

RICARDO BREGANON

**CONTROLE DE ARFAGEM E GUINADA DE UM SISTEMA
DE HÉLICES PARALELAS**

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Mecânica. Área de concentração: Aeronaves.

Orientador: Prof. Titular Eduardo Morgado Belo

São Carlos

2009

Aos meus pais, Bento e Cleonice por entenderem os momentos de ausência.

AGRADECIMENTOS

Ao Professor Titular Eduardo Morgado Belo pela orientação e por ter me oferecido a oportunidade de realizar este trabalho.

Ao amigo Márcio Aurélio Furtado Montezuma pela contribuição e discussão na elaboração deste trabalho.

Aos professores e funcionários da Escola de Engenharia de São Carlos.

Aos amigos de república, Ricardo A. Angélico, Ricardo Hasegawa, Gianluca, Luis Francisco Micheli, Paulo César Moselli pelo companheirismo e amizade.

Ao meu irmão Rogério Breganon pelo apoio.

Aos meus pais pelo carinho e incentivo.

À CAPES pela bolsa de estudos concedida durante a realização deste trabalho.

Resumo

BREGANON, Ricardo. *Controle de arfagem e guinada de um sistema de hélices paralelas*. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

A aplicação de técnicas de controle multivariáveis com o auxílio de ferramentas computacionais vem sendo bastante empregada em projetos de sistemas de controle complexos, que tem como base o conceito de variáveis de estado. Neste trabalho, visamos desenvolver um sistema de controle seguidor com realimentação de estados para um protótipo de um sistema de hélices paralelas onde a dinâmica é não linear e possui três graus de liberdade sendo dois graus de liberdade controlados. Serão utilizados dois modelos matemáticos: um modelo linear e um modelo não linear. A representação linear será utilizada para gerar as matrizes de ganho do sistema de controle. O modelo não linear é obtido através de prototipagem virtual no ambiente de desenvolvimento *ADAMS*[®], que utiliza técnicas de modelagem de sistemas multicorpos para a obtenção das equações dinâmicas do movimento. Em seguida o modelo não linear é exportado para o ambiente de simulações em *MATLAB*[®]. A simulação é utilizada para verificar se a técnica de controle linear utilizada tem a capacidade de controlar o sistema não linear e analisar o comportamento do conjunto controlador mais a planta não linear já que está sendo feita atribuição de auto-estrutura completa. Os resultados obtidos ilustram o desempenho e a eficiência da metodologia proposta.

Palavras chave: sistema de hélices paralelas, prototipagem virtual, controle multivariável, sistema seguidor.

Abstract

BREGANON, Ricardo *Pitch and Yaw Control of a parallel propeller system*. Thesis (Master) – School of Engineering of São Carlos, University of São Paulo, São Carlos, SP, Brazil, 2009.

The application of multivariable control techniques with the aid of computing tools has been largely used in projects of complex control systems that are based on the concept of state-variables. The objective of this paper is to show the development of a tracking-system with state-feedback for a prototype of a parallel propeller system presenting nonlinear dynamics with three degrees-of-freedom but having only two controlled. Two mathematical models are used: a linear model and a nonlinear model. The linear representation is used to design the gain matrices of the linear control system. The nonlinear model is obtained through the virtual prototyping environment system *ADAMS*[®], using modeling techniques of multi-body systems to obtain the dynamic motion equations. Then, the nonlinear model is exported to *SIMULINK*[®] (*MATLAB*[®]). The simulation is used to verify if the applied linear control techniques are able to control the nonlinear model and to analyze the system behavior (control system plus nonlinear plant). The design of the control system is done using the entire eigenstructure assignment technique. The results show the performance and the efficiency of the proposed methodology.

Keywords: parallel propeller system, virtual prototyping, multivariable control, tracking system.

LISTA DE FIGURAS

Figura 1 Aeroestabilizador	6
Figura 2 Planta experimental desenvolvida pela Quanser.....	7
Figura 3 Diagrama de blocos do sistema de controle proposto por Lopes (2006)	7
Figura 4 Diagrama de blocos apresentado por WITT (2007).....	7
Figura 5 Modelo Experimental.....	11
Figura 6 Curva experimental do empuxo do motor em relação à variação do PWM	13
Figura 7 Contra peso	14
Figura 8 Encoders utilizado para posicionamento do sistema de hélices paralelas.....	15
Figura 9 Placa de comunicação do encoder (RS-422).....	15
Figura 10 Modelo Virtual desenvolvido no ambiente multicorpos <i>ADAMS</i> [®]	25
Figura 11 Sistema de controle seguidor	29
Figura 12 Planta com um observador de estados	33
Figura 13 Planta do sistema representada na forma de espaço de estados	37
Figura 14 Planta Experimental	38
Figura 15 Diagrama de blocos <i>MATLAB</i> [®] / <i>SIMULINK</i> [®] de um sistema seguidor	39
Figura 16 Resposta a uma entrada degrau da planta linear gerada pelo <i>ADAMS</i> [®]	39
Figura 17 Velocidades angulares de arfagem, guinada e rolagem	40
Figura 18 Resposta a uma entrada degrau com ruído branco, de planta linear	40
Figura 19 Velocidades angulares de arfagem, guinada e rolagem com ruído branco	41
Figura 20 Resposta a uma entrada rampa da planta linear geradas pelo <i>ADAMS</i> [®]	41
Figura 21 Velocidades angulares de arfagem, guinada e rolagem	42
Figura 22 Resposta a uma entrada rampa com ruído	42
Figura 23 Velocidades angulares com ruído	43

Figura 24 Diagrama de blocos da planta não linear gerada pelo <i>ADAMS</i> [®]	43
Figura 25 Resposta a uma entrada degrau da planta não linear geradas pelo <i>ADAMS</i> [®]	44
Figura 26 Velocidades angulares de arfagem, guinada e rolagem	44
Figura 27 Resposta a uma entrada degrau com ruído da planta não linear	45
Figura 28 Velocidades angulares de arfagem, guinada e rolagem	45
Figura 29 Entrada Rampa aplicada da planta não linear geradas pelo <i>ADAMS</i> [®]	46
Figura 30 Velocidades angulares de arfagem, guinada e rolagem	46
Figura 31 Entrada rampa com ruído, aplicada da planta não linear geradas pelo <i>ADAMS</i> [®]	47
Figura 32 Velocidades angulares de arfagem, guinada e rolagem	47
Figura 33 Sistema de controle seguidor com observador de estados	48
Figura 34 Resposta a uma entrada degrau, de uma planta com observador de estados	49
Figura 35 Resposta a uma entrada degrau com ruído branco, aplicado na ação de controle ...	49
Figura 36 Resposta a uma entrada rampa, de uma planta com observador de estados	50
Figura 37 Resposta a uma entrada rampa, com ruído, de uma planta com observador	50
Figura 38 Módulo de Potência dos Motores.....	51
Figura 39 Esquema de aquisição de dados	52
Figura 40 Diagrama de Blocos do sistema de controle aplicado à planta real	53
Figura 41 Resposta do ângulo de arfagem referente à variação do contra peso traseiro.....	54
Figura 42 Resposta do ângulo de guinada referente à variação do contra peso traseiro.	55
Figura 43 Resposta a uma entrada degrau para o ângulo de arfagem.	56
Figura 44 Resposta a uma entrada degrau para o ângulo de guinada.....	57
Figura 45 Resposta do ângulo de arfagem referente à variação do contra peso lateral.....	57
Figura 46 Resposta do ângulo de guinada referente à variação do contra peso lateral.	58
Figura 47 Resposta do ângulo de arfagem referente aos diferentes ganhos dos motores	59
Figura 48 Resposta do ângulo de guinada referente aos diferentes ganhos dos motores.....	59

LISTA DE SÍMBOLOS

A	Matriz de estados.
$ADAMS^{\text{®}}$	Automatic Dynamics Analysis of Mechanical Systems
B	Matriz de entrada ou de controle.
C	Matriz de saída.
D	Matriz de transmissão direta
$DAE's$	Equações diferenciais e algébricas
DOF	Degrees of freedom
e	Erro do observador de estados
E	Matriz de saída relativa às saídas controláveis
F	Matriz de saída relativa às saídas não controláveis.
F_M	Força total dos motores
K_1, K_2	Matriz de ganhos do controle seguidor
L	Matriz de ganho do observador
MBS	Multy Body Systems
M_c	Matriz de controlabilidade
MSS	Mechanical System Simulation
m	Número de entradas
n	Número de estados
p	Número de saídas
PID	Proporcional-Integral-Derivativo
PWM	Pulse-Width Modulation
r	Comando de entrada
t	Tempo [s]

u	Vetor ação de controle
v	Conjunto de autovetores
v_1, v_2, \dots, v_{n+m}	Autovetores
w	Vetor das saídas controláveis
x	Vetor de estados do modelo linear do sistema de hélices paralelas
z	Vetor de estados do controle seguidor
$\lambda_1, \lambda_2, \dots, \lambda_{n+m}$	Autovalores [rad/s]
σ	Conjunto de autovalores

SUMÁRIO

INTRODUÇÃO.....	1
1.1 Motivação	2
1.2 Objetivos.....	2
1.3 Organização do Texto.....	3
REVISÃO DA LITERATURA	5
MODELAGEM DO SISTEMA DE HÉLICES PARALELAS	10
3.1 Introdução	10
3.2 Apresentação do Sistema	11
3.3 Modelagem Matemática.....	16
3.3.1 Técnicas MBS de Modelagem	17
3.3.2 <i>ADAMS</i> [®] - Ferramenta Computacional para MBS.	19
3.3.3 Sistema de Coordenadas	21
3.3.4 Tipos de Análises	22
3.4 Modelagem do Sistema de Hélices Paralelas.....	24
3.5 Análise do Modelo Linear na forma de Espaço de Estados.....	25
SISTEMA DE CONTROLE	27
4.1 Sistema de Controle Seguidor com Realimentação de Estados.....	27
4.2 TEORIA DAS VARIÁVEIS DE ESTADO	33
SIMULAÇÃO USANDO <i>SIMULINK</i> [®] / <i>ADAMS</i> [®]	36
5.1 <i>SIMULINK</i> [®]	36
5.2 Simulação e Controle usando <i>SIMULINK</i> [®] / <i>ADAMS</i> [®]	36
MATERIAIS E MÉTODOS.....	38
6.1 Materiais utilizados	38

6.2 Validação do Modelo e Resultados Experimentais	51
CONCLUSÕES E PERSPECTIVAS FUTURAS.....	60
BIBLIOGRAFIA.....	61
Apêndice I	63

Capítulo 1

INTRODUÇÃO

O desenvolvimento de protótipos de sistemas didáticos que possibilitem o estudo de técnicas de controle pode resultar em ótimos benefícios para a apresentação de temas presentes num curso introdutório de controle (CAMPO, 2007).

Em razão da necessidade de atender às crescentes e rigorosas exigências de desempenho dos sistemas de controle, ao aumento da complexidade dos sistemas e ao acesso fácil e em larga escala aos computadores, a teoria de controle moderno, que é uma nova abordagem para a análise e o projeto de sistemas de controle complexos, têm como base o conceito de estado.

Por definição, o estado de um sistema dinâmico é o menor conjunto de variáveis (variáveis de estado) tal que o conhecimento destas variáveis em $t = t_0$, juntamente com a entrada do sistema para $t \geq t_0$, determina completamente o comportamento do sistema para qualquer instante. Já as variáveis de estado (x) podem ser definidas como menor conjunto de variáveis que determina o estado do sistema dinâmico (ALBUQUERQUE, 2007).

O modelo matemático de um sistema dinâmico é definido como um conjunto de equações que representa com precisão ou, pelo menos, razoavelmente bem a dinâmica do sistema. Note que um modelo matemático não é único para determinado sistema. Um sistema é representado de muitas maneiras diferentes e, portanto, pode ter vários modelos matemáticos, dependendo da perspectiva a ser considerada (OGATA, 2003).

Em estudos de engenharia a palavra modelo possui dois significados principais: um está associado aos modelos físicos construídos em escala e o outro está associado aos modelos matemáticos utilizados na descrição das relações e leis físicas que regem o comportamento de um dado sistema. O modelo físico se apresenta como um arranjo similar à situação real,

construído de acordo com leis e escalas apropriadas, tal que o modelo funcione de forma similar ao objeto real em estudo. Por modelo matemático se entende o processo de descrever o sistema real em termos matemáticos, isto é, equações matemáticas que correspondam aos princípios e às leis físicas que regem o comportamento do sistema que se deseja analisar (SILVA, 1999).

Implementações computacionais no campo de simulação de sistemas multicorpos permitindo desenvolver, testar, alterar e otimizar o desempenho de sistemas mecânicos submetidos a grandes deslocamentos é conhecida como “prototipagem virtual”. Mas esta poderosa técnica não elimina a construção de protótipos reais. O uso desta tecnologia reduz o tempo e custos nos inúmeros testes realizados nos protótipos reais. Constrói-se o protótipo real com o intuito de validar os resultados da simulação (MONTEZUMA, 2003).

1.1 Motivação

A motivação deste trabalho é desenvolver um sistema para simular os movimentos de arfagem e guinada de uma aeronave de asas rotativas, e implementar um sistema de controle em espaço de estados para uma aplicação do setor aeronáutico. Também como motivação deste trabalho está à utilização deste sistema como demonstrador didático de técnicas de controle.

1.2 Objetivos

O propósito deste trabalho consiste em confeccionar um simulador controlador de arfagem e guinada de um sistema didático com três graus de liberdade de movimento,

composto por duas hélices em paralelo, para testes de diferentes técnicas de controle. Para melhor entendimento o objetivo geral pode ser subdividido em objetivos específicos:

1. Projetar e desenvolver um modelo físico (experimental) de um sistema de duas hélices com três graus de liberdade de movimento, porém somente dois graus de liberdade controlados;
2. Desenvolver um modelo virtual utilizando o software de Sistemas Multicorpos *ADAMS*[®];
3. Implementar um sistema de controle de arfagem e guinada, utilizando espaço de estados;
4. Analisar os resultados do controlador;
5. Analisar as principais vantagens e desvantagens da metodologia empregada para obtenção do modelo matemático e das respostas do sistema.

O projeto do controlador foi baseado em um sistema de controle seguidor, obtido através de um modelo não linear da planta, linearizado em torno de um ponto de operação. O controlador resultante foi utilizado em simulações no ambiente *Matlab/Simulink*, e também na implementação em tempo real utilizando a planta experimental.

Os resultados são empregados para ilustrar a capacidade do controlador em fazer com que a saída possa seguir uma entrada especificada.

1.3 Organização do Texto

Este texto está dividido em 7 capítulos que estão estruturados como segue.

O Capítulo 2 trata de uma breve revisão da literatura relacionada à de uma planta não linear com três graus de liberdade, e dos diversos sistemas de controle aplicados.

O Capítulo 3 apresenta alguns conceitos de técnicas de Sistemas Multicorpos, implementadas em *ADAMS*[®] e introduz alguns aspectos sobre modelagem matemática e características do modelo utilizado.

O Capítulo 4 relata uma técnica de controle com atribuição de autoestrutura completa através de um sistema seguidor com realimentação de estados.

O Capítulo 5 mostra como foi feita a implementação dinâmica do sistema de hélices paralelas utilizando-se o *SIMULINK*[®], e o diagrama de blocos aplicado no controle do sistema.

O Capítulo 6 trata dos materiais e da metodologia utilizados para a realização das simulações e da aplicação do controle em uma planta não linear real, com a finalidade de analisar o comportamento dinâmico do modelo mais o sistema de controle multivariável e verificar sua eficiência.

O Capítulo 7 apresenta as conclusões e as propostas para trabalhos futuros.

Capítulo 2

REVISÃO DA LITERATURA

O objetivo deste capítulo é apresentar uma revisão de literatura sobre os aspectos mais relevantes para o desenvolvimento deste trabalho.

A disseminação de diversas técnicas modernas de controle de sistemas foi viabilizada recentemente pelo uso de ferramentas computacionais de baixo custo. Além dos programas dedicados ao desenvolvimento dessas aplicações, como *LabVIEW* e o *MATLAB*, o uso de sistemas digitais com grande capacidade de processamento também possibilitou essas implementações.

No trabalho de Campo (2005), é apresentado um sistema idealizado, que trata de um denominado aeroestabilizador com dois graus de liberdade: elevação e translação, controlados por um conjunto de motores de corrente contínua acoplados a hélices, que deverão gerar as forças necessárias para a movimentação do sistema, mostrado na Figura 1. O foco principal consiste na apresentação de plantas experimentais desenvolvidas para o estudo de técnicas de controle por meio de ferramentas computacionais voltadas para análise e síntese de controladores. O aeroestabilizador foi modelado supondo que os dois graus de liberdade do sistema podem ser desacoplados, obtendo-se então duas funções de transferência, uma função de transferência do movimento de translação e outra função de transferência referente ao movimento de elevação. Os controladores projetados para os dois graus de liberdade do sistema foram do tipo PID (Proporcional-Integral-Derivativo), e o procedimento de ajuste foi baseado em um processo iterativo de ajuste e verificação de desempenho.

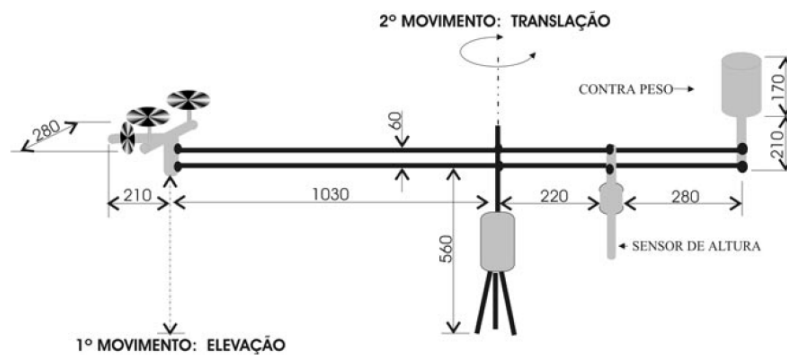


Figura 1 Aeroestabilizador

Lopes (2006) utiliza uma planta piloto na forma de um helicóptero com três graus de liberdade (Helicóptero 3DOF), construído pela Quanser Consulting®, representado na Figura 2, para implementar uma lei de controle preditivo no espaço de estados, com duas variáveis de entrada. A dinâmica do sistema é não-linear, de sexta ordem e com amortecimento muito pequeno. Linearizando-se o modelo físico nominal em torno de um ponto de equilíbrio, obteve-se um modelo de predição marginalmente estável. A implementação em tempo real envolve a solução de um problema de programação quadrática a cada período de amostragem de 50ms. Resultados experimentais são empregados para estudar o efeito de alterações nos horizontes de predição e controle, bem como para ilustrar a capacidade do controlador em fazer respeitar as restrições de saída. A Figura 3 mostra o diagrama de blocos do sistema de controle empregado.



Figura 2 Planta experimental desenvolvida pela Quanser

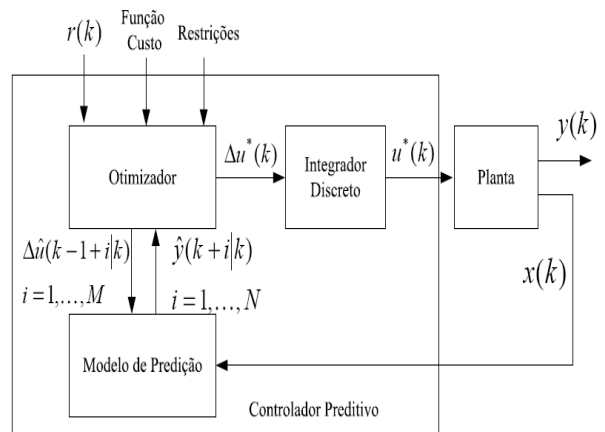


Figura 3 Diagrama de blocos do sistema de controle proposto por Lopes (2006)

No trabalho de Witt (2007) é apresentado um sistema de controle preditivo aproximado, representado na Figura4, aplicado ao modelo da Quanser, baseado em uma rede neural de um modelo de planta não linear, e a linearização em cada instante de tempo, ou seja, a cada momento da amostragem um modelo linear é extraído do modelo da rede neural, e usado para prever a saída futura, dentro de um espaço de tempo fixo em vez de um modelo linear estático, resultando assim em um modelo linear adaptável.

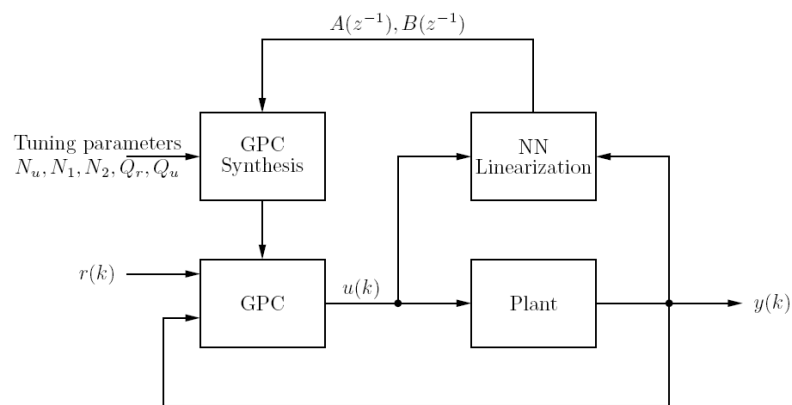


Figura 4 Diagrama de blocos apresentado por WITT (2007)

Yu e Zhong (2009) desenvolveram um controlador robusto de atitude com múltiplos pontos de operação, para uma planta nominal e uma planta com perturbação. O método de controle robusto com realimentação de saída, baseado em um sinal de compensação é proposto em um sistema linear e invariante no tempo.

O controlador robusto baseado em compensação é projetado em dois passos: primeiro, um controlador é desenvolvido para seguir a saída da planta nominal; então um compensador robusto é somado para compensar os efeitos da perturbação, a fim de alcançar as propriedades do controle robusto equivalente. O controlador foi implementado em uma planta experimental (Quanser), observado na Figura 2, mostrando que o controle robusto de atitude apresentou alta precisão em múltiplos pontos de operação.

Carvalho e Hemerly (2008) aplicaram um controle adaptativo em uma planta de dinâmica não linear usando redes neurais. Simulações realizadas por computadores comparam o desempenho desta metodologia com um controlador linear, quando perturbações paramétricas são introduzidas, mostrando como o controle consegue atuar bem nesta condição. Além disso, foram utilizados dois métodos de controle adaptativo não linear com redes neurais, um com realimentação de saída e outro com realimentação de estados, ilustrando os efeitos de conhecimento prévio sobre a dinâmica da planta, arquitetura da rede, e informações disponíveis pelos sensores. Porém eles observaram que a realimentação de estados apresenta um bom desempenho, mesmo quando os estados estimados foram usados no lugar das variáveis reais mensuradas.

Outro aspecto importante é a utilização do software *ADAMS*[®] para modelagem de sistemas dinâmicos, e o emprego de um sistema de controle seguidor com realimentação de estados.

No trabalho de Montezuma (2003), é utilizada essa metodologia de modelagem em *ADAMS*[®] e a implementação de um controle de posição e orientação aplicado a um sistema

seguidor com realimentação de estados, em uma plataforma de Stewart com seis graus de liberdade. A utilização desta metodologia visa obter uma economia de tempo e esforço na modelagem de sistemas mecânicos complexos, além de facilitar a análise de várias configurações de plataformas de Stewart.

Albuquerque (2007) obtém o modelo para simulação de uma mão robótica com 20 graus de liberdade auxiliando-se do programa MBS numérico *ADAMS*[®], que possui rotinas para a geração das equações e para a solução do sistema. Este programa possui aplicativos de pré e pós processamento. Estas ferramentas facilitaram a criação do modelo da mão robótica e a posterior simulação com análises e apresentação gráfica dos resultados. Um sistema de controle seguidor com realimentação de estados foi aplicado para controlar os movimentos da mão robótica, e testar a eficiência desta metodologia.

Capítulo 3

MODELAGEM DO SISTEMA DE HÉLICES PARALELAS

3.1 Introdução

O extraordinário aumento de desempenho dos computadores aliado à evolução da teoria de sistemas multicorpos na última década levou ao surgimento de softwares capazes de formular e resolver automaticamente as equações de sistemas mecânicos complexos.

Técnicas de modelamento MBS são utilizadas para analisar o comportamento de tais sistemas, uma tarefa que geralmente é apoiada por programas de computador capazes de modelar sistemas dinâmicos MBS e simular o seu comportamento.

Um MBS caracteriza-se por ser um sistema mecânico que pode ser modelado como um conjunto de corpos rígidos conectados por articulações, os quais podem apresentar um alto grau de movimentação relativa entre si. O sistema está sujeito à ação de forças, obedecendo a um conjunto de restrições, e tem seu comportamento governando por equações dinâmicas de movimento (COSTA, 1992).

Segundo Huston & Kamman (2000), as questões mais importantes ao se utilizar simulação numérica de sistemas multicorpos são: o quão bem o sistema multicorpos está representando o sistema real, quanto eficiente é a simulação e quanta acuidade tem a simulação.

Um corpo rígido é definido por um número de pontos que devem ser movimentados juntamente. Um ponto não pode se mover em relação ao outro, mas os pontos movem-se como um todo, no caso de pontos que definem polígonos e superfícies de forma livre. O movimento de um corpo rígido é especificado por seis graus de liberdade, translação e rotação em x , y e z .

Assim, este capítulo tem o objetivo de apresentar a modelagem do sistema de hélices paralelas utilizada neste trabalho para a determinação das características dinâmicas. Tal modelo foi construído utilizando o software de simulação de sistemas mecânicos *ADAMS*[®], que utiliza técnicas de modelagem de sistemas multicorpos para a obtenção de equações dinâmicas do movimento. Este software oferece uma interface gráfica para auxiliar a criação do modelo matemático e para a visualização dos resultados de simulação onde os parâmetros desejados são fornecidos de uma maneira simples e rápida.

3.2 Apresentação do Sistema

A estratégia de controle seguidor desenvolvida neste trabalho foi direcionada para trabalhar em uma planta (modelo experimental) com duas hélices paralelas, possuindo dois graus de liberdade controlados, sendo eles os movimentos de arfagem e guinada, e um terceiro grau de liberdade não controlado em rolagem, responsável em fazer com que o sistema possa guinar, pelo fato do mesmo não possuir um rotor de calda. (Figura 5).



Figura 5 Modelo Experimental

Este sistema, composto basicamente por duas hastes de tamanhos diferentes, em uma das extremidades possui dois motores de corrente contínua, cada um com uma hélice, que serão utilizadas para gerarem o empuxo, necessário para a realização dos movimentos de arfagem e guinada do sistema. A relação existente entre a variação do PWM (Pulse-Width Modulation) e a magnitude das forças em cada motor, foi determinada experimentalmente utilizando uma célula de carga. Esse procedimento foi repetido duas vezes e em seguida tirou-se o valor médio para cada variação do PWM sendo que os resultados obtidos estão mostrados na tabela 1.

TABELA 1 Resultados experimentais obtidos para a determinação da força de sustentação em relação à variação do PWM

PWM Variação do Duty Cycle (%)	Ensaio 1 (N)	Ensaio 2 (N)	Força de sustentação (N)
0	0,0000	0,0000	0,0000
10	0,0745	0,0783	0,0764
20	0,2794	0,3079	0,2937
30	0,5079	0,7120	0,6100
40	0,8885	1,0238	0,9561
50	1,2905	1,3395	1,3150
60	1,6052	1,7720	1,6887
70	1,9260	2,1476	2,0368
80	2,2898	2,4565	2,3732
90	2,6056	2,8184	2,7120
100	3,0018	3,1312	3,0665

Após os ensaios, traçou-se o gráfico da força de empuxo nos motores em função do *duty cycle* aplicado. A Figura 6 apresenta o resultado obtido para F_{M1} ou F_{M2} .

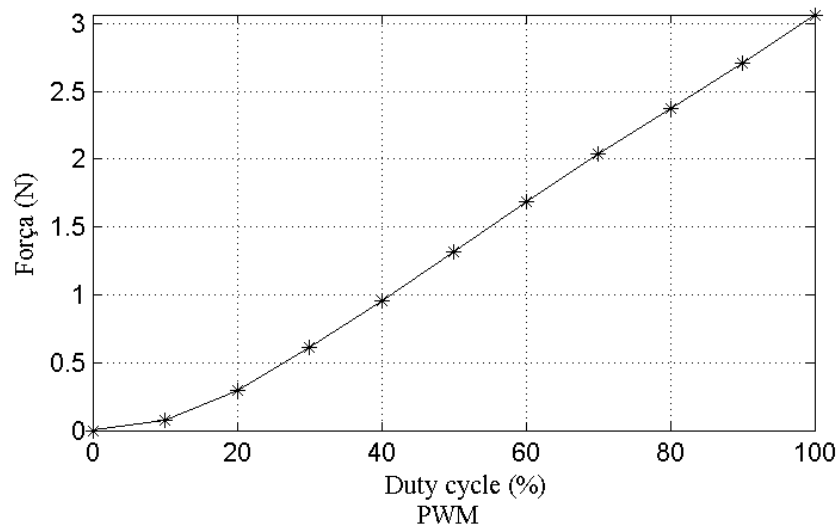


Figura 6 Curva experimental do empuxo do motor em relação à variação do PWM

Os pontos experimentais foram interpolados por um polinômio de terceiro grau, e a função FIT do *MATLAB*[®] foi utilizada para determinar os coeficientes da equação característica, conforme o resultado mostrado abaixo.

$$F_{M1} = -2,4645e^{-6}u_1^3 + 4,5357e^{-4}u_1^2 + 0,0077u_1$$

em que, u_1 pode ser substituído por u_2 para se obter a força gerada pelo segundo motor (F_{M2}).

Porém a força total de sustentação é dada pela soma das forças geradas por cada um dos motores.

$$F_M = F_{M1} + F_{M2}$$

Na outra extremidade da haste maior, está colocado um contrapeso que tem a finalidade de fazer o equilíbrio das forças, auxiliando para que o conjunto se mantenha suspenso no ar, e que pode também ser utilizado para simular alterações nas características físicas do sistema, como pode ser observado na Figura 7. Este sistema dispensa a necessidade de um rotor de cauda.



Figura 7 Contra peso

Os movimentos do sistema serão controlados através da variação de tensão de alimentação nos dois motores, permitindo realizar os seguintes movimentos:

- arfagem: é definido como o movimento em torno do eixo transversal; para que isso ocorra, é necessário alterar, os níveis de tensão dos motores, de forma igual;

- guinada: é definido como o movimento em torno do eixo vertical e pode ser efetuada em dois sentidos:

- guinada para a direita;
- guinada para a esquerda.

Para que ocorra a guinada, é necessário que se aplique tensões diferentes a cada motor, com isso, haverá a rolagem do eixo dos motores, fazendo com que todo o sistema se movimente em torno do eixo vertical.

Rolagem: É definido como o movimento em torno do eixo longitudinal, e também pode ser efetuado em dois sentidos:

- rolamento para a esquerda;

- rolamento para a direita.

O sistema possui também três encoders com uma resolução de 2000 pontos por revolução, que são responsáveis em fornecer o posicionamento do modelo em cada instante de tempo (Figura8).

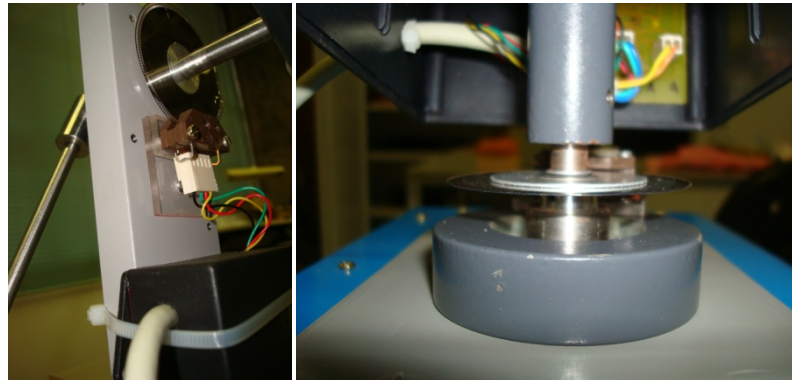


Figura 8 Encoders utilizados para medição do posicionamento do sistema de hélices paralelas

A comunicação entre o encoder e o sistema de aquisição de dados dSPACE é feito através de uma placa com comunicação RS-422, que é mais utilizada em comunicação ponto a ponto, permitindo o envio de informações, de maneira extremamente confiável e tornando o sinal completamente imune de ruídos. A Figura 9 mostra a placa de circuito impresso confeccionada para fazer a leitura dos encoders.

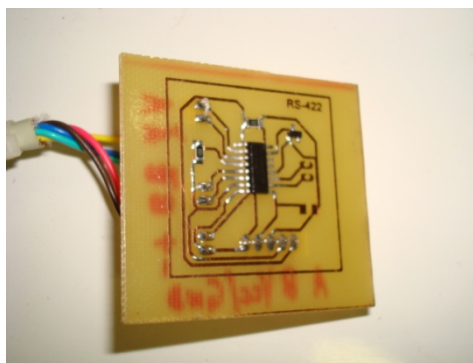


Figura 9 Placa de comunicação do encoder (RS-422)

3.3 Modelagem Matemática

Esta seção tem como objetivo apresentar os conceitos básicos relativos a um processo de modelagem, discutindo técnicas de modelagem de Sistemas Multicorpos (MBS - *MultiBody Systems*). Uma breve descrição do *software ADAMS®* utilizado no desenvolvimento do trabalho também é apresentada.

No estudo de um problema de modelagem, vários níveis de complexidades distintas são envolvidos. A capacidade da definição apropriada dos aspectos relevantes de um problema de modelagem é uma qualidade requerida de cientistas e engenheiros. De qualquer maneira, os seguintes passos são parte de um processo de modelagem (MONTEZUMA, 2003).

1. Descrever um modelo físico de um sistema o qual contém os aspectos relevantes para o estudo desejado, com suas hipóteses simplificadoras;
2. Obter equações que descrevem matematicamente o comportamento do sistema;
3. Resolver as equações resultantes analítica ou numericamente com o objetivo de estimar o comportamento do sistema;
4. Verificar os resultados do modelo por comparação com o comportamento do sistema real;
5. Modificar o sistema físico, se necessário, ou usá-lo para propostas de análise e projeto.

O uso de computadores para realizar as tarefas 2 e 3 é uma prática comum nestes dias, como por exemplo, os métodos dos elementos finitos, programas de geração de equações MBS, linguagens de simulação para propostas diversas, etc., e eles permitem aos cientistas e engenheiros concentrarem-se nos aspectos mais importantes do estudo, como as tarefas 1 e 5.

3.3.1 Técnicas MBS de Modelagem

A engenharia mecânica tem utilizado muito os computadores no processo de desenvolvimento de projetos, desde a geração automática de desenhos de projeto até a simulação de possíveis produtos mecânicos. Uma das técnicas de modelamento empregada em sistemas mecânicos é a técnica de Sistemas Multicorpos, implementada em vários sistemas de software disponíveis comercialmente.

Técnicas de modelagem MBS podem ser usadas para realizar estudos de análise e projeto de qualquer sistema mecânico que possa ser modelado com o conjunto de corpos rígidos interconectados por juntas, influenciados por forças, dirigido por movimentos prescritos e limitado por vínculos.

As equações do movimento para estes sistemas são muito laboriosas e difíceis de serem geradas manualmente, até para um sistema compreendido por um número pequeno de corpos interconectados. Portanto, foi um grande progresso o fato dos formalismos serem desenvolvidos, através dos quais as equações do movimento para um MBS podem ser geradas por computador, baseado em dados simples sobre os corpos (geométricos e inerciais) e as suas interconexões (vínculos cinemáticos e leis de força). Sistemas típicos que podem ser estudados por tais técnicas incluem espaçonaves articuladas, veículos terrestres, mecanismos e máquinas em geral, manipuladores, dispositivos eletromecânicos de alta velocidade, etc. (MONTEZUMA, 2003).

Os Engenheiros utilizam tais implementações computacionais no campo de Simulação de Sistemas Mecânicos (MSS - *Mechanical System Simulation*), permitindo desenvolver, testar, reajustar e otimizar o desempenho de sistemas mecânicos submetidos a grandes movimentos. Esta poderosa capacidade é conhecida como "prototipagem virtual".

Tradicionalmente, as empresas constroem protótipos para aprender sobre sistemas complicados, possibilitando a tomada de decisões críticas de projeto. Quando se está aprendendo através de um protótipo real, o processo é (PASQUINI, 1998):

1. Projetar, construir e montar o sistema completo;
2. Instrumentá-lo com acelerômetros, extensômetros, etc.;
3. Executar um conjunto de testes;
4. Coletar e organizar os dados;
5. Interpretar os dados;
6. Finalmente, tomar decisões de projeto, as quais frequentemente envolvem mudanças no protótipo e a repetição do processo.

No caso da prototipagem virtual, o processo é (PASQUINI, 1998):

1. Construir (modelar) o sistema;
2. Instrumentá-lo, apenas solicitando certas saídas de interesse ao programa de MBS;
3. Executar um conjunto de testes (simulações);
4. Comparar visualmente o desempenho de várias alternativas de projeto através de animações gráficas;
5. Interpretar os dados, já automaticamente coletados, organizados e plotados;
6. Rapidamente tomar decisões de projeto.

A prototipagem virtual libera o engenheiro para focalizar os aspectos criativos do projeto, interpretando os dados de testes, e tomando novas decisões. Em adição, uma vez que se avaliaram os testes iniciais do protótipo virtual, pode-se necessitar de novos dados ou de modificações substanciais de projeto para o protótipo. Os programas de MBS são, em geral, adaptados para esta necessidade; rapidamente pode-se alterar o modelo, pedir novos dados de saída, e executar novamente os testes. Em contraste, trabalhando com protótipos reais, o

processo de construção e instrumentação deve ser completamente reproduzido, e o tempo reprogramado.

Deve-se atentar para o fato de que a prototipagem virtual não elimina a construção de protótipos reais. O uso desta tecnologia reduz o tempo e o custo perdidos nos inúmeros testes normalmente realizados nos protótipos reais, otimizando facilmente parâmetros críticos de projeto, com a simulação do modelo nas mais diversas condições de operação. Assim, pode-se construir o protótipo real, sabendo-se que será necessário executar poucos testes, para confirmar e validar os resultados das simulações.

3.3.2 ADAMS® - Ferramenta Computacional para MBS.

Segundo Pasquini (1998) durante uma simulação, o *ADAMS*® descreve a posição e a orientação de todas as partes do sistema mecânico em termos de seis coordenadas, três translacionais, e três angulares. Armazena os deslocamentos, velocidades e acelerações, translacionais e angulares, dependentes do tempo, no vetor de estado. O vetor de estado também contém os valores correntes das reações e das forças aplicadas em cada uma das partes nas posições inerciais e nos vínculos. Assim, o vetor de estado fornece uma descrição completa do sistema mecânico para a simulação dos mesmos.

Com base nos resultados da simulação, pode-se ajustar os parâmetros de projeto para melhorar o desempenho do sistema.

As leis da física formam a base para o campo de MBS. As ferramentas de simulação MBS implementam as leis de Newton do movimento, ou uma formulação equivalente dos princípios físicos. O *ADAMS*® usa o sistema de equações de Euler-Lagrange do movimento. Este sistema, composto de equações algébricas e diferenciais de segunda ordem, requer métodos numéricos sofisticados para sua solução.

O *ADAMS*[®] define o seguinte, para especificar o modelo mecânico para uma simulação:

- Características inerciais das partes;
- Interação entre as partes;
- Movimentos e forças para o sistema.

O modelo pode também incluir equações diferenciais (de primeira ordem) e algébricas adicionais acopladas ao, ou independentes do, sistema mecânico.

Os dados de entrada para uma simulação inclui os seguintes parâmetros:

- Massa e inércia dos corpos rígidos ou partes;
- Definição dos aspectos geométricos do sistema incluindo os centros de massa para as partes, as posições nas partes das juntas que unem os elementos do sistema, e os pontos nos quais as funções de movimento específicas e forças se aplicam;
- Conectividade para o sistema (o mecanismo para a conexão das partes) definidos em termos das juntas mecânicas, e outros vínculos e elementos elásticos;
- Descrição das forças externas e excitações agindo no sistema;
- Atributos gráficos que possibilitam a animação e a visualização do comportamento do sistema.

Uma biblioteca de juntas padrões simplifica a tarefa da definição das conexões entre as partes do sistema.

Cada simulação começa com uma análise dos dados de entrada para verificar se a descrição do sistema mecânico é completa, correta e consistente.

O processo de verificação é minucioso. Um conjunto completo de mensagens identificará quaisquer erros que poderão ocorrer nos dados de entrada. A montagem das equações diferenciais e algébricas (DAE's) para a simulação real do problema apenas acontecerá após os dados de entrada passarem pelos padrões listados acima.

Segundo Montezuma (2003) uma desvantagem do processo de modelagem utilizando o software de sistemas multicorpos *ADAMS*[®] é o fato de não apresentar as equações dinâmicas que governam o movimento. Isso impossibilita uma análise para conhecer melhor os tipos de não linearidades presentes no sistema.

3.3.3 Sistema de Coordenadas

Segundo Chace (1984) um problema computacional essencial em simulação dinâmica de grandes deslocamentos é a construção automática e a simultânea solução numérica de um conjunto de equações diferenciais ordinárias sendo conhecidas as condições iniciais. A cada passo de integração todos os coeficientes devem ser numericamente calculados e o conjunto de equações resolvido, sendo que este possui um alto grau de variáveis dependentes. A velocidade desse processo depende muito da escolha das coordenadas para representar o sistema. O *ADAMS*[®] utiliza um sistema de coordenadas que consiste simplesmente de três deslocamentos retangulares do centro de massa de cada parte mais a rotação das mesmas com os ângulos medidos em relação ao sistema de coordenadas global. Isto tende a resultar em um grande mas esparsa sistema de equações, onde é possível o uso de métodos de matrizes esparsas para solução simultânea.

A restrição de movimento no *ADAMS*[®], seja através de juntas ou outros vínculos, remove graus de liberdade do modelo através da inclusão de equações algébricas ao conjunto de equações do sistema, essas expressões de restrição de movimento são numerosas e variadas, mas somente envolvem duas partes adjacentes normalmente ligadas por uma junta, e que dessa forma as equações são esparsas. O *ADAMS*[®] pode gerar centenas de equações diferenciais e algébricas, mas que são resolvidas rapidamente por possuir um algoritmo usando matrizes esparsas e um avançado sistema de integração.

3.3.4 Tipos de Análises

Podemos realizar seis diferentes tipos de análises dependendo das características do problema e do interesse que se tem. Cada um destes tipos de análises é útil em um contexto apropriado. As várias análises fornecem diferentes informações sobre o sistema, como é descrito a seguir (CHACE, 1984; PASQUINI, 1998).

1. O *ADAMS*[®] realiza uma análise de condições iniciais (o processo de montagem) antes das análises estática, quase-estática e dinâmica. Após processar os dados de entrada e antes da simulação começar, o *ADAMS*[®] requer um consistente conjunto de valores para o estado do sistema. Isto é, os deslocamentos e as velocidades têm que satisfazer as equações de vínculo que definem o sistema. Em sistemas complexos, os valores iniciais que se especificam para deslocamentos e velocidades frequentemente violam as equações de vínculo. A análise das condições iniciais modifica os dados de entrada para os deslocamentos e velocidades necessários para satisfazer os vínculos do sistema enquanto minimiza a mudança nos dados. Frequentemente, é conveniente especificar apenas pequenos deslocamentos e velocidades iniciais, e permitir que a análise das condições iniciais monte o resto para o sistema de forma adequada.
2. Uma análise cinemática simula o movimento do sistema. Ela permite ao engenheiro determinar a faixa de valores possíveis para o deslocamento, velocidade e aceleração de qualquer ponto de interesse em um equipamento mecânico. Especificando-se as propriedades de massa e de inércia de uma parte, o *ADAMS*[®] também calcula as correspondentes forças aplicadas e de reação requeridas para gerar os movimentos prescritos. Uma aplicação típica desta análise

é a avaliação preliminar de um modelo complicado para posterior análise dinâmica.

3. A análise de equilíbrio estático determina um estado para o sistema para balancear todas as forças internas e externas na ausência de quaisquer movimentos ou forças inerciais. Todas as velocidades e acelerações do sistema são ajustadas para zero. Frequentemente acha-se um ponto de partida para uma análise dinâmica pelo uso de uma análise estática para remover transientes indesejados no início da simulação.
4. Uma análise quase-estática é uma sequência de análises estáticas realizadas para diferentes configurações do sistema, tipicamente, em intervalos fixos de tempo por todo um movimento prescrito para o sistema.
5. O tipo mais complexo de análise é a análise dinâmica. Vários integradores diferentes estão disponíveis para achar a solução para um sistema completo de equações diferenciais e algébricas (DAE's). A análise dinâmica fornece solução no tempo (história no tempo) para todos os deslocamentos, velocidades, acelerações, forças de reação internas em um sistema mecânico dirigido por um conjunto de forças e excitações externas.
6. Na análise linear, o *ADAMS*[®] lineariza o sistema de equações não-lineares para o modelo sobre um ponto de operação. Isto resulta em um conjunto de equações lineares invariantes no tempo na forma

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{3.1}$$

onde x é o vetor de estados, y vetor de saídas, u vetor de entradas da planta e A , B , C e D matrizes de espaço de estados invariantes no tempo (SOHONI, 1989).

3.4 Modelagem do Sistema de Hélices Paralelas

O modelo foi construído utilizando o software de simulação de sistemas mecânicos *ADAMS*[®], através do módulo *ADAMS*[®]/*View*, que utiliza as técnicas de modelagem de sistemas multicorpos para a obtenção de equações dinâmicas do movimento. Este software oferece uma interface gráfica para auxiliar a criação do modelo matemático e para a visualização dos resultados da simulação, onde os parâmetros necessários são fornecidos de uma maneira simples e rápida.

Como já mencionado, o *ADAMS*[®] usa o sistema de equações de Euler-Lagrange do movimento sendo composto de equações algébricas e diferenciais de segunda ordem. Em geral, todo o equacionamento de programas como o *ADAMS*[®] fica oculto. Uma interface simples, com ícones para a criação de modelos multicorpos torna fácil a sua manipulação.

O trabalho de Orlandea (ORLANDEA, 1973; ORLANDEA & CHACE, 1977) originou o programa para simulação de sistema multicorpos *ADAMS*[®] (“Automatic Dynamics Analysis of Mechanical Systems”). O *ADAMS*[®] é um programa para simulação de sistemas em três dimensões que utiliza a técnica de matrizes esparsas para resolver as equações algébricas lineares e o método de Gear para a integração das equações diferenciais. As equações dinâmicas do sistema são descritas como equações de Lagrange e os vínculos são descritos pelos multiplicadores de Lagrange.

O modelo do Sistema de Hélices Paralelas, mostrado na Figura 10, foi desenvolvido com características geométricas e inerciais aproximadas do modelo experimental que se deseja simular. O cálculo de inércia foi realizado pelo próprio *ADAMS*[®] através da geometria das peças e das massas de cada material. Posteriormente foram obtidas as massas reais com o auxílio de uma célula de carga, com a finalidade de melhorar a aproximação do modelo virtual com o modelo real.

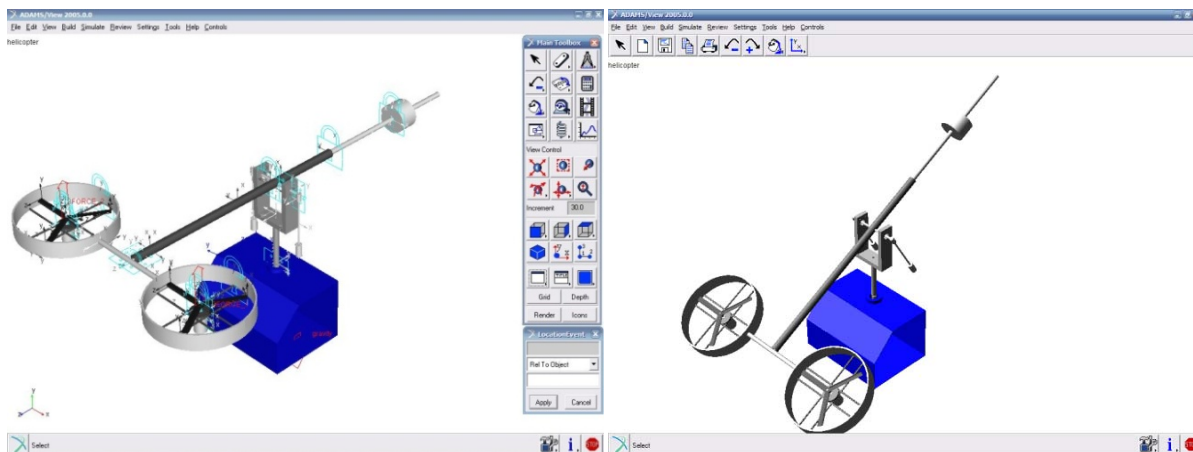


Figura 10 Modelo Virtual desenvolvido no ambiente multicorpos *ADAMS*[®]

3.5 Análise do Modelo Linear na forma de Espaço de Estados

O modelo linear descrito nesta seção é obtido através do comando *LINEAR*, presente no programa *ADAMS*[®], que calcula tanto os modos e autovalores do modelo como a representação na forma de espaço de estados (*A*, *B*, *C* e *D*). A representação linear fornece importantes características do comportamento dinâmico do modelo. Propriedades de estabilidade do sistema têm relação direta com a parte real dos autovalores complexos. Autovalores com parte real positiva representam modos instáveis. A análise dos autovalores ainda fornece características do amortecimento e do comportamento oscilatório através de suas componentes reais e imaginárias (BARBIERE, 2002).

O vetor de estados, automaticamente gerado pelo *ADAMS*[®], é composto pelas coordenadas generalizadas *q* e pelas velocidades generalizadas *u*. Segundo Sohoni (1986) as coordenadas generalizadas escolhidas como variáveis de estado são aquelas que apresentam as mais rápidas variações dentro do modelo (BARBIERE, 2002). As linearizações devem ser realizadas em torno do ponto de operação, após a realização de uma simulação estática ou dinâmica. As matrizes das equações de estado e de saída (*A*, *B*, *C* e *D*) foram obtidas através

de uma linearização com o sistema de hélices paralelas na sua posição de equilíbrio sendo elas:

$$A = \begin{bmatrix} 0 & -1,9392 & 0 & 0,6464 & 0 & -0,0003 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,6851 & 0 & -1,9004 & 0 & 0,0003 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1,2647 & 0 & -1,1931 & 0 & 0,0009 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -155,6325 & 408,8657 \\ 0 & 0 \\ 416,4449 & -148,0091 \\ 0 & 0 \\ -11,1962 & 11,1962 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0,0011 & 0 & 0,0011 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0,0022 \\ 0 & 0,0022 & 0 & -0,0021 & 0 & 0 \\ 0,0011 & 0 & 0,0011 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0,0022 & 0 \\ 0,0022 & 0 & -0,0021 & 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Capítulo 4

SISTEMA DE CONTROLE

4.1 Sistema de Controle Seguidor com Realimentação de Estados

O sistema linearizado de malha aberta é representado abaixo pelas equações de estado de n -ésima ordem e as equações de saída de p -ésima ordem.

$$\dot{x} = Ax + Bu \quad (4.1)$$

$$y = Cx = \begin{bmatrix} E \\ F \end{bmatrix} x \quad (4.2)$$

onde A é chamada de matriz de estado, B , de matriz de entrada, C , de matriz de saída, D , de matriz de transição direta, y é um vetor $p \times 1$ e $w = Ex$ é um vetor $m \times 1$ representando as saídas que são requeridas para seguir o vetor de entrada r .

O controle por realimentação de estados é projetado com o intuito de que o vetor w siga o comando de entrada r no sentido de que a resposta em regime seja

$$\lim_{t \rightarrow \infty} w(t) = r(t) \quad (4.3)$$

quando r é um comando de entrada constante por partes. Segundo D'Azzo e Houpis (1995) o método do projeto consiste em adicionar um vetor comparador e integrador que satisfaça a equação

$$\dot{z} = r - w = r - Ex \quad (4.4)$$

O sistema de malha aberta composto é então governado pelas equações aumentadas de estado e saída formadas a partir das equações (4.1) a (4.4)

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -E & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r = \bar{A}x' + \bar{B}u + \bar{B}' r \quad (4.5)$$

$$y = [C \ 0] \begin{bmatrix} x \\ z \end{bmatrix} = \bar{C}x' \quad (4.6)$$

onde

$$\bar{A} = \begin{bmatrix} A & 0 \\ -E & 0 \end{bmatrix}; \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}; \quad \bar{B}' = \begin{bmatrix} 0 \\ I \end{bmatrix}; \quad \bar{C} = [C \ 0]. \quad (4.7)$$

$$\bar{A} = \begin{bmatrix} 0 & -1,9392 & 0 & 0,6464 & 0 & -0,0003 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,6851 & 0 & -1,9004 & 0 & 0,0003 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,2647 & 0 & -1,1931 & 0 & 0,0009 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -0,0011 & 0 & -0,0011 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,0022 & 0 & 0 \end{bmatrix};$$

$$\bar{B} = \begin{bmatrix} -155,6324 & 408,8657 \\ 0 & 0 \\ 416,4449 & -148,0091 \\ 0 & 0 \\ -11,1962 & 11,1962 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

D'Azzo e Houpis (1995) mostram que a lei de controle a ser usada é

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = K_1 x + K_2 z = [K_1 \ K_2] \begin{bmatrix} x \\ z \end{bmatrix} \quad (4.8)$$

$$\text{sendo } \bar{K} = [K_1 \ K_2]. \quad (4.9)$$

$$\bar{K} = \begin{bmatrix} -0,0070 & -0,0321 & -0,0132 & 0,0003 & -0,0549 & -0,0811 & 9,2680 & -18,5786 \\ -0,0243 & -0,1076 & 0,0031 & 0,0715 & -0,1808 & -0,2053 & 9,2680 & -40,5089 \end{bmatrix}$$

A obtenção da matriz \bar{K} é realizada a partir da seleção dos autovalores a serem atribuídos à matriz da planta de malha fechada A'_{cl} na equação (4.12)

$$\sigma(\bar{A} + \bar{B}\bar{K}) = \{\lambda_1, \lambda_2, \dots, \lambda_{n+m}\} \quad (4.13)$$

e um conjunto associado de autovetores

$$v(\bar{A} + \bar{B}\bar{K}) = \{v_1, v_2, \dots, v_{n+m}\} \quad (4.14)$$

que são selecionados a fim de se obter as características de resposta no tempo desejadas. Os autovalores e autovetores são relacionados pela equação

$$(\bar{A} + \bar{B}\bar{K})v_i = \lambda_i v_i \quad (4.15)$$

que pode ser colocada na forma

$$[\bar{A} - \lambda_i I \ \bar{B}] \begin{bmatrix} v_i \\ g_i \end{bmatrix} = 0 \quad \text{para } i = 1, 2, \dots, n + m, \quad (4.16)$$

onde v_i é o autovetor e

$$g_i = \bar{K}v_i \quad (4.17)$$

Para satisfazer a equação (4.16), o vetor $[v_i^T \ g_i^T]$ deve pertencer ao *kernel* de

$$\bar{S}(\lambda_i) = [\bar{A} - \lambda_i I \ \bar{B}] \quad \text{para } i = 1, 2, \dots, n + m \quad (4.18)$$

A notação $\ker \bar{S}(\lambda_i)$ é usada para definir o espaço nulo que contém todos os vetores $[v_i^T \ g_i^T]$ para que a equação (4.16) seja satisfeita (D'AZZO E HOUPIS, 1995). A equação (4.17) pode ser usada para formar a igualdade matricial

$$[g_1 \ g_2 \ \dots \ g_{n+m}] = [\bar{K}v_1 \ \bar{K}v_2 \ \dots \ \bar{K}v_{n+m}] \quad (4.19)$$

de onde se obtém a matriz \bar{K} como segue:

$$\bar{K} = [g_1 \ g_2 \ \dots \ g_{n+m}][v_1 \ v_2 \ \dots \ v_{n+m}]^{-1} = QV^{-1} \quad (4.20)$$

Observa-se que os autovalores podem ter valores repetidos de número igual às entradas do sistema. Isso se deve ao fato do espaço nulo ter dimensão igual ao número de entradas. Assim associa-se um autovalor repetido a um vetor da base do espaço nulo. Com isso, todas as colunas da matriz V continuam sendo linearmente independentes e, por isso, a matriz V^{-1} existe.

A síntese do controle por realimentação de estados, apresentada anteriormente assume que todos os estados x são mensuráveis ou que possam ser gerados a partir da saída. Na prática em muitos sistemas de controle é impossível fisicamente ou economicamente impraticável instalar todos os sensores que seriam necessários para medir os estados. Essa metodologia foi apresentada por Montezuma, (2009) em sua qualificação de doutorado. Para que se possam reconstruir os estados da planta, a partir das saídas, requer que todos os estados sejam observáveis. A condição necessária para observabilidade completa é dada pela equação (4.21).

$$\text{posto } M_o = \text{posto} [C^T \ A^T C^T \ (A^T)^2 C^T \ \dots \ (A^T)^{n-1} C^T] = n \quad (4.21)$$

O objetivo é apresentar uma forma de reconstruir os estados a partir das saídas medidas de um sistema dinâmico, utilizando um observador de estados. O vetor de estados estimados \hat{x} pode, então, ser usado para implementar a lei de controle com realimentação de estados $u = K\hat{x}$.

Um método básico de reconstrução dos estados é simular as equações de estado e de saída da planta. Essas equações serão simuladas utilizando a mesma entrada u que é aplicada ao sistema físico. Os estados do sistema simulado e do sistema físico deverão ser idênticos, somente se as condições iniciais de simulação e do sistema físico forem iguais.

Como a planta física pode estar sujeita a distúrbios não mensuráveis que não podem ser aplicados a simulação, então, a diferença entre a saída atual da planta y e a saída simulada

\hat{y} é usada como outra entrada na equação da simulação. Assim, as equações de estado e de saída do observador se tornam (D'AZZO E HOUPIS, 1995).

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (4.22)$$

$$\hat{y} = C\hat{x} \quad (4.23)$$

onde L é a matriz de ganho $n \times l$ do observador.

Um método para sintetizar L usa a reconstrução do erro do observador definido por

$$e \equiv x - \hat{x} \quad (4.24)$$

Subtraindo a equação (4.22) da equação (4.1) e usando as equações (4.2), (4.23) e (4.24) resulta na equação de estado do erro do observador

$$\dot{e} = (A - LC)e \quad (4.25)$$

Para uma escolha apropriada da matriz L do observador, todos os autovetores de $(A - LC)$ atribuídos devem estar no semiplano complexo esquerdo, de tal forma que o valor em regime do erro $e(t)$ para qualquer condição inicial seja zero:

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (4.26)$$

A equação (4.25) indica que a equação do erro do observador não possui entrada e é excitada somente pela condição inicial, assim o erro do observador independe da entrada do sistema. O valor do erro em regime é desta forma igual a zero. A importância disto é que haverá a convergência dos estados do observador para os da planta, em regime, independentemente de quais sejam as entradas.

Os autovalores de $(A - LC)$ são usualmente escolhidos de tal modo que fiquem a esquerda dos autovalores de A . Assim, se o observador de estados tiver um erro inicial em relação à planta física, este rapidamente tenderá a zero (D'AZZO E HOUPIS, 1995).

A representação da planta física dada pelas equações (4.1) e (4.2) e o observador pelas equações (4.22) e (4.23) são mostrados na Figura 12.

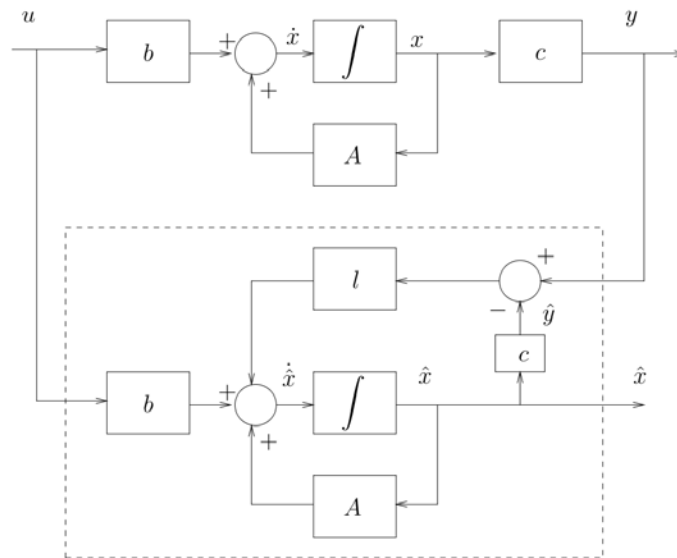


Figura 12 Planta com um observador de estados

4.2 TEORIA DAS VARIÁVEIS DE ESTADO

Para fins de controle é costume dividir os sistemas em SISO (single input, single output) e MIMO (multiple input, multiple output), havendo ainda as classificações intermediárias SIMO e MISO. Como as designações em inglês indicam, um sistema SISO caracteriza-se por uma entrada e uma saída, enquanto os MIMO são sistemas multivariáveis dispendo de várias entradas e saídas. À luz destas considerações facilmente concluímos que o sistema que nos propomos a estudar pode ser visto como um MIMO onde as entradas serão as tensões aplicadas nos motores, e as saídas serão os valores dos diferentes graus de liberdade que definem a posição em cada instante.

As técnicas de controle clássicas baseiam-se na determinação de funções de transferência e na utilização de lugares de raízes, diagramas de Bode, curvas de Nyquist, etc.

Estes procedimentos são particularmente bem sucedidos para sistemas SISO. Num sistema MIMO a interação entre as entradas e saídas (a entrada u normalmente influencia todas as saídas) dificulta muito esta análise. A extensão é normalmente feita implementando controladores que anulem a interação de modo a que, em malha fechada, cada entrada só influencie uma saída.

Alternativamente a isto, e de forma mais frequente, recorre-se à representação por equações de estado. O estado é uma estrutura matemática constituída por um conjunto de n variáveis $x(t) = [x_1(t); x_2(t), \dots, x_n(t)]^T$ de tal forma que, se tivermos o estado inicial $x(t_0)$ e os valores das entradas aplicadas $u(t)$ durante um intervalo de tempo $T = [t_0; t_f]$, conseguimos determinar univocamente os valores das saídas $y(t)$ ao longo desse período. Estas variáveis de estado nem sempre são quantidades fisicamente mensuráveis, podendo ser grandezas puramente matemáticas sem qualquer significado.

Se o sistema for linear e $x(t)$, $x(t_0)$, $u(t)$, $y(t)$ forem, respectivamente, o vetor de estado, o estado inicial, as entradas e as saídas, temos a expressão abaixo. A , B , C e D são matrizes cuja dimensão depende do número de variáveis de estado, de entradas e de saídas.

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad \text{Equação de Estado (dinâmica do sistema)}$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad \text{Equação de Saída (observação do sistema)}$$

No caso do sistema não ser linear as relações acima podem ser descritas por

$$\dot{x}(t) = F(x; x_0; u; t)$$

$$y(t) = H(x; u; t)$$

A colocação do nosso sistema nesta última forma teria uma dupla vantagem. Por um lado permitiria a aplicação e exploração direta de uma série de técnicas de controle já

estudadas para sistemas MIMO. Por outro, e se repararmos nas considerações que fizemos na apresentação dos algoritmos “ode”, facilmente os poderíamos usar para computar os resultados.

Capítulo 5

SIMULAÇÃO USANDO *SIMULINK*®/*ADAMS*®

5.1 *SIMULINK*®

SIMULINK® é um pacote de software que faz parte do *MATLAB*® desenvolvido pela *Mathworks Inc.* que permite modelar, simular e analisar sistemas cujas saídas variam no tempo. Tais sistemas são frequentemente sistemas dinâmicos podendo ser lineares ou não lineares modelados em tempo contínuo ou discreto. O *SIMULINK*® pode ser usado para explorar o comportamento de uma ampla faixa de sistemas dinâmicos reais incluindo circuitos elétricos, sistemas de freios, de absorção de choques e muito outros sistemas elétricos, mecânicos e termodinâmicos.

O processo de simulação usando o *SIMULINK*® consiste de duas etapas. Primeiro cria-se um modelo gráfico (diagrama de blocos) do sistema a ser simulado usando-se o editor de modelos do *SIMULINK*®. O modelo representa as relações matemáticas dependentes no tempo entre os sistemas de entradas, estados e saídas. Então, simula-se o comportamento do sistema durante um determinado período de tempo, que no caso é o modelo ou a planta do sistema gerado pelo *ADAMS*® e exportado para o *SIMULINK*® como um bloco *s-function*. O controle será realizado no próprio *SIMULINK*® (MONTEZUMA, 2003).

5.2 Simulação e Controle usando *SIMULINK*®/*ADAMS*®

O sistema de controle será montado através de um diagrama de blocos que represente a planta e a lei de controle desejados. São utilizados dois modelos de um sistema de hélices paralelas como planta, gerados a partir do software de modelagem dinâmica *ADAMS*® e

exportado para o *SIMULINK*®. Um linear e um não linear apresentando ambos duas entradas que são geradas pelo empuxo dos motores e seis saídas, das quais três são os movimentos de arfagem, guinada e rolagem, e as outras três são as suas respectivas velocidades angulares.

O modelo linearizado em termos de variáveis de estados é utilizado para realizar o controle em torno da posição de linearização. É representado por quatro matrizes: A , B , C e D . A matriz A possui dimensões $n \times n$ onde n é o número de estados que no caso é igual a duas vezes o número de graus de liberdade do sistema. Como o sistema de hélices paralelas possui três graus de liberdade, n é igual a seis. A matriz B possui dimensões $n \times m$ onde m é igual a dois que é o número de entradas do sistema. A matriz C é $p \times n$ onde p é o número de saídas do sistema e a matriz D é $p \times m$ que no caso tem todos os seus elementos nulos.

Foi atribuído que o vetor de entrada (u) no sistema são as tensões aplicadas nos motores, através da modulação da largura de pulso, e o vetor de saída (y) são as posições e as velocidades angulares de cada eixo (Figura 13).

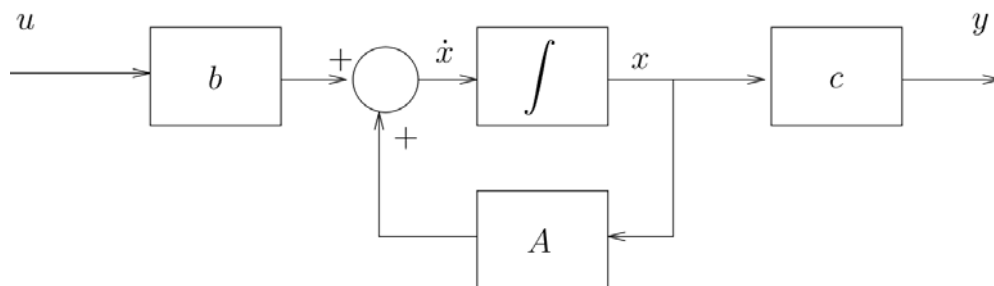


Figura 13 Planta do sistema representada na forma de espaço de estados

O modelo não linear é utilizado para simular o comportamento do sistema de hélices paralelas real (modelo experimental) e testar a eficiência do controle utilizado, isto é, verificar se o controle gerado a partir do modelo linear é capaz de controlar o modelo não linear de forma adequada em torno do ponto de linearização.

Capítulo 6

MATERIAIS E MÉTODOS

O desenvolvimento desse trabalho pode ser dividido em duas etapas distintas, quais sejam, simulação do modelo linear e do modelo não linear em ambiente *MATLAB*[®]/*SIMULINK*[®] e a implementação em tempo real do controlador utilizando a planta experimental do sistema de hélices paralelas, mostrado na Figura 14. A seguir será descrito o procedimento e os materiais utilizados em cada uma destas etapas.

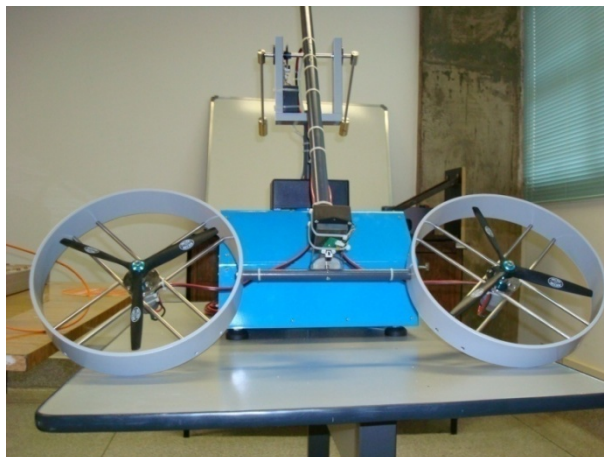


Figura 14 Planta Experimental

6.1 Materiais utilizados

Foi utilizado na simulação um microcomputador com sistema operacional Windows XP, juntamente com o software *ADAMS*[®] e *MATLAB*[®].

O diagrama de blocos utilizados foi implementado em ambiente *MATLAB*[®]/*SIMULINK*[®] e foi utilizado em um modelo linear, gerados pelo software *ADAMS*[®] conforme Figura 15.

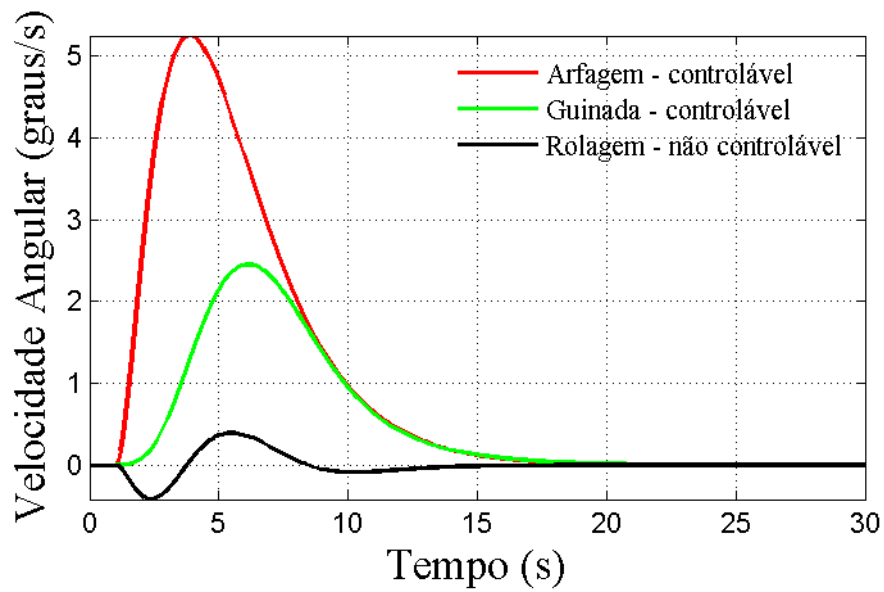


Figura 17 Velocidades angulares de arfagem, guinada e rolagem

Aplicou-se um ruído branco na ação de controle, para testar a eficiência do controle e verificar se o mesmo continua sendo capaz de seguir o valor de entrada requerido. A Figura 18 mostra o comportamento dos deslocamentos dos ângulos de arfagem e guinada com uma perturbação, e a Figura 19 mostra as respectivas velocidades angulares

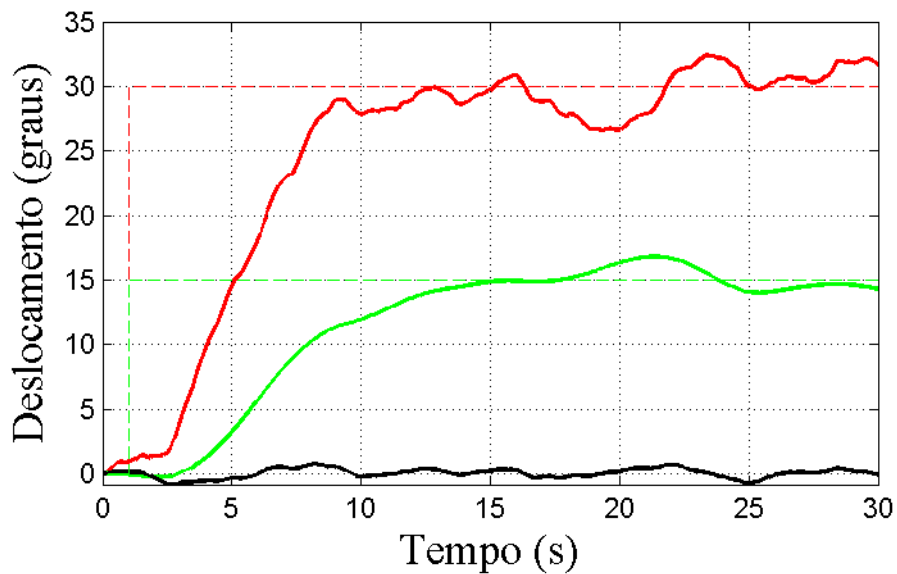


Figura 18 Resposta a uma entrada degrau com ruído branco, de planta linear

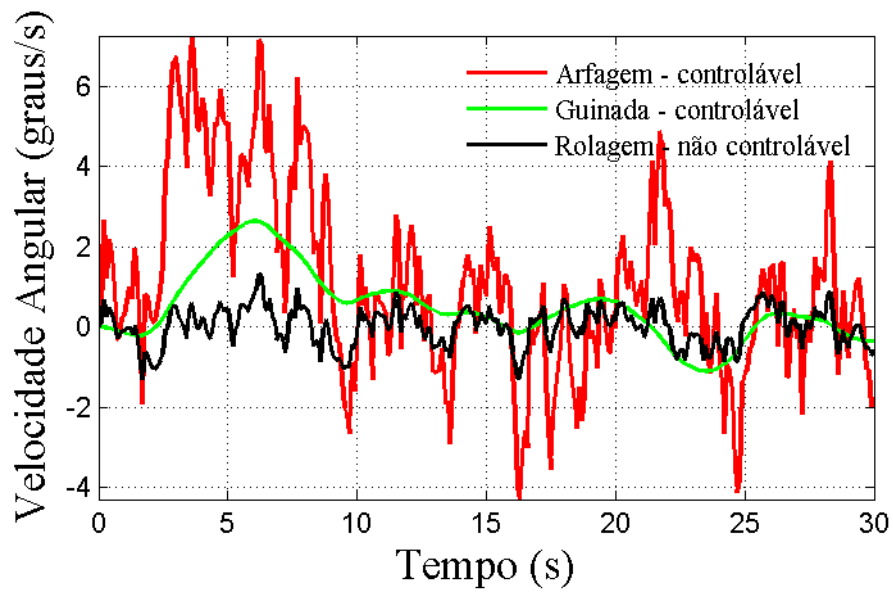


Figura 19 Velocidades angulares de arfagem, guinada e rolagem com ruído branco

Troca-se a entrada degrau por uma entrada rampa para analisar se o controlador ainda continua seguindo o sinal da entrada. Os valores dos ângulos são os mesmos fixados anteriormente, e a resposta dos ângulos de arfagem e guinada são mostrados na Figura 20.

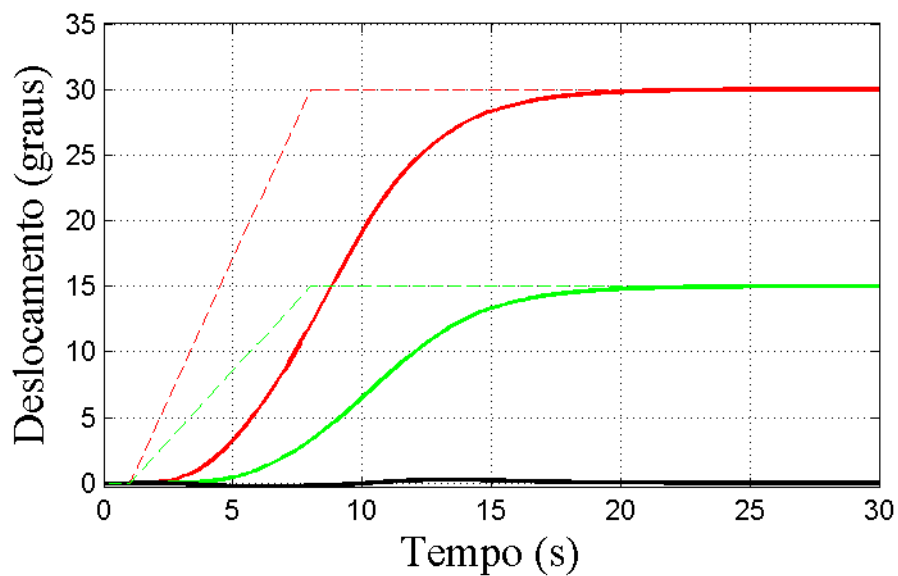


Figura 20 Resposta a uma entrada rampa da planta linear geradas pelo *ADAMS*[®]

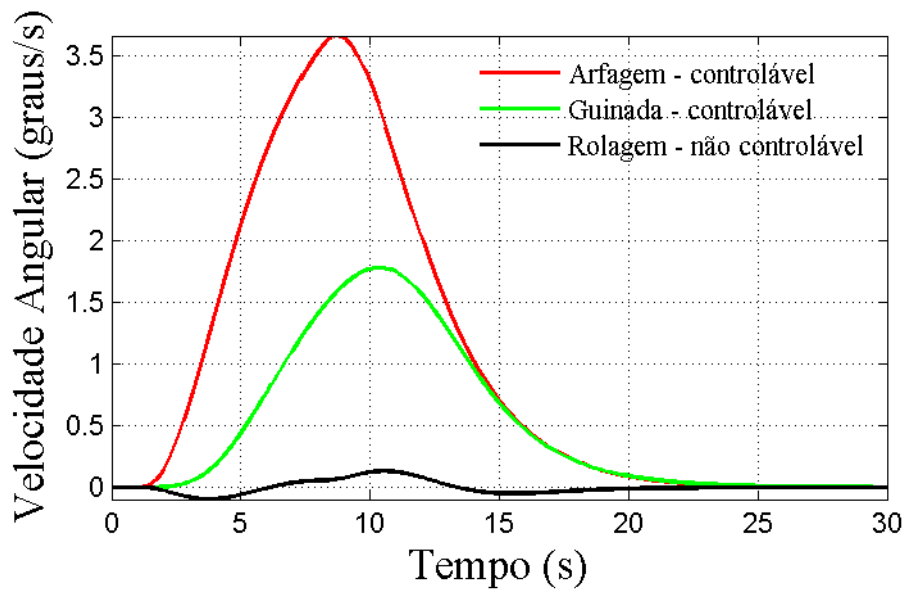


Figura 21 Velocidades angulares de arfagem, guinada e rolagem

Aplicou-se novamente um ruído branco na ação de controle, para entrada rampa, testando a eficiência do controle e verificar se o mesmo continua sendo capaz de seguir o valor de entrada requerido. A Figura 22 retrata o comportamento do controlador em relação aos ângulos de arfagem e guinada da planta linear geradas pelo *ADAMS*[®], e a Figura 33 as velocidades angulares respectivas.

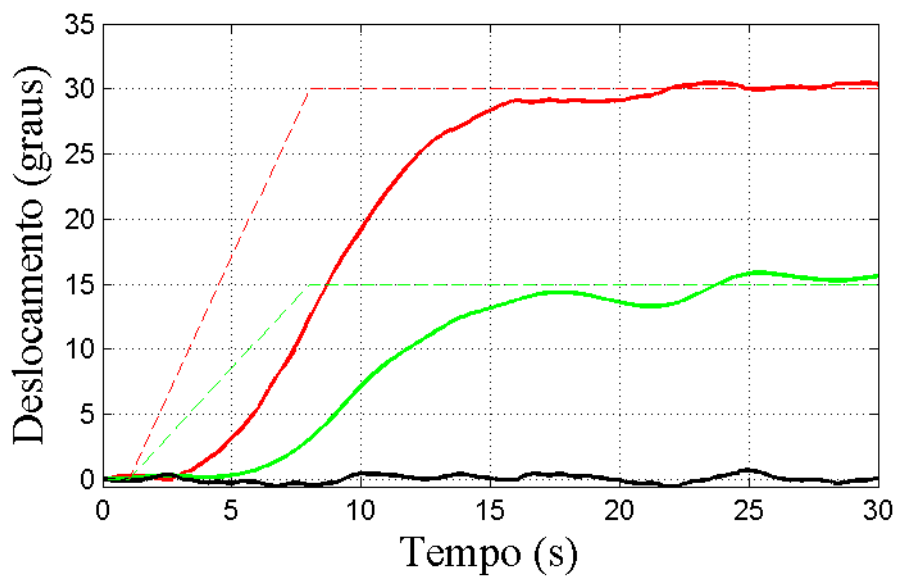


Figura 22 Resposta a uma entrada rampa com ruído

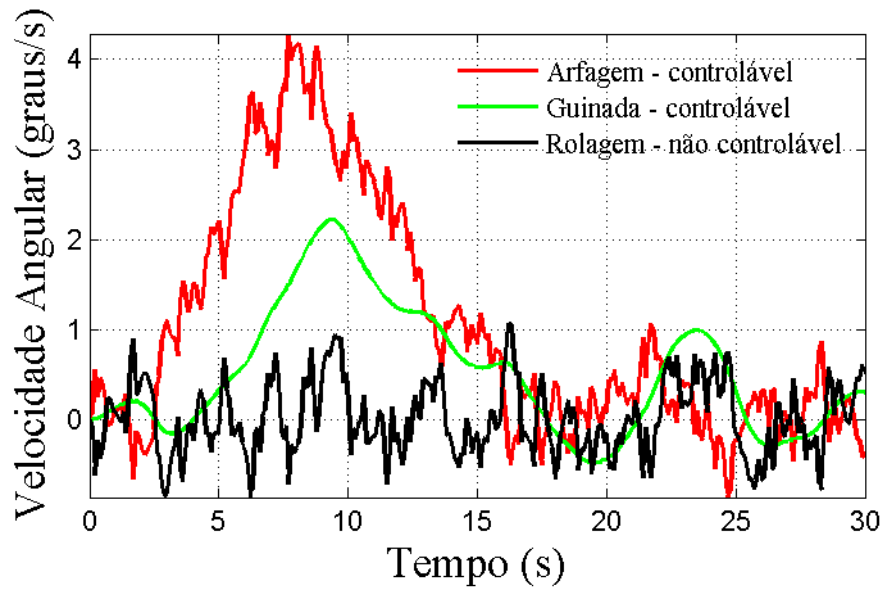


Figura 23 Velocidades angulares com ruído

A mesma simulação foi realizada em um modelo não linear, gerados pelo *ADAMS*[®], para observar se o controlador desenvolvido no modelo linear é capaz de controlar o sistema de hélices paralelas não linear. O diagrama de blocos utilizado na planta não linear é mostrado na Figura 24.

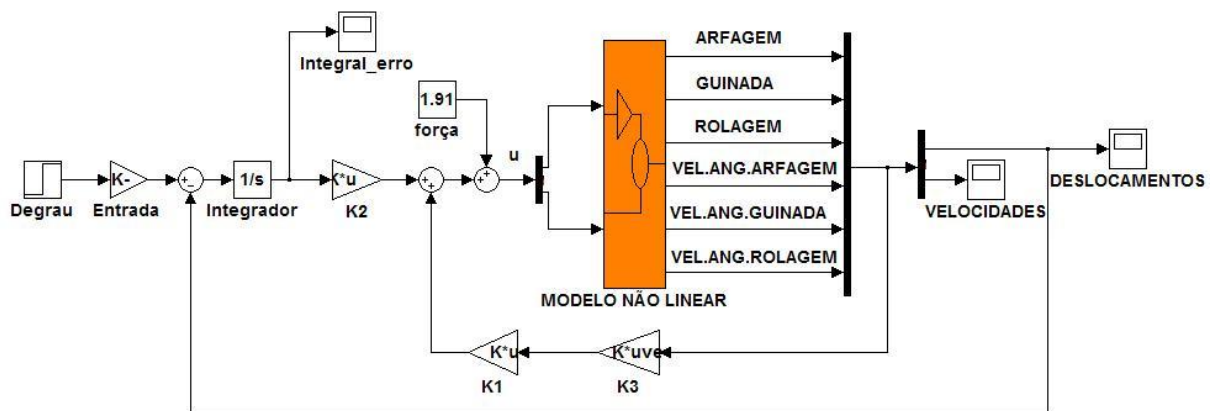


Figura 24 Diagrama de blocos da planta não linear gerada pelo *ADAMS*[®]

Os ângulos utilizados na simulação não linear foram os mesmos utilizados na simulação linear, e a resposta dos ângulos de arfagem e guinada para esta condição é mostrada na Figura 25, com as respectivas velocidades angulares mostradas na Figura 26

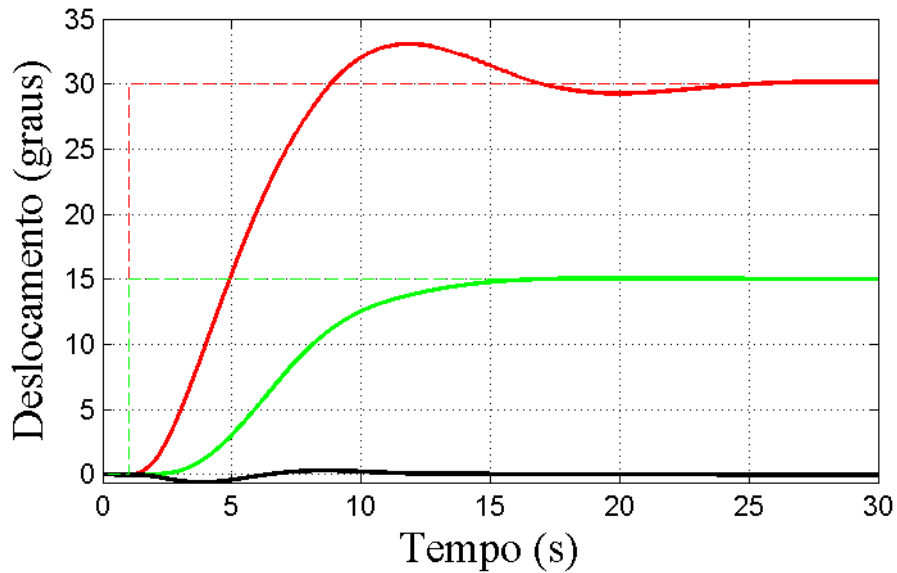


Figura 25 Resposta a uma entrada degrau da planta não linear geradas pelo *ADAMS*[®]

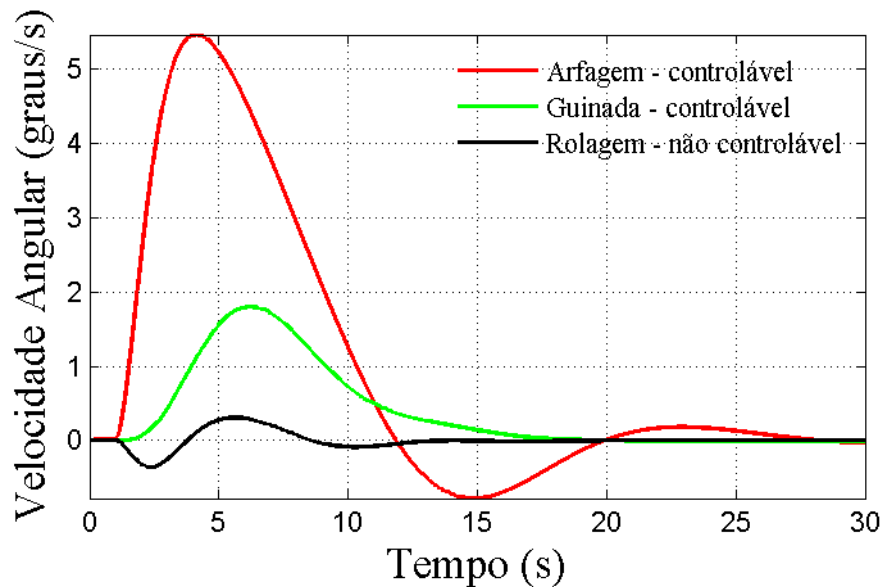


Figura 26 Velocidades angulares de arfagem, guinada e rolagem

Assim como no modelo linear, no modelo não linear foi aplicado um ruído branco na ação de controle, em uma entrada degrau, para analisar o comportamento do controle com uma perturbação externa. A Figura 27 mostra que mesmo com oscilações o sistema de controle seguidor ainda continua atuando. A Figura 28 mostra o comportamento das velocidades angulares do sistema.

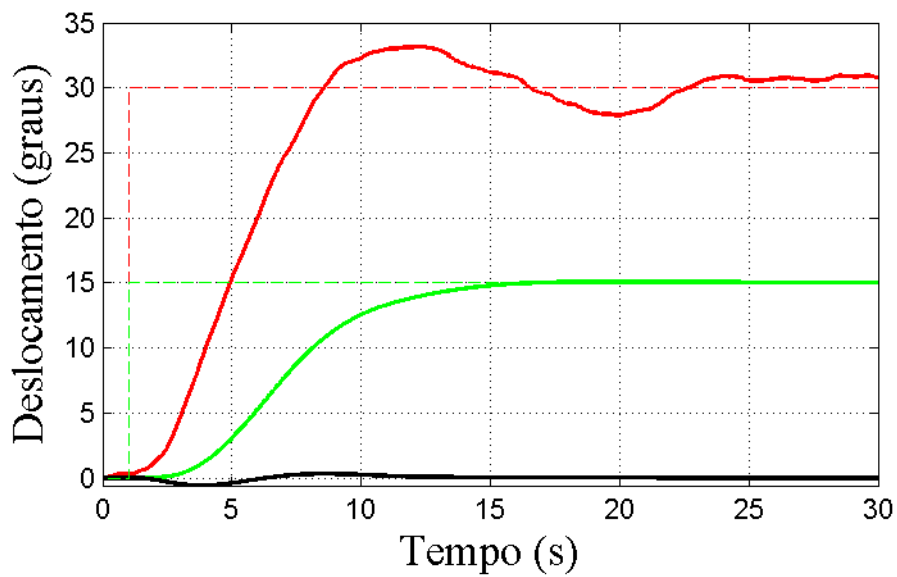


Figura 27 Resposta a uma entrada degrau com ruído da planta não linear

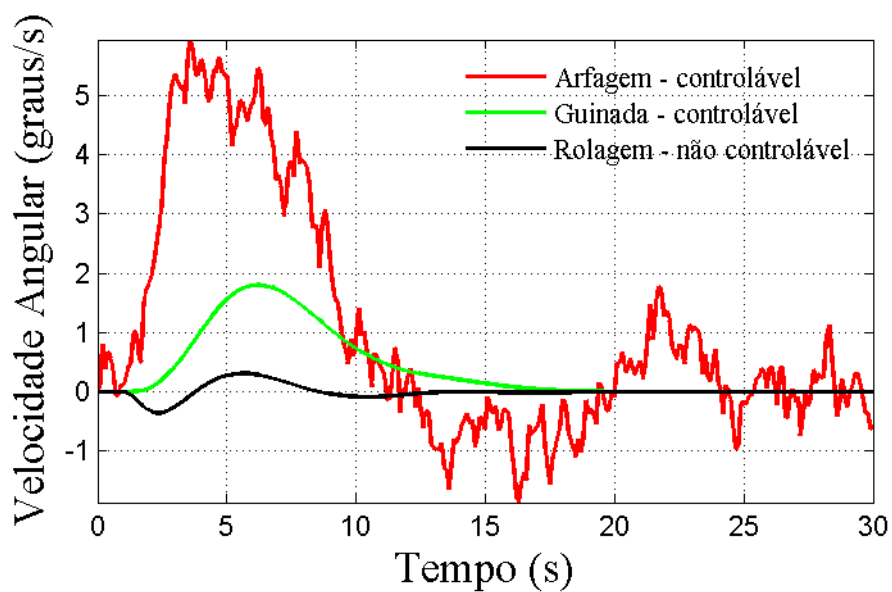


Figura 28 Velocidades angulares de arfagem, guinada e rolagem

Obedecendo a mesma condição do modelo linear, uma entrada rampa também foi utilizada no modelo não linear, com a finalidade de comparar os resultados obtidos e analisar a eficiência do controlador em diferentes situações. A Figura 29 é possível visualizar a resposta dos ângulos de arfagem e guinada para a entrada rampa e na Figura 30 as velocidades angulares respectivas.

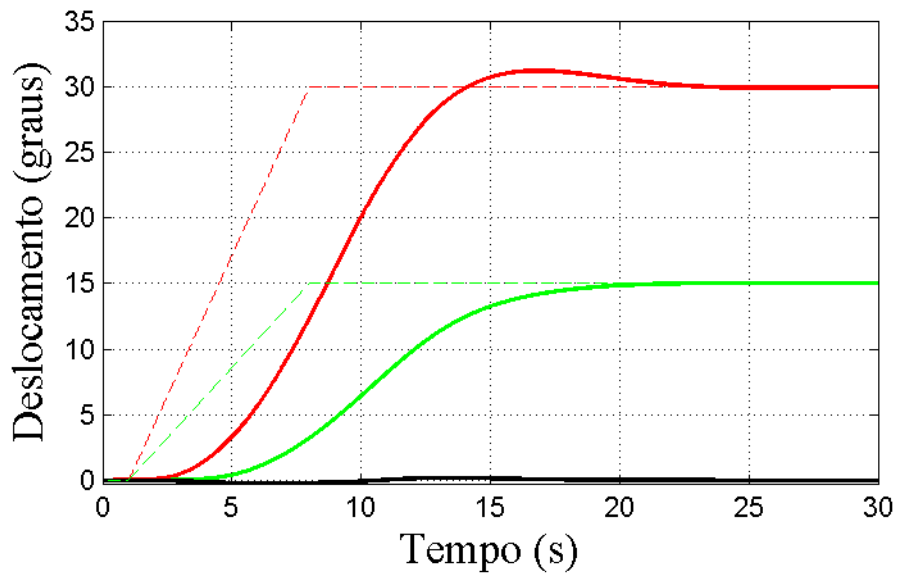


Figura 29 Entrada Rampa aplicada da planta não linear geradas pelo *ADAMS*[®]

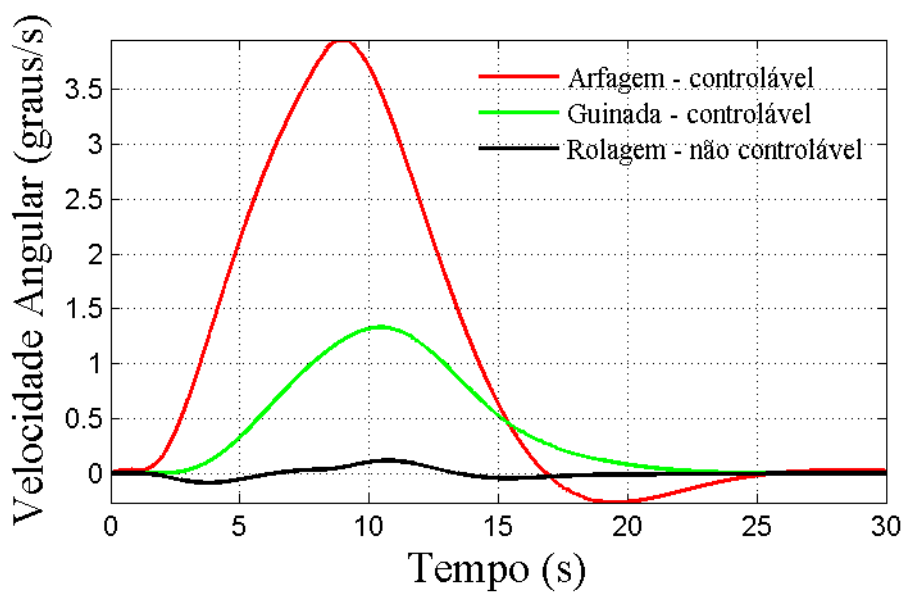


Figura 30 Velocidades angulares de arfagem, guinada e rolagem

Por fim, aplica-se novamente um ruído na ação de controle para uma entrada rampa, com o intuito de verificar o comportamento dinâmico do sistema de hélices paralelas e analisar a capacidade que o controle tem de atuar em situações diversas (Figuras 31 e 32).

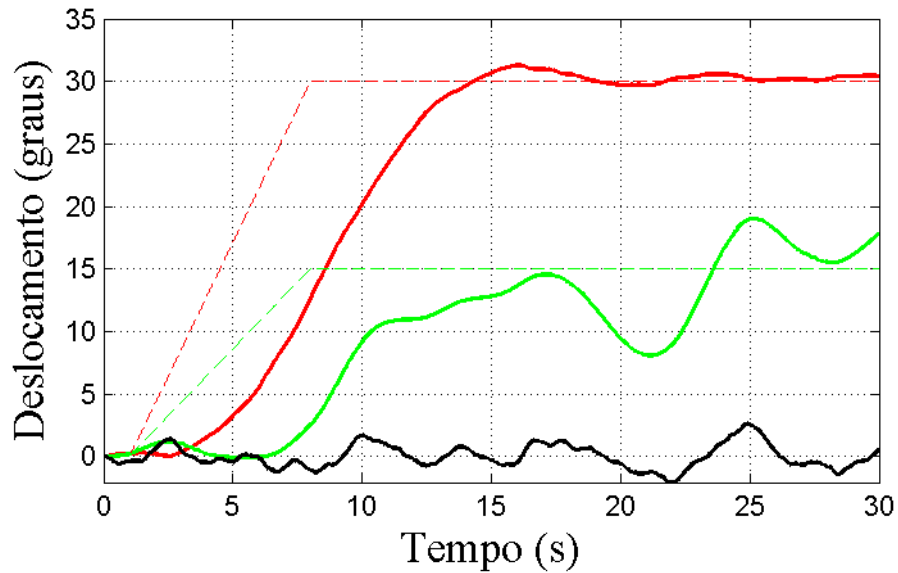


Figura 31 Entrada rampa com ruído, aplicada da planta não linear geradas pelo *ADAMS*[®]

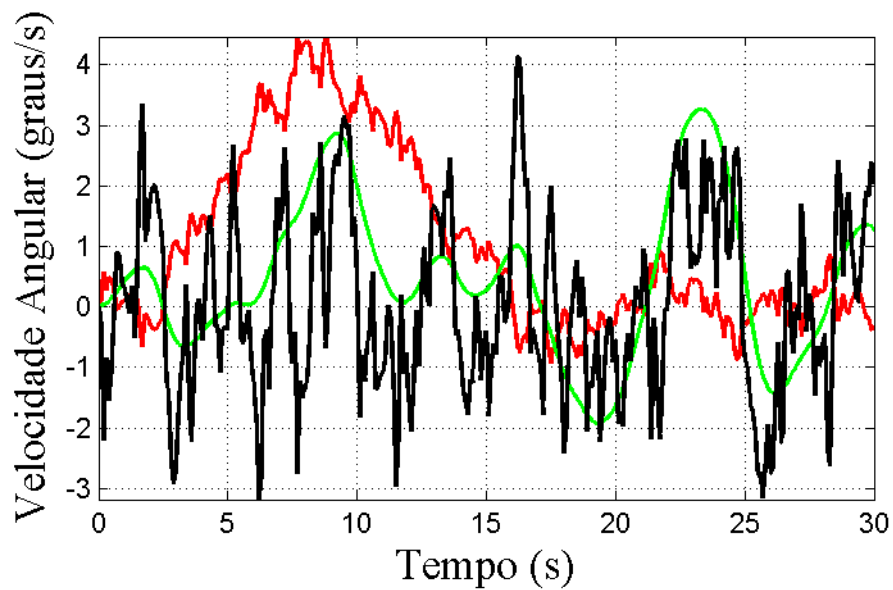


Figura 32 Velocidades angulares de arfagem, guinada e rolagem

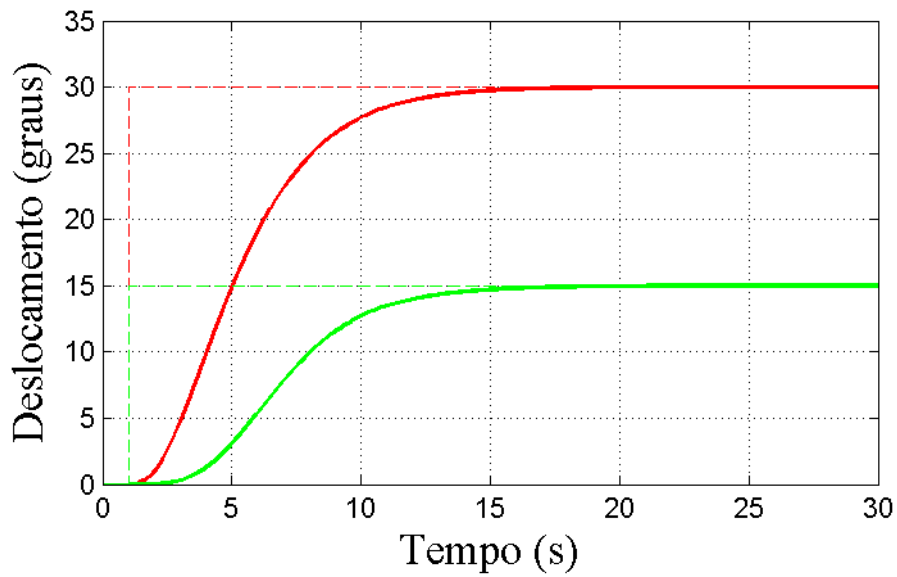


Figura 34 Resposta a uma entrada degrau, de uma planta com observador de estados

O mesmo procedimento adotado no sistema de controle seguidor sem o observador de estados, foi utilizado novamente, introduz-se um ruído branco na ação de controle, como forma de uma perturbação externa, para analisar o comportamento do controle (Figura 35)

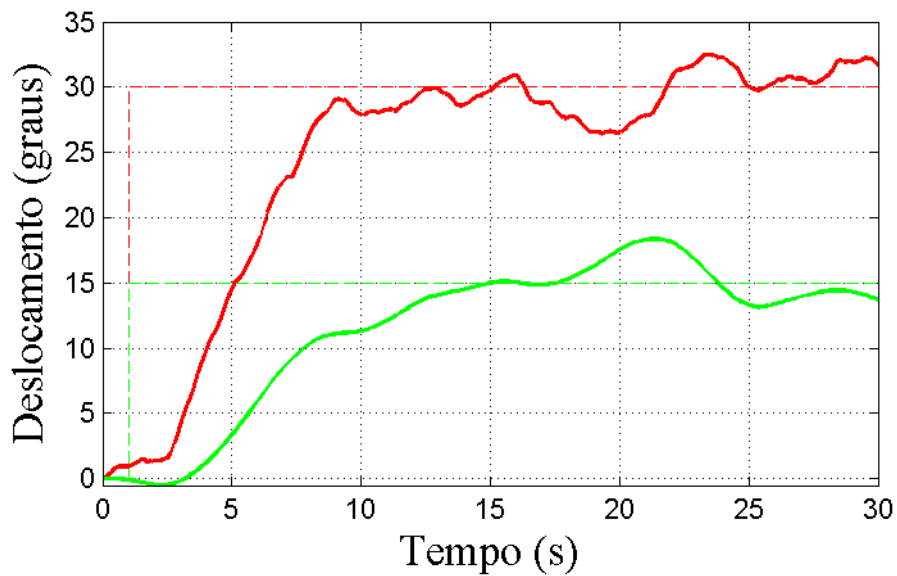


Figura 35 Resposta a uma entrada degrau com ruído branco, aplicado na ação de controle

Uma entrada rampa também foi testada na planta com o observador de estados, para verificar a capacidade do controle em relação a variação do ângulo de arfagem e guinada (Figura 36).

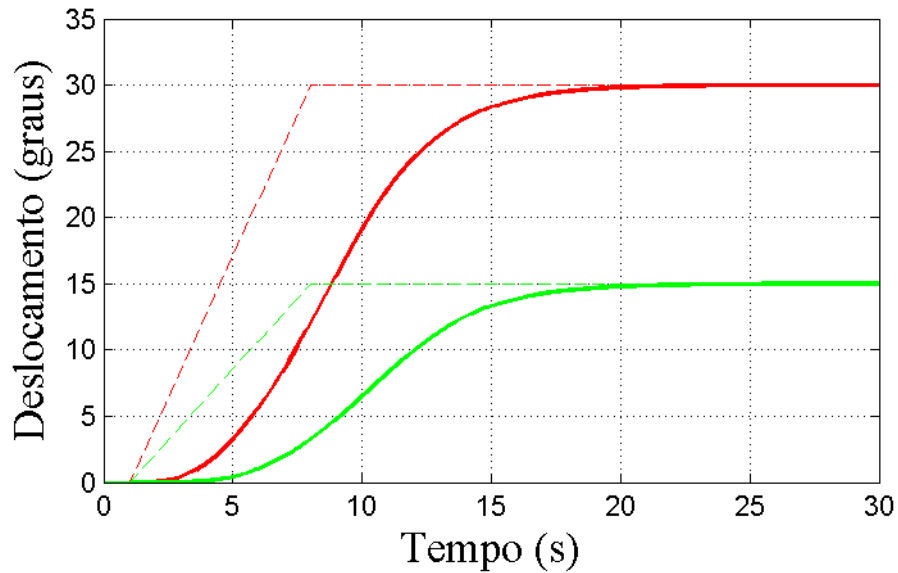


Figura 36 Resposta a uma entrada rampa, de uma planta com observador de estados

O ruído branco também é utilizado na ação de controle, para uma entrada rampa, com o objetivo de testar a eficiência do controlador (Figura 37)

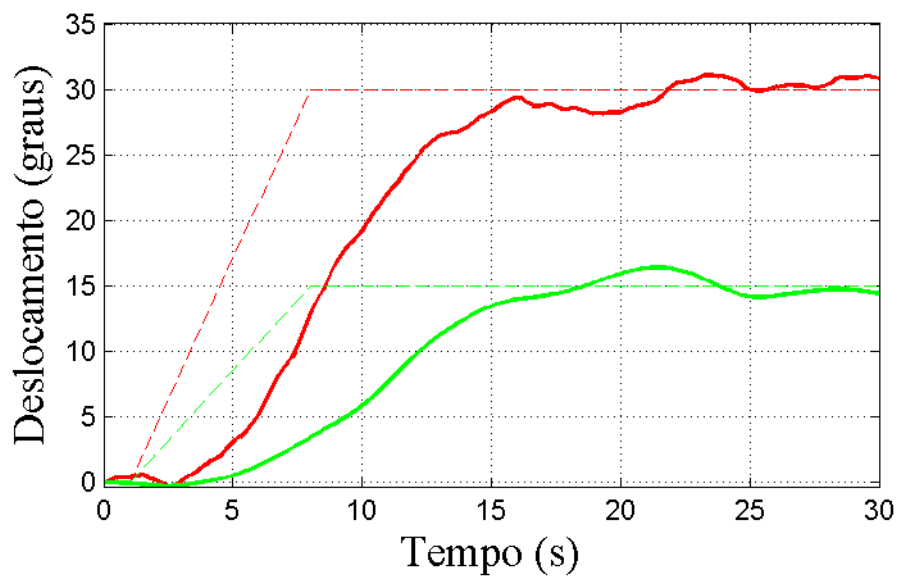


Figura 37 Resposta a uma entrada rampa, com ruído, de uma planta com observador

Para validar os resultados obtidos com as simulações, e verificar se o modelo matemático gerado através do *ADAMS*[®] é satisfatório, os testes realizados neste capítulo serão aplicados em uma planta experimental real e comparados com as simulações.

6.2 Validação do Modelo e Resultados Experimentais

Para a realização do experimento, foram utilizados os seguintes equipamentos:

- Um microcomputador com sistema operacional Windows,
- Uma placa controladora para controle de prototipagem rápida dSPACE 1103, que possui boa capacidade de processamento, vitais para aplicações que envolvem inúmeros sensores e atuadores. A dSPACE também trabalha com uma interface em tempo real, onde o controlador é totalmente programável em diagramas de blocos no *SIMULINK*[®].
- Foi construído um módulo de potência dos motores, utilizado para receber a variação do PWM e transformá-lo em tensão para cada um dos motores (Figura 38).

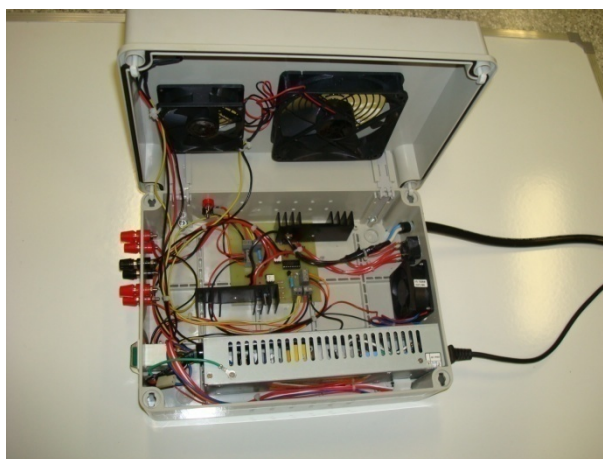


Figura 38 Módulo de Potência dos Motores

- Um protótipo (planta piloto) de um sistema de hélices paralelas, construído para a validação do modelo matemático gerado pelo ADAMS[®] (Figura 39).



Figura 39 Esquema de aquisição de dados

O programa utilizado para realizar o controle seguidor com observador de estados, em tempo real do sistema de hélices paralelas é criado através de diagramas de blocos, via *SIMULINK*[®] e compilado para a placa de aquisição dSPACE (Figura 40). O posicionamento do sistema é corrigido através da resposta dos sensores (encoders), através da atuação dos motores 1 e 2.

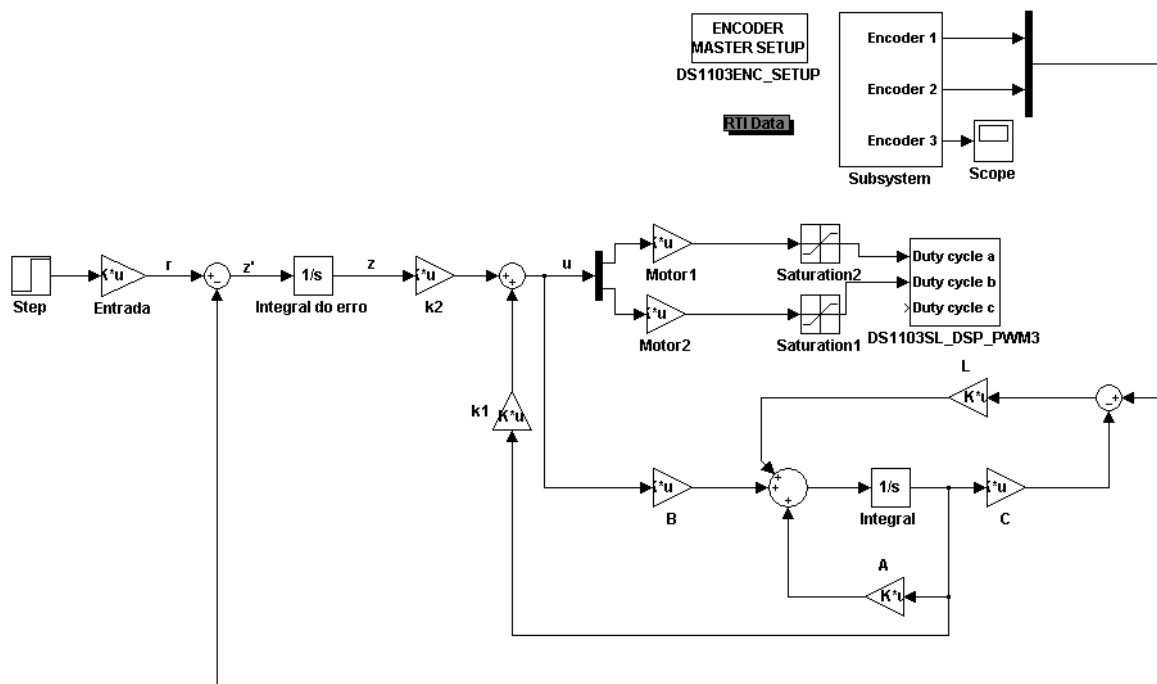


Figura 40 Diagrama de Blocos do sistema de controle aplicado à planta real

Alguns experimentos foram feitos, para testar a capacidade de resposta do controle seguidor na planta de hélices paralelas real. Vale lembrar que o efeito solo das hélices é desprezível em todos os experimentos.

O sistema tem como limitação o ângulo de guinada dependente do ângulo de arfagem, uma vez que não será possível guinar o sistema de hélices paralelas sem antes ter dado o comando de arfagem.

- Experimento 1. O primeiro experimento foi realizado variando o contrapeso localizado na extremidade traseira do sistema de hélices paralelas, simulando uma perturbação externa no sistema. Utilizaram-se duas posições diferentes para comparar as respostas, sendo essas [100 mm; 50 mm] respectivamente. Uma característica do sistema, é que ele apresenta uma resposta lenta, isso ocorre devido a algumas limitações físicas do modelo.

Os resultados da variação do ângulo de arfagem foram obtidos a partir de uma entrada degrau de 30° , onde foi contado um tempo de 100 segundos, e posteriormente foi aplicado um novo comando de entrada de -15° , também com um tempo de 100 segundos, e em seguida, o sistema retornou ao ponto de partida (0°). Os resultados de resposta do ângulo de arfagem foram satisfatórios, mostrando que o controle seguidor consegue atuar bem em condições adversas, em relação à variação desse ângulo (Figura 41).

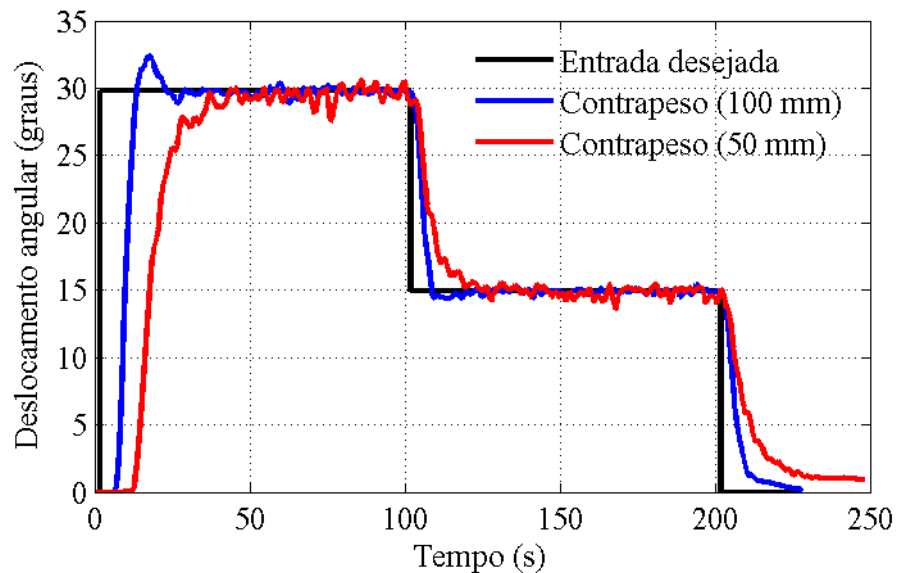


Figura 41 Resposta do ângulo de arfagem referente à variação do contra peso traseiro.

O ângulo de guinada permaneceu em 0° , porém pode-se observar que o sistema não teve uma boa precisão, apesar de tentar corrigir o erro, o sistema oscilou muito em torno do ponto desejado (Figura 42). Uma justificativa para que essas oscilações aconteçam é o fato de que se está supondo que o fluxo aerodinâmico gerado pelas hélices tem um comportamento uniforme, o que não é garantido. A outra simplificação que se fez, e que pode contribuir para essas oscilações, foi considerar que a variação

da força de empuxo é instantânea para uma variação de tensão de alimentação dos motores.

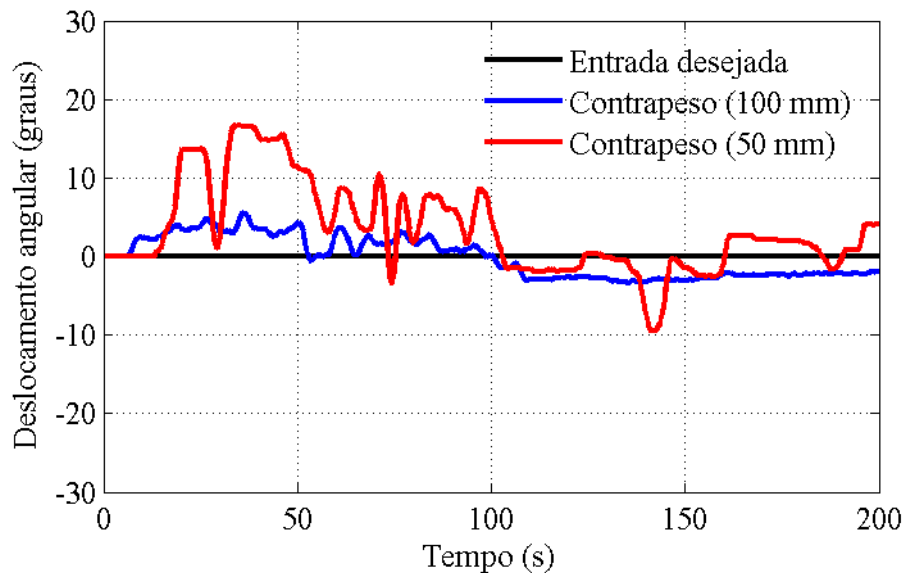


Figura 42 Resposta do ângulo de guinada referente à variação do contra peso traseiro.

- Experimento 2. A planta continua na mesma condição anteriormente citada, onde somente o contrapeso traseiro foi alterado para as posições [100 mm; 50 mm], foi dado apenas uma entrada degrau para o ângulo de arfagem, onde foi contado um tempo de 100 segundos e em seguida foi dada uma entrada degrau para o ângulo de guinada de 15° , mais uma vez é possível analisar que a saída conseguiu seguir o comando de entrada do sistema de hélices paralelas, apesar de apresentar oscilações. Essas respostas foram comparadas com a resposta obtida através das simulações feitas em *MATLAB*[®].

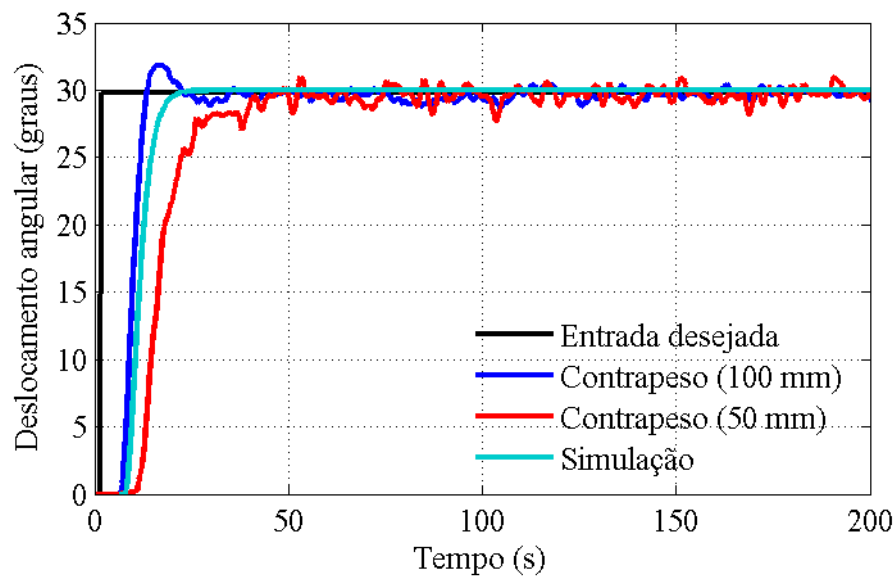


Figura 43 Resposta a uma entrada degrau para o ângulo de arfagem.

É possível observar na Figura 44 que o ângulo de guinada tem uma tendência a divergir, e possui uma dinâmica lenta, o erro é devido aos autovalores associados ao sistema de controle seguidor. No entanto, mesmo assim o controlador busca corrigir o erro a todo o momento. Outro detalhe é devido ao fato de que as hélices dos motores possuem o mesmo sentido de rotação, fazendo com que apareça uma força de Coriolis e o efeito giroscópico. Os autovalores atribuídos ao sistema de malha fechada e utilizados em todos os experimentos são:

$$\sigma(\bar{A} + \bar{B}\bar{K}) = \{-0.9 + 0.5i; -0.9 - 0.5i; -0.8 + 0.4i; -0.8 - 0.4i; -0.05 + 0.03i; -0.05 - 0.03i; -0.5; -0.5\}$$

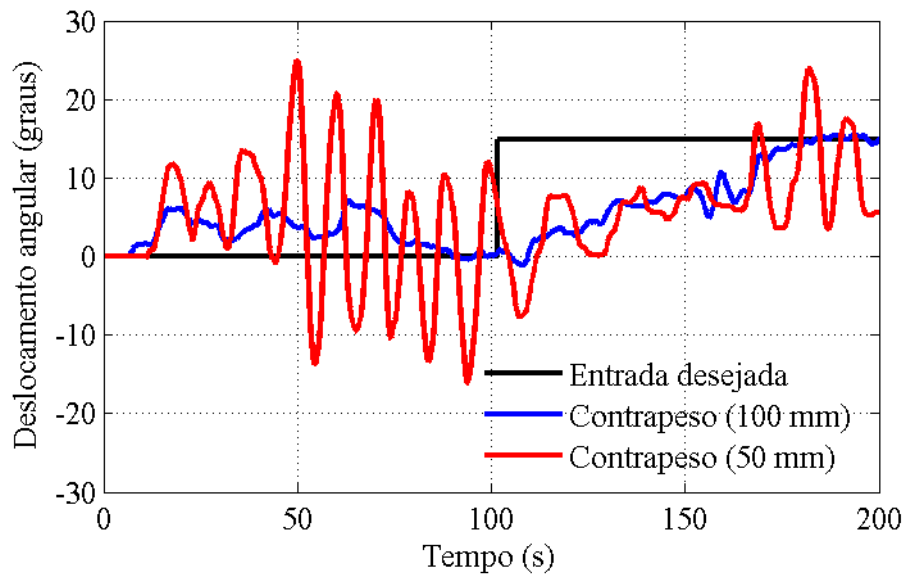


Figura 44 Resposta a uma entrada degrau para o ângulo de guinada

- Experimento 3. Esse experimento foi feito, deixando o contrapeso traseiro na posição 100 mm e alteraram-se os contrapesos laterais, recuando-os do curso máximo para seu curso mínimo, simulando outro tipo de perturbação do sistema. O resultado obtido em relação o ângulo de arfagem é mostrado na Figura 45.

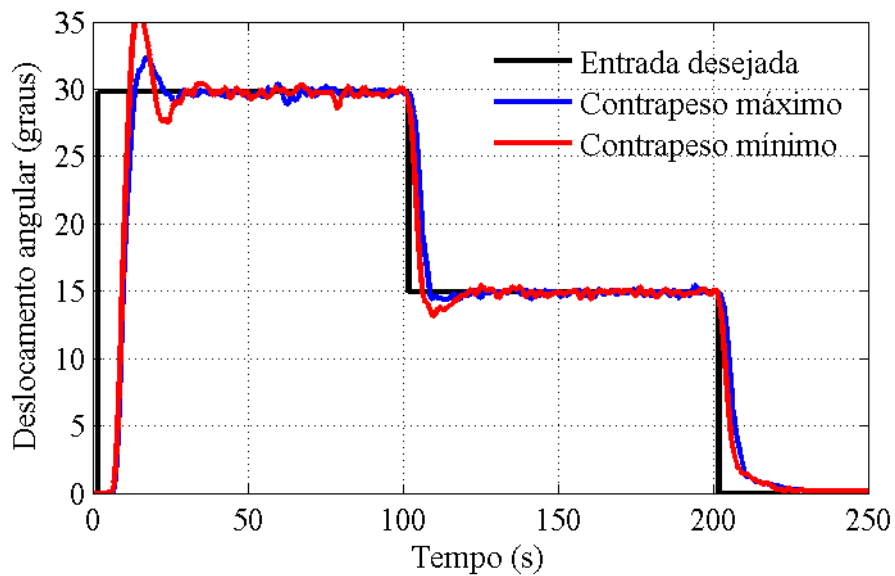


Figura 45 Resposta do ângulo de arfagem referente à variação do contra peso lateral.

O ângulo de guinada desejado permaneceu em 0° , nota-se que depois de 100 segundos, ocorreu à minimização do erro, e o sistema conseguiu estabilizar em torno do ângulo desejado, com isso é possível observar que a variação do contrapeso lateral melhora significativamente as não linearidades do ângulo de guinada.

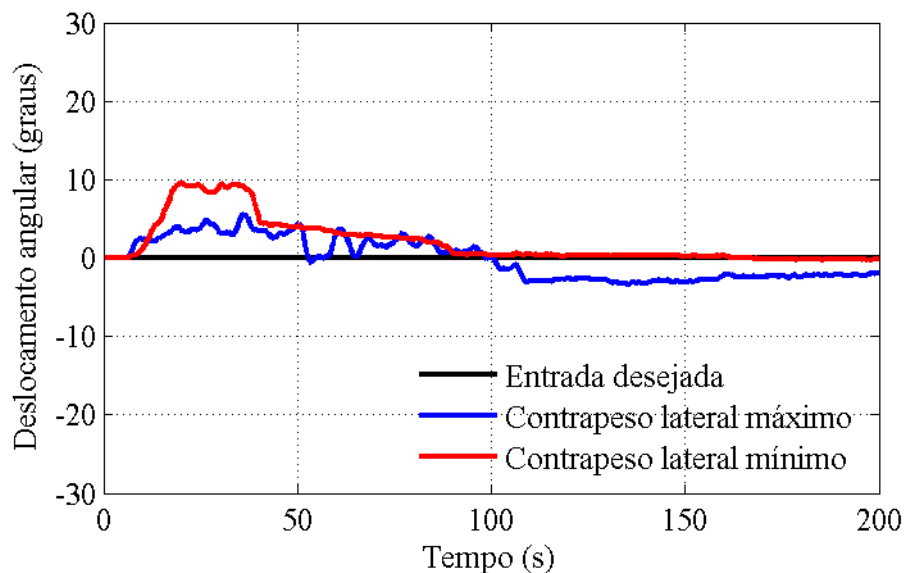


Figura 46 Resposta do ângulo de guinada referente à variação do contra peso lateral.

- Experimento 4. Por fim são aplicados três ganhos diferentes nos motores, [0,05; 0,1; 0,15], para analisar o comportamento do sistema de hélices paralelas em relação ao tempo de resposta do sistema. É possível notar que quanto maior o ganho aplicado nos motores, maior será o sobre sinal do sistema. Isto se explica na medida em que o ganho (proporcional) do sistema como um todo aumenta e certamente tende a desestabilizar o sistema como seria de esperar. Neste caso o contrapeso traseiro também estava na posição 100 milímetros, e a entrada foi de 30° com decaimento de 15° após 100 segundos. O controlador novamente conseguiu seguir a entrada requerida para o ângulo de arfagem do sistema (Figura 47).

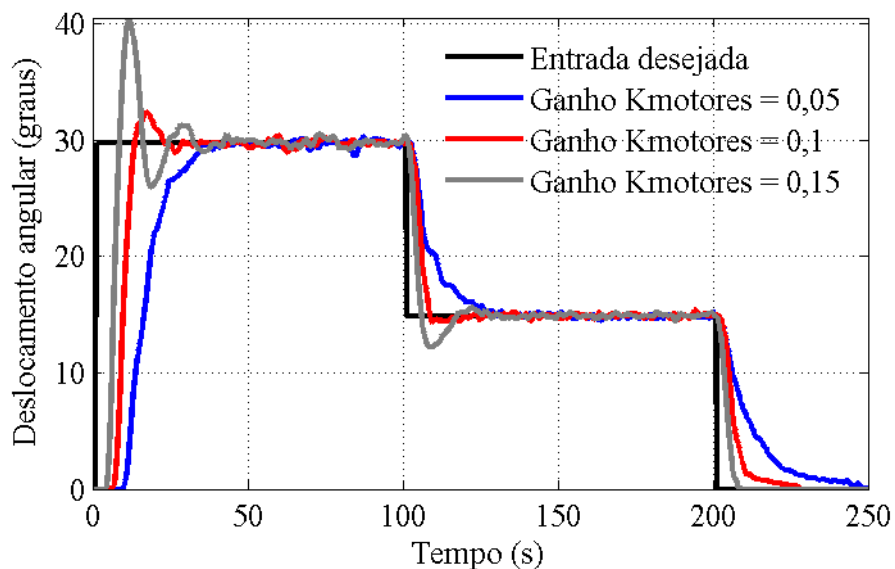


Figura 47 Resposta do ângulo de arfagem referente aos diferentes ganhos dos motores

Os ângulos de guinada também permaneceram na posição 0° conforme a entrada requerida, mostrando que mais uma vez o controle oscilou em torno desse ponto (Figura 48).

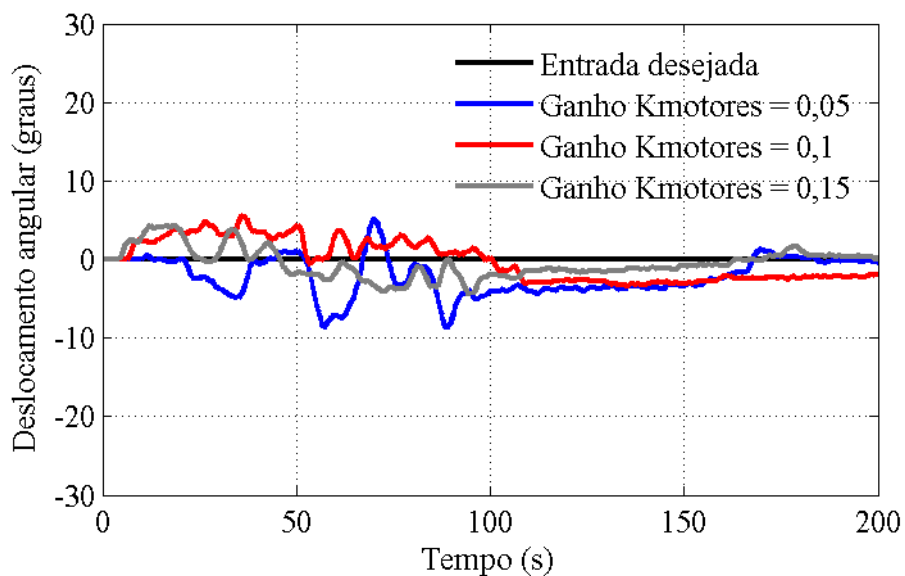


Figura 48 Resposta do ângulo de guinada referente aos diferentes ganhos dos motores

Capítulo 7

CONCLUSÕES E PERSPECTIVAS FUTURAS

Neste trabalho realizou-se um estudo sobre modelagem de um sistema de hélices paralelas, utilizando o pacote de modelagem dinâmica *ADAMS*[®], através de uma interface gráfica de fácil visualização, permitindo maior flexibilidade e rapidez na obtenção do modelo dinâmico da planta.

Conclui-se através das simulações que o sistema seguidor utilizado é eficiente para controlar os movimentos de arfagem e guinada apresentados neste trabalho.

Embora este processo de modelagem e simulação tenha sido aplicado a um sistema de hélices paralelas com o intuito de estudar o comportamento deste sistema, este método pode ser aplicado para modelar e simular outros tipos de sistemas complexos com muitos graus de liberdade de forma rápida e com bons resultados.

Os resultados obtidos experimentalmente para o ângulo de arfagem foram satisfatórios, com valores próximos dos resultados das simulações, isso mostra que o controlador consegue atuar bem em diferentes condições de trabalho e que o modelo linear consegue controlar uma planta não linear, porém o ângulo de guinada foi deficiente, apresentando oscilações em torno do ponto desejado. Isto certamente deve-se a algumas simplificações já comentadas no texto tais como considerar fluxo gerado pelas hélices uniforme e resposta instantânea de força de empuxo a entrada de tensão nos motores.

Como perspectiva de trabalhos futuros, tem-se a aplicação de outras técnicas de controle, com a finalidade de compará-las com o sistema de controle estudado.

BIBLIOGRAFIA

ALBUQUERQUE, A.R..L. *Aplicações de Hardware-in-the-loop no desenvolvimento de uma mão robótica* 173p Tese (Doutorado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos,2007

BARBIERI, F. A. A. *Prototipagem virtual: modelagem, simulação, controle e otimização de dinâmica veicular*. 194 p. Tese (Doutorado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos,2002.

CAMPO, A.B., “*Project and Simulation of a Digital Controller for an Aero-Stabilizing System*”, Integração, São Paulo, Vol. 48, pp 61-65, 2007.

CAMPO, A. B., “*Development of Pedagogical Prototypes for the Study of Digital Control Theory*”. Integração, São Paulo, Vol. 43, pp. 347-351, 2005.

COSTA NETO, A *Application of Multibody Systems (MBS) Modelling Techniques to Automotive Vehicle Chassis Simulation for Motion Control Studies* Tese (Doutorado) University of Warwick, WARWICK, Inglaterra, 1992.

CHANCE, M. C. *Modeling of dynamic mechanical system*. In: CAD/CAM ROBOTICS AND AUTOMATIONS INSTITUTE AND INTERNATIONAL CONFERENCE, 1985, Tucson. Proceedings... Arizona. February,1985.

D’AZZO,J.J., and HOUPIS, H.C., “*Linear Control System Analysis and Design: Conventional and Modern*, New York, McGraw Hill, Inc, 1995.

HALSEY, K.M., and GLOVER, K., “*Analysis and Synthesis of Nested Feedback Systems*”, IEEE Transactions on Automatic Control, Vol. 50, pp. 984-996, 2005.

HUSTON, R.L.; KAMMAN,J.W., *Testing the accuracy of numerical simulations of multibody systems* Mechanics Research Communications, Volume 28, Issue 1, January-February 2000, Pages 1-6

LOPES, R.V., GALVÃO, R.K.H., MILHAN, A.P., BECERRA, V.M. and YONEYAMA, T., “*Modelling and Constrained Predictive Control of a 3DOF Helicopter*”, XVI Brazilian Automation Conference, Salvador, Brazil, pp.429-434, 2006.

MONTEZUMA, M. A. F., CAPORALI, A. S., and BELO, E. M., “*Modeling and position and orientation control of a Stewart platform*”. In: XIV Brazilian Automation Conference, Natal –RN, Brazil, 2002.

MONTEZUMA, M.A.F. . *Modelagem e controle de posição e orientação de uma plataforma de stewart*. 124p. Dissertação (Mestrado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003.

OGATA, K. *Engenharia de controle moderno*. 4ª Ed. São Paulo, Prentice-Hall do Brasil, 2003.

ORLANDEA, N., CHACE, M. A., AND CALAHAN, D. A., "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Part 1," *Journal of Engineering for Industry*, pp. 773-779, 1977

SILVA, V. R. V. *Análise estrutural de uma bicicleta através de simulação conjunta de técnicas de multicorpos e elementos finitos*. São Carlos. 137p. Tese (Doutorado) – Escola de Engenharia de São Carlos. Universidade de São Paulo, 1999.

SOHONI, V. N., WHITESELL, J. *Automatic linearization of constrained dynamical models* Transactions of the ASME. Journal of Mechanisms, Transmissions, and Automations in Design, v.108, n3, p.300-304, September, 1986.

WITT, J., BOONTO, S., and WERNER, H., “*Approximate Model Predictive Control of a 3-DOF Helicopter*”, Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA, pp.4501-4506, 2007.

YU, Y., ZHONG, Y., *Robust attitude control of a 3dof helicopter with multi operation points*, Springer Science + Business Media, LLC, 2009.

Apêndice I

```
% Programa Desenvolvido para Determinação dos Ganhos K1 e K2 de um Sistema
Seguidor Usando Realimentação de Estados
```

```
% % Matrizes A, B, C, D referentes à dinâmica do sistema de hélices
% % paralelas.
```

```
a = load('A');
b = load('B');
c = load('C');
d = load('D');
```

```
% Condição de controlabilidade onde posto de Mcc = número de estados +
% número de entradas
```

```
Mcc=[b a;d(1:2,:) -c(1:2,:)];
r=rank(Mcc);
```

```
% Montagem das matrizes Abarra e Bbarra
```

```
e=zeros(6,2);
abar=[a e;-c(1:2,:) d(1:2,:)];
bbar=[b;d(1:2,:)];
```

```
% Conjunto de autovalores a serem atribuídos ao sistema
```

```
l_1=-0.9+0.5i;
l_2=-0.9-0.5i;
l_3=-0.8+0.4i;
l_4=-0.8-0.4i;
l_5=-0.05+0.03i;
l_6=-0.05-0.03i;
l_7=-0.5;
l_8=-0.5;
```

```
s1=[abar-l_1*eye(8) bbar];
s2=[abar-l_2*eye(8) bbar];
s3=[abar-l_3*eye(8) bbar];
s4=[abar-l_4*eye(8) bbar];
s5=[abar-l_5*eye(8) bbar];
s6=[abar-l_6*eye(8) bbar];
```

```
s7=[abar-l_7*eye(8) bbar];  
s8=[abar-l_8*eye(8) bbar];
```

```
r1=rref(s1);  
r2=rref(s2);  
r3=rref(s3);  
r4=rref(s4);  
r5=rref(s5);  
r6=rref(s6);  
r7=rref(s7);  
r8=rref(s8);
```

```
a1=[r1;zeros(2,10)];  
a2=[r2;zeros(2,10)];  
a3=[r3;zeros(2,10)];  
a4=[r4;zeros(2,10)];  
a5=[r5;zeros(2,10)];  
a6=[r6;zeros(2,10)];  
a7=[r7;zeros(2,10)];  
a8=[r8;zeros(2,10)];
```

```
for i=9:10
```

```
    a1(i,i)=-1;  
    a2(i,i)=-1;  
    a3(i,i)=-1;  
    a4(i,i)=-1;  
    a5(i,i)=-1;  
    a6(i,i)=-1;  
    a7(i,i)=-1;  
    a8(i,i)=-1;
```

```
end
```

```
v(:,1)=a1(1:8,9)/(a1(1,9));  
v(:,2)=a2(1:8,10)/(a2(1,10));  
v(:,3)=a3(1:8,9)/(a3(1,9));  
v(:,4)=a4(1:8,10)/(a4(1,10));  
v(:,5)=a5(1:8,9)/(a5(1,9));  
v(:,6)=a6(1:8,10)/(a6(1,10));  
v(:,7)=a7(1:8,9)/(a7(1,9));  
v(:,8)=a8(1:8,10)/(a8(1,10));
```

```
q(:,1)=a1(9:10,9)/(a1(1,9));
q(:,2)=a2(9:10,10)/(a2(1,10));
q(:,3)=a3(9:10,9)/(a3(1,9));
q(:,4)=a4(9:10,10)/(a4(1,10));
q(:,5)=a5(9:10,9)/(a5(1,9));
q(:,6)=a6(9:10,10)/(a6(1,10));
q(:,7)=a7(9:10,9)/(a7(1,9));
q(:,8)=a8(9:10,10)/(a8(1,10));
```

```
% Matrizes de Ganhos k1 e k2
```

```
cc=inv(c);
kbar=q*inv(v);
c_e=c(1:2,:);
c_f=c(3:6,:);
k1=real(kbar(:,1:6)); % Ganho proporcional aos estados
k2=real(kbar(:,7:8)); % Ganho proporcional a integral do erro
```

```
acl=[abar+bbar*kbar];
bcl=[zeros(6,2);eye(2)];
ccl=[c zeros(6,2)];
dcl=zeros(6,2);
```

```
% Observador de estados
```

```
% Calculo da matriz de ganho L
```

```
x=-25;
y=-26;
z=-27;
w=-28;
```

```
p = [x z w x y w];
L = place(a',c',p);
L1=L';
```

```
% Programa desenvolvido para plotar os gráficos do sistema de hélices
% paralelas

close all;
n=1;

%Deslocamento
figure(1);
plot(pitch.time(1:n:size(pitch.time,1)), ...
     pitch.signals.values(1:n:size(pitch.signals.values,1)),...
     'LineWidth',3,'Color',[1 0 0]);

hold on;
plot(yaw.time(1:n:size(yaw.time,1)), ...
     yaw.signals.values(1:n:size(yaw.signals.values,1)),...
     'LineWidth', 3,'Color',[0 0 1]);
plot(roll.time(1:n:size(roll.time,1)), ...
     roll.signals.values(1:n:size(roll.signals.values,1)),...
     'LineWidth', 3,'Color',[0 0 0]);
plot(pitch1.time(1:n:size(pitch1.time,1)), ...
     pitch1.signals.values(1:n:size(pitch1.signals.values,1)),...
     '--','Color',[1 0 0])

plot(yawl.time(1:n:size(yawl.time,1)), ...
     yawl.signals.values(1:n:size(yawl.signals.values,1)),...
     '--','Color',[0 0 1]);

xlabel('Tempo (s)');
ylabel('Deslocamento (graus)');

set(get(gca,'xlabel'),'FontName','Times New Roman');
set(get(gca,'xlabel'),'FontSize',16);
set(get(gca,'ylabel'),'FontName','Times New Roman');
set(get(gca,'ylabel'),'FontSize',16);
set(gcf,'Position',[200 200 500 300]); % Tamanho da figura
set(gcf,'PaperPositionMode','auto');
set(gca,'Box','on');

axis([0 20 -inf 35]); % Escala dos eixos
grid on; % Grid do gráfico
```

```
set(gcf, 'Color', [1 1 1]); % Cor do fundo do gráfico

% Configurações de legenda
lh = legend('Elevação - controlável', 'Deslocamento - controlável',
'Rolagem - não controlável'); % Entradas da legenda

set(lh, 'Location', 'EastOutside'); % Posição da legenda
set(lh, 'Box', 'off'); % Borda da legenda
set(lh, 'FontName', 'Times New Roman');
set(lh, 'FontSize', 10);

% Salva a figura em arquivo
print('-dpng', 'Resposta');

% Velocidades Angulares
figure(2);
plot(vel_ang_pitch.time(1:n:size(vel_ang_pitch.time,1)), ...

vel_ang_pitch.signals.values(1:n:size(vel_ang_pitch.signals.values,1)), ...
    'LineWidth', 3, 'Color', [1 0 0]);

hold on;
plot(vel_ang_yaw.time(1:n:size(vel_ang_yaw.time,1)), ...
    vel_ang_yaw.signals.values(1:n:size(vel_ang_yaw.signals.values,1)), ...
    'LineWidth', 3, 'Color', [0 0 1]);
plot(vel_ang_roll.time(1:n:size(vel_ang_roll.time,1)), ...

vel_ang_roll.signals.values(1:n:size(vel_ang_roll.signals.values,1)), ...
    'LineWidth', 3, 'Color', [0 0 0]);

xlabel('Tempo (s)');
ylabel('Velocidade Angular (graus/s)');

set(get(gca, 'xlabel'), 'FontName', 'Times New Roman');
set(get(gca, 'xlabel'), 'FontSize', 16);
set(get(gca, 'ylabel'), 'FontName', 'Times New Roman');
set(get(gca, 'ylabel'), 'FontSize', 16);
set(gcf, 'Position', [200 200 500 300]); % Tamanho da figura
set(gcf, 'PaperPositionMode', 'auto');
set(gca, 'Box', 'on');
```

```
axis([0 20 -inf inf]); % Escala dos eixos
grid on; % Grid do gráfico

set(gcf, 'Color', [1 1 1]); % Cor do fundo do gráfico

% Configurações de legenda
lh = legend('Elevação - controlável', 'Deslocamento - controlável',
'Rolagem - não controlável'); % Entradas da legenda

set(lh, 'Location', 'EastOutside'); % Posição da legenda
set(lh, 'Box', 'off'); % Borda da legenda
set(lh, 'FontName', 'Times New Roman');
set(lh, 'FontSize', 10);

% Salva a figura em arquivo
print('-dpng', 'velocidades');
```

```
% Programa desenvolvido para plotar os gráficos obtidos na aquisição de
% dados.
close all;
clear all;

% Experimento 1
arq1 = open('experimento5.mat');
data(1) = arq1.experimento5;

x_data1 = data(1).X.Data;
y1_data1 = data(1).Y(1).Data;
y2_data1 = data(1).Y(2).Data;
y3_data1 = data(1).Y(3).Data;
y4_data1 = data(1).Y(4).Data;

% Experimento 5
arq2 = open('experimento1.mat');
data(2) = arq2.experimento1;

x_data2 = data(2).X.Data;
y1_data2 = data(2).Y(1).Data;
y2_data2 = data(2).Y(2).Data;
y3_data2 = data(2).Y(3).Data;
y4_data2 = data(2).Y(4).Data;

% Experimento 7
arq3 = open('experimento7.mat');
data(3) = arq3.experimento7;

x_data3 = data(3).X.Data;
y1_data3 = data(3).Y(1).Data;
y2_data3 = data(3).Y(2).Data;
y3_data3 = data(3).Y(3).Data;
y4_data3 = data(3).Y(4).Data;

% figure(1);
plot(x_data1,(180/pi)*y1_data1,'Color',[0 0 0],'Linewidth',2);
hold on;
plot(x_data1,(180/pi)*y3_data1,'Color',[0 0 1],'Linewidth',2);
plot(x_data2,(180/pi)*y3_data2,'Color',[1 0 0],'Linewidth',2);
```



```
plot(x_data3,(180/pi)*y3_data3,'Color',[0.5 0.5 0.5],'Linewidth',2);
lh = legend('Entrada desejada','Ganho Km = 0,05','Ganho Km = 0,1','Ganho Km
= 0,15');

xlabel('Tempo (s)');
ylabel('Deslocamento angular (graus)');

set(gca,'FontSize',12,'FontName','Times New Roman');

set(get(gca,'xlabel'),'FontName','Times New Roman');
set(get(gca,'xlabel'),'FontSize',12);
set(get(gca,'ylabel'),'FontName','Times New Roman');
set(get(gca,'ylabel'),'FontSize',12);
set(gcf,'Position',[200 200 500 300]); % Tamanho da figura
set(gcf,'PaperPositionMode','auto');
set(gca,'Box','on');

axis([0 250 -inf inf]); % Escala dos eixos
grid on; % Grid do gráfico

set(gcf,'Color',[1 1 1]); % Cor do fundo do gráfico

%set(lh,'Location','EastOutside'); % Posição da legenda
set(lh,'Box','off'); % Borda da legenda
set(lh,'FontName','Times New Roman');
set(lh,'FontSize',12);

% % Salva a figura em arquivo
print('-dpng','pitch_comp_3ganhos');

% figure(2);

plot(x_data2,(180/pi)*y2_data2,'Color',[0 0 0],'Linewidth',2);
hold on;
plot(x_data1,(180/pi)*y4_data1,'Color',[0 0 1],'Linewidth',2);
plot(x_data2,(180/pi)*y4_data2,'Color',[1 0 0],'Linewidth',2);
plot(x_data3,(180/pi)*y4_data3,'Color',[0.5 0.5 0.5],'Linewidth',2);
lh = legend('Entrada desejada','Ganho Km = 0,05','Ganho Km = 0,1','Ganho Km
= 0,15');
```

```
xlabel('Tempo (s)');
ylabel('Deslocamento angular (graus)');

set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');

set(get(gca, 'xlabel'), 'FontName', 'Times New Roman');
set(get(gca, 'xlabel'), 'FontSize', 12);
set(get(gca, 'ylabel'), 'FontName', 'Times New Roman');
set(get(gca, 'ylabel'), 'FontSize', 12);
set(gcf, 'Position', [200 200 500 300]); % Tamanho da figura
set(gcf, 'PaperPositionMode', 'auto');
set(gca, 'Box', 'on');

axis([0 200 -10 10]); % Escala dos eixos
grid on; % Grid do gráfico

set(gcf, 'Color', [1 1 1]); % Cor do fundo do gráfico

%set(lh, 'Location', 'EastOutside'); % Posição da legenda
set(lh, 'Box', 'off'); % Borda da legenda
set(lh, 'FontName', 'Times New Roman');
set(lh, 'FontSize', 12);

% % Salva a figura em arquivo
print('-dpng', 'pitch_comp_3ganhos');
```