

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**GUILHERME LOPES SILVEIRA**

**PROJETO DE DIPLOMAÇÃO**

**ANÁLISE COMPARATIVA DE MÉTODOS DE  
CONTROLE BASEADO EM DADOS**

Porto Alegre

2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**ANÁLISE COMPARATIVA DE MÉTODOS DE  
CONTROLE BASEADO EM DADOS**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: MSc. Manoel Fabricio Flores da Cunha

Porto Alegre

2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**GUILHERME LOPES SILVEIRA**

**ANÁLISE COMPARATIVA DE MÉTODOS DE  
CONTROLE BASEADO EM DADOS**

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. MSc. Manoel Fabricio Flores da Cunha, UFRGS

Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof. MSc. Manoel Fabricio Flores da Cunha, UFRGS

Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dra. Lucíola Campestrini, UFRGS

Doutora pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Eng. Everton Luis Granado Ghignatti Júnior, NSCAD

Bacharel pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Porto Alegre, dezembro de 2011

## RESUMO

Um bom controlador é aquele que faz com que o sistema atenda às especificações de operação. A partir dessa idéia, métodos de projeto de controladores baseados em dados buscam projetar um controlador que minimize algum critério de desempenho pré-determinado utilizando apenas dados coletados do sistema em operação. Não é mais necessário o conhecimento do modelo da planta, o qual se torna uma ferramenta desnecessária no projeto do controlador. Este trabalho apresenta uma comparação entre dois métodos de controle baseados em dados, sendo eles o Virtual Reference Feedback Tuning (VRFT) e o Iterative Feedback Tuning (IFT), aplicados no problema prático de controle de um motor de corrente contínua. A aplicação prática é demonstrada desde a estruturação conceitual e física até os resultados bem sucedidos no problema de sintonização de um controlador para o sistema com o uso de ambos os métodos. São analisados os resultados obtidos bem como as particularidades que cada método proporciona.

**Palavras-chaves: Sistemas de controle, Métodos de controle baseado em dados, Virtual Reference Feedback Tuning, Iterative Feedback Tuning .**

## **ABSTRACT**

A good controller is one which makes the system achieve the operating specifications. From this idea, data-based controller design methods seek to design a controller that minimizes some pre-determined performance criterion using only data collected from the system in operation. It is no longer necessary to know the plant model, which becomes an unnecessary tool in the controller design. This paper presents a comparison between two data-based control methods, Virtual Reference Feedback Tuning (VRFT) and Iterative Feedback Tuning (IFT), applied to the practical problem of controlling a DC motor. The practical application is demonstrated from the conceptual and physical structure to the successful results in the problem of tuning a controller for the system using both methods. The results are analyzed as well as the peculiarities that each method provides.

**Keywords: Control systems, Data based control methods, Virtual Reference Feedback Tuning, Iterative Feedback Tuning.**

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
<b>2</b>	<b>ITERATIVE FEEDBACK TUNING .....</b>	<b>16</b>
<b>2.1</b>	<b>Revisão Bibliográfica .....</b>	<b>16</b>
<b>2.2</b>	<b>Desenvolvimento do método.....</b>	<b>16</b>
<b>2.2.1</b>	<b>Critério de Minimização .....</b>	<b>19</b>
<b>2.2.2</b>	<b>Estimativa do gradiente.....</b>	<b>21</b>
<b>2.2.3</b>	<b>O algoritmo .....</b>	<b>22</b>
<b>2.2.4</b>	<b>Convergência.....</b>	<b>22</b>
<b>3</b>	<b>VIRTUAL REFERENCE FEEDBACK TUNING.....</b>	<b>24</b>
<b>3.1</b>	<b>Revisão bibliográfica .....</b>	<b>24</b>
<b>3.2</b>	<b>Considerações iniciais.....</b>	<b>25</b>
<b>3.2.1</b>	<b>O problema de controle do modelo de referência .....</b>	<b>27</b>
<b>3.3</b>	<b>Desenvolvimento do método.....</b>	<b>28</b>
<b>3.4</b>	<b>Algoritmo .....</b>	<b>30</b>
<b>4</b>	<b>ARQUITETURA DO SISTEMA .....</b>	<b>32</b>
<b>4.1</b>	<b>Hardware Utilizado .....</b>	<b>32</b>
<b>4.2</b>	<b>Softwares utilizados .....</b>	<b>34</b>
<b>4.3</b>	<b>Estruturação do método IFT .....</b>	<b>35</b>
<b>4.4</b>	<b>Estruturação do método VRFT .....</b>	<b>37</b>
<b>4.5</b>	<b>Modelo de Referência .....</b>	<b>37</b>
<b>5</b>	<b>RESULTADOS .....</b>	<b>38</b>
<b>5.1</b>	<b>Método IFT.....</b>	<b>39</b>
<b>5.2</b>	<b>Método VRFT.....</b>	<b>45</b>
<b>6</b>	<b>CONCLUSÃO.....</b>	<b>51</b>

<b>6.1</b>	<b>Conclusões .....</b>	<b>51</b>
<b>6.2</b>	<b>Sugestões para trabalhos futuros .....</b>	<b>52</b>
<b>ANEXO A:</b>	<b>ROTINA PARA O CÁLCULO DO VETOR DE PARÂMETROS DO CONTROLADOR ATRAVÉS DO MÉTODO VRFT .....</b>	<b>54</b>
<b>ANEXO B:</b>	<b>ROTINA PARA O CÁLCULO DO VETOR DE PARÂMETROS DO CONTROLADOR ATRAVÉS DO MÉTODO IFT .....</b>	<b>56</b>
<b>ANEXO C:</b>	<b>FUNÇÃO PIDIFT.M .....</b>	<b>58</b>
<b>REFERÊNCIAS</b>	<b>.....</b>	<b>63</b>

## LISTA DE ILUSTRAÇÕES

Figura 1:	Elementos conceituais de um sistema de controle.....	13
Figura 2:	Plataforma didática QET- DCMCT.....	15
Figura 3:	Diagrama de blocos do sistema considerado no método IFT.....	17
Figura 4:	Diagrama de blocos do sistema considerado para o método VRFT.....	25
Figura 5:	Indicação dos elementos da plataforma QET- DCMCT.....	32
Figura 6:	Topologia dos componentes do sistema e suas interações.....	34
Figura 7:	Sistema de ensaios para o desenvolvimento do método IFT.....	35
Figura 8:	Fluxograma da operação do sistema.....	36
Figura 9:	Sistema de ensaios para o desenvolvimento do método VRFT.....	37
Figura 10:	Sinal de referência utilizado nos ensaios.....	38
Figura 11:	Progressão do custo versus número de iterações.....	40
Figura 12:	Progressão do passo $\gamma$ versus número de iterações.....	40
Figura 13:	Progressão do ganho proporcional $K_p$ versus número de iterações.....	41
Figura 14:	Progressão do ganho proporcional $K_i$ versus número de iterações.....	41
Figura 15:	Ação de controle utilizando-se os parâmetros iniciais do problema (linha traço-ponto), após três iterações (linha tracejada) e após nove iterações (linha contínua) com a utilização dos parâmetros obtidos com o uso do algoritmo IFT. .....	42
Figura 16:	Progressão do sinal de saída para o uso dos parâmetros: iniciais (linha traço-ponto), após uma iteração (linha tracejada), após três iterações (linha pontilhada espessa) e após nove iterações (linha contínua).....	43

Figura 17:	Comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua). .....	44
Figura 18:	Detalhe da figura 17 apresentando a comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua). .....	44
Figura 19:	Progressão do custo ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT. ....	46
Figura 20:	Progressão do ganho proporcional $K_p$ ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT. ....	46
Figura 21:	Progressão do ganho proporcional $K_i$ ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT. ....	47
Figura 22:	Progressão da ação de controle utilizando-se os parâmetros iniciais do problema (linha traço-ponto), ação de controle obtida no primeiro ensaio (linha tracejada) e a ação de controle obtida após nove ensaios para sintonizar o controlador (linha contínua). .....	47
Figura 23:	Detalhe apresentando a comparação da ação de controle obtida no primeiro ensaio (linha pontilhada), após uma segunda sintonização (linha tracejada) e após nove sintonizações sucessivas (linha contínua), com o uso do método VRFT. ....	48
Figura 24:	Comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a	

utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua). .....49

Figura 25: Detalhe da figura 25 figura 17 apresentando o sinal de saída do sistema após uma única sintonia do método VRFT (linha tracejada) e após nove sintonias sucessivas (linha tracejada) em contraste com a saída desejada do sistema (linha contínua). .....50

## LISTA DE TABELAS

Tabela 1	Nomenclatura dos elementos componentes da plataforma QET-DCMCT.....	33
Tabela 2	Descrição das variáveis apresentadas na figura 6 .....	35
Tabela 3	Parâmetros do sistema e custo ao longo das iterações realizadas para o método IFT.....	39
Tabela 4	Parâmetros do sistema e custo ao longo das iterações realizadas para o método VRFT .....	45

## LISTA DE ABREVIATURAS

DA: Conversor de Dados Digital-Analógico

DAQ: Data Acquisition

DCMCT: DC Motor Control Trainer

IFT: Iterative Feedback Tuning

PI: Proporcional-Integral

PID: Proporcional-Integral-Derivativo

QET: Quanser Engineering Trainer

QIC: Quanser Integrated Controller

QuaRC: Quanser Real-Time Control

SISO: Single-Input and Single-Output

UFRGS: Universidade Federal do Rio Grande do Sul

USB: Universal Serial Bus

VRFT: Virtual Reference Feedback Tuning

## 1 INTRODUÇÃO

Um sistema de controle tem por objetivo fundamental garantir com que o sinal de saída de outro sistema assuma um comportamento previamente especificado. Esse outro sistema a ser controlado é chamado de processo ou planta e a sua saída é chamada de variável controlada ou então variável de processo. Já o sinal de entrada do processo é chamado de sinal de controle ou variável manipulada por se tratar de um sinal passível de ser manipulado (BAZANELLA, 2005). Tais entidades relevantes de um sistema de controle são apresentadas conforme figura 1.

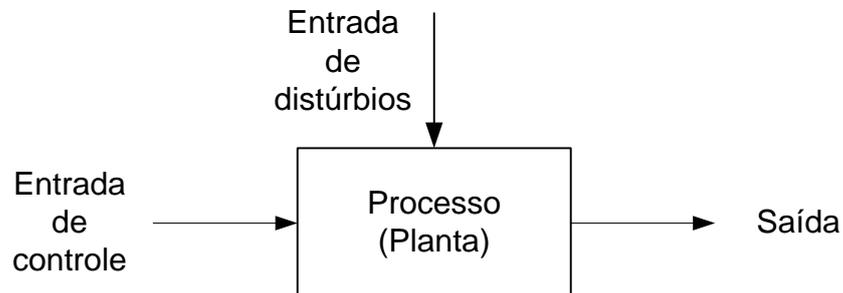


Figura 1: Elementos conceituais de um sistema de controle.

Sistemas de controle são estudados desde o século XVIII quando James Watt desenvolveu um sistema de controle de velocidade para uma máquina a vapor com o uso de um regulador centrífugo. O desenvolvimento da teoria de controle vem sendo aprimorada desde então através de métodos e ferramentas que possibilitem sistemas de controle que possam executar o maior número de operações da maneira mais robusta possível (OGATA, 2003).

Logo em uma fase inicial do desenvolvimento da teoria de controle foi provado que é possível determinar a estabilidade de sistemas a partir de sua modelagem através de equações diferenciais (MINORSKY, 1922). Posteriormente foi desenvolvido um procedimento que determina a estabilidade de um sistema quando em malha fechada através de ensaios

realizados sobre a planta em malha aberta. O procedimento consiste em aplicar ondas senoidais estacionárias no sistema em malha aberta com o intuito de traçar um diagrama polar da sua resposta em frequência (NYQUIST, 1932). Outra importante contribuição foi o desenvolvimento pleno do método da alocação de pólos através do lugar das raízes, que consiste em realizar uma análise gráfica de como os pólos do sistema irão se comportar à medida que os parâmetros do sistema são alterados (EVANS,1948).

Entretanto o processo por muitas vezes tem um alto nível de complexidade e não-linearidades, o que requer a desconsideração de propriedades físicas próprias desse sistema de tal forma que uma modelagem matemática seja possível. Nesse caso o modelo matemático do processo é uma aproximação simplificada do comportamento real e ele poderá não ser válido para todas as condições de operação (OGATA, 2003).

Tendo-se em mente que o objetivo no projeto de um controlador é a concepção do próprio controlador e a modelagem matemática do processo não é uma etapa essencial (GEVERS, 2002) foram então desenvolvidos métodos que fazem com que a identificação do processo seja uma etapa desnecessária para o projeto, tais como o *Virtual Reference Feedback Tuning* (VRFT) e o *Iterative Feedback Tuning* (IFT). Tais métodos se propõem a sintonizar um controlador de estrutura pré-fixada operando na planta em malha fechada, através da minimização de critérios de desempenho que utilizam dados adquiridos de pontos de interesse do sistema. É dessa metodologia, que utiliza dados do sistema em operação dispensando o modelo do processo, que surge o conceito de controle baseado em dados.

Motivado pela abordagem de projeto de controladores baseados em dados e pela carência de literatura que apresente aplicações práticas em sistemas mecânicos, este trabalho se propõe a realizar uma análise comparativa entre dois métodos de projeto de controladores, sendo eles o *Virtual Reference Feedback Tuning* (VRFT) e o *Iterative Feedback Tuning* (IFT). Para a aplicação dos métodos e análise de resultados será utilizada a plataforma

didática *Quanser Engineering Trainer (QET) DC Motor Control Trainer (DCMCT)* produzida pelo grupo *Quanser Consulting of Markham* apresentada na figura 2.

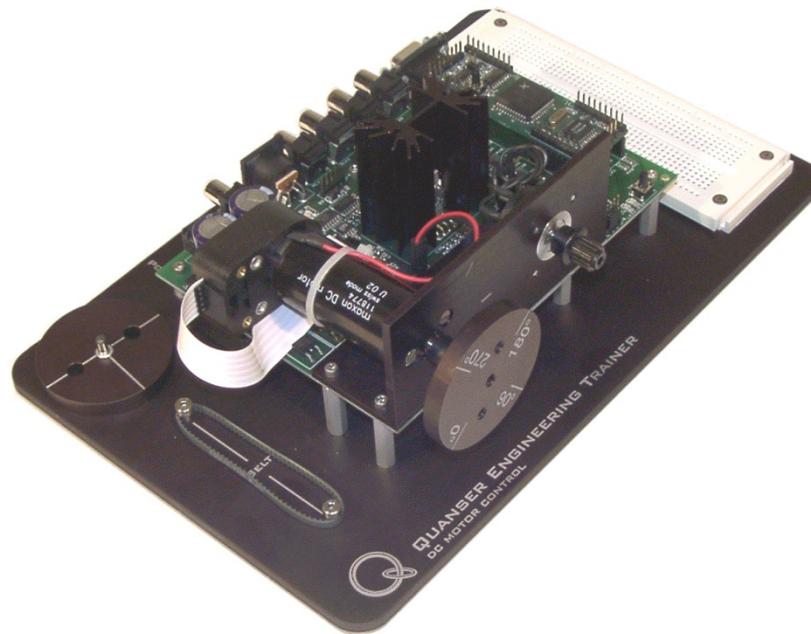


Figura 2: Plataforma didática QET- DCMCT.

O presente trabalho foi inicialmente desenvolvido baseado na programação do micro controlador presente na plataforma didática de forma a realizar a aquisição de dados, o controle do motor e a comunicação com o computador. Devido à dificuldades técnicas na comunicação da plataforma com o computador e a própria programação do micro controlador se fez necessária uma abordagem alternativa para a validação desses resultados. Assim sendo, optou-se pelo uso de uma placa de aquisição de dados e o software *Quanser Real-Time Controller (QuaRC)*, o qual realiza a comunicação e controle do sistema com o computador, possibilitando a realização desse trabalho. Essa nova abordagem é apresentada ao longo deste trabalho.

## 2 ITERATIVE FEEDBACK TUNING

### 2.1 REVISÃO BIBLIOGRÁFICA

O método Iterative Feedback Tuning (IFT) visa sintonizar um controlador através de uma minimização de um critério estabelecido com o uso de um método de gradiente. Através de um experimento específico na planta, dados de entrada e saída da planta são coletados e uma estimativa do gradiente da função do critério de desempenho pode ser obtida.

Para controladores com um grau de liberdade são necessários dois experimentos distintos, sendo que o segundo é um experimento “especial” a ser realizado; o primeiro é um experimento que tem por função adquirir dados de interesse durante condições de operação normal da planta. Esse segundo experimento consiste em realimentar o sinal de saída coletado durante a operação normal (coletado na primeira batelada de experimentos) juntamente com o sinal de referência sendo aplicado no sistema nesse segundo experimento. Da operação de realimentação do sistema com dados provenientes de experimentos anteriores surge o título de *Iterative Feedback Tuning* (IFT) (HJALMARSSON et al., 1998).

### 2.2 DESENVOLVIMENTO DO MÉTODO

Para o desenvolvimento do trabalho considera-se um sistema real em tempo discreto, cujo modelo da planta é desconhecido, descrito por

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

Nessa consideração  $G(z)$  é o modelo desconhecido da planta,  $v(t)$  é um distúrbio não-mensurável do processo,  $t$  representa os instantes de tempo discretos e  $z$  é o operador de avanço no tempo.

O distúrbio  $v(t)$  é considerado como sendo um processo estocástico estacionário no sentido amplo com média nula, o qual pode ser definido como  $v(t) = H(z)w(t)$ , sendo  $w(t)$  um ruído branco com variância  $\sigma_w^2$ .

Para o trabalho a ser desenvolvido considera-se um sistema que possui o controlador com um grau de liberdade e o sistema de controle é apresentado na figura 2 (FLORES DA CUNHA, 2010).

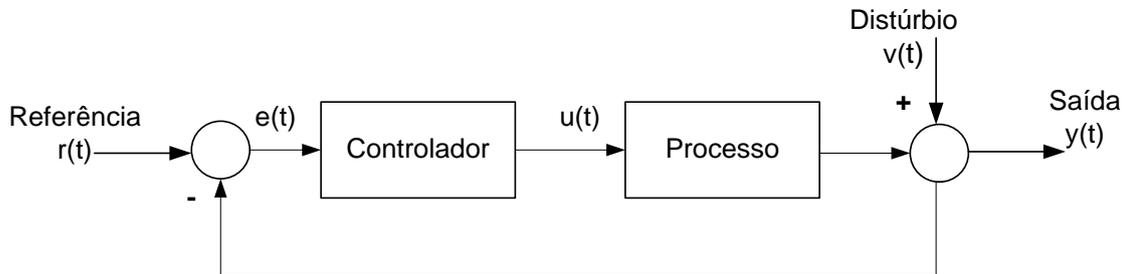


Figura 3: Diagrama de blocos do sistema considerado no método IFT.

Tem-se pelo diagrama de blocos o sinal de saída do controlador, chamado de ação de controle, como sendo:

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

O controlador apresentado em (2) pode ser parametrizado por um vetor de parâmetros  $\rho \in \mathfrak{R}^n$  como a seguinte função de transferência linear e invariante no tempo:

$$C(\rho, z) = \rho^T \bar{C}(z), \quad (3)$$

sendo que  $\bar{C}(z)$  é um vetor coluna de funções racionais.

Seja  $y_d(t)$  a resposta desejada do sistema a um sinal de referência  $r(t)$  para o sistema em laço fechado, essa resposta pode ser definida como sendo a saída de um modelo de referência  $T_d(z)$  como em

$$y_d(t) = T_d(z)r(t) \quad (4)$$

O erro entre a saída real  $y(\rho)$  e a saída desejada  $y_d(\rho)$  é

$$\begin{aligned} \tilde{y}(\rho, t) &\triangleq y(\rho, t) - y_d(\rho) = \\ &= \left( \frac{C(\rho, z)G(z)}{1 + C(\rho, z)G(z)} r(t) - y_d \right) + \frac{1}{1 + C(\rho, z)G(z)} v(t) \end{aligned} \quad (5)$$

Tomando-se o modelo de referência como definido em (4), pode-se reescrever a equação (5) como sendo:

$$\tilde{y}(\rho, t) = \left( \frac{C(\rho, z)G(z)}{1 + C(\rho, z)G(z)} - T_d(z) \right) r(t) + \frac{1}{1 + C(\rho, z)G(z)} v(t) \quad (6)$$

O erro formulado em (6) consiste devido a dois fatores: o erro de seguimento de referência do sinal  $r(t)$  e o erro devido ao acréscimo do distúrbio  $v(t)$  no sistema.

É intuitivo que se formule um objetivo de projeto de controle sustentado na idéia de minimização de um critério de desempenho baseado no erro  $\tilde{y}(\rho, t)$  apresentado em (6). Para essa decisão é escolhido o erro quadrático conforme a seguir:

$$J(\rho) = \frac{1}{2N} \mathbf{E} \left[ \sum_{i=1}^N \left( L_y(z) \tilde{y}(\rho, t) \right)^2 + \lambda \left( L_u(z) u(\rho, t) \right)^2 \right] \quad (7)$$

Na equação (7)  $\mathbf{E}[\cdot]$  denota a esperança em relação ao distúrbio estacionário no sentido amplo  $v(t)$ .  $L_y(z)$  e  $L_u(z)$  são pré-filtros que possibilitam que o conjunto de dados obtidos possa ser selecionado de forma a corresponder a somente determinadas frequências de interesse. No trabalho apresentado os filtros são considerados passa-tudo, optando-se por trabalhar com o conjunto universo dos dados.  $\lambda$  é o termo relativo ao esforço de controle e é proveniente da formulação original do controlador Linear Quadrático Gaussiano (LQG) e será considerado nulo no presente trabalho por se tratar de uma análise completa do método e não uma aplicação específica, o que inviabilizaria uma análise quantitativa dos dados, além de o método IFT não ser dependente de tal termo. Se for considerado apenas o erro entre a saída real e a saída desejada, o problema anteriormente citado não ocorre (FLORES DA CUNHA, 2010) e o desenvolvimento completo do método considerando o termo de esforço de controle pode ser encontrado em HJALMARSSON et al. (1998).

O objetivo do projeto de controladores é encontrar o vetor de parâmetros tal que o controlador utilizado faça com que o sistema atue o mais similarmente ao desejado. Tal vetor de parâmetros ótimo é encontrado, para o método IFT, minimizando-se (7) conforme:

$$\rho^* = \arg \min_{\rho} (J(\rho)) \quad (8)$$

Sendo  $T_0(\rho, z)$  e  $S_0(\rho, z)$  as funções de transferência de malha fechada e de sensibilidade do sistema com a atuação do controlador  $C(\rho, z)$ , respectivamente, tem-se:

$$T_0(\rho, z) = \frac{C(\rho, z)G(z)}{1 + C(\rho, z)G(z)}, \quad (9)$$

$$S_0(\rho, z) = \frac{1}{1 + C(\rho, z)G(z)}. \quad (10)$$

Dessa forma a saída do sistema real pode ser descrita como:

$$y(t) = T_0(\rho, z)r(t) + S_0(\rho, z)v(t). \quad (11)$$

Dada a independência entre os processos  $r(t)$  e  $v(t)$ ,  $J(\rho)$  pode ser reescrito da seguinte forma:

$$J(\rho) = \frac{1}{2N} \sum_{i=1}^N \{y_d(t) - T_0(\rho, z)r(t)\}^2 + \frac{1}{2N} E[S_0(\rho, z)v(t)]. \quad (12)$$

### 2.2.1 Critério de Minimização

A minimização de  $J(\rho)$  apresentada em (8) tem como objetivo encontrar o vetor de parâmetros  $\rho$  para um controlador com estrutura pré-especificada. Para encontrar o mínimo de  $J(\rho)$  é necessário encontrar uma solução para  $\rho$  considerando a equação:

$$0 = \frac{\partial J}{\partial \rho}(\rho) = \frac{1}{N} E \left[ \sum_{i=1}^N \tilde{y}(\rho, t) \frac{\partial \tilde{y}(\rho, t)}{\partial \rho}(\rho, t) \right]. \quad (13)$$

Partindo-se da definição de erro apresentada em (5), pode-se perceber que  $\tilde{y}(\rho, t)$  é a diferença entre a saída real obtida com o controlador operando no sistema e a saída desejada levando-se em consideração o modelo de referência proposto em (4). Em tal modelo de referência, a saída desejada não possui nenhum termo dependente dos parâmetros do controlador e pode-se escrever:

$$\frac{\partial \tilde{y}(\rho, t)}{\partial \rho} = \frac{\partial y(\rho, t)}{\partial \rho} \quad (14)$$

Resolvendo-se o termo à direita da equação (14), tem-se

$$\frac{\partial y(\rho, t)}{\partial \rho} = \frac{\frac{\partial C(\rho, z)}{\partial \rho} G(z) - \frac{\partial C(\rho, z)}{\partial \rho} C(\rho, z) G^2(z)}{[1 + C(\rho, z) G(z)]^2} r(t) - \frac{\frac{\partial C(\rho, z)}{\partial \rho} G(z)}{[1 + C(\rho, z) G(z)]^2} v(t) \quad (15)$$

$$\frac{\partial y(\rho, t)}{\partial \rho} = \frac{\partial C(\rho, z)}{\partial \rho} \left[ \frac{T_0(\rho, z)}{C(\rho, z)} - \frac{T_0^2(\rho, z)}{C(\rho, z)} \right] r(t) - \frac{\partial C(\rho, z)}{\partial \rho} \frac{1}{C(\rho, z)} T_0(\rho, z) S_0(\rho, z) v(t) \quad (16)$$

$$\frac{\partial y(\rho, t)}{\partial \rho} = \frac{\partial C(\rho, z)}{\partial \rho} \frac{1}{C(\rho, z)} [T_0(\rho, z) r(t) - T_0^2(\rho, z) r(t) - T_0(\rho, z) S_0(\rho, z) v(t)]. \quad (17)$$

Tomando-se a equação (11) e multiplicando-se por  $T_0(\rho, z)$ , obtém-se

$$T_0(\rho, z) y(t) = T_0^2(\rho, z) r(t) + T_0(\rho, z) S_0(\rho, z) v(t). \quad (18)$$

Aplicando a equação (18) em (17) é possível reescrevê-la como

$$\frac{\partial y(\rho, t)}{\partial \rho} = \frac{\partial C(\rho, z)}{\partial \rho} \frac{1}{C(\rho, z)} [T_0(\rho, z) \{r(t) - y(t)\}]. \quad (19)$$

Conforme formulação do método, o controlador que atua na planta é de estrutura pré-fixada e, portanto, conhecida. Dessa forma os termos  $\frac{\partial C(\rho, z)}{\partial \rho}$  e  $\frac{1}{C(\rho, z)}$  também o são, o que possibilita que o termo  $T_0(\rho, z) r(t) \{r(t) - y(t)\}$  seja obtido através de dois experimentos. O sinal de saída  $y(t)$ , adquirido no primeiro experimento, é subtraído do sinal de referência  $r(t)$  e esse novo sinal obtido é então utilizado como a própria referência no segundo ensaio. Para os dois experimentos a serem realizados consideram-se:

$$r^1 = r(t); y^1(\rho, t) = T_0(\rho, z) r(t) + T_0(\rho, z) v^1(t) \quad (20)$$

$$r^2 = r(t) - y^1(\rho, t); y^2(\rho, t) = T_0(\rho, z) \{r(t) - y^1(\rho, t)\} + S_0(\rho, z) v^2(t) \quad (21)$$

onde  $y^1(\rho, t)$  e  $y^2(\rho, t)$  são as saídas do sistema no primeiro e segundo experimento, respectivamente, e  $r^1(\rho, t)$  e  $r^2(\rho, t)$  são os sinais de referência a serem aplicados no sistema no primeiro e segundo experimento, respectivamente. Os termos  $v^1(t)$  e  $v^2(t)$  são os distúrbios que atuam no sistema durante o primeiro e segundo experimento, respectivamente. Assume-se que tais distúrbios são mutuamente independentes, devido ao fato que a duração de um experimento tende a ser maior do que o tempo de correlação dos distúrbios.

Esses dois experimentos permitem, então, o cálculo exato do sinal de erro  $\tilde{y}(\rho, t)$ , sendo ele

$$\tilde{y}(\rho, t) = y^1(\rho, t) - y_d(\rho), \quad (22)$$

que é a equação que descreve o erro para o método IFT aplicado ao problema de sintonia de controlador considerando o termo de esforço de controle nulo (FLORES DA CUNHA, 2010).

### 2.2.2 Estimativa do gradiente

A estimativa do gradiente de  $J(\rho)$  pode ser obtida, com o uso dos sinais definidos anteriormente, inicialmente aplicando-se a equação (21) em (19):

$$\frac{\partial y(\rho, t)}{\partial \rho} = \frac{\partial C(\rho, z)}{\partial \rho} \frac{1}{C(\rho, z)} y^2(\rho, t). \quad (23)$$

Ao se comparar a equação obtida em (23) com a equação (19), obtém-se

$$est \left[ \frac{\partial y(\rho, t)}{\partial \rho} \right] = \frac{\partial y(\rho, t)}{\partial \rho} + \frac{\partial C(\rho, z)}{\partial \rho} S_0(\rho, z) v^2(t). \quad (24)$$

Conforme apresentado em (23) a equação obtida é uma estimativa da taxa de variação do sinal de saída do sistema, levando-se em consideração o acréscimo do distúrbio  $v^2(t)$  no sistema, que nada mais é do que a estimativa do gradiente. Como  $y(\rho, t)$  pode ser visualizada como uma relação entre o sinal de erro  $\tilde{y}(\rho, t)$  e o sinal desejado  $y_d(\rho)$ , é possível se obter estimativa do gradiente da função custo  $J(\rho)$ :

$$est \left[ \frac{\partial J(\rho)}{\partial \rho} \right] = \frac{1}{N} \sum_{j=1}^N \tilde{y}(\rho, t) est \left[ \frac{\partial y(\rho, t)}{\partial \rho} \right]. \quad (25)$$

É necessário, para que a aproximação estocástica desse método ocorra corretamente, que a estimativa demonstrada em (24) seja não-polarizada:

$$\mathbf{E} \left\{ est \left[ \frac{\partial J(\rho)}{\partial \rho} \right] \right\} = \frac{\partial J(\rho, t)}{\partial \rho}. \quad (26)$$

Tais equações apresentadas na presente seção garantem a estimativa do gradiente para o caso onde a parcela do custo referente ao esforço de controle é nula (FLORES DA CUNHA, 2010).

### 2.2.3 O algoritmo

Com um controlador  $C(\rho, z) = \rho^T \bar{C}(z)$  operando na planta, geram-se os sinais  $y^1(\rho, t)$  e  $y^2(\rho, t)$ , das respectivas equações (20) e (21), e calcula-se o erro conforme (22). Tal erro calculado é então utilizado em (23) para se obter a estimativa do gradiente do sinal de saída.

Calcula-se então a estimativa do gradiente da função de custo  $J(\rho)$  conforme (24), e o próximo vetor de parâmetros do controlador é calculado por

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \text{est} \left[ \frac{\partial J(\rho)}{\partial \rho} \right] \quad (27)$$

onde  $\gamma_i$  é uma seqüência de números reais positivos que determina o tamanho de passo do algoritmo utilizado e  $R_i$  é uma seqüência de matrizes positivas definidas que irão definir a direção de busca (HJALMARSSON et al., 1998).

### 2.2.4 Convergência

Para que o algoritmo utilizado tenha sua convergência local garantida, é necessário que as seguintes condições sejam satisfeitas: em qualquer experimento, a seqüência de sinais de distúrbio  $v_t$ , para  $t = 1, \dots, N$ , consista de sinais limitados de média zero; a estimativa do gradiente utilizada deve ser não-polarizada; os elementos da seqüência de passos  $\gamma_i$  devem satisfazer (HJALMARSSON et al., 1998):

$$\gamma_i \geq 0 \quad e \quad \sum_{i=1}^{\infty} \gamma_i = \infty \quad (28)$$

$$\sum_{i=1}^{\infty} \gamma_i^2 \leq \infty. \quad (29)$$

Para este trabalho a escolha da seqüência de passos utilizada se faz da seguinte maneira. Utilizando-se um passo inicial  $\gamma_0$  é realizada uma iteração do método de maneira a se obter dois valores de custo. O primeiro valor de custo é referente ao sistema operando com os parâmetros iniciais do controlador e o segundo custo diz respeito ao sistema operando com os parâmetros obtidos após uma iteração. Devido ao elevado tempo que o experimento prático estruturado para este trabalho exige para a realização de cada uma das iterações, foi estipulada a seguinte regra para a redução do número de iterações necessárias e maior velocidade na minimização do custo: o custo obtido após uma iteração deve ser duas vezes menor do que o custo inicial ou menor. Caso contrário, o passo sofre um aumento de duas vezes e o experimento é reiniciado a partir da primeira iteração.

Após esse critério ser atendido as iterações são realizadas normalmente e sempre que houver um aumento de custo de uma iteração para a próxima, o valor de passo é reduzido pela metade e descarta-se o vetor de parâmetros obtido nessa iteração, mantendo-se o vetor de parâmetros calculado na iteração anterior em que houve uma redução do custo.

### 3 VIRTUAL REFERENCE FEEDBACK TUNING

#### 3.1 REVISÃO BIBLIOGRÁFICA

O método *Virtual Reference Feedback Tuning* (VRFT) visa sintonizar um controlador para uma planta de maneira direta (a identificação do modelo da planta utilizada não é necessária) baseado em medições de entrada e saída adquiridas através de um único ensaio, não sendo necessárias iterações ou outros experimentos específicos.

Um conjunto de dados adquiridos através de um ensaio consiste realizar medições do sinal  $u(t)$  (valores do sinal de saída do controlador) e as respectivas medições do sinal  $y(t)$  (valores do sinal de saída do sistema), sendo que o ensaio em si e a forma como esses dados foram adquiridos é irrelevante para a discussão nesse momento. O método consiste na utilização de uma referência virtual, assim chamada por não ser a referência utilizada para obtenção do conjunto de dados adquiridos no ensaio, que leve a saída do sistema a seguir certo modelo de referência desejado. De posse do sinal de saída e da referência virtual utilizada calcula-se então o sinal de erro que deve ser aplicado ao controlador ideal que leve o sinal de saída desse controlador a atingir os valores de saída adquiridos.

O problema resume-se então à escolha dos parâmetros do controlador que minimizem os critérios do método, que são baseados apenas no conjunto de dados adquirido em ensaio e serão explicados em detalhe a seguir.

### 3.2 CONSIDERAÇÕES INICIAIS

Considera-se para o problema um sistema dinâmico com uma planta linear em tempo discreto, *Single-Input and Single-Output* (SISO) tal que

$$y(t) = G(z)u(t) + v(t), \quad (30)$$

onde  $z$  é o operador de avanço no tempo,  $G(z)$  é a função de transferência desconhecida do processo,  $u(t)$  é o sinal de saída do controlador e  $v(t)$  é um distúrbio do processo não-mensurável. O distúrbio  $v(t)$  é considerado como sendo um processo estocástico estacionário no sentido amplo com média nula, o qual pode ser definido como  $v(t) = H(z)w(t)$ , sendo  $w(t)$  um ruído branco com variância  $\sigma_w^2$ . A estrutura do sistema analisado encontra-se na figura 4.

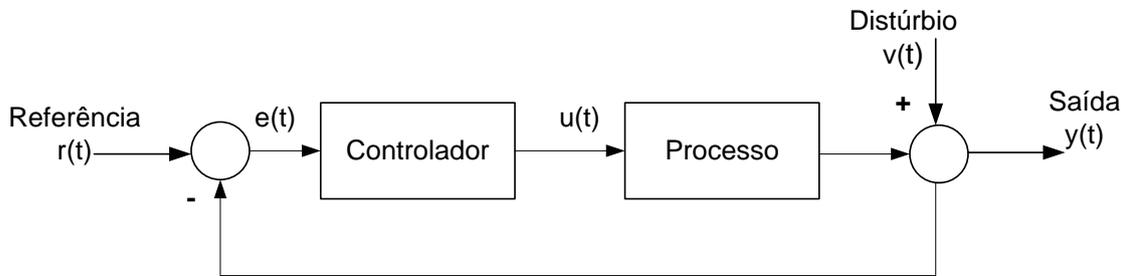


Figura 4: Diagrama de blocos do sistema considerado para o método VRFT.

O processo é controlado por um controlador linear e invariante no tempo  $C(z, \rho)$  e se supõe que tal controlador possa ser parametrizado com a seguinte estrutura:

$$C(z, \rho) = \rho^T \bar{C}(z), \quad (31)$$

onde  $\bar{C}(z)$  é um vetor coluna  $n$ -dimensional de funções racionais e  $\rho^T$  é um vetor linha  $n$ -dimensional de parâmetros, tal como:

$$\bar{C}(z) = [C_1(z) \ C_2(z) \ \dots \ C_n(z)]^T \quad (32)$$

$$\rho^T = [\rho_1 \ \rho_2 \ \dots \ \rho_n]. \quad (33)$$

Como exemplo, uma estrutura de controladores amplamente utilizada é do tipo proporcional-integral-derivativo (PID) com o pólo derivativo fixado. A parametrização desse tipo de controlador pode ser escrita como:

$$C(z, \rho) = [K_p \ K_i \ K_d] \left[ 1 \ \frac{1}{z-1} \ \frac{z-1}{z} \right]^T. \quad (34)$$

A saída do controlador, chamada de ação de controle, é dada por

$$u(t) = C(z, \rho)(r(t) - y(t)), \quad (35)$$

onde  $r(t)$  é o sinal de referência, o qual é assumido como sendo um processo estacionário no sentido amplo e descorrelacionado com o ruído, ou seja,

$$\bar{E}[r(t)e(s)] = 0 \ \forall t, s, \quad (36)$$

onde

$$\bar{E}[r(t)e(s)] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \mathbf{E}[r(t)e(s)] \quad (37)$$

e  $\mathbf{E}[\cdot]$  denota a esperança matemática (LJUNG, 1999).

Aplicando-se (33) em (30) no sistema de malha fechada, se obtém

$$y(t, \rho) = T(z, \rho)r(t) + S(z, \rho)v(t) \quad (38)$$

onde  $T(z, \rho)$  e  $S(z, \rho)$  são, respectivamente, as funções de transferência de malha fechada e de sensibilidade do sistema, tal como:

$$S(z, \rho) \triangleq \frac{1}{1 + C(z, \rho)G(z)} \quad (39)$$

$$T(z, \rho) \triangleq \frac{C(z, \rho)G(z)}{1 + C(z, \rho)G(z)} = C(z)G(z)S(z, \rho) \quad (40)$$

Assim sendo, explicita-se a dependência do sistema de malha fechada em relação aos parâmetros  $\rho$  do controlador.

### 3.2.1 O problema de controle do modelo de referência

Deseja-se que o sistema de malha fechada alcance determinado desempenho, o qual é especificado por uma função de transferência desejada de malha fechada  $T_d(z)$ , tal como

$$y_d(t) = T_d(z)r(t). \quad (41)$$

A função de transferência de malha fechada  $T_d(z)$  é chamada de modelo de referência na literatura e se procura pelos parâmetros do controlador  $C(z, \rho)$  que façam com que o sinal de saída do sistema seja o mais próximo possível da saída desejada  $y_d(t)$ . Pode-se dessa maneira determinar o erro entre a saída real do sistema  $y(t)$  e a saída desejada  $y_d(t)$  definido como

$$\tilde{y}(\rho, t) \triangleq y(\rho, t) - y_d(\rho). \quad (42)$$

Surge então um critério de desempenho, baseado no erro  $\tilde{y}(\rho, t)$ , o qual se deseja minimizar com intuito de obter os parâmetros do controlador, conforme a seguir:

$$\min_{\rho} [J(\rho)], \quad (43)$$

onde:

$$J(\rho) \triangleq \bar{E} \left[ (y(\rho, t) - y_d(\rho))^2 \right] = \bar{E} [(\tilde{y}(\rho, t))^2]. \quad (44)$$

Conforme (36), o ruído e o sinal de referência são descorrelacionados, o que permite reescrever (44) em dois termos. Um é referente ao critério de seguimento de referência  $J_y(\rho)$  e o segundo é referente ao critério de rejeição de ruído  $J_e(\rho)$ , conforme a seguir:

$$J(\rho) = J_y(\rho) + J_e(\rho) \quad (45)$$

$$J_y(\rho) = \bar{E} \left[ \left( (T_d(z) - T(z, \rho))r(t) \right)^2 \right] \quad (46)$$

$$J_e(\rho) = \bar{E} \left[ (S(z)v(t))^2 \right]. \quad (47)$$

Uma vez que o modelo de referência seja escolhido, o correspondente desempenho do sistema de malha fechada pode ser obtido unicamente através do controlador ideal  $C_d(z)$ :

$$C_d(z) = \frac{T_d(z)}{G(z)(1 - T_d(z))} \quad (48)$$

Esse é o controlador que minimiza o critério de seguimento de referência  $J_y(\rho)$ . O sistema somente irá se comportar exatamente como especificado no modelo de referência caso o controlador ideal  $C_d(z)$  pertença à classe de controladores em consideração. Essa suposição é formalizada da seguinte maneira:

$$\exists \rho_d \in \mathcal{D}_\rho: C_d(z) = \rho_d^T \bar{C}(z) \quad (49)$$

Essa suposição pode ser sempre plenamente satisfeita se as estruturas do modelo de referência e do controlador são adequadamente pré-determinadas. O conhecimento da estrutura da função de transferência da planta é suficiente para que as estruturas do modelo de referência e do controlador sejam escolhidas adequadamente.

### 3.3 DESENVOLVIMENTO DO MÉTODO

O VRFT é um método de projeto de controladores não iterativo e que lida apenas com o problema de seguimento de referência, ou seja, procura minimizar apenas o critério  $J_y(\rho)$ . A idéia central do método é fazer com que a tarefa de procurar o controlador seja reduzida para um problema de identificação que descreva a relação dinâmica entre os sinais  $\bar{e}(t)$  e  $u(t)$ . Supõe-se que um controlador ideal  $C_d(z)$  esteja presente em um sistema de malha fechada em operação normal e dois conjuntos de dados (o sinal de entrada do processo  $u(t)$  e o respectivo sinal de saída  $y(t)$ ) são adquiridos do sistema. Para que isso seja feito, um sinal de referência é calculado tal que, quando aplicado ao sistema operando em malha fechada com o controlador ideal, a saída seja idêntica ao sinal de saída obtido durante o experimento  $\bar{r}(t) = T_d^{-1}y(t)$ . Essa referência calculada é chamada de “virtual” por não ter sido a referência utilizada no experimento que gerou os conjuntos de dados ( $u(t)$  e  $y(t)$ ), e desse conceito que surge a denominação do método. De posse da referência virtual e do conjunto de dados de saída  $y(t)$  adquiridos durante a operação normal do sistema é possível calcular um

erro virtual  $\bar{e}(t) = \bar{r}(t) - y(t)$ , mesmo que a função de transferência do processo  $G(z)$  seja desconhecida. Sabe-se que quando o processo  $G(z)$  é alimentado pelo sinal  $u(t)$  (o conjunto de dados de entrada do processo de que temos posse), ele irá fornecer na sua saída o sinal  $y(t)$ . Um bom controlador é aquele que fornece de resposta  $u(t)$  quando alimentado pelo sinal de erro  $e(t)$ . A idéia é então procurar pelo controlador que forneça  $u(t)$  quando alimentado pelo erro virtual  $\bar{e}(t)$ . Como ambos os sinais  $u(t)$  e  $\bar{e}(t)$  são conhecidos, a tarefa de procurar o controlador ideal  $C_d(z)$  consiste em resolver o seguinte problema

$$\min[ J_{VR}^N(\rho) ], \quad (50)$$

onde:

$$J_{VR}^N(\rho) = \frac{1}{N} \sum_{t=1}^N (u(t) - C(z, \rho)\bar{e}(t))^2. \quad (51)$$

Como o controlador  $C(z, \rho)$  pode ser parametrizado conforme (32), a equação (51) pode ser reescrita da forma

$$J_{VR}^N(\rho) = \frac{1}{N} \sum_{t=1}^N (u(t) - \rho^T \varphi(t))^2, \quad (52)$$

onde:

$$\varphi(t) = \bar{C}(z)\bar{e}(t). \quad (53)$$

Visto que  $J_{VR}^N(\rho)$  é uma equação quadrática, nota-se que a solução para a minimização desse critério é da forma de mínimos quadrados e é feita de maneira direta, sem necessidade de iterações tal como no método IFT. Uma estimativa do vetor de parâmetros do controlador pode ser encontrada resolvendo-se a seguinte equação:

$$\hat{\rho} = \left[ \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) u(t). \quad (54)$$

A utilização dessa equação em conjunto com os dados obtidos, que possuem ruído, irá resultar em um vetor de parâmetros do controlador polarizado que, por sua vez, irá reduzir significativamente o desempenho desejado do controlador. Isso pode ser observado através da

inspeção da expressão no domínio frequência do critério assintótico de  $J_{VR}(\rho)$ , quando dados com ruído e dados sem ruído são comparados. Para dados sem ruído obtém-se:

$$J_{VR}^L(\rho) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |P|^2 |C(\rho) - C_0|^2 |1 - T(z)|^2 \frac{|L|^2}{|T|^2} \phi_u d\omega. \quad (55)$$

Para dados com presença de ruído:

$$J_{VR}^R(\rho) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |P|^2 |C(\rho) - C_0|^2 |1 - T(z)|^2 \frac{|L|^2}{|T|^2} \phi_u + \frac{|C(\rho)|^2}{|P|^2 |C_0|^2} |L|^2 \phi_d d\omega, \quad (56)$$

onde  $\phi_d$  é a densidade espectral do ruído.

Para contemplar o ruído presente, a equação (54) torna-se então

$$\hat{\rho} = \left[ \sum_{t=1}^N \zeta(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \zeta(t) u_\varphi(t), \quad (57)$$

onde:

$$\varphi(t) = \bar{C}(z)(T_d^{-1}(z) - 1)y(t). \quad (58)$$

Na equação (58) o termo  $u_\varphi(t)$  representa o sinal de controle que foi adquirido no mesmo experimento que gerou  $\varphi(t)$ . O termo  $\zeta(t)$  é uma variável instrumental definida como

$$\zeta(t) = \bar{C}(z)(T_d^{-1}(z) - 1)y_s(t) \quad (59)$$

onde  $y_s(t)$  é o sinal de saída do sistema adquirido na realização de um segundo experimento, que é igual ao primeiro experimento exceto pela realização do ruído presente no sistema. Essa variável instrumental se destina a eliminar a polarização dos parâmetros estimados do controlador originada devido à presença de ruído.

### 3.4 ALGORITMO

A idéia acima pode ser desenvolvida com um algoritmo que consiste em três passos (onde um filtro  $L(z)$  escolhido pelo usuário pode ser utilizado para utilizar uma seleção definida dos dados). Esse algoritmo representa a estrutura do método VRFT.

Dado um conjunto de dados de entrada e saída  $\{u(t), y(t)\}_{t=1,2,\dots,N}$ , faça o seguinte:

Calcule uma referência virtual  $\bar{r}(t)$  tal que:

$$y(t) = T_d(z)\bar{r}(t) \quad (60)$$

e com isso calcule o correspondente erro virtual

$$\bar{e}(t) = \bar{r}(t) - y(t). \quad (61)$$

Seleciona-se o vetor de parâmetros do controlador,  $\hat{\rho}_N$ , que minimize o seguinte critério:

$$J_{VR}^N(\rho) = \frac{1}{N} \sum_{t=1}^N (u(t) - C(z; \rho)\bar{e}(t))^2. \quad (62)$$

O controlador pode ser parametrizado com um vetor de parâmetros tal como:

$$C(z; \rho) = \rho^T(z)\bar{C} \quad (63)$$

e a equação (32) pode ser reescrita da forma:

$$J_{VR}^N(\rho) = \frac{1}{N} \sum_{t=1}^N (u(t) - \rho^T \varphi(t))^2, \quad (64)$$

onde  $\varphi(t)$  é definido pela equação (53).

Dessa forma o vetor de parâmetros  $\hat{\rho}_N$  é dado por:

$$\hat{\rho}_N = \left[ \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) u(t). \quad (65)$$

## 4 ARQUITETURA DO SISTEMA

### 4.1 HARDWARE UTILIZADO

Para a aplicação prática dos métodos a plataforma didática QET-DCMCT foi escolhida por apresentar todos os elementos necessários de um sistema de controle e um detalhamento desses elementos e sua localização podem ser visualizados na figura 5.

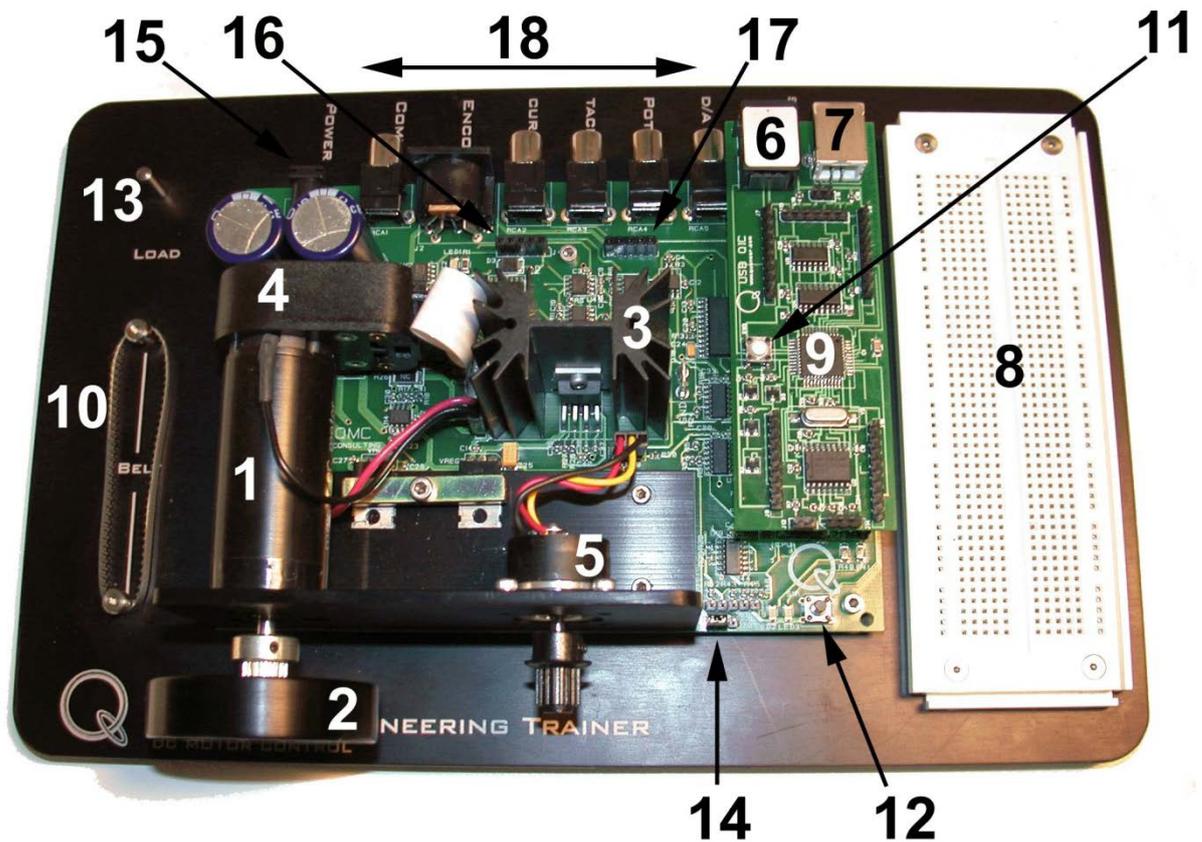


Figura 5: Indicação dos elementos da plataforma QET- DCMCT.

A descrição de cada um dos elementos da plataforma QET-DCMCT é apresentada na tabela 1.

**Tabela 1 Nomenclatura dos elementos componentes da plataforma QET-DCMCT**

Identificação	Descrição
1	Motor de corrente contínua Maxon
2	Massa inercial removível
3	Amplificador de potência linear
4	Encoder ótico de alta resolução
5	Potenciômetro de rolamento de esferas
6	Porta RJ11 do QIC: Para instalação de hardware com a utilização de um dispositivo de programação compatível
7	Porta USB do QIC: Utilizada para sintonia do controlador e comunicação (utilizando-se o software QICii)
8	Protoboard externa: para desenvolvimento de circuitos analógicos
9	Microcontrolador PIC18F4550
10	Correia removível: para acoplar o eixo do motor ao potenciômetro
11	Botão de reset do micro controlador
12	Botão do usuário: para interações manuais necessárias pelo usuário
13	Pino para fixação da massa inercial quando não utilizada
14	Jumper J6: alterna entre a operação do QIC ou de um DAQ
15	Plugue de 6mm para alimentação
16	Bornes J4 para acessar níveis de tensão de utilização intermediários
17	Bornes J11 para acessar sinais analógicos intermediários
18	Terminais RCA para realizar interface com computador através de uma placa de aquisição de dados.

## 4.2 SOFTWARES UTILIZADOS

Para o desenvolvimento dos métodos foi utilizado o software MATLAB o qual permite um tratamento elaborado dos dados adquiridos. Os algoritmos desenvolvidos para cada um dos métodos são apresentados na seção Anexo ao final deste trabalho. A estruturação do sistema para cada um dos métodos foi desenvolvida com o conjunto de ferramentas Simulink, do MATLAB, e é apresentada em detalhes nas seções seguintes.

Para que o MATLAB pudesse se comunicar com a plataforma didática QET-DCMCT foi utilizado o software *Quanser Real-Time Controller* (QuaRC), o qual permite que o MATLAB envie e receba dados da plataforma em tempo real com o uso de uma placa de aquisição de dados. A topologia da estrutura física do sistema desenvolvido para este trabalho, demonstrando a integração dos elementos e suas atribuições, é apresentada na figura 6.

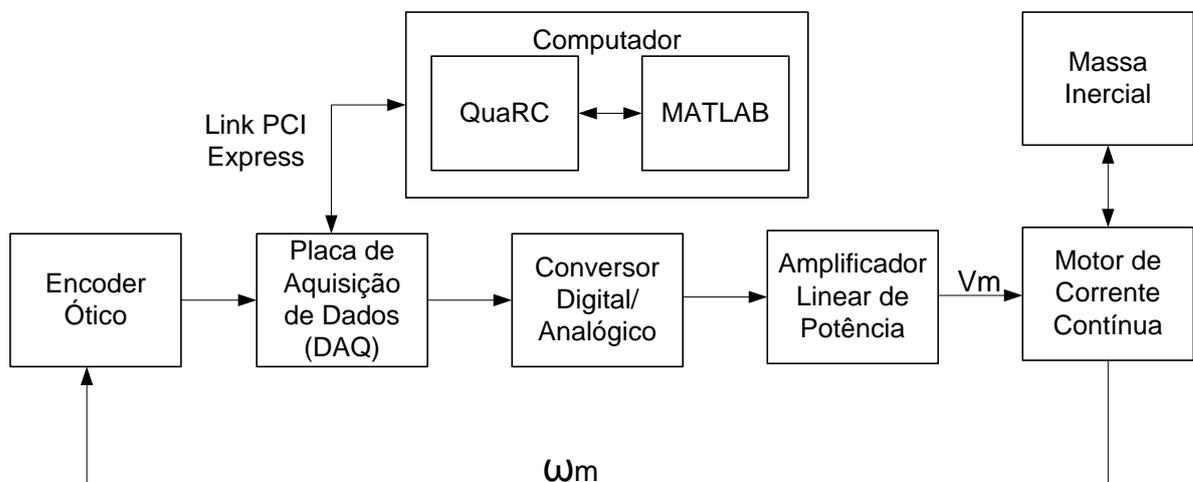


Figura 6: Topologia dos componentes do sistema e suas interações.

Tabela 2 Descrição das variáveis apresentadas na figura 6

Símbolo	Descrição	Unidade
$\omega_m$	Velocidade angular do motor	rad\s
$V_m$	Tensão aplicada no motor pelo amplificador linear	V

### 4.3 ESTRUTURAÇÃO DO MÉTODO IFT

Para o desenvolvimento dos experimentos foi desenvolvido o sistema em Simulink apresentado na figura 7 e a estrutura do algoritmo para a escolha do controlador pode ser visualizada na figura 8 em formato de fluxograma.

Os algoritmos que realizam o cálculo do vetor de parâmetros do controlador são disponibilizados na seção Anexos ao final deste trabalho.

#### QET: IFT/PI/Controle de Velocidade

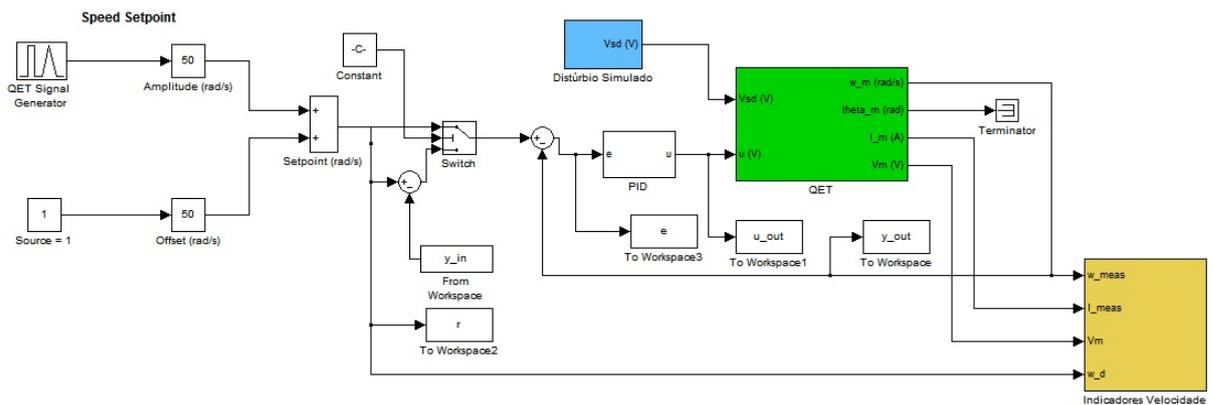


Figura 7: Sistema de ensaios para o desenvolvimento do método IFT.

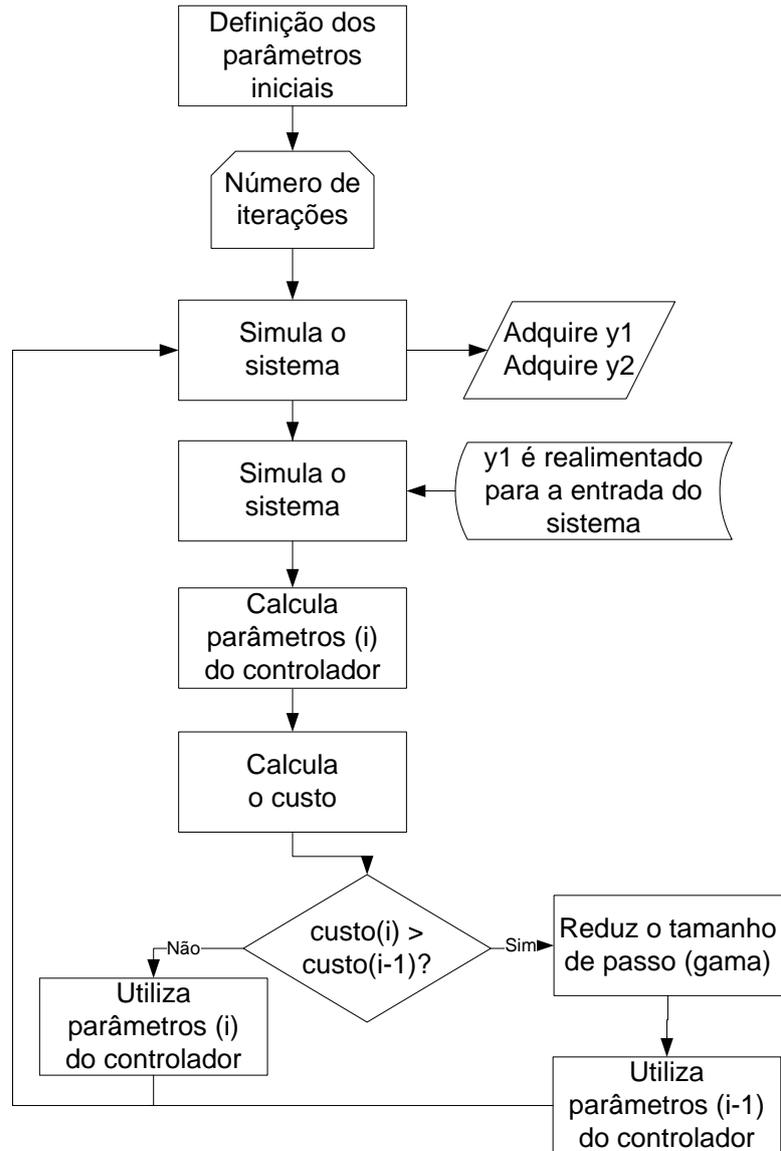


Figura 8: Fluxograma da operação do sistema.

#### 4.4 ESTRUTURAÇÃO DO MÉTODO VRFT

Para o método VRFT foi desenvolvido o sistema em Simulink apresentado na figura 9. O algoritmo que realiza o cálculo do vetor de parâmetros do controlador é disponibilizado na seção Anexo ao final deste trabalho.

##### QET: VRFT/PI/Controle de Velocidade

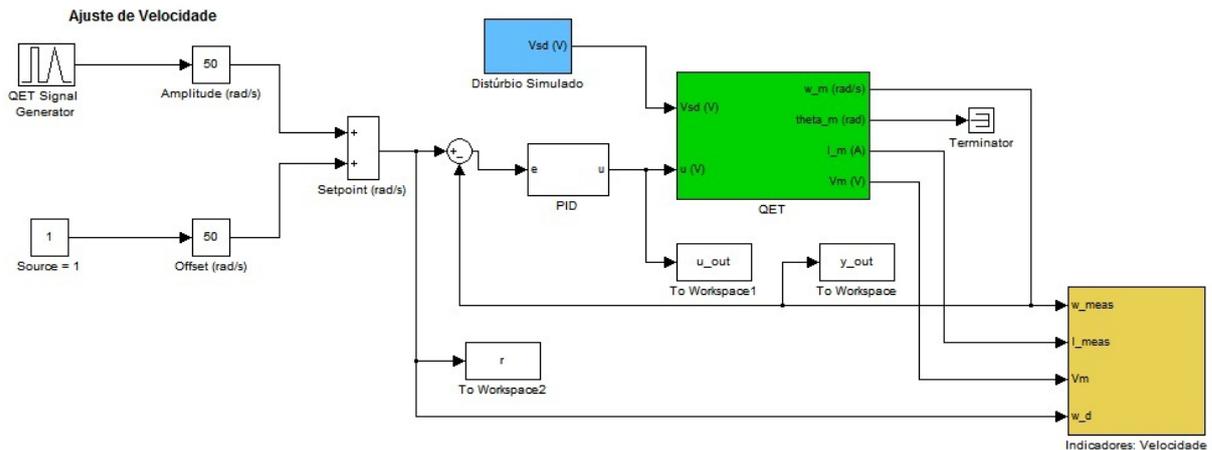


Figura 9: Sistema de ensaios para o desenvolvimento do método VRFT.

#### 4.5 MODELO DE REFERÊNCIA

Para ambos os métodos foi utilizado o seguinte modelo de referência

$$T_d(z) = \frac{0,1}{z - 0,9}. \quad (66)$$

Usando-se esse modelo de referência, as dinâmicas rápidas do motor (pólo elétrico) não serão excitadas e, dessa forma, pode-se considerar um sistema de primeira ordem. Assim sendo, o modelo de referência (67) foi escolhido para atender ao objetivo de comparar os métodos VRFT e IFT com o uso dos mesmos requisitos.

## 5 RESULTADOS

Neste capítulo serão apresentados os resultados alcançados em cada um dos métodos para a aplicação desenvolvida. Utilizando-se da topologia descrita no capítulo 4 os métodos IFT e VRFT foram ensaiados em uma estrutura que utiliza um controlador PI gerando um ensaio para cada um dos métodos analisados.

É possível determinar o melhor controlador possível para o processo ensaiado através de sua modelagem, entretanto tal objetivo não é escopo deste trabalho. Deseja-se analisar o desempenho de cada um dos dois métodos no projeto de um controlador para alguma condição específica, não sendo necessariamente o melhor controlador possível. Dessa maneira estipulou-se para ambos os métodos o uso de uma estrutura de controlador do tipo proporcional-integral, o mesmo vetor de parâmetros do controlador inicial

$$\rho^0 = \begin{bmatrix} 0,0010 \\ 0,0010 \end{bmatrix}, \quad (67)$$

o mesmo modelo de referência apresentado em (67) e o mesmo sinal de referência conforme apresentado na figura 10.

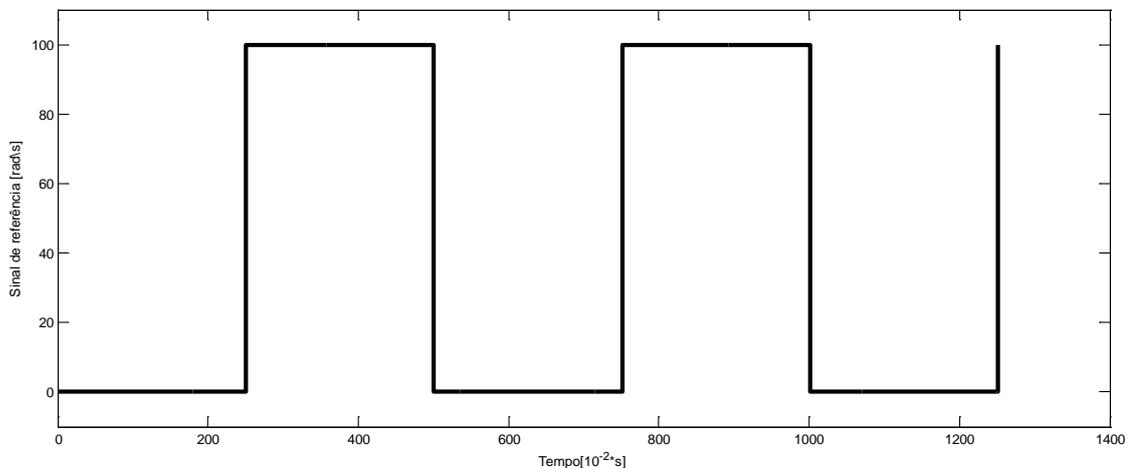


Figura 10: Sinal de referência utilizado nos ensaios.

## 5.1 MÉTODO IFT

Com base na figura 11 é possível perceber que o método IFT obteve uma redução de custo para um valor virtualmente nulo já a partir da terceira iteração. Nota-se pela quinta coluna da tabela 3 que o valor mínimo do custo é obtido de fato a partir da terceira oscilação e sofre pequenas oscilações devido ao ruído presente no sistema durante os ensaios. Salienta-se que o custo é uma função decrescente no método IFT, ou seja, sempre que ocorre um aumento do custo em uma iteração, os parâmetros do controlador da iteração anterior são mantidos, o passo de busca é reduzido pela metade e uma nova iteração é realizada.

**Tabela 3 Parâmetros do sistema e custo ao longo das iterações realizadas para o método IFT**

Iteração	$K_p$	$K_i$	$\gamma$	Custo
Inicial	0,0010	0,0010	1,0000	322,8410
1	0,0433	0,0017	1,0000	60,1601
2	0,0826	0,0030	1,0000	11,2054
3	0,0693	0,0047	1,0000	1,0397
4	0,0583	0,0052	1,0000	0,2073
5	0,0586	0,0052	0,5000	0,2192
6	0,0583	0,0052	0,5000	0,2023
7	0,0590	0,0052	0,5000	0,2000
8	0,0583	0,0052	0,2500	0,2020
9	0,0589	0,0052	0,2500	0,2007

Podem-se observar pela figura 11 as reduções do tamanho de passo  $\gamma$  da busca do gradiente de minimização da função custo na quinta e na oitava iterações, que são as iterações onde o custo encontrado foi superior ao custo obtido na quarta e na quinta iterações, respectivamente. Essa redução do valor do passo não resulta na obtenção de um vetor de parâmetros do controlador que leve o sistema a apresentar um custo sensivelmente menor, entretanto garante que o custo do sistema é uma função decrescente.

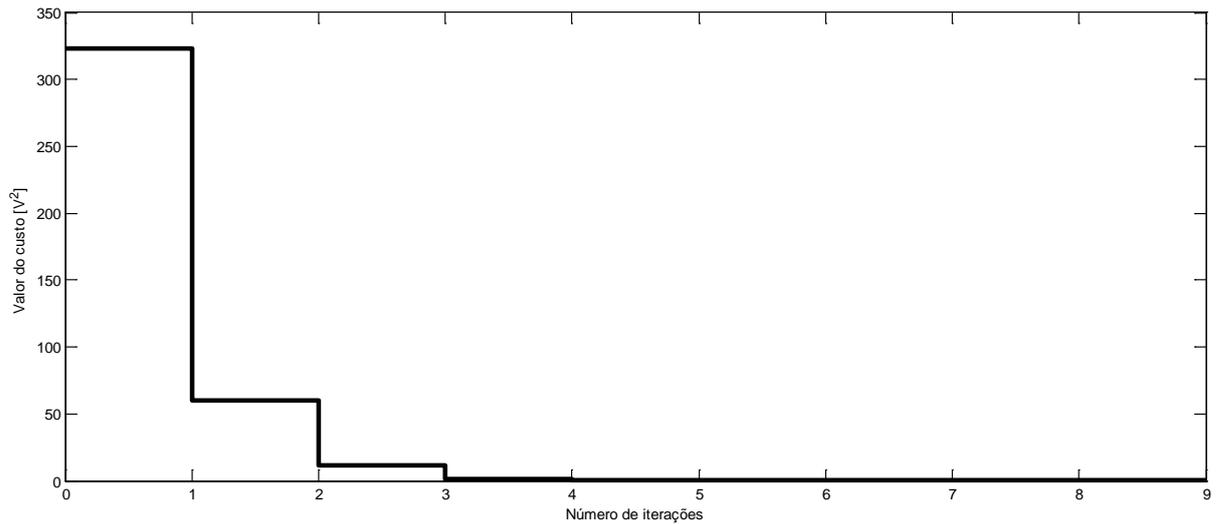


Figura 11: Progressão do custo versus número de iterações.

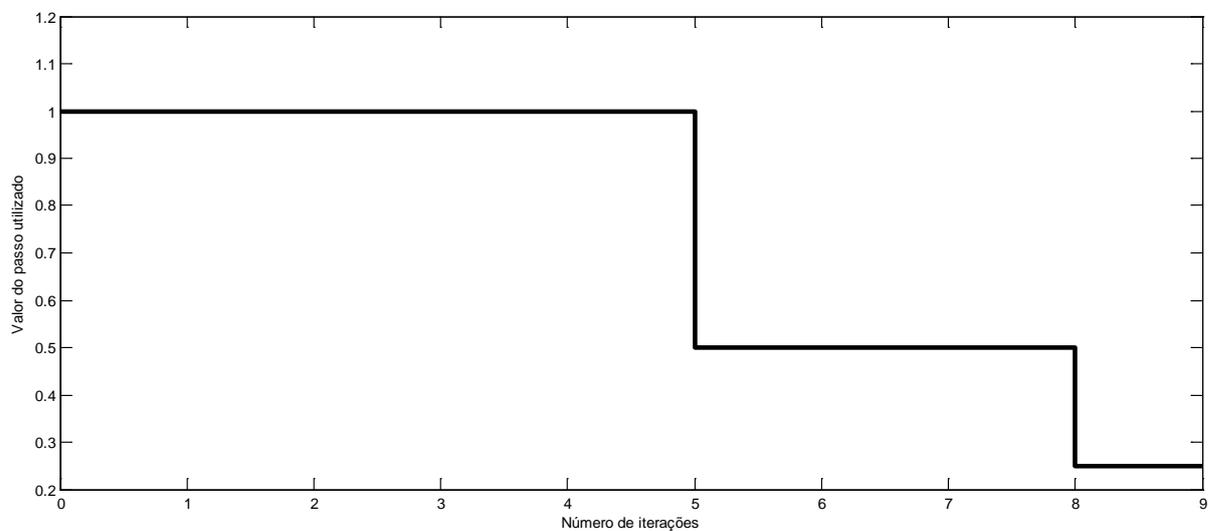


Figura 12: Progressão do passo  $\gamma$  versus número de iterações.

Conforme a figura 13 e a figura 14, o vetor de parâmetros do controlador final é obtido a partir da quarta iteração, indo de encontro com o valor do custo próximo do mínimo visto na quinta coluna da tabela 3.

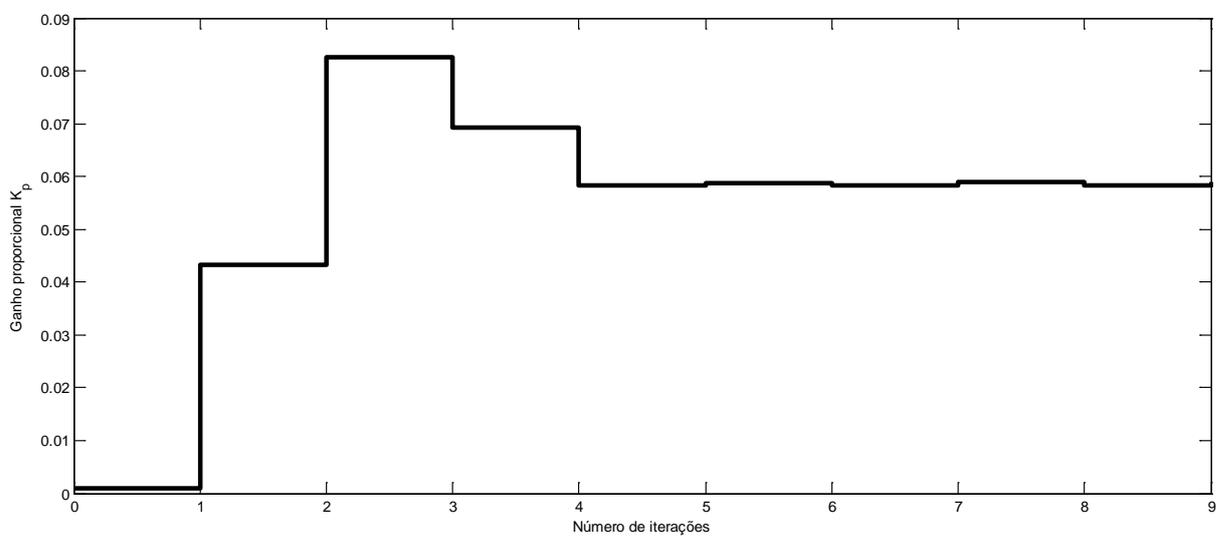


Figura 13: Progressão do ganho proporcional  $K_p$  versus número de iterações.

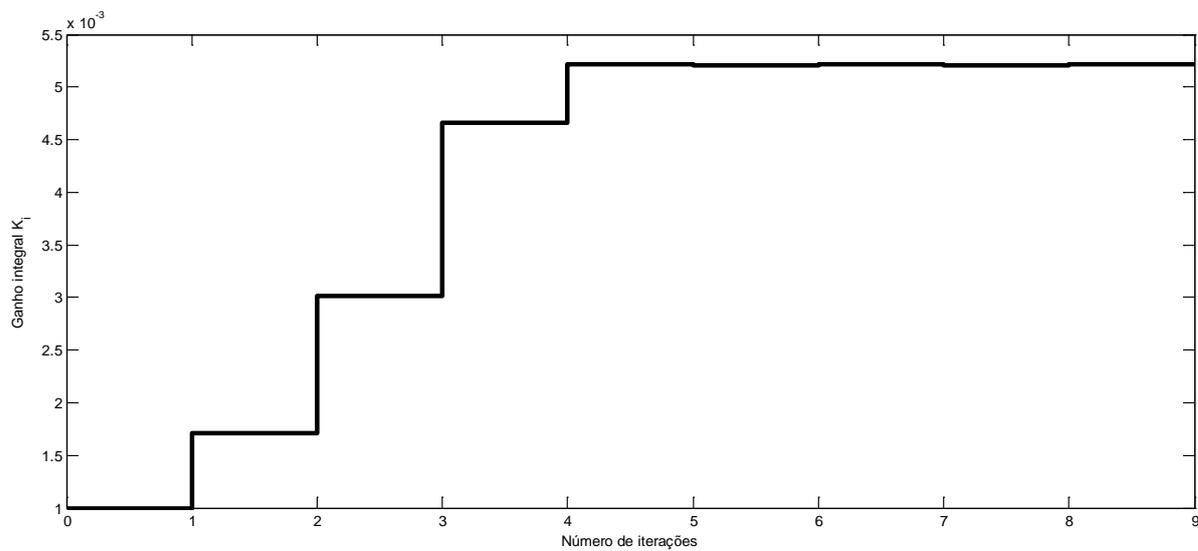


Figura 14: Progressão do ganho proporcional  $K_i$  versus número de iterações.

O esforço de controle, apresentado na figura 15, mostra que nas condições iniciais o controlador apresentava uma resposta equivalente ao próprio sinal de saída. Isso demonstra que o esforço de controle era tão pequeno que praticamente não influenciava a saída do sistema. Após nove iterações, a ação de controle apresenta uma resposta mais veloz do que a inicialmente observada, levando o sistema a atingir a resposta desejada.

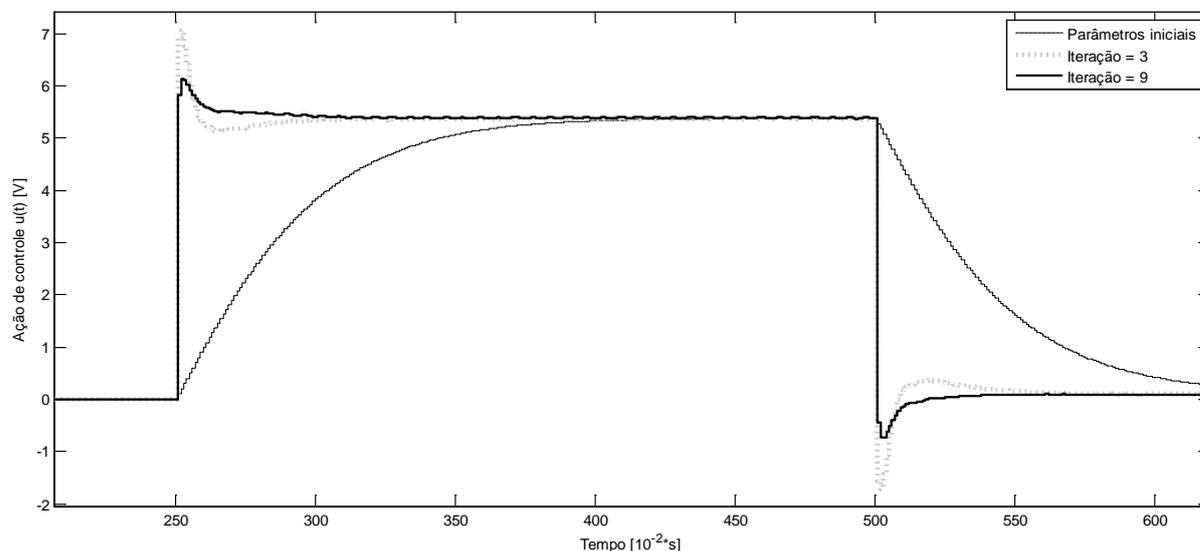


Figura 15: Ação de controle utilizando-se os parâmetros iniciais do problema (linha traço-ponto), após três iterações (linha tracejada) e após nove iterações (linha contínua) com a utilização dos parâmetros obtidos com o uso do algoritmo IFT.

A figura 16 apresenta uma progressão do sinal de saída do sistema em relação à iteração do algoritmo realizada. Observa-se como o sinal de saída vai sendo alterado a cada iteração, aproximando-se cada vez mais ao modelo de referência desejado.

Na figura 17, apresentam-se os sinais de saída do sistema com o uso dos parâmetros iniciais e com o uso dos parâmetros obtidos após nove iterações em contraste com o sinal de saída desejado. Como os parâmetros do controlador obtidos com o algoritmo resultam em um sinal de saída muito próximo ao modelo de referência desejado, as curvas praticamente se sobrepõem e, dessa maneira, a figura 18 apresenta um detalhamento da figura 17 para que se torne mais evidente o bom desempenho obtido com o algoritmo em um problema prático.

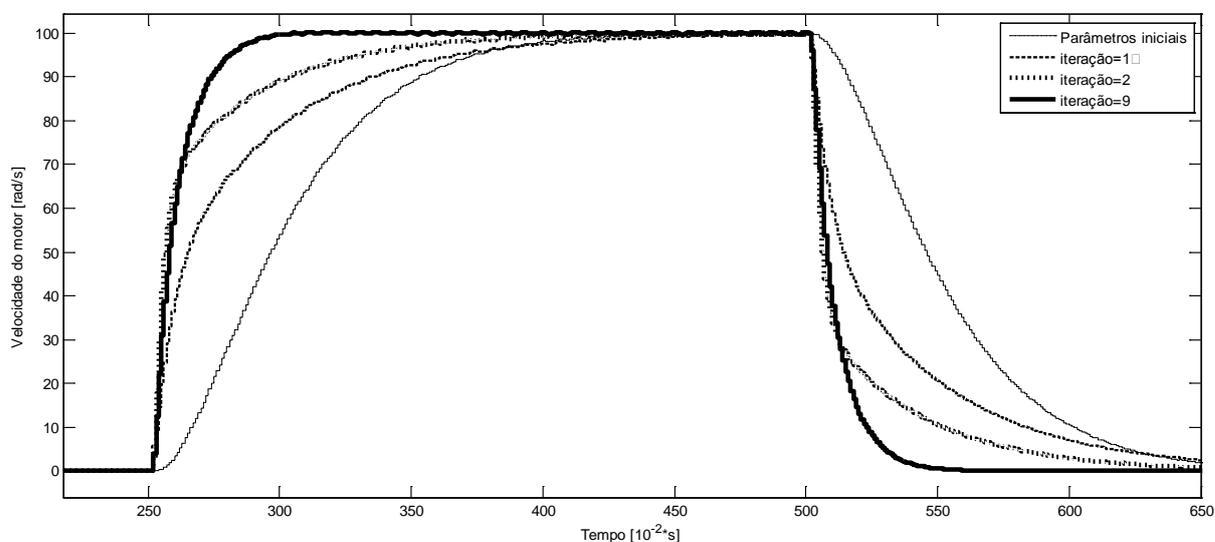


Figura 16: Progressão do sinal de saída para o uso dos parâmetros: iniciais (linha traço-ponto), após uma iteração (linha tracejada), após três iterações (linha pontilhada espessa) e após nove iterações (linha contínua).

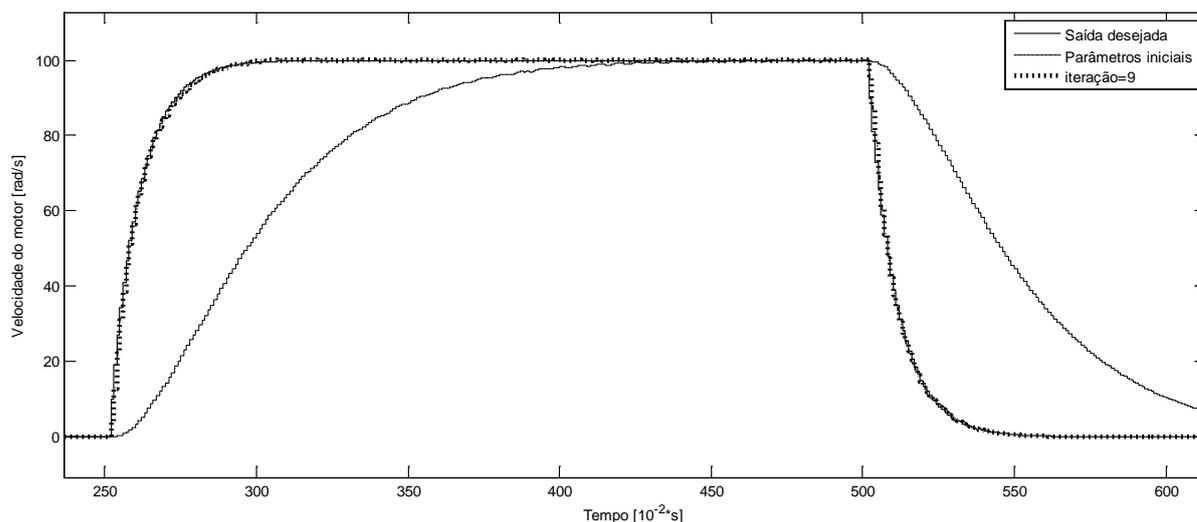


Figura 17: Comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua).

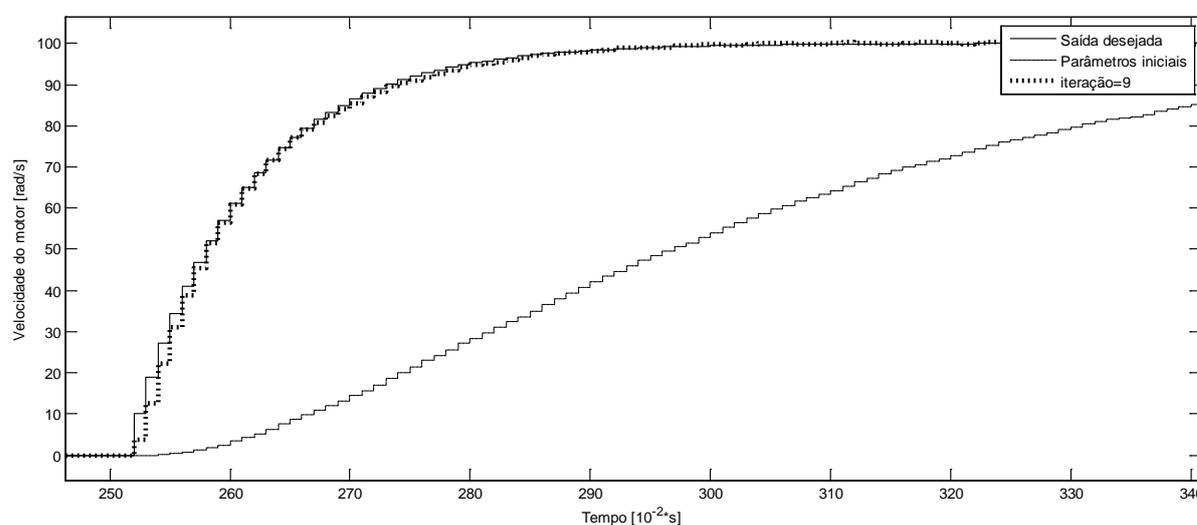


Figura 18: Detalhe da figura 17 apresentando a comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua).

## 5.2 MÉTODO VRFT

O método VRFT difere-se do método IFT por ser direto, não necessitando de iterações nos experimentos para a obtenção do vetor de parâmetros do controlador. Entretanto, foram realizadas repetidas sintonizações com o método VRFT para uma análise proporcional com o método IFT, permitindo analisar certas particularidades tais como velocidade e capacidade de minimização do custo. Não se deve confundir, dessa maneira, o número de iterações apresentados para o método IFT com a quantidade de ensaios realizados para o presente método.

A figura 19 mostra que o custo é levado a um valor virtualmente mínimo com a utilização de uma única sintonia do controlador. Conforme a quarta coluna intitulada '*Custo*', conforme tabela 4, um segundo experimento de sintonia do controlador reduz ainda mais o custo, porém de forma mais sutil.

**Tabela 4 Parâmetros do sistema e custo ao longo das iterações realizadas para o método VRFT**

Ensaio	$K_p$	$K_i$	Custo
Inicial	0,0010	0,0010	268,3124
1	0,0728	0,0054	1,0580
2	0,0564	0,0054	0,2686
3	0,0576	0,0054	0,2322
4	0,0577	0,0054	0,2475
5	0,0569	0,0054	0,2183
6	0,0577	0,0054	0,2487
7	0,0565	0,0054	0,2200
8	0,0561	0,0054	0,2578
9	0,0563	0,0054	0,2473

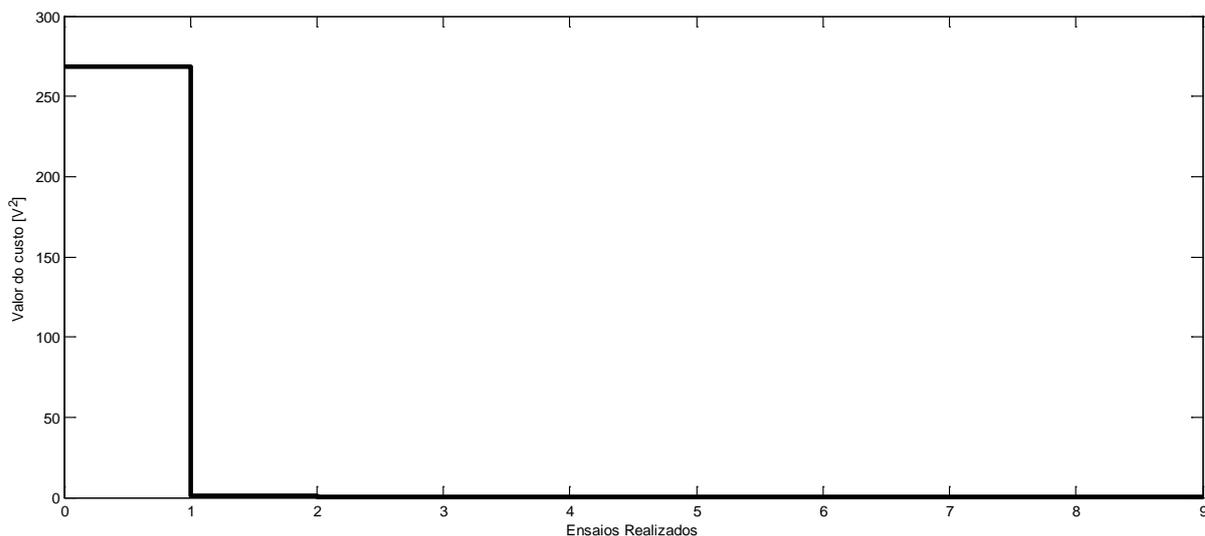


Figura 19: Progressão do custo ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT.

Essa estabilidade do custo se reflete na figura 20 e na figura 21, as quais demonstram a progressão dos ganhos proporcional e integral, respectivamente. O ganho integral é estável desde o primeiro ensaio e o ganho proporcional é sutilmente modificado após o segundo ensaio e sofre pequenas oscilações nos ensaios subseqüentes. Tais oscilações no ganho proporcional justificam-se pela presença de distúrbio na realização dos ensaios.

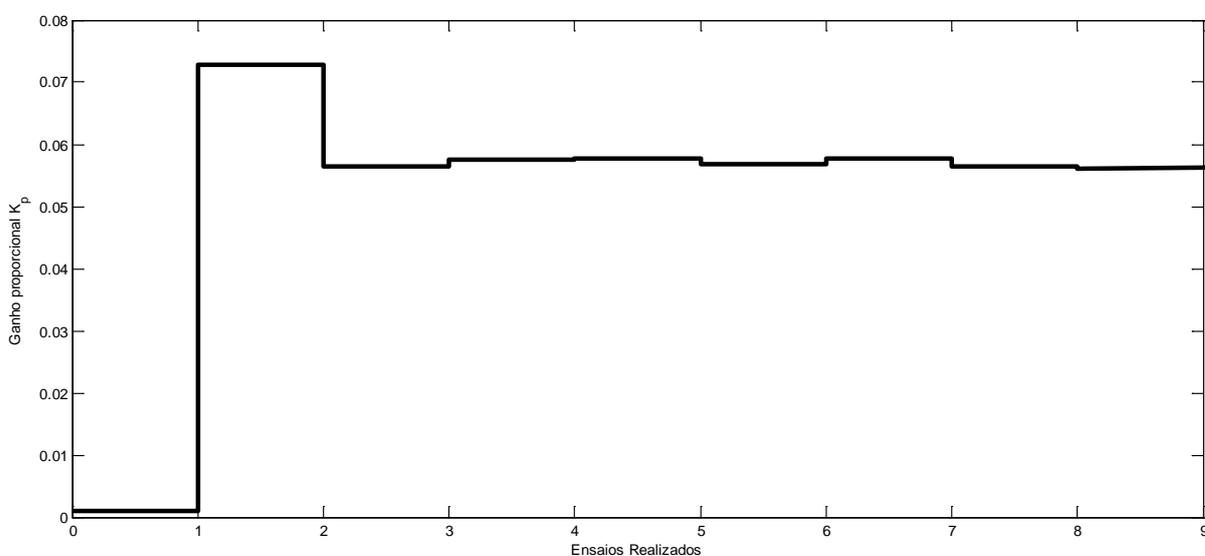


Figura 20: Progressão do ganho proporcional  $K_p$  ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT.

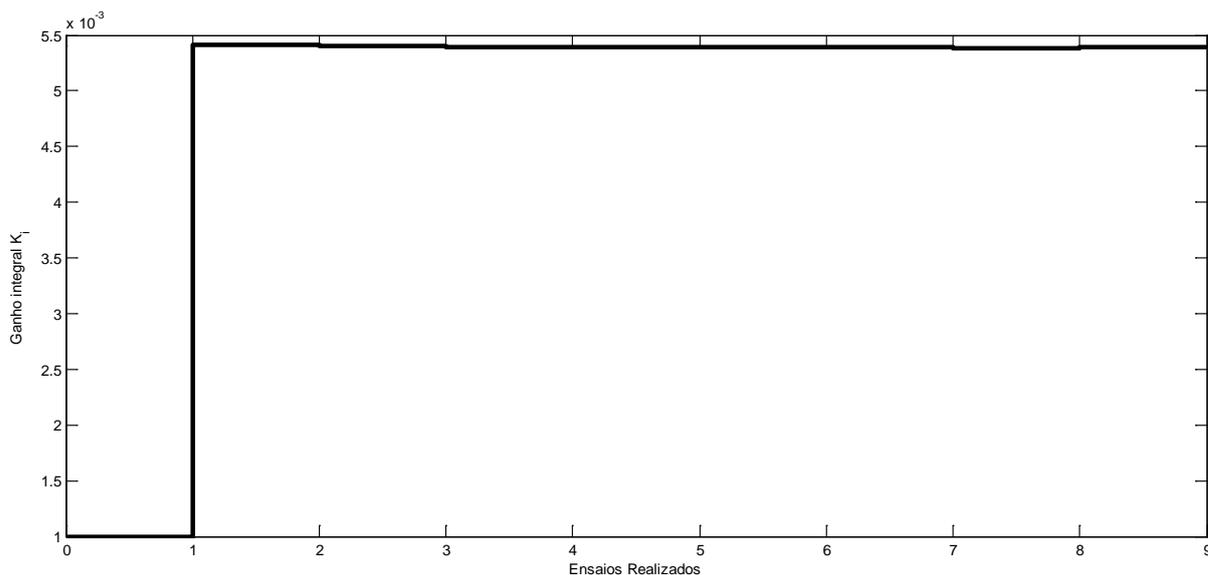


Figura 21: Progressão do ganho proporcional  $K_i$  ao longo de sucessivos ensaios de sintonização do controlador com o uso do método VRFT.

Através da figura 22 é possível perceber que a ação de controle sofre uma grande alteração após o primeiro ensaio de sintonização, o que é um ponto contrastante em relação ao método IFT que necessita de três iterações para alcançar um resultado semelhante, como pode ser observado na figura 15.

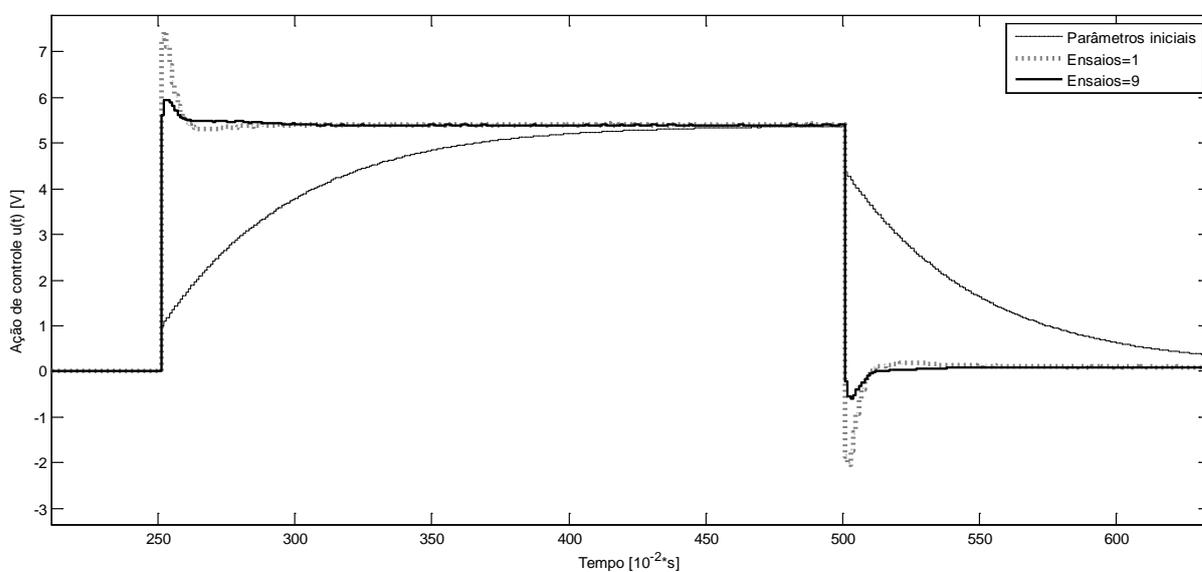


Figura 22: Progressão da ação de controle utilizando-se os parâmetros iniciais do problema (linha traço-ponto), ação de controle obtida no primeiro ensaio (linha tracejada) e a ação de controle obtida após nove ensaios para sintonizar o controlador (linha contínua).

A mudança que a ação de controle sofre após um segundo ensaio pode ser visualizada em detalhe na figura 23. A diferença da ação de controle após um segundo ensaio e após nove ensaios é de um nível muito baixo, contudo pode ser a grande diferença da resposta que o controlador com os parâmetros iniciais apresentava em relação ao modelo de referência. Após a realização de um ensaio a ação de controle já está muito mais próxima da desejada e a realização de um segundo ensaio permite que o custo, de valor 1,0580, possa ser reduzido para o valor de 0,2686, o qual sofre pequenas oscilações nos ensaios subsequentes devido ao ruído presente no experimento.

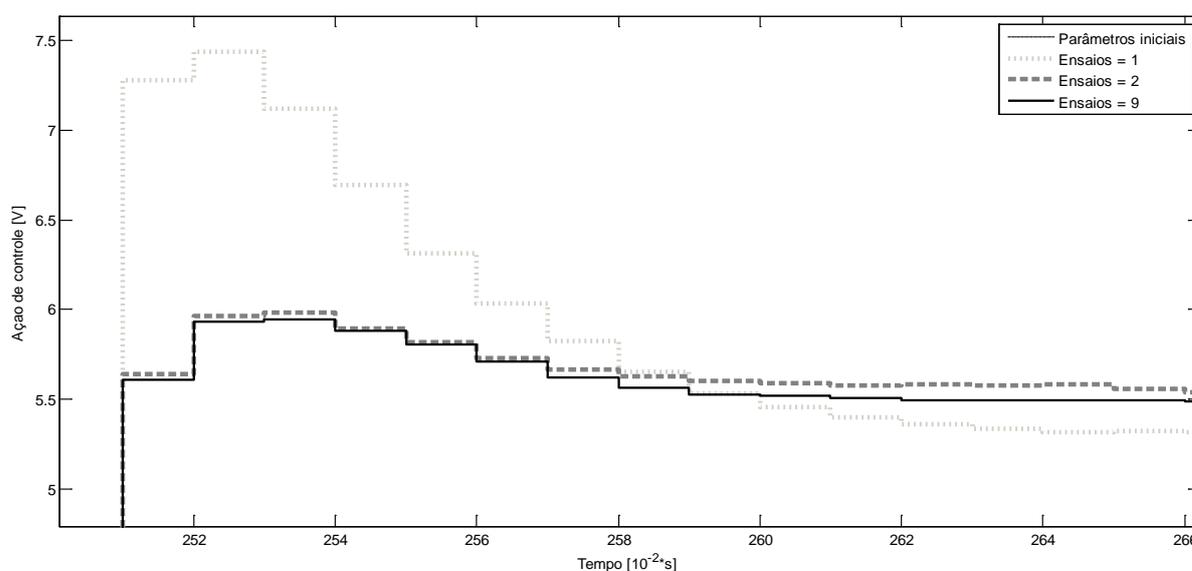


Figura 23: Detalhe apresentando a comparação da ação de controle obtida no primeiro ensaio (linha pontilhada), após uma segunda sintonização (linha tracejada) e após nove sintonizações sucessivas (linha contínua), com o uso do método VRFT.

Os sinais de saída do sistema após os ensaios de sintonia podem ser visualizado na figura 24, onde fica evidente que após uma única sintonia com o método VRFT o sistema apresenta uma resposta muito próxima ao modelo de referência desejado. A figura 25 apresenta em detalhe as diferenças que o sinal de saída possui após um único ensaio e após nove ensaios de sintonização em contraste com a saída desejada.

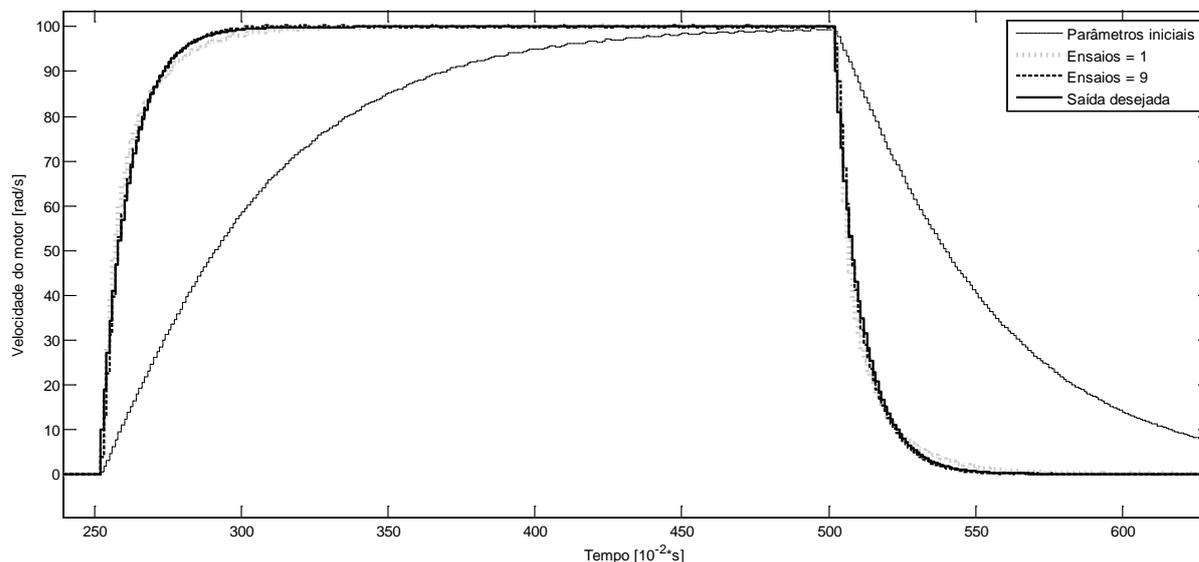


Figura 24: Comparação do sinal de saída do sistema com o uso dos parâmetros iniciais do controlador (linha traço-ponto) e com o uso dos parâmetros obtidos com a utilização do algoritmo (linha tracejada) em contraste com a saída desejada do sistema (linha contínua).

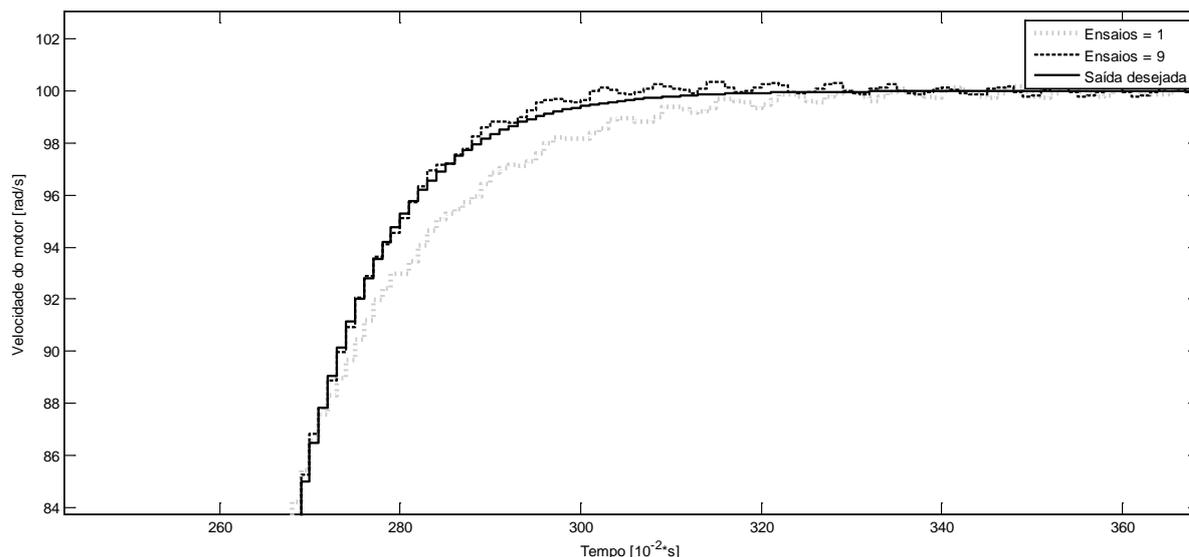


Figura 25: Detalhe da figura 24 apresentando o sinal de saída do sistema após uma única sintonia do método VRFT (linha tracejada) e após nove sintonias sucessivas (linha tracejada) em contraste com a saída desejada do sistema (linha contínua).

O método VRFT atinge uma resposta satisfatória após um único ensaio de sintonização, conforme a formulação teórica propõe. Um melhor vetor de parâmetros do controlador foi encontrado após uma segunda sintonização e manteve-se estável após outras sete sintonizações. Dessa forma, não é possível afirmar que ensaios sucessivos irão resultar em uma minimização efetiva dos custos para o sistema em condições diferentes da utilizada. Pode-se afirmar que um único ensaio de sintonização é suficiente para que o sistema apresente uma resposta muito próxima do modelo de referência proposto e que outras sintonizações podem reduzir sutilmente o custo, mas são mais afetadas pelo ruído do que no uso do método IFT.

## 6 CONCLUSÃO

### 6.1 CONCLUSÕES

Neste trabalho foi demonstrado que em um problema prático o uso dos métodos de controle baseado em dados VRFT e IFT atingiu o resultado desejado para o sistema descrito no trabalho. Ambos os métodos conseguiram minimizar o custo a um valor condizente com simulações apresentadas em FLORES DA CUNHA (2010). Dessa forma, os métodos demonstram-se como soluções confiáveis para o projeto de um controlador.

É interessante salientar que existem particularidades muito importantes a serem levadas em consideração ao se escolher um dos dois métodos para o projeto de um controlador. Para o método VRFT foram necessárias duas sintonizações para que a resposta do sistema alcançasse o modelo de referência desejado, embora o resultado após um único ensaio já é satisfatório. Com uso de dois ensaios de sintonização para o método VRFT foi necessário colocar o processo em operação duas vezes. Para obter um resultado análogo com o método IFT foram necessárias quatro iterações, sendo que para cada iteração é necessário colocar o processo em operação duas vezes, resultando em um total de 8 operações da planta.

Em contrapartida o método VRFT não é robusto em relação a ruídos, diferentemente do método IFT, e nos ensaios realizados neste trabalho foi possível perceber variações no controlador obtido mesmo em um ambiente com pouca presença de ruído. O método IFT mostra-se muito mais robusto e, para as condições especificadas neste trabalho, mais preciso por alcançar um valor de custo inferior ao método VRFT.

Sendo assim, pode-se imaginar que determinados processos são mais propensos à utilização do método VRFT ou do método IFT. Para processos em que o custo ou o tempo de operação da planta são restritivos o método VRFT se mostra como uma solução mais

desejável que o IFT por necessitar de uma única operação da planta para obter um resultado satisfatório. No entanto, para processos com elevado nível de ruído e que se necessite de uma maior precisão do resultado o método IFT é uma alternativa mais interessante.

## **6.2 SUGESTÕES PARA TRABALHOS FUTUROS**

Um estudo sobre as limitações práticas da aplicação e o desempenho de cada um dos métodos nessas condições é algo que o presente trabalho não abordou e é muito interessante de forma a complementar os resultados aqui obtidos. Analisar os efeitos em cada um dos métodos em situações limites, onde o modelo de referência é muito próximo da máxima resposta do sistema, irá apresentar maiores particularidades do VRFT e do IFT.

Outro tema a ser tratado no futuro é a presença de não-linearidades no sistema tais como zona-morta e folga (também encontrada na literatura como *backlash*) e qual o desempenho alcançado em uma aplicação prática em contraste com resultados teóricos como os obtidos em FLORES DA CUNHA (2010).

ANEXO:

Algoritmos Desenvolvidos para o Desenvolvimento do Trabalho.

## ANEXO A: ROTINA PARA O CÁLCULO DO VETOR DE PARÂMETROS DO CONTROLADOR ATRAVÉS DO MÉTODO VRFT

Após a aquisição dos dados de interesse no ensaio do sistema, é necessário calcular o novo vetor de parâmetros do controlador. Para isso, utiliza-se a seguinte rotina desenvolvida em MATLAB.

### VRFT1.m

```
clear custo Phi;
estrutura_controlador = 'PI';

%Ts é o tempo de amostragem
Ts = Tf;

%Vetor de parâmetros inicial do controlador
Teta=[0.01 0.001 0]';

%% Formulação da função de transferência desejada %%
%-----%
Td=tf([1],[1 0],Ts);
Td=tf([0.1],[1 -0.9],Ts);
Td=tf([0.5],[1 -0.5],Ts);
Td=tf([0.2],[1 -0.8],Ts);
%-----%

%Formulação do vetor de tempo de ensaio
t_aquis=length(r)*Ts;
tout=[0:Tf:(t_aquis-Ts)]';

%Formula a resposta desejada (yd) do modelo de referência(Td)
%ao sinal de referência utilizado no ensaio
yd=lsim(Td,r,tout);

%Salva os vetores de resposta em outras variáveis para tratamento posterior
Yd_inicial=yd;           %Salva o vetor saída desejada pré sintonia
Ref_inicial=r;          %Salva o vetor referência pré sintonia
Saida_inicial=y_out;    %Salva o vetor referência pré sintonia
Tempo_inicial=tout;     %Salva o vetor tempo pré sintonia

%Calculo do custo/critério de desempenho associado ao controlador inicial
custo(:,1) = sum((y_out-yd).^2)/(2*length(y_out));

%Função de transferência do termo proporcional do controlador
P = tf([1],[1],Ts);
%Função de transferência do termo integral do controlador
I = tf([1],[1 -1],Ts);
%Função de transferência do termo derivativo do controlador
D = tf([1],[1 -1],Ts);
```

```

%Formulação da função de transferência do modelo de referência escolhido
indice = 1;
grau_relativo = 0;
vetor_numerador=cell2mat(Td.num);

while (vetor_numerador(indice) == 0)
    indice = indice + 1;
    grau_relativo = grau_relativo + 1;
end

Td=inv(Td);
Td=Td*tf([1],[1 zeros(1,grau_relativo)],Ts);

%Formulação da referência virtual
r_virtual = lsim(Td,y_out,tout);
r_virtual = [r_virtual(1+grau_relativo:end)];

%Formulação do erro virtual
erro = r_virtual - y_out(1:end-grau_relativo);

%Adequação da dimensão do vetor de ação de controle para entrar em
%conformidade com demais vetores para fins de cálculo
u_out=u_out(1:end-grau_relativo);

%Cálculo do vetor Phi(controlador multiplicado pelo erro virtual)
Phi(:,1)=[lsim(P,erro,tout(1:end-grau_relativo))];
Phi(:,2)=[lsim(I,erro,tout(1:end-grau_relativo))'];

%Cálculo do vetor Teta que fornece o controlador desejado
Teta= (inv(Phi' * Phi))*Phi'*u_out;
Teta=Teta';

%Apresenta o vetor de parâmetros do controlador de acordo
%com a estrutura escolhida
%-----%
switch estrutura_controlador
    case 'P'
        Teta = [Teta(1);0;0]
    case 'PD'
        Teta = [Teta(1);0;Teta(2)]
    case 'PI'
        Teta = [Teta(1);Teta(2);0]
    case 'PID'
        Teta = [Teta(1);Teta(2);Teta(3)]
end
%-----%

```

## ANEXO B: ROTINA PARA O CÁLCULO DO VETOR DE PARÂMETROS DO CONTROLADOR ATRAVÉS DO MÉTODO IFT

Para o método IFT utilizou-se de três diferentes rotinas em MATLAB para que o método fosse aplicado sendo elas *passo1\_ift.m*, *passo2\_ift.m* e *passo3\_ift.m*. A cada ensaio realizado cada uma das rotinas é invocada, sendo que a cada realização das três rotinas uma iteração do método IFT é feita. A seguir apresentam-se as três rotinas.

### passo1\_ift.m

```
%% Rotina para aquisição dos dados referentes à primeira batelada de
%% experimentos do método IFT
```

```
% Define a estrutura do controlador utilizado
estrutura_controlador='PI';
```

```
Ts=Tf;
gama=1;
lambda=0;
g=tf([0.1],[1 -0.9],Ts);
no_iteracoes=10;
t_aquis=length(r)*Ts;
tout=[0:Tf:(t_aquis-Ts)];
yd=lsim(g,r,tout);
seletor=1;
y(:,1)=y_out;
u(:,1)=u_out;
y_in=[tout y(:,1)];
seletor=-1;
```

### passo2\_ift.m

```
%% Rotina para aquisição dos dados referentes à segunda batelada de
%% experimentos (experimento especial) do método IFT
```

```
y(:,2)=y_out;
u(:,2)=u_out;
seletor=1;
```

### passo3\_ift.m

```
%% Rotina para cálculo do gradiente da função custo e do vetor de
%% parâmetros do método IFT após a terceira batelada de experimentos
```

```
progressao_Kp(:,iteracao)=K(1);
progressao_Ki(:,iteracao)=K(2);
progressao_Kd(:,iteracao)=K(3);
```

```
progressao_saida(iteracao,:)=y_out;
progressao_entrada(iteracao,:)=u_out;
```

```
[K,J,R] = pidift(K, yd, y, u, gama, lambda, Ts, estrutura_controlador);
gradiente_custo(iteracao,:)=J;
```

```
custo(:,iteracao) = (sum((y_out-  
yd).^2)+lambda*sum(u_out.^2))/(2*length(y_out));  
  
custo_estimado(1)=custo(1);  
  
if(iteracao>1)  
if(custo(iteracao)>custo(iteracao-1))  
    gama = gama/2;  
    %gama = gama/i;  
    K(1)=progressao_Kp(iteracao-1);  
    K(2)=progressao_Ki(iteracao-1);  
    K(3)=progressao_Kd(iteracao-1);  
end  
  
custo_estimado(iteracao)=custo_estimado(iteracao-1) - gama*J'*inv(R)*J;  
  
end  
  
progressao_gama(:,iteracao)=gama;  
  
iteracao=iteracao+1;
```

## ANEXO C: FUNÇÃO PIDIFT.M

A rotina *loopift\_2.m* apresentada no Anexo B se utiliza da função *pidift.m* descrita a seguir.

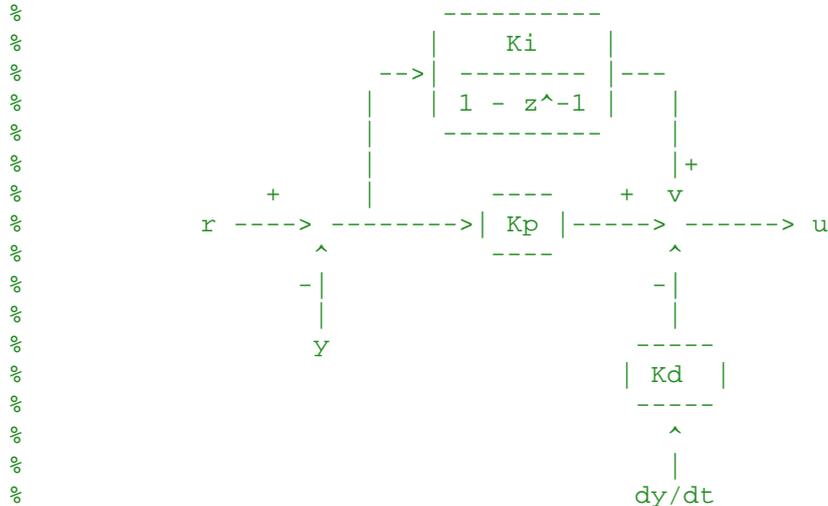
### pidift.m

```
function [K,J,R] = pidift(K, yd, y, u, gamma, lambda, Ts,
controller_structure)
```

```
% K = pidift - Calculate new PID gains from existing gains of a discrete
% PID controller
```

```
%
% K = pidift(K, yd, y, u)
```

```
% Calculate the new PID gains that will move the current system
% response closer to the desired system response. K is a vector of
% the current PID gains, [Kp, Ki, Kd], where the controller structure
% is of the following form:
```



```
% yd is the desired system output. y is an Nx3 column vector
% resulting from the following experiments:
```

```
% Start with a reference signal r. yd should be the desired
% closed-loop response of the system to the reference input r.
```

```
% y1,u1 = generated by system with the reference input r
% y2,u2 = generated by system with the reference input r - y1
% y3,u3 = generated by system with the reference input r
```

```
% Note that y1,u3 and y3,u3 must be generated from _separate_
% experiments.
```

```
% Once complete, y(:,1) = y1, etc.
```

```
% K = pidift(K, yd, y, u, gamma)
```

```
% Generate new gains with a step-size of gamma used in the
% gradient-descent of the cost function. gamma=0.1 when omitted.
```

```

%
% K = pidift(K, yd, y, u, gamma, lambda)
%
% Add input cost to the IFT cost-function. u is an Nx3 column vector
% of input signals generated from the same experiments as y.

%
% Source:
% H. Hjalmarsson et al., Iterative feedback tuning: theory and
% applications, Control Systems Magazine, IEEE, vol. 18, 1998, pp.
% 26-41.
%
% (C) 2008 Dan Miller danielmiller@ucsd.edu

% Force K into column form.
K = K(:);

if nargin < 4
    gamma = 0.1;
end
if nargin < 5
    lambda = 0;
end

% Extract Kp and Ki, Kd is not used directly in the cost calculations.
Kp = K(1);
Ki = K(2);
Kd = K(3);
%Ks = K(4);

% Extract the test signals.
y1 = y(:,1);
y2 = y(:,2);

u1 = u(:,1);
u2 = u(:,2);

% Construct the gradient PID controller functions.

switch controller_structure

    case 'P'

        dKp=tf([1 -1],[Kp -Kp],Ts);
        dKi=0;
        dKd=0;
        K=[K(1)]';

        % Calculate the gradient signals.
        dydKp = lsim(dKp, y2);
        dudKp = lsim(dKp, u2);

% Calculate the gradient of the cost function for each gain.
dJ = 0;
for i = 1:length(y1)
    dJ = dJ + (y1(i) - yd(i)) * [dydKp(i)] ...
        + lambda * u1(i) * [dudKp(i)];
end
dJ = dJ/length(y1);

```

```

% Calculate the Gauss-Newton-ish gradient matrix.
R = 0;
for i = 1:length(y1)
    dydrho = [dydKp(i)];
    dudrho = [dudKp(i)];

    R = R + dydrho*dydrho' + lambda*dudrho*dudrho';
end
R = R/length(y1);

% Calculate the new gains.
K = K - gamma*inv(R)*dJ;
K=[K(1) 0 0]';
J=dJ;
R=R;

    case 'PI'
        dKp=tf([1 -1],[Kp Ki-Kp],Ts);
        dKi = tf([1], [1 -1], Ts)*dKp;
        K=[K(1) K(2)]';

        % Calculate the gradient signals.
dydKp = lsim(dKp, y2);
dydKi = lsim(dKi, y2);

dudKp = lsim(dKp, u2);
dudKi = lsim(dKi, u2);

%dydKp
%dydKi

% Calculate the gradient of the cost function for each gain.
dJ = 0;
for i = 1:length(y1)
    dJ = dJ + (y1(i) - yd(i)) * [dydKp(i); dydKi(i)] ...
        + lambda * u1(i) * [dudKp(i); dudKi(i)];
end
dJ = dJ/length(y1);

% Calculate the Gauss-Newton-ish gradient matrix.
R = 0;
for i = 1:length(y1)
    dydrho = [dydKp(i); dydKi(i)];
    dudrho = [dudKp(i); dudKi(i)];

    R = R + dydrho*dydrho' + lambda*dudrho*dudrho';
end
R = R/length(y1);

%R=eye(2);

% Calculate the new gains.
%K = K - gamma*((inv(R)*dJ)/(norm(inv(R)*dJ)));

K = K - gamma*inv(R)*dJ;
K=[K(1) K(2) 0]';
J=dJ;
R=R;

```

```

case 'PD'
    dKp = tf([1 -1 0],[Kp+Kd -Kp-2*Kd Kd],Ts);
    dKd = tf([1 -1],[1 0], Ts)*dKp;
    dKi=0;
    K=[K(1) K(3)]';

    % Calculate the gradient signals.
    dydKp = lsim(dKp, y2);
    dydKd = lsim(dKd, y2);

    dudKp = lsim(dKp, u2);
    dudKd = lsim(dKd, u2);

    % Calculate the gradient of the cost function for each gain.
    dJ = 0;
    for i = 1:length(y1)
        dJ = dJ + (y1(i) - yd(i)) * [dydKp(i); dydKd(i)] ...
            + lambda * u1(i) * [dudKp(i); dudKd(i)];
    end
    dJ = dJ/length(y1);

    % Calculate the Guass-Newton-ish gradient matrix.
    R = 0;
    for i = 1:length(y1)
        dydrho = [dydKp(i); dydKd(i)];
        dudrho = [dudKp(i); dudKd(i)];

        R = R + dydrho*dydrho' + lambda*dudrho*dudrho';
    end
    R = R/length(y1);

    % Calculate the new gains.
    K = K - gamma*inv(R)*dJ;
    K = [K(1) 0 K(2)]';
    J=dJ;
    R=R;

case 'PID'
    dKp = tf([1 -1 0],[Kp+Kd Ki-Kp-2*Kd Kd],Ts);
    dKi = tf([1],[1 -1], Ts)*dKp;
    dKd = tf([1 -1],[1 0], Ts)*dKp;

    % Calculate the gradient signals.
    dydKp = lsim(dKp, y2);
    dydKi = lsim(dKi, y2);
    dydKd = lsim(dKd, y2);

    dudKp = lsim(dKp, u2);
    dudKi = lsim(dKi, u2);
    dudKd = lsim(dKd, u2);

    % Calculate the gradient of the cost function for each gain.
    dJ = 0;
    for i = 1:length(y1)
        dJ = dJ + (y1(i) - yd(i)) * [dydKp(i); dydKi(i); dydKd(i)] ...
            + lambda * u1(i) * [dudKp(i); dudKi(i); dudKd(i)];
    end
    dJ = dJ/length(y1);

```

```

% Calculate the Gauss-Newton-ish gradient matrix.
R = 0;
for i = 1:length(y1)
    dydrho = [dydKp(i); dydKi(i); dydKd(i)];
    dudrho = [dudKp(i); dudKi(i); dudKd(i)];

    R = R + dydrho*dydrho' + lambda*dudrho*dudrho';
end
R = R/length(y1);

% Calculate the new gains.
K = K - gamma*inv(R)*dJ;
J=dJ;
R=R;

end
%dKp = tf([1, -1], [Kp + Ki, -Kp], 1);
%dKi = tf([1, 0], [Kp + Ki, -Kp], 1);
%dKd = tf([1, -2, 1], [Kp + Ki, -Kp, 0], 1);

%dKp = tf([1, -1], [Kp + Ki, -Kp], Ts);

% p/ integrador z/z-1
%dKp = tf([1 -1 0],[Kp+Ki+Kd -Kp-2*Kd Kd],Ts);

% p/ integrador 1/z-1
%dKp = tf([1 -1 0],[Kp+Kd Ki-Kp-2*Kd Kd],Ts);

% p/ integrador z/z-1
%dKi = tf([1, 0], [Kp + Ki, -Kp], Ts);

% p/ integrador 1/z-1
%dKi = tf([1], [1 -1], Ts)*dKp;

%dKd = tf([1, -2, 1], [Kp + Ki, -Kp, 0], Ts);

%dKd = tf([1 -1], [1 0], Ts)*dKp;

%dKs = tf([1],[1],Ts)*dKp;

%R=eye(3);

end

```

## REFERÊNCIAS

BAZANELLA, A.S. GOMES DA SILVA JR. J. M.; **Sistemas de Controle: princípios e métodos de projeto**. Porto Alegre: Editora da Universidade/UFRGS, 2005. 297p. : il.

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

EVANS, W. R.; **Graphical Analysis of Control Systems**. Trans. AIEE 67 (1), p.547–551, 1948.

FLORES DA CUNHA, M.F.; **Iterative Feedback Tuning em Sistemas Sujeitos a Não-Linearidades Não-Diferenciáveis**. 2010. 113p. Tese (Mestrado em engenharia) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.

GEVERS, M. A decade of progress in interative process control design: from theory to practice. **Journal of Process Control**, Aachen, v.12, n.4, p519-531, June 2002.

HJALMARSSON, H.; GUNNARSSON, S.; GEVERS, M. A convergent iterative restricted complexity control design scheme. In: IEEE CONFERENCE ON DECISION AND CONTROL, 33., 1994, Lake Buena Vista. **Anais**. . . New York: IEEE, 1994. v.2, p.1735–1740.

LJUNG, L.; **System Identification: Theory for the User**. 2ed. Prentice Hall PTR, 1999.

MINORSKY, N.; **Directional stability of automatically steered bodies.** J. Amer. SOC. of Naval Engineers, p. 280-309, v.34, 1922.

NYQUIST, H.; **Regeneration theory.** Bell System Technical Journal, vol. 11, p.126-147, 1932.

OGATA, K.; **Engenharia de Controle Moderno.** 4.ed. São Paulo: Prentice Hall, 2003.