

**UNIVERSIDADE FEDERAL  
DO CEARÁ  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**

**CONTROLE DE VELOCIDADE APLICADO A UM  
SISTEMA DE ACIONAMENTO COM MOTOR DE  
RELUTÂNCIA VARIÁVEL**

Dissertação de Mestrado submetida à Universidade Federal do Ceará como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica.

**Fernando Barros Sobreira**

Fortaleza, Dezembro de 2008

# CONTROLE DE VELOCIDADE APLICADO A UM SISTEMA DE ACIONAMENTO COM MOTOR DE RELUTÂNCIA VARIÁVEL

Fernando Barros Sobreira

Esta dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia Elétrica, Área de Controle e Acionamentos e aprovado em sua forma final pelo programa de Pós-Graduação em Engenharia Elétrica na Universidade Federal do Ceará.

---

Otacílio da Mota Almeida, Dr.  
Orientador

---

Luiz Henrique Barreto, Dr.  
Coordenador do Programa de  
Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

---

Otacílio da Mota Almeida, Dr.  
Presidente

---

José Carlos Teles Campos, Dr.

---

Ricardo Silva Thé Ponte, Dr.

---

Cláudio Marques de Sa Medeiros, Dr.

*Dedico este trabalho a toda minha família que muito contribuíram de forma direta ou indiretamente para a elaboração e execução desta dissertação . . .*

“Não importa quantas pedras há no caminho, o mais importante é saber desvia-las.”

**Sandro Kretus**

## AGRADECIMENTOS

Aos meus pais, Fernando e Célia, por seus ensinamentos, investimentos, compreensão, amizade e conselhos. A minha irmã, Júlia, pelo apoio e amizade. À minha namorada, Ticiania, pelo apoio, conselhos, paciência e compreensão.

Aos meus amigos, Davi e Andre, por sempre estarem presentes.

Aos meus amigos Eber, Paulo e Tobias pelo incentivos e apoio no mestrado.

Aos meus colegas de mestrado pelo companheirismo, experiências e pelos bons momentos.

Ao professor doutor Otacílio, pelas orientações de projeto e pesquisas dentro da universidade.

A professora doutora Laurinda, pela compartilhamento de experiências e ajudas no projeto.

Ao CNPQ pelo apoio financeiro dado durante o mestrado na UFC.

E a todos os que contribuíram direta ou indiretamente para a realização desse trabalho.

Resumo do Projeto de dissertação de Mestrado submetido à Universidade Federal do Ceará.

# **CONTROLE DE VELOCIDADE APLICADO A UM SISTEMA DE ACIONAMENTO COM MOTOR DE RELUTÂNCIA VARIÁVEL**

**Fernando Barros Sobreira**

Dezembro / 2008

Orientador: Otacílio da Mota Almeida, Dr.

Área de Concentração: Controle e Acionamentos.

Palavras-chave: Motor de Relutância Variável, Controlador PID, Identificação, Método do Relé, DSP, Eletrônica de Potência e Acionamentos Elétricos.

Número de Páginas: 92

Este trabalho apresenta estudos e implementações práticas do controle de velocidade de um sistema de acionamento com máquinas de relutância variável (MRV). Inicialmente é feita uma revisão bibliográfica sobre o sistema de acionamento da máquina, na qual aspectos de identificação e controle das malhas de corrente e velocidade são abordados. Com base na revisão da literatura, descreve-se os elementos básicos na implementação prática do sistema composto pelo conversor estático de potência, sistema de condicionamento e aquisição de dados, o motor de relutância variável e a carga. Aspectos teóricos e práticos da malha de controle de corrente e velocidade são tratados. A malha de corrente foi implementada com um controlador PI em uma estrutura de auto-sintonia baseado na introdução de uma não linearidade do tipo relé, posicionado na malha de controle para o propósito de identificação da dinâmica da malha de corrente. Especial interesse foi dispensando ao controle de velocidade implementado através de uma estrutura RST que permite a associação com um estimador dos mínimos quadrados para identificação da dinâmica malha de velocidade. Considerações sobre estabilidade das malhas projetadas são estabelecidas.

Devido à velocidade necessária para o processamento dos polinômios de controle, utilizou-se uma estrutura apropriada de hardware para o processamento digital dos sinais. O sistema encontra-se montado na laboratório do GPAR (Grupo de Pesquisa em Automação e Robótica).

Abstract of dissertation presented to UFC as a fulfillment of the requirements for the degree of Master in Electrical Engineering.

# **SPEED CONTROL APPLIED TO A SWITCHED RELUCTANCE MOTOR SYSTEM DRIVER CONTROL**

**Fernando Barros Sobreira**

December / 2008

Advisor: Otacílio da Mota Almeida, Dr.

Area of Concentration: Control and Systems drivers.

Keywords: Switched Reluctance Motor, PID Controller, Identification, Relay Method, DSP, Power Electronics and Drivers.

Number of Pages: 92

This work presents studies and practical implementations of a speed control driver for a switched reluctance motors. The work begins with a literature review on the system drive of the machine, aspects of identification and control of the current and speed loops are addressed. Based on literature review, it describes the basic elements in the practical implementation of the system composed of a static power conversion, conditioning system and data acquisition, the switched reluctance motor and the load. Theoretical and practical aspects of the current and speed loops are treated. The current loop was implemented with a PI controller in a structure of self-tuning based on the introduction of a non-linearity of the relay type, positioned in the control loop for the purpose of identifying the dynamics of the current loop. Special interest was given to speed control implemented through a structure that allows the RST with a combination of least squares estimator to identify the dynamics of the speed loop. Considerations of loop stability are established.

Due to the speed required for the processing of polynomials of control, was used a hardware structure suitable for the processing of digital signals. The system is mounted on the laboratory GPAR (Research Group on Automation and Robotics).

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Abreviaturas</b>	<b>xiv</b>
<b>Lista de Símbolos</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	5
1.3 Estrutura da Dissertação . . . . .	6
<b>2 Identificação e Controle de Corrente e Velocidade de um Motor de Re-</b>	
<b>lutância Variável</b>	<b>7</b>
2.1 Introdução . . . . .	7
2.2 Modelagem do MRV . . . . .	8
2.3 Modelagem em Tempo Real do MRV: Identificação do Processo . . . . .	13
2.3.1 Método da Função Descritiva . . . . .	13
2.3.2 Identificação em Frequência - Método do Relé Setpoint . . . . .	15
2.3.3 Identificação da Malha de Velocidade - Mínimos Quadrados . . . . .	18
2.4 Estratégias de Controle e Sintonia . . . . .	19
2.4.1 Método de Ziegler-Nichols para Sintonia de Controladores PID . . . . .	21
2.4.2 Método de Ziegler-Nichols Modificado . . . . .	21
2.4.3 Projeto e Sintonia PID via Estrutura RST . . . . .	23



<b>3</b>	<b>Sistema de acionamento do Motor de Relutância Variável</b>	<b>26</b>
3.1	Estrutura do Sistema de Acionamento: Hardware e Software. . . . .	26
3.1.1	O Conversor de Potência . . . . .	28
3.1.2	O Sistema de Controle . . . . .	33
3.1.3	Unidade Central de Controle . . . . .	33
3.1.4	Observações Finais . . . . .	38
<b>4</b>	<b>Resultados Experimentais</b>	<b>40</b>
4.1	Introdução . . . . .	40
4.2	Identificação da malha de corrente . . . . .	40
4.3	Sintonia dos Controladores para Malha de Corrente . . . . .	43
4.4	Experimentos para identificação da malha de velocidade . . . . .	46
4.5	Sintonia PID para malha de Velocidade . . . . .	50
4.5.1	Sintonia do PID com Estrutura RST para malha de Velocidade . . . . .	52
4.6	Alteração do ponto de operação . . . . .	54
4.7	Software de Controle . . . . .	60
4.7.1	Considerações Finais . . . . .	63
<b>5</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>65</b>
5.1	Conclusões . . . . .	65
5.2	Propostas para trabalhos futuros . . . . .	67
<b>A</b>	<b>O processador DSP <i>TMS320F2812</i></b>	<b>69</b>
<b>B</b>	<b>Transdutores / Bancada Experimental</b>	<b>76</b>
B.0.1	Transdutores de Corrente e Tensão . . . . .	76
B.0.2	Encoder . . . . .	76
B.0.3	Bancada Experimental . . . . .	77
<b>C</b>	<b>Software</b>	<b>78</b>
C.1	Controlador PI corrente + PID Velocidade para <i>TMS320F2812</i> . . . . .	78

# Lista de Figuras

1.1	Sistema de acionamento do MRV com malha de realimentação de corrente.	4
2.1	MRV 12/8 3 fases . . . . .	8
2.2	Configuração de MRVs: a) Única saliência. b) Dupla saliência (4/2) . . . . .	9
2.3	Família de curvas de fluxo concatenado/corrente/posição para um MRV. (a) motor sem saturação (b) motor com saturação. . . . .	10
2.4	Co-energia no plano $\lambda \times i$ . . . . .	11
2.5	Circuito equivalente do MRV. . . . .	12
2.6	(a) Diagrama de blocos do processo com relé. (b) Formas de onda típicas do relé. (c) Diagrama de Nyquist $G(s)$ . . . . .	14
2.7	Diagrama de blocos para identificação do sistema pelo método do relé. . . . .	15
2.8	Sistema em malha fechada. . . . .	16
2.9	Ações do controlador PID no diagrama de Nyquist. . . . .	22
2.10	Estrutura RST. . . . .	24
3.1	Estrutura do sistema de acionamento. . . . .	26
3.2	Diagrama de blocos da bancada experimental para o sistema de controle da corrente do MRV. . . . .	27
3.3	Conexão entre dispositivos no sistema de acionamento. . . . .	28
3.4	Conversor Meia Ponte Assimétrica Trifásico. . . . .	30
3.5	Modos de condução por fase. . . . .	31
3.6	Interface de acionamento para MOSFET's. . . . .	32
3.7	Diagrama de blocos do sistema de acionamento. . . . .	34
3.8	Esquema de ligação para o Buffer de saída. . . . .	36
3.9	Divisor de tensão para o conversor $A/D$ . . . . .	37
3.10	Esquema de ligação do encoder 845GM – F1GSHC1024R. . . . .	37

3.11	Processo de configuração do algoritmo de controle. . . . .	38
3.12	Rotina principal do <i>software</i> . . . . .	39
4.1	Forma de onda de corrente do método do relé setpoint para $d = 1.5$ e $\varepsilon = 0.05$	42
4.2	Forma de onda de corrente do método do relé setpoint para $d = 1.5$ e $\varepsilon = 0.35$	42
4.3	Traçado dos pontos de identificação pelo método do relé no diagrama de Nyquist. . . . .	42
4.4	Diagrama de Nyquist da função identificada . . . . .	43
4.5	Resposta do controlador PI Ziegler-Nichols para $i = 2.5A$ , $\omega = 220rpm$ , $K_c = 5.49$ e $T_i = 6.7 \cdot 10^{-4}$ . . . . .	44
4.6	Variável de controle PI Ziegler-Nichols para $i = 2.5A$ , $\omega = 220rpm$ , $K_c = 5.49$ e $T_i = 6.7 \cdot 10^{-4}$ . . . . .	44
4.7	Resposta controlador PI Ziegler-Nichols modificado para $i = 2.5A$ , $\omega = 220rpm$ , $K_c = 172$ , $T_i = 1.23 \cdot 10^{-4}$ . . . . .	45
4.8	Variável de controle PI Ziegler-Nichols modificado para $i = 2.5A$ , $\omega = 220rpm$ , $K_c = 172$ , $T_i = 1.23 \cdot 10^{-4}$ . . . . .	46
4.9	Diagrama de blocos para simulação do método de identificação da malha de velocidade. . . . .	47
4.10	Resposta a um degrau em malha aberta de velocidade do MRV. . . . .	48
4.11	Variação dos parâmetros identificados para função de transferência de velocidade do MRV. . . . .	49
4.12	Diagrama de Nyquist da função de transferência de velocidade do MRV. . . . .	50
4.13	Resposta do sistema ao controlador PID aplicado a malha de velocidade do MRV com referência de $650rpm$ . . . . .	51
4.14	Variável de controle para o controlador PID aplicado a malha de velocidade do MRV com referência de $650rpm$ . . . . .	51
4.15	Pólos da função de transferência em malha fechada na estrutura RST. . . . .	53
4.16	Resposta do sistema ao controlador PID aplicado a malha de velocidade do MRV na estrutura RST com referência de $650rpm$ . . . . .	53
4.17	Variável de controle do para o controlador PID aplicado a malha de velocidade do MRV na estrutura RST com referência de $650rpm$ . . . . .	54
4.18	Resposta ao controlador de corrente PI Ziegler-Nichols modificado para uma mudança de referência de $2.5$ para $3.5A$ . . . . .	55

4.19	Resposta ao controlador de corrente PI Ziegler-Nichols modificado para uma mudança de referência de 3.5 para 2.5A . . . . .	55
4.20	Resposta do sistema ao controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 650 para 800rpm . . . .	56
4.21	Variável de controle para o controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 650 para 800rpm . .	56
4.22	Resposta do sistema ao controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 800 para 650rpm . . . .	57
4.23	Variável de controle para o controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 800 para 650rpm . .	57
4.24	Erro quadrático do controlador de corrente PI Ziegler-Nichols e Ziegler-Nichols Modificado. . . . .	59
4.25	Erro quadrático do controlador de velocidade PID na estrutura RST . . . .	59
4.26	Diagrama de fluxo do <i>software</i> de controle. . . . .	62
A.1	Diagrama de blocos do EVA. . . . .	71
A.2	Funcionamento do contador Timer 3. . . . .	72
A.3	Diagrama de blocos do circuito equivalente para a geração de PWMs. . . .	72
A.4	Funcionamento do PWM Assimétrico. . . . .	73
A.5	Diagrama de funcionamento circuito ADC. . . . .	74
B.1	Sensor gray 10 – <i>bits Allen-Bradley</i> . . . . .	77

# Lista de Tabelas

2.1	Tabela de ajuste de Ziegler-Nichols para controladores . . . . .	21
3.1	Dispositivos do sistema de acionamento. . . . .	29
4.1	Resultados de amplitude e período para relé setpoint para malha de corrente	41
4.2	Parâmetros do controlador PID Velocidade . . . . .	50
4.3	Parâmetros do controlador PID aplicado a malha velocidade na estrutura RST . . . . .	52
4.4	Desempenho dos controladores de corrente do MRV para regime transitório ( $M_s$ e $T_r$ ). . . . .	58
4.5	Desempenho dos controladores de velocidade do MRV para regime tran- sitório ( $M_s$ e $T_r$ ). . . . .	58
4.6	Desempenho dos controladores de corrente para regime permanente (erro quadrático e variância do erro). . . . .	60
4.7	Desempenho dos controladores de velocidade para regime permanente (erro quadrático e variância do erro). . . . .	60
B.1	Características elétricas do transdutor de tensão e corrente . . . . .	76

# Lista de Abreviaturas

<b>ADC</b>	<i>Analog to Digital Converter</i>
<b>CA</b>	<i>Corrente Alternada</i>
<b>CC</b>	<i>Corrente Continua</i>
<b>DSP</b>	<i>Digital Signal Processor</i>
<b>MQ</b>	<i>Mínimos Quadrados</i>
<b>MQR</b>	<i>Mínimos Quadrados Recursivos</i>
<b>MRV</b>	<i>Motor de Relutância Variável</i>
<b>PI</b>	<i>Proporcional e integral</i>
<b>PID</b>	<i>Proporcional, integral e Derivativo</i>
<b>PWM</b>	<i>Pulse Width Modulation</i>
<b>SRM</b>	<i>Switched Reluctance Motor</i>
<b>ZN</b>	<i>Ziegler-Nichols</i>
<b>ZNM</b>	<i>Ziegler-nichols Modificado</i>

# Lista de Símbolos

## Variáveis e Constantes

$L_a (L_{max})$	Indutância máxima
$L_u (L_{min})$	Indutância mínima
$N_r$	Número de pólos do rotor
$\theta_r$	Ângulo do pólo do rotor
$\theta_s$	Ângulo do pólo do estator
$\theta_{on}$	Ângulo de ligar a fase
$\theta_{off}$	Ângulo de desligar a fase
$\theta_c$	Ângulo de comutação
$\omega_b$	Velocidade de base
$\omega_r$	Velocidade angular
$V_{cc}$	Tensão do barramento
$U$	Tensão aplicada ao PWM
$i_{ref}$	Corrente de referencia
$v_{ref}$	Tensão de referencia
$D$	Ciclo de trabalho
$\lambda_j(\theta_r, i_j)$	Fluxo concatenado da fase $j$
$r_s$	Resistência de estator
$V_j$	Tensão da fase $j$
$\psi_s$	Fluxo constante
$W'$	Co-energia
$T_j$	Conjugado da fase $j$
$L_t(r, i)$	Indutância transitória
$E_j$	Força contra-eletromotriz da fase $j$
$kt$	Constante de conjugado

$\mathfrak{R}_r$	Relutância
$d$	atraso de transporte no tempo discreto
$e(t)$	erro de previsão
$J$	função custo
$L$	atraso de transporte no tempo contínuo
$q - 1$	operador atraso
$t$	tempo contínuo
$T_s$	período de amostragem
$u, u(t)$	sinal de entrada controle
$Y$	vetor de saídas
$y, y(t)$	sinal de saída
$K_c$	Ganho proporcional controlador PID
$T_i$	Tempo integral controlador PID
$T_d$	Tempo derivativo controlador PID



# Capítulo 1

## Introdução

### 1.1 Motivação

Atualmente, com o desenvolvimento da eletrônica de potência e avanços teóricos na área, tornou-se possível controlar a velocidade tanto de motores de corrente contínua (CC) como de corrente alternada (CA). Desta forma viabilizou-se o controle eficiente de corrente e velocidade das malhas de acionamento dos motores, não somente foi obtido maior eficiência na utilização da energia elétrica como também alcançado o objetivo final da aplicação. Na indústria, o motor de indução é o tipo mais aplicado, mas com a diversidade de aplicações e a necessidade de maior eficiência nos sistemas motrizes, ocorreu um impulso no sentido de obter construções mais simplificada e velocidades elevadas, possuindo estruturas robustas (Rômulo, 2007).

Os motores de relutância variável são potencialmente atrativos para muitas aplicações industriais por atender aos requisitos estabelecidos acima. Estes motores apresentam estrutura simples, alta velocidade e baixo custo de fabricação e manutenção (Oliveira *et al.*, 2006).

Um motor de relutância variável - MRV (Motor de Relutância Variável) consiste de um estator com enrolamento e um rotor sólido magnético de pólos salientes, sem enrolamentos que dispensa o uso de ímãs permanentes. Através da energização de um enrolamento um conjugado de relutância faz com que a parte móvel da máquina, rotor, mova-se. Em máquinas rotativas o rotor gira acompanhando a seqüência de chaveamento da corrente nas fases da máquina. Este movimento provoca variações na relutância do circuito magnético, e conseqüentemente variações na indutância dos enrolamentos de uma

mesma fase. Este comportamento justifica o nome da máquina como Máquina à Relutância Variável, ou também Máquina à Relutância Chaveada devido ao princípio de funcionamento das mesmas.

Um dos motores de relutância variável mais antigos que se tem registro foi construído na Escócia em 1838 (Henriques, 2004). Era utilizado para movimentar a locomotiva na estrada de ferro Glasgow-Edinburgh. Na década de 20 a patente do motor de passo registrada por C. L Walker em Alberden, inclui muitas características dos MRVs modernas. Topologicamente e eletromagneticamente os MRVs são idênticos aos motores de passo. A diferença principal é que os MRVs, diferente dos motores de passo, normalmente são operados com a realimentação da posição do eixo para a sincronia de correntes de fase com a posição do rotor (Rômulo, 2007).

O conceito do MRV foi estabelecido no século XIX (Nicolai, 1995), mas somente começou a ter o seu potencial explorado a partir do desenvolvimento da tecnologia dos semicondutores de potência, dos microprocessadores e de ferramentas matemáticas computacionais para o projeto de máquinas. Estes avanços permitiram a construção de acionamentos com motor de relutância variável com elevado desempenho. Com centenas de pesquisas e desenvolvimentos teóricos e práticos divulgados através de periódicos científicos, chegou-se ao conhecimento atual sobre os MRV (Miller, 2001). Como pontos favoráveis à utilização dos acionamentos com o MRV, são citados geralmete, a simplicidade de construção e economia comparado a motores de indução e síncronos (Harris e Miller, 1989). O MRV possui algumas desvantagens como a necessidade de sensor de posição, a oscilação de conjugado acompanhando as oscilações de corrente e o alto nível de ruído sonoro em relação aos motores de indução de mesma potência (Bass *et al.*, 1986). Além disso, a modelagem matemática é dificultada por possuir não linearidade magnética. Dessa forma, em relação aos aspectos da implementação de algoritmos de controle das malhas de corrente e velocidade, os MRVs constituem atualmente um desafio.

O acionamento eficiente de uma máquina de relutância variável para o controle adequado de conjugado e velocidade esta associado a um eficiente controle de corrente que é influenciado pela posição rotórica e conseqüentemente pela variação na indutância que provoca o surgimento de regimes dinâmicos não lineares que exigem linearizações em torno do ponto de operação ou aplicação de técnicas de controle não lineares ao sistema. Entre as estratégias de controle de corrente mais utilizadas pode-se citar: o con-

trole por histerese (Rim *et al.*, 2001), (Bianco *et al.*, 1996), controle proporcional-integral (PI) com ou sem malha de linearização por realimentação (“Feedback Linearization”) (Schulz e Rahuman, 2004), (Rômulo, 2007), (Sobreira *et al.*, 2008), (Reis *et al.*, 2008b), (Reis *et al.*, 2008a), controle ”Sliding Mode”, controladores baseados em lógica Fuzzy (Tanaka, 2001) e (Henriques, 2004), entre outros. Todas as estratégias de controle citadas utilizam técnicas de PWM (“*pulse width modulation*”) ou modulação delta.

Os controladores por histerese são os de mais fácil implementação, além de não requerem o conhecimento dos parâmetros da máquina. O modo de operação “on-off” destes controladores impõe variações à frequência de chaveamento do conversor, determinadas pela: i) variação na indutância da máquina, ii) banda de histerese, que tem impacto na oscilação da corrente e iii) tensão do barramento CC. Essas variações podem elevar a frequência de chaveamento a valores que ultrapassam o limite de operação das chaves do conversor e aumentando as perdas de chaveamento (Sung-Jun *et al.*, 2001).

Os controladores PI e PID com controle de tensão por PWM permitem operar o conversor em frequência fixa, realizando o controle da corrente a partir da variação do período ativo de cada ciclo de chaveamento. Apesar disto, sua resposta no domínio do tempo é de difícil otimização se nenhuma técnica de adaptação em tempo real de seus parâmetros for utilizada. Isto se deve à variação da indutância própria das fases da máquina de relutância variável.

A variação da indutância do MRV durante sua operação e seu comportamento não linear dificultam o projeto e a sintonia de um controlador PI ou PID com parâmetros fixos, para um desempenho uniforme em toda faixa de operação da máquina. Para contornar estes problemas, controladores que têm mecanismos de adaptação dos ganhos do controlador em função da posição rotórica, tem sido empregados como apresentados em (Schulz e Rahuman, 2004), (Hwu e Liaw, 2000), (Bae *et al.*, 2000). No entanto, estas estruturas exigem um esforço computacional maior que as implementações dos controladores simples e também de controladores por histerese e de tipo modulação delta.

A análise do desempenho dos controladores de corrente utilizados em sistemas de acionamentos de máquinas de relutância variável geralmente é baseada na resposta transitória, oscilação do conjugado, produção de ruído e vibração mecânica. Entre os fatores que interferem no desempenho dos controladores podem ser citados a faixa de velocidade de projeto e a indutância mútua entre fases, (Blaabjerg *et al.*, 1999).

Em alta velocidade, o efeito da força contra-eletromotriz se acentua, além de ocorrer um aumento na taxa de mudança da indutância. Estes fatores degradam a resposta do controle PI e beneficiam o controle por histerese que pode passar a apresentar um desempenho superior, (Blaabjerg *et al.*, 1999).

Em geral, nos sistemas de acionamento, o conversor de potência utilizado para acionar a máquina é uma fonte de tensão. A Fig. 1.1 apresenta os componentes básicos de um sistema de acionamento típico. O Conversor fonte de tensão é constituído basicamente por chaves eletrônicas e diodos, apresentando um mínimo de elementos ativos (apenas capacitores para filtro de entrada). Por sua vez as chaves do dispositivo de potência são acionadas através de um sistema de disparo, ou gatilho (“Gate Driver”) que recebem seus sinais de uma unidade de modulação por largura de pulso. Por fim, o controlador de corrente através de sua lei de controle determina de que forma a entrada de controle do sistema deve atuar para que o perfil desejado seja alcançado. O controlador de velocidade por sua vez utiliza a referência defini a referência para a malha de corrente. Embora as duas malhas trabalhem em conjunto, a malha de corrente é rápida enquanto a malha de velocidade é lenta em termos de constante de tempo.

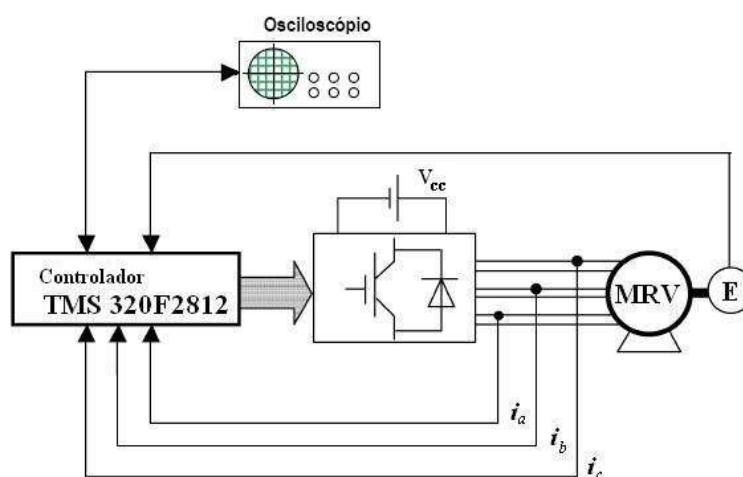


Figura 1.1: Sistema de acionamento do MRV com malha de realimentação de corrente.

A motivação para o desenvolvimento desta dissertação assenta-se sobre três aspectos principais.

- Incrementar a bancada de testes de MRV do GPAR com algoritmos de controle de corrente e velocidade.

- A utilização de técnicas de controle de corrente e velocidade que são típicas de controle de processos industriais em sistemas de acionamento de MRV que possui uma malha de corrente rápida e uma malha de velocidade lenta com dinâmicas não lineares e não modeladas.
- A aplicação de técnicas de controle ao MRV é um tema, que embora recorrente, continua a apresentar desafios que hoje são possíveis de serem trabalhados dado ao avanço tecnológico na área de eletrônica de potência e na área de microprocessadores.

## 1.2 Objetivos

O objetivo geral da dissertação é atualizar a bancada de controle do MRV do laboratório GPAR através da implementação de malhas eficientes para o controle de corrente e velocidade. Para isto alguns objetivos intermediários foram estabelecidos e podem ser enumerados como:

1. Reprojetar e reconstruir o hardware/software da bancada MRV para conferir mais funcionalidade e melhorias quanto à redução de ruídos (hardware) e atendimento do período de amostragem do sistema (software).

O projeto da bancada MRV envolve a remontagem do conversor de potência e as placas de condicionamento de sinais com o objetivo de reduzir o ruído nos componentes do sistema. Esta fase teve como base as dissertações desenvolvidas no laboratório GPEC que possuem experiências em montagem de conversores estáticos de potência: Souza (2003), BASTOS (2006) e Melo (2007).

2. Desenvolver o controle da malha de corrente do MRV adequando-a à implementação da malha de velocidade.

Para o adequado projeto de controle da malha de corrente utilizou-se a experiência que o GPAR vem adquirindo com as dissertações e publicações já desenvolvidas: Lima (2005), Rômulo (2007), Reis (2008) e Oliveira (2008).

O reprojetado de uma malha de controle envolve a determinação da dinâmica da malha e o projeto do controlador. Desde que a malha de corrente é não linear em função das variações da relutância, utilizou um método de identificação em frequência baseado na introdução de uma não linearidade do tipo relé na malha direta, para identificar

um ponto de operação da malha de corrente. Dos algoritmos testados no ambiente do GPAR, o método da ASTROM (1984), uma evolução do método de Ziegler Nichols freqüencial, foi utilizado para o projeto do controlador de corrente.

3. Desenvolver os algoritmos de controle de velocidade.

Para a modelagem da malha de velocidade utilizou-se o método dos mínimos quadrados enquanto o controlador foi implementado através de uma estrutura RST. Desde que a malha de velocidade é lenta (7 mais lenta que a malha de corrente) foi possível implementar sem dificuldades estas estruturas. A escolha destas estruturas, embora não tenha uma justificativa específica, visa acompanhar uma linha de desenvolvimento de algoritmos que possibilita a análise de robustez do sistema desacoplando o problema de estabilidade robusta.

### **1.3 Estrutura da Dissertação**

Inicialmente nessa dissertação foram abordados os aspectos gerais sobre o sistema de acionamento com máquina de relutância variável. Aspectos como técnicas utilizadas na identificação da malha de corrente e velocidades propostas, e implementações dos algoritmos de controle foram apresentados no segundo capítulo. No terceiro capítulo descreve-se a bancada experimental do GPAR, e são mostradas as estruturas do software e hardware utilizada nos experimentos. No capítulo seguinte apresenta-se os resultados experimentais obtidos com a implementação do protótipo do acionamento do MRV. As conclusões do trabalho são abordadas no quinto capítulo, também são sugeridas possíveis implementações e contribuições para trabalhos futuros.

## Capítulo 2

# Identificação e Controle de Corrente e Velocidade de um Motor de Relutância Variável

### 2.1 Introdução

O MRV é um motor elétrico em que o conjugado é produzido pela tendência do deslocamento de sua parte móvel para uma posição onde a indutância do enrolamento energizado é máxima (Miller, 2001). Ambos, o rotor e estator do MRV têm pólos salientes, mas nenhum enrolamento no rotor. O número de pólos do estator é geralmente diferente do número de pólos do rotor para evitar a eventualidade do rotor estar em um estado sem produção de conjugado inicial, que ocorre quando todos os pólos do rotor coincidem com os pólos do estator. O enrolamento do estator é bobinado em pólos opostos e conectado em serie ou paralelo, constituído de um número de circuitos separados eletricamente ou fases (Teixeira, 2008). Estes enrolamentos de fase podem ser energizados separadamente ou juntos dependendo do esquema de controle.

A Fig. 2.1 ilustra a seção transversal de um motor MRV 12/8, que tem 12 pólos no estator e 8 pólos no rotor. Este é um motor trifásico onde cada fase compreende três bobinas enroladas em pólos opostos. Sendo um motor deste tipo que está montado na bancada experimental do Laboratório de Automação e Controle do curso de Engenharia Elétrica da UFC.

Os MRVs apresentam várias configurações geométricas de acordo com o número de

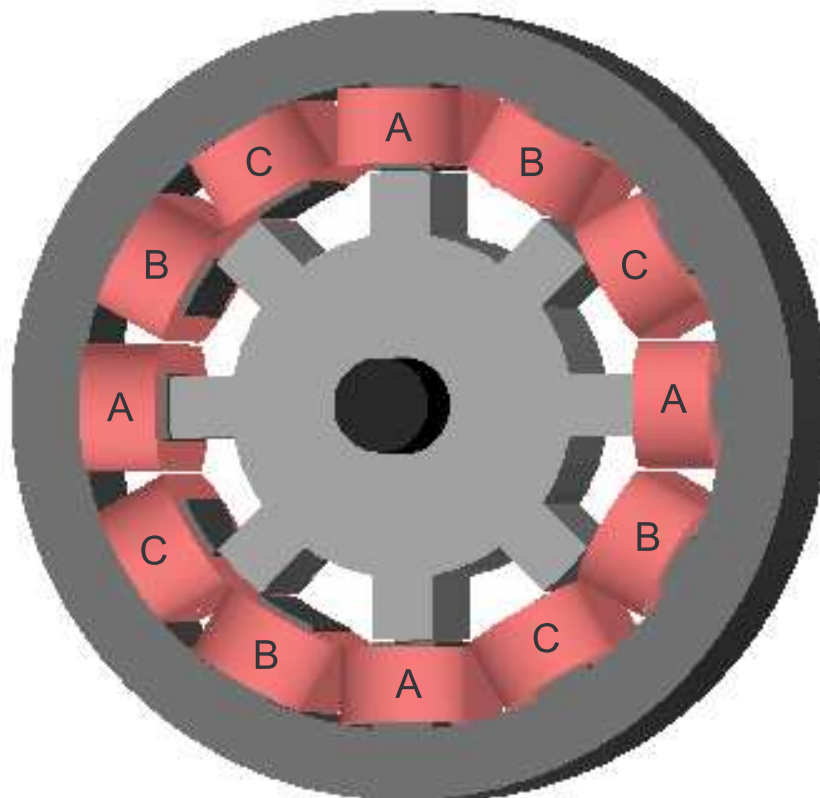


Figura 2.1: MRV 12/8 3 fases

pólos no rotor e estator, tais como, motores de  $4/2$  (duas fases),  $6/4$  (três fases),  $8/6$  (quatro fases),  $10/8$  (cinco fases), e  $12/8$  (quatro pólos e três fases) (Miller, 2001), (Teixeira, 2008). A Fig. 2.2 ilustra MRVs de polos com única e dupla saliência. Uma discussão de vantagens e desvantagens de diferentes geometrias é dada em (Miller, 2001), (Krishnan, 2001).

## 2.2 Modelagem do MRV

Como não existem enrolamentos no rotor conseqüentemente, não existe dinâmica elétrica associada ao rotor do motor, simplificando a sua análise. Entretanto, na prática estes motores trabalham em saturação; assim, o modelo matemático do MRV é não linear devido à influência da saturação magnética (Miller, 2001) e (Krishnan, 2001). A família de curvas  $(\theta_r, i)$  da Fig. 2.3 representa o fluxo concatenado com relação a corrente e a posição do rotor para um motor sem saturação e com saturação 2.3(a) e 2.3(b) respectivamente.

O MRV é descrito eletricamente da mesma forma que outros motores de CA. As



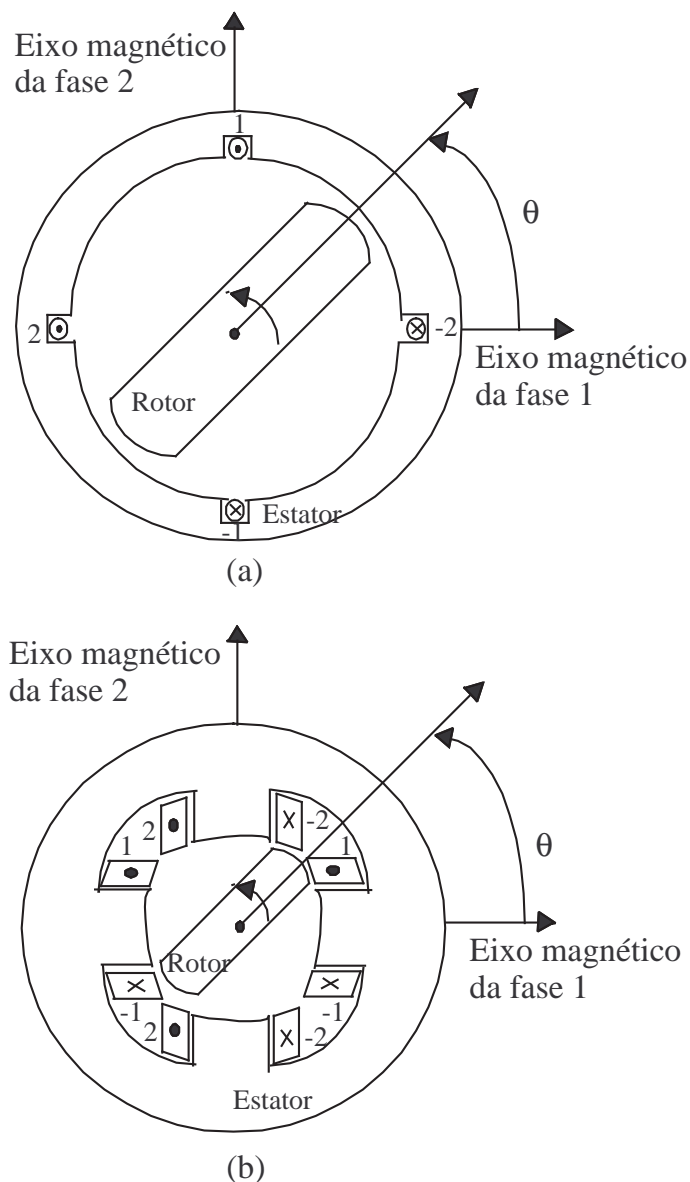


Figura 2.2: Configuração de MRVs: a) Única saliência. b) Dupla saliência (4/2)

equações para as fases do motor são associadas ao estator, isto é:

$$V_j = r_s i_j + \frac{d\lambda_j(\theta_r, i_j)}{dt} \quad (2.1)$$

$j = 1, 2, 3, \dots, m$ , e com a família de curvas  $\lambda_j(\theta_r, i_j)$  obtidas somente para uma fase, onde  $i_j$  é a corrente da fase  $j$ .

Da equação 2.1, o fluxo concatenado é uma função não linear da posição do rotor  $\theta_r$ , em relação ao ponto de alinhamento entre os pólos do estator e rotor, e também, da corrente. A característica de fluxo concatenado, em geral, é modelada utilizando-se linearização por

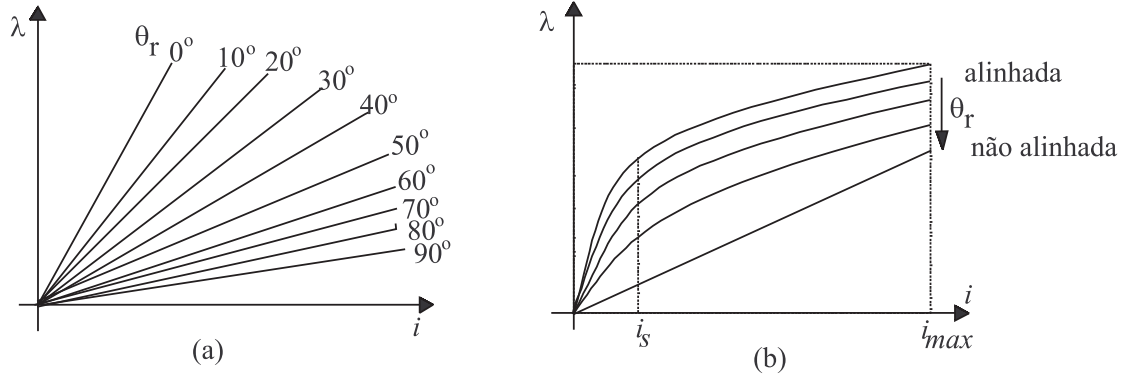


Figura 2.3: Família de curvas de fluxo concatenado/corrente/posição para um MRV. (a) motor sem saturação (b) motor com saturação.

partes (Ray, 1986). De acordo com (Ilic-Spong *et al.*, 1987), o fluxo pode ser aproximado por uma função contínua, que leva em conta estas não linearidades.

O MRV usa o princípio eletromagnético básico que converte energia elétrica em energia mecânica. Porém, nem toda energia fornecida pela fonte de potência é convertida em mecânica. Parte dessa energia é armazenada no campo magnético durante o processo de “magnetização”, e pode retornar à fonte durante o período de “desmagnetização” (Miller, 2001), (Krishnan, 2001). A região da curva  $\lambda \times i$  envolvida pela trajetória em um ciclo elétrico ( $\Delta W'$ ) representa a co-energia eletromagnética total convertida em conjugado eletromecânico, conforme ilustra a figura 2.4.  $\delta W'$  é o trabalho produzido pela variação de  $\delta\theta_r$  (GAO, 2000).

Um modo de se controlar o MRV é energizando uma fase por vez. Assim, o conjugado de saída é o resultado da contribuição das fases individualmente, uma depois da outra. Logo, a análise pode ser feita durante a operação de uma fase e repetida para outras fases (Ray *et al.*, 1986). A expressão para o conjugado produzido pela fase  $j$  é determinada pela derivada da co-energia  $W'_j$  com relação à  $\theta_r$  (GAO, 2000):

$$T_j(\theta_r, i_j) = \frac{\partial W'_j}{\partial \theta_r}(\theta_r, i_j) = \frac{\partial}{\partial \theta_r} \int_0^{i_j} \lambda_j(\theta_r, i_j) di_j \quad (2.2)$$

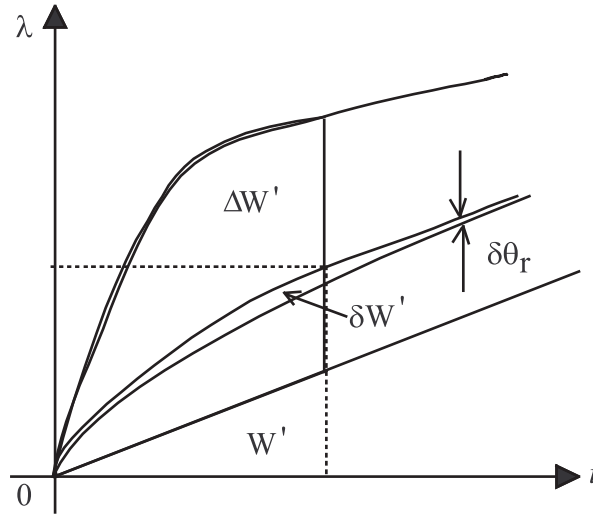


Figura 2.4: Co-energia no plano  $\lambda \times i$ .

sendo,

$$\partial W'_j = \int_0^{i_j} \lambda_j(\theta_r, i_j) di_j. \quad (2.3)$$

As equações dinâmicas de movimento são:

$$J \frac{d\omega_r}{dt} = T_e - B_m \omega_r - T_L \quad (2.4)$$

em que

$$\omega_r = \frac{d\theta_r}{dt}, \quad (2.5)$$

$J$  é o momento de inércia do rotor,  $B_m$  é o coeficiente de atrito do rotor,  $T_L$  é o torque da carga e  $\omega_r$  é denotado como velocidade angular do rotor.

Da equação 2.1 tem-se:

$$V_j = R_j i_j + \frac{\partial \lambda_j}{\partial i_j} \frac{di_j}{dt} + \frac{\partial \lambda_j}{\partial \theta_r} \frac{d\theta_r}{dt}. \quad (2.6)$$

Denomina-se  $\frac{\partial \lambda_j}{\partial i_j}$  como a indutância transitória  $L_t$ :

$$L_t(\theta_r, i_j) = \frac{\partial \lambda_j(\theta_r, i_j)}{\partial i_j}. \quad (2.7)$$

Considerando a equação (2.5), e o último termo da equação (2.6), representa-se a

f.c.e.m.  $E_j$  como,

$$E_j = \frac{\partial \lambda_j}{\partial \theta_r} \omega_r. \quad (2.8)$$

Assim, a equação (2.6) torna-se:

$$V_j = R_j i_j + L_t(\theta_r, i_j) \frac{di_j}{dt} + E_j(\omega_r, \theta_r, i_j) \quad (2.9)$$

similar à de um motor CC série devido a curva de conjugado inverso com a corrente.

A equação (2.9) é bastante genérica e mostra que nos terminais do MRV tem-se um circuito equivalente que consta, em cada fase, de uma resistência, uma indutância, e uma f.c.e.m., de forma geral, similar a outros motores (Tormey e Torrey, 1991), e apresentada na Fig. 2.5.

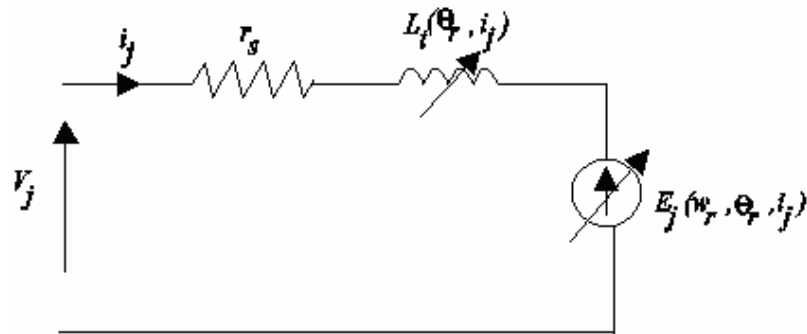


Figura 2.5: Circuito equivalente do MRV.

Nem toda toda energia obtida da fonte de alimentação é convertida em trabalho mecânico. Parte desta energia é armazenada no campo magnético (GAO, 2000). Se não existir saturação magnética, as curvas magnéticas serão linhas retas. Neste caso, em qualquer posição  $\theta_r$  do rotor, a co-energia e a energia magnética armazenada são iguais e dados por (GAO, 2000):

$$W_f = W' = \frac{1}{2} L(\theta_r) i^2, \quad (2.10)$$

então o conjugado instantâneo, equação (2.2), é reduzido para (GAO, 2000):

$$T_j = \frac{1}{2} i_j^2 \frac{dL_j(\theta_r)}{d\theta_r}. \quad (2.11)$$

Em MRVs de múltiplas fases, o conjugado instantâneo é dado por um somatório (GAO,

2000)

$$T_e = \sum_{j=1}^m T_j, \quad (2.12)$$

onde  $m$  é o total de fases.

## 2.3 Modelagem em Tempo Real do MRV: Identificação do Processo

Para desenvolver um controlador adequado para um sistema é necessário ter um conhecimento prévio deste. Neste Trabalho utiliza-se métodos de identificação baseados em dados de entrada e saída das malhas de corrente e velocidade. Duas abordagens típicas de modelagem baseadas em dados serão utilizadas com o propósito de identificação das dinâmicas das malhas de corrente e velocidade do MRV.

1. “*relé setpoint*”, que utiliza o método da função descritiva com o propósito de identificação em frequência da malha de corrente utilizado para identificação do modelo da malha de corrente;
2. mínimos quadrados, utilizado para identificação do modelo da malha de velocidade.

### 2.3.1 Método da Função Descritiva

O método da função descritiva constitui-se em uma técnica para modelagem de sistemas não lineares (Khalil, 1996). A idéia básica do método consiste em determinar a relação entre um sinal de entrada e um sinal de saída representados por componentes harmônicas fundamentais senoidais.

Relés do tipo “*on-off*” com histerese e saturações são elementos não lineares que quando posicionados na malha direta com outro sistema linear ou não linear, conforme a figura 2.6, sob certas condições estabelecem ciclos limites de oscilações que determinam pontos de operações no plano complexo de Nyquist. A partir da definição do ponto de operação pode-se determinar funções de transferências do processo no ponto considerado.

Com o relé na malha de realimentação, as oscilações podem ser controladas e a função do sistema  $G(s)$  pode ser determinada. A condição para a oscilação ser mantida na malha

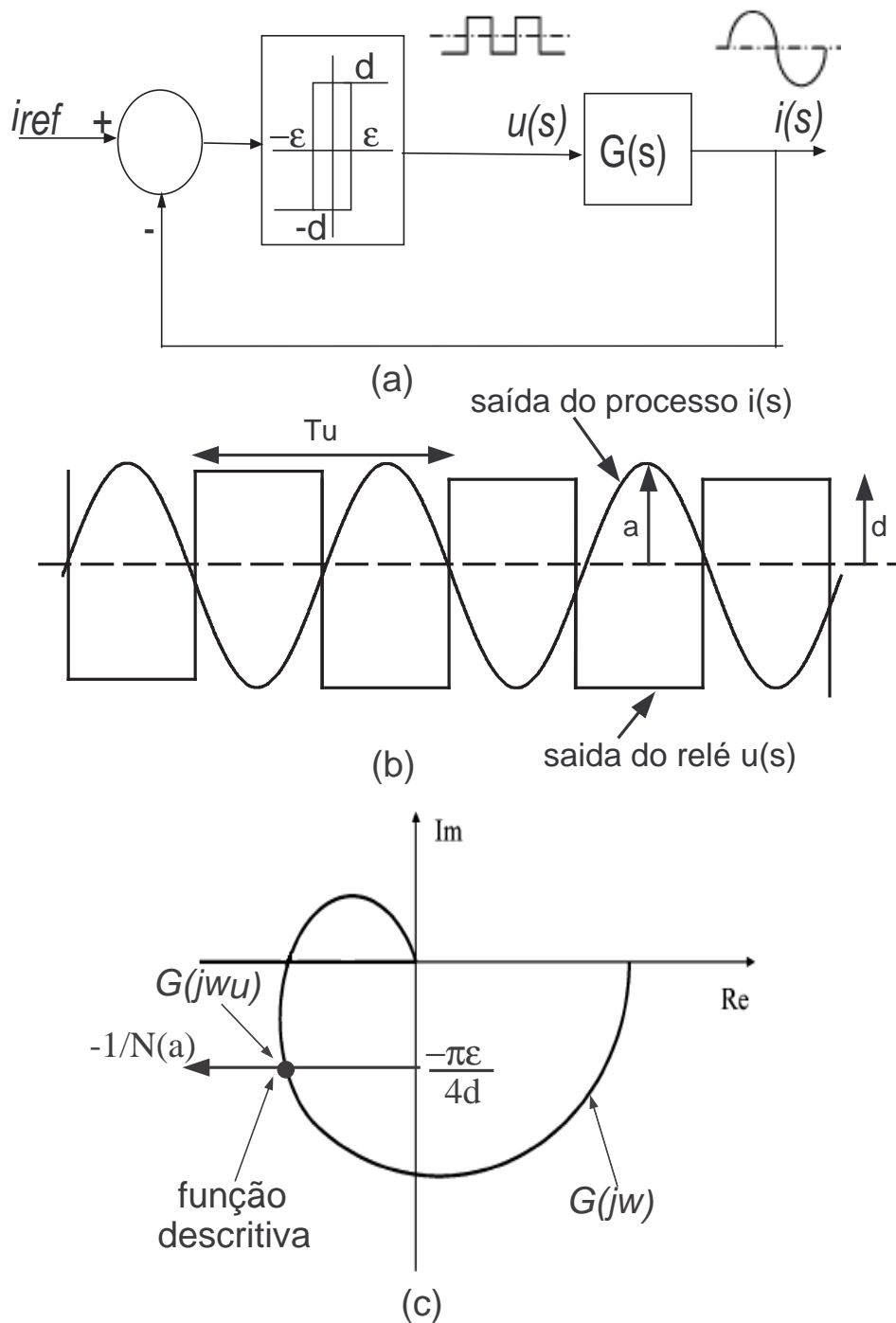


Figura 2.6: (a) Diagrama de blocos do processo com relé. (b) Formas de onda típicas do relé. (c) Diagrama de Nyquist  $G(s)$ .

é dada por

$$N(a).G(j\omega) = -1 \quad (2.13)$$

onde  $\omega$  é a frequência de oscilação da saída do sistema. A condição (2.13) também é representada pelo diagrama de Nyquist da Fig. 2.6(c) onde  $G(j\omega)$  intercepta  $-1/N(a)$  e estabelece um ciclo limite que oscila o sistema no ponto de operação. Se um relé sem histerese ( $\varepsilon \rightarrow 0$ ) é utilizado, pode-se determinar no diagrama de Nyquist o ponto de cruzamento de ganho. Se reles com histereses são utilizados, outros pontos do diagrama são determinados. Dado um valor de  $\varepsilon$  e  $d$ , a função de transferência na frequência de operação é dada por

$$G(j\omega_0) = \frac{-\pi}{4d} \sqrt{a^2 - \varepsilon^2} - \frac{\pi\varepsilon}{4d}, \quad (2.14)$$

onde  $d$  é a amplitude do relé,  $\varepsilon$  é a largura da histerese, e  $a$  é a amplitude das oscilações resultantes na saída da planta com controle do relé com histerese (Reis *et al.*, 2008a).

A função  $1/N(a)$  pode ser representada como uma linha reta paralela ao eixo real no plano complexo, como mostra a figura 2.6(c). O método de identificação através do relé, facilita a sintonia de controladores PID (Åström e Hägglund, 1986). Os mais comuns entre estes métodos são os métodos de Ziegler-Nichols e Ziegler-Nichols modificado. Estes foram utilizados no desenvolvimento deste trabalho.

### 2.3.2 Identificação em Frequência - Método do Relé Setpoint

O método relé “setpoint” foi proposto por (Luo *et al.*, 1998) e constitui uma extensão do método proposto por (Åström e Hägglund, 1986). A Fig. 2.7 apresenta o diagrama de blocos da aplicação do método.

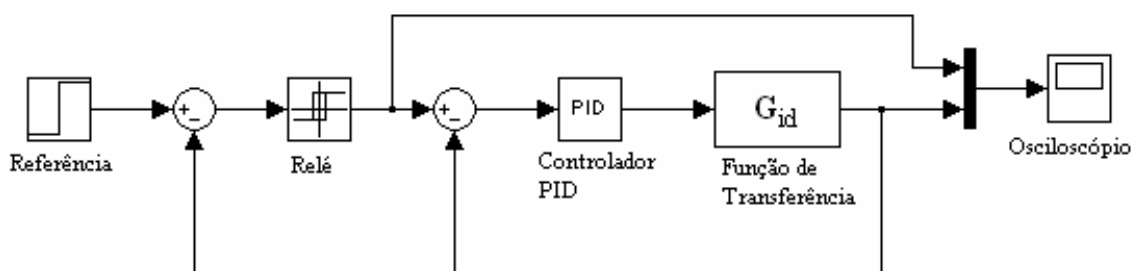


Figura 2.7: Diagrama de blocos para identificação do sistema pelo método do relé.

O método tem como objetivo determinar um ponto na curva de Nyquist para a função

de transferência da planta. Este ponto é chamado de ponto crítico,  $K_u$  e  $T_u$ , em que os parâmetros do controlador são determinados a partir destes ou de um outro ponto qualquer do diagrama de Nyquist, obtidos através de experimentações.

Considerando-se a modelagem não linear por função descritiva, o diagrama de blocos do sistema resultante para o relé e o sistema é apresentado na Fig. 2.8.

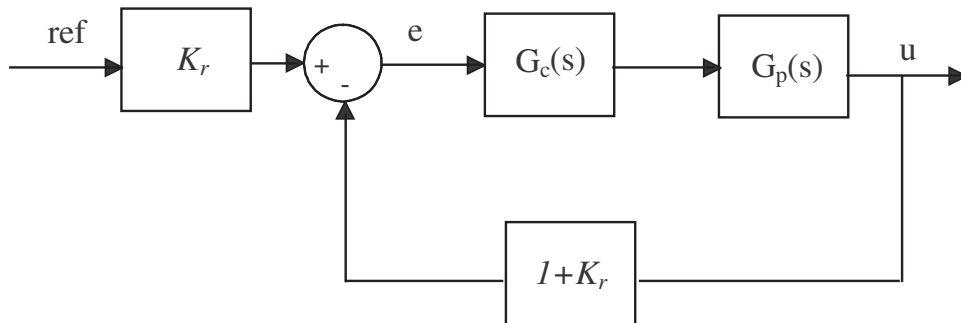


Figura 2.8: Sistema em malha fechada.

A função de transferência de malha aberta é dada como:

$$(1 + K_r) G_p(s) G_c(s). \quad (2.15)$$

O erro  $e(t)$  é calculado como:

$$e(t) = K_r r_{ef} - (1 + K_r) u(t) \quad (2.16)$$

onde  $K_r$  é o ganho do relé,  $ref$  é a referência e  $u$  é a variável de controle.

Muitos processos industriais podem ser modelados por funções de transferências de baixa ordem com atrasos de transporte (Luyben, 1996) e (Chien e Fruehalf, 1990). Neste trabalho, para propósito de modelagem em frequência com o método do relé da malha de corrente, utiliza-se a função de transferência contendo somente um ganho, um atraso de transporte e um pólo fora da origem:

$$G_{p(i)}(s) = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (2.17)$$

sendo  $\theta$  é o atraso de transporte,  $\tau$  é a constante de tempo dominante do sistema e  $K_p$  é o ganho do processo.

Considera-se a função de transferência do controlador PI da malha de corrente dada



por:

$$G_c(s) = K_c \left( 1 + \frac{1}{T_i s} \right) \quad (2.18)$$

sendo,  $K_c$ , e  $T_i$ , representam respectivamente o ganho proporcional, e o tempo integral.

Para oscilações na saída da Fig. 2.8 obtém-se:

$$\arg [(1 + K_r) G_p(j\omega_o) G_c(j\omega_o)] = -\pi \quad (2.19)$$

ou seja,

$$\arg \left[ (1 + K_r) \frac{K_p e^{-\theta s}}{\tau s + 1} \frac{K_c (T_i s + 1)}{T_i s} \right] \Bigg|_{s = -j\omega_o} = -\pi, \quad (2.20)$$

e

$$|(1 + K_r) G_p(j\omega_o) G_c(j\omega_o)| = 1 \quad (2.21)$$

ou seja,

$$\left| (1 + K_r) \frac{K_p e^{-\theta s}}{\tau s + 1} \frac{K_c (T_i s + 1)}{T_i s} \right| \Bigg|_{s = j\omega_o} = 1. \quad (2.22)$$

Da condição de fase, equação (2.20), o atraso de transporte é dado por:

$$\theta = \frac{\left[ \frac{\pi}{2} + \tan^{-1}(T_i \omega_o) - \tan^{-1}(T_i \omega_o) \right]}{\omega_o}. \quad (2.23)$$

Da equação (2.22) a constante de tempo é dada por:

$$\tau = \frac{K_p \sqrt{(1 + K_r) K_c (T_i^2 \omega_o^2 + 1)} - T_i \omega_o}{\omega_o \sqrt{T_i \omega_o}}. \quad (2.24)$$

A maior vantagem da aplicação desse método é a possibilidade de manter o sistema em malha fechada, garantindo a estabilidade do sistema, uma vez que existe um controlador pré-ajustado atuando durante o processo de identificação (Padhy e Majhia, 2006), além de proteger a integridade da máquina. Por outro lado, é necessário uma pré-sintonia do controlador utilizado. Neste trabalho foi realizada a identificação da função de transferência da malha de corrente do MRV utilizando-se o método do relé “*setpoint*”. A modelagem desenvolvida nesta seção será utilizada nos tópicos seguintes para o projeto dos controladores.

### 2.3.3 Identificação da Malha de Velocidade - Mínimos Quadrados

Para a identificação da malha de velocidade do MRV, utilizou-se o método dos mínimos quadrados. Este método foi utilizado por ter como base uma estrutura de regressão linear que permite a implementação do controle adaptativo, além de proporcionar uma identificação rápida e ser de fácil interpretação.

Controladores adaptativos são relativamente simples de serem implementados na malha de velocidade desde que esta malha tenha constante de tempo mais lenta quando comparada com a malha de corrente.

O método dos mínimos quadrados é uma técnica de estimação que minimiza o erro quadrático médio entre o valor medido da saída e o valor estimado (Aguirre, 2007).

Seja um processo físico caracterizado por uma entrada,  $u(t)$ , uma saída,  $y(t)$  e um ruído  $\ell(t)$  então a função de transferência discreta linear é dada por (Aguirre, 2007):

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t) + \ell(t) \quad (2.25)$$

sendo:

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_naz^{-na} \quad (2.26)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + \dots + b_nbz^{-nb} \quad (2.27)$$

polinômios que definem a estrutura do modelo do processo em malha aberta. A variável  $d$  corresponde ao tempo morto de processo.

Escrevendo (2.25) na forma de equação a diferenças tem-se:

$$y(t) = -a_1y(t-1) - \dots - a_nay(t-na) + b_0u(t-d) + \dots + b_nbu(t-d-nb) + \ell(t), \quad (2.28)$$

sendo  $\ell(t)$  um ruído de medição,

Se o ruído de medição apresenta autocorrelação diferente de zero ou apresenta amostras correlacionadas com a entrada ou saída, este é classificado como ruído colorido, pois apresenta dinâmica estimável. No entanto, quando a média é nula e há decorrelação com entrada ou saída, o ruído é branco e nenhuma dinâmica poderá ser estimada. Desta

forma o vetor de medidas é definido somente pelas entradas e saídas passadas dado por:

$$\varphi^T(t) = \begin{bmatrix} -y(t-1) & -y(t-2) & \dots & -y(t-na) \\ u(t-d) & \dots & u(t-d-nb) \end{bmatrix} \quad (2.29)$$

e o vetor de parâmetros dado por:

$$\theta^T(t) = [ a_1 \ a_2 \ \dots \ a_{na} \ b_0 \ b_1 \ \dots \ b_{nb} ]. \quad (2.30)$$

Pode-se então escrever o modelo da equação 2.28 na forma matricial:

$$y(t) = \varphi^T(t)\theta^T(t), \quad (2.31)$$

Admitindo-se  $N$  medidas suficientes para se determinar os parâmetros tem-se:

$$\begin{bmatrix} y(0) \\ y(1) \\ \dots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} \varphi^T(0) \\ \varphi^T(1) \\ \dots \\ \varphi^T(N-1) \end{bmatrix} \theta. \quad (2.32)$$

Fazendo o uso da pseudo-inversa (Meister, 2006), a matriz de coeficientes do sistema,  $\theta$ , pode ser determinado por:

$$\theta = [\varphi^T \varphi]^{-1} \varphi^T y \quad (2.33)$$

## 2.4 Estratégias de Controle e Sintonia

O controle efetivo de corrente de um MRV depende do controle de dois parâmetros: a magnitude de corrente no enrolamento do motor e o instante em que essa entra em condução com relação à posição do rotor. Da equação de conjugado, (2.11), observa-se que este depende do quadrado da corrente e da variação de indutância com a posição do rotor. Como o último termo depende das características construtivas da máquina, o controle da corrente adequado deve ser eficiente para se obter um perfil de conjugado desejado. Dessa forma, o MRV pode ser controlado levando-se em conta a corrente ou pelo comportamento de sua indutância, ou de ambos, de modo a se obter o perfil de conjugado

desejado.

Devido à simplicidade de implementação e à capacidade de satisfazer a maioria dos requisitos do projeto, controladores PI e PID de estrutura fixa são muito utilizados em nível industrial (Åström e Hägglund, 1986) e (Takatsu e Itoh, 1999). Entretanto, mesmo sendo estratégias conhecidas e dominadas pelos engenheiros de controle, os aspectos de sintonia continuam um desafio tanto na engenharia quanto na pesquisa. As dificuldades nos projetos são proporcionais à complexidade dos requisitos de controle exigidos (por exemplo: resposta rápida do sistema e restrições nas variáveis) e às complexidades presentes na dinâmica do processo a ser controlado, ou seja não-linearidades, atraso de transporte, fase não-mínima, interações nas malhas.

Pode-se citar o método de Åström e Hägglund (1986) como um exemplo de sucesso no meio industrial para a sintonia de controladores PID. Esta técnica constitui um marco no desenvolvimento de métodos sistemáticos de ajuste de controladores PID, apresentando duas características de suma importância para o controle de processos industriais, tais como (Åström e Wittenmark, 1998):

1. simples de projetar e implementar;
2. não requer o conhecimento dos parâmetros do processo, ou seja, a identificação explícita de um modelo para o processo.

Nesta dissertação aplica-se uma estratégia de controle PI para a malha de corrente a partir da linearização da malha em torno de um ponto de operação. Um método de identificação freqüencial denominado de relé “setpoint” (Luo *et al.*, 1998) é utilizado para realizar a identificação do sistema e as estratégias de Ziegler-Nichols e Ziegler-Nichols modificado proposto por Åström e Hägglund (1986) são aplicadas para realizar o ajuste do controlador PI.

Para a malha de velocidade implementou-se uma estratégia de controle PID. O ajuste dos parâmetros do controlador foi realizado através da estratégia de Ziegler-Nichols sendo configurado em um estrutura de dois graus de liberdade, como apresentado em Wellstead e Zarrop (1991).

### 2.4.1 Método de Ziegler-Nichols para Sintonia de Controladores PID

O método freqüencial proposto por Ziegler e Nichols (1943), baseia-se no conhecimento de um ponto do diagrama de Nyquist da função de transferência do processo denominado ponto crítico  $(\omega_u, K_u)$  e caracterizado pelo ponto no qual o mapeamento de  $G_p(j\omega)$  intercepta o eixo real negativo. Neste ponto tem-se que:

$$K_u = \frac{1}{|G(j\omega_u)|}. \quad (2.34)$$

O inverso da freqüência de cruzamento  $\omega_u$  é conhecido como período crítico, tal como a equação 2.35.

$$T_u = \frac{2\pi}{\omega_u} \quad (2.35)$$

O ajuste dos parâmetros dos controladores PI e PID é realizado seguindo a orientação apresentada na tabela 2.1.

Tabela 2.1: Tabela de ajuste de Ziegler-Nichols para controladores

Controlador	$K_c$	$T_i$	$T_d$
P	$0,5K_u$	-	-
PI	$0,4K_u$	$0,8T_u$	-
PID	$0,6K_u$	$0,5T_u$	$0,12T_u$

Este método não apresenta flexibilidade de projeto, as especificações são rígidas e sua característica oscilatória leva o processo a operar com pequena margem de estabilidade, o que sob a influência de dinâmicas não modeladas pode levar à instabilidade do sistema.

### 2.4.2 Método de Ziegler-Nichols Modificado

Este método foi proposto por Åström e Hägglund com o objetivo de superar deficiências do método tradicional de Ziegler-Nichols (Åström e Hägglund, 1999). Neste método apenas um teste experimental é necessário, a amplitude de saída do processo pode ser restringida pelo ajuste da amplitude do relé, o processo não é levado ao limite da estabilidade e pode ser facilmente automatizado para utilização em produtos comerciais (Åström e Hägglund, 1999). Uma forma de equacionar o método pode ser realizada a partir da interpretação do método de Ziegler-Nichols no domínio da freqüência. Neste domínio o método pode

ser interpretado com sendo uma modificação através de um controlador PID da função de transferência de malha aberta do processo. Ou seja, com um controlador PID é possível mover um determinado ponto “A” do diagrama de Nyquist  $G(j\omega)$ , para uma posição especificada em termos de fase e ganho para o sistema em malha fechada. As ações proporcional (P), integral (I) e derivativa (D) do controlador PID, em termos de diagrama de Nyquist, são apresentadas na Fig. 2.9.

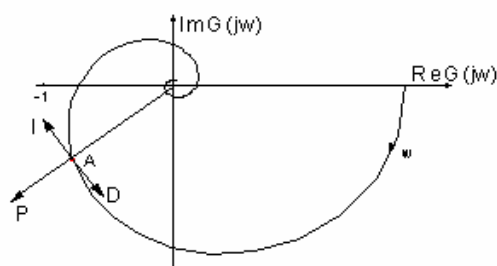


Figura 2.9: Ações do controlador PID no diagrama de Nyquist.

A partir da escolha de um ponto na curva de Nyquist, determina-se o controlador que moverá este ponto para a localização adequada. Assim, escolhendo-se as funções de transferência para o processo no ponto de operação original dadas por

$$A = G_p(j\omega) = r_p e^{j\varphi_p}, \quad (2.36)$$

do ponto de operação escolhido como

$$B = G_l(j\omega) = r_b e^{j\varphi_b}, \quad (2.37)$$

e do controlador,

$$G_c(j\omega) = r_c e^{j\varphi_c}, \quad (2.38)$$

obtém-se a função de transferência de malha aberta como

$$r_b e^{j\varphi_b} = r_p r_c e^{(j\varphi_p + j\varphi_c)}. \quad (2.39)$$

O controlador será escolhido de modo que:

$$r_c = \frac{r_b}{r_p}.$$

Para um controlador PI tem-se que (Åström e Hägglund, 1999):

$$K_c = \frac{r_b \cos(\varphi_b - \varphi_p)}{r_p}$$

e

$$T_i = \frac{1}{\omega \tan(\varphi_p - \varphi_b)}. \quad (2.40)$$

Neste caso, deve-se ter  $\varphi_p > \varphi_b$  para se obter o valor  $T_i$  positivo. O ganho  $K_c$  do controlador é calculado através da equação (2.40), esta tem um parâmetro ( $T_i$ ) a ser determinado e um grau de liberdade.

Supondo-se agora que o experimento de Ziegler-Nichols seja utilizado para determinar o ponto adequado, tem-se que  $r_p = 1/K_u$  e  $\varphi_p = 0$  e, desta forma, os parâmetros do controlador PI ficam (Åström e Hägglund, 1999):

$$K_c = K_u r_b \cos(\varphi_b)$$

e

$$T_i = \frac{T_u}{2\pi \tan(\varphi_b)}. \quad (2.41)$$

Neste caso,  $\varphi_b$  deve ser negativo de modo a se obter os parâmetros positivos.

O controlador obtido a partir das equações (2.41) desloca um ponto determinado da função de transferência de malha aberta do processo no plano  $G(s)$  para o ponto especificado por  $\varphi_b$  e  $r_b$ . O ponto a ser movido geralmente é o ponto de última ressonância, que pode ser identificado pelo método do relé (Åström e Hägglund, 1999).

### 2.4.3 Projeto e Sintonia PID via Estrutura RST

A estrutura RST (Fig. 2.10) é amplamente utilizada no meio acadêmico e industrial e possui a vantagem de resultar em um controlador de dois graus de liberdade, ou seja, é possível resolver o problema da dinâmica e rejeição à perturbação de forma desacoplada. Nesta estrutura  $R$ ,  $S$  e  $T$  são polinômios de projeto que definem o desempenho do sis-

tema (Neto, 2005). Uma outra forma de implementar a estrutura RST é a partir de um controlador conhecido (Wellstead e Zarrop, 1991) e (Salamanca, 2007).

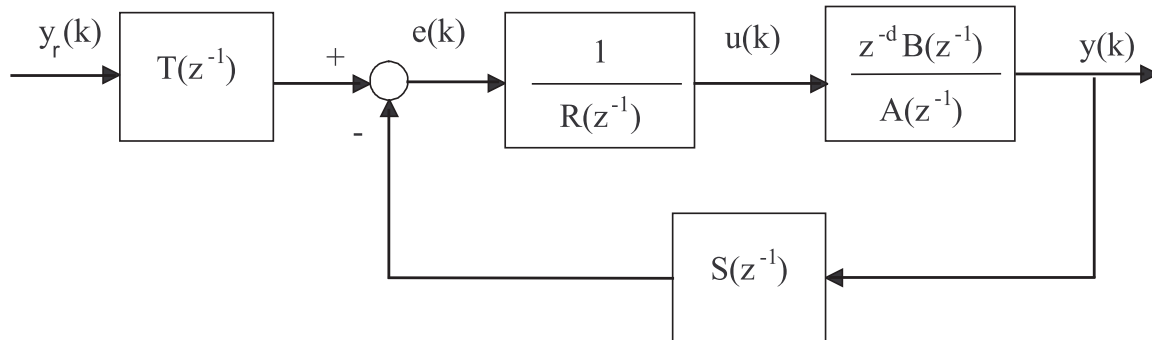


Figura 2.10: Estrutura RST.

A lei de controle da estrutura RST é representada por:

$$R(z^{-1})\Delta u(k) = T(z^{-1})y_r(k) - S(z^{-1})y(k) \quad (2.42)$$

e a função de transferência discreta em malha fechada a partir da figura 2.10 é dada por:

$$y(k) = \frac{z^{-d}B(z^{-1})T(z^{-1})}{A(z^{-1})\Delta R(z^{-1}) + z^{-d}B(z^{-1})S(z^{-1})}y_r(k) \quad (2.43)$$

onde a ordem dos polinômios é dada por:  $nR = nB + d - 1$ ,  $nS = nA$ , respectivamente, e  $T(1) = S(1)$  para garantir o rastreamento da referência e a eliminação da perturbação de carga. Na síntese RST da equação (2.42) os polinômios  $R(z^{-1})$ ,  $S(z^{-1})$  e  $T(z^{-1})$  da estrutura RST quando comparados aos parâmetros do controlador PID paralelo são dados por (Salamanca, 2007):

$$\begin{cases} R(z^{-1}) = 1 \\ S(z^{-1}) = s_0 + s_1z^{-1} + s_2z^{-2} \\ T(z^{-1}) = S(z^{-1}) \end{cases} \quad (2.44)$$

onde os coeficientes  $s_0$ ,  $s_1$  e  $s_2$  são calculados por (Salamanca, 2007):

$$s_0 = K_c \left( 1 + \frac{T_s}{T_i} + \frac{T_d}{T_s} \right); \quad s_1 = -K_c \left( 1 + 2\frac{T_d}{T_s} \right); \quad s_2 = K_c \left( \frac{T_d}{T_s} \right). \quad (2.45)$$



Estes coeficientes gerados são função dos parâmetros ( $K_p$ ,  $T_i$  e  $T_d$ ) do controlador PID convencional que agora compõem um controlador de dois graus de liberdade (2DOF). A utilização de controladores nesta estrutura possibilita a realização de estudos de estabilidade e robustez para controladores lineares (Salamanca, 2007).

# Capítulo 3

## Sistema de acionamento do Motor de Relutância Variável

### 3.1 Estrutura do Sistema de Acionamento: Hardware e Software.

O acionamento de um MRV pode ser dividido em duas partes distintas: o conversor de potência e o sistema de controle, conforme apresentado na Fig. 3.1.

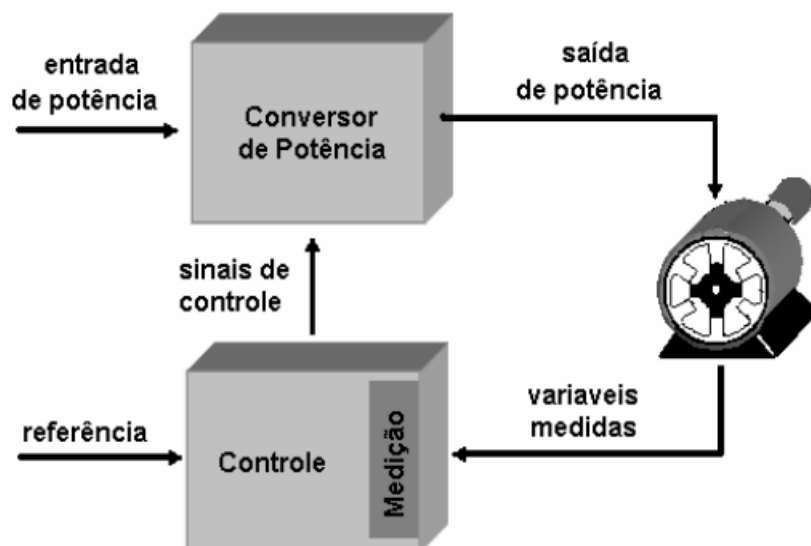


Figura 3.1: Estrutura do sistema de acionamento.

No acionamento do motor de relutância, cada fase deve ter sua corrente controlada em forma e amplitude mediante aplicação de tensão, durante o crescimento da indutância de

fase (geração de torque positivo); e ter esta mesma corrente extinta antes do período de decréscimo da indutância de fase.

O sistema de controle também é responsável pelo sincronismo de alimentação de fase do MRV. A figura 3.2 apresenta em diagramas de blocos funcionais a bancada experimental para o sistema de controle da corrente e velocidade do MRV.

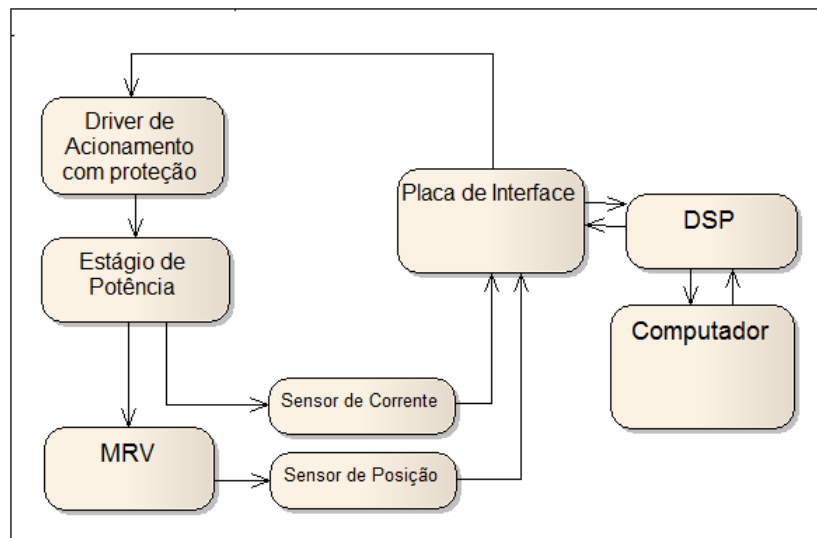


Figura 3.2: Diagrama de blocos da bancada experimental para o sistema de controle da corrente do MRV.

O sistema de acionamento utilizado foi montado no Laboratório de Automação e Robótica do Departamento de Engenharia Elétrica da UFC e é composto dos seguintes subsistemas: Máquina à relutância variável (MRV), “*encoder gray*” de 10 “*bits*”, retificador não-controlado, conversor meia ponte assimétrica trifásico (Fig. 3.4) com semicondutores de potência do tipo, “IGBTs/MOSFETs”, sensores de tensão e corrente utilizado no sistema de medição, microcomputador e processador digital de sinais (“Digital Signal Processor - DSP”).

As conexões físicas entre os dispositivos utilizados, assim como a descrição dos mesmos pode ser observada na figura 3.3 e na tabela 3.1, respectivamente.

Para o sistema de controle de acionamento do MRV, os principais periféricos utilizados são os módulos Gerenciadores de Eventos (EVA e EVB), que controlam os temporizadores e os módulos responsáveis pelo PWM, e o conversor *A/D*. Uma descrição mais completa dos respectivos dispositivos está nos Anexos A e B.

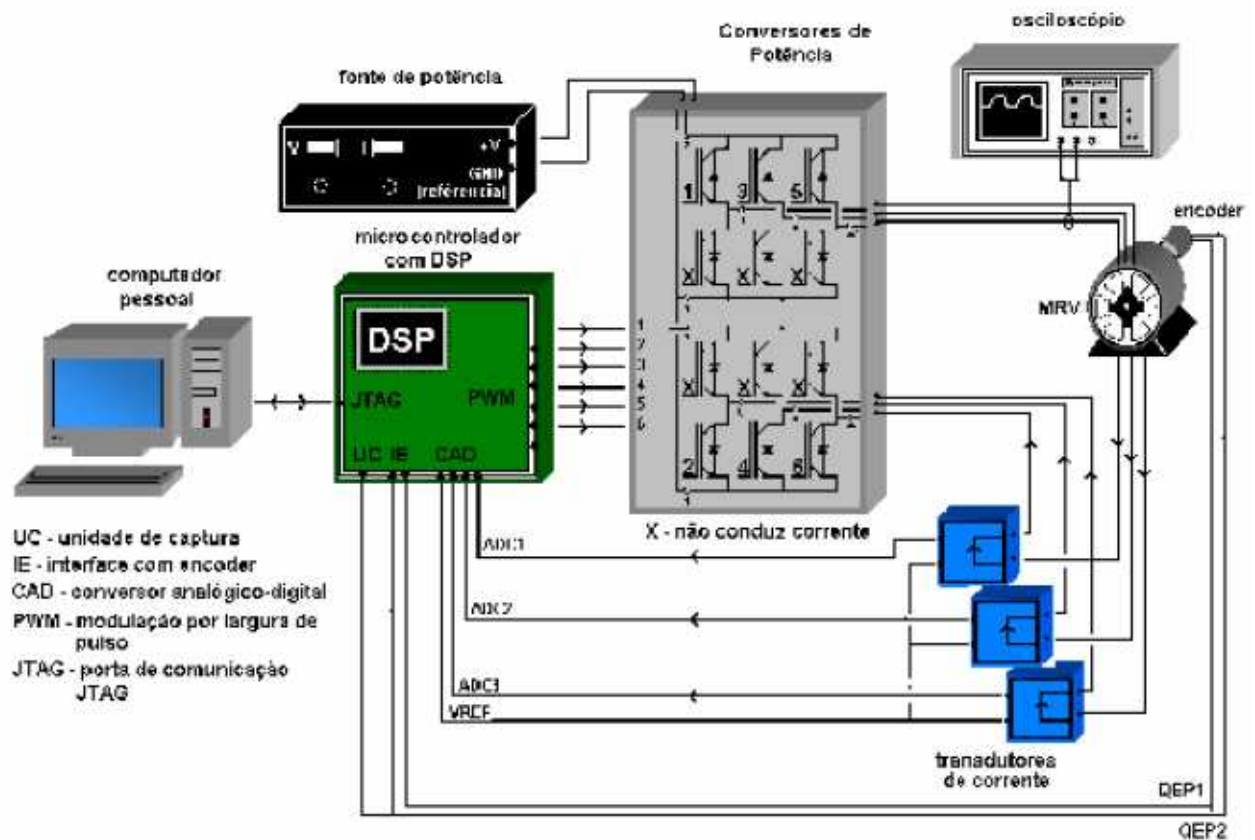


Figura 3.3: Conexão entre dispositivos no sistema de acionamento.

### 3.1.1 O Conversor de Potência

Conversores para acionamentos de máquinas de relutância variável são indispensáveis em seu funcionamento com velocidade variável. Comparativamente, as topologias desenvolvidas para máquinas de relutância variável contam com uma maior diversidade, em geral apresentam menor número de dispositivos de chaveamento, maior proteção contra curto-circuito do barramento CC e maior liberdade de acionamento entre fases. Em contrapartida nos motores de relutância variável, a frequência de chaveamento é maior que nos sistemas de acionamentos CA com o mesmo número de fases e a não uniformidade do conjugado pode exigir o uso de grandes capacitores de filtro para o barramento CC (Reis, 2008).

Uma característica do MRV é sua operação apenas com correntes unipolares, o que faz o projeto do conversor bastante simples. O conversor deve fornecer pulsos de corrente de forma sincronizada com a posição relativa do rotor. O perfil e a magnitude da corrente devem satisfazer os requisitos de torque e velocidade além de garantir a operação segura

Tabela 3.1: Dispositivos do sistema de acionamento.

Dispositivo	Modelo	Descrição
Processador digital	<i>TMS320F2812</i>	Centro de controle do sistema de acionamento
Conversor de potência	Ponte Simétrica	Condicionamento de potência ao circuito motor
Transdutor de corrente	<i>LEM LA25 – NP</i>	Geração de um sinal de corrente proporcional à corrente medida
Transdutor de tensão	<i>LEM LV25 – P</i>	Geração de um sinal de corrente proporcional à tensão medida
Encoder	<i>AB 845GM – F1GSHC1024R</i>	Gera uma saída paralela de 10 “bits gray” que representa a posição angular absoluta

da máquina e das chaves de potência do sistema de acionamento.

A escolha da topologia do conversor deve ser feita com base em informações como geometria da máquina, condições de operação e velocidade. Basicamente, conversores para acionamento de MRV enquadram-se em duas categorias: Fontes de Tensão e Fontes de Corrente (Barnes e Pollock, 1998). Conversores do tipo Fonte de Tensão possuem uma maior aplicação em operações de alta velocidade onde a implementação eficiente de um acionamento por corrente seria dificultada por conta da elevada força eletromotriz (Lima, 2005). Quando há a necessidade de um controle do conjugado de saída onde é importante a minimização da ondulação do torque em baixas velocidades a alternativa do conversor do tipo fonte de corrente torna-se mais freqüente (Le-Huy *et al.*, 1991).

Dentre as diversas configurações de conversores pode-se destacar aquelas que apresentam chaveamento dissipativo, chaveamento não dissipativo, e novas topologias com capacidade de operação regenerativa (Vukosavic e Stefanovic, 1991).

Existem muitos tipos de conversores sendo usados na indústria pelo MRV (Miller, 1993) e (Krishnan, 2001). Os conversores influenciam diretamente em muitos aspectos relevantes tais como, desempenho e flexibilidade do controle, custo, volume, capacidade de conjugado, perdas e eficiência do sistema de acionamento.

Mesmo os conversores tipo fonte de tensão serem mais indicados em acionamentos de altas velocidades, quando associados a uma controlador de corrente são uma ótima sugestão para acionamentos em baixas e médias velocidades (Lima, 2005). Tal estratégia tem sido utilizada com bastante êxito no controle de MRVs (Oliveira *et al.*, 2002), (Saraiva, 2006) e (Rômulo, 2007). Como o objetivo deste trabalho é aplicar um controle para malha

de velocidade em médias velocidades, um Conversor do tipo Fonte de Tensão acionando um MRV foi o conjunto adotado.

Um conversor do tipo meia ponte assimétrica é um dos mais comuns para o acionamento de MRVs. Sua estrutura com duas chaves por fase, mostrada na figura 3.4, possibilita uma máxima flexibilização do controle, elevada eficiência juntamente com um mínimo de componentes passivos no circuito (Miller, 1993). O conversor foi especificado para um MRV com três fases, uma geometria de doze pólos no estator e oito pólos no rotor (12/8) e com um valor de indutância máxima de  $52mH$  e de  $8mH$  para indutância mínima identificados em Reis *et al.* (2007). Seus valores nominais para tensão e corrente são respectivamente de 120 Volts e 2,5 A.

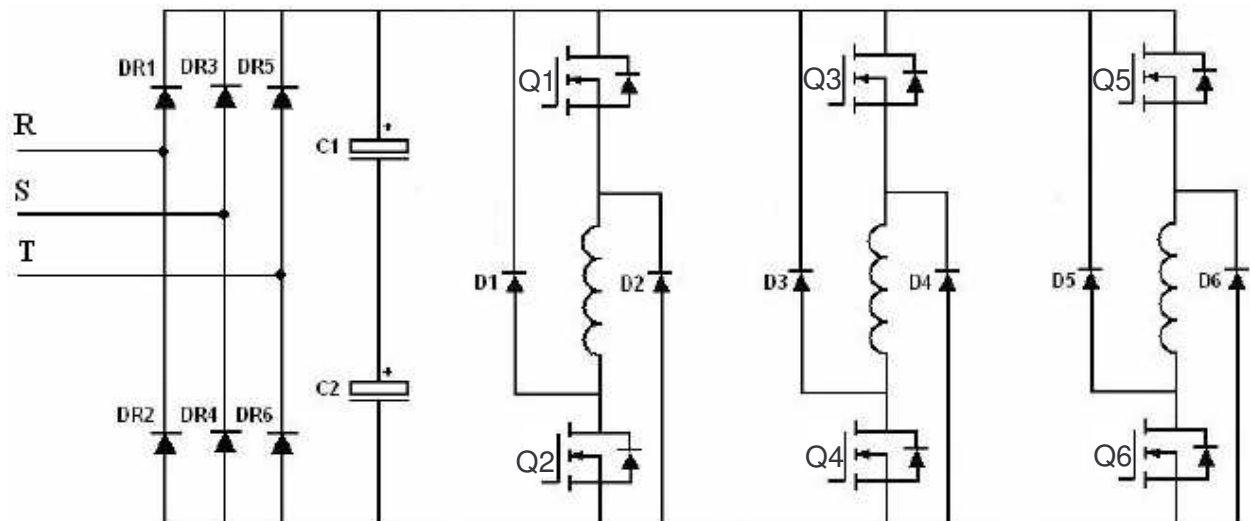


Figura 3.4: Conversor Meia Ponte Assimétrica Trifásico.

A utilização desta topologia no acionamento de MRVs é devido a capacidade de operação utilizando altas tensões e potência, bem como a possibilidade do controle independente de cada fase mesmo quando uma sobreposição de fases em condução do circuito magnético é requerida. Sua principal vantagem é a alta eficiência por conta da regeneração da energia armazenada nos enrolamentos de fase do circuito magnético durante o intervalo em que a fase encontra-se desligada.

Assumindo que as fases da máquina são alimentadas pelo circuito descrito na figura 3.4, a estratégia de chaveamento do conversor pode operar em três modos de condução distintos em cada fase. Estes modos de condução são mostrados na Fig. 3.5.

Na Fig. 3.5 (a), tem-se o primeiro modo de condução, com as duas chaves,  $Q1$  e

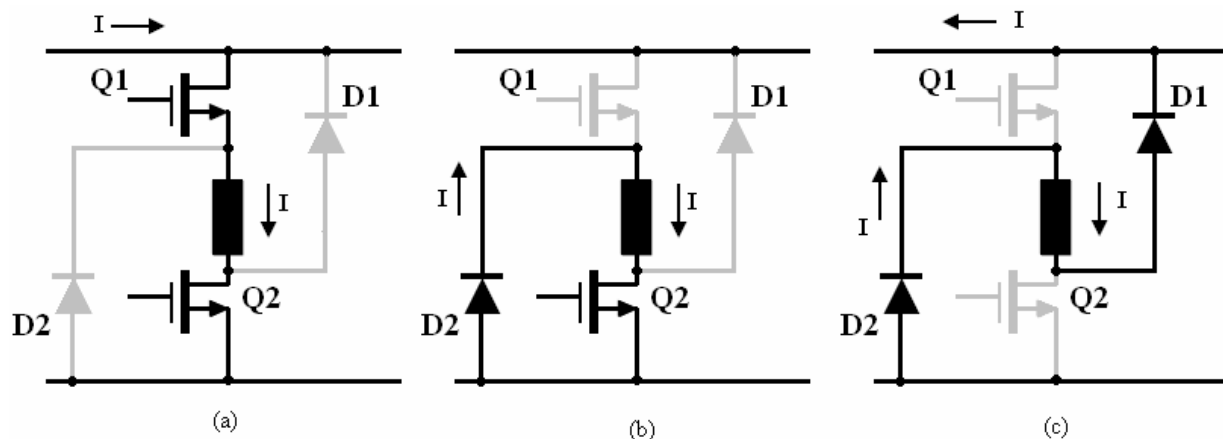


Figura 3.5: Modos de condução por fase.

$Q2$ , fechadas e com isso a tensão  $CC$  disponível encontra-se aplicada ao enrolamento da máquina. Quando  $Q1$  é desligada e  $Q2$  se mantém fechada, o diodo de “roda livre”  $D2$  garante a continuidade do circuito para a corrente do enrolamento (situação  $b$ ). Por fim, na situação  $(c)$ , quando as duas chaves são desligadas os diodos de “roda livre”,  $D1$  e  $D2$ , entram em condução. Este tipo de operação é válida tanto para geradores quanto para motores, já que a potência elétrica ( $V \cdot I$ ) pode ser tanto positiva quanto negativa. O modo de condução das chaves ainda é necessário para o estabelecimento do fluxo, que é estabelecido e zerado a cada passo. A integral da tensão no tempo durante a condução das chaves e durante o período de roda livre dos diodos deve ser aproximadamente igual, independente do funcionamento da máquina como motor ou gerador.

O conversor mostrado na figura 3.4 é composto de três braços cada um dos quais ligados em série com os enrolamentos de fase da máquina. Para as seis chaves do conversor, semicondutores de potência do tipo “MOSFET” e diodos de “roda livre” foram utilizados, onde as chaves superiores do conversor (MOSFET’s  $Q1$ ,  $Q3$  e  $Q5$ ) são acionados com uma estratégia de “chaveamento suave” a partir de um PWM de tensão em alta frequência (20kHz) onde a razão cíclica é determinado pela saída do controlador de corrente implementado e as chaves inferiores de cada braço (MOSFET’s  $Q2$ ,  $Q4$  e  $Q6$ ) são energizadas e desligadas de acordo com a posição rotórica da máquina, ou seja, de acordo com os ângulos de energização e de desligamento ( $\theta_{on}$  e  $\theta_{off}$ ) de cada fase que dependem da geometria da máquina.

A integração do controle com o conversor de potência é realizada por uma interface

de acionamento a qual provê o sinal adequado para cada chave do conversor. A figura 3.6 mostra o esquema da interface para o acionamento de cada fase.

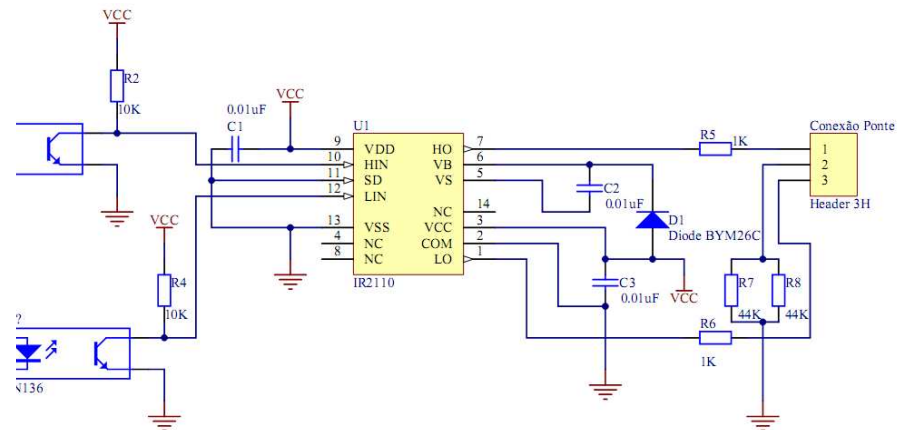


Figura 3.6: Interface de acionamento para MOSFET 's.

A interface de acionamento é composta por um optoacoplador 6n136 que garante o isolamento efetivo do sistema de controle com o conversor de potência do sistema de acionamento. O outro estágio da interface de acionamento é composto por um IR2110, um dispositivo para chaveamento de MOSFET 's e IGBT 's em alta velocidade e alta tensão, condiciona um sinal de “gate” entre 12V e 18V para o funcionamento da chave de potência juntamente de um circuito para a detecção de sobrecorrente da mesma.

A realimentação do sistema de controle de corrente é feita com transdutores instalados para fornecer amostras de corrente drenadas das principais grandezas do Conversor. Para o transdutor de corrente, optou-se pelo de sensores por efeito “Hall” do tipo LEM LA25 – NP para a menor interferência de ruídos provenientes do sistema de potência com a capacidade de efetuar leituras que vão de 0A a 25A, e para um sensor de tensão do tipo LEM LV25 – P, com a capacidade de efetuar leituras que vão de 10V a 500V. Os dois transdutores fornecem uma corrente proporcional ao sinal medido, sendo utilizado um resistor e um potenciômetro para converter esse valor em tensão ajustada para o conversor analógico/digital do controlador.

A realimentação do sistema de controle de velocidade é feita com a utilização de um sensor “gray” Allen – Bradley 845GM – F1GSHC1024R de 10 bits, que fornece uma saída paralela de 10 bits referente à posição absoluta do rotor.



### 3.1.2 O Sistema de Controle

O sistema de controle utilizado no acionamento do motor de relutância, indicado na figura 3.1 é constituído praticamente pelo DSP, o conversor de potência e o motor. No DSP estão implementadas as rotinas computacionais responsáveis pela operação do sistema de acionamento. O microcomputador mostrado na figura 3.3, além de ser utilizado para programar estas rotinas computacionais, também é utilizado na visualização instantânea das variáveis durante a operação do motor.

O controle do sistema é realizado em três etapas independentes. A primeira, corresponde ao controle de corrente, constituído basicamente de reguladores PI independente para cada fase do motor. A referência para o controle de corrente é fornecida pelo segundo controlador, controlador de velocidade. A segunda etapa é constituído por um controlador PID. Todos os controladores foram implementados digitalmente no DSP. A terceira etapa do processo de controle é a comutação de fase e é realizada pelo DSP a partir dos sinais  $QEP1$  e  $QEP2$  (figura 3.3) do encoder, referentes à posição angular instantânea.

De acordo com a figura 3.3, o sistema de controle é responsável pela recepção das variáveis medidas, execução de cálculos e rotinas regidas pela lei de controle, e geração de sinais de controle. Os sinais de controle são responsáveis pelo rastreamento da velocidade em relação à referência determinada, gerando uma corrente de referência aplicada no controlador de corrente. Sendo esta referência aplicada às fases do motor através do conversor de potência.

O controlador atua automaticamente na transição entre os modos de operação de regulação da corrente e pulso único. À medida que a corrente de referência é elevada, o ciclo de trabalho dos transistores é aumentado proporcionalmente, mantendo o chaveamento durante o angulo de condução. Quando o ciclo de trabalho chega a 100%, em função do aumento da corrente de referência, o motor deixa de operar sob o modo de regulação da corrente e passa a operar em pulso único. O processo de transição reversa também é realizado automaticamente na ocorrência de redução da corrente de referência.

O diagrama de blocos representativo do sistema é mostrado na figura 3.7.

### 3.1.3 Unidade Central de Controle

O sistema de controle para o Acionamento do MRV é constituído de duas partes básicas e distintas. O núcleo, que é o responsável por toda “inteligência” do sistema de controle e a

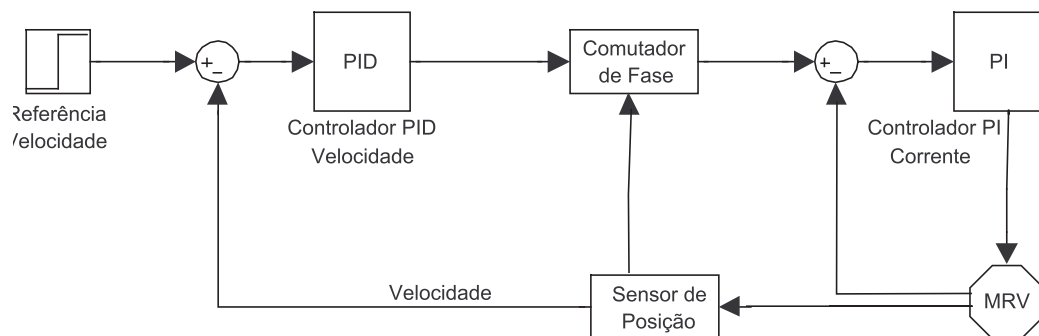


Figura 3.7: Diagrama de blocos do sistema de acionamento.

interface, responsável por compatibilizar o núcleo para a atuação com a parte do módulo de Potência do Sistema de Controle. O sistema utiliza um processador digital de sinais (DSP) *TMS320F2812* como centro de controle. O DSP é constituído basicamente de um processador, módulos de memória e periféricos integrados. Tal dispositivo possui a particularidade de processar exclusivamente sinais digitais, utilizando-se de um conversor analógico-digital, quando há a necessidade de interagir com sinais de natureza analógica. No sistema de acionamento do MRV, os sinais analógicos presentes são a tensão contínua do barramento CC dos conversores de potência, e as correntes das fases; ambos fornecidos pelos sensores de tensão e corrente, respectivamente.

Uma particularidade deste DSP que deve ser ressaltada, é o fato de este ser um processador de ponto fixo. Devido a esta característica, o seu processamento se limita à operação com dados digitais inteiros (sem representação de parte fracionária). Um artifício de programação, “formato Q”, foi utilizado para possibilitar o processamento e operação de valores fracionários garantindo a precisão necessária. Este artifício consiste em atribuir (em nível de *software* exclusivamente) um ponto flutuante à variável no programa.

O sistema de controle utiliza-se de temporizadores, denominados de *Timer3* e *Timer2* (Apêndice A), do DSP para garantir a frequência de amostragem das três correntes de fase da máquina em  $40\mu s$  e a frequência de amostragem na malha de velocidade da máquina em  $120\mu s$ . Para atingir esse período, inicialmente é configurado o multiplicador (“prescales”) do “clock” da unidade central de processamento (CPU), que serve como sinal de entrada para a frequência do contador. Em seguida o contador é configurado para trabalhar em modo de contagem crescente contínua. Neste modo de operação o contador

é incrementado em sincronismo com o sinal de “clock” da CPU, então quando este atinge um valor previamente configurado no registrador de período, uma rotina de interrupção é ativada e a contagem é reiniciada. Para a obtenção desta frequência de amostragem em exatos  $40\mu s$ , por exemplo, a seguinte equação (3.1) é utilizada:

$$T_s = T_{cin} \times N, \quad (3.1)$$

onde  $T_s$  é o período de amostragem desejado,  $T_{cin}$ , é o período do “clock” de entrada e  $N$  é o valor de contagem configurado no registrador de período do contador. Com isto calculamos a necessidade de 3000 ciclos de clock de  $75MHz$ , para que a frequência de amostragem seja  $40\mu s$ .

O controlador atua no MRV a partir de um PWM que utiliza *Timer1* (Apêndice A) do processador para gerar a portadora do sinal. Três unidades de comparação do gerenciador de “Eventos A” (*CMPR1*, *CMPR2* e *CMPR3*) do processador definem os ciclos de trabalho modulados pelos controladores implementados para a malha de corrente. Para obter uma portadora de  $20kHz$  com um “clock” de CPU de  $75MHz$ , utiliza-se a equação (3.1) para calcular o valor necessário do registrador de período do *Timer1*.

A integração do sinal de realimentação analógico medido com o controle é realizado através do módulo ADC (Apêndice A) configurado como um único módulo de 16 canais. O gerenciador de eventos A através do contador *Timer3* é o responsável pela sincronização do período de amostragem como visto anteriormente. Assim um “início de conversão” (SOC) dispara o funcionamento do ADC onde são realizadas as 8 conversões. O conversor de potência disponibiliza três sinais de corrente (correspondentes as três fases do MRV) e um sinal de tensão (correspondente a tensão do barramento CC) como mostrado na figura 3.3. É realizado então um processo de “*over sampling*” onde cada medida é realizada 2 vezes em canais diferentes para diminuir a interferência de ruídos inerentes do circuito. Terminado este processo os resultados das conversões são armazenados em registradores de 16 bits onde destes apenas os 12 bits mais significativos representam a leitura específica da tensão amostrada. É realizado então um deslocamento de 4 bits para a direita para eliminar os bits descartáveis do registrador para que o valor efetivo da conversão possa ser utilizado pelo controlador de corrente implementado.

A integração do sensor de posição responsável pelos sinais de sincronismo de fase e entrada da malha de velocidade, é realizado através da *PORT B*, onde é realizada à

leitura de toda a porta paralelamente. Uma máscara de 10 *bits* é aplicada descartando os outros *bits*. A conversão do valor “gray” proveniente do encoder para um valor entre 0° e 360° é realizada em *software*. Por se tratar de um sensor absoluto, na inicialização do sistema, o motor é travado em uma fase conhecida e então disparado um sinal digital pela *PORTB.15* responsável por indicar a posição “zero”.

O *TMS320F2812* é um circuito integrado que faz uso da tecnologia *CMOS* para 3V. Para possibilitar que o DSP possa trabalhar com os circuitos de acionamento das chaves e com os circuitos transdutores de corrente e tensão do MRV, foi construída uma placa de interface com o propósito de compatibilizar os níveis de tensão e corrente utilizados pelo DSP com os da tecnologia TTL que foi utilizada para padronizar o sistema de acionamento. Três tipos básicos de circuitos construídos possibilitam a compatibilização dos módulos.

“*Buffers*” do tipo *LS7407*, de coletor aberto, compatibilizam os pinos de saída do DSP (PWM e I/O's de uso geral) com o sistema de gatilhamento das chaves como mostra a figura 3.8.

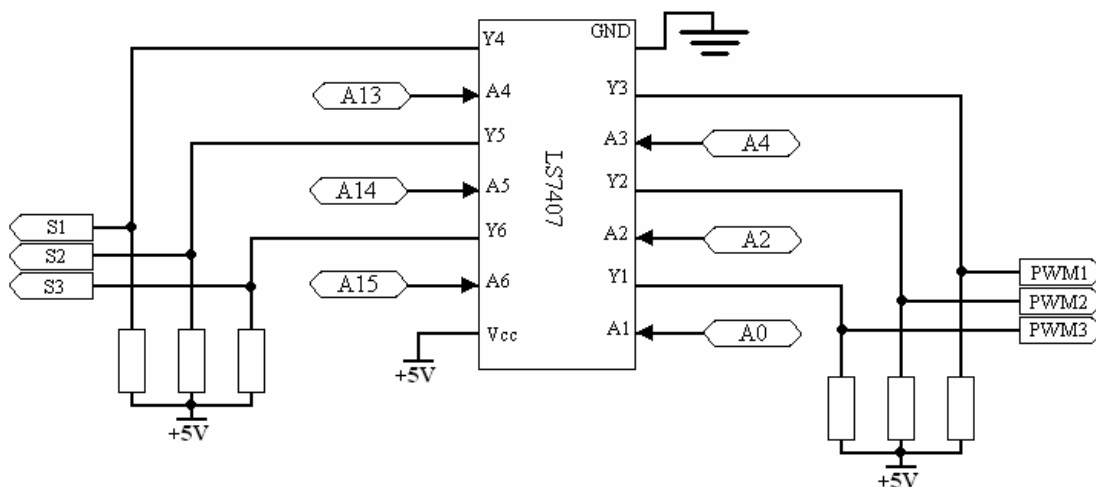


Figura 3.8: Esquema de ligação para o Buffer de saída.

Para o ajuste das tensões capturadas através do módulo ADC, divisores de tensão são ajustados como mostra a figura 3.9, de forma que os máximos valores de saída dos transdutores não ofereçam risco à integridade da DSP.

O sensor de posição *845GM – F1GSHC1024R* citado anteriormente possui suas saídas em coletor aberto facilitando a compatibilização com os pinos de I/O do DSP através de resistores “PULL-UP” associados a uma fonte de alimentação de 3V, cujo esquema de ligação é mostrado na figura 3.10.

O *software* de controle do DSP é constituído de uma série de algoritmos implementados de forma que os controladores da malha de corrente, malha de velocidade e as sincronizações dos ângulos de disparo baseadas na posição rotórica ocorram sem produzir interferência entre seus resultados. O algoritmo de controle utiliza linguagem “C” compatível com processadores digitais da “Texas Instruments” por meio do ambiente de programação do “*Code Composer Studio - CCS*”.

O bloco de configuração trata da inicialização do controlador para que todas as tare-

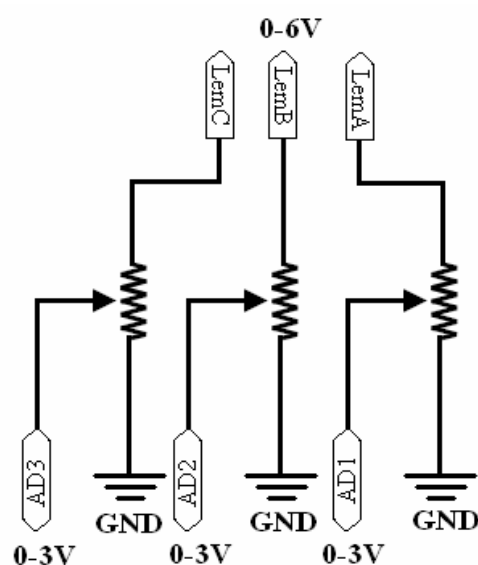


Figura 3.9: Divisor de tensão para o conversor *A/D*.

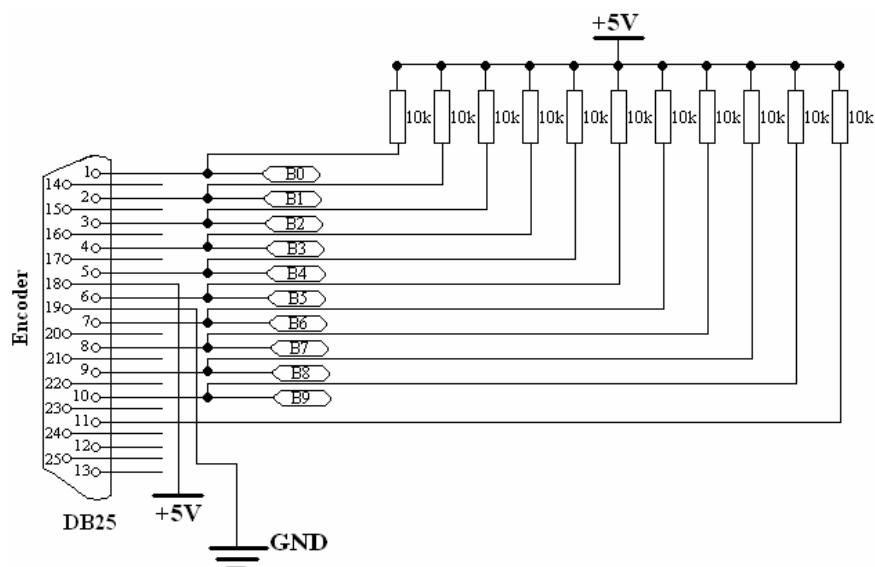


Figura 3.10: Esquema de ligação do encoder 845GM – F1GSHC1024R.

fas da rotina principal possam ser executadas. Consiste basicamente na habilitação das interrupções temporizadas que serão utilizadas (“*Sample Time*” da malha de corrente em  $25kHz$ , “*Sample Time*” da malha de velocidade em  $8.3kHz$  e de atualização do PWM em  $20kHz$ ) na configuração dos periféricos de controle (conversor *A/D* cascadeado em modo seqüencial e unidade de PWM com geração assimétrica), na configuração da porta para leitura do encoder, e por fim na inicialização de algumas variáveis e parâmetros necessários para o restante do programa. Fig. 3.11.

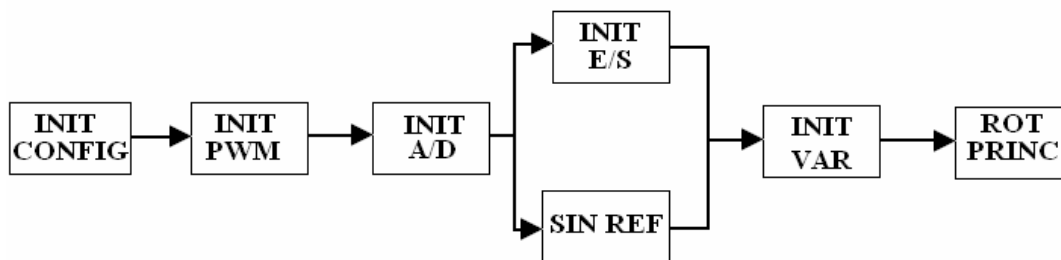


Figura 3.11: Processo de configuração do algoritmo de controle.

Ao atingir a rotina principal, o algoritmo da malha de velocidade gera uma saída (corrente de referência) que servirá como entrada para a malha de corrente, em seguida o algoritmo principal mantém o funcionamento da máquina verificando constantemente qual fase deve ser energizada de acordo com a posição do rotor. Uma vez selecionada qual fase deve permanecer ativa, a lei de controle, que depende do tipo do controlador de corrente implementado, determina com base nas correntes verificadas através do conversor *A/D* durante a interrupção de amostragem, qual será o próximo ciclo de trabalho que deverá ser atualizado durante a interrupção do PWM como mostra a figura 3.12.

### 3.1.4 Observações Finais

O acionamento do MRV pode ser dividido em duas partes principais, o conversor de potência e o controle. Ao contrário do acionamento em máquinas CA, o conversor é indispensável para o seu funcionamento. Em geral estes apresentam uma menor quantidade de elementos dissipativos, maior proteção contra curto circuito no barramento CC, maior liberdade no acionamento entre as chaves, operação apenas com correntes unipolares, frequência de chaveamento maior que em sistemas de acionamento CA com o mesmo número de fase e exige o uso de grandes capacitores de filtro no barramento CC.

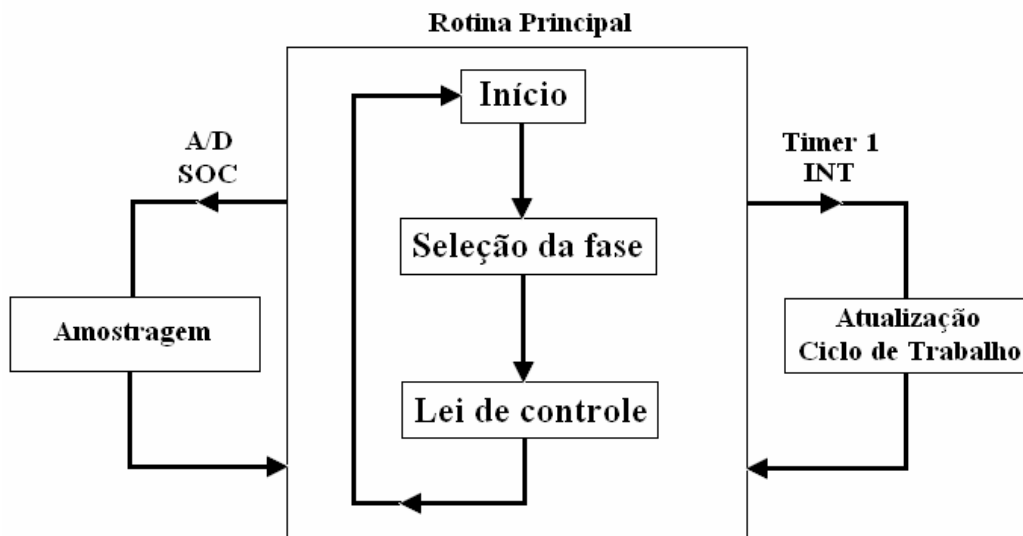


Figura 3.12: Rotina principal do *software*.

Existem muitas topologias de conversores que podem ser aplicadas ao MRV, sendo conversores do tipo meia ponte assimétrica um dos mais comuns e o utilizado neste trabalho, projetado para um MRV (12/8), valor nominal de tensão de 120 Volts e corrente de 2,5A, com valores de indutância máxima e mínima de  $52mH$  e de  $8mH$  respectivamente.

A utilização do *IR2110* mostrou-se ideal para realizar o acionamento dos “MOS-FET’s”, por reduziu a quantidade componentes na placa de acionamento diminuindo assim o ruído e possibilitou uma maior proteção do sistema de controle.

O sensor de corrente por efeito “HALL” utilizado, mostrou-se robusto. O reposicionamento dos componentes da placa de do conversor, diminuindo a distancia entre o sensor de corrente e a placa do DSP, possibilitou a redução de ruídos na leitura do sinal referente as variáveis medidas.

A placa de desenvolvimento utilizada possui um DSP com um alto poder de processamento, com conversões *A/D* rápida. O que possibilitou um ótimo desempenho do *software* de controle, podendo atender a todos os tempos críticos. Porém, possui memória RAM do DSP limitada em  $36Kb$ , sendo assim necessário a utilização da memória “*flash*” do DSP para que fosse possível implementar todo o *software*. Para não prejudicar o desempenho do sistema, na inicialização, as funções críticas que exigem muito processamento são copiadas para memória RAM, possibilitando uma execução mais rápida da ação de controle desejada.

# Capítulo 4

## Resultados Experimentais

### 4.1 Introdução

Este capítulo tem como objetivo apresentar os procedimentos experimentais realizados no sistema de acionamento com o motor de relutância variável (MRV). Os experimentos tem como objetivo avaliar o desempenho do controlador de velocidade para o sistema de acionamento do MRV que se encontra no laboratório GPAR da UFC com os seguintes requisitos: O controlador de corrente do sistema em malha fechada deve conferir uma resposta ao degrau de referência com erro nulo em regime permanente, máximo de sobre-sinal limitado a 10% e tempo de acomodação menor do que  $1ms$ ; O controlador de velocidade do sistema em malha fechada deve responder ao degrau de referência com erro nulo em regime permanente, máximo de sobre-sinal limitado a 10% e tempo de acomodação menor do que  $0,5s$ .

Na implementação do sistema utilizou-se um DSP da Texas *TMS320F2812* para implementar as malhas de controle ao mesmo tempo que armazena os dados coletado de corrente, tensão e parâmetros do controlador em memória e envia ao computador o qual está conectado. O *software* “Code Composer Studio”, é responsável por realizar a leitura dos valores e apresentar também a resposta em forma gráfica.

### 4.2 Identificação da malha de corrente

O método do relé *setpoint* descrito na seção 2.3.2 foi aplicado ao sistema de acionamento proposto nesse trabalho, sob condição de carga constante, com parâmetros nominais de



tensão e corrente (120V e 2.5A), velocidade fixa em 220rpm, um controlador PI pré-ajustado com os valores em  $K_c = 100$  e  $T_i = 2.222 \cdot 10^{-4}$  e com os parâmetros do relé ajustados com  $d = 1.5$  e  $\varepsilon$  variando de 0.05 a 0.35. A tabela 4.1 mostra os valores de amplitude e período de oscilação da corrente para os valores descritos.

Tabela 4.1: Resultados de amplitude e período para relé setpoint para malha de corrente

Epson	Amplitude	Período(ms)
0.05	0.240	1.4
0.08	0.292	1.6
0.10	0.298	1.7
0.13	0.298	1.7
0.15	0.340	1.8
0.18	0.345	2.0
0.20	0.350	2.1
0.25	0.380	2.3
0.30	0.500	2.6
0.35	0.500	2.8

As figuras 4.1 e 4.2 apresentam os resultados destes experimentos para os valores de  $\varepsilon = 0.05$  e  $\varepsilon = 0.35$ .

A figura 4.3 apresenta o traçado dos pontos de identificação pelo método do relé apresentados na tabela 4.1 no diagrama de Nyquist.

Com os resultados gerados, aplicando-se as equações 2.17, 2.23, 2.24, pode-se chegar a equação baixo:

$$G_p(s) = \frac{89e^{-2 \cdot 10^{-4}s}}{0.1092s + 1} \quad (4.1)$$

A Fig. 4.4 mostra o diagrama de Nyquist considerando a equação 4.1.

Conforme pode ser observado nas Fig. 4.1 e 4.2, o sistema de malha de corrente com o controlador PI pré-ajustado resulta em um sistema estável em malha fechada e de 1<sup>a</sup> ordem conforme figura 4.4.

Conforme pode-se verificar a partir do diagrama de Nyquist da fig. 4.3 bons resultados são obtidos para a estimação da função de transferência em frequência da malha de corrente.

Uma vez identificada a função de transferência para a malha de corrente do MRV, é possível realizar a sintonia dos controladores de corrente propostos neste trabalho.

Para a sintonia dos parâmetros PI foi foram utilizado os métodos de Ziegler-Nichols

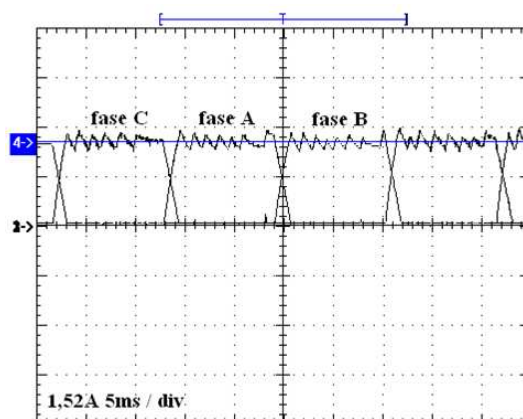


Figura 4.1: Forma de onda de corrente do método do relé setpoint para  $d = 1.5$  e  $\varepsilon = 0.05$

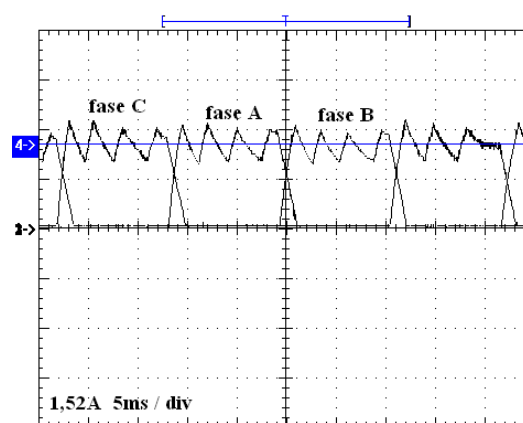


Figura 4.2: Forma de onda de corrente do método do relé setpoint para  $d = 1.5$  e  $\varepsilon = 0.35$

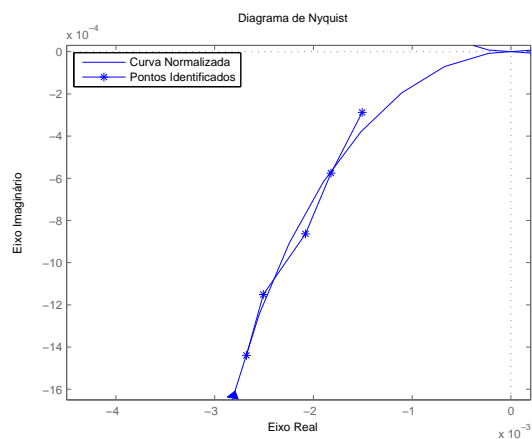


Figura 4.3: Traçado dos pontos de identificação pelo método do relé no diagrama de Nyquist.

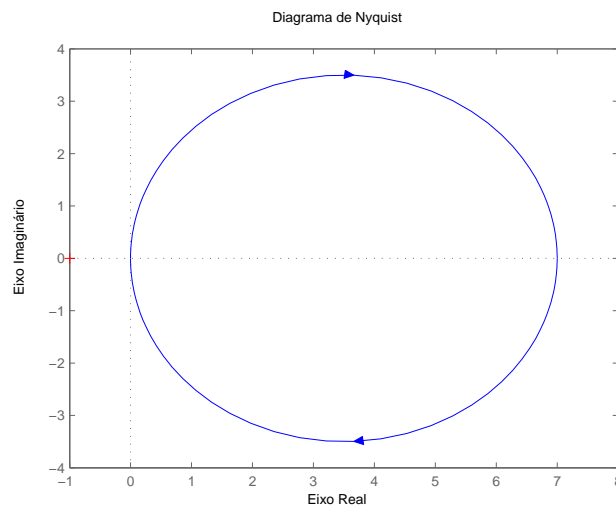


Figura 4.4: Diagrama de Nyquist da função identificada

e de Ziegler-Nichols modificado. Ambos os parâmetros de sintonia foram aplicados ao sistema real. Conforme descrito na seção 4.3.

### 4.3 Sintonia dos Controladores para Malha de Corrente

O ajuste dos parâmetros do controlador PI pelo método de Ziegler-Nichols utiliza o ponto crítico  $(\omega_u, K_u)$ , como citado na seção 2.4.1. O ponto crítico é obtido a partir função de transferência 4.1. Aplicando-se na tabela 2.1 os valores de  $\omega_u$  e  $K_u$  encontrados, chega-se aos parâmetros do controlador correspondem com  $K_c = 5.49$  e  $T_i = 6.7 \cdot 10^{-4}$ . Estes valores foram aplicados ao sistema de controle de corrente implementado, obtendo-se as Figs. 4.5 e 4.6, correspondentes a resposta de corrente de fase e a variável de controle respectivamente.

Analisando a Fig. 4.5 é possível notar que o controlador com os parâmetros ajustado pelo método Ziegler-Nichols fez com que o sistema alcançasse a referência desejada de 2.5A com um sobre-sinal. Analisando a variável de controle referente a este controlador, Fig. 4.6, é possível perceber que o controlador apresenta um comportamento na saída do sistema dentro das especificações desejadas.

Como apresentado na seção 2.4.2, o ajuste por Ziegler-Nichols modificado é realizado deslocando-se um ponto no diagrama de Nyquist com um controlador PI. O ajuste do

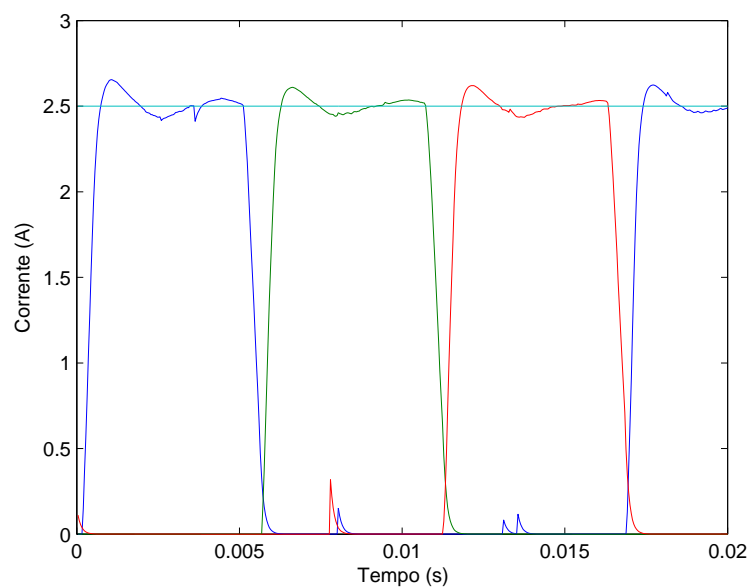


Figura 4.5: Resposta do controlador PI Ziegler-Nichols para  $i = 2.5A$ ,  $\omega = 220rpm$ ,  $K_c = 5.49$  e  $T_i = 6.7 \cdot 10^{-4}$ .

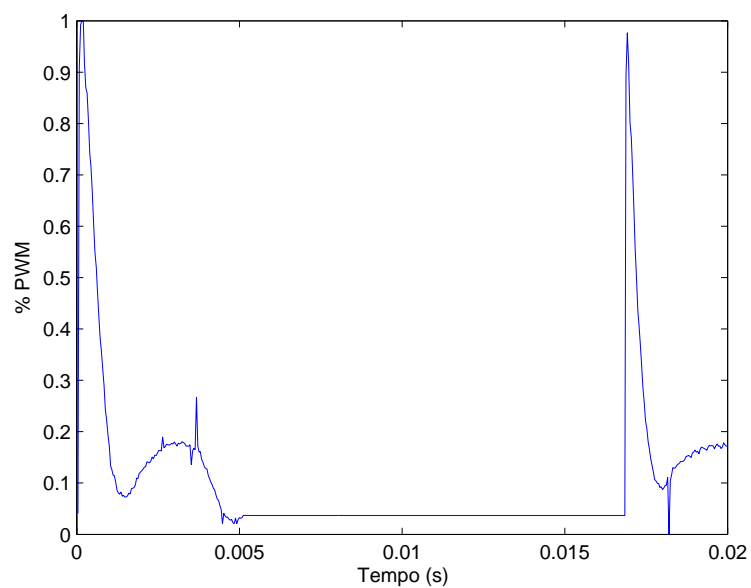


Figura 4.6: Variável de controle PI Ziegler-Nichols para  $i = 2.5A$ ,  $\omega = 220rpm$ ,  $K_c = 5.49$  e  $T_i = 6.7 \cdot 10^{-4}$ .

controlador foi projetado para mover o ponto utilizado no ajuste por Ziegler-Nichols para  $r_b = 0.26$  e  $\theta_b = 46$ , por ser um ponto sugerido pela literatura (Åström e Hägglund, 1995). Aplicando-se então a equação 2.41, chega-se aos seguintes valores para o controlador PI:  $K_c = 172$ ,  $T_i = 1.23 \cdot 10^{-4}$ .

A resposta de corrente e a variável de controle para esse ajuste podem ser vistas nas Figs. 4.7 e 4.8 respectivamente.

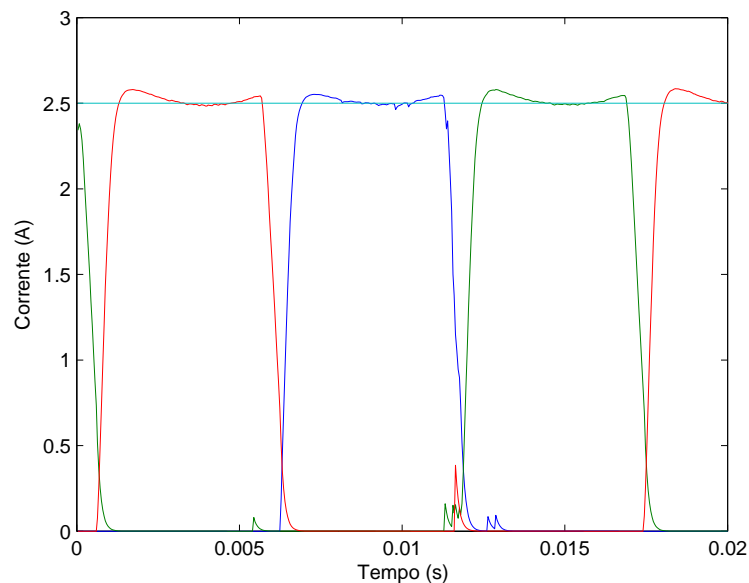


Figura 4.7: Resposta controlador PI Ziegler-Nichols modificado para  $i = 2.5A$ ,  $\omega = 220rpm$ ,  $K_c = 172$ ,  $T_i = 1.23 \cdot 10^{-4}$ .

Analisando a Fig. 4.7 é possível notar que o controlador com os parâmetros ajustados pelo método de Ziegler-Nichols modificado fez com que o sistema alcançasse a referência desejada de  $2.5A$  com um pequeno sobre-sinal, menor ainda que o do controlador anterior. É possível notar também que este controlador praticamente não apresentou a oscilação mostrada no controlador com os parâmetros ajustados pelo método de Ziegler-Nichols. Analisando a variável de controle referente a este controlador, Fig. 4.8, é possível perceber que logo após o a seleção da nova fase o controlador aplica uma saída para o sistema de aproximadamente 87% da valor máximo de tensão, não saturando assim a variável de controle. Logo em seguida controlador aplica uma saída para o sistema com uma variação média de 5%.

O controlador com parâmetros ajustados pelo método de Ziegler-Nichols modificado, além de apresentar um comportamento dentro das especificações desejadas, mostro-se melhor quando comparado ao controlador ajustado pelo método de Ziegler-Nichols, tanto

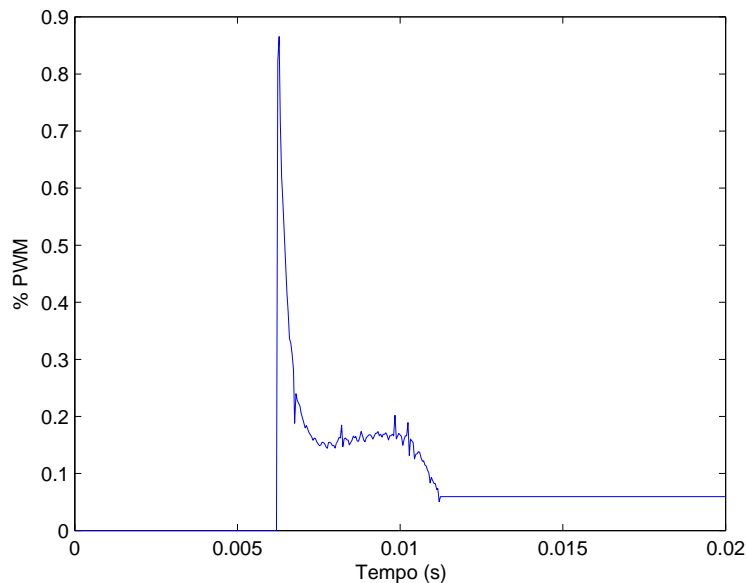


Figura 4.8: Variável de controle PI Ziegler-Nichols modificado para  $i = 2.5A$ ,  $\omega = 220rpm$ ,  $K_c = 172$ ,  $T_i = 1.23 \cdot 10^{-4}$ .

na resposta do sistema quanto no comportamento da variável de controle.

## 4.4 Experimentos para identificação da malha de velocidade

Para identificação da malha de velocidade é necessário que a malha de corrente esteja sintonizada e em funcionamento. A partir dos resultados apresentados na seção 4.3 deste capítulo, foi definida a utilização dos parâmetros do controlador PI por Ziegler-Nichols modificado devido ao desempenho superior deste controlador quanto ao tempo de resposta, sobres-sinal, erro quadrático e variável de controle.

O método dos mínimos quadrados foi utilizado para a identificação experimental da malha de velocidade de acordo com o descrito na seção 2.3.3 do capítulo 2.

Para a malha de velocidade utilizou-se um modelo com dois pólos e um zero expressa na equação 4.2.

$$G_{p(v)}(z^{-1}) = \frac{b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + 1} \quad (4.2)$$

Esta estrutura para malha de velocidade foi adotada por mostrar-se experimentalmente mais adequada para representar a dinâmica da malha.

A Fig. 4.9 ilustra o diagrama de blocos da malha de velocidade com o estimador dos mínimos quadrados.

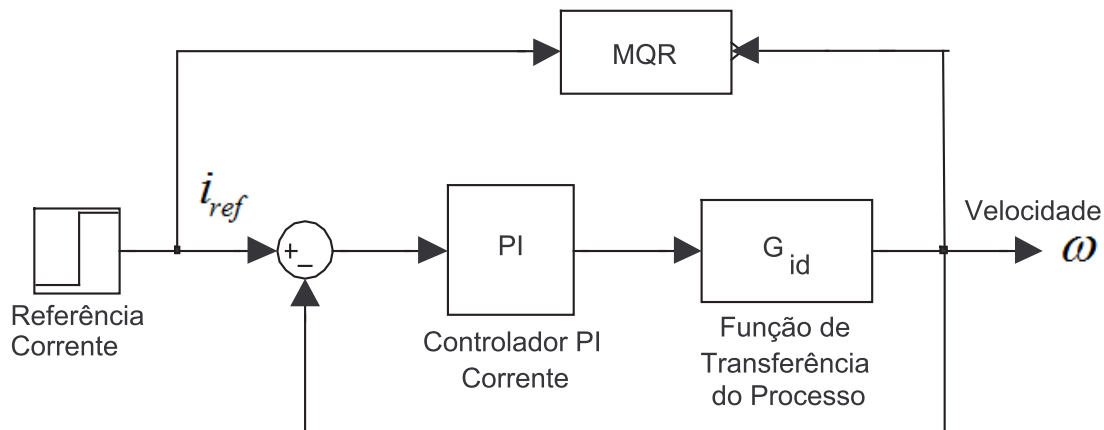


Figura 4.9: Diagrama de blocos para simulação do método de identificação da malha de velocidade.

Pode-se observar da Fig. 4.9 que para a implementação da malha de velocidade, deve-se estimar a influência da malha de corrente na velocidade do motor. Desta forma o vetor de regressores considerando o modelo adotado resulta em:

$$\phi(t) = \begin{bmatrix} -\omega(t-1) & -\omega(t-2) & i_{ref}(t) & i_{ref}(t-1) \end{bmatrix}, \quad (4.3)$$

e o vetor de parâmetros estimados torna-se

$$\hat{\theta} = \begin{bmatrix} a_1 & a_2 & b_0 & b_1 \end{bmatrix} \quad (4.4)$$

A equação à diferença para o sistema é dada por:

$$\omega(t) = -a_1\omega(t-1) - a_2\omega(t-2) + b_0i_{ref}(t) + b_1i_{ref}(t-1). \quad (4.5)$$

Foram utilizados quatro entradas do tipo degrau na malha de corrente, dados por 3.5, 4.5, 5.5 e 6.5A. A utilização do degrau em bateladas na estimação pelo método dos mínimos quadrados para este caso foi possível devido ao ruído branco intrínseco do sistema. A saída de velocidade para cada degrau de corrente é ilustrada na Fig. 4.10. Estes dados foram aplicados de formar “*off-line*” no algoritmo implementado dos mínimos quadrados recursivos.

A utilização de quatro referências em pontos de operações distintos possibilitou uma

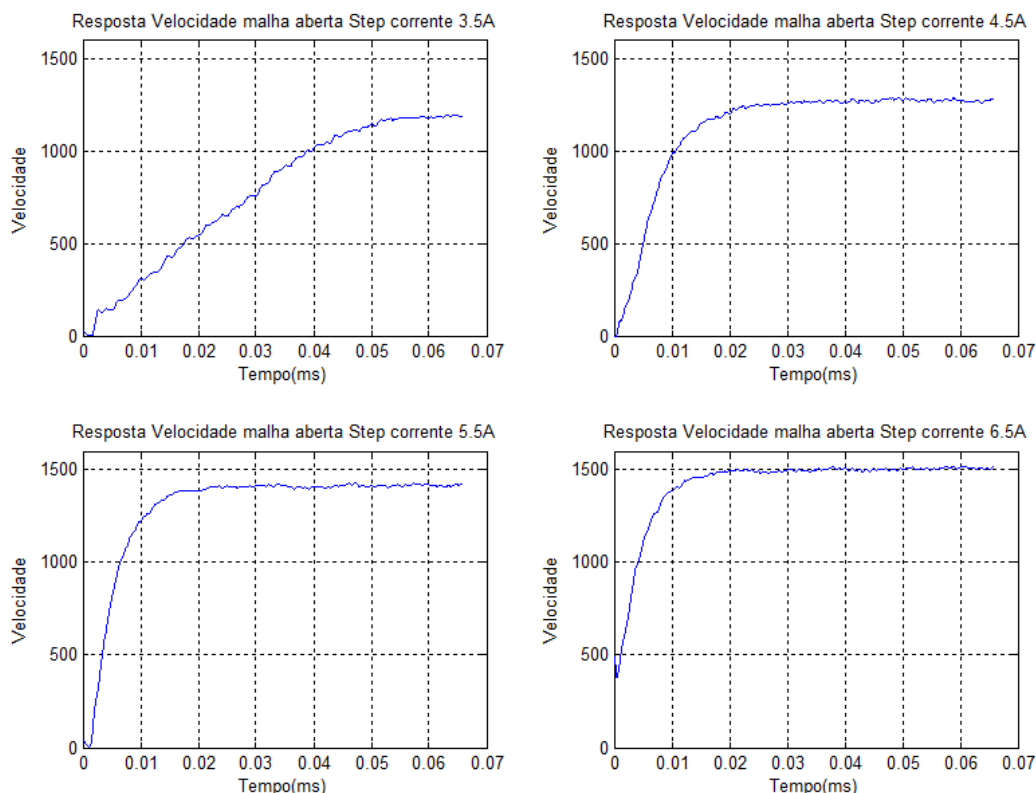


Figura 4.10: Resposta a um degrau em malha aberta de velocidade do MRV.

robustez da função de transferência estimada, pois esta passou a conter informações da dinâmica do sistema em mais de um ponto de operação. Na Fig. 4.10 é possível verificar ainda que a malha de velocidade possui uma dinâmica diferente em cada um dos pontos identificado, ou seja: É um sistema não-linear.

Assim, supondo-se uma estrutura para o modelo da malha de velocidade dada a partir da equação 2.25, com  $n_b = 1$  e  $n_a = 2$ , ou seja, dois pólos e um zero, tem-se:

$$G_{p(v)}(z^{-1}) = \frac{b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + 1}, \quad (4.6)$$

que será utilizada no projeto de controlador PID de velocidade para o MRV.

A aplicação do método citado resulta na função de transferência da malha de velocidade descrita pela equação 4.7. A escolha de uma equação com dois pólos e um zero foi realizada de forma empírica. Experimentos foram realizados com o motor em questão utilizando equações de primeiro, segundo e terceiro grau, com um e dois pólos. Sendo o modelo citado o que obteve uma melhor resposta quando comparado com os dados



coletados para validação do modelo.

$$G_{p(v)}(z^{-1}) = \frac{-0.0105z^{-1} + 0.0818}{1 + 0.4108z^{-1} + 0.5600z^{-2}}. \quad (4.7)$$

A figura 4.11 mostra a convergência dos parâmetros para os valores do modelo.

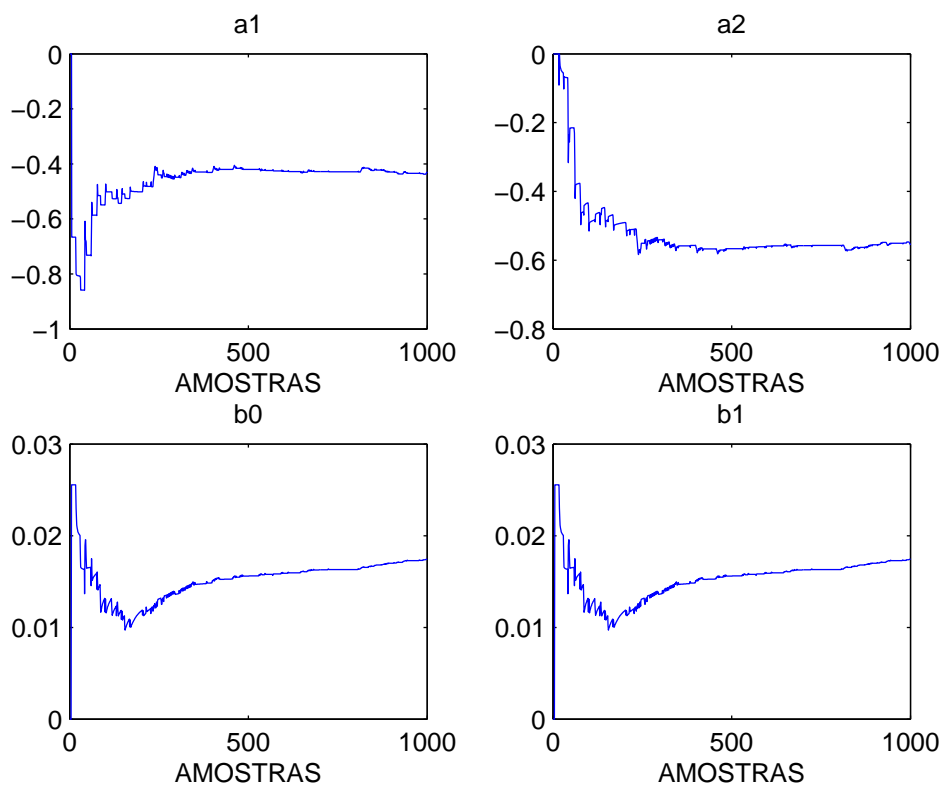


Figura 4.11: Variação dos parâmetros identificados para função de transferência de velocidade do MRV.

Uma vez identificada a função de transferência para a malha de velocidade do MRV, realiza-se o ajuste dos controladores de velocidade propostos neste trabalho. Inicialmente foi realizada a sintonia para um controlador PID tradicional pelo método de Ziegler-Nichols, e em seguida utilizou-se um PID na estrutura RST, conforme apresentado na seção 2.4. Ambos os controladores foram aplicados ao sistema real. Os resultados foram coletados e analisados de modo a selecionar o melhor, prosseguindo os experimentos, demonstrados a seguir.

## 4.5 Sintonia PID para malha de Velocidade

A sintonia dos parâmetros do controlador para a malha de velocidade foi realizada utilizando-se a teoria descrita nas seções 2.4.1 e 2.4.3.

Inicialmente foi traçado o diagrama de Nyquist que pode ser visualizado na Fig. 4.12, obtendo-se o ponto crítico  $(\omega_u, K_u)$ . Aplicando-se a tabela 2.1, chega-se aos valores das constantes  $K_c$ ,  $T_i$  e  $T_d$  do controlador PID ajustado pelo método de Ziegler-Nichols. Os valores dos parâmetros são apresentados na tabela 4.2.

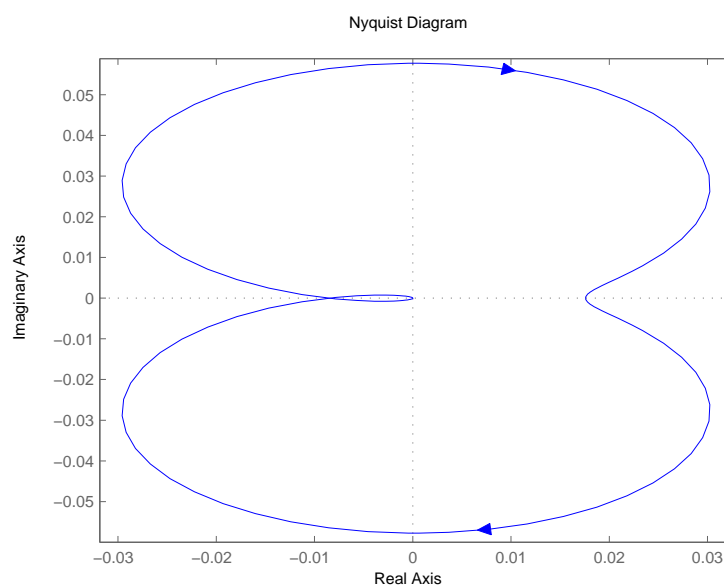


Figura 4.12: Diagrama de Nyquist da função de transferência de velocidade do MRV.

Tabela 4.2: Parâmetros do controlador PID Velocidade

Parâmetro	Valor
$K_c$	2.787
$T_i$	$1.10 \cdot 10^{-3}$
$T_d$	$2.52 \cdot 10^{-5}$

As Figs. 4.13 e 4.14 apresentam a resposta do controlador PID de velocidade do MRV com os parâmetros ajustados pelo método de Ziegler-Nichols, e a variável de controle (corrente de referência aplicada), respectivamente para um “setpoint” de  $650rpm$ .

A partir da análise da Fig. 4.13 é possível verificar que o sistema controlado atingiu a referência especificada de  $650rpm$  com um pequeno sobre-sinal, mas dentro das especificações desejadas com relação ao percentual de sobre-sinal e tempo de acomodação.

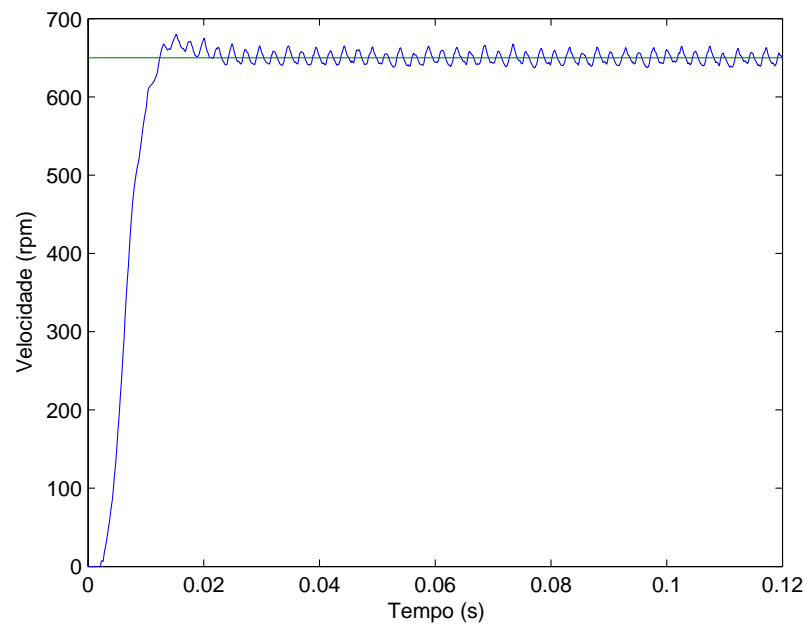


Figura 4.13: Resposta do sistema ao controlador PID aplicado a malha de velocidade do MRV com referência de  $650rpm$

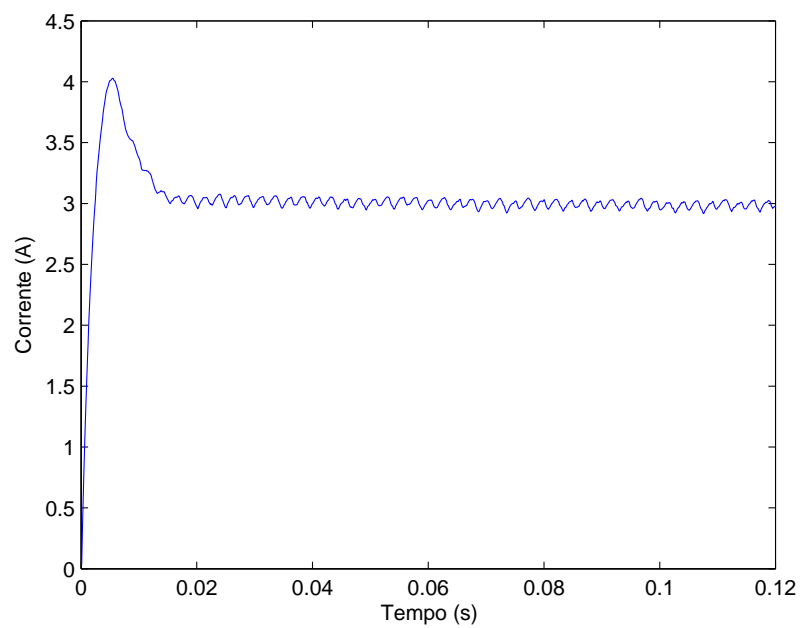


Figura 4.14: Variável de controle para o controlador PID aplicado a malha de velocidade do MRV com referência de  $650rpm$

O sistema apresentou uma resposta oscilatória em regime permanente devido ao erro do sensor de posição. Afim de reduzir esta oscilação, foi proposta a aplicação de uma outra estrutura de controladores, apresentada na seção 4.5.1.

Analisando a figura 4.14, referente a variável de controle do sistema para o controlador PID com parâmetros ajustados pelo método de Ziegler-Nichols, nota-se que o controlador aplica um pico de corrente e acomodou logo em seguida com uma oscilação mínima, também devido ao erro do sensor de posição.

### 4.5.1 Sintonia do PID com Estrutura RST para malha de Velocidade

A fim de se analisar as melhorias possíveis a partir dos parâmetros do controlador PID sintonizados na seção os valores 4.5, aplica-se a equação (2.45). Podendo chegar aos parâmetros do controlador na estrutura RST apresentados na tabela 4.3.

Tabela 4.3: Parâmetros do controlador PID aplicado a malha velocidade na estrutura RST

Parâmetro	Valor
$S_0$	12.355
$S_1$	-2.826
$S_2$	0.0195

Os novos pólos alocados pela estrutura RST podem ser obtidos para análise utilizando-se as equações (2.43) e (4.7). A nova função de transferência em malha fechada é dada pela equação (4.8) cujos pólos estão posicionados dentro do ciclo unitário do plano  $Z$ , conforme ilustra Fig. 4.15.

$$H(z) = \frac{0.216z^3 + 0.167z^2 - 0.049z + 0.0003}{z^4 - 0.355z^3 + 0.293z^2 - 0.605z + 0.0003} \quad (4.8)$$

As Figs. 4.16 e 4.17 apresentam a resposta do sistema ao controlador de velocidade na estrutura RST e a variável de controle (corrente de referência aplicada) respectivamente para um “setpoint” de 650rpm.

A partir da análise da Fig. 4.16 é possível verificar que o sistema controlado atingiu a referência especificada de 650rpm com um pequeno sobre-sinal, mas dentro das especificações desejadas com relação ao percentual de sobre-sinal e tempo de acomodação. É

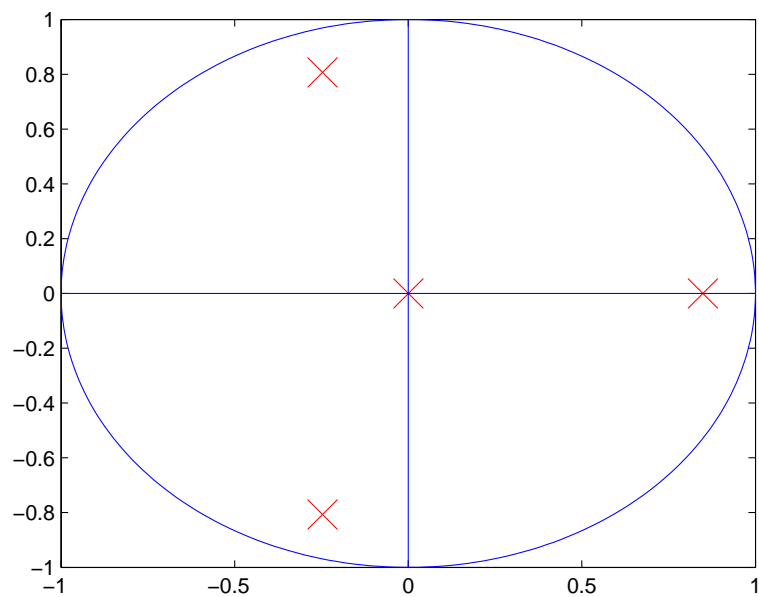


Figura 4.15: Pólos da função de transferência em malha fechada na estrutura RST.

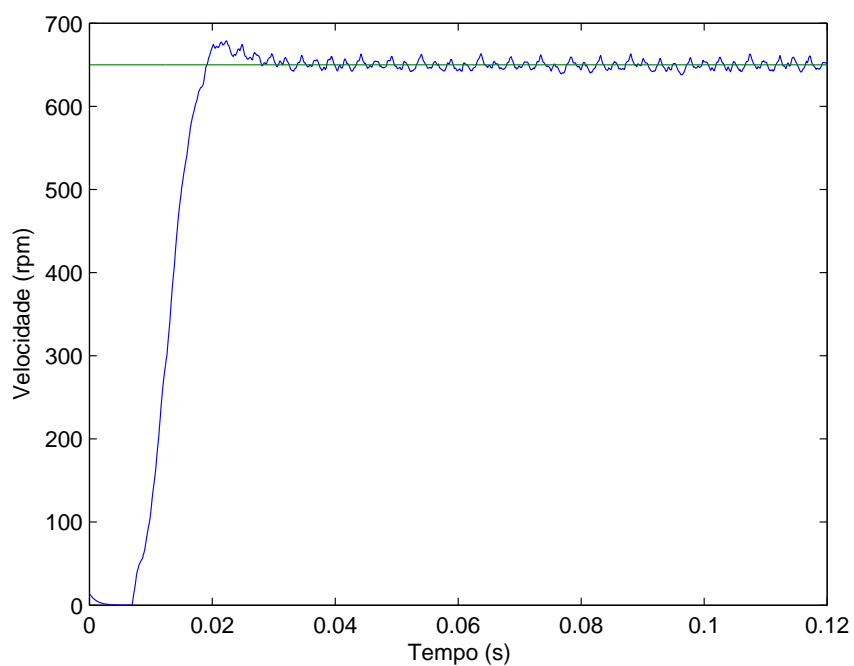


Figura 4.16: Resposta do sistema ao controlador PID aplicado a malha de velocidade do MRV na estrutura RST com referência de  $650rpm$

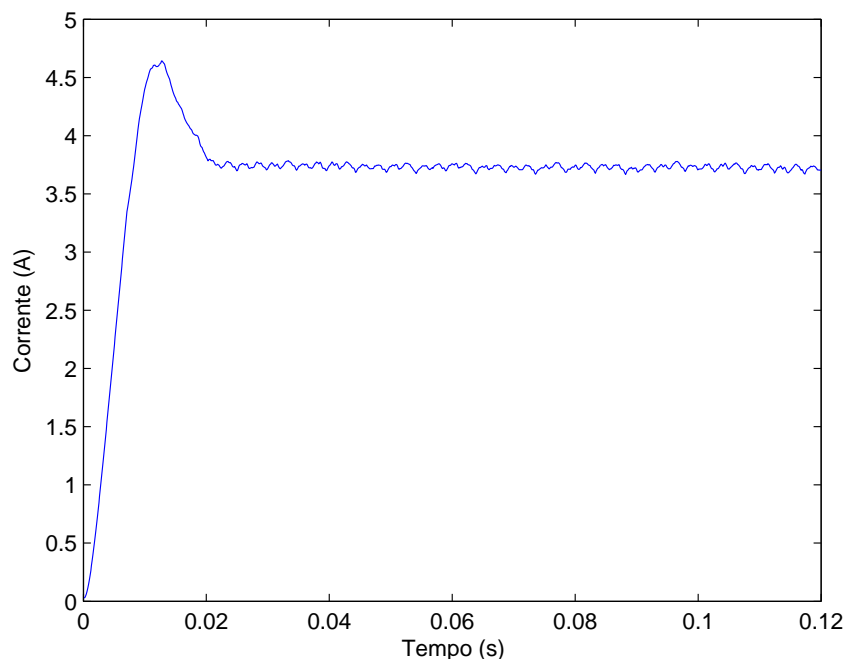


Figura 4.17: Variável de controle do para o controlador PID aplicado a malha de velocidade do MRV na estrutura RST com referência de  $650rpm$

possível verificar também que este apresentou um sobre-sinal menor do que o controlador PID clássico, mas com um tempo de acomodação maior. Comparando a resposta em regime permanente do controlador da estrutura RST e o controlador PID clássico nota-se que o primeiro apresentou um melhora considerável com relação a oscilação devido ao erro do sensor de posição, mostrando-se um sistema mais robusto a ruídos.

Analisando a figura 4.17, referente a variável de controle do sistema para o controlador na estrutura RST, nota-se que o controlador também aplica um pico de corrente e estabiliza logo em seguida com uma oscilação menor, mais ainda existente, também devido ao erro do sensor de posição.

## 4.6 Alteração do ponto de operação

As funções de transferência para malha de corrente e para malha de velocidade, dada pelas equações (4.1) e (4.7), respectivamente, estão associadas a um ponto de operação. Nesta seção de experimentos verifica-se a consistência dos modelos e dos controladores em outros pontos de operação.

As mudanças dos pontos de operação são realizadas automaticamente pelo *software* de

controle desenvolvido. Na malha de corrente esta mudança é realizada no final do quarto ciclo de mudança de fase do início da captura de dados. Na malha de velocidade esta mudança ocorre após  $60ms$  do início da captura de dados.

As Figuras 4.18 e 4.19 mostram a resposta do controlador de corrente PID com parâmetros ajustados pelo método de Ziegler-Nichols modificado utilizado na malha de velocidade do MRV para uma mudança de referência de  $2.5$  para  $3.5A$  e de  $3.5$  para  $2.5A$ , respectivamente.

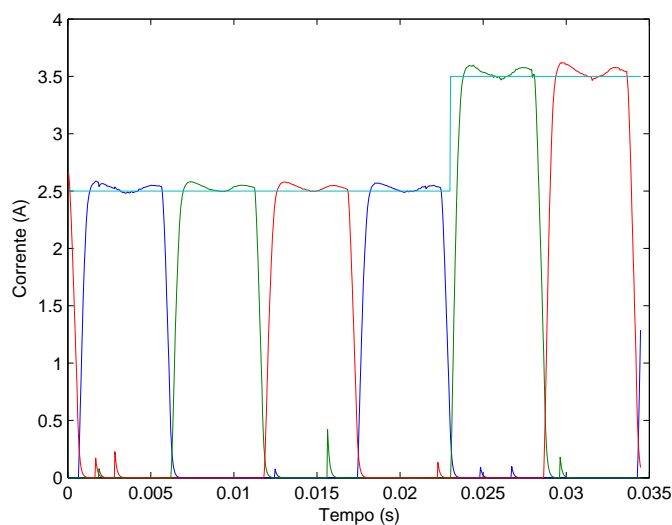


Figura 4.18: Resposta ao controlador de corrente PI Ziegler-Nichols modificado para uma mudança de referência de  $2.5$  para  $3.5A$

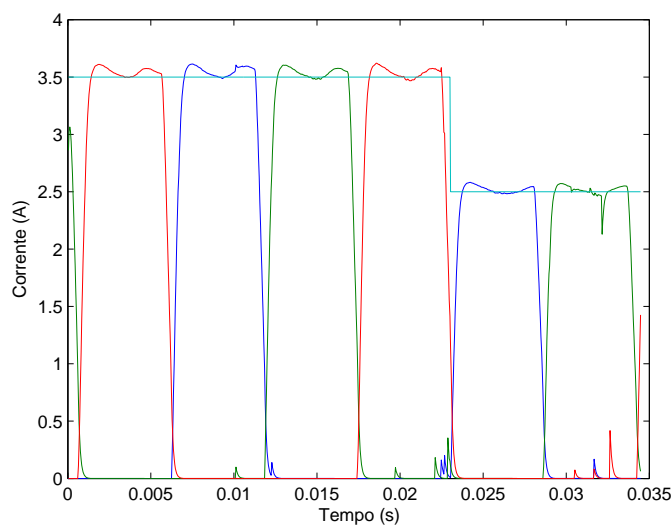


Figura 4.19: Resposta ao controlador de corrente PI Ziegler-Nichols modificado para uma mudança de referência de  $3.5$  para  $2.5A$

As Figuras 4.20 e 4.21 mostram a resposta do controlador de velocidade PID com estrutura RST aplicado ao MRV e variável de controle para uma mudança de referência de 650 para 800rpm, respectivamente.

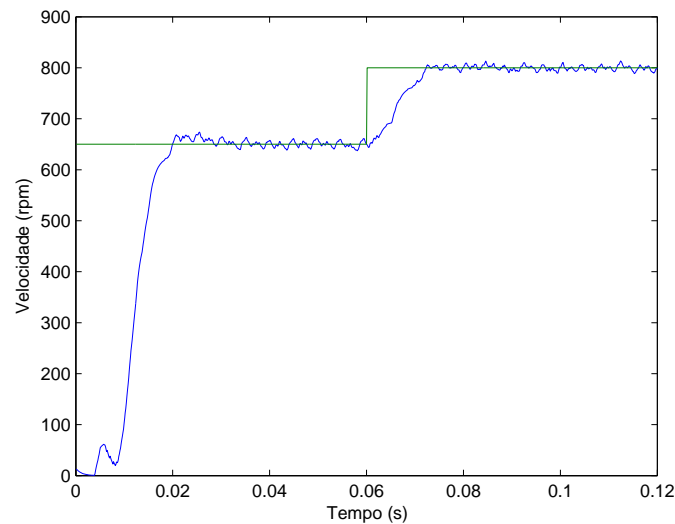


Figura 4.20: Resposta do sistema ao controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 650 para 800rpm

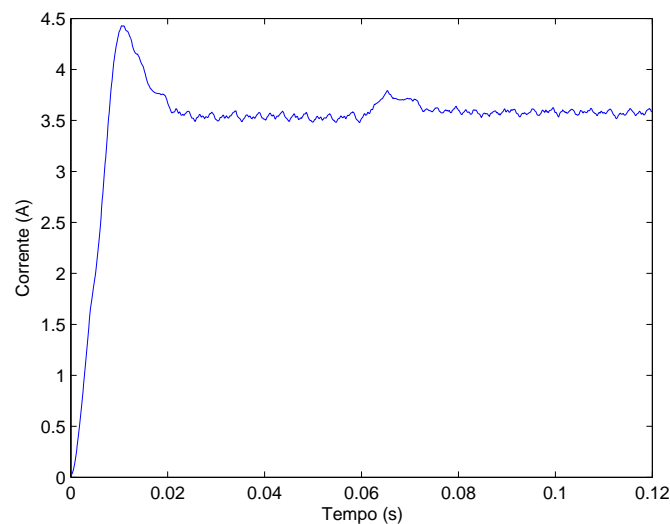


Figura 4.21: Variável de controle para o controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 650 para 800rpm

As Figuras 4.22 e 4.23 mostram a resposta do controlador de velocidade PID com estrutura RST aplicado ao MRV e variável de controle para uma mudança de referência de 800 para 650rpm, respectivamente.

Analisando-se estas Figuras pode-se concluir que mesmo fora do ponto de operação em



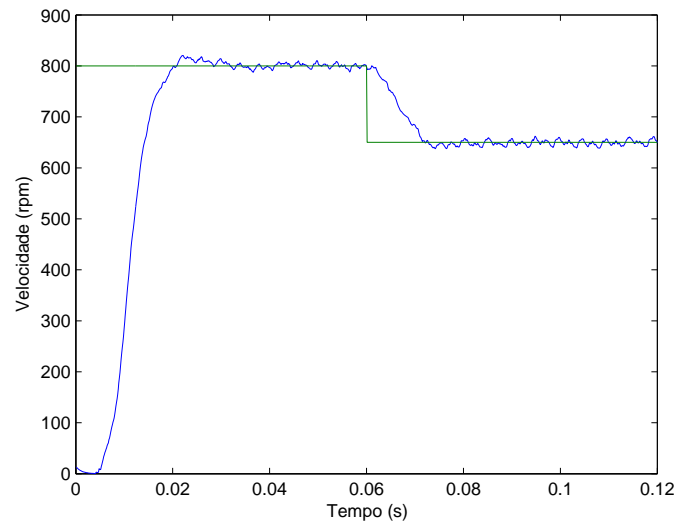


Figura 4.22: Resposta do sistema ao controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 800 para 650rpm

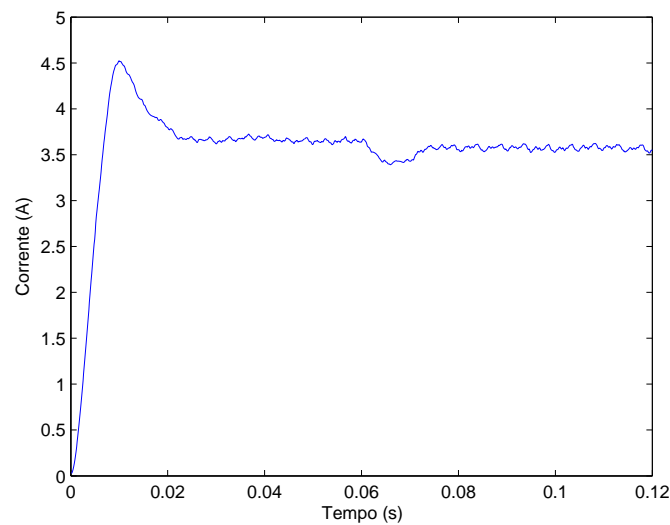


Figura 4.23: Variável de controle para o controlador PID aplicado a malha de velocidade na estrutura RST para uma mudança de referência de 800 para 650rpm

que a função de transferência dos sistemas foram identificadas, os controladores projetados atenderam as especificações de sobre-sinal e tempo de acomodação desejados, mesmo durante uma mudanças de “*setpoint*” de corrente e velocidade. Além disso, mostrarem o ótimo desempenho dos métodos de identificação, que apresentaram funções de transferência com informações da dinâmica do sistema que possibilitou o controle do mesmo fora do ponto de operação em que foram identificadas em um sistema não-linear.

O funcionamento adequado do controlador pode ser qualificado por meio de medidas de índice de desempenho. Neste trabalho foram analisados alguns índices como: resposta

transitória, erro médio quadrático, variância do erro e esforço de controle.

Os dados foram analisados com auxílio do *software Matlab* a partir dos dados coletados pelo DSP. O erro médio quadrático é dado pela média da soma dos quadrados do erro dos dados coletados, e a variância é dada por uma média de dispersão estatística, indicando quão longe em geral os seus valores se encontram do valor esperado. O cálculo da variância é realizado com a função “*var*” no *Matlab* com a diferença dos valores coletados pelo DSP e a referência desejada.

Na análise de resposta transitória verifica-se o sobres-sinal,  $M_s$ , e o tempo de acomodação  $T_r$ .  $T_r$  é dado para malha de corrente como o tempo decorrido para a corrente variar de 5% a 95% da referência, e para a malha de velocidade como o tempo decorrido para a velocidade variar de 5% a 95% da referência. As tabelas 4.4 e 4.5 apresentam estes resultados de forma quantificada para os dois controladores:

Tabela 4.4: Desempenho dos controladores de corrente do MRV para regime transitório ( $M_s$  e  $T_r$ ).

Controlador Corrente	$M_s$	$T_r$
PI Ziegler-Nichols	6%	0.6ms
PI Ziegler-Nichols Modificado	3.2%	0.6ms

Tabela 4.5: Desempenho dos controladores de velocidade do MRV para regime transitório ( $M_s$  e  $T_r$ ).

Controlador Velocidade	$M_s$	$T_r$
PID Ziegler-Nichols	4.6%	9.6ms
PID RST	4.3%	11.64ms

Para a análise dos controladores em regime permanente, as curvas de resposta de corrente do MRV, Figuras 4.5 e 4.7, e resposta de velocidade do MRV, Figuras 4.13 e 4.16 foram utilizadas, podendo-se verificar o erro médio quadrático e a variância do erro, apresentados nas tabelas 4.6 e 4.7. As Figs. 4.24 e 4.25 mostram o erro médio quadrático para os controladores de corrente PI Ziegler-Nichols e PI Ziegler-Nichols Modificado e para o controlador de velocidade PID na estrutura RST respectivamente.

As tabelas 4.4 e 4.6 que expressam os resultados de forma quantitativa em regime transitório e permanente do controladores de corrente aplicados ao sistema, mostram o ótimo desempenho dos controladores aplicados, e as melhoras do controlador PI Ziegler-Nichols modificado quando comparado ao controlador PI Ziegler-Nichols com relação a

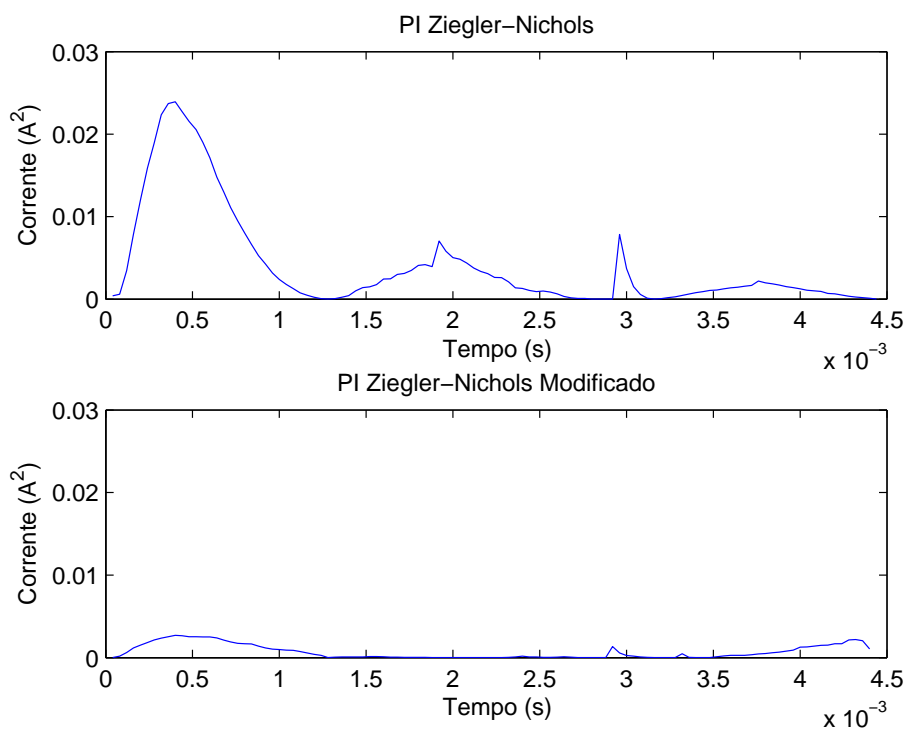


Figura 4.24: Erro quadrático do controlador de corrente PI Ziegler-Nichols e Ziegler-Nichols Modificado.

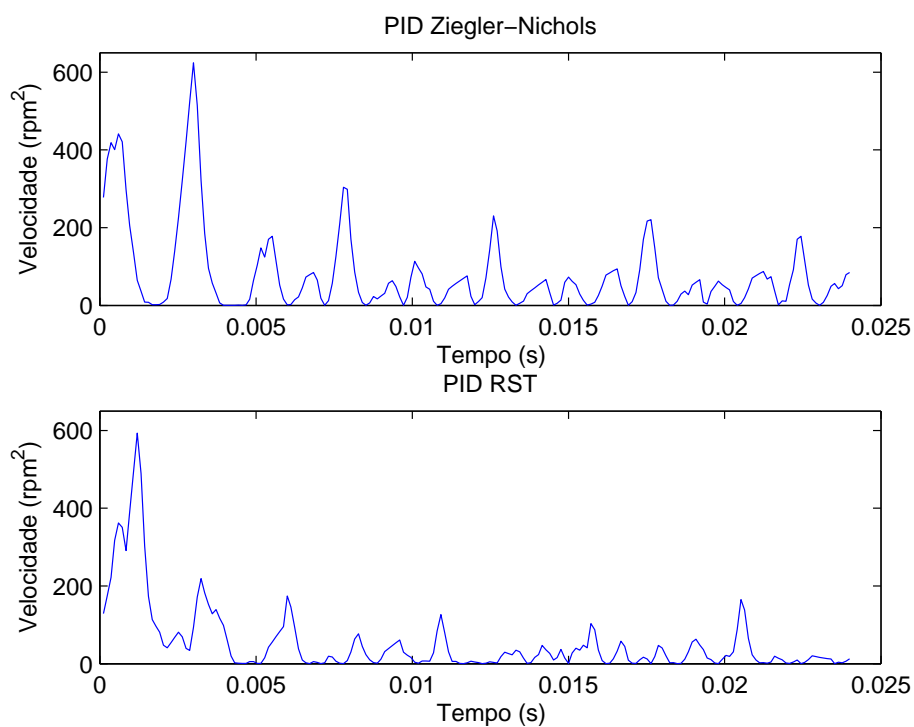


Figura 4.25: Erro quadrático do controlador de velocidade PID na estrutura RST

Tabela 4.6: Desempenho dos controladores de corrente para regime permanente (erro quadrático e variância do erro).

Controlador Corrente	Média do erro quadrático	Variância do erro
PI Ziegler-Nichols	$4.0 \cdot 10^{-3}$	$3.6 \cdot 10^{-5}$
PI Ziegler-Nichols Modificado	$6.9 \cdot 10^{-4}$	$7.1 \cdot 10^{-7}$

Tabela 4.7: Desempenho dos controladores de velocidade para regime permanente (erro quadrático e variância do erro).

Controlador Velocidade	Média do erro quadrático	Variância do erro
PID Ziegler-Nichols	$0.79 \cdot 10^{-2}$	$11.6 \cdot 10^{-3}$
PID RST	$0.52 \cdot 10^{-2}$	$8.2 \cdot 10^{-3}$

sobre-sinal, média do erro quadrático e variância do erro.

As tabelas 4.5 e 4.7 que expressam os resultados de forma quantitativa em regime transitório e permanente do controladores de velocidade aplicados ao sistema, mostram o ótimo desempenho dos controladores aplicados, e as melhoras do controlador PID com estrutura RST quando comparado ao controlador PID clássico com relação a sobre-sinal, média do erro quadrático e variância do erro. Mas apresenta um tempo de acomodação maior, ainda dentro das especificações.

## 4.7 Software de Controle

Desenvolver um programa de controle em tempo-real é um desafio que envolve não apenas a produção de um programa que atenda aos requisitos funcionais, mas que também atenda os requisitos de restrições de tempo.

Este trabalho além do desenvolvimento do controle de velocidade para um MRV, focou no aprimoramento do programa de controle, visando a interação com usuário e o funcionamento adequado, obedecendo aos tempos críticos de controle do sistemas em tempo real.

Para isto, todos os processos críticos devem ser executados de modo que o seu tratamento seja garantido. Para isso estes são executados em interrupções. Na maioria das vezes essas são interrupções de tempo. Também é necessário levar em consideração o acontecimento de mais de uma interrupção ao mesmo tempo. Uma interrupção de maior prioridade (processo mais crítico) não pode ser interrompida por outra interrupção. O processador utilizado possui suporte para mais de uma interrupção de forma seqüencial,

onde é possível tratar suas prioridades.

O projeto do programa pode ser explicado a partir do diagrama de blocos da Fig. 4.26, que pode ser descrita da como a seguir:

Inicialmente todas as variáveis são inicializadas assim como as interrupções e funções primárias e secundárias das portas de entrada e saída. Em seguida um tipo de controle é selecionado. Neste momento é passado como proteção de *software* os valores máximo e mínimo de corrente por fase. Então é inicializado o sistema de posicionamento do rotor, onde é aplicada uma tensão fixa por um certo período, travando o rotor em uma posição para se configurar o sensor de posição. Esta posição é adotada como marco zero. Este procedimento causa um impacto no motor. Devido a isso a tensão aplicada deve ser regulada para que o motor trave de maneira mais suave possível naquela posição.

Uma vez configurados a posição zero e o controlador inicial, o sistema habilita a interrupção global, fazendo com que as interrupções configuradas anteriormente possam atuar. A partir de então o controlador entra em um laço infinito onde são realizados os processos menos críticos como interação com o usuário.

Todos os processos críticos funcionam em um plano a parte, por meio de interrupção de tempo. Entre esses processos estão o controle de corrente, o controle de velocidade, o ajuste do sinal de tensão por PWM e a energização de uma nova fase.

O controlador faz uso de leituras de variáveis do sistema, tanto digitais, no caso da posição do rotor, quanto analógicas, no caso das correntes de fase. A conversão dos valores da corrente de fase para valores digitais apresenta um tempo de atraso que pode influenciar no desempenho do controlador através da violação do período de amostragem. A garantia do período de amostragem do sistema deve ser encarada como um ponto crítico e deve ser implementado artifícios de software e técnicas de programação para garantir o atendimento deste requisito. O DSP escolhido para este trabalho consegue realizar a conversão de uma variável em  $80ns$ , isto possibilita a que todas as variáveis requisitadas pelo controlador estejam disponíveis em menos de  $1\mu s$ . Em microcontroladores rápidos como dsPIC, o tempo de converção para variáveis de 12 bits chega ser de  $2\mu s$ .

O DSP em uso possibilita além de uma medição mais rápida, quando comparada a microcontroladores, a realização de medidas de forma cíclica, podendo executar até dezesseis medições em fila. Esse artifício foi utilizado para se realizar mais de uma medida de uma mesma fase podendo-se trabalhar com uma média entre elas.

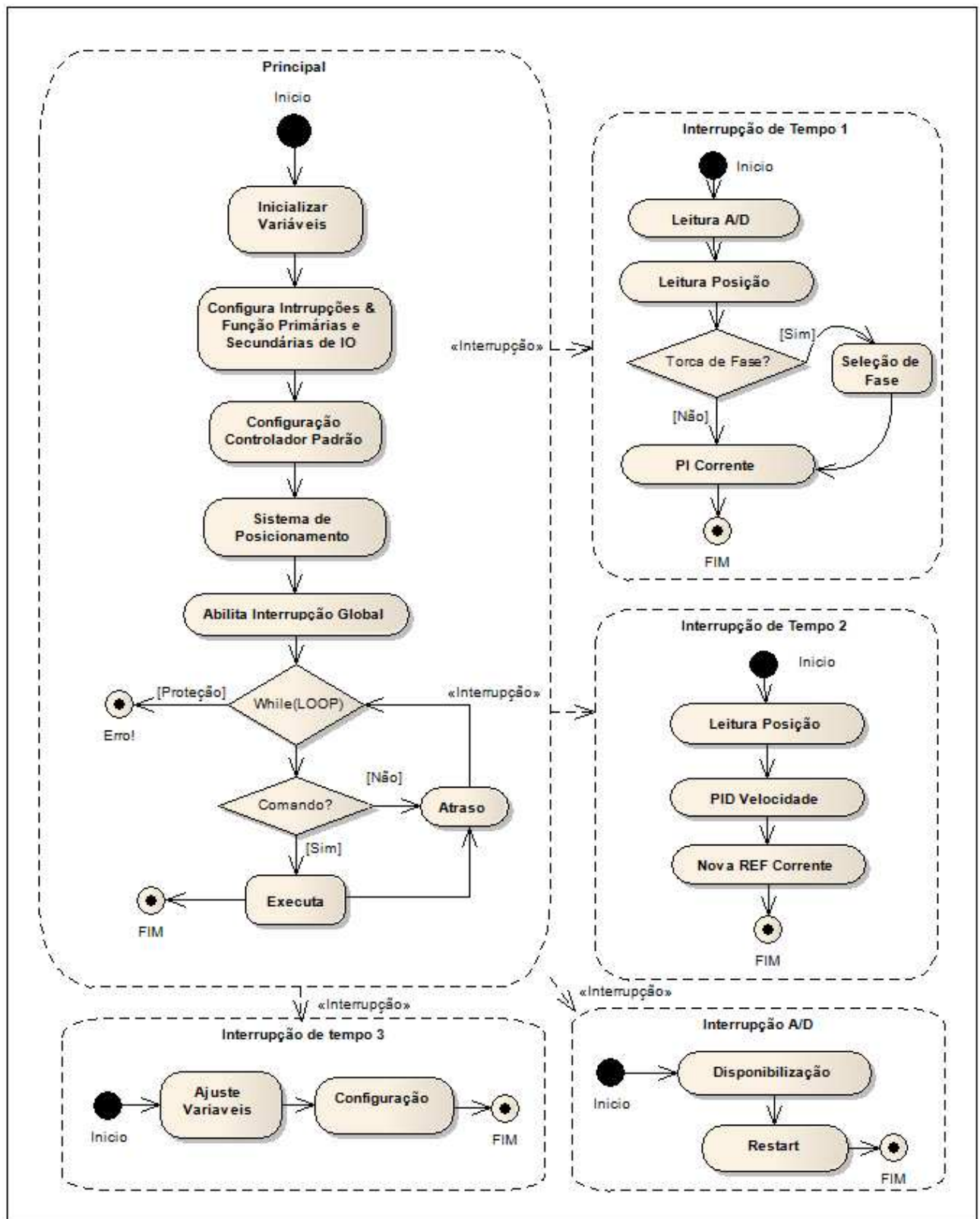


Figura 4.26: Diagrama de fluxo do *software* de controle.

A interrupção de tempo utilizada para controle de corrente do MRV também é utilizada para troca de fase. Ao ser executada a interrupção, o controlador coleta os dados de corrente já disponíveis pelo conversor analógico. Esses dados são aplicados ao controlador PI com os parâmetros ajustados, que são configurados por meio de comandos dados pelo usuário. Nesse momento também é realizada a leitura do encoder de posição aplicando-se o desvio do ângulo de acionamento configurado pelo usuário, também através de comandos do usuário. Por fim o valor resultante é aplicado à variável que ajusta o percentual do sinal de tensão por PWM (variável de saída do sistema).

A interrupção de tempo utilizada no controle de velocidade faz uso do encoder de posição. Comparando o valor atual do encoder com o valor da medição anterior na base de tempo, é possível identificar a velocidade do MRV. Os valores de velocidade são aplicados ao PID selecionado pelo usuário, e então a resposta do controle de velocidade é salva na variável de referência de corrente.

Uma outra interrupção de tempo é utilizada para a realização do PWM, nesta interrupção os valores de saída do controlador de corrente correspondente a cada fase do MRV é aplicado em seu respectivo PWM, produzindo uma modulação de tensão sobre as fases do motor.

O processo de interação do usuário com o controle é realizado como segue: no laço infinito, uma variável indica a existência de um novo comando. Este valor é modificado pelo usuário através do sistema de “*debug*” da interface de desenvolvimento. Assim de tempos em tempos, sem prioridade e muito menos um período exato, é realizada a ação desejada. Dentre os comando existentes tem-se: configurar limites de correntes, seleção de controladores de corrente e velocidade, mudança de referência, configuração de tempo para mudança automática de referencia, captura de dados, início e termino de operações, entre outros comandos. Estes comandos são também aceitos com o sistema em repouso.

#### 4.7.1 Considerações Finais

A aplicação do método do relé “*setpoint*” possibilitou a identificação da malha de corrente de maneira simples e prática. A função identificada foi validade com a resposta dos controladores ajustados a partir dos parâmetros da função estimada.

Foi possível obter uma resposta dentro das especificações de projeto tanto para o controlador com os parâmetros ajustados pelo método de Ziegler-Nichols como pelo método

de Ziegler-Nichols Modificado, tendo o segundo método uma melhor resposta quanto ao sobre-sinal, erro quadrático médio e variância do erro. Os controladores foram testado também fora do ponto de operação para os quais foram projetados, e ainda assim obtiveram uma resposta dentro das especificações, possibilitando comprovar o ótimo desempenho no MRV tanto do método de identificação como dos métodos de ajuste de controladores apresentados.

O desempenho adequado do método de identificação utilizado na malha de velocidade, mínimos quadrados recursivos “*off-line*”, também foi comprovado com as respostas dentro das especificações de projeto dos controladores ajustados a partir da função de transferência estimada. O bom desempenho dos controladores de velocidade também avaliado com as respostas deste em pontos de operações diferentes para os quais foram projetados.

Mesmo com um bom resultado, a utilização do PID convencional na malha de velocidade apresentou uma pequena oscilações em regime permanente devido ao erro gerado pelo sensor de posição, onde na taxa de amostragem utilizada a variação de  $\pm 1bit$  acarreta um erro de  $\pm 17rpm$ . A utilização da estrutura RST no controlador PID para malha de velocidade mostrou melhores resultados em regime permanente com relação a este problema. Esta oscilação não compromete o controlador por estar dentro de um erro aceitável menor que 3%.



# Capítulo 5

## Conclusão e Trabalhos Futuros

### 5.1 Conclusões

Neste trabalho foram implementados algoritmos de identificação e controle aplicados à um sistema de acionamento para uma máquina de relutância variável. Inicialmente foram apresentados os aspectos gerais sobre o MRV e suas principais características. Vantagens e desvantagens do MRV foram discutidas e observou-se que a aplicação do MRV é vantajosa onde se requer alto conjugado de partida e ampla faixa de velocidade são necessárias, pois nestes casos existe uma maior relação custo-benefício sobre o motor CC convencional de ímã permanente sem escovas.

Apresentou-se também aspectos construtivos e operacionais dos MRVs. Observou-se que variando o número de fases, o número de pólos no estator e o número de pólos do rotor, muitas geometrias diferentes de MRVs podem ser realizadas. Geralmente o aumento do número de fases reduz a oscilação de conjugado, entretanto, aumenta-se a complexidade dos arranjos eletrônicos para a operação do MRV. Além disso, em geral, as fases do motor são independentes, não existindo problemas como curto de braço e tempo morto de chaveamento em sistemas com MRVs.

O conversor de potência e o sistema de controle que foram utilizados no sistema de acionamento do MRV para este trabalho, foram descritos mostrando os aspectos construtivos e o interfaceamento do conversor de potência com o processador digital de sinal - DSP. Destacou-se algumas características principais dos componentes utilizados, DSP *TMS320F2812*, sensor de posição *845GM – F1GSHC1024R*, transdutor de corrente *LA25 – NP*.

Foi proposta uma identificação da malha de corrente por meio de um método de identificação em frequência do tipo relé *setpoint*, que mostrou-se eficaz pois os controladores do tipo PI ajustados utilizando métodos de Zielger-Nichols e Ziegler-Nichols Modificado a partir dos parâmetros encontrados na identificação, apresentaram ótimos resultados baseados nos requisitos de projeto apresentados no capítulo 4. Os controladores de corrente implementados também mostraram resultados satisfatórios quando baseados no mesmo requisito de projeto. Na avaliação do desempenho dos controladores fora do ponto de operação observou-se, embora não matematicamente comprovada, uma robustez do controlador em atender os requisitos de projeto.

As análises de transitório realizadas com os controladores de corrente resultaram em um menor valor de sobre sinal para a resposta do PI sintonizado pelo método de Ziegler-Nichols Modificado, mas um mesmo tempo de acomodação. Nas análises em regime permanente, o controlador sintonizado pelo método de Ziegler-Nichols Modificado, apresentou uma média no erro quadrático 5,7 vezes menor e uma variância do erro 50,7 vezes menor, com relação ao controlador PI com os parâmetros ajustado pelo método de Ziegler-Nichols.

A partir da análise realizado em regimes transitório e permanente dos controladores de correntes, foi selecionado o controlador PI com os parâmetros ajustados pelo método Ziegler-Nichols Modificado para compor o controlador de velocidade devido o seu melhor desempenho.

Antes de sintonizar o controlador de velocidade foi realizada a identificação da malha de velocidade. Para isso foi utilizado o método de identificação por meios dos mínimos quadrados, que também mostrou-se eficaz uma vez que os controladores do tipo PID ajustados por Zielger-Nichols utilizando a função de transferência identificada também apresentaram ótimos resultados baseado nas especificações de projeto do capítulo 4, mesmo fora do ponto de operação inicialmente especificado.

Foram realizados testes com dois tipos de estruturas de controladores para malha de velocidade, uma primeira PID convencional e uma PID na estrutura RST. A análise de transitório realizadas com os controladores de velocidade mostraram uma pequena melhora com relação ao sobre-sinal na resposta do PID na estrutura RST, e uma melhora considerável de aproximadamente  $2ms$  com relação ao tempo de acomodação.

A utilização do PID convencional na malha de velocidade apresentou uma pequena

oscilações em regime permanente devido ao erro gerado pelo sensor de posição. A utilização da estrutura RST no controlador PID para malha de velocidade mostrou melhores resultados em regime permanente com relação a este problema. O erro apresentado foi menor que 3%, e com uma média nula.

O bom desempenho dos controladores apresentados é um reflexo das melhorias e considerações implementadas no *software* de controle como: tratamento dos controladores por interrupção; conversão  $A/D$  de forma cíclica; melhor seleção dos tipos de variáveis.

Os estudos realizados neste trabalho contribuíram para o aperfeiçoamento do sistema de acionamento da máquina de relutância variável do laboratório GPAR na Universidade Federal do Ceará. Melhorando o hardware de acionamento, resultando na redução de ruídos de medição. Inserindo novas técnicas de programação, resultando no melhor desempenho do DSP e na possibilidade de captura de resultados com o mesmo. Desenvolvimento do controlador de velocidade, viabilizando o uso do sistema em muitas aplicações industriais que exigem esse controle.

Mesmo com as melhorias realizadas, durante o período de desenvolvimento deste trabalho foi possível destacar alguns pontos interessantes que devem ser tratados no futuro. Este são citados na seção seguinte.

## 5.2 Propostas para trabalhos futuros

A fim de melhorar o desempenho do sistema desenvolvido e aprofundar o conhecimento no tema abordado, além dos estudos realizados, são apresentadas abaixo algumas sugestões para trabalhos futuros.

1. Desenvolvimento de um supervisor para avaliar o desempenho da malha de corrente e resintonizar o controlador quando os índices de desempenho divergirem das especificações;
2. Desenvolvimento de funções de correlação entre corrente e torque para o estudo do conjugado, viabilizando o uso do sistema em aplicações que exigem o torque contínuo;
3. Implementação de algoritmos de controle adaptativos, afim de se obter um melhor desempenho no ponto de operação de cada instante tendo visto que tanto a malha

de velocidade com a malha de corrente são não lineares;

4. Desenvolvimento de um algoritmo para estimar a posição do rotor podendo partir para aplicações “sensorless”, diminuindo assim o custo do hardware utilizado tornando aplicações com MRV mais competitivas no mercado.

# Apêndice A

## O processador DSP *TMS320F2812*

O sistema de controle para o Acionamento do MRV exibe duas partes básicas e distintas. O núcleo, que é o responsável por toda "inteligência" do Sistema de Controle e a Interface, responsável por compatibilizar o núcleo para a atuação com a parte o módulo de Potência do Sistema de Controle. Para o núcleo do sistema de controle realizado neste trabalho foi utilizada uma DSP da Texas Instruments do tipo *TMS320F2812*, a qual oferece uma performance de 150 milhões de instruções por segundo (150 MIPS) em uma CPU de 32 bits. Além disso, este processador disponibiliza uma série de periféricos integrados listados a seguir (Texas, Instruments, 2002):

- 2 Gerenciadores de Eventos (EVA e EVB);
- Temporizadores de uso geral;
- Geradores de PWM para aplicações de controle;
- Conversores Analógicos Digitais de 12 bits
- Interface do tipo "Controller Área Network" (eCAN);
- Interface periférica serial (SPI)- porta serial síncrona;
- Interface de Comunicação Serial (SCI) - porta serial assíncrona;
- Pinos de entrada e saída para uso geral do tipo bidirecionais (GPIO);
- Dispositivo de reset para integridade do sistema ("Watchdog Timer").

Para o sistema de controle de acionamento do MRV, os principais periféricos utilizados são os módulos Gerenciadores de Eventos (EVA e EVB), que controlam os temporizadores e os módulos responsáveis pelo PWM, e o conversor AD. A seguir discute-se o papel de cada um destes periféricos para o funcionamento deste sistema de controle. O processador DSP *TMS320F2812* possuiu dois módulos Gerenciadores de Eventos (EVA e EVB) idênticos em termos de funcionalidades e de definição de registros, porém possuem diferentes nomes e endereços para seus registros. Os registradores do EVA iniciam-se no endereço 7400h enquanto os do EVB no endereço 7500h. Tais módulos EV's proporcionam ao núcleo de controle uma ampla faixa de funções e opções particularmente úteis em aplicações para controle de máquinas elétricas citadas a seguir:

- Lógica para interrupções;
- 2 temporizadores de uso geral para cada EV;
- 3 unidades de comparação para cada EV;
- Unidades de PWM com opção para PWM vetorial, programação de “dead bands”
- Circuito de Quadrature Encoder Pulse (QEP).

A Fig. A.1 mostra o diagrama de blocos do Gerenciador de Eventos A. Para o Gerenciador de Eventos B tem-se um diagrama funcional praticamente idêntico.

Há dois “*Timers*” de uso geral em cada modulo EV, GP “*Timers*” 1 e 2 para o EVA enquanto os GP “*timers*” 2 e 4 respondem ao EVB. Os “*Timers*” podem ser operados independentemente ou de forma sincronizada, se o sinal de entrada pode ser tanto o “clock” da CPU quanto um sinal de “clock” externo. Uma ampla variedade de multiplicadores pode ser utilizada para variar a frequência de cada contador. Cada “*Timer*” é configurado para seu modo de contagem no registrador de Controle de Temporização respectivo (*T·CON*). Assim quando o contador é habilitado ele funcionará em um dos 4 modos básicos de funcionamento seguinte:

- Modo “Stop/Hold”;
- Contagem Crescente Contínua;
- Contagem Crescente/Decrescente Direcional;

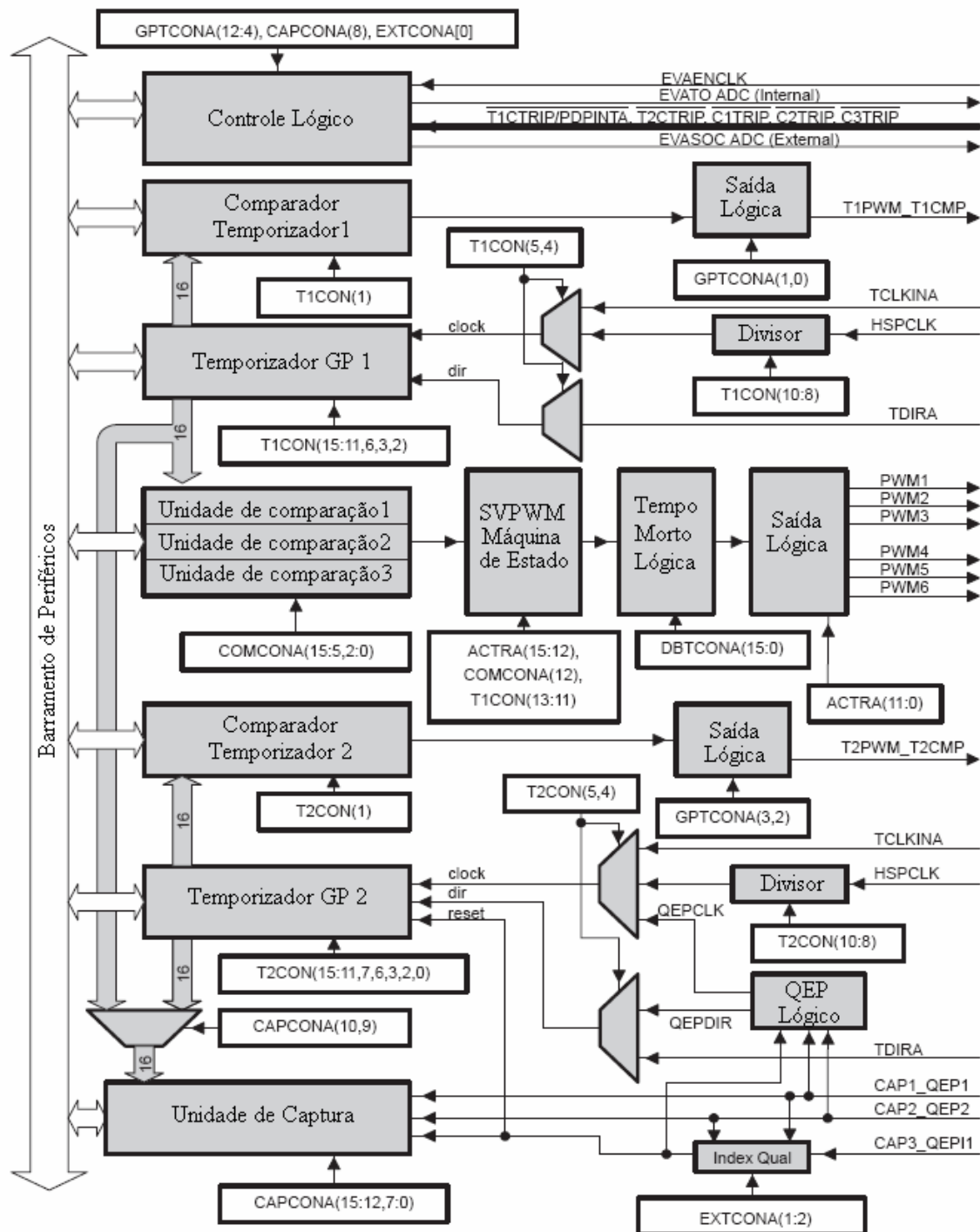


Figura A.1: Diagrama de blocos do EVA.

- Contagem Crescente e Decrescente Contínua.

A Fig. A.2 ilustra o funcionamento do Timer 3 para a situação descrita anteriormente.

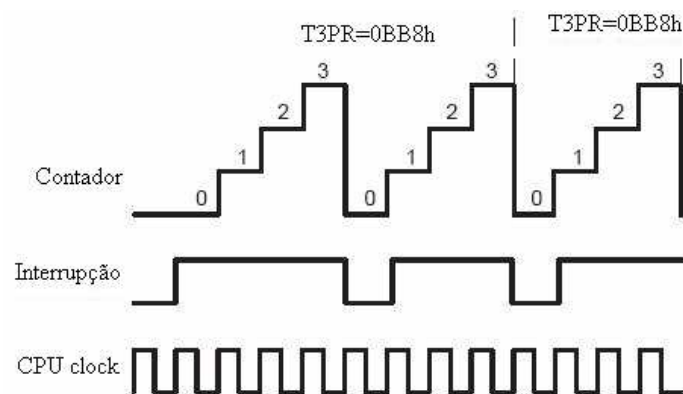


Figura A.2: Funcionamento do contador Timer 3.

A Fig. A.3 mostra o diagrama de blocos do circuito equivalente para a geração de PWMs.

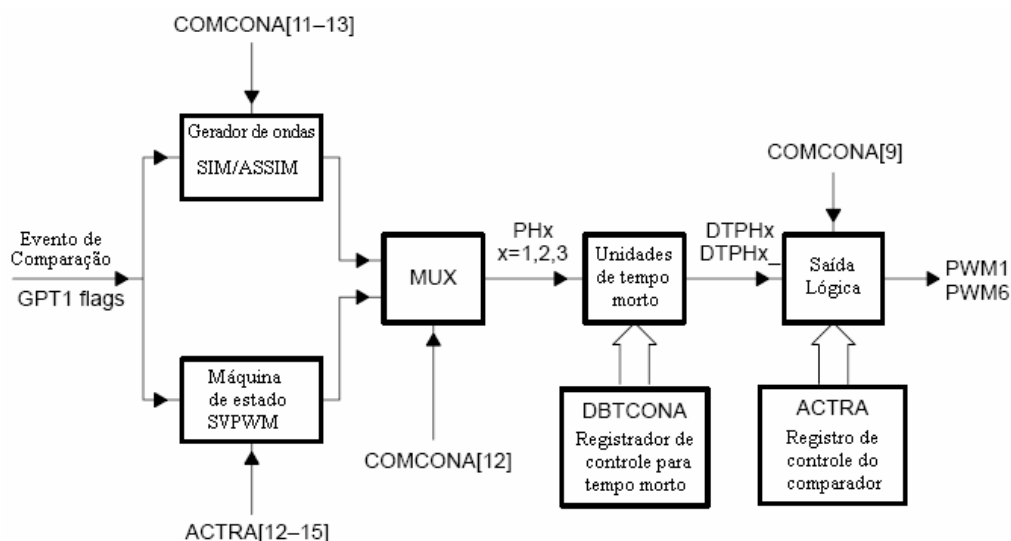


Figura A.3: Diagrama de blocos do circuito equivalente para a geração de PWMs.

Para a geração de um sinal assimétrico de PWM um contador é selecionado para repetir a contagem que será o período da portadora. Com registrador de comparação é utilizado para os valores do sinal modulado. O valor do registrador de comparação é constantemente comparado com o valor atual do contador. Quando seus valores se igualam uma transição, que pode ser de alto para baixo ou de baixo para alto, acontece no pino de saída associado. Assim um pulso de saída, alto ou baixo, é gerado de forma que sua duração é proporcional ao valor do registrador de comparação. Este processo é repetido para cada período da portadora com larguras de pulsos diferentes determinados



pelo registro de comparação. A Fig.A.4 ilustra o funcionamento do PWM assimétrico.

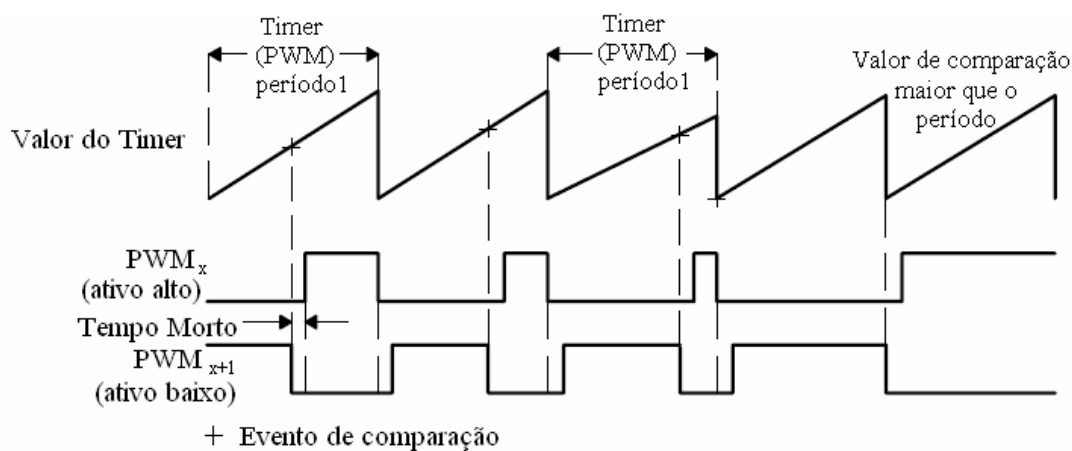


Figura A.4: Funcionamento do PWM Assimétrico.

O módulo Conversor Digital-Analógico (ADC) da *TMS320F2812* é um conversor de 12 bits com 16 canais que também podem ser configurados para o funcionamento como dois módulos distintos de oito canais a serviço dos gerenciadores de eventos A e B. Apesar da possibilidade do funcionamento como dois módulos de oito canais distintos, fisicamente há apenas um módulo ADC multiplexado como mostra a Fig. A.5.

Algumas das principais funcionalidades do Módulo ADC incluem:

- Núcleo de conversão de 12 bits com amostrador (S/H) duplo;
- Modos de amostragem simultânea ou seqüencial;
- Entrada analógica de 0V a 3V;
- Tempo de conversão de  $200nS$  para conversão única ou de  $60ns$  para conversão em fila;
- 16 canais de entrada;
- Livre escolha para sequenciamento de conversões dos 16 canais;
- Múltiplos gatilhos para "Início de Conversão" que podem atuar de forma independente para o modo de funcionamento distinto.

Um sinal de Modulação por Largura de Pulso, ("Pulse Width Modulation -PWM"), é uma seqüência de pulsos de largura variável. Tais pulsos são separados entre si por

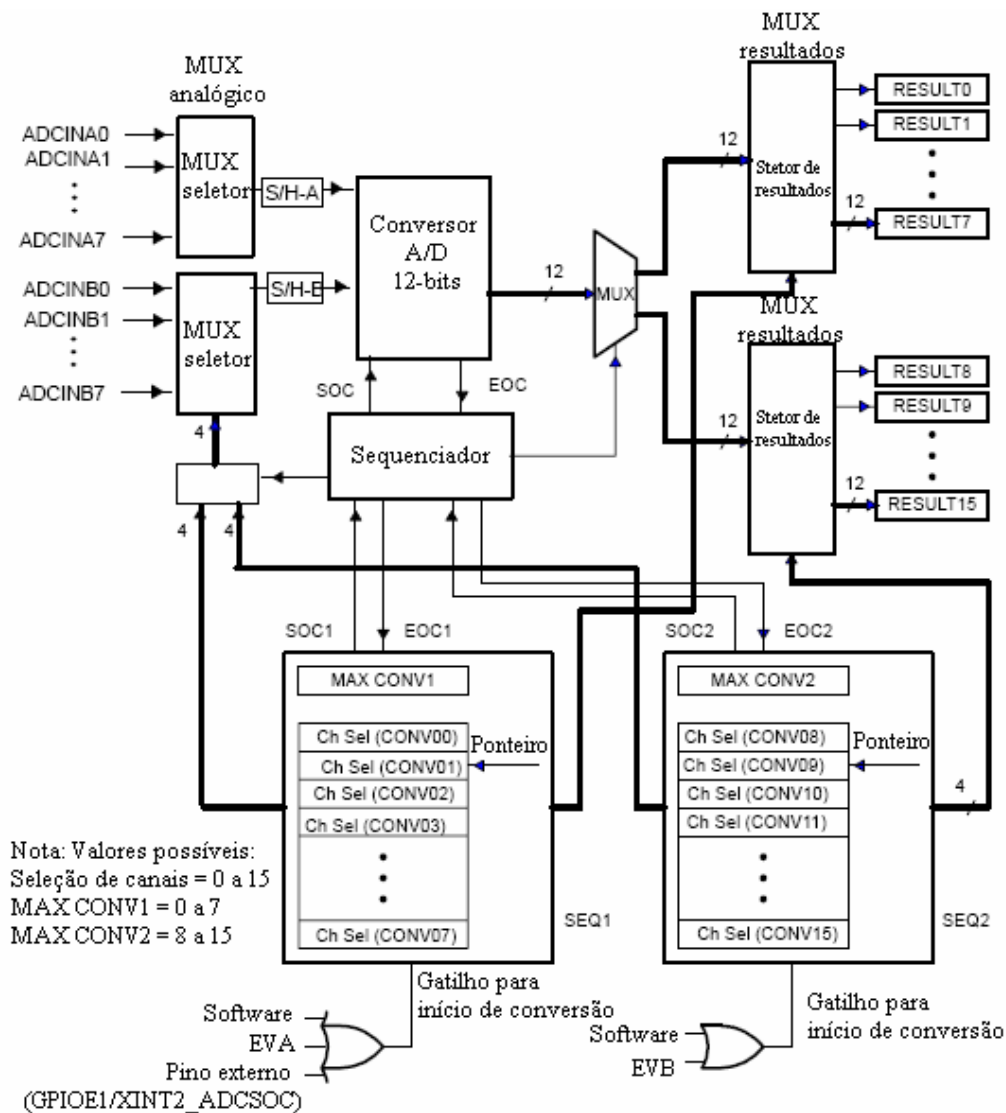


Figura A.5: Diagrama de funcionamento circuito ADC.

um sinal de período fixo de forma que há apenas um pulso em cada período. O sinal de período fixo é chamado de período da portadora PWM e o seu inverso é chamado de frequência da portadora PWM. As larguras dos pulsos do PWM são determinadas ou moduladas de acordo com uma seqüência de valores desejados conhecidos como sinal modulado. Nos sistemas de controle, os sinais de PWM são utilizados para controlar o momento de abertura e fechamento das chaves de potência responsáveis pela corrente e energia entregues aos enrolamentos das máquinas. As unidades e PWM têm por objetivo minimizar a intervenção do usuário na geração de PWM's para sistemas de controle, bem como eficientizar o processamento desta funcionalidade pela CPU. A capacidade para

geração de PWM's de cada Gerenciador de Eventos é resumida a seguir:

- Cinco PWMs independentes. Três dos quais gerados pelas unidades de PWM enquanto dois restantes são gerados pelos comparadores dos contadores;
- Três PWM adicionais dependentes dos sinais gerados através das unidades de PWM;
- “Tempo Morto” programável para os pares das unidades de PWM dependentes;
- Largura de pulso e incremento/decremento da largura mínima de um “pulso de clock”;
- PWM com resolução de 16 bits;
- Mudanças da frequência da portadora e das larguras de pulso “em execução”;
- Interrupção para proteção dos sistemas de acionamento;
- Geração de PWMs simétricos, assimétricos e vetoriais;
- Comprometimento mínimo da CPU coma geração através dos circuitos dedicados para o PWM.

# Apêndice B

## Transdutores / Bancada Experimental

### B.0.1 Transdutores de Corrente e Tensão

Transdutor de corrente *LEM LA25 – P*

Tabela B.1: Características elétricas do transdutor de tensão e corrente

Referência	Range	Saida
<i>LEM LA25 – P</i>	25A	4 – 24mA
<i>LEM LV25 – P</i>	10 a 500V	4 – 24mA

- Fabricante - LEM Components. O transdutor de corrente *LEM LA25 – P* vem conectado às saídas de potência do conversor de potência.

### B.0.2 Encoder

A figura B.1 apresenta o sensor gray *Allen-Bradley 845GM – F1GSHC1024R* de 10 – bits utilizado para realimentar a malha de velocidade e identificar o ponto de chaveamento.

Características:

- Gray Code Absoluto, Binário Natural ou Decimal Codificado em Binário;
- 8-15 bits de resolução (256 a 32,768);
- Coletor Aberto, TTL, saídas push-pull ou SSI;
- Pino de ajuste eletrônico zero;

- Saída opcional de nível baixo ou alto;
- NEMA Tipo 4 e 13;
- Proteção contra polaridade inversa;
- Temperatura de operação de 85°C;
- Identificação CE para todas as diretrizes aplicáveis.



Figura B.1: Sensor gray 10 – bits Allen-Bradley.

### B.0.3 Bancada Experimental

MOSFET tipo *IRFP460A* ( $V_{DSS} = 500V$ ,  $R_{DS} = 0,27V$ ,  $I_D = 20A$ ) da *International Rectifier*. Para os diodos de “roda livre” foram selecionados seis dispositivos do tipo *MUR460* ( $I_O = 40A$ ).

# Apêndice C

## Software

### C.1 Controlador PI corrente + PID Velocidade para *TMS320F2812*

Listing C.1: MRV.h

```
0 //Variables for Current Control
struct tCCVars {
    //PWM Variables
    Uint32 TogA;
    Uint32 TogB;
    Uint32 TogC;
    //PWM Lock Variables
    Uint32 TogForLock;
    //Control
    Uint32 uA,lastUA, uB, lastUB, uC, lastUC;
10 //Erro Variables
    int16 erro, lastErro, lastLastErro;
};

//Variables for Speed Control
struct tSCVars {
    //Control
    Uint16 current, lastCurrent;
    int speed, lasSpeed, lasLastSpeed;
    Uint16 YinterruptForSpeed;
    Uint16 DesvioForSpeed;
20 //Erro Variables
    int16 erro, lastErro, lastLastErro;

    int filtredOutPut;
};

//Motor Status
struct tStatus {
30 //Position
    //Angle and Switch Angle Variables
    int angle, lastAngle; //The diference of lastAngle & angle in a timer period results in Speed
    /* Phase Controller Variables */
    int phase, lastPhase; // Which is the current phase and the last phase

    //Lock phase variables
    int lock;
    //Current
40 int16 currentA;
    int16 currentB;
    int16 currentC;
    //Speed
    int speed;
};

//Referencies for controller
struct tRefs {
50 Uint16 MaxCurrent, MinCurrent; // Set the MAX and MIN current for current control
    // 600 = 6A, 100 = 1A
    Uint16 current;
    Uint16 REFlock;

    Uint16 speed;
};

//PID Parameters
struct tPID {
```

```

60     int32 P;
        int32 I;
        int32 D;
    };

    //Controller
    struct tController {
        int switchAngle; //This variable modify the instante of phase activation
        int WPhase; // Which phase start the machine

70     struct tPID current;
        struct tCCVars currentVars;
        Uint32 (*currentControl)();

        struct tPID speed;
        struct tSCVars speedVars;
        Uint32 (*speedControl)();
    };

    //Motor
80     struct tMRV {
        struct tController controller;
        struct tStatus status;
        struct tRefs ref;
        //Command System variable
        int COMMAND; //Variable used to input command to the machine
    };

90     void init_VarController(int16 Ref, int16 RefLock, int16 MaxCurrent, int16 MinCurrent, int16 Speed){
        mrv.ref.MaxCurrent = MaxCurrent;
        mrv.ref.MinCurrent = MinCurrent;
        mrv.ref.current = Ref;
        mrv.ref.REFLock = RefLock;
        mrv.ref.speed = Speed;
    }

100     Uint32 setParam(struct tPID *contoller, Uint32 Kp, Ki, Kd){
        contoller->P = Kp;
        contoller->I = Ki;
        contoller->D = Kd;
    }

```

Listing C.2: main.c

```

0 //#####
//
// TITLE: Switched Reluctance Motor Speed Control with DSP TMS320F2812
//#####
//
// Ver | dd mmm yyyy | Who | Description of changes
//====|=====|=====|=====
10 // 1.00 | 23 Jul 2008 | Sobreira | First Implementation (Objects Structure)
//#####

#include "DSP281x_Device.h" // DSP281x Headerfile Include File
#include "DSP281x_Examples.h" // DSP281x Examples Include File

#include "MRV.h" // Motor Variables

#include "TimerControl.h" // Timer Control
#include "System.h" // System core
20 #include "CurrentControl.h" // Current Control
#include "SpeedControl.h" // Speed Control

//#####
// Global Variables //
//#####

30 struct tMRV mrv;
    Uint16 countC, countV;
    int16 vSpeed[1000];
    Uint32 vSpeedControl[1000];

    int16 vCurrent[3][1000];
    Uint32 vCurrentControl[1000];

//#####
// Prototype statements for functions //
40 //#####
void switchPhase(void); // Make the switch of the current phase to next phase
extern void CheckAngle(void); // Check position encoder and save the value
int gray2bin(int numgray); // Convert the encoder position value to Degrees
void Delay1(int count); // Just wait

//#####
// Main Core //
//#####
50 void main(void)
{
    InitSystem(); // Set the basic functions of the processor

    //SetUP the switch point
    mrv.controller.switchAngle = 0; // 0 make switch Angle 15 Degrees

    //StartUP Current Control System

```

```

setupCurrentPI(93312,89129); //Coefs ZNm (Value * 1000)
//StartUP Speed Control System
setupSpeedRST(TRUE); //enable PID RST struct

60 //StartUP Control System
//((Current Reference, Current For StartUP, Max Current, Min Current, Speed(RPM))
init_VarController(250, 250, 500, 50, 800); // Value scale 100x (A)

//System LOOP Status
loop=1;

70 while(loop){
//Verify angle and switch phase
CheckAngle();

// In case of COMMAND
switch(mrv.COMMAND){

case 0:
// DO NOTHING
break;

80 case SetUPCurrentZN:
setupCurrentPI(5490,1675); // (Value * 1000)
clearErrors(); //Clear all variables
break;
case SetUPCurrentZNm:
setupCurrentPI(17200,3075); // (Value * 1000)
clearErrors(); //Clear all variables
break;
case SetUPSpeedZNm:
setupSpeedPID(TRUE); //enable PID convecional MODE
setParam(&mrv.controller.speed, 2787, 916, 210);
clearErrors(); //Clear all variables
break;
90 case SetUPSpeedRST:
setupSpeedPID(FALSE); //enable PID RST struct
setParam(&mrv.controller.speed, 2787, 916, 210);
clearErrors(); //Clear all variables
break;

case ManualParamSetUP:
//Need to put a breakpoint here
clearErrors(); //Clear all variables
break;

case STARTCapture:
clearErrors(); //Clear all variables
countC=0;
countV=0;
break;

100 default:
powerDOWN(); // Take system down (FAIL)
}

mrv.COMMAND = 0;
}

120 //#####
// Active the current Phase //
//#####
void switchPhase(void){
GpioDataRegs.GPADAT.bit.GPIOA13=1; //Phase C
GpioDataRegs.GPADAT.bit.GPIOA14=1; //Phase B
GpioDataRegs.GPADAT.bit.GPIOA15=1; //Phase A

130 switch(phase) {
case 0:
GpioDataRegs.GPADAT.bit.GPIOA15=0; //Phase A
break;
case 1:
GpioDataRegs.GPADAT.bit.GPIOA14=0; //Phase A
break;
case 2:
GpioDataRegs.GPADAT.bit.GPIOA13=0; //Phase A
break;
}
}

140 //#####
// Read Sensor and Convert to Degrees //
//#####
extern void CheckAngle(void){
int ValorGray=GpioDataRegs.GPB DAT.all; //GET encoder value
int ValorBin=gray2bin(ValorGray); //Convert value to Degrees
angle=ValorBin*0.3515;

150 angle--angleSwitch; //Make the angle compensation

if (angle<0) angle=360-angle; //Verify if have a 0<->360 Transaction
if (angle>360) angle--=360;

//Set the current phase according to the angle
if (((angle>=0.0)&&(angle<15.0))||((angle>=45.0)&&(angle<60.0))||
((angle>=90.0)&&(angle<105.0))||((angle>=135.0)&&(angle<150.0))||
160 ((angle>=180.0)&&(angle<195.0))||((angle>=225.0)&&(angle<240.0))||
((angle>=270.0)&&(angle<285.0))||((angle>=315.0)&&(angle<330.0)))
phase=2;
if (((angle>=15.0)&&(angle<30.0))||((angle>=60.0)&&(angle<75.0))||
((angle>=105.0)&&(angle<120.0))||((angle>=150.0)&&(angle<165.0))||
((angle>=195.0)&&(angle<210.0))||((angle>=240.0)&&(angle<255.0))||

```



```

(( angle >=285.0)&&(angle <300.0))||((angle >=330.0)&&(angle <345.0))
    phase=1;
    if((( angle >=30.0)&&(angle <45.0))||(( angle >=75.0)&&(angle <90.0))||
(( angle >=120.0)&&(angle <135.0))||(( angle >=165.0)&&(angle <180.0))||
(( angle >=210.0)&&(angle <225.0))||(( angle >=255.0)&&(angle <270.0))||
(( angle >=300.0)&&(angle <315.0))||(( angle >=345.0)&&(angle <360.0))
    phase=0;

// If the currente phase is diferent SWITCH
if (lastPhase!=phase){
    lastPhase=phase;
    switchPhase();
}
}

#####
//          Conver gray Value to Binary          //
#####
int gray2bin(int numgray)
{
    int grayL, binario ;

    grayL=numgray&0x03FF;

    grayL^=(grayL>>8);
    grayL^=(grayL>>4);
    grayL^=(grayL>>2);
    grayL^=(grayL>>1);

    binario=grayL;

    return binario;
}

#####
//          Just Wait          //
#####
void Delay1(int count){
    Uint32 y;
    //2500, 900
    for (y=0;y<count;y++)
        DELAY_US(1000);
}

```

Listing C.3: System.h

```

0  int loop=0;

#####
//          Prototype statements for functions          //
#####
void InitSystem(void);
void Gpio_Select(void);
void init_ADC(void);

10 void powerDOWN(void);

```

Listing C.4: System.c

```

0  #include "System.h"
#include "MRV.h" // Motor Variables

#####
//          Setup the processor          //
#####
void InitSystem(void){
    // Step 1. Initialize System Control:
    // PLL, WatchDog, enable Peripheral Clocks
    InitSysCtrl();

    // Step 2. Initialize GPIO:
    Gpio_Select();
    GpioDataRegs.GPADAT.bit.GPIOA13=1; //Phase C
    GpioDataRegs.GPADAT.bit.GPIOA14=1; //Phase B
    GpioDataRegs.GPADAT.bit.GPIOA15=1; //Phase A

    // Step 3. Clear all interrupts and initialize PIE vector table:
    // Disable CPU interrupts
    DINT;

    // Initialize PIE control registers to their default state.
    // The default state is all PIE interrupts disabled and flags are cleared.
    InitPieCtrl();

    // Disable CPU interrupts and clear all CPU interrupt flags:
    IER = 0x0000;
    IFR = 0x0000;

    // Initialize the PIE vector table with pointers to the shell Interrupt Service Routines (ISR).
    // This will populate the entire table, even if the interrupt is not used.
    // This is useful for debug purposes.
    InitPieVectTable();

    // Interrupts that are used are re-mapped to ISR functions found within this file.
    EALLOW; // This is needed to write to EALLOW protected registers

```

```

PieVectTable.T1PINT = &eva_timer1_isr;
PieVectTable.T3PINT = &evb_timer3_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize all the Device Peripherals:
init_ADC(); //INIT the ADC
init_eva_timer1(); //INIT Timer 1
init_evb_timer3(); //INIT Timer 3

// Enable PIE group 2 interrupt 4 for T1PINT
PieCtrlRegs.PIEIER2.all = M_INT4;

// Enable PIE group 3 interrupt 1 for T2PINT
PieCtrlRegs.PIEIER3.all = M_INT1;

// Enable PIE group 4 interrupt 4 for T3PINT
PieCtrlRegs.PIEIER4.all = M_INT4;

// Enable CPU Interrupts
IER |= (M_INT2 | M_INT4 | M_INT3);

// Enable global Interrupts and higher priority real-time debug events:
EINT; // Enable Global interrupt INTM
ERTM; // Enable Global realtime interrupt DBGM
}

#####
// Initialize ADC hardware
#####
void init_ADC(void){

//Reset Flag
AdcRegs.ADCCTRL1.bit.RESET=1;
DELAY_US(30); // Just wait
AdcRegs.ADCCTRL1.bit.RESET=0;
//Set Reference
AdcRegs.ADCCTRL3.bit.ADCBGRFDN = 0x3; // Power up bandgap/reference circuitry
DELAY_US(30); // Just wait
// Delay before powering up rest of ADC
AdcRegs.ADCCTRL3.bit.ADCPWDN = 1; // Power up rest of ADC
DELAY_US(30); // Just wait
AdcRegs.ADCCTRL3.bit.SMODE_SEL=0; // Sequential sampling mode converted
AdcRegs.ADCCTRL3.bit.ADCCLKPS=1;
/*Core clock divider. 28x Peripheral Clock, HSPCLK, is divided by 2*ADCLKPS[3 - 0],
except when ADCCLKPS[3 - 0] is 0000, in which case HSPCLK is directly passed
on. The divided clock is further divided by ACTRL1[7]+1 to generate the core clock, ADCCLK.*/

AdcRegs.ADCCTRL1.bit.SUSMOD=0; //conversion ignore break points
AdcRegs.ADCCTRL1.bit.ACQ_PS=1; // Acquisition time 4xTclk
AdcRegs.ADCCTRL1.bit.CPS=0; //fclk=CLK/1
AdcRegs.ADCCTRL1.bit.CONT_RUN=0; //Start-stop mode
AdcRegs.ADCCTRL1.bit.SEQ_CASC=1; //Cascaded mode

AdcRegs.ADCCTRL2.bit.EVB_SOC_SEQ=1; //Allows the cascaded mode sequencer to be started by EVB signal
AdcRegs.ADCMAXCONV.bit.MAX_CONV1=0x05; //6 conversions
//Configure ADC Sequence
AdcRegs.ADCCHSELSEQ1.bit.CONV00=0;
AdcRegs.ADCCHSELSEQ1.bit.CONV01=0x3;
AdcRegs.ADCCHSELSEQ1.bit.CONV02=0x6;
AdcRegs.ADCCHSELSEQ1.bit.CONV03=0;
AdcRegs.ADCCHSELSEQ2.bit.CONV04=0x3;
AdcRegs.ADCCHSELSEQ2.bit.CONV05=0x6;

AdcRegs.ADCCTRL2.bit.RST_SEQ1=1; //sequence reset
}

#####
// Set Input, Output and Mode
#####
void Gpio_Select(void)
{
EALLOW;

//SetUP mux PWM for PWM 1-6 GPA e GPB
GpioMuxRegs.GPAMUX.bit.PWM1.GPIOA0=1;
GpioMuxRegs.GPAMUX.bit.PWM2.GPIOA1=1;
GpioMuxRegs.GPAMUX.bit.PWM3.GPIOA2=1;
GpioMuxRegs.GPAMUX.bit.PWM4.GPIOA3=1;
GpioMuxRegs.GPAMUX.bit.PWM5.GPIOA4=1;
GpioMuxRegs.GPAMUX.bit.PWM6.GPIOA5=1;
GpioMuxRegs.GPAMUX.bit.C1TRIP.GPIOA13=0;
GpioMuxRegs.GPAMUX.bit.C2TRIP.GPIOA14=0;
GpioMuxRegs.GPAMUX.bit.C3TRIP.GPIOA15=0;
GpioMuxRegs.GPADIR.bit.GPIOA13=1;
GpioMuxRegs.GPADIR.bit.GPIOA14=1;
GpioMuxRegs.GPADIR.bit.GPIOA15=1;

//SetUP Sensor Position Inputs and Outputs
GpioMuxRegs.GPBMUX.all=0x0000; //Set all as primary function
GpioMuxRegs.GPBDIR.all=0x0000; //Set all as Input
GpioMuxRegs.GPBDIR.bit.GPIOB11=1; //PIN B11 output (reset encoder)

EDIS;
}

#####
// Turn off Motor
#####
void powerDOWN(void){
GpioDataRegs.GPADAT.bit.GPIOA13=1; //Phase C
GpioDataRegs.GPADAT.bit.GPIOA14=1; //Phase B
GpioDataRegs.GPADAT.bit.GPIOA15=1; //Phase A
loop=0;
while(!loop);
}

```

Listing C.5: CurrentControl.h

```

0 //#####
//          Prototype statements for functions          //
//#####
  Uint32 CurrentPI(Uint32 UAnt);
  void setupCurrentPI(int32 Kp, int32 Ki);
  void setCurrentREF(Uint16 REF);

```

Listing C.6: CurrentControl.c

```

0 #include <math.h>
#include "MRV.h" // Motor Variables
#include "CurrentControl.h"

  Uint32 CurrentPI(Uint32 UAnt){
  int32 TMP;
  int32 Coef, int32 Coef1;

  Coef1 =mrv.controller.current.P;
  Coef  = mrv.controller.current.I+mrv.controller.current.P;

10   TMP= UAnt + (Coef*mrv.controller.currentVars.erro)+
      (Coef1*mrv.controller.currentVars.lastErro);

      if (TMP<=0) {
      TMP=0;
      }

      return TMP;
20 }

  void setupCurrentPI(int32 Kp, int32 Ki){
  //Coef trapezoidal struct
  setParam(&mrv.controller.current, Kp, Ki,0); //Coefs ZN
  mrv.controller.currentControl = &CurrentPI(Uint32 UAnt)
  }

  void setCurrentREF(Uint16 REF){
30   mrv.ref.current = REF;
  }

```

Listing C.7: SpeedControl.h

```

0 //#####
//          Prototype statements for functions          //
//#####
  Uint32 VelocidadePID ();
  Uint32 VelocidadeRST ();
  void setupSpeedRST(int isRST);

```

Listing C.8: SpeedControl.c

```

0 #include "SpeedControl.h"
#include <math.h>
#include "MRV.h" // Motor Variables

  Uint32 VelocidadePID(){
  static int32 ErroSum;
  int32 TMP;

  ErroSum+= mrv.controller.speedVars.erro;

10   TMP = ((mrv.controller.speed.P*mrv.controller.speedVars.erro)+
          (mrv.controller.speed.I*ErroSum)+
          (mrv.controller.speed.D*
          (mrv.controller.speedVars.lastErro-mrv.controller.speedVars.erro)));

      if (TMP<=0) {
      TMP=0;
      }

      return TMP;
20 }

  Uint32 VelocidadeRST(){
  int32 TMP;

  TMP = (mrv.controller.speedVars.current+
          ((mrv.controller.speed.P+mrv.controller.speed.I+mrv.controller.speed.D)*
          mrv.ref.speed) - (mrv.controller.speed.P*mrv.controller.speedVars.speed) -
          (mrv.controller.speed.I*mrv.controller.speedVars.lasSpeed) -
          (mrv.controller.speed.D*mrv.controller.speedVars.lasLastSpeed))

30   if (TMP<=0) {
      TMP=0;
      }

      return TMP;

```

```

}
40 void setupSpeedRST(int isRST){
    if (isRST) {
        mrv.controller.currentControl=&VelocidadeRST;
    } else {
        mrv.controller.currentControl=&VelocidadePID;
    }
}
}

```

Listing C.9: TimerControl.h

```

0 //Timer Count
  Uint32 Timer2Count;
  Uint16 VEZ;

  //#####
  //          Prototype statements for functions          //
  //#####

10 interrupt void eva_timer1_isr(void);
   interrupt void evb_timer2_isr(void);
   interrupt void evb_timer3_isr(void);

   void init_eva_timer1(void);
   void init_eva_timer2(void);
   void init_evb_timer3(void);

```

Listing C.10: TimerControl.c

```

0 #include "TimerControl.h"
  #include <math.h>
  #include "CurrentControl.h"
  #include "SpeedControl.h"

  #include "MRV.h" // Motor Variables

  //#####
  //          PWM TIMER INIT          //
  //#####
10 void init_eva_timer1(void)
  {
    // Initialize EVA Timer 1:
    // Setup Timer 1 Registers (EV A)
    EvaRegs.GPTCONA.all = 0;

    // Set the Period for the GP timer 1
    EvaRegs.T1PR = 0x0EA6; // Period
    EvaRegs.T1CMPR = 0x0000; // Compare Reg

20 // Enable Period interrupt bits for GP timer 1
    // Count up, x128, internal clk, enable compare, use own period
    EvaRegs.EVAIMRA.bit.T1PINT = 1;
    EvaRegs.EVAIFRA.bit.T1PINT = 1;

    // Clear the counter for GP timer 1
    EvaRegs.T1CNT = 0x0000;

30 EvaRegs.T1CON.all = 0x1044;

    EvaRegs.ACTRA.all=0x0AA5; //pin1 high, pin 2 low etc
    //PWM 1, 3 e 5 = 1 not Inverted, = 2 Inverted
    EvaRegs.ACTRA.bit.CMP1ACT=1;
    EvaRegs.ACTRA.bit.CMP3ACT=1;
    EvaRegs.ACTRA.bit.CMP5ACT=1;
    EvaRegs.COMCONA.all=0xA600;
    EvaRegs.CMPR1=0x0752;
    EvaRegs.CMPR2=0x0752;
    EvaRegs.CMPR3=0x0752;
40 }

  //#####
  //          Speed TIMER INIT          //
  //#####
  void init_evb_timer2(void){
    // Initialize EVB Timer 2:
    // Setup Timer 2 registers (EV B)
    EvaRegs.GPTCONA.all = 0;

50 // Set the Period for the GP timer 2
    EvaRegs.T2PR = 0x2328; // Period
    EvaRegs.T2CMPR = 0x0000; // Compare Reg

    // Enable Period interrupt bits for GP timer 2
    // Count up, x128, internal clk, enable compare, use own period
    EvaRegs.EVAIMRB.bit.T2PINT = 1;
    EvaRegs.EVAIFRB.bit.T2PINT = 1;

60 // Clear the counter for GP timer 2
    EvaRegs.T2CNT = 0x0000;
    EvaRegs.T2CON.all = 0x1042;

    Timer2Count=0; // scale Var
  }
}

```

```

70 //#####
//          Current TIMER INIT          //
//#####
void init_evb_timer3(void)
{
    // Initialize EVB Timer 3:
    // Setup Timer 3 Registers (EV B)
    EvbRegs.GPTCONB.all = 0;

    // Set the Period for the GP timer 3 to 0x0200;
    EvbRegs.T3PR = 0x0BB8; // Period
80 EvbRegs.T3CMPR = 0x0000; // Compare Reg

    // Enable Period interrupt bits for GP timer 3
    // Count up, x128, internal clk, enable compare, use own period
    EvbRegs.EVBIMRA.bit.T3PINT = 1;
    EvbRegs.EVBIFRA.bit.T3PINT = 1;

    // Clear the counter for GP timer 3
    EvbRegs.T3CNT = 0x0000;
90 EvbRegs.T3CON.all = 0x1042;

    // Start EVA ADC Conversion on timer 3 Period interrupt
    EvbRegs.GPTCONB.bit.T3TOADC = 2;
}

//#####
//          PWM Routine          //
//#####
100 interrupt void eva_timer1_isr(void)
{
    //System StratUP, lock system to start sensor position
    if (mrv.status.lock==0) {
        //Make all togs equal TogForLock
        TogA=TogB=TogC=mrv.controller.currentVars.TogForLock;
    }else{
110 //Calculate tog for especific controller
        mrv.controller.currentVars.TogA=-0.1249*mrv.controller.currentVars.uA+3749;
        mrv.controller.currentVars.TogB=-0.1249*mrv.controller.currentVars.uB+3749;
        mrv.controller.currentVars.TogC=-0.1249*mrv.controller.currentVars.uC+3749;

        //PWM Limits
        if (mrv.controller.currentVars.TogA > 0xEA0) mrv.controller.currentVars.TogA = 0xEA0;
        if (mrv.controller.currentVars.TogA < 0x252) mrv.controller.currentVars.TogA = 0x252;
        if (mrv.controller.currentVars.TogB > 0xEA0) mrv.controller.currentVars.TogB = 0xEA0;
        if (mrv.controller.currentVars.TogB < 0x252) mrv.controller.currentVars.TogB = 0x252;
120 if (mrv.controller.currentVars.TogC > 0xEA0) mrv.controller.currentVars.TogC = 0xEA0;
        if (mrv.controller.currentVars.TogC < 0x252) mrv.controller.currentVars.TogC = 0x252;
    }
    //Set PWM register
    EvaRegs.CMPR1=mrv.controller.currentVars.TogA;
    EvaRegs.CMPR2=mrv.controller.currentVars.TogB;
    EvaRegs.CMPR3=mrv.controller.currentVars.TogC;

    // Enable more interrupts from this timer
    EvaRegs.EVAIMRA.bit.T1PINT = 1;
130 // Note: To be safe, use a mask value to write to the entire
    // EVAIFRA register. Writing to one bit will cause a read-modify-write
    // operation that may have the result of writing 1's to clear
    // bits other than those intended.
    EvaRegs.EVAIFRA.all = BIT7;

    // Acknowledge interrupt to receive more interrupts from PIE group 2
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP2;
}

//#####
//          Speed Routine          //
//#####
140 interrupt void eva_timer2_isr(void){
    extern void CheckAngle(void);

    CheckAngle();

    static Uint16 difAngleSum=0;
    static VEZ=0;
150 Uint16 difAngle;
    Uint16 TMP;

    // Verify prescale for speed calculation
    if (++Timer2Count>=20) {
        VEZ++;

        if (mrv.status.angle>=mrv.status.lastAngle) {
            // Make sure that the motor have not completed the cicle
160 if ((mrv.status.angle>300) && ((mrv.status.lastAngle>=0) && (mrv.status.lastAngle<60))) {
                DifAngulo=360-mrv.status.angle+mrv.status.lastAngle;
            }else {
                DifAngulo=mrv.status.angle-mrv.status.lastAngle;
            }
        }else {
            // Make sure that the motor have not completed the cicle
            if ((mrv.status.lastAngle>300) && ((mrv.status.angle>=0) && (mrv.status.angle<60))) {
                DifAngulo=360-mrv.status.lastAngle+mrv.status.angle;
            }else {
170 DifAngulo=mrv.status.lastAngle-mrv.status.angle;
            }
        }
    }
    // Save Angle
    mrv.status.lastAngle=mrv.status.angle;
}

```

```

//Clear prescale
Timer2Count=0;
}

180 if (VEZ>=5) {
    VEZ=0;
    mrv.controller.speedVars.lastSpeed=
    mrv.controller.speedVars.lastLastSpeed;
    mrv.controller.speedVars.lastSpeed=
    mrv.controller.speedVars.speed;
    //Speed= ((100000*(difAngleSum))/(120*6*Periodo*20));
    mrv.controller.speedVars.speed=
190 ((625*(difAngleSum))/(9*5));
    difAngleSum=0;

    if (mrv.controller.speedVars.speed<0)
        mrv.controller.speedVars.speed= -(mrv.controller.speedVars.speed);

    mrv.controller.speedVars.lastLastErro=mrv.controller.speedVars.lastErro;
    mrv.controller.speedVars.lastErro=mrv.controller.speedVars.erro;
    mrv.controller.speedVars.erro= mrv.ref.speed-mrv.controller.speedVars.speed;

200 //MQR();

    mrv.controller.speedVars.lastCurrent = mrv.controller.speedVars.current;

    mrv.controller.speedVars.current=mrv.controller.speedControl();

    if (mrv.controller.speedVars.current>60000) mrv.controller.speedVars.current=60000;
    //CCorrente -> 60.000 - 600
    // X - TMP
210 TMP = (0.01*mrv.controller.speedVars.current);

    setCurrentREF(TMP);

    vSpeed[countV]=mrv.controller.speedVars.speed;
    vSpeedControl[countV+]=mrv.controller.speedVars.current;

}

220 // Note: To be safe, use a mask value to write to the entire
// EVAIFRB register. Writing to one bit will cause a read-modify-write
// operation that may have the result of writing 1's to clear
// bits other than those intended.
EvaRegs.EVAIFRB.all = BIT0;

// Acknowledge interrupt to receive more interrupts from PIE group 3
PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;

230 }

#####
// Current Routine //
#####
interrupt void evb_timer3_isr(void)
{
240 extern void CheckAngle(void);

    Uint16 currentTMP1, currentTMP2;
    Uint16 current1, current2, current3;

    //Phase A Current
    currentTMP1=(AdcRegs.ADCRESULT0>>4);
    currentTMP2=(AdcRegs.ADCRESULT3>>4);
    current1=(currentTMP1+currentTMP2)/2;
    current1=current1*0.22075; //max a/d 6v
250 mrv.status.currentA=current1;

    //Phase B Current
    currentTMP1=(AdcRegs.ADCRESULT1>>4);
    currentTMP2=(AdcRegs.ADCRESULT4>>4);
    current2=(currentTMP1+currentTMP2)/2;
    current2=current2*0.22075; //max a/d 6v
    mrv.status.currentB=current2;

    //Phase C Current
260 currentTMP1=(AdcRegs.ADCRESULT2>>4);
    currentTMP2=(AdcRegs.ADCRESULT5>>4);
    current3=(currentTMP1+currentTMP2)/2;
    current3=current3*0.22075; //max a/d 6v
    mrv.status.currentC=current3;

    CheckAngle();

270 //System lock phase
    if(mrv.status.lock==0){
        //Just with locked phase
        switch (mrv.controller.WPhase) {
            case 0:
                if (mrv.status.currentA>=mrv.status.REFLock) {
                    GpioDataRegs.GPADAT.bit.GPIOA15=1;
                }else {
                    GpioDataRegs.GPADAT.bit.GPIOA15=0;
                }
            break;
            case 1:
                if (mrv.status.currentB>=mrv.status.REFLock) {
                    GpioDataRegs.GPADAT.bit.GPIOA14=1;
                }else {
280 }

```

```

        GpioDataRegs.GPADAT.bit.GPIOA14=0;
    }
    break;
    case 2:
        if (mrv.status.currentC>=mrv.status.REFLock) {
290         GpioDataRegs.GPADAT.bit.GPIOA13=1;
        }else {
            GpioDataRegs.GPADAT.bit.GPIOA13=0;
        }
        break;
    }else {
        if (mrv.status.phase==0) {
300         mrv.controller.currentVars.error=
            mrv.status.current -mrv.status.currentA;
            mrv.controller.currentVars.uA=
                mrv.controller.currentControl(mrv.controller.currentVars.lastUA);
            mrv.controller.currentVars.lastUA = mrv.controller.currentVars.uA;
        }else if (mrv.status.phase==1){
            mrv.controller.currentVars.error=
            mrv.status.current -mrv.status.currentB;
            mrv.controller.currentVars.uB=
            mrv.controller.currentControl(mrv.controller.currentVars.lastUB);
            mrv.controller.currentVars.lastUB = mrv.controller.currentVars.uB;
310         }else if (mrv.status.phase==2) {
            mrv.controller.currentVars.error=
            mrv.status.current -mrv.status.currentC;
            mrv.controller.currentVars.uC=
            mrv.controller.currentControl(mrv.controller.currentVars.lastUC);
            mrv.controller.currentVars.lastUC = mrv.controller.currentVars.uC;
        }
        mrv.controller.currentVars.lastLastErro=mrv.controller.currentVars.lastErro;
        mrv.controller.currentVars.lastErro=mrv.controller.currentVars.error;
    }
}

320 if (countC<1000){
    vCurrent[0][countC]=current1;
    vCurrent[1][countC]=current2;
    vCurrent[2][countC]=current3;
    vCurrentControl[countC++]=mrv.controller.currentVars.uA;
}

    AdcRegs.ADCTRL2.bit.RST_SEQ1=1;

330 // Note: To be safe, use a mask value to write to the entire
// EVBIFRA register. Writing to one bit will cause a read-modify-write
// operation that may have the result of writing 1's to clear
// bits other than those intended.
EvbRegs.EVBIFRA.all = BIT7;

// Acknowledge interrupt to receive more interrupts from PIE group 4
PieCtrlRegs.PIEACK.all = PIEACK_GROUP4;
}

```

# Referências Bibliográficas

- Aguirre, L. (2007). *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora da UFMG, MG, 3ª edição.
- Åström, K. e Hägglund, T. (1986). *Advanced PID Control*. ISA-Instrumentation, Systems, and Automation Society, USA.
- Åström, K. e Hägglund, T. (1995). *Adaptive Control*. Addison-Wesley, New York.
- Åström, K. e Hägglund, T. (1999). *PID Controllers: Theory, Design, and Tuning*. ISA-Instrumentation, Systems, and Automation Society, 2nd edition, USA.
- Åström, K. e Wittenmark, B. (1998). *Computer-Controlled Systems - Theory and Design*. Prentice-Hall, Upper Saddle River, NJ.
- Bae, H., Lee, B., Vijayraghavan, P., e Krishnan, R. (2000). A linear switched reluctance motor: Converter and control. *IEEE Trans. on Ind. Appl.*, Vol. 36, No. 5, pp. 1351–1359.
- Barnes, M. e Pollock, C. (1998). Power electronic converters for switched reluctance drives. *IEEE Transactions on Power Electronics*, Vol. 13, No. 6, pp. 1100–1111.
- Bass, J., Ehsani, M., e Miller, T. (1986). Robust torque control of switched reluctance motors without a shaft-position sensor. *IEEE Transactions on Industry Electronics*, Vol. 33, No. 3, pp. 212–216.
- BASTOS, G. (2006). *Retificador Monofásico com Alto Fator de Potência com Conexão Comum entre Entrada e Saída*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Bianco, C., Tonielli, A., e Filicori, F. (1996). A prototype controller for variable reluctance motors. *IEEE Tans. on Power Electron.*, Vol. 43pp. 207–216.



- Blaabjerg, F., Kjaer, P., Rasmussen, P., e Cossar, C. (1999). Improved digital current control methods in switched reluctance motor. *IEEE Trans. on Power Electron.*, Vol. 14, No. 3, pp. 563–672.
- Chien, I. e Fruehalf, P. (1990). Consider imc tuning to improve performance. *Chemical Engineering Progress*, Vol.pp. 133–141.
- GAO, Y. (2000). *Speed Control of Switched Reluctance Motors*. Tese de Doutorado, The Hong Kong University of Science and Technology, Hong Kong.
- Harris, W. e Miller, T. (1989). Comparison of design and performance parameter in switched reluctance and induction motors. *Fourth Int. Conf. on Electrical Machines and Drives - ICMD*, Vol.pp. 303–307.
- Henriques, L. (2004). *Implementação de Estratégia de Minimização de Oscilações de Torque e Remoção de Sensor de Posição para um Acionamento de Relutância Variável usando Técnica Neuro-fuzzy*. Tese de Doutorado, Universidade Federal do Rio de Janeiro, UFRJ, Brasil.
- Hwu, L. e Liaw, C. (2000). Dc-link voltage boosting and switching control for switched reluctance motor drives. *IEE Proceedings Electric Power Applications*, Vol.pp. 337–344.
- Ilic-Spong, M., Marino, R., Persada, S., e Taylor, D. (1987). Feedback linearization control of switched motors. *IEEE Trans. on Autom. Control*, Vol. AC-32, No. 5, pp. 371–379.
- Khalil, H. (1996). *Nonlinear Systems*. Prentice Hall, London.
- Krishnan, R. (2001). *Switched Reluctance Motor Drives - Modeling, Simulation, Analysis, Design, and Applications*. CRC Press, Industrial Electronics Series.
- Le-Huy, H., Slimani, K., e Viarouge, P. (1991). A current-controlled quasi-resonant converter for switched-reluctance motor. *IEEE Trans. on Ind. Electron.*, Vol. 38, No. 4, pp. 355–362.
- Lima, R. (2005). *Acionamento de uma Máquina à Relutância Variável*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Luo, R., Qin, S. J., , e Chen, D. (1998). A new approach to closed loop auto-tuning for pid controllers. *Proceedings of the American Control Conference*, Vol.pp. 348–352.

- Luyben, W. (1996). Tuning proportional-integral-derivative controllers for integrator/deadtime processes. *Ind. Eng. Chem.*, Vol. 35pp. 3480–3483.
- Meister, D. (2006). *Aplicação do Método dos Mínimos Quadrados na Estimação de Parâmetros do Modelo de um Transformador*. Dissertação de Mestrado, Universidade de Brasília, UnB, Brasil.
- Melo, N. (2007). *Conversor Boost para a Mitigação de Afundamentos de Tensão em Acionamentos de Velocidade Variável*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Miller, T. (1993). *Switched Reluctance Motors and Their Control*. Magna Physics publishing and Clarendon Press Oxford, Oxford, NY, first edição.
- Miller, T. (2001). *Electronic Control os Switched Reluctance Machines*. Newnes.
- Neto, A. H. (2005). *Técnicas Anti-Windup Em Estruturas De Controle PID, RST E GPC*. Dissertação de Mestrado, Universidade Federal de Santa Catarina, UFSC, Brasil.
- Nicolai, J. (1995). Simplified electronics bring the switched reluctance motor to the mass market. *Proc. IEEE Proceedings Power Electronics Specialists Conference - PESC*, pp. 3903–3907.
- Oliveira, A., Lima, A., e Jacobina, C. (2002). Sistema de acionamento de mrc usando regulador de corrente e estratégia de fluxo corrente para comutação eletrônica de corrente entre fases. *Proc. Congresso Brasileiro de Automática - CBA*, pp. 3145–3150, Natal, RN.
- Oliveira, D. (2008). *Proposta e implementação de sistema para caracterização eletromecânica de máquinas linear e rotativa de relutância chaveada*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Oliveira, D., Almeida, R., Teixeira, V., Aguiar, V., Cunha, H., Reis, L., e Pontes, R. (2006). Método iterativo para cálculo da variação da indutância do motor rotativo de relutância chaveada. *IX - Encontro de Modelagem Computacional*, Vol. 1.
- Padhy, P. e Majhia, S. (2006). Simulação do controle de posição de um motor de indução trifásico utilizando controle vetorial indireto. *Computers & Chemical Engineering*, Vol. 30, No. 15, pp. 790–796.

- Ray, W. F., Lawrenson, P., Davis, R., Stephenson, J., Foulton, N., e R.J.Blake (1986). High-performance switched reluctance brushless drives. *IEEE Trans. on Ind. Appl.*, Vol. IA-22, No. 4, pp. 722–729.
- Reis, L. (2008). *Modelagem e Controle Preditivo Aplicado a um Sistema de Acionamento com Motor de Relutância Variável*. Tese de Doutorado, Universidade Federal de Santa Catarina, UFSC, Brasil.
- Reis, L., Coelho, A., Almeida, O., Campos, J., e Almeida, R. (2007). Current control of switched reluctance motor based on generalized minimum variance controller. *ACC - American Control Conference*, Vol. 1pp. 3541–3545.
- Reis, L., Coelho, A., Almeida, O., Sobreira, F., e Almeida, R. (2008a). Performance assessment of pi and imc controller design based on setpoint relay for a switched reluctance motor drive. *Induscon - Conferência Internacional de Aplicações Industriais*, Vol. 1, No. 41141, pp. 64.
- Reis, L., Coelho, A., Almeida, O., Sobreira, F., e Braga, A. (2008b). A practical method for the current control of switched reluctance motor drive. *Industrial Electronics Magazine*, Vol. 1.
- Rim, G., Kim, W., e Cho, J. (2001). An improved sequential hysteresis current controller for three-phase inverter: design and hardware implementation. *Proc. Proceedings of the 2001 IEEE International Conference on Control Applications*, pp. 294–300, México City.
- Rômulo, N. C. A. (2007). *Estratégias de Controle de Corrente para uma Máquina à Relutância Variável*. Dissertação de Mestrado, PGEEL / UFC, Fortaleza, CE.
- Salamanca, H. (2007). *Análise, Sintomnia e Experimentação de controladores PID via Projetos GMV e GPC*. Dissertação de Mestrado, Universidade Federal de Santa Catarina, UFSC, Brasil.
- Saraiva, R. (2006). *Autosintonia de Controladores PI para Malha de Corrente de uma Máquina à Relutância Variável*. Dissertação de Mestrado, PGEEL / UFC, Fortaleza, CE.

- Schulz, S. e Rahuman, K. M. (2004). High performance current control for switched reluctance motors with on-line modeling. *European Conference on Power Electronics and Applications*, Vol. CD-ROM Paper No.1246-12076.
- Sobreira, F., Reis, L., Almeida, O., e Braga, A. (2008). Sintonia baseada em algoritmos genéticos de controlador pi aplicado a motores de relutância variável. *CBA - Congresso Brasileiro de Automática*, Vol. 1.
- Souza, K. (2003). *Conversor CC-CA monofásico para interligar painéis fotovoltaicos ao sistema elétrico*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Sung-Jun, P., Jin-Woo, A., Man-Hyung, L., e Lipo, T. (2001). Novel encoder for srm drive with high resolution angle control. *Proc. Proc. Conf. Rec. of the IEEE Int. Symp. on Ind. Electron. - ISIE*, Vol. 3, pp. 1781–1785.
- Takatsu, H. e Itoh, T. (1999). Future needs for control theory in industry - report of the control technology survey in japanese industry. *IEEE Transaction on Control Systems Technology*, Vol. 37, No. 3, pp. 298–305.
- Tanaka, K. (2001). *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley.
- Teixeira, V. (2008). *Projeto de Motores a Relutância Variável e Ferramenta Computacional para Determinação das Características Estáticas da Máquina*. Dissertação de Mestrado, Universidade Federal do Ceará, UFC, Brasil.
- Tormey, D. e Torrey, D. (1991). A comprehensive design procedure for low torque-ripple variable-reluctance motor drives. *Proc. IEEE Proc. Ind. Appl. Society Ann. Meeting - IAS*, pp. 886–891.
- Vukosavic, S. e Stefanovic, V. (1991). SRM inverter topologies: A comparative evaluation. *IEEE Trans. on Ind. Appl.*, Vol. 27, No. 6, pp. 1034–1047.
- Wellstead, P. e Zarrop, M. (1991). *Self-Tuning Systems. Control and Signal Processing*. John Wiley & Sons, New York.
- Ziegler, J. e Nichols, N. (1943). Optimum settings for automatic controllers. *Transactions ASME*, Vol. 65pp. 433–444.