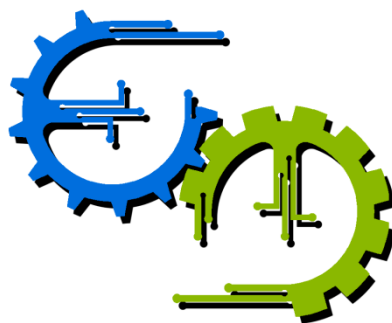


TRABALHO DE GRADUAÇÃO

**SISTEMA DE CONTROLE DE ATITUDE PARA
FOGUETE DE PEQUENO PORTE**

Por,
Marconi Campos Ferreira Porto

Brasília, Julho de 2017



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

SISTEMA DE CONTROLE DE ATITUDE PARA FOGUETE DE PEQUENO PORTE

POR,

Marconi Campos Ferreira Porto

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Alex da Rosa, UnB/ ENE (Orientador)

Prof. Sébastien R.M.J. Rondineau, UnB/ FGA

Prof. Renato A. Borges, UnB/ ENE

Prof. Paolo Gessini, UnB/ FGA

Brasília, Julho de 2017

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

CP853s Campos Ferreira Porto, Marconi
Sistema de Controle de Atitude para Foguete de Pequeno Porte / Marconi Campos Ferreira Porto; orientador Alex da Rosa; co-orientador Sébastien Roland Marie Joseph Rondineau. -- Brasília, 2017. 180 p.

Monografia (Graduação - Engenharia de Controle e Automação) -- Universidade de Brasília, 2017.

1. Controle de foguetes. 2. Controle de atitude. 3. Controle de arfagem. 4. Controlador de ganhos programados. 5. Controlador PID. I. da Rosa, Alex, orient. II. Roland Marie Joseph Rondineau, Sébastien, co-orient. III. Título.

REFERÊNCIA BIBLIOGRÁFICA

PORTO, M. C. F., (2017). Sistema de Controle de Atitude para Foguete de Pequeno Porte. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 005/2017, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 123p.

CESSÃO DE DIREITOS

AUTOR: Marconi Campos Ferreira Porto.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Sistema de Controle de Atitude para Foguete de Pequeno Porte.

GRAU: Engenheiro ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Marconi Campos Ferreira Porto
SQSW 102 bloco F ap 103 – Setor Sudoeste.
70670-206 Brasília – DF – Brasil.

AGRADECIMENTOS

Agradeço aos parentes, amigos e professores que me apoiaram quando necessário durante todo o curso.

Marconi Campos Ferreira Porto.

RESUMO

O trabalho consiste no projeto de um controlador de ganhos programados do ângulo de arfagem do foguete LILE-2 da Hybrid Propulsion Team.

A Hybrid Propulsion Team é uma equipe de alunos de várias engenharias da Universidade de Brasília dos campi Darcy Ribeiro e Gama. Tem como objetivo produzir um foguete de pequeno porte para um concurso (N-Prize) promovido pela Universidade de Cambridge (Reino Unido) com a intenção de colocar um pequeno satélite em órbita que dê três voltas em torno da Terra. Já foram testados quatro foguetes pela equipe. O nome da equipe (propulsão híbrida, em português) dá-se devido ao fato de se usar propelentes em estados diferentes: combustível sólido (parafina) e oxidante líquido (óxido nitroso – N_2O). No início das pesquisas com propelentes híbridos, houve muita relutância em seu uso por pesquisadores, mas hoje é largamente usado.

Na área de Dinâmica de Vôo, o foguete possui três eixos que rotacionam nos ângulos Pitch (arfagem), Yaw (guinada), Roll (rolagem). A orientação desses três ângulos definem a Atitude do foguete. Para essa aeronave o projeto de controle dos ângulos de arfagem e guinada são idênticos, enquanto que o ângulo de rolagem necessita de um projeto diferente. Os controles do três ângulos podem ser feitos independentemente um do outro. O Pitch é considerado o ângulo mais relevante visto ser o usado para se calcular a trajetória de vôo.

O ângulo de ataque, pequena perturbação que altera a orientação do vetor velocidade, será controlado pela mudança na orientação do jato de combustão. O ângulo de ataque será discretizado em 20 amostras e para cada amostra serão calculadas numericamente três derivadas de estabilidade (parâmetros aerodinâmicos) que serão adicionadas aos outros parâmetros do sistema de controle que não variam com o ângulo de ataque. Desse modo, serão 20 conjuntos de parâmetros e será projetado um sistema de controle contínuo para cada conjunto. Em seguida, será usado um simulador computacional (MATLAB) para validar o modelo. Nisso consiste o controlador de ganhos programados.

Palavras Chave: controlador de ganhos programados, pitch, derivadas de estabilidade, ângulo de ataque.

ABSTRACT

The work consist of the design of a programmed gain controller of the Pitch of the LILE-2 rocket of the *Hybrid Propulsion Team*.

The Hybrid Propulsion Team is a team of students of various engineering at the University of Brasilia of the Darcy Ribeiro and Gama campuses. It aims to produce a small rocket for a competition (N-Prize) promoted by the University of Cambridge (United Kingdom) with the intention of putting a small satellite in orbit that turns three round the Earth. Four rockets have already been tested by the team. The name of the team (hybrid propulsion, Portuguese) is due to the use of propellants in different states: solid fuel (paraffin) and liquid oxidant (nitrous oxide – N_2O). At the beginning of research with hybrid propellants, there was much reluctance in its use by researchers, but today it is widely used.

In the area of Flight Dynamics, the rocket has three axes that rotate in the Pitch, Yaw, Roll angles. The orientation of these three angles defines the Attitude of the rocket. For this rocket the design of control of the pitch and yaw angles are identical, while the roll angle requires a different design. The controls at the three angles can be made independently of one another. Pitch is considered the most relevant angle since it is used to calculate the flight trajectory.

The angle of attack, small perturbation that changes the orientation of the velocity vector, will be controlled by the change in orientation of the combustion jet. The angle of attack will be discretized in 18 samples and for each sample will be numerically calculated three stability derivatives (aerodynamic parameters) that will be added to the other parameters of the control system that do not vary with the angle of attack. In this way, there will be 18 sets of parameters and a continuous control system will be designed for each set. Next, a computer simulator (MATLAB) will be used to validate the model. This is the programmed gain controller.

Keywords: programmed gain controller, pitch, stability derivatives, angle of attack

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	SOFTWARES UTILIZADOS	2
1.3	DEFINIÇÃO DO PROBLEMA	3
1.4	OBJETIVOS DO TRABALHO	3
1.5	RESULTADOS ALCANÇADOS	3
1.6	APRESENTAÇÃO DO MANUSCRITO	4
2	REVISÃO BIBLIOGRÁFICA	5
2.1	FUNÇÕES MATEMÁTICAS	5
2.1.1	Função degrau unitário	5
2.1.2	Função decaimento exponencial	5
2.2	MODELOS MATEMÁTICOS DE UM SISTEMA FÍSICO	6
2.3	AERODINÂMICA BÁSICA	10
2.4	DINÂMICA DE VÔO	11
2.5	CONTROLE DE ATITUDE DE FOGUETES	13
2.6	CONTROLADOR PID	15
2.7	CONTROLADOR DE GANHOS PROGRAMADOS	16
3	MODELAGEM MATEMÁTICA	17
3.1	PLANTA DO FOGUETE	17
3.2	MODELO MATEMÁTICO DO CONTROLADOR PID	18
4	DESENVOLVIMENTO	19
4.1	PROCEDIMENTO DE PROJETO	19
4.2	O SISTEMA FÍSICO DA PLANTA	19
4.3	DESENHO MECÂNICO DA FUSELAGEM DO LILE-2	19
4.4	PARÂMETROS DA PLANTA	20
4.5	PLANTA DO LILE-2	21
4.6	CONTROLADOR PID E CONTROLADOR DE GANHOS PROGRAMADOS	21
4.7	TESTES NO SISTEMA COMPENSADO	22
5	RESULTADOS	23
5.1	DESENHO MECÂNICO DA FUSELAGEM DO LILE-2	23
5.2	PARÂMETROS DA PLANTA	24
5.3	PLANTA DO LILE-2, DETERMINAÇÃO DOS PARÂMETROS DO CONTROLADOR PID, SISTEMA COMPENSADO E TESTES	30
5.3.1	Simulação para $\alpha = -4,5^\circ$	30
5.3.2	Simulação para $\alpha = -4,0^\circ$	33
5.3.3	Simulação para $\alpha = 0^\circ$	34
5.3.4	Simulação para $\alpha = 1,0^\circ$	35
5.3.5	Simulação para $\alpha = 4,5^\circ$	36
5.3.6	Simulação da queima de combustível	37
5.4	PROGRAMAS PARA O CONTROLADOR DE GANHOS PROGRAMADOS	38
6	ANÁLISE DOS RESULTADOS	49
6.1	DESENHO MECÂNICO DA FUSELAGEM	49
6.2	PARÂMETROS DA PLANTA	49
6.3	PLANTA DO LILE-2, DETERMINAÇÃO DOS PARÂMETROS DO CONTROLADOR PID, SISTEMA COMPENSADO E TESTES	49
6.4	O CONTROLADOR DE GANHOS PROGRAMADOS	50
7	CONCLUSÕES	51
7.1	CONCLUSÕES	51
7.2	TRABALHOS FUTUROS	52
REFERÊNCIAS BIBLIOGRÁFICAS		54
ANEXOS		55
I	Simulação de vôo para o LILE-2	55
II	Simulações para os valores discretizados do ângulo de ataque	55
II.1	Simulação para $\alpha = -4,5^\circ$	55
II.2	Simulação para $\alpha = -4,0^\circ$	61
II.3	Simulação para $\alpha = -3,5^\circ$	67
II.4	Simulação para $\alpha = -3,0^\circ$	73
II.5	Simulação para $\alpha = -2,5^\circ$	79
II.6	Simulação para $\alpha = -2,0^\circ$	85

II.7	Simulação para $\alpha = -1,5^\circ$	91
II.8	Simulação para $\alpha = -0,8^\circ$	97
II.9	Simulação para $\alpha = 0^\circ$	103
II.10	Simulação para $\alpha = 0,5^\circ$	109
II.11	Simulação para $\alpha = 1,0^\circ$	115
II.12	Simulação para $\alpha = 1,5^\circ$	121
II.13	Simulação para $\alpha = 2,0^\circ$	127
II.14	Simulação para $\alpha = 2,5^\circ$	133
II.15	Simulação para $\alpha = 3,0^\circ$	139
II.16	Simulação para $\alpha = 3,5^\circ$	145
II.17	Simulação para $\alpha = 4,0^\circ$	151
II.18	Simulação para $\alpha = 4,5^\circ$	157
III	Códigos usados e dados gerados na simulação de queima de combustível.	163
IV	Códigos usados para a implementação do controlador de ganhos programados e gráficos de teste	172

LISTA DE FIGURAS

Figura 1.1. Símbolo da Hybrid Propulsion Team.	1
Figura 1.2. Bancada de teste de motor híbrido.	2
Figura 2.1. Gráfico da função degrau unitário.	5
Figura 2.2. Gráfico do decaimento exponencial.	6
Figura 2.3 Circuito elétrico.	8
Figura 2.4. Configurações da força de Sustentação.	10
Figura 2.5. Forças e momentos.	11
Figura 2.6. Variáveis para o controle de atitude.	14
Figura 2.7. Diagrama de blocos.	15
Figura 4.1. Sistema físico da planta.	19
Figura 5.1. Vista em perspectiva.	23
Figura 5.2. Vista lateral.	23
Figura 5.3. Vista traseira.	24
Figura 5.4. Arquivo de entrada para Digital Datcom.	28
Figura 5.5. Dados do arquivo de saída (1ª parte).	29
Figura 5.6. Dados do arquivo de saída (2ª parte).	29
Figura 5.7. Dados de simulação.	31
Figura 5.8. Dados de simulação.	32
Figura 5.9. Gráfico da saída para a entrada degrau unitário.	32
Figura 5.10. Dados de simulação.	33
Figura 5.11. Gráfico da saída para entrada decaimento exponencial.	33
Figura 5.12. Dados de simulação.	34
Figura 5.13. Dados de simulação.	35
Figura 5.14. Dados de simulação.	36
Figura 5.15. Dados de simulação.	37
Figura 5.16. Curvas para entrada degrau unitário.	37
Figura 5.17. Curvas para entrada decaimento exponencial.	38
Figura 5.18. Dados de simulação.	40
Figura 5.19. Gráfico da saída para entrada degrau unitário.	41
Figura 5.20. Dados de simulação.	42
Figura 5.21. Gráfico da saída para entrada decaimento exponencial.	43
Figura 5.22. Interpolação de $CZ\alpha$	44
Figura 5.23. Interpolação de $CM\alpha$	44
Figura 5.24. Dados de simulação.	46
Figura 5.25. Dados de simulação.	46
Figura 5.26. Dados de simulação.	48
Figura 5.27. Dados de simulação.	48
Figura I.1. Altura, velocidade e aceleração.	55
Figura II.1. Programa em MATLAB.	57
Figura II.2. Dados de simulação.	58
Figura II.3. Dados de simulação.	59
Figura II.4. Gráfico da saída para a entrada degrau unitário.	59
Figura II.5. Programa em MATLAB.	60
Figura II.6. Dados de simulação.	60
Figura II.7. Gráfico da saída para a entrada decaimento exponencial.	61
Figura II.8. Programa em MATLAB.	63
Figura II.9. Dados de simulação.	64
Figura II.10. Dados de simulação.	65
Figura II.11. Gráfico da saída para a entrada degrau.	65
Figura II.12. Programa em MATLAB.	66
Figura II.13. Dados de simulação.	66
Figura II.14. Gráfico da saída para entrada decaimento exponencial.	67
Figura II.15. Programa em MATLAB.	69

Figura II.16. Dados de simulação.....	70
Figura II.17. Dados de simulação.....	71
Figura II.18. Gráfico da saída para entrada degrau unitário.....	71
Figura II.19. Programa em MATLAB.....	72
Figura II.20. Dados de simulação.....	72
Figura II.21. Gráfico da saída para entrada decaimento exponencial.....	73
Figura II.22. Programa em MATLAB.....	75
Figura II.23. Dados de simulação.....	76
Figura II.24. Dados de simulação.....	77
Figura II.25. Gráfico da saída para entrada degrau.....	77
Figura II.26. Programa em MATLAB.....	78
Figura II.27. Dados de simulação.....	78
Figura II.28. Gráfico da saída para a entrada degrau unitário.....	79
Figura II.29. Programa em MATLAB.....	81
Figura II.30. Dados de simulação.....	82
Figura II.31. Dados de simulação.....	83
Figura II.32. Gráfico da saída para entrada degrau unitário.....	83
Figura II.33. Programa em MATLAB.....	84
Figura II.34. Dados de simulação.....	84
Figura II.35. Gráfico da saída para entrada decaimento exponencial.....	85
Figura II.36. Programa em MATLAB.....	87
Figura II.37. Dados de simulação.....	88
Figura II.38. Dados de simulação.....	89
Figura II.39. Gráfico da saída para entrada degrau unitário.....	89
Figura II.40. Programa em MATLAB.....	90
Figura II.41. Dados de simulação.....	90
Figura II.42. Gráfico da saída para a entrada decaimento exponencial.....	91
Figura II.43. Programa para MATLAB.....	93
Figura II.44. Dados de simulação.....	94
Figura II.45. Dados de simulação.....	95
Figura II.46. Gráfico da saída para entrada degrau unitário.....	95
Figura II.47. Programa em MATLAB.....	96
Figura II.48. Dados de simulação.....	96
Figura II.49. Gráfico da saída para entrada decaimento exponencial.....	97
Figura II.50. Programa para MATLAB.....	99
Figura II.51. Dados de simulação.....	100
Figura II.52. Dados de simulação.....	101
Figura II.53. Gráfico da saída para entrada degrau.....	101
Figura II.54. Programa para MATLAB.....	102
Figura II.55. Dados de simulação.....	102
Figura II.56. Gráfico da saída para entrada decaimento exponencial.....	103
Figura II.57. Programa para MATLAB.....	105
Figura II.58. Dados de simulação.....	106
Figura II.59. Dados de simulação.....	107
Figura II.60. Gráfico da saída para entrada degrau unitário.....	107
Figura II.61. Programa para MATLAB.....	108
Figura II.62. Dados de simulação.....	108
Figura II.63. Gráfico da saída para a entrada decaimento exponencial.....	109
Figura II.64. Programa para MATLAB.....	111
Figura II.65. Dados de simulação.....	112
Figura II.66. Dados de simulação.....	113
Figura II.67. Gráfico da saída para a entrada degrau.....	113
Figura II.68. Programa para MATLAB.....	114
Figura II.69. Dados de simulação.....	114
Figura II.70. Gráfico da saída para entrada decaimento exponencial.....	115
Figura II.71. Programa para MATLAB.....	117

Figura II.72. Dados de simulação.	118
Figura II.73. Dados de simulação.	119
Figura II.74. Gráfico da saída para entrada degrau unitário.....	119
Figura II.75. Programa para MATLAB.	120
Figura II.76. Dados de simulação.	120
Figura II.77. Gráfico da saída para decaimento exponencial.	121
Figura II.78. Programa para MATLAB.	123
Figura II.79. Dados de simulação.	124
Figura II.80. Dados de simulação.	125
Figura II.81. Gráfico da saída para a entrada degrau unitário.....	125
Figura II.82. Programa para MATLAB.	126
Figura II.83. Dados de simulação.	126
Figura II.84. Gráfico da saída para entrada decaimento exponencial.	127
Figura II.85. Programa para MATLAB.	129
Figura II.86. Dados de simulação.	130
Figura II.87. Dados de simulação.	131
Figura II.88. Gráfico da saída para entrada degrau unitário.....	131
Figura II.89. Programa para MATLAB.	132
Figura II.90. Dados de simulação.	132
Figura II.91. Gráfico da saída para a entrada decaimento exponencial.	133
Figura II.92. Programa para MATLAB.	135
Figura II.93. Dados de simulação.	136
Figura II.94. Dados de simulação.	137
Figura II.95. Gráfico da saída para entrada degrau unitário.....	137
Figura II.96. Programa para MATLAB.	138
Figura II.97. Dados de simulação.	138
Figura II.98. Gráfico da saída para entrada decaimento exponencial.	139
Figura II.99. Programa para MATLAB.	141
Figura II.100. Dados de simulação.	142
Figura II.101. Dados de simulação.	143
Figura II.102. Gráfico da saída para entrada degrau unitário.....	143
Figura II.103. Programa para MATLAB.	144
Figura II.104. Dados de simulação.	144
Figura II.105. Gráfico da saída para entrada decaimento exponencial.	145
Figura II.106. Programa para MATLAB.	147
Figura II.107. Dados de simulação.	148
Figura II.108. Dados de simulação.	149
Figura II.109. Gráfico da saída para entrada degrau.	149
Figura II.110. Programa para MATLAB.	150
Figura II.111. Dados de simulação.	150
Figura II.112. Gráfico da saída para a entrada decaimento exponencial.	151
Figura II.113. Programa para MATLAB.	153
Figura II.114. Dados de simulação.	154
Figura II.115. Dados de simulação.	155
Figura II.116. Gráfico da saída para entrada degrau unitário.....	155
Figura II.117. Programa para MATLAB.	156
Figura II.118. Dados de simulação.	156
Figura II.119. Gráfico da saída para entrada decaimento exponencial.	157
Figura II.120. Programa para MATLAB.	159
Figura II.121. Dados de simulação.	160
Figura II.122. Dados de simulação.	161
Figura II.123. Gráfico da saída para entrada degrau unitário.....	161
Figura II.124. Programa para MATLAB.	162
Figura II.125. Dados de simulação.	162
Figura II.126. Gráfico da saída para entrada decaimento unitário.	163
Figura III.1. Código para o cálculo das curvas no mesmo gráfico.	164

Figura III.2. Còdigo para o cálculo dos dados.....	165
Figura III.3. Dados sobre a saída para a entrada degrau unitário. Valores máximo e mínimo, valor final e sobrevalor percentual para cada uma das 9 curvas.	170
Figura III.4. Dados sobre a curva para a entrada decaimento exponencial. Valor final para cada uma das 9 curvas.	172
Figura IV.1. Còdigo principal para o controlador de ganhos programados por chaveamento.	174
Figura IV.2. Função FunTran.m para o controlador de ganhos programados por chaveamento.....	174
Figura IV.3. Função saída.m para o controlador de ganhos programados por chaveamento.	175
Figura IV.4. Còdigo principal para o controlador de ganhos programados por interpolação.	175
Figura IV.5. Função FunTranConti.m para o controlador de ganhos programados por interpolação.....	176
Figura IV.6. Função Cza.m para o controlador de ganhos programados por interpolação..	177
Figura IV.7. Função Cma.m para o controlador de ganhos programados por interpolação.	177
Figura IV.8. Função saída.m para o controlador de ganhos programados por interpolação.	178
Figura IV.9. Gráfico da saída para $\alpha = 1,852^\circ$ e entrada degrau unitário.....	178
Figura IV.10. Gráfico da saída para $\alpha = 1,852^\circ$ e entrada decaimento exponencial.	179
Figura IV.11. Gráfico da saída para $\alpha = -4,279^\circ$ e entrada degrau unitário.	179
Figura IV.12. Gráfico da saída para $\alpha = -4,279^\circ$ e entrada decaimento exponencial.....	180

LISTA DE SÍMBOLOS

Símbolos Latinos

c	Comprimento do foguete	[m]
h_c	Comprimento do corpo do foguete	[m]
h_n	Comprimento do nariz do foguete	[m]
J_{yy_n}	Momento de inércia do nariz do foguete em torno de y	[kg.m ²]
J_{yy_c}	Momento de inércia do corpo do foguete em torno de y	[kg.m ²]
J_{yy}	Momento de inércia em torno do eixo y	[kg.m ²]
l	Distância entre o CP e o CG	[m]
m	Massa do foguete	[kg]
m_c	Massa do corpo do foguete	[m]
m_n	Massa do nariz do foguete	[m]
M_α	Derivada de estabilidade dimensional	[kg/m.s]
M_q	Derivada de estabilidade dimensional	[kg/s]
q	Velocidade de rotação do eixo y	[rad/s]
r	Raio da secção transversal do foguete	[m]
S	Área da secção transversal	[m ²]
t	Tempo	[s]
T	Empuxo	[N]
v	Velocidade média do foguete	[m/s]
$v(t)$	Função velocidade do foguete	[m/s]
V_c	Volume do corpo do foguete	[m]
V_n	Volume do nariz do foguete	[m ³]
x'_{CG}	Posição do CG em relação À ponta do foguete	[m]
Z_α	Derivada de estabilidade dimensional	[kg/m ² .s]

Símbolos Gregos

α	Ângulo de ataque	[rad]
μ_0	Velocidade média do foguete	[m/s]
μ_1	Ângulo de controle	[rad]
ρ	Densidade média do ar atmosférico	[kg/m ³]
$\rho(h)$	Densidade do ar atmosférico em função da altitude	[kg/m ³]
ρ_c	Massa específica do PVC	[kg/m ³]
ρ_n	Massa específica do Nylon 6.0	[kg/m ³]

\bar{z}	Distância entre a posição do CG e a base do foguete	[m]
\bar{z}_n	Distância entre a posição dos sensores e o CG	[m]
Δa_z^0	Variação da velocidade no eixo z sentida pelo acelerômetro	[m/s ²]

Grupos Adimensionais

a	Parâmetro do controlador PID
A	Matriz do sistema
B	Matriz de entradas
C	Matriz de saídas
$C_{Y\beta}$	Derivada de estabilidade não-dimensional
$C_{z\alpha}$	Derivada de estabilidade não-dimensional
$C_{M\alpha}$	Derivada de estabilidade não-dimensional
C_{Mq}	Derivada de estabilidade não-dimensional
D	Matriz de realimentação
e^{-t}	Função decaimento exponencial
K	Ganho do controlador PID
u	Vetor de entradas
$u(t)$	Função degrau unitário
x	Vetor de estado
\dot{x}	Derivada do vetor de estado em relação ao tempo
y	Vetor de saídas

Subscritos

n	nariz do foguete
y	eixo y
z	eixo z

Sobrescritos

\cdot	Variação temporal
0	Posição do sensor acelerômetro

Siglas

CG	Centro de gravidade
CP	Centro de pressão

IMU	<i>Inertial mass unit</i>
MEMS	<i>Micro electro mechanical system</i>
PID	Proporcional-integral-derivativo

1 INTRODUÇÃO

Este capítulo tem por finalidade introduzir o contexto em que o projeto está envolvido e apresentar ferramentas utilizadas, o problema em si e as partes que compõem o texto.

1.1 CONTEXTUALIZAÇÃO

O presente trabalho insere-se no desenvolvimento de um projeto da *Hybrid Propulsion Team* (Fig. 1.1), uma equipe de alunos de vários cursos de engenharia da Universidade de Brasília dos campi Darcy Ribeiro e Gama, como Engenharia Mecatrônica, Engenharia Aeroespacial, Engenharia Elétrica e Engenharia Mecânica. A finalidade da equipe é participar de competições na área de foguetes e desenvolver o conhecimento e a experiência dos participantes em várias áreas. A principal competição na qual está inserida, e que motivou o início deste trabalho, é o *N-Prize* promovido pela Universidade de Cambridge (Reino Unido). O objetivo da competição é premiar a primeira universidade que colocar um pequeno satélite em órbita que dê três voltas em torno da Terra por um baixo custo. Ela já decorre há 3292 dias sem uma equipe vencedora.



Figura 1.1. Símbolo da *Hybrid Propulsion Team*.

Para se chegar ao modelo final do foguete, que colocará o satélite em órbita, a equipe propôs uma concepção de foguetes em ordem crescente de tamanho e complexidade. Foram criadas bancadas de teste de motor, onde motores foram testados (Fig. 1.2), e quatro foguetes com apenas fuselagem e motor foram testados. Para o próximo teste, Foguete LILE-2, está previsto o uso de um sistema de controle de atitude e um paraquedas de recuperação como acréscimo.



Figura 1.2. Bancada de teste de motor híbrido.

A origem do nome da equipe (propulsão híbrida, em português) refere-se ao fato dos propelentes estarem em estados físicos diferentes: combustível sólido (parafina) e oxidante líquido

(óxido nitroso – N_2O). No início das pesquisas com propulsão híbrida, houve muita relutância em seu uso, mas hoje é largamente usado.

Na área de Dinâmica de Vôo, o foguete possui três eixos que rotacionam nos ângulos Pitch (arfagem), Yaw (guinada), Roll (rolagem). A orientação desses três ângulos definem a Atitude do foguete. Para essa aeronave o projeto de controle dos ângulos de arfagem e guinada são idênticos, enquanto que o ângulo de rolagem necessita de um projeto diferente. Os controles do três ângulos podem ser feitos independentemente um do outro. O Pitch é considerado o ângulo mais relevante visto ser o usado para se calcular a trajetória de vôo. Para o presente trabalho será realizado o projeto do controlador para a Atitude Pitch.

1.2 SOFTWARES UTILIZADOS

Foram utilizados quatro softwares para o desenvolvimento do trabalho: MATLAB R2010a, OpenRocket, Digital Datcom e SolidWorks 2013.

O *MATLAB R2010a* foi usado para simular os sistemas projetados. O *OpenRocket* foi utilizado para se determinar a posição do centro de pressão (CP). O *Digital Datcom* foi usado para se determinar as derivadas de estabilidade não-dimensionais. E *SolidWorks 2013* foi utilizado para se fazer o desenho mecânico da fuselagem do foguete.

1.3 DEFINIÇÃO DO PROBLEMA

O foguete LILE-2 decolará com um ângulo de 87° em relação à linha horizontal. Após isso deverá manter-se com trajetória retilínea por todo o vôo de teste. No entanto, o arrasto, força aerodinâmica de resistência do ar, e a gravidade alteram a orientação. Este é o problema abordado. Desse modo, faz-se necessário, projetar um sistema de controle capaz de controlar a direção do jato de combustão a fim de corrigir qualquer mudança na orientação.

1.4 OBJETIVOS DO TRABALHO

Como mencionado, o objetivo principal do trabalho é projetar um sistema de controle que altere a direção do jato de combustão para corrigir mudanças na orientação do LILE-2 causadas pelo arrasto e pela gravidade. A Atitude, grandeza física que define a orientação de um corpo extenso, para uma aeronave, e por consequência para o foguete LILE-2, consiste na orientação de cada um dos três eixos do sistema de eixos associado ao corpo e tendo sua origem no centro de gravidade. A Atitude é medida pela posição angular. Os três ângulos em uma aeronave são conhecidos por Pitch (movimento de arfagem), Yaw (movimento de guinada) e Roll (movimento de rolagem). Para o LILE-2, o projeto de controle dos ângulos Pitch e Yaw são idênticos e o ângulo Roll precisa de um projeto diferente. Os três projetos podem ser feitos independentemente. Pelo fato de a trajetória de vôo (no caso, a trajetória é retilínea com ângulo de 87° em relação à horizontal) ser projetada no ângulo Pitch, foi escolhido, para o presente trabalho, realizar o projeto do controlador para esse ângulo. O projeto foi feito em nível de simulação computacional, e algumas simplificações foram necessárias.

1.5 RESULTADOS ALCANÇADOS

Realizou-se neste trabalho o projeto de um controlador de ganhos programados para o controle do movimento de arfagem (Pitch) do LILE-2. A parte analógica é dada por controladores PID e a parte discretizada é dada pela discretização em 18 valores da faixa de valores do ângulo de ataque utilizadas na simulação ($-4,5^\circ$ a $4,5^\circ$). Tal discretização eliminou a característica de a planta ser variante no tempo. A característica inviabilizaria o trabalho, não porque o projeto não seria fisicamente possível, mas porque é um conhecimento não ministrado nas disciplinas obrigatórias da graduação e assimilá-lo deixaria mais extensa a já extensa revisão bibliográfica e faltaria tempo para a conclusão do projeto. Parte dos dados foram dados reais de projeto, como a geometria do LILE-2 e outra parte foram obtidas por meio ferramentas de *software*. Por fim, o controlador híbrido foi validado por meio de simulação computacional em *MATLAB R2010a*.

1.6 APRESENTAÇÃO DO MANUSCRITO

No Capítulo 2, é feita uma revisão bibliográfica dos principais conceitos abordados no trabalho. No Capítulo 3, faz-se a modelagem matemática da planta e do controlador trabalhados. O Capítulo 4, fala do procedimento de projeto e das etapas seguidas no desenvolvimento. O Capítulo 5, apresenta os principais resultados alcançados no desenvolvimento. No Capítulo 6, tem-se a análise dos resultados alcançados. O Capítulo 7, apresenta as conclusões e sugestões de trabalhos futuros. E os anexos, apresentam códigos em MATLAB desenvolvidos, os demais resultados não-apresentados no texto e uma descrição do conteúdo do CD.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a parte mais relevante da carga teórica utilizada no trabalho.

2.1 FUNÇÕES MATEMÁTICAS

Apresenta-se a seguir as funções degrau unitário e decaimento exponencial.

2.1.1 Função degrau unitário

A função degrau unitário apresenta um valor constante e igual a um a partir do instante zero e um valor constante e igual a zero para valores menores que zero de t . Em linguagem matemática, tem a seguinte representação:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.1)$$

O gráfico da função degrau unitário encontra-se na Fig. 2.1.

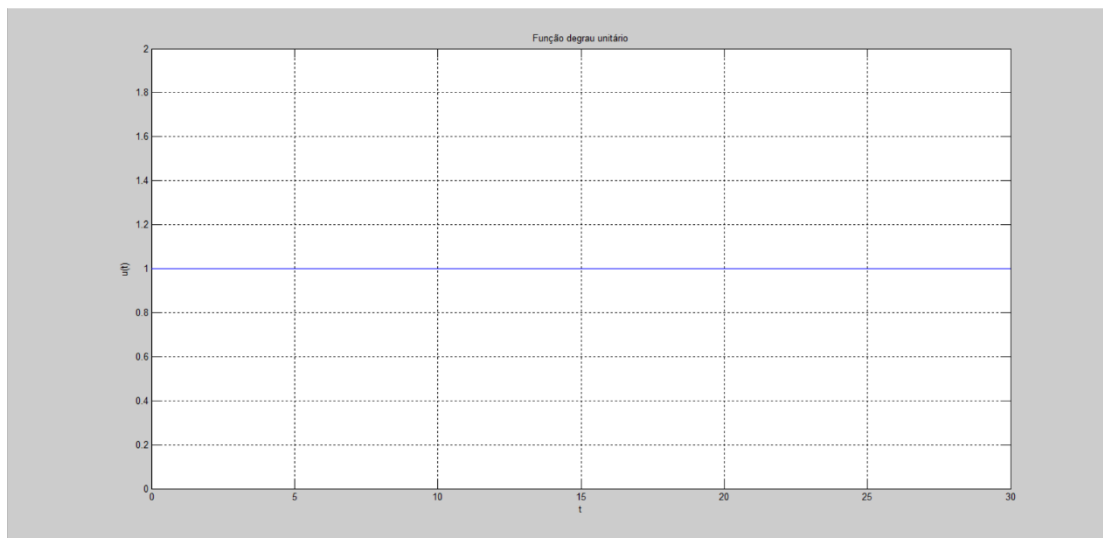


Figura 2.1. Gráfico da função degrau unitário.

2.1.2 Função decaimento exponencial

A função decaimento exponencial tem a forma [1]

$$y = y_0 e^{ct} \quad (2.2)$$

onde $c < 0$.

Um caso particular é a função da forma

$$y = e^{-t} \quad (2.3)$$

Essa função inicia-se com valor 1 e tende a zero no infinito. O gráfico se encontra na Fig. 2.2.

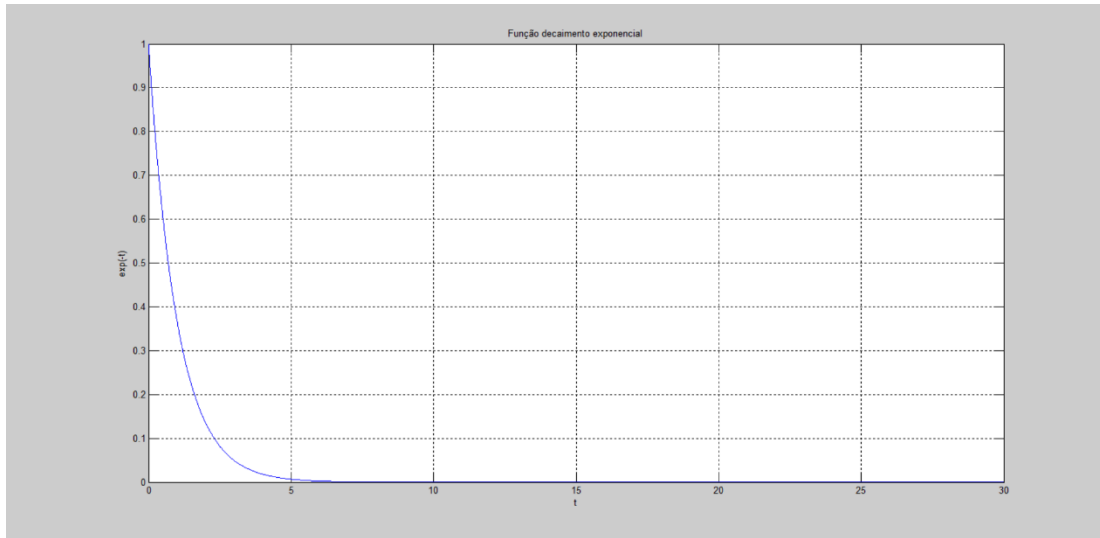


Figura 2.2. Gráfico do decaimento exponencial.

2.2 MODELOS MATEMÁTICOS DE UM SISTEMA FÍSICO

Existem três modos de representação matemática de um sistema físico: equação diferencial, função de transferência e espaço de estados.

Cada sistema físico é descrito por leis físicas específicas, como a lei de Newton para sistemas mecânicos e as leis de Kirchhoff para circuitos elétricos. Tais leis físicas levam a modelos matemáticos que descrevem a relação entre entrada e saída dos sistemas físicos. A equação diferencial linear e invariante no tempo é um exemplo dessa modelagem [2]:

$$a_n \frac{d^n c(t)}{dt^n} + a_{n-1} \frac{d^{n-1} c(t)}{dt^{n-1}} + \dots + a_0 c(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \dots + b_0 r(t) \quad (2.4)$$

O uso da representação em função de transferência permite representar a entrada, a saída e os sistema como entidades separadas. Além disso, as inter-relações são simplesmente algébricas. Inicialmente, apresenta-se a transformada de Laplace:

$$L[f(t)] = F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (2.5)$$

onde $s = \sigma + j\omega$ é uma variável complexa. Sendo tal integral possível, $F(s)$ é dita *transformada de Laplace* de $f(t)$.

Para se obter $f(t)$ a partir da transformada de Laplace, usa-se a transformada de Laplace inversa e é definida como:

$$L^{-1}[F(s)] = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s) e^{st} ds = f(t)u(t) \quad (2.6)$$

onde

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.7)$$

Apresenta-se agora o conceito de função de transferência. Ao aplicar-se a transformada de Laplace a ambos os lados da Eq. 2.4, obtém-se

$$a_n s^n C(s) + a_{n-1} s^{n-1} C(s) + \dots + a_0 C(s) + \text{condição inicial com termos envolvendo } c(t) = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \dots + b_0 R(s) + \text{condição inicial com termos envolvendo } r(t) \quad (2.8)$$

A função de transferência é calculada com as condições iniciais nulas. Dessa forma, a Eq. 2.8 torna-se

$$(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0) C(s) = (b_m s^m + b_{m-1} s^{m-1} + \dots + b_0) R(s) \quad (2.9)$$

Explicitando-se a relação entre a transformada de saída, $C(s)$, e a transformada de entrada, $R(s)$, tem-se

$$\frac{C(s)}{R(s)} = G(s) = \frac{(b_m s^m + b_{m-1} s^{m-1} + \dots + b_0)}{(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0)} \quad (2.10)$$

$G(s)$ é chamada função de transferência.

É possível obter-se a saída, $C(s)$, explicitando-a na Eq. (2.10):

$$C(s) = R(s)G(s) \quad (2.11)$$

Trata-se agora da representação em espaço de estados. Diferentemente das outras representações, o espaço de estados pode representar sistemas com múltiplas entradas e múltiplas saídas. Além disso, é próprio para utilização computacional. Um sistema físico é representado por estas equações, no espaço de estados:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (2.12)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (2.13)$$

Válido para $t \geq t_0$ e condições iniciais, $\mathbf{x}(t_0)$, onde

\mathbf{x} = vetor de estado

$\dot{\mathbf{x}}$ = derivada do vetor de estado em relação ao tempo

\mathbf{y} = vetor de saídas

\mathbf{u} = vetor de entradas (vetor de controle)

\mathbf{A} = matriz do sistema (matriz de estado)

\mathbf{B} = matriz de entradas

\mathbf{C} = matriz de saídas

\mathbf{D} = matriz de realimentação

A Equação 2.12 é chamada *equação de estado* e a Eq. 2.13 é chamada *equação de saídas*. Um exemplo [2] modificado mostra a representação de sistema físico (um circuito elétrico) em espaço de estados. O circuito encontra-se na Fig. 2.3.

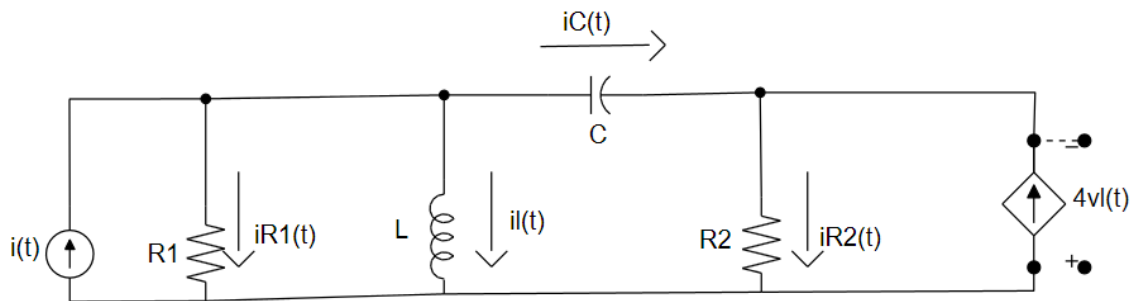


Figura 2.3 Circuito elétrico.

Etapa 1: Nomear todas as correntes no circuito, Fig. 2.3.

Etapa 2: Selecionar as variáveis de estado a partir das relações tensão-corrente de todos os armazenadores de energia:

$$L \frac{di_L}{dt} = v_L \quad (2.14)$$

$$C \frac{dv_C}{dt} = i_C \quad (2.15)$$

As variáveis de estados devem ser aquelas passíveis de serem derivadas. Assim,

$$x_1 = i_L \quad (2.16)$$

$$x_2 = v_C \quad (2.17)$$

Usando-se as leis de Kirchhoff, chega-se à equação de estado.

Da malha que contém L e C , tem-se

$$v_L = v_C + v_{R_2} = v_C + i_{R_2} R_2$$

$$v_L = v_C + (i_C + 4v_L) R_2$$

$$v_L = \frac{1}{1-4R_2} (v_C + i_C R_2) \quad (2.18)$$

Para o cálculo de i_L , tem-se

$$i_C = i(t) - i_{R_1} - i_L$$

$$i_C = i(t) - \frac{v_{R_1}}{R_1} - i_L$$

$$i_C = i(t) - \frac{v_L}{R_1} - i_L \quad (2.19)$$

Pode-se reescrever as Eq. 2.18 e 2.19 para se obter as equações simultâneas que definem v_L e i_C como combinações lineares das variáveis de estado:

$$(1 - 4R_2)v_L - R_2i_C = v_C \quad (2.20)$$

$$-\frac{1}{R_1}v_L - i_C = i_L - i(t) \quad (2.21)$$

O resultado das soluções das Eq. 2.20 e 2.21 é

$$v_L = \frac{1}{\Delta} [R_2i_L - v_C - R_2i(t)] \quad (2.22)$$

$$i_C = \frac{1}{\Delta} [(1 - 4R_2)i_L + \frac{1}{R_1}v_C - (1 - 4R_2)i(t)] \quad (2.23)$$

$$\Delta = - \left[(1 - 4R_2) + \frac{R_2}{R_1} \right] \quad (2.24)$$

Fazendo-se a substituição das Eq. 2.22 e 2.23 nas Eq. 2.14 e 2.15, e escrevendo tal resultado na forma vetorial-matricial, tem-se a equação de estado a seguir:

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} \frac{R_2}{(L\Delta)} & -\frac{1}{(L\Delta)} \\ \frac{(1-4R_2)}{(C\Delta)} & \frac{1}{(R_1C\Delta)} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{-R_2}{(L\Delta)} \\ \frac{-(1-4R_2)}{(C\Delta)} \end{bmatrix} i(t) \quad (2.25)$$

Para a equação de saída com apenas uma saída, no caso, v_{R_2} , pode-se escrever a partir da malha que contém C , L e R_2

$$v_{R_2} = -v_C + v_L \quad (2.26)$$

Fazendo-se a substituição da Eq. 2.22 na Eq. 2.26 e escrevendo-se o resultado na forma vetorial-matricial, tem-se:

$$v_{R_2} = \left[\frac{R_2}{\Delta} \left| -\left(1 + \frac{1}{\Delta}\right) \right. \right] \begin{bmatrix} \dot{i}_L \\ v_C \end{bmatrix} - \frac{R_2}{\Delta} i(t) \quad (2.27)$$

Caso queira-se especificar duas saídas, por exemplo, v_{R_2} e i_{R_2} , pode-se escrever a partir da malha que contém C , L e R_2

$$v_{R_2} = -v_C + v_L \quad (2.26)$$

$$i_{R_2} = i_C + 4v_L \quad (2.28)$$

Fazendo-se a substituição das Eq. 2.22 e 2.23 nas Eq. 2.26 e 2.28 e escrevendo-as na forma vetorial-matricial, tem-se

$$\begin{bmatrix} v_{R_2} \\ i_{R_2} \end{bmatrix} = \begin{bmatrix} \frac{R_2}{\Delta} & -\left(1 + \frac{1}{\Delta}\right) \\ \frac{1}{\Delta} & \frac{(1-4R_2)}{(\Delta R_1)} \end{bmatrix} \begin{bmatrix} \dot{i}_L \\ v_C \end{bmatrix} + \begin{bmatrix} -\frac{R_2}{\Delta} \\ -\frac{1}{\Delta} \end{bmatrix} i(t) \quad (2.29)$$

O acréscimo de mais uma saída no modelo implica no acréscimo de mais uma linha na equação de saída referente à nova saída e tal procedimento não altera a equação de saída

referente às outras saídas previamente consideradas. Do mesmo modo, a retirada de uma saída do modelo não afeta as outras saídas.

2.3 AERODINÂMICA BÁSICA

A Aerodinâmica é o estudo da atuação das forças de um fluido (no caso, o ar) na fuselagem da aeronave.

Na literatura, encontra-se de forma mais comum, o esclarecimento sobre aerodinâmica fazendo alusão à aviões. No entanto, o mesmo se aplica a foguetes.

Ao passar pelo corpo do avião, o ar precisa alterar seu caminho, o que altera a sua velocidade. Além a viscosidade do ar gera forças de fricção. Esses fatores fazem existir forças e momentos aerodinâmicos no avião. Decompondo-se tais forças e momentos sobre o avião nas três dimensões temos as grandezas que se seguem [3].

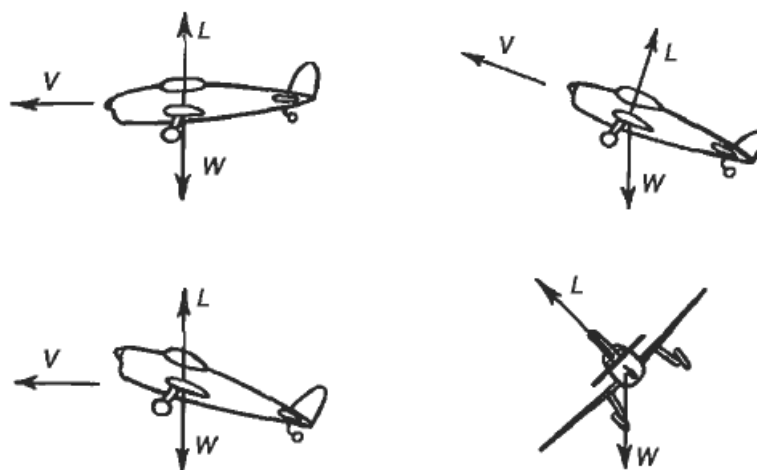


Figura 2.4. Configurações da força de Sustentação.

A Sustentação, denotada por L (de Lift, em inglês), ou por $-Z$, se for representada no sistema de eixos associado à aeronave localizado no centro de gravidade (CG), é força ocasionada nas asas devido à diferença de pressão entre as superfícies inferior e superior. Ela é sempre perpendicular à direção de vôo. A Figura 2.4 [3] mostra as diferentes configurações possíveis para a Sustentação durante o vôo.

O Arrasto, denotado por D (de Drag, em inglês), ou por $-X$, se for representada no sistema de eixos associado à aeronave, é a força oposta à direção da linha de vôo e que resiste ao movimento do avião.

A Força de Vento Cruzado, denotada por Y , é a componente mutuamente perpendicular à Sustentação e ao Arrasto e positivo no sentido para a direita do piloto.

O Momento de Arfagem, denotado por M , refere-se à rotação do eixo y do sistema de eixos da aeronave e é positivo quando o nariz do avião levanta.

O Momento de Rolagem, denotado por L_R , refere-se à rotação do eixo x do sistema de eixos da aeronave e é positivo quando asa direita desce.

O Momento de Guinada, denotado por N , refere-se à rotação do eixo z do sistema de eixos associado ao avião e é positivo quando o nariz da aeronave vira à direita.

A Figura 2.5 [3] mostra tais componentes no sistema de eixos da aeronave. As setas indicam o valor positivo.

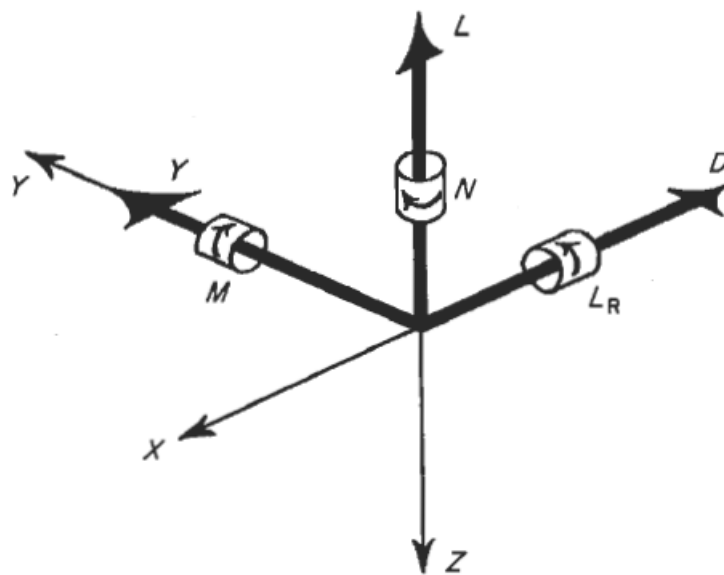


Figura 2.5. Forças e momentos.

No entanto, para estudos de controle e estabilidade, é mais conveniente uma representação ligeiramente diferente que será vista mais adiante.

2.4 DINÂMICA DE VÔO

A Dinâmica de Vôo é o estudo da estabilidade de uma aeronave. Será apresentado nesta seção, os conceitos da linearização do modelo matemático da planta e das derivadas de estabilidade.

As equações de movimento para aeronaves consistem em 15 equações diferenciais acopladas e mais 3 algébricas. Essas equações são não-lineares e, dessa forma, existe uma dificuldade extrema de tratá-las analiticamente. Com isso, em muitos problemas práticos, é admissível linearizar as equações por meio de pequenas perturbações em torno da condição de referência (*Teoria de Pequenas Perturbações*). Obtendo-se assim um conjunto de

equações diferenciais lineares homogêneas com coeficientes constantes que podem ser resolvidas mais fácil e rapidamente. [4]

A *Teoria de Pequenas Perturbações* restringe a região de estabilidade da aeronave, mas, em compensação, transforma as equações em lineares. Tais restrições surgem dos seguintes pressupostos a fim de fundamentar a teoria: Primeiro, o movimento consiste de pequenos desvios (perturbações) em torno das condições de vôo de equilíbrio; segundo, o movimento da aeronave pode ser analisado pela separação das equações em dois grupos: equações longitudinais (as de interesse no trabalho) e as equações laterais. O primeiro grupo é formado pelas equações da força X, da força Z e do momento de arfagem. O segundo grupo é formado pelas equações da força Y e dos momentos de guinada e rolagem [9]. Os dois grupos devem estar desacoplados. Sob esses pressupostos, a aeronave não deve passar por amplitudes grandes e manobras muito rápidas, como será o caso do LILE-2. O modelo é estável nessas condições e instável para valores grandes de perturbações. Por exemplo, o ângulo de rotação do eixo z não deve ser 90°.

A seguir estão considerações sobre a linearização proporcionada pela *Teoria das Pequenas Perturbações*.

Uma nova notação para as variáveis é pertinente. Os valores de referência de todas as variáveis recebem o subscrito zero e as pequenas perturbações recebem o prefixo Δ . Nos casos em que o valor de referência é zero, o prefixo Δ sofre omissão. Também são feitas simplificações trigonométricas, considerando-se o fato de as perturbações serem pequenas [4].

Ao se introduzir a notação das pequenas perturbações nas 18 equações não-lineares de movimento, os pressupostos acima incorporados, apenas os termos de primeira ordem são mantidos e, dessa forma, tem-se 12 equações lineares.

A demonstração das equações gerais de movimento de uma aeronave, como o avião, assim como apresentá-las fogem do escopo do presente trabalho. Mais à frente será apresentado o equivalente para o foguete deste projeto.

Na *Teoria de Pequenas Perturbações*, tem-se que as variáveis assumem um valor de equilíbrio fixo e um valor de perturbação pequeno associado à variável é linearizado por séries de Taylor. Tais séries geram as Derivadas de Estabilidade. Um exemplo é a Eq. a seguir para a perturbação na força Z,

$$\Delta Z = Z_u \Delta u + Z_w w + Z_{\dot{w}} \dot{w} + Z_q q + \Delta Z_c \quad (2.30)$$

onde $\Delta Z, \Delta u, w, \dot{w}$ e q são pequenas perturbações, $Z_u, Z_w, Z_{\dot{w}}$ e Z_q são derivadas de estabilidade e ΔZ_c é uma força de controle.

Dessa equação, pode-se tirar que há mais de um ponto de operação e esses pontos são os valores de referência das variáveis [5].

As derivadas de estabilidade são

$$Z_w = \frac{\partial \Delta Z}{\partial w}, \text{ etc.} \quad (2.31)$$

Para este trabalho as derivadas de estabilidade de interesse são Z_α , M_α e M_q . Após o cálculo das derivadas, tem-se

$$Z_\alpha = \frac{1}{2} \rho \mu_0 S C_{Z_\alpha} \quad (2.32)$$

$$M_\alpha = \frac{1}{2} \rho \mu_0 l S C_{M_\alpha} \quad (2.33)$$

$$M_q = \frac{1}{4} \rho \mu_0 l^2 S C_{M_q} \quad (2.34)$$

Os fatores C_{Z_α} , C_{M_α} e C_{M_q} são chamados derivadas de estabilidade não-dimensionais, enquanto que Z_α , M_α e M_q são chamados, mais especificamente, de derivadas de estabilidade dimensionais. As derivadas de estabilidade não dimensionais possuem definição semelhante às das derivadas de estabilidade dimensionais. A sua definição não é tão relevante para o escopo deste trabalho e, assim, não serão detalhadas.

Sobre o significado dos outros fatores das equações dessas três derivadas, eles serão apresentados em momento oportuno.

2.5 CONTROLE DE ATITUDE DE FOGUETES

A atitude de um foguete é a orientação dos três eixos do sistema de eixos associado ao foguete com a origem localizada no centro de gravidade. A necessidade de se ter um sistema de controle para a atitude do foguete se deve ao fato de as forças aerodinâmicas e a gravidade desalinham o eixo x com o vetor velocidade.

A Figura 2.6 [6] mostra características importantes de projeto para o controle de atitude de um foguete. Diferentemente da abordagem em Aerodinâmica, as forças sobre o foguete têm sua orientação positiva no mesmo sentido dos eixos, mas os momentos continuam os mesmos. T é a força de empuxo proporcionada pelo jato de combustão, ξ é a distância do CG para a base, ξ_n é a distância da posição dos sensores para o CG, α e β , são os ângulos de desalinhamento do eixo x para o vetor velocidade, v , e μ_1 e μ_2 são ângulos do empuxo para o controle do desalinhamento.

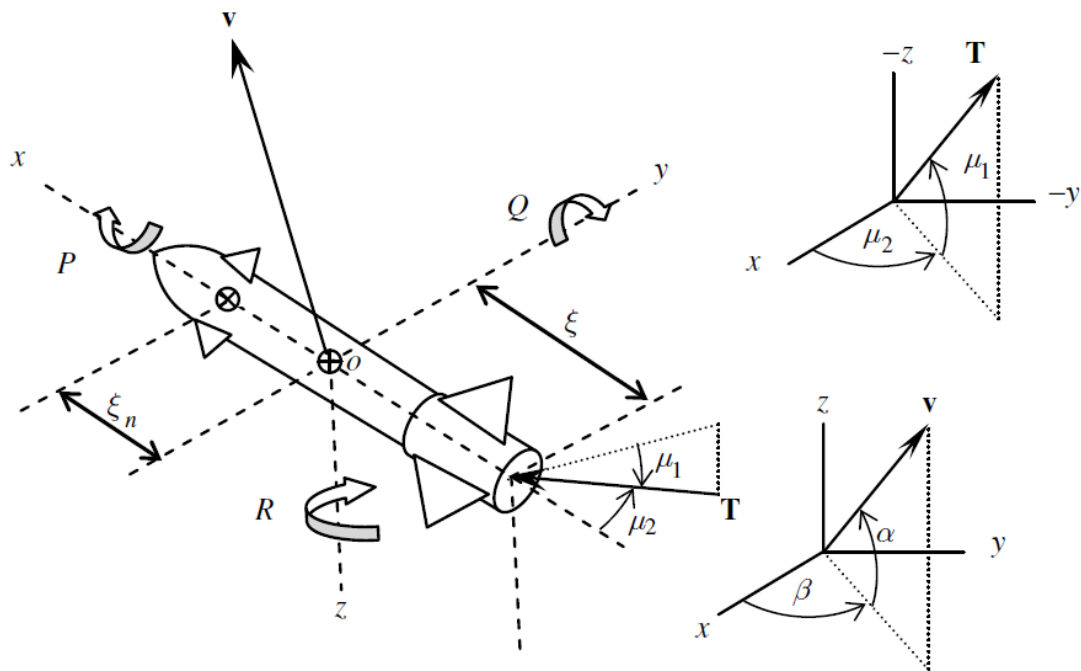


Figura 2.6. Variáveis para o controle de atitude.

O controle da orientação de eixo do foguete pode ser separadamente, desde que se mantenha o momento de rolagem igual a zero (para manter as equações desacopladas). O projeto do controlador para os ângulos Pitch (arfagem) e Yam (guinada) para um foguete são idênticos devido à simetria geométrica, enquanto que o projeto para o ângulo Roll (rolagem) precisa de uma abordagem diferente.

Quando convém ter-se uma trajetória de vôo que varia, ela é projetada e calculada com base no ângulo Pitch por meio do Sistema de Guiagem. Sendo tal ângulo o que possui maior importância no projeto de controle de atitude de um foguete. Por razões de limitação de tempo, para o presente trabalho, será projetado um controlador apenas o movimento de arfagem (Pitch) e tal escolha se deve justamente ao fato de ser o movimento mais importante dos três. Desse modo, algumas variáveis apresentadas na Fig. Não serão usadas no projeto.

A planta que representa a dinâmica do movimento de arfagem (ângulo Pitch), sugerida por [6], em representação espaço de estados é apresentada abaixo

$$\dot{x} = Ax + Bu \quad (2.35)$$

$$y = Cx + Du \quad (2.36)$$

onde

$$x = \begin{bmatrix} q \\ \alpha \end{bmatrix} \quad (2.37)$$

$$u = \mu_1 \quad (2.38)$$

$$y = \begin{bmatrix} q \\ \Delta a_z^0 \end{bmatrix} \quad (2.39)$$

$$A = \begin{bmatrix} \frac{M_q}{J_{yy}} & \frac{M_\alpha}{J_{yy}} \\ 1 & \frac{Z_\alpha}{mv} \end{bmatrix} \quad (2.40)$$

$$B = \begin{bmatrix} -\frac{T\bar{z}}{J_{yy}} \\ \frac{T}{mv} \end{bmatrix} \quad (2.41)$$

$$C = \begin{bmatrix} 1 & 0 \\ -\frac{M_q\bar{z}_n}{J_{yy}} & \frac{Z_\alpha}{m} - \frac{M_\alpha\bar{z}_n}{J_{yy}} \end{bmatrix} \quad (2.42)$$

$$D = \begin{bmatrix} 0 \\ T(\frac{\bar{z}_n\bar{z}}{J_{yy}} - \frac{1}{m}) \end{bmatrix} \quad (2.43)$$

2.6 CONTROLADOR PID

O controlador PID (controlador proporcional integral derivativo) possui duas características importantes: melhora a resposta transiente e aumenta o tipo do sistema (o que proporciona a possibilidade de levar o erro em regime permanente a zero).

A função de transferência de um controlador PID é

$$G_c(s) = K_1 + \frac{K_2}{s} + K_3s \quad (2.44)$$

Na Figura 2.7 [7], encontra-se o diagrama de blocos para um sistema de controle em malha fechada com o controlador PID.

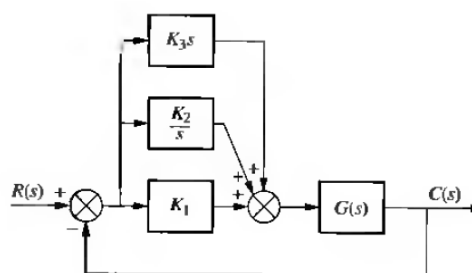


Figura 2.7. Diagrama de blocos.

2.7 CONTROLADOR DE GANHOS PROGRAMADOS

O sistema de controle de ganhos programados é uma técnica a ser aplicada a sistemas lineares variantes no tempo. A síntese do controlador compreende três etapas: determinar um conjunto de modelos locais para a planta em condições operacionais diferentes; é projetado um controlador, chamado controlador local (no caso, PID), para cada um dos modelos locais; realiza-se o chaveamento ou a interpolação dos controladores locais [8].

3 MODELAGEM MATEMÁTICA

Apresenta-se agora o desenvolvimento dos principais modelos matemáticos utilizados no projeto.

3.1 PLANTA DO FOGUETE

Foram cogitados dois modelos-base para a planta do foguete. Um proposto em [9] e outro proposto em [6]. O modelo escolhido foi o de [6] pelo fato de que, tal modelo foi concebido propriamente para foguetes e modela inclusive os sensores giroscópio e acelerômetro, enquanto que o modelo de [9] foi concebido para aeronaves de modo geral e não inclui os sensores. Dessa forma, segue-se abaixo a planta escolhida em representação de espaço de estados:

$$\dot{x} = Ax + Bu \quad (3.1)$$

$$y = Cx + Du \quad (3.2)$$

onde

$$x = \begin{bmatrix} q \\ \alpha \end{bmatrix} \quad (3.3)$$

$$u = \mu_1 \quad (3.4)$$

$$y = \begin{bmatrix} q \\ \Delta a_z^0 \end{bmatrix} \quad (3.5)$$

$$A = \begin{bmatrix} \frac{M_q}{J_{yy}} & \frac{M_\alpha}{J_{yy}} \\ 1 & \frac{Z_\alpha}{mv} \end{bmatrix} \quad (3.6)$$

$$B = \begin{bmatrix} -\frac{T\bar{z}}{J_{yy}} \\ \frac{T}{mv} \end{bmatrix} \quad (3.7)$$

$$C = \begin{bmatrix} 1 & 0 \\ -\frac{M_q\bar{z}_n}{J_{yy}} & \frac{Z_\alpha}{m} - \frac{M_\alpha\bar{z}_n}{J_{yy}} \end{bmatrix} \quad (3.8)$$

$$D = \begin{bmatrix} 0 \\ T(\frac{\bar{z}_n\bar{z}}{J_{yy}} - \frac{1}{m}) \end{bmatrix} \quad (3.9)$$

Entretanto, a saída representada pela variação da aceleração normal sentida pelo acelerômetro, Δa_z^0 , não será necessária no projeto e pode ser removida da equação de saída como demonstrado na seção 2.2. Assim, a planta utilizada para o foguete tem a seguinte representação em espaço de estados:

$$\dot{x} = Ax + Bu \quad (3.10)$$

$$y = Cx + Du \quad (3.11)$$

onde

$$\mathbf{x} = \begin{bmatrix} q \\ \alpha \end{bmatrix} \quad (3.12)$$

$$u = \mu_1 \quad (3.13)$$

$$y = q \quad (3.14)$$

$$\mathbf{A} = \begin{bmatrix} \frac{M_q}{J_{yy}} & \frac{M_\alpha}{J_{yy}} \\ 1 & \frac{Z_\alpha}{mv} \end{bmatrix} \quad (3.15)$$

$$\mathbf{B} = \begin{bmatrix} -\frac{T_z^z}{J_{yy}} \\ \frac{T}{mv} \end{bmatrix} \quad (3.16)$$

$$\mathbf{C} = [1|0] \quad (3.17)$$

$$D = 0 \quad (3.18)$$

3.2 MODELO MATEMÁTICO DO CONTROLADOR PID

A função de transferência para um controlador PID é

$$G_c(s) = K_1 + \frac{K_2}{s} + K_3s$$

$$G_c(s) = \frac{K_1s + K_2 + K_3s^2}{s}$$

$$G_c(s) = \frac{K_3(s^2 + \frac{K_1}{K_3}s + \frac{K_2}{K_3})}{s} \quad (3.19)$$

A expressão entre parênteses pode ser simplificada para dois zeros iguais, obtendo-se assim um caso particular de controlador PID:

$$G_c(s) = \frac{K(s+a)^2}{s} \quad (3.20)$$

Os parâmetros K e a precisam ser determinados.

4 DESENVOLVIMENTO

Este capítulo tem por finalidade descrever passo-a-passo o procedimento utilizado para se desenvolver o projeto e chegar aos resultados finais.

4.1 PROCEDIMENTO DE PROJETO

O procedimento de projeto seguido resumidamente passou pelas seguintes etapas: Determinação do sistema físico, determinação das especificações, obtenção do modelo matemático, análise, projeto e testes.

4.2 O SISTEMA FÍSICO DA PLANTA

Uma concepção do sistema físico da planta é a considerada na Fig. 4.1 [6].

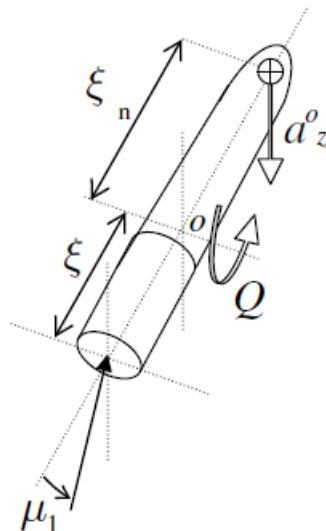


Figura 4.1. Sistema físico da planta.

Como variável nova apresentada tem-se a_z^0 , que é a componente da aceleração em z no ponto onde se encontram-se os sensores.

4.3 DESENHO MECÂNICO DA FUSELAGEM DO LILE-2

Um dos primeiros passos foi a concepção do desenho mecânico da fuselagem do LILE-2 no software SolidWorks 2013 para uso no projeto do controlador. Originalmente o LILE-2 possuía aletas e um nariz arredondado. Devido às condições do modelo matemático usado, as aletas foram retiradas e, devido às condições do software Digital Datcom para cálculo das derivadas de estabilidade não-dimensionais, o nariz foi considerado cônico.

4.4 PARÂMETROS DA PLANTA

Em seguida, foi feita a identificação de quais parâmetros são originalmente invariantes no tempo e quais são variantes no tempo. Após isso, foram escolhidos quais parâmetros variantes seriam aproximados por invariantes e quais permaneceriam variantes. A aproximação foi considerada devido ao fato do nível de complexidade exigido para o trabalho e devido ao tempo hábil disponível para realizá-lo. Mais especificamente, os parâmetros são formados por fatores e tais fatores foram trabalhados individualmente como invariantes e variantes.

Os valores dos fatores foram identificados.

Foram pesquisados os valores de velocidade, v , e empuxo, T , que seriam desenvolvidos pelo LILE-2 durante o vôo. Tais valores são intrínsecos a trabalhos de simulação de outros membros da Hybrid Propulsion Team, que forneceram as informações.

O fator v varia ao longo do vôo. Este é um fator aproximado para invariante. Dessa forma, calculou-se a velocidade média para a função $v(t)$.

O fator T originalmente é constante.

Foi calculada a massa, m , do LILE-2. A massa m varia com o vôo devido à queima do combustível, no entanto, foi considerada constante. O corpo do foguete foi considerado como uma massa sólida de PVC soldável e o nariz, uma massa sólida de nylon 6.0. Pelas dimensões do foguete e a densidade de cada material, foram calculadas as massas do corpo, do nariz e total.

O momento de inércia em torno do eixo y , J_{yy} , varia com o tempo, mas também foi considerado constante. Foi calculado o momento de inércia para o nariz e o momento de inércia para o corpo e, em seguida, somados.

Foi calculada a distância \bar{x} . Esse valor é a distância da origem do sistema de eixos do LILE-2 para a base do LILE-2. A origem do sistema de eixos foi localizada no centro de massa. Dessa forma, foi calculada a posição do centro de massa e, em seguida, o valor de \bar{x} .

As derivadas de estabilidade dimensionais foram trabalhadas. Elas permaneceram variantes no projeto, mas com uma ressalva: seriam discretizadas em 18 valores. As derivadas de estabilidade dimensionais são Z_α , M_α e M_q . Tais derivadas são compostas por fatores que incluem a densidade do ar na atmosfera, ρ , a velocidade do foguete, $\mu_0 = v$, a área da maior secção transversal do foguete, S , a distância entre o centro de gravidade (CG) e o centro de pressão (CP), l , e as derivadas de estabilidade não-dimensionais, C_{z_α} , C_{M_α} e C_{M_q} .

A densidade do ar na atmosférica varia com a altitude. No entanto, foi considerada constante. Para o cálculo dela, foram considerados os valores da Atmosfera Padrão Internacional descrita em [3]. Nela, há valores discretizados para diferentes níveis em toda a extensão da

atmosfera. A altura que seria atingida pelo LILE-2 é um valor intrínseco a trabalhos de simulação de outros membros da Hybrid Propulsion Team, que a forneceram. Também foi considerada para o cálculo, a altitude relativa à cidade de Brasília, Distrito Federal, Brasil. Para os valores de altitude que seriam desenvolvidos pelo LILE-2, a Atmosfera Padrão Internacional possui seis valores discretizados. Foi calculada uma função de interpolação para esses seis valores e, em seguida, calculada a média da função.

A velocidade μ_0 do foguete é a mesma de v , já mencionada.

A secção transversal do foguete, S , foi calculada a partir das dimensões do foguete.

A posição do CP varia de acordo com o valor do ângulo de ataque, α . A relação entre a posição do CP e α foi obtida a partir de um *software* chamado *OpenRocket*. Tal *software* pode associar um valor da posição de CP para cada valor de α . Dentro da faixa de valores de α selecionada para o projeto, é possível simular seis valores discretizados. Dessa forma, foi criada uma função de interpolação para esses seis valores. E para cada um dos 18 valores de α considerados no projeto, a posição de CP em relação à ponta do foguete foi calculada a partir da função de interpolação.

A distância entre a posição do CG e a posição do CP, l , foi calculada para cada valor da posição de CP pela diferença entre as duas posições.

As derivadas de estabilidade não-dimensionais, $C_{z\alpha}$, $C_{M\alpha}$ e C_{Mq} foram calculados pelo *software Digital Datcom* para 18 valores do ângulo de ataque, α .

Com esses fatores em mãos, foram calculados as derivadas de estabilidade dimensionais, Z_α , M_α e M_q para cada um dos 18 valores de α selecionados.

4.5 PLANTA DO LILE-2

Mais especificamente, a planta calculada do LILE-2 é a planta do ângulo de arfagem (Pitch) do LILE-2. A discretização do ângulo de ataque, α , tornaram discretizados os parâmetros variantes da planta. Desse modo, passou-se a ter 18 plantas diferentes no trabalho.

Cada planta foi inicialmente representada no espaço de estados e, em seguida, convertida para função de transferência por meio de um programa criado para o *software* MATLAB R2010a.

4.6 CONTROLADOR PID E CONTROLADOR DE GANHOS PROGRAMADOS

Para cada uma das 18 plantas, foi projetado um controlador PID por meio de programas adaptados de [10] criados para o *software* MATLAB R2010a. As especificações de projeto são Sobrevalor = 1,2 e Tempo de assentamento = 1s. O programa é necessário, visto que seriam projetados 18 controladores, um para cada planta. O controlador PID, originalmente,

possui três constantes a serem determinadas, K_1, K_2 e K_3 , no entanto, para se calcular por meio do programa seriam necessários quatro comandos *For*, um dentro do outro, o que levaria a um tempo de processamento absurdamente longo. Dessa forma, foi escolhido um caso particular de controlador PID com apenas duas constantes a determinar, K e α .

O funcionamento do programa consiste na seguinte descrição. Há dois parâmetros a serem calculados, K e α . Foram escolhidas duas faixas de valores, uma para cada parâmetro. Cada faixa foi variada a intervalos regulares e foram calculados um controlador e um sistema compensado para cada combinação dos valores de K e α . Na primeira combinação em que os requisitos são atingidos, o programa se encerra e apresenta os parâmetros calculados.

Após isso foi desenvolvido dois controladores de ganhos programados: um por chaveamento e outro por interpolação.

4.7 TESTES NO SISTEMA COMPENSADO

Para cada uma das plantas compensadas em malha fechada foram testadas duas entradas: entrada degrau unitário e decaimento exponencial. Como o LILE-2 precisará manter uma trajetória retilínea do início ao fim da propulsão, a entrada de referência deverá ser zero em todo o vôo. No entanto, uma entrada zero para teste geraria uma saída zero e o teste seria inconclusivo. Dessa forma, escolheu-se uma entrada que inicialmente assume um valor maior que zero e termina com valor zero (o decaimento exponencial) a fim de se verificar a questão da saída seguir uma referência zero.

Para o valor de $\alpha = 3^\circ$, a fim de se simular a queima de combustível e verificar a robustez do sistema de controle para tal situação, variou-se a massa de 10% a 90%, incrementando-se de 10 em 10 e obtiveram-se dados e os gráficos das saídas para as entradas degrau unitário e decaimento exponencial.

5 RESULTADOS

Neste capítulo apresenta-se os resultados obtidos no trabalho.

5.1 DESENHO MECÂNICO DA FUSELASEM DO LILE-2

O desenho mecânico da fuselagem do LILE-2 gerado em perspectiva encontra-se na Fig. 5.1. O Desenho mecânico em vista lateral com as principais medidas em metros encontra-se na Fig. 5.2. E, em vista traseira, na Fig. 5.3.

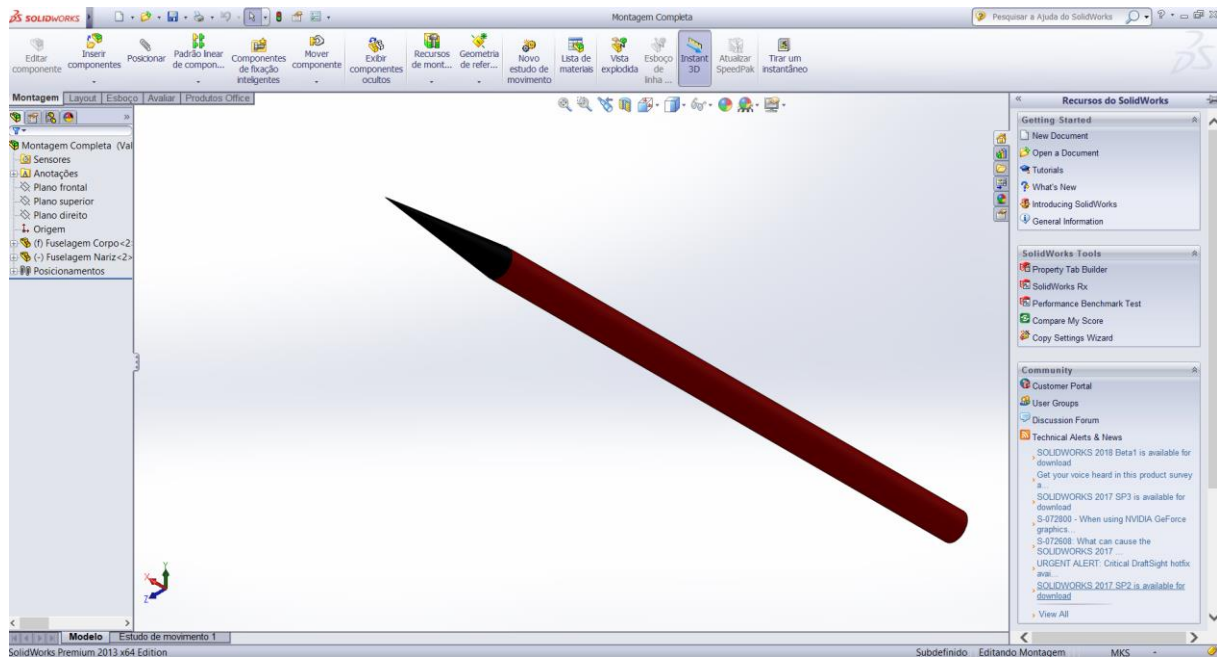


Figura 5.1. Vista em perspectiva.

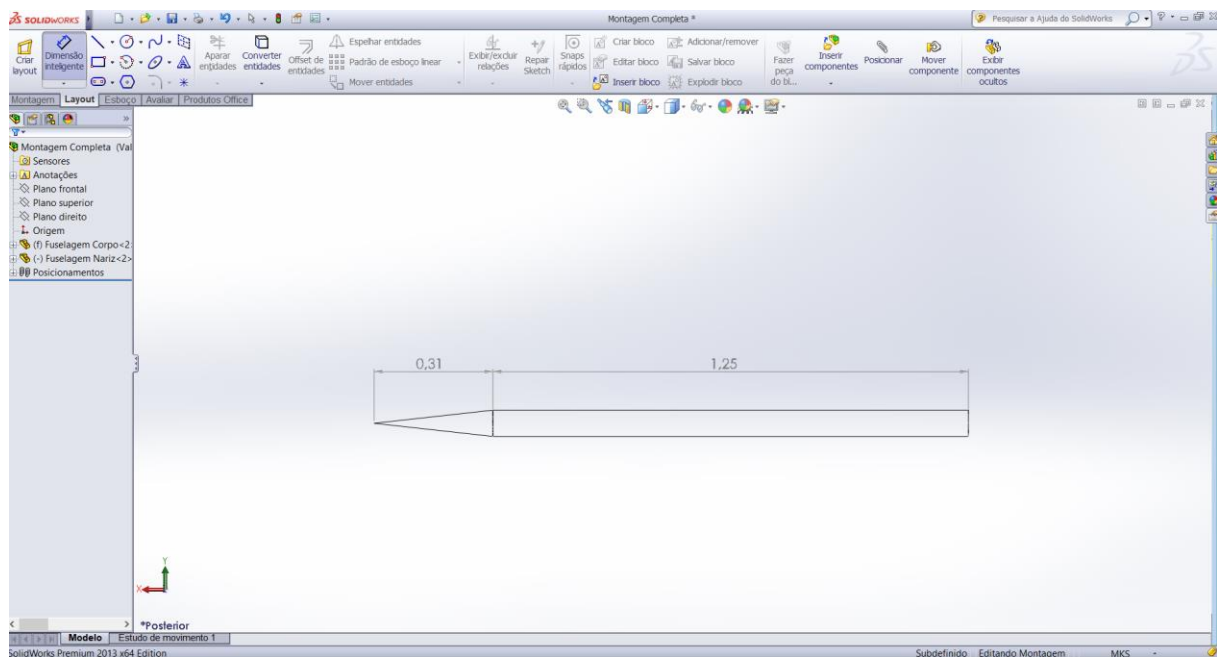


Figura 5.2. Vista lateral.

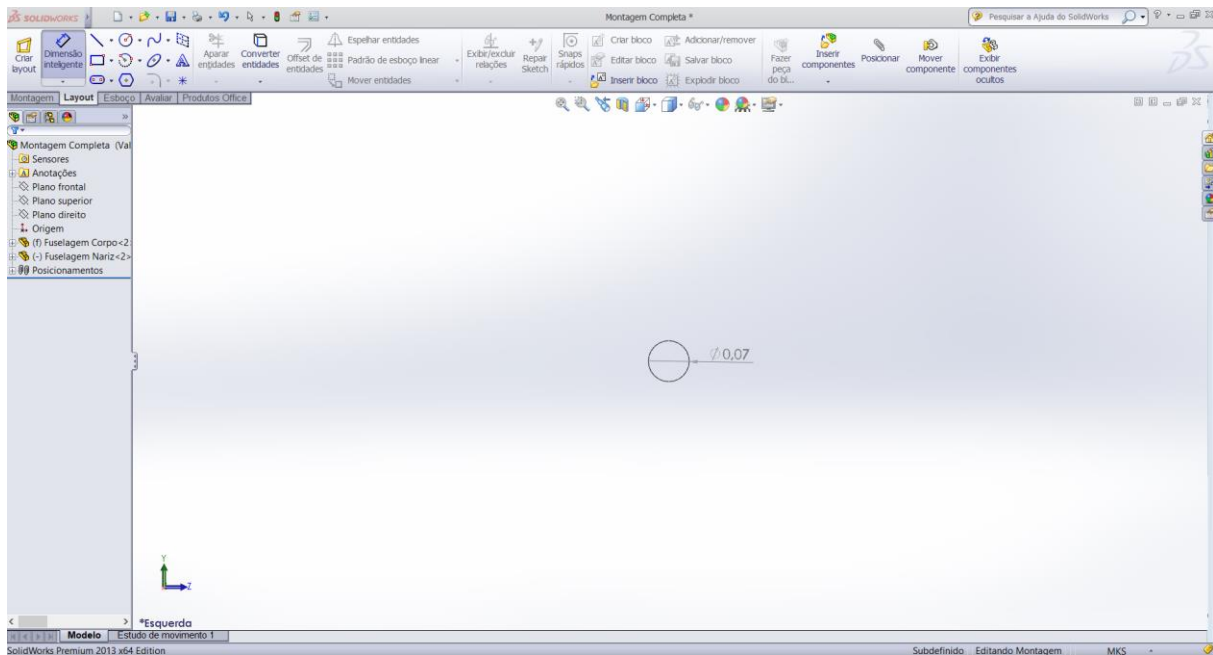


Figura 5.3. Vista traseira.

5.2 PARÂMETROS DA PLANTA

Os parâmetros da planta foram identificados de acordo com a simbologia abaixo:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (5.1)$$

$$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (5.2)$$

$$\mathbf{C} = [c_1 | c_2] \quad (5.3)$$

$$D = d_1 \quad (5.4)$$

Parâmetros originalmente invariantes no tempo: $a_3 = 1$, $c_1 = 1$, $c_2 = 0$ e $d_1 = 0$. Os demais parâmetros, a_1, a_2, a_4, b_1 e b_2 , são originalmente variantes no tempo. Esses parâmetros são iguais a

$$a_1 = \frac{M_q}{J_{yy}} \quad (5.5)$$

$$a_2 = \frac{M_\alpha}{J_{yy}} \quad (5.6)$$

$$a_4 = \frac{Z_\alpha}{mv} \quad (5.7)$$

$$b_1 = -\frac{T\bar{z}}{J_{yy}} \quad (5.8)$$

$$b_2 = -\frac{T}{mv} \quad (5.9)$$

As unidades de medida no trabalho estão em concordância com Sistema Internacional de Unidades (S.I.).

A função $v(t)$, para o cálculo de aproximação de v , foi estimada a partir do gráfico de simulação que se encontra no anexo 1 do texto. Enquanto há propulsão, trecho do vôo em que o controlador está em ação, a curva da velocidade pode ser aproximada por uma função do 1º grau. Dessa forma, $v(t)$ é estimado como

$$v(t) = 40,0256t \quad (5.10)$$

O valor médio de uma função em um intervalo fechado $[a, b]$ é dado pela integral

$$f_m = \frac{1}{b-a} \int_a^b f(x)dx \quad (5.11)$$

Por meio da equação (130), foi calculada a velocidade média, v , usada no modelo. O limite inferior da integral considerado é $a = 0 \text{ s}$ e o limite superior considerado é $b = 7,1231 \text{ s}$, que foi estimado por medição direta no gráfico. Dessa forma,

$$v = \frac{1}{7,1231-0} \int_0^{7,1231} 40,0256t dt$$

$$v = 142,55315 \text{ m/s} \quad (5.12)$$

Devido ao modo de implementação do arquivo de entrada para o cálculo das derivadas de estabilidade não-dimensionais no *software Digital Datcom*, a velocidade considerada para o trabalho foi de

$$v = 142,91 \text{ m/s} \quad (5.13)$$

Para o empuxo, T , o valor, fornecido por outros membros da *Hybrid Propulsion Team* a partir de simulação computacional em seus trabalhos e utilizado aqui, é

$$T = 400N \quad (5.14)$$

Para o cálculo da massa m do LILE-2, foram calculadas separadamente as massas do corpo e do nariz, visto serem de materiais diferentes. Para o cálculo de cada uma das massas foram calculadas primeiramente os volumes de cada parte. Em seguida, por meio da fórmula de densidade, foram calculadas as massas individuais. Após isso, foram somadas.

O volume do nariz é o volume de um cone:

$$V_n = \frac{\pi r^2 h_n}{3} \quad (5.15)$$

onde

r = raio da base do cone (raio da secção transversal do foguete),
 h_n = altura do cone.

$$V_n = \frac{\pi \times 0,03455^2 \times 0,312}{3}$$

$$V_n = 3,90001 \times 10^{-4} m^3 \quad (5.16)$$

O volume do corpo é o volume de um cilindro:

$$V_c = \pi r^2 h_c \quad (5.17)$$

onde

r = raio da base do cone (raio da secção transversal do foguete),
 h_c = altura do cilindro.

$$V_c = \pi \times 0,03455^2 \times 1,248$$

$$V_c = 4,6802 \times 10^{-3} m^3 \quad (5.18)$$

As densidades do PVC soldável (corpo) e do Nylon 6.0 (nariz) são, respectivamente,

$$\rho_c = 1,34 \times 10^3 \text{ kg/m}^3 \quad (5.19)$$

$$\rho_n = 1,14 \times 10^3 \text{ kg/m}^3 \quad (5.20)$$

Assim, as massas do corpo, da nariz e total são, respectivamente,

$$m_c = \rho_c V_c$$

$$m_c = 6,27141 \text{ kg} \quad (5.21)$$

$$m_n = \rho_n V_n$$

$$m_n = 4,4461 \times 10^{-1} \text{ kg} \quad (5.22)$$

$$m = m_c + m_n$$

$$m = 6,71602 \text{ kg} \quad (5.23)$$

O momento de inércia do LILE-2 em torno do eixo y , J_{yy} , foi cálculo por meio dos cálculos separados do momento de inércia do nariz e do momento de inércia do corpo e em seguida somados. Para cada parte, foi usado teorema dos eixos paralelos. Em seguida, os momentos de inércia foram somados.

$$J_{yy_n} = 2,44956 \times 10^{-1} \text{ kgm}^2 \quad (5.24)$$

$$J_{yy_c} = 8,56890 \times 10^{-1} \text{ kgm}^2 \quad (5.25)$$

$$J_{yy} = J_{yy_n} + J_{yy_c}$$

$$J_{yy} = 1,10185 \text{ kgm}^2 \quad (5.26)$$

Para o cálculo da distância ξ , foi necessário se determinar primeiramente a coordenada x'_{CG} (a linha serve para indicar que não é o mesmo eixo x do sistema de eixos doo CG) da posição do CG do foguete em relação à ponta do foguete:

$$x'_{CG} = 8,55110 \times 10^{-1} m \quad (5.27)$$

O comprimento do foguete é

$$c = 1,56000 m \quad (5.28)$$

E a distância ξ é

$$\xi = c - x'_{CG}$$

$$\xi = 7,04890 \times 10^{-1} m \quad (5.29)$$

Seguem-se agora os cálculos dos fatores das derivadas de estabilidade dimensionais.

A altura atingida pelo LILE-2 em simulação é de 1250m. Como a altitude de Brasília é 1000m, a faixa de altitude percorrida pelo LILE-2 seria de 1000m a 2250m. A função interpolação da densidade do ar na atmosfera foi calculada para seis valores dentro dessa faixa.

O cálculo da função de interpolação da densidade do ar na atmosfera, ρ , calculado é dado pela equação

$$\rho(h) = 2,8970 \times 10^{-16} h^5 - 2,2688 \times 10^{-12} h^4 + 6,4035 \times 10^{-9} h^3 - 9,3114 \times 10^{-6} h^2 + 6,5559 \times 10^{-3} h - 6,5680 \times 10^{-1} \quad (5.30)$$

A densidade média calculada com a mesma Eq. 5.30 é

$$\rho = 1,0483 \text{ kg/m}^3 \quad (5.31)$$

A velocidade $\mu_0 = v$ do foguete é

$$\mu_0 = 142,91 \text{ m/s} \quad (5.32)$$

A área da secção transversal do foguete, S , é

$$S = \pi r^2$$

$$S = 3,02880 \times 10^{-3} \text{ m}^2 \quad (5.33)$$

A função de interpolação para o cálculo da posição do CP em relação à ponta do foguete é dada pela equação

$$x'_{CP}(\alpha) = 15436,7\alpha^5 - 6286,48\alpha^4 + 1105,03\alpha^3 - 116,266\alpha^2 + 9,1158\alpha + 2,0800 \times 10^{-1} \quad (5.34)$$

A distância entre a posição do CG em relação à ponta do foguete e a posição do CP em relação à ponta do foguete, l , foi calculada para cada planta por meio da equação

$$l = x'_{CG} - x'_{CP}(\alpha) \quad (5.35)$$

As derivadas de estabilidade não-dimensionais, $C_{z\alpha}$, $C_{M\alpha}$ e C_{Mq} , foram calculadas pelo *software* Digital Datcom. $C_{z\alpha}$ e $C_{M\alpha}$ possuem um valor diferente para cada um dos 18 valores de α considerados, enquanto que C_{Mq} é constante.

Para o cálculo do *software* foi necessário o desenvolvimento de arquivo de entrada denominado *de.inp*. O este arquivo o *software* gera um arquivo de saída denominado *datcom.out* com os valores das derivadas de estabilidade não-dimensionais. O arquivo de entrada encontra-se na Fig. 5.4, os valores $C_{z\alpha}$ ($= C_{Y\beta}$) e $C_{M\alpha}$ do arquivo de saída encontram-se na Fig. 5.5 e o valor de C_{Mq} do arquivo de saída encontra-se na figura 5.6. Tais derivadas estão representadas em $graus^{-1}$ no arquivo de saída e foram convertidas para rad^{-1} .

```

$FLTCON NMACH=1.0, MACH(1)=0.42, NALPHA=20.0,
        ALSCHD(1)=-5.0,-4.5,-4.0,-3.5,-3.0,-2.5,-2.0,
                -1.5,-0.75,0.0,0.5, 1.0,1.5,2.0,2.5,
                3.0,3.5,4.0,4.5,5.0,
        NALT=20.0,
        ALT(1)=0.0,92.0,184.0,276.0,368.0,460.0,552.0,644.0,
                736.0,828.0,920.0,1012.0,1104.0,1196.0,1288.0,
                1380.0,1472.0,1564.0,1656.0,1750.0,
        WT=65.8168,
        GAMMA=87.0,
        LOOP=3.0$
$OPTINS SREF=0.00375012, CBARR=1.56, BLREF=0.0691$
$SYNTHS XCG=0.85511, ZCG=0.0$
$BODY NX=20.0,
        X(1)=0.0,0.0347,0.0694,0.1041,0.1388,0.1735,0.2082,
                0.2429,0.2776,0.312,0.4348,0.5616,0.6864,0.8112,
                0.936,1.0608,1.1856,1.3104,1.4352,1.56,
        R(1)=0.0,0.003843,0.007685,0.011528,0.015370,0.019213,
                0.0230555,0.026891,0.030741,0.03455,0.03455,0.03455,
                0.03455,0.03455,0.03455,0.03455,0.03455,0.03455,
                0.03455,0.03455,
        S(1)=0.0,0.00004640,0.00018554,0.00041750,0.00074216,
                0.00115969,0.00166993,0.00227177,0.00296883,
                0.00375013,0.00375013,0.00375013,0.00375013,
                0.00375013,0.00375013,0.00375013,0.00375013,
                0.00375013,0.00375013,0.00375013,
        P(1)=0.0,0.02415,0.04829,0.07243,0.09657,0.12072,
                0.14486,0.16891,0.19315,0.21708,0.21708,0.21708,
                0.21708,0.21708,0.21708,0.21708,0.21708,0.21708,
                0.21708,0.21708$
$BODY BNOSE=1.0, BLN=0.312, BLA=1.248$
DIM M
DAMP
CASEID TRABALHO DE GRADUAÇÃO, DERIVADAS DE ESTABILIDADE
NEXT CASE

```

Figura 5.4. Arquivo de entrada para *Digital Datcom*.

Ø INPUT DIMENSIONS ARE IN M , SCALE FACTOR IS 1.0000

1 AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
 CHARACTERISTICS AT ANGLE OF ATTACK AND IN SIDESLIP
 DATCOM BODY ALONE CONFIGURATION
 TRABALHO DE GRADUAÇÃO, DERIVADAS DE ESTABILIDADE

MACH NUMBER	FLIGHT CONDITIONS					REFERENCE DIMENSIONS						
	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LONG.	LENGTH LAT.	MOMENT HORIZ	REF. CENTER VERT		
	M	M/SEC	N/ M**2	DEG K	1/ M	M**2	M	M	M	M		
Ø 0.420	0.00	142.91	1.0133E+05	288.150	9.7397E+06	0.004	1.560	0.069	0.855	0.000		
Ø	DERIVATIVE (PER DEGREE)											
Ø ALPHA	CD	CL	CM	CN	CA	XCP	CLA	CMA	CYB	CNB	CLB	
Ø												
-5.0	0.313	-0.190	-0.0631	-0.216	0.295	0.292	3.412E-02	1.417E-02	-3.412E-02	-3.199E-01	0.000E+00	
-4.5	0.310	-0.169	-0.0575	-0.193	0.295	0.298	4.101E-02	1.139E-02	-4.101E-02	-2.571E-01	0.000E+00	
-4.0	0.307	-0.149	-0.0517	-0.170	0.296	0.305	4.025E-02	1.169E-02	-4.025E-02	-2.640E-01	0.000E+00	
-3.5	0.304	-0.129	-0.0458	-0.147	0.296	0.311	3.949E-02	1.200E-02	-3.949E-02	-2.710E-01	0.000E+00	
-3.0	0.302	-0.109	-0.0397	-0.125	0.296	0.318	3.872E-02	1.231E-02	-3.872E-02	-2.779E-01	0.000E+00	
-2.5	0.300	-0.090	-0.0335	-0.103	0.296	0.325	3.796E-02	1.262E-02	-3.796E-02	-2.849E-01	0.000E+00	
-2.0	0.299	-0.071	-0.0271	-0.082	0.296	0.332	3.719E-02	1.293E-02	-3.719E-02	-2.919E-01	0.000E+00	
-1.5	0.298	-0.053	-0.0206	-0.061	0.296	0.339	3.643E-02	1.324E-02	-3.643E-02	-2.989E-01	0.000E+00	
-0.8	0.297	-0.026	-0.0105	-0.030	0.296	0.350	3.528E-02	1.371E-02	-3.528E-02	-3.094E-01	0.000E+00	
0.0	0.296	0.000	0.0000	0.000	0.296	*****	3.458E-02	1.399E-02	-3.458E-02	-3.157E-01	0.000E+00	
0.5	0.296	0.017	0.0070	0.020	0.296	0.353	3.489E-02	1.386E-02	-3.489E-02	-3.129E-01	0.000E+00	
1.0	0.297	0.035	0.0139	0.040	0.296	0.346	3.566E-02	1.355E-02	-3.566E-02	-3.059E-01	0.000E+00	
1.5	0.298	0.053	0.0206	0.061	0.296	0.339	3.643E-02	1.324E-02	-3.643E-02	-2.989E-01	0.000E+00	
2.0	0.299	0.071	0.0271	0.082	0.296	0.332	3.719E-02	1.293E-02	-3.719E-02	-2.919E-01	0.000E+00	
2.5	0.300	0.090	0.0335	0.103	0.296	0.325	3.796E-02	1.262E-02	-3.796E-02	-2.849E-01	0.000E+00	
3.0	0.302	0.109	0.0397	0.125	0.296	0.318	3.872E-02	1.231E-02	-3.872E-02	-2.779E-01	0.000E+00	
3.5	0.304	0.129	0.0458	0.147	0.296	0.311	3.949E-02	1.200E-02	-3.949E-02	-2.710E-01	0.000E+00	
4.0	0.307	0.149	0.0517	0.170	0.296	0.305	4.025E-02	1.169E-02	-4.025E-02	-2.640E-01	0.000E+00	
4.5	0.310	0.169	0.0575	0.193	0.295	0.298	4.101E-02	1.139E-02	-4.101E-02	-2.571E-01	0.000E+00	
5.0	0.313	0.190	0.0631	0.216	0.295	0.292	4.177E-02	1.108E-02	-4.177E-02	-2.502E-01	0.000E+00	

Figura 5.5. Dados do arquivo de saída (1ª parte).

1 AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
 DYNAMIC DERIVATIVES
 DATCOM BODY ALONE CONFIGURATION
 TRABALHO DE GRADUAÇÃO, DERIVADAS DE ESTABILIDADE

MACH NUMBER	FLIGHT CONDITIONS					REFERENCE DIMENSIONS				
	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LONG.	LENGTH LAT.	MOMENT HORIZ	REF. CENTER VERT
	M	M/SEC	N/ M**2	DEG K	1/ M	M**2	M	M	M	M
Ø 0.420	0.00	142.91	1.0133E+05	288.150	9.7397E+06	0.004	1.560	0.069	0.855	0.000
Ø	DYNAMIC DERIVATIVES (PER DEGREE)									
Ø ALPHA	PITCHING		ACCELERATION		ROLLING		YAWING			
Ø	CLQ	CMQ	CLAD	CMAD	CLP	CYP	CNP	CNR	CLR	
Ø										
-5.00	3.084E-02	-1.292E-02	5.918E-02	-1.013E-03	NDM	NDM	NDM	NDM	0.000E+00	
-4.50									0.000E+00	
-4.00									0.000E+00	
-3.50									0.000E+00	
-3.00									0.000E+00	
-2.50									0.000E+00	
-2.00									0.000E+00	
-1.50									0.000E+00	
-0.75									0.000E+00	
0.00									0.000E+00	
0.50									0.000E+00	
1.00									0.000E+00	
1.50									0.000E+00	
2.00									0.000E+00	
2.50									0.000E+00	
3.00									0.000E+00	
3.50									0.000E+00	
4.00									0.000E+00	
4.50									0.000E+00	
5.00									0.000E+00	

Figura 5.6. Dados do arquivo de saída (2ª parte).

O cálculo das derivadas de estabilidade dimensionais, Z_α , M_α e M_q , de acordo com [4] faz-se pelas seguintes equações

$$Z_\alpha = \frac{1}{2} \rho \mu_0 S C_{Z_\alpha} \quad (5.36)$$

$$M_\alpha = \frac{1}{2} \rho \mu_0 l S C_{M_\alpha} \quad (5.37)$$

$$M_q = \frac{1}{4} \rho \mu_0 l^2 S C_{M_q} \quad (5.38)$$

Os resultados dos cálculos dos fatores e parâmetros da planta variantes no tempo serem apresentados posteriormente ao se apresentar cada uma das 18 plantas consideradas.

5.3 PLANTA DO LILE-2, DETERMINAÇÃO DOS PARÂMETROS DO CONTROLADOR PID, SISTEMA COMPENSADO E TESTES

Para cada um dos 18 valores do ângulo de ataque, α , considerados, foi desenvolvida uma planta específica para o LILE-2. E, por consequência, foi projetado um controlador PID para cada planta. Para tal tarefa repetitiva e trabalhosa foi desenvolvido um programa em MATLAB R2010a com base em sugestões de [10]. O programa foi criado em partes separadas para cada planta. Tal programa também realiza o teste para entrada degrau unitário. Em seguida, foi criado um programa para o teste da entrada decaimento exponencial. E, posteriormente, os parâmetros do controlador e da planta foram usados para criar um programa que representasse o sistema compensado final que trabalha com uma faixa discretizada de valores de α .

Os programas utilizados nas simulação, os dados e os gráficos produzidos por elas encontram-se todos no Anexo II.

5.3.1 Simulação para $\alpha = -4,5^\circ$

Para $\alpha = -4,5^\circ$, foram alcançados os seguintes resultados.

O programa usado para se calcular os parâmetros do controlador encontra-se no Anexo II.

Na Figura 5.7, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.007139  0.04113
x2      1 -0.0005554

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1593
-----
s^2 - 0.006583 s - 0.04113

```

Figura 5.7. Dados de simulação.

Na Figura 5.8, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura 5.8. Dados de simulação.

Na Figura 5.9, encontra-se o gráfico da saída para entrada degrau.

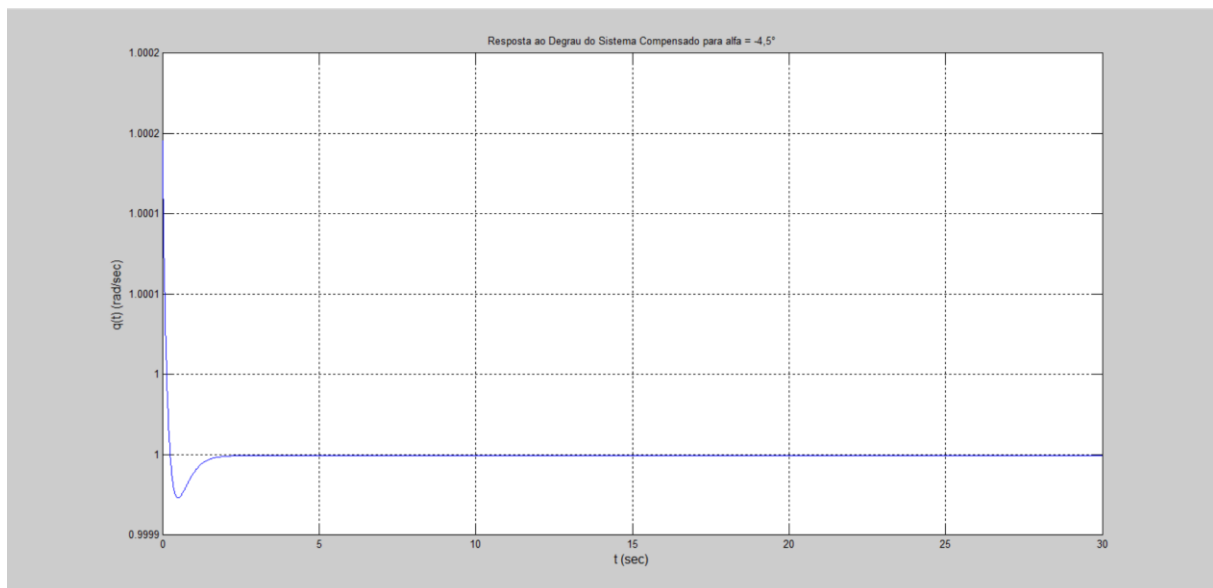


Figura 5.9. Gráfico da saída para a entrada degrau unitário.

O programa usado para se testar a entrada decaimento exponencial encontra-se no Anexo II.

Na Figura 5.10, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada

decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =  
    20  
  
a =  
    4  
  
z_max =  
    1.0002  
  
z_min =  
    3.0719e-010  
  
z_inf =  
    3.0719e-010
```

Figura 5.10. Dados de simulação.

Na Figura 5.11, encontra-se o gráfico da saída para entrada decaimento exponencial.

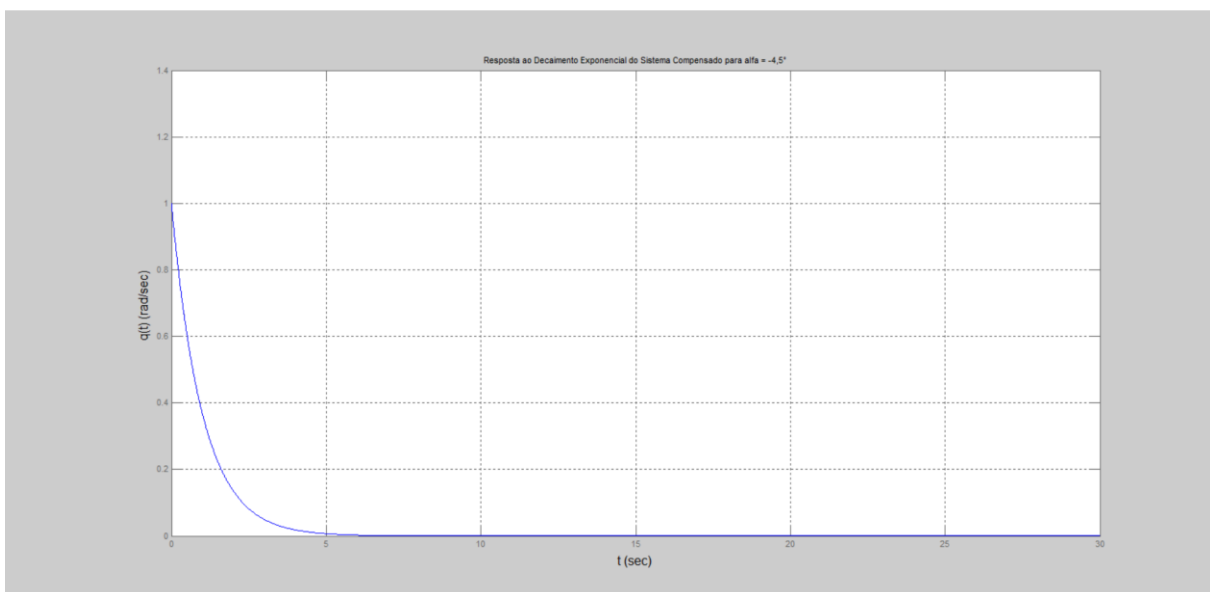


Figura 5.11. Gráfico da saída para entrada decaimento exponencial.

5.3.2 Simulação para $\alpha = -4, 0^\circ$

Para $\alpha = -4, 0^\circ$, tem-se a se destacar o valor da saída no infinito para a entrada decaimento exponencial (Fig. 5.12), para ilustrar o fato de todas as simulações apresentarem valores diferentes neste ponto.

```
K =  
    20  
  
a =  
    4  
  
z_max =  
    1.0002  
  
z_min =  
    3.3265e-010  
  
z_inf =  
    3.3265e-010
```

Figura 5.12. Dados de simulação.

Os programas e os demais dados relativos a esta simulação encontram-se no Anexo II.

5.3.3 Simulação para $\alpha = 0^\circ$

Para $\alpha = 0^\circ$, vale ressaltar o sobrevalor percentual alcançado que se mostrou ligeiramente maior aos das outras simulações (Fig. 5.13), exceto para $\alpha = 0,5^\circ$, que também apresentou o mesmo sobrevalor percentual.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0197
```

Figura 5.13. Dados de simulação.

5.3.4 Simulação para $\alpha = 1,0^\circ$

A simulação para esse valor de α foi escolhida para ilustrar o fato de que as plantas para cada simulação são diferentes (mas, não necessariamente diferentes) de simulação para simulação (Fig. 5.14). Basta-se comparar com o obtido para $\alpha = -4,5^\circ$.

```

a =
      x1      x2
x1  0.02046  0.08282
x2      1 -0.000483

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1581
-----
s^2 - 0.01998 s - 0.08283

```

Figura 5.14. Dados de simulação.

5.3.5 Simulação para $\alpha = 4,5^\circ$

Esta simulação foi escolhida para ilustrar o fato de que para valores simétricos α , no caso, para $\alpha = 4,5^\circ$ e $\alpha = -4,5^\circ$ as plantas são iguais (Fig. 5.15). Isso deve-se ao fato de possuírem as mesmas derivadas de estabilidade.

```

a =
      x1      x2
x1  0.007139  0.04113
x2      1 -0.0005554

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1593
-----
s^2 - 0.006583 s - 0.04113

```

Figura 5.15. Dados de simulação.

5.3.6 Simulação da queima de combustível

As 9 curvas de saída para cada um dos 9 valores (10%, 20%, 30%, ... , 90%) de massa simulados para $\alpha = 3^\circ$ para a entrada degrau unitário encontram-se plotadas no mesmo gráfico na Fig. 5.16 (devido a possuírem valores próximos, as curvas aparentam estarem sobrepostas).

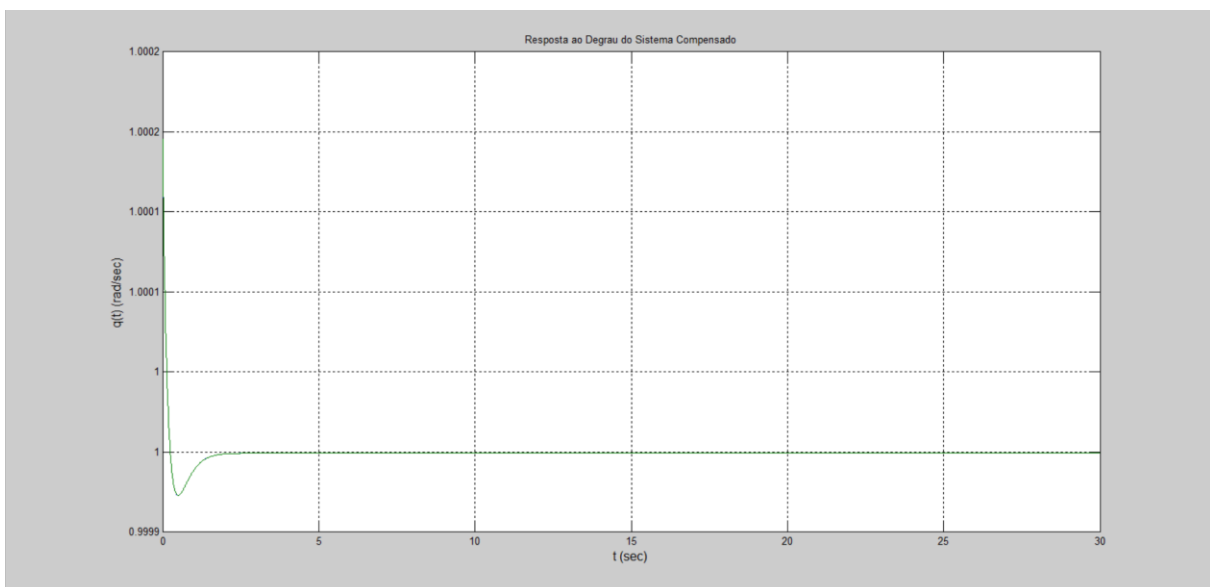


Figura 5.16. Curvas para entrada degrau unitário.

As 9 curvas de saída para cada um dos 9 valores (10%, 20%, 30%, ... , 90%) de massa simulados para $\alpha = 3^\circ$ para a entrada decaimento exponencial encontram-se plotadas no mesmo gráfico na Fig. 5.17 (devido a possuírem valores próximos, as curvas aparentam estarem sobrepostas).

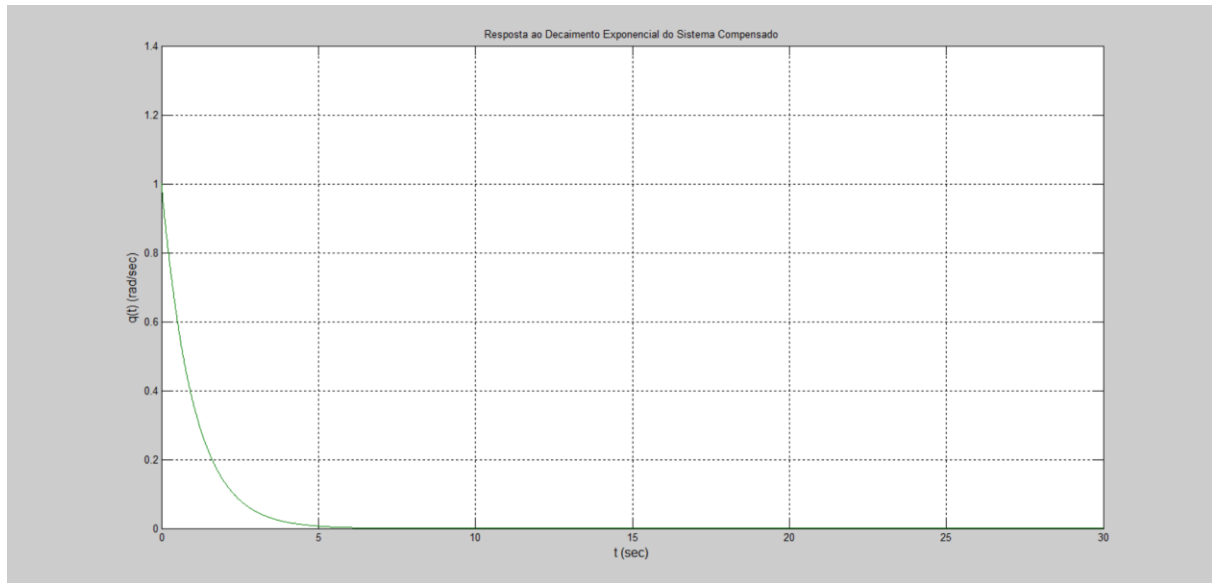


Figura 5.17. Curvas para entrada decaimento exponencial.

Os dados de valores máximo e mínimo, valor final e sobrevalor percentual para cada uma das curvas para entrada degrau unitário encontram-se no Anexo III.

Os dados de valor final cada uma das curvas para entrada decaimento exponencial encontram-se no Anexo III.

Os códigos dos programas utilizados também encontram-se no Anexo III.

5.4 PROGRAMAS PARA O CONTROLADOR DE GANHOS PROGRAMADOS

Com base nos resultados da seção anterior, foi escrito um programa em MATLAB que define o comportamento do controlador de ganhos programados por chaveamento proposto.

Foram escritos o código principal e mais duas funções: *FunTran* e *saida*. O código principal recebe como entrada um valor do ângulo de ataque, α , dentre os valores testados e um valor referente a cada uma das entradas testadas: 1 (um) para entrada degrau unitário e 2 (dois) para entrada decaimento exponencial. A função *FunTran* recebe como entrada o valor do ângulo de ataque e as derivadas de estabilidade não-dimensionais, $C_{Z\alpha}$ e $C_{M\alpha}$, e gera a função de transferência em malha fechada. E a função *saida* recebe como entrada a função de transferência em malha fechada e o valor do tipo de entrada selecionada e gera a saída correspondente em gráfico. Além disso, o programa gera as matrizes $A, B, C, e D$, da representação em espaço de estados da planta, a função de transferência da planta, os

parâmetros do controlador, os valores máximo e mínimo da saída, o valor final (em estado estacionário) da saída e, para a entrada degrau unitário, o sobrevalor percentual.

O código principal, a função *FunTran* e a função *saida* encontram-se no Anexo 3.

A seguir são apresentados resultados do programa.

Na Figura 5.18, para $\alpha = -3,5^\circ$ e entrada degrau unitário, encontra-se as linhas de pedidos de dados e os dados gerados pelo programa e, na Fig. 5.19, o gráfico de saída.

```

Digite o valor do ângulo de ataque em graus: -3.5
Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: 1

a =
      x1      x2
x1  0.009157  0.04907
x2      1 -0.0005348

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

Transfer function:
      -255.9 s - 0.1573
-----
s^2 - 0.008622 s - 0.04908

K =
      20

a =
      4

y_max =
      1.0002

y_min =
      1.0000

y_inf =
      1.0000

SVP =
      0.0196

```

Figura 5.18. Dados de simulação.

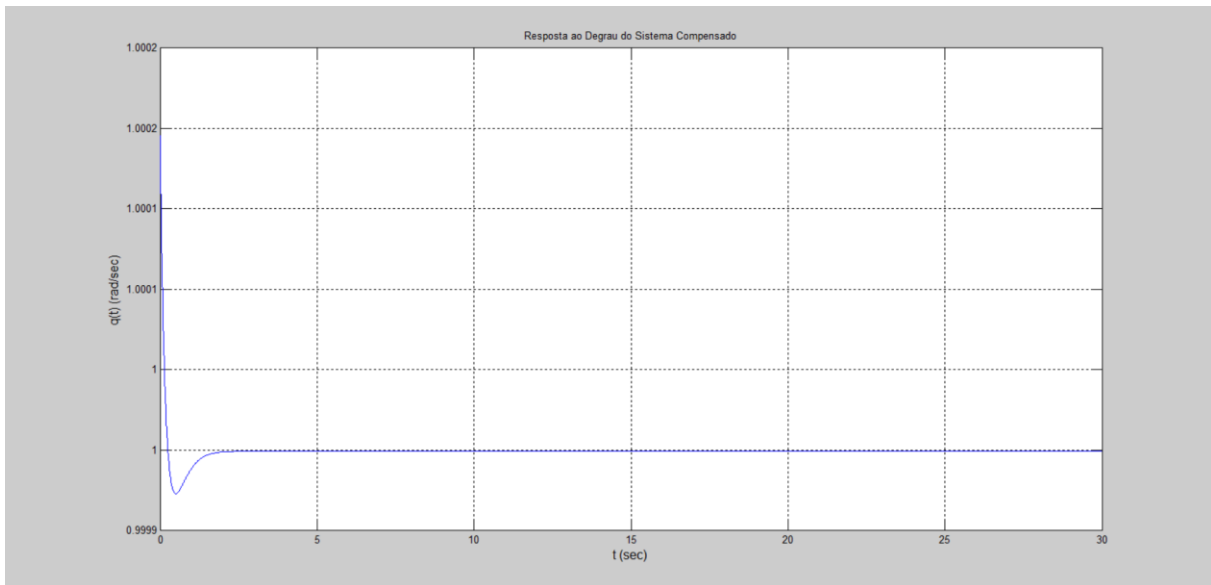


Figura 5.19. Gráfico da saída para entrada degrau unitário.

Na Figura 5.20, para $\alpha = 1,0^\circ$ e para entrada decaimento exponencial, as linhas de pedidos de dados e os dados gerados pelo programa e, na Fig. 5.21, o gráfico de saída.

```

Digite o valor do ângulo de ataque em graus: 1
Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: 2

a =
      x1      x2
x1  0.02046  0.08282
x2      1 -0.000483

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

Transfer function:
      -255.9 s - 0.1581
-----
s^2 - 0.01998 s - 0.08283

K =

      20

a =

      4

z_max =

      1.0002

z_min =

      6.1409e-010

z_inf =

      6.1409e-010

```

Figura 5.20. Dados de simulação.

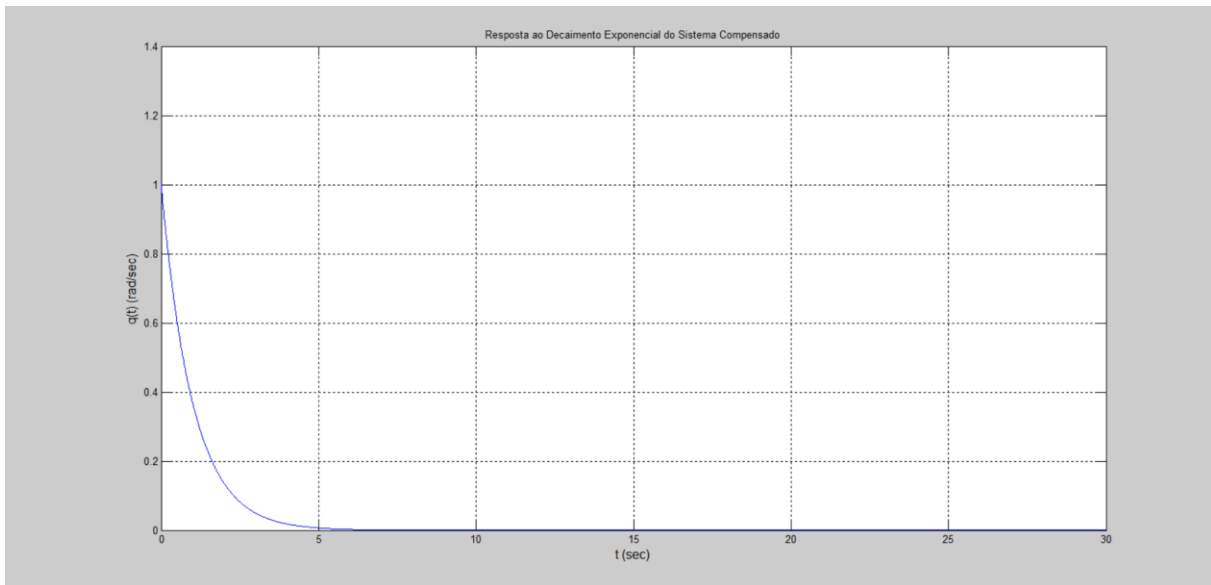


Figura 5.21. Gráfico da saída para entrada decaimento exponencial.

Após isso, foi desenvolvido um controlador de ganhos programados por interpolação. Similar para o caso anterior, foi escrito um programa em MATLAB. Ele possui um código principal e quatro funções: as funções $C_{z\alpha}$ e $C_{m\alpha}$, que calculam a derivadas de estabilidade não-dimensiionais $C_{z\alpha}$ e $C_{m\alpha}$, respectivamente, por meio de interpolação; A função $FunTranConti$, que calcula a função de transferência em malha fechada; e a função $saída$, que gera os gráficos para as saídas. Além disso, o programa gera as matrizes $A, B, C, e D$, da representação em espaço de estados da planta, a função de transferência da planta, os parâmetros do controlador, os valores máximo e mínimo da saída, o valor final (em estado estacionário) da saída e o sobrevalor percentual para a entrada degrau unitário; para a entrada decaimento exponencial, o programa gera o valor final. O programa também informa os valores de $C_{z\alpha}$ e $C_{m\alpha}$ estimados pelas interpolações.

O *software Digital Datcom* calculou 18 valores de $C_{z\alpha}$ e $C_{m\alpha}$ em função de α . O presente program em MATLAB faz a interpolação desses valores para $C_{z\alpha}$ e para $C_{m\alpha}$, substituindo o chaveamento de 18 sistemas compensados por uma única equação com valores contínuos para α . Os gráficos da interpolação de $C_{z\alpha}$ e $C_{m\alpha}$ encontram-se nas Fig. 5.22 e 5.23, respectivamente.

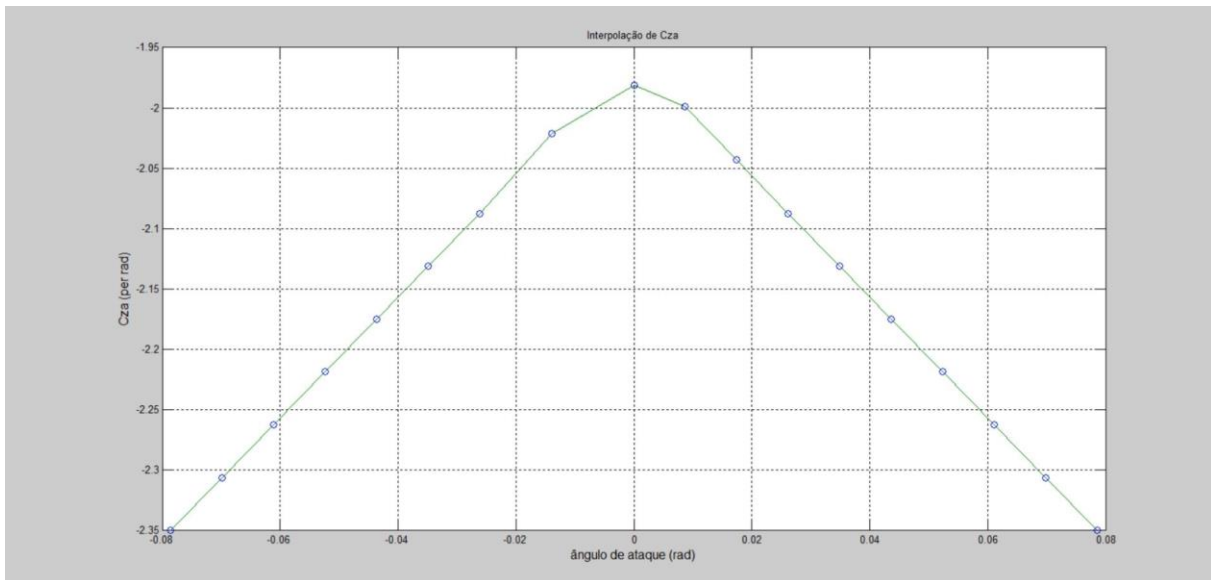


Figura 5.22. Interpolação de $C_{z\alpha}$.

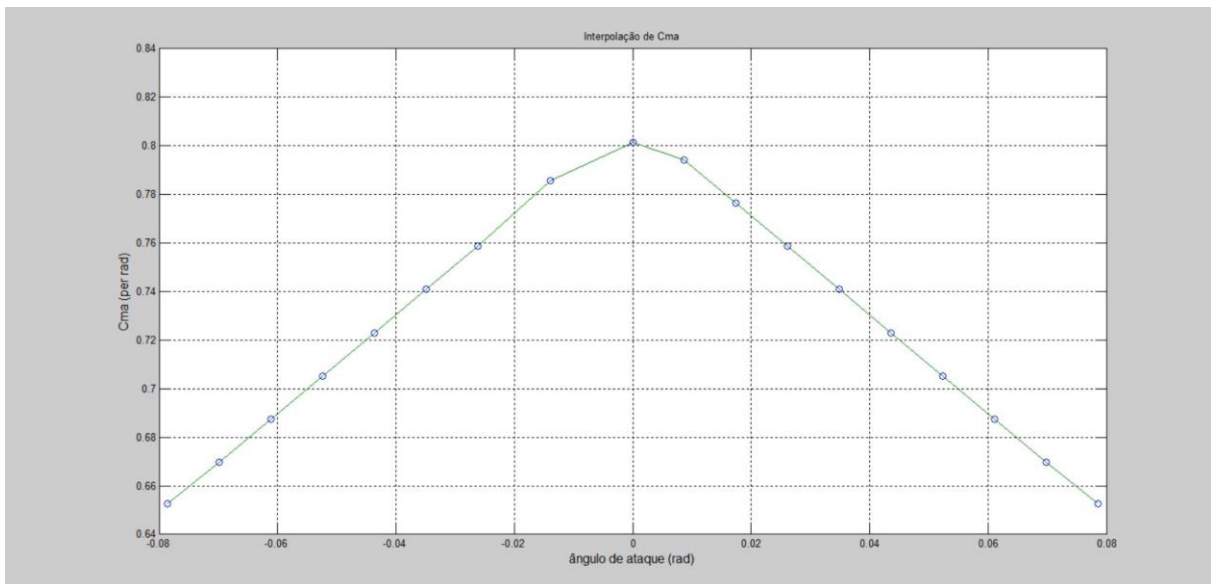


Figura 5.23. Interpolação de $C_{m\alpha}$

Foram feitos testes para entrada degrau unitário e decaimento exponencial para os valores $\alpha = 1,852^\circ$ e $\alpha = -4,279^\circ$. Os dados para $\alpha = 1,852^\circ$ e entrada degrau unitário encontram-se na Fig. 5.24, para entrada decaimento exponencial encontra-se na Fig. 5.25. Os dados para $\alpha = -4,279^\circ$ e entrada degrau unitário encontram-se na Fig. 5.26, para entrada decaimento exponencial encontra-se na Fig. 5.27. Os gráficos para essas saídas encontram-se no Anexo IV.

```

Cza =
    -2.1179

Cma =
    0.7461

a =
           x1      x2
x1    0.01495    0.06804
x2         1   -0.0006258

b =
           u1
x1   -255.9
x2   -0.5209

c =
           x1  x2
y1    1    0

d =
           u1
y1    0

Continuous-time model.

Transfer function:
    -255.9 s - 0.1956
-----
s^2 - 0.01433 s - 0.06805

```

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura 5.24. Dados de simulação.

```
z_inf =  
    6.2153e-010
```

Figura 5.25. Dados de simulação.

Cza =

-2.3305

Cma =

0.6602

a =

	x1	x2
x1	0.007524	0.04271
x2	1	-0.0006886

b =

	u1
x1	-255.9
x2	-0.5209

c =

	x1	x2
y1	1	0

d =

	u1
y1	0

Continuous-time model.

Transfer function:

-255.9 s - 0.1985

s^2 - 0.006835 s - 0.04272

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura 5.26. Dados de simulação.

```
z_inf =  
    3.9577e-010
```

Figura 5.27. Dados de simulação.

6 ANÁLISE DOS RESULTADOS

Apresenta-se aqui uma análise dos resultados obtidos na seção anterior.

6.1 DESENHO MECÂNICO DA FUSELAGEM

O projeto mecânico original da fuselagem possui apresenta um nariz arredondado e quatro aletas na base. No entanto, foram feitas simplificações. O modelo matemático adotado é um foguete sem aletas e, de acordo com o manual do *software Digital Datcom*, o nariz precisava ser cônico. As mudanças foram mínimas e apropriadas para o trabalho. Visto isso, o desenho mecânico considerado mostra-se adequado.

6.2 PARÂMETROS DA PLANTA

Os fatores que formam os parâmetros da planta foram pesquisados, escolhidos ou calculados de forma minuciosa para que gerassem parâmetros e modelos precisos, coerentes e os resultados esperados. Foram muitos fatores, de pesquisa extensa e detalhosa para determiná-los. Os modelos permitiram resultados com sucesso, levando a crer que os parâmetros e seus fatores foram trabalhados.

6.3 PLANTA DO LILE-2, DETERMINAÇÃO DOS PARÂMETROS DO CONTROLADOR PID, SISTEMA COMPENSADO E TESTES

Para cada um dos valores do ângulo de ataque selecionados, foi calculada uma planta diferente. Cada uma das plantas se apresentou dentro do esperado com menos zeros finitos do que pólos finitos.

Apesar de as plantas para cada ângulo de ataque serem diferentes, os parâmetros do controlador foram iguais para todos, $K = 20$ e $a = 4$, ou seja, o controlador é igual para todas as plantas.

Para a entrada degrau unitário, as especificações de projeto para o sistema compensado eram Sobrevalor Percentual: $\%SP = 20\%$ e Tempo de Assentamento: $T_s = 1s$. Para uma parte dos sistemas compensados (um para cada ângulo de ataque), o Sobrevalor Percentual atingido foi $\%SP = 0,0196\%$ e para a outra foi $\%SP = 0,0197\%$. Em todos os sistemas compensados, os valores máximo e mínimo da saída foram respectivamente, $1,0002$ e $1,0000$, para uma margem de assentamento entre $0,98$ e $1,02$, ou seja, a saída manteve dentro do tempo de assentamento desde o instante inicial. Tanto o Sobrevalor Percentual, quanto o Tempo de Assentamento atingidos se mantiveram bem abaixo do especificado. O erro em regime estacionário não foi especificado, mas foi tentado anulá-lo mesmo assim por meio de

um controlador PID que aumentasse a ordem do sistema. Para todos os sistemas compensados o erro em regime estacionário foi nulo.

Para a entrada decaimento exponencial, usada para se testar o sistema com valor zero na entrada (visto que o LILE-2 terá como referência o valor zero), temos que, em todos os sistemas compensados, o valor máximo (valor inicial) é 1,0002, e o valor mínimo (valor final) encontra-se na casa de 10^{-10} . Tal como o esperado, saída tende a zero no infinito.

Para a simulação da queima de combustível, os dados mostram que, para a entrada degrau unitário, o sistema de controle se apresentou dentro das especificações para todos os valores de massa; para a entrada decaimento exponencial a saída acompanhou a entrada no valor final para todos os valores de massa. Portanto, o sistema de controle pode ser considerado robusto.

6.4 O CONTROLADOR DE GANHOS PROGRAMADOS

O controlador de ganhos programados por chaveamento, apresenta o mesmo comportamento que os testes individuais anteriores, mostrando que foi bem implementado.

O controlador de ganhos programados por interpolação apresentou valores estimados de C_{z_α} e C_{m_α} conforme esperado e, para os valores testados (diferentes dos valores por chaveamento e com três casas decimais), foram geradas saídas dentro das especificações. Portanto, foi obtido sucesso para esse controlador.

7 CONCLUSÕES

A seguir tem-se as conclusões finais relativas ao projeto e sugestões de trabalhos futuros.

7.1 CONCLUSÕES

Foi desenvolvido dois controladores de ganhos programados, um por chaveamento e outro por interpolação, para controlar, em nível de simulação computacional, o movimento de arfagem do foguete LILE-2 da *Hybrid Propulsion Team*, equipe de alunos de Engenharias da UnB. A parte discretizada refere-se à discretização da faixa de valores do ângulo de ataque, α , considerado, $-4,5^\circ$ a $4,5^\circ$. Foram escolhido 18 valores ao longo dessa faixa. Termos mantidos como variantes no tempo são todos dependentes de α . Dessa forma, discretizando os valores, foi possível obter 18 plantas (diferentes) invariantes no tempo. Para cada planta foi projetado (parte contínua) um controlador PID. E por fim, foi criado um programa em *MATLAB R2010a* para unir todos os sistemas compensados originários desse controle em um único controlador híbrido em que a seleção de determinado ângulo de ataque α leva à efetivação de um dos 18 sistemas compensados e, por consequência, à correção precisa da orientação.

Como principais resultados, tem-se a determinação dos parâmetros da planta (ou melhor de cada uma das 18 plantas), especificações alcançadas bem menores que as especificações de projeto e a concepção, em nível de simulação, de controladores de ganhos programados, por chaveamento e por interpolação.

A determinação dos parâmetros é considerada um dos principais resultados, pois foi um trabalho extenso que envolveu pesquisa para saber como obtê-los, dados de projeto de subsistemas do foguete, páginas e páginas de cálculos, consultas a trabalhos de outros colegas da *Hybrid Propulsion Team*. Em particular, fazer a revisão bibliográfica sobre as derivadas de estabilidade e ainda encontrar (e aprender a usar) um programa (*Digital Datcom*) que calculasse as derivadas de estabilidade não-dimensionais foram etapas que consumiram muitíssima leitura, testes no programa e tempo e proporcionaram muita aprendizagem. Ademais, a etapa da determinação dos parâmetros foi feita minuciosamente, pois o sucesso no desenvolvimento dos controladores dependeria diretamente deles.

Quanto às especificações alcançadas, como visto na análise dos resultados, foram bem abaixo do especificado para sobrevalor percentual e tempo de assentamento. O sobrevalor percentual foi especificado em 20% para entrada degrau unitário e o atingido foi de 0,0196% para algumas plantas e 0,0197% para as outras. E, o tempo de assentamento especificado foi 1s, enquanto que a saída para entrada degrau unitário manteve-se na margem de 2% em desde o instante inicial. Para o erro em regime estacionário, foi atingido o valor zero tanto para entrada degrau unitário, quanto para entrada decaimento exponencial.

A implementação dos controladores de ganhos programados foi uma consequência natural dos testes em cada planta e sistema compensado. Para surpresa, foi conseguido um controlador PID idêntico para todos os valores do ângulo de ataque considerados discretizado. Foi possível os dois tipos controladores de ganhos programados com êxito, tendo-os como produtos finais do trabalho.

Houve tentativas de outras metodologias para o projeto do controlador. No entanto, os resultados alcançados não foram melhores. Houve casos de grande erro de estado estacionário e ou de tempo de assentamento absurdamente longo.

Foram encontradas muitas dificuldades. Iniciar o trabalho sem um professor-orientador específico da área (foi possível ter acesso a um em um curto período) e também terminar sem um. Para encontrar bibliografia afim sob essas condições, foram necessárias muita leitura que não proporcionavam proveito algum ao trabalho. Após encontrar a bibliografia, foram necessárias muitas releituras, visto que muito conteúdo não tinha com tirar dúvidas e eram assuntos novos escritos em outros idiomas. Houve dificuldades para me informar sobre *softwares* necessários para o projeto e como usá-los (não somente o *Digital Datcom*, mencionado acima). Foram testados três tipos de controladores até se encontrar um que atendesse às especificações. Além de, como mencionado, foram necessárias inúmeras páginas de cálculos. Apesar disso, houve persistência e o trabalho foi concluído.

Os resultados foram dentro do esperado ou acima, como no caso das especificações, as aprendizagens com projeto foram muitas e intensas na área de Controle e na área Aeroespacial, na idealização, execução e concepção de um projeto de engenharia chegando a resultados satisfatórios e à superação de todas as dificuldades encontradas. Visto isso, pode-se concluir que houve pleno êxito na execução do trabalho.

7.2 TRABALHOS FUTUROS

Para trabalhos futuros, faz-se as seguintes recomendações.

Fazer o cálculo dos parâmetros do controlador incrementando os valores de K e a , em vez de decrementar, como foi feito neste trabalho, para se encontrar controladores com valores menores nos parâmetros para, por exemplo, ter um ganho menor e, consequência, um sobrevalor menor e um menor custo na implementação física.

Fazer um controle adaptativo a fim de se considerar os termos variantes no tempo.

Tentar implementar fisicamente o sistema de controle com atuador e jet-vanes. Para atuador, recomenda-se um motor DC por possuir um modelo matemático linearizado e para o material dos jet-vanes recomendo aço 1020 [11].

Recomenda-se também replicar o projeto do controlador para controle do movimento de guinada, já que para o presente trabalho ambos os projetos são idênticos.

Caso se faça o controle do três ângulos de atitude e integre-os num único trabalho com o uso de jet-vanes, haveria um par de jet-vanes para um movimento e outro para para dois movimentos. Recomenda-se um par para o movimento de arfagem (Pitch), pois além do controle Pitch pode-se implementar o sistema de guiagem; e outro par para os movimentos de guinada (Yaw) e rolagem (Roll). Os movimentos guinada e rolagem não poderão ocorrer simultaneamente. Como é imprescindível que o movimento que rolagem seja nulo para as equações dos três sistemas não sejam acopladas (independentes), recomenda-se prioridade ao movimento de rolagem. Dessa forma, pode ser tentado o uso de um circuito digital para decidir a escolha entre esses dois movimentos.

Para os sensores pode-se usar uma IMU do tipo MEMS (*micro electro mechanical system*) de baixo custo e calibrá-lo conforme [12].

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SWOKOWSKI, E. W. **Cálculo com geometria analítica**. 2ª ed. São Paulo, SP: Makron Books, 1994. Vol. 2
- [2] NISE, N. S. **Engenharia de sistemas de controle**. 5ª ed. Rio de Janeiro, RJ: LTC, 2009, 684 p.
- [3] HOUGHTON, E. L., CARPENTER, P. W. **Aerodynamics for engineering students**. 5ª ed. Burlington, MA, UK: Butterworth-Heinemann, 2003, 590 p.
- [4] ETKIN, B., REID, L. D. **Dynamics of flight: stability and control**. 3ª ed. Toronto, Canada: John Wiley & Sons, 1996, 382 p.
- [5] ROSA, C. F. **Série de Taylor e aplicações**. 2013. 56 p. Monografia de TCC (Licenciatura em Matemática) – Universidade Federal de Santa Catarina, Florianópolis, SC.
- [6] TEWARI, A. **Advanced control of aircraft, spacecraft and rockets**. Noida, India: John Wiley & Sons, 2011. 390 p.
- [7] NISE, N. S. **Control systems engineering**. 4ª ed. EUA: John Wiley & Sons, 2004, 969 p.
- [8] NOGUEIRA, F. G., TORRICO, B. C., BARRA JÚNIOR, W., DA COSTA JÚNIOR, C. T., BARREIROS, J. A. L., DE LANA, J. J. **Identificação e controle LPV aplicados ao projeto de um estabilizador de sistemas de potência em um gerador de 350 MVA**. In: XX Congresso brasileiro de automática. Belo Horizonte, MG. 2014. (Anais)
- [9] NELSON, C. R., **Flight stability and automatic control**. EUA: McGraw-Hill, 1989. 284 p.
- [10] OGATA, K., **Modern control engineering**. 4ª ed. New Jersey, EUA: Prentice Hall, 2002. 964 p.
- [11] GILADETT, L. V., WINEMAN, A. R., Investigation of vanes immersed in the jet of a solid-fuel rocket motor. In: NACA. Washington, EUA. 1952.
- [12] Tedaldi, D., Pretto, A., Menegatti, E. A robust and easy to implement method for IMU calibration without external equipments. In: ICRA. Hong Kong, China. 2014.

ANEXOS

I Simulação de vôo para o LILE-2

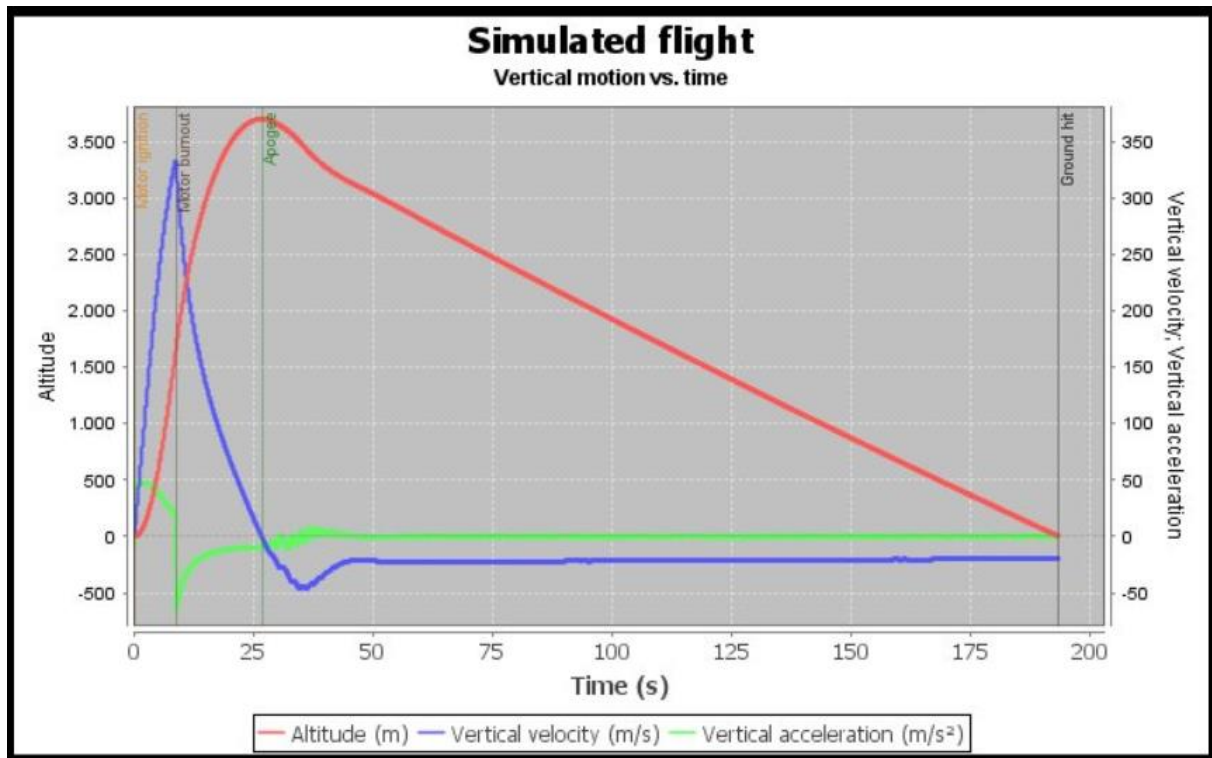


Figura I.1. Altura, velocidade e aceleração.

II Simulações para os valores discretizados do ângulo de ataque

II.1 Simulação para $\alpha = -4,5^\circ$

Na Figura II.1, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-4.5;

Czalfag=-0.04101;
Cmalfag=0.01139;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```



```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end

    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -4,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.1. Programa em MATLAB.

Na Figura II.2, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.007139  0.04113
x2      1 -0.0005554

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1593
-----
s^2 - 0.006583 s - 0.04113

```

Figura II.2. Dados de simulação.

Na Figura II.3, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.3. Dados de simulação.

Na Figura II.4, encontra-se o gráfico da saída para entrada degrau.

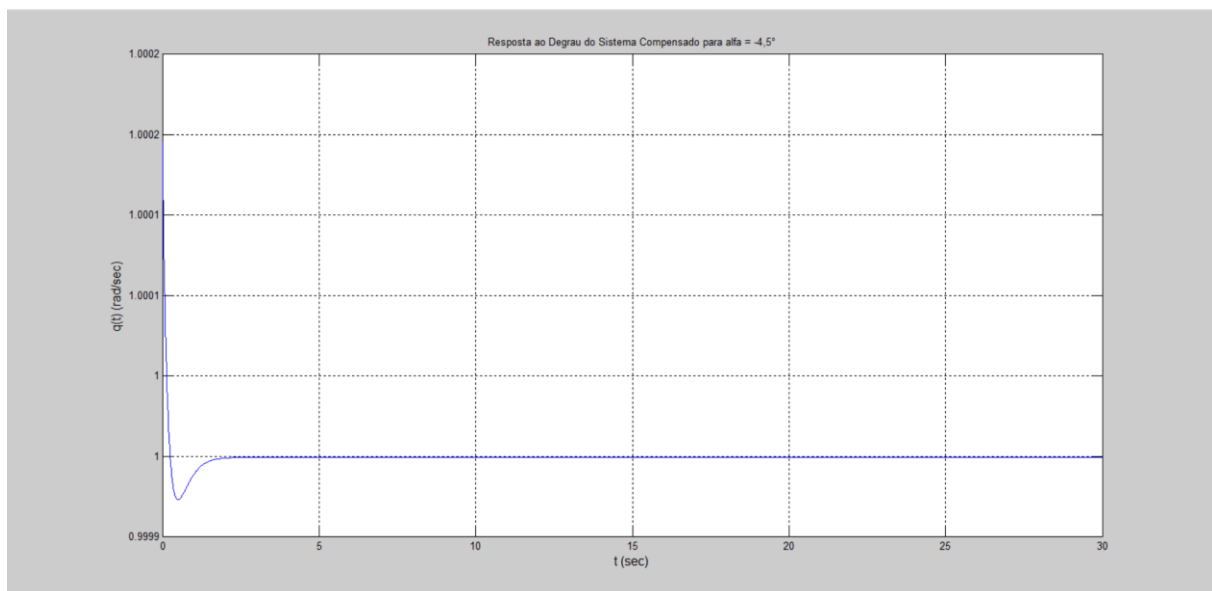


Figura II.4. Gráfico da saída para a entrada degrau unitário.

Na Figura II.5, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgastadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
numl=K*[1 2*a a^2];
denl=[0 1 0];
Gc=tf(numl,denl);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -4,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.5. Programa em MATLAB.

Na Figura II.6, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
     4

z_max =
    1.0002

z_min =
    3.0719e-010

z_inf =
    3.0719e-010

```

Figura II.6. Dados de simulação.

Na Figura II.7, encontra-se o gráfico da saída para entrada decaimento exponencial.

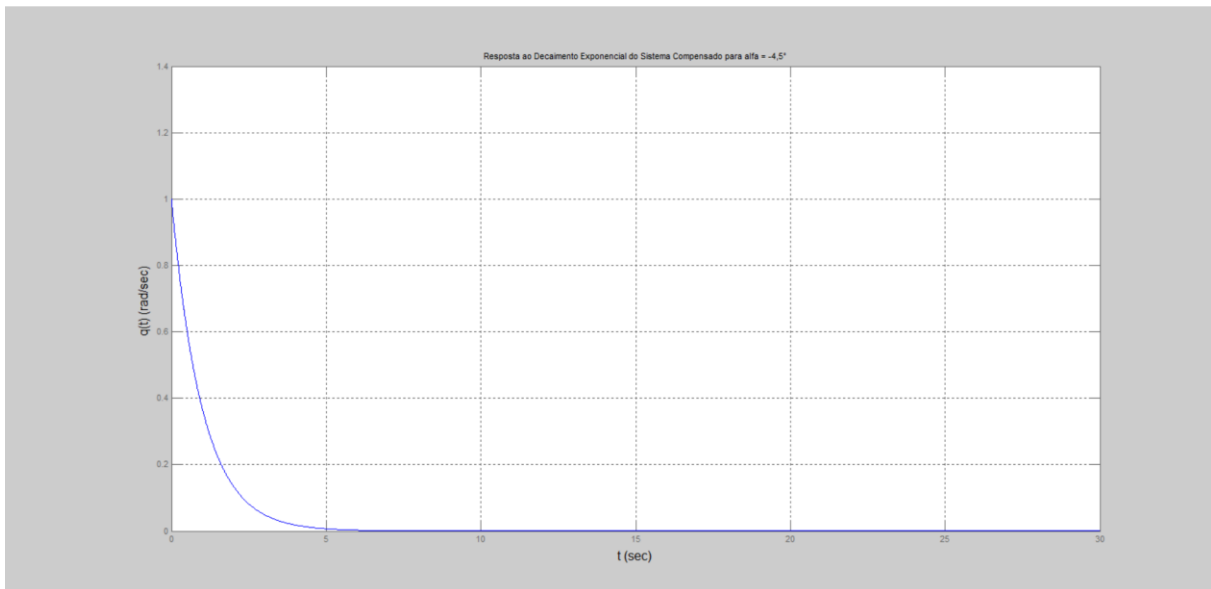


Figura II.7. Gráfico da saída para a entrada decaimento exponencial.

II.2 Simulação para $\alpha = -4, 0^\circ$

Na Figura II.8, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-4;

Czalfag=-0.04025;
Cmalfag=0.01169;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -4,0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.8. Programa em MATLAB.

Na Figura II.9, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.008055  0.04484
x2      1 -0.0005451

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1582
-----
s^2 - 0.00751 s - 0.04484

```

Figura II.9. Dados de simulação.

Na Figura II.10, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.


```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
|  
SVP =  
    0.0196
```

Figura II.10. Dados de simulação.

Na Figura II.11, encontra-se o gráfico da saída para entrada degrau.

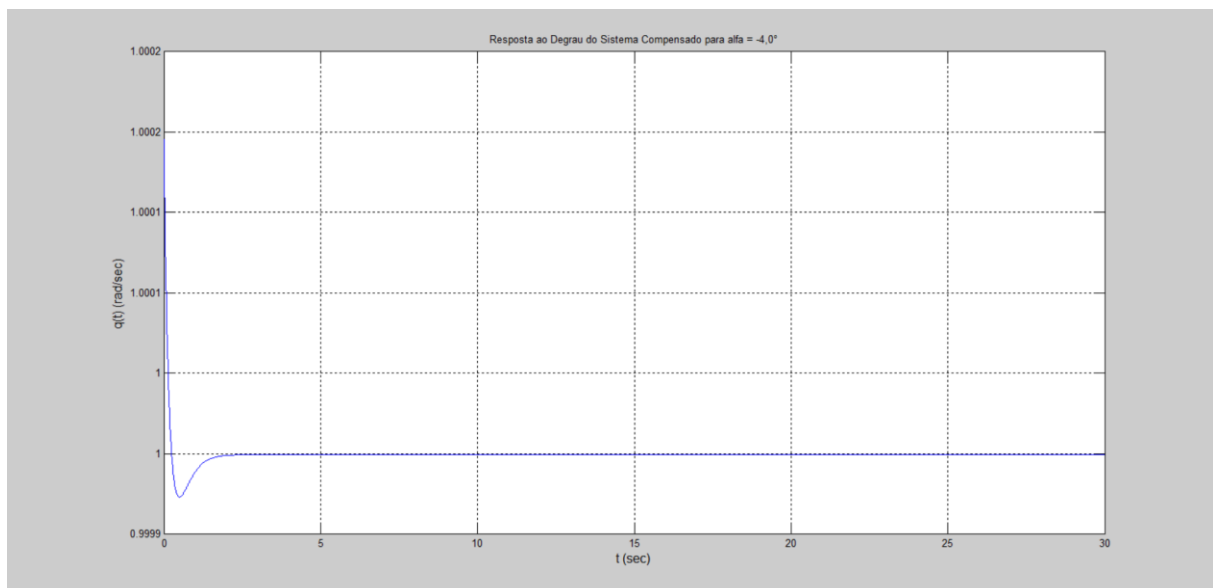


Figura II.11. Gráfico da saída para a entrada degrau.

Na Figura II.12, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -4,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.12. Programa em MATLAB.

Na Figura II.13, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    3.3265e-010

z_inf =
    3.3265e-010

```

Figura II.13. Dados de simulação.

Na Figura II.14, encontra-se o gráfico da saída para entrada decaimento exponencial.

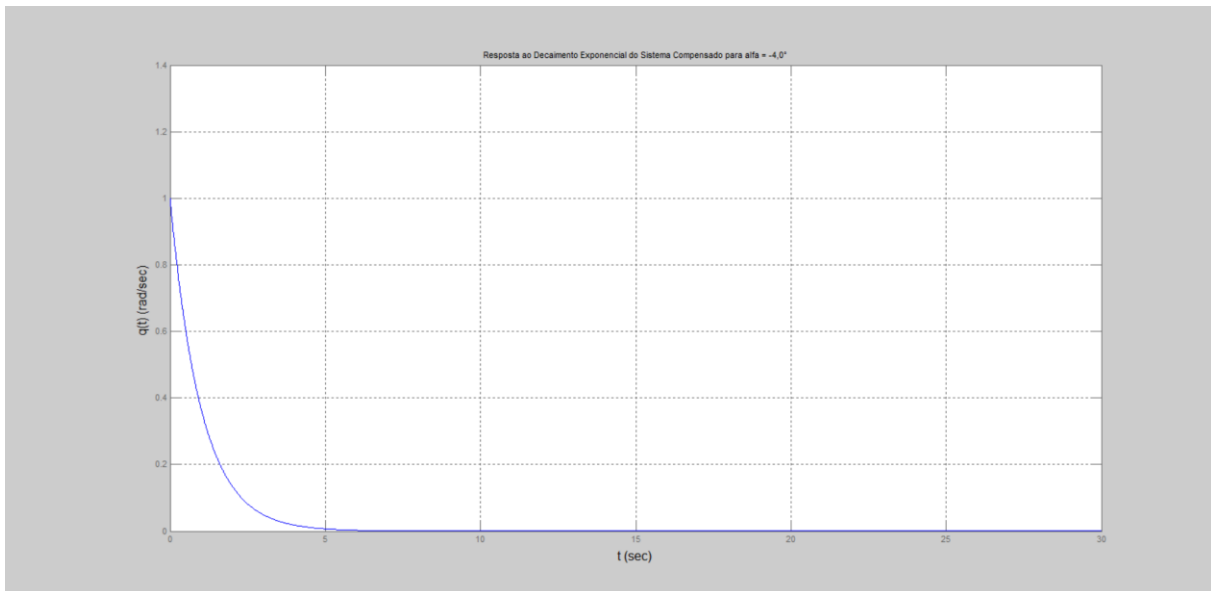


Figura II.14. Gráfico da saída para entrada decaimento exponencial.

II.3 Simulação para $\alpha = -3,5^\circ$

Na Figura II.15, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-3.5;

Czalfag=-0.03949;
Cmalfag=0.012;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

!G(s)=!
G=tf(sys)

```

```

for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -3,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.15. Programa em MATLAB.

Na Figura II.16, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.009157  0.04907
x2      1 -0.0005348

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1573
-----
      s^2 - 0.008622 s - 0.04908

```

Figura II.16. Dados de simulação.

Na Figura II.17, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.17. Dados de simulação.

Na Figura II.18, encontra-se o gráfico da saída para entrada degrau.

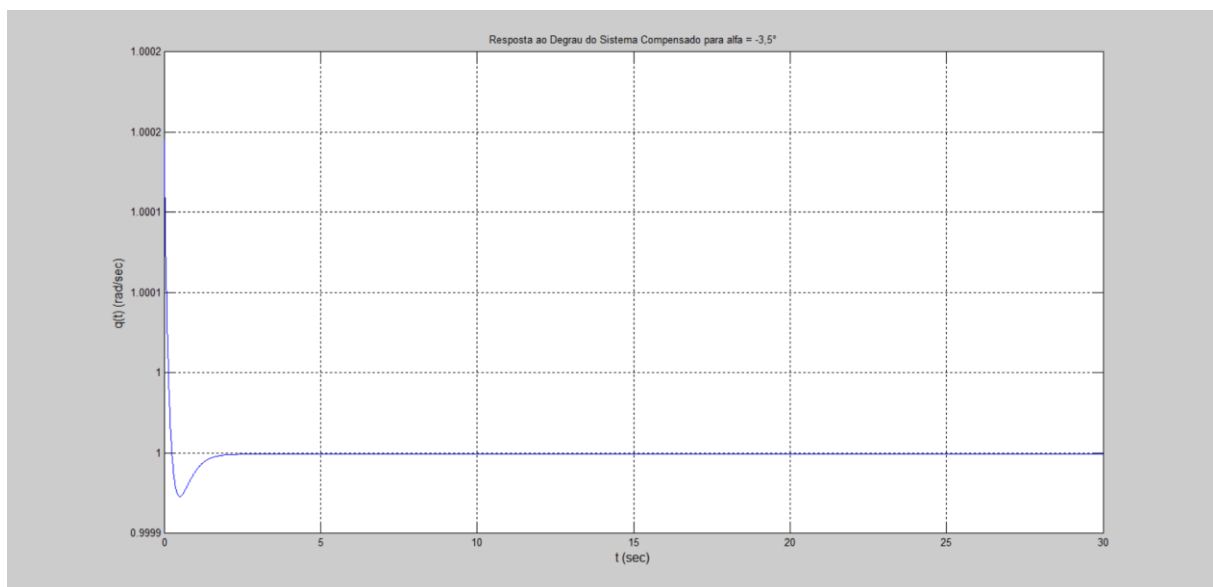


Figura II.18. Gráfico da saída para entrada degrau unitário.

Na Figura II.19, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```
K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -3,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)
```

Figura II.19. Programa em MATLAB.

Na Figura II.20, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =
    20

a =
     4

z_max =
    1.0002

z_min =
    3.6210e-010

z_inf =
    3.6210e-010
```

Figura II.20. Dados de simulação.

Na Figura II.21, encontra-se o gráfico da saída para entrada decaimento exponencial.

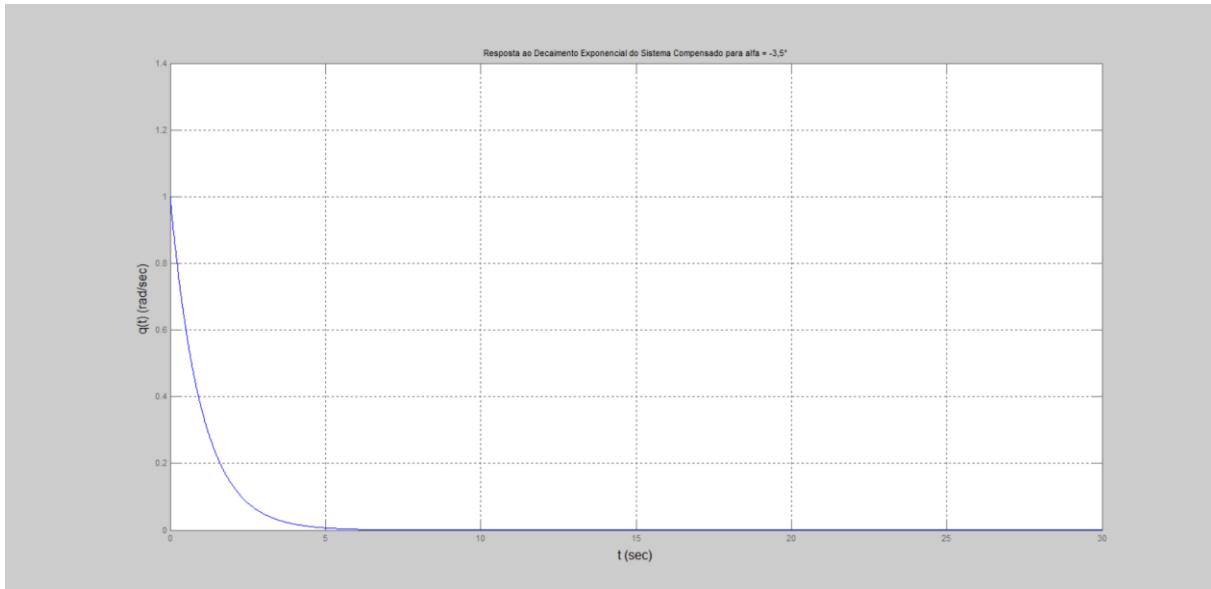


Figura II.21. Gráfico da saída para entrada decaimento exponencial.

II.4 Simulação para $\alpha = -3, 0^\circ$

Na Figura II.22, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-3;

Czalfag=-0.03872;
Cmalfag=0.01231;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -3,0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.22. Programa em MATLAB.

Na Figura II.23, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.0105  0.05389
x2      1 -0.0005244

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1567
-----
s^2 - 0.009972 s - 0.0539

```

Figura II.23. Dados de simulação.

Na Figura II.24, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.24. Dados de simulação.

Na Figura II.25, encontra-se o gráfico da saída para entrada degrau.

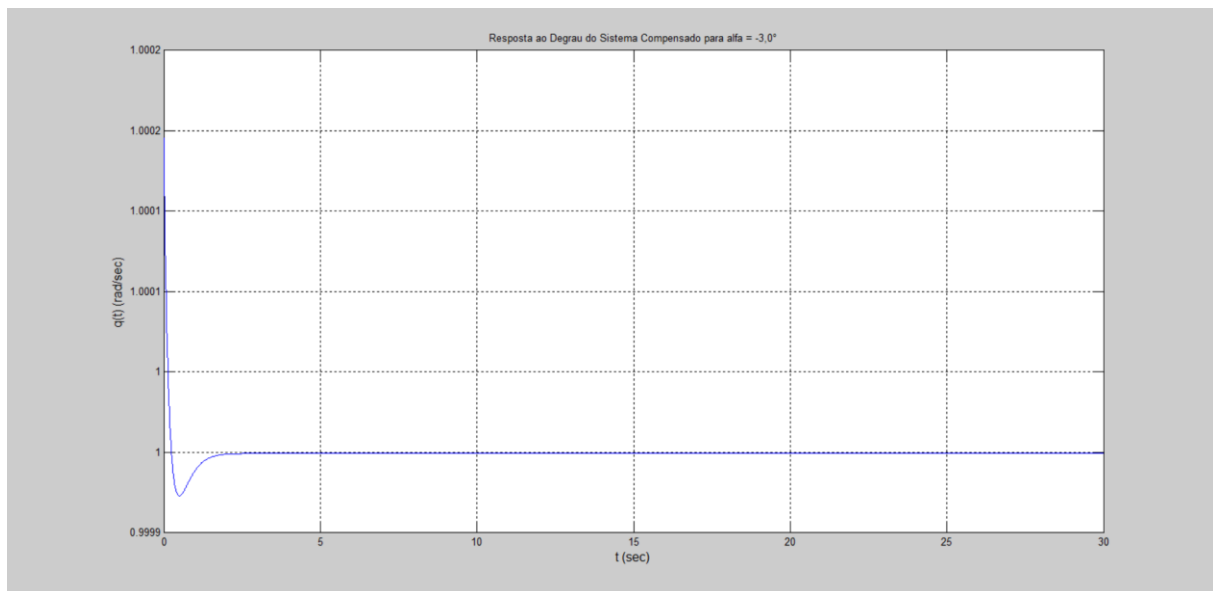


Figura II.25. Gráfico da saída para entrada degrau.

Na Figura II.26, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```
K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -3,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)
```

Figura II.26. Programa em MATLAB.

Na Figura II.27, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =
    20

a =
     4

z_max =
    1.0002

z_min =
    3.9604e-010

z_inf =
    3.9604e-010
```

Figura II.27. Dados de simulação.

Na Figura II.28, encontra-se o gráfico da saída para entrada decaimento exponencial.

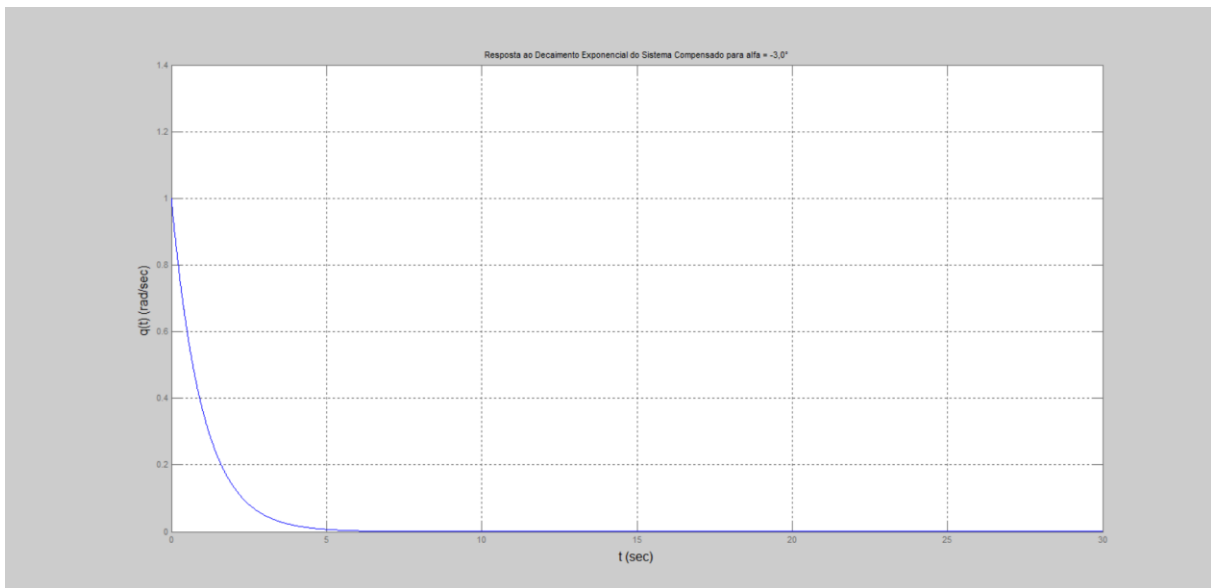


Figura II.28. Gráfico da saída para a entrada degrau unitário.

II.5 Simulação para $\alpha = -2,5^\circ$

Na Figura II.29, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-2.5;

Czalfag=-0.03796;
Cmalfag=0.01262;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*1*S*Cmalfa;
Mq=0.25*roh*mi0*1^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```



```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -2,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.29. Programa em MATLAB.

Na Figura II.30, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01215  0.05944
x2      1 -0.0005141

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1563
-----
s^2 - 0.01163 s - 0.05945

```

Figura II.30. Dados de simulação.

Na Figura II.31, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.31. Dados de simulação.

Na Figura II.32, encontra-se o gráfico da saída para entrada degrau.

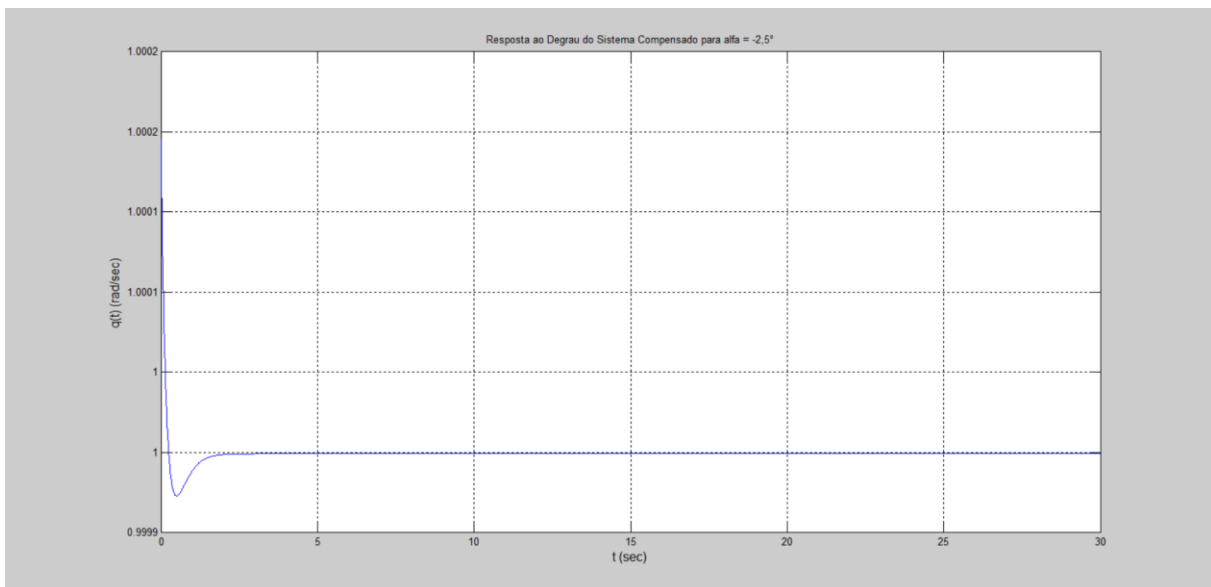


Figura II.32. Gráfico da saída para entrada degrau unitário.

Na Figura II.33, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -2,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.33. Programa em MATLAB,

Na Figura II.34, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    4.3593e-010

z_inf =
    4.3593e-010

```

Figura II.34. Dados de simulação.

Na Figura II.35, encontra-se o gráfico da saída para entrada decaimento exponencial.

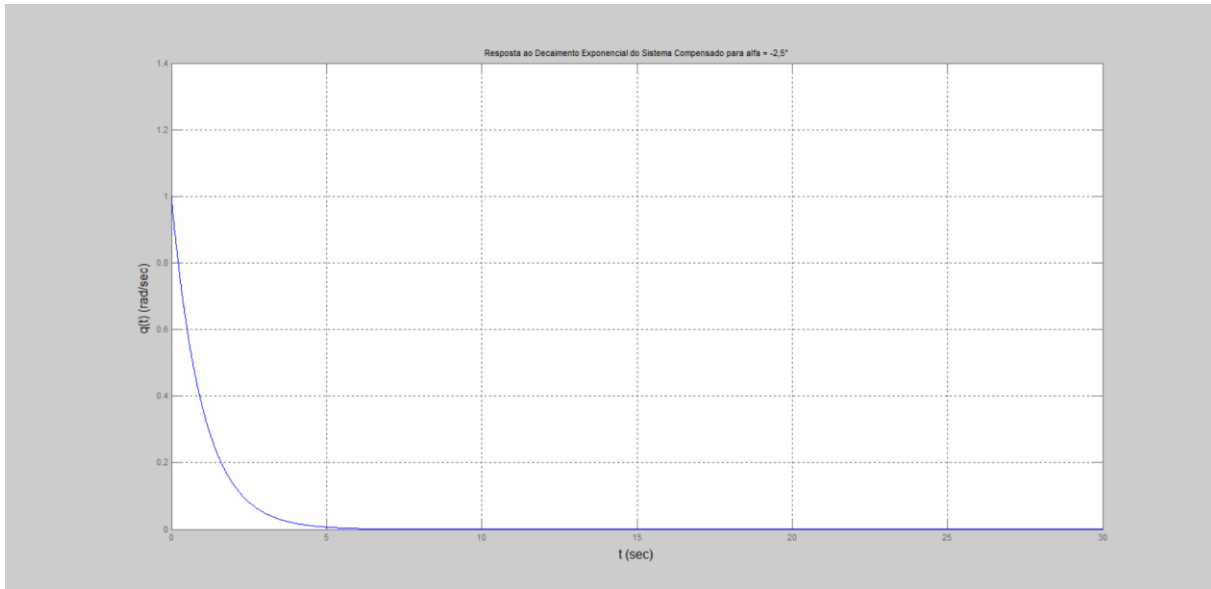


Figura II.35. Gráfico da saída para entrada decaimento exponencial.

II.6 Simulação para $\alpha = -2, 0^\circ$

Na Figura II.36, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-2;

Czalfag=-0.03719;
Cmalfag=0.01293;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*1*S*Cmalfa;
Mq=0.25*roh*mi0*1^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=l;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -2,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.36. Programa em MATLAB.

Na Figura II.37, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01423  0.06591
x2      1 -0.0005037

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1564
-----
s^2 - 0.01373 s - 0.06592

```

Figura II.37. Dados de simulação.

Na Figura II.38, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.


```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.38. Dados de simulação.

Na Figura II.39, encontra-se o gráfico da saída para entrada degrau.

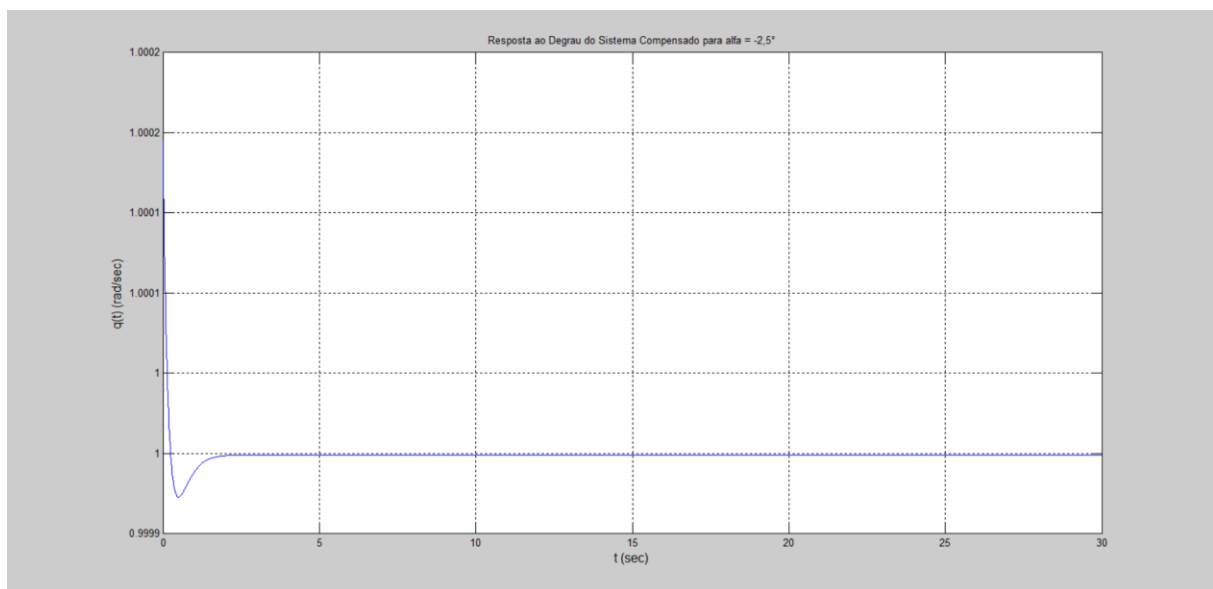


Figura II.39. Gráfico da saída para entrada degrau unitário.

Na Figura II.40, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```
K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -2,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)
```

Figura II.40. Programa em MATLAB.

Na Figura II.41, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =
    20

a =
     4

z_max =
    1.0002

z_min =
    4.8345e-010

z_inf =
    4.8345e-010
```

Figura II.41. Dados de simulação.

Na Figura II.42, encontra-se o gráfico da saída para entrada decaimento exponencial.

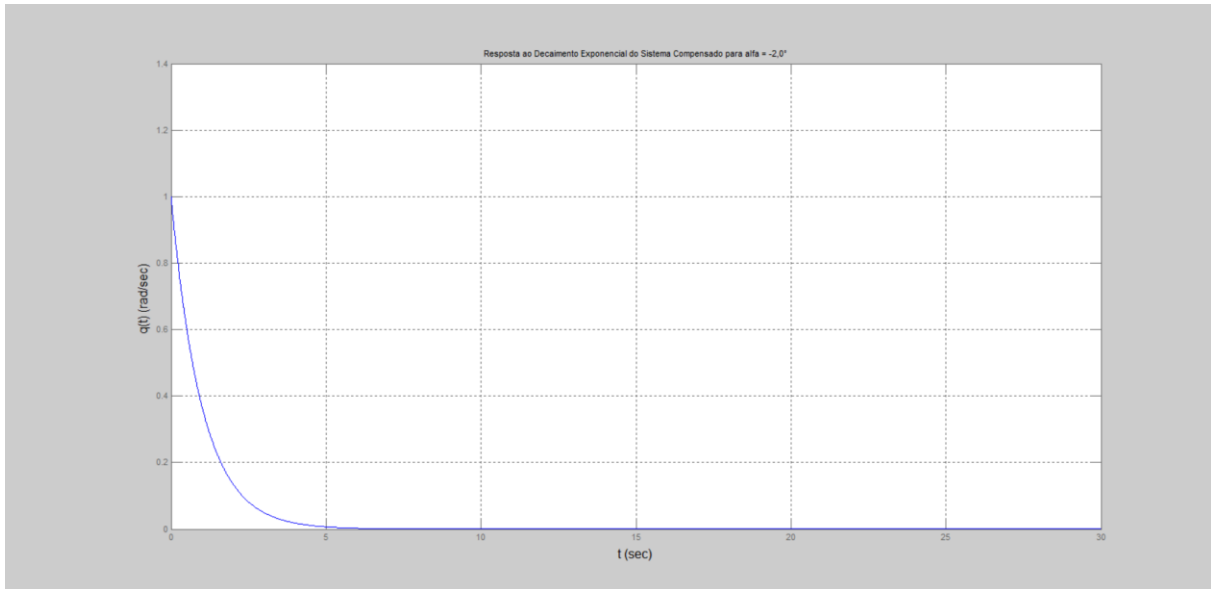


Figura II.42. Gráfico da saída para a entrada decaimento exponencial.

II.7 Simulação para $\alpha = -1,5^\circ$

Na Figura II.43, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-1.5;

Czalfag=-0.03643;
Cmalfag=0.01324;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*1*S*Cmalfa;
Mq=0.25*roh*mi0*1^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -1,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.43. Programa para MATLAB.

Na Figura II.44, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01691  0.07358
x2      1 -0.0004934

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1569
-----
s^2 - 0.01642 s - 0.07359

```

Figura II.44. Dados de simulação.

Na Figura II.45, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
|  
SVP =  
    0.0196
```

Figura II.45. Dados de simulação.

Na Figura II.46, encontra-se o gráfico da saída para entrada degrau.

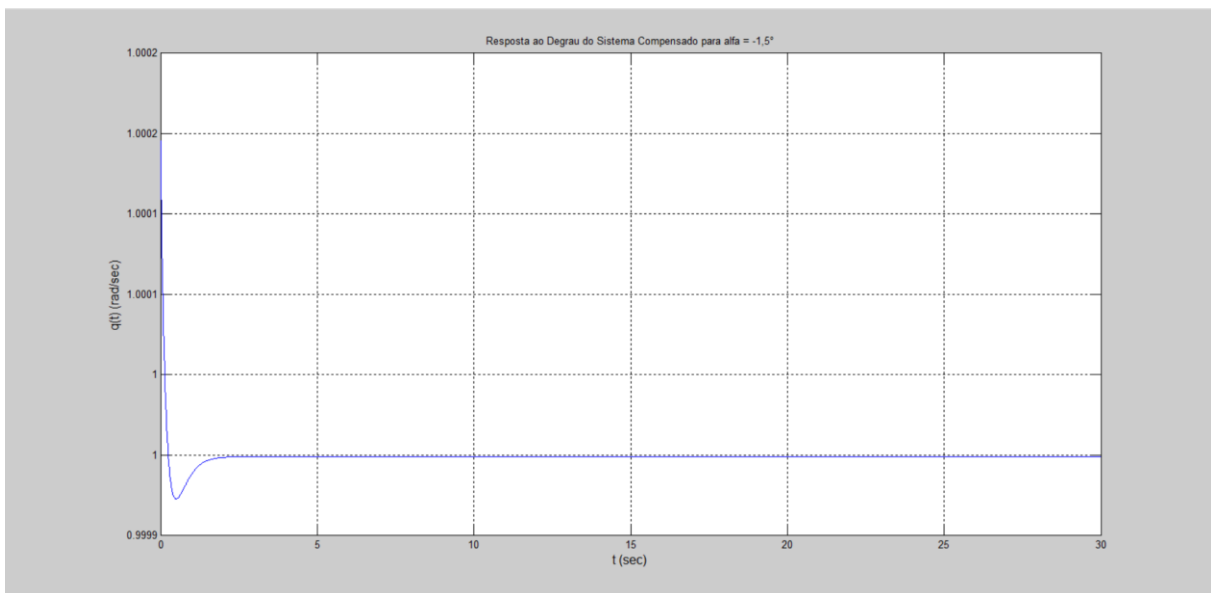


Figura II.46. Gráfico da saída para entrada degrau unitário.

Na Figura II.47, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```
K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -1,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)
```

Figura II.47. Programa em MATLAB.

Na Figura II.48, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =
    20

a =
     4

z_max =
    1.0002

z_min =
    5.4159e-010

z_inf =
    5.4159e-010
```

Figura II.48. Dados de simulação.

Na Figura II.49, encontra-se o gráfico da saída para entrada decaimento exponencial.

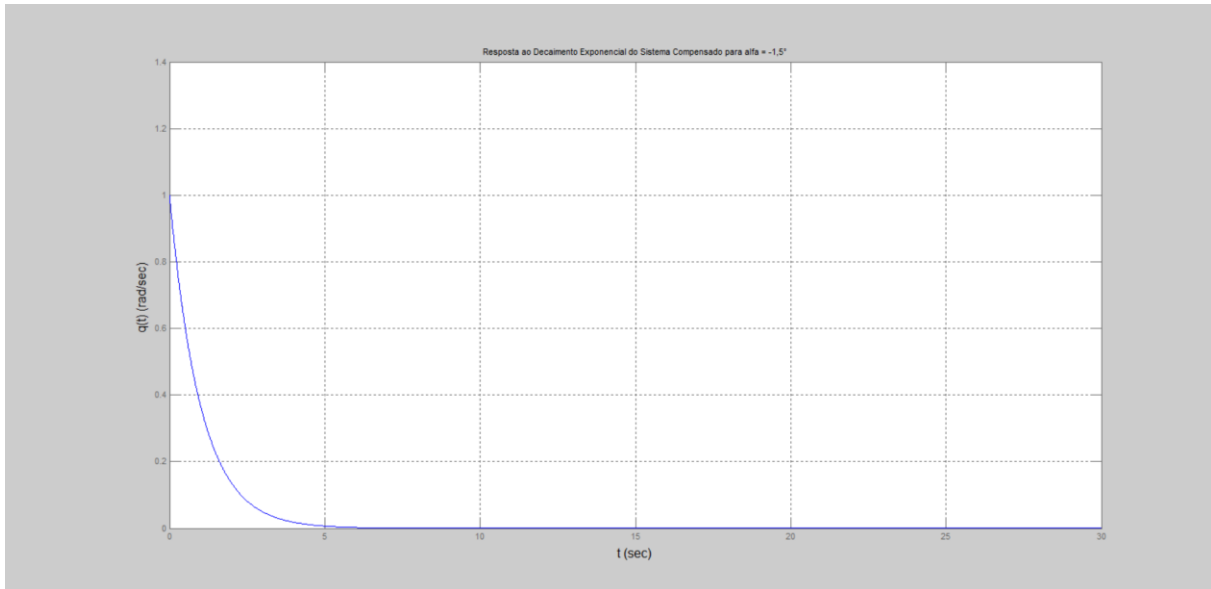


Figura II.49. Gráfico da saída para entrada decaimento exponencial.

II.8 Simulação para $\alpha = -0,8^\circ$

Na Figura II.50, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=-0.8;

Czalfag=-0.03528;
Cmalfag=0.01371;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end

    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = -0,8°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.50. Programa para MATLAB.

Na Figura II.51, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1    0.0222    0.0873
x2     1 -0.0004778

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1    1  0

d =
      u1
y1    0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1587
-----
s^2 - 0.02172 s - 0.08731

```

Figura II.51. Dados de simulação.

Na Figura II.52, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.52. Dados de simulação.

Na Figura II.53, encontra-se o gráfico da saída para entrada degrau.

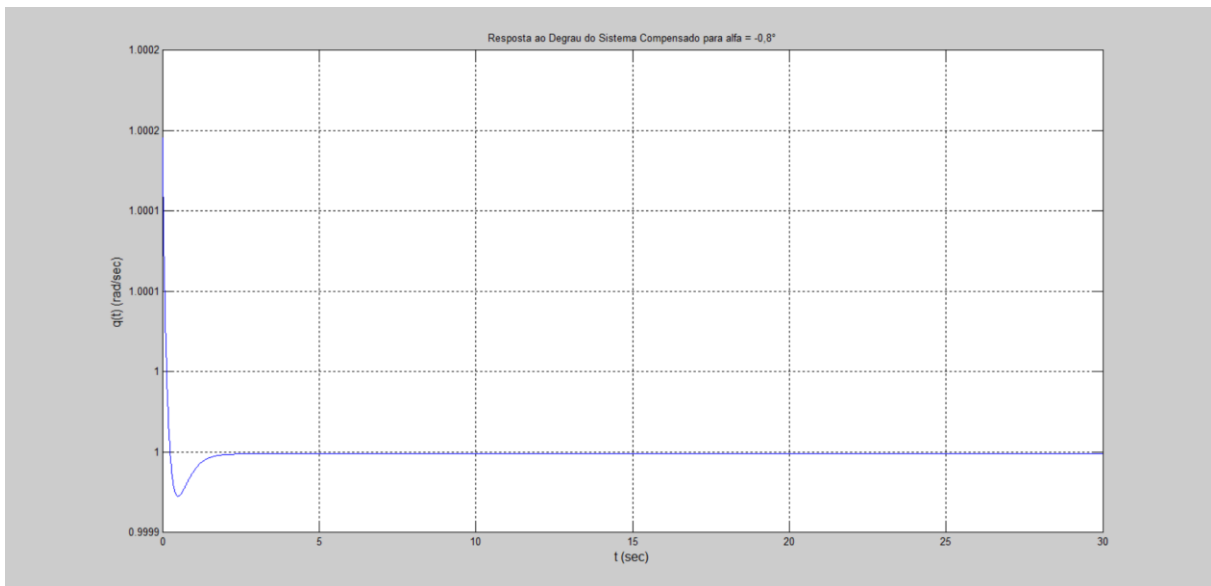


Figura II.53. Gráfico da saída para entrada degrau.

Na Figura II.54, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = -0,8°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.54. Programa para MATLAB.

Na Figura II.55, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    6.4945e-010

z_inf =
    6.4945e-010

```

Figura II.55. Dados de simulação.

Na Figura II.56, encontra-se o gráfico da saída para entrada decaimento exponencial.

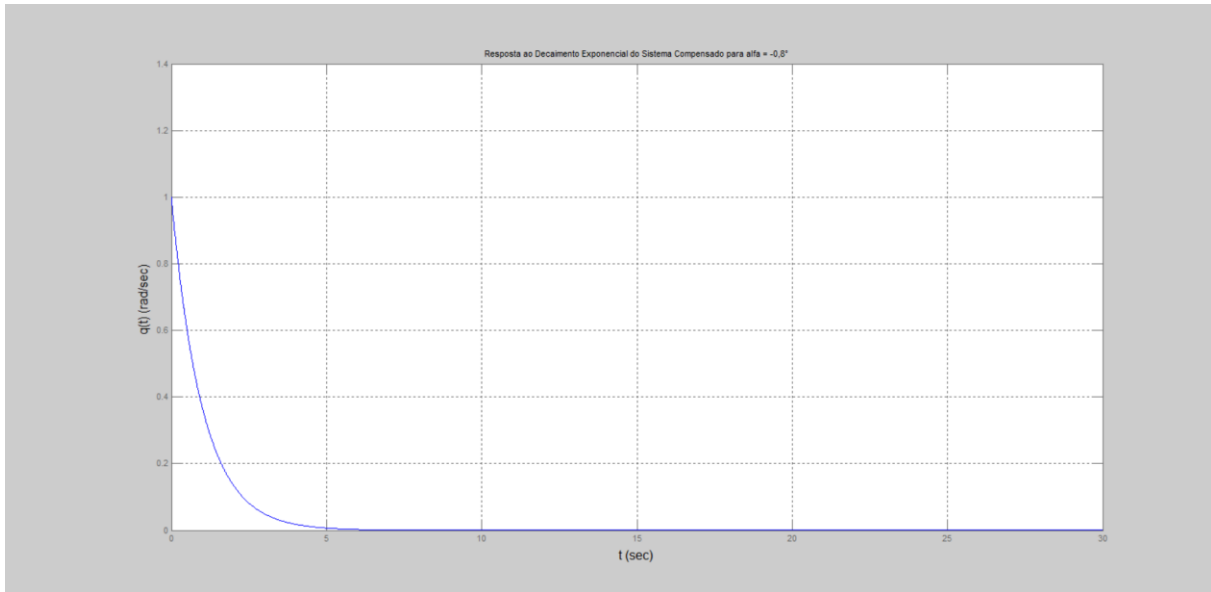


Figura II.56. Gráfico da saída para entrada decaimento exponencial.

II.9 Simulação para $\alpha = 0^\circ$

Na Figura II.57, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=0;

Czalfag=-0.03458;
Cmalfag=0.01399;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```



```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.57. Programa para MATLAB.

Na Figura II.58, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.03191  0.1068
x2      1 -0.0004683

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1644
-----
s^2 - 0.03145 s - 0.1068

```

Figura II.58. Dados de simulação.

Na Figura II.59, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

K =
20
a =
4
y_max =
1.0002
y_min =
1.0000
y_inf =
1.0000
SVP =
0.0197

Figura II.59. Dados de simulação.

Na Figura II.60, encontra-se o gráfico da saída para entrada degrau.

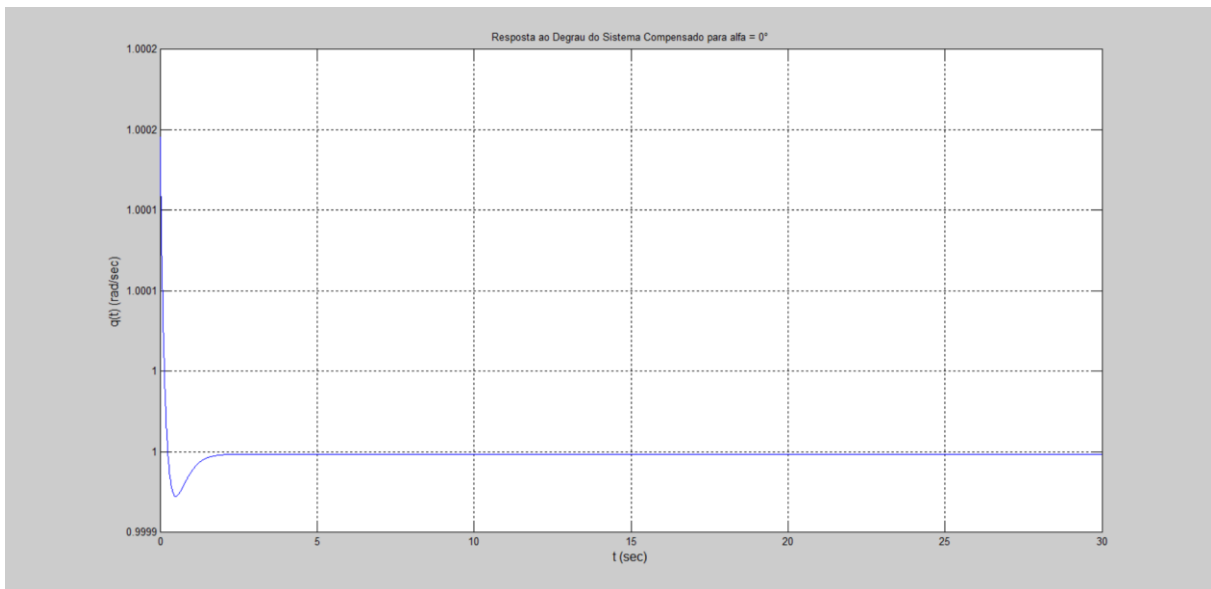


Figura II.60. Gráfico da saída para entrada degrau unitário.

Na Figura II.61, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.61. Programa para MATLAB.

Na Figura II.62, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
     4

z_max =
    1.0002

z_min =
    8.2259e-010

z_inf =
    8.2259e-010

```

Figura II.62. Dados de simulação.

Na Figura II.63, encontra-se o gráfico da saída para entrada decaimento exponencial.

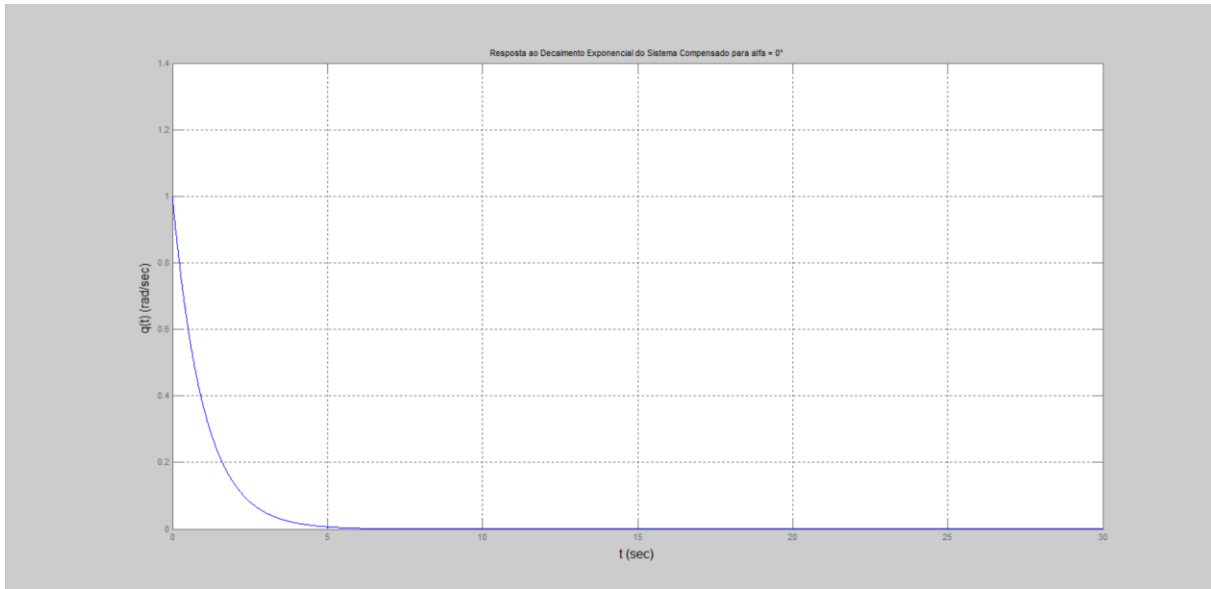


Figura II.63. Gráfico da saída para a entrada decaimento exponencial.

II.10 Simulação para $\alpha = 0,5^\circ$

Na Figura II.64, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=0.5;

Czalfag=-0.03489;
Cmalfag=0.01386;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*1*S*Cmalfa;
Mq=0.25*roh*mi0*1^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=l;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 0,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.64. Programa para MATLAB.

Na Figura II.65, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.02526  0.09414
x2      1 -0.0004725

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1602
-----
s^2 - 0.02479 s - 0.09415

```

Figura II.65. Dados de simulação.

Na Figura II.66, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.


```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0197
```

Figura II.66. Dados de simulação.

Na Figura II.67, encontra-se o gráfico da saída para entrada degrau.

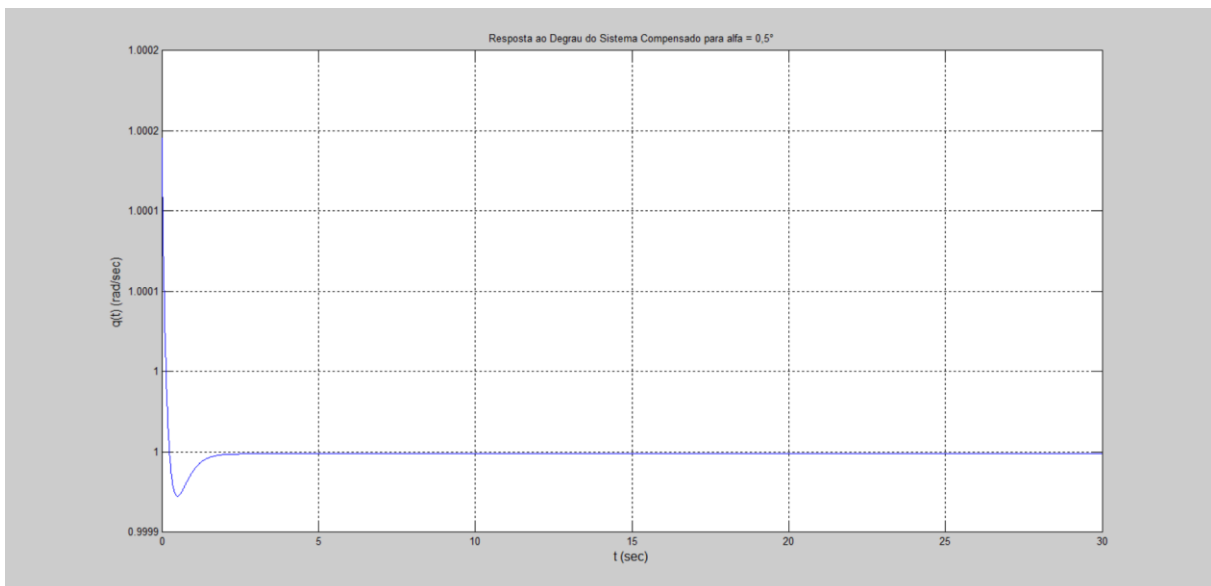


Figura II.67. Gráfico da saída para a entrada degrau.

Na Figura II.68, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 0,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.68. Programa para MATLAB.

Na Figura II.69, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
     4

z_max =
    1.0002

z_min =
    7.0684e-010

z_inf =
    7.0684e-010

```

Figura II.69. Dados de simulação.

Na Figura II.70, encontra-se o gráfico da saída para entrada decaimento exponencial.

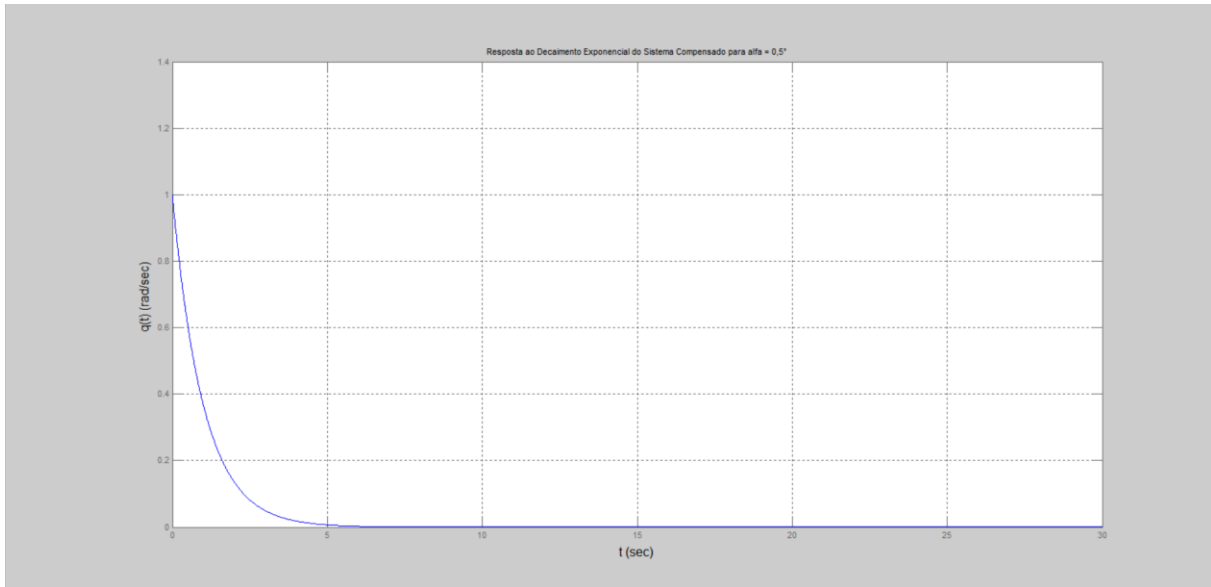


Figura II.70. Gráfico da saída para entrada decaimento exponencial.

II.11 Simulação para $\alpha = 1, 0^\circ$

Na Figura II.71, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=1;

Czalfag=-0.03566;
Cmalfag=0.01355;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 1,0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.71. Programa para MATLAB.

Na Figura II.72, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.02046  0.08282
x2      1 -0.000483

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1581
-----
s^2 - 0.01998 s - 0.08283

```

Figura II.72. Dados de simulação.

Na Figura II.73, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura II.73. Dados de simulação.

Na Figura II.74, encontra-se o gráfico da saída para entrada degrau.

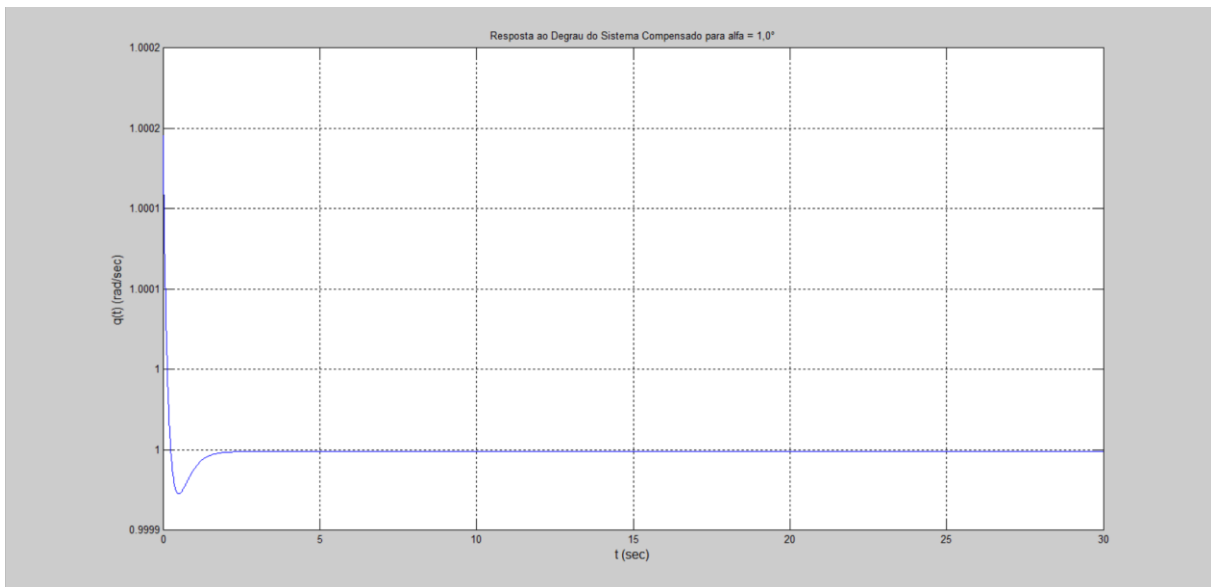


Figura II.74. Gráfico da saída para entrada degrau unitário.

Na Figura II.75, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 1,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.75. Programa para MATLAB.

Na Figura II.76, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
     4

z_max =
    1.0002

z_min =
    6.1409e-010

z_inf =
    6.1409e-010

```

Figura II.76. Dados de simulação.

Na Figura II.77, encontra-se o gráfico da saída para entrada decaimento exponencial.

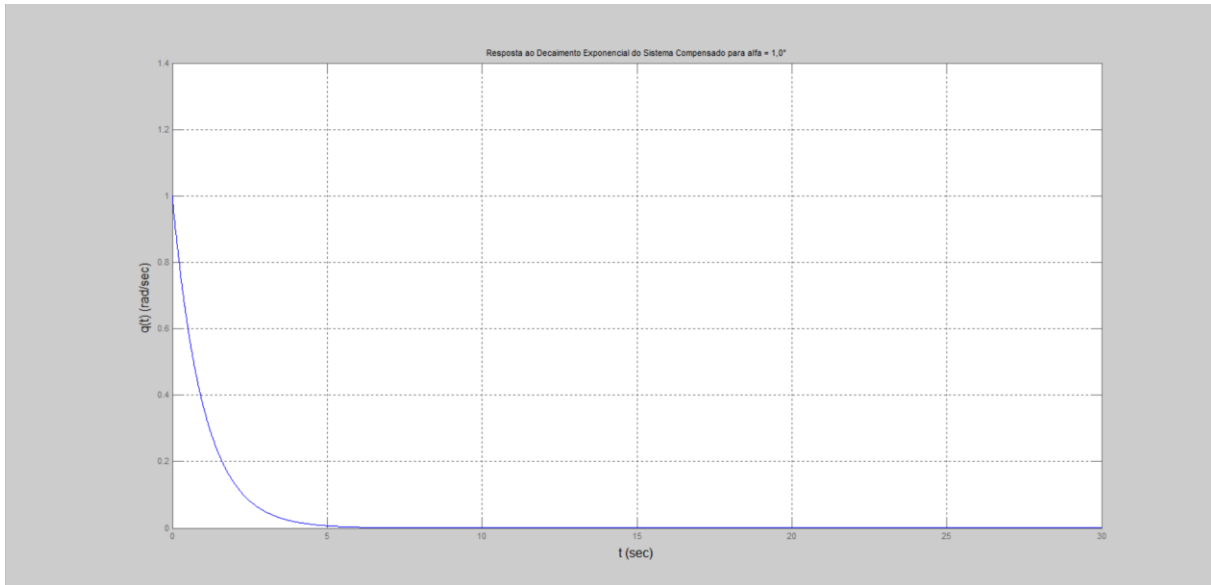


Figura II.77. Gráfico da saída para decaimento exponencial.

II.12 Simulação para $\alpha = 1,5^\circ$

Na Figura II.78, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=1.5;

Czalfag=-0.03643;
Cmalfag=0.01324;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 1,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.78. Programa para MATLAB.

Na Figura II.79, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01691  0.07358
x2      1 -0.0004934

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1569
-----
s^2 - 0.01642 s - 0.07359

```

Figura II.79. Dados de simulação.

Na Figura II.80, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =
    20

a =
    4

y_max =
    1.0002

y_min =
    1.0000

y_inf =
    1.0000

SVP =
    0.0196
```

Figura II.80. Dados de simulação.

Na Figura II.81, encontra-se o gráfico da saída para entrada degrau.

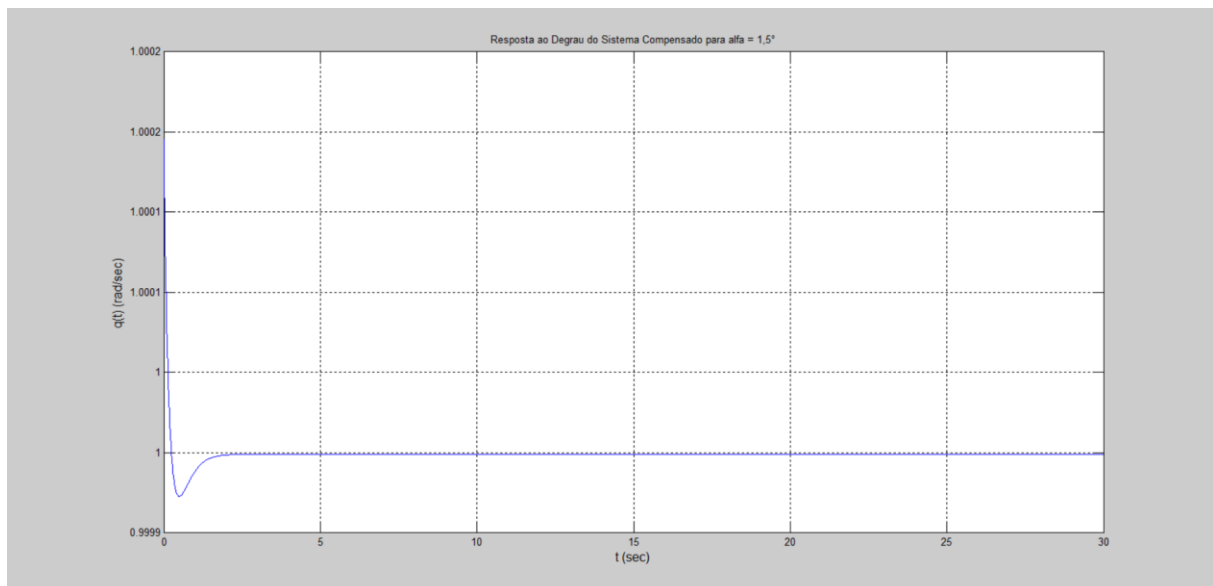


Figura II.81. Gráfico da saída para a entrada degrau unitário.

Na Figura II.82, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 1,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.82. Programa para MATLAB.

Na Figura II.83, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
     4

z_max =
    1.0002

z_min =
    5.4159e-010

z_inf =
    5.4159e-010

```

Figura II.83. Dados de simulação.

Na Figura II.84, encontra-se o gráfico da saída para entrada decaimento exponencial.

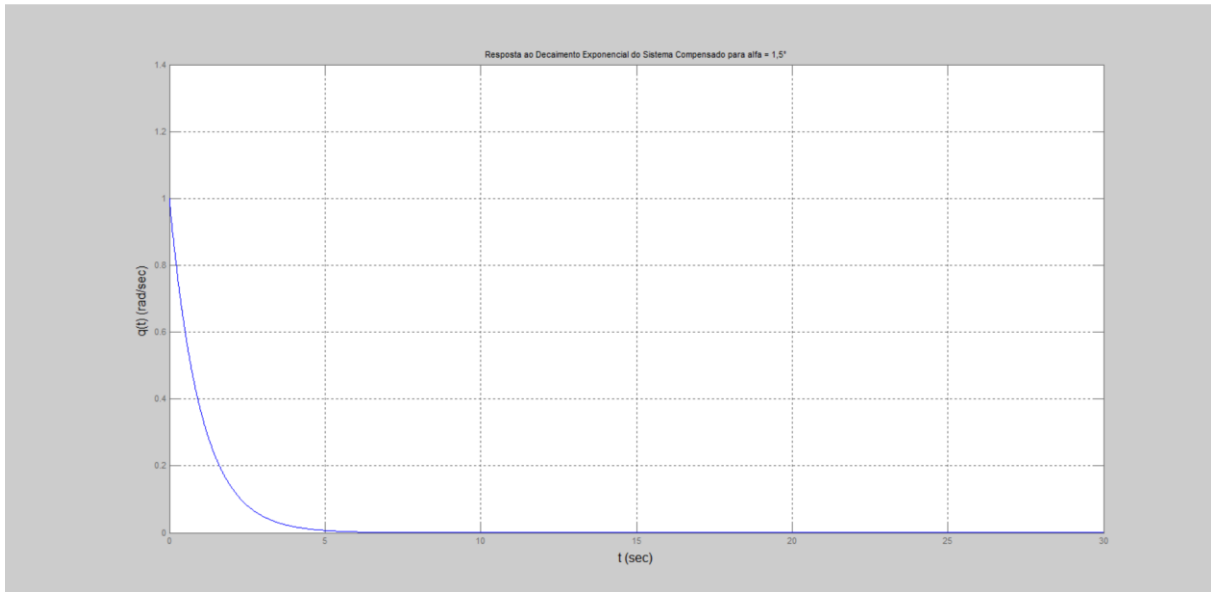


Figura II.84. Gráfico da saída para entrada decaimento exponencial.

II.13 Simulação para $\alpha = 2,0^\circ$

Na Figura II.85, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=2;

Czalfag=-0.03719;
Cmalfag=0.01293;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```



```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 2,0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.85. Programa para MATLAB.

Na Figura II.86, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01423  0.06591
x2      1 -0.0005037

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1564
  -----
  s^2 - 0.01373 s - 0.06592

```

Figura II.86. Dados de simulação.

Na Figura II.87, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP = |  
    0.0196
```

Figura II.87. Dados de simulação.

Na Figura II.88, encontra-se o gráfico da saída para entrada degrau.

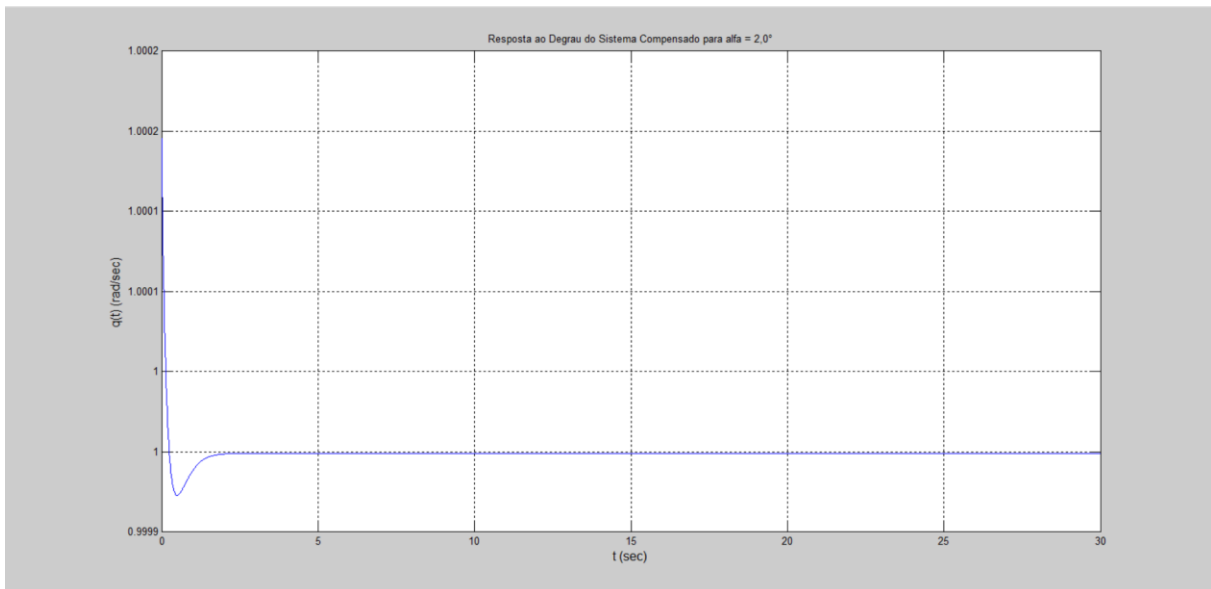


Figura II.88. Gráfico da saída para entrada degrau unitário.

Na Figura II.89, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 2,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.89. Programa para MATLAB.

Na Figura II.90, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    4.8345e-010

z_inf =
    4.8345e-010

```

Figura II.90. Dados de simulação.

Na Figura II.91, encontra-se o gráfico da saída para entrada decaimento exponencial.

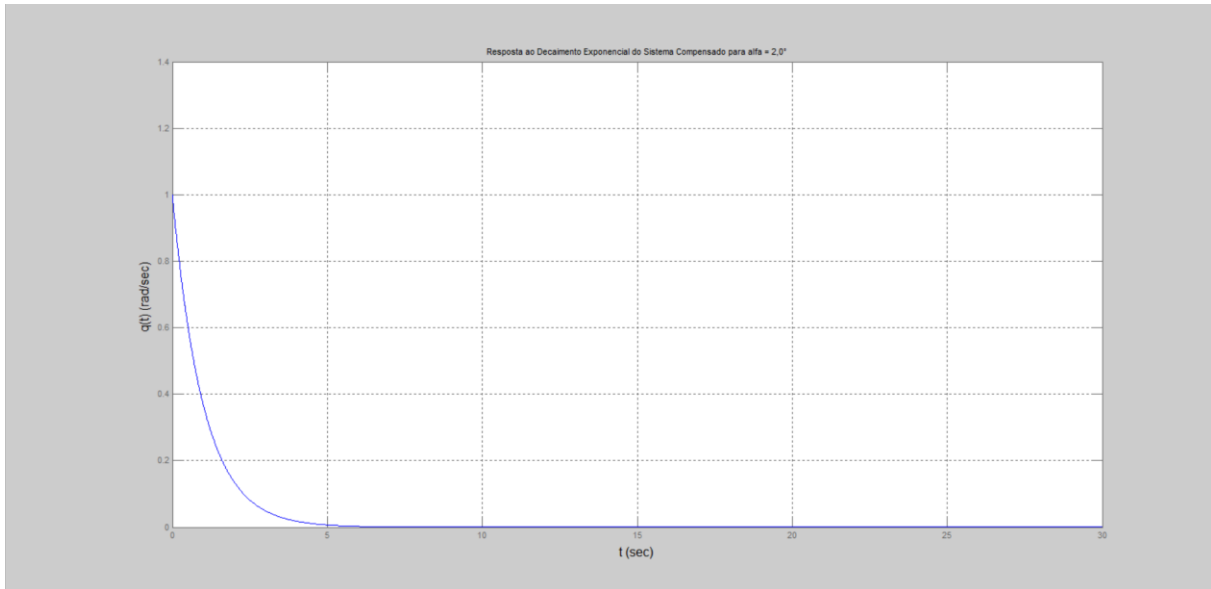


Figura II.91. Gráfico da saída para a entrada decaimento exponencial.

II.14 Simulação para $\alpha = 2,5^\circ$

Na Figura II.92, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=2.5;

Czalfag=-0.03796;
Cmalfag=0.01262;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=l;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 2, S^p')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.92. Programa para MATLAB.

Na Figura II.93, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.01215  0.05944
x2      1 -0.0005141

```

```

b =
      u1
x1 -255.9
x2 -0.4168

```

```

c =
      x1  x2
y1  1   0

```

```

d =
      u1
y1  0

```

Continuous-time model.

```
ans =
```

```
G(s)=
```

```

Transfer function:
      -255.9 s - 0.1563
-----
s^2 - 0.01163 s - 0.05945

```

Figura II.93. Dados de simulação.

Na Figura II.94, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.


```
K =  
    20  
  
a =  
    4  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
|  
SVP =  
    0.0196
```

Figura II.94. Dados de simulação.

Na Figura II.95, encontra-se o gráfico da saída para entrada degrau.

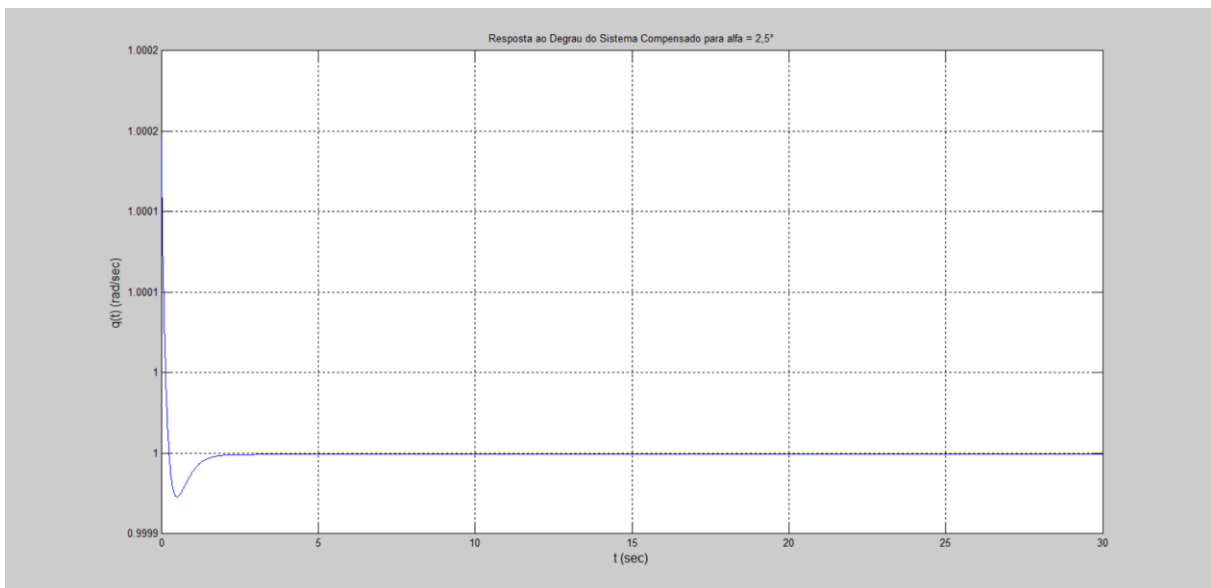


Figura II.95. Gráfico da saída para entrada degrau unitário.

Na Figura II.96, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 2,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.96. Programa para MATLAB.

Na Figura II.97, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    4.3593e-010

z_inf =
    4.3593e-010

```

Figura II.97. Dados de simulação.

Na Figura II.98, encontra-se o gráfico da saída para entrada decaimento exponencial.

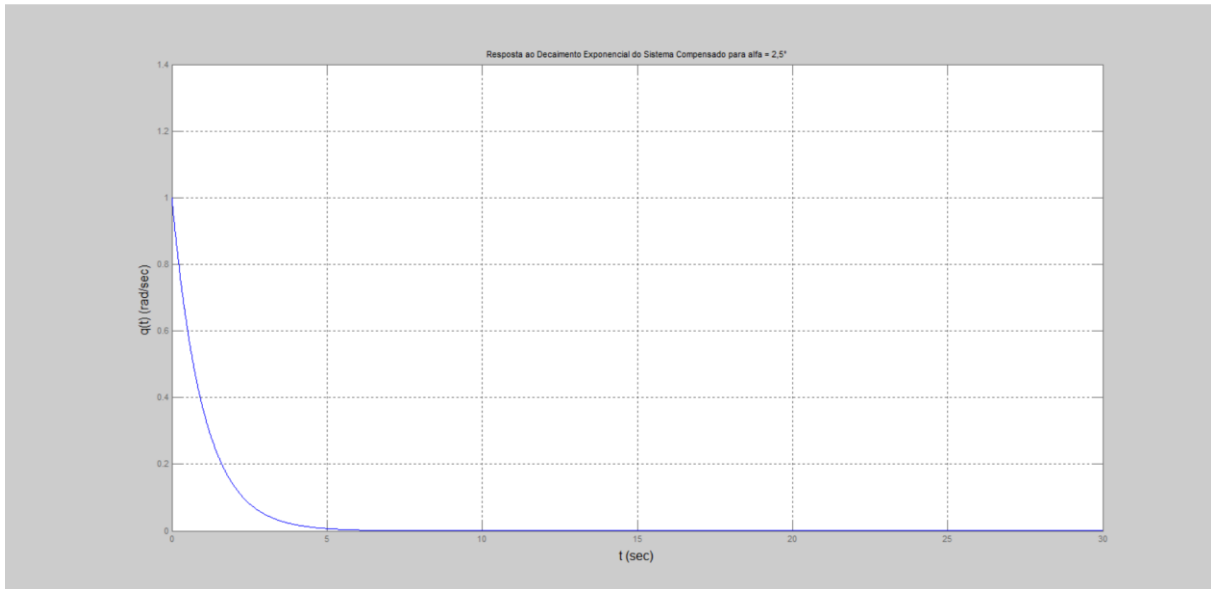


Figura II.98. Gráfico da saída para entrada decaimento exponencial.

II.15 Simulação para $\alpha = 3,0^\circ$

Na Figura II.99, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=2.5;

Czalfag=-0.03796;
Cmalfag=0.01262;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=l;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 2, S^p')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.99. Programa para MATLAB.

Na Figura II.100, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1    0.01215  0.05944
x2      1 -0.0005141

```

```

b =
      u1
x1 -255.9
x2 -0.4168

```

```

c =
      x1  x2
y1    1   0

```

```

d =
      u1
y1    0

```

Continuous-time model.

```
ans =
```

```
G(s)=
```

```

Transfer function:
      -255.9 s - 0.1563
-----
s^2 - 0.01163 s - 0.05945

```

Figura II.100. Dados de simulação.

Na Figura II.101, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, $SV\%P$, calculados pelo programa.

```
K =
    20

a =
    4

y_max =
    1.0002

y_min =
    1.0000

y_inf =
    1.0000

|
SVP =
    0.0196
```

Figura II.101. Dados de simulação.

Na Figura II.102, encontra-se o gráfico da saída para entrada degrau.

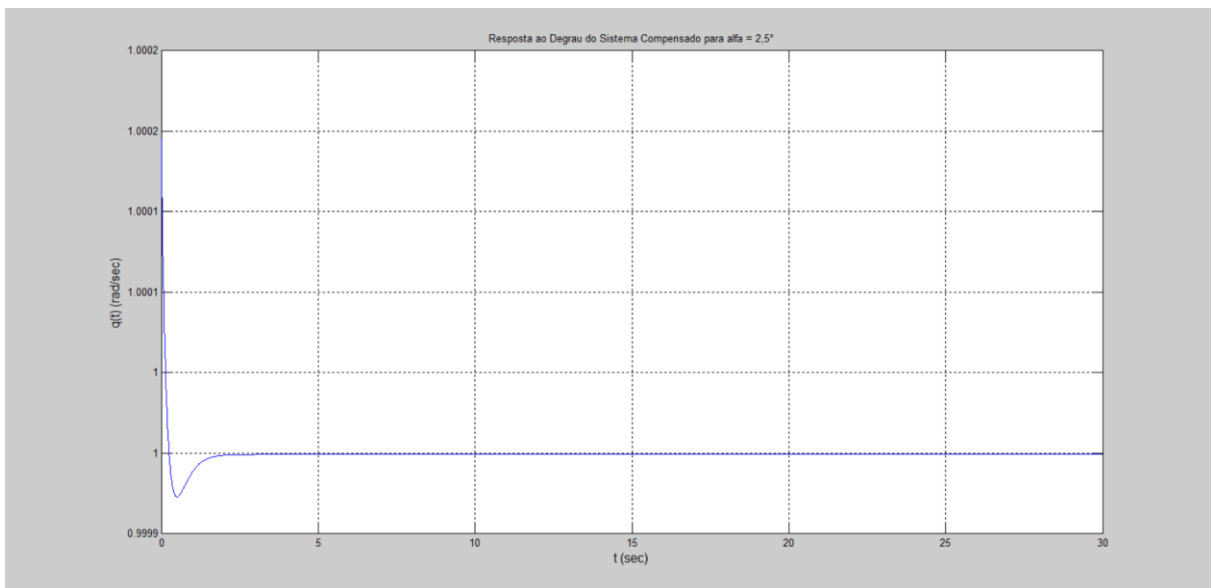


Figura II.102. Gráfico da saída para entrada degrau unitário.

Na Figura II.103, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 2,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.103. Programa para MATLAB.

Na Figura II.104, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    4.3593e-010

z_inf =
    4.3593e-010

```

Figura II.104. Dados de simulação.

Na Figura II.105, encontra-se o gráfico da saída para entrada decaimento exponencial.

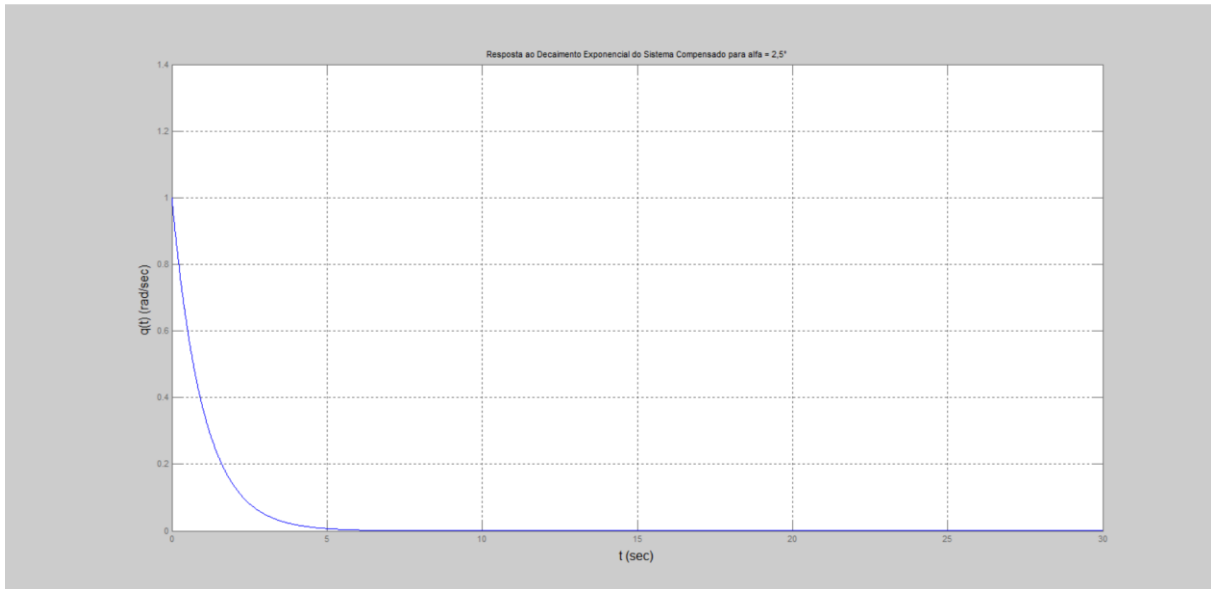


Figura II.105. Gráfico da saída para entrada decaimento exponencial.

II.16 Simulação para $\alpha = 3,5^\circ$

Na Figura II.106, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=3.5;

Czalfag=-0.03949;
Cmalfag=0.012;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

!G(s)=
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 3,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.106. Programa para MATLAB.

Na Figura II.107, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.009157  0.04907
x2      1 -0.0005348

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1573
-----
s^2 - 0.008622 s - 0.04908

```

Figura II.107. Dados de simulação.

Na Figura II.108, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

K =
20
a =
4
y_max =
1.0002
y_min =
1.0000
y_inf =
1.0000
SVP =
0.0196

Figura II.108. Dados de simulação.

Na Figura II.109, encontra-se o gráfico da saída para entrada degrau.

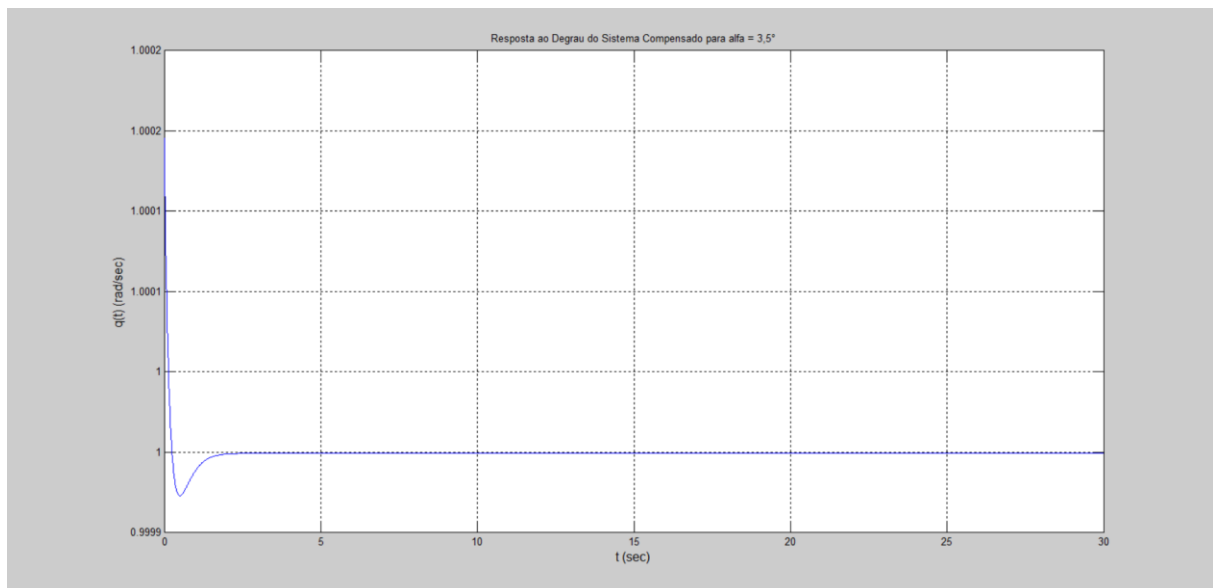


Figura II.109. Gráfico da saída para entrada degrau.

Na Figura II.110, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```
K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 3,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)
```

Figura II.110. Programa para MATLAB.

Na Figura II.111, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```
K =
    20

a =
     4

z_max =
    1.0002

z_min =
    3.6210e-010

z_inf =
    3.6210e-010
```

Figura II.111. Dados de simulação.

Na Figura II.112, encontra-se o gráfico da saída para entrada decaimento exponencial.

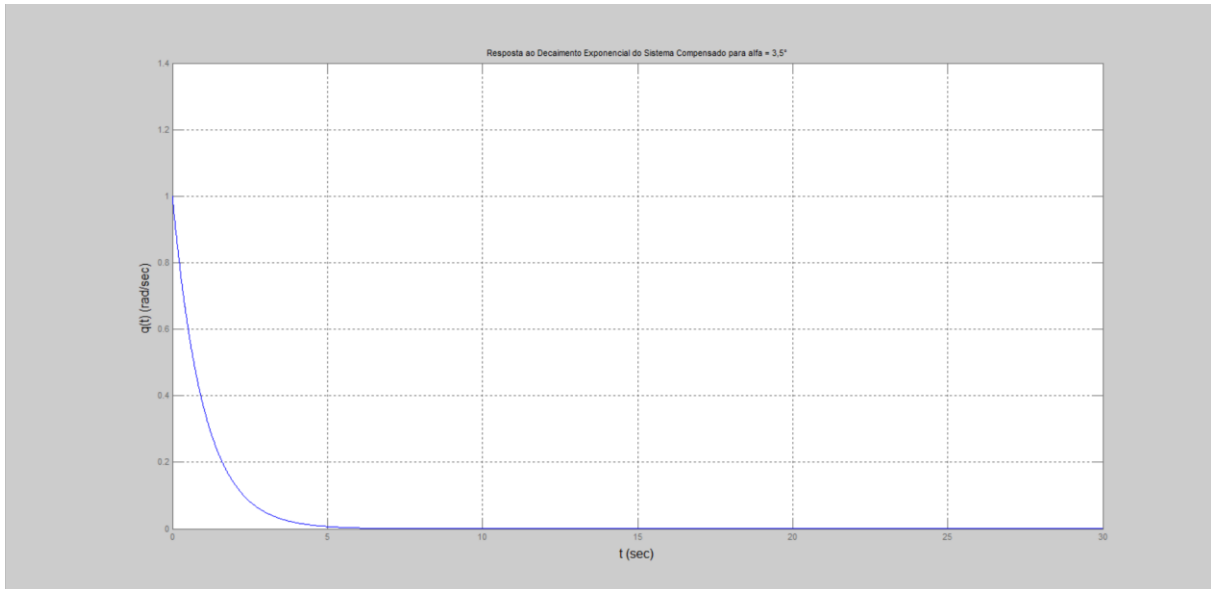


Figura II.112. Gráfico da saída para a entrada decaimento exponencial.

II.17 Simulação para $\alpha = 4,0^\circ$

Na Figura II.113, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=4;

Czalfag=-0.04025;
Cmalfag=0.01169;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=l;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=l;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```



```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 4,0°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.113. Programa para MATLAB.

Na Figura II.114, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.008055  0.04484
x2      1 -0.0005451

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1  x2
y1  1    0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
  -255.9 s - 0.1582
-----
s^2 - 0.00751 s - 0.04484

```

Figura II.114. Dados de simulação.

Na Figura II.115, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, SVP , calculados pelo programa.

K =
20
a =
4
y_max =
1.0002
y_min =
1.0000
y_inf =
1.0000
SVP =
0.0196

Figura II.115. Dados de simulação.

Na Figura II.116, encontra-se o gráfico da saída para entrada degrau.

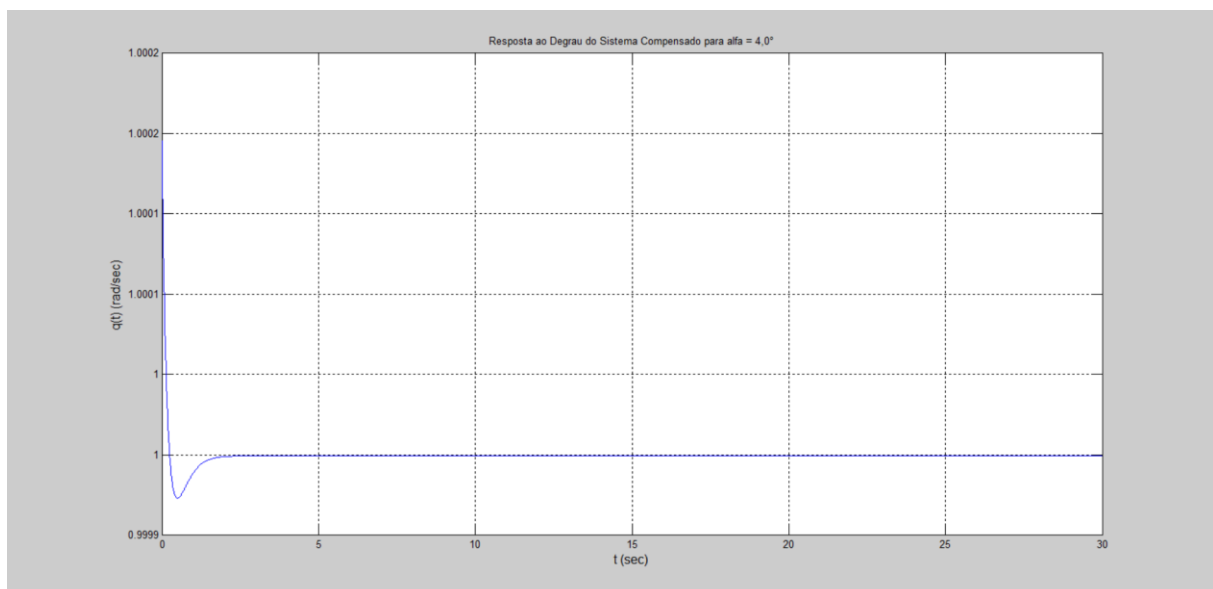


Figura II.116. Gráfico da saída para entrada degrau unitário.

Na Figura II.117, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 4,0°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.117. Programa para MATLAB.

Na Figura II.118, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    3.3265e-010

z_inf =
    3.3265e-010

```

Figura II.118. Dados de simulação.

Na Figura II.119, encontra-se o gráfico da saída para entrada decaimento exponencial.

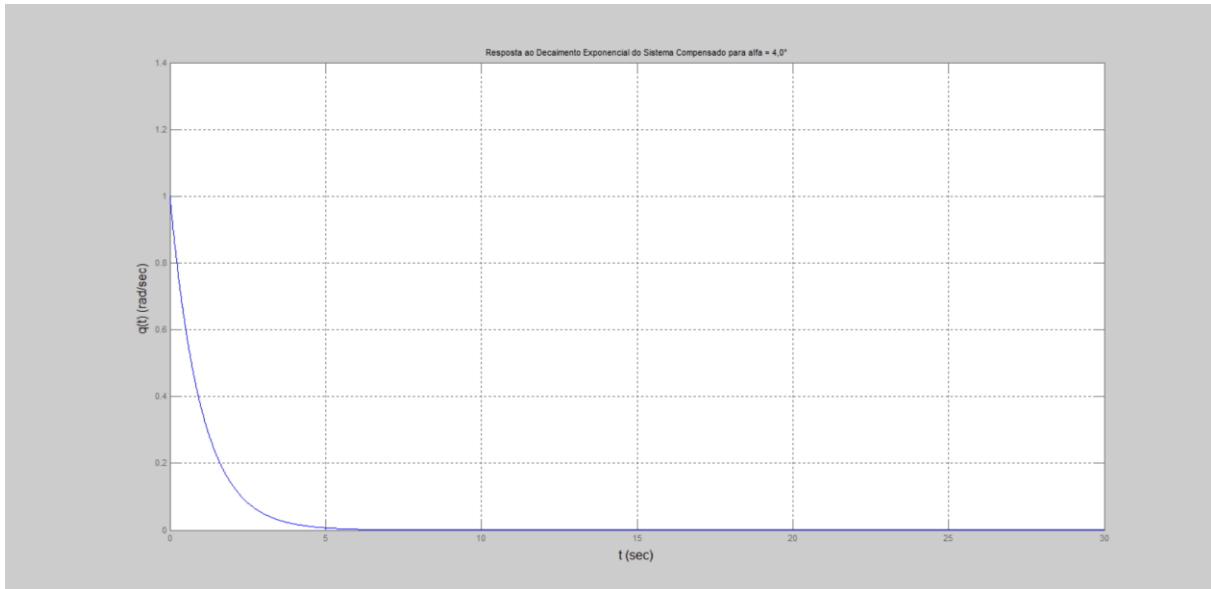


Figura II.119. Gráfico da saída para entrada decaimento exponencial.

II.18 Simulação para $\alpha = 4,5^\circ$

Na Figura II.120, encontra-se o programa usado para se calcular os parâmetros do controlador.

```

alfag=4.5;

Czalfag=-0.04101;
Cmalfag=0.01139;

Cmqg=0.01292;

Czalfa=Czalfag/(pi/180);
Cmalfa=Cmalfag/(pi/180);

Cmq=Cmqg/(pi/180);

ma=6.71602;
v=142.91;
zeta=0.70489;
Jyy=1.10185;
T=400;
roh=1.0483;
mi0=v;
S=0.0030288;
xcg=0.85511;

alfa=alfag*pi/180;

aa=abs(alfa);

xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

l=xcg-xcp;

Zalfa=0.5*roh*mi0*S*Czalfa;
Malfa=0.5*roh*mi0*l*S*Cmalfa;
Mq=0.25*roh*mi0*l^2*S*Cmq;

a1=Mq/Jyy;
a2=Malfa/Jyy;
a3=1;
a4=Zalfa/(ma*v);
b1=-T*zeta/Jyy;
b2=-T/(ma*v);
c1=1;
c2=0;
d1=0;

A=[a1 a2; a3 a4];
B=[b1; b2];
C=[c1 c2];
D=d1;

sys=ss(A,B,C,D)

'G(s)='
G=tf(sys)

```

```

t=0:0.01:30;
for K=20:-0.2:2;
    for a=4:-0.2:0.5;
        num1=K*[1 2*a a^2];
        den1=[0 1 0];
        Gc=tf(num1,den1);
        GcG=Gc*G;
        T=feedback(GcG,1);
        y=step(T,t);
        m=min(y);
        for i=3000:-1:1;
            y1=y(i);
            if y1==1 & i<100
                break;
            end
        end
        if m>0.80 & y1<1.02 & y1>0.98 & i<100
            break;
        end
    end
    if m>0.80 & y1<1.02 & y1>0.98 & i<100
        break;
    end
end

plot(t,y)
xlabel('t (sec)', 'FontSize',12)
ylabel('q(t) (rad/sec)', 'FontSize',12)
title('Resposta ao Degrau do Sistema Compensado para alfa = 4,5°')
grid
K
a
y_max=max(y)
y_min=m
y_inf=y(3001)
s=stepinfo(y,t);
SVP=s.Overshoot

```

Figura II.120. Programa para MATLAB.

Na Figura II.121, encontram-se, calculadas pelo programa, as matrizes, A, B, C e D , da representação em espaço de estados e a função de transferência da planta.

```

a =
      x1      x2
x1  0.007139  0.04113
x2      1 -0.0005554

b =
      u1
x1 -255.9
x2 -0.4168

c =
      x1 x2
y1  1  0

d =
      u1
y1  0

Continuous-time model.

ans =

G(s)=

Transfer function:
      -255.9 s - 0.1593
-----
s^2 - 0.006583 s - 0.04113

```

Figura II.121. Dados de simulação.

Na Figura II.122, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada degrau, y_{max} , o mínimo valor da saída para a entrada degrau, y_{min} , o último valor de vetor de saída do MATLAB para a entrada degrau, y_{inf} , e o sobrevalor percentual para a entrada degrau, $SV\%P$, calculados pelo programa.


```
K =  
20  
  
a =  
4  
  
y_max =  
1.0002  
  
y_min =  
1.0000  
  
y_inf =  
1.0000  
  
SVP =  
0.0196
```

Figura II.122. Dados de simulação.

Na Figura II.123, encontra-se o gráfico da saída para entrada degrau.

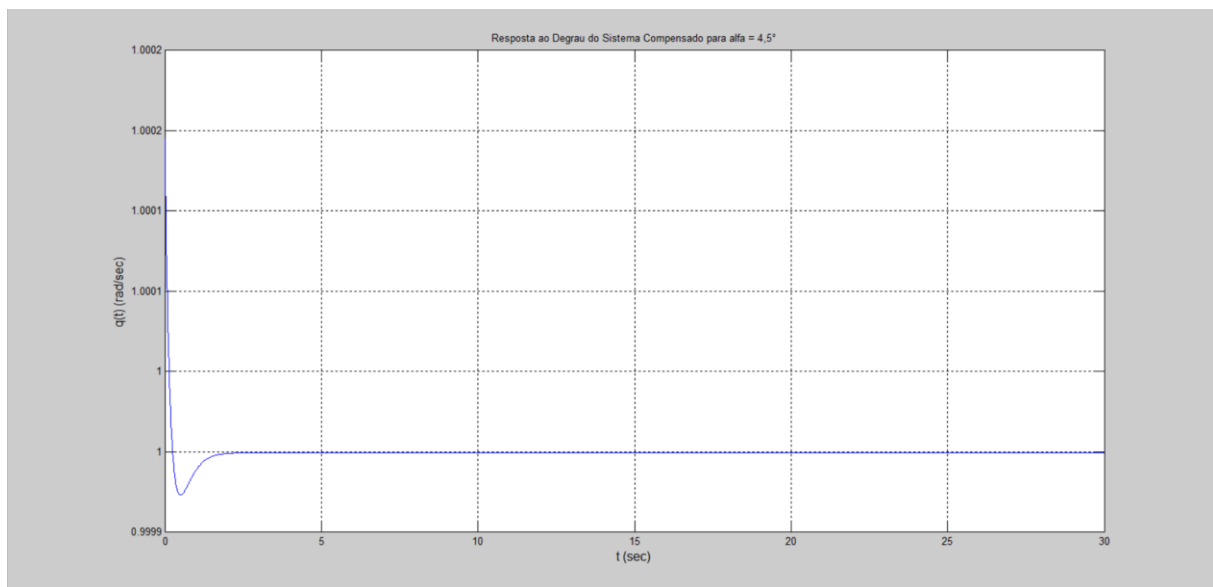


Figura II.123. Gráfico da saída para entrada degrau unitário.

Na Figura II.124, encontra-se o programa usado para se testar a entrada decaimento exponencial. As variáveis G e t são resgatadas pelo programa através do *workspace* (local onde estão armazenadas as variáveis calculadas pelo programa anterior).

```

K=20;
a=4;
num1=K*[1 2*a a^2];
den1=[0 1 0];
Gc=tf(num1,den1);
GcG=Gc*G;
T=feedback(GcG,1);
u=exp(-t);
z=lsim(T,u,t);
lsim(T,u,t)
xlabel('t','FontSize',12)
ylabel('q(t) (rad/sec)','FontSize',12)
title('Resposta ao Decaimento Exponencial do Sistema Compensado para alfa = 4,5°')
grid
K
a
z_max=max(z)
z_min=min(z)
z_inf=z(3001)

```

Figura II.124. Programa para MATLAB.

Na Figura II.125, encontram-se os parâmetros do controlador, K e a , o máximo valor da saída para a entrada decaimento exponencial, z_{max} , o mínimo valor da saída para a entrada decaimento exponencial, z_{min} , o último valor de vetor de saída do MATLAB para a entrada decaimento exponencial, z_{inf} .

```

K =
    20

a =
    4

z_max =
    1.0002

z_min =
    3.0719e-010

z_inf =
    3.0719e-010

```

Figura II.125. Dados de simulação.

Na Figura II.126, encontra-se o gráfico da saída para entrada decaimento exponencial.

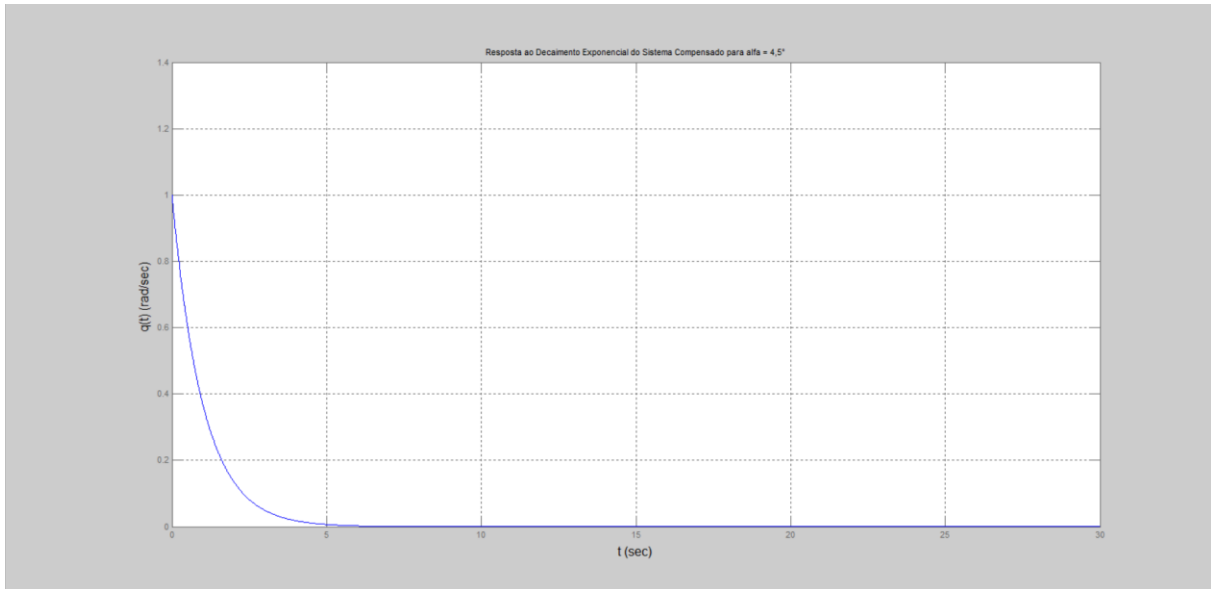


Figura II.126. Gráfico da saída para entrada decaimento unitário.

III Códigos usados e dados gerados na simulação de queima de combustível.

```

entrada=input('Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: ');
t=(0:0.01:30)';
for k=1:9
    alfag=3;
    Czalfag=-0.03872;
    Cmalvag=0.01231;
    Cmqg=0.01292;

    Czalfa=Czalfag/(pi/180);
    Cmalva=Cmalvag/(pi/180);

    Cmq=Cmqg/(pi/180);

    ma=k*0.1*6.71602;
    v=142.91;
    zeta=0.70489;
    Jyy=1.10185;
    Th=400;
    roh=1.0483;
    mi0=v;
    S=0.0030288;
    xcg=0.85511;

    alfa=alfag*pi/180;

    aa=abs(alfa);

    xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

    l=xcg-xcp;

    Zalva=0.5*roh*mi0*S*Czalfa;
    Malva=0.5*roh*mi0*l*S*Cmalva;
    Mq=0.25*roh*mi0*l^2*S*Cmq;

    al=Mq/Jyy;
    a2=Malva/Jyy;
    a3=l;
    a4=Zalva/(ma*v);
    b1=-Th*zeta/Jyy;
    b2=-Th/(ma*v);
    c1=l;
    c2=0;
    d1=0;

    A=[a1 a2; a3 a4];
    B=[b1; b2];
    C=[c1 c2];
    D=d1;

    sys=ss(A,B,C,D);

    G=tf(sys);
    K=20;
    a=4;
    num1=K*[1 2*a a^2];
    den1=[0 1 0];
    Gc=tf(num1,den1);
    GcG=Gc*G;
    T=feedback(GcG,1);
    y(:,k)=step(T,t);
    u=exp(-t);
    z(:,k)=lsim(T,u,t);
end

```

Figura III.1. Código para o cálculo das curvas no mesmo gráfico.

```

entrada=input('Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: ');
for k=1:9
    t=(0:0.01:30)';
    alfag=3;
    Czalfag=-0.03872;
    Cmalvag=0.01231;
    Cmvg=0.01292;

    Czalfa=Czalfag/(pi/180);
    Cmalva=Cmalvag/(pi/180);

    Cmvg=Cmvg/(pi/180);

    ma=k*0.1*6.71602;
    v=142.91;
    zeta=0.70489;
    Jyy=1.10185;
    Th=400;
    roh=1.0483;
    mi0=v;
    S=0.0030288;
    xcg=0.85511;

    alfa=alfag*pi/180;

    aa=abs(alfa);

    xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

    l=xcg-xcp;

    Zalva=0.5*roh*mi0*S*Czalva;
    Malva=0.5*roh*mi0*S*Cmalva;
    Mvg=0.25*roh*mi0^2*S*Cmvg;

    a1=Mvg/Jyy;
    a2=Malva/Jyy;
    a3=l;
    a4=Zalva/(ma*v);
    b1=-Th*zeta/Jyy;
    b2=-Th/(ma*v);
    c1=l;
    c2=0;
    d1=0;

    A=[a1 a2; a3 a4];
    B=[b1; b2];
    C=[c1 c2];
    D=d1;

    sys=ss(A,B,C,D);

    G=tf(sys);
    K=20;
    a=4;
    num1=K*[1 2*a a^2];
    den1=[0 1 0];
    Gc=tf(num1,den1);
    GcG=Gc*G;
    T=feedback(GcG,1);
    y=step(T,t);
    u=exp(-t);
    z=lsim(T,u,t);
    perc=k*10
    if entrada==1
        y_max=max(y)
        y_min=min(y)
        y_inf=y(3001)
        s=stepinfo(y,t);
        SVP=s.Overshoot
    elseif entrada==2
        z_inf=z(3001)
    end
end
end

```

Figura III.2. Código para o cálculo dos dados.

```
perc =  
    10  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196  
perc =  
    20  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

```
perc =  
    30  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

```
perc =  
    40  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

```
perc =  
    50  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196  
perc =  
    60  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```



```
perc =  
    70  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

```
perc =  
    80  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

```
perc =  
    90  
  
y_max =  
    1.0002  
  
y_min =  
    1.0000  
  
y_inf =  
    1.0000  
  
SVP =  
    0.0196
```

Figura III.3. Dados sobre a saída para a entrada degrau unitário. Valores máximo e mínimo, valor final e sobrevalor percentual para cada uma das 9 curvas.

```
perc =  
    10  
  
z_inf =  
    3.3834e-009  
  
perc =  
    20  
  
z_inf =  
    1.8462e-009  
  
perc =  
    30  
  
z_inf =  
    1.2672e-009  
  
perc =  
    40  
  
z_inf =  
    9.6437e-010  
  
perc =  
    50  
  
z_inf =  
    7.7828e-010  
  
perc =  
    60  
  
z_inf =  
    6.5237e-010
```

```
perc =  
    70  
  
z_inf =  
    5.6152e-010  
  
perc =  
    80  
  
z_inf =  
    4.9288e-010  
  
perc =  
    90  
  
z_inf =  
    4.3919e-010
```

Figura III.4. Dados sobre a curva para a entrada decaimento exponencial. Valor final para cada uma das 9 curvas.

IV Códigos usados para a implementação do controlador de ganhos programados e gráficos de teste

```

alfag=input('Digite o valor do ângulo de ataque em graus: ');
entrada=input('Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: ');
if alfag== -4.5
    Czalfag=-0.04101;
    Cmalvag=0.01139;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -4
    Czalfag=-0.04025;
    Cmalvag=0.01169;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -3.5
    Czalfag=-0.03949;
    Cmalvag=0.012;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -3
    Czalfag=-0.03872;
    Cmalvag=0.01231;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -2.5
    Czalfag=-0.03796;
    Cmalvag=0.01262;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -2
    Czalfag=-0.03719;
    Cmalvag=0.01293;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -1.5
    Czalfag=-0.03643;
    Cmalvag=0.01324;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== -0.8
    Czalfag=-0.03528;
    Cmalvag=0.01371;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== 0
    Czalfag=-0.03458;
    Cmalvag=0.01399;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== 1
    Czalfag=-0.03566;
    Cmalvag=0.01355;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== 1.5
    Czalfag=-0.03643;
    Cmalvag=0.01324;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);
elseif alfag== 2
    Czalfag=-0.03719;
    Cmalvag=0.01293;
    T=FunTran(alfag,Czalfag,Cmalvag);
    saida(T,entrada);

```

```

elseif alfag==2.5
    Czalfag=-0.03796;
    Cmalfag=0.01262;
    T=FunTran(alfag,Czalfag,Cmalfag);
    saida(T,entrada);
elseif alfag==3
    Czalfag=-0.03872;
    Cmalfag=0.01231;
    T=FunTran(alfag,Czalfag,Cmalfag);
    saida(T,entrada);
elseif alfag==3.5
    Czalfag=-0.03949;
    Cmalfag=0.012;
    T=FunTran(alfag,Czalfag,Cmalfag);
    saida(T,entrada);
elseif alfag==4
    Czalfag=-0.04025;
    Cmalfag=0.01169;
    T=FunTran(alfag,Czalfag,Cmalfag);
    saida(T,entrada);
elseif alfag==4.5
    Czalfag=-0.04101;
    Cmalfag=0.01139;
    T=FunTran(alfag,Czalfag,Cmalfag);
    saida(T,entrada);
end

```

Figura IV.1. Código principal para o controlador de ganhos programados por chaveamento.

```

function T=FunTran(alfag,Czalfag,Cmalfag)
    Cmqq=0.01292;

    Czalfa=Czalfag/(pi/180);
    Cmalfa=Cmalfag/(pi/180);

    Cmqq=Cmqq/(pi/180);

    ma=6.71602;
    v=142.91;
    zeta=0.70489;
    Jyy=1.10185;
    T=400;
    roh=1.0483;
    mi0=v;
    S=0.0030288;
    xcg=0.85511;

    alfa=alfag*pi/180;

    aa=abs(alfa);

    xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

    l=xcg-xcp;

    Zalfa=0.5*roh*mi0*S*Czalfa;
    Malfa=0.5*roh*mi0*l*S*Cmalfa;
    Mq=0.25*roh*mi0*l^2*S*Cmqq;

```

Figura IV.2. Função *FunTran.m* para o controlador de ganhos programados por chaveamento.

```

function saida(T,entrada)
t=0:0.01:30;
if entrada==1
    y=step(T,t);
    plot(t,y)
    xlabel('t (sec)','FontSize',12)
    ylabel('q(t) (rad/sec)','FontSize',12)
    title('Resposta ao Degrau do Sistema Compensado')
    grid
    y_max=max(y)
    y_min=min(y)
    y_inf=y(3001)
    s=stepinfo(y,t);
    SVP=s.Overshoot
elseif entrada==2
    u=exp(-t);
    z=lsim(T,u,t);
    plot(t,z)
    xlabel('t','FontSize',12)
    ylabel('q(t) (rad/sec)','FontSize',12)
    title('Resposta ao Decaimento Exponencial do Sistema Compensado')
    grid
    z_max=max(z)
    z_min=min(z)
    z_inf=z(3001)
end

```

Figura IV.3. Função *saida.m* para o controlador de ganhos programados por chaveamento.

```

alfag=input('Digite o valor do ângulo de ataque em graus: ');
entrada=input('Digite 1(um) para entrada degrau ou digite 2(dois) para entrada decaimento exponencial: ');
Cza=Czalfa(alfag);
Cma=Cmalfa(alfag);
T=FunTranConti(alfag,Cza,Cma);
saida(T,entrada);

```

Figura IV.4. Código principal para o controlador de ganhos programados por interpolação.

```

function T=FunTranConti(alfag,Cza,Cma)
    Cmqq=0.01292;

    Cmqq=Cmqq/(pi/180);

    ma=0.8*6.71602;
    v=142.91;
    zeta=0.70489;
    Jyy=1.10185;
    T=400;
    roh=1.0483;
    mi0=v;
    S=0.0030288;
    xcg=0.85511;

    alfa=alfag*pi/180;

    aa=abs(alfa);

    xcp=15436.7*aa^5-6286.48*aa^4+1105.03*aa^3-116.266*aa^2+9.1158*aa+0.208;

    l=xcg-xcp;

    Zalfa=0.5*roh*mi0*S*Cza;
    Malfa=0.5*roh*mi0*1*S*Cma;
    Mq=0.25*roh*mi0*1^2*S*Cmq;

    a1=Mq/Jyy;
    a2=Malfa/Jyy;
    a3=1;
    a4=Zalfa/(ma*v);
    b1=-T*zeta/Jyy;
    b2=-T/(ma*v);
    c1=1;
    c2=0;
    d1=0;

    A=[a1 a2; a3 a4];
    B=[b1; b2];
    C=[c1 c2];
    D=d1;

    sys=ss(A,B,C,D)

    G=tf(sys)
    K=20
    a=4
    num1=K*[1 2*a a^2];
    den1=[0 1 0];
    Gc=tf(num1,den1);
    GcG=Gc*G;
    T=feedback(GcG,1);

```

Figura IV.5. Função *FunTranConti.m* para o controlador de ganhos programados por interpolação.


```

function Cza=Czalfa(alfag)
alfa=alfag*pi/180;
alfagg=[-4.5 -4 -3.5 -3 -2.5 -2 -1.5 -0.8 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5];
Czalfagg=[-0.04101 -0.04025 -0.03949 -0.03872 -0.03796 -0.03719 -0.03643 ...
|-0.03528 -0.03458 -0.03489 -0.03566 -0.03643 -0.03719 -0.03796 -0.03872 -0.03949 -0.04025 -0.04101];
alfar=alfagg*pi/180;
Czalfar=Czalfagg/(pi/180);
x2=alfar;
y1=interp1(x2,Czalfar,alfar);
plot(x2,Czalfar,'o',alfar,y1)
xlabel('ângulo de ataque (rad)','FontSize',12)
ylabel('Cza (per rad)','FontSize',12)
title('Interpolação de Cza')
grid
Cza=interp1(x2,Czalfar,alfa)

```

Figura IV.6. Função *Cza.m* para o controlador de ganhos programados por interpolação.

```

function Cma=Cmalfa(alfag)
alfa=alfag*pi/180;
alfagg=[-4.5 -4 -3.5 -3 -2.5 -2 -1.5 -0.8 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5];
Cmalfagg=[0.01139 0.01169 0.012 0.01231 0.01262 0.01293 0.01324 0.01371 ...
|0.01399 0.01386 0.01355 0.01324 0.01293 0.01262 0.01231 0.012 0.01169 0.01139];
alfar=alfagg*pi/180;
Cmalfar=Cmalfagg/(pi/180);
x2=alfar;
y1=interp1(x2,Cmalfar,alfar);
plot(x2,Cmalfar,'o',alfar,y1)
xlabel('ângulo de ataque (rad)','FontSize',12)
ylabel('Cma (per rad)','FontSize',12)
title('Interpolação de Cma')
grid
Cma=interp1(x2,Cmalfar,alfa)

```

Figura IV.7. Função *Cma.m* para o controlador de ganhos programados por interpolação.

```

function saida(T,entrada)
t=0:0.01:30;
if entrada==1
    y=step(T,t);
    plot(t,y)
    xlabel('t (sec)','FontSize',12)
    ylabel('q(t) (rad/sec)','FontSize',12)
    title('Resposta ao Degrau do Sistema Compensado')
    grid
    y_max=max(y)
    y_min=min(y)
    y_inf=y(3001)
    s=stepinfo(y,t);
    SVP=s.Overshoot
elseif entrada==2
    u=exp(-t);
    z=lsim(T,u,t);
    plot(t,z)
    xlabel('t (sec)','FontSize',12)
    ylabel('q(t) (rad/sec)','FontSize',12)
    title('Resposta ao Decaimento Exponencial do Sistema Compensado')
    grid
    z_max=max(z)
    z_min=min(z)
    z_inf=z(3001)
end

```

Figura IV.8. Função *saida.m* para o controlador de ganhos programados por interpolação.

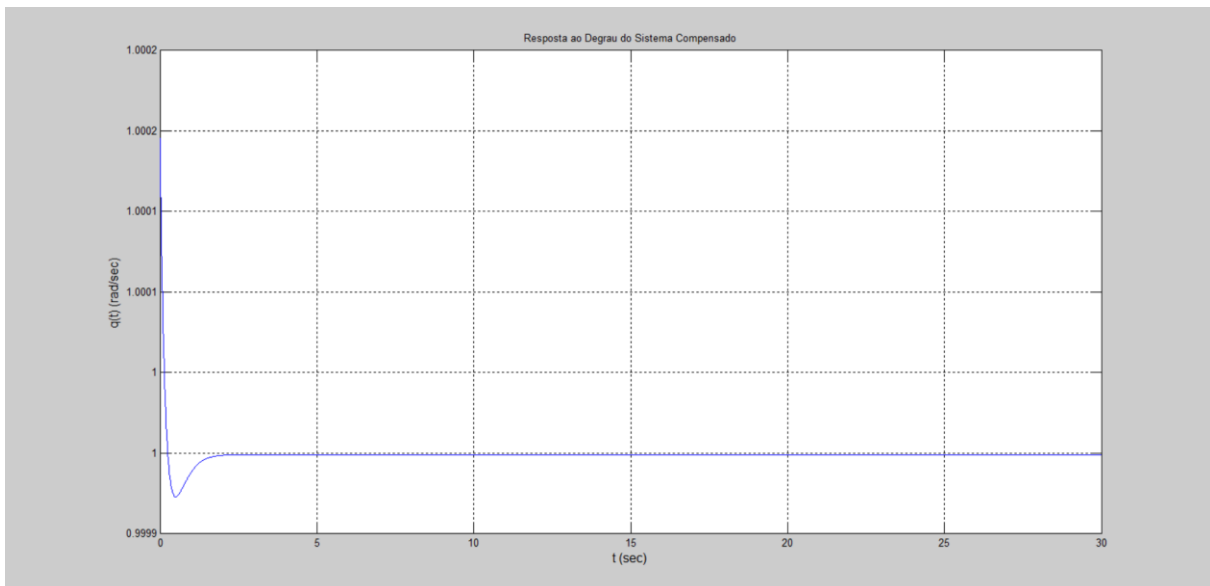


Figura IV.9. Gráfico da saída para $\alpha = 1.852^\circ$ e entrada degrau unitário.

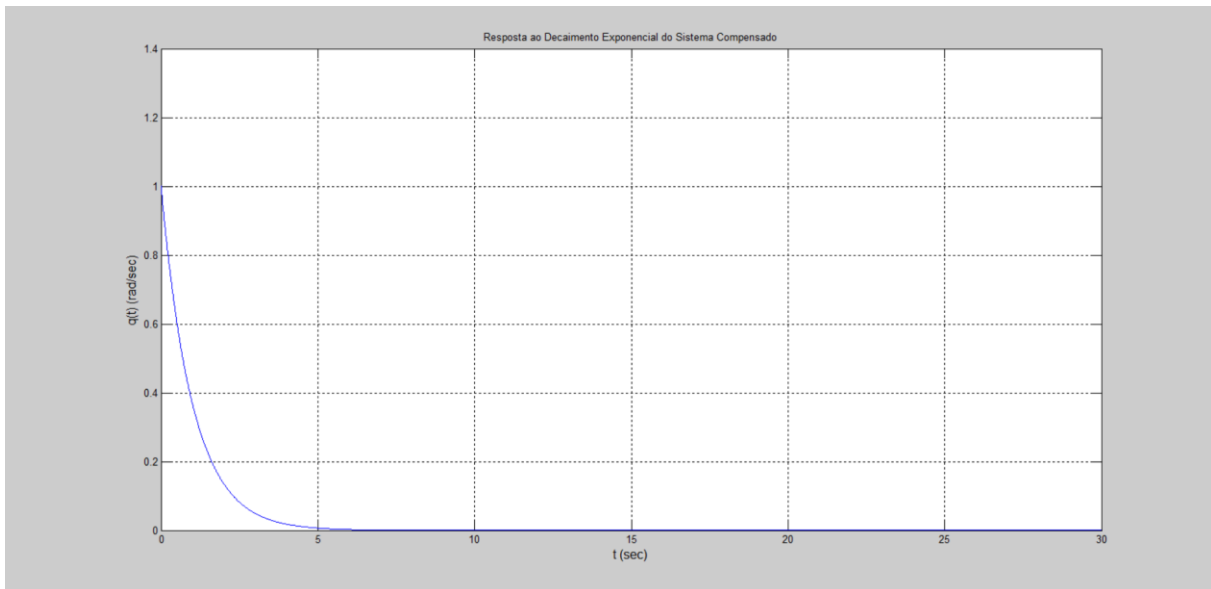


Figura IV.10. Gráfico da saída para $\alpha = 1,852^\circ$ e entrada decaimento exponencial.

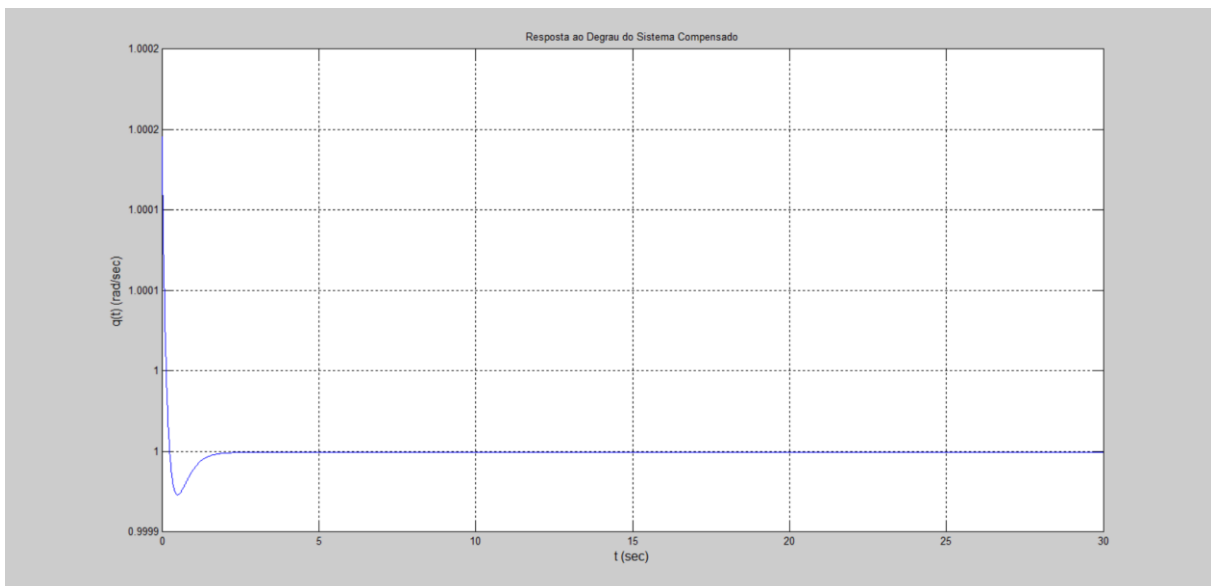


Figura IV.11. Gráfico da saída para $\alpha = -4,279^\circ$ e entrada degrau unitário.

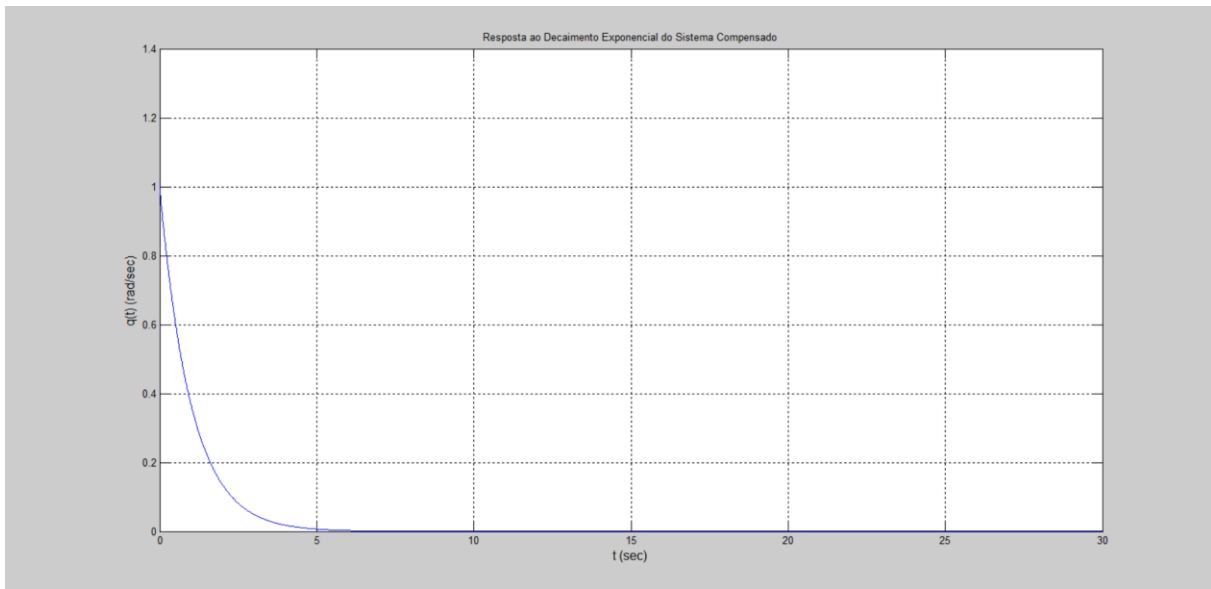


Figura IV.12. Gráfico da saída para $\alpha = -4,279^\circ$ e entrada decaimento exponencial.