID, PID com escalonamento de ganhos e PID com dois graus de liberdade. Também foram levantadas publicações que servem como base de conhecimento para o desenvolvimento de pilotos automático, modelagem de aeronaves e métodos de controle.

Neste contexto este trabalho propõe uma nova configuração de sistema de piloto automático para aeronaves utilizando controladores avançados. Em cada controlador é combinado o controle PID tradicional com escalonamento de ganhos, utilizando lógica *fuzzy*, e um método de filtragem do sinal de referência, o filtro ISA. Também é proposto neste trabalho procedimentos de otimização dos controladores PID pois, além de ainda representarem uma técnica potencialmente relevante em pilotos automáticos modernos (Sartori, 2014), é possível com isso avaliar qual o desempenho alcançável com o uso desses controladores e comparar o controle PID otimizado com o controle avançado proposto.

1.4 Objetivos e Contribuições

O objetivo geral deste trabalho é apresentar um projeto de desenvolvimento e implementação de um piloto automático, com o uso de controladores avançados, para aeronaves, utilizando como referência um Cessna 182, já iniciado em Splendor et al. (2015). O projeto deste piloto automático é composto por um sistema de controle longitudinal, responsável pela manutenção da altitude e do ângulo de arfagem, e de um sistema de controle latero-direcional, responsável pela manutenção dos ângulos de rolamento e

guinada. Os controladores propostos são desenvolvidos com o uso das equações dinâmicas que representam o Cessna 182 (Roskam, 1995).

Uma das técnicas muito empregada em projetos relacionados ao desenvolvimento de piloto automático é baseada nas leis de Newton para um corpo rígido. As equações diferenciais que descrevem a dinâmica de uma aeronave geralmente são formadas por equações não-lineares que são apresentadas de forma simplificadas para os sistemas reais por envolverem muitas características que tornariam o sistema ainda mais complexo. Os modelos matemáticos do comportamento dinâmico de uma aeronave são de fundamental importância para a simulação dos controladores (Santos e Oliveira, 2011).

A partir desses controladores já desenvolvidos, são projetados novos controladores utilizando lógica *fuzzy* para torná-los capazes de realizar a compensação de incertezas e perturbações. Com isso espera-se contribuir com o avanço no desenvolvimento de sistemas de piloto automático e no domínio da tecnologia de pilotos automáticos para aeronaves.

No trabalho de Splendor et al. (2015) foram desenvolvidos controladores PID para as malhas de controle de longitudinal (arfagem e altitude) e latero-direcional (rolamento e guinada) para o Cessna 182. Neste trabalho foi feita a otimização dos ganhos desses controladores utilizando o *Simulink Design Optimization* disponível para o Matlab/Simulink. Os resultados são apresentados no capítulo 3. Outros estudos serão realizados para a análise de rastreamento da referência desejada em relação à tarefa realizada com o intuito de obtenção dos ganhos dos controladores PID.

Também é dada continuidade ao levantamento bibliográfico com trabalhos relacionados às técnicas de controle avançadas como PID *fuzzy* utilizadas nos desenvolvimentos de pilotos automáticos. Com isso espera-se revelar o estado da arte em relação a sistemas de piloto automático para aeronaves.

Por fim, é feito o desenvolvimento de um piloto automático avançado com o uso dos controladores propostos para o Cessna 182. O diferencial deste trabalho em relação aos já existentes está na integração da técnica de controle PID com o escalonamento de ganhos *fuzzy* e filtro de referência ISA com o sistema de piloto automático para aeronaves de asa fixa. Isso contribui para o aumento da estabilidade durante o voo quando sujeito a incidência de incertezas e/ou perturbações. Os controladores projetados foram implementados em um microcontrolador Arduino e testados por meio de simulações integrando o Matlab/Simulink e o simulador voo X-Plane 10. Os resultados apresentados demonstram a efetividade da estratégia de controle proposta.

1.5 Organização do Texto

Este trabalho está dividido em 6 capítulos. O capítulo 2 faz uma breve descrição dos conceitos básicos sobre controle automático, dos principais componentes de uma aeronave, das forças e movimentos que atuam em um avião. Os conceitos sobre sistemas de coordenadas e controle de aeronaves são também apresentados. Por fim é feita uma revisão bibliográfica sobre trabalhos relacionados com piloto automático de aeronaves usando controles avançados. No capítulo 3 são apresentados procedimentos de otimização dos ganhos dos controladores PID, a partir das funções de transferência simuladas no software Matlab/Simulink, estabelecendo um sistema *model-in-the-loop* (MIL) para o piloto automático do avião Cessna 182 e, posteriormente, com a integração do software Matlab/Simulink e o simulador de voo X-Plane, constituindo num sistema *software-in-the-loop* (SIL), para o piloto automático do avião Cessna 172. Também foi feita a implementação dos controladores usando uma placa Arduino que foi testada em conjunto com o simulador de voo, formando um sistema *hardware-in-the-loop* (HIL). O capítulo 4 apresenta um procedimento para determinação inicial dos ganhos de um controlador PID. Depois são descritas algumas variações e alternativas ao controle PID clássico, considerados controles avançados, que são experimentados e comparados. Com base neles é definido, no capítulo 5, o controlador ISA-PID-EG-*Fuzzy*, proposto neste trabalho. Também são apresentados os resultados obtidos nas simulações realizadas com o controlador proposto. Por fim, no capítulo 6 são apresentadas as conclusões, dificuldades encontradas no desenvolvimento deste trabalho e os trabalhos futuros.

Fundamentação teórica

Este capítulo apresenta os conceitos básicos sobre controle automático. Também é feita uma breve descrição dos principais componentes de uma aeronave, as forças e movimentos que atuam nela. Os conceitos sobre sistemas de coordenadas e controle de aeronaves são também apresentados. Por fim, é feita uma revisão bibliográfica sobre trabalhos relacionados com piloto automático de aeronaves usando controles avançados.

2.1 Controle Automático

As teorias de controle comumente usadas hoje são a teoria de controle clássica (também chamada de teoria de controle convencional), teoria de controle moderna e teoria de controle robusta. O controle automático é essencial em qualquer campo de engenharia e ciência. É uma parte importante dos sistemas de veículos aeroespaciais, sistemas robotizados, sistemas de fabricação modernos e quaisquer operações industriais que envolvam controle de temperatura, pressão, umidade, fluxo, etc (Ogata, 2010).

Os métodos de resposta no domínio da frequência e do lugar das raízes, que são o núcleo da teoria clássica de controle, levam a sistemas estáveis que satisfazem um conjunto de requisitos de desempenho mais ou menos arbitrários. Tais sistemas são, em geral, aceitáveis, mas não ótimos em qualquer sentido significativo. À medida que as plantas modernas com muitas entradas e saídas se tornam cada vez mais complexas, a descrição de um sistema de controle moderno requer uma grande quantidade de equações. A teoria do controle clássico, que trata apenas de sistemas de entrada única e de saída única, torna-se impotente para sistemas de múltipla entrada e múltipla saída. A disponibilidade de

computadores digitais possibilitou a análise no domínio do tempo de sistemas complexos. A teoria do controle moderno, baseada na análise e síntese no domínio do tempo, usando variáveis de estado, foi desenvolvida para lidar com a crescente complexidade das plantas modernas e com requisitos rigorosos de precisão, peso e custo em aplicações militares, espaciais e industriais (Ogata, 2010).

2.2 Definições

Para se discutir sistemas de controle, algumas terminologias básicas devem ser definidas (Ogata, 2010).

- **Variável Controlada.** A variável controlada é a quantidade ou condição que é medida e controlada. É a saída do sistema.
- **Sinal de Controle.** O sinal de controle (também chamado de variável manipulada) é a quantidade ou condição que é variada pelo controlador. Controle significa medir a variável controlada do sistema e aplicar o sinal de controle ao sistema para corrigir ou limitar o desvio do valor medido de um valor desejado.
- **Plantas.** Uma planta pode ser um equipamento ou apenas um conjunto de peças de uma máquina funcionando juntas, cujo objetivo é realizar uma operação específica. Qualquer objeto físico a ser controlado (como um dispositivo mecânico, um forno de aquecimento, um reator químico ou uma nave espacial) pode ser chamado de planta.
- **Processo.** Um processo é uma operação ou desenvolvimento natural, progressivamente contínuo, marcado por uma série de mudanças graduais que se sucedem de forma relativamente fixa e conduzem a um resultado ou fim particular. Também pode ser uma operação artificial ou voluntária, progressivamente contínua, que consiste em uma série de ações ou movimentos controlados direcionados sistematicamente para um resultado ou fim particular. Qualquer operação controlada pode ser chamada de processo. Entre os exemplos estão os processos químicos, econômicos e biológicos.
- **Sistema.** Um sistema é uma combinação de componentes que atuam em conjunto e realizam um determinado objetivo. Um sistema não precisa ser físico. O conceito de sistema pode ser aplicado a fenômenos abstratos e dinâmicos como os encontrados na economia. A palavra sistema deve, portanto, ser interpretada como implicando sistemas físicos, biológicos, econômicos e similares.

- **Perturbação.** Uma perturbação é um sinal que tende a afetar adversamente o valor da saída de um sistema.
- **Controle com realimentação.** Controle com realimentação, ou de malha fechada, refere-se a uma operação que, na presença de distúrbios, tende a reduzir a diferença entre a saída de um sistema e alguma entrada de referência e faz isso com base nessa diferença.

Os controladores do tipo PID, controle em cascata e *feedforward* são considerados controladores clássicos, enquanto os controladores adaptativos, preditivos, ótimo, não-linear e inteligentes são classificados como controles avançados (Almeida e Coelho, 2001). Técnicas de controle avançado são, geralmente, empregadas para superar limitações das técnicas convencionais e seu uso deve ser justificado, já que envolve algoritmos mais complexos e conhecimentos matemáticos mais aprimorados que podem dificultar a compreensão das estruturas empregadas, principalmente por parte de operadores não especializados (Coelho et al., 1998).

2.3 Controladores PID

Mais da metade dos controladores industriais em uso hoje são controladores PID ou controladores PID modificados. A utilidade dos controles PID está na sua aplicabilidade geral na maioria dos sistemas de controle. Em particular, quando o modelo matemático da planta não é conhecido e, portanto, métodos de projeto analítico não podem ser utilizados, controles PID se mostram os mais úteis (Ogata, 2010). Essa abordagem é muito popular em uma larga variedade de aplicações, incluindo aviação, onde ainda é dominante hoje em dia (Sartori, 2014). O princípio básico do controlador PID é manipular a variável controlada por meio de três ações de controle: proporcional, integral e derivativa (equação 2.1).

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2.1)$$

onde $u(t)$ é o sinal aplicado para o controle do atuador no instante t , K_p é o ganho proporcional, $e(t)$ é o erro no instante t , T_i é constante de tempo integral, T_d é a constante de tempo derivativo. Assim K_p , $K_i = K_p/T_i$ e $K_d = K_p \times T_d$ são os três parâmetros de sintonia do controlador PID.

A ação proporcional visa realizar um controle razoável do erro. A ação integral garante que a resposta de compensação do sistema seja igual ao valor de referência para que o

erro tenda assintoticamente para zero (erro em regime estacionário). A ação derivativa promove a estabilidade do sistema, diminuindo o tempo necessário para atingir o regime de estado.

Em aplicações práticas alguns cuidados devem ser tomados na utilização desses controladores, de forma que se obtenha o desempenho desejado. Os principais efeitos que devem ser considerados são: a saturação do sinal atuante; a limitação do ganho derivativo em altas frequências; e a escolha do sinal de referência (Neris e Trindade Junior, 2001).

Para obter um desempenho satisfatório para uma determinada aplicação deve-se ajustar os parâmetros do controlador, alterando seu comportamento. A maior parte dos controladores PID é ajustada em campo. Diferentes métodos de ajuste foram sendo propostos, como o método de Cohen e Coon, Astrom e Hagglund, Ziegler e Nichols, entre outros (Ogata, 2010).

2.4 Controle de Aeronave

Um avião com asa fixa é um tipo de aeronave capaz de movimentar-se na atmosfera por meio de sua força motriz, utilizando uma hélice ou turbina para manter seu equilíbrio em relação às forças aerodinâmicas que agem em sua estrutura (Prochaska, 2011). Os aviões são projetados para os mais diversos fins, entretanto, possuem basicamente os mesmos componentes com características operacionais e dimensões diferenciadas de acordo com a sua finalidade (Rodrigues, 2013).

Tem sido desenvolvidas muitas técnicas para controlar um sistema dinâmico usando retroalimentação mas muito poucas técnicas de controle são realmente implementadas nas aplicações de voos no mundo real. A principal razão por trás de não implementar as técnicas avançadas de controle é que elas não são intuitivas e no setor aeroespacial, onde a segurança é uma prioridade alta, técnicas não intuitivas não são suficientemente confiáveis para serem implementadas em aeronaves reais. Nos aviões modernos avançados, os controladores convencionais PID são amplamente utilizados, mesmo que não sejam muito eficientes para sistemas dinâmicos não-lineares, principalmente por sua natureza intuitiva, facilidade de operação e baixo custo (Narenathreyas, 2013).

2.4.1 Controle Longitudinal

O sistema de piloto automático longitudinal tem como objetivo controlar as variáveis relacionadas ao ângulo de arfagem, ao ângulo de ataque e as velocidades vertical e horizontal (Santos e Oliveira, 2011). O piloto automático longitudinal para o Cessna

182 é composto por uma malha de controle interna que controla o ângulo de arfagem e uma malha de controle externa responsável por controlar a altitude da aeronave.

A estabilidade dinâmica longitudinal de uma aeronave é dividida em dois modos: modo de período curto e modo de período longo. Esses modos são gerados em situações em que a aeronave sofre uma perturbação em seu estado de equilíbrio (Cook, 2007).

A Tabela 2.1 apresenta as especificações utilizadas na malha de controle longitudinal para a resposta a uma entrada do tipo degrau unitário.

Tabela 2.1: Especificações de projeto para o controle longitudinal

Controlador	Característica	Valores
Arfagem	Tempo de pico (t_p)	$\leq 1s$
	Máximo sobre-sinal (M_p)	35%
	Amortecimento (ζ)	0.7
Altitude	Tempo de pico (t_p)	$\leq 15s$
	Máximo sobre-sinal (M_p)	30%
	Tempo de acomodação (t_s) critério de 2%	$\leq 20s$

Fonte: Splendor et al. (2015)

Como referência ao controle da malha de arfagem compara-se o ângulo de arfagem desejado θ_{ref} com o valor medido por um giroscópio vertical gerando um sinal de erro θ_e (Splendor et al., 2015). Geralmente o sinal do giroscópio precisa ser amplificado antes de ser enviado para a superfície de controle gerando uma deflexão do profundor (δ_e). Essa amplificação é feita por um controlador P (proporcional) e a magnitude é dada por um ganho chamado aqui de K_p . O movimento causado pela deflexão do profundor proporciona uma mudança no ângulo de arfagem provocando alterações nas forças e momentos atuantes na aeronave em relação ao seu centro de gravidade (Nelson, 1998).

Para melhorar o desempenho do sistema de controle do ângulo de arfagem pode-se adicionar uma malha de controle interna de forma a modificar o coeficiente de amortecimento do modo de período curto (Nelson, 1998). Com a diferença entre o erro do ângulo de arfagem e_a e o erro da taxa de arfagem e_{rg} obtém-se a relação e_{δ_a} resultando assim na malha de controle completa para o ângulo de arfagem. Com os parâmetros de ganho K_p e K_{rg} é possível ajustar o coeficiente de amortecimento, o tempo de subida, o máximo sobre-sinal e, conseqüentemente, ajustar a estabilidade da aeronave (Splendor et al., 2015).

Para determinar os valores dos ganhos relacionados à malha de controle é utilizado o método do lugar das raízes (Ogata, 2010). Primeiramente, determina-se o ganho $K_{rg} = 1.18$ da malha interna utilizando as funções de transferência de malha aberta dadas

pelas equações 2.2 (Blakelock, 1991) e 2.3 (Splendor et al., 2015). Para o coeficiente de amortecimento $\zeta = 0.7$ e utilizando a equação 2.4 é obtido o ganho $K_p = 14.3$.

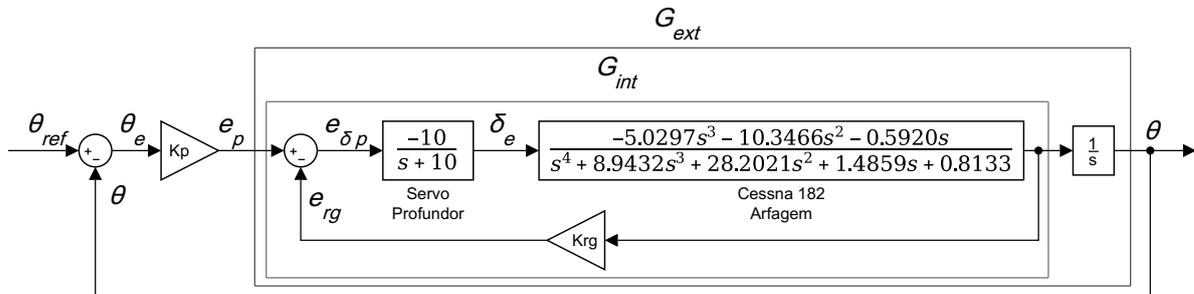
$$G_{ser} = \frac{\delta_e(s)}{e_{\delta a}} = \frac{-10}{s + 10} \quad (2.2)$$

$$G_{\theta} = \frac{\theta(s)}{\delta_e(s)} = \frac{-5.0297s^2 - 10.3466s - 0.5920}{s^4 + 8.9432s^3 + 28.2021s^2 + 1.4859s + 0.8133} \quad (2.3)$$

$$G_{ext} = \frac{1}{s} G_{int} \quad (2.4)$$

A Figura 2.1 mostra a malha de controle do ângulo de arfagem obtida

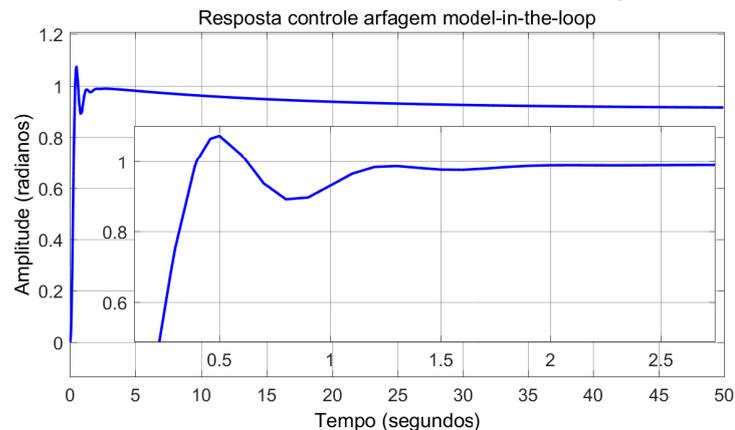
Figura 2.1: Malha de controle do ângulo de arfagem



Fonte: Adaptado de Splendor et al. (2015)

A Figura 2.2 mostra a resposta da malha de controle do ângulo de arfagem.

Figura 2.2: Resposta da malha de controle do ângulo de arfagem



Fonte: Adaptado de Splendor et al. (2015)

Analisando a Figura 2.2 observa-se que a resposta apresenta um tempo de pico de 0.5s e um sobre-sinal de 7.36%, atendendo assim as especificações de projeto do controlador de

arfagem (Tabela 2.1) (Splendor et al., 2015). Mais adiante neste trabalho será mostrado como os ganhos K_p e K_{rg} podem ser otimizados (Subseções 3.2.1, 3.3.1 e 3.4.1).

A malha de controle de altitude visa a manter a altitude h da aeronave minimizando o erro h_e entre a altitude atual h a desejada h_{ref} . Considerando que a velocidade horizontal é controlada por outra malha de controle e conhecendo a equação dinâmica que relaciona a mudança de altitude ocasionada por uma deflexão do profundor é possível obter a função de transferência para o controlador de altitude (Santos e Oliveira, 2011). A função de transferência que relaciona a altitude a uma deflexão do profundor é mostrada na equação 2.5 (Splendor et al., 2015).

$$\frac{h(s)}{\delta_e(s)} = \frac{402.8584s^3 + 68698.4772s^2 + 902.5666s + 2831.2003}{s^5 + 8.9432s^4 + 28.2021s^3 + 1.4859s^2 + 0.8133s} \quad (2.5)$$

A relação entre a variação de altitude e a deflexão do profundor em condições de voo pode ser convertida para $h(s)/\theta(s)$ passando a ter como entrada o ângulo de arfagem do avião utilizando a equação 2.6 (Santos e Oliveira, 2011).

$$\frac{h(s)}{\theta(s)} = \frac{h(s)}{\delta_e(s)} \frac{\delta_e(s)}{\theta(s)} \quad (2.6)$$

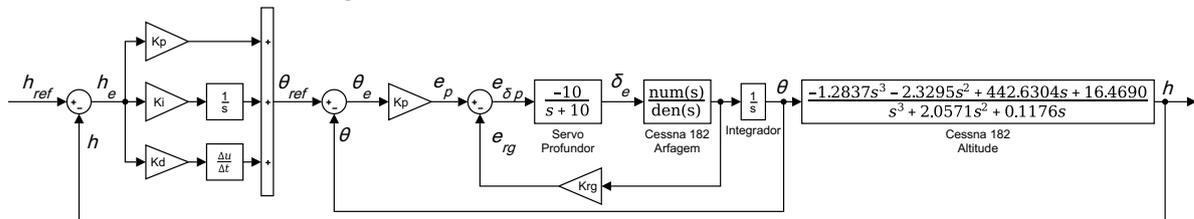
Da equação 2.6 é obtida a função de transferência utilizada no controle de altitude da aeronave como apresentado na equação 2.7 (Splendor et al., 2015).

$$\frac{h(s)}{\theta(s)} = \frac{-1.2837s^3 - 2.3295s^2 + 442.6304s + 16.4690}{s^3 + 2.0571s^2 + 0.1176s} \quad (2.7)$$

Para o controle da malha de altitude é utilizado um controlador PID como mostra a Figura 2.3, onde a função de transferência da malha de controle de altitude é representada pela equação 2.7 (Splendor et al., 2015).

A Figura 2.3 mostra a malha de controle de altitude obtida.

Figura 2.3: Malha de controle de altitude



Fonte: Adaptado de Splendor et al. (2015)

A escolha do controlador PID deve-se ao fato da sua aplicabilidade na maior parte dos sistemas de controle e dos diversos métodos de sintonia disponíveis (Ogata, 2010). Dentre

esses métodos pode-se destacar a segunda regra de *Ziegler-Nichols* (ZN) que propõe a utilização do ganho crítico K_{cr} e o período crítico P_{cr} da malha a ser controlada para determinar os ganhos K_p , K_i e K_d do controlador PID (Splendor et al., 2015).

Com a utilização do método do lugar das raízes e a função de transferência de malha aberta dada pela equação 2.7 é possível localizar o ganho crítico e a parcela imaginária S_i que corresponde ao valor do ponto de cruzamento do eixo imaginário e a curva do lugar das raízes (Santos e Oliveira, 2011).

Foram obtidos os valores de $K_{cr} = 0.0246$ e $S_i = 2.9$. Conhecendo esses valores é possível determinar os ganhos do controlador PID, utilizando as equações 2.8 a 2.11.

$$P_{cr} = 2\pi/S_i = 2.17s \quad (2.8)$$

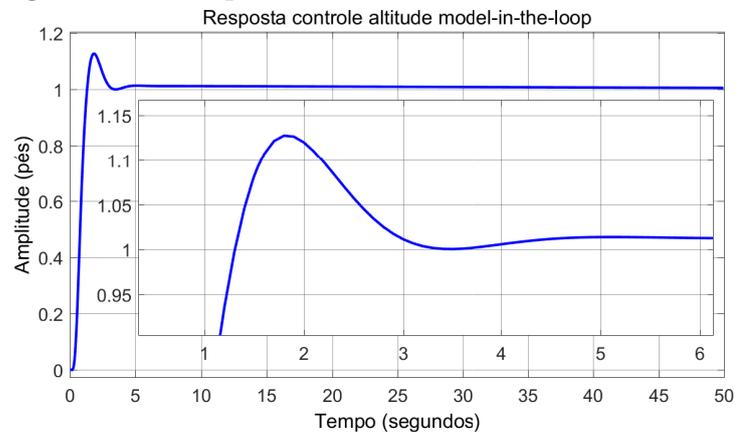
$$K_p = 0.6 \times K_{cr} = 0.0148 \quad (2.9)$$

$$K_i = 0.6 \times K_{cr}/0.5 \times P_{cr} = 0.0136 \quad (2.10)$$

$$K_d = 0.6 \times K_{cr} \times 0.125 \times P_{cr} = 0.0040 \quad (2.11)$$

Como o método de sintonia utilizado fornece uma estimativa para os valores dos ganhos do controlador é possível efetuar um ajuste manual fino para obter-se uma resposta melhor. Por esse motivo os valores utilizados para os ganhos são $K_p = 0.0112$, $K_i = 0.00038$ e $K_d = 0.0032$ (Splendor et al., 2015). A Figura 2.4 mostra a resposta obtida para a malha de controle de altitude.

Figura 2.4: Resposta da malha de controle de altitude



Fonte: Adaptado de Splendor et al. (2015)

Os valores para resposta a uma entrada do tipo degrau obtidos para a malha de controle apresentam um tempo de pico de 4.40s, sobre-sinal de 13.13% e um tempo de acomodação de 5.4s (Splendor et al., 2015).

2.4.2 Controle Latero-Direcional

O sistema de piloto automático latero-direcional está relacionado aos ângulos de rolamento, guinada e derrapagem (Ribeiro e Oliveira, 2010). Para simplificar os estudos dos controladores latero-direcionais para o Cessna 182, é apresentada uma malha de controle para o ângulo de rolamento e para o ângulo de guinada visando proporcionar uma maior estabilidade da aeronave (Splendor et al., 2015).

As especificações utilizadas na malha de controle latero-direcional para a resposta a uma entrada do tipo degrau unitário são apresentadas na Tabela 2.2.

Tabela 2.2: Especificações de projeto para o controle latero-direcional

Controlador	Característica	Valores
Rolamento	Tempo de pico (t_p)	$\leq 5s$
	Máximo sobre-sinal (M_p)	20%
	Tempo de acomodação (t_s) critério de 2%	$\leq 35s$
	Amortecimento (ζ)	0.7
Guinada	Tempo de pico (t_p)	$\leq 10s$
	Máximo sobre-sinal (M_p)	30%
	Tempo de acomodação (t_s) critério de 2%	$\leq 20s$
	Amortecimento (ζ)	$3 \leq \zeta \leq 4$

Fonte: Splendor et al. (2015)

O valor obtido por um giroscópio de inclinação é usado como referência para o controle do ângulo de rolamento ϕ . O valor fornecido pelo giroscópio é comparado a um valor de referência ϕ_{ref} gerando um sinal de erro ϕ_e que atua no controle do servo gerando uma deflexão δ_a nos *aileron*s (os quais são equipamentos móveis instalados no bordo de fuga da asa movimentando-se para cima e para baixo de forma contrária um do outro. A principal finalidade do *aileron* é inclinar o avião para um dos lados. Mantendo-se essa inclinação, a aeronave pode entrar em curva para o lado da asa baixada, acarretando em um movimento de rolamento). Essa deflexão causa uma mudança no ângulo de rolamento procurando mantê-lo no valor desejado (Splendor et al., 2015).

Para a malha de controle do ângulo de rolamento atingir um bom desempenho é necessário incluir uma malha de realimentação interna no sistema de controle com o intuito de aumentar o valor do coeficiente de amortecimento (McLean, 1990).

Com a diferença entre o erro do ângulo de rolamento e_{pid} e o erro da taxa do ângulo de rolamento e_{rg} obtém-se a relação $e_{\delta pid}$, resultando assim na malha de controle interna completa para o ângulo de rolamento. Ressalta-se aqui que $\dot{\phi}$ é a derivada do ângulo de rolamento ϕ (Splendor et al., 2015).

Com a malha de controle interna completa para o ângulo de rolamento é possível utilizar um controlador PID para o controle do ângulo de rolamento ϕ . Para determinar os valores dos ganhos K_{rg} , K_p , K_i e K_d relacionados à malha de controle é utilizado o método do lugar das raízes. Primeiramente, determina-se o ganho $K_{rg} = 0.0803$ da malha interna utilizando as funções de transferência de malha aberta dadas pelas equações 2.12 e 2.13 (Splendor et al., 2015). Utilizando a equação 2.14 são obtidos os valores para $K_{cr} = 2.42$ e $S_i = 12.9$.

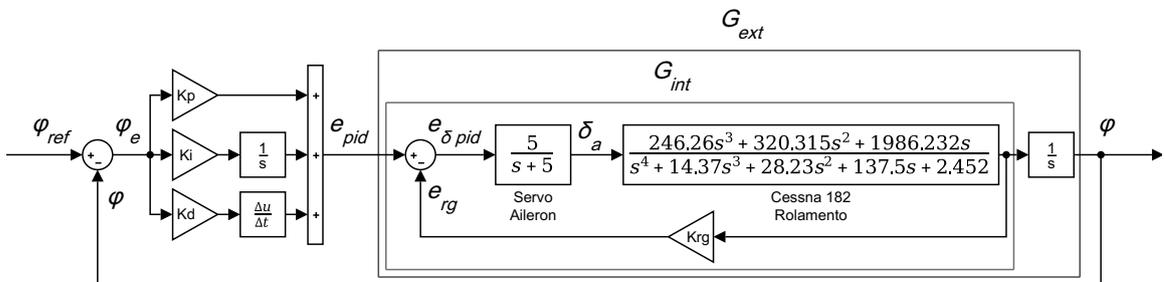
$$G_{ser} = \frac{\delta_a(s)}{e_{\delta pid}} = \frac{5}{s + 5} \quad (2.12)$$

$$G_{\dot{\phi}} = \frac{\dot{\phi}}{\delta_a} = \frac{246.25965s^2 + 320.31486s + 1986.23189}{s^4 + 14.3713s^3 + 28.2324s^2 + 137.5079s + 2.4524} \quad (2.13)$$

$$G_{ext} = \frac{1}{s} G_{int} \quad (2.14)$$

Com os valores de K_{cr} e S_i é possível calcular os valores para os ganhos do controlador PID. Os valores para os ganhos após um ajuste manual fino são $K_p = 2.6430$, $K_i = 1.6498$ e $K_d = 1.1090$ (Splendor et al., 2015). A Figura 2.5 mostra a malha de controle do ângulo de rolamento obtida.

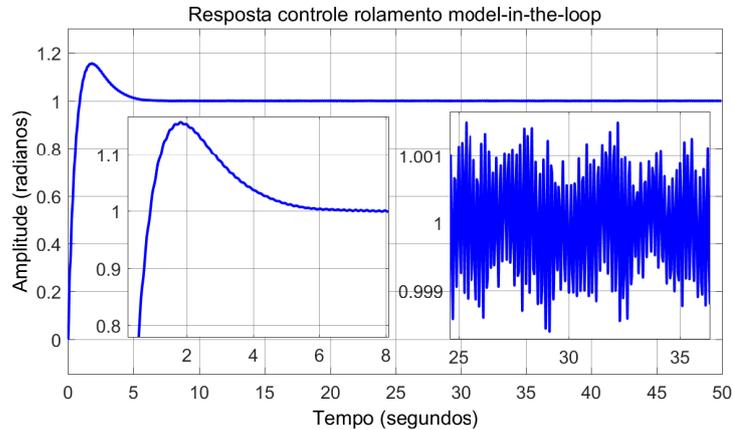
Figura 2.5: Malha de controle do ângulo de rolamento



Fonte: Adaptado de Splendor et al. (2015)

A Figura 2.6 mostra a resposta obtida para a malha de controle do ângulo de rolamento.

Figura 2.6: Resposta da malha de controle do ângulo de rolamento



Fonte: Adaptado de Splendor et al. (2015)

Os valores de resposta a uma entrada do tipo degrau obtidos para a malha de controle apresentam um tempo de pico de 2.80s, sobre-sinal de 15.71% e um tempo de acomodação de 5.55s.

Normalmente em aeronaves convencionais o giroscópio utilizado para medir a taxa de guinada pode ser substituído pela somatória das taxas de rolamento p , de guinada r e de um circuito *washout*. O circuito *washout* é um giroscópio vertical que contribui para o aumento da estabilidade da aeronave quando imposta a perturbações (Roskam, 1995). A somatória das taxas de rolamento p e de guinada r são representados pela equação 2.15, onde p é a derivada do ângulo de rolamento e r é a derivada do ângulo de guinada.

$$G_r = (p \times \sin(\alpha + \alpha_R)) + r \times \cos(\alpha + \alpha_R) \quad (2.15)$$

Com a utilização do ângulo de inclinação $(\alpha + \alpha_R)$ na malha de controle da taxa de guinada é possível aumentar o desempenho dinâmico do seu coeficiente de amortecimento (Splendor et al., 2015). Com a malha de controle completa é possível determinar os valores dos ganhos K_{rg} , K_p , K_i e K_d relacionados à malha de controle utilizando o método do lugar das raízes. Primeiramente, determina-se o ganho K_{rg} da malha de controle utilizando as funções de transferência (equações 2.16 a 2.19) considerando o ângulo de inclinação $(\alpha + \alpha_R) = 0$. Utilizando essas mesmas equações e considerando o ângulo de inclinação $(\alpha + \alpha_R) = 1$ são obtidos os valores para K_{cr} e S_i .

$$\frac{\delta_r(s)}{e_{\delta pid}} = \frac{5}{s + 5} \quad (2.16)$$

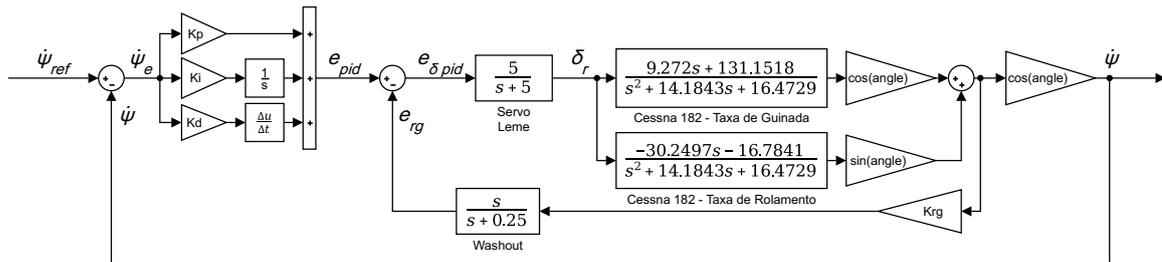
$$H_{was} = \frac{5}{s + 0.25} \quad (2.17)$$

$$\frac{r(s)}{\delta_r(s)} = \frac{9.2717s + 131.1518}{s^2 + 14.1843s + 16.4729} \quad (2.18)$$

$$\frac{r(s)}{\delta_r(s)} = \frac{-30.2497s - 16.7841}{s^2 + 14.1843s + 16.4729} \quad (2.19)$$

Com os valores obtidos para $K_{rg} = 0.1$, $K_{cr} = 1.13$ e $S_i = 3.85$ é possível calcular, usando o segundo método de ZN, os valores para os ganhos do controlador PID para o controlador da taxa de guinada. Os valores obtidos após um ajuste manual fino são $K_p = 0.2143$, $K_i = 0.5698$ e $K_d = 0.0089$ (Splendor et al., 2015). A Figura 2.7 mostra a malha de controle da taxa de guinada obtida.

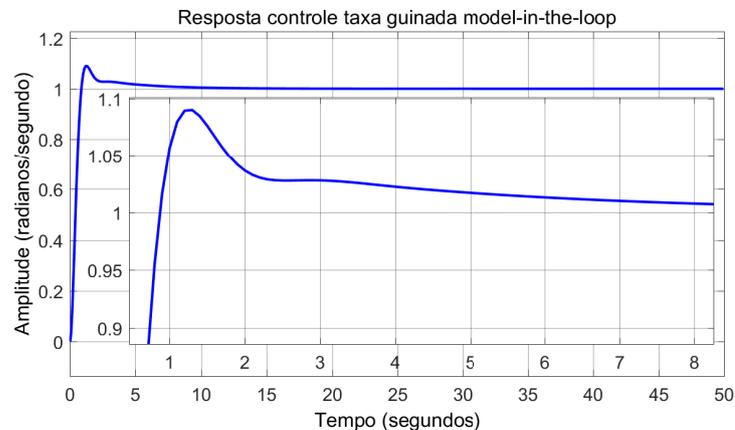
Figura 2.7: Malha de controle da taxa de guinada



Fonte: Adaptado de Splendor et al. (2015)

A Figura 2.8 ilustra a resposta obtida para a malha de controle da taxa de guinada utilizando um ângulo de inclinação de 0° .

Figura 2.8: Resposta da malha de controle da taxa de guinada



Fonte: Adaptado de Splendor et al. (2015)

A resposta obtida para uma entrada do tipo degrau para a malha de controle da taxa de guinada apresenta um tempo de pico de 7.86s, sobre-sinal de 6.08% e um tempo de acomodação de 13.17s (Splendor et al., 2015).

2.5 Trabalhos Relacionados

Roskam (1995) apresenta métodos para a síntese e análise do estado estacionário e perturbado (malhas aberta e fechada) da estabilidade e controle de aeronaves de asa fixa, formulando as equações gerais do movimento para um avião rígido, as quais são especializadas em determinados conjuntos que se aplicam às condições de voo em estado perturbado e estado estacionário, respectivamente. Também fornece uma visão geral dos fundamentos aerodinâmicos necessários para compreender e utilizar força e momento de modelos aerodinâmicos. Várias propriedades importantes de aerofólios e superfícies de elevação são vistos. O efeito da fuselagem no centro aerodinâmico é discutido e alguns aspectos fundamentais de características das superfícies de controle e de *flaps* são cobertas. Também trata da modelagem real de forças de momentos aerodinâmicos e axiais. Faz uma introdução sobre conceito e utilização de derivadas de estabilidade e controle e são dadas explicações físicas e exemplos de sinais e magnitudes desses derivadas.

Shengyi et al. (2009) desenvolveu um piloto automático para o canal longitudinal de veículos aéreos não tripulados (VANTs). Apesar do trabalho afirmar que é baseado em controle PID *fuzzy* adaptativo, no trabalho não está claro o uso da teoria de controle adaptativo. Usa, de fato, um controle avançado que combina o método tradicional de controle PID com o método de controle *fuzzy*. Para o sistema *fuzzy*, são utilizadas funções de pertinência triangulares por serem convenientes de computar. As tabelas com as regras *fuzzy* são apresentadas, mas não são apresentados todos os dados necessários para a replicação desse trabalho. É adotado o método de Mamdani que forma as regras *fuzzy* pelo método máximo-mínimo. Os autores afirmam que os resultados das simulações mostram que o algoritmo do controlador PID *fuzzy* adaptativo é fácil, mas não deixam claro tal afirmação no trabalho.

No trabalho de Ribeiro e Oliveira (2010) é proposta uma plataforma de teste para controladores de piloto automático empregando Matlab/Simulink, o simulador de voo X-Plane com o modelo de aeronave a ser comandada e um microcontrolador para comandar os controles da aeronave. Esses recursos são interconectados usando barramentos de comunicação. Assim o controle de piloto automático desenvolvido no Matlab/Simulink é testado controlando uma aeronave no X-Plane. As entradas das superfícies de controle de voo da aeronave no X-Plane são simultaneamente enviadas ao microcontrolador que

as traduzem em comandos efetivos no controle de movimento do servo. Por meio dessa plataforma, sistemas de piloto automático podem ser aplicados em modelos similares a aeronaves reais, minimizando riscos e aumentando a flexibilidade de mudanças. Como estudo de caso, é apresentado resultados de testes com um sistema de piloto automático para controle do ângulo de rolamento.

O trabalho de Santos et al. (2011) consiste no desenvolvimento de uma plataforma dedicada para a implementação de sistemas de controle longitudinal para o Boeing 747-400, utilizando diferentes topologias para a malha de controle. Inicialmente são estudados os conceitos de movimentação longitudinal para um corpo rígido, determinando o modelo aerodinâmico longitudinal completo e também os modelos aproximados para períodos curto e longo de uma aeronave genérica. Usando os coeficientes de estabilidade e as derivadas longitudinais do Boeing 747-400, foram determinadas as equações dinâmicas para esse avião. Com a determinação das equações dinâmicas para o Boeing 747-400 os controladores foram projetados usando duas topologias distintas de malha de controle. A malha de controle longitudinal composta por uma realimentação de taxa de arfagem é a topologia mais abordada nesse trabalho. Utilizando tal abordagem foram projetados os controladores contínuos de altitude e velocidade vertical. Em uma outra forma de abordagem, no projeto desses controladores se faz a substituição da malha interna de taxa de arfagem por um sistema de compensação contínua, o qual é usado para estabilizar a resposta de saída da malha interna de controle do ângulo de arfagem. Os controladores contínuos projetados foram devidamente discretizados. As equações à diferenças obtidas no processo de discretização foram utilizadas para a implementação dos controladores no microprocessador *Rabbit 2000*. Os controladores digitais implementados no *Rabbit 2000* foram experimentados, verificados e validados usando o Boeing 747-400 disponível no simulador de voo X-Plane. A interação entre o hardware dedicado e o computador executando o X-Plane é feita por meio do envio e recebimento de pacotes de dados do módulo de comunicação *Ethernet (TCP/IP)* utilizando o protocolo *UDP (Uniform Datagram Protocol)* disponíveis em ambos os sistemas. Os resultados obtidos nesses ensaios são comparados com os resultados das simulações do sistema de controle longitudinal usando os controladores projetados e as equações dinâmicas do Boeing 747-400 implementadas no Matlab/Simulink.

Em Santos e Oliveira (2011) é descrito um ambiente de simulação *software-in-the-loop* para o teste e validação de um algoritmo de orientação e controle hierárquico para um VANT de asa fixa, utilizando o simulador de voo X-Plane e Matlab/Simulink. Este *framework* experimental permite o desenvolvimento, a avaliação e a validação de algoritmos para orientação de voo e controle usando modelos realísticos de aeronaves

não tripuladas em ambientes modelados do mundo real, por meio do simulador X-Plane. Apresenta um *framework* de algoritmo para implementação geral do planejamento de caminho e da arquitetura de controle. O algoritmo de orientação de voo proposto utiliza a informação do ângulo de apontamento do vetor magnético e do vetor de velocidade, obtidos por meio do sistema de navegação da aeronave não tripulada virtual, disponível no simulador X-Plane. Por meio dessa plataforma, sistemas de orientação de voo e de piloto automático projetados podem ser aplicados em modelos semelhantes a uma aeronave real, minimizando riscos e aumentando a flexibilidade para mudanças de projeto. Como estudo de caso, se determinou uma orientação de voo consistindo em geração de trajetória, orientação de direção e um controlador de piloto automático para o VANT e os resultados obtidos a partir do sistema de orientação e controle foram apresentados.

Kada e Ghazzawi (2011) descreve os aspectos da estrutura e do projeto de um controlador PID robusto para sistemas de alta ordem. Esse trabalho apresenta um projeto que combina resposta *deadbeat*, controle robusto, e técnicas de redução de modelo para melhorar desempenho e robustez do controlador PID. Ao contrário de controladores *deadbeat* convencionais, os parâmetros de ajuste são reduzidos a um ganho em cascata que produz um prático método de ajuste. O controlador é testado em um piloto automático sobre o eixo de arfagem de um VANT. As simulações computacionais mostram que o método proposto melhora os desempenhos de resposta no domínio do tempo e exibe propriedade mais forte de robustez para compensar incertezas do sistema.

Em Thums et al. (2012) é apresentado uma metodologia para a sintonia de controladores PID para processos com múltiplas malhas considerando as incertezas paramétricas do modelo obtido. O foco do trabalho é na sintonia de controladores para VANTs, nos quais os controladores PID em cascata são amplamente utilizados. Na metodologia desenvolvida, os melhores ganhos para os controladores PID são obtidos por meio de um algoritmo genético, considerando o modelo não-linear do sistema.

Fiuzy et al. (2013) fizeram uma proposta para determinar os coeficientes ótimos do controlador PID em piloto automático baseado em algoritmo de otimização, como algoritmos evolucionário, com o objetivo de regular a altura desejada de um VANT. A função de custo proposta simultaneamente otimiza as especificações de desempenho do sistema. A otimização é feita por algoritmos evolucionários como algoritmo genético, otimização de enxame de partículas e algoritmo de otimização por política social. A resposta do sistema baseada em algoritmos evolucionários é comparada com as vantagens do método de otimização clássico baseado na definição de restrições.

Espinoza et al. (2013) fizeram uma comparação de controladores, os quais são aplicados para um VANT de asa fixa. A comparação é realizada entre leis controles lineares

clássicas e não-lineares. Os controles lineares são o Proporcional-Derivativo (PD) e Proporcional-Integral-Derivativo (PID) e os não-lineares são *backstepping*, modo deslizante, *nested saturation* e *fuzzy*. Esses controles são comparados e analisados para altitude, guinada e rolamento em testes simulados. Os ganhos dos controles PD e PID foram definidos inicialmente pelo segundo método de ZN e depois foram feitos ajustes manuais finos para obter a resposta desejada. Para o controle *fuzzy* de altitude são usadas como entradas o erro e a taxa do erro. São considerados três casos de entrada: negativa (N), zero (Z) ou positiva (P). Para obter a saída é usada a inferência *max-min*. São considerados 5 casos de saída: negativa grande (NG), negativa pequena (NP), zero (Z), positiva pequena (PP) ou positiva grande (PG). Para os controles *fuzzy* de guinada e rolamento são considerados 7 casos de entrada e saída: negativa grande (NG), negativa média (NM), negativa pequena (NP), zero (Z), positiva pequena (PP), positiva média (PM) ou positiva grande (PG). No trabalho não fica claro como o controle *fuzzy* é estruturado, como é definido o número de casos nas entradas e saídas ou como cada caso é definido. Os autores concluem que o controlador *fuzzy* mostra bom desempenho já que combina boa resposta com baixo esforço de controle para cada uma das três dinâmicas, bem como comentam que a escolha por um controlador que estabilize o voo de uma aeronave de asa fixa não é trivial pois ela é uma muito instável para certos movimentos das superfícies de controle.

Mystkowski (2013) apresenta um procedimento de projeto de controle robusto não-linear para um micro VANT, que utiliza o chamado valor singular (μ) e a técnica μ -síntese. O controle robusto definido, combina incertezas paramétricas e aglomeradas do micro VANT que são aplicadas por conexão serial do piloto automático *Kestrel* e do microprocessador *Gumstix*. Assim, os laços de *feedback* do controle robusto, que lidam com as incertezas derivadas da aerodinâmica, são usados para garantir a estabilidade robusta das dinâmicas locais do VANT, em direções de controle longitudinal e lateral.

Em Farid (2013) é apresentado um sistema de inferência *neuro-fuzzy* adaptativo (ANFIS - *adaptive neuro-fuzzy inference system*) empregado para controlar um VANT. A definição da estrutura do piloto automático é apresentada e aplicada no controle de posição lateral do VANT. Os resultados dos controles ANFIS e PID laterais são comparados, onde é demonstrado que os dois controladores possuem resultados similares. O controlador ANFIS é capaz de se adaptar em condições não-lineares, enquanto o PID precisa ser reajustado para preservar controle apropriado em algumas condições. Os resultados foram gerados pelas funções da *toolbox Aerosim Aeronautical Simulation Block Set*, que provê um conjunto completo de ferramentas de desenvolvimento com seis graus de liberdade. O modelo não-linear do VANT *Aerosonde* com o controle ANFIS é simulado para verificar

a capacidade do sistema. Posteriormente, os resultados são validados com o simulador de voo *FlightGear*.

Sarhan e Ashry (2013) desenvolvem um controlador PID adaptativo baseado em otimização de parâmetros com inferência *fuzzy* como piloto automático do VANT de asa fixa *Aerosonde*. A inferência *fuzzy* em tempo real é utilizada como mecanismo de auto-adaptação para ajuste dos ganhos PID. O controlador PID adaptativo proposto é comparado com dois outros controladores. O primeiro é o PID geneticamente ajustado e o outro é o controlador por lógica *fuzzy*. Para o controle *fuzzy* são usadas como entradas o erro, acumulação (integral) do erro e a taxa (derivada) do erro e são considerados 7 casos para cada entrada com funções de pertinência da forma triangular. Não é apresentada uma tabela com as regras *fuzzy* usadas e nem detalhes da implementação do controle PID adaptativo. Os resultados de simulação mostraram boas características de desempenho e robustez do controlador proposto.

No trabalho de Yadav e Gaur (2014) o objetivo é projetar um sistema de piloto automático para VANT com o intuito de controlar a velocidade e a altitude usando sistema de controle eletrônico de acelerador (*ETCS - Electronic Throttle Control System*) e profundor, respectivamente. Um servomotor *DC* é usado para o projeto do *ETCS* para controlar a posição do acelerador para o fluxo de massa de ar apropriado. Controladores baseados em Inteligência Artificial (IA) como PD com lógica *fuzzy*, PD mais I com lógica *fuzzy*, controlador PID adaptativo com lógica *fuzzy* e controlador com lógica *fuzzy* baseada em modo deslizante adaptativo são projetados para sistema de piloto automático estável e comparados com o controlador PI convencional. O objetivo dos controles de velocidade, aceleração e altitude é atingir uma grande faixa de variação da velocidade do ar e melhorar a eficiência energética. Nesse trabalho também é apresentada a eficiência energética utilizando a taxa específica de energia por velocidade do VANT.

Em Aian et al. (2014) é abordado o desenvolvimento de código relacionado com a análise de estabilidade longitudinal em um modelo de aeronave leve com hélice. O código é concebido de tal forma que, ao considerar os dados da aeronave se produz resultados em termos de: derivadas de estabilidade longitudinais em forma não dimensional, bem como na sua forma dimensional; características de estabilidade longitudinais da aeronave relacionada com a posição do comando em ponto neutro (alavanca livre ou fixa) e também forças solicitadas para mover a superfície de controle; função de transferência em resposta à deflexão da superfície de controle para a componente de velocidade e ângulo de arfagem. Por meio desse código desenvolvido pode-se fazer uma avaliação do comportamento longitudinal para aeronave leve com hélice como o Cessna 182.

Em Melkou et al. (2014) é tratado o projeto de dois controladores (longitudinal e latero-direcional) para estabilização de um VANT de asa fixa tipo Cessna 182. Os controles são baseados em modos deslizantes terminais e controle em cascata PID-TSMC para uma lei de controle robusto. É feito um estudo comparativo afim de determinar qual controlador oferece melhor desempenho. Foi demonstrado a robustez do TSMC frente a incertezas paramétricas e perturbações externas. Esse comando é melhorado em termos de precisão adicionando um PID no laço externo.

Famularo et al. (2015) desenvolveu uma arquitetura de controle de supervisão híbrida em um ambiente de tempo real para sistemas de controle limitados. A estratégia baseia-se na ideia de Comando Governador (CG) que é especializado para considerar *set-points* de tempo variável e restrições. A importância do método reside essencialmente na sua capacidade de evitar violação de restrições e a perda de estabilidade, independente de qualquer alteração na configuração da estrutura da planta ou restrição, substituindo o CG corrente por uma nova unidade calculado *on-line*. Um esquema em tempo real é uma escolha extremamente atraente por causa de suas inúmeras aplicações em engenharia: indústria automobilística, defesa e aeroespacial, química e aplicações de usinas nucleares, multimídia e telecomunicações, e assim por diante. Os resultados experimentais em um laboratório com tanques e simulações sobre um modelo de aeronave Cessna 182 mostram a eficácia da estratégia proposta.

No trabalho de Murillo et al. (2015) é utilizado o Modelo de Controle Preditivo (MPC). É uma técnica moderna que, hoje em dia, encapsula diferentes técnicas de controle ótimo. Para o caso das dinâmicas não-lineares, muitas variantes possíveis podem ser desenvolvidas, que podem conduzir a novos algoritmos de controle. Nesse trabalho um novo método de sistema de controle genérico é apresentado. Tal método pode ser aplicado para controlar, de uma forma otimizada, diferentes sistemas tendo dinâmicas não-lineares. Particularmente, nesse trabalho, a técnica proposta é apresentada no contexto do desenvolvimento de um sistema de controle de voo autônomo de VANTs. Essa técnica pode ser aplicada em diferentes tipos de veículos aéreos, que tenham qualquer tipo de dinâmica genérica não-linear. O método apresentado usa um processo de linearização generalizado que é executado ao longo de trajetórias de espaço de estados candidatas definidas de forma iterativa no espaço de estados global. Ao contrário dos métodos clássicos, esse não necessita de ser predefinido num certo estado de equilíbrio, mas em vez disso, é executado ao longo de qualquer estado arbitrário. A técnica permite a inclusão de restrições com facilidade. É utilizada como uma unidade de sistema de controle centralizado, que é capaz de controlar a dinâmica completa de aeronaves sem a necessidade de se dissociar o sistema em diferentes modos reduzidos. A técnica é experimentada em

um modelo simulado do avião Cessna 172 (que é similar ao 182). O teste consistiu em executar manobras autônomas como subir a uma altitude desejada em velocidade constante, mudança de direção de voo e executar curva coordenada.

No trabalho de Efe (2015) são abordadas aplicações da teoria de controle por modos deslizantes em VANTs. São considerados os veículos de pequena escala e seu piloto automático como um controlador por modos deslizantes. São discutidas as necessidades em aplicações práticas e um exemplo simulado é apresentado para dar suporte às hipóteses. Vê-se que a propriedade de robustez do controle por modos deslizantes é uma característica importante em aplicações sujeitas às incertezas e imprecisões. Aplicações em VANTs, portanto, se beneficiam da versatilidade dessa técnica.

O trabalho de Zhiteckii et al. (2015) trata do avançado conceito de otimização- L_1 na teoria de controle moderno com a aplicação no projeto de um sistema de piloto automático lateral L_1 -ótimo. O objetivo do controlador é manter a orientação desejada de rolamento de um VANT na presença de perturbações, especialmente rajadas de vento. Para atingir um índice de alto desempenho, são propostos dois algoritmos separados. O primeiro é projetado para implementar o controle PI L_1 -ótimo para a estabilização de uma dada velocidade de rolamento. O segundo algoritmo garante o controle P L_1 -ótimo necessário para estabilizar o rolamento da aeronave. Os resultados das simulações mostram as propriedades do método de controle proposto.

Splendor et al. (2015) apresenta um projeto de um piloto automático para o Cessna 182. A implementação dos controladores associados ao piloto automático é feita em um microcontrolador com o intuito de realizar sua simulação. A comunicação entre os dispositivos da aeronave é realizada por meio de uma rede de comunicação de dados. Assim, as informações processadas pelos controladores são enviadas diretamente para as superfícies de controle da aeronave representada pelo simulador de voo. Como estudo de caso, o piloto automático projetado foi simulado em diversas condições de voo e, por fim, com a apresentação dos resultados obtidos por meio das simulações, comprova-se que o objetivo de manter a aeronave estabilizada foi alcançado.

No trabalho Beygi et al. (2015) é desenvolvido controlador PID *fuzzy* auto-sintonizável (FSPID - *fuzzy self-tuning* PID) para amenizar as desvantagens do PID convencional (CPID) em pilotos automáticos de aeronaves. O controle proposto é aplicado para o ângulo de arfagem da aeronave e as habilidades do controlador proposto são comparadas com os controladores PID convencional e proporcional. O retorno da malha interna que atua como amortecimento de oscilações em esquemas tradicionais é removido para a comparação das capacidades do PID *fuzzy* auto-sintonizável, PID convencional, e controle proporcional. Baseado em simulações, tanto o PID convencional quanto o

fuzzy auto-sintonizável conseguem adequadamente amortecer as oscilações na falta de um retorno interno da malha, mas o controlador proporcional não consegue. Considera-se então que a aproximação de período curto avalia a função dos controladores FSPID e CPID em confronto com distúrbios abruptos e contínuos, além do ajuste inadequado dos parâmetros. Para o sistema de inferência *fuzzy* são usadas como entradas o erro e a taxa (derivada) do erro. Tanto para as entradas como para saídas com as variações dos ganhos é considerado o intervalo de -5 a 5, e são considerados 7 casos: negativa grande (NG), negativa média (NM), negativa pequena (NP), zero (Z), positiva pequena (PP), positiva média (PM) ou positiva grande (PG). São usadas funções gaussianas e triangulares para as entradas e saídas, respectivamente, e são apresentadas as tabelas com as regras *fuzzy* usadas. As regras *fuzzy* são definidas pelo método de tentativa-e-erro combinada com conhecimento de especialista. É utilizado o método de inferência de Mamdani e operadores de mínimo e máximo são usados para definir as saídas. O método de centróide (centro de gravidade) foi usado para defuzzificação. Os resultados de simulação de aproximação de período curto mostram uma melhor função anti-perturbação para o PID *fuzzy* auto-sintonizável comparado ao PID convencional. O PID *fuzzy* auto-sintonizável pode ajustar os parâmetros PID para obter a resposta ótima em função da velocidade, sobre-sinal e erro de estado estacionário em condições de ajuste inadequado de parâmetros PID, com base nos resultados da simulação em aproximação de curto período, O controlador pode melhorar adaptativamente a resposta do sistema por meio da definição *on-line* de parâmetros PID. Considerando que a inferência *fuzzy* foi desenvolvida por tentativa-e-erro com conhecimento de especialista, os autores sugerem que o método proposto pode ser melhorado com o emprego de métodos inteligentes como algoritmos genéticos e redes neurais.

Sarhan e Qin (2016) desenvolve um controlador PID adaptativo baseado em otimização de parâmetros com inferência *fuzzy* para controlar a dinâmica de altitude do VANT *Aerobsonde*. A inferência *fuzzy* em tempo real é utilizada como mecanismo de auto-adaptação para ajuste dos ganhos PID. O controlador PID adaptativo proposto é comparado com dois outros controladores. O primeiro é o PID geneticamente ajustado e o outro é o controlador por lógica *fuzzy*. Para o controle *fuzzy* são usadas como entradas o erro e a taxa (derivada) do erro no intervalo de -10 a 10. São considerados 5 casos de entrada: negativa grande (NG), negativa pequena (NP), zero (Z), positiva pequena (PP) ou positiva grande (PG). As funções de pertinência são da forma triangular e é apresentada uma tabela com as regras *fuzzy* usadas. Para o controle PID adaptativo não foi apresentado como são definidos os ganhos iniciais. Novamente são usadas como entradas o erro e a taxa (derivada) do erro mas no intervalo de -3 a 3. São considerados 7 casos: negativa

grande (NG), negativa média (NM), negativa pequena (NP), zero (Z), positiva pequena (PP), positiva média (PM) ou positiva grande (PG). Para as saídas das variações dos ganhos são considerados 3 casos: negativa (N), zero (Z) ou positiva (P). Os intervalos para as saídas são de -3 a 3 para a variação de K_p e de -1 a 1 para a variação de K_i e K_d e são apresentadas as tabelas com as regras *fuzzy* usadas. As regras *fuzzy* são definidas com base em experiência humana e são um critério para a tomada de decisão sobre o sistema. Os resultados de simulação mostraram boas características de desempenho e boa robustez quanto a estabilidade do controlador PID adaptativo proposto.

2.6 Considerações Finais

Neste capítulo foram apresentados conceitos básicos sobre controle automático. Também foi feita uma breve descrição dos principais componentes de uma aeronave, as forças e movimentos que atuam em um avião. Por fim, foi feita uma revisão bibliográfica sobre trabalhos relacionados com piloto automático de aeronaves usando controles avançados.

Otimização de Ganhos de Sintonia de Controladores PID

Neste capítulo são apresentados procedimentos de otimização dos ganhos dos controladores PID, a partir das funções de transferência simuladas no software Matlab/Simulink, estabelecendo um sistema *model-in-the-loop* (MIL) para o piloto automático do avião Cessna 182 e, posteriormente, com a integração do software Matlab/Simulink e o simulador de voo X-Plane, constituindo num sistema *software-in-the-loop* (SIL), para o piloto automático do avião Cessna 172. Também são implementados os controladores usando uma placa Arduino que foi testada em conjunto com o simulador de voo, formando um sistema *hardware-in-the-loop* (HIL). Como vantagens, a otimização pode minimizar consideravelmente a exigência de sintonia em voo e reduzir substancialmente os riscos e custos envolvidos nos testes em voo. Neste contexto, resultados de simulações são obtidos para comprovar a eficácia do uso dessas propostas de otimização.

3.1 Otimização dos Ganhos: Procedimentos e Índices de Desempenho

O objetivo da proposta de otimização é minimizar o erro entre o valor de referência e o valor atual obtido por cada controlador. A otimização dos ganhos dos controladores é feita utilizando as ferramentas *Simulink Design Optimization* do software Matlab/Simulink (MathWorks, 2016). Para isso, os blocos *Check Step Response Characteristics* e *Check Against Reference*, pertencentes ao *Simulink Design Optimization*, são inseridos na saída

da malha de controle (Figura 3.1). O bloco *Check Step Response Characteristics* fornece os critérios de projeto, tais como tempo de subida, percentual de subida, tempo de acomodação, percentual de acomodação, percentual de sobre-sinal, percentual de sub-sinal, necessários à otimização dos ganhos de cada controlador, baseado nas restrições das especificações de projeto definidas por Splendor et al. (2015) e mostradas nas Tabelas 2.1 e 2.2. Deve ser salientado que o *Simulink Design Optimization* é uma boa alternativa para otimizar variáveis dependentes do tempo e funções objetivo implícitas (Ahsan et al., 2013).

Os critérios percentual de subida e tempo de subida especificam que a resposta a uma entrada do tipo degrau unitário atinja um percentual maior ou igual ao dado pelo percentual de subida em um tempo menor ou igual ao determinado pelo tempo de subida. No bloco *Check Against Reference* é especificado um sinal de referência, ao qual a resposta fornecida (valor atual obtido) pelo controlador deve se aproximar ou coincidir. Essa resposta é definida para um degrau unitário pela função:

$$f(x) = 1 - e^{(-a.x)} \quad (3.1)$$

a é definido pela equação:

$$0.02 = e^{(-t_s.a)} \quad (3.2)$$

e t_s é obtido das Tabelas 2.1 e 2.2.

Além disso, para realizar a otimização, os mesmos ganhos obtidos na Seção 2.4 são utilizados como ganhos iniciais dos controladores, assim como as mesmas funções de transferência (TF) das malhas de controle longitudinal e latero-direcional descritas na Seção 2.4. Além disso, a otimização tem o intuito de tornar mais satisfatórios os resultados já obtidos em Splendor et al. (2015).

O índice de desempenho baseado no erro médio quadrático (EMQ) representa uma estimativa da precisão do sistema e é definido pela média dos quadrados dos erros (equação 3.3) (Lehmann e Casella, 2006).

$$EMQ = \frac{\sum_{i=1}^n (y(i) - y_{ref})^2}{n} \quad (3.3)$$

O esforço de controle médio (ECM) representa uma estimativa da precisão do sistema e é definida pela média dos quadrados das saídas do controlador (equação 3.4) (Vale, 2007).

$$ECM = \frac{\sum_{i=1}^n (u(i))^2}{n} \quad (3.4)$$

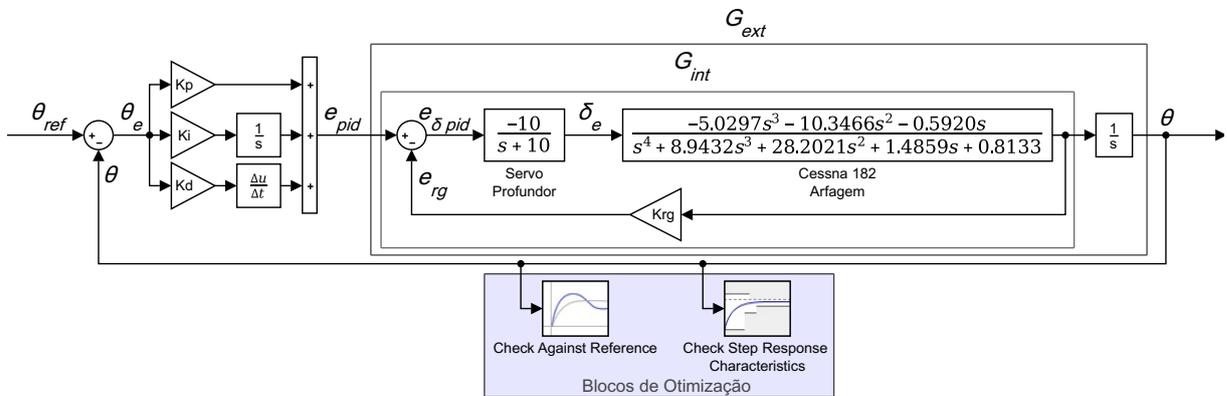
A variância do sinal de controle (VSC) (equação 3.5), permite avaliar o esforço do controle sobre os atuadores. Para uma maior vida útil do elemento atuador, a princípio, este valor deve ser o menor possível (do Carmo et al., 2012).

$$VSC = \frac{\sum_{i=1}^n (u(i) - \bar{u})^2}{n} \quad (3.5)$$

onde

$$\bar{u} = \frac{\sum_{i=1}^n u(i)}{n} \quad (3.6)$$

Figura 3.1: Blocos de otimização em conjunto com malha de controle do ângulo de arfagem



Fonte: Elaborado pelo autor

3.2 Otimização com Plataforma MIL a partir das Funções de Transferência

Nesta seção são utilizadas as mesmas funções de transferência das malhas de controles longitudinal e latero-direcional descritas na Seção 2.4 para a otimização.

3.2.1 Controle Longitudinal

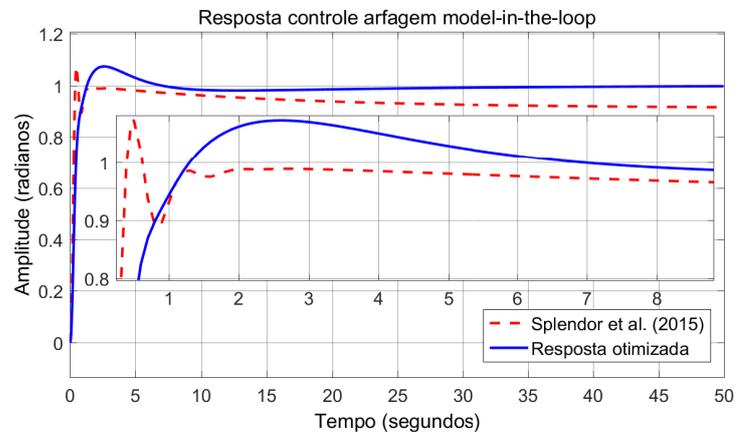
Arfagem

Na malha de controle do ângulo de arfagem, o controlador proporcional (P) foi substituído por um PID para possibilitar a obtenção de uma resposta ainda melhor e eliminar o erro estacionário. Com isso foram adicionados os ganhos K_i e K_d , conforme já mostrado na Figura 3.1 (Thums et al., 2012). Como a Tabela 2.1 não especifica um valor para

o tempo de acomodação, t_s , no controle do ângulo de arfagem foi definido $t_s = 10s$. O tempo de pico t_p foi desconsiderado pois não foi possível obter uma resposta realística (testada no X-Plane) suave atendendo esse critério. Logo, priorizou-se somente minimizar o tempo de acomodação e o máximo sobre-sinal. O valor K_p foi limitado no máximo 18.59 (14.3*120%), K_i sem máximo, K_d e K_{rg} no máximo 1.18 (1.18*100%). O valor zero foi definido para os valores mínimos.

A Figura 3.2 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com os ganhos $K_p = 7.1278$, $K_i = 2.0630$, $K_d = 0.0$ e $K_{rg} = 1.18$ obtidos da otimização, a qual apresenta um tempo de pico de 2.6s, sobre-sinal de 7.36% e tempo de acomodação de 5.5s. Considerando que o valor K_d convergiu para 0, pode-se afirmar que a otimização do controle PID convergiu para um PI-D, que é uma variante do esquema de controle PID, mas com o ganho K_{rg} no lugar de K_d . Nessa variante, a ação derivativa é colocada somente no ramo de realimentação para que a diferenciação ocorra apenas no sinal de realimentação e não no sinal de referência (Ogata, 2010).

Figura 3.2: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com otimização dos ganhos na plataforma MIL.



Fonte: Elaborado pelo autor

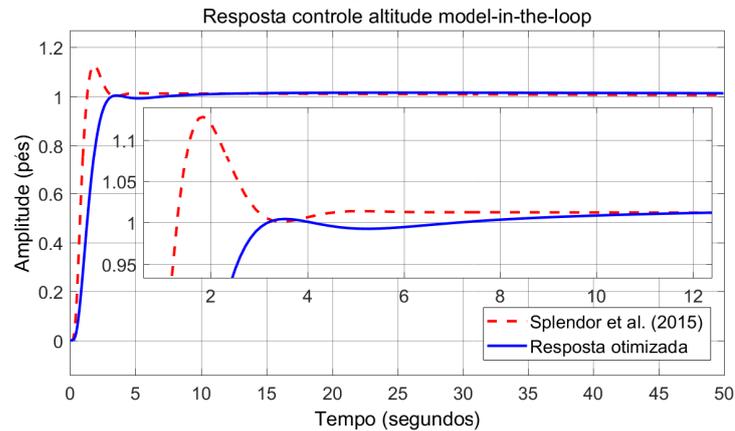
Altitude

O valor K_p foi limitado no máximo 0.0146 (0.0112*130%), K_i sem máximo e K_d no máximo 0.0032 (0.0032*100%). O valor zero foi definido para os valores mínimos.

Na Figura 3.3 é mostrada a resposta a uma entrada do tipo degrau unitário da malha de controle de altitude com os ganhos $K_p = 0.0052$, $K_i = 0.0001$ e $K_d = 0.0032$ após

otimização. A resposta apresenta tempo de pico de 2.25s, sobre-sinal de 6.12%, tempo de acomodação de 3s.

Figura 3.3: Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude com otimização dos ganhos na plataforma MIL.



Fonte: Elaborado pelo autor

3.2.2 Controle Latero-direcional

Rolamento

O valor K_p foi limitado no máximo 3.4359 ($2.6430 * 130\%$), K_i sem máximo, K_d no máximo 1.1090 ($1.1090 * 100\%$) e K_{rg} no máximo 0.0803 ($0.0803 * 100\%$). Os valores mínimos foram definidos em zero.

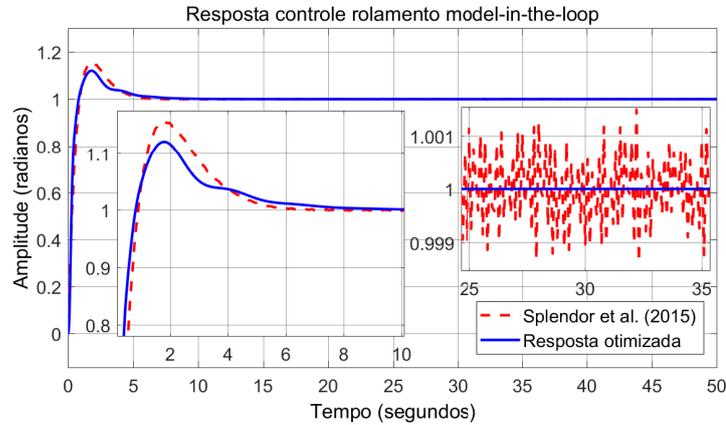
Na Figura 3.4, com a otimização dos ganhos para $K_p = 0.5915$, $K_i = 0.2574$, $K_d = 0.0662$, $K_{rg} = 0.0803$, para a malha de controle do ângulo de rolamento, verifica-se que a resposta apresenta tempo de pico de 1.8s, sobre-sinal de 12.07%, tempo de acomodação de 5.01s.

Guinada

O valor K_p foi limitado no máximo 0.2786 ($0.2143 * 130\%$), K_i sem máximo, K_d no máximo 0.0089 ($0.0089 * 100\%$) e K_{rg} no máximo 0.1 ($0.1 * 100\%$). Os valores mínimos foram definidos em zero.

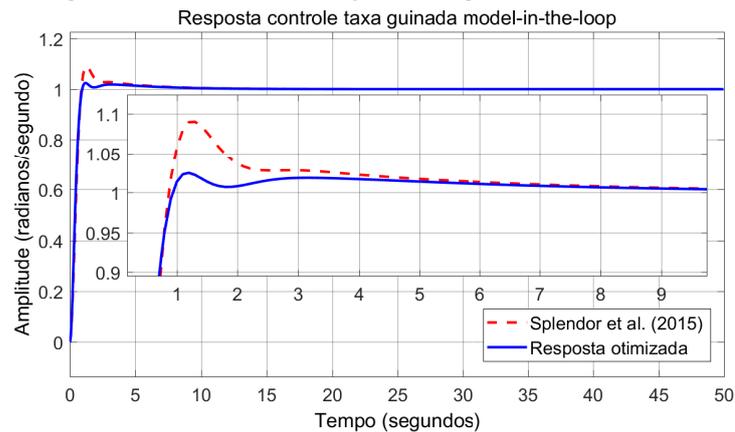
Ao analisar a Figura 3.5, constata-se que a resposta apresenta um tempo de pico de 3.27s, sobre-sinal de 1.76%, tempo de acomodação de 0.91s, levando em conta os ganhos $K_p = 0.2786$, $K_i = 0.4322$ e $K_d = 0.0089$, $K_{rg} = 0.0562$ obtidos da otimização para a malha de controle da taxa de guinada.

Figura 3.4: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento com otimização dos ganhos na plataforma MIL.



Fonte: Elaborado pelo autor

Figura 3.5: Resposta a uma entrada do tipo degrau unitário da malha de controle da taxa de guinada com otimização dos ganhos na plataforma MIL.



Fonte: Elaborado pelo autor

A Tabela 3.1 mostra a comparação de índices de desempenho obtidos antes e depois da otimização na plataforma MIL. A taxa mostra a variação do índice, sendo a proporção entre a otimização e o obtido por Splendor et al. (2015). Quanto ao ângulo de arfagem houve um pequeno aumento no EMQ, mas diminuíram significativamente no ECM e VSC. Isso se deve ao fato que, com os ganhos otimizados, a resposta ficou levemente mais lenta mas muito mais suave. Isso se refletiu na resposta do controle de altitude. Já no ângulo de rolamento além de uma boa diminuição quanto ao EMQ, houve uma diminuição significativa nos valores de erro quanto ao ECM e VSC. Isso evidencia que antes da otimização havia uma oscilação de alta frequência no controle. Também na taxa de guinada é possível ver melhora em todos os índices.

Tabela 3.1: Comparação de índices de desempenho da otimização na plataforma MIL.

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	0.95×10^{-2}	1.46×10^0	9.64×10^{-1}
Otimização	1.15×10^{-2}	0.84×10^0	3.16×10^{-1}
taxa	1.21	0.58	0.33
Altitude			
Splendor et al. (2015)	1.85×10^{-2}	2.45×10^{-5}	2.48×10^{-5}
Otimização	2.55×10^{-2}	2.21×10^{-5}	2.21×10^{-5}
taxa	1.38	0.90	0.89
Rolamento			
Splendor et al. (2015)	9.24×10^{-3}	7703×10^7	7703×10^7
Otimização	5.40×10^{-3}	0.01×10^0	0.01×10^0
taxa	0.58	0.01	0.01
Guinada			
Splendor et al. (2015)	1.17×10^{-2}	2.23×10^{-2}	1.17×10^{-3}
Otimização	1.06×10^{-2}	1.98×10^{-2}	0.79×10^{-3}
taxa	0.91	0.89	0.68

Fonte: Elaborado pelo autor

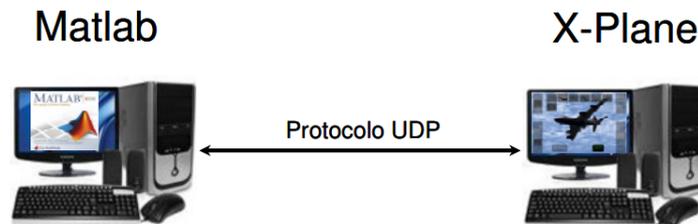
3.3 Otimização com Plataforma SIL

Na Seção 3.2, a otimização dos ganhos é realizada para as malhas de controle longitudinal e latero-direcional desenvolvidas por Splendor et al. (2015) utilizando funções de transferência para o piloto automático do avião Cessna 182. Nesta Seção, os ganhos da otimização são obtidos e testados para o piloto automático do avião Cessna 172 disponível no simulador de voo X-Plane, como mostrado na Figura 3.1, pelo uso dos blocos *Check Step Response Characteristics* e *Check Against Reference*. Os critérios de otimização são definidos a partir das especificações definidas em Splendor et al. (2015).

Para a realização das otimizações e dos testes, utilizando o simulador de voo X-Plane, é desenvolvida uma plataforma SIL. Nela os dados de voo gerados pelo simulador de voo X-Plane com o avião Cessna 172 são passados para os controladores implementados no Matlab/Simulink que faz a leitura dos dados e retorna os comandos de controle para o simulador de voo X-Plane. A comunicação é feita por interface de rede utilizando o protocolo UDP, conforme a Figura 3.6 (Ribeiro e Oliveira, 2010). O bloco *Real-Time*

Synchronization é utilizado nas simulações para fazer com que a execução ocorra em tempo real, assim como ocorre no X-Plane.

Figura 3.6: Plataforma SIL.



Fonte: Elaborado pelo autor

3.3.1 Controle Longitudinal

Arfagem

Com o intuito de otimização dos ganhos da malha de controle do ângulo de arfagem, K_p , K_i , K_d e K_{rg} , Figura 3.7, é definida uma resposta a uma entrada do tipo degrau unitário com o ângulo de arfagem variando de zero para 1 grau. A altitude foi mantida próxima dos 4800 pés, o ângulo de rolamento foi mantido nivelado e o ângulo de guinada não foi controlado por não interferir nessa resposta. A saída precisou ser dividida por 100 para que, com ganhos iniciais, a resposta fosse realizável.

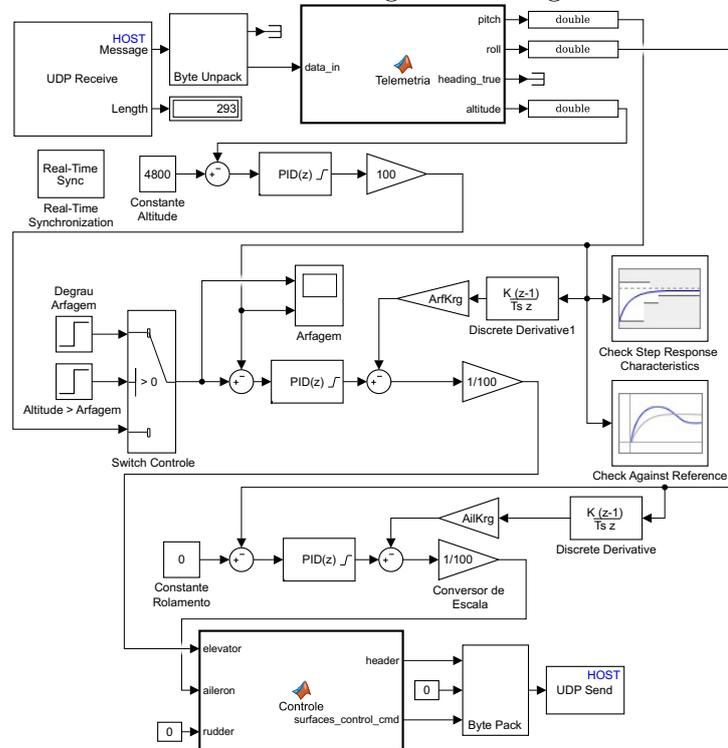
Os ganhos iniciais utilizados foram os obtidos em Splendor et al. (2015) ($K_p = 14.3$, $K_i = 0.0$, $K_d = 0.0$, já que era um controlador proporcional, e $K_{rg} = 1.18$). O valor K_p foi limitado no máximo 18.59 ($14.3 \cdot 130\%$), K_i sem máximo, K_d e K_{rg} no máximo 1.18 ($1.18 \cdot 100\%$). Os valores mínimos foram definidos em zero.

A Figura 3.8 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com os ganhos obtidos por Splendor et al. (2015), a qual apresenta pico máximo de 98.44% aos de 3.0s e não acomoda. Também mostra a resposta com os ganhos $K_p = 7.5933$, $K_i = 3.8066$, $K_d = 0.1058$ e $K_{rg} = 0.5688$ obtidos da otimização, a qual apresenta um tempo de pico de 2.4s, sobre-sinal de 12.54% e tempo de acomodação de 14.4s.

Altitude

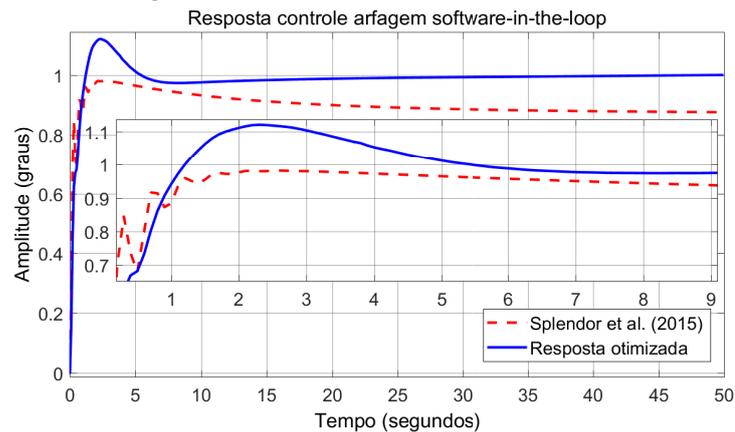
Com relação à otimização dos ganhos da malha de controle de altitude, K_p , K_i , K_d , Figura 3.9, é definida uma resposta a uma entrada do tipo degrau unitário variando a altitude de 4800 pés para 4801 pés. O ângulo de rolamento foi mantido nivelado e o

Figura 3.7: Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de arfagem.



Fonte: Elaborado pelo autor

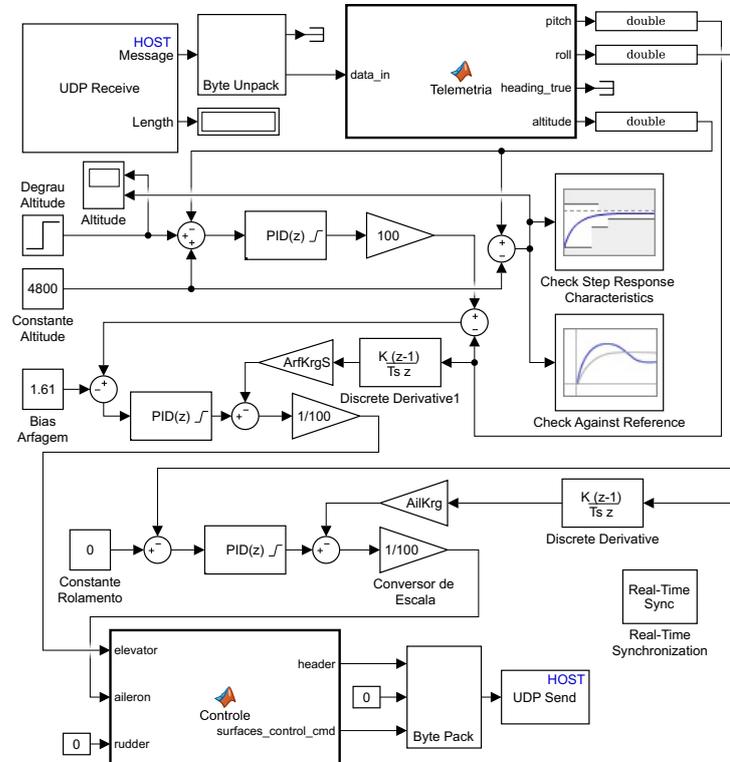
Figura 3.8: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem na plataforma SIL.



Fonte: Elaborado pelo autor

ângulo de guinada não foi controlado por não interferir nessa resposta. A saída precisou ser multiplicada por 100 para que, com ganhos iniciais, a resposta fosse realizável.

Figura 3.9: Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle de altitude.

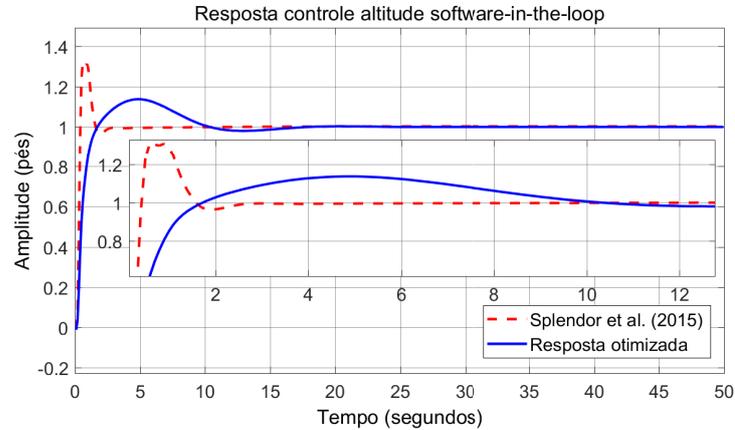


Fonte: Elaborado pelo autor

Os ganhos iniciais utilizados foram os obtidos em Splendor et al. (2015) ($K_p = 0.0112$, $K_i = 0.00038$ e $K_d = 0.0032$). O valor K_p foi limitado no máximo 0.01456 ($0.0112 \cdot 130\%$), K_i sem máximo e K_d no máximo 0.0032 ($0.0032 \cdot 100\%$). Os valores mínimos foram definidos em zero.

A Figura 3.10 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle de altitude com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 1s, sobre-sinal de 15.28% e tempo de acomodação de 1.53s. Também mostra a resposta do controle de altitude com os ganhos $K_p = 0.0023$, $K_i = 0.0011$, $K_d = 0.0030$ obtidos da otimização, a qual apresenta um tempo de pico de 5.6s, sobre-sinal de 11.57% e tempo de acomodação de 10.13s. É possível perceber que houve uma resposta mais suave em detrimento do tempo de acomodação.

Figura 3.10: Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude na plataforma SIL.



Fonte: Elaborado pelo autor

3.3.2 Controle Latero-direcional

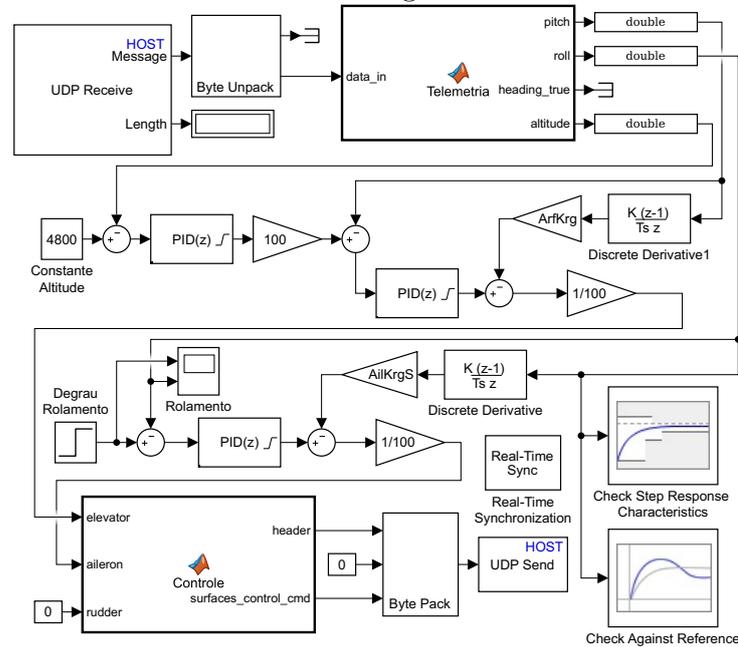
Rolamento

Com o intuito de otimização dos ganhos da malha de controle do ângulo de rolamento, K_p , K_i , K_d e K_{rg} , Figura 3.11, é definida uma resposta a uma entrada do tipo degrau unitário com o ângulo de rolamento variando de zero para 1 grau. A altitude foi mantida próxima dos 4800 pés e o ângulo de guinada não foi controlado para não interferir nessa resposta. A saída precisou ser dividida por 100 para que, com ganhos iniciais, a resposta fosse realizável.

Os ganhos iniciais utilizados foram os obtidos em Splendor et al. (2015) ($K_p = 2.6430$, $K_i = 1.6498$, $K_d = 1.1090$ e $K_{rg} = 0.0803$). O valor K_p foi limitado no máximo 3.4359 ($2.6430 \cdot 130\%$), K_i sem máximo, K_d no máximo 1.1090 ($1.1090 \cdot 100\%$) e K_{rg} no máximo 0.0803 ($0.0803 \cdot 100\%$). Os valores mínimos foram definidos em zero.

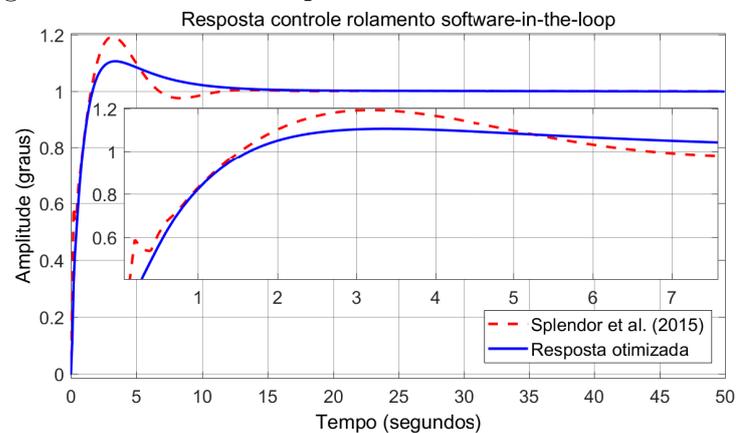
A Figura 3.12 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 3.1s, sobre-sinal de 17.24% e tempo de acomodação de 8.97s. Também mostra a resposta com os ganhos $K_p = 3.31053$, $K_i = 0.7376$, $K_d = 0.3488$ e $K_{rg} = 0.0803$ obtidos da otimização, a qual apresenta um tempo de pico de 3.2s, sobre-sinal de 9.73% e tempo de acomodação de 9.65s, mostrando significativa melhora quanto ao sobre-sinal sem muito prejuízo ao tempo de acomodação.

Figura 3.11: Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de rolamento.



Fonte: Elaborado pelo autor

Figura 3.12: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento na plataforma SIL.



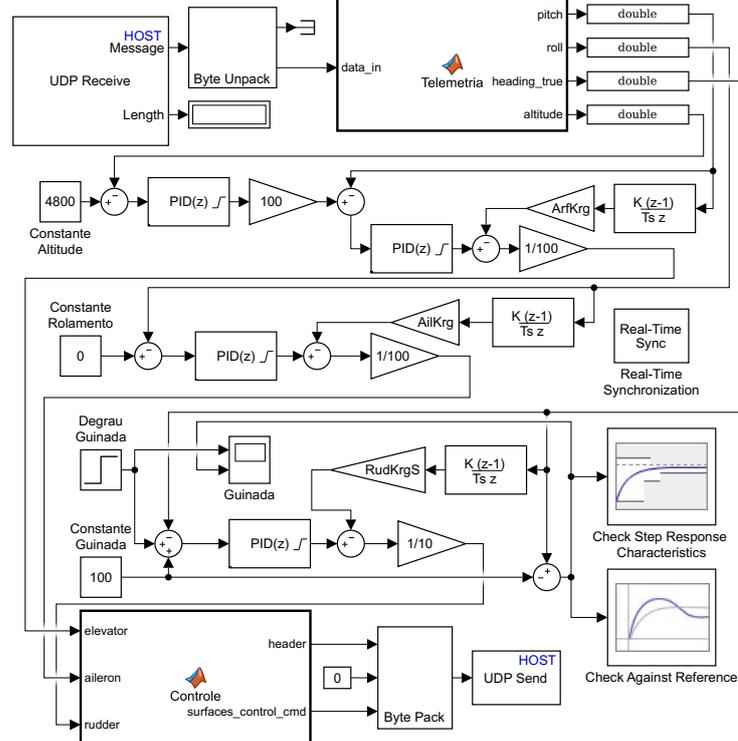
Fonte: Elaborado pelo autor

Guinada

Nas plataformas *SIL* e *HIL* foram feitas otimizações para o controle do ângulo de guinada e não para a taxa. Para a otimização dos ganhos da malha de controle do ângulo de guinada, K_p , K_i , K_d e K_{rg} , Figura 3.13, é definida uma resposta a uma entrada do tipo degrau unitário com o ângulo de guinada variando de 100 para 101 graus. Novamente, a altitude

foi mantida próxima dos 4800 pés e o ângulo de rolamento foi mantido nivelado. A saída precisou ser dividida por 10 para que, com ganhos iniciais, a resposta fosse realizável.

Figura 3.13: Modelo do Simulink usado na plataforma SIL com os blocos de otimização para a malha de controle do ângulo de guinada.



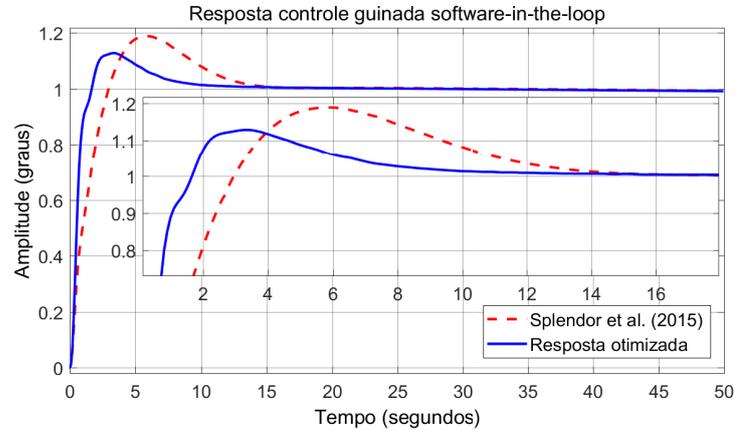
Fonte: Elaborado pelo autor

Os ganhos iniciais utilizados foram os obtidos em Splendor et al. (2015) ($K_p = 0.2143$, $K_i = 0.5698$, $K_d = 0.0089$ e $K_{rg} = 0.1$). O valor K_p foi limitado no máximo 0.2786 ($0.2143 \cdot 130\%$), K_i sem máximo, K_d no máximo 0.0089 ($0.0089 \cdot 100\%$) e K_{rg} no máximo 0.1 ($0.1 \cdot 100\%$). Os valores mínimos foram definidos em zero.

A Figura 3.14 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 6.0s, sobre-sinal de 18.55% e tempo de acomodação de 13.36s. Também mostra a resposta com os ganhos $K_p = 0.2786$, $K_i = 1.2987$, $K_d = 0.0089$ e $K_{rg} = 0.0781$ obtidos da otimização, a qual apresenta um tempo de pico de 3.4s, sobre-sinal de 13.01% e tempo de acomodação de 8.75s, mostrando uma melhora em todos os aspectos.

A Tabela 3.2 mostra a comparação de índices de desempenho obtidos antes e depois da otimização na plataforma SIL. Quanto ao ângulo de arfagem houve uma redução no EMQ, pequeno aumento no ECM mas redução significativa na VSC. No controle de

Figura 3.14: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada na plataforma SIL.



Fonte: Elaborado pelo autor

altitude houve pequeno aumento no EMQ devido ao controle do ângulo de arfagem ser mais suave, mas houve consideráveis reduções quanto ao ECM e VSC. Já no ângulo de rolamento além da redução quanto ao EMC, houve uma redução significativa quanto ao ECM e VSC. Também no ângulo de guinada é possível ver redução em todos os índices.

3.4 Otimização com Plataforma HIL

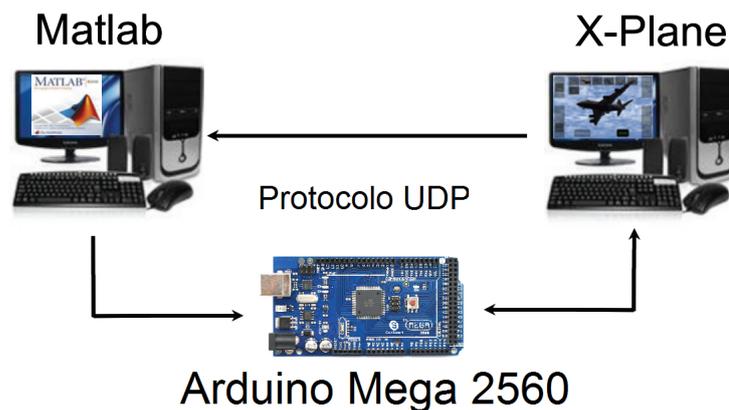
Nesta Seção, os ganhos da otimização são obtidos e testados para o piloto automático do avião Cessna 172 disponível no simulador de voo X-Plane em uma plataforma HIL utilizando Arduino Mega 2560, simulador X-Plane e Matlab/Simulink como plataforma de controle e monitoramento. Os critérios de otimização, os ganhos iniciais e as faixas de determinação dos ganhos são os mesmos utilizados na Seção 3.3.

Na plataforma HIL desenvolvida, os dados de voo gerados pelo simulador de voo X-Plane com o avião Cessna 172 são passados para os controladores implementados no microcontrolador Arduino MEGA 2560 que faz a leitura dos dados e retorna os comandos de controle para o simulador de voo X-Plane. Os dados gerados pelo simulador também são passados para o Matlab/Simulink para geração dos gráficos. Além disso, o Matlab/Simulink controla a simulação passando para o Arduino as referências de arfagem, altitude, rolamento e guinada que devem ser mantidas e os ganhos que devem ser utilizados. Toda a comunicação é feita por interface de rede utilizando o protocolo UDP, conforme a Figura 3.15 (Ribeiro e Oliveira, 2010).

Tabela 3.2: Comparação de índices de desempenho da otimização na plataforma SIL.

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	4.74×10^{-2}	1.00×10^1	6.96×10^0
Otimização	4.17×10^{-2}	1.19×10^1	2.46×10^0
taxa	0.88	1.19	0.35
Altitude			
Splendor et al. (2015)	4.09×10^{-2}	3.94×10^{-4}	7.19×10^{-5}
Otimização	4.23×10^{-2}	2.70×10^{-4}	3.93×10^{-5}
taxa	1.03	0.69	0.55
Rolamento			
Splendor et al. (2015)	4.47×10^{-2}	7.25×10^0	7.24×10^0
Otimização	4.23×10^{-2}	1.90×10^0	1.78×10^0
taxa	0.95	0.26	0.26
Guinada			
Splendor et al. (2015)	5.75×10^{-2}	1.02×10^{-1}	5.04×10^{-2}
Otimização	4.24×10^{-2}	0.98×10^{-1}	0.53×10^{-2}
taxa	0.74	0.96	0.11

Fonte: Elaborado pelo autor

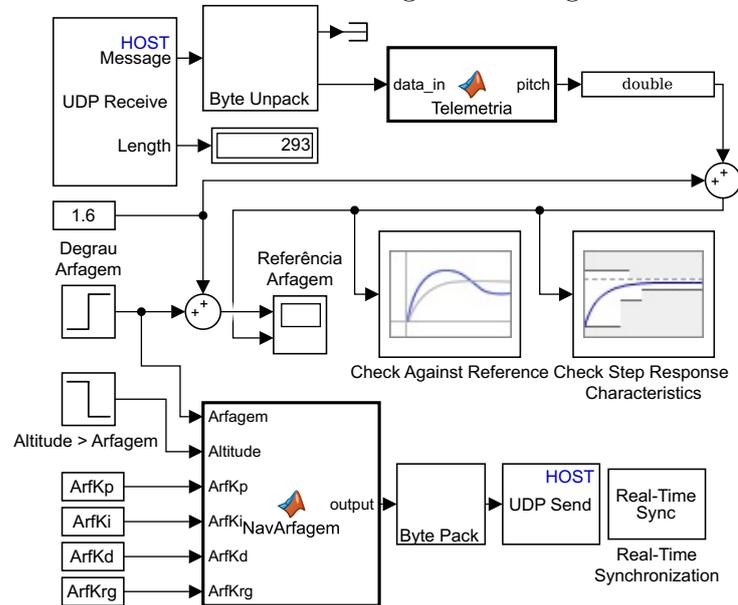
**Figura 3.15:** Plataforma HIL.

3.4.1 Controle Longitudinal

Arfagem

A Figura 3.16 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de arfagem. A cada iteração, novos ganhos são passados para o microcontrolador para serem utilizados.

Figura 3.16: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de arfagem.



Fonte: Elaborado pelo autor

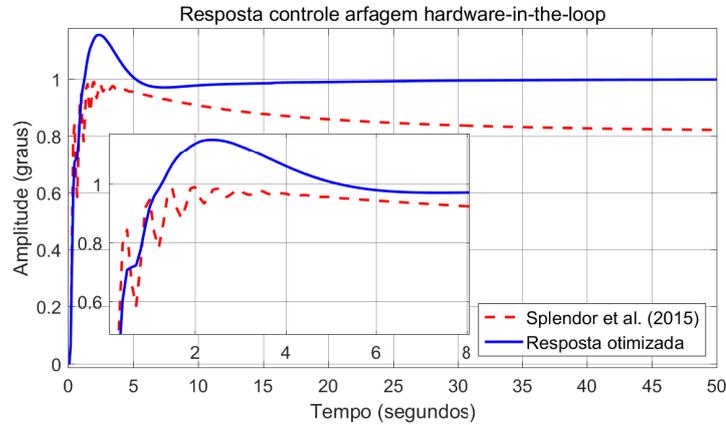
A Figura 3.17 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com os ganhos obtidos por Splendor et al. (2015), a qual apresenta pico máximo de 97.55% aos de 2.9s e não acomoda. Também mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem com os ganhos $K_p = 8.9398$, $K_i = 5.1016$, $K_d = 0.0797$ e $K_{rg} = 0.5284$ obtidos da otimização, a qual apresenta um tempo de pico de 2.2s, sobre-sinal de 13.94% e tempo de acomodação de 8.96s.

Altitude

A Figura 3.18 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de altitude.

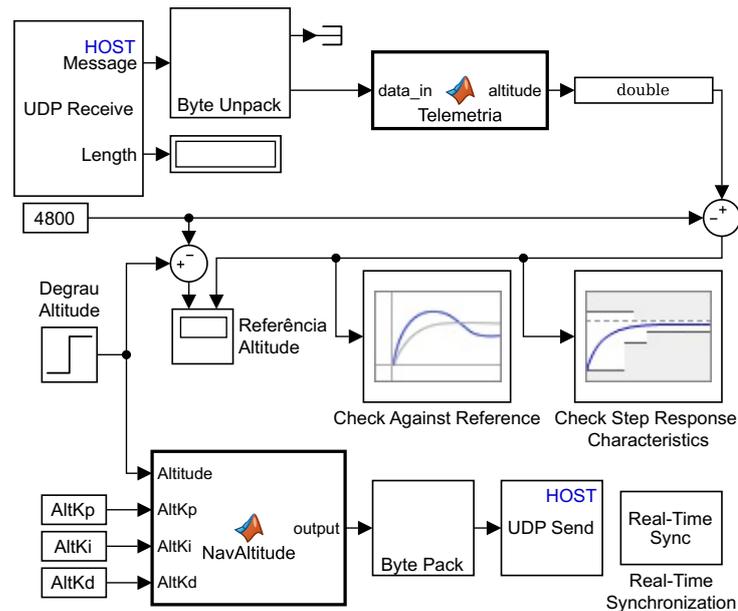
A Figura 3.19 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle de altitude com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 1.3s, sobre-sinal de 24.17% e tempo de acomodação de 2.67s. Também mostra a resposta do controle de altitude com os ganhos $K_p = 0.00381$, $K_i = 0.00161$, $K_d = 0.00300$ obtidos da otimização, a qual apresenta um tempo de pico de 3.1s, sobre-sinal de 9.38% e tempo de acomodação de 8.41s. Com isso se obteve uma redução considerável do sobre-sinal mantendo um tempo de acomodação razoável.

Figura 3.17: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de arfagem na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 3.18: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle de altitude.



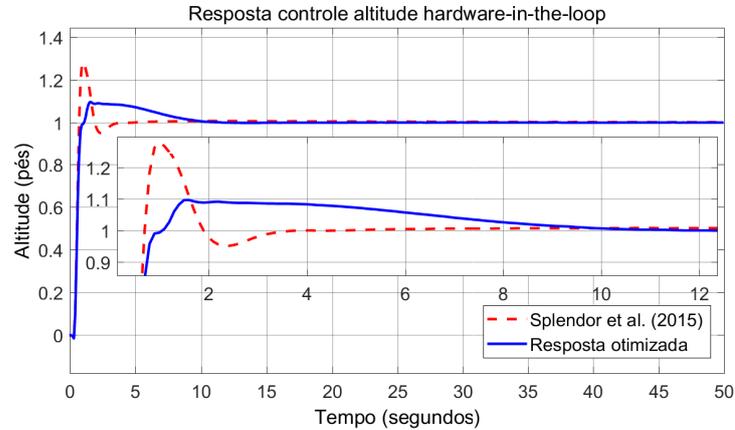
Fonte: Elaborado pelo autor

3.4.2 Controle Latero-direcional

Rolamento

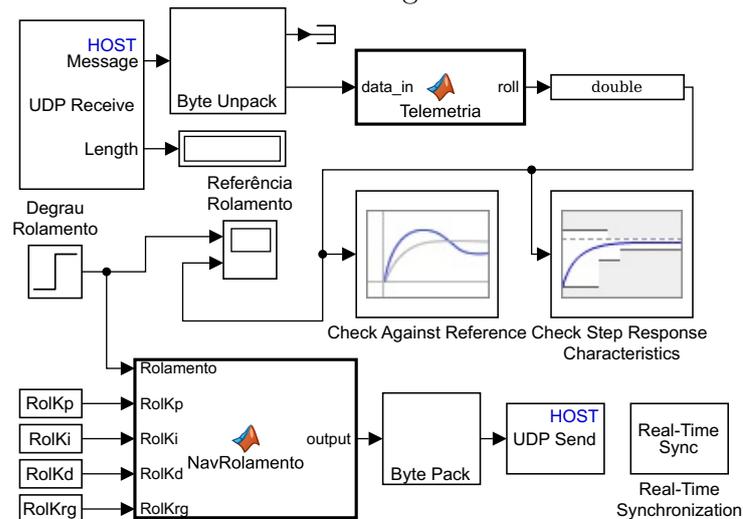
A Figura 3.20 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de rolamento.

Figura 3.19: Resposta a uma entrada do tipo degrau unitário da malha de controle de altitude na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 3.20: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de rolamento.

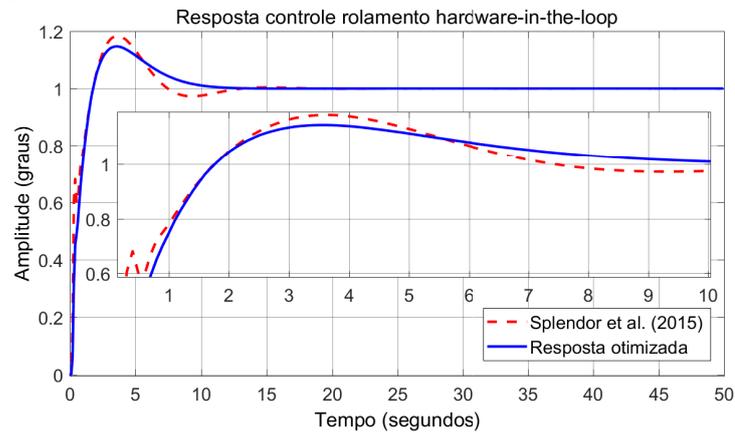


Fonte: Elaborado pelo autor

A Figura 3.21 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 3.0s, sobre-sinal de 10.32% e tempo de acomodação de 10.8s. Também mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento com os ganhos $K_p = 3.4081$, $K_i = 1.1158$, $K_d = 0.6903$ e $K_{rg} = 0.0$ obtidos da otimização, a qual apresenta um tempo de pico de 2.4s, sobre-sinal de 9.86% e tempo de acomodação de 11.24s. Observando a Figura 3.21 é possível ver que

houve remoção de uma forte oscilação durante a resposta ao degrau usando os ganhos otimizados, mantendo um desempenho similar.

Figura 3.21: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de rolamento na plataforma HIL.

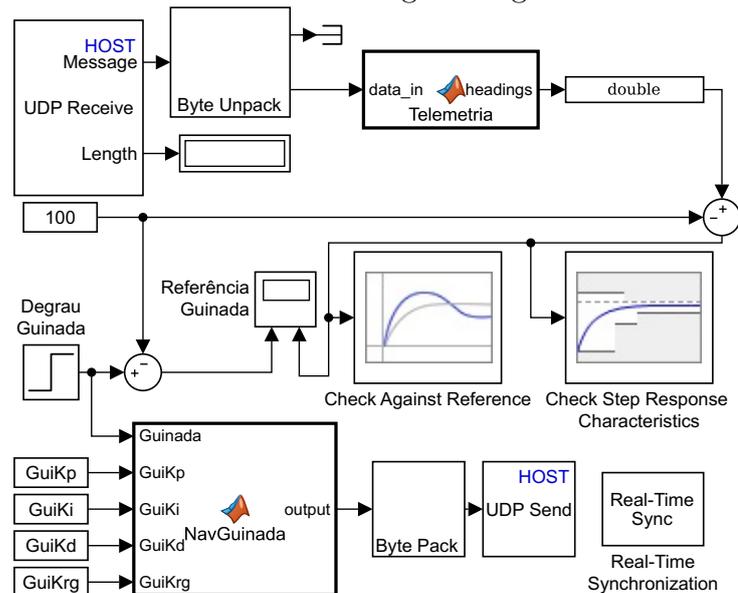


Fonte: Elaborado pelo autor

Guinada

A Figura 3.22 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de guinada.

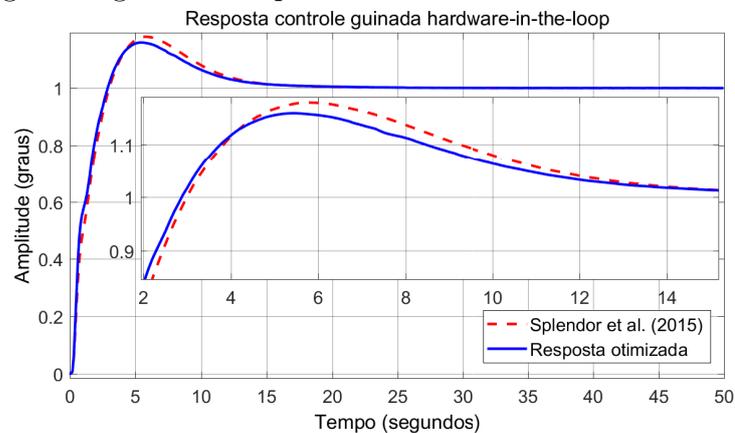
Figura 3.22: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de guinada.



Fonte: Elaborado pelo autor

A Figura 3.23 mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada com os ganhos obtidos por Splendor et al. (2015), a qual apresenta um tempo de pico de 5.3s, sobre-sinal de 15.64% e tempo de acomodação de 12.89s. Também mostra a resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada com os ganhos $K_p = 0.2786$, $K_i = 0.6584$, $K_d = 0.0084$ e $K_{rg} = 0.0943$ obtidos da otimização, a qual apresenta um tempo de pico de 4.7s, sobre-sinal de 14.62% e tempo de acomodação de 11.65s, obtendo uma leve melhora em todos os critérios.

Figura 3.23: Resposta a uma entrada do tipo degrau unitário da malha de controle do ângulo de guinada na plataforma HIL.



Fonte: Elaborado pelo autor

A Tabela 3.3 mostra a comparação de índices de desempenho obtidos antes e depois da otimização na plataforma HIL. Quanto ao ângulo de arfagem houve uma redução no EMQ, aumento no ECM mas uma redução considerável na VSC. No controle de altitude houve redução no EMQ mesmo com controle do ângulo de arfagem mais suave, e também ocorreram consideráveis reduções quanto ao ECM e VSC. Já no ângulo de rolamento apesar de uma aumento quanto ao EMQ, houve uma redução quanto ao ECM e VSC, devido ao novo controle ter ficado com uma resposta mais suave. Já no ângulo de guinada é possível ver uma leve redução em todos os índices.

A coluna A da Tabela 3.4 relaciona os ganhos definidos por Splendor et al. (2015) e descritos na Seção 2.4. Ainda nesta tabela, a coluna B refere-se aos ganhos obtidos com a otimização das malhas de controle definidas na plataforma MIL com funções de transferência e tratadas na Seção 3.2. Por fim, com a otimização direta nas plataformas SIL e HIL os ganhos obtidos são listados nas coluna C e D respectivamente.

Tabela 3.3: Comparação de índices de desempenho da otimização na plataforma HIL.

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	2.25×10^{-3}	0.48×10^0	3.33×10^{-1}
Otimização	1.24×10^{-3}	1.58×10^0	1.75×10^{-1}
taxa	0.55	3.29	0.53
Altitude			
Splendor et al. (2015)	2.09×10^{-3}	4.78×10^{-5}	9.80×10^{-7}
Otimização	1.46×10^{-3}	2.90×10^{-5}	4.91×10^{-7}
taxa	0.70	0.61	0.50
Rolamento			
Splendor et al. (2015)	0.71×10^{-3}	8.54×10^{-2}	8.54×10^{-2}
Otimização	1.47×10^{-3}	5.04×10^{-2}	5.02×10^{-2}
taxa	2.07	0.59	0.59
Guinada			
Splendor et al. (2015)	4.24×10^{-3}	1.43×10^{-2}	7.74×10^{-3}
Otimização	3.58×10^{-3}	1.31×10^{-2}	7.46×10^{-3}
taxa	0.84	0.92	0.96

Fonte: Elaborado pelo autor

3.5 Experimento

Diferentemente da otimização, no experimento as respostas das malhas de controle estão sujeitas ao forte acoplamento entre elas. Além disso, no experimento, uma aplicação é simulada, que busca abordar uma situação de voo real, ao contrário da otimização, onde há somente o processo de otimização dos ganhos dos controladores na resposta ao degrau unitário, cujos ganhos são usados nesta aplicação. Ambas as questões acima podem influenciar diretamente nos resultados obtidos.

3.5.1 Plataforma SIL

Os experimentos com os controladores projetados para o piloto automático, Figura 3.24, estão divididos em 3 etapas. Na etapa 1, o avião encontra-se em um voo nivelado com ângulo de guinada de 100 graus, após 600s. Na etapa 2, a aeronave executa uma curva de 50 graus para a direita (150 graus), por mais 600s. Na etapa 3, a aeronave executa uma

Tabela 3.4: Ganhos obtidos das simulações.

Malha	Ganhos	A (Splendor et al., 2015)	B (MIL)	C (SIL)	D (HIL)
Arfagem	K_p	14.300	7.1278	7.5933	8.9398
	K_i	0.0000	2.0630	3.8066	5.1016
	K_d	0.0000	0.0000	0.1058	0.0797
	K_{rg}	1.1800	1.1800	0.5688	0.5284
Altitude	K_p	0.0112	0.0052	0.0023	0.0038
	K_i	0.0004	0.0001	0.0011	0.0016
	K_d	0.0032	0.0032	0.0030	0.0030
Rolamento	K_p	2.6430	0.5915	3.3105	3.4081
	K_i	1.6498	0.2574	0.7376	1.1158
	K_d	1.1090	0.0662	0.3488	0.6903
	K_{rg}	0.0803	0.0803	0.0803	0.0000
Guinada	K_p	0.2143	0.2786	0.2786	0.2786
	K_i	0.5698	0.4322	1.2987	0.6584
	K_d	0.0089	0.0089	0.0089	0.0084
	K_{rg}	0.1000	0.0562	0.0781	0.0943

Fonte: Elaborado pelo autor

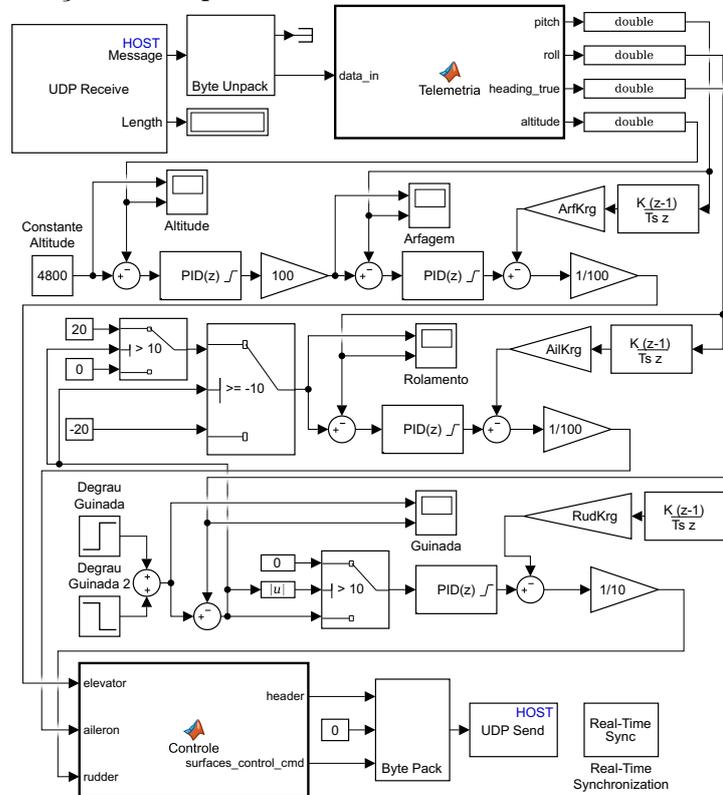
curva de 50 graus para esquerda (100 graus). Em cada curva, se o erro de guinada for maior que 10 graus, o avião se inclina 20 graus na direção da curva para corrigir o erro. Se não, o controle do ângulo de guinada (leme) é usado para corrigir o erro. A altitude é mantida em 4800 pés e a velocidade horizontal considerada é de aproximadamente 144.3mph.

Os resultados obtidos nesse experimento para os controladores projetados para o piloto automático do avião Cessna 182 aplicados ao avião Cessna 172 no simulador de voo X-Plane são apresentados nas Figuras 3.25 a 3.28. Nas figuras o eixo da ordenada dos gráficos (eixo Y) representa a variável a ser controlada e o eixo da abscissa dos gráficos (eixo X) representa o tempo. Dessa forma, tem-se a variável a ser controlada em relação ao tempo.

Malhas de Controle Longitudinal

As respostas obtidas na malha de controle do ângulo de arfagem são apresentadas na forma gráfica na Figura 3.25, utilizando os ganhos das colunas A, B e C da Tabela 3.4. No início da simulação a acomodação ocorre na faixa dos 500 segundos iniciais (Etapa

Figura 3.24: Modelo do Simulink usado na plataforma SIL com malhas de controle para realização dos experimentos.

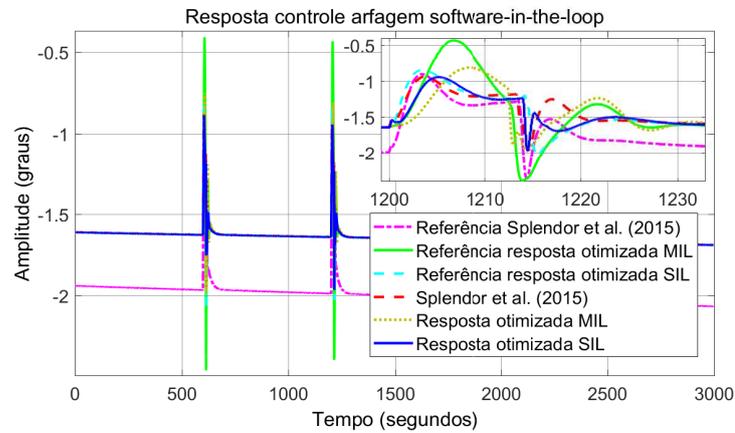


Fonte: Elaborado pelo autor

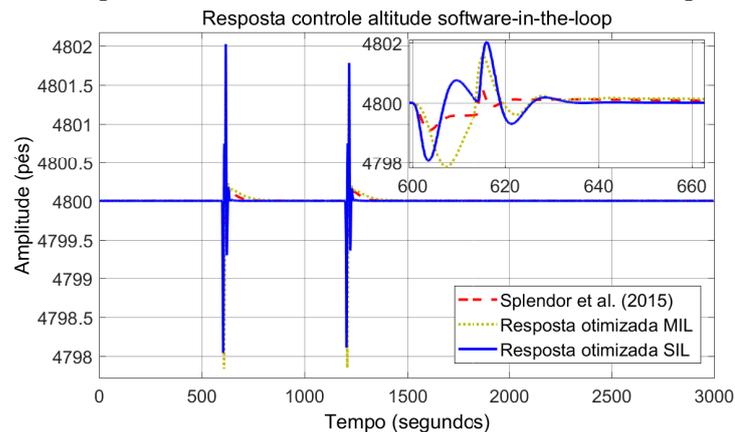
1). A primeira oscilação, entre 500 e 1000 segundos, representa a curva para a direita (Etapa 2). A segunda oscilação, entre 1000 e 1500 segundos, representa a curva para a esquerda (Etapa 3). Uma análise desse gráfico mostra a eliminação do erro estacionário pela utilização do parâmetro integrativo dos controles otimizados.

Considerando as mesmas etapas e faixas de tempo simuladas para a malha de controle do ângulo de arfagem, a resposta obtida na malha de controle de altitude é apresentada graficamente na Figura 3.26, utilizando os ganhos das colunas A, B e C da Tabela 3.4.

Visualizando a Tabela 3.5 pode-se realizar uma análise dos resultados de picos (M_p) e de tempos de pico (t_p) obtidos das simulações. Nela é possível perceber que houve um aumento na amplitude dos picos da malha de altitude com a otimização. Um possível motivo para isso é o atraso adicional provocado pelo ganho integrativo na malha de controle do ângulo de arfagem, que é uma malha interna da malha de altitude.

Figura 3.25: Resposta da malha de controle do ângulo de arfagem na plataforma SIL.

Fonte: Elaborado pelo autor

Figura 3.26: Resposta da malha de controle de altitude na plataforma SIL.

Fonte: Elaborado pelo autor

Malhas de Controle Latero-direcional

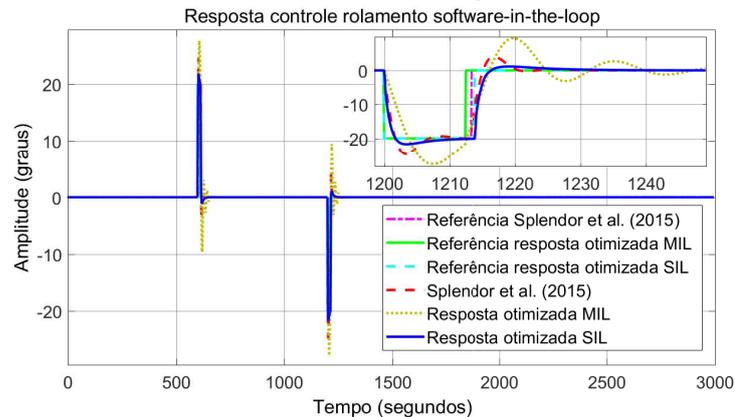
A Figura 3.27 mostra as respostas fornecidas pela malha de controle do ângulo de rolamento levando em conta a utilização dos ganhos das colunas A, B e C da Tabela 3.4. Lembrando que, neste caso, as mesmas etapas e faixas de tempo simuladas para a malhas de controle do ângulo de arfagem e de altitude são também consideradas.

Da mesma forma, as respostas fornecidas pela malha de controle do ângulo de guinada são apresentadas na Figura 3.28 com os ganhos das colunas A, B e C da Tabela 3.4.

Na Tabela 3.6 é possível analisar os resultados obtidos dos picos (Mp), dos tempos de pico (tp) das simulações e dos tempos de acomodação (ts) pelo critério de 2%. Nos tempos de acomodação (ts) marcados com * não houve tempo para a acomodação.

Tabela 3.5: Respostas da malha de controle de altitude na plataforma SIL.

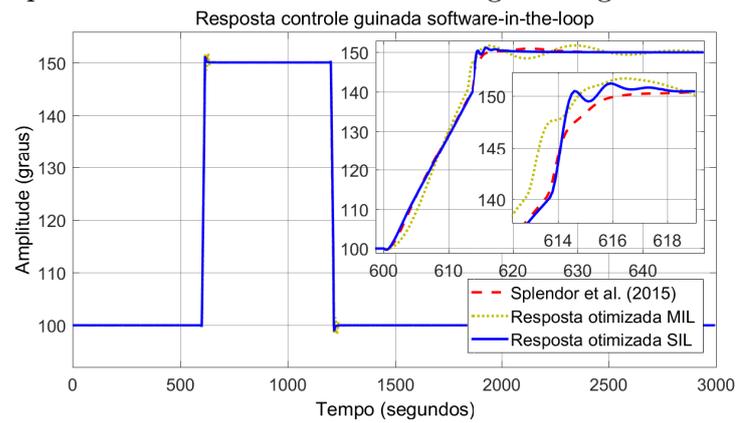
Ganhos da coluna A da Tabela 3.4.					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.5s	-0.0185%	Abaixo	3.5s	-0.0182%
Acima	15.3s	0.0109%	Acima	15.1s	0.0094%
Ganhos da coluna B da Tabela 3.4.					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	6.9s	-0.0457%	Abaixo	6.8s	-0.0438%
Acima	13.8s	0.0296%	Acima	13.8s	0.0294%
Ganhos da coluna C da Tabela 3.4.					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.1s	-0.0407%	Abaixo	3.1s	-0.0392%
Acima	15.0s	0.0448%	Acima	15.1s	0.0369%

Figura 3.27: Resposta da malha de controle do ângulo de rolamento na plataforma SIL.

Fonte: Elaborado pelo autor

A Tabela 3.7 mostra a comparação de índices de desempenho da obtidos no experimento na plataforma SIL. Quanto ao ângulo de arfagem houve uma diminuição expressiva no valor do EMQ devido a eliminação do erro estacionário durante o experimento, mas pouca variação nos índices ECM e VSC. No controle de altitude houve redução em todos os índices. Já nos ângulos de rolamento e de guinada os índices se mantiveram similares.

Figura 3.28: Resposta da malha de controle do ângulo de guinada na plataforma SIL.



Fonte: Elaborado pelo autor

Tabela 3.6: Respostas da malha de controle latero-direcional na plataforma SIL.

Rolamento				Guinada		
Ganhos da coluna A da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.4s	21.66%	*	22.9s	1.97%	15.65s
0	3.5s	20.63%	6.8s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.3s	21.99%	*	22.6s	1.99%	15.40s
0	3.6s	21.33%	6.8s			
Ganhos da coluna B da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	7.1s	37.79%	*	27.8s	3.61%	30.35s
0	7.1s	46.88%	43.96s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	7.0s	36.92%	*	27.5s	3.75%	30.23s
0	7.1s	49.13%	44.20s			
Ganhos da coluna C da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.1s	8.35%	8.90s	13.6s	3.87%	15.22s
0	4.7s	6.67%	13.56s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.1s	7.20%	8.69s	14.9s	2.06%	14.95s
0	5.5s	5.08%	12.85s			

Tabela 3.7: Comparação de índices de desempenho da otimização no experimento na plataforma SIL.

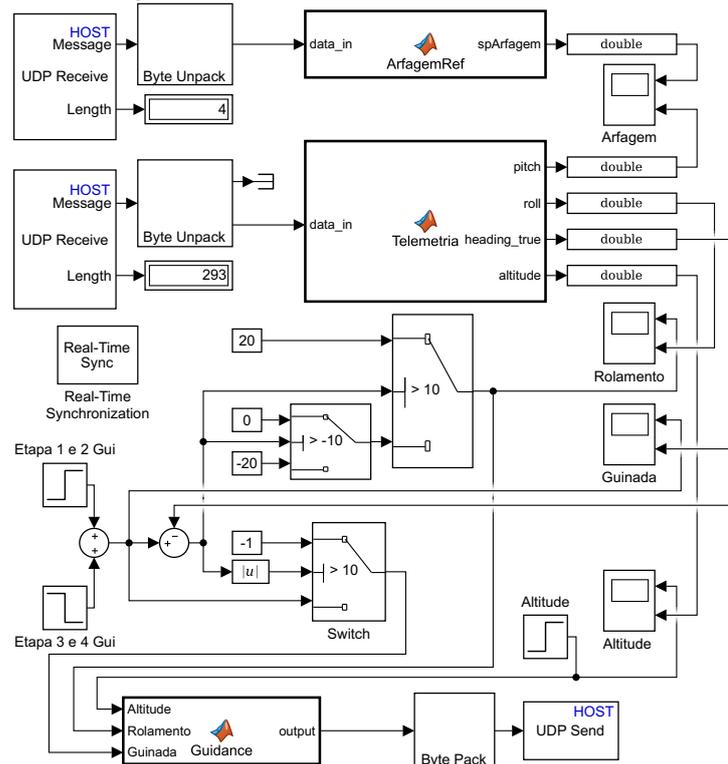
Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	124×10^{-3}	2.55×10^1	5.75×10^{-1}
Otimização MIL	2.78×10^{-3}	2.55×10^1	5.93×10^{-1}
taxa MIL	0.02	1.00	1.03
Otimização SIL	4.01×10^{-3}	2.58×10^1	5.63×10^{-1}
taxa SIL	0.03	1.01	0.98
Altitude			
Splendor et al. (2015)	14.8×10^{-2}	3.97×10^{-4}	8.05×10^{-7}
Otimização MIL	9.15×10^{-2}	2.71×10^{-4}	7.19×10^{-7}
taxa MIL	0.62	0.68	0.89
Otimização SIL	4.57×10^{-2}	2.72×10^{-4}	5.84×10^{-7}
Rolamento			
Splendor et al. (2015)	3.37×10^{-1}	6.73×10^0	6.29×10^0
Otimização MIL	10.4×10^{-1}	1.57×10^0	1.13×10^0
taxa MIL	3.09	0.23	0.18
Otimização SIL	3.64×10^{-1}	7.26×10^0	6.82×10^0
taxa SIL	1.08	1.08	1.08
Guinada			
Splendor et al. (2015)	1.35×10^{-2}	6.98×10^{-2}	6.92×10^{-2}
Otimização MIL	3.13×10^{-2}	6.13×10^{-2}	6.06×10^{-2}
taxa MIL	2.32	0.88	0.88
Otimização SIL	4.13×10^{-2}	7.18×10^{-2}	7.11×10^{-2}
taxa SIL	3.06	1.03	1.03

Fonte: Elaborado pelo autor

3.5.2 Plataforma HIL

Os experimentos com os controladores projetados para o piloto automático, Figura 3.29, são similares aos realizados na plataforma SIL.

Figura 3.29: Modelo do Simulink usado na plataforma HIL com o controle de navegação para realização dos experimentos.



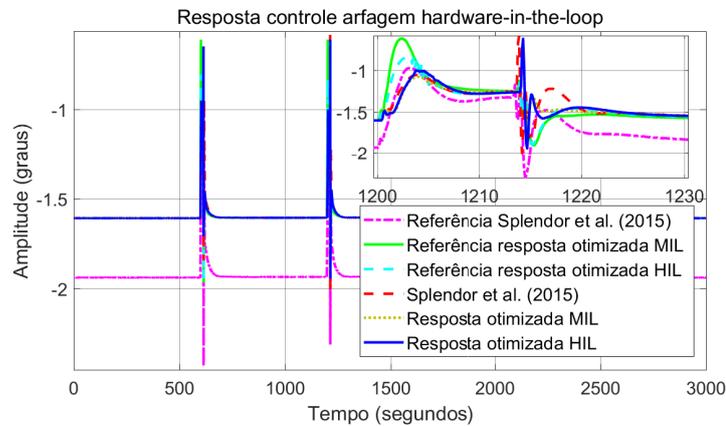
Fonte: Elaborado pelo autor

Malhas de Controle Longitudinal

As respostas adquiridas da malha de controle do ângulo de arfagem são apresentadas nos gráficos da Figura 3.30 e as respostas adquiridas da malha de controle de altitude são apresentadas nos gráficos da Figura 3.31 utilizando os ganhos das colunas A, B e D da Tabela 3.4.

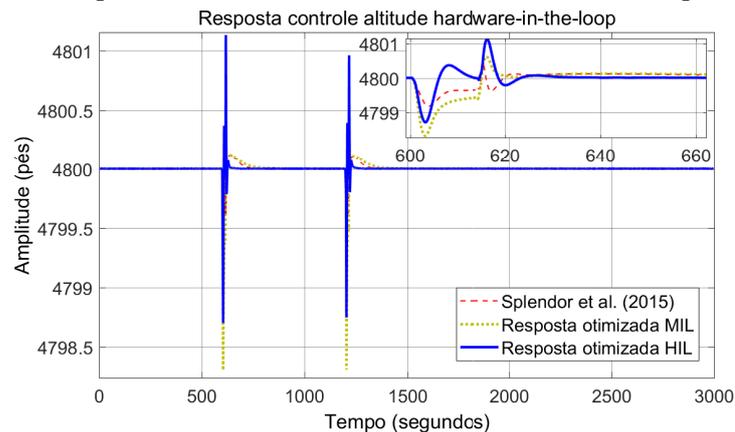
Por meio dos dados apresentados na Tabela 3.8 é possível analisar os resultados de picos (M_p) e de tempos de pico (t_p) obtidos nas simulações. Nela é possível perceber que houve um aumento na amplitude dos picos da malha de altitude com a otimização. Um possível motivo para ter ocorrido esse aumento é o atraso adicional provocado pelo ganho integrativo na malha do ângulo de arfagem, que é uma malha interna da malha de altitude.

Figura 3.30: Resposta da malha de controle do ângulo de arfagem na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 3.31: Resposta da malha de controle de altitude na plataforma HIL.



Fonte: Elaborado pelo autor

Malhas de Controle Latero-direcional

Os gráficos apresentados na Figura 3.32 mostram as respostas obtidas na malha de controle do ângulo de rolamento ao se considerar os ganhos das colunas A, B e D da Tabela 3.4. Lembrando que, nesse caso, as mesmas etapas e faixas de tempo simuladas para a malhas de controle do ângulo de arfagem e de altitude também são consideradas.

Da mesma forma, as respostas obtidas na malha de controle do ângulo de guinada são apresentadas nos gráficos da Figura 3.33 com os ganhos das colunas A, B e D da Tabela 3.4.

Na Tabela 3.9 é possível analisar os resultados obtidos dos picos (M_p), dos tempos de pico (t_p) das simulações e dos tempos de acomodação (t_s) pelo critério de %2. Nos tempos de acomodação (t_s) marcados com ‘*’ não houve tempo para a acomodação. Nela

Tabela 3.8: Respostas da malha de controle de altitude na plataforma HIL.
Ganhos da coluna A da Tabela 3.4.

Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	38.0s	-0.0391%	Abaixo	37.2s	-0.0374%
Acima	45.9s	0.0574%	Acima	45.4s	0.0569%

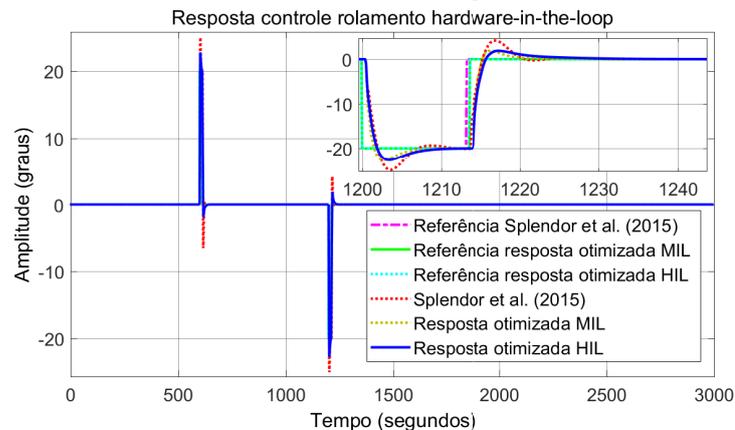
Ganhos da coluna B da Tabela 3.4.

Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	10.2s	-0.0576%	Abaixo	9.1s	-0.0562%
Acima	30.2s	0.0414%	Acima	29.7s	0.0401%

Ganhos da coluna D da Tabela 3.4.

Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	39.9s	-0.0486%	Abaixo	40.5s	-0.0410%
Acima	36.2s	0.1069%	Acima	36.3s	0.1163%

Figura 3.32: Resposta da malha de controle do ângulo de rolamento na plataforma HIL.

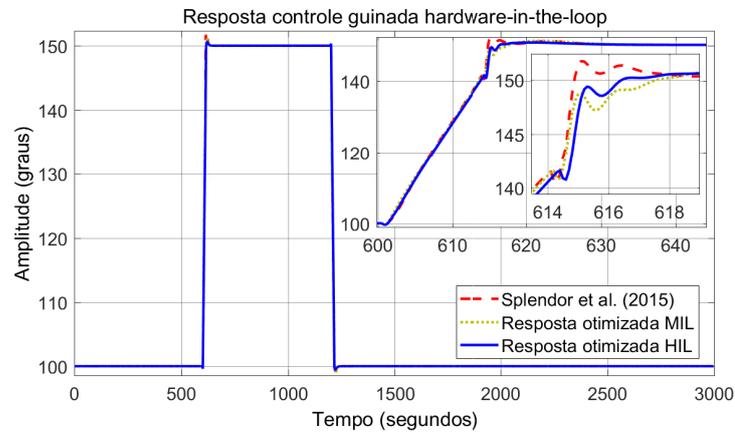


Fonte: Elaborado pelo autor

é possível ver que houve uma piora considerável no desempenho da malha do ângulo de rolamento quando se utiliza os ganhos otimizados por função de transferência, mas uma melhora considerável com a otimização direta na plataforma HIL.

A Tabela 3.10 apresenta uma comparação de índices de desempenho obtidos no experimento na plataforma HIL. Quanto ao ângulo de arfagem houve uma diminuição expressiva no EMQ devido a eliminação do erro estacionário durante o experimento, mas pouca variação no índice ECM e um certo aumento na VSC. No controle de altitude houve aumento no EMQ e na VSC devido a resposta mais lenta do controle do ângulo

Figura 3.33: Resposta da malha de controle do ângulo de guinada na plataforma HIL.



Fonte: Elaborado pelo autor

de arfagem. Já no ângulo de rolamento houve diminuição nos índices mas no ângulo de guinada os índices aumentaram com a otimização.

3.6 Considerações

Foram apresentadas propostas de otimização dos ganhos de controladores de um sistema de piloto automático para o avião Cessna 172/182. O motivo para uso do avião Cessna 182 para desenvolvimento das funções de transferência se justifica pela disponibilidade das derivadas aerodinâmicas na literatura de referência, enquanto que no simulador de voo X-Plane (instalação original) está disponível para uso apenas o modelo para o avião Cessna 172. Apesar de serem diferentes, esses aviões apresentam, em sua maioria, características semelhantes. Apesar disso a otimização na plataforma MIL utilizando funções de transferência nem sempre se reflete em melhorias nos testes com o simulador. Isso evidencia a importância de um modelo matemático preciso que é difícil de se obter.

Nas otimizações, as respostas suaves foram prioridade em todas as malhas de controle. A segunda prioridade foi a eliminação de erros de estado estacionário, especialmente na malha de controle do ângulo de arfagem, com o uso de um controlador PID, em vez do controlador proporcional usado em Splendor et al. (2015). Em terceiro lugar, a minimização de sobre-sinais e tempos de acomodação. Todas estas prioridades foram feitas com o intuito de encontrar um equilíbrio satisfatório entre erros e esforços de controle e, ao mesmo tempo, atender às especificações de projeto. Deve-se enfatizar que existem situações em que existe uma relação inversa entre os índices de desempenho, ou seja,

Tabela 3.9: Respostas da malha de controle latero-direcional na plataforma HIL.

Rolamento			Guinada			
Ganhos da coluna A da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.7s	32.54%	11.28s	44.5s	5.33%	48.17s
0	4.2s	29.75%	14.47s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.7s	32.65%	11.26s	44.0s	5.08%	47.66s
0	4.2s	29.59%	14.47s			
Ganhos da coluna B da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	9.1s	47.93%	*	30.8s	3.02%	48.75s
0	10.1s	50.38%	59.93s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	8.9s	48.58%	*	29.2s	2.97%	47.96s
0	10.1s	49.62%	59.97s			
Ganhos da coluna D da Tabela 3.4.						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.7s	19.64%	7.81s	36.6s	2.00%	36.60s
0	3.6s	15.73%	8.01s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.7s	19.54%	7.74s	43.5s	0.34%	35.82s
0	3.9s	14.92%	7.94s			

forçar a melhora de um deles implica piorar os outros, sem comprometer a estabilidade do sistema.

Outro ponto importante a ser observado é que, nas otimizações usando as plataformas SIL e HIL, com o simulador de voo X-Plane, foram adotados procedimentos para minimizar o acoplamento entre as malhas. Finalmente, nos experimentos, onde foram simuladas uma situação de voo real, o forte acoplamento das malhas de controle foi refletido nos resultados obtidos.

No trabalho de Splendor et al. (2015) os ganhos das malhas de controle foram obtidos por meio dos métodos do lugar das raízes e de estimativa pelo método de ZN. Posteriormente, esses ganhos passaram por um ajuste manual fino, que caracteriza

Tabela 3.10: Comparação de índices de desempenho da otimização no experimento na plataforma HIL.

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	181×10^{-4}	3.70×10^0	5.80×10^{-2}
Otimização MIL	2.69×10^{-4}	4.29×10^0	5.65×10^{-2}
taxa MIL	0.01	1.16	0.97
Otimização HIL	4.49×10^{-4}	4.10×10^0	10.6×10^{-2}
taxa HIL	0.02	1.11	1.82
Altitude			
Splendor et al. (2015)	3.88×10^{-3}	6.07×10^{-5}	8.91×10^{-7}
Otimização MIL	6.08×10^{-3}	4.88×10^{-5}	2.12×10^{-7}
taxa MIL	1.57	0.80	0.24
Otimização HIL	13.4×10^{-3}	4.58×10^{-5}	12.7×10^{-7}
taxa HIL	3.45	0.75	1.43
Rolamento			
Splendor et al. (2015)	5.63×10^{-2}	14.8×10^{-1}	14.1×10^{-1}
Otimização MIL	24.6×10^{-2}	4.08×10^{-1}	3.25×10^{-1}
taxa MIL	4.34	0.28	0.23
Otimização HIL	4.29×10^{-2}	12.9×10^{-1}	12.0×10^{-1}
taxa HIL	0.76	0.87	0.85
Guinada			
Splendor et al. (2015)	4.61×10^{-3}	3.65×10^{-2}	3.64×10^{-2}
Otimização MIL	6.60×10^{-3}	1.32×10^{-2}	1.31×10^{-2}
taxa MIL	1.43	0.36	0.36
Otimização HIL	5.72×10^{-3}	6.19×10^{-2}	6.17×10^{-2}
taxa HIL	1.24	1.70	1.70

Fonte: Elaborado pelo autor

como método de tentativa-e-erro, a partir dos quais foram realizadas as simulações. Já as propostas de otimização apresentadas neste trabalho oferecem uma forma simples, eficiente, sistematizada e replicável de obtenção dos ganhos, com os resultados tendo uma melhora significativa em relação aos obtidos no trabalho de Splendor et al. (2015). Além disto, as propostas de otimização dispensam o uso de métodos clássicos de determinação dos ganhos das malhas de controle, bem como de tentativa-e-erro. Outras vantagens de

seus usos podem ser minimizar consideravelmente a exigência de sintonia em voo, reduzir substancialmente os riscos e custos envolvidos nos testes em voo, e atingir objetivos das tarefas predeterminadas com melhora significativa. Logo, o *Simulink Design Optimization*, por meio de seus blocos de otimização *Check Step Response Characteristics* e *Check Against Reference*, é uma boa alternativa para sintonia de ganhos de controladores.

Embora tenham sido utilizadas especificações de projeto e critérios de otimização diferentes, as três abordagens de controle obtiveram resultados satisfatórios na realização da tarefa predeterminada, de manter a aeronave estabilizada e executar manobras simples como curvas para ambos os lados. Deve-se ressaltar que os modelos obtidos a partir de funções de transferência das malhas de controle podem ser vistos como modelos simplificados do sistema, enquanto que os modelos provenientes do simulador de voo X-Plane podem ser considerados modelos realísticos do sistema. Em suma, esses últimos modelos representam de forma fidedigna o avião real. Os critérios para a escolha final do controlador devem ser avaliados definindo um função custo para cada caso. Essa função custo deve ponderar os critérios de resposta transiente e os índices desempenho. Com isso, o controlador que obter menor custo será conseqüentemente o que irá ter maiores benefícios.

Controles Avançados

Neste capítulo é apresentado um procedimento para determinar os ganhos de um controlador PID com o uso das respostas de um controlador de duas posições (*Relay*). Também são apresentadas algumas variações e alternativas ao controle PID clássico. A proposta desses controladores (*ISA*-PID, *Fuzzy* PID e PID-EG-*Fuzzy*) mais avançados é contornar as limitações dos controladores PID, apresentadas no Capítulo 2. Também se espera superar o desempenho do controle PID clássico otimizado, apresentado no Capítulo 3. Com base nos dados e resultados obtidos neste capítulo, é definido e experimentado, no Capítulo 5 o controlador proposto neste trabalho.

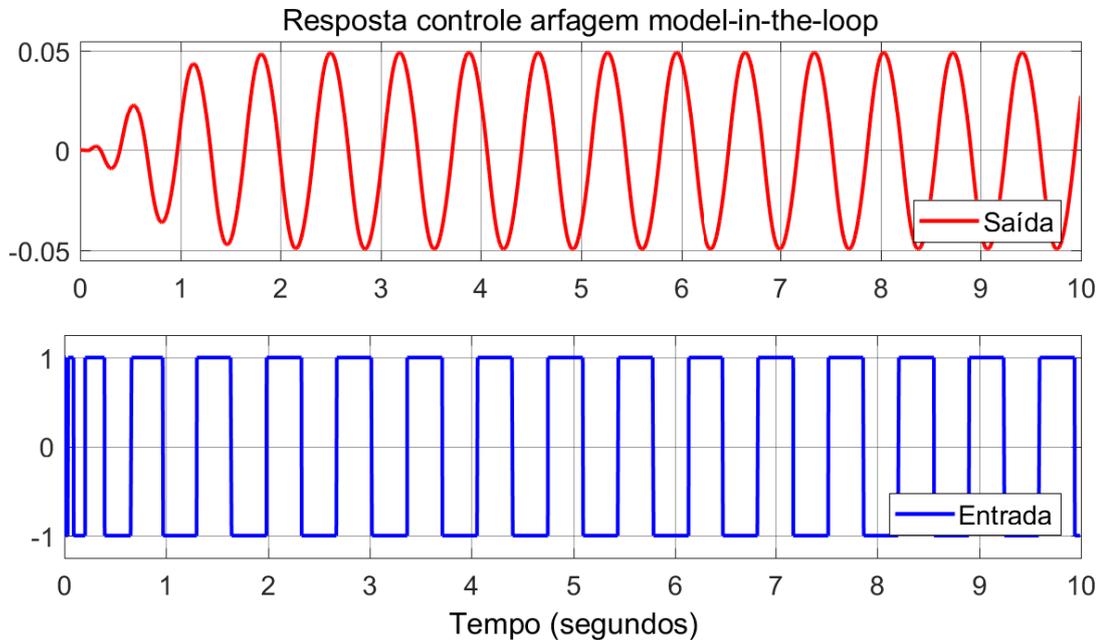
4.1 Procedimento de Auto-sintonia do Controlador PID com o Método Relay

Aplicando o procedimento de auto-sintonia, para determinação de ganhos, descrito por Wilson (2005), com $d = 1$, para a malha do ângulo de arfagem na plataforma *MIL*, obtém-se a resposta apresentada na Figura 4.1.

Assim é possível definir o período crítico $P_{cr} = 0.69$ e a amplitude $a = 0.049$. Pela equação 4.1 pode-se obter o valor de $K_{cr} = 26$ e com a regra de sintonia para o segundo método de ZN define-se $K_p = 0.6K_{cr} = 15.6$, $K_i = K_p/0.5P_{cr} = 45.22$, $K_d = K_p \times 0.125P_{cr} = 1.346$.

$$K_{cr} = \frac{4d}{\pi a} \quad (4.1)$$

Figura 4.1: Resposta controle *relay* da malha de controle do ângulo de arfagem na plataforma MIL.



Fonte: Elaborado pelo autor

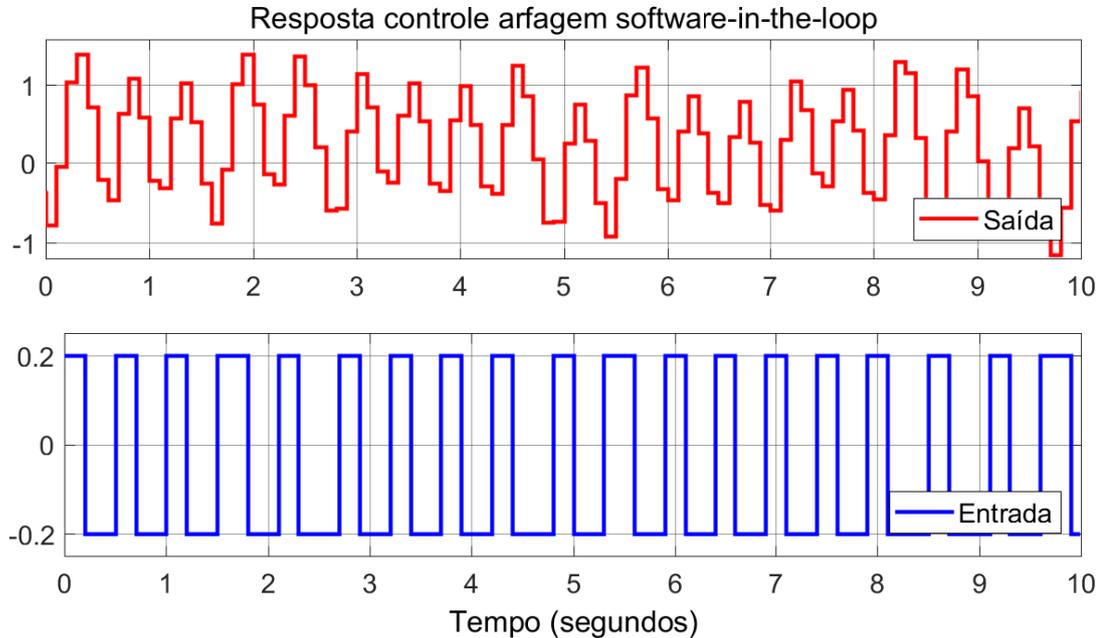
Ao aplicar esse mesmo procedimento de *relay* para a malha do ângulo de arfagem na plataforma *SIL*, com $d = 0.2$, que corresponde a 20% do valor máximo, obtêm-se a resposta apresentada na Figura 4.2.

Considerando que ocorrem 18 ciclos em 10 segundos, foi usado $P_{cr} = 10/18 = 0.56$. Usando a média dos picos altos e baixos, foi considerado $a = 0.9205$. Com isso obtêm-se $K_{cr} = 0.2766$ e, pela regra de ZN, $K_p = 0.1660$, $K_i = 0.5929$, $K_d = 0.01162$.

4.2 Controlador ISA-PID

O controlador *ISA*-PID é um controlador PID com estrutura melhorada, descrito no trabalho de Åström e Hägglund (2006), sendo que *ISA* se refere à Sociedade Internacional de Automação (em inglês *International Society of Automation*). Essa estrutura tem por objetivo melhorar o desempenho do controle PID quanto a rejeição de perturbação e rastreamento de trajetória. Frequentemente um controle PID não consegue ter bom desempenho nesses dois aspectos já que uma otimização para melhorar um desses aspectos muitas vezes acaba prejudicando o outro. Uma solução simples para fazer com que um controle PID tenha um melhor desempenho nos dois aspectos é alterando-o para um controlador *ISA*-PID. Isso melhora a resposta quanto ao rastreamento de trajetória

Figura 4.2: Resposta controle *relay* da malha de controle do ângulo de arfagem na plataforma SIL.

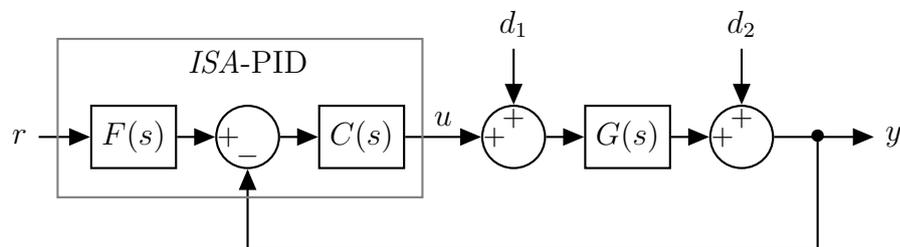


Fonte: Elaborado pelo autor

adicionando mais um parâmetro b que permite um controle independente do impacto do sinal de referência sobre a ação proporcional (The MathWorks, 2017).

Na estrutura da Figura 4.3 há um controle em malha fechada C , que nesse caso é um controle PID, e um filtro F aplicado à referência r , descrito na equação 4.2. O sinal de controle é indicado por u , y é a saída o sistema, d_1 e d_2 são perturbações aplicadas na saída do controle e na saída do sistema respectivamente.

Figura 4.3: Estrutura do Controlador *ISA*-PID.



Fonte: Elaborado pelo autor

$$F(s) = \frac{bK_p s + K_i}{K_p s + K_i} \quad (4.2)$$

Na equação 4.2, K_p e K_i são ganhos de C . b é um número real entre 0 e 1 sendo que, quanto menor o valor de b , menor o sobre-sinal do rastreamento de referência, mas se pequeno demais, pode prejudicar o tempo de acomodação. Em The MathWorks (2017) é sugerido $b = 0.7$.

Com o uso do filtro é possível otimizar o controle PID para melhor desempenho quanto a rejeição de perturbação e ajustar o ganho b para melhorar o desempenho quanto ao rastreamento de referência.

Considerando o degrau unitário no sinal de referência dado por

$$R(s) = \frac{1}{s} \quad (4.3)$$

tem-se que

$$R(s)F(s) = \frac{bK_p s + K_i}{K_p s^2 + K_i s} \quad (4.4)$$

e aplicando a transformada inversa de Laplace tem-se

$$\mathcal{L}^{-1} \left[\frac{bK_p s + K_i}{K_p s^2 + K_i s} \right] = 1 + (b - 1)e^{-\frac{K_i t}{K_p}} \quad (4.5)$$

Essa transformada inversa de Laplace é necessária para implementação do filtro nas plataformas *SIL* e *HIL*.

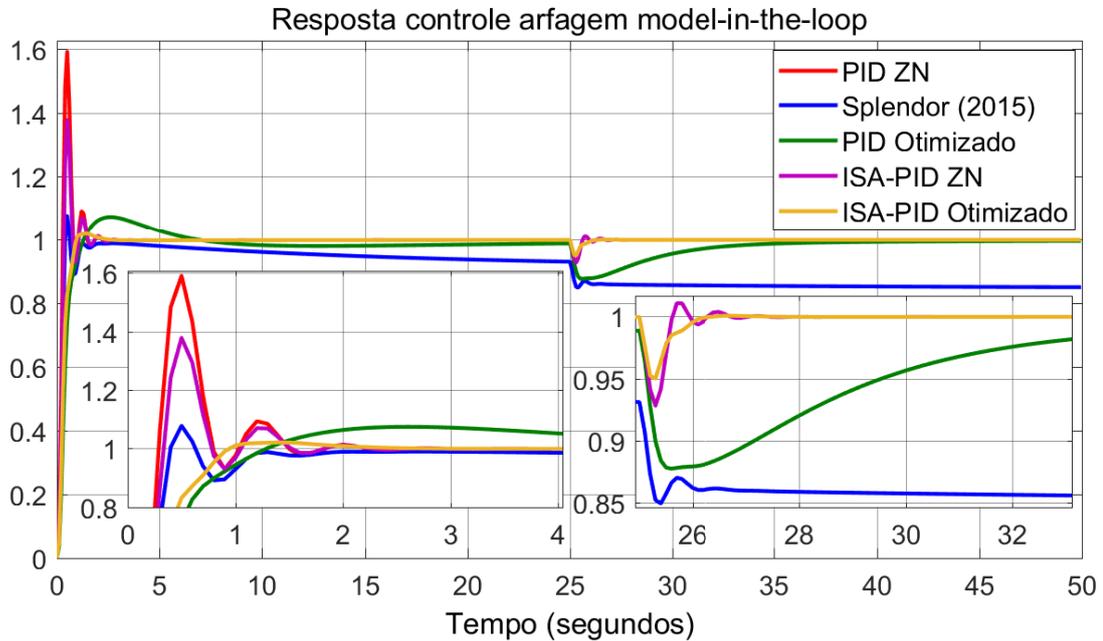
4.2.1 Teste do Controlador ISA-PID na Malha do Ângulo de Arfagem Usando Plataforma MIL

Usando $b = 0.7$ e a regra de ZN como ganhos iniciais, foi feita a otimização do controlador, e foram encontrados os ganhos $b = 0.2952$, $K_p = 20.28$, $K_i = 45.17$, $K_d = 1.346$, $K_{rg} = 1.346$.

A Figura 4.4 mostra as respostas ao degrau unitário da malha de controle do ângulo de arfagem com ganhos obtidos da regra de ZN usando o procedimento *relay*, com os definidos em Splendor et al. (2015), com PID otimizado a partir dos ganhos de Splendor et al. (2015), com a estrutura *ISA-PID* aplicado aos ganhos definidos pelo método de ZN e, por fim, *ISA-PID* com ganhos otimizados a partir de ZN. Aos 25 segundos de simulação é aplicada uma perturbação unitária.

A Tabela 4.1 mostra as características obtidas com cada conjunto de ganhos.

Figura 4.4: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *ISA*-PID na plataforma MIL.



Fonte: Elaborado pelo autor

Tabela 4.1: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *ISA*-PID na plataforma MIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID ZN	0.20s	0.5s	60.6%	1.5s	-7.00%	0.6s
Splendor et al. (2015)	0.23s	0.5s	7.36%	*	*	*
PID Otimizado	0.34s	2.6s	7.36%	5.5s	-12.3%	7.8s
<i>ISA</i> -PID ZN	0.23s	0.5s	38.7%	1.5s	-7.00%	0.6s
<i>ISA</i> -PID Otimizado	0.29s	1.3s	2.00%	0.9s	-5.06%	0.6s

Fonte: Elaborado pelo autor

4.2.2 Teste do Controlador *ISA*-PID na Malha do Ângulo de Arfagem Usando Plataforma SIL

Ao considerar os ganhos obtidos com a otimização na plataforma *MIL* e fazendo o processo inverso do segundo método de ZN podem ser estabelecidas as seguintes regras: $K_p = 0.78K_{cr}$, $T_i = 0.65P_{cr}$, $T_d = 0.096P_{cr}$. Com isso, tem-se $K_p = 0.2157$, $K_i = 0.645$, $K_d = 0.01160$. Foi definido $K_{rg} = K_d = 0.01160$ e foi mantido $b = 0.2952$. Com esses

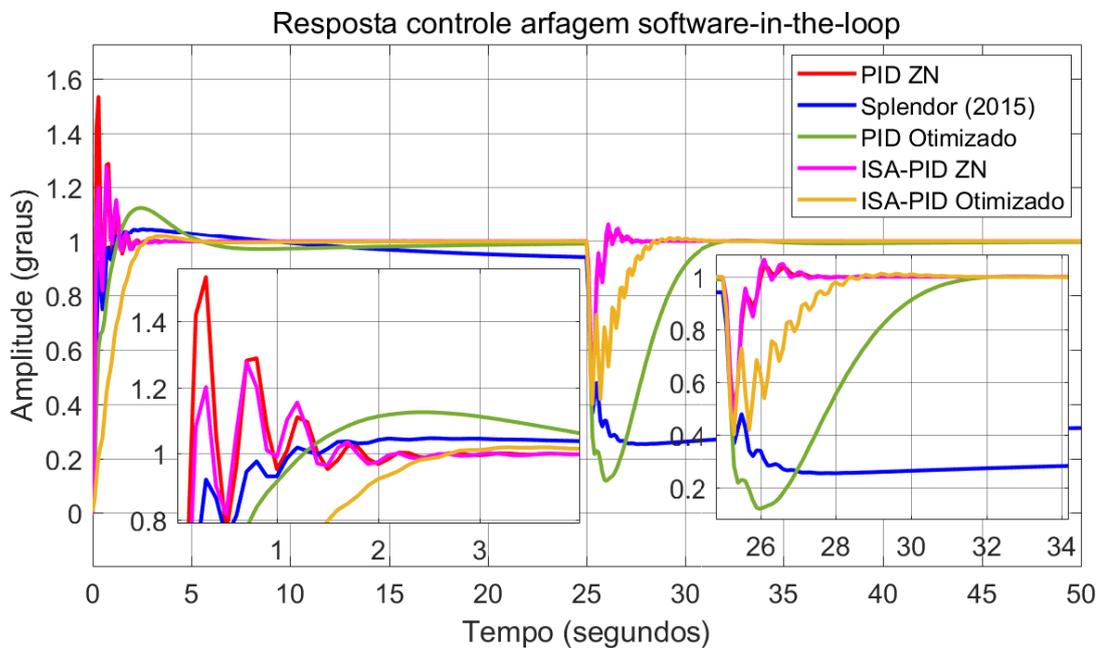
ganhos o controle se tornou instável e por isso não foram feitos mais testes. Ao invés disso foi feita a otimização direta na plataforma *SIL*.

Após a primeira tentativa de otimização foram encontrados os ganhos $K_p = 0.1123$, $K_i = 0.06922$, $K_d = 2.040 \times 10^{-4}$, $K_{rg} = 8.3345 \times 10^{-3}$, $b = 0.2809$ com $t_d = 0.89s$, $t_p = 5.5s$, $M_p = 0\%$, $t_s = 4.3s$, $M_p(D) = -64\%$, $t_s(D) = 5.0s$. Com esses ganhos não foi obtida uma resposta suave, então foi feita uma nova tentativa de otimização.

Após a segunda otimização se obteve um resposta suave quanto ao degrau mas não quanto a perturbação. Aparentemente, quanto mais suave a resposta a perturbação, maior será o pico de erro provocado por ela e maior será o tempo necessário para o retorno ao valor de referência.

A Figura 4.5 mostra as respostas ao degrau unitário da malha do ângulo de arfagem com ganhos obtidos da regra de ZN usando o procedimento *relay*, com os definidos em Splendor et al. (2015), com PID otimizado a partir dos ganhos de Splendor et al. (2015), com filtro na referência aplicado ao ZN e, por fim, *ISA-PID* com ganhos otimizados a partir de ZN.

Figura 4.5: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma *SIL*.



Fonte: Elaborado pelo autor

A Tabela 4.2 mostra as características obtidas com cada conjunto de ganhos.

Tabela 4.2: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma SIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID ZN	0.10s	0.3s	53.3%	2.1s	51.9%	1.76s
Splendor et al. (2015)	0.16s	2.5s	4.62%	*	*	*
PID Otimizado	0.25s	2.4s	12.5%	15.2s	88.3%	6.14s
ISA-PID ZN	0.12s	0.7s	27.6%	2.0s	55.0%	1.70s
ISA-PID Otimizado	0.83s	3.3s	1.87%	2.4s	60.0%	3.25s

Fonte: Elaborado pelo autor

4.3 Controladores Fuzzy PID

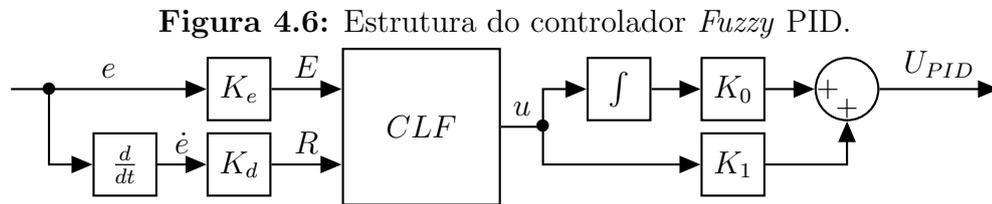
As estratégias de controle inteligente buscam modelar o comportamento humano. Uma das ferramentas mais poderosas capaz de converter regras de controle linguísticas baseadas no conhecimento de um especialista é o controle baseado em lógica *fuzzy* (CLF). A sua filosofia difere dos métodos anteriores ao adaptar o conhecimento especializado do operador no projeto do controlador (Le-Huy, 1995).

A lógica *fuzzy*, também chamada de nebulosa ou difusa, teve seu início com o trabalho de Zadeh (1965) que introduziu o conceito de conjuntos *fuzzy*, apresentando um novo paradigma computacional capaz de modelar as incertezas próprias do raciocínio humano (Dualibe et al., 2007; Tanaka, 1996).

Como um tipo de método de controle não-linear, o campo de aplicação do CLF ampliou para incluir muitas aplicações de controle industrial devido o CLF ser mais robusto do que o controle PID convencional (Mudi e Pal, 1999; Sugeno, 1985). A maioria das aplicações pertence à classe de controladores *Fuzzy* PID (Mann et al., 1999). O controlador *Fuzzy* PID geralmente possui três configurações com base no número de entradas. Existem os de uma entrada, duas entradas e de três ou mais entradas. Os de uma entrada e de três ou mais entradas não foram amplamente utilizados devido a poucas regras e as regras serem complexas demais, respectivamente. Portanto, a maioria das aplicações pertence a controladores *Fuzzy* PID de duas entradas. Nesses controladores, apenas em alguns casos os aspectos de sintonia foram discutidos (Duan et al., 2008).

CLFs não dependem de um modelo matemático preciso do dispositivo controlado, mas, por outro lado, requerem conhecimento especialista para modelagem do sistema. Assim o projeto do controlador terá um desempenho satisfatório quanto maior for o conhecimento do especialista que o projetou (Mrad e Deeb, 2002).

O controlador *Fuzzy* PID é mostrado na Figura 4.6, o qual combina recursos dos controladores *Fuzzy* PD e *Fuzzy* PI (Li et al., 1997).



Fonte: Elaborado pelo autor

Escolher um número apropriado de regras é importante em um sistema *fuzzy*, pois regras demais resultam em um sistema *fuzzy* complexo que pode ser desnecessário para o problema, enquanto que regras de menos produz um sistema *fuzzy* menos robusto que pode ser insuficiente para atingir o objetivo (Wang, 1997).

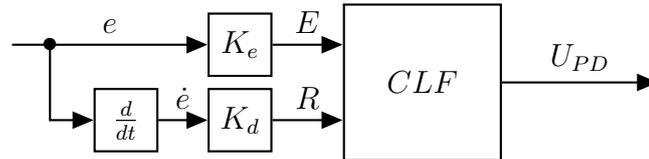
Se por um lado o desempenho de controladores *Fuzzy* PID tem dado bons resultados em muitas aplicações, por outro lado seu desempenho depende também de seu ajuste, que deve ser feito na base da tentativa-e-erro. De fato, ao se projetar qualquer controlador *fuzzy*, existem mais opções do que no caso de controladores convencionais, sendo assim influenciados por uma grande quantidade de graus de liberdade. Frequentemente alega-se que o projeto de controladores *fuzzy* é feito *ad hoc* e que, de fato, não existe metodologia sistemática de projeto. Os que manifestam tais opiniões parecem esquecer-se que nenhum procedimento de projeto, por mais sistemático que seja, será válido a menos que o projetista esteja familiarizado com o processo físico ou químico a ser controlado, incluindo seus problemas e peculiaridades (Simões e Shaw, 1999).

4.3.1 Teste do Controlador Fuzzy PID na Malha do Ângulo de Arfagem Usando Plataforma SIL

No trabalho de Wahid e Rahmat (2010) é implementado um CLF para o controle do ângulo de arfagem do que é chamado de Avião de Aviação Geral (*General Aviation Airplane*). Esse CLF é do tipo PD conforme mostrado na Figura 4.7.

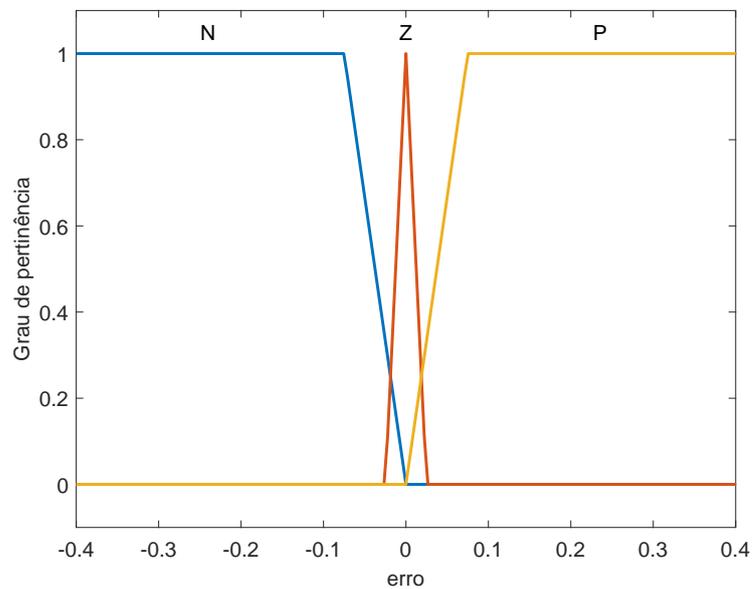
Cada conjunto *fuzzy* consiste em três tipos de funções de pertinência: N de negativo (*negative*), Z de zero e P de positivo (*positive*). A função de pertinência apropriada para representar cada conjunto *fuzzy* precisa ter um universo de discurso apropriado. Em Wahid e Rahmat (2010) é definido o universo de discurso entre -0.4 e 0.4 para o erro e -0.09 e 0.09 para a derivada do erro. As Figuras 4.8, 4.9 e 4.10 mostram os conjuntos *fuzzy* das entradas e da saída.

Figura 4.7: Estrutura do controlador *Fuzzy* PD.



Fonte: Elaborado pelo autor

Figura 4.8: Conjunto *fuzzy* para a entrada do erro.



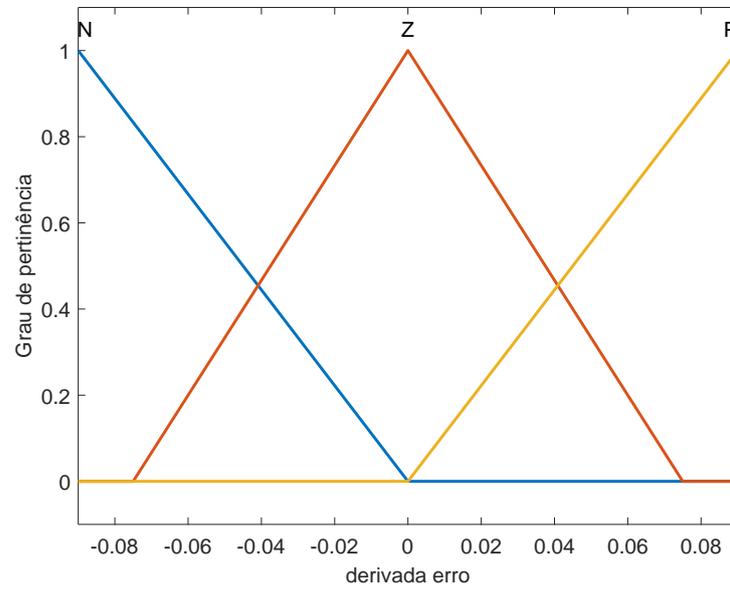
Fonte: Elaborado pelo autor

O trabalho de Wahid e Rahmat (2010) não esclarece como esses valores foram definidos e também não esclarece os detalhes da implementação. Por isso, esse trabalho não pode ser replicado fidedignamente.

Para o teste foi desenvolvido um controlador do tipo *Fuzzy* PID. As funções de pertinência são mostradas na Figura 4.11, onde *N* significa negativo (*negative*), *P* positivo (*positive*), *ZE* zero, *S* pequeno (*small*), *M* médio (*medium*) e *B* grande (*big*) e as regras na Tabela 4.3.

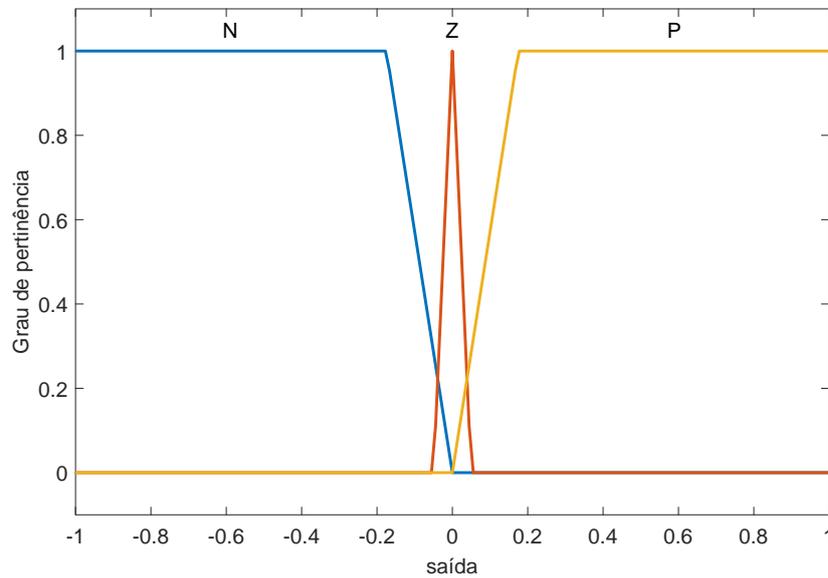
Foi utilizado o *Simulink Design Optimization* de otimização para encontrar os ganhos $K_e = 1.41$, $K_d = 0.3$, $K_0 = 4$, $K_1 = 12$ conforme a Figura 4.6. A Figura 4.12 mostra as respostas ao degrau unitário da malha de controle do ângulo de arfagem comparadas com o controlador *Fuzzy* PID na plataforma MIL.

Figura 4.9: Conjunto *fuzzy* para a entrada da derivada do erro.



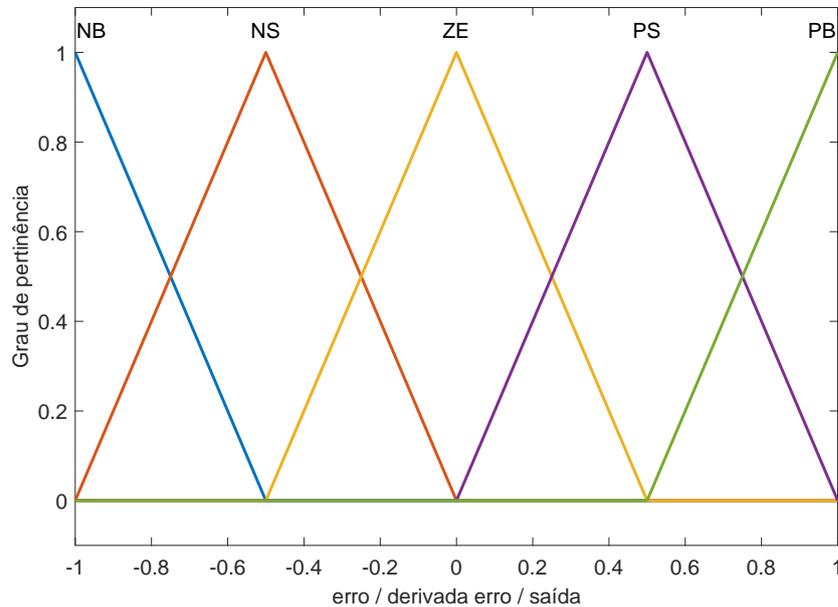
Fonte: Elaborado pelo autor

Figura 4.10: Conjunto *fuzzy* para a saída.



Fonte: Elaborado pelo autor

A Tabela 4.4 mostra as características obtidas com cada conjunto de ganhos. Como pode ser observado, apesar do *Fuzzy* PID ter obtido resultados melhores que o PID otimizado, o *ISA*-PID ainda é mais satisfatório.

Figura 4.11: Conjunto *fuzzy* para as entradas e saída.

Fonte: Elaborado pelo autor

Tabela 4.3: Regras *fuzzy* para o controlador.

	$\Delta e(k)$				
	NB	NS	ZE	PS	PB
$e(k)$	NB	NB	NB	NS	ZE
	NS	NB	NS	ZE	PS
	ZE	NB	NS	ZE	PS
	PS	NS	ZE	PS	PB
	PB	ZE	PS	PB	PB

Fonte: Elaborado pelo autor

Tabela 4.4: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *Fuzzy* PID na plataforma MIL.

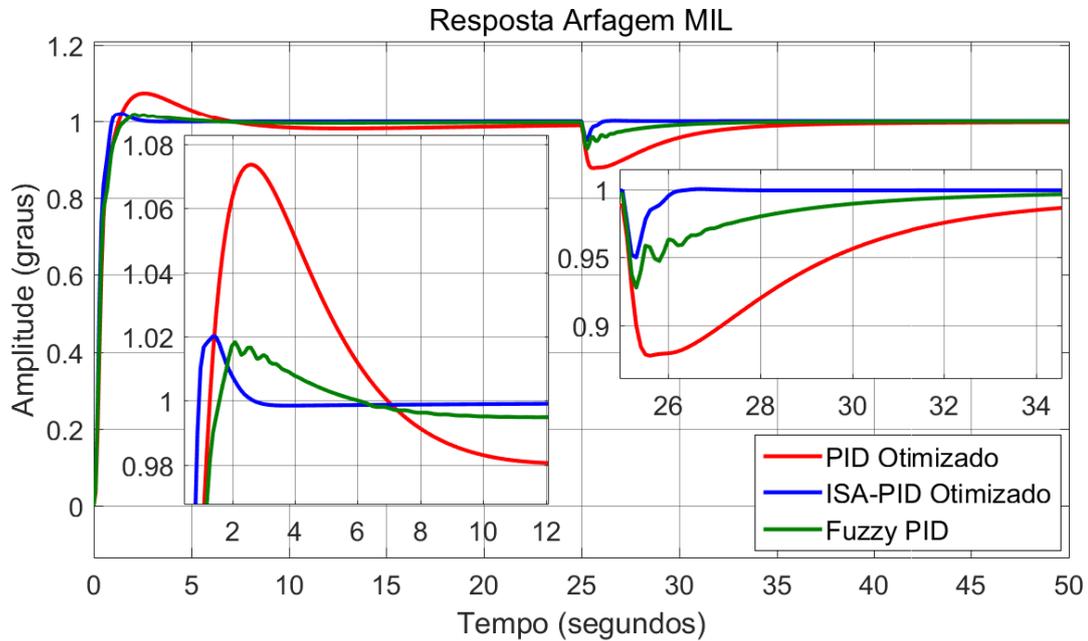
Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID Otimizado	0.34s	2.6s	7.36%	5.5s	-12.3%	7.8s
<i>ISA</i> -PID Otimizado	0.29s	1.3s	2.00%	0.9s	-5.06%	0.6s
<i>Fuzzy</i> PID	0.32s	2.1s	1.83%	1.3s	-7.2%	2.9s

Fonte: Elaborado pelo autor

4.4 Controlador PID-EG-Fuzzy

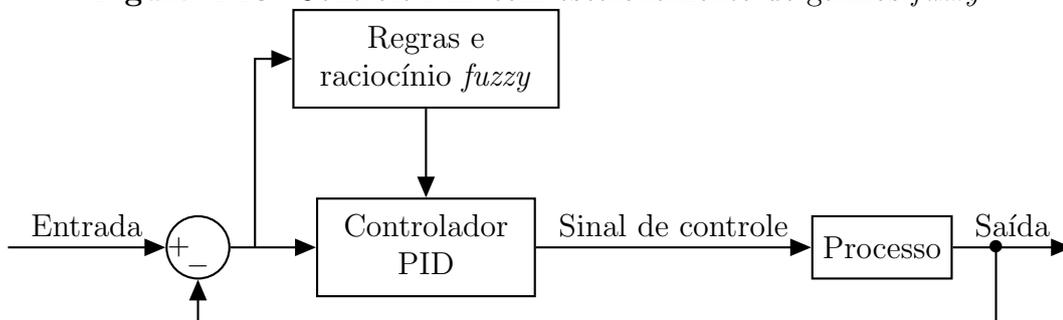
A Figura 4.13 mostra um controle PID com escalonamento de ganhos *fuzzy*. A ideia é explorar as regras e o raciocínio *fuzzy* para gerar os ganhos do controlador (Zhao et al., 1993).

Figura 4.12: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *Fuzzy* PID na plataforma MIL.



Fonte: Elaborado pelo autor

Figura 4.13: Controle PID com escalonamento de ganhos *fuzzy*.



Fonte: Adaptado de Zhao et al. (1993)

É assumido que K_p e K_d são definidos nos intervalos $[K_{p,min}, K_{p,max}]$ e $[K_{d,min}, K_{d,max}]$, respectivamente. Os intervalos apropriados são determinados experimentalmente. Por conveniência, K_p e K_d são normalizados no intervalo entre 0 e 1 pela transformação linear das equações 4.6 e 4.7.

$$K'_p = \frac{K_p - K_{p,min}}{K_{p,max} - K_{p,min}} \quad (4.6)$$

$$K'_d = \frac{K_d - K_{d,min}}{K_{d,max} - K_{d,min}} \quad (4.7)$$

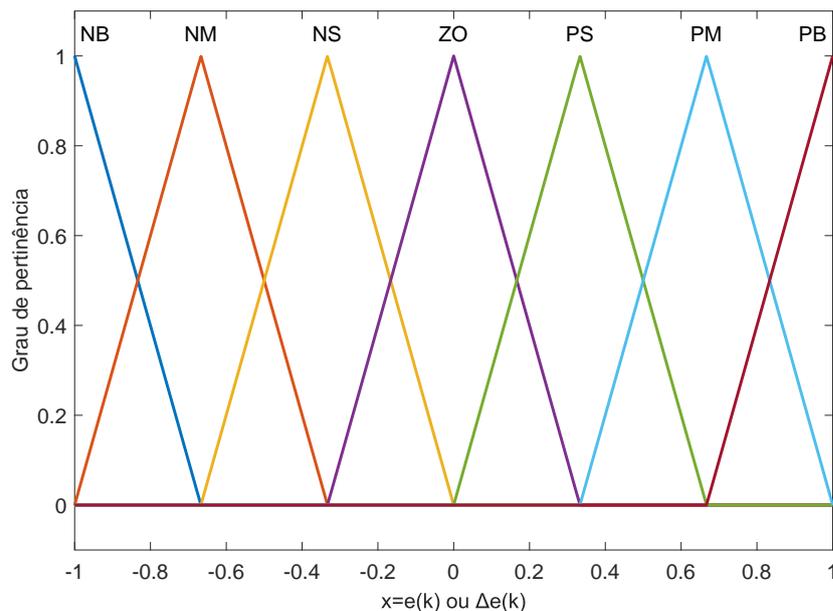
No esquema proposto por Zhao et al. (1993), os ganhos do PID são determinados com base no erro atual $e(k)$ e sua primeira derivada $\Delta e(k)$. A constante de tempo integral é determinada com base na constante de tempo derivativa, conforme a equação 4.8, e o ganho integral é então obtido pela equação 4.9.

$$T_i = \alpha T_d \quad (4.8)$$

$$K_i = \frac{K_p}{\alpha T_d} = \frac{K_p^2}{\alpha K_d} \quad (4.9)$$

Os parâmetros K'_p , K'_d , e α são determinados pelo conjunto de regras *fuzzy*, que podem ser extraídas da experiência do operador (Zhao et al., 1993). As funções de pertinência para $e(k)$ e $\Delta e(k)$ são mostradas na Figura 4.14.

Figura 4.14: Funções de pertinência para $e(k)$ e $\Delta e(k)$.

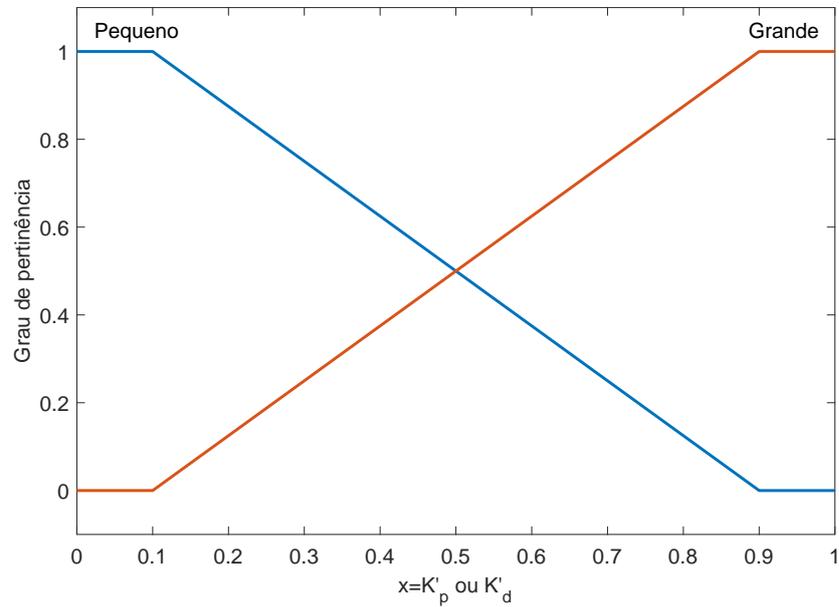


Fonte: Adaptado de Zhao et al. (1993)

As funções de pertinência para as saídas K'_p e K'_d foram adaptadas para funções trapezoidais para facilitar a implementação e são mostradas na Figura 4.15.

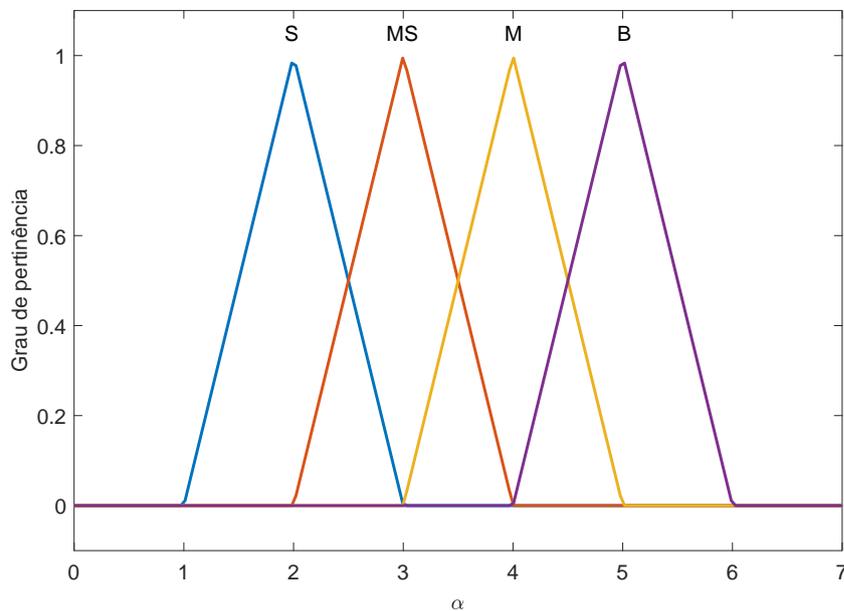
No método de ZN, a constante de tempo integral T_i é definida como quatro vezes a constante de tempo derivativa. Isto é, α é igual a 4. No esquema proposto por Zhao et al. (1993), α pode assumir um valor menor que 4 (por exemplo, 2) para gerar uma ação integral que diminua mais rapidamente o erro. As funções de pertinência para os valores de α foram adaptadas para funções triangulares e são mostradas na Figura 4.16.

Figura 4.15: Funções de pertinência para as saídas K'_p e K'_d .



Fonte: Adaptado de Zhao et al. (1993)

Figura 4.16: Funções de pertinência para α .



Fonte: Adaptado de Zhao et al. (1993)

Foi feita a correspondência de 2 para S (*Small*), 3 para MS (*Medium Small*), 4 para M (*Medium*) e 5 para B (*Big*). Os conjuntos de regras usados para definir os valores de K'_p , K'_d e α são dados nas Tabelas 4.5, 4.6 e 4.7 respectivamente.

Tabela 4.5: Regras *fuzzy* para K'_p .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZE	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Fonte: Adaptado de Zhao et al. (1993)

Tabela 4.6: Regras *fuzzy* para K'_d .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZE	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

Fonte: Adaptado de Zhao et al. (1993)

Tabela 4.7: Regras *fuzzy* para α .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	S	S	S	S	S	S	S
	NM	MS	MS	S	S	S	MS	MS
	NS	M	MS	MS	S	MS	MS	M
	ZE	B	M	MS	MS	MS	M	B
	PS	M	MS	MS	S	MS	MS	M
	PM	MS	MS	S	S	S	MS	MS
	PB	S	S	S	S	S	S	S

Fonte: Adaptado de Zhao et al. (1993)

Uma vez que K'_p , K'_d e α são obtidos, os ganhos do controlador PID são calculados usando as equações 4.10, 4.11 e 4.12.

$$K_p = (K_{p,max} - K_{p,min})K'_p + K_{p,min} \quad (4.10)$$

$$K_d = (K_{d,max} - K_{d,min})K'_d + K_{d,min} \quad (4.11)$$

$$K_i = K_p^2 / (\alpha K_d) \quad (4.12)$$

Com base em extensivo estudo de simulação em vários processos, Zhao et al. (1993) define uma regra geral para determinar o intervalo de K_p e K_d , dado pelas equações 4.13 e 4.14, onde K_{cr} e P_{cr} são, respectivamente, o ganho e o período de oscilação no limite de estabilidade sobre o controle proporcional (Ziegler e Nichols, 1942).

$$K_{p,min} = 0.32K_{cr} \quad K_{p,max} = 0.6K_{cr} \quad (4.13)$$

$$K_{d,min} = 0.08K_{cr}P_{cr} \quad K_{d,max} = 0.15K_{cr}P_{cr} \quad (4.14)$$

No trabalho de Wahid e Hassan (2012) é apresentado um controlador PID *Fuzzy* auto ajustado específico para o controle do ângulo de arfagem de aeronaves. As funções de pertinência das entradas são apresentadas nas Figuras 4.17 e 4.18. Ainda, os valores mínimos de K_p , K_i e K_d são somados diretamente as saídas do controlador *fuzzy* para definir os ganhos que serão utilizados no PID. As funções de pertinência das saídas são apresentadas nas Figuras 4.19, 4.20 e 4.21.

Os conjuntos de regras usados para definir os valores de K_p , K_i e K_d são dados nas Tabelas 4.8, 4.9 e 4.10 respectivamente.

4.4.1 Teste do Controlador PID-EG-Fuzzy Usando Plataforma MIL

Para o teste do controlador proposto em Zhao et al. (1993), considerando $P_{cr} = 0.69$ e $K_{cr} = 26$, definidos anteriormente, tem-se que $K_{p,min} = 8.32$, $K_{p,max} = 15.6$, $K_{d,min} = 1.4352$ e $K_{d,max} = 2.691$. Para o controlador proposto por Wahid e Hassan (2012) não fica explícito como os ganhos são determinados. Por isso foi usado o *Simulink Design Optimization* para encontrar os ganhos. Foram encontrados os seguintes ganhos: para a entrada do erro no controlador *fuzzy* $K_e = 2.9520$; para a entrada da derivada do erro no

Tabela 4.8: Regras *fuzzy* para K_p .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	P	P	P	P	P	P	P
	NM	Z	P	P	P	P	P	Z
	NS	N	Z	Z	P	Z	Z	N
	ZE	N	N	N	Z	N	N	N
	PS	N	Z	Z	P	Z	Z	N
	PM	Z	P	P	P	P	P	Z
	PB	P	P	P	P	P	P	P

Fonte: Adaptado de Wahid e Hassan (2012)

Tabela 4.9: Regras *fuzzy* para K_d .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	N	N	N	N	N	N	N
	NM	P	Z	N	N	N	Z	P
	NS	P	P	Z	N	Z	P	P
	ZE	P	P	P	Z	P	P	P
	PS	P	P	Z	N	Z	P	P
	PM	P	Z	N	N	N	Z	P
	PB	N	N	N	N	N	N	N

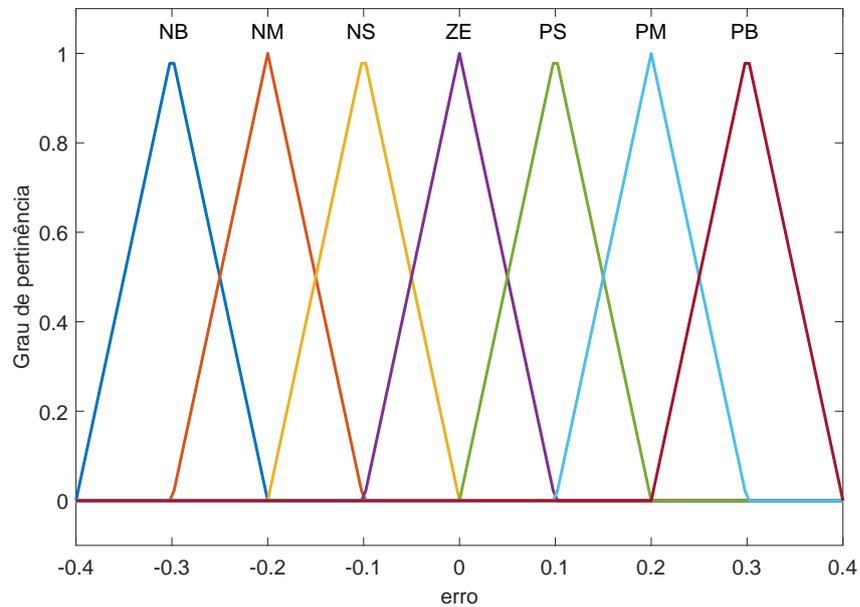
Fonte: Adaptado de Wahid e Hassan (2012)

Tabela 4.10: Regras *fuzzy* para K_i .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	P	P	P	P	P	P	P
	NM	Z	Z	P	P	P	Z	Z
	NS	N	Z	Z	P	Z	Z	N
	ZE	N	N	N	Z	N	N	N
	PS	N	Z	Z	P	Z	Z	N
	PM	Z	Z	P	P	P	Z	Z
	PB	P	P	P	P	P	P	P

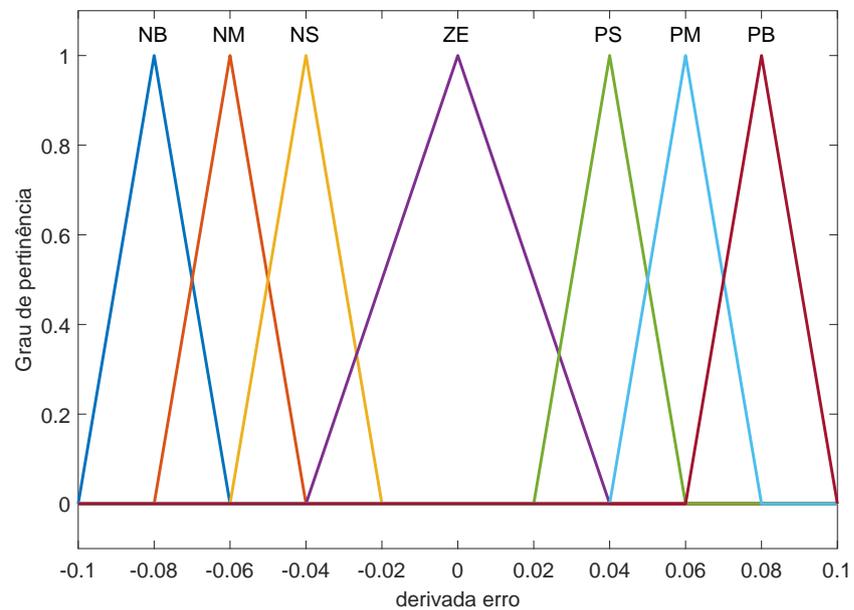
Fonte: Adaptado de Wahid e Hassan (2012)

Figura 4.17: Funções de pertinência para a entrada do erro.



Fonte: Adaptado de Wahid e Hassan (2012)

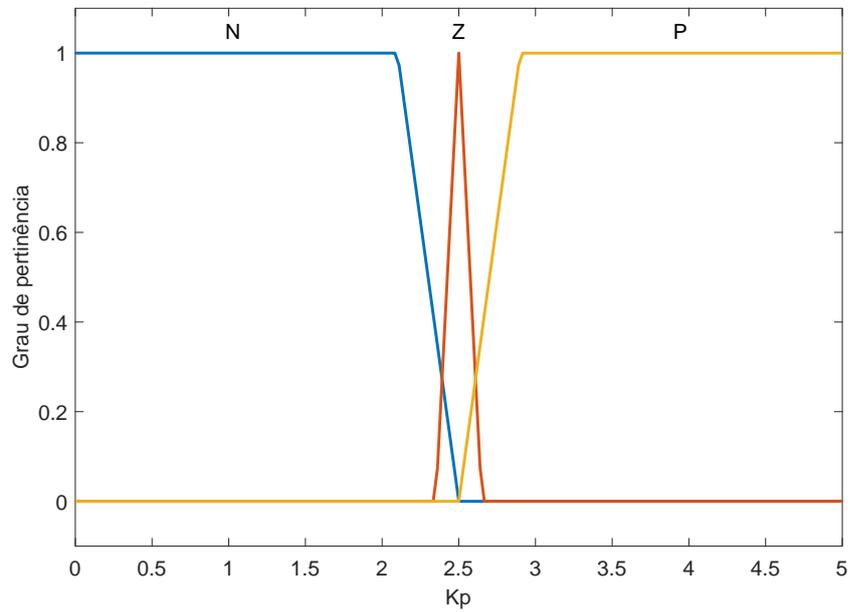
Figura 4.18: Funções de pertinência para a entrada da derivada do erro.



Fonte: Adaptado de Wahid e Hassan (2012)

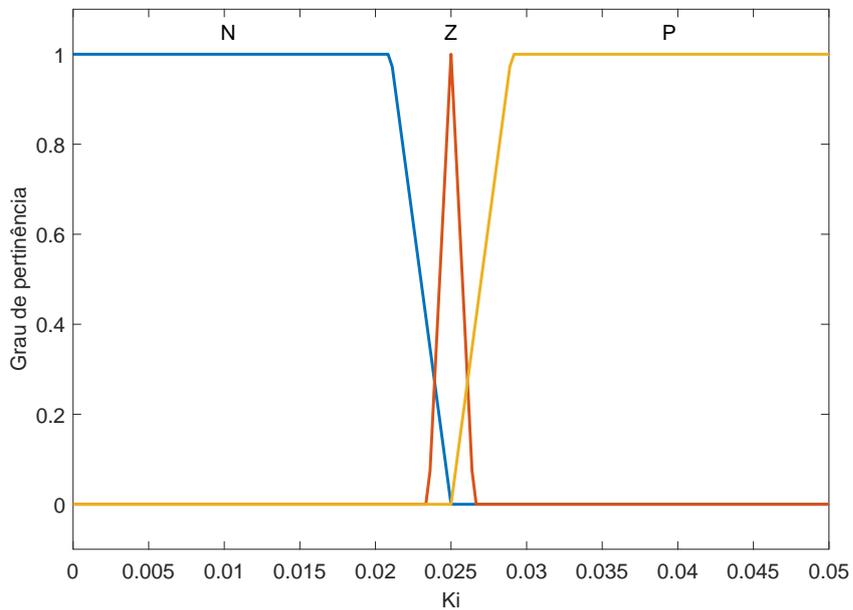
controlador *fuzzy* $K_{ec} = 0$; $K_{p,min} = 13.506$; $K_{i,min} = 8.3951$; $K_{d,min} = 0.49596$; para as saídas do controlador *fuzzy* $K_{pr} = 5.2328$; $K_{ir} = 1$; $K_{dr} = 2.9847$; $K_{rg} = 5.3161$.

Figura 4.19: Funções de pertinência para a saída K_p .



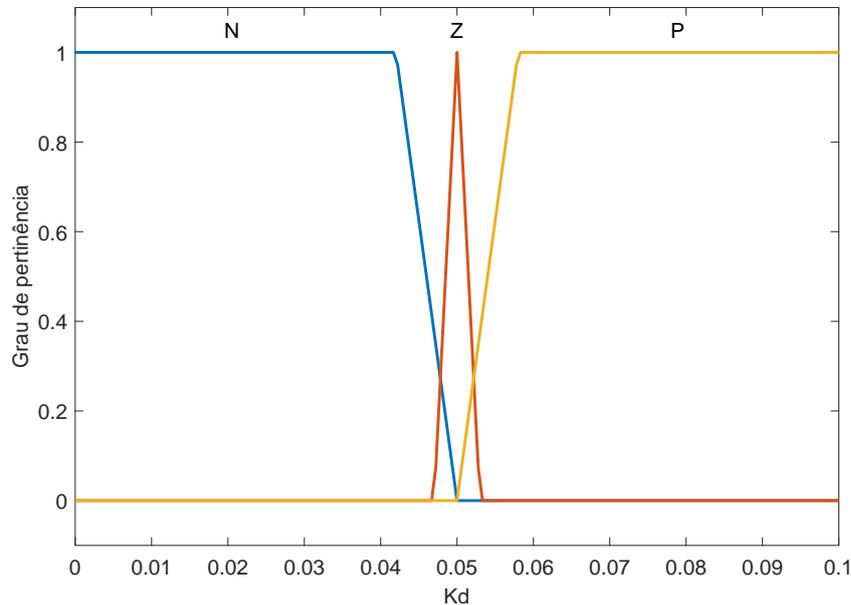
Fonte: Adaptado de Wahid e Hassan (2012)

Figura 4.20: Funções de pertinência para a saída K_i .



Fonte: Adaptado de Wahid e Hassan (2012)

A Figura 4.22 mostra as respostas ao degrau unitário com perturbação de carga da malha do ângulo de arfagem com ganhos obtidos usando PID otimizado a partir dos ganhos de Splendor et al. (2015), ISA-PID com ganhos otimizados a partir de ZN,

Figura 4.21: Funções de pertinência para a saída K_d .

Fonte: Adaptado de Wahid e Hassan (2012)

PID-EG-*Fuzzy* otimizado baseado no trabalho de Zhao et al. (1993) e PID-EG-*Fuzzy* otimizado baseado no trabalho de Wahid e Hassan (2012).

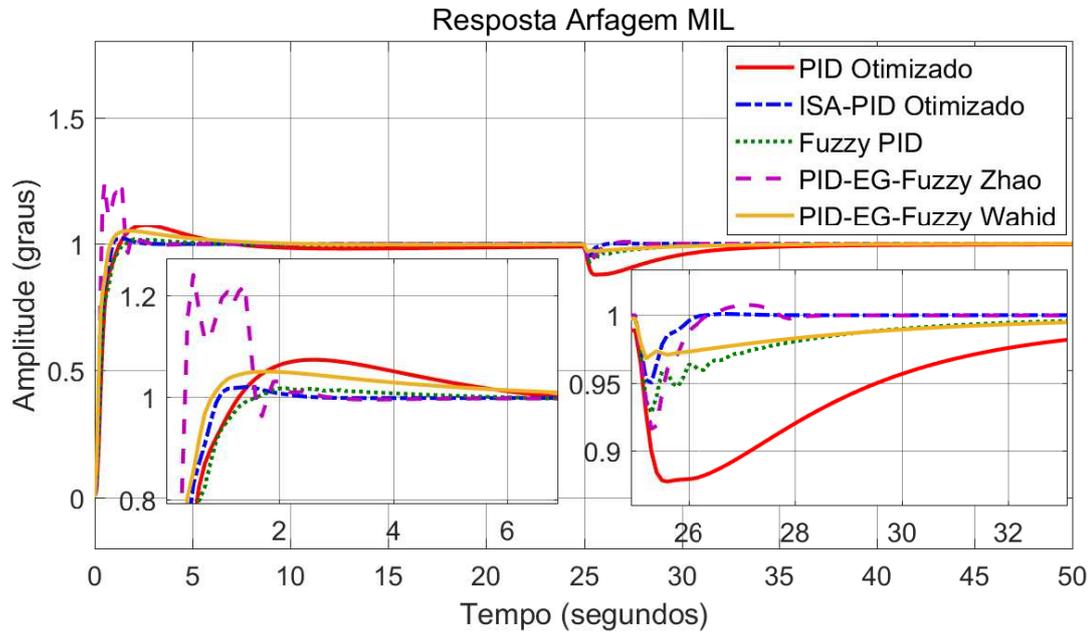
Considerando que o valor de K_{ec} convergiu para 0, usando o controlador proposto por Wahid e Hassan (2012), pode-se interpretar que, apesar do bom desempenho, o controlador *fuzzy* está utilizando somente a grandeza do erro para ajustar os ganhos do PID.

A Tabela 4.11 mostra as características obtidas com cada conjunto de ganhos testados. Esperava-se que o controle PID com escalonamento de ganhos (EG) *fuzzy* teria desempenho sempre igual ou superior ao PID otimizado, mas para o controlador sugerido em Zhao et al. (1993), isso não ocorreu.

Uma hipótese para isto é que a definição indireta do ganho K_i em Zhao et al. (1993) tenha prejudicado o desempenho pois ela limita as possibilidades. Então foi feita uma conversão para a definição independente do ganho K_i , assim como é feito com K_p e K_d e foram usadas as mesmas regras de K'_p , apresentadas na tabela 4.5.

Após a otimização foram obtidos os ganhos $K_{p,min} = 4.03$, $K_{p,max} = 13.58$, $K_{d,min} = 0.68$, $K_{d,max} = 0.99$, $K_{i,min} = 2.45$, $K_{i,max} = 4.03$. A Figura 4.23 mostra a resposta obtida do controlador proposto por Zhao et al. (1993) depois de modificado, comparado com outras respostas.

Figura 4.22: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controladores PID-EG-*Fuzzy* na plataforma MIL.



Fonte: Elaborado pelo autor

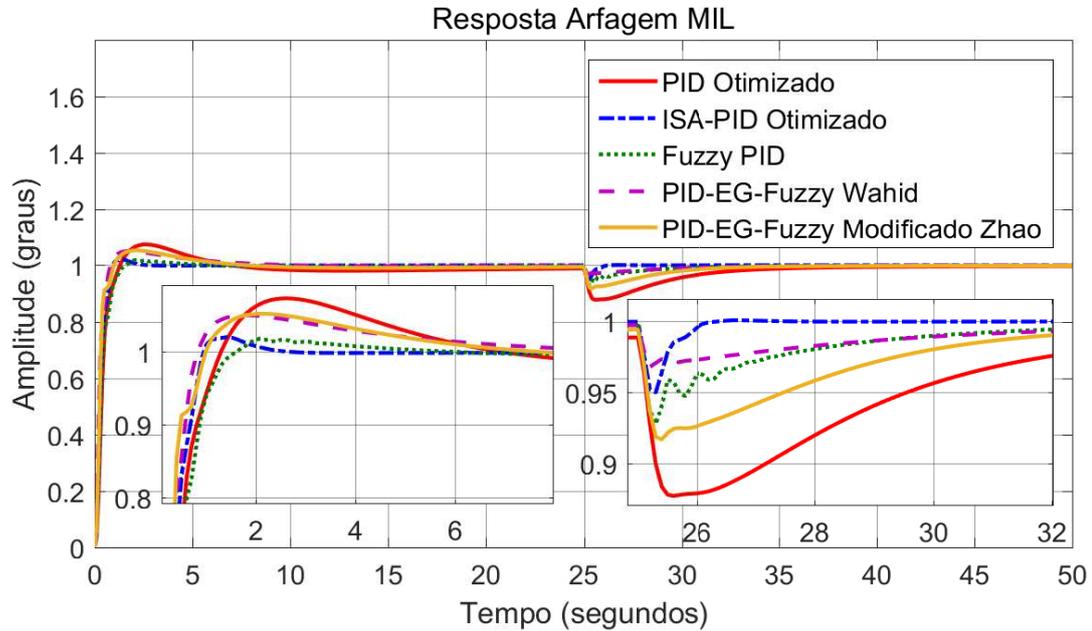
Tabela 4.11: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controladores PID-EG-*Fuzzy* na plataforma MIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID Otimizado	0.34s	2.6s	7.36%	5.5s	-12.3%	7.8s
ISA-PID Otimizado	0.29s	1.3s	2.00%	0.9s	-5.06%	0.6s
<i>Fuzzy</i> PID	0.32s	2.1s	1.83%	1.3s	-7.2%	2.9s
PID-EG- <i>Fuzzy</i> Zhao et al. (1993)	0.22s	0.5s	24.0%	2.1s	-8.4%	0.9s
PID-EG- <i>Fuzzy</i> Wahid e Hassan (2012)	0.21s	1.8s	5.03%	5.1s	-3.22%	2.4s

Fonte: Elaborado pelo autor

A Tabela 4.12 mostra as características obtidas com cada conjunto de ganhos e com controlador PID-EG-*Fuzzy* modificado, baseado em Zhao et al. (1993), obteve-se uma resposta mais esperada.

Figura 4.23: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador PID-EG-*Fuzzy* modificado, baseado em Zhao et al. (1993), na plataforma MIL.



Fonte: Elaborado pelo autor

Tabela 4.12: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador PID-EG-*Fuzzy* modificado, baseado em Zhao et al. (1993), na plataforma MIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID Otimizado	0.34s	2.6s	7.36%	5.5s	-12.3%	7.8s
ISA-PID Otimizado	0.29s	1.3s	2.00%	0.9s	-5.06%	0.6s
<i>Fuzzy</i> PID	0.32s	2.1s	1.83%	1.3s	-7.20%	2.9s
PID-EG- <i>Fuzzy</i> Wahid e Hassan (2012)	0.21s	1.8s	5.03%	5.1s	-3.22%	2.4s
PID-EG- <i>Fuzzy</i> mod Zhao et al. (1993)	0.25s	2.1s	5.24%	5.0s	-8.35%	5.0s

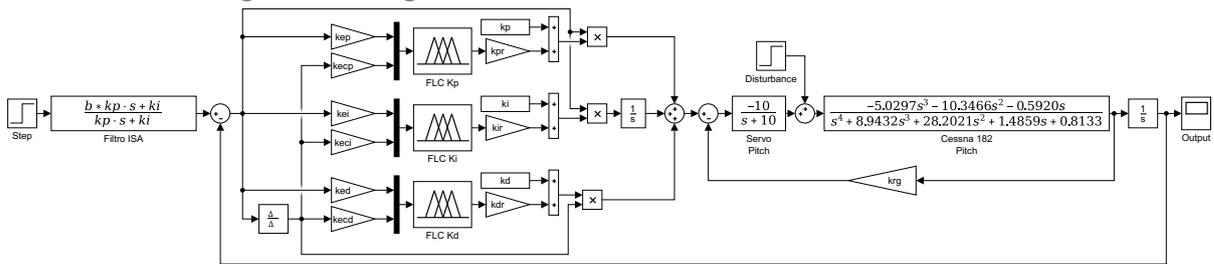
Fonte: Elaborado pelo autor

4.5 Controlador PID-EG-Fuzzy integrado com estrutura ISA-PID

Com base nos dados e resultados anteriores, surgiu a ideia da possibilidade obter resultados também satisfatórios utilizando um controlador *fuzzy* para cada um dos ganhos do controlador PID. Com isso seria possível usar intervalos diferentes para as entradas

dos controladores *fuzzy*. Assim foram realizados experimentos com um controlador PID-EG-*Fuzzy* com estrutura ISA-PID, ressaltando a utilização integrada do filtro no sinal de referência. As regras utilizadas são as mesmas propostas em Wahid e Hassan (2012). A Figura 4.24 mostra o modelo do controlador ISA-PID-EG-*Fuzzy*.

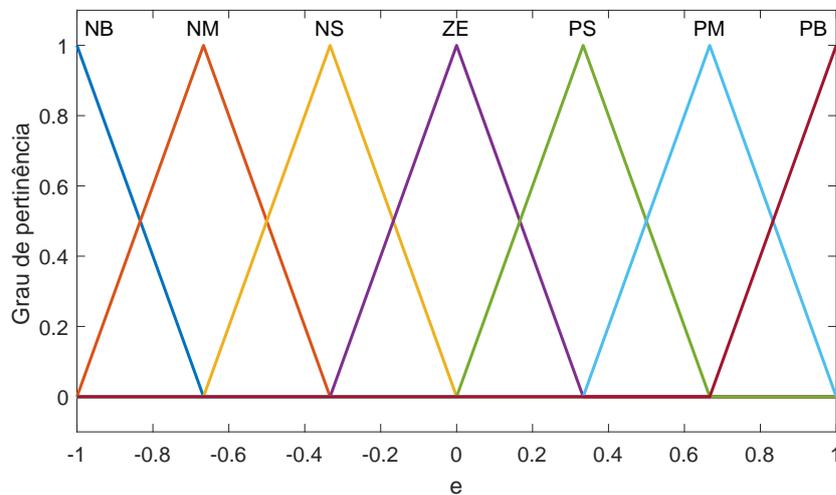
Figura 4.24: Modelo do controlador ISA-PID-EG-*Fuzzy* para a malha de controle do ângulo de arfagem.



Fonte: Elaborado pelo autor

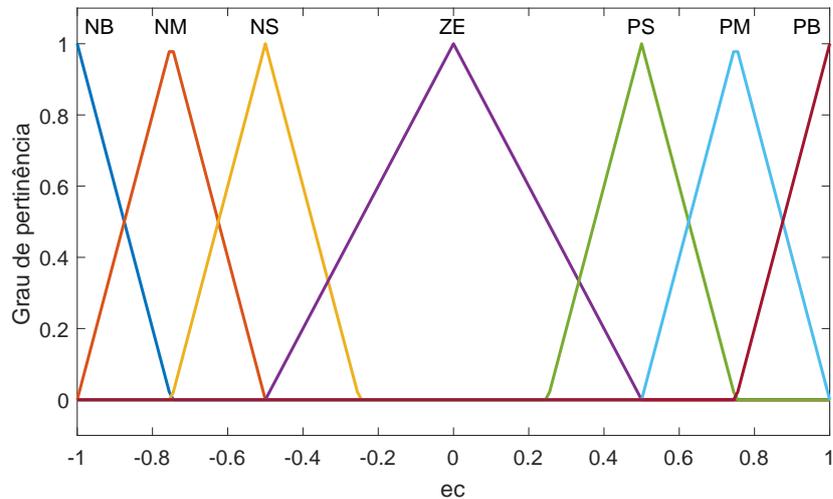
As Figuras 4.25 e 4.26 mostram respectivamente as entradas do erro e da derivada do erro nos controladores *fuzzy*, usados na determinação dos três ganhos, K_p , K_i e K_d .

Figura 4.25: Funções de pertinência para a entrada do erro.

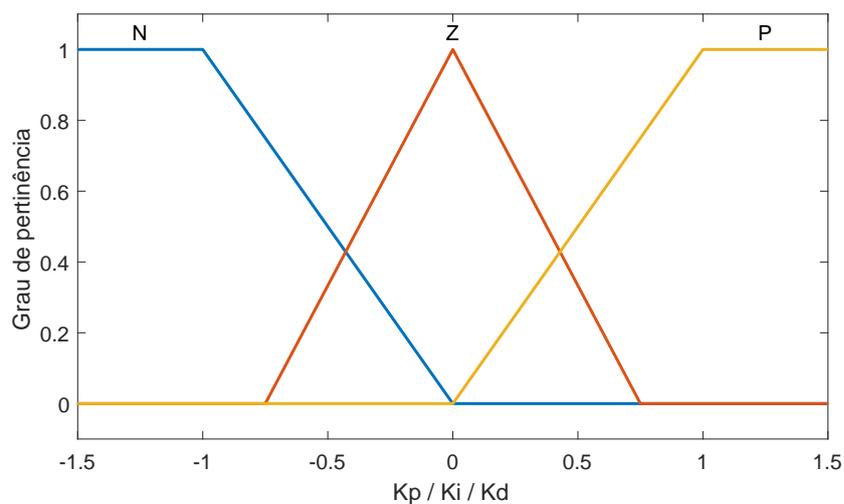


Fonte: Adaptado de Wahid e Hassan (2012)

Para esse controlador, os valores médios de K_p , K_i e K_d são somados com as saídas amplificadas dos controladores *fuzzy* para definir os ganhos que serão utilizados no PID. As funções de pertinência das saídas são apresentadas na Figura 4.27.

Figura 4.26: Funções de pertinência para a entrada da derivada do erro.

Fonte: Adaptado de Wahid e Hassan (2012)

Figura 4.27: Funções de pertinência para a saída K_p , K_i e K_d .

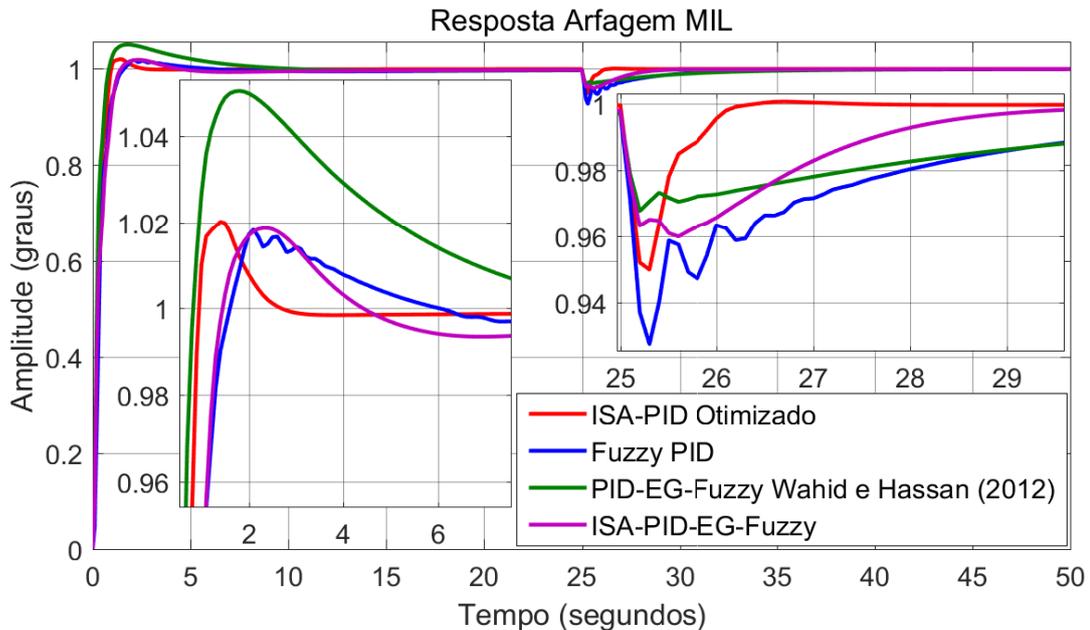
Fonte: Adaptado de Wahid e Hassan (2012)

4.5.1 Teste do Controlador ISA-PID-EG-Fuzzy Usando Plataforma MIL

Utilizando o *Simulink Design Optimization* foram obtidos os seguintes ganhos para o controlador do ângulo de arfagem na plataforma MIL: $b = 0.7747$, $K_p = 14.793$ (médio), $K_{pr} = 10.066$ ($K_{p,min} = K_p - K_{pr}$, $K_{p,max} = K_p + K_{pr}$), $K_{ep} = 3.8821$, $K_{ecp} = 1.1203$, $K_i = 9.7919$ (médio), $K_{ir} = 8.2975$ ($K_{i,min} = K_i - K_{ir}$, $K_{i,max} = K_i + K_{ir}$), $K_{ei} = 7$, $K_{eci} = 1.003$, $K_d = 2.5409$ (médio), $K_{dr} = 1$ ($K_{d,min} = K_d - K_{dr}$, $K_{d,max} = K_d + K_{dr}$), $K_{ed} = 1.6358$, $K_{ecd} = 2.9783$, $K_{rg} = 3$.

A Figura 4.28 mostra a resposta do controlador *ISA-PID-EG-Fuzzy* comparada com os outros controladores.

Figura 4.28: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma MIL.



Fonte: Elaborado pelo autor

A Tabela 4.13 mostra os dados obtidos da resposta do controlador *ISA-PID-EG-Fuzzy* comparado com os outros controladores. Não foi obtido o melhor valor em nenhum dos requisitos mas se obteve bons valores em todos os requisitos ao mesmo tempo que se obteve uma resposta suave, como pode ser visto na Figura 4.28.

Tabela 4.13: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma MIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
<i>ISA-PID Otimizado</i>	0.29s	1.3s	2.00%	0.9s	-5.06%	0.6s
<i>Fuzzy PID</i>	0.32s	2.1s	1.83%	1.3s	-7.20%	2.9s
PID-EG- <i>Fuzzy</i> Wahid e Hassan (2012)	0.21s	1.8s	5.03%	5.1s	-3.22%	2.4s
<i>ISA-PID-EG-Fuzzy</i>	0.26s	2.4s	1.87%	1.3s	-4.01%	1.9s

Fonte: Elaborado pelo autor

4.5.2 Teste do Controlador ISA-PID-EG-Fuzzy Usando Plataforma SIL

Na seção anterior o controlador *ISA-PID-EG-Fuzzy*, desenvolvido com base no proposto por Wahid e Hassan (2012), foi testado na plataforma MIL. Nesta seção esse controlador é testado na plataforma SIL. Para isso é necessário fazer uma conversão dos ganhos de uma plataforma para outra. Considerando que na plataforma MIL o período crítico é $P_{cr} = 0.69$, o ganho crítico é $K_{cr} = 26$ e os ganhos obtidos na plataforma MIL são: $b = 0.7747$, $K_p = 14.793$ (médio), $K_{pr} = 10.066$ ($K_{p,min} = K_p - K_{pr}$, $K_{p,max} = K_p + K_{pr}$), $K_{ep} = 3.8821$, $K_{ecp} = 1.1203$, $K_i = 9.7919$ (médio), $K_{ir} = 8.2975$ ($K_{i,min} = K_i - K_{ir}$, $K_{i,max} = K_i + K_{ir}$), $K_{ei} = 7$, $K_{eci} = 1.003$, $K_d = 2.5409$ (médio), $K_{dr} = 1$ ($K_{d,min} = K_d - K_{dr}$, $K_{d,max} = K_d + K_{dr}$), $K_{ed} = 1.6358$, $K_{ecd} = 2.9783$, $K_{rg} = 3$. Então, convertendo os ganhos para a plataforma SIL, onde $P_{cr} = 0.56$ e $K_{cr} = 0.2766$: $K_p = 0.15737$, $K_{pr} = 0.10709$, $K_i = 0.12835$, $K_{ir} = 0.10876$, $K_d = 0.021938$, $K_{dr} = 0.0086339$, $K_{rg} = 0.025902$. Para os outros ganhos são utilizados os mesmos valores.

A resposta inicial obtida mostrou-se ser muito oscilatória. Com isso foram feitas tentativas de otimização, utilizando um intervalo de 0 a 130% para todos os ganhos. Mas, mesmo após várias tentativas de otimização, não foi possível obter uma resposta satisfatória utilizando o controlador proposto em Wahid e Hassan (2012). Então, várias modificações foram experimentadas e a resposta satisfatória encontrada foi com um controlador baseado em Zhao et al. (1993) com a modificação para determinação direta do ganho K_i .

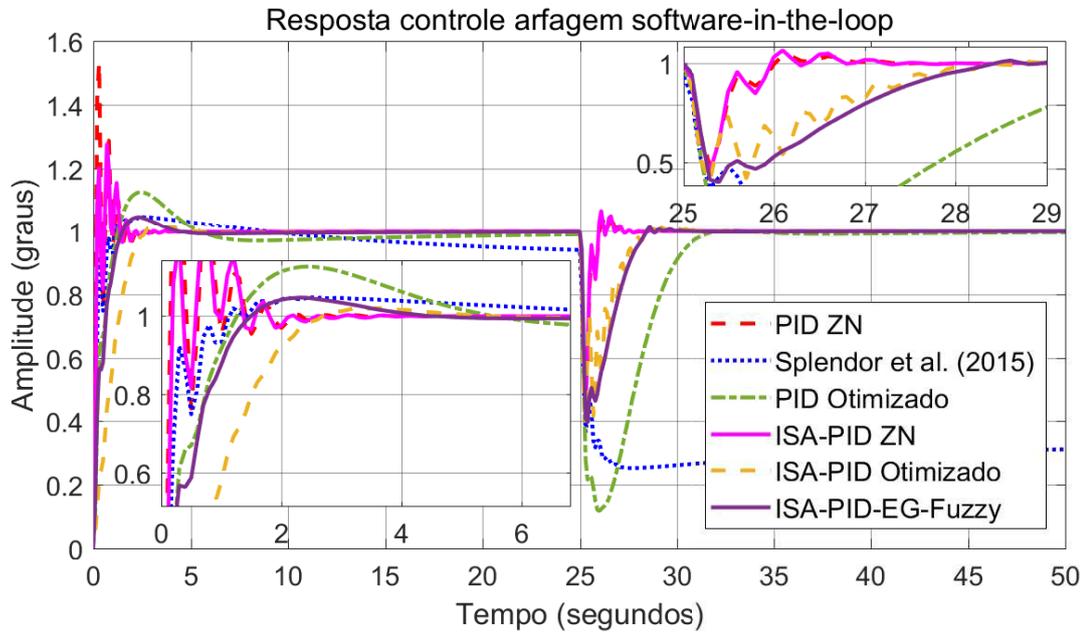
Os ganhos obtidos foram $b = 0.6000$, $K_e = 0.6625$, $K_{ec} = 0.2270$, $K_p = 0.09722$, $K_{pr} = 0.04154$, $K_i = 0.09999$, $K_{ir} = 0.01787$, $K_d = 0.001828$, $K_{dr} = 0.0001814$, $K_{rg} = 0.008086$.

Foram utilizadas as mesmas regras e funções de pertinência para K_p e K_i , o que tornou o controlador mais simples. A resposta obtida é apresentada na Figura 4.29 para comparação e os dados apresentados na Tabela 4.14. Novamente, é possível observar que, apesar do controlador *ISA-PID-EG-Fuzzy* não ser o melhor em nenhuma característica, apresentou bons valores em todas elas, ao mesmo tempo em que se tem uma resposta suave.

4.6 Considerações

O uso do filtro suaviza a função degrau da referência. Um controlador PID com ganhos fixos pode ser otimizado para um melhor desempenho na resposta transitória (degrau) ou perturbação (rastreamento de referência). Portanto, o bom desempenho do controlador

Figura 4.29: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma SIL.



Fonte: Elaborado pelo autor

Tabela 4.14: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma SIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
PID ZN	0.10s	0.3s	53.3%	2.1s	51.9%	1.76s
Splendor et al. (2015)	0.16s	2.5s	4.62%	*	*	*
PID Otimizado	0.25s	2.4s	12.5%	15.2s	88.3%	6.14s
<i>ISA-PID ZN</i>	0.12s	0.7s	27.6%	2.0s	55.0%	1.70s
<i>ISA-PID Otimizado</i>	0.83s	3.3s	1.87%	2.4s	60.0%	3.25s
<i>ISA-PID-EG-Fuzzy</i>	0.25s	2.3s	4.59%	3.5s	60.3%	3.26s

Fonte: Elaborado pelo autor

ISA-PID é devido a um controle PID otimizado para o rastreamento de referência combinado com o filtro que suaviza a função degrau tornando a referência rastreável.

Dois problemas são encontrados com essa abordagem. Para grandes mudanças de referência, ou seja, quando o erro for grande, ocorrerá a saturação do atuador. No caso de aeronaves isso ocorre quando as superfícies de controle atingem seus ângulos limites (Franklin et al., 1994). Com isso o rastreamento da referência gerado pelo filtro é perdido e, conseqüentemente, a resposta transitória será ruim.

Outro problema é que, situações em que controle em cascata é aplicado, usar o filtro na malha interna, necessariamente piorará o desempenho da malha externa. Isso porque a malha externa é que define a referência da interna. Então basta que a malha externa passe uma referência rastreável para a malha interna. O filtro, nesse caso, somente atrasará essa passagem de referência.

Em um controlador PID com ganhos variáveis, os ganhos podem ser escalonados de acordo com a situação. Pode-se definir um conjunto de ganhos para quando o erro for grande, para uma boa resposta transitória, e outro conjunto para quando o erro for pequeno para boa compensação a perturbação. Porém, um novo desafio surge para definir o que é considerado grande e pequeno em cada sistema, algo que pode ser melhor dimensionado por um especialista. Outro fator que pode ser considerado, da mesma forma, é a variação do erro.

Otimizações nos controladores feitas na plataforma MIL podem se tornar adaptadas aos modelos que não são totalmente precisos ao representar um sistema real. Com isso as melhorias obtidas em um não necessariamente se refletem em melhorias no outro, o que limita os benefícios desse tipo de otimização.

Controlador ISA-PID-EG-Fuzzy Proposto: Projeto, Resultados e Análises

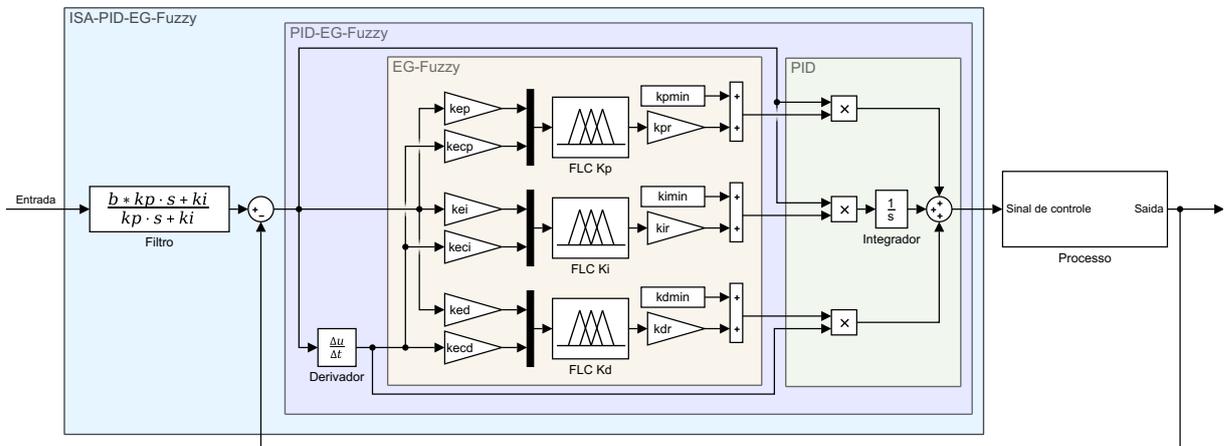
Com base no estudo e análises apresentados no Capítulo 4, é definido neste capítulo o projeto do controlador proposto. Ele consiste em um controle com estrutura *ISA-PID*, que possui um filtro no sinal de referência, integrado com o escalonamento dos ganhos (EG) utilizando controladores *fuzzy*. Assim o controlador é denominado como *ISA-PID-EG-Fuzzy* e o detalhamento deste controlador é feito no decorrer deste capítulo. Também é feita a avaliação e comparação, por meio de simulações, do piloto automático que utiliza o controlador proposto neste trabalho com outros dois. O primeiro é o proposto por Splendor et al. (2015) e o segundo é o piloto automático descrito no Capítulo 3, que utiliza controladores PID otimizados.

5.1 Controlador ISA-PID-EG-Fuzzy Proposto

A Figura 5.1 mostra o projeto do sistema de controle *ISA-PID-EG-Fuzzy* proposto. A ideia é integrar o escalonamento de ganhos, utilizando lógica *fuzzy*, do controlador PID a estrutura *ISA-PID*, que utiliza um filtro no sinal de referência. A função de transferência do filtro F é dado pela equação 5.1.

$$F(s) = \frac{bK_p s + K_i}{K_p s + K_i} \quad (5.1)$$

Figura 5.1: Projeto do sistema de controle *ISA-PID-EG-Fuzzy*.



Fonte: Elaborado pelo autor

onde K_p e K_i são os ganhos proporcional e integrativo do controlador PID e b é um número real entre 0 e 1. A expressão equivalente em tempo discreto do filtro F é dada por

$$r_f(k) = br(k) + \frac{K_i}{K_p} r_i(k) \quad (5.2)$$

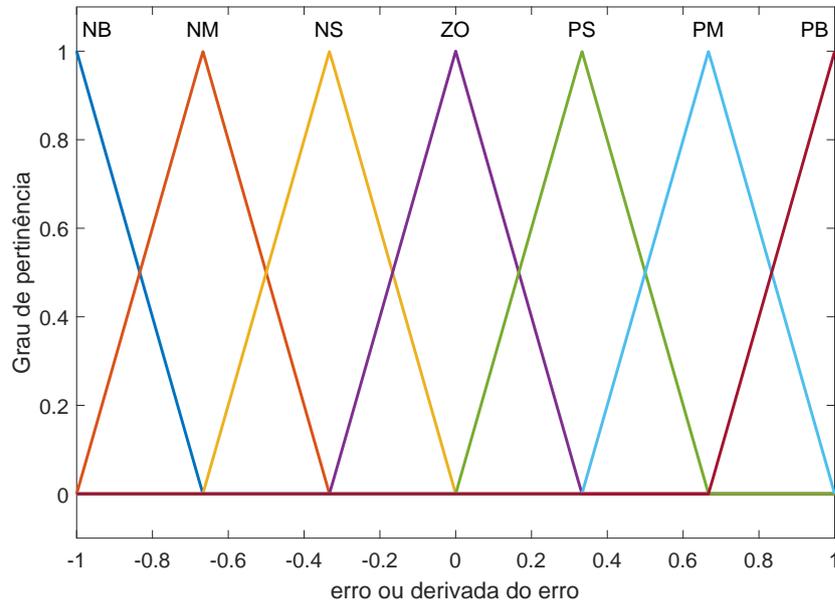
onde $r_f(k)$ é a referência filtrada no instante k e

$$r_i(k) = \begin{cases} 0 & \text{se } k = 0 \\ r_i(k-1) + T_s((1-b)r(k) - \frac{K_i}{K_p} r_i(k-1)) & \text{se } k > 0 \end{cases} \quad (5.3)$$

onde $r(k)$ é a referência de entrada no instante k e T_s é intervalo de tempo.

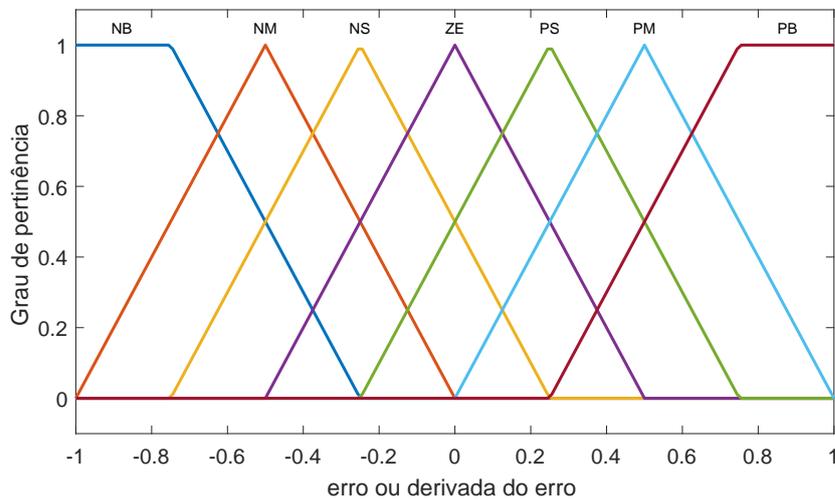
Os ganhos K_p , K_i e K_d do controlador PID são definidos nos intervalos $[K_{p,min}, K_{p,max}]$, $[K_{i,min}, K_{i,max}]$ e $[K_{d,min}, K_{d,max}]$ respectivamente. O escalonamento de ganhos *fuzzy*, identificado como *EG-Fuzzy* na Figura 5.1, define os ganhos do *PID* entre os valores mínimos e máximos baseados no erro e na derivada do erro. As entradas de cada *FLC* são normalizadas entre -1 e 1, conforme a Figura 5.3, e a saída entre 0 e 1, conforme a Figura 5.4. Para o *FLC Kp*, o erro é considerado *negative big (NB)* quando este é menor que $-1/kep$ e *positive big (PB)* quando este é maior que $-1/kep$. De forma análoga, as mesmas considerações podem ser feitas para a derivada do erro e, conseqüentemente, para o *FLC Ki* e *FLC Kd*. Os valores k_{pmin} , k_{imin} e k_{dmin} , da Figura 5.1, correspondem a $K_{p,min}$, $K_{i,min}$ e $K_{d,min}$ respectivamente. Os valores $k_{pmin} + k_{pmax}$, $k_{imin} + k_{imax}$ e $k_{dmin} + k_{dmax}$, da Figura 5.1, correspondem a $K_{p,max}$, $K_{i,max}$ e $K_{d,max}$ respectivamente, quando as saídas dos *FLCs* são iguais a 1.

Figura 5.2: Funções de pertinência para as entradas de erro e derivada do erro na plataforma HIL antes das modificações.



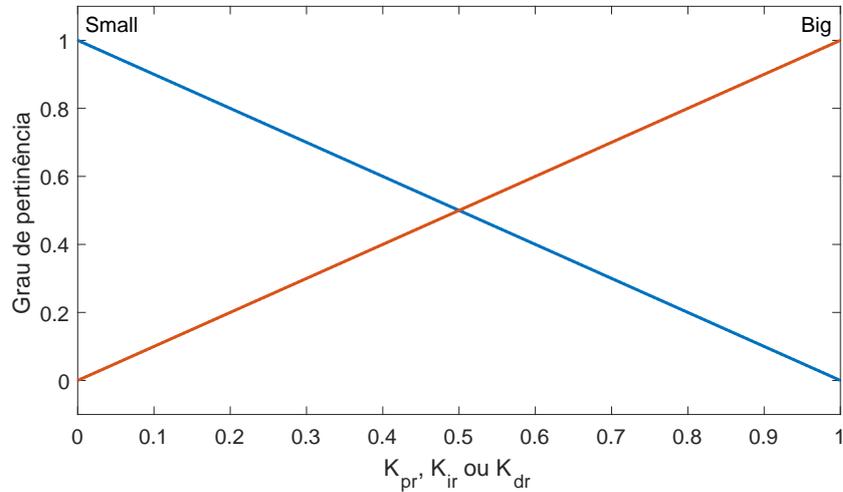
Fonte: Elaborado pelo autor

Figura 5.3: Funções de pertinência para as entradas de erro e derivada do erro na plataforma HIL depois das modificações.



Fonte: Elaborado pelo autor

O conjunto de regras usado para ajustar os valores de K_p e K_i é mostrado na Tabela 5.1, onde N significa negativo (*negative*), P positivo (*positive*), ZE zero, S pequeno (*small*), M médio (*medium*) e B grande (*big*). O conjunto de regras para K_d é mostrado na Tabela 5.2.

Figura 5.4: Funções de pertinência para as saídas K_{pr} , K_{ir} e K_{dr} na plataforma HIL.

Fonte: Elaborado pelo autor

Tabela 5.1: Regras *fuzzy* para K_p e K_i .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZE	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Fonte: Elaborado pelo autor

Tabela 5.2: Regras *fuzzy* para K_d .

		$\Delta e(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZE	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

Fonte: Elaborado pelo autor

Foi notado um chaveamento entre as funções de pertinência, mostradas na Figura 5.2, durante os testes, principalmente com o erro muito próximo de zero. Por isso elas foram

modificadas, com uma base mais larga, conforme a Figura 5.3. As funções de pertinência usadas nas saídas são mostradas na Figura 5.4. O controlador proposto foi implementado no piloto automático na plataforma HIL.

5.1.1 Implementação do Controlador ISA-PID-EG-Fuzzy na Plataforma HIL

Para o desenvolvimento do controlador *fuzzy* no Arduino foi utilizada a eFLL (*Embedded Fuzzy Logic Library*), uma biblioteca *fuzzy* para Arduino e sistemas embarcados, apresentada em Alves (2012). Essa biblioteca utiliza o método de inferência de Mamdani, que tem como base a regra de composição MAX-MIN. O método de defuzzificação utilizado é o centro de área, no universo contínuo. O código implementado encontra-se no Apêndice A. Cada conjunto de ganhos foi definido a partir dos obtidos na plataforma HIL por meio do *Simulink Design Optimization*.

Os conjuntos *fuzzy* são moldados na forma “FuzzySet(float a, float b, float c, float d)” (Alves, 2012). Por exemplo, o conjunto para a entrada do erro igual a NB (*negative big*) é dado por

```
FuzzySet* eNB = new FuzzySet(-1.00, -1.00, -0.75, -0.25);
```

Neste exemplo, para uma entrada menor ou igual a -0.75, o grau de pertinência é 1. Para uma entrada entre -0.75 e -0.25, o grau de pertinência fica entre 1 e 0, e para uma entrada maior ou igual a -0.25, o grau de pertinência é 0.

Já as regras *fuzzy* são moldadas na forma “FuzzyRule fr = new FuzzyRule(ID, antecedente, consequente)” (Alves, 2012). Por exemplo, o antecedente para erro igual a NM (*negative medium*) e derivada do erro igual a PS (*positive small*) é dado por

```
FuzzyRuleAntecedent* ifENMAndEcPS = new FuzzyRuleAntecedent();
ifENMAndEcPS->joinWithAND(eNM, ecPS);
```

O consequente, que no caso de K_p e K_i são iguais a B (*big*) e de K_d igual a S (*small*), é dado por

```
FuzzyRuleConsequent* thenKpkiBKdS = new FuzzyRuleConsequent();
thenKpkiBKdS->addOutput(Bkpk);
thenKpkiBKdS->addOutput(Skd);
```

Então, a regra é criada e adicionada ao sistema com

```
FuzzyRule* fuzzyRule12 = new FuzzyRule(12, ifENMAndEcPS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule12);
```

As demais regras foram implementadas desta mesma forma e se encontraram no apêndice A.

5.1.2 Teste do Controlador ISA-PID-EG-Fuzzy Usando Plataforma HIL

Arfagem

Os ganhos obtidos na plataforma SIL foram usados como ganhos iniciais para a otimização na plataforma HIL. A Figura 5.5 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de arfagem proposta. Após a otimização, os ganhos encontrados para a malha de controle do ângulo de arfagem foram $K_p = 0.08474$, $K_{pr} = 0.04125$, $K_i = 0.09958$, $K_{ir} = 0.01786$, $K_d = 0.001746$, $K_{dr} = 0.0001700$, $K_{rg} = 0.008583$, $K_e = 0.6251$, $K_{ec} = 0.2077$, $b = 0.5198$. O distúrbio foi provocado introduzindo uma variação de 10% do valor máximo na saída do controlador. A resposta obtida é apresentada na Figura 5.6 para comparação e os dados obtidos estão na Tabela 5.3.

Tabela 5.3: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de arfagem na plataforma HIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
Splendor et al. (2015)	0.28s	2.3s	0.00%	*	*	*
PID Otimizado	0.32s	2.3s	12.3%	10.9s	77%	5.56s
<i>ISA-PID-EG-Fuzzy</i>	0.40s	2.4s	8.31%	1.41s	69%	3.17s

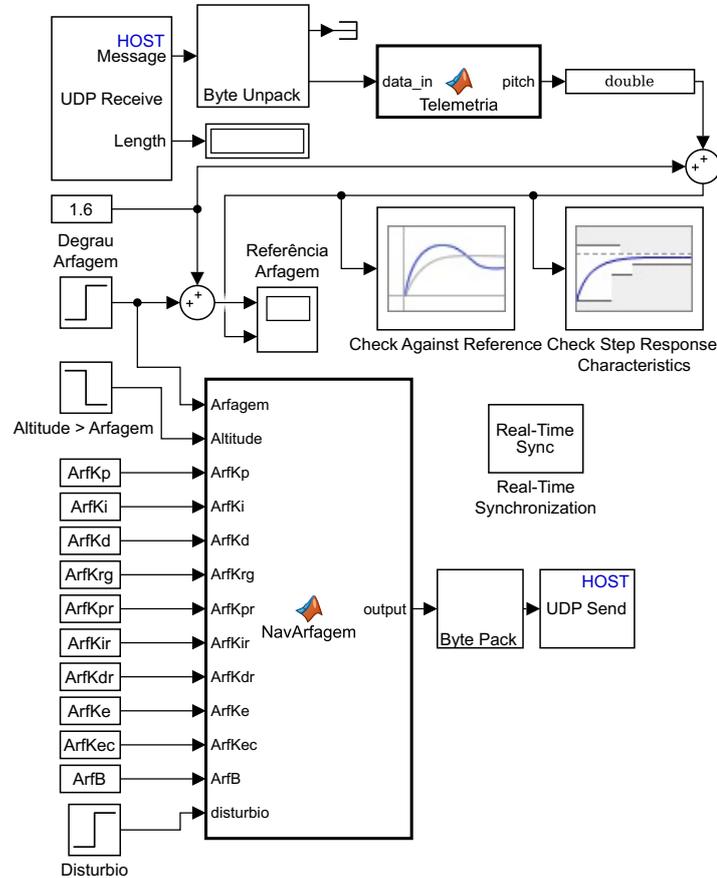
Fonte: Elaborado pelo autor

É possível observar que a resposta ao degrau no controle do ângulo de arfagem utilizando o controlador *ISA-PID-EG-Fuzzy* é suave, ou seja, pouco oscilatória, assim como com o controlador PID otimizado, e ainda conseguiu melhorar os valores de sobre-sinal e tempo de acomodação. Além disso, foi possível diminuir o pico provocado pelo distúrbio e melhorar o tempo de acomodação ao retornar para o valor de referência.

Altitude

A Figura 5.7 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle de altitude proposta. Os ganhos encontrados, por meio do

Figura 5.5: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de arfagem proposta.



Fonte: Elaborado pelo autor

Simulink Design Optimization, para a malha de controle de altitude foram $K_p = 0.00550$, $K_{pr} = 0.00143$, $K_i = 0.00276$, $K_{ir} = 0.00100$, $K_d = 0.002122$, $K_{dr} = 0.00344$, $K_{rg} = 0.008583$, $K_e = 0.8518$, $K_{ec} = 1.0383$, $b = 0.4687$. A resposta obtida é apresentada na Figura 5.8 para comparação e os dados obtidos estão na Tabela 5.4. O distúrbio foi provocado alterando a quantidade de combustível no avião de 10 lb para 380 lb, que provoca um aumento de carga e, conseqüentemente, perda de altitude.

A resposta ao degrau no controle de altitude utilizando o controlador *ISA-PID-EG-Fuzzy* conseguiu melhorar o valor de sobre-sinal e também obteve um tempo de acomodação razoável. Além disso, foi possível diminuir o pico provocado pelo distúrbio em relação ao PID otimizado e melhorar o tempo de acomodação ao retornar para o valor de referência.

Figura 5.6: Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

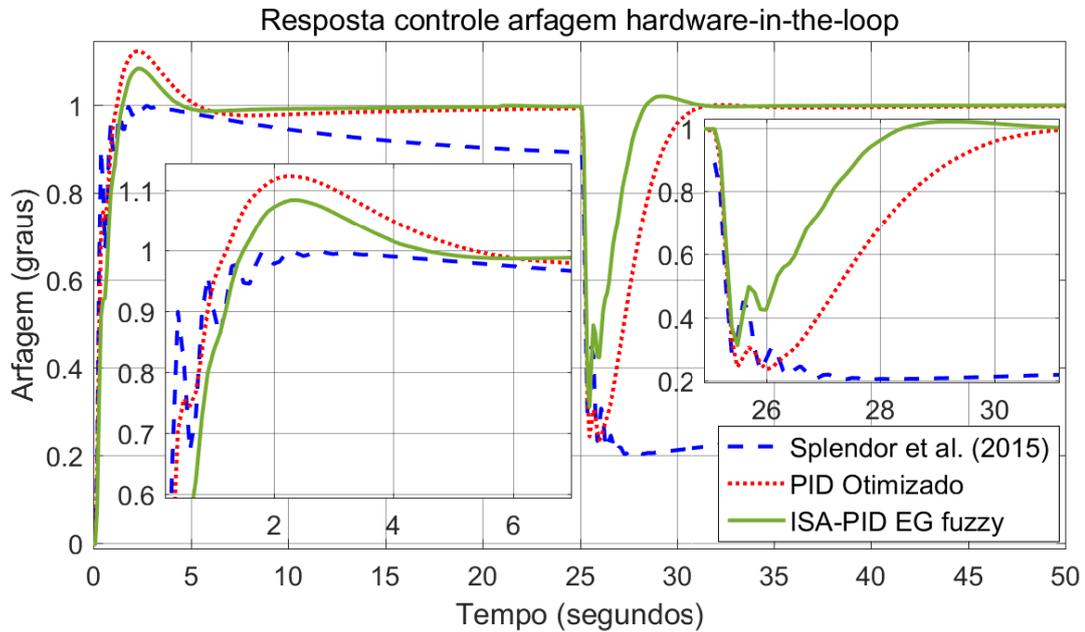


Tabela 5.4: Comparação da resposta ao degrau com distúrbio (D) da malha de controle de altitude na plataforma HIL.

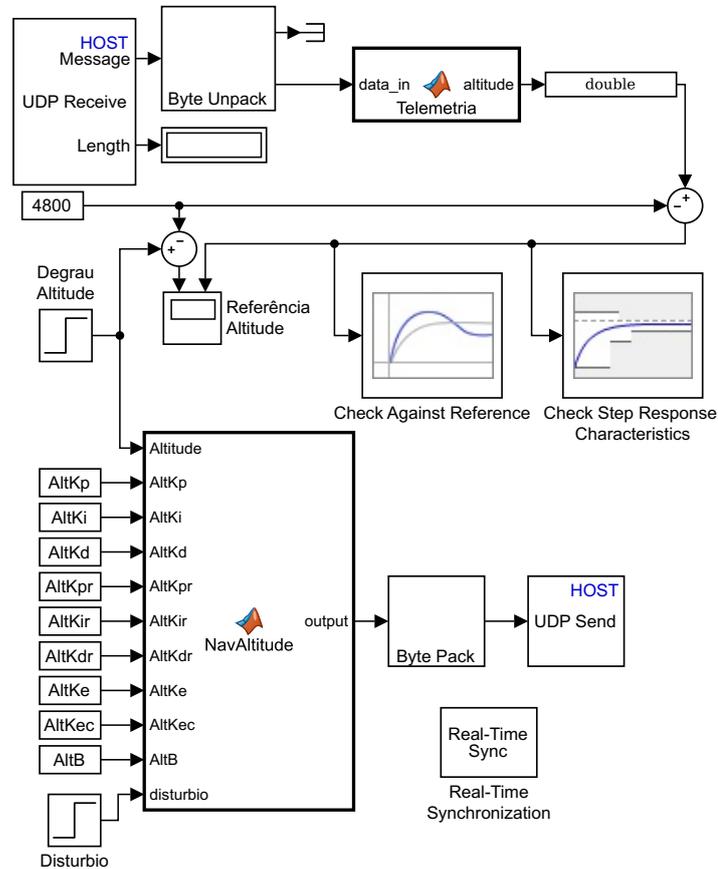
Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
Splendor et al. (2015)	0.43s	0.7s	42.43%	1.42s	61.67%	>25s
PID Otimizado	0.51s	1.4s	8.89%	8.46s	117.18%	11.86s
<i>ISA-PID-EG-Fuzzy</i>	0.59s	5.9s	0.44%	4.10s	70.37%	4.53s

Fonte: Elaborado pelo autor

Rolamento

A Figura 5.9 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de rolamento proposta. Os ganhos encontrados, por meio do *Simulink Design Optimization*, para a malha de controle do ângulo de rolamento foram $K_p = 8.04026$, $K_{pr} = 6.91750$, $K_i = 3.49074$, $K_{ir} = 1.27627$, $K_d = 0.31478$, $K_{dr} = 0.10541$, $K_{rg} = 0.000$, $K_e = 0.70133$, $K_{ec} = 1.44124$, $b = 0.59581$. A resposta obtida é apresentada na Figura 5.10 para comparação e os dados obtidos estão na Tabela 5.5. É possível observar que, com o controlador proposto, não houve sobre-sinal ($M_p = 0$) e, portanto, não é possível determinar o tempo de pico t_p . O distúrbio foi provocado introduzindo uma variação de 10% do valor máximo na saída do controlador.

Figura 5.7: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de altitude proposta.



Fonte: Elaborado pelo autor

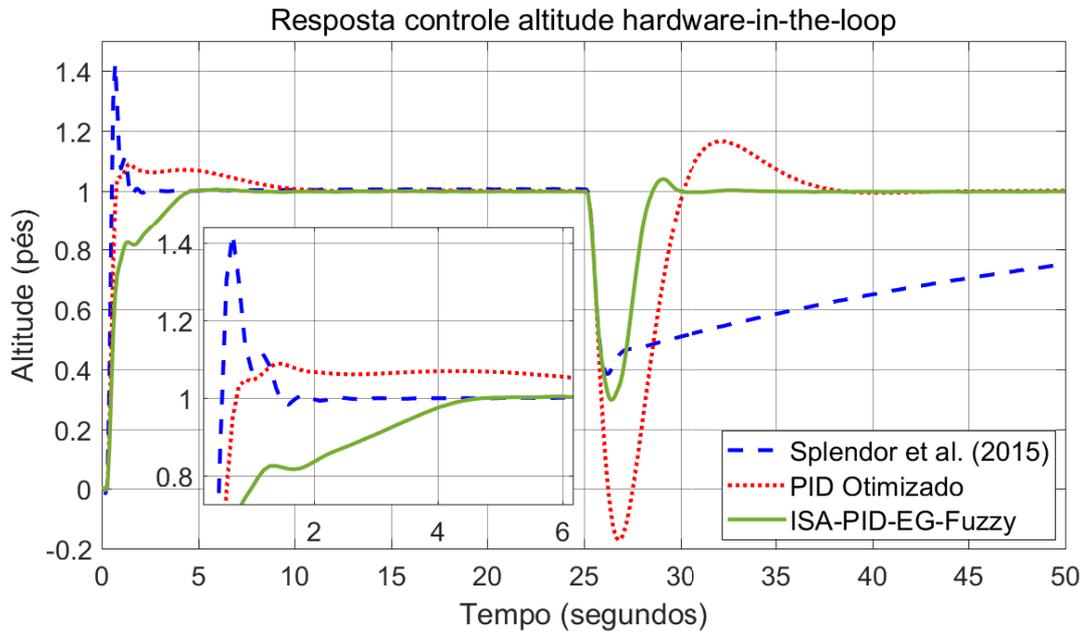
Tabela 5.5: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de rolamento na plataforma HIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
Splendor et al. (2015)	0.27s	2.9s	30.54%	9.85s	228.45%	13.7s
PID Otimizado	0.34s	3.0s	12.00%	7.66s	221.97%	11.3s
<i>ISA-PID-EG-Fuzzy</i>	0.38s	*	0.00%	5.98s	107.88%	9.3s

Fonte: Elaborado pelo autor

O controlador proposto conseguiu eliminar o sobre-sinal e também obteve o menor tempo de acomodação, na resposta ao degrau no controle do ângulo de rolamento. Além disso, foi possível diminuir o pico provocado pelo distúrbio e melhorar o tempo de acomodação ao retornar para o valor de referência.

Figura 5.8: Comparação da resposta ao degrau com distúrbio da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Guinada

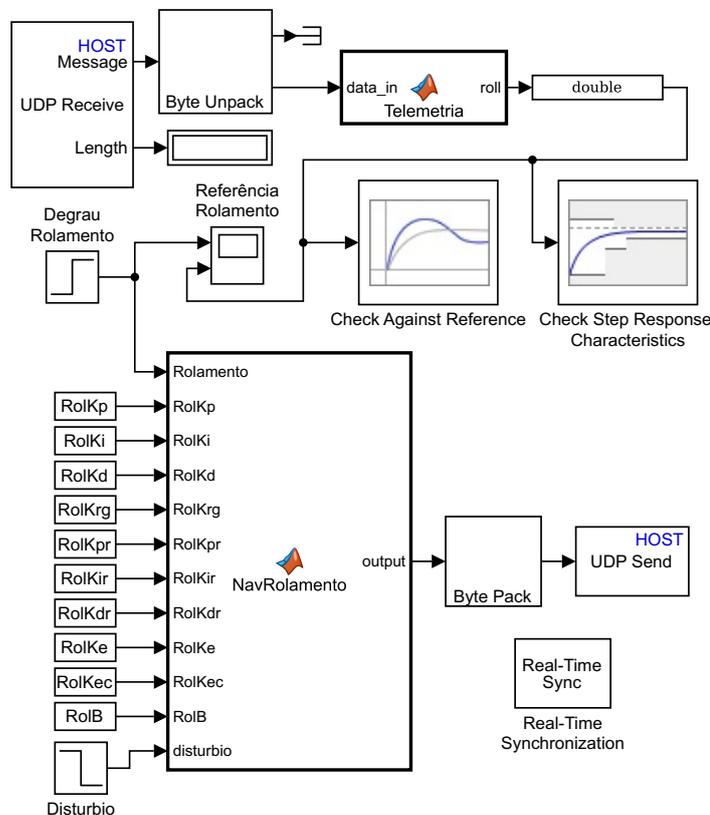
A Figura 5.11 mostra o modelo do Simulink utilizado junto com o Arduino para a otimização da malha de controle do ângulo de guinada proposta. Os ganhos encontrados, por meio do *Simulink Design Optimization*, para a malha de controle do ângulo de guinada foram $K_p = 0.29020$, $K_{pr} = 0.03028$, $K_i = 1.14190$, $K_{ir} = 0.07578$, $K_d = 0.00851$, $K_{dr} = 0.01289$, $K_{rg} = 0.13595$, $K_e = 0.97483$, $K_{ec} = 1.09403$, $b = 0.60078$. A resposta obtida é apresentada na Figura 5.12 para comparação e os dados obtidos estão na Tabela 5.6. O distúrbio foi provocado introduzindo uma variação de 10% do valor máximo na saída do controlador.

Tabela 5.6: Comparação da resposta ao degrau com distúrbio (D) da malha de controle do ângulo de guinada na plataforma HIL.

Ganhos	t_d	t_p	M_p	t_s	M_p (D)	t_s (D)
Splendor et al. (2015)	1.07s	6.1s	17.79%	9.55s	103.4%	13.45s
PID Otimizado	0.79s	5.7s	15.85%	9.07s	101.1%	12.19s
<i>ISA-PID-EG-Fuzzy</i>	0.70s	3.6s	13.55%	5.98s	91.25%	6.80s

Fonte: Elaborado pelo autor

Figura 5.9: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de rolamento proposta.



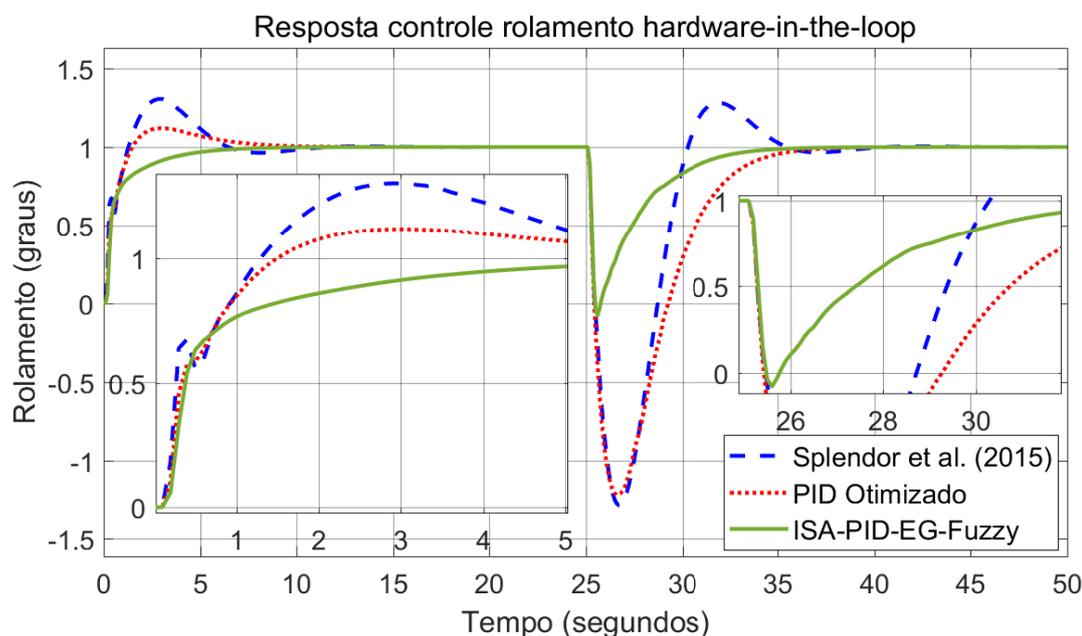
Fonte: Elaborado pelo autor

É possível observar que a resposta ao degrau no controle do ângulo de guinada utilizando o controlador *ISA-PID-EG-Fuzzy* é suave, ou seja, pouco oscilatória, e ainda conseguiu melhorar os valores em todas as características comparadas.

5.2 Casos de Simulação

Foram feitas simulações com total de 300 segundos onde a altitude de referência é mantida em 4800 pés e é usada sempre aceleração máxima, que resulta em uma velocidade aproximada de 143 mph. Inicialmente a aeronave permanece nivelada com ângulo de guinada em 100 graus em relação ao Norte. Após 100 segundos, a referência para o ângulo de guinada é alterado para 150 graus e, mais 100 segundos depois, retorna ao valor de 100 graus. Quando o erro no ângulo de guinada é maior que 10 graus, o controle do leme é desabilitado e a referência para o ângulo de rolamento muda de 0 para 20 graus. Isso faz com que a aeronave inicie a curva para a direita. Quando o erro no ângulo de guinada se

Figura 5.10: Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de rolamento com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Fonte: Elaborado pelo autor

torna menor que 10 graus, o controle do leme é novamente habilitado e a referência para o ângulo de rolamento volta para 0 graus, o que nivela a aeronave. O mesmo acontece de forma análoga na mudança do ângulo de guinada de 150 para 100 graus, na curva para a esquerda. As simulações também foram feitas com diferentes condições de voo, definindo no simulador valores para a velocidade do vento, rajadas de vento e turbulência. Os índices de desempenho EMQ, ECM e VSC obtidos nas simulações foram calculados no Matlab/Simulink.

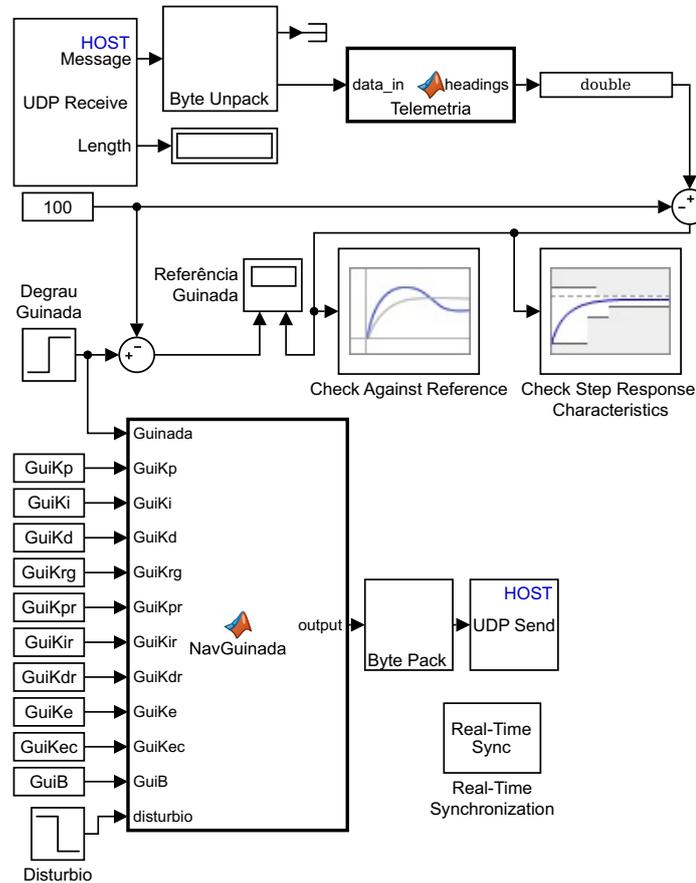
5.2.1 Simulação 1

Essa primeira simulação é feita sem a presença de vento ou turbulência. A Figura 5.13 mostra a resposta obtida pela malha de controle do ângulo de arfagem.

A Figura 5.14 mostra a resposta obtida pela malha de controle de altitude. É possível observar que, com os controles definidos em Splendor et al. (2015), apesar dos picos superiores serem menores, a altitude demora mais para retornar ao valor de referência.

A Tabela 5.7 apresenta valores obtidos no controle de altitude. Nela se destaca que o controlador definido por Splendor et al. (2015) obteve os menores picos positivos e o controlador proposto obteve os menores picos negativos. Isso ocorre pois o controlador

Figura 5.11: Modelo do Simulink usado na plataforma HIL com os blocos de otimização para a malha de controle do ângulo de guinada proposta.



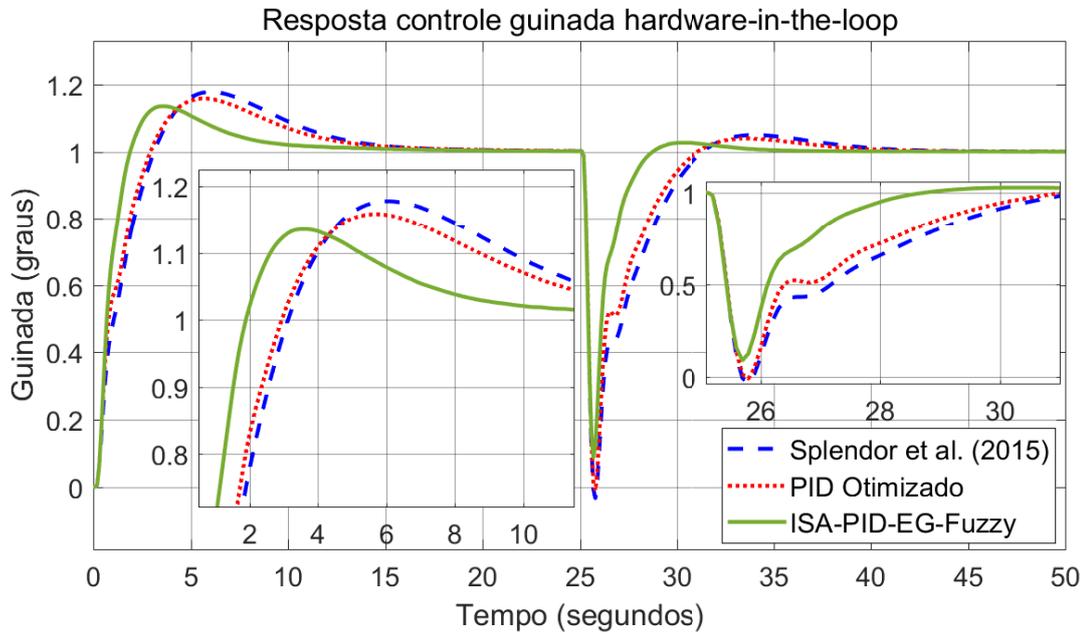
Fonte: Elaborado pelo autor

proposto reage mais rapidamente a queda de altitude provocada pela inclinação da aeronave, retornando ao valor de referência. Então, no instante que a aeronave nivela novamente, ocorre um aumento da altitude, e como com o controlador definido por Splendor et al. (2015) está ainda abaixo do valor de referência, os picos positivos acabam sendo menores.

A Figura 5.15 mostra a resposta obtida pela malha de controle do ângulo de rolamento. É possível ver a mudança da referência de 0 para 20 e de volta para 0 graus na curva para direita e de 0 para -20 e de volta para 0 graus na curva para esquerda. Também é visível que o controlador proposto praticamente eliminou o sobre-sinal da resposta nas mudanças de referência para o ângulo de rolamento.

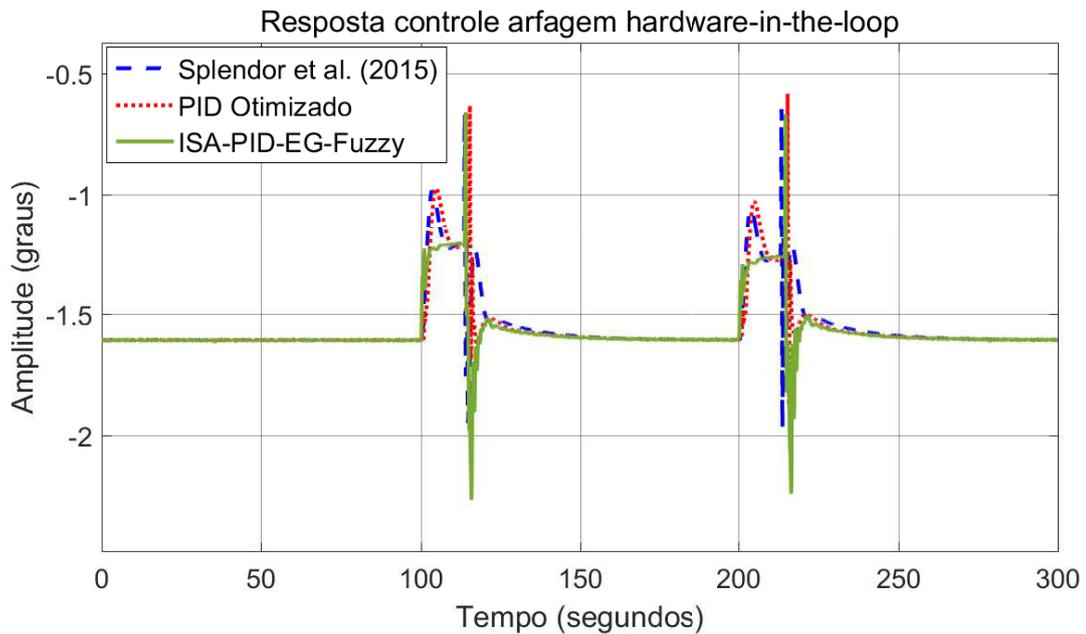
A Figura 5.16 mostra a resposta obtida pela malha de controle do ângulo de guinada. É possível ver a mudança da referência para o ângulo de guinada, de 100 para 150 na curva para direita e de 150 para 100 graus na curva para esquerda.

Figura 5.12: Comparação da resposta ao degrau com distúrbio da malha de controle do ângulo de guinada com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



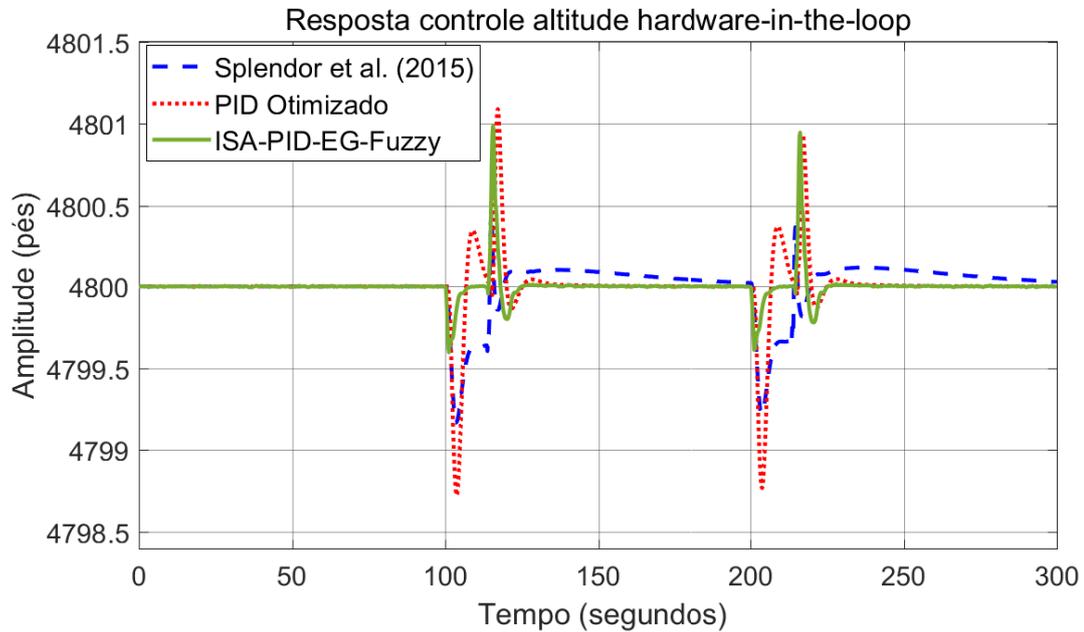
Fonte: Elaborado pelo autor

Figura 5.13: Comparação da resposta na simulação 1 da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 5.14: Comparação da resposta na simulação 1 da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Fonte: Elaborado pelo autor

Tabela 5.7: Respostas da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

Splendor et al. (2015)					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.6s	-0.8315 pés	Abaixo	3.6s	-0.7783 pés
Acima	15.0s	0.4454 pés	Acima	14.6s	0.3628 pés
PID Otimizado					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.8s	-1.2769 pés	Abaixo	3.8s	-1.2319 pés
Acima	17.2s	1.0899 pés	Acima	17.2s	0.9317 pés
<i>ISA-PID-EG-Fuzzy</i>					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	1.2s	-0.4033 pés	Abaixo	1.2s	-0.3885 pés
Acima	15.6s	0.9868 pés	Acima	16.2s	0.9463 pés

Fonte: Elaborado pelo autor

A Tabela 5.8 apresenta valores obtidos no controle latero-direcional, que consiste nos controles dos ângulos de rolamento e guinada. Nela está destacado que o controlador proposto obteve melhor resultado em 10 das 18 comparações feitas.

Figura 5.15: Comparação da resposta na simulação 1 da malha de controle do ângulo de rolamento com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

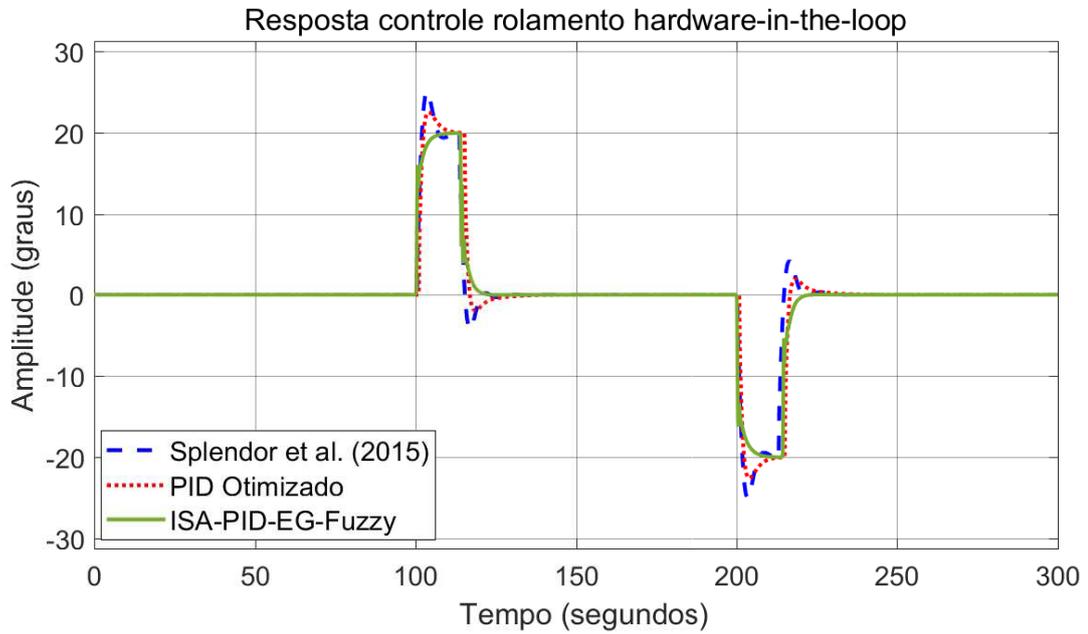


Figura 5.16: Comparação da resposta na simulação 1 da malha de controle do ângulo de guinada com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

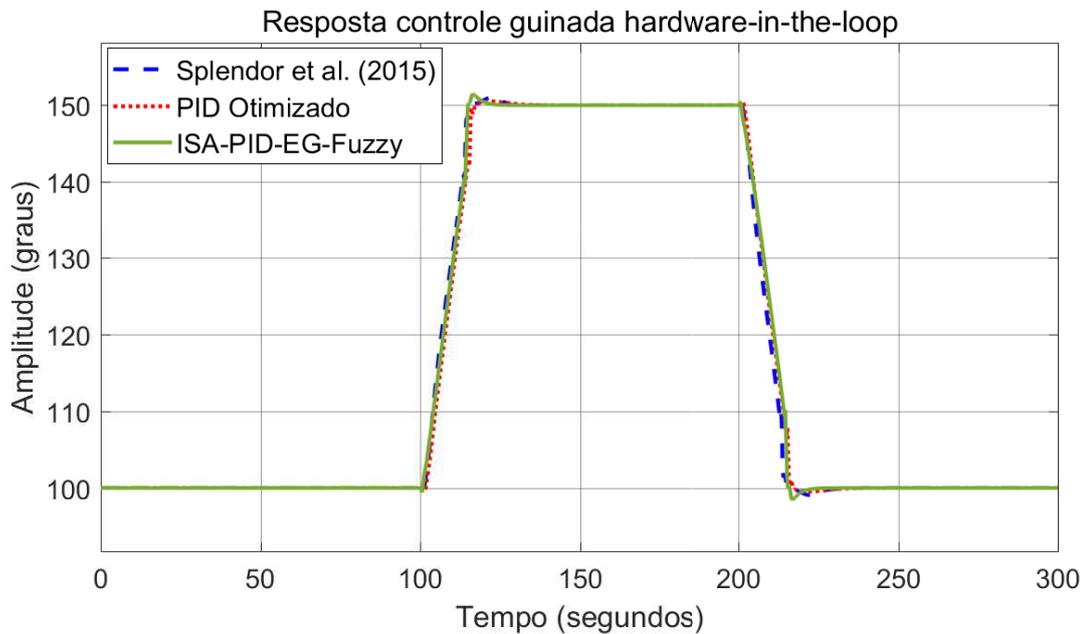


Tabela 5.8: Respostas da malha de controle latero-direcional com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL na simulação 1.

Rolamento			Guinada			
Splendor et al. (2015)						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.5s	25.11%	10.1s	22.1s	1.93%	15.27s
0	3.4s	20.88%	6.52s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.5s	24.81%	9.93s	22.2s	1.89%	15.36s
0	3.4s	21.02%	6.85s			
PID Otimizado						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	4.0s	12.83%	9.30s	22.3s	0.99%	16.69s
0	3.3s	9.81%	10.5s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	4.0s	12.99%	9.30s	22.5s	0.97%	16.57s
0	3.3s	9.93%	10.4s			
<i>ISA-PID-EG-Fuzzy</i>						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	*	0.00%	5.99s	16.4s	2.97%	17.96s
0	11.7s	0.13%	5.82s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	*	0.00%	6.67s	16.9s	3.19%	18.37s
0	11.4s	0.13%	5.50s			

Fonte: Elaborado pelo autor

Considerando as doze comparações (três índices e quatro malhas) o controlador proposto obteve os melhores valores em oito delas, como pode ser visto na Tabela 5.9.

5.2.2 Simulação 2

A simulação 2 foi feita com vento de 8 nós e rajadas de 4 nós, mas sem a presença de turbulência. Com isso o vento oscila entre 8 e 12 nós. A Figura 5.17 mostra a resposta obtida pela malha de controle do ângulo de arfagem.

A Figura 5.18 mostra a resposta obtida pela malha de controle de altitude.

Tabela 5.9: Índices de desempenho da simulação 1 na plataforma HIL .

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	9.6205×10^{-2}	1.9405×10^{-3}	1.5074×10^{-4}
PID Otimizado	4.7049×10^{-3}	1.9258×10^{-3}	1.2671×10^{-4}
<i>ISA-PID-EG-Fuzzy</i>	1.1244×10^{-3}	1.9498×10^{-3}	1.0101×10^{-4}
Altitude			
Splendor et al. (2015)	2.5449×10^{-2}	3.4547×10^0	3.9590×10^{-2}
PID Otimizado	3.9377×10^{-2}	2.4382×10^0	1.9779×10^{-2}
<i>ISA-PID-EG-Fuzzy</i>	9.7387×10^{-3}	2.4805×10^0	1.4910×10^{-2}
Rolamento			
Splendor et al. (2015)	4.7461×10^0	2.8123×10^{-3}	2.7712×10^{-3}
PID Otimizado	4.9423×10^0	2.8529×10^{-3}	2.8122×10^{-3}
<i>ISA-PID-EG-Fuzzy</i>	2.6658×10^0	6.4161×10^{-3}	6.3729×10^{-3}
Guinada			
Splendor et al. (2015)	5.8924×10^{-1}	8.9788×10^{-3}	8.9749×10^{-3}
PID Otimizado	6.0311×10^{-1}	7.4832×10^{-3}	7.4781×10^{-3}
<i>ISA-PID-EG-Fuzzy</i>	3.7000×10^{-1}	7.2001×10^{-3}	7.1943×10^{-3}

Fonte: Elaborado pelo autor

A Tabela 5.10 apresenta valores obtidos no controle de altitude. Nela está destacado que o controlador definido por Splendor et al. (2015) obteve os menores picos positivos e o controlador proposto obteve os menores picos negativos. Assim como na Simulação 1, isso ocorre pois o controlador proposto reage mais rapidamente a queda de altitude provocada pela inclinação da aeronave, retornando ao valor de referência. Então, no instante que a aeronave nivela novamente, ocorre um aumento da altitude, e como com o controlador definido por Splendor et al. (2015) está ainda abaixo do valor de referência, os picos positivos acabam sendo menores.

A Figura 5.19 mostra a resposta obtida pela malha de controle do ângulo de rolamento. Novamente é visível que o controlador proposto praticamente eliminou o sobre-sinal da resposta nas mudanças de referência para o ângulo de rolamento.

A Figura 5.20 mostra a resposta obtida pela malha de controle do ângulo de guinada.

A Tabela 5.11 apresenta valores obtidos no controle latero-direcional, que consiste nos controles dos ângulos de rolamento e guinada. Nela está destacado que o controlador proposto obteve melhor resultado em 11 das 18 comparações feitas.

Figura 5.17: Comparação da resposta na simulação 2 da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

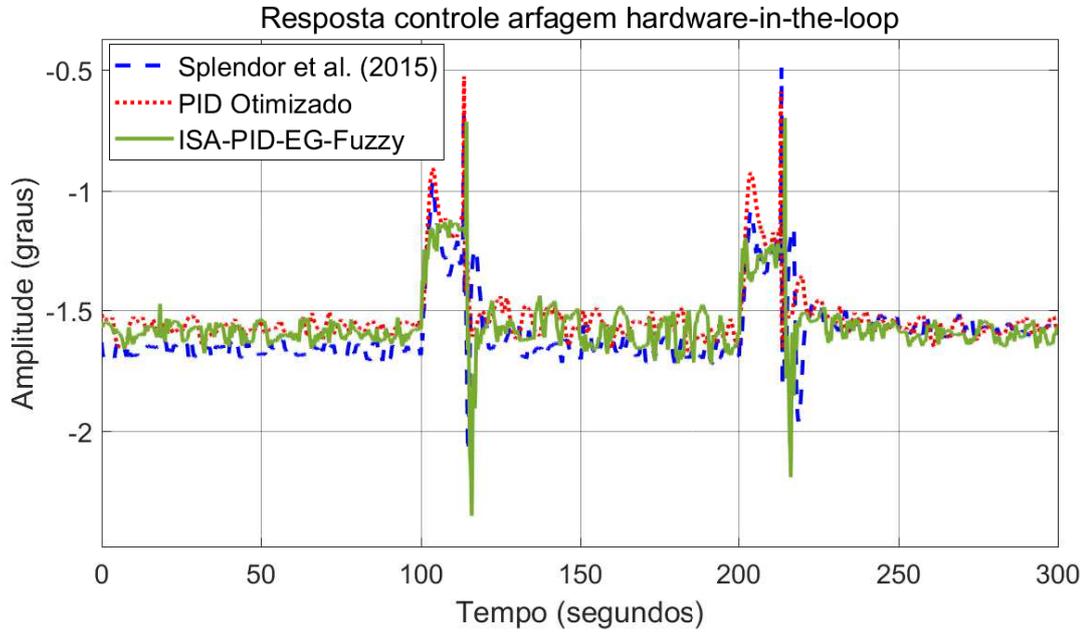


Figura 5.18: Comparação da resposta na simulação 2 da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

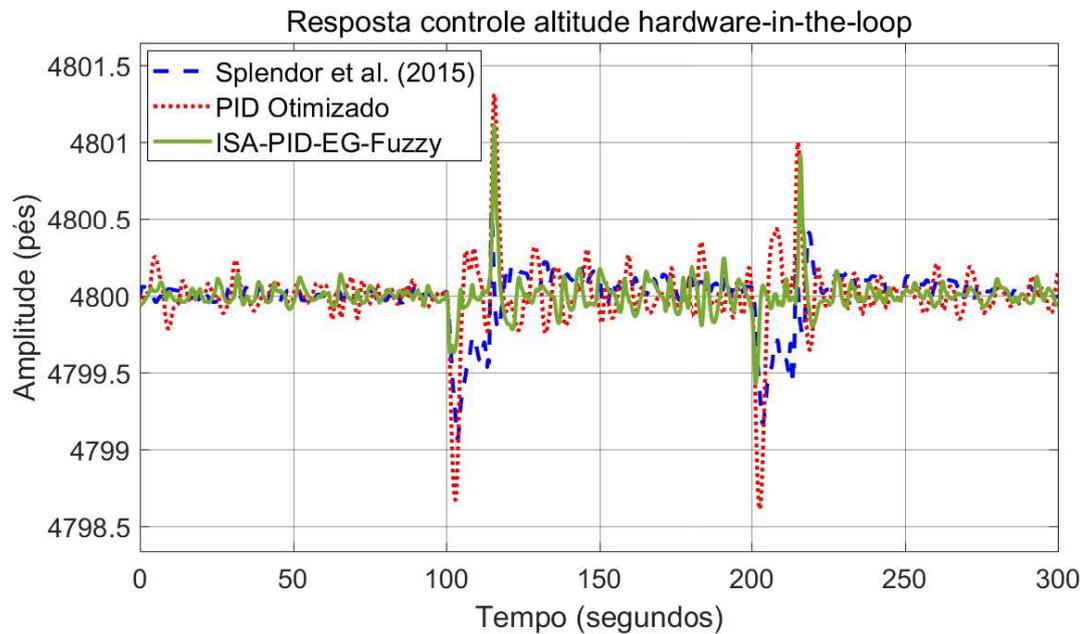
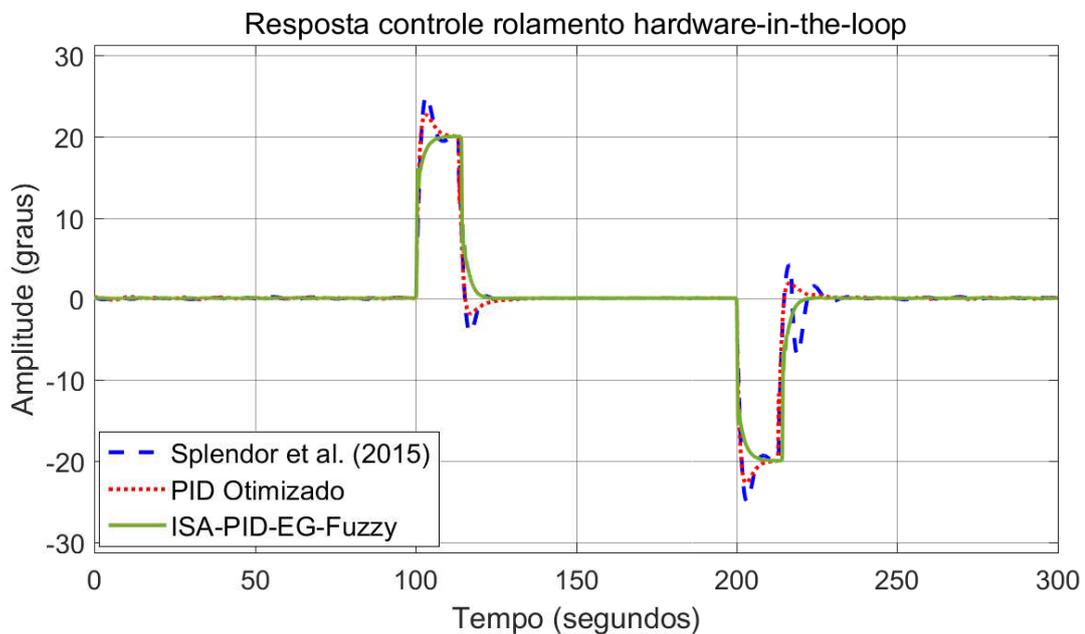


Tabela 5.10: Respostas da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL na simulação 2.

Splendor et al. (2015)					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.8s	-0.9374 pés	Abaixo	3.6s	-0.8369 pés
Acima	14.8s	0.5435 pés	Acima	18.8s	0.4072 pés
PID Otimizado					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	3.0s	-1.3333 pés	Abaixo	2.6s	-1.3917 pés
Acima	15.8s	1.3154 pés	Acima	15.2s	0.9995 pés
<i>ISA-PID-EG-Fuzzy</i>					
Direita	tp	Mp	Esquerda	tp	Mp
Abaixo	1.2s	-0.3788 pés	Abaixo	1.4s	-0.5765 pés
Acima	15.6s	1.1108 pés	Acima	16.0s	0.9150 pés

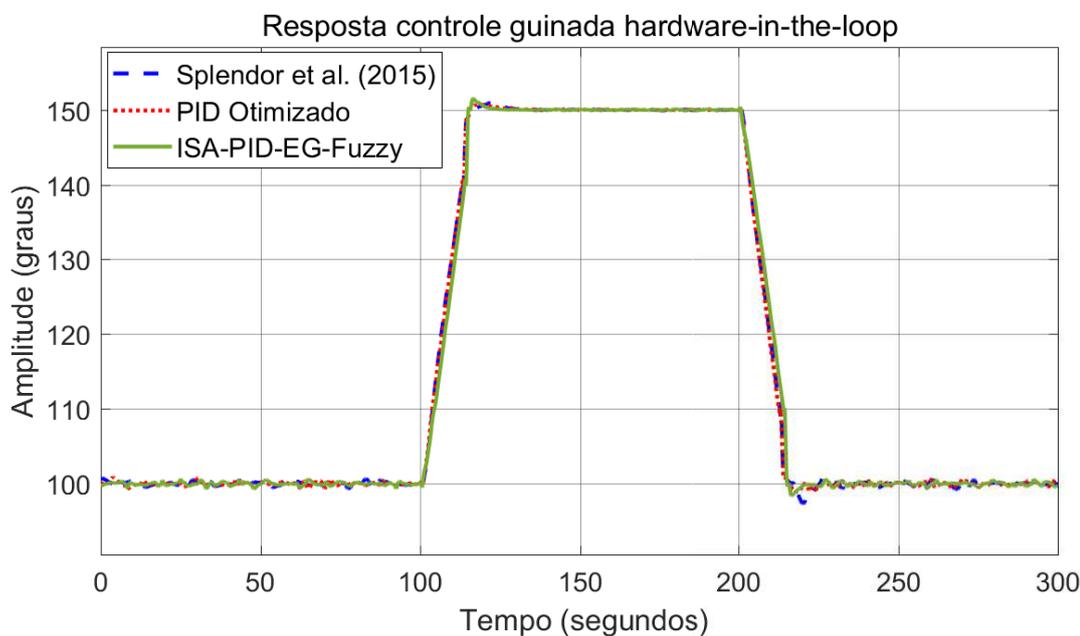
Fonte: Elaborado pelo autor

Figura 5.19: Comparação da resposta na simulação 2 da malha de controle do ângulo de rolamento com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Considerando as doze comparações de desempenho (três índices e quatro malhas) o controlador proposto obteve os melhores valores em oito delas, como pode ser visto na Tabela 5.12.

Figura 5.20: Comparação da resposta na simulação 2 da malha de controle do ângulo de guinada com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



5.2.3 Simulação 3

A simulação 3 é feita com vento de 12 nós e rajadas de 6 nós. Com isso o vento oscila entre 12 e 18 nós. O simulador de voo X-Plane permite definir um valor de turbulência entre 0 e 10 e nessa simulação é usada o valor 1. A Figura 5.21 mostra a resposta obtida pela malha de controle do ângulo de arfagem.

A Figura 5.22 mostra a resposta obtida pela malha de controle de altitude. É possível observar que os picos causados pelas perturbações são maiores que os causados pelos movimentos da aeronave.

A Figura 5.23 mostra a resposta obtida pela malha de controle do ângulo de rolamento. É visível que o controlador proposto minimizou o sobre-sinal da resposta nas mudanças de referência para o ângulo de rolamento e conseguiu manter menores amplitudes frente as oscilações, em relação aos outros dois controladores.

A Figura 5.24 mostra a resposta obtida pela malha de controle do ângulo de guinada. Neste caso não houve diferença visualmente significativa entre os controladores.

Considerando as doze comparações de desempenho (três índices e quatro malhas) o controlador proposto obteve os melhores valores em duas delas, como pode ser visto na Tabela 5.13.

Tabela 5.11: Respostas da malha de controle latero-direcional com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL na simulação 2.

Rolamento			Guinada			
Splendor et al. (2015)						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.3s	24.80%	9.80s	22.4s	1.87%	14.94s
0	3.7s	21.32%	6.32s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.3s	25.13%	9.75s	22.2s	5.30%	23.65s
0	3.5s	20.60%	16.97s			
PID Otimizado						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	3.2s	13.33%	7.99s	19.9s	1.38%	14.50s
0	3.7s	9.70%	10.94s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	3.0s	13.70%	7.89s	18.2s	2.32%	23.53s
0	3.6s	9.53%	12.75s			
<i>ISA-PID-EG-Fuzzy</i>						
Direita (150)	tp	Mp	ts	tp	Mp	ts
20	*	0.00%	5.96s	16.6s	3.15%	18.37s
0	11.6s	0.13%	5.79s			
Esquerda (100)	tp	Mp	ts	tp	Mp	ts
-20	*	0.00%	6.11s	16.7s	3.28%	18.13s
0	12.0s	0.31%	5.78s			

Fonte: Elaborado pelo autor

5.3 Considerações

Neste capítulo foram apresentados resultados comparativos entre 3 sistemas de piloto automático para aeronaves. O primeiro é o proposto por Splendor et al. (2015). O segundo, desenvolvido com base no primeiro, utiliza controladores PID otimizados, conforme apresentado no capítulo 3. Por fim, o terceiro utiliza os controladores *ISA-PID-EG-Fuzzy*, proposto neste trabalho.

Tabela 5.12: Índices de desempenho da simulação 2 na plataforma HIL.

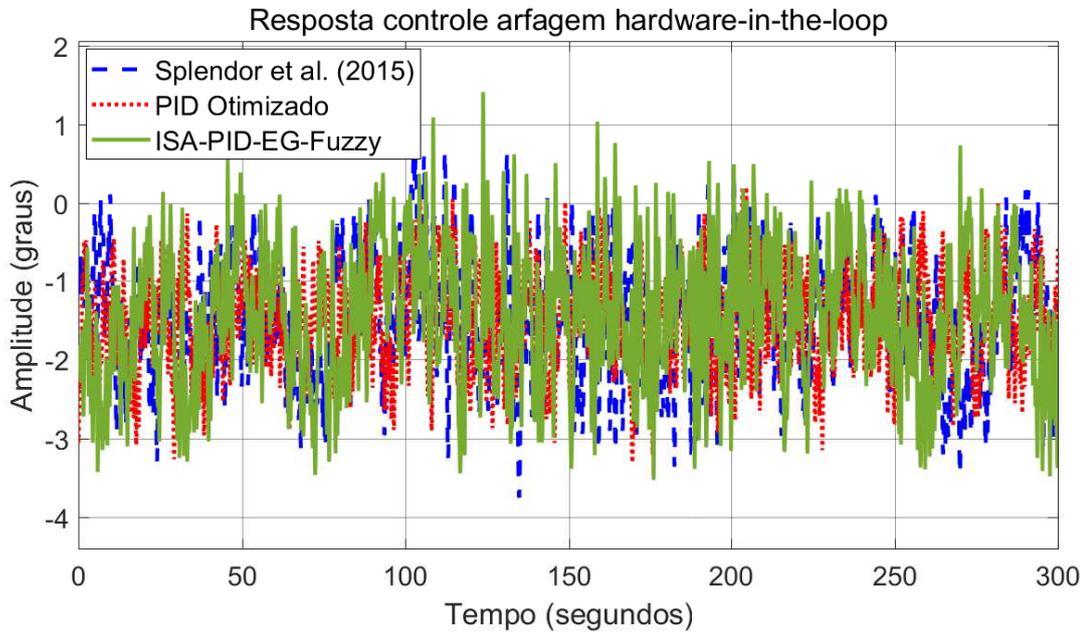
Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	9.6794×10^{-2}	1.9798×10^{-3}	1.6338×10^{-4}
PID Otimizado	5.0362×10^{-3}	1.8854×10^{-3}	1.5561×10^{-4}
<i>ISA</i> -PID-EG- <i>Fuzzy</i>	2.0832×10^{-3}	1.9423×10^{-3}	1.2646×10^{-4}
Altitude			
Splendor et al. (2015)	4.0797×10^{-2}	3.5600×10^0	1.4274×10^{-1}
PID Otimizado	5.3581×10^{-2}	2.3465×10^0	2.9341×10^{-2}
<i>ISA</i> -PID-EG- <i>Fuzzy</i>	1.4349×10^{-2}	2.4509×10^0	1.8764×10^{-2}
Rolamento			
Splendor et al. (2015)	3.9106×10^0	4.3527×10^{-3}	4.3128×10^{-3}
PID Otimizado	3.0122×10^0	4.4035×10^{-3}	4.3616×10^{-3}
<i>ISA</i> -PID-EG- <i>Fuzzy</i>	2.6000×10^0	5.3977×10^{-3}	5.3552×10^{-3}
Guinada			
Splendor et al. (2015)	4.7533×10^{-1}	9.1463×10^{-3}	9.1398×10^{-3}
PID Otimizado	4.3868×10^{-1}	8.4053×10^{-3}	8.3980×10^{-3}
<i>ISA</i> -PID-EG- <i>Fuzzy</i>	4.2764×10^{-1}	7.7700×10^{-3}	7.7651×10^{-3}

Fonte: Elaborado pelo autor

Para cada piloto automático experimentado, foram feitas três simulações com diferentes condições climáticas. Os resultados foram demonstrados por meio de gráficos que representam as respostas para os controladores para as diferentes condições de voo utilizadas nas simulações. Foram apresentadas também as características de desempenho dos controladores em termos de resposta transitória para o tempo de pico (tp), sobre-sinal (Mp) e o tempo de acomodação (ts) para as duas primeiras simulações. Não foram apresentadas as características de desempenho para terceira simulação pois as amplitudes das oscilações provocadas pelas perturbações tornam esses dados poucos significativos, provocando picos grandes capazes tirar a resposta da faixa de acomodação.

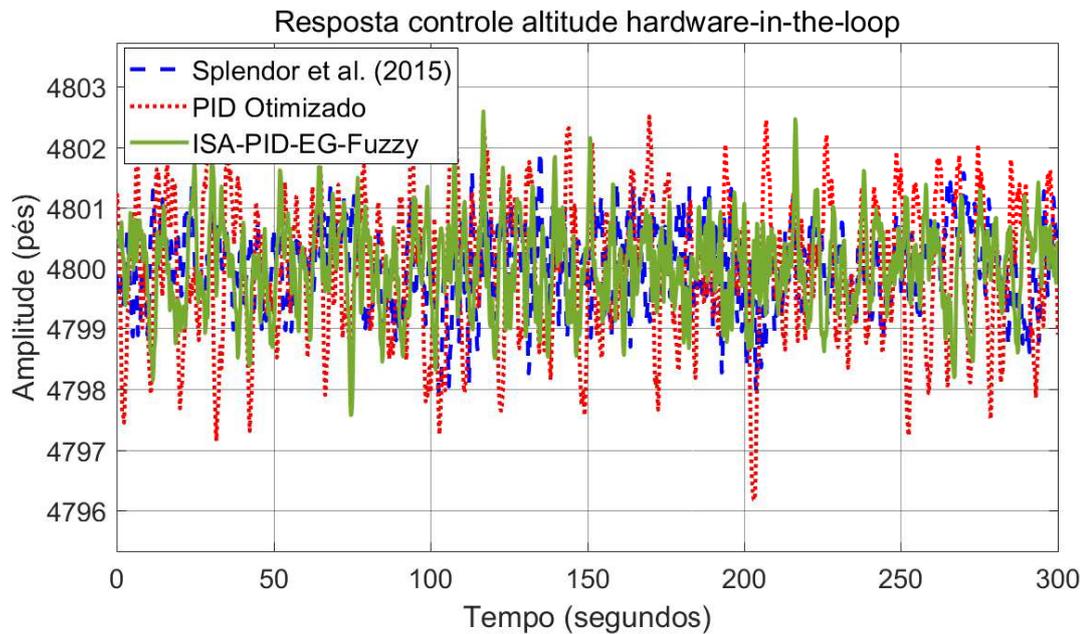
Nas otimizações feitas usando a plataforma HIL, com o simulador de voo X-Plane, foram adotados procedimentos para minimizar o acoplamento entre as malhas, da mesma forma que os relatados no capítulo 3. Diferentemente das otimizações, nas simulações as respostas das malhas de controle estão sujeitas ao forte acoplamento entre elas. Além disso, elas simulam uma aplicação, que busca abordar uma situação de voo real, ao contrário da otimização, onde há somente o processo de otimização dos ganhos dos

Figura 5.21: Comparação da resposta na simulação 3 da malha de controle do ângulo de arfagem com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 5.22: Comparação da resposta na simulação 3 da malha de controle de altitude com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.



Fonte: Elaborado pelo autor

Figura 5.23: Comparação da resposta na simulação 3 da malha de controle do ângulo de rolamento com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

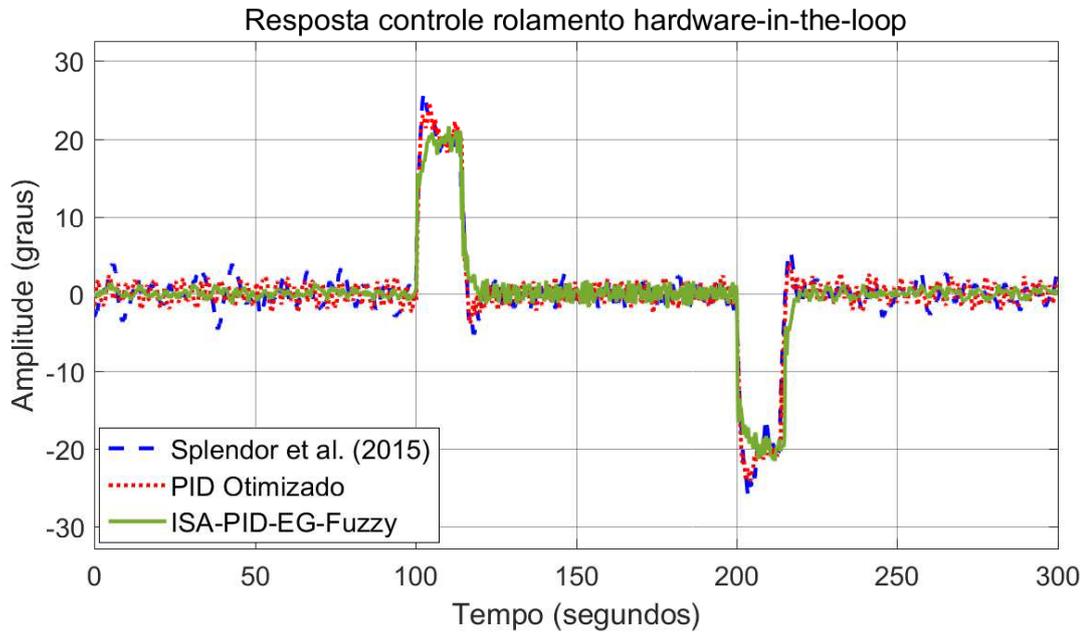


Figura 5.24: Comparação da resposta na simulação 3 da malha de controle do ângulo de guinada com controlador *ISA-PID-EG-Fuzzy* na plataforma HIL.

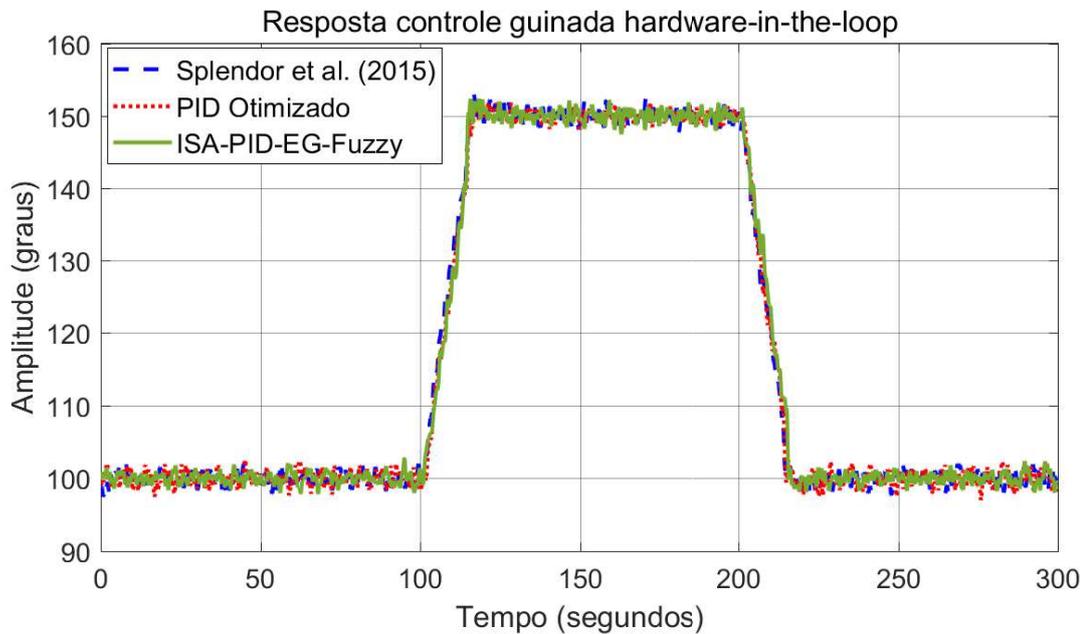


Tabela 5.13: Índices de desempenho da simulação 3 na plataforma HIL.

Ganhos	EMQ	ECM	VSC
Arfagem			
Splendor et al. (2015)	2.2001×10^{-1}	2.8734×10^{-3}	1.3503×10^{-3}
PID Otimizado	8.9773×10^{-2}	2.5134×10^{-3}	8.3243×10^{-4}
<i>ISA-PID-EG-Fuzzy</i>	3.6394×10^{-1}	4.9882×10^{-3}	3.5692×10^{-3}
Altitude			
Splendor et al. (2015)	4.3724×10^{-1}	3.9325×10^0	8.8439×10^{-1}
PID Otimizado	1.3870×10^0	2.7620×10^0	4.4989×10^{-1}
<i>ISA-PID-EG-Fuzzy</i>	5.4778×10^{-1}	2.9943×10^0	9.5410×10^{-1}
Rolamento			
Splendor et al. (2015)	5.3822×10^0	6.6488×10^{-3}	6.6441×10^{-3}
PID Otimizado	4.5544×10^0	5.9001×10^{-3}	5.8697×10^{-3}
<i>ISA-PID-EG-Fuzzy</i>	3.1610×10^0	1.3080×10^{-2}	1.3070×10^{-2}
Guinada			
Splendor et al. (2015)	1.1082×10^0	1.2132×10^{-2}	1.2152×10^{-2}
PID Otimizado	1.1790×10^0	1.2363×10^{-2}	1.2341×10^{-2}
<i>ISA-PID-EG-Fuzzy</i>	9.6629×10^{-1}	1.3655×10^{-2}	1.3625×10^{-2}

Fonte: Elaborado pelo autor

controladores na resposta ao degrau unitário e uma perturbação fixa, cujos ganhos são usados nesta aplicação. Ambas as questões acima podem ter influenciado diretamente nos resultados obtidos.

No próximo capítulo são apresentadas as conclusões deste trabalho, com algumas contribuições, juntamente com algumas propostas para possíveis trabalhos futuros.

Conclusão

Este trabalho apresentou o projeto e a implementação de um piloto automático para o controle de uma aeronave, com o uso de controladores avançados.

Com o estudo de controladores clássicos PID, que até hoje são comumente usados no desenvolvimento de pilotos automáticos, foram propostos, neste trabalho, procedimentos de otimização dos controladores, com os resultados tendo uma melhora significativa em relação aos obtidos no trabalho de Splendor et al. (2015). Também foram estudados controladores que utilizam técnicas diversas no desenvolvimento de pilotos automáticos, possibilitando fazer a comparação entre eles e entre os controladores PID otimizados.

A partir da comparação desses controladores, foram projetados novos controladores, que utilizam lógica *fuzzy* para fazer o escalonamento de ganhos do controlador PID clássico, com a intenção de torná-los capazes de realizar a compensação de incertezas e perturbações. Além disso, também foi utilizado o filtro no sinal de referência de cada controle para melhorar as características de desempenho em termos de resposta transitória.

O diferencial desta proposta em relação aos trabalhos relacionados está na integração do controlador PID com o escalonamento de ganhos utilizando lógica *fuzzy* com a estrutura ISA-PID.

Os resultados obtidos neste trabalho, conforme discutidos no capítulo anterior, apontam que o controlador proposto obteve melhores resultados que os outros controladores comparados, na maioria das comparações feitas.

6.1 Contribuições

Como principais contribuições geradas pelo desenvolvimento deste trabalho pode-se destacar que:

- As técnicas utilizadas no desenvolvimento dos controladores do piloto automático proposto podem ser adaptadas para outros tipos de aeronaves, pois dependem apenas da substituição dos valores das derivadas aerodinâmicas do Cessna 182 pelos valores das derivadas aerodinâmicas da aeronave a ser utilizada.
- Utilização dos controladores projetados como uma plataforma de testes auxiliando no estudo sobre piloto automático, pois seus parâmetros podem ser alterados e suas respostas monitoradas com certo grau de realismo.
- Os procedimentos de otimização propostos oferecem uma forma simples, eficiente, sistematizada e replicável de obtenção dos ganhos de controladores PID, dispensando o uso de métodos clássicos de determinação dos ganhos das malhas de controle, bem como de tentativa-e-erro. Com isso é minimizada consideravelmente a exigência de sintonia na planta, reduzindo substancialmente os riscos e custos envolvidos nos testes práticos e, mesmo assim, atingindo os objetivos das tarefas predeterminadas.

6.2 Dificuldades Encontradas

A maior dificuldade no desenvolvimento deste trabalho foi encontrar trabalhos de referência, específicos do tema abordado, com qualidade. Muitos artigos encontrados que aparentemente apresentavam bons resultados não foram replicados por falta de dados suficientes para isso ou, quando replicados, não se obtinham os mesmos resultados.

Muitos trabalhos encontrados como referências não foram mais profundamente explorados pela falta de conhecimento na área de controle e automação. Isso se deve a uma não formação prévia e à necessidade de disciplinas específicas para tal aprofundamento.

6.3 Trabalhos Futuros

O trabalho desenvolvido teve como principal objetivo o projeto de desenvolvimento e implementação de um piloto automático, com o uso de controladores avançados, para aeronaves. Por meio da análise dos resultados obtidos, são identificadas possíveis continuações para o trabalho ou outras linhas de pesquisas mais gerais para a teoria de

controle de algumas questões que permanecem em aberto. Assim as principais direções a serem seguidas são:

- Exploração das muitas modificações e ajustes com diferentes objetivos que podem ser feitos no controlador proposto neste trabalho, tanto na estrutura do sistema de inferência *fuzzy* quanto nos parâmetros.
- Implementação de um controlador para a velocidade horizontal atuando no ajuste do comando de aceleração da aeronave, já que neste trabalho o controle da velocidade horizontal da aeronave é realizado de forma manual pelo teclado ou com a utilização de um joystick.
- Implementação de um controlador para a velocidade vertical para atuar no controle de subida e descida da aeronave para diferentes referências de altitude, considerando que neste trabalho o controle de altitude visa somente manter a altitude de referência.
- O desenvolvimento de um sistema com escalonamento de ganhos para os controladores que cubra todas as situações de voo referentes a, principalmente, velocidade da aeronave e a altitude.
- Implementação de um controle de trajetória para que a aeronave possa executar tarefas relacionadas à navegação com o mínimo de intervenções do usuário.
- Implementação dos controladores projetados em diferentes plataformas de hardware para comparação do desempenho na velocidade de processamento e na transmissão de dados em relação à arquitetura utilizada neste trabalho.
- Projetar novos controladores utilizando outras técnicas de controle e a comparação dos resultados obtidos com os apresentados neste trabalho.
- Aplicar os resultados obtidos via simulação em um VANT real com a realização de voos autônomos.

REFERÊNCIAS

AHSAN, M.; RAFIQUE, K.; MAZHAR, F. Optimization based tuning of autopilot gains for a fixed wing UAV. *World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, v. 7, n. 5, p. 781–786, 2013.

AIAN, A. E. A.; BASUNO, B.; BIN OMAR, Z. On the determination of longitudinal transfer function of the Cessna - 182 aircraft model. *Applied Mechanics and Materials*, v. 465-466, n. 1662-7482, p. 363–367, 2014.

Disponível em <http://www.scientific.net/AMM.465-466.363> (Último Acesso em 5/5/2017)

ALMEIDA, O. M.; COELHO, A. A. R. Controlador PID com escalonamento nebuloso dos ganhos: auto-sintonia, análise e implementação. *V Simpósio Brasileiro de Automação Inteligente - V SBAI*, 2001.

ALVES, A. J. eFLL - uma biblioteca fuzzy para arduino e sistemas embarcados. 2012. Disponível em <http://www.zerokol.com/2012/09/arduinfuzzy-uma-biblioteca-fuzzy-para.html> (Último Acesso em 17/11/2017)

ÅSTRÖM, K. J.; HÄGGLUND, T. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

BEYGI, N.; BEIGY, M.; SIAHI, M. Design of fuzzy self-tuning PID controller for pitch control system of aircraft autopilot. *arXiv preprint arXiv:1510.02588*, 2015.

BLAKELOCK, J. H. *Automatic control of aircraft and missiles*. A Wiley-Interscience publication. Wiley, 1991.

Disponível em <https://books.google.com.br/books?id=ubcczZUDCsMC> (Último Acesso em 06/04/2016)

DO CARMO, M. J.; OLIVEIRA, Â. R.; CARVALHO, J. R. Utilização de estatística como índices não intrusivos na avaliação do comportamento das malhas de controle: um estudo de caso para sistemas com atraso de transporte. In: *XXXVI Congresso Nacional de Matemática Aplicada e Computacional*, 2012, p. 1034–1040.

COELHO, L. S.; ALMEIDA, O. M.; SIMAS, H.; COELHO, A. A. R. Intelligent and self-tuning PID controllers: methods and application. In: *12º CBA*, 1998, p. 375–380.

COOK, M. V. *Flight dynamics principles (second edition)*. Second edition ed. Oxford: Butterworth-Heinemann, 2007.

DUALIBE, C.; VERLEYSSEN, M.; JESPERS, P. *Design of analog fuzzy logic controllers in CMOS technologies: implementation, test and application*. Springer Science & Business Media, 2007.

DUAN, X. G.; LI, H. X.; DENG, H. A simple tuning method for fuzzy PID control. In: *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, 2008, p. 271–275.

EFE, M. Ö. Sliding mode control for unmanned aerial vehicles research. In: YU, X.; ÖNDER EFE, M., eds. *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*, v. 24 de *Studies in Systems, Decision and Control*, Springer International Publishing, p. 239–255, 2015.

Disponível em http://dx.doi.org/10.1007/978-3-319-18290-2_12 (Último Acesso em 22/02/2016)

ESPINOZA, T.; DZUL, A.; LLAMA, M. Linear and nonlinear controllers applied to fixed-wing UAV. *International Journal of Advanced Robotic Systems*, v. 10, 2013.

FAMULARO, D.; FRANZE, G.; FURFARO, A.; MATTEI, M. A hybrid real-time command governor supervisory scheme for constrained control systems. *Control Systems Technology, IEEE Transactions on*, v. 23, n. 3, p. 924–936, 2015.

FARID, A. M. UAV controller based on adaptive neuro-fuzzy inference system and PID. *IAES International Journal of Robotics and Automation*, v. 2, n. 2, p. 73, 2013.

FIUZY, M.; HADDADNIA, J.; MASHHADI, S. Designing an optimal PID controller for control the plan's height, based on control of autopilot by using evolutionary algorithms. *Journal of mathematics and computer Science*, v. 6, p. 260–271, 2013.

FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A.; POWELL, J. D. *Feedback control of dynamic systems*, v. 3. Addison-Wesley Reading, MA, 1994.

KADA, B.; GHAZZAWI, Y. Robust PID controller design for an UAV flight control system. In: *Lecture Notes in Engineering and Computer Science*, 2011.

Disponível em http://www.iaeng.org/publication/WCECS2011/WCECS2011_pp945-950.pdf (Último Acesso em 10/05/2016)

LE-HUY, H. An adaptive fuzzy controller for permanent-magnet ac servo drives. In: *Industry Applications Conference, 1995. Thirtieth IAS Annual Meeting, IAS '95., Conference Record of the 1995 IEEE*, 1995, p. 104–110 vol.1.

LEHMANN, E. L.; CASELLA, G. *Theory of point estimation*. Springer Science & Business Media, 2006.

LI, H. X.; GATLAND, H. B.; GREEN, A. W. Fuzzy variable structure control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 27, n. 2, p. 306–312, 1997.

MANN, G. K. I.; HU, B.-G.; GOSINE, R. G. Analysis of direct action fuzzy PID controller structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 29, n. 3, p. 371–388, 1999.

MATHWORKS Simulink design optimization. 2016.

Disponível em <http://www.mathworks.com/help/slido/index.html> (Último Acesso em 20/01/2016)

MCLEAN, D. *Automatic flight control systems*. Prentice-Hall international series in systems and control engineering. Prentice Hall, 1990.

Disponível em <https://books.google.co.in/books?id=cJNTAAAAMAAJ> (Último Acesso em 22/02/2016)

MELKOU, L.; REZOUQ, A.; HAMERLAIN, M. PID-terminal sliding mode control of aircraft UAV. In: *Modelling Symposium (EMS), 2014 European*, 2014, p. 233–238.

MRAD, F.; DEEB, G. Experimental comparative analysis of adaptive fuzzy logic controllers. *IEEE Transactions on Control Systems Technology*, v. 10, n. 2, p. 250–255, 2002.

MUDI, R. K.; PAL, N. R. A robust self-tuning scheme for PI- and PD-type fuzzy controllers. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 1, p. 2–16, 1999.

MURILLO, M. H.; LIMACHE, A. C.; ROJAS FREDINI, P. S.; GIOVANINI, L. L. Generalized nonlinear optimal predictive control using iterative state-space trajectories: Applications to autonomous flight of UAVs. *International Journal of Control, Automation and Systems*, v. 13, n. 2, p. 361–370, 2015.

Disponível em <http://dx.doi.org/10.1007/s12555-013-0416-y> (Último Acesso em 22/02/2016)

MYSTKOWSKI, A. Robust control of the micro UAV dynamics with an autopilot. *Journal of Theoretical and Applied Mechanics*, v. 51, n. 3, 2013.

Disponível em <http://www.ptmts.org.pl/jtam/index.php/jtam/article/view/v51n3p751> (Último Acesso em 22/02/2016)

NARENATHREYAS, K. B. *Fuzzy logic control for aircraft longitudinal motion*. Dissertação de mestrado, Czech Technical University, 2013.

NELSON, R. C. *Flight stability and automatic control*, v. 2. WCB/McGraw Hill, 1998.

NERIS, L. D. O.; TRINDADE JUNIOR, O. *Um piloto automático para as aeronaves do projeto ARARA*. Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Paulo, 2001.

Disponível em <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-12092002-130939/> (Último Acesso em 22/02/2016)

OGATA, K. *Modern control engineering*. Instrumentation and controls series. Prentice Hall, 2010.

PROCHASKA, E. O. C. *Aprendendo a voar: manual de treinamento básico – piloto de avião*. Phorte Editora, 2011.

RIBEIRO, L. R.; OLIVEIRA, N. M. F. UAV autopilot controllers test platform using Matlab/Simulink and X-plane. In: *Frontiers in Education Conference (FIE), 2010 IEEE*, 2010, p. S2H–1–S2H–6.

RODRIGUES, L. E. M. J. *Fundamentos da engenharia aeronáutica*. Cengage learning, 2013.

ROSKAM, J. *Airplane flight dynamics and automatic flight controls*. N. p. 1 in *Airplane Flight Dynamics and Automatic Flight Controls*. Roskam Aviation and Engineering Corporation, 1995.

Disponível em <https://books.google.com.br/books?id=RDZUG--o9HMC> (Último Acesso em 22/02/2016)

SANTOS, S. R. B. D.; JÚNIOR, C. L. N.; JUNIOR, S. N. G.; BITTAR, A.; DE OLIVEIRA, N. M. F. Experimental framework for evaluation of guidance and control algorithms for UAVs. In: *21st Brazilian Congress of Mechanical Engineering, COBEM*, 2011.

SANTOS, S. R. B. D.; OLIVEIRA, N. M. F. D. Longitudinal autopilot controllers test platform hardware in the loop. In: *Systems Conference (SysCon), 2011 IEEE International*, 2011, p. 379–386.

SARHAN, A.; ASHRY, M. Self-tuned PID controller for the aerosonde UAV autopilot. In: *International Journal of Engineering Research and Technology*, ESRSA Publications, 2013.

SARHAN, A.; QIN, S. Adaptive PID control of UAV altitude dynamics based on parameter optimization with fuzzy inference. *International Journal of Modeling and Optimization*, v. 6, n. 4, p. 246, 2016.

SARTORI, D. *Design, implementation and testing of advanced control laws for fixed-wing UAVs*. Tese de Doutorado, Politecnico di Torino, 2014.

Disponível em <http://porto.polito.it/2571146/> (Último Acesso em 20/07/2017)

SHENGYI, Y.; KUNQIN, L.; JIAO, S. Design and simulation of the longitudinal autopilot of UAV based on self-adaptive fuzzy PID control. In: *Computational Intelligence and Security, 2009. CIS '09. International Conference on*, 2009, p. 634–638.

SIMÕES, M. G.; SHAW, I. S. *Controle e modelagem fuzzy*. A Wiley-Interscience publication. Editora Edgar Blücher Ltda., 1999.

Disponível em <https://books.google.com.br/books?id=ubcczZUDcSMC> (Último Acesso em 22/02/2016)

SPLENDOR, F.; MARTINS, N. A.; DE SOUZA GIMENES, I. M.; MARTINI, J. A. Design of an autopilot for Cessna 182. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, v. 13, n. 1, p. 27–36, 2015.

SUGENO, M. *Industrial applications of fuzzy control*. New York, NY, USA: Elsevier Science Inc., 1985.

TANAKA, K. *An introduction to fuzzy logic for practical applications*. Springer New York, 1996.

Disponível em <https://books.google.com.br/books?id=Fq-eXnV56coC> (Último Acesso em 22/02/2016)

THE MATHWORKS, I. Designing PID for disturbance rejection with PID tuner - Matlab/Simulink example. 2017.

Disponível em <https://www.mathworks.com/help/control/examples/designing-pid-for-disturbance-rejection-with-pid-tuner.html> (Último Acesso em 14/02/2017)

THUMS, G. D.; TORRES, L. A. B.; PALHARES, R. M. Metodologia de sintonia PID multi-malha para veículos aéreos não tripulados: dinâmica longitudinal. In: *XIX Congresso Brasileiro de Automática, CBA*, 2012, p. 358–365.

TURKOGLU, K.; OZDEMIR, U.; NIKBAY, M.; JAFAROV, E. M. PID parameter optimization of an UAV longitudinal flight control system. *World Academy of Science, Engineering and Technology*, v. 21, n. 9, p. 1031 – 1036, 2008.

Disponível em <http://waset.org/Publications?p=21> (Último Acesso em 22/02/2016)

VALE, M. R. B. G. *Análise comparativa do desempenho de um controlador fuzzy acoplado a um PID neural sintonizado por um algoritmo genético com controladores inteligentes convencionais*. Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, 2007.

Disponível em <http://repositorio.ufrn.br:8080/jspui/handle/123456789/15184> (Último Acesso em 05/09/2017)

VURAL, S. Y.; HAJIYEV, C. A comparison of longitudinal controllers for autonomous UAV. *International Journal of Sustainable Aviation*, v. 1, n. 1, p. 58–71, 2014.

WAHID, N.; HASSAN, N. Self-tuning fuzzy PID controller design for aircraft pitch control. In: *2012 Third International Conference on Intelligent Systems Modelling and Simulation*, IEEE, 2012, p. 19–24.

WAHID, N.; RAHMAT, M. F. Pitch control system using LQR and fuzzy logic controller. In: *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, 2010, p. 389–394.

WANG, L. *A course in fuzzy systems and control*. Prentice Hall PTR, 1997.

Disponível em <https://books.google.com.br/books?id=wbJQAAAAMAAJ> (Último Acesso em 25/10/2017)

WILSON, D. I. Relay-based PID tuning. *Automation and Control*, p. 10–11, 2005.

YADAV, A. K.; GAUR, P. Ai-based adaptive control and design of autopilot system for nonlinear UAV. *Sadhana*, v. 39, n. 4, p. 765–783, 2014.

Disponível em <http://dx.doi.org/10.1007/s12046-014-0275-0> (Último Acesso em 22/02/2016)

ZADEH, L. A. Fuzzy sets. *Information and control*, v. 8, n. 3, p. 338–353, 1965.

ZHAO, Z.-Y.; TOMIZUKA, M.; ISAKA, S. Fuzzy gain scheduling of PID controllers. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 23, n. 5, p. 1392–1398, 1993.

ZHITECKII, L.; PILCHEVSKY, A.; KRAVCHENKO, A.; BYKOV, B. Modern control theory for designing lateral autopilot systems of UAV. In: *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2015 IEEE International Conference, 2015*, p. 160–164.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. *trans. ASME*, v. 64, n. 11, 1942.

Código do Controlador Implementado no Arduino

```
/*
  Autor: Fernando Palmieri Freire
  Contato: fpfreire@gmail.com / pg48331@uem.br

  Códigos adaptados de Fábio Splendor

  Configuração X-Plane 10:
  Entrada e Saída de Dados: UDP
    3 speeds
    4 Mach, VVI, G-load
    8 joystick ail/elv/rud
    16 angular velocities
    17 pitch, roll, headings
    20 lat, lon, altitude
    25 comando de aceleração

  Biblioteca Fuzzy: http://www.zerokol.com/2012/09/arduino-fuzzy-uma-biblioteca-fuzzy-para.html
*/

#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>

float convFloat(byte index);
// Retorna o valor dos 4 bytes, onde index é a posição do primeiro byte
// no pacote enviado pelo X-Plane, convertido para ponto flutuante.
// index = 5 + 36*Linha + 4*Coluna
// http://svglobe.com/arduino/udpdata.html

void buildFuzzy();
```

```

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 0, 177); // Microcontrolador Arduino Mega 2560
IPAddress remIp(192, 168, 0, 2); //1, 100); Computador rodando o X-Plane
IPAddress groundStationIp(192, 168, 0, 1); //1, 100); Computador rodando o Matlab/Simulink

// Porta remota de entrada de dados saídos do Arduino Mega para o X-Plane
unsigned int remPort = 49000;
// Porta local de entrada de dados saídos do X-Plane
unsigned int localPort = 49001;
// Porta remota de entrada de dados saídos do Arduino Mega para o Matlab/Simulink
unsigned int remPortGroundStation = 49002;

float

// Ganhos dissertação Splendor
/*
  arfKp = 0.1430,  arfKi = 0.0000,  arfKd = 0.0000,  arfKrg = 0.011800,
  arfKpr = 0.0,  arfKir = 0.0,  arfKdr = 0.0,  arfKe = 0.0,  arfKec = 0.0,  arfB = 1.0,
  altKp = 0.0112,  altKi = 0.00038,  altKd = 0.0032,
  altKpr = 0.0,  altKir = 0.0,  altKdr = 0.0,  altKe = 0.0,  altKec = 0.0,  altB = 1.0,
  rolKp = 2.6430,  rolKi = 1.6498,  rolKd = 1.1090,  rolKrg = 0.0803,
  rolKpr = 0.0,  rolKir = 0.0,  rolKdr = 0.0,  rolKe = 0.0,  rolKec = 0.0,  rolB = 1.0,
  guiKp = 0.2143,  guiKi = 0.5698,  guiKd = 0.0089,  guiKrg = 0.1000,
  guiKpr = 0.0,  guiKir = 0.0,  guiKdr = 0.0,  guiKe = 0.0,  guiKec = 0.0,  guiB = 1.0,
  /**/

// Ganhos otimizados MIL
/*
  arfKp = 0.071278,  arfKi = 0.020630,  arfKd = 0.0000,  arfKrg = 0.011800,
  arfKpr = 0.0,  arfKir = 0.0,  arfKdr = 0.0,  arfKe = 0.0,  arfKec = 0.0,  arfB = 1.0,
  altKp = 0.0052,  altKi = 0.0001,  altKd = 0.0032,
  altKpr = 0.0,  altKir = 0.0,  altKdr = 0.0,  altKe = 0.0,  altKec = 0.0,  altB = 1.0,
  rolKp = 5.915,  rolKi = 2.574,  rolKd = 0.662,  rolKrg = 0.803,
  rolKpr = 0.0,  rolKir = 0.0,  rolKdr = 0.0,  rolKe = 0.0,  rolKec = 0.0,  rolB = 1.0,
  guiKp = 0.2786,  guiKi = 0.4322,  guiKd = 0.0089,  guiKrg = 0.0562,
  guiKpr = 0.0,  guiKir = 0.0,  guiKdr = 0.0,  guiKe = 0.0,  guiKec = 0.0,  guiB = 1.0,
  /**/

// Ganhos otimizados PID SIL
/*altKp = 0.0009577,  altKi = 0.000182,  altKd = 0.0003,
  arfKp = 5.19039,  arfKi = 5.53814,  arfKd = 0.011586,  arfKrg = 0.04328,
  rolKp = 1.39062,  rolKi = 0.18858,  rolKd = 0.74935,  rolKrg = 0.014377,
  guiKp = 0.359892,  guiKi = 0.219012,  guiKd = 0.005109,  guiKrg = 0.00115377,
  */
// Ganhos otimizados PID HIL
/*
  arfKp = 0.089398,  arfKi = 0.051016,  arfKd = 0.000797,  arfKrg = 0.005284,
  arfKpr = 0.0,  arfKir = 0.0,  arfKdr = 0.0,  arfKe = 0.0,  arfKec = 0.0,  arfB = 1.0,
  altKp = 0.0038,  altKi = 0.0016,  altKd = 0.0030,
  altKpr = 0.0,  altKir = 0.0,  altKdr = 0.0,  altKe = 0.0,  altKec = 0.0,  altB = 1.0,
  rolKp = 3.4081,  rolKi = 1.1158,  rolKd = 0.6903,  rolKrg = 0.0000,
  rolKpr = 0.0,  rolKir = 0.0,  rolKdr = 0.0,  rolKe = 0.0,  rolKec = 0.0,  rolB = 1.0,

```

```

guiKp = 0.2786,   guiKi = 0.6584,   guiKd = 0.0084,   guiKrg = 0.0943,
guiKpr = 0.0,   guiKir = 0.0,   guiKdr = 0.0,   guiKe = 0.0,   guiKec = 0.0,   guiB = 1.0,
*/

// Ganhos HIL ISA-PID-EG-Fuzzy

arfKp = 0.08474, arfKi = 0.09958, arfKd = 0.001746, arfKrg = 0.008583,
arfKpr = 0.04125, arfKir = 0.01786, arfKdr = 0.00017, arfKe = 0.6251, arfKec = 0.2077, arfB = 0.5198,
altKp = 0.00550, altKi = 0.00276, altKd = 0.002122,
altKpr = 0.00143, altKir = 0.00100, altKdr = 0.00344, altKe = 0.8518, altKec = 1.0383, altB = 0.4687,
rolKp = 8.04026, rolKi = 3.49074, rolKd = 0.314783, rolKrg = 0.0000,
rolKpr = 6.91750, rolKir = 1.27627, rolKdr = 0.10541, rolKe = 0.7013, rolKec = 1.4412, rolB = 0.5958,
guiKp = 0.29020, guiKi = 1.14190, guiKd = 0.008513, guiKrg = 0.1359,
guiKpr = 0.03028, guiKir = 0.07578, guiKdr = 0.01289, guiKe = 0.9748, guiKec = 1.0940, guiB = 0.6007,
/**/

// Set points (sp) iniciais
spAltitude = 4800.0, spArfagem = 0.0, spRolamento = 0.0, spGuinada = 100.0, // 77.74,
spLatitude = 0.0, spLongitude = 0.0,

altAtual = 0.0,
altErroAnterior = 0.0, altErroAtual = 0.0, altIntegralErro = 0.0, altDerivadaErro = 0.0,
arfAtual = 0.0, arfAnterior = 0.0, arfDerivada = 0.0,
arfErroAnterior = 0.0, arfErroAtual = 0.0, arfIntegralErro = 0.0, arfDerivadaErro = 0.0,
rolAtual = 0.0, rolAnterior = 0.0, rolDerivada = 0.0,
rolErroAnterior = 0.0, rolErroAtual = 0.0, rolIntegralErro = 0.0, rolDerivadaErro = 0.0,
guiAtual = 0.0, guiAnterior = 0.0, guiDerivada = 0.0,
guiErroAnterior = 0.0, guiErroAtual = 0.0, guiIntegralErro = 0.0, guiDerivadaErro = 0.0,
lonAtual = 0.0, lonAnterior = 0.0, latAtual = 0.0, latAnterior = 0.0,

parcelaIntegralArf, parcelaIntegralAlt, parcelaIntegralRol, parcelaIntegralGui,
parcelaIntegralSpArf = 0.0, integralSpArf = 0.0, spArfagemISA = 0.0, disturbio = 0.1,
parcelaIntegralSpAlt = 0.0, integralSpAlt = 0.0, spAltitudeISA = 0.0,
parcelaIntegralSpRol = 0.0, integralSpRol = 0.0, spRolamentoISA = 0.0,
parcelaIntegralSpGui = 0.0, integralSpGui = 0.0, spGuinadaISA = 0.0,
saidaArf, saidaRol, saidaGui,
temp = 0.0, segundos;

EthernetUDP Udp;
EthernetUDP UdpGroundStation;

char readBuffer[257];
char writeBuffer[149] = {
    68, 65, 84, 65, 48, // DATA0 [0-4]
    8, 0, 0, 0, // joystick ail/elv/rud [5-8]
    0, 192, 121, 196, // -999 [9-12] Profundor
    0, 192, 121, 196, // -999 [13-16] Aileron
    0, 192, 121, 196, // -999 [17-20] Leme
    0, 192, 121, 196, // -999 [21-24]
    0, 192, 121, 196, // -999 [25-28]
    0, 192, 121, 196, // -999 [29-32]
    0, 192, 121, 196, // -999 [33-36]
    0, 192, 121, 196, // -999 [37-40]
    8, 0, 0, 0, // Enviado para ler o joystick em paralelo com o controlador

```

```

0, 192, 121, 196, // -999 [45-48]
0, 192, 121, 196, // 49
0, 192, 121, 196, // 53
0, 192, 121, 196, // 57
0, 192, 121, 196, // 61
0, 192, 121, 196, // 65
0, 192, 121, 196, // 69
0, 192, 121, 196, // 73
25, 0, 0, 0, // Acelerador
0, 192, 121, 196, // 81
0, 192, 121, 196, // 85
0, 192, 121, 196, // 89
0, 192, 121, 196, // 93
0, 192, 121, 196, // 97
0, 192, 121, 196, // 101
0, 192, 121, 196, // 105
0, 192, 121, 196, // 109
62, 0, 0, 0, // Peso combustível
0, 192, 121, 196, // 117
0, 192, 121, 196, // 121
0, 192, 121, 196, // 125
0, 192, 121, 196, // 129
0, 192, 121, 196, // 133
0, 192, 121, 196, // 137
0, 192, 121, 196, // 141
0, 192, 121, 196 // 145
};
// Buffer para enviar a referência do ângulo de arfagem para o Simulink.
char writeBufferGroundStation[4];
unsigned long time, timeBack;
int i, j, k;

// Instanciando um objeto da biblioteca
Fuzzy* fuzzy = new Fuzzy();

// Entradas para o erro e derivada do erro
FuzzySet* eNB = new FuzzySet(-1.00, -1.00, -0.75, -0.25);
FuzzySet* eNM = new FuzzySet(-1.00, -0.50, -0.50, 0.00);
FuzzySet* eNS = new FuzzySet(-0.75, -0.25, -0.25, 0.25);
FuzzySet* eZO = new FuzzySet(-0.50, 0.00, 0.00, 0.50);
FuzzySet* ePS = new FuzzySet(-0.25, 0.25, 0.25, 0.75);
FuzzySet* ePM = new FuzzySet(0.00, 0.50, 0.50, 1.00);
FuzzySet* ePB = new FuzzySet(0.25, 0.75, 1.00, 1.00);

FuzzySet* ecNB = new FuzzySet(-1.00, -1.00, -0.75, -0.25);
FuzzySet* ecNM = new FuzzySet(-1.00, -0.50, -0.50, 0.00);
FuzzySet* ecNS = new FuzzySet(-0.75, -0.25, -0.25, 0.25);
FuzzySet* ecZO = new FuzzySet(-0.50, 0.00, 0.00, 0.50);
FuzzySet* ecPS = new FuzzySet(-0.25, 0.25, 0.25, 0.75);
FuzzySet* ecPM = new FuzzySet(0.00, 0.50, 0.50, 1.00);
FuzzySet* ecPB = new FuzzySet(0.25, 0.75, 1.00, 1.00);

// Saídas para Kpr e Kir
FuzzySet* Skpki = new FuzzySet(-1.0, 0.0, 0.0, 1.0);

```

```

FuzzySet* Bkpci = new FuzzySet(0.0, 1.0, 1.0, 2.0);

// Saídas para Kdr
FuzzySet* Skd = new FuzzySet(-1.0, 0.0, 0.0, 1.0);
FuzzySet* Bkd = new FuzzySet(0.0, 1.0, 1.0, 2.0);

void setup() {
    buildFuzzy();
    // Start the Ethernet and UDP:
    Ethernet.begin(mac, ip);
    Udp.begin(localPort);           // Porta de entrada dos dados da navegação (X-Plane).
    UdpGroundStation.begin(49177); // Porta de entrada dos dados da estação solo (Matlab/Simulink).
}

void loop() {
    // Leitura dados da estação solo (Matlab/Simulink).
    if (UdpGroundStation.parsePacket()) { // Se a estação solo enviar dados.
        UdpGroundStation.read(readBuffer, 176);
        if (readBuffer[2] != 121) {
            temp = convFloat(0);
            if (temp != spArfagem) spArfagem = temp;
        }
        if (readBuffer[6] != 121) {
            temp = convFloat(4);
            if (temp != spAltitude) spAltitude = temp;
        }
        if (readBuffer[10] != 121) {
            temp = convFloat(8);
            if (temp != spRolamento) spRolamento = temp;
        }
        if (readBuffer[14] != 121) {
            temp = convFloat(12);
            if (temp != spGuinada) spGuinada = temp;
        }
        if (readBuffer[18] != 121) {
            temp = convFloat(16);
            if (temp != arfKp) arfKp = temp;
        }
        if (readBuffer[22] != 121) {
            temp = convFloat(20);
            if (temp != arfKi) arfKi = temp;
        }
        if (readBuffer[26] != 121) {
            temp = convFloat(24);
            if (temp != arfKd) arfKd = temp;
        }
        if (readBuffer[30] != 121) {
            temp = convFloat(28);
            if (temp != arfKrg) arfKrg = temp;
        }
        if (readBuffer[34] != 121) {
            temp = convFloat(32);
            if (temp != altKp) altKp = temp;
        }
    }
}

```

```
if (readBuffer[38] != 121) {
    temp = convFloat(36);
    if (temp != altKi) altKi = temp;
}
if (readBuffer[42] != 121) {
    temp = convFloat(40);
    if (temp != altKd) altKd = temp;
}
if (readBuffer[46] != 121) {
    temp = convFloat(44);
    if (temp != rolKp) rolKp = temp;
}
if (readBuffer[50] != 121) {
    temp = convFloat(48);
    if (temp != rolKi) rolKi = temp;
}
if (readBuffer[54] != 121) {
    temp = convFloat(52);
    if (temp != rolKd) rolKd = temp;
}
if (readBuffer[58] != 121) {
    temp = convFloat(56);
    if (temp != rolKrg) rolKrg = temp;
}
if (readBuffer[62] != 121) {
    temp = convFloat(60);
    if (temp != guiKp) guiKp = temp;
}
if (readBuffer[66] != 121) {
    temp = convFloat(64);
    if (temp != guiKi) guiKi = temp;
}
if (readBuffer[70] != 121) {
    temp = convFloat(68);
    if (temp != guiKd) guiKd = temp;
}
if (readBuffer[74] != 121) {
    temp = convFloat(72);
    if (temp != guiKrg) guiKrg = temp;
}
if (readBuffer[78] != 121) {
    temp = convFloat(76);
    if (temp != arfKpr) arfKpr = temp;
}
if (readBuffer[82] != 121) {
    temp = convFloat(80);
    if (temp != arfKir) arfKir = temp;
}
if (readBuffer[86] != 121) {
    temp = convFloat(84);
    if (temp != arfKdr) arfKdr = temp;
}
if (readBuffer[90] != 121) {
    temp = convFloat(88);
```

```

    if (temp != arfKe) arfKe = temp;
}
if (readBuffer[94] != 121) {
    temp = convFloat(92);
    if (temp != arfKec) arfKec = temp;
}
if (readBuffer[98] != 121) {
    temp = convFloat(96);
    if (temp != arfB) arfB = temp;
}
if (readBuffer[102] != 121) {
    temp = convFloat(100);
    if (temp != disturbio) disturbio = temp;
}
if (readBuffer[106] != 121) {
    temp = convFloat(104);
    if (temp != altKpr) altKpr = temp;
}
if (readBuffer[110] != 121) {
    temp = convFloat(108);
    if (temp != altKir) altKir = temp;
}
if (readBuffer[114] != 121) {
    temp = convFloat(112);
    if (temp != altKdr) altKdr = temp;
}
if (readBuffer[118] != 121) {
    temp = convFloat(116);
    if (temp != altKe) altKe = temp;
}
if (readBuffer[122] != 121) {
    temp = convFloat(120);
    if (temp != altKec) altKec = temp;
}
if (readBuffer[126] != 121) {
    temp = convFloat(124);
    if (temp != altB) altB = temp;
}
if (readBuffer[130] != 121) {
    temp = convFloat(128);
    if (temp != rolKpr) rolKpr = temp;
}
if (readBuffer[134] != 121) {
    temp = convFloat(132);
    if (temp != rolKir) rolKir = temp;
}
if (readBuffer[138] != 121) {
    temp = convFloat(136);
    if (temp != rolKdr) rolKdr = temp;
}
if (readBuffer[142] != 121) {
    temp = convFloat(140);
    if (temp != rolKe) rolKe = temp;
}
}

```

```

if (readBuffer[146] != 121) {
    temp = convFloat(144);
    if (temp != rolKec) rolKec = temp;
}
if (readBuffer[150] != 121) {
    temp = convFloat(148);
    if (temp != rolB) rolB = temp;
}
if (readBuffer[154] != 121) {
    temp = convFloat(152);
    if (temp != guiKpr) guiKpr = temp;
}
if (readBuffer[158] != 121) {
    temp = convFloat(156);
    if (temp != guiKir) guiKir = temp;
}
if (readBuffer[162] != 121) {
    temp = convFloat(160);
    if (temp != guiKdr) guiKdr = temp;
}
if (readBuffer[166] != 121) {
    temp = convFloat(164);
    if (temp != guiKe) guiKe = temp;
}
if (readBuffer[170] != 121) {
    temp = convFloat(168);
    if (temp != guiKec) guiKec = temp;
}
if (readBuffer[174] != 121) {
    temp = convFloat(172);
    if (temp != guiB) guiB = temp;
}
}

// Leitura dados da navegação (X-Plane).
if (Udp.parsePacket()) { // Se o simulador enviar dados.
    Udp.read(readBuffer, 257);
    segundos = (millis() - time) / 1000.0; // Tempo em segundos desde o último comando.
    time = millis();

    // CONTROLE DE ALTITUDE
    // A altitude só é controlada se a referência for maior que um, se não, a arfagem é controlada
    // usando o valor de referência para ela.
    if (spAltitude > 1.0) {
        // Altitude em pés nível do mar fornecido pelo X-Plane.
        altAtual = convFloat(197);
        // Erro anterior altitude para cálculo da derivada e da integral.
        altErroAnterior = altErroAtual;
        // Filtro referência
        parcelaIntegralSpAlt = (spAltitude * (1.0 - altB) - (integralSpAlt * altKi / altKp)) * segundos;
        integralSpAlt = integralSpAlt + parcelaIntegralSpAlt;
        spAltitudeISA = spAltitude * altB + integralSpAlt * altKi / altKp;
        if (abs(spAltitude - altAtual) < abs(spAltitudeISA - altAtual)) spAltitudeISA = spAltitude;
        // Erro atual da altitude em pés
    }
}

```

```

altErroAtual = spAltitudeISA - altAtual;
// Derivada do erro no tempo em pés/segundo.
altDerivadaErro = (altErroAtual - altErroAnterior) / segundos;
// Parcela final da integral do erro no tempo em graus*segundos.
parcelaIntegralAlt = ((altErroAnterior + altErroAtual) / 2.0) * segundos;
fuzzy->setInput(1, altErroAtual * altKe);
fuzzy->setInput(2, altDerivadaErro * altKec);
fuzzy->fuzzify();
// Comando servo Profundor com conversão na saída
spArfagem = ((altKp + altKpr * fuzzy->defuzzify(1)) * altErroAtual
             + (altKi + altKir * fuzzy->defuzzify(1)) * (altIntegralErro + parcelaIntegralAlt)
             + (altKd + altKdr * fuzzy->defuzzify(2)) * altDerivadaErro) * 100.0;
// Limite para o argulo de arfagem: Controle simplório da velocidade vertical.
if (spArfagem > 10) spArfagem = 10;
else if (spArfagem < -30) spArfagem = -30;
// Anti wind-up - Integral do erro no tempo em graus*segundos.
else altIntegralErro = altIntegralErro + parcelaIntegralAlt;
// Limitar a integral na interrupção do dados do X-Plane
if (((altKi + altKir * fuzzy->defuzzify(1)) * altIntegralErro) > 15.0)
    altIntegralErro = 15.0 / (altKi + altKir * fuzzy->defuzzify(1));
if (((altKi + altKir * fuzzy->defuzzify(1)) * altIntegralErro) < -30.0)
    altIntegralErro = -30.0 / (altKi + altKir * fuzzy->defuzzify(1));
convByte(9, spArfagem); // Conversão para envio ao Simulink
for (int i = 0; i < 4; i++) writeBufferGroundStation[i] = writeBuffer[i + 9];
arfB = 1.0; // Comentar essa linha para usar filtro de referência do ângulo de arfagem.
}

// CONTROLE DE ARFAGEM
// Ângulo de arfagem anterior em graus para cálculo de derivada.
arfAnterior = arfAtual;
// Ângulo de arfagem em graus fornecido pelo X-Plane.
arfAtual = convFloat(153);
// Derivada do ângulo de arfagem em graus/segundo
arfDerivada = (arfAtual - arfAnterior) / segundos;
// Erro anterior do ângulo de arfagem em radianos para cálculo da derivada e da integral.
arfErroAnterior = arfErroAtual;
// Filtro referência
parcelaIntegralSpArf = (spArfagem * (1.0 - arfB) - (integralSpArf * arfKi / arfKp)) * segundos;
integralSpArf = integralSpArf + parcelaIntegralSpArf;
spArfagemISA = spArfagem * arfB + integralSpArf * arfKi / arfKp;
// Erro atual do ângulo de arfagem em graus
arfErroAtual = spArfagemISA - arfAtual;
// Derivada do erro no tempo em graus/segundos.
arfDerivadaErro = (arfErroAtual - arfErroAnterior) / segundos;
// Parcela final da integral do erro no tempo em graus*segundos.
parcelaIntegralArf = ((arfErroAnterior + arfErroAtual) / 2.0) * segundos;
fuzzy->setInput(1, arfErroAtual * arfKe);
fuzzy->setInput(2, arfDerivadaErro * arfKec);
fuzzy->fuzzify();
// Comando servo Profundor com conversão na saída
saidaArf = ((arfKp + arfKpr * fuzzy->defuzzify(1)) * arfErroAtual
            + (arfKi + arfKir * fuzzy->defuzzify(1)) * (arfIntegralErro + parcelaIntegralArf)
            + (arfKd + arfKdr * fuzzy->defuzzify(2)) * arfDerivadaErro - arfKrg * arfDerivada);
if (saidaArf > -1 && saidaArf < 1) { // Anti wind-up

```

```

// Integral acumulada do erro no tempo em graus*segundos.
arfIntegralErro = arfIntegralErro + parcelaIntegralArf;
}
// Limitar a integral na interrupção do dados
if ((arfKi * arfIntegralErro) > 1.0) arfIntegralErro = 1.0 / arfKi;
if ((arfKi * arfIntegralErro) < -1.0) arfIntegralErro = -1.0 / arfKi;
convByte(9, saidaArf); // Comando Profundor

// CONTROLE DE ROLAMENTO/
// Ângulo de rolamento anterior em graus para cálculo de derivada.
rolAnterior = rolAtual;
// Ângulo de rolamento atual em graus fornecido pelo X-Plane.
rolAtual = convFloat(157);
// Derivada do ângulo de rolamento em graus/segundo
rolDerivada = (rolAtual - rolAnterior) / segundos;
// Erro anterior do ângulo de rolamento em graus para cálculo de derivada e de integral do erro.
rolErroAnterior = rolErroAtual;
// Filtro referência
parcelaIntegralSpRol = (spRolamento * (1.0 - rolB) - (integralSpRol * rolKi / rolKp)) * segundos;
integralSpRol = integralSpRol + parcelaIntegralSpRol;
spRolamentoISA = spRolamento * rolB + integralSpRol * rolKi / rolKp;
// Erro atual do ângulo de rolamento em graus
rolErroAtual = spRolamentoISA - rolAtual;
// Derivada do erro no tempo em graus/segundos.
rolDerivadaErro = (rolErroAtual - rolErroAnterior) / segundos;
// Parcela final da integral do erro no tempo em graus*segundos.
parcelaIntegralRol = ((rolErroAnterior + rolErroAtual) / 2.0) * segundos;
fuzzy->setInput(1, rolErroAtual * rolKe);
fuzzy->setInput(2, rolDerivadaErro * rolKec);
fuzzy->fuzzify();
// Comando servo Aileron com conversão na saída
saidaRol = ((rolKp + rolKpr * fuzzy->defuzzify(1)) * rolErroAtual
            + (rolKi + rolKir * fuzzy->defuzzify(1)) * (rolIntegralErro + parcelaIntegralRol)
            + (rolKd + rolKdr * fuzzy->defuzzify(2)) * rolDerivadaErro - rolKrg * rolDerivada)
            / 100.0;
convByte(13, saidaRol); // Comando Aileron
// Anti wind-up
if (saidaRol > -1 && saidaRol < 1) {
// Integral do erro no tempo em graus*segundos.
rolIntegralErro = rolIntegralErro + parcelaIntegralRol;
}
// Limitar a integral na interrupção do dados
if ((rolKi * rolIntegralErro) > 100.0) rolIntegralErro = 100.0 / rolKi;
if ((rolKi * rolIntegralErro) < -100.0) rolIntegralErro = -100.0 / rolKi;

// CONTROLE DE GUINADA/
// Se spGuinada >= 0 então controlar o leme
if (spGuinada >= 0) {
// Ângulo de guinada anterior em graus para cálculo de derivada.
guiAnterior = guiAtual;
// Ângulo de guinada em graus fornecido pelo X-Plane
guiAtual = convFloat(161);
// Filtro referência
if (spGuinada - spGuinadaISA < 180 && spGuinada - spGuinadaISA > -180) {

```

```

    parcelaIntegralSpGui = (spGuinada * (1.0 - guiB)
        - (integralSpGui * guiKi / guiKp)) * segundos;
    integralSpGui = integralSpGui + parcelaIntegralSpGui;
    spGuinadaISA = spGuinada * guiB + integralSpGui * guiKi / guiKp;
}
else spGuinadaISA = spGuinada;
if (abs(spGuinada - guiAtual) > abs(spGuinadaISA - guiAtual)) spGuinadaISA = spGuinada;
// Erro anterior do ângulo de guinada em graus para cálculo da derivada e da integral.
guiErroAnterior = guiErroAtual;
// Erro atual do ângulo de guinada em graus
guiErroAtual = (spGuinadaISA - guiAtual);
if ((guiAtual - guiAnterior) < 180 && (guiAtual - guiAnterior) > -180)
    // Derivada do ângulo de guinada em graus/segundo
    guiDerivada = (guiAtual - guiAnterior) / segundos;
if (guiErroAtual > 180.0) guiErroAtual = guiErroAtual - 360;
if (guiErroAtual < -180.0) guiErroAtual = guiErroAtual + 360;
// Derivada do erro no tempo em graus/segundos.
guiDerivadaErro = (guiErroAtual - guiErroAnterior) / segundos;
// Parcela final da integral do erro no tempo em graus*segundos.
parcelaIntegralGui = ((guiErroAnterior + guiErroAtual) / 2.0) * segundos;
fuzzy->setInput(1, guiErroAtual * guiKe);
fuzzy->setInput(2, guiDerivadaErro * guiKec);
fuzzy->fuzzify();
// Comando servo Leme
saidaGui = ((guiKp + guiKpr * fuzzy->defuzzify(1)) * guiErroAtual
    + (guiKi + guiKir * fuzzy->defuzzify(1)) * (guiIntegralErro + parcelaIntegralGui)
    + (guiKd + guiKdr * fuzzy->defuzzify(2)) * guiDerivadaErro - guiKrg * guiDerivada)
    / 10.0;
convByte(17, saidaGui + disturbio);
// Anti wind-up.
if (saidaGui > -1 && saidaGui < 1) {
    // Integral do erro no tempo em graus*segundos.
    guiIntegralErro = guiIntegralErro + parcelaIntegralGui;
}
if ((guiKi * guiIntegralErro) > 10) guiIntegralErro = 10.0 / guiKi;
if ((guiKi * guiIntegralErro) < -10) guiIntegralErro = -10.0 / guiKi;
}
else convByte(17, 0.0); // Comando Leme

// Manter a aeronave abastecida.
convByte(117, 170); // Tanque de combustível 1
convByte(121, 170); // Tanque de combustível 2

// Envio de dados ao X-Plane
Udp.beginPacket(remIp, remPort);
Udp.write(writeBuffer, 149);
Udp.endPacket();

// Envio de dados ao Simulink
UdpGroundStation.beginPacket(groundStationIp, remPortGroundStation);
UdpGroundStation.write(writeBufferGroundStation, 4);
UdpGroundStation.endPacket();
}
} // Fim Main Loop

```

```

void buildFuzzy() {
    // FuzzyInput
    FuzzyInput* error = new FuzzyInput(1);
    error->addFuzzySet(eNB);
    error->addFuzzySet(eNM);
    error->addFuzzySet(eNS);
    error->addFuzzySet(eZO);
    error->addFuzzySet(ePS);
    error->addFuzzySet(ePM);
    error->addFuzzySet(ePB);

    fuzzy->addFuzzyInput(error);

    // FuzzyInput
    FuzzyInput* errorChange = new FuzzyInput(2);
    errorChange->addFuzzySet(ecNB);
    errorChange->addFuzzySet(ecNM);
    errorChange->addFuzzySet(ecNS);
    errorChange->addFuzzySet(ecZO);
    errorChange->addFuzzySet(ecPS);
    errorChange->addFuzzySet(ecPM);
    errorChange->addFuzzySet(ecPB);

    fuzzy->addFuzzyInput(errorChange);

    // FuzzyOutput
    FuzzyOutput* kpki = new FuzzyOutput(1);
    kpki->addFuzzySet(Skpki);
    kpki->addFuzzySet(Bkpki);

    fuzzy->addFuzzyOutput(kpki);

    // FuzzyOutput
    FuzzyOutput* kd = new FuzzyOutput(2);
    kd->addFuzzySet(Skd);
    kd->addFuzzySet(Bkd);

    fuzzy->addFuzzyOutput(kd);

    // FuzzyRules Consequents

    FuzzyRuleConsequent* thenKpkiSKdB = new FuzzyRuleConsequent();
    thenKpkiSKdB->addOutput(Skpki);
    thenKpkiSKdB->addOutput(Bkd);

    FuzzyRuleConsequent* thenKpkiBKdS = new FuzzyRuleConsequent();
    thenKpkiBKdS->addOutput(Bkpki);
    thenKpkiBKdS->addOutput(Skd);

    FuzzyRuleConsequent* thenKpkiBKdB = new FuzzyRuleConsequent();
    thenKpkiBKdB->addOutput(Bkpki);
    thenKpkiBKdB->addOutput(Bkd);
}

```

```

// Building FuzzyRule 1
FuzzyRuleAntecedent* ifENBAndEcNB = new FuzzyRuleAntecedent();
ifENBAndEcNB->joinWithAND(eNB, ecNB);

FuzzyRule* fuzzyRule1 = new FuzzyRule(1, ifENBAndEcNB, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule1);

// Building FuzzyRule 2
FuzzyRuleAntecedent* ifENBAndEcNM = new FuzzyRuleAntecedent();
ifENBAndEcNM->joinWithAND(eNB, ecNM);

FuzzyRule* fuzzyRule2 = new FuzzyRule(2, ifENBAndEcNM, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule2);

// Building FuzzyRule 3
FuzzyRuleAntecedent* ifENBAndEcNS = new FuzzyRuleAntecedent();
ifENBAndEcNS->joinWithAND(eNB, ecNS);

FuzzyRule* fuzzyRule3 = new FuzzyRule(3, ifENBAndEcNS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule3);

// Building FuzzyRule 4
FuzzyRuleAntecedent* ifENBAndEcZO = new FuzzyRuleAntecedent();
ifENBAndEcZO->joinWithAND(eNB, ecZO);

FuzzyRule* fuzzyRule4 = new FuzzyRule(4, ifENBAndEcZO, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule4);

// Building FuzzyRule 5
FuzzyRuleAntecedent* ifENBAndEcPS = new FuzzyRuleAntecedent();
ifENBAndEcPS->joinWithAND(eNB, ecPS);

FuzzyRule* fuzzyRule5 = new FuzzyRule(5, ifENBAndEcPS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule5);

// Building FuzzyRule 6
FuzzyRuleAntecedent* ifENBAndEcPM = new FuzzyRuleAntecedent();
ifENBAndEcPM->joinWithAND(eNB, ecPM);

FuzzyRule* fuzzyRule6 = new FuzzyRule(6, ifENBAndEcPM, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule6);

// Building FuzzyRule 7
FuzzyRuleAntecedent* ifENBAndEcPB = new FuzzyRuleAntecedent();
ifENBAndEcPB->joinWithAND(eNB, ecPB);

FuzzyRule* fuzzyRule7 = new FuzzyRule(7, ifENBAndEcPB, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule7);

// Building FuzzyRule 8
FuzzyRuleAntecedent* ifENMAndEcNB = new FuzzyRuleAntecedent();
ifENMAndEcNB->joinWithAND(eNM, ecNB);

FuzzyRule* fuzzyRule8 = new FuzzyRule(8, ifENMAndEcNB, thenKpkiSKdB);

```

```

fuzzy->addFuzzyRule(fuzzyRule8);

// Building FuzzyRule 9
FuzzyRuleAntecedent* ifENMAndEcNM = new FuzzyRuleAntecedent();
ifENMAndEcNM->joinWithAND(eNM, ecNM);

FuzzyRule* fuzzyRule9 = new FuzzyRule(9, ifENMAndEcNM, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule9);

// Building FuzzyRule 10
FuzzyRuleAntecedent* ifENMAndEcNS = new FuzzyRuleAntecedent();
ifENMAndEcNS->joinWithAND(eNM, ecNS);

FuzzyRule* fuzzyRule10 = new FuzzyRule(10, ifENMAndEcNS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule10);

// Building FuzzyRule 11
FuzzyRuleAntecedent* ifENMAndEcZO = new FuzzyRuleAntecedent();
ifENMAndEcZO->joinWithAND(eNM, ecZO);

FuzzyRule* fuzzyRule11 = new FuzzyRule(11, ifENMAndEcZO, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule11);

// Building FuzzyRule 12
FuzzyRuleAntecedent* ifENMAndEcPS = new FuzzyRuleAntecedent();
ifENMAndEcPS->joinWithAND(eNM, ecPS);

FuzzyRule* fuzzyRule12 = new FuzzyRule(12, ifENMAndEcPS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule12);

// Building FuzzyRule 13
FuzzyRuleAntecedent* ifENMAndEcPM = new FuzzyRuleAntecedent();
ifENMAndEcPM->joinWithAND(eNM, ecPM);

FuzzyRule* fuzzyRule13 = new FuzzyRule(13, ifENMAndEcPM, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule13);

// Building FuzzyRule 14
FuzzyRuleAntecedent* ifENMAndEcPB = new FuzzyRuleAntecedent();
ifENMAndEcPB->joinWithAND(eNM, ecPB);

FuzzyRule* fuzzyRule14 = new FuzzyRule(14, ifENMAndEcPB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule14);

// Building FuzzyRule 15
FuzzyRuleAntecedent* ifENSAndEcNB = new FuzzyRuleAntecedent();
ifENSAndEcNB->joinWithAND(eNS, ecNB);

FuzzyRule* fuzzyRule15 = new FuzzyRule(15, ifENSAndEcNB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule15);

// Building FuzzyRule 16
FuzzyRuleAntecedent* ifENSAndEcNM = new FuzzyRuleAntecedent();
ifENSAndEcNM->joinWithAND(eNS, ecNM);

```

```

FuzzyRule* fuzzyRule16 = new FuzzyRule(16, ifENSAndEcNM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule16);

// Building FuzzyRule 17
FuzzyRuleAntecedent* ifENSAndEcNS = new FuzzyRuleAntecedent();
ifENSAndEcNS->joinWithAND(eNS, ecNS);

FuzzyRule* fuzzyRule17 = new FuzzyRule(17, ifENSAndEcNS, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule17);

// Building FuzzyRule 18
FuzzyRuleAntecedent* ifENSAndEcZO = new FuzzyRuleAntecedent();
ifENSAndEcZO->joinWithAND(eNS, ecZO);

FuzzyRule* fuzzyRule18 = new FuzzyRule(18, ifENSAndEcZO, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule18);

// Building FuzzyRule 19
FuzzyRuleAntecedent* ifENSAndEcPS = new FuzzyRuleAntecedent();
ifENSAndEcPS->joinWithAND(eNS, ecPS);

FuzzyRule* fuzzyRule19 = new FuzzyRule(19, ifENSAndEcPS, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule19);

// Building FuzzyRule 20
FuzzyRuleAntecedent* ifENSAndEcPM = new FuzzyRuleAntecedent();
ifENSAndEcPM->joinWithAND(eNS, ecPM);

FuzzyRule* fuzzyRule20 = new FuzzyRule(20, ifENSAndEcPM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule20);

// Building FuzzyRule 21
FuzzyRuleAntecedent* ifENSAndEcPB = new FuzzyRuleAntecedent();
ifENSAndEcPB->joinWithAND(eNS, ecPB);

FuzzyRule* fuzzyRule21 = new FuzzyRule(21, ifENSAndEcPB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule21);

// Building FuzzyRule 22
FuzzyRuleAntecedent* ifEZOAndEcNB = new FuzzyRuleAntecedent();
ifEZOAndEcNB->joinWithAND(eZO, ecNB);

FuzzyRule* fuzzyRule22 = new FuzzyRule(22, ifEZOAndEcNB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule22);

// Building FuzzyRule 23
FuzzyRuleAntecedent* ifEZOAndEcNM = new FuzzyRuleAntecedent();
ifEZOAndEcNM->joinWithAND(eZO, ecNM);

FuzzyRule* fuzzyRule23 = new FuzzyRule(23, ifEZOAndEcNM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule23);

// Building FuzzyRule 24

```

```

FuzzyRuleAntecedent* ifEZOAndEcNS = new FuzzyRuleAntecedent();
ifEZOAndEcNS->joinWithAND(eZO, ecNS);

FuzzyRule* fuzzyRule24 = new FuzzyRule(24, ifEZOAndEcNS, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule24);

// Building FuzzyRule 25
FuzzyRuleAntecedent* ifEZOAndEcZO = new FuzzyRuleAntecedent();
ifEZOAndEcZO->joinWithAND(eZO, ecZO);

FuzzyRule* fuzzyRule25 = new FuzzyRule(25, ifEZOAndEcZO, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule25);

// Building FuzzyRule 26
FuzzyRuleAntecedent* ifEZOAndEcPS = new FuzzyRuleAntecedent();
ifEZOAndEcPS->joinWithAND(eZO, ecPS);

FuzzyRule* fuzzyRule26 = new FuzzyRule(26, ifEZOAndEcPS, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule26);

// Building FuzzyRule 27
FuzzyRuleAntecedent* ifEZOAndEcPM = new FuzzyRuleAntecedent();
ifEZOAndEcPM->joinWithAND(eZO, ecPM);

FuzzyRule* fuzzyRule27 = new FuzzyRule(27, ifEZOAndEcPM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule27);

// Building FuzzyRule 28
FuzzyRuleAntecedent* ifEZOAndEcPB = new FuzzyRuleAntecedent();
ifEZOAndEcPB->joinWithAND(eZO, ecPB);

FuzzyRule* fuzzyRule28 = new FuzzyRule(28, ifEZOAndEcPB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule28);

// Building FuzzyRule 29
FuzzyRuleAntecedent* ifEPSAndEcNB = new FuzzyRuleAntecedent();
ifEPSAndEcNB->joinWithAND(ePS, ecNB);

FuzzyRule* fuzzyRule29 = new FuzzyRule(29, ifEPSAndEcNB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule29);

// Building FuzzyRule 30
FuzzyRuleAntecedent* ifEPSAndEcNM = new FuzzyRuleAntecedent();
ifEPSAndEcNM->joinWithAND(ePS, ecNM);

FuzzyRule* fuzzyRule30 = new FuzzyRule(30, ifEPSAndEcNM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule30);

// Building FuzzyRule 31
FuzzyRuleAntecedent* ifEPSAndEcNS = new FuzzyRuleAntecedent();
ifEPSAndEcNS->joinWithAND(ePS, ecNS);

FuzzyRule* fuzzyRule31 = new FuzzyRule(31, ifEPSAndEcNS, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule31);

```

```

// Building FuzzyRule 32
FuzzyRuleAntecedent* ifEPSAndEcZ0 = new FuzzyRuleAntecedent();
ifEPSAndEcZ0->joinWithAND(ePS, ecZ0);

FuzzyRule* fuzzyRule32 = new FuzzyRule(32, ifEPSAndEcZ0, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule32);

// Building FuzzyRule 33
FuzzyRuleAntecedent* ifEPSAndEcPS = new FuzzyRuleAntecedent();
ifEPSAndEcPS->joinWithAND(ePS, ecPS);

FuzzyRule* fuzzyRule33 = new FuzzyRule(33, ifEPSAndEcPS, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule33);

// Building FuzzyRule 34
FuzzyRuleAntecedent* ifEPSAndEcPM = new FuzzyRuleAntecedent();
ifEPSAndEcPM->joinWithAND(ePS, ecPM);

FuzzyRule* fuzzyRule34 = new FuzzyRule(34, ifEPSAndEcPM, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule34);

// Building FuzzyRule 35
FuzzyRuleAntecedent* ifEPSAndEcPB = new FuzzyRuleAntecedent();
ifEPSAndEcPB->joinWithAND(ePS, ecPB);

FuzzyRule* fuzzyRule35 = new FuzzyRule(35, ifEPSAndEcPB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule35);

// Building FuzzyRule 36
FuzzyRuleAntecedent* ifEPMAndEcNB = new FuzzyRuleAntecedent();
ifEPMAndEcNB->joinWithAND(ePM, ecNB);

FuzzyRule* fuzzyRule36 = new FuzzyRule(36, ifEPMAndEcNB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule36);

// Building FuzzyRule 37
FuzzyRuleAntecedent* ifEPMAndEcNM = new FuzzyRuleAntecedent();
ifEPMAndEcNM->joinWithAND(ePM, ecNM);

FuzzyRule* fuzzyRule37 = new FuzzyRule(37, ifEPMAndEcNM, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule37);

// Building FuzzyRule 38
FuzzyRuleAntecedent* ifEPMAndEcNS = new FuzzyRuleAntecedent();
ifEPMAndEcNS->joinWithAND(ePM, ecNS);

FuzzyRule* fuzzyRule38 = new FuzzyRule(38, ifEPMAndEcNS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule38);

// Building FuzzyRule 39
FuzzyRuleAntecedent* ifEPMAndEcZ0 = new FuzzyRuleAntecedent();
ifEPMAndEcZ0->joinWithAND(ePM, ecZ0);

```

```

FuzzyRule* fuzzyRule39 = new FuzzyRule(39, ifEPMAAndEcZ0, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule39);

// Building FuzzyRule 40
FuzzyRuleAntecedent* ifEPMAAndEcPS = new FuzzyRuleAntecedent();
ifEPMAAndEcPS->joinWithAND(ePM, ecPS);

FuzzyRule* fuzzyRule40 = new FuzzyRule(40, ifEPMAAndEcPS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule40);

// Building FuzzyRule 41
FuzzyRuleAntecedent* ifEPMAAndEcPM = new FuzzyRuleAntecedent();
ifEPMAAndEcPM->joinWithAND(ePM, ecPM);

FuzzyRule* fuzzyRule41 = new FuzzyRule(41, ifEPMAAndEcPM, thenKpkiBKdB);
fuzzy->addFuzzyRule(fuzzyRule41);

// Building FuzzyRule 42
FuzzyRuleAntecedent* ifEPMAAndEcPB = new FuzzyRuleAntecedent();
ifEPMAAndEcPB->joinWithAND(ePM, ecPB);

FuzzyRule* fuzzyRule42 = new FuzzyRule(42, ifEPMAAndEcPB, thenKpkiSKdB);
fuzzy->addFuzzyRule(fuzzyRule42);

// Building FuzzyRule 43
FuzzyRuleAntecedent* ifEPBAndEcNB = new FuzzyRuleAntecedent();
ifEPBAndEcNB->joinWithAND(ePB, ecNB);

FuzzyRule* fuzzyRule43 = new FuzzyRule(43, ifEPBAndEcNB, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule43);

// Building FuzzyRule 44
FuzzyRuleAntecedent* ifEPBAndEcNM = new FuzzyRuleAntecedent();
ifEPBAndEcNM->joinWithAND(ePB, ecNM);

FuzzyRule* fuzzyRule44 = new FuzzyRule(44, ifEPBAndEcNM, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule44);

// Building FuzzyRule 45
FuzzyRuleAntecedent* ifEPBAndEcNS = new FuzzyRuleAntecedent();
ifEPBAndEcNS->joinWithAND(ePB, ecNS);

FuzzyRule* fuzzyRule45 = new FuzzyRule(45, ifEPBAndEcNS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule45);

// Building FuzzyRule 46
FuzzyRuleAntecedent* ifEPBAndEcZ0 = new FuzzyRuleAntecedent();
ifEPBAndEcZ0->joinWithAND(ePB, ecZ0);

FuzzyRule* fuzzyRule46 = new FuzzyRule(46, ifEPBAndEcZ0, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule46);

// Building FuzzyRule 47
FuzzyRuleAntecedent* ifEPBAndEcPS = new FuzzyRuleAntecedent();

```

```

ifEPBAndEcPS->joinWithAND(ePB, ecPS);

FuzzyRule* fuzzyRule47 = new FuzzyRule(47, ifEPBAndEcPS, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule47);

// Building FuzzyRule 48
FuzzyRuleAntecedent* ifEPBAndEcPM = new FuzzyRuleAntecedent();
ifEPBAndEcPM->joinWithAND(ePB, ecPM);

FuzzyRule* fuzzyRule48 = new FuzzyRule(48, ifEPBAndEcPM, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule48);

// Building FuzzyRule 49
FuzzyRuleAntecedent* ifEPBAndEcPB = new FuzzyRuleAntecedent();
ifEPBAndEcPB->joinWithAND(ePB, ecPB);

FuzzyRule* fuzzyRule49 = new FuzzyRule(49, ifEPBAndEcPB, thenKpkiBKdS);
fuzzy->addFuzzyRule(fuzzyRule49);

}

float convFloat(byte index) {
    int i;
    float valor = 0.0;
    long fracao, aux = 0.0;
    byte sinal = 0, expoente = 0;
    for (i = 3; i >= 0; i--) {
        aux = aux | (readBuffer[index + i] & 0x000000FF);
        if (i > 0) aux = aux << 8;
    }
    sinal = aux >> 31;
    expoente = (aux >> 23) & 0xFF;
    fracao = aux & 0x007FFFFFFF;
    for (i = 1; i < 24; i++) {
        if (bitRead(fracao, 23 - (i - 1))) {
            valor = valor + pow(2.0, -(i - 1));
        }
    }
    return (pow(-1, sinal) * (1.0 + valor)) * pow(2, (expoente - 127));
}

void convByte(byte index, float valor) {
    int i = 0, j = 6, f = 2;
    byte sinal = 0, expoente = 0;
    if (valor < 0.0) {
        sinal = 1;
        valor = (-1.0) * valor;
    }
    if (valor >= 1) while (valor >= 2) {
        valor = valor / 2;
        expoente++;
    }
    else while ((valor > 0) && (valor < 1)) {
        valor = valor * 2;

```

```
    expoente--;
  }
  if (valor != 0.0) expoente = expoente + 127;
  writeBuffer[index + 3] = sinal << 7;
  writeBuffer[index + 3] = writeBuffer[index + 3] | (expoente >> 1);
  expoente = expoente << 7;
  writeBuffer[index + 2] = expoente;
  if (valor >= 1.0) valor = valor - 1.0;
  for (i = 0; i < 23; i++) {
    valor = valor * 2.0;
    if (valor >= 1.0) {
      valor = valor - 1.0;
      bitWrite(writeBuffer[index + f], j, 1);
    }
    else bitWrite(writeBuffer[index + f], j, 0);
    j--;
    if (j < 0) {
      j = 7;
      f--;
    }
  }
}
```