

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DOUGLAS ANTUNES TESCH

**EXTENSÃO DO ITERATIVE FEEDBACK
TUNING PARA SISTEMAS EM
CASCATA COM APLICAÇÃO EM
CONTROLE DE QUADRICÓPTEROS**

Porto Alegre
2016

DOUGLAS ANTUNES TESCH

**EXTENSÃO DO ITERATIVE FEEDBACK
TUNING PARA SISTEMAS EM
CASCATA COM APLICAÇÃO EM
CONTROLE DE QUADRICÓPTEROS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Diego Eckhard

CO-ORIENTADOR: Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre
2016

DOUGLAS ANTUNES TESCH

**EXTENSÃO DO ITERATIVE FEEDBACK
TUNING PARA SISTEMAS EM
CASCATA COM APLICAÇÃO EM
CONTROLE DE QUADRICÓPTEROS**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Diego Eckhard, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul –
Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Aurelio Tergolina Salton, PUCRS
Doutor pela Universidade de Newcastle – Newcastle, Austrália

Prof. Dr. Guilherme Araujo Pimentel, PUCRS
Doutor pela Université de Mons – Mons, Bélgica

Prof. Dr. Jeferson Vieira Flores, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: _____
Prof. Dr. Luis Fernando Alves Pereira

Porto Alegre, março de 2016.

AGRADECIMENTOS

Gostaria de começar agradecendo a Nathalia, pela dedicação, compreensão e apoio nesta jornada.

Aos meus pais que dedicaram partes importantes de suas vidas para que hoje eu esteja aqui.

Aos meus irmãos Deivid e Carla que sempre me apoiam.

Aos amigos Israel, Raphael e Tamara pelas horas de trabalho e descontração.

Aos bolsistas Gabriel e William pela ajuda nos experimentos.

Ao professor Bazanella pela coorientação e ajuda.

Ao professor Diego Eckhard pela impecável orientação durante o período de mestrado, pela amizade e perseverança.

Ao programa PPGEE e ao CNPq pela oportunidade e concessão da bolsa.

RESUMO

Os métodos de controle baseados em dados têm como diferencial utilizar os dados coletados de experimentos para realizar a sintonia do controlador de uma forma direta e sem utilizar nenhum modelo do processo. Um dos métodos já consagrados é o conhecido *Iterative Feedback Tuning*, método este caracterizado como um algoritmo iterativo de otimização que busca a minimização do critério de desempenho norma \mathcal{H}_2 . O presente trabalho propõe uma extensão baseada no clássico algoritmo *Iterative Feedback Tuning* para sistemas em formato cascata com duas malhas de controle. O formato cascata com duas malhas de controle é uma estrutura comum e muito utilizada em aplicações de cunho industrial. A sintonia deste tipo de estrutura em metodologias clássicas de controle envolve geralmente duas etapas. Primeiramente a malha interna é sintonizada para então posteriormente a malha externa. A vantagem do método de controle baseado em dados proposto está em realizar a sintonia de ambas as malhas de forma simultânea e concomitantemente. É escolhida como aplicação um veículo aéreo não tripulado com quatro rotores, mais conhecido como quadricóptero. Esta aplicação tem uma dinâmica complexa, uma vez que o sistema é classificado como sendo não-linear e multivariável, algo demonstrado através do equacionamento dinâmico do quadricóptero que é apresentado. Uma série de simulações e experimentos práticos são apresentadas com o intuito de demonstrar a dinâmica descrita no modelo e também demonstrar a aplicação do algoritmo *Cascade Iterative Feedback Tuning* para os mesmos.

Palavras-chave: Controle baseado em dados, Critério \mathcal{H}_2 , Quadricóptero, Controle cascata.

ABSTRACT

The methods of data-driven control have the distinction to use the data collected from experiments to perform the tuning of the controller directly and without using any process model. One method already established is the Iterative Feedback Tuning, this method is characterized as an iterative optimization algorithm that aims to minimize the \mathcal{H}_2 norm performance criterion. The present work proposes an extension based on the classical Iterative Feedback Tuning algorithm for cascade systems with two control loops. The cascade format with two loops is a common structure and widely used in industrial applications. In classical control methods, the tuning of this type of structure usually involves two steps. First, the inner loop is tuned and subsequently the outer loop. The advantage of the proposed data-driven control method is to perform this tuning of both controller simultaneously and concurrently. We choose as an application a unmanned aerial vehicle (UAV) with four rotors, better known as quadcopter. This application has a complex dynamic, since the system is classified as being non-linear and multivariable, demonstrated by the quadcopter dynamic equations. A series of simulations and practical experiments are presented to demonstrate the dynamic model described and also to validate the method CIFT algorithms.

Keywords: Electrical Engineering, Signal Processing, Automation and Control, Electronic and Instrumentation.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	11
LISTA DE ABREVIATURAS	12
LISTA DE SÍMBOLOS	13
1 INTRODUÇÃO	15
2 ITERATIVE FEEDBACK TUNING - IFT	18
2.1 Definições	19
2.1.1 Processo	19
2.1.2 Controle	19
2.1.3 Processo em malha fechada	20
2.2 Problema de otimização	20
2.2.1 Seguimento da resposta desejada	22
2.2.2 Rejeição a ruído	22
2.2.3 Esforço de controle	22
2.3 Algoritmo do gradiente	23
2.3.1 Convergência do algoritmo	24
2.4 Gradiente da função custo $J(\rho)$	25
2.5 Algoritmo IFT	28
2.6 Exemplo Numérico	29
2.7 Sintonia IFT em sistemas com integrador	30
3 CASCADE ITERATIVE FEEDBACK TUNING - CIFT	35
3.1 Definições	36
3.1.1 Processo em cascata	36
3.1.2 Estrutura do controlador	37
3.1.3 Processo em malha fechada	38
3.2 Problema de Otimização	38
3.3 Gradiente da função custo $J(\rho_1, \rho_2)$	39
3.3.1 Derivada parcial de $y_2(t)$ em função de ρ_1	39
3.3.2 Derivada parcial de $y_2(t)$ em função de ρ_2	41
3.4 Estimativa $\frac{\partial \hat{J}}{\partial \rho}$	43
3.5 Algoritmo CIFT	44
3.6 Exemplo numérico	45

3.6.1	Processo sem ruído	46
3.6.2	Processo com ruído	47
3.7	Comparativo entre os algoritmos IFT e CIFT	51
3.7.1	<i>Iterative Feedback Tuning</i>	53
3.7.2	<i>Cascade Iterative Feedback Tuning</i>	54
4	QUADRICÓPTERO	57
4.1	Cinemática	58
4.1.1	Matriz de rotação	58
4.2	Dinâmica	60
4.2.1	Estados de translação	60
4.2.2	Estados Angulares	62
4.2.3	Força e momento	64
4.3	Modelo completo	65
4.4	Simulação	66
4.4.1	Simulink	68
4.4.2	Resultados simulados do método <i>Cascade Iterative Feedback Tuning</i>	70
5	EXPERIMENTO PRÁTICO	74
5.1	Quadricóptero	74
5.1.1	Rotores	74
5.1.2	Placa controladora	78
5.1.3	Inertial Measurement Unit - IMU	78
5.1.4	Baterias	79
5.2	Plataforma de testes	79
5.3	Experimento	80
5.4	Resultados	82
6	CONCLUSÃO	87
	REFERÊNCIAS	88

LISTA DE ILUSTRAÇÕES

Figura 1:	Diagrama de blocos, sistema em malha fechada.	20
Figura 2:	Saída $y_0(t, \rho)$ do processo com controlador inicial (44).	30
Figura 3:	Saída $y_{20}(t)$ do processo com controlador obtido pelo algoritmo IFT.	31
Figura 4:	Convergência dos parâmetros do controlador.	31
Figura 5:	Resposta inicial, final e desejada para o sistema com integrador e sintonizado com IFT.	33
Figura 6:	Processo com controladores em formato cascata.	37
Figura 7:	Saída desejada $y_d(t)$ e saída inicial $y_2(t)$, caso sem ruído, com os controladores iniciais (102).	46
Figura 8:	Saída final $y_1(t)$ do processo sem ruído, depois de realizadas 20 iterações do algoritmo CIFT.	47
Figura 9:	Saída final $y_2(t)$ do processo sem ruído, depois de realizadas 20 iterações do algoritmo CIFT.	47
Figura 10:	Convergência dos parâmetros do controlador $C_1(z)$ malha interna, caso sem ruído.	49
Figura 11:	Convergência dos parâmetros do controlador $C_2(z)$ malha externa, caso com ruído.	49
Figura 12:	Saída final $y_1(t)$, malha interna do processo com ruído, depois de realizadas 20 iterações do algoritmo CIFT.	51
Figura 13:	Saída final $y_2(t)$, malha externa do processo com ruído, depois de realizadas 20 iterações do algoritmo CIFT.	51
Figura 14:	Convergência dos parâmetros do controlador $C_1(z)$ malha interna, caso com ruído.	52
Figura 15:	Convergência dos parâmetros do controlador $C_2(z)$ malha externa, caso com ruído.	52
Figura 16:	Saída do processo $y_2(t)$ com os controladores iniciais (108) e (109) (linha contínua); resposta desejada $y_d(t)$ (linha pontilhada).	53
Figura 17:	Demonstração gráfica da convergência dos controladores $C_1(z, \rho_1)$ e $C_2(z, \rho_2)$ utilizando o algoritmo CIFT.	56
Figura 18:	Comparação entre as saída $y_2(t)$ obtida com os métodos IFT e CIFT.	56
Figura 19:	Exemplo rotação de um dado ponto p	58
Figura 20:	Representação dos referenciais e orientação dos eixos do quadricóptero.	59
Figura 21:	Representação dos estados associadas ao quadricóptero.	61
Figura 22:	Forças e torques atuantes no quadricóptero (STEVENS; LEWIS, 2003).	64
Figura 23:	Estrutura simplificada para cálculo da matriz de inércia (BEARD, 2008).	67

Figura 24:	Diagrama de blocos do rotor realizada com auxílio da ferramenta <i>simulink</i> do <i>software</i> Matlab.	68
Figura 25:	Diagrama do sub-bloco de controle do quadricóptero com auxílio da ferramenta <i>simulink</i> do <i>software</i> Matlab.	68
Figura 26:	Diagrama do sub-bloco de controle angular com auxílio da ferramenta <i>simulink</i> do <i>software</i> Matlab.	69
Figura 27:	Diagrama do sub-bloco de controle da velocidade angular com auxílio da ferramenta <i>simulink</i> do <i>software</i> Matlab.	69
Figura 28:	Diagrama do sub-bloco de controle angular com auxílio da ferramenta <i>simulink</i> do <i>software</i> Matlab.	70
Figura 29:	Saída inicial com os controladores da equação (158), saída desejada e resultado final $y_2(t)$ utilizando o método de sintonia CIFT para o modelo simulado.	71
Figura 30:	Saída final $y_1(t)$ com os controladores iniciais da equação (158), utilizando o método de sintonia CIFT para o modelo simulado.	72
Figura 31:	<i>Frame</i> do quadricóptero utilizado para os experimentos práticos (HOBBY-KING.COM, ???).	75
Figura 32:	Exemplo genérico das zonas de funcionamento do motor de corrente contínua <i>brushless</i> (COSTA, 2008).	75
Figura 33:	Motor modelo NTM Prop Drive Series 2826 1100kv utilizado no experimento do quadricóptero (HOBBYKING.COM, ???).	76
Figura 34:	ESC utilizado para controle de velocidade dos motores do quadricóptero (ISEEROBOT.COM, ???).	76
Figura 35:	Hélice 8X4.5 utilizada no quadricóptero dos experimentos práticos.	77
Figura 36:	Placa modelo <i>Multiwii Flight Controller</i> utilizada como controlador para o quadricóptero dos experimentos (MULTICOPTERS.CO.UK, ???).	78
Figura 37:	Baterias utilizadas durante os experimentos práticos.	79
Figura 38:	Plataforma de testes.	80
Figura 39:	Diagrama de blocos da estrutura utilizada para o controle do ângulo <i>roll</i> do quadricóptero.	80
Figura 40:	Resposta inicial da saída $y_2(t)$ ângulo, utilizando os controladores iniciais (166).	82
Figura 41:	Resposta inicial com os controladores (166) e obtida após o uso do algoritmo CIFT para a saída da velocidade angular $y_1(t)$. Sendo o valor do termo integral de 0,01.	83
Figura 42:	Resposta inicial com os controladores (166), respostas desejada $y_d(t)$ e obtida após o uso do algoritmo CIFT para a saída do ângulo $y_2(t)$. Sendo o valor do termo integral de 0,01.	83
Figura 43:	Resposta inicial com os controladores (170) e obtida após o uso do algoritmo CIFT para a saída da velocidade angular $y_1(t)$. Sendo o valor do termo integral de 0,1.	85
Figura 44:	Resposta inicial com os controladores (170), respostas desejada $y_d(t)$ e obtida após o uso do algoritmo CIFT para a saída do ângulo $y_2(t)$. Sendo o valor do termo integral de 0,1.	85
Figura 45:	Convergência dos parâmetros do controlador $C_1(z)$ malha interna, para o experimento prático do quadricóptero.	86

Figura 46: Convergência dos parâmetros do controlador $C_2(z)$ malha externa, para o experimento prático do quadricóptero. 86

LISTA DE TABELAS

Tabela 1:	Convergência dos parâmetros do controlador e custo para o sistema exemplo.	32
Tabela 2:	Valor da função custo e convergência dos parâmetros do controlador, utilizando IFT em um sistema com integrador	34
Tabela 3:	Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o caso do exemplo sem ruído. . . .	48
Tabela 4:	Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o caso do exemplo com ruído. . . .	50
Tabela 5:	Convergência dos parâmetros da malha interna, comparativo usando IFT.	54
Tabela 6:	Convergência dos parâmetros da malha externa, comparativo usando IFT.	55
Tabela 7:	Convergência dos parâmetros da malha interna e externa, comparativo usando CIFT.	55
Tabela 8:	Características do motor AXI 2212-12 (HOBBYKING.COM, ???). . .	67
Tabela 9:	Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o modelo simulado do quadricóptero. . . .	73
Tabela 10:	Características do motor Prop Drive 2826 1100kv.	76
Tabela 11:	Lista dos componentes por peso do quadricóptero.	77
Tabela 12:	Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o experimento prático utilizando quadricóptero real. Valor do termo integral de 0,01.	84
Tabela 13:	Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o experimento prático utilizando quadricóptero real. Valor do termo integral de 0,1.	84

LISTA DE ABREVIATURAS

BMI	<i>Bilinear Matrix Inequality</i>
CIFT	<i>Cascade Iterative Feedback Tuning</i>
CbT	<i>Correlation based Tuning</i>
ESC	<i>Electronic Speed Control</i>
FDT	<i>Frequency Domain Tuning</i>
IMU	<i>Inertial Measurement Unit</i>
IFT	<i>Iterative Feedback Tuning</i>
LMI	<i>Linear Matrix Inequality</i>
LQ	<i>Linear Quadratic</i>
LQG	<i>Linear Quadratic Gaussian</i>
LTI	<i>Linear Time Invariant</i>
PD	Proporcional Derivativo
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PWM	<i>Pulse Width Modulation</i>
QN	<i>Quasi Newton</i>
RPM	Rotações Por Minuto
SD	<i>Steepest Descent</i>
UAV	<i>Unmanned Aerial Vehicle</i>
VANT	Veículo Aéreo Não Tripulados
VRFT	<i>Virtual Reference Feedback Tuning</i>

LISTA DE SÍMBOLOS

z	Operador de avanço
σ_e^2	Variância do ruído e
E	Valor esperado
\sum	Somatório
ρ	Vetor contendo os parâmetro do controlador $C(z, \rho)$
k_p	Ganho proporcional
k_i	Ganho integral
k_d	Ganho derivativo
$\overline{C}(z)$	Vetor contendo as funções de transferência para o formato vetorizada de $C(z, \rho)$
$\ \cdot \ _2^2$	Norma 2 ao quadrado
N	Tamanho da amostra
\overline{E}	$\lim_{N \rightarrow \infty} \sum_{t=1}^N E[\cdot]$
γ	Tamanho do passo do algoritmo iterativo
λ	Valor de escolhido para minimização do sinal de controle no algoritmo IFT
∇J	Gradiente de J
$\frac{\partial}{\partial}$	Derivada parcial
n	Parâmetro de projeto para tamanho do passo <i>Steepest Descent</i>
i	Iteração do algoritmo
L	Número máximo de iterações
$\hat{\cdot}$	Valor estimado
$\frac{\partial^2}{\partial}$	Derivada parcial de segunda ordem
$R(\cdot)_a^b$	Rotação do ponto a para o ponto b
ϕ	Ângulo roll
θ	Ângulo pitch
ψ	Ângulo yaw
$\frac{d}{dt}$	Derivada no tempo

× Produto vetorial
” Polegadas

1 INTRODUÇÃO

Quando projetistas de sistemas de controle se deparam com um novo desafio, surge uma comum dúvida: quais das diversas possibilidades dos métodos de controle devem ser usados? Uma resposta direta e simples para esta pergunta é difícil e demanda muito conhecimento por parte do projetista. De uma forma geral, quando é necessário lidar com processos desconhecidos a intuição nos força a acreditar que o correto procedimento é realizar a identificação do processo e com base no modelo obtido construir um controlador que alcance o desempenho desejado. Os métodos que utilizam este procedimento de identificação são conhecidos como métodos baseados em modelo.

A identificação de modelos é uma subárea de suma importância na área de controle de processos. O procedimento de identificação é tipicamente descrito em 3 etapas segundo (LJUNG, 1987) como: a coleta de dados, a escolha da estrutura do modelo candidato e a determinação dos parâmetros do modelo. A grande vantagem de usar um modelo para controlar um processo está na possibilidade de prever seu comportamento em diferentes pontos de operação, desde que o modelo tenha um certo grau de precisão. Definir o grau de precisão necessário e garantir que o modelo seja suficientemente robusto é uma área de grande interesse e tema de trabalhos como (HJALMARSSON, 2005) e (PROCHAZKA et al., 2005).

Uma possibilidade diferente dos métodos baseados em modelo são os chamados métodos baseados em dados. Este tipo de método deu seus primeiros passos graças ao trabalho de (HJALMARSSON; GUNNARSSON; GEVERS, 1994), no qual é proposto um método chamado *Iterative Feedback Tuning* (IFT). O método é classificado como um algoritmo iterativo de otimização que utiliza o critério de desempenho \mathcal{H}_2 , sendo o problema de otimização resolvido através do algoritmo do gradiente. O grande avanço no algoritmo foi a proposta de um estimador não polarizado para o gradiente da função custo que considera em sua forma o problema de seguimento de referência, atenuação de ruído e minimização do esforço de controle.

A metodologia baseada em dados tem como diferencial abandonar a etapa de identificação de um modelo e utilizar os dados coletados a partir de experimentos para realizar de forma direta a sintonia dos controladores, devido a isto este método é também conhecido como *direct approach*. Vários trabalhos foram propostos ao decorrer das últimas duas décadas mantendo o mesmo diferencial fundamental do controle baseado em dados, como exemplo alguns dos principais trabalhos são: (KAMMER; BITMEAD; BARTLETT, 2000) que propõe um método chamado de *Frequency Domain Tuning* (FDT), utilizando o mesmo critério de desempenho \mathcal{H}_2 e se assemelhando muito ao IFT original, sua diferença reside em ser um método para sistemas com referencial nulo e seu desenvolvimento ser realizado no domínio da frequência; em (KARIMI; MISKOVIC; BONVIN, 2004) é proposto um método chamado de *Correlation based Tuning* (CbT), neste método

são utilizadas variáveis instrumentais para reduzir o efeito do ruído, outra diferença reside no problema de otimização onde o objetivo é a solução para um sistema de equações de correlação; por último temos o trabalho de (CAMPI; LECCHINI; SAVARESI, 2002) que propõe o conhecido método *Virtual Reference Feedback Tuning* (VRFT), este método se diferencia por não ser iterativo onde o critério de desempenho \mathcal{H}_2 é transformado em um critério quadrático, sendo possível realizar a minimização da função custo através de uma única iteração.

O presente trabalho propõe uma extensão para sistemas em formato cascata utilizando como base o *Iterative Feedback Tuning*. Processos em formato cascata são largamente utilizados em diversas aplicações, as mais conhecidas são para o setores químico-industrial e petroquímico (CORRIPIO, 2000). Uma estrutura em formato cascata tem por definição mais de uma malha de controle. Muitos processos possuem somente duas malhas, no entanto, em alguns processos mais complexos este número geralmente é maior. O trabalho se limita a usar um formato mais comum de estrutura com duas malhas de controle definidas como interna e externa.

O objetivo do algoritmo proposto é realizar a sintonia dos controladores em formato cascata com duas malhas, interna e externa. O seu desenvolvimento foi realizado levando em consideração as limitações apresentadas pelo clássico algoritmo IFT para estruturas em cascata. O uso do algoritmo clássico ocasionaria dois principais problemas: primeiro, seria necessário desacoplar as malhas e realizar a sintonia de forma independente, duplicando assim o trabalho de sintonia; segundo, seria necessária a escolha de uma resposta desejada para a malha interna. Esta escolha é muito difícil de ser realizada, pois não há como ter o correto conhecimento sobre a interferência da dinâmica da malha interna no resultado final, uma vez que desconhecemos o processo que estamos lidando. O trabalho de (JENG; LIAO, 2013) utiliza controle baseado em dados para realizar a sintonia de controladores em formato cascata com duas malhas, sendo a única bibliografia encontrada sobre o tema, todavia o trabalho utiliza um algoritmo não iterativo e há necessidade da escolha de uma resposta desejada para a malha interna.

Para validar o algoritmo proposto é utilizado como aplicação um *Unmanned Aerial Vehicle* (VANT) real. O UAV em questão é composto de 4 hélices que sustentam e realizam seus movimentos. Sendo conhecido como quadricóptero, quadcoptero ou quadrotor. Quadricópteros são bem populares entre os praticantes de aerodelismo dada a sua simplicidade construtiva e baixo custo. Sua grande aplicabilidade nas mais diversas áreas de atuação também é um fator que impulsiona o mercado de aerodelismo no mundo (GUPTE; MOHANDAS; CONRAD, 2012). As aplicações mais convencionais são para as áreas de supervisão, fotos, análises topográficas, monitoramento e exploração (PURI, 2005).

O quadricóptero apresenta um comportamento dinâmico bem próprio e um desafio de controle com um grau de dificuldade elevado, isto em grande parte por se tratar de um modelo não-linear e multivariável. Este tipo de UAV serve como aplicação alvo em testes das atuais propostas de controle avançado. Como exemplo, são citados os seguintes trabalhos: (BOUABDALLAH; NOTH; SIEGWART, 2004) que compara o uso do controlador Proporcional-Integral-Derivativo (PID) com *Linear-Quadratic* (LQ); no trabalho de (ALTUĞ; OSTROWSKI; MAHONY, 2002) é usada uma forma diferente de sensoriamento, nela é usada uma câmera como realimentação de posição; em (MADANI; BENALLI-GUE, 2006) é utilizada a técnica de controle conhecida como *Backstepping*; em (COZA; MACNAB, 2006) é utilizado um controle adaptativo baseado na lógica fuzzy.

A modelagem dinâmica do quadricóptero pode ser encontrada em muitas referências,

entretanto é reservado um capítulo para trazer uma explicação baseada nos trabalhos de (BEARD, 2008) e (STEVENS; LEWIS, 2003), a cinemática envolvida em seu movimento é bem explicada em (CRAIG, 2005) onde é encontrada a maioria dos conceitos presentes neste trabalho. Com o auxílio da ferramenta Simulink do *software* Matlab um modelo de quadricóptero é construído, com a sua ajuda são explicadas todas as malhas de controle que compõem o controle angular de um quadricóptero, bem como sua estabilização em voo. Com todos os conceitos abordados sobre a dinâmica do UAV, são apresentados os resultados para o caso real e simulado do quadricóptero, como forma de complementar o entendimento do método proposto.

Esta dissertação está organizada da seguinte forma: No Capítulo 2 é realizada uma revisão bibliográfica a respeito do algoritmo *Iterative Feedback Tuning* e é apresentado um exemplo numérico para o mesmo. No Capítulo 3 uma extensão do algoritmo *Iterative Feedback Tuning* para sistemas em formato cascata é proposta, o qual é nomeado como *Cascade Iterative Feedback Tuning* (CIFT). Ao final deste capítulo é demonstrado um exemplo numérico comparando os métodos de sintonia IFT e CIFT. No Capítulo 4 é descrito o comportamento dinâmico de um quadricóptero, e uma demonstração simulada do algoritmo *Cascade Iterative Feedback Tuning* aplicado ao quadricóptero é apresentada. No Capítulo 5 são demonstrados os resultados dos experimentos utilizando o algoritmo de sintonia proposto aplicado experimentalmente no quadricóptero, juntamente com todos os aspectos que envolvem o experimento, como plataforma de testes e descrição detalhada do quadricóptero.

2 ITERATIVE FEEDBACK TUNING - IFT

O *Iterative Feedback Tuning* (IFT) é um método de controle baseado em dados que foi primeiramente proposto em (HJALMARSSON; GUNNARSSON; GEVERS, 1994), a mais de duas décadas. Com o passar dos anos o método foi subindo os degraus da notoriedade e atualmente é possível afirmar que se encontra bem estabelecido, como pode ser observado nos outros trabalhos de (HJALMARSSON et al., 1998) e (HJALMARSSON, 2002).

O controle baseado em dados tem como premissa fundamental ser uma metodologia para projetar os parâmetros de um controlador, utilizando somente dados de entrada e saída do processo, de uma forma direta e sem utilizar nenhum modelo (BAZANELLA; CAMPESTRINI; ECKHARD, 2011). Para projetar os parâmetros do controlador a metodologia aborda-o como sendo um problema de otimização. Comumente o objetivo de otimização, trata-se em encontrar o controlador ótimo para um dado desempenho desejado do sistema em malhada fechada, seja ele seguimento de referência, atenuação de ruído ou minimização do esforço de controle.

O método de controle baseado em dados se diferencia do baseado em modelo por não utilizar nenhum conhecimento da função de transferência do sistema *a priori*. Ao invés disto utiliza somente dados de entrada e saída para calcular os parâmetros do controlador, dentro da subárea dos algoritmos adaptativos este procedimento é conhecido como *direct approach*, diferente do *indirect approach*, onde os parâmetros do controlador são obtidos a partir de um modelo.

Ao longo dos anos vários métodos de controle baseado em dados foram propostos tendo como meta tais objetivos mencionados, o que diferencia os métodos é a abordagem dada ao problema como o problema de otimização ou o uso de diferentes funções custos para específicas aplicações. Pode-se apontar como mais representativos os trabalhos: (HJALMARSSON; GUNNARSSON; GEVERS, 1994) - *Iterative Feedback Tuning* (IFT); (KAMMER; BITMEAD; BARTLETT, 2000) - *Frequency Domain Tuning* (FDT); (CAMPI; LECCHINI; SAVARESI, 2002) - *Virtual Reference Feedback Tuning* (VRFT); (KARIMI; MISKOVIC; BONVIN, 2004) - *Correlation-based Tuning* (CbT); (SHI; SKELTON, 2000) - *Markov Linear Quadratic Gaussian Control*.

O método IFT se diferencia por ser iterativo e utilizar como critério de desempenho a norma \mathcal{H}_2 , largamente utilizada em algoritmos de otimização. O objetivo da otimização é encontrar o controlador ótimo que minimize a norma \mathcal{H}_2 da diferença entre o sinal de saída do sistema e uma dada resposta desejada para o mesmo.

Na sequência é apresentado com profundidade o algoritmo IFT, grande parte do desenvolvimento pode ser consultado em (BAZANELLA; CAMPESTRINI; ECKHARD, 2011), seguido pela explicação do seu uso com o algoritmo do gradiente, finalizando o capítulo com um exemplo numérico que engloba todo o conhecimento apresentado.

2.1 Definições

2.1.1 Processo

Considera-se um processo monovariável linear discreto e invariante no tempo (LTI), dado por

$$y(t) = G(z)u(t) + v(t), \quad (1)$$

onde $y(t)$ é a saída do processo, $u(t)$ é a entrada de controle, $G(z)$ é a função de transferência discreta do processo, z é o operador de avanço no tempo e $v(t)$ é o ruído do processo. Considera-se o ruído sendo um processo quasi-estacionário, resultado da filtragem de um ruído branco gaussiano pelo sistema LTI conforme, $v(t) = H(z)e(t)$, sendo $e(t)$ um ruído branco com variância de σ_e^2 .

Ambas funções de transferências $G(z)$ e $H(z)$ são racionais, causais e desconhecidas, assume-se que $H(\infty) = 1$.

Define-se um sinal quasi-estacionário como:

Definição 1. (LJUNG, 1987): *Um sinal $s(t)$ é dito quasi-estacionário se:*

- $E[s(t)] = m_s, |m_s(t)| \leq C, \forall t$
- $E[s(t)s(r)] = R_s(t, r), |R_s(t, r)| \leq C, \forall t, r$
- $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N R_s(t, \tau) = R_s(\tau), \forall t$

onde $E[\cdot]$, é o valor esperado do sinal.

O processo é controlado por um controlador linear invariante no tempo, sendo sua estrutura pertencente a uma dada classe \mathcal{C} de funções de transferência, sendo sua escolha dada conforme a necessidade da aplicação. O controlador é parametrizado em um formato vetorial tendo todos os parâmetros reais $\rho \in \mathbb{R}^p$. O sinal de controle $u(t)$ pode ser escrito como

$$u(t) = C(z, \rho)(r(t) - y(t)), \quad (2)$$

onde $r(t)$ é o sinal de referência, ρ é o vetor de parâmetros do controlador $C(z, \rho)$, sendo sua função de transferência racional e causal. O sinal de referência $r(t)$ é um sinal determinístico e considerado quasi-estacionário, logo, o sinal de referência não é correlacionado com o ruído do processo, portanto

$$E[r(t)e(\tau)] = 0 \quad \forall t, \tau.$$

2.1.2 Controle

O conjunto de todos os possíveis controladores que podem ser implementados conforme uma dada estrutura específica é definido como classe de controlador \mathcal{C} sendo:

$$\mathcal{C} = \{C(z, \rho) : \rho \in \mathcal{P} \subseteq \mathbb{R}^p\}. \quad (3)$$

É possível utilizar o clássico controlador Proporcional-Integral-Derivativo (PID) como exemplo de estrutura de controle. A estrutura de controle PID é largamente conhecida e utilizado nas mais variadas aplicações, o controlador tem sua estrutura dada por

$$C(z, k_p, k_i, k_d) = k_p + k_i \frac{z}{z-1} + k_d \frac{z-1}{z}, \quad (4)$$

onde k_p , k_i e k_d são os ganhos proporcional, integral e derivativo, respectivamente. O ajuste dos ganhos do controlador PID é o principal objetivo ao lidar com este tipo de estrutura, e comumente o principal desafio de controle nos processos.

O sistema linear (4) pode ser parametrizado em formato vetorial na forma

$$C(z, \rho) = \rho^T \bar{C}(z), \quad (5)$$

onde ρ é o vetor dos parâmetros do controlador a serem ajustados e $\bar{C}(z)$ é o vetor com as funções de transferências, ambas definidas para controladores PID como:

$$\rho = \begin{bmatrix} k_p \\ k_i \\ k_d \end{bmatrix} \quad \bar{C}(z) = \begin{bmatrix} 1 \\ \frac{z}{z-1} \\ \frac{z-1}{z} \end{bmatrix}. \quad (6)$$

2.1.3 Processo em malha fechada

O processo em malha fechada é definido como o resultante dos sinais de entrada $r(t)$ e $e(t)$, com o sinal de saída $y(t)$ como:

$$\begin{aligned} y(t, \rho) &= T(z, \rho)r(t) + S(z, \rho)H(z)e(t) \\ S(z, \rho) &= \frac{1}{1 + C(z, \rho)G(z)} \\ T(z, \rho) &= \frac{C(z, \rho)G(z)}{1 + C(z, \rho)G(z)} = C(z, \rho)G(z)S(z, \rho). \end{aligned}$$

onde as funções de transferência em malha fechada $S(z, \rho)$ e $T(z, \rho)$, são definidas respectivamente como sensibilidade e sensibilidade complementar do sistema. A representação em diagrama de blocos do sistema em malha fechada é demonstrada na Figura 1.

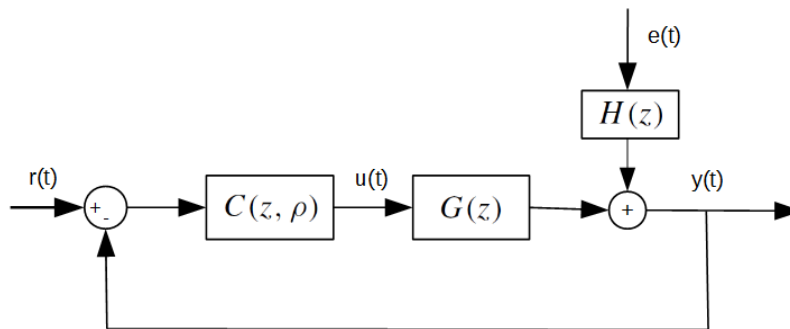


Figura 1: Diagrama de blocos, sistema em malha fechada.

2.2 Problema de otimização

O principal objetivo da maioria dos algoritmos de otimização, é a minimização ou maximização de uma função custo em específico. O problema tratado neste trabalho não é diferente, o objetivo do método IFT é a minimização da função custo norma \mathcal{H}_2 dada por

$$J(\rho) = \|y_d(t) - y(t, \rho)\|_2^2 = \frac{1}{N} \sum_{t=1}^N (y_d(t) - y(t, \rho))^2, \quad (7)$$

onde ρ é o vetor com os parâmetros do controlador, $y_d(t)$ é definido como o sinal de saída desejado do sistema, ou seja, o comportamento desejado para o sistema a ser controlado. A abordagem mais clássica para obter este sinal é utilizar um modelo de referência $T_d(z)$ para o sistema em malha fechada como:

$$y_d(t) = T_d(z)r(t). \quad (8)$$

A função custo $J(\rho)$ também pode ser vista como um critério de desempenho da otimização, quanto menor o seu valor, mais próxima a solução do problema se encontra. O problema de otimização do método *Iterative Feedback Tuning* pode ser sintetizado como:

$$\min_{\rho \in P} J(\rho) \quad (9)$$

De acordo com a necessidade do projeto de controle, é possível adicionar um efeito de penalidade sobre o esforço de controle, a equação (7) pode ser reescrita como

$$J(\rho) = \frac{1}{N} \sum_{t=1}^N (y_d(t) - y(t, \rho))^2 + \lambda u(t, \rho)^2, \quad (10)$$

sendo λ a variável de escolha do projetista que define o peso entre seguimento de referência e a minimização do esforço de controle.

Os problemas de otimização enfrentados pelo método *Iterative Feedback Tuning*, quase sempre se deparam com sistemas com mínimos locais em sua função custo, isto porque raramente encontramos funções custo convexas em situações práticas de controle, dificultando ou impossibilitando a convergência.

Este tipo de formulação de função custo apresentada na equação (10) é muito utilizada nos métodos de projeto de controladores baseados em modelo, por exemplo, LQR/LQG (ANDERSON; MOORE, 1971) e controle preditivo generalizado (BITMEAD; GEVERS; WERTZ, 1990). Ambos os métodos utilizam, para encontrar a solução de (10), ferramentas como: equação de Riccati, Inequações Matriciais Lineares (LMIs), Inequações Matriciais Bilineares (BMIs)(BOYD et al., 1994). Entretanto como mencionado, estes métodos utilizam ferramentas que necessariamente precisam do modelo do sistema, previamente obtido em alguma etapa, etapa esta que não é necessária em *Iterative Feedback Tuning*.

A minimização da função custo de (10) não engloba somente alcançar uma resposta desejada ou reduzir o esforço de controle, é possível também realizar a minimização do ruído que incorpora a saída do sistema. Estas propriedades são melhores percebidas ao dividir em três partes a função custo conforme (GOODWIN; SIN, 1984):

$$J(\rho) = [J_y(\rho) + J_e(\rho)] + \lambda J_u(\rho); \quad (11)$$

onde são definidos

$$J_y(\rho) = \overline{E}[(T(z, \rho) - T_d(z))r(t)^2] \quad (12)$$

$$J_e(\rho) = \overline{E}[(S(\rho)v(t))^2] \quad (13)$$

$$J_u(\rho) = \overline{E}[(u(t, \rho))^2] \quad (14)$$

sendo $\overline{E}[\cdot]$ dado por:

$$\overline{E}[\cdot] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N E[\cdot] \quad (15)$$

Cada uma destas parcelas pode ser interpretada como uma parte significativa da otimização do algoritmo, $J_y(\rho)$ é responsável pela parte de seguimento de referência, $J_e(\rho)$ é responsável pela parte da minimização do ruído e $J_u(\rho)$ atua na otimização do esforço de controle.

2.2.1 Seguimento da resposta desejada

Esta parte da função custo é responsável pelo seguimento da resposta desejada em malha fechada do sistema, desconsiderando ruído e esforços de controle. Em alguns casos práticos não é possível garantir a resposta desejada $y_d(t)$, pois a classe \mathcal{C} é muito restritiva. Neste caso o projetista pode alterar a classe do controlador ou alterar os critérios de desempenho para a resposta desejada, atribuindo nesta um conjunto de especificações que variam conforme a necessidade e aplicação, especificações conhecidas como máximo sobressinal, tempo de subida, tempo de acomodação entre outros.

Caso seja considerado somente o seguimento da resposta desejada, é possível idealizar um controlador desejado $C_d(z)$ para um dado sistema em malha fechada, este controlador tem a capacidade de atingir exatamente os requisitos desejados pelo modelo de referência $T_d(z)$. O controlador desejado é definido como

$$C_d(z) = \frac{T_d(z)}{G(z)(1 - T_d(z))}. \quad (16)$$

Portanto para atingir o objetivo de seguimento da resposta desejada, o controlador $C_d(z, \rho)$ deve pertencer a classe \mathcal{C} . Calcular de forma analítica o controlador $C_d(z)$ é algo trivial quando se tem o completo conhecimento de $G(z)$, sem este conhecimento, seria necessária uma etapa anterior de modelagem ou identificação a fim de levantar a função de transferência. A etapa de obtenção do modelo de processo não existe na metodologia IFT e sempre são considerados como processos desconhecidos.

2.2.2 Rejeição a ruído

A minimização do custo associado ao ruído do processo é outro tópico importante para o projeto do controlador e comumente conhecida por controle de mínima variância. Um exemplo envolvendo controle adaptativo é demonstrado em (ÅSTRÖM; WITTENMARK, 1973).

É impossível obter um controlador que elimine por completo o efeito do ruído no processo, ou seja, $J_e(\rho) = 0$. Para isto seria necessário que $S(z, \rho) = 0 \forall z$, o que por sua vez demandaria $C(z, \rho)G(z) \rightarrow \infty \forall z$. Entretanto atenuações são perfeitamente possíveis de serem realizadas, algo demonstrado em (ÅSTRÖM, 1970), sendo o controlador dado por

$$C_e(z) = \frac{H(z) - 1}{G(z)}, \quad (17)$$

semelhante ao seguimento de referência, calcular analiticamente o controlador $C_e(z)$ é algo fácil quando temos o conhecimento da função de transferência do sistema $G(z)$.

2.2.3 Esforço de controle

Caso o objetivo principal da minimização seja o esforço de controle $J_u(\rho)$, o resultado da minimização será sempre $u(t) = 0$, ou seja, desligar o controle. É por este motivo que a minimização da função custo de (14) sempre deve ser combinada com outra.

O projetista deve definir o peso $\lambda \in \mathbb{R}$ com a finalidade de optar por quais das parcelas da função custo, deve ser dada maior importância na minimização. Por exemplo, na equação (11) é possível dar maior prioridade na minimização do custo de $J_y(\rho)$ e $J_e(\rho)$, ao escolher valores entre $0 < \lambda \leq 1$.

2.3 Algoritmo do gradiente

Vários algoritmos de otimização podem ser utilizados para resolver o problema de otimização da minimização de (7), por exemplo Gauss-Newton, Levenberg-Marquardt ou Newton-Raphson. Os algoritmos citados são semelhantes por serem todos baseados no gradiente da função custo para realizar sua minimização. Visto que os algoritmos baseados em gradiente são largamente empregados em otimizações, serão utilizados neste trabalho dois algoritmos comuns na área de controle baseado em dados os conhecidos *Steepest Descent* e Quasi-Newton (BAZANELLA; CAMPESTRINI; ECKHARD, 2011).

A premissa fundamental dos algoritmos que utilizam o gradiente é dar sucessivos passos na direção inversa indicada pelo gradiente da função custo. Este tipo de procedimento é largamente conhecido e foi primeiramente proposto em (CAUCHY, 1847) para encontrar mínimos de funções, formalmente definimos o algoritmo *Steepest Descent* como:

$$\rho_{i+1} = \rho_i - \gamma_i \nabla J(\rho_i), \quad (18)$$

onde $\nabla J(\rho) = \frac{\partial J(\rho)}{\partial \rho}$ e $\gamma_i > 0$ é o tamanho do passo dado a cada iteração. Portanto a cada iteração i são atualizados os parâmetros ρ na direção oposta pelo gradiente, assim um valor menor de custo é alcançado a cada iteração. A convergência deste algoritmo também depende das condições iniciais ρ_0 e do tamanho do passo escolhido γ_i .

Outro método derivado do gradiente é o conhecido Quasi-Newton definido como

$$\rho_{i+1} = \rho_i - \gamma_i H^{-1} \nabla J(\rho_i) \quad (19)$$

onde H é uma aproximação para a matriz Hessiana definida por

$$H = \frac{1}{N} \sum_{t=0}^N \frac{\partial y(t, \rho)}{\partial \rho} \frac{\partial y(t, \rho)}{\partial \rho}^T. \quad (20)$$

onde N é o tamanho da amostra coletada.

Diferente do *Steepest Descent*, o método Quasi-Newton não foi proposto para encontrar mínimos da função custo, o algoritmo foi elaborado com a finalidade de encontrar as raízes da equação $\nabla J(\rho) = 0$, devido a isto dependendo do ponto inicial de convergência o algoritmo converge para pontos onde a derivada de $\nabla J(\rho)$ é nula, por exemplo máximos e mínimos e pontos de sela. O recomendado e utilizado neste trabalho, é inicializar o algoritmo em uma região próxima do mínimo, ou seja, utilizar o algoritmo em uma segunda etapa de convergência quando notada uma redução na taxa de decaimento da função custo $J(\rho)$, o que ocasiona uma perda de eficiência por parte do algoritmo *Steepest Descent*.

Na bibliografia atual, é muito difícil chegar no consenso sobre a correta metodologia que guie a tomada de decisão na troca para o algoritmo Quasi-Newton. Isto porque cada convergência é única para uma dada aplicação de controle, sendo assim não conseguimos utilizar a mesma metodologia em todos os experimentos. Neste trabalho a tomada de decisão para a troca de algoritmo foi realizada de modo empíricos e escolhemos a convergência que obteve o melhor resultado.

2.3.1 Convergência do algoritmo

O tamanho do passo γ é um parâmetro fundamental no algoritmo, sua escolha traz reflexos diretos na sua convergência. Caso o tamanho do passo seja muito grande o algoritmo pode não convergir (ECKHARD, 2008). Em teoria, um valor muito pequeno não traria maiores problemas do que um baixo desempenho na velocidade de convergência.

Dada a sua grande importância para a convergência, muitos trabalhos foram propostos sobre o tema da escolha do tamanho de passo como em (HJALMARSSON et al., 1998), onde é proposto um método baseado em (ROBBINS; MONRO, 1951) que garante a convergência para um mínimo da função custo. Outros trabalhos também podem ser relacionados como em (GOLDSTEIN, 1962) e (ECKHARD, 2008). Como já citado acima, são utilizados neste trabalho os algoritmos *Steepest Descent* e Quasi-Newton para solucionar o problema de otimização. Na sequência é discutida a convergência de ambos algoritmos.

O algoritmo do gradiente converge em probabilidade para um mínimo global da função custo caso utilizado a seguinte regra para a escolha do tamanho de passo

$$\sum_{i=1}^{\infty} \gamma_i = \infty \quad \sum_{i=1}^{\infty} \gamma_i^2 < \infty, \quad (21)$$

Esta sequência garante a convergência para o valor exato do mínimo global e é bastante utilizada em varias aplicações de controle baseado em dados, o inconveniente desta sequência reside no fato que a taxa de convergência é geralmente muito baixa, necessitando assim muitos experimentos para alcançar o mínimo.

Neste trabalho é utilizado o algoritmo de Quasi-Newton para garantir a convergência para o mínimo global segundo o seguinte teorema:

Definição 2. (BAZANELLA; CAMPESTRINI; ECKHARD, 2011) *Seja o ρ_* um extremo isolado da função custo $J(\rho)$, então o ρ_* é um ponto-fixo atrativo para o algoritmo do Quasi-Newton.*

O teorema acima garante que existe uma região de atração em torno do mínimo global da função custo para o qual se garante a convergência. Sabe-se que o domínio de atração é geralmente muito pequeno e portanto é recomendado neste trabalho que se utilize o algoritmo do gradiente para encontrar a condição inicial para o algoritmo de Quasi-Newton. Contudo, não é necessário utilizar o algoritmo do gradiente caso esteja disponível um ponto inicial perto do mínimo.

Neste trabalho utilizamos processos que são contaminados por ruídos, portanto as estimativas apresentam um erro relacionado a este ruído. Contudo de acordo com o Teorema 4.6 do livro (BAZANELLA; CAMPESTRINI; ECKHARD, 2011), quando as estimativas do gradiente da função custo são contaminadas por ruído então o algoritmo converge para uma região entorno do mínimo global e o tamanho desta região depende da quantidade de ruído.

Neste trabalho quando o algoritmo *Steepest Descent* é utilizado foi escolhido o seguinte sequência de passo:

$$\text{Steepest Descent} : \gamma_i = \frac{n}{\|\nabla J_{i-1}\|}, \quad (22)$$

onde n é uma constante de projeto. Esta sequência é uma derivação da recomendação apresentada em (BAZANELLA; CAMPESTRINI; ECKHARD, 2011). A sequência

acima não garante a convergência para o mínimo global da função custo, entretanto é utilizada para aumentar a velocidade de convergência do algoritmo nas primeiras iterações. Para garantir a convergência, é utilizado o algoritmo Quasi-Newton em uma segunda etapa de convergência.

É escolhido a seguinte sequência de tamanho de passo para o algoritmo Quasi-Newton:

$$\text{Quasi-Newton} : \gamma_i = \frac{i}{L}, \quad (23)$$

onde L é o número máximo de iterações utilizadas no experimento. O algoritmo Quasi-Newton é uma derivação do conhecido Newton-Raphson, a principal diferença entre eles é a adição do tamanho do passo como variável de projeto e uso de uma aproximação da matriz hessiana.

Uma qualidade importante do algoritmo de Newton-Raphson está em realizar a convergência em uma única iteração, caso a função custo seja quadrática. Isto também pode acontecer em funções custo não quadráticas, caso o algoritmo se encontre perto o suficiente do mínimo global. A escolha por estas sequências de tamanho de passo advém da possibilidade de usar somente as propriedades benéficas de cada algoritmo. Comumente o algoritmo *Steepest Descent* tem uma larga região de atração, mas tende a ter uma baixa taxa de convergência, ao contrário da Quasi-Newton que tem uma alta taxa de convergência entretanto com uma menor região atração (RALSTON; RABINOWITZ, 1965).

2.4 Gradiente da função custo $J(\rho)$

Como visto na seção anterior, os algoritmos *Steepest Descent* (18) e Quasi-Newton (19) são capazes de minimizar a função custo (7) realizando a convergência dos parâmetros do controlador, o principal objetivo do *Iterative Feedback Tuning*.

Entretanto para fazer uso destes algoritmos é necessário obter o gradiente da função custo $\nabla J(\rho)$, o qual é a principal contribuição do algoritmo *Iterative Feedback Tuning*. Uma outra forma de explicar o algoritmo, é como um método capaz de obter a estimativa não polarizada do gradiente. Grande parte do desenvolvimento apresentando a seguir é minusiosamente apresentado em (BAZANELLA; CAMPESTRINI; ECKHARD, 2011), sendo a bibliografia fonte da fundamentação teórica deste trabalho.

Considerando a seguinte norma \mathcal{H}_2 como função custo a ser minimizada por

$$J(\rho) = \frac{1}{N} \sum_{t=0}^N [(y_d(t) - y(t, \rho))^2 + \lambda u^2(t, \rho)] \quad (24)$$

Derivando (24) com respeito ao parâmetro ρ obtemos

$$\frac{\partial J(\rho)}{\partial \rho} = \frac{2}{N} \sum_{t=0}^N \left[[y_d(t) - y(t, \rho)] \frac{\partial y(t, \rho)}{\partial \rho} + \lambda u(t, \rho) \frac{\partial u(t, \rho)}{\partial \rho} \right] \quad (25)$$

onde λ é um valor compreendido entre $0 \leq \lambda \leq 1$ que corresponde ao seu valor proporcional do quanto é desejado minimizar o sinal de controle na função custo, esta escolha é um valor de projeto e definida conforme a necessidade da aplicação.

Para a implementação do gradiente é necessário computar a equação anterior, entretanto a equação depende de processos estocásticos que por sua vez são desconhecidos. Observando todas as parcelas de (25), é possível agrupa-los em: Sinais que são diretamente medidos como $y(t, \rho)$ e $u(t, \rho)$; sintetizado como em $y_d(t)$. Entretanto, as parcelas

envolvendo as derivadas parciais de $\frac{\partial y(t, \rho)}{\partial \rho}$ e $\frac{\partial u(t, \rho)}{\partial \rho}$, não são medidas de uma forma direta, portanto elas devem ser sintetizadas de alguma maneira afim de obter o gradiente da função custo. É demonstrado na sequência como obter as parcelas necessárias para o cálculo do gradiente.

Relembrando das definições de malha fechada para o processo temos

$$\begin{aligned} y(t, \rho) &= T(z, \rho)r(t) + S(z, \rho)v(t) \\ S(z, \rho) &= \frac{1}{1 + C(z, \rho)G(z)} \\ T(z, \rho) &= \frac{C(z, \rho)G(z)}{1 + C(z, \rho)G(z)} = C(z, \rho)G(z)S(z, \rho). \end{aligned}$$

A derivada parcial de $y(t, \rho)$ é obtida por

$$\begin{aligned} \frac{\partial y(t, \rho)}{\partial \rho} &= \frac{G(z)}{1 + C(z, \rho)G(z)} \frac{\partial C(z, \rho)}{\partial \rho} r(t) - \frac{C(z, \rho)G(z)^2}{(1 + C(z, \rho)G(z))^2} \frac{\partial C(z, \rho)}{\partial \rho} r(t) \\ &\quad - \frac{G(z)}{(1 + C(z, \rho)G(z))^2} \frac{\partial C(z, \rho)}{\partial \rho} v(t) \end{aligned}$$

a qual pode ser reescrita como

$$\frac{\partial y(t, \rho)}{\partial \rho} = \frac{1}{C(z, \rho)} \frac{\partial C(z, \rho)}{\partial \rho} \{T(z, \rho)r(t) - T(z, \rho)[T(z, \rho)r(t) + S(z, \rho)v(t)]\}.$$

Observando a parcela dentro da multiplicação entre chaves, percebe-se que não é possível obtê-la sem o completo conhecimento da função de transferência $G(z)$ e do ruído $v(z)$ que compõe o sistema. É possível reescrever a equação acima ao perceber que $y(t, \rho) = T(z, \rho)r(t) + S(z, \rho)v(t)$, substituindo este termo a equação pode ser reescrita como

$$\frac{\partial y(t, \rho)}{\partial \rho} = \frac{1}{C(z, \rho)} \frac{\partial C(z, \rho)}{\partial \rho} \{T(z, \rho)[r(t) - y(t, \rho)]\}. \quad (26)$$

A principal informação retirada da equação anterior é que possível sintetizar o sinal entre chaves ao aplicar o seu respectivo sinal de erro, ou seja, $r(t) - y(t, \rho)$ como referência em um novo experimento. Este conhecimento e procedimento é o fruto de todo desenvolvimento do *Iterative Feedback Tuning*. Para realizá-la é necessário seguir o seguinte procedimento: coletar os dados de $y(t, \rho)$ em condição normal de operação, realizar um novo experimento utilizando como referência o sinal de $r(t) - y(t, \rho)$ e coletar a resposta desta excitação.

É importante notar que ao realizar um novo experimento, o sinal $T(z, \rho)[r(t) - y(t, \rho)]$ estará contaminado com um novo ruído $S(z, \rho)v(t)$ indesejado para o cálculo exato do gradiente, entretanto uma vez que o ruído do processo tenha média zero, a estimativa do sinal não será polarizada, sendo

$$E [T(z, \rho)\{r(t) - y(t, \rho)\} + S(z, \rho)v(t)] = T(z, \rho)\{r(t) - y(t, \rho)\}. \quad (27)$$

A derivada parcial de $\frac{\partial u(t, \rho)}{\partial \rho}$ é obtida de uma maneira similar e seguindo os mesmos passos demonstrados anteriormente, lembrando que

$$u(t, \rho) = \frac{C(z, \rho)}{1 + C(z, \rho)G(z)} [r(t) - v(t)] = S(z, \rho)C(z, \rho)[r(t) - v(t)]. \quad (28)$$

Derivando a equação anterior com relação ao parâmetro ρ

$$\frac{\partial u(t, \rho)}{\partial \rho} = S(z, \rho) \frac{\partial C(z, \rho)}{\partial \rho} [r(t) - v(t)] + \frac{\partial S(z, \rho)}{\partial \rho} C(z, \rho) [r(t) - v(t)]. \quad (29)$$

Percebendo que a seguinte relação é possível

$$\frac{\partial S(z, \rho)}{\partial \rho} = -\frac{1}{C(z, \rho)} \frac{\partial C(z, \rho)}{\partial \rho} T(z, \rho) S(z, \rho). \quad (30)$$

A seguinte equação é obtida ao manipular as duas últimas equações e agrupando conforme:

$$\begin{aligned} \frac{\partial u(t, \rho)}{\partial \rho} &= S(z, \rho) \frac{\partial C(z, \rho)}{\partial \rho} [r(t) - v(t)] - \frac{\partial C(z, \rho)}{\partial \rho} T(z, \rho) S(z, \rho) [r(t) - v(t)] \\ &= S(z, \rho) \frac{\partial C(z, \rho)}{\partial \rho} [r(t) - T(z, \rho)r(t) - (1 - T(z, \rho))v(t)] \\ &= \frac{\partial C(z, \rho)}{\partial \rho} S(z, \rho) [r(t) - y(z, \rho)] \\ &= \frac{1}{C(z, \rho)} \frac{\partial C(z, \rho)}{\partial \rho} \{C(z, \rho) S(z, \rho) [r(t) - y(t, \rho)]\} \end{aligned}$$

De maneira similar a anterior, nota-se que para obter a derivada parcial de $\frac{\partial u(t, \rho)}{\partial \rho}$ um experimento deve ser realizado excitando o sistema com uma entrada do tipo $r(t) - y(t, \rho)$.

Ambas derivadas parciais são corrompidas com os ruídos de outros experimentos impossibilitando o cálculo preciso de suas parcelas. Entretanto, como o ruído tem média zero, o valor estimado das derivadas parciais não é polarizada para o cálculo do gradiente. O gradiente da função custo (25) pode ser estimado por

$$\frac{\partial \widehat{J}(\rho)}{\partial \rho} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y(t, \rho)] \frac{\partial \widehat{y}(t, \rho)}{\partial \rho} + \lambda \frac{2}{N} \sum_{t=1}^N u(t, \rho) \frac{\partial \widehat{u}(t, \rho)}{\partial \rho} \quad (31)$$

onde o símbolo matemático $\widehat{\cdot}$ representa uma estimativa do seu valor exato, neste trabalho os sinais estimados são assim chamados quando obtemos um sinal corrompido por ruído e não o seu valor exato.

Note que o sinal $y(t, \rho)$ não é correlacionado com $\frac{\partial \widehat{y}(t, \rho)}{\partial \rho}$, uma vez que o ruído presente neles advém de experimentos diferentes, o mesmo vale para $u(t)$ e $\frac{\partial \widehat{u}(t, \rho)}{\partial \rho}$, portanto

$$E[\widehat{\nabla J(\rho)}] = \nabla J(\rho). \quad (32)$$

Considerando a equação anterior válida, torna-se possível o uso do algoritmo do gradiente devido a aproximação estocástica. Esta condição é a chave nas provas de convergência para mínimos locais como apresentada em (ROBBINS; MONRO, 1951) (HJALMARSSON et al., 1998).

A Hessiana para a função custo (7) é definida como a matriz

$$H = \frac{\partial^2 J(\rho)}{\partial \rho^2}, \quad (33)$$

levando em consideração as aproximações estocásticas para as estimativas das derivadas parciais, é possível escrever a Hessiana como

$$\frac{\partial^2 J(\rho)}{\partial \rho^2} = \frac{2}{N} \sum_{t=1}^N \left[[y_d(t) - y(t, \rho)] \frac{\partial^2 y(t, \rho)}{\partial \rho^2} + \frac{\partial y(t, \rho)}{\partial \rho} \frac{\partial y(t, \rho)}{\partial \rho}^T \right]. \quad (34)$$

Para obter a derivada parcial segunda de $y(t, \rho)$ e obter a Hessiana completa, é necessário o incremento de um experimento adicional no algoritmo IFT. Entretanto em quesitos práticos o uso da Hessiana é aconselhável quando a diferença $(y_d(t) - y(t, \rho))$ é relativamente pequena, ou seja, um custo muito baixo. Nestes casos é viável utilizar uma aproximação para a equação anterior (HJALMARSSON; GUNNARSSON; GEVERS, 1994), dada por

$$\widehat{H} = \frac{1}{N} \sum_{t=1}^N \frac{\partial \widehat{y}(t, \rho)}{\partial \rho} \frac{\partial \widehat{y}(t, \rho)}{\partial \rho}^T. \quad (35)$$

Como bibliografia adicional sobre o tema, é recomendado o trabalho de (SOLARI; GEVERS, 2004), nele é encontrado um caso particular de referência nula e sem o peso sobre o esforço de controle ($\lambda = 0$).

A próxima sessão sintetiza o algoritmo IFT em um passo-a-passo simplificado.

2.5 Algoritmo IFT

A cada iteração o algoritmo IFT é utilizado o conforme o seguinte procedimento:

- Coletar os sinais em malha fechada das saídas do controlador e do processo em condições normais de operação, ou seja, $r_1(t)$ como referência, e nomeá-los como $u_1(t)$ e $y_1(t)$, respectivamente.
- Computar o sinal $r_2(t) = r_1(t) - y_1(t)$.
- Excitar o sistema em malhada fechada com o sinal $r_2(t)$, novamente coletar os dados das saídas do controlador e do processo, entretanto os nomeando como $u_2(t)$ e $y_2(t)$.
- Calcular

$$\begin{aligned} \frac{\partial \widehat{y}(t, \rho_i)}{\partial \rho} &= \frac{1}{C(z, \rho_i)} \frac{\partial C(z, \rho_i)}{\partial \rho} y_2(t) \\ \frac{\partial \widehat{u}(t, \rho_i)}{\partial \rho} &= \frac{1}{C(z, \rho_i)} \frac{\partial C(z, \rho_i)}{\partial \rho} u_2(t) \end{aligned}$$

- Calcular a aproximação da Hessiana.

$$\widehat{H} = \frac{1}{N} \sum_{t=1}^N \frac{\partial \widehat{y}(t, \rho)}{\partial \rho} \frac{\partial \widehat{y}(t, \rho)}{\partial \rho}^T. \quad (36)$$

- Estimar o gradiente

$$\nabla \widehat{J} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y_1(t)] \frac{\partial \widehat{y}(t, \rho_i)}{\partial \rho} + \lambda \sum_{t=1}^N u_1(t) \frac{\partial \widehat{u}(t, \rho_i)}{\partial \rho}$$

- Calcular o incremento nos parâmetros utilizando um dos algoritmos:

$$\rho_{i+1} = \rho_i - \gamma_i \nabla \widehat{J}(\rho_i), \quad (37)$$

$$\rho_{i+1} = \rho_i - \gamma_i \widehat{H}^{-1} \nabla \widehat{J}(\rho_i) \quad (38)$$

2.6 Exemplo Numérico

O intuito desta sessão é demonstrar um exemplo a respeito do *Iterative Feedback Tuning*. A convergência dos parâmetros do controlador e do custo a cada iteração serão analisados com objetivo de esclarecer as nuances do algoritmo. Para este exemplo, a estrutura de controle escolhida é o controlador PID, pois é considerado um dos controladores industriais mais comuns por sua flexibilidade e popularidade (CORRIPIO, 2000).

A estrutura de controle escolhida é dada por

$$C(z) = k_p + k_i \frac{z}{z-1} + k_d \frac{z-1}{z} \quad (39)$$

com esta classe de controlador temos o vetor de parâmetros ρ e o vetor \overline{C} conforme

$$\rho = \begin{bmatrix} k_p \\ k_i \\ k_d \end{bmatrix}^T \quad \overline{C}(z) = \begin{bmatrix} 1 \\ z \\ \frac{z-1}{z-1} \\ \frac{z-1}{z} \end{bmatrix}. \quad (40)$$

Vamos assumir que o processo tem uma função de transferência $G(z)$ dada por

$$G(z) = \frac{0,5}{z-0,9}, \quad (41)$$

onde é adicionado um ruído branco gaussiano com média zero e variância de 10^{-3} .

Escolhemos como modelo de referência a função de transferência dada por

$$T_d(z) = \frac{0,25z^2 - 0,3z + 0,1}{z^3 - 1,65z^2 + 0,6z + 0,1}. \quad (42)$$

Utilizando a definição de $C_d(z)$ (16), obtemos o controlador desejado que alcança exatamente o objetivo $T_d(z)$,

$$C_d(z) = \begin{bmatrix} 0,2 \\ 0,1 \\ 0,2 \end{bmatrix}^T \begin{bmatrix} 1 \\ z \\ \frac{z-1}{z-1} \\ \frac{z-1}{z} \end{bmatrix} \quad (43)$$

em uma situação com ausência de ruído o algoritmo é capaz de encontrar o mínimo global da função custo, obtendo (43). Entretanto com a adição do ruído já é esperado que os parâmetros fiquem concentrados em uma circunferência com raio centrado no mínimo da função, como também obtido em (ECKHARD; BAZANELLA, 2012).

O processo é inicialmente controlado pelo valores arbitrados por

$$C(z, \rho^0) = \begin{bmatrix} 0, 1 \\ 0, 01 \\ 0, 02 \end{bmatrix}^T \begin{bmatrix} 1 \\ z \\ \frac{z-1}{z} \\ \frac{z-1}{z} \\ z \end{bmatrix}. \quad (44)$$

A Figura 2 demonstra o sinal desejado $y_d(t)$ e a saída do sistema $y_0(t)$ com as condições iniciais do controlador indicadas pela equação anterior. É definido para este experimento, um período de amostragem de 0,01s e como referência uma onda quadrada com amplitude unitária e período de 2s, dado por

$$r(t) = \text{square} \left(\frac{2\pi t}{2} \right). \quad (45)$$

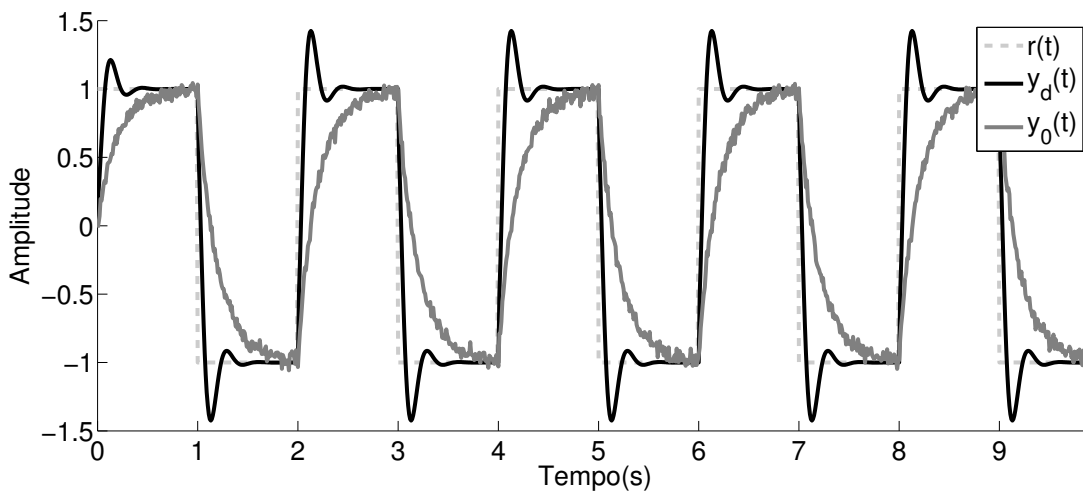


Figura 2: Saída $y_0(t, \rho)$ do processo com controlador inicial (44).

Como procedimento de convergência, utilizamos o método *Steepest Descent* (18) nas primeiras 5 iterações. O método de Quasi-Newton (19) é empregado nas últimas 15 iterações finalizando a convergência, sendo que neste é utilizada a aproximação da Hessiana demonstrada em (35). O passo escolhido para ambos os métodos de convergência segue o definido em (22) e (23), sendo a constante n escolhida com valor de 0,01. A Figura 3 demonstra a saída alcançada após a convergência dos parâmetros.

A Tabela (1) demonstra a convergência dos parâmetros do controlador e do seu respectivo custo a cada iteração realizada.

A Figura 4 exemplifica graficamente como os parâmetros do controlador k_p , k_i e k_d , movimentam-se no espaço \mathbb{R}^3 dos ganhos $C(z, \rho)$.

2.7 Sintonia IFT em sistemas com integrador

Em uma situação prática de controle onde o processo é desconhecido, é possível empregar o algoritmo IFT para realizar a sua sintonia, entretanto caso o projetista opte por uma estrutura de controle com integral, por exemplo PID, o projetista pode encontrar dificuldades caso o sistema sintonizado seja um sistema com integrador. Um sistema

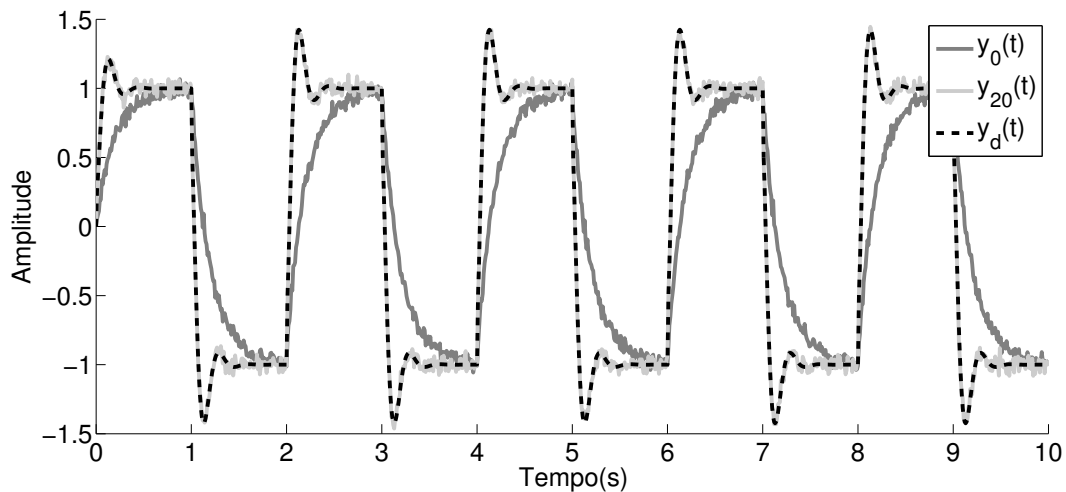


Figura 3: Saída $y_{20}(t)$ do processo com controlador obtido pelo algoritmo IFT.

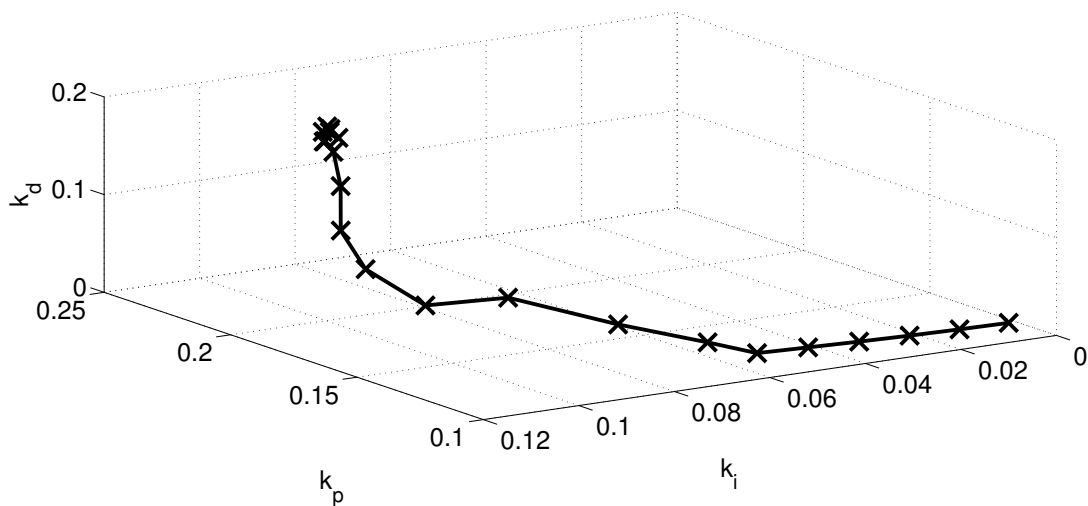


Figura 4: Convergência dos parâmetros do controlador.

com integrador é definido como sendo qualquer sistema em malha aberta que tem em sua própria dinâmica um fator integral. Nesta sessão são apresentados os problemas que o algoritmo IFT sofre em sua convergência ao realizar a sintonia de um sistema com integrador. Para explicar melhor os inconvenientes deste sistema, apresentamos um exemplo numérico da sua sintonia.

Para o exemplo em questão é escolhido o processo de primeira ordem em série com um integrador conforme

$$G(z) = \frac{z}{z-1} \frac{1}{z-0,6}. \quad (46)$$

note que este sistema segundo o princípio do modelo interno, não precisa de um integrador na estrutura de controle para obter o erro nulo em regime permanente.

Tendo como meta utilizar o algoritmo IFT em um sistema com integrador, é escolhido

Tabela 1: Convergência dos parâmetros do controlador e custo para o sistema exemplo.

Ite	$J(\rho)$	k_p	k_i	k_d	Método
1	0,2866	0,1000	0,0100	0,0200	SD
2	0,1505	0,1006	0,0200	0,0200	SD
3	0,0946	0,1015	0,0299	0,0200	SD
4	0,0629	0,1027	0,0399	0,0199	SD
5	0,0403	0,1042	0,0498	0,0198	SD
6	0,0254	0,1058	0,0596	0,0197	QN
7	0,0192	0,1211	0,0619	0,0191	QN
8	0,0119	0,1478	0,0665	0,0175	QN
9	0,0057	0,1775	0,0738	0,0243	QN
10	0,0029	0,1966	0,0810	0,0049	QN
11	0,0016	0,2054	0,0888	0,0400	QN
12	0,0014	0,2055	0,0940	0,0832	QN
13	0,0012	0,2007	0,0966	0,1348	QN
14	0,0011	0,2007	0,0981	0,1709	QN
15	0,0011	0,2009	0,1000	0,1825	QN
16	0,0012	0,2000	0,0997	0,1986	QN
17	0,0011	0,1985	0,1000	0,1947	QN
18	0,0012	0,2008	0,1003	0,1929	QN
19	0,0012	0,1981	0,0984	0,1877	QN
20	0,0012	0,1975	0,1004	0,1986	QN

o controlador PI parametrizado por

$$C(z) = \begin{bmatrix} k_p \\ k_i \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}.$$

São escolhidos os seguinte pontos iniciais de convergência para os parâmetros do controlador:

$$C(z) = \begin{bmatrix} 0, 1 \\ 0, 01 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}.$$

É escolhido um modelo de referência de primeira ordem com ganho em regime permanente unitário para o processo conforme

$$T_d(z) = \frac{0, 2}{z - 0, 8}, \quad (47)$$

utilizando a definição de $C_d(z)$ em (16), é possível calcular o controlador ideal para o sistema com integrador, o qual é dada por

$$C_d = \begin{bmatrix} 0, 08 \\ 0, 12 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix},$$

ou seja, o controlador desejado não pertence a classe \mathcal{C} dos possíveis controladores da estrutura escolhida, portanto não é possível atingir a resposta desejada $y_d(t)$ com esta estrutura de controle. Neste exemplo seria necessário um controlador PD ao contrario do

controlador PI escolhido, uma vez que o sistema não precisa do termo integral para obter erro em regime permanente.

O ruído não será adicionado neste exemplo numérico, porque temos a intenção de não adicionar efeitos indesejados que possam atrapalhar na análise do efeito do sistema com integrador na sintonia IFT. Igualmente ao realizado ao longo deste trabalho, é utilizado como referência um sinal em formato de onda quadrada com período de 2s e amplitude unitária, dada por

$$r(t) = \text{square} \left(\frac{2\pi t}{2} \right). \quad (48)$$

Como procedimento de convergência, utilizamos o método *Steepest Descent* (18) nas primeiras 10 iterações. O método de Quasi-Newton (19) é empregado nas últimas 10 iterações finalizando a convergência, sendo que neste utilizamos a aproximação da Hessiana demonstrada em (35). O passo escolhido para ambos os métodos de convergência segue o definido em (22) e (23), sendo a constante n escolhida com valor de 0,001.

Na Tabela 2 são demonstrados os dados referentes a convergência do algoritmo IFT atuando na sintonia do sistema com integrador. É notado que o termo integral, inicializado com o valor de 0,01, teve seu valor reduzido a cada iteração e convergindo para um valor aproximado de -0,0002. Este resultado é esperado, uma vez que o controlador desejado $C_d(z)$ é composto pela estrutura PD, o valor correto para o termo integral é zero e o algoritmo como demonstrado, realiza esta convergência para um valor próximo do mesmo.

Um estudo mais profundo sobre a sintonia utilizando IFT em sistemas com integrador é uma área que se encontra ainda em desenvolvimento e carece de trabalhos sobre o tema. Não é encontrada na bibliografia, maiores detalhes sobre como resolver ou contornar este problema de convergência para este caso particular de sistema.

Na Figura 5 são apresentadas as respostas iniciais, finais e desejadas para sistema. É notado que a resposta do sistema após a sintonia utilizando o algoritmo IFT fica mais próxima da resposta desejada $y_d(t)$, entretanto não alcança a convergência exata.

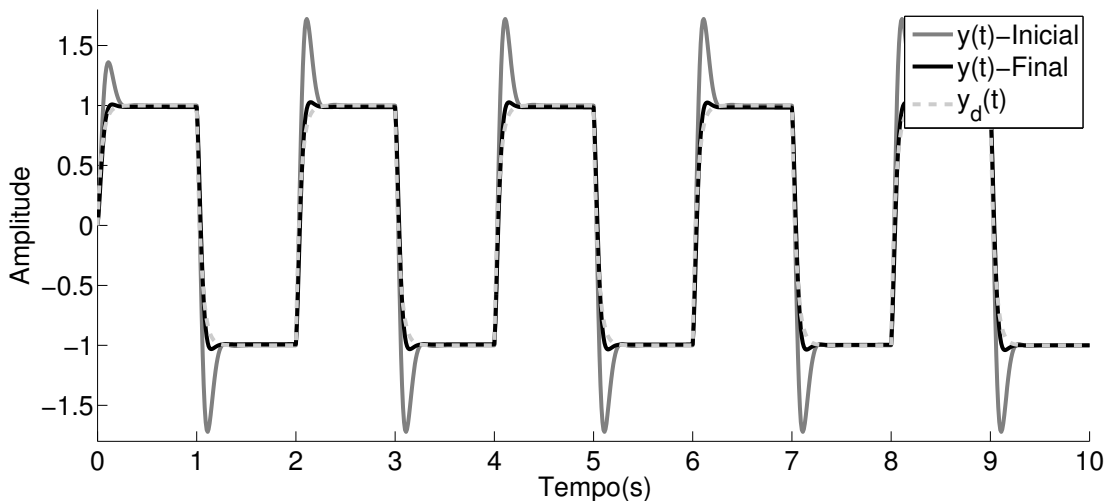


Figura 5: Resposta inicial, final e desejada para o sistema com integrador e sintonizado com IFT.

Tabela 2: Valor da função custo e convergência dos parâmetros do controlador, utilizando IFT em um sistema com integrador

Ite	$J(\rho)$	k_p	k_i	Método
1	0,0652	0,1000	0,0100	SD
2	0,0596	0,1001	0,0090	SD
3	0,0538	0,1002	0,0080	SD
4	0,0481	0,1002	0,0070	SD
5	0,0423	0,1003	0,0060	SD
6	0,0364	0,1003	0,0050	SD
7	0,0303	0,1003	0,0040	SD
8	0,0238	0,1004	0,0030	SD
9	0,0168	0,1004	0,0020	SD
10	0,0101	0,1004	0,0010	SD
11	0,0059	0,1003	0,0000	QN
12	0,0056	0,0990	-0,0001	QN
13	0,0050	0,0964	-0,0002	QN
14	0,0045	0,0932	-0,0002	QN
15	0,0041	0,0899	-0,0002	QN
16	0,0039	0,0872	-0,0002	QN
17	0,0038	0,0852	-0,0002	QN
18	0,0038	0,0841	-0,0002	QN
19	0,0038	0,0835	-0,0002	QN
20	0,0038	0,0832	-0,0002	QN

3 CASCADE ITERATIVE FEEDBACK TUNING - CIFT

O controle em formato cascata é uma estrutura muito conhecida para muitos projetistas, principalmente do ramo químico-industrial. Ela se diferencia dos demais sistemas de controle por ter em sua estrutura múltiplas realimentações negativas em formato seriado, também conhecida por malhas de controle. Quando a estrutura apresenta duas malhas, convencionalmente são nomeadas por malhas interna e externa de controle. Dada sua grande aplicabilidade (CORRIPIO, 2000), será utilizada esta estrutura de controle para o trabalho.

Comumente a estrutura de controle em cascata é usada para aumentar o desempenho do sistema a distúrbios na variável manipulada. Entretanto não é todo processo que tem melhoras significativas no seu desempenho ao utilizar esta estrutura de controle. A tomada de decisão pelo seu uso não é trivial, no trabalho de (KRISHNASWAMY et al., 1990) é realizado uma análise comparativa demonstrando quando a escolha por este controlador é recomendado.

Uma vez decidido pelo uso da estrutura cascata, realizar a sintonia dos controladores envolve a sintonia mútua de dois controladores, malha interna e externa, o que torna o projeto mais complexo comparado a uma única malha de controle. Comumente se realiza a sintonia da malha interna em uma primeira etapa para posteriormente realizar a sintonia da malha externa. Nesta sintonia a malha interna é projetada para ter a dinâmica mais rápida possível, uma tentativa de tornar sua dinâmica desprezível frente ao da malha externa.

Atualmente há muitos métodos complexos relacionados com a sintonia desta estrutura de controle tão comumente utilizada. Existem muitas variações de metodologias, desde métodos baseados em modelos ou baseados em dados, como nos trabalhos de (LEE; PARK; LEE, 1998), (ALFARO; VILANOVA; ARRIETA, 2009), (KAYA; TAN; ATHERTON, 2007), (LIU; GU; ZHANG, 2005) e (JENG; LIAO, 2013).

O objetivo geral do trabalho é utilizar o algoritmo IFT para realizar a sintonia dos controladores em formato cascata com duas malhas, interna e externa. Porém o algoritmo IFT clássico foi desenvolvido para sistemas com uma única malha de controle. O seu uso no formato atual ocasionaria dois principais problemas: primeiro, é necessário desacoplar as malhas e realizar a sintonia de forma independente, como o algoritmo IFT utiliza 2 experimentos para cada iteração, isto duplicaria o número de experimentos; segundo, seria necessária a escolha de um modelo de referência para a malha interna. Esta escolha é muito difícil de ser realizada, pois não há como ter o correto conhecimento sobre a interferência da dinâmica da malha interna no resultado final, uma vez que o processo é considerado desconhecido.

Tendo conhecimento dos dois problemas apresentados, não foi encontrada na bibliografia nenhuma alternativa para o uso do algoritmo IFT empregado em estruturas de

controle em formato cascata. O único trabalho encontrado foi o de (JENG; LIAO, 2013) que propõe um método de sintonia baseado em dados, o método utiliza somente 1 experimento, entretanto há a obrigatoriedade de definir ambos modelos de referência para as malhas interna e externa. Visto esta possibilidade de contribuição o presente trabalho propõe uma extensão que diminua o número de experimentos necessários em comparação ao IFT clássico, e realize a sintonia mútua dos controladores sem o uso de um modelo de referência para a malha interna, essa extensão definimos como *Cascade Iterative Feedback Tuning*.

3.1 Definições

3.1.1 Processo em cascata

Um processo com controle em formato cascata é caracterizado por ter controladores seriados realimentados, ou como também nomeados malhas de controle, jargão utilizado neste trabalho. De um modo geral os processos que optam por este tipo de controle buscam atenuação a distúrbios da variável manipulada (KRISHNASWAMY et al., 1990).

É adotado neste trabalho o sistema em formato cascata com duas malhas definidas como malha interna e malha externa, muito comum em plantas industriais. Vamos considerar os sistemas monovariáveis lineares discretos invariantes no tempo dados por

$$y_1(t) = G_1(z)u_1(t) + v_1(t) \quad (49)$$

$$y_2(t) = G_2(z)y_1(t) + v_2(t), \quad (50)$$

sendo $y_1(t)$ e $y_2(t)$ as saídas das malhas interna e externa do processo em cascata. $G_1(z)$ e $G_2(z)$ são as funções de transferência das malhas interna e externa, enquanto que $v_1(t)$ e $v_2(t)$ são os ruídos pertencentes as suas respectivas malhas. Consideram-se os ruídos sendo processos quasi-estacionários, resultados da filtragem de ruído branco por sistemas LTIs conforme

$$v_1(t) = H_1(z)e_1(t) \quad (51)$$

$$v_2(t) = H_2(z)e_2(t), \quad (52)$$

sendo ambos os ruídos $e_1(t)$ e $e_2(t)$ ruídos brancos gaussianos com variâncias de $\sigma_{e_1}^2$ e $\sigma_{e_2}^2$.

Todas as funções de transferência $G_1(z)$, $G_2(z)$, $H_1(z)$ e $H_2(z)$ são racionais, causais e desconhecidas, consideramos $H_1(\infty) = 1$ e $H_2(\infty) = 1$.

A Figura 6 exemplifica o sistema controlado na forma cascata.

O processo é controlado por controladores lineares discretos invariantes no tempo, sendo sua estrutura pertencente a uma dada classe \mathcal{C} de funções de transferência. Os sinais de controle gerados por estes controladores em malha fechada são definidos como

$$u_1(t) = C_1(z, \rho_1)(r_1(t) - y_1(t)) \quad (53)$$

$$u_2(t) = C_2(z, \rho_2)(r_2(t) - y_2(t)), \quad (54)$$

sendo $r_1(t)$ e $r_2(t)$ os sinais de referência para a malha interna e externa. A referência $r_1(t)$ é sintetizada através do controlador $C_2(z, \rho_2)$, ou seja, $u_2(t)$

$$r_1(t) = u_2(t) = C_2(z, \rho_2)(r_2(t) - y_2(t)). \quad (55)$$

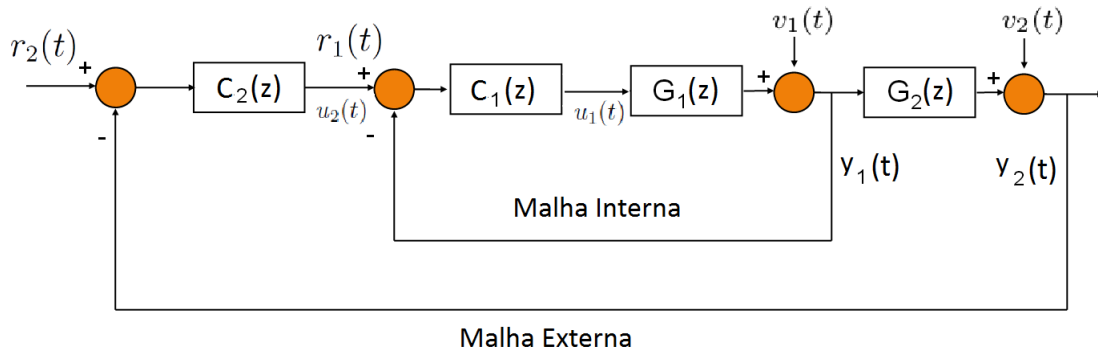


Figura 6: Processo com controladores em formato cascata.

O sinal de referência $r_2(t)$ é um sinal determinístico e quasi-estacionário, ou seja, o sinal de referência não é correlacionado com os ruídos do processo, portanto

$$E[r_2(t)e_1(\tau)] = 0 \quad \forall t, \tau$$

$$E[r_2(t)e_2(\tau)] = 0 \quad \forall t, \tau.$$

3.1.2 Estrutura do controlador

O conjunto de todos os controladores que podem ser implementados conforme uma dada estrutura é definido como classe de controlador \mathcal{C} sendo:

$$\mathcal{C} = \left\{ \begin{bmatrix} C_1(z, \rho_1) \\ C_2(z, \rho_2) \end{bmatrix} : \rho_1 \in P_1 \subseteq \mathbb{R}^{p_1}, \rho_2 \in P_2 \subseteq \mathbb{R}^{p_2} \right\}. \quad (56)$$

sendo ρ_1 e ρ_2 os parâmetros dos controladores a serem ajustados das malhas interna e externa. Observe que ambos os controladores não precisam ter a mesma estrutura.

Os controladores podem ser linearmente parametrizados por

$$C_1(z, \rho_1) = \rho_1^T \overline{C}_1(z) \quad (57)$$

$$C_2(z, \rho_2) = \rho_2^T \overline{C}_2(z), \quad (58)$$

$\overline{C}_1(z)$ e $\overline{C}_2(z)$ são os vetores contendo as funções de transferência pertencentes a classe \mathcal{C} .

Como exemplo de estrutura, é possível utilizar novamente o controlador PID para ambas as classes, esta escolha é tanto quanto propícia, pois além de muito utilizado em diversos processos simples, também é utilizado para controle de processos em cascata (CORRIPIO, 2000) (LEE; PARK; LEE, 1998). Os controladores PIDs são parametrizados para o processo em cascata como

$$C_i(z, \rho_i) = \begin{bmatrix} k_{p,i} \\ k_{i,i} \\ k_{d,i} \end{bmatrix}^T \begin{bmatrix} 1 \\ z \\ \frac{z-1}{z} \\ \frac{z-1}{z} \\ z \end{bmatrix}. \quad (59)$$

3.1.3 Processo em malha fechada

Utilizando os sinais das saídas $y_1(t)$ e $y_2(t)$ definidos em (49) e (50), é possível rescrever o processo em malha fechada como:

$$y_1(t, \rho_1, \rho_2) = T_1(z, \rho_1, \rho_2)(r_2(t) - v_2(t)) + \frac{S_1(z, \rho_1, \rho_2)}{G_2(z)}v_1(t) \quad (60)$$

$$y_2(t, \rho_1, \rho_2) = T_2(z, \rho_1, \rho_2)r_2(t) + S_2(z, \rho_1, \rho_2)v_2(t) + S_1(z, \rho_1, \rho_2)v_1(t), \quad (61)$$

onde $T_1(z, \rho_1, \rho_2)$ e $T_2(z, \rho_1, \rho_2)$ são as funções de transferência do processo em malha fechada entre as saídas $y_1(t)$ (49) e $y_2(t)$ (50) e a entrada $r_2(t)$. As funções de transferência são dadas por

$$T_1(z, \rho_1, \rho_2) = \frac{y_1(t)}{r_2(t)} = \frac{C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)}{1 + C_1(z, \rho_1)G_1(z) + C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)G_2(z)}, \quad (62)$$

$$T_2(z, \rho_1, \rho_2) = \frac{y_2(t)}{r_2(t)} = \frac{C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)G_2(z)}{1 + C_1(z, \rho_1)G_1(z) + C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)G_2(z)}. \quad (63)$$

$S_1(z, \rho_1, \rho_2)$ e $S_2(z, \rho_1, \rho_2)$, são respectivamente as funções de transferência do processo em malha fechada entre a saída $y_2(t)$ (50) e as entradas $v_1(t)$ (51) e $v_2(t)$ (52). As funções de transferência são dadas por

$$S_1(z, \rho_1, \rho_2) = \frac{y_2(t)}{v_1(t)} = \frac{G_2(z)}{1 + C_1(z, \rho_1)G_1(z) + C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)G_2(z)}, \quad (64)$$

$$S_2(z, \rho_1, \rho_2) = \frac{y_2(t)}{v_2(t)} = \frac{1 + C_1(z, \rho_1)G_1(z)}{1 + C_1(z, \rho_1)G_1(z) + C_1(z, \rho_1)C_2(z, \rho_2)G_1(z)G_2(z)}. \quad (65)$$

Vale salientar que por conveniência o sinal $y_1(t, \rho_1, \rho_2)$ é escrito em um formato que reutiliza as definições de malha fechada do sinal $y_2(t, \rho_1, \rho_2)$, algo que é percebido ao observar os termos pertencentes aos ruídos

$$T_1(z, \rho_1, \rho_2)v_2(t), \quad \frac{S_1(z, \rho_1, \rho_2)}{G_2(z)}v_1(t).$$

3.2 Problema de Otimização

O problema de otimização do IFT clássico apresentado na equação (9), trata da otimização do vetor de parâmetros do controlador. Entretanto, no controle cascata há dois vetores de parâmetros que constituem os controladores das malhas internas e externas. A otimização neste caso tem como objetivo ajustar em conjunto estes dois vetores independentes. O problema de otimização para controladores em cascata é definida como

$$\min_{\rho_1, \rho_2} J(\rho_1, \rho_2) \quad (66)$$

sendo ρ_1 e ρ_2 os vetores contendo os parâmetros dos controladores das malhas internas e externas.

O critério de desempenho J , escolhido para esta minimização é o mesmo utilizado na metodologia clássica, a norma \mathcal{H}_2 do erro entre o sinal desejado $y_d(t)$ e o obtido $y_2(t)$. É considerado que o sinal de interesse é a saída da malha externa $y_2(t)$. O critério de desempenho é dado por

$$J(\rho_1, \rho_2) = \|y_d(t) - y_2(t, \rho_1, \rho_2)\|^2 = \frac{1}{N} \sum_{t=1}^N [y_d(t) - y_2(t, \rho_1, \rho_2)]^2, \quad (67)$$

sendo $y_d(t)$ a saída desejada para a malha externa e N o tamanho da amostra coletada. Como demonstrado em (8), a saída desejada pode ser obtida ao utilizar um modelo de referência $T_d(z)$ para o sistema em malha fechada sendo

$$y_d(t) = T_d(z)r_2(t). \quad (68)$$

Para minimizar a função custo descrita nesta sessão, é possível utilizar inúmeros algoritmos como os já mencionados Gauss-Newton, Levenberg-Marquardt, Newton-Raphson, entretanto serão utilizados os algoritmos já definidos na sessão anterior *Steepest Descent* e Quasi-Newton. Na próxima sessão é demonstrado como são obtidos os gradientes da função custo para o caso cascata, a principal contribuição deste trabalho.

3.3 Gradiente da função custo $J(\rho_1, \rho_2)$

Para obter a estimativa do gradiente da função custo da equação (67), é proposta uma extensão baseada no método *Iterative Feedback Tuning*. Com esta extensão é possível realizar a convergência dos parâmetros ρ_1 e ρ_2 de ambos os controladores, malha interna e externa, e de forma simultânea.

Utilizamos a mesma norma \mathcal{H}_2 do método IFT clássico (67) como função custo, o gradiente é calculado derivando a função custo em relação aos parâmetros ρ_1 e ρ_2 conforme:

$$\frac{\partial J}{\partial \rho_1} = \frac{2}{N} \sum_{t=1}^N \left((y_d(t) - y_2(t, \rho_1, \rho_2)) \frac{\partial y_2(t, \rho_1, \rho_2)}{\partial \rho_1} \right) \quad (69)$$

$$\frac{\partial J}{\partial \rho_2} = \frac{2}{N} \sum_{t=1}^N \left((y_d(t) - y_2(t, \rho_1, \rho_2)) \frac{\partial y_2(t, \rho_1, \rho_2)}{\partial \rho_2} \right). \quad (70)$$

Para facilitar o entendimento vamos utilizar a seguinte notação

$$\rho = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad \nabla J = \begin{bmatrix} \frac{\partial J}{\partial \rho_1} \\ \frac{\partial J}{\partial \rho_2} \end{bmatrix}.$$

Igualmente ao caso do IFT clássico, para implementar o gradiente são necessárias as derivadas parciais em relação a cada variável dependente. Os próximos passos demonstram como são obtidas as derivadas parciais para o cálculo do gradiente, onde devido ao tamanho das equações envolvidas, optou-se por ocultar as dependências de ρ_1 , ρ_2 e z .

3.3.1 Derivada parcial de $y_2(t)$ em função de ρ_1

Utilizando as definições do processo em malha fechada no formato cascata (60)-(65), a derivada parcial de $y_2(t)$ em função de ρ_1 pode ser calculada como

$$\frac{\partial y_2(t)}{\partial \rho_1} = \frac{\partial}{\partial \rho_1} \{T_2 r_2(t) + S_1 v_1(t) + S_2 v_2(t)\}. \quad (71)$$

Para facilitar o entendimento, é apresentado a derivada parcial de cada parcela da

equação anterior. A derivada parcial de $T_2(z)$ em relação a ρ_1 é dada por

$$\frac{\partial T_2}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[\frac{C_1 C_2 G_1 G_2}{1 + C_1 G_1 + C_1 C_2 G_1 G_2} - \frac{1}{C_2} \frac{(C_1 C_2 G_1 G_2)(C_1 C_2 G_1)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} - \frac{(C_1 C_2 G_1 G_2)^2}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] r_2(t).$$

É possível reescrever a equação anterior como

$$\frac{\partial T_1}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[T_2 - \frac{1}{C_2} T_1 T_2 - T_2^2 \right] r_2(t). \quad (72)$$

Seguindo as demonstrações, a derivada parcial de $S_1(z)$ em relação a ρ_1 é dada por

$$\frac{\partial S_1}{\partial \rho_1} = -\frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[\frac{1}{C_2} \frac{G_2 (C_1 C_2 G_1)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} + \frac{(C_1 C_2 G_1 G_2)^2}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] v_1(t).$$

a qual pode ser reescrita como

$$\frac{\partial S_1}{\partial \rho_1} = -\frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[\frac{1}{C_2} S_1 T_1 + S_1 T_2 \right] v_1(t). \quad (73)$$

A terceira e última derivada parcial de $S_2(z)$ em relação ρ_1 é dada por

$$\frac{\partial S_2}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[\frac{1}{C_2} \frac{C_1 C_2 G_1}{1 + C_1 G_1 + C_1 C_2 G_1 G_2} - \frac{1}{C_2} \frac{(1 + C_1 G_1)(C_1 C_2 G_1)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} - \frac{(1 + C_1 G_1)(C_1 C_2 G_1 G_2)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] v_2(t).$$

a qual pode ser reescrita como

$$\frac{\partial S_2(t)}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[\frac{1}{C_2} (T_1 - T_1 S_2) - T_2 S_2 \right] v_2(t). \quad (74)$$

Relembrando as definições de malha fechada definidas em (60) e (61) para as saídas $y_1(t)$ e $y_2(t)$, e as parcelas correspondentes aos ruídos (64) e (65), é possível reescrever a derivada parcial de $y_2(t)$ em função de ρ_1 como

$$\frac{\partial y_2(t)}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[T_2(r_2(t) - y_2(t)) - \frac{1}{C_2} \left(T_1(y_2(t) - v_2(t)) \right) \right]. \quad (75)$$

A equação anterior demonstra que não é possível obter a derivada parcial de $y_2(t)$ em relação a ρ_1 de uma forma direta e analítica, porém é possível sintetizar este sinal realizando experimentos práticos no próprio processo, este é o principal fundamento do controle baseado em dados usado nesta extensão.

Para ter acesso a informação das parcelas $T_2(r_2(t) - y_2(t))$ e $T_1(y_2(t) - v_2(t))$ da equação anterior, basta identificar que de acordo com as definições (62) e (63), elas correspondem aos sinais $y_2^2(t)$ e $y_1^1 - y_1^2$ quando excitados com o sinal $r_2(t) - y_2^1(t)$, o sobrescrito significa que os dados foram coletados no experimento em questão. É importante

notar que estes novos sinais sintetizados estarão contaminados pelos ruídos $S_1(z)v_1^2(t)$ e $S_2(z)v_2^2(t)$, ruídos estes indesejados para o cálculo exato do gradiente, entretanto uma vez que o ruído tenha média zero, a estimativa não será polarizada.

Na sequência é demonstrado o correto procedimento dos experimentos e todos os sinais envolvidos.

- Primeiro experimento: A referência do sistema é $r_2(t) = r^1(t)$, neste experimento os sinais medidos $y_1(t)$ e $y_2(t)$ são dados por

$$y_1^1(t) = T_1 r^1(t) + \frac{S_1}{G_2} v_1^1(t) - T_1 v_2^1(t) \quad (76)$$

$$y_2^1(t) = T_2 r^1(t) + S_1 v_1^1(t) + S_2 v_2^1(t). \quad (77)$$

- Segundo experimento: A referência do sistema é $r_2(t) = r^1 - y_2^1(t)$, neste experimento os sinais medidos $y_1(t)$ e $y_2(t)$ são dados por

$$y_1^2(t) = T_1 (r^1(t) - y_2^1(t)) + \frac{S_1}{G_2} v_1^2(t) - T_1 v_2^2(t) \quad (78)$$

$$y_2^2(t) = T_2 (r^1(t) - y_2^1(t)) + S_1 v_1^2(t) + S_2 v_2^2(t). \quad (79)$$

Utilizando os sinais de $y_1^1(t)$, $y_1^2(t)$ e $y_2^2(t)$ definimos uma estimativa para a derivada parcial de $y_2(t)$ em função de ρ_1 como

$$\widehat{\frac{\partial y_2(t)}{\partial \rho_1}} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[y_2^2(t) - \frac{1}{C_2} (y_1^1(t) - y_1^2(t)) \right]. \quad (80)$$

A estimativa obtida com os dois experimentos alcança o valor exato do gradiente quando não há ruído. Na presença de ruído nossa estimativa é composta por uma parte correspondente ao valor exato do gradiente, mais uma parte correspondente aos ruídos oriundos dos experimentos realizados para obtenção do gradiente, como demonstrado na equação abaixo.

$$\widehat{\frac{\partial y_2(t)}{\partial \rho_1}} = \frac{\partial y_2(t)}{\partial \rho_1} + \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[S_2 v_2^2(t) + S_1 v_1^2(t) - \frac{1}{C_2} \left\{ \frac{S_1}{G_2} (v_1^1 - v_1^2) + T_1 v_2^2 \right\} \right],$$

é possível notar que a estimativa é corrompida tanto pelo ruído oriundo do primeiro experimento $v_1^1(t)$ e $v_2^1(t)$, tanto quanto do segundo experimento $v_1^2(t)$ e $v_2^2(t)$.

3.3.2 Derivada parcial de $y_2(t)$ em função de ρ_2

Será utilizado o mesmo procedimento adotado na sessão anterior para obter a derivada parcial de $y_2(t)$ em função de ρ_2 . Utilizando as definições do processo em malha fechada no formato cascata (60)-(65), a derivada parcial de $y_2(t)$ em função de ρ_2 pode ser calculada como

$$\frac{\partial y_2(t)}{\partial \rho_2} = \frac{\partial}{\partial \rho_2} \{ T_2 r_2(t) + S_1 v_1(t) + S_2 v_2(t) \}. \quad (81)$$

Para facilitar o entendimento é apresentado a derivada parcial de cada parcela da equação anterior. A derivada parcial de $T_2(z)$ em relação a ρ_2 é dada por

$$\frac{\partial T_2}{\partial \rho_2} = \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} \left[\frac{C_1 C_2 G_1 G_2}{1 + C_1 G_1 + C_1 C_2 G_1 G_2} - \frac{(C_1 C_2 G_1 G_2)^2}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] r_2(t).$$

a qual pode ser reescrita como

$$\frac{\partial T_2}{\partial \rho_2} = \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [T_2 - T_2^2] r_2(t). \quad (82)$$

A derivada parcial de $S_1(z)$ em relação a ρ_2 é dada por

$$\frac{\partial S_1}{\partial \rho_2} = -\frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} \left[\frac{G_2(C_1 C_2 G_1 G_2)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] v_1(t).$$

a qual pode ser reescrita como

$$\frac{\partial S_1}{\partial \rho_2} = -\frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [S_2 T_2] v_1(t). \quad (83)$$

A derivada da terceira e última derivada parcial de $S_2(z)$ em relação a ρ_2 é dada por

$$\frac{\partial S_2(t)}{\partial \rho_2} = -\frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} \left[\frac{(1 + C_1 G_1)(C_1 C_2 G_1 G_2)}{(1 + C_1 G_1 + C_1 C_2 G_1 G_2)^2} \right] v_2(t).$$

a qual pode ser reescrita como

$$\frac{\partial S_2(t)}{\partial \rho_2} = -\frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [S_1 T_2] v_2(t). \quad (84)$$

Do mesmo modo realizado na sessão anterior, usamos as definições de malha fechada definidas em (60) e (61) para as saídas $y_1(t)$ e $y_2(t)$, assim podemos reorganizar escrevendo a derivada parcial de $y_2(t)$ em função de ρ_2 como

$$\frac{\partial y_2(t)}{\partial \rho_2} = \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_1} \left[T_2(r_2(t) - y_2(t)) \right]. \quad (85)$$

A derivada parcial em relação a ρ_2 é bem semelhante em sua estrutura ao obtido com o método clássico IFT (26), a semelhança em sua estrutura não é a única, os mesmos fundamentos também são válidos para esta derivada parcial. A equação anterior demonstra que é possível sintetizar o sinal através do procedimento de experimentos. Propositamente é utilizado o mesmo experimento e sinais para obter a derivada parcial anterior (79), evitando o custo de novos experimentos, portanto a estimativa para derivada parcial de $y_2(t)$ em função de ρ_2 é definida como

$$\widehat{\frac{\partial y_2(t)}{\partial \rho_2}} = \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [y_2^2(t)]. \quad (86)$$

Igualmente ao caso anterior, a estimativa apresentada, obtida com os dois experimentos, alcança o exato valor para o correto gradiente quando não há presença de ruído como pode ser observado

$$\widehat{\frac{\partial y_2(t)}{\partial \rho_2}} = \frac{\partial y_2(t)}{\partial \rho_2} + \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [S_1 v_1^2(t) + S_2 v_2^2(t)]. \quad (87)$$

É possível notar que neste caso, a estimativa é corrompida somente pelo ruído oriundo do segundo experimento $v_1^2(t)$ e $v_2^2(t)$.

3.4 Estimativa $\frac{\partial \hat{J}}{\partial \rho}$

Com o auxílio de um segundo experimento foi demonstrado que é possível obter o gradiente exato de ambas derivadas parciais de $\frac{\partial y_2(t)}{\partial \rho}$ quando não há presença de ruído no processo. De acordo com a estimativa proposta, na presença de ruído em ambas as malhas, os gradientes calculados (80) e (86), serão corrompidos por estes ruídos, como observado no formato vetorial dos respectivos gradientes

$$\begin{bmatrix} \widehat{\frac{\partial y_2(t)}{\partial \rho_1}} \\ \widehat{\frac{\partial y_2(t)}{\partial \rho_2}} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_2(t)}{\partial \rho_1} + \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[S_2 v_2^2(t) + S_1 v_1^2(t) - \frac{1}{C_2} \left\{ \frac{S_1}{G_2} (v_1^1 - v_1^2) + T_1 v_2^2 \right\} \right] \\ \frac{\partial y_2(t)}{\partial \rho_2} + \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [S_1 v_1^2(t) + S_2 v_2^2(t)] \end{bmatrix}.$$

entretanto desde que os ruídos tenham média zero e sejam descorrelacionados a estimativa dos gradientes não será polarizada, sendo assim o valor estimado deste termo é dado por

$$E \begin{bmatrix} \widehat{\frac{\partial y_2(t)}{\partial \rho_1}} \\ \widehat{\frac{\partial y_2(t)}{\partial \rho_2}} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_2(t)}{\partial \rho_1} \\ \frac{\partial y_2(t)}{\partial \rho_2} \end{bmatrix}.$$

Com as estimativas $\widehat{\frac{\partial y_2(t)}{\partial \rho_1}}$ e $\widehat{\frac{\partial y_2(t)}{\partial \rho_2}}$, é possível calcular o gradiente da função custo conforme

$$\frac{\partial \hat{J}}{\partial \rho_1} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y_2^1(t, \rho_1, \rho_2)] \frac{\partial y_2(t, \rho_1, \rho_2)}{\partial \rho_1} \quad (88)$$

$$\frac{\partial \hat{J}}{\partial \rho_2} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y_2^1(t, \rho_1, \rho_2)] \frac{\partial y_2(t, \rho_1, \rho_2)}{\partial \rho_2}. \quad (89)$$

Analisando as equações anteriores, é possível notar que a estimativa da derivada parcial de $\widehat{\frac{\partial y_2(t)}{\partial \rho_1}}$ contém o termo indesejado $\frac{S_1}{G_2} (v_1^1)$ que corresponde ao ruído do primeiro experimento. Caso a equação (88) fosse utiliza o ruído presente no sinal $y_2^1(t)$ seria correlacionado com esta pequena parcela o que seria capaz de polarizar o estimador dos gradientes conforme

$$E \begin{bmatrix} \widehat{\frac{\partial J}{\partial \rho_1}} \\ \widehat{\frac{\partial J}{\partial \rho_2}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial \rho_1} + \frac{2}{N} \sum_{t=1}^N S_1 v_1^2(t) \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \frac{S_1}{C_2} \frac{1}{C_2} v_1^2(t) \\ \frac{\partial J}{\partial \rho_2} \end{bmatrix}.$$

Devido à polarização apresentada na função custo, é proposto o seguinte estimador para o gradiente da função custo:

$$\frac{\partial \hat{J}}{\partial \rho} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y_2^3(t)] \frac{\partial y_2(t)}{\partial \rho}. \quad (90)$$

A pequena mudança proposta é a adição de um experimento no procedimento, com a finalidade exclusiva de descorrelacionar o ruído do primeiro experimento presente na estimativa do gradiente, portanto o terceiro experimento deve ter o mesmo sinal de referência

usado no primeiro $r_2^3(t) = r_2^1(t)$, o experimento em questão será realizado da seguinte forma

- Terceiro experimento: a referência usado é $r_2^3(t) = r_2^1(t)$, neste experimento somente é necessário coletar o sinal de $y_2(t)$, todavia os sinais medidos $y_1(t)$ e $y_2(t)$ são dados por

$$y_1^3(t) = T_1(r_2^1(t)) + \frac{S_1}{G_2}v_1^3(t) - T_1v_2^3(t) \quad (91)$$

$$y_2^3(t) = T_2(r_2^1(t)) + S_1v_1^3(t) + S_2v_2^3(t). \quad (92)$$

Com a adição do terceiro experimento a equação (90) não tem nenhum sinal correlacionado, portanto

$$E \left[\frac{\partial \widehat{J}}{\partial \rho} \right] = \frac{\partial J}{\partial \rho}. \quad (93)$$

A próxima sessão sintetiza um passo a passo simplificado para o algoritmo *Cascade Iterative Feedback Tuning*.

3.5 Algoritmo CIFT

A cada iteração, o algoritmo CIFT é calculado conforme o seguinte procedimento:

- Coletar as saídas das malhas internas e externas do processo em condições normais de operação, ou seja, $r_2(t) = r^1(t)$, e nomeá-los como $y_1^1(t)$ e $y_2^1(t)$ respectivamente.
- Computar o erro do experimento anterior: $r_2^2(t) = r^1(t) - y_2^1(t)$.
- Excitar o processo em malha fechada com a referência calculada $r_2^2(t)$, novamente coletar as saídas das malhas internas e externas do processo, entretanto nomeando as como y_1^2 e y_2^2 respectivamente.
- Calcular

$$\frac{\partial \widehat{y_2(t)}}{\partial \rho_2} = \frac{1}{C_2} \frac{\partial C_2}{\partial \rho_2} [y_2^2(t)] \quad (94)$$

$$\frac{\partial \widehat{y_2(t)}}{\partial \rho_1} = \frac{1}{C_1} \frac{\partial C_1}{\partial \rho_1} \left[y_2^2(t) - \frac{1}{C_2} (y_1^2(t) - y_1^1(t)) \right] \quad (95)$$

- Calcular a aproximação da Hessiana.

$$\widehat{H} = \frac{1}{N} \sum_{t=1}^N \frac{\partial \widehat{y(t, \rho)}}{\partial \rho} \frac{\partial \widehat{y(t, \rho)}}{\partial \rho}^T \quad (96)$$

- Realizar um terceiro experimento com a mesma referência do primeiro, ou seja, $r_2(t) = r^1(t)$. Coletar o sinal da saída externa nomeando como $y_2^3(t)$
- Calcular o gradiente CIFT

$$\widehat{\nabla J(\rho_i)} = \frac{2}{N} \sum_{t=1}^N [y_d(t) - y_2^3(t)] \frac{\partial \widehat{y_2(t)}}{\partial \rho}$$

- Calcular o incremento nos parâmetros utilizando um dos algoritmos:

$$\rho_{i+1} = \rho_i - \gamma_i \nabla \widehat{J}(\rho_i) \quad (97)$$

$$\rho_{i+1} = \rho_i - \gamma_i \widehat{H}^{-1} \nabla \widehat{J}(\rho_i). \quad (98)$$

A demonstração da prova de convergência para o algoritmo CIFT não será abordada. Pois nos trabalhos de (BAZANELLA; CAMPESTRINI; ECKHARD, 2011), (ECKHARD; BAZANELLA, 2012) e (ECKHARD, 2008), a prova de convergência é apresentada e facilmente consultada para algoritmos que pertencem a metodologia de convergência baseada em gradiente, o caso deste algoritmo proposto.

3.6 Exemplo numérico

O principal foco desta sessão é demonstrar um exemplo numérico que sintetize todo o conhecimento sobre o algoritmo proposto *Cascade Iterative Feedback Tuning*. Para isto, é demonstrado a convergência dos parâmetros dos controladores das malhas internas e externas e o seu respectivo custo a cada iteração.

Primeiramente é exemplificado o uso do algoritmo CIFT para o caso com ausência de ruído. Neste exemplo é demonstrado que o algoritmo realiza a convergência dos parâmetros dos controladores para os valores exatos objetivados. Logo após são adicionados ruídos gaussianos em ambas as malhas, interna e externa, com finalidade de demonstrar que o algoritmo proposto alcança o objetivo mesmo na presença de ruído.

Propiciamente escolhemos para ambos controladores, malha interna e externa, controladores PIs, teremos portanto 4 parâmetros livres k_p e k_i de ambas as malhas em \mathbb{R}^4 . O controlador PID já foi definido e parametrizado em (59) na forma cascata de controle, entretanto é utilizado neste exemplo numérico o controlador PI, os controladores PIs para as malhas interna e externa são respectivamente definidos como

$$C_1(z) = \begin{bmatrix} k_{p,1} \\ k_{i,1} \end{bmatrix}^T \overline{C}(z) \quad C_2(z) = \begin{bmatrix} k_{p,2} \\ k_{i,2} \end{bmatrix}^T \overline{C}(z), \quad (99)$$

sendo $\overline{C}(z)$ o vetor contendo as funções de transferências parametrizadas por

$$\overline{C}(z) = \begin{bmatrix} 1 & \frac{z}{z-1} \end{bmatrix}^T.$$

O processo é cascadeado com dois sistemas de primeira ordem LTI $G_1(z)$ e $G_2(z)$ dados por

$$G_1(z) = \frac{1}{z-0,7} \quad G_2(z) = \frac{1}{z-0,8},$$

sendo $G_1(z)$ e $G_2(z)$ as funções de transferência das malhas internas e externas respectivamente.

Vale a pena salientar que o método proposto tem como objetivo alcançar o desempenho desejado para a malha externa, não sendo necessária assim a escolha de um modelo

de referência para a malha interna. É escolhida de forma arbitrária o seguinte modelo de referência dada por

$$T_d(z) = \frac{0,05z^2 - 0,085z + 0,036}{z^4 - 3,4z^3 + 4,34z^2 - 2,463z + 0,524}. \quad (100)$$

Os controladores desejados que alcançam exatamente o objetivo $T_d(z)$, para o caso sem ruído, são dados por

$$C_{d,1}(z) = \begin{bmatrix} 0,1 \\ 0,01 \end{bmatrix}^T \bar{C}(z) \quad C_{d,2}(z) = \begin{bmatrix} 0,5 \\ 0,1 \end{bmatrix}^T \bar{C}(z). \quad (101)$$

O processo é inicialmente controlado pelos controladores dados por

$$C_1(z) = \begin{bmatrix} 0,3 \\ 0,01 \end{bmatrix}^T \bar{C}(z) \quad C_2(z) = \begin{bmatrix} 0,3 \\ 0,03 \end{bmatrix}^T \bar{C}(z). \quad (102)$$

Definimos para ambos os exemplos, com e sem ruído, o período de amostragem de 0,01s e como referência uma onda quadrada com amplitude unitária e período de 2 segundos, dado por

$$r_2(t) = \text{square} \left(\frac{2\pi t}{2} \right). \quad (103)$$

Igualmente ao caso do algoritmo IFT clássico comentado, em uma situação com ausência de ruído o algoritmo é capaz de encontrar o mínimo global da função custo, isto porque o algoritmo CIFT obtém o gradiente exato da função custo, para o caso na ausência de ruído.

3.6.1 Processo sem ruído

A Figura 7 demonstra a saída desejada $y_d(t)$ e a saída inicial do processo $y_2(t)$ com as condições iniciais do controlador indicadas em (102).

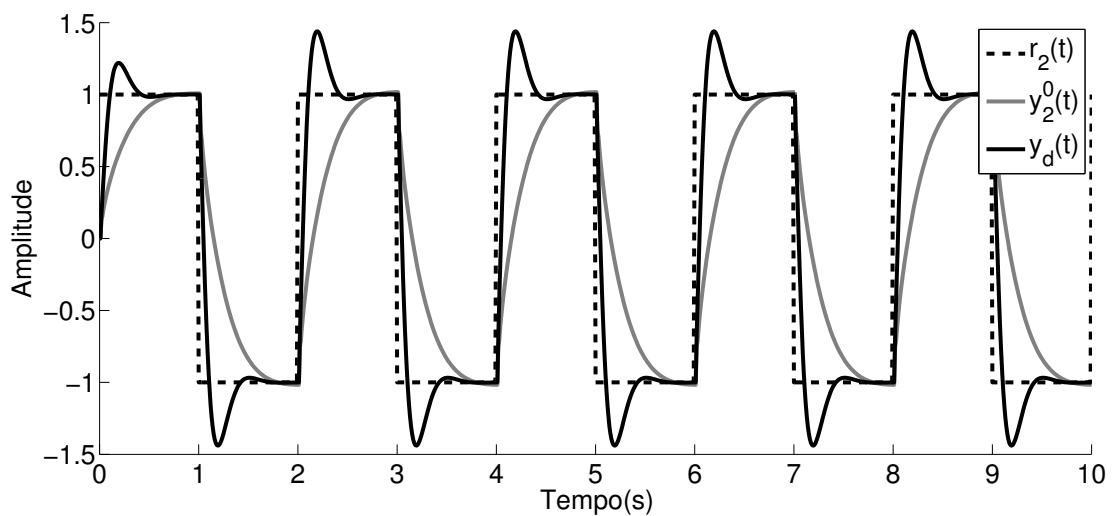


Figura 7: Saída desejada $y_d(t)$ e saída inicial $y_2(t)$, caso sem ruído, com os controladores iniciais (102).

Utilizamos o método *Steepest Descent* (18) como algoritmo de convergência nas primeiras 10 iterações. O algoritmo de Quasi-Newton (19) é empregado nas últimas 10

iterações finalizando a convergência, sendo que neste utilizamos a aproximação da Hessiana demonstrada em (35). O passo escolhido para ambos os algoritmos de convergência seguem o definido em (22) e (23), sendo a constante n escolhida com valor de 0,01. As Figuras 8 e 9 demonstram as saídas alcançadas após a convergência dos parâmetros.

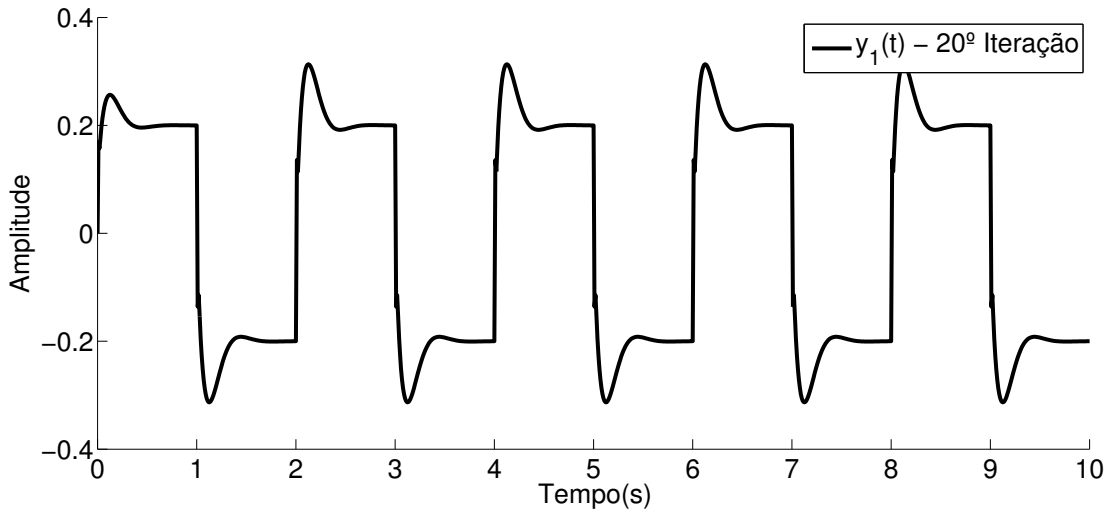


Figura 8: Saída final $y_1(t)$ do processo sem ruído, depois de realizadas 20 iterações do algoritmo CIFT.

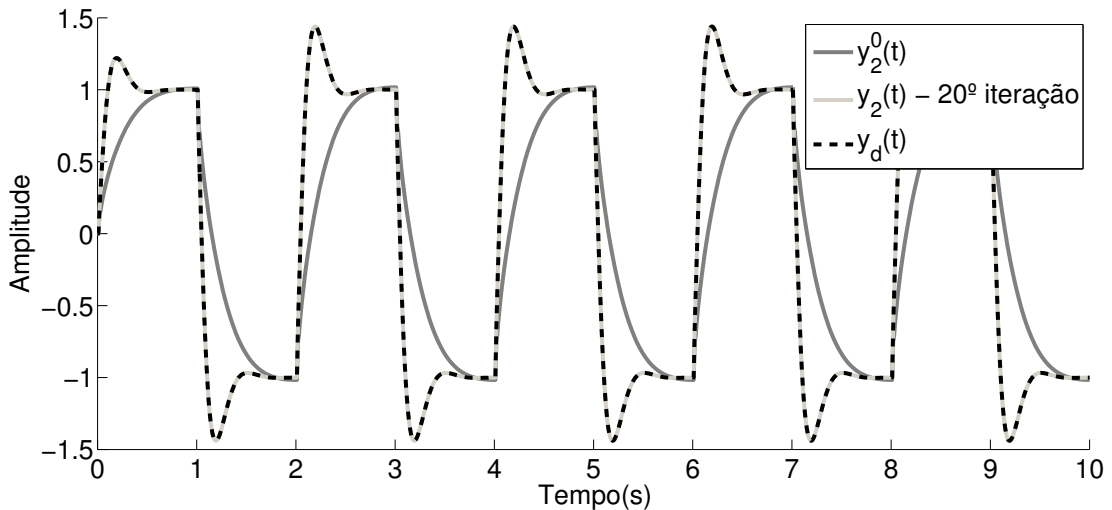


Figura 9: Saída final $y_2(t)$ do processo sem ruído, depois de realizadas 20 iterações do algoritmo CIFT.

A Tabela 3 demonstra a convergência dos parâmetros dos controladores e do seu respectivo custo a cada iteração realizada.

As Figuras 10 e 11 exemplificam graficamente como os parâmetros do controlador k_p e k_i , movimentam-se no espaço \mathbb{R}^2 dos ganhos $C(z, \rho)$.

3.6.2 Processo com ruído

O processo é contaminado por dois ruídos brancos $v_1(t)$ e $v_2(t)$, gaussianos com média zero e variância de $\sigma^2 = 10^{-4}$ descorrelacionados, ou seja, $E[v_1(t)v_2(t)] = 0$, de acordo

Tabela 3: Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o caso do exemplo sem ruído.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		$k_{p,1}$	$k_{i,1}$	$k_{p,2}$	$k_{i,2}$	
1	0,2187	0,3000	0,0300	0,3000	0,0100	SD
2	0,1274	0,2999	0,0319	0,3004	0,0198	SD
3	0,0757	0,2997	0,0339	0,3004	0,0296	SD
4	0,0467	0,2993	0,0360	0,3002	0,0394	SD
5	0,0329	0,2985	0,0374	0,2992	0,0492	SD
6	0,0278	0,2967	0,0346	0,2963	0,0582	SD
7	0,0248	0,2941	0,0259	0,2922	0,0575	SD
8	0,0237	0,2912	0,0286	0,2887	0,0660	SD
9	0,0226	0,2890	0,0210	0,2859	0,0606	SD
10	0,0218	0,2868	0,0272	0,2833	0,0676	QN
11	0,0208	0,2846	0,0199	0,2808	0,0617	QN
12	0,0156	0,2402	0,0202	0,3017	0,0626	QN
13	0,0060	0,1646	0,0185	0,3566	0,0658	QN
14	0,0025	0,1354	0,0152	0,3961	0,0731	QN
15	0,0010	0,1228	0,0129	0,4199	0,0817	QN
16	0,0003	0,1112	0,0113	0,4519	0,0899	QN
17	0,0001	0,1042	0,0104	0,4787	0,0959	QN
18	0,0000	0,1010	0,0101	0,4942	0,0990	QN
19	0,0000	0,1001	0,0100	0,4993	0,0999	QN
20	0,0000	0,1000	0,0100	0,5000	0,1000	QN

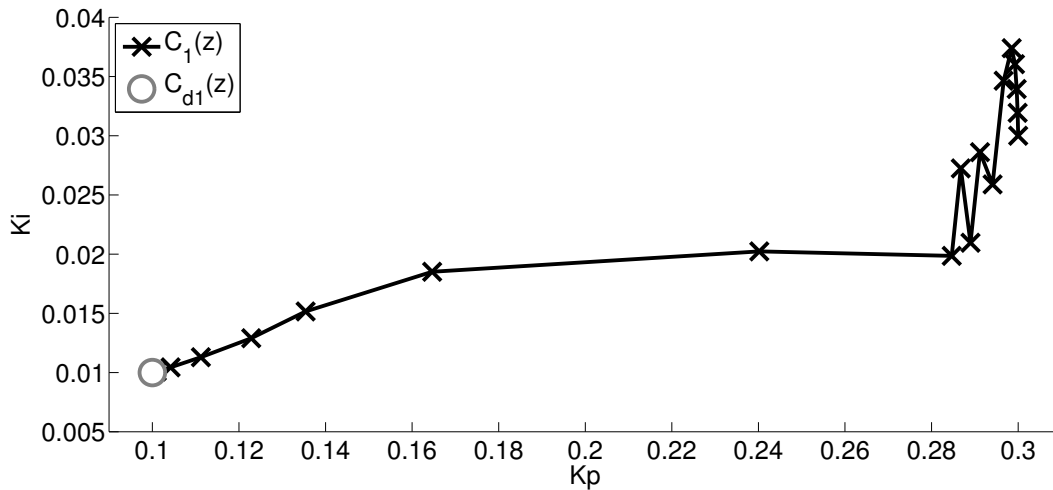


Figura 10: Convergência dos parâmetros do controlador $C_1(z)$ malha interna, caso sem ruído.

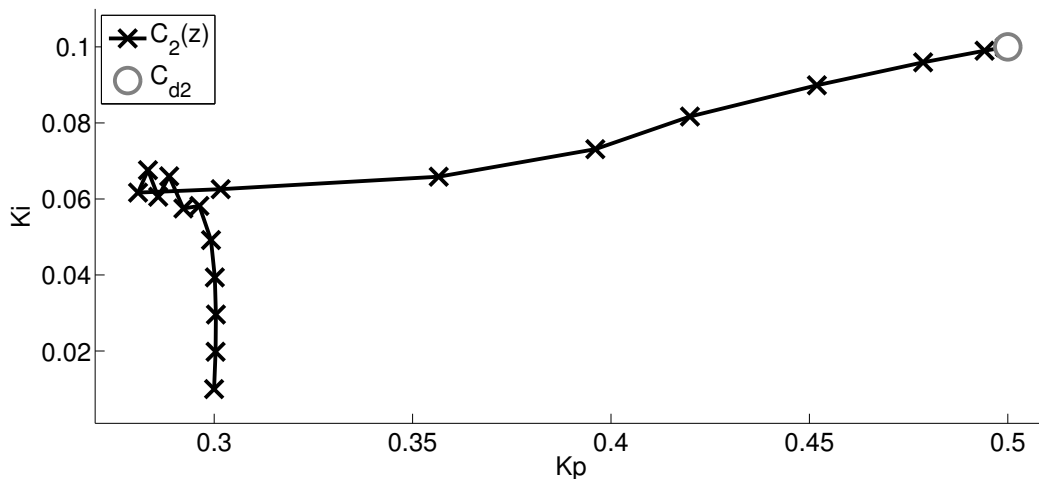


Figura 11: Convergência dos parâmetros do controlador $C_2(z)$ malha externa, caso com ruído.

ao demonstrado na Figura 6.

Utilizamos como procedimento de convergência o mesmo aplicado para o processo sem ruído, justamente para criar um paralelo entre as convergências do algoritmo. As Figuras 12 e 13 demonstram as saídas do processo após a convergência dos parâmetros.

A Tabela 4 demonstra a convergência dos parâmetros dos controladores e do seu respectivo custo a cada iteração realizada.

As Figuras 14 e 15 exemplificam graficamente como os parâmetros k_p e k_i dos controladores, movimentam-se no espaço \mathbb{R}^2 dos ganhos $C(z, \rho)$.

Ao término da convergência do algoritmo CIFT, é verificado que o valor da função custo alcança valores muito próximos à própria variância dos ruídos do processo, por isto é considerado que a convergência alcançou valores aceitáveis para os parâmetros dos controladores dados por

$$C_1(z) = \frac{0.1057z - 0.0957}{z - 1} \quad C_2(z) = \frac{0.4686z - 0.3693}{z - 1}. \quad (104)$$

Tabela 4: Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o caso do exemplo com ruído.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		$k_{p,1}$	$k_{i,1}$	$k_{p,2}$	$k_{i,2}$	
1	0,2198	0,3000	0,0300	0,3000	0,0100	SD
2	0,1270	0,2999	0,0319	0,3004	0,0198	SD
3	0,0759	0,2997	0,0339	0,3004	0,0296	SD
4	0,0469	0,2993	0,0360	0,3002	0,0394	SD
5	0,0334	0,2985	0,0374	0,2992	0,0492	SD
6	0,0280	0,2967	0,0348	0,2963	0,0582	SD
7	0,0253	0,2941	0,0261	0,2923	0,0574	SD
8	0,0236	0,2912	0,0285	0,2886	0,0658	SD
9	0,0229	0,2890	0,0209	0,2859	0,0605	SD
10	0,0220	0,2868	0,0272	0,2834	0,0675	SD
11	0,0209	0,2846	0,0198	0,2808	0,0616	QN
12	0,0175	0,2746	0,0203	0,2721	0,0622	QN
13	0,0109	0,2582	0,0203	0,2560	0,0639	QN
14	0,0057	0,2344	0,0185	0,2456	0,0676	QN
15	0,0023	0,2064	0,0173	0,2489	0,0710	QN
16	0,0011	0,1779	0,0142	0,2697	0,0776	QN
17	0,0006	0,1526	0,0135	0,3099	0,0819	QN
18	0,0004	0,1283	0,0114	0,3670	0,0895	QN
19	0,0003	0,1147	0,0107	0,4208	0,0948	QN
20	0,0001	0,1057	0,0100	0,4686	0,0993	QN

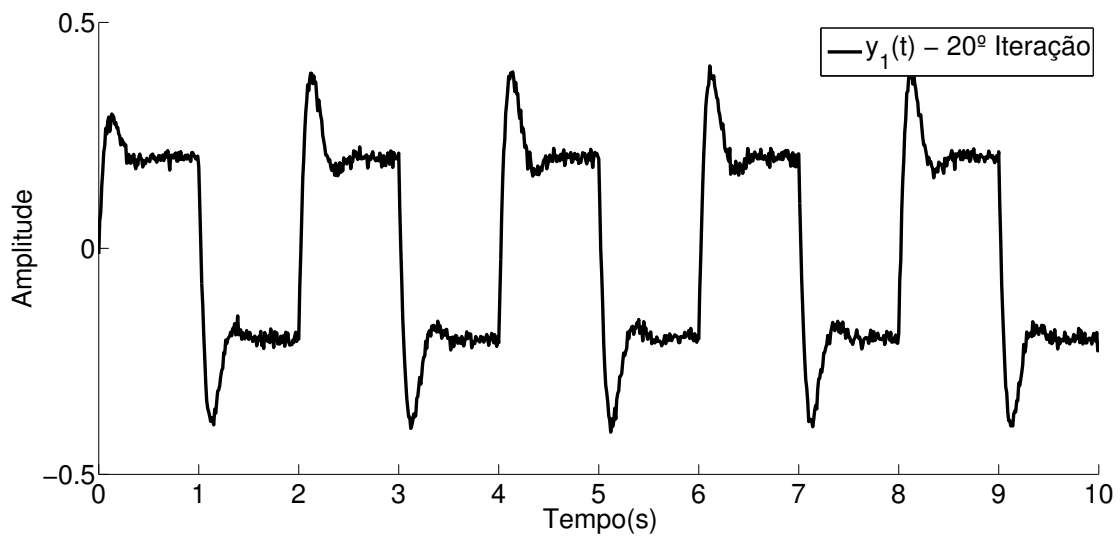


Figura 12: Saída final $y_1(t)$, malha interna do processo com ruído, depois de realizadas 20 iterações do algoritmo CIFT.

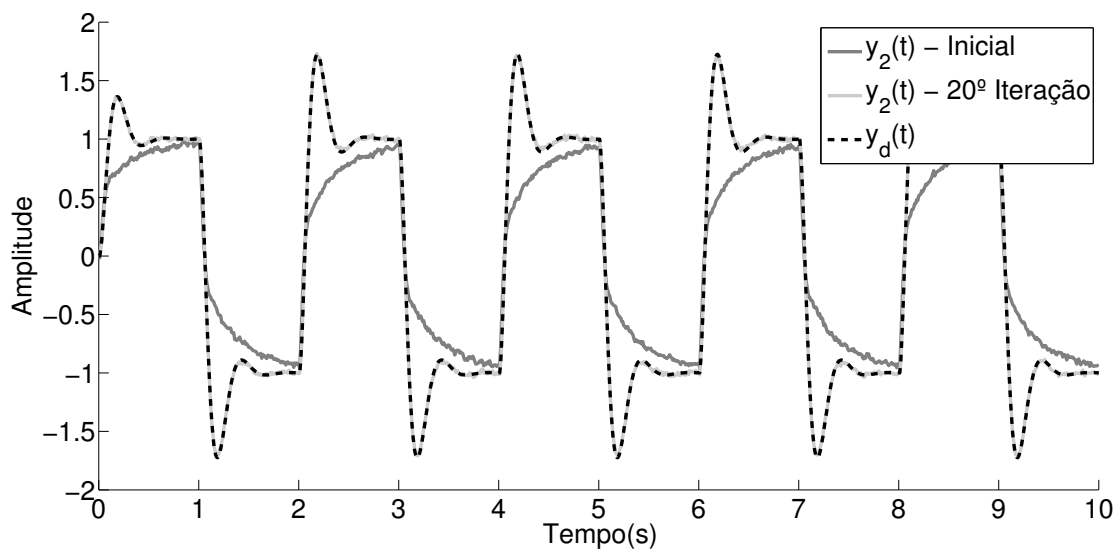


Figura 13: Saída final $y_2(t)$, malha externa do processo com ruído, depois de realizadas 20 iterações do algoritmo CIFT.

3.7 Comparativo entre os algoritmos IFT e CIFT

Nesta sessão é realizada uma comparação entre a metodologia clássica IFT e o método CIFT proposto. Para isto, é empregado em ambos os métodos o mesmo processo dinâmico e demonstrado as diferenças encontradas nas técnicas de sintonia. Juntamente são apresentadas as respostas finais e convergências para ambos algoritmos.

As funções de transferência do processo escolhido são dadas por

$$G_1(z) = \frac{1}{z - 0,8} \quad G_2(z) = \frac{1}{z - 0,9},$$

os processos são contaminados por dois ruídos brancos $v_1(t)$ e $v_2(t)$, gaussianos com média zero e variância de $\sigma^2 = 10^{-4}$.

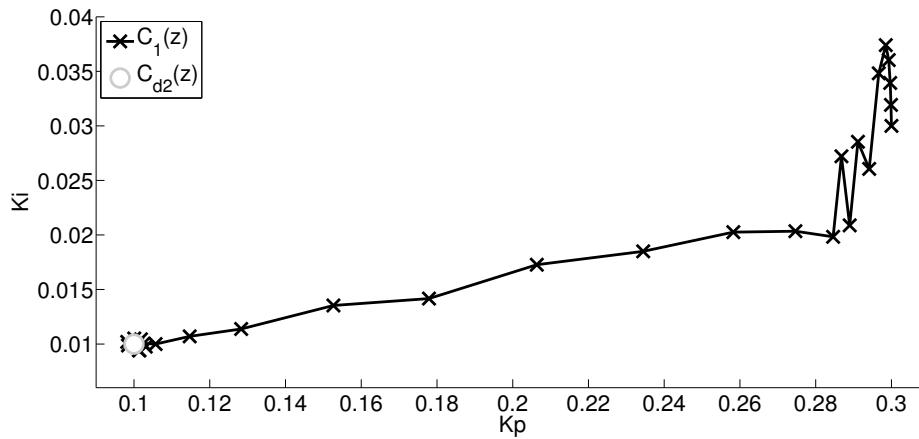


Figura 14: Convergência dos parâmetros do controlador $C_1(z)$ malha interna, caso com ruído.

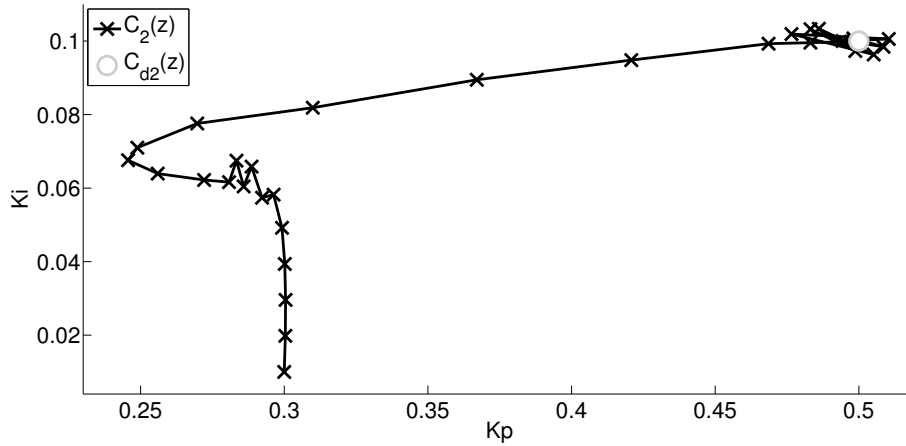


Figura 15: Convergência dos parâmetros do controlador $C_2(z)$ malha externa, caso com ruído.

É definido para ambos controladores $C_1(z, \rho_1)$ e $C_2(z, \rho_2)$ a estrutura do tipo PI dada por

$$\rho = \begin{bmatrix} k_p \\ k_i \end{bmatrix} \quad \bar{C}(z) = \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}. \quad (105)$$

É arbitrado como modelo de referência a função de transferência dada por

$$T_d(z) = \frac{0,02403z^2 - 0,0394z + 0,016}{z^4 - 3,43z^3 + 4,431z^2 - 2,556z + 0,556},$$

os controladores que alcançam o correspondente modelo de referência $T_d(z)$ são calculados como:

$$C_{d1}(q) = \begin{bmatrix} 0,2 \\ 0,07 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix} \quad (106)$$

$$C_{d2}(q) = \begin{bmatrix} 0,08 \\ 0,009 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}. \quad (107)$$

O processo em cascata é inicialmente controlado pelos controladores escolhidos como:

$$C_1(z) = \begin{bmatrix} 0,1 \\ 0,05 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix} \quad (108)$$

$$C_2(z) = \begin{bmatrix} 0,05 \\ 0,004 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}. \quad (109)$$

Como referência $r_2(t)$, escolhemos novamente uma onda quadrada com amplitude unitária e período de 2s definida em (103). A Figura 16 demonstra a saída obtida ao utilizar os controladores iniciais escolhidos (108) e (109) aplicando a referência descrita.

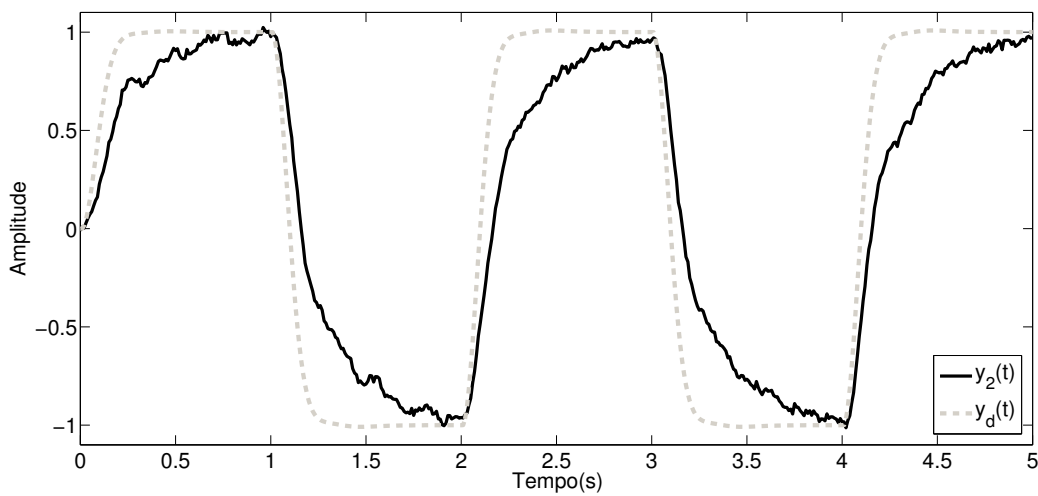


Figura 16: Saída do processo $y_2(t)$ com os controladores iniciais (108) e (109) (linha contínua); resposta desejada $y_d(t)$ (linha pontilhada).

O procedimento de convergência adotado para esta comparação é semelhante ao utilizado nos exemplos anteriores. Utilizamos o método *Steepest Descent* (18) como algoritmo de convergência nas primeiras iterações. O algoritmo de Quasi-Newton (19) é empregado nas últimas para finalizar a convergência, sendo que neste utilizamos a aproximação da Hessiana demonstrada em (35). O passo escolhido para ambos os algoritmos de convergência seguem o definido em (22) e (23), sendo a constante n escolhida com valor de 0,001.

3.7.1 Iterative Feedback Tuning

Utilizamos o método clássico IFT para realizar a sintonia de ambos controladores, malhas interna e externa. Para esta metodologia é necessário executar a sintonia por etapas. Primeiramente é ajustado o controlador da malha interna $C_1(z, \rho_1)$. Esta etapa tem uma dificuldade adicional para o projeto, pois tem a necessidade de escolher um modelo de referência para a malha interna. Na segunda etapa o controlador $C_2(z, \rho_2)$ da malha externa é sintonizado, finalizando o procedimento.

Como a malha interna é sintonizada desacoplada do restante do processo, é necessário definir um correto procedimento de sintonia para esta malha em especial. O sinal de referência $r_1(t)$ utilizado é o mesmo sinal definido em (103). É escolhido para o experimento uma função de transferência unitária com um atraso como modelo de referência $T_d(z)$,

Tabela 5: Convergência dos parâmetros da malha interna, comparativo usando IFT.

Int	$C_1(z)$		Método
	$k_{p,1}$	$k_{i,1}$	
1	0,1000	0,0500	SD
2	0,1003	0,0509	SD
3	0,1006	0,0519	SD
4	0,1010	0,0528	SD
5	0,1013	0,0538	SD
6	0,1017	0,0547	QN
7	0,1088	0,0553	QN
8	0,1233	0,0564	QN
9	0,1458	0,0583	QN
10	0,1771	0,0609	QN
20	0,7718	0,1698	QN
30	0,7982	0,2002	QN
40	0,7990	0,2008	QN

ou seja, a malha interna é projetada para ser a mais rápida possível como forma de tornar a dinâmica desta malha desprezível em comparação a externa. Este procedimento é uma forma clássica de sintonia para estruturas em cascata e recomendada em (CORRIPIO, 2000). A Tabela 5 demonstra a convergência dos parâmetros do controlador $C_1(z, \rho_1)$. Ao final desta convergência obtemos o controlador PI dado por

$$C_1(z, \rho_1) = \begin{bmatrix} 0,7990 \\ 0,2008 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix} \quad (110)$$

A Tabela 6 demonstra a convergência dos parâmetros do controlador $C_2(z, \rho_2)$. Ao final desta convergência obtemos como resultado o controlador PI dado por

$$C_2(z, \rho_2) = \begin{bmatrix} 0,0571 \\ 0,0097 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix} \quad (111)$$

3.7.2 Cascade Iterative Feedback Tuning

Nesta metodologia realizamos a sintonia de ambos controladores das malhas interna e externa utilizando o algoritmo proposto *Cascade Iterative Feedback Tuning*. A convergência dos parâmetros de ambos controladores $C_1(z, \rho_1)$ and $C_2(z, \rho_2)$ são demonstrados na Tabela 7.

Ao final de 40 iterações do algoritmo CIPT obtemos os controladores dados por

$$C_1(z, \rho_1) = \begin{bmatrix} 0,1986 \\ 0,0691 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}$$

$$C_2(z, \rho_2) = \begin{bmatrix} 0,0797 \\ 0,0090 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \end{bmatrix}.$$

Comparando a Tabela 6 e 7, é possível verificar que o algoritmo proposto obtém um melhor desempenho, ao comparar suas respectivas funções custo $J(\rho)$. O algoritmo alcança um valor 9 vezes menor que o método convencional que realiza a sintonia das malhas interna e externa de forma separada.

Tabela 6: Convergência dos parâmetros da malha externa, comparativo usando IFT.

Int	$J(\rho)$	$C_2(z)$		Método
		$k_{p,2}$	$k_{i,2}$	
1	0,0890	0,0500	0,0040	SD
2	0,0511	0,0500	0,0050	SD
3	0,0280	0,0500	0,0060	SD
4	0,0139	0,0501	0,0070	SD
5	0,0066	0,0501	0,0080	SD
6	0,0032	0,0501	0,0090	QN
7	0,0031	0,0504	0,0090	QN
8	0,0031	0,0508	0,0091	QN
9	0,0028	0,0515	0,0091	QN
10	0,0026	0,0523	0,0092	QN
20	0,0022	0,0570	0,0097	QN
30	0,0022	0,0570	0,0097	QN
40	0,0022	0,0571	0,0097	QN

Tabela 7: Convergência dos parâmetros da malha interna e externa, comparativo usando CIFT.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		$k_{p,1}$	$k_{i,1}$	$k_{p,2}$	$k_{i,2}$	
1	0,1071	0,1000	0,0500	0,0500	0,0040	SD
2	0,0651	0,1000	0,0500	0,0500	0,0050	SD
3	0,0401	0,1000	0,0500	0,0501	0,0060	SD
4	0,0284	0,1000	0,0501	0,0502	0,0070	SD
5	0,0246	0,1000	0,0502	0,0503	0,0080	SD
6	0,0240	0,1000	0,0509	0,0511	0,0078	QN
7	0,0042	0,1046	0,0533	0,0777	0,0081	QN
8	0,0023	0,1123	0,0574	0,0826	0,0084	QN
9	0,0016	0,1206	0,0657	0,0773	0,0087	QN
10	0,0013	0,1257	0,0698	0,0766	0,0087	QN
20	0,0004	0,1912	0,0714	0,0789	0,0090	QN
30	0,0004	0,1971	0,0693	0,0792	0,0090	QN
40	0,0004	0,1986	0,0691	0,0797	0,0090	QN

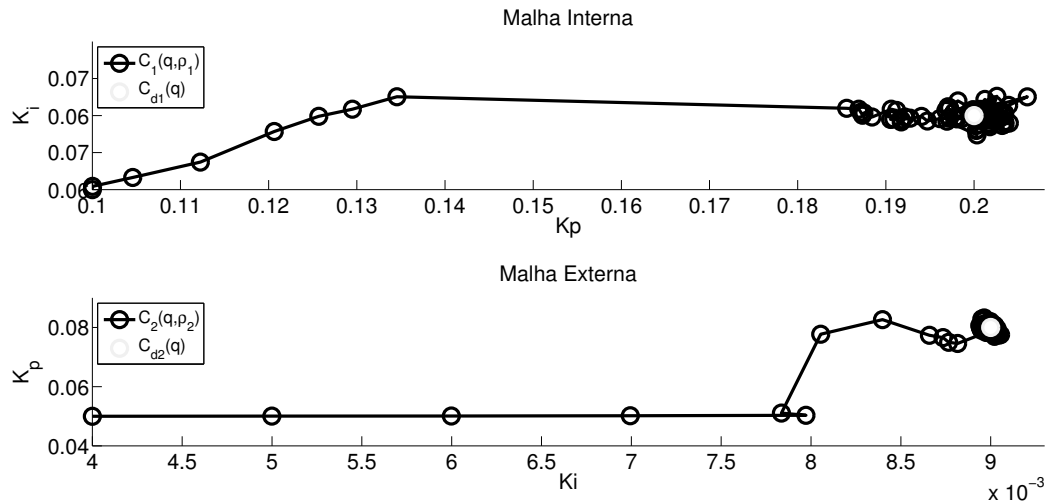


Figura 17: Demonstração gráfica da convergência dos controladores $C_1(z, \rho_1)$ e $C_2(z, \rho_2)$ utilizando o algoritmo CIFT.

A Figura 17 demonstra graficamente a convergência para ambos controladores.

É possível analisar observando os gráficos da Figura 17 e as tabelas com as convergências 6 e 7 que o após 40 iterações o algoritmo já alcança valores bem próximos do seu mínimo. Entretanto percebemos que depois de 20 iterações o método já alcança uma redução considerável na função custo e um bom desempenho como resposta final, sendo este o ponto aconselhável para finalizar o experimento. Outro detalhe importante que após somente 8 iterações o algoritmo proposto obtém um custo menor em comparação ao método IFT clássico.

A resposta final da saída $y_2(t)$ com os controladores obtidos ao término de cada metodologia é apresentada na Figura 18.

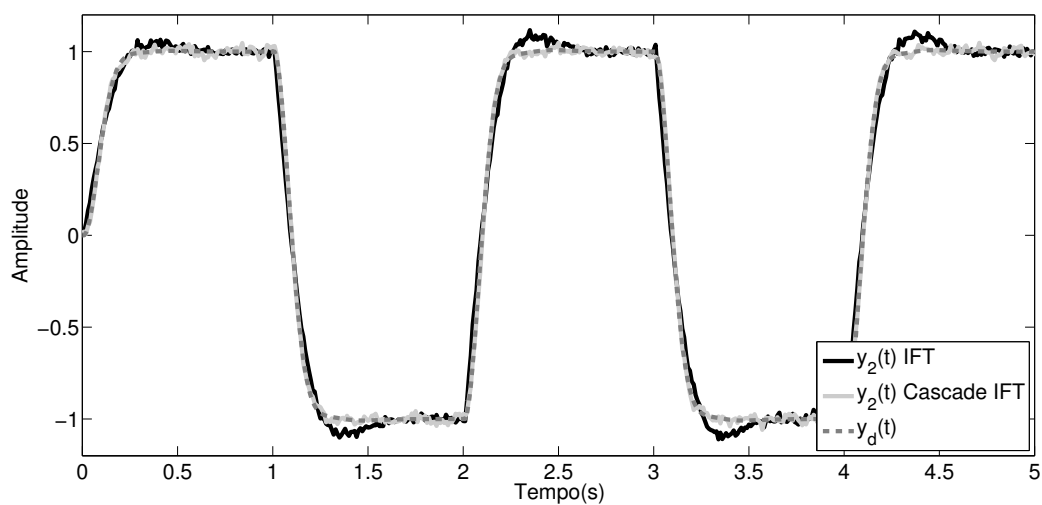


Figura 18: Comparação entre as saída $y_2(t)$ obtida com os métodos IFT e CIFT.

4 QUADRICÓPTERO

Veículos aéreos não tripulados (VANTs) ou em inglês *Unmanned Aerial Vehicle* (UAVs), tiveram um aumento considerável em seu uso nos últimos anos principalmente pela redução do custo envolvido na fabricação e pela sua popularidade nas mais diversas áreas de atuação (GUPTÉ; MOHANDAS; CONRAD, 2012). Seja para fim civil como: supervisão de rachaduras ou trincas em fachadas de prédios, fotos aéreas privilegiadas, esportes em geral, uso em plantações e exploração aérea diversa. Seja para fins militares como: monitoramento de queimadas, patrulha de fronteira, ataques aéreos, uso em situações de extremo risco para o piloto e monitoramento aéreo diverso (PURI, 2005).

Vamos utilizar neste trabalho o VANTs chamado de quadricóptero, quadcoptero ou quadrotor, este VANT é bem popular entre os praticantes de aeromodelismo dada a sua simplicidade construtiva e baixo custo. Entretanto o controle utilizado nesta aplicação não é algo trivial por se tratar de um sistema não-linear e multivariável, devido a isto é utilizado como aplicação teste para muitas propostas de sistemas controle como em (BOUABDALLAH; NOTH; SIEGWART, 2004), (ALTUŽ; OSTROWSKI; MAHONY, 2002), (MADANI; BENALLEGUE, 2006), (COZA; MACNAB, 2006) e (RAFFO; ORTEGA; RUBIO, 2010). Comumente o controle utilizado nos quadricópteros não profissionais e até em alguns ditos profissionais, é o clássico controle linear PID organizado em formato cascata, onde a malha interna é responsável pelo controle da velocidade angular e a malha externa é responsável pelo controle do ângulo do quadricóptero.

O grande objetivo almejado deste trabalho é propor uma metodologia para realizar a sintonia dos controladores de um quadricóptero em uma situação real de operação, fazendo uso do algoritmo *Cascade Iterative Feedback Tuning* e utilizando experimentos práticos para o devido fim. Como segundo objetivo, é almejado verificar o comportamento do algoritmo proposto atuando em sistemas não-lineares, juntamente validando se o algoritmo é recomendado para esta aplicação em específico. Já é conhecido que o algoritmo IFT linear utilizado em sistemas não-lineares introduz um erro na convergência que depende da dinâmica do processo (HJALMARSSON, 2002). É esperado que o CIFT herde este mesmo comportamento para sistemas não-lineares, uma vez que o algoritmo é baseado no clássico IFT.

A contribuição deste trabalho surgiu da necessidade de utilizar o IFT clássico na sintonia do controle em formato cascata do quadricóptero. De acordo ao já apresentado no Capítulo 3.5, o algoritmo IFT apresenta limitações para esta estrutura de controle, como necessidade da escolha do modelo de referência para a malha interna e uso de 4 iteração por ciclo. Visto estas limitações o trabalho se voltou a obter alguma solução para os mesmos, obtendo assim o *Cascade Iterative Feedback Tuning*.

As sessões que se seguem tem como objetivo tentar esclarecer a dinâmica do quadricóptero através da obtenção de um modelo simplificado do mesmo, com o intuito de

demonstrar o desafio de controle enfrentado. Por fim são demonstrados algumas simulações realizadas com o modelo obtido, utilizando controladores lineares PIDs em ambas as malhas, algo também utilizado nos experimentos práticos.

4.1 Cinemática

Esta sessão é responsável por descrever os vários sistemas de coordenadas que o quadricóptero esta atuando, coordenadas estas que descrevem seu posicionamento e orientação no espaço \mathbb{R}^3 cartesiano. Grande parte da informação contida nesta sessão tem como base o material de (BEARD, 2008), (STEVENS; LEWIS, 2003) e (CRAIG, 2005).

4.1.1 Matriz de rotação

Transformar um sistema de coordenadas em outro comumente requer duas operações básicas: rotação e translação. Translação é uma operação linear simples de ser realizada, por exemplo

$$p^a = T p^b \quad (112)$$

onde T é a matriz de translação. A equação basicamente descreve que para representar o ponto p^b no sistema de coordenadas a , devemos aplicar a matriz de translação T . Rotação é uma transformação bem similar em sua estrutura, por exemplo

$$p^a = R(\zeta) p^b \quad (113)$$

onde $R(\zeta)$ é conhecida como matriz de rotação. A equação basicamente descreve que para rotacionar o ponto p^b em ζ radianos, devemos aplicar a matriz de rotação $R(\zeta)$. A matriz de rotação tem um papel fundamental para o desenvolvimento das equações de movimento e por isto vamos dar maior atenção a sua explicação.

A matriz $R(\zeta)$ realiza uma rotação entorno de um dado eixo de coordenadas, por exemplo a Figura 19 demonstra uma rotação entorno da do eixo Z. Obter a matriz de rotação demanda um conhecimento algébrico e geométrico básico, algo que não será demonstrado neste trabalho.

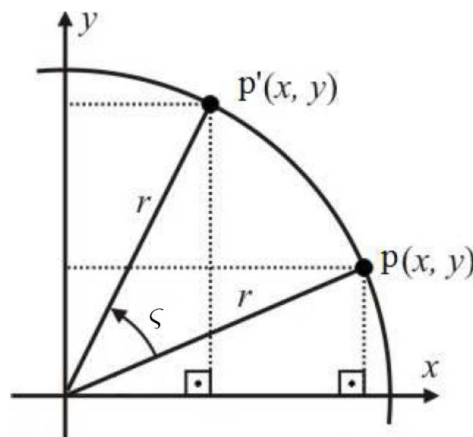


Figura 19: Exemplo rotação de um dado ponto p .

Definimos as matrizes de rotação em torno do eixo X, Y e Z como:

- Rotação em torno do eixo X

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \text{sen}(\phi) \\ 0 & -\text{sen}(\phi) & \cos(\phi) \end{bmatrix} \quad (114)$$

- Rotação em torno do eixo Y

$$R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\text{sen}(\theta) \\ 0 & 1 & 0 \\ \text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (115)$$

- Rotação em torno do eixo Z

$$R(\psi) = \begin{bmatrix} \cos(\psi) & \text{sen}(\psi) & 0 \\ -\text{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (116)$$

sendo *sen* e *cos* respectivamente as abreviaturas para seno e cosseno dos seus respectivos ângulos em questão. Os ângulos ϕ , θ e ψ são definidos como sendo os ângulos de rotação desejados, respectivamente nos eixos X, Y e Z.

O quadricóptero atua sobre dois referenciais, um inercial alinhado a eixo norte terrestre e outro centrado coincidente ao centro de massa do veículo aéreo. Neste trabalho usaremos o subscrito i para citar o referencial inercial e c para citar o referencial no corpo do quadricóptero, o alinhamento dos eixos se dão pela orientação conhecida como "+" sendo exemplificada na Figura 20. Como o cerne da matriz de rotação é realizar a mudança de um sistema de coordenadas para outro, vamos definir R_i^c como sendo a matriz de rotação que transporta a informação do referencial i para o referencial c .

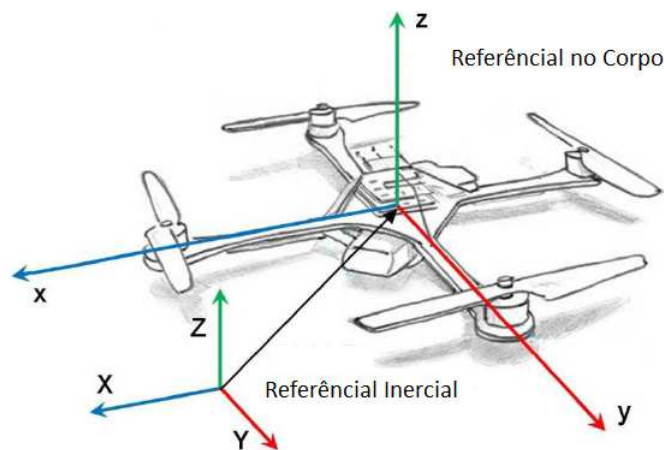


Figura 20: Representação dos referenciais e orientação dos eixos do quadricóptero.

Duas propriedades das matrizes de rotação devem ser explicadas devido a sua grande serventia para os cálculos que seguem: A primeira propriedade define que a inversa da matriz $R(\varsigma)$ pode ser facilmente obtida ao perceber que a sua inversa é igual a transposta, portanto

$$R_c^i(\varsigma) = R_i^c(\varsigma)^{-1} = R_i^c(\varsigma)^T, \quad (117)$$

note que R_c^i é a matriz de rotação que transporta a informação do referencial do corpo c para o referencial inercial i . A segunda propriedade diz respeito a sequência de rotações, é possível descrever qualquer orientação no espaço \mathbb{R}^3 ao realizar 3 rotações consecutivas em uma sequência pré determinada, as sequências de rotações comumente utilizadas são XYZ, ZYX e ZYZ. Vamos adotar neste trabalho a sequência XYZ dada por

$$\begin{aligned}
 R_c^i(\phi, \theta, \psi) &= R_{V_2}^c(\phi)R_{V_1}^{V_2}(\theta)R_i^{V_1}(\psi) \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\phi)c(\theta) \\ c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta) \end{bmatrix} \quad (118)
 \end{aligned}$$

onde V_1 e V_2 são referenciais intermediários na sequência de rotações executada, realizamos novamente uma simplificação na equação acima para poupar espaço sendo $s = \text{seno}$ e $c = \text{cosseno}$.

4.2 Dinâmica

Esta sessão tem como objetivo demonstrar o equacionamento dinâmico do quadricóptero. O modelo do UAV em questão tem doze estados descritos abaixo:

- X, Y e Z estados inerciais de posicionamento do quadricóptero, grandeza em metros.
- U, V e W estados de velocidade com referencial no centro do corpo do quadricóptero, grandeza em m/s.
- ϕ , θ e ψ ângulos de Euler que definem a orientação no \mathbb{R}^3 , grandeza em radianos.
- P, Q e R estados de velocidade angular com referencial no centro do corpo do quadricóptero, grandezas em radianos/s.

Na Figura 21 são demonstrados todos os estados atuantes na dinâmica do quadricóptero e os seus referenciais.

Na sequência é demonstrado como são obtidos os estados da dinâmica do UAV em estudo, para melhor organização, os equacionamentos estão separados em estados de translação (XYZ e UVW) e estados angulares ($\phi\theta\psi$ e PQR).

4.2.1 Estados de translação

A dinâmica dos estados de translação do quadricóptero são descritos no referencial alinhado ao centro do seu corpo definida por c , entretanto para obter o seu posicionamento no plano cartesiano \mathbb{R}^3 é necessária a mudança para o referencial inercial definido como i . A matriz de transformação que realiza esta operação é definida em (118), podemos escrever os estados inerciais de posicionamento X, Y e Z como:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = R_c^i(\phi, \theta, \psi) \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \quad (119)$$

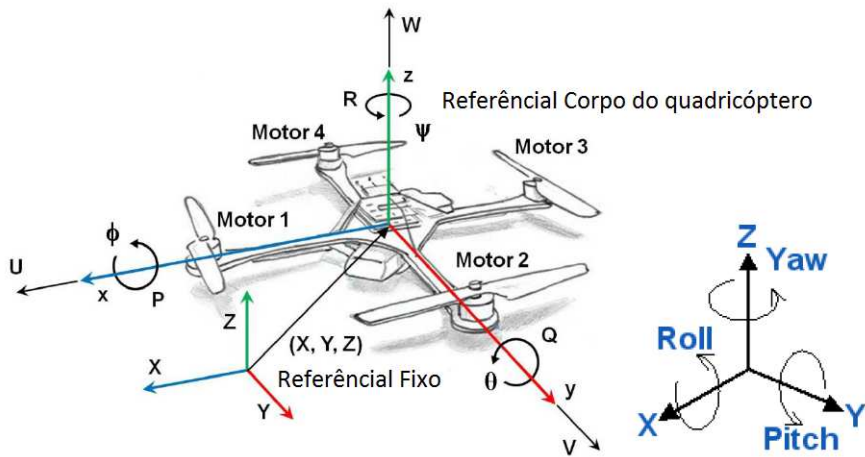


Figura 21: Representação dos estados associadas ao quadricóptero.

relembrando que

$$R_c^i(\phi, \theta, \psi) = R_i^c(\phi, \theta, \psi)^T. \quad (120)$$

Os estados das velocidades de translação U, V e W podem ser obtidos ao utilizar a clássica 2ª Lei de Newton para equacionar a dinâmica do quadricóptero, portanto teremos:

$$m \frac{dv}{dt} = f^i, \quad (121)$$

sendo m a massa total e f^c a força total aplicado em cada eixo do quadricóptero. Lembrando que as Leis de Newton são válidas somente para referenciais inerciais podemos escrever

$$\frac{d}{dt} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \frac{f^i}{m}, \quad (122)$$

substituindo (119) na equação anterior obtemos:

$$\frac{d}{dt} \left(R_c^i \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} \right) = \frac{f^i}{m}, \quad (123)$$

onde ocultamos as dependências das variáveis ϕ, θ e ψ para facilitar entendimento. Realizando a derivada da equação anterior e agrupando obtemos:

$$\frac{f_i}{m} = R_c^i \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} + \dot{R}_c^i \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \quad (124)$$

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = -R_c^i \dot{R}_c^i \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \frac{R_c^i f_i}{m}. \quad (125)$$

Em (STEVENS; LEWIS, 2003) é demonstrada uma propriedade matricial que facilita

em muito os cálculos, a propriedade em questão é dada por

$$\begin{aligned} \Omega &= -R_i^c \dot{R}_c^i, \\ \Omega &= \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}, \end{aligned} \quad (126)$$

podemos também definir Ω em um outro formato conforme

$$\Omega X = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \times X \quad (127)$$

ou seja, ΩX pode ser escrito como o produto vetorial dos estados P , Q e R e o vetor arbitrário X .

Substituindo (126) ou (127) em (125) e realizando os devidos cálculos obtemos finalmente os estados \dot{U} , \dot{V} e \dot{W} dados por

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} RV - QW \\ PW - RU \\ QU - PV \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (128)$$

onde $[f_x, f_y, f_z]^T$ são as forças exercidas separadamente em cada eixo. A equação acima descreve o comportamento das velocidades de translação executadas no quadricóptero.

4.2.2 Estados Angulares

A dinâmica dos estados de rotação do quadricóptero igualmente ao de translação, são descritas no referencial alinhado ao centro do seu corpo, entretanto para obter a sua orientação no plano cartesiano \mathbb{R}^3 vamos utilizar a matriz de transformação demonstrada em (STEVENS; LEWIS, 2003), podemos escrever os estados angulares de orientação ϕ , θ e ψ como:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\phi)\tan(\theta) & \text{cos}(\phi)\tan(\theta) \\ 0 & \text{cos}(\phi) & -\text{sen}(\phi) \\ 0 & \frac{\text{sen}(\phi)}{\text{cos}(\theta)} & \frac{\text{cos}(\phi)}{\text{cos}(\theta)} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \quad (129)$$

Vamos utilizar um procedimento semelhante ao empregado para obter as equações dinâmicas de translação. Os estados P , Q e R podem ser obtidos ao utilizar a 2ª lei de Newton que descreve movimentos de rotação como

$$\frac{dh}{dt} = \tau^i, \quad (130)$$

onde h e τ^i são respectivamente os vetores dos momentos angulares e torques aplicados no quadricóptero para cada eixo do seu referencial inercial i . Uma propriedade muito útil do momento angular é que ela pode ser calculada através do seu momento de inércia J e sua velocidade angular ω como $h = J\omega$, sendo $\omega = [P \ Q \ R]^T$. Lembrando que a segunda Lei de Newton é válida apenas para referenciais inerciais e utilizando $h = J\omega$, obtemos:

$$\frac{d}{dt} (R_c^i J_c^i \omega) = \tau^i \quad (131)$$

Calculando a derivada da equação anterior chegamos a seguinte equação:

$$\tau_i = \dot{R}_c^i J_i^c \begin{bmatrix} P \\ Q \\ R \end{bmatrix} + R_c^i J_i^c \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix}. \quad (132)$$

Isolando a variável de interesse obtemos:

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = -J_i^c{}^{-1} \left(R_c^i \dot{R}_c^i J_i^c \begin{bmatrix} P \\ Q \\ R \end{bmatrix} + R_c^i \tau^i \right). \quad (133)$$

O momento de inércia J pode ser obtido através da matriz constante conhecida como matriz de inércia, dada por

$$J = \begin{bmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2) dm \end{bmatrix}$$

$$J = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{bmatrix} \quad (134)$$

entretanto podemos simplificar a matriz definida acima ao adotar que o quadricóptero em estudo é simétrico e seu centro é coincidente com o centro de massa, esta simplificação não é somente teórica, na prática sempre é objetivado que o quadricóptero seja o mais simétrico possível. A matriz de inércia simplificada é dada por

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}, \quad (135)$$

ou seja, a matriz de inércia é diagonal.

Substituindo a matriz de inércia definida na equação anterior e usando a propriedade matricial (126) em (133), obtemos a seguinte equação

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix} \left(\begin{bmatrix} 0 & R & -Q \\ -R & 0 & P \\ Q & -P & 0 \end{bmatrix} \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} + \begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} \right), \quad (136)$$

onde $\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$ é definido como o torque associado a cada eixo no referencial c do quadricóptero. Realizando as operações necessárias obtemos

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \frac{(J_y - J_z)}{J_x} QR \\ \frac{(J_z - J_x)}{J_y} PR \\ \frac{(J_x - J_y)}{J_z} PQ \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} \tau_{roll} \\ \frac{1}{J_y} \tau_{pitch} \\ \frac{1}{J_z} \tau_{yaw} \end{bmatrix} \quad (137)$$

Os estados equacionados acima tem como referencial o corpo do quadricóptero, e descrevem o comportamento dinâmico dos estados de rotação.

4.2.3 Força e momento

O objetivo desta sessão é descrever todas as forças e torques que atuam no movimento do quadricóptero em estudo. Na Figura 22 são demonstradas as forças e torques atuantes, onde F_f , F_e , F_d e F_a são definidas como as forças exercidas pelos rotores para cada lado do quadricóptero. τ_f , τ_e , τ_d e τ_a são os torques aplicados pelos rotores em cada lado.

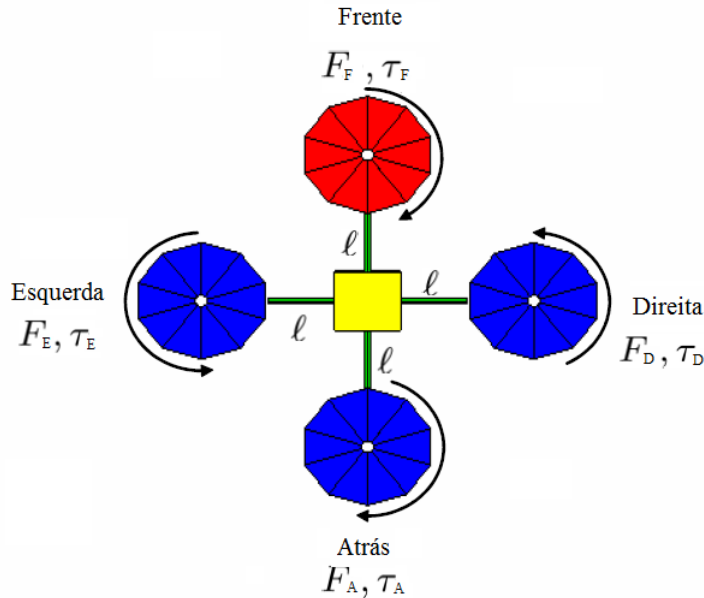


Figura 22: Forças e torques atuantes no quadricóptero (STEVENS; LEWIS, 2003).

As forças que atuam no quadricóptero são definidas como força de empuxo e torque ambas exercidas pelos 4 rotores. Podemos definir o total de empuxo exercido pelos rotores do quadricóptero como a soma direta de cada componente de força, portanto temos

$$\sum F = F_f + F_e + F_d + F_a. \quad (138)$$

Observando a Figura 22, podemos perceber que para executar um movimento positivo de *roll*, movimento de giro entorno do eixo X, o motor da esquerda deve receber um incremento de força positivo ao contrário do motor da direita que deve receber um incremento negativo de força, podemos equacionar esta análise pela seguinte equação

$$\tau_{roll} = l(F_e - F_d) \quad (139)$$

sendo l a distância compreendida entre o centro do quadricóptero até o centro do rotor.

Utilizando a mesma análise, podemos equacionar o torque para o movimento positivo de *pitch*, movimento de rotação entorno do eixo Y, o motor da frente deve receber um incremento positivo de força ao contrário do motor de trás que deve receber um incremento negativo de força, portanto

$$\tau_{pitch} = l(F_f - F_a) \quad (140)$$

O movimento de rotação positivo de *yaw*, movimento de giro entorno do eixo Z, é um pouco diferentes dos dois anteriores, de acordo com a terceira lei de Newton o empuxo exercido pelo rotor produz um torque no eixo do quadricóptero, devido a isto que os rotores da posição frente e trás são rotacionados no sentido horário e os rotores da posição

esquerda e direita são rotacionados no sentido anti-horário, assim é mantido o equilíbrio de torques. Portanto para realizar um movimento intencional de *yaw* no sentido positivo podemos basicamente diminuir este equilíbrio como demonstrado na equação abaixo.

$$\tau_{yaw} = \tau_e + \tau_d - \tau_f - \tau_a. \quad (141)$$

Para um modelo completo do quadricóptero, temos que realizar a modelagem do principal componentes de força e torque, o rotor. A modelagem completa deste componente exige conhecimento de vários parâmetros como área, raio, coeficiente de empuxo e densidade do ar. Na prática utilizamos um coeficiente obtido experimentalmente C_t e C_p que descreve o comportamento do empuxo e torque conforme

$$F = C_t w^2 \quad (142)$$

$$\tau = C_p w^2 \quad (143)$$

portanto, força e torque são consideradas grandezas diretamente proporcionais ao quadrado da velocidade angular w (MAHONY; KUMAR; CORKE, 2012).

Organizando em um formato matricial podemos escrever todas as componentes de força e torque como

$$\begin{bmatrix} \sum F \\ \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} C_t & C_t & C_t & C_t \\ 0 & -lC_t & 0 & lC_t \\ lC_t & 0 & -lC_t & 0 \\ -C_p & C_p & -C_p & C_p \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \quad (144)$$

Uma última força que ainda não foi considerada nos cálculos apresentadas, é a tão importante força da gravidade, ela atua na dinâmica de translação somente no eixo Z do referencial inercial, portanto devemos transportar esta influência para o referencial não inercial no centro do quadricóptero. A transformação desejada é dada por

$$F_g = R(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} mg \sen(\theta) \\ -mg \cos(\theta) \sen(\phi) \\ -mg \cos(\theta) \cos(\phi) \end{bmatrix} \quad (145)$$

onde g é a constante de aceleração da gravidade.

4.3 Modelo completo

Abaixo são organizadas os doze estados do quadricóptero equacionados nas sessões anteriores, 6 representando o movimento de translação e 6 representando os movimentos de rotação.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\phi)c(\theta) \\ c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta) \end{bmatrix}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (146)$$

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} RV - QW \\ PW - RU \\ QU - PV \end{bmatrix} + mg \begin{bmatrix} \sen(\theta) \\ -\cos(\theta) \sen(\phi) \\ -\cos(\theta) \cos(\phi) \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ \sum F \end{bmatrix} \quad (147)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\text{sen}(\phi) \\ 0 & \frac{\text{sen}(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (148)$$

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \frac{(J_y - J_z)QR}{J_x} \\ \frac{(J_z - J_x)PR}{J_y} \\ \frac{(J_x - J_y)PQ}{J_z} \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x}\tau_{roll} \\ \frac{1}{J_y}\tau_{pitch} \\ \frac{1}{J_z}\tau_{yaw} \end{bmatrix} \quad (149)$$

4.4 Simulação

Nesta sessão utilizamos os estados equacionados na sessão anterior para simular o comportamento dinâmico completo do quadricóptero, todas as constantes e variáveis compreendidas no equacionamento foram obtidas do trabalho de (COSTA, 2008), neste trabalho o autor obtém todas as características do quadricóptero através de experimentos específicos.

A matriz de inércia (135) pode ser obtida por três métodos, pelo seu cálculo exato, por construção de um modelo 3D em algum *software* específico ou pelo cálculo aproximado ao utilizar uma representação simplificado para da sua estrutura, por exemplo, esferas, quadrados e cilindros.

O cálculo preciso é algo dispendioso de ser realizado manualmente, sendo tecnicamente inviável de ser realizado. Construir o modelo 3D em algum *software* é um recurso poderoso para obter a matriz de inércia, um modelo bem construído obtém elevados graus de precisão quanto ao valor desejado, sendo portanto uma ótima opção, todavia, dependendo do grau de precisão que desejamos para o nosso modelo simulado, consome elevado gasto no quesito tempo de projeto, o que torna muitas vezes o projeto inviável. Utilizar uma aproximação para a estrutura através de elementos básico é uma opção rápida, mas não mais precisa que os dois métodos citados anteriormente, entretanto é algo muito utilizado dada pela sua facilidade e rapidez de projeto.

Vamos utilizar o terceiro método para obter a matriz de inércia, sendo a aproximação para a estrutura do quadricóptero dada pela Figura 23, onde utilizaremos uma esfera de raio R e massa M para representar o corpo do quadricóptero, e os rotores como massas pontuais com massa m a uma distância l do eixo de rotação.

A inércia de uma esfera e de uma massa pontual podem ser obtidas por (RESNICK; HALLIDAY; WALKER, 1988)

$$J_{esfera} = \frac{2MR^2}{5} \quad J_{Pontual} = ml^2, \quad (150)$$

portanto o cálculo das componentes da matriz de inércia são dadas por

$$\begin{aligned} J_x &= \frac{2MR^2}{5} + 2ml^2 \\ J_y &= \frac{2MR^2}{5} + 2ml^2 \\ J_z &= \frac{2MR^2}{5} + 4ml^2, \end{aligned}$$

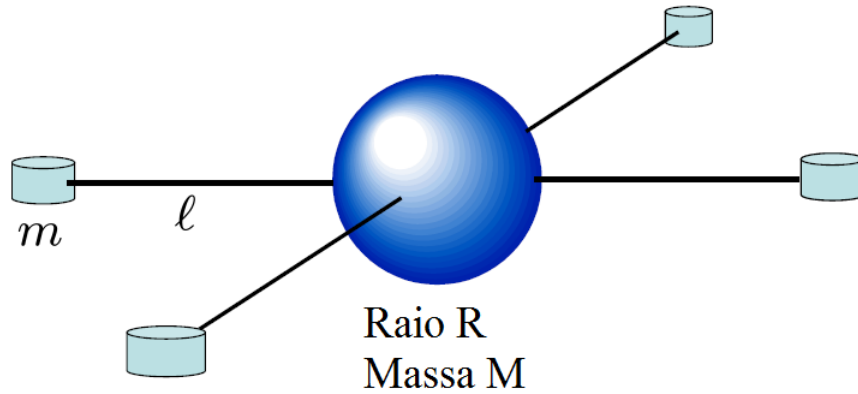


Figura 23: Estrutura simplificada para cálculo da matriz de inércia (BEARD, 2008).

Tabela 8: Características do motor AXI 2212-12 (HOBBYKING.COM, ????).

RPM/V	2000
Gama entrada [V]	9-12
RPM máximo	6000
Dimensão do eixo	3,17mm
Resistência Interna	45mΩ
Corrente sem carga	1,2A
Capacidade de corrente	28A/60s

usando os valores de $R = 0.1m$, $M = 1.2Kg$, $m = 0.060Kg$ e $l = 0.60m$ obtemos a seguinte matriz de inércia.

$$J = \begin{bmatrix} 0.048 & 0 & 0 \\ 0 & 0.048 & 0 \\ 0 & 0 & 0.0912 \end{bmatrix} kg m^2. \quad (151)$$

Vamos utilizar nesta simulação o motor AXI 2212-12, o mesmo utilizado no trabalho de (COSTA, 2008), sendo suas especificações dadas na Tabela 8.

Como já mencionado anteriormente estes rotores tem um comportamento semelhante a sistemas de 1º ordem, por serem extremamente rápidos o seu regime transitório é considerado desprezível e desconsiderado nesta simulação, podemos portanto equacionar o rotor utilizando suas características apresentadas por

$$\omega = 0 \quad |V| \leq V_0 \quad (152)$$

$$\omega = K_v(V - 9) = \frac{2000 \times 2\pi}{60}(V - 9) \quad |V| > V_0 \quad (153)$$

$$G_m(s) = \frac{K_v}{t s + 1} \quad (154)$$

sendo w a velocidade angular do rotor, V a tensão de entrada no motor, V_0 é a tensão de zona morta e t é a constante de tempo do sistema de primeira ordem sendo seu valor de $t = 20ms$.

Na Figura 24 é demonstrado como o motor pode ser modelado utilizando a ferramenta

de simulação *Simulink* do *software* Matlab, nele podemos perceber dois componentes não lineares, saturação e zona morta.

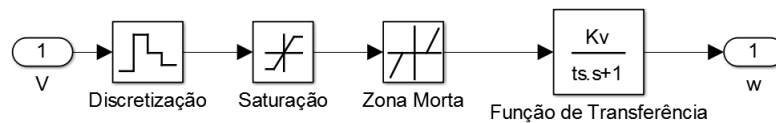


Figura 24: Diagrama de blocos do rotor realizada com auxílio da ferramenta *simulink* do *software* Matlab.

4.4.1 Simulink

O modelo do quadricóptero descrito nas sessões anteriores é recriado nesta sessão com o auxílio da ferramenta de simulação *simulink*, todas as constantes e parâmetros necessários já foram definidos. Para melhor organização vamos apresentar o modelo simulado divididos em sub-blocos de acordo com a sua utilidade.

4.4.1.1 Sub-bloco de controle

A estrutura de controle realizada para o quadricóptero nesta simulação é o formato cascata, na malha externa temos o controle angular de ϕ , θ e ψ , na malha interna temos o controle da velocidade angular P, Q e R. Diferentes dos demais temos o controle da altura Z, executada com uma única malha de controle. A Figura 25 exemplifica as diferentes malhas existentes.

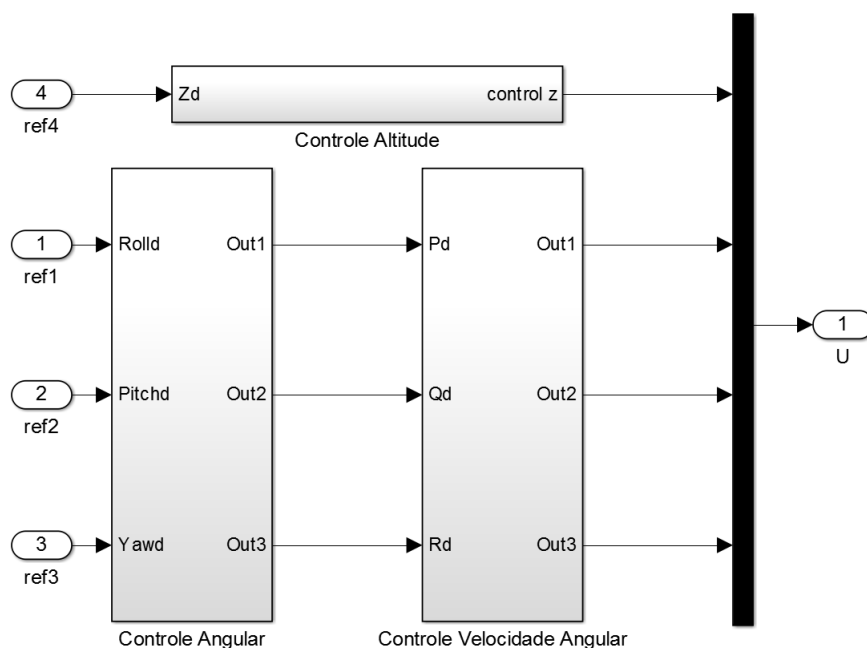


Figura 25: Diagrama do sub-bloco de controle do quadricóptero com auxílio da ferramenta *simulink* do *software* Matlab.

Os sub-blocos de controle angular e velocidade angular tem em comum a classe de controle PID escolhida, uma vez que é pretendido utilizar esta classe com o método *Cas-*

cade *Iterative Feedback Tuning*, nas Figuras 26 e 27 são demonstrados os sub-blocos de controle.

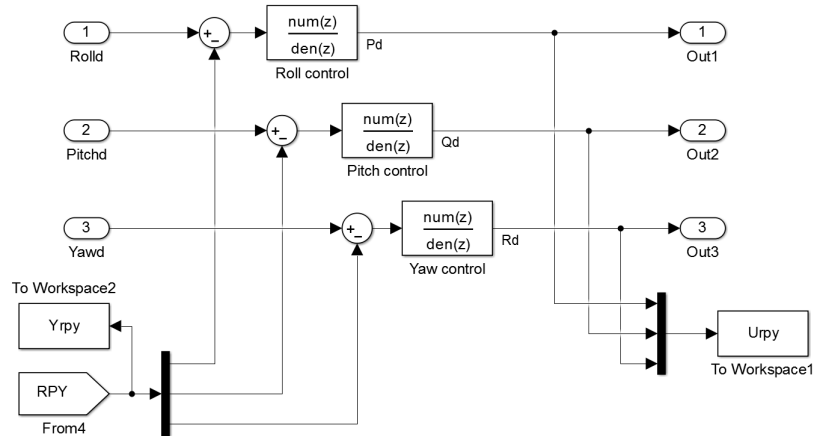


Figura 26: Diagrama do sub-bloco de controle angular com auxílio da ferramenta *simulink* do *software* Matlab.

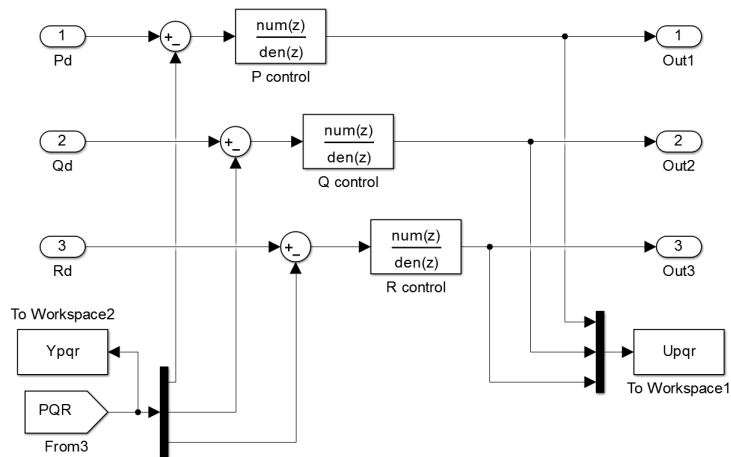


Figura 27: Diagrama do sub-bloco de controle da velocidade angular com auxílio da ferramenta *simulink* do *software* Matlab.

4.4.1.2 Sub-blocos de comando e força

Na Figura 28 são demonstrados os sub-blocos de comando, os sub-blocos correspondentes aos rotores e o sub-bloco de força e momento. Os sub-blocos dos motores já foram mencionados na Figura 24. No sub-bloco de comando temos uma derivação da equação (144), o objetivo é separar os devidos sinais de comando para os respectivos motores, o

equacionamento é dado por

$$\begin{bmatrix} \sum F \\ \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} 0,25 & 0,25 & 0,25 & 0,25 \\ 0 & -0,5 & 0 & 0,5 \\ 0,5 & 0 & -0,5 & 0 \\ -0,25 & 0,25 & -0,25 & 0,25 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} = \mathcal{M} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad (155)$$

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} = \mathcal{M}^{-1} \begin{bmatrix} \sum F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (156)$$

onde Q_i são os respectivos comandos para os motores. No sub-bloco força e momento temos a aplicação da equação (144) para a obtenção da força e momento de cada motor.

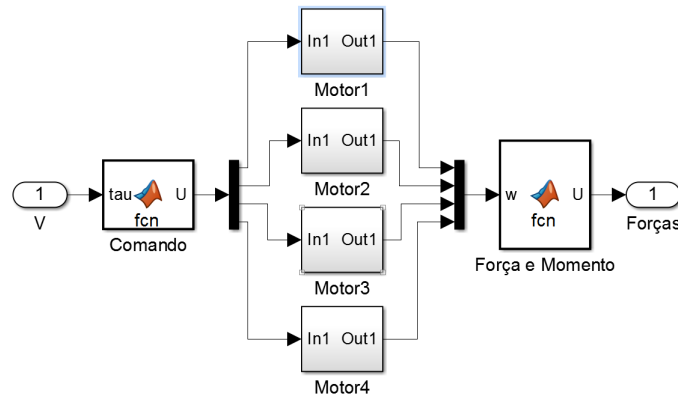


Figura 28: Diagrama do sub-bloco de controle angular com auxílio da ferramenta *simulink* do *software* Matlab.

4.4.2 Resultados simulados do método *Cascade Iterative Feedback Tuning*

Nesta sessão vamos utilizar o método apresentado na sessão 3.5 para realizar a sintonia das malhas interna e externa dos controladores do quadricóptero simulado em *simulink*. A classe de controladores \mathcal{C} adotada em ambas as malhas são os controladores PDs, este tipo de controlador é largamente utilizado em aplicações convencionais de voo, por isto sua escolha (SALIH et al., 2010; ARGENTIM et al., 2013). Definimos os controladores PDs das malhas interna e externa utilizados no método CIFT respectivamente como

$$C_1 = \begin{bmatrix} k_{p,1} \\ k_{d,1} \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \quad C_2 = \begin{bmatrix} k_{p,2} \\ k_{d,2} \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}. \quad (157)$$

Para facilitar a demonstração dos resultados, vamos apresentar somente a convergência realizada para a malha de controle do ângulo *roll* ϕ , como os outros estados apresentaram pequenas variações entre suas dinâmicas, a sintonia não apresenta diferença significativa para demonstração.

Todos os outros controladores, que não estão sendo sintonizados pelo método CIFT, são mantidos constantes e ativos durante a sintonia, suas escolhas foram realizadas de maneira empírica e buscando somente uma região estável de operação para os estados,

seus valores são dados por

$$\begin{aligned}
 C_\theta &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} & C_\psi &= \begin{bmatrix} 10 \\ 10 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \\
 C_Q &= \begin{bmatrix} 0,02 \\ 0,1 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} & C_R &= \begin{bmatrix} 1 \\ 10 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \\
 C_Z &= \begin{bmatrix} 50 \\ 2 \\ 400 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{z-1} \\ \frac{z-1}{z} \end{bmatrix}
 \end{aligned} \tag{158}$$

Escolhemos como modelo de referência a função de transferência dada por

$$T_d(z) = \frac{0,02}{z - 0,98} \tag{159}$$

O processo é contaminado por dois ruídos gaussianos com média zero e variância de $\sigma^2 = 10^{-5}$, de acordo ao demonstrado na Figura 6, o nível de variância empregado neste experimento tem como objetivo obter um ruído com amplitude igual a 10% da amplitude da variável controlada. Definimos como período de amostragem o valor de 0,01s e como referência uma onda quadrada com amplitude 0,1 radianos e período de 8 segundos, dada por

$$r_2(t) = 0,1 \text{square} \left(\frac{2\pi t}{8} \right) \tag{160}$$

Como procedimento de convergência, utilizamos o método *Steepest Descent* (18) nas primeiras 5 iterações. O método de Quasi-Newton (19) é empregado nas últimas 15 iterações finalizando a convergência, sendo que neste utilizamos a aproximação da Hessiana demonstrada em (35). O passo escolhido para ambos os métodos de convergência segue o definido em (22) e (23), sendo a constante n escolhida com valor de 0,2. Nas Figuras 29 e 30 são demonstradas as respostas inicial, final e desejada para o ângulo *roll*, utilizando a metodologia CIFT.

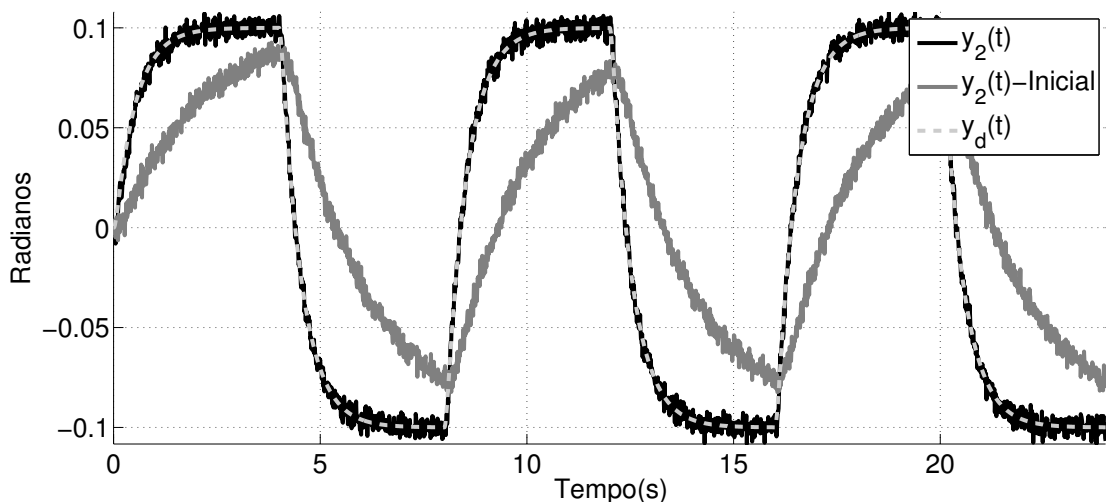


Figura 29: Saída inicial com os controladores da equação (158), saída desejada e resultado final $y_2(t)$ utilizando o método de sintonia CIFT para o modelo simulado.

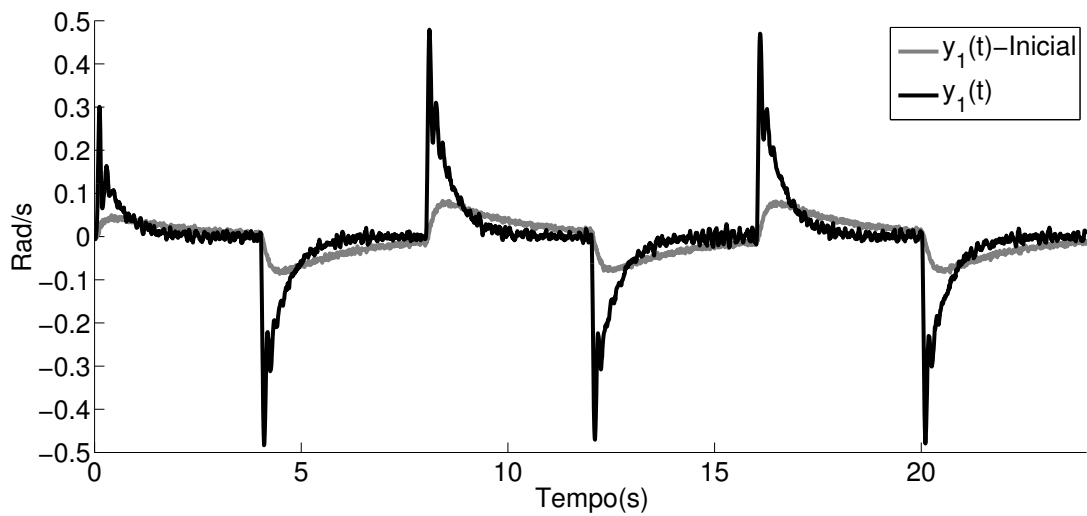


Figura 30: Saída final $y_1(t)$ com os controladores iniciais da equação (158), utilizando o método de sintonia CIFT para o modelo simulado.

Na Tabela 9 são apresentados os dados de convergência dos parâmetros dos controladores em conjunto com o seu respectivo custo.

Quanto a convergência do algoritmo CIFT, é percebido um comportamento diferente em comparação aos exemplos lineares demonstrados neste trabalho. Como esperado, o uso do método CIFT para sistemas não lineares introduziu um pequeno erro que, ao longo das iterações, distorceu a convergência, entretanto o erro introduzido não afetou a resposta obtida que ficou muito próxima ao sinal desejado como mostrado na Figura 29. Nestas circunstâncias de simulação o algoritmo CIFT se mostrou capaz de realizar a convergência deste sistema não-linear.

O erro introduzido pela não linearidade do sistema começa a ficar aparente a partir da iteração 16. A partir desta iteração a função custo apresenta uma tendência crescente, por isso é aconselhável parar o experimento. Segundo (HJALMARSSON, 2002), o comportamento obtido é esperado, uma vez que estamos utilizando um método para sistemas lineares em um sistema não linear, que é o caso do quadricóptero.

Tabela 9: Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o modelo simulado do quadricóptero.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		k_p	k_d	k_p	k_d	
1	0,3016	0,1000	0,1000	0,5000	0,1000	SD
2	0,2872	0,2145	0,0996	0,6640	0,1013	SD
3	0,2469	0,2424	0,0992	0,8620	0,1024	SD
4	0,1980	0,2676	0,0989	1,0604	0,1034	SD
5	0,1764	0,2916	0,0987	1,2590	0,1043	SD
6	0,1300	0,3165	0,0986	1,4574	0,1054	QN
7	0,1282	0,3414	0,0782	1,5182	0,1276	QN
8	0,1186	0,3887	0,0416	1,6311	0,1726	QN
9	0,0745	0,4440	-0,0038	1,7665	0,2631	QN
10	0,0722	0,5117	-0,0658	1,8889	0,3937	QN
11	0,0670	0,5697	-0,1197	1,9624	0,5655	QN
12	0,0568	0,6220	-0,1750	1,9942	0,7747	QN
13	0,0472	0,6684	-0,2154	1,9985	1,1143	QN
14	0,0557	0,7507	-0,2598	1,9953	1,1756	QN
15	0,0756	0,7327	-0,2764	2,0019	1,5209	QN
16	0,0339	0,7681	-0,3027	1,9968	1,8374	QN
17	0,0509	0,7972	-0,3168	2,0053	1,9384	QN
18	0,0806	0,7538	-0,3050	1,9989	2,3287	QN
19	0,0708	0,7247	-0,3066	2,0010	2,8141	QN
20	0,0875	0,6814	-0,2898	1,9991	3,6548	QN

5 EXPERIMENTO PRÁTICO

Temos como objetivo deste capítulo colocar em prática todo o conhecimento apresentado neste trabalho, portanto usaremos o algoritmo *Cascade Iterative Feedback Tuning* para realizar a sintonia dos controladores das malhas interna e externa de um quadricóptero real. O modelo dinâmico descrito na sessão anterior tem como intuito demonstrar as diferenças encontradas, no controle e sintonia, ao utilizar um modelo simulado em comparação ao caso real prático.

O capítulo está organizado na seguinte sequência: o quadricóptero utilizado nos experimentos práticos é definido, a plataforma dos testes e os detalhes dos experimentos são apresentados, uma descrição detalhada sobre os procedimentos dos experimentos realizados e finalizando pela demonstração dos resultados alcançados.

5.1 Quadricóptero

Uma característica deste trabalho é a opção de utilizar uma estrutura de quadricóptero, ou conhecido também como *frame*, de baixo custo e popular entre os aeromodelistas. Isto advém principalmente do objetivo de desenvolver uma metodologia que ajude estes praticantes a realizar a sintonia dos controladores dos seus quadricópteros de um modo mais simples e eficiente.

O *frame* escolhido é o modelo F450, uma estrutura de baixo custo e de fácil montagem, na Figura 31 é demonstrado o frame mencionado. O frame vem dotado de encaixes para os rotores, encaixes para a placa controladora e várias possíveis personalizações podem ser alcançadas, por exemplo: adição de câmeras, apoios, suspensão e compartimento para cargas.

Na sequência são descritos os principais equipamentos que constituem o quadricóptero.

5.1.1 Rotores

A única força exercida pelo quadricóptero é gerada pela rotação das suas 4 hélices, esta rotação por sua vez deve ser desempenhada pelo conjunto dos seus 4 motores, dimensionar o correto conjunto motor/hélice comumente chamado de rotor, é de extrema importância para o desempenho dinâmico do quadricóptero (MCCORMICK, 1995). Neste trabalho optamos por escolher o conjunto motor e hélice considerados comuns para um quadricóptero, lembrando que o intuito é ter como aplicação um UAV considerado popular.

O motor utilizado para este tipo de aplicação é do tipo corrente contínua sem escovas, ou do inglês *Brushless*. Os *Brushless* são motores com características bem distintas dos



Figura 31: *Frame* do quadricóptero utilizado para os experimentos práticos (HOBBY-KING.COM, ???).

seus pares, por exemplo: elevada velocidade de reação, alto RPM (Rotações Por Minuto), confiabilidade elevada, ruído reduzido, maior vida útil (devido a ausência de desgaste da escova). Propriedades muito úteis para esta aplicação.

Outra propriedade interessante deste tipo de motor é a chamada zona morta, região onde a velocidade angular é nula até um limiar inferior de tensão ser ultrapassada. O motor também tem um limite superior para a tensão aplicada nele, qualquer acréscimo de tensão não resulta em um acréscimo na rotação angular, permanecendo este valor constante para qualquer tensão superior ao limite superior, esta propriedade é conhecida como saturação da velocidade angular ou RPM. Na Figura 32 é demonstrado um exemplo genérico de motor e a exemplificação dos limiares de tensão comentados.

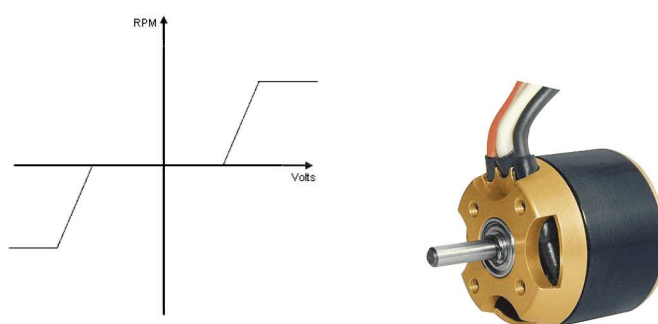


Figura 32: Exemplo genérico das zonas de funcionamento do motor de corrente contínua *brushless* (COSTA, 2008).

Na Figura 33 é demonstrado o motor utilizado nos experimentos do quadricóptero. O motor em questão se trata do modelo 2826 1100kv da marca NTM Prop Drive. Na Tabela 10 são apresentadas as suas principais características construtivas.

Os motores do tipo *brushless* são acionados por meio de um dispositivo eletrônico de velocidade conhecido também como inversor, *driver* ou ESC (do inglês *Electronic Speed Control*), esta última nomenclatura será utilizada neste trabalho. Este circuito de potência



Figura 33: Motor modelo NTM Prop Drive Series 2826 1100kv utilizado no experimento do quadricóptero (HOBBYKING.COM, ???).

Tabela 10: Características do motor Prop Drive 2826 1100kv.

RPM/V	1100
Gama entrada [V]	0,1-12,6(3S)
RPM máximo	13750
Polos	12
Dimensão do eixo	3mm
Máxima Corrente	20A
Peso	57g

realiza o controle da velocidade do motor e seu acionamento é dado por um sinal DC modulado ou também conhecido como sinal PWM (do inglês *Pulse Width Modulation*). Neste trabalho utilizamos o ESC modelo SkyWalker-20A da marca Hobbywing demonstrado na Figura 34.



Figura 34: ESC utilizado para controle de velocidade dos motores do quadricóptero (ISEE-ROBOT.COM, ???).

A hélice usada no quadricóptero é um item essencial que deve harmonizar com o restante do *frame*, uma vez que é o item responsável por exercer a força de empuxo jogando o ar para baixo, ela deve ser capaz de exercer uma força maior que o peso pretendido a suportar, ou seja, o peso total do quadricóptero. Escolhemos utilizar hélices capazes de aplicar a força de empuxo com valor nominal de duas vezes ao peso do quadricóptero, isto para que o UAV fique em estado estacionário planando com 50% da sua força total. O peso total do quadricóptero é calculado na Tabela 11 onde são listados seus componentes.

As especificações para as hélices comumente dizem respeito ao tamanho e passo. O tamanho nada mais é que o comprimento entre as duas pontas da hélice e o passo é

Tabela 11: Lista dos componentes por peso do quadricóptero.

<i>Frame</i>	272
Motor	228g
Hélice	168g
Placa de Controle	14g
Bateria	187g
Esc	76g
Diversos	50g
Soma	995g

uma medida teórica de distância que mede o quanto a hélice se deslocaria para frente em condições ideais e com 100% de eficiência.

As especificações seguem medidas inglesas e são dadas por unidades em polegadas, alguns formatos de nomenclatura podem ser encontrados por exemplo como 10X45, 1045, 10X4.5 e 10X45. Nestes exemplos a hélice tem o tamanho de 10 polegadas e um passo de 4.5 polegadas. A hélice escolhida para a nossa aplicação é do tipo 8X4.5, ou seja, tem 8" de tamanho e 4.5" de passo.

Na Figura 35 é demonstrada a hélice utilizada no experimento prático.



Figura 35: Hélice 8X4.5 utilizada no quadricóptero dos experimentos práticos.

Existem duas maneiras de conhecer o valor da força de empuxo que o conjunto motor/hélice alcança: podemos utilizar coeficientes teóricos tabelados com isto calcular o seu valor para condições ideais ou obter por meios experimentais onde o próprio conjunto motor/hélice que será utilizado na aplicação é testado (MCCORMICK, 1995). Escolhemos o meio experimental como meio para obter o valor de força, entretanto não foi necessário realizar os experimentos práticos pois é possível encontrar muitas fontes para este valor em consultas rápidas, como este é um valor sempre consultado para a escolha dos conjuntos de propulsão, os próprios fornecedores muitas vezes disponibilizam este tipo de informação.

A força de empuxo de cada motor é obtida através de (HOBBYKING.COM, ????) sendo seu valor de 5,2214N, em (ADAMONE.RCHOMEPAGE.COM, ????) temos uma outra fonte muito eficiente para consulta a respeito de valores como força de empuxo, torque e potência e para verificar discrepâncias em valores fornecidos pelos fornecedores, nele é possível calcular os valores teóricos para diversas hélices comerciais e também personalização dos mesmo. O valor da força de empuxo para o quadricóptero é dado por

$$F_e = 4.5, 2214N = 20, 8844N \quad (161)$$

utilizando o peso total do quadricóptero definido na Tabela 11 calculamos

$$F_p = 0,995kg * 9,81 \frac{m}{s^2} = 9,7610N \quad (162)$$

portanto comparando a força de empuxo F_e e a força peso do quadricóptero F_p é percebido que o objetivo de aplicar duas vezes o valor da força peso do quadricóptero como empuxo foi obtido, ou seja, os rotores utilizam 50% da sua força para se manter em estado estacionário.

5.1.2 Placa controladora

A placa controladora escolhida utiliza o mesmo microcontrolador que a franquia tão popular Arduino. O microcontrolador em questão é o ATmega 328P de 8 bits e 16Mhz que executa todos os comandos de controle e processamento de sinais dos sensores, nesta placa temos também integrado os sensores inerciais giroscópio, acelerômetro, barômetro e magnetômetro ou seja, esta placa foi desenvolvida especificamente para a aplicação de aeromodelismo. Entretanto não utilizamos a biblioteca própria Arduino na programação, o grupo de pesquisa em questão desenvolveu suas próprias bibliotecas na linguagem de programação C/C++.

O placa controladora é do modelo conhecido como *Multiwii Flight Controller*, ele possui 6 entradas analógicas e 14 digitais sendo destas 6 como saídas para PWM, a Figura 36 demonstra o controlador utilizado.

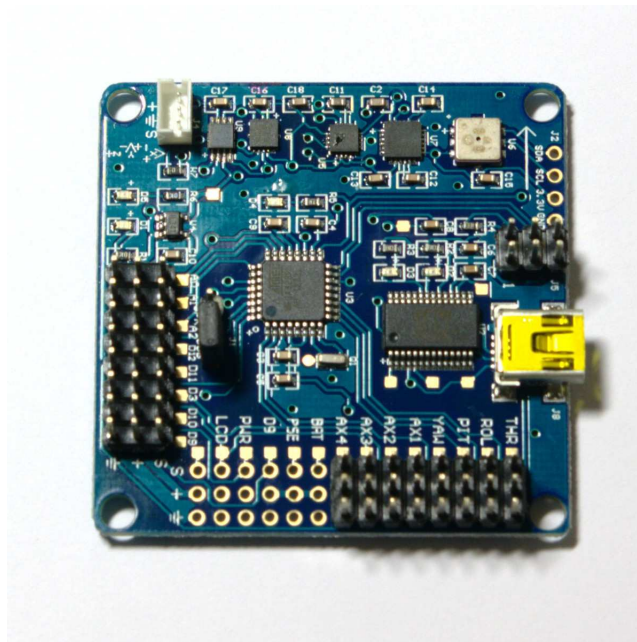


Figura 36: Placa modelo *Multiwii Flight Controller* utilizada como controlador para o quadricóptero dos experimentos (MULTICOPTERS.CO.UK, ???).

5.1.3 Inertial Measurement Unit - IMU

A unidade de medição inercial mais conhecida como IMU (do inglês, *Inertial Measurement Unit*) é um dispositivo eletrônico capaz de medir as forças que atual em seu referencial, acoplado ao quadricóptero o dispositivo informa as medidas inerciais do que chamamos anteriormente de referencial no corpo do quadricóptero c .

O dispositivo integra em seu circuito acelerômetros e giroscópios e às vezes magnetômetros, a informação combinada de todos os sensores é fundida utilizando técnicas de fusão de dados para gerar os estados estimados de velocidade angular P , Q e R e as velocidades de translação U , V e W . Nesta aplicação a placa de controle já possui o dispositivo IMU integrado, nele temos vários componentes de sensoriamento como: giroscópio, acelerômetro, barômetro e magnetômetro.

5.1.4 Baterias

Os UAVs em geral precisam de muita energia para suprir os vários componentes eletrônicos que o compõem, mas em especial o componente que mais consome este recurso são os 4 motores *Brushless*, eles demandam um fornecimento constante de valores altos de corrente, por exemplo os 4 circuitos de potência ESC utilizado são projetadas para cada um suportar 20 amperes em momento de pico. As baterias de polímero de lítio são indicadas para esta aplicação, pois permitem elevadas correntes de descarga e possuem maior carga e menor peso comparada com outros tipos de baterias com a mesma capacidade. Na Figura 37 são demonstradas as baterias utilizadas ao longo do experimento.



Figura 37: Baterias utilizadas durante os experimentos práticos.

5.2 Plataforma de testes

O experimento se limita demonstrar a sintonia dos controladores das malhas interna e externa para o controle do ângulo *roll*. O controle do ângulo *pitch* obtêm resultados muito semelhantes devido a sua dinâmica ser praticamente espelhada, portanto não é vital sua demonstração. O experimento é realizado utilizando restrições de movimento para o ângulo *pitch*. Uma forma de realizar esta restrição é simplesmente amarrando os lados correspondentes ao movimento de giro *pitch*. Na Figura 38 é demonstrada a plataforma de testes utilizada no experimento.

Este tipo de restrição diminui as interferências não lineares típicas da dinâmica do sistema, sem comentar no quesito segurança dos participantes do experimento. Restringindo seus movimentos como demonstrado na Figura 38, é possível simplificar o modelo apresentado em (148) e (149) como:

$$\dot{\phi} = P \quad (163)$$

$$\dot{P} = \frac{1}{J_x} \tau_{roll} \quad (164)$$



Figura 38: Plataforma de testes.

Restringir os movimentos do UAV para facilitar a sintonia dos controladores é uma prática bem comum, podemos citar o trabalho de (GONGORA et al., 2009) que utiliza uma plataforma para restringir o movimento de um helicóptero, e o trabalho de (KIRLI et al., 2010) que demonstra o uso de uma plataforma bem sofisticada que somente restringe o ganho de altura, ou seja, o movimento de subida.

5.3 Experimento

Vamos utilizar a mesma estrutura de controle apresentada na sessão de simulação 4.4.1.1, entretanto só usaremos as malhas de controle pertencentes ao controle da variável de interesse, o ângulo *roll*. O controle utilizado é classificado como monovariável discreto e invariante no tempo em formato cascata como demonstrado na Figura 39 sendo ρ_1 e ρ_2 os vetores de parâmetros dos controlador das malhas interna e externa.

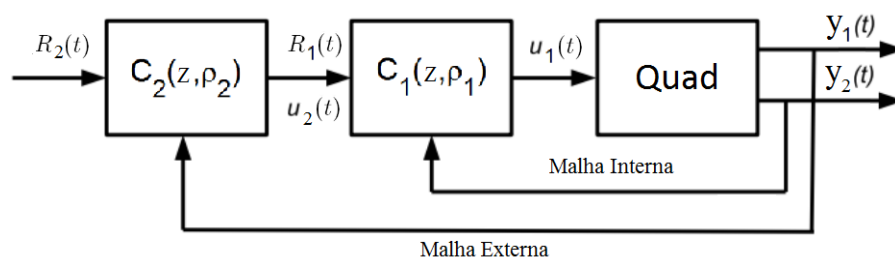


Figura 39: Diagrama de blocos da estrutura utilizada para o controle do ângulo *roll* do quadricóptero.

A classe \mathcal{C} de controladores adotada nos experimentos contempla a estrutura PD, de-

finida novamente por conveniência como

$$C_1 = \begin{bmatrix} k_{p,1} \\ k_{d,1} \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \quad C_2 = \begin{bmatrix} k_{p,2} \\ k_{d,2} \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}.$$

utilizamos esta mesma estrutura de controle no exemplo do quadricóptero simulado utilizando as equações dinâmicas apresentadas em (146)-(149). Entretanto o modelo apresentado é uma simplificação do modelo prático utilizado, pois desconsidera alguns efeitos aerodinâmicos e considera o UAV como sendo simétrico e equilibrado em seu centro de massa, algo que na prática é dificilmente alcançado.

Para corrigir em parte estes problemas vamos adicionar o efeito integrador no controle. Todavia o sistema já possui em sua dinâmica o efeito integrador, algo que pode ser observado com mais facilidade na equação simplificada do sistema com restrição de movimento em *pitch* (163)-(164). Ou seja, com a adição do termo integrador no controle, o sistema fica caracterizado como sendo um sistema sobre efeito de dois integradores, termo este definido na sessão 2.7.

Como visto na sessão 2.7, onde é explicado com maiores detalhes, os problemas que o algoritmo IFT sofre ao lidar com sistemas com integrador. O algoritmo IFT convencional apresenta um problema em sua convergência, onde o algoritmo tende a anular o termo integral quando realiza a sintonia destes sistemas.

Como o algoritmo CIFT é baseado no IFT ele herda o mesmo problema de convergência que dificulta a sintonia do quadricóptero. Foi observado nos experimentos práticos que utilizando o controlador com o termo integral, o algoritmo CIFT apresenta na sintonia o mesmo problema de convergência descrito acima, tende a convergir o termo integral para um valor próximo de zero. Devido a isto, é proposto neste trabalho utilizar somente controladores PD e fixar um valor arbitrado para o termo integral, este termo não será considerado na convergência e por conseguinte, o algoritmo CIFT incorporará esta dinâmica como se fosse parte do processo.

O procedimento para escolha do valor integral é por meio experimental, durante a convergência do algoritmo CIFT seu valor é acrescido conforme verificada uma queda considerável na função custo, os momentos de acréscimo são demonstrados na sessão de resultados desta sessão.

Vale a pena salientar, com finalidade de não criar maiores confusões, que o termo $\frac{\partial C_2}{\partial \rho_2}$ da equação (94), para este caso específico será dado por

$$\frac{\partial C_2}{\partial \rho_2} = \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}, \quad (165)$$

ou seja, como o termo integral é considerado constante, a derivada de $C_2(z, \rho_2)$ contempla somente os termos dos ganhos proporcional e derivativo.

A posição inicial para os parâmetros dos controladores PD das malhas interna e externa foram escolhidos como:

$$C_1 = \begin{bmatrix} 0,25 \\ 0 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \quad C_2 = \begin{bmatrix} 1,5 \\ 0 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}. \quad (166)$$

com estes controladores iniciais obtemos como resposta a saída angular $y_2(t)$ dada pela Figura 40.

Escolhemos como modelo de referência o sistema linear discreto definido como:

$$T_d(z, t) = \frac{0,00484z + 0,004687}{z^2 - 1,899z + 0,9085}, \quad (167)$$

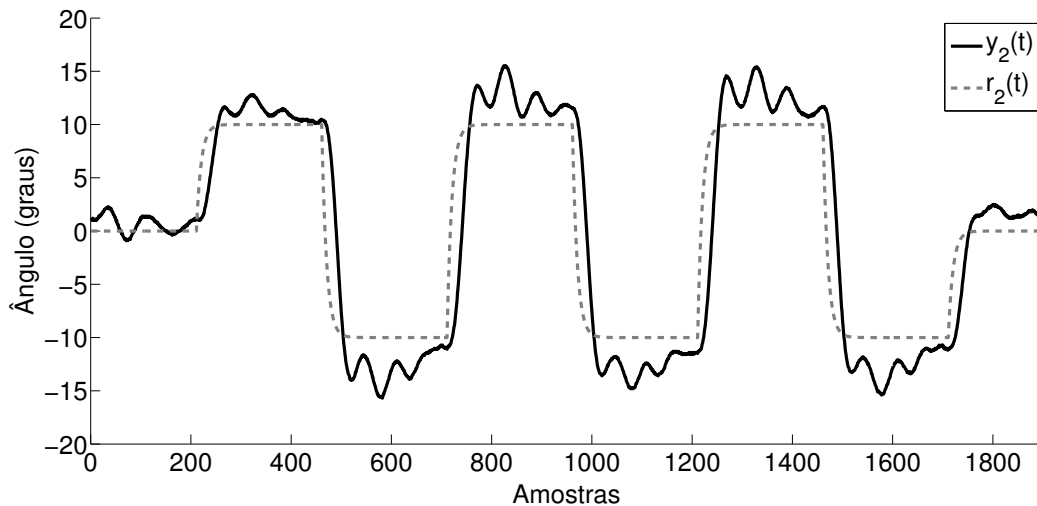


Figura 40: Resposta inicial da saída $y_2(t)$ ângulo, utilizando os controladores iniciais (166).

este modelo é obtido através da discretização de uma função de transferência de segunda ordem em tempo contínuo, utilizando o método *zero-order hold* com um período de amostragem de 0,01s. A função de transferência em tempo contínuo é dada por

$$T_d(s, t) = \frac{100}{s^2 + 9,6s + 100}. \quad (168)$$

a escolha de um modelo de segunda ordem como resposta desejada é devido a natureza do processo. Como demonstrado na Figura 40 o processo é bastante oscilatório, como medida conservadora decidimos por reduzir o sobressinal ao nível aceitável de 2 graus e diminuir o erro em regime permanente.

O sinal de referência adotado para o experimento é o resultado da filtragem de um sinal do tipo onda quadrada com período de 5s e amplitude de 10 graus como:

$$r_2(t) = 10 \frac{0,1}{z - 0,9} \text{square} \left(\frac{2\pi t}{5} \right), \quad (169)$$

todos os sinais e processos discretos utilizados no experimento tem período de amostragem de 0,01s.

5.4 Resultados

A convergência do algoritmo CIFT é realizada com uma pequena diferença dos exemplos realizados anteriormente, a diferença consiste em não utilizar o método *Steepest Descent* (18) nos experimentos. Observamos que o método Quasi-Newton (19) converge mesmo nas primeiras iterações, isto facilita o processo de convergência, pois o método Quasi-Newton converge com menos iterações comparado o método *Steepest Descent*.

É utilizada a aproximação da Hessiana demonstrada em (35). O passo escolhido para o método de convergência Quasi-Newton segue o definido em (23).

Nas Figuras 41 e 42 são demonstradas as respostas inicial, final e desejada para o ângulo *roll* de ambas as malhas, utilizando o valor fixo do ganho integral de 0,01. Nestas figuras percebemos que o comportamento obtido já começa a se assemelhar ao desejado, entretanto o erro em regime permanente permanece elevado.

Outro detalhe interessante na Figura 41 é que o ruído na malha interna de controle da velocidade angular, é mais visível que na malha externa de controle do ângulo Figura 42, duas explicações podem ser dadas para todo este ruído: primeiro, vibrações oriundas dos rotores são medidas pelos sensores; segundo, a informação do ângulo sofre um processamento digital mais elevado em comparação a informação da velocidade angular, devido a isto o sinal do ângulo tem este aspecto filtrado.

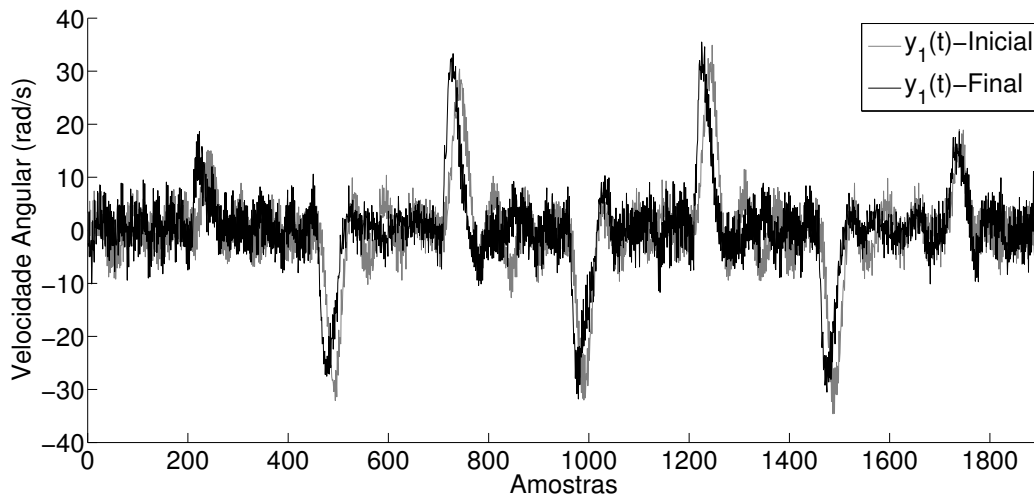


Figura 41: Resposta inicial com os controladores (166) e obtida após o uso do algoritmo CIFT para a saída da velocidade angular $y_1(t)$. Sendo o valor do termo integral de 0,01.

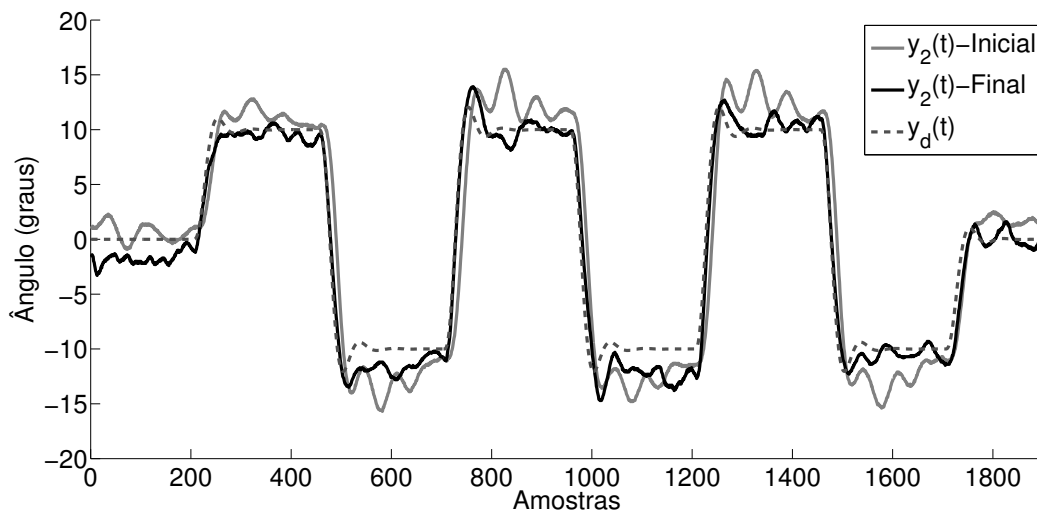


Figura 42: Resposta inicial com os controladores (166), respostas desejada $y_d(t)$ e obtida após o uso do algoritmo CIFT para a saída do ângulo $y_2(t)$. Sendo o valor do termo integral de 0,01.

Na Tabela 12 é demonstrada a convergência dos parâmetros dos controladores e seu respectivo custo agregado, utilizando o valor do ganho integral fixo de 0,01.

Com o término da primeira etapa dos experimento, utilizando o valor do ganho inte-

Tabela 12: Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o experimento prático utilizando quadricóptero real. Valor do termo integral de 0,01.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		$k_{p,1}$	$k_{d,1}$	$k_{p,2}$	$k_{d,2}$	
1	10,7745	0,2500	0,0000	1,5000	0,0000	QN
2	9,9094	0,2339	0,0055	1,5911	1,7665	QN
3	9,2690	0,1945	0,0199	1,8031	6,4950	QN
4	7,9713	0,1374	0,0652	2,1849	15,7710	QN
5	5,4768	0,1095	0,2429	2,7135	23,1586	QN
6	3,8289	0,1059	0,5407	3,2310	24,1985	QN
7	2,3369	0,1046	0,7154	3,6574	29,6420	QN
8	2,2816	0,1021	0,7507	4,4572	39,7880	QN
9	1,8867	0,0751	0,6275	6,2907	63,0767	QN

Tabela 13: Valor da função custo e convergência dos parâmetros dos controladores, malhas interna e externa, para o experimento prático utilizando quadricóptero real. Valor do termo integral de 0,1.

Ite	$J(\rho)$	$C_1(z)$		$C_2(z)$		Método
		$k_{p,1}$	$k_{d,1}$	$k_{p,2}$	$k_{d,2}$	
1	6,0890	0,0751	0,6275	6,2907	63,0767	QN
2	4,9008	0,0657	0,2273	4,1183	50,8743	QN
3	4,9891	0,0581	0,3442	4,7846	52,8966	QN
4	5,4560	0,0583	0,6180	5,4356	40,6063	QN
5	3,0883	0,0522	0,4310	6,4154	70,4375	QN
6	1,7042	0,0462	0,2893	8,0423	108,1760	QN
7	0,6032	0,0375	0,0533	10,2284	175,2910	QN
8	0,3321	0,0354	0,0938	11,5207	175,5750	QN

gral fixo 0,01, alcançamos os controladores dados por

$$C_1 = \begin{bmatrix} 0,0751 \\ 0,6275 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \quad C_2 = \begin{bmatrix} 6,2907 \\ 63,0767 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}. \quad (170)$$

Utilizamos os controladores obtidos ao final da primeira etapa de convergência equação (170), como parâmetros iniciais e realizamos uma nova série de experimentos, contudo aumentamos o valor do termo integral para 0,1. Somente os parâmetros iniciais e o valor do termo integral foram alterados, todos as outras variáveis do experimento como tamanho de passo e método de convergência foram mantidas constantes. Realizando esta nova série de experimentos obtemos a convergência demonstrada na Tabela 13.

Analisando os dados obtidos das convergências das Tabelas 12 e 13, percebemos que o experimento alcançou um resultado significativo na redução do custo, comparando o seu primeiro valor na Tabela 12 de 10,7745 com o seu último valor na Tabela 13 de 0,3321, calculamos uma redução aproximada de 33 vezes. A partir da iteração 8 utilizando o valor integral de 0,1, não foi obtida redução significativa na função custo e devido a isto, finalizamos o experimento.

Ao término desta segunda etapa de convergência o algoritmo CIFT alcança os contro-

ladores dados por

$$C_1 = \begin{bmatrix} 0,0354 \\ 0,0938 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix} \quad C_2 = \begin{bmatrix} 11,5207 \\ 175,5750 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z-1}{z} \end{bmatrix}, \quad (171)$$

com estes controladores obtemos como resposta final as saídas de ambas as malhas demonstradas nas Figuras 43 e 44. Com a mudança do ganho integral para o valor de 0,1 conseguimos uma redução do erro em regime permanente e uma redução considerável na função custo, comparado com valor de 0,01 usado na primeira série de experimentos.

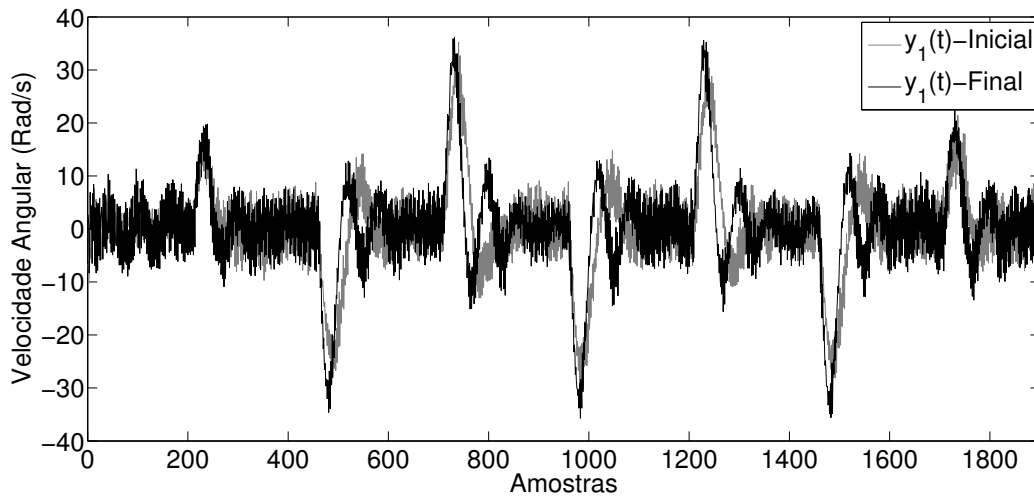


Figura 43: Resposta inicial com os controladores (170) e obtida após o uso do algoritmo CIFT para a saída da velocidade angular $y_1(t)$. Sendo o valor do termo integral de 0,1.

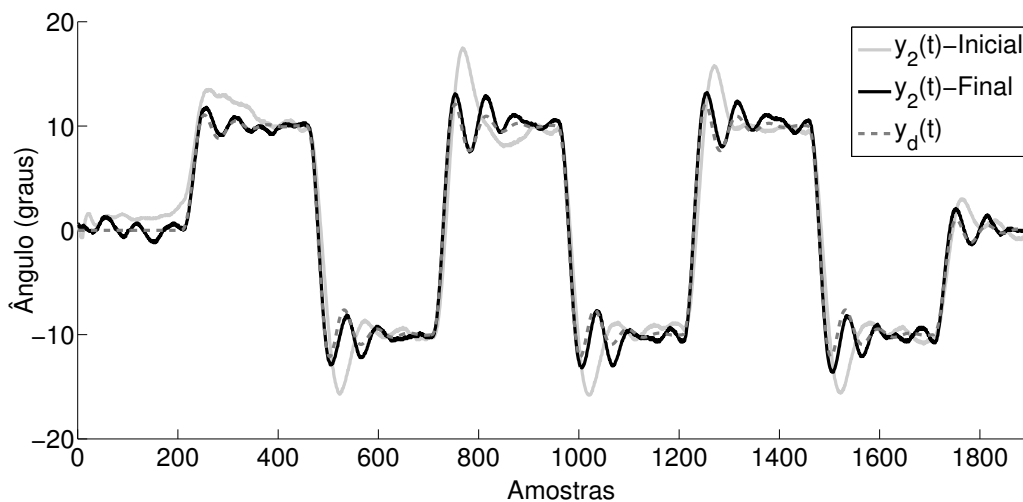


Figura 44: Resposta inicial com os controladores (170), respostas desejada $y_d(t)$ e obtida após o uso do algoritmo CIFT para a saída do ângulo $y_2(t)$. Sendo o valor do termo integral de 0,1.

Nas Figuras 45 e 46 demonstramos uma análise gráfica da convergência dos parâmetros de ambos controladores malhas interna e externa para ambas etapas dos experimentos, fazemos questão de adicionar o parâmetro do termo integral nos gráficos para melhor elucidar as convergências.

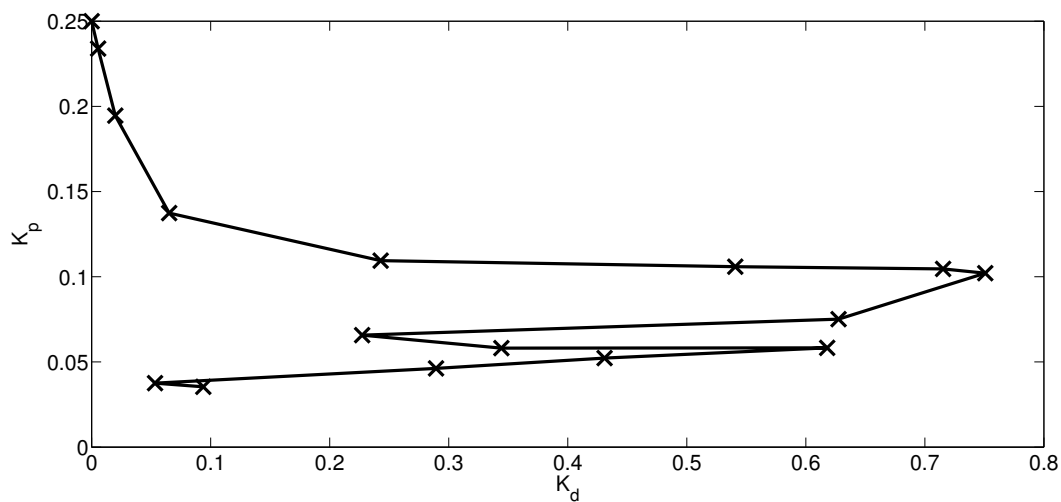


Figura 45: Convergência dos parâmetros do controlador $C_1(z)$ malha interna, para o experimento prático do quadricóptero.

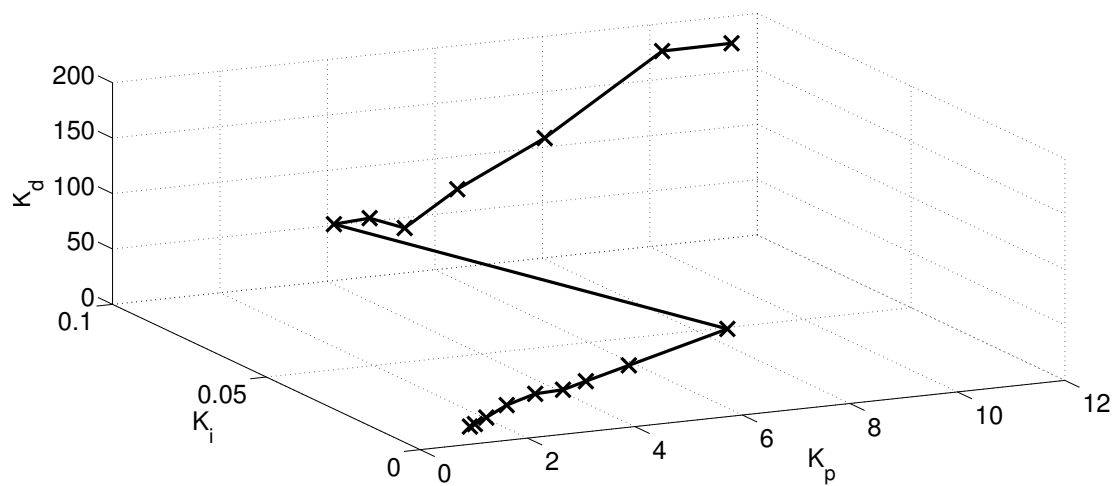


Figura 46: Convergência dos parâmetros do controlador $C_2(z)$ malha externa, para o experimento prático do quadricóptero.

6 CONCLUSÃO

A principal contribuição deste trabalho está na proposta de uma extensão baseada no método *Iterative Feedback Tuning* para sistemas em formato cascata com duas malhas de controle, nomeado como *Cascade Iterative Feedback Tuning* (CIFT). Foi demonstrado que o algoritmo em questão consegue realizar a convergência dos parâmetros dos controladores de forma simultânea para ambas as malhas interna e externa. Apresentamos exemplos numéricos que demonstram sua eficiência em situações com presença e ausência de ruído. Também realizamos um comparativo entre os métodos IFT e o CIFT, onde foi demonstrado que o algoritmo proposto alcança um desempenho elevado e uma rápida convergência em comparação ao método convencional de sintonia utilizando IFT.

Neste trabalho utilizamos como aplicação a sintonia de um UAV comum e muito conhecido na área de aeromodelismo, o chamado quadricóptero. Demonstramos através do modelamento matemático os estados da dinâmica que compõem o seu modelo, juntamente da descrição das forças atuantes, não linearidades e incertezas do modelo.

A aplicação prática do quadricóptero também foi realizada. Descrevemos todos os componentes e circuitos básicos que compõem o quadricóptero utilizado. Nele demonstramos todas as malhas e subsistemas de controle para uma correta estabilização em voo e realizamos o principal objetivo do trabalho, que é a sintonia de um quadricóptero real utilizando o método de controle baseado em dados CIFT.

Como trabalhos futuros, pode-se comparar o desempenho do algoritmo proposto com outros métodos de sintonia para estruturas em formato cascata. Além disso outras diferentes estruturas para o formato cascata podem ser consideradas no desenvolvimento do algoritmo, bem como a adição de mais malhas de controle. Outra possibilidade é tentar aplicar o mesmo desenvolvimento o qual se obteve o algoritmo CIFT e estendê-lo para outros métodos de controle baseados em dados, por exemplo VRFT.

REFERÊNCIAS

ADAMONE.RCHOMEPAGE.COM. **Estimate Propeller Static Thrust**. Disponível em: <<http://adamone.rchomepage.com/calc-thrust.html>>. Acesso em: 26 de janeiro de 2016.

ALFARO, V.; VILANOVA, R.; ARRIETA, O. Robust tuning of two-degree-of-freedom (2-DoF) PI/PID based cascade control systems. **Journal of Process Control**, Paris, v.19, n.10, p.1658–1670, 2009.

ALTUĞ, E.; OSTROWSKI, J. P.; MAHONY, R. Control of a quadrotor helicopter using visual feedback. In: ROBOTICS AND AUTOMATION (ICRA), 2002 IEEE INTERNATIONAL CONFERENCE ON, 2002, WASHINGTON. **Proceedings...**, New York: IEEE, 2002. v.1, p.72–77.

ANDERSON, B. D. O.; MOORE, J. B. **Linear optimal control**. Englewood Cliffs, New Jersey: Prentice Hall, 1971.

ARGENTIM, L. M. et al. PID, LQR and LQR-PID on a quadcopter platform. In: INFORMATICS, ELECTRONICS & VISION (ICIEV), 2013 INTERNATIONAL CONFERENCE ON, 2013, DHAKA. **Proceedings...**, New York: IEEE, 2009. p.1–6.

ÅSTRÖM, K. J. **Introduction to Stochastic Control Theory**. New York: Academic Press, 1970.

ÅSTRÖM, K. J.; WITTENMARK, B. On self tuning regulators. **Automatica**, New York, v.9, n.2, p.185–199, 1973.

BAZANELLA, A. S.; CAMPESTRINI, L.; ECKHARD, D. **Data-driven controller design: the H_2 approach**. Netherlands: Springer Science & Business Media, 2011.

BEARD, R. W. **Quadrotor dynamics and control**. Provo: Brigham Young University, 2008.

BITMEAD, R. R.; GEVERS, M.; WERTZ, V. **Adaptive optimal control: the thinking man's gpc**. New York: Prentice Hall International, 1990.

BOUABDALLAH, S.; NOTH, A.; SIEGWART, R. PID vs LQ control techniques applied to an indoor micro quadrotor. In: INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2004 IEEE/RSJ INTERNATIONAL CONFERENCE ON, 2004, SENDAI. **Proceedings...**, New York: IEEE, 2004. v.3, p.2451–2456.

BOYD, S. et al. **Linear matrix inequalities in system & control theory**. Philadelphia: Society for Industrial and Applied Mathematic, 1994.

CAMPI, M. C.; LECCHINI, A.; SAVARESI, S. M. Virtual reference feedback tuning: a direct method for the design of feedback controllers. **Automatica**, Elmsford, v.38, n.8, p.1337–1346, 2002.

CAUCHY, A. Méthode générale pour la résolution des systemes d'équations simultanées. **Comptes Rendus de Paris**, Paris, v.25, n.1847, p.536–538, 1847.

CORRIPIO, A. B. **Tuning of industrial control systems**. Maggie Valley: Isa, 2000.

COSTA, S. **Controlo e simulação de um quadrirotor convencional**. 2008. 103p. Dissertação (Mestrado em Engenharia Aeroespacial — Universidade Técnica de Lisboa, Lisbon, 2008.

COZA, C.; MACNAB, C. A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. In: FUZZY INFORMATION PROCESSING SOCIETY (NAFIPS), 2006 ANNUAL MEETING OF THE NORTH AMERICAN ON, 2006, MONTREAL. **Proceedings...**, New York: IEEE, 2006. p.454–458.

CRAIG, J. J. **Introduction to robotics: mechanics and control**. New Jersey: Pearson Prentice Hall, 2005.

ECKHARD, D. **Projeto de controladores baseado em dados** : convergência dos métodos iterativos. 2008. 94p. Dissertação (Mestrado em engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

ECKHARD, D.; BAZANELLA, A. S. Robust convergence of the steepest descent method for data-based control. **International Journal of Systems Science**, Hants, v.43, n.10, p.1969–1975, 2012.

GOLDSTEIN, A. A. Cauchy's method of minimization. **Numerische Mathematik**, Berlin, v.4, p.146–150, 1962.

GONGORA, M. et al. Robustness analysis of evolutionary controller tuning using real systems. In: EVOLUTIONARY COMPUTATION (CEC), 2009 IEEE CONGRESS ON, 2009, TRONDHEIM. **Proceedings...**, New York: IEEE, 2009. p.606–613.

GOODWIN, G. C.; SIN, K. S. **Adaptive filtering, prediction and control**. Englewood Cliffs, NJ, EUA: Prentice Hall, 1984.

GUPTE, S.; MOHANDAS, P. I. T.; CONRAD, J. M. A survey of quadrotor unmanned aerial vehicles. In: SOUTHEASTCON, 2012 PROCEEDINGS OF IEEE, 2012, ORLANDO. **Proceedings...**, New York: IEEE, 2012. p.1–6.

HJALMARSSON, H. Iterative feedback tuning - an overview. **International journal of adaptive control and signal processing**, Glasgow, v.16, n.5, p.373–395, 2002.

HJALMARSSON, H. From experiment design to closed-loop control. **Automatica**, Elmsford, v.41, n.3, p.393–438, 2005.

HJALMARSSON, H. et al. Iterative feedback tuning: theory and applications. **IEEE control systems magazine**, Piscataway, v.18, n.4, p.26–41, 1998.

HJALMARSSON, H.; GUNNARSSON, S.; GEVERS, M. A convergent iterative restricted complexity control design scheme. In: CONFERENCE ON DECISION AND CONTROL (CDC), 1994 33RD IEEE CONFERENCE, 1994, LAKE BUENA VISTA. **Proceedings...**, New York: IEEE, 1994. p.1735–1740.

HOBBYKING.COM. **F450 Multi-Rotor Air Frame**. Disponível em: <<http://www.amazon.com/Neewer-Black-Multi-Rotor-Frame-FlameWheel/dp/B00I0001V2>>. Acesso em: 26 de janeiro de 2016.

HOBBYKING.COM. **NTM Prop Drive Series 28-26 1100kv**. Disponível em: <<https://www.hobbyking.com/mobile/products.asp?idparentcat=520>>. Acesso em: 26 de janeiro de 2016.

ISEEROBOT.COM. **ESC Hobbywing UBEC SKYWALKER 2-3S Lipo 20A**. Disponível em: <<http://www.iseerobot.com/produk-1928-esc-hobbywing-ubec-skywalker-23s-lipo-20a.html>>. Acesso em: 26 de janeiro de 2016.

JENG, J.-C.; LIAO, S.-J. A simultaneous tuning method for cascade control systems based on direct use of plant data. **Industrial & Engineering Chemistry Research**, Washington, v.52, n.47, p.16820–16831, 2013.

KAMMER, L. C.; BITMEAD, R. R.; BARTLETT, P. L. Direct iterative tuning via spectral analysis. **Automatica**, Elmsford, v.36, n.9, p.1301–1307, 2000.

KARIMI, A.; MISKOVIC, L.; BONVIN, D. Iterative correlation-based controller tuning. **International journal of adaptive control and signal processing**, Glasgow, v.18, n.8, p.645–664, 2004.

KAYA, I.; TAN, N.; ATHERTON, D. P. Improved cascade control structure for enhanced performance. **Journal of Process Control**, London, v.17, n.1, p.3–16, 2007.

KIRLI, A. et al. Self tuning fuzzy PD application on TI TMS320F28335 for an experimental stationary quadrotor. In: EDUCATION AND RESEARCH CONFERENCE (EDERC), 2010 4TH EUROPEAN CONFERENCE, 2010, NICE. **Proceedings...**, New York: IEEE, 2010. p.42–46.

KRISHNASWAMY, P. R. et al. When to use cascade control. **Industrial & engineering chemistry research**, Washington, v.29, n.10, p.2163–2166, 1990.

LEE, Y.; PARK, S.; LEE, M. PID controller tuning to obtain desired closed loop responses for cascade control systems. **Industrial & engineering chemistry research**, Washington, v.37, n.5, p.1859–1865, 1998.

LIU, T.; GU, D.; ZHANG, W. Decoupling two-degree-of-freedom control strategy for cascade control systems. **Journal of Process Control**, Oxford, v.15, n.2, p.159–167, 2005.

LJUNG, L. **System identification: theory for the user**. Englewood Cliffs: Prentice Hall, 1987.

MADANI, T.; BENALLEGUE, A. Backstepping control for a quadrotor helicopter. In: INTELLIGENT ROBOTS AND SYSTEMS (RSJ), 2006 INTERNATIONAL CONFERENCE ON, 2006, BEIJING. **Proceedings...**, New York: IEEE, 2006. p.3255–3260.

MAHONY, R.; KUMAR, V.; CORKE, P. Multicopter aerial vehicles: modeling, estimation, and control of quadrotor. **IEEE Robotics & Automation Magazine**, Saint Paul, n.19, p.20–32, 2012.

MCCORMICK, B. W. **Aerodynamics, aeronautics, and flight mechanics**. New York: Wiley, 1995.

MULTICOPTERS.CO.UK. **MultiWii 328P Flight Controller**. Disponível em: <<https://www.multicopters.co.uk/multiwii-328p-flight-controller-with-ftdi-dsm2-port.html>>. Acesso em: 26 de janeiro de 2016.

PROCHAZKA, H. et al. Iterative feedback tuning for robust controller design and optimization. In: CONFERENCE ON DECISION AND CONTROL AND EUROPEAN CONTROL (CDC-ECC), 2005 44TH IEEE CONFERENCE, 2005, SEVILLE. **Proceedings...**, New York: IEEE, 2005. p.3602–3607.

PURI, A. A survey of unmanned aerial vehicles (UAV) for traffic surveillance. **Department of computer science and engineering, University of South Florida**, Tampa, 2005.

RAFFO, G. V.; ORTEGA, M. G.; RUBIO, F. R. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. **Automatica**, Washington, v.46, n.1, p.29–39, 2010.

RALSTON, A.; RABINOWITZ, P. **A first course in numerical analysis**. Mineola: McGraw-Hill, 1965.

RESNICK, R.; HALLIDAY, D.; WALKER, J. **Fundamentals of physics**. New York: Wiley, 1988.

ROBBINS, H.; MONRO, S. A Stochastic Approximation Method. **The annals of mathematical statistics**, Beachwood, v.22, n.3, p.400–407, 1951.

SALIH, A. L. et al. Flight PID controller design for a UAV quadrotor. **Scientific Research and Essays**, Lagos, v.5, n.23, p.3660–3667, 2010.

SHI, G.; SKELTON, R. E. Markov data-based LQG control. **Journal of dynamic systems, measurement, and control**, Texas, v.122, n.3, p.551–559, 2000.

SOLARI, G.; GEVERS, M. Unbiased estimation of the Hessian for iterative feedback tuning (IFT). In: CONFERENCE ON DECISION AND CONTROL (CDC), 2004 43RD IEEE CONFERENCE, 2004, PARADISE ISLAND. **Proceedings...**, New York: IEEE, 2004. v.2, p.1759–1760 Vol.2.

STEVENS, B. L.; LEWIS, F. L. **Aircraft control and simulation**. New York: John Wiley & Sons, 2003.