



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE SOBRAL**  
**CURSO DE ENGENHARIA ELÉTRICA**

**PAULO ROBSON MELO COSTA**

**CONTROLE DE POSIÇÃO E VELOCIDADE DO MOTOR DE INDUÇÃO  
TRIFÁSICO POR ORIENTAÇÃO DO CAMPO BASEADO EM DSP**

**SOBRAL**

**2016**

PAULO ROBSON MELO COSTA

CONTROLE DE POSIÇÃO E VELOCIDADE DO MOTOR DE INDUÇÃO TRIFÁSICO  
POR ORIENTAÇÃO DO CAMPO BASEADO EM DSP

Trabalho de Conclusão de Curso apresentado à coordenação do curso de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de graduado em Engenharia Elétrica. Área de concentração: Máquinas Elétricas, Eletrônica de Potência e Microprocessadores.

Orientador: Prof. Dr. Marcus Rogério de Castro.

SOBRAL

2016

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do *Campus* de Sobral

- 
- C872c Costa, Paulo Robson Melo.  
Controle de posição e velocidade do motor de indução trifásico por orientação do campo baseado em DSP. / Paulo Robson Melo Costa. – 2016.  
124 f. : il., enc. ; 30 cm.
- Monografia (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia Elétrica, Sobral, 2016.
- Orientação: Prof. Dr. Marcus Rogério de Castro.
1. Máquinas Elétricas. 2. Eletrônica de Potência. 3. Microprocessadores I. Título.

---

CDD 621.3

PAULO ROBSON MELO COSTA

CONTROLE DE POSIÇÃO E VELOCIDADE DO MOTOR DE INDUÇÃO TRIFÁSICO  
POR ORIENTAÇÃO DO CAMPO BASEADO EM DSP

Trabalho de Conclusão de Curso apresentado à coordenação do curso de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de graduado em Engenharia Elétrica. Área de concentração: Máquinas Elétricas, Eletrônica de Potência e Microprocessadores.

Aprovado em: 04/02/2016

BANCA EXAMINADORA

---

Prof. Dr. Marcus Rogério de Castro (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Éber de Castro Diniz  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Rômulo Nunes de C. Almeida  
Universidade Federal do Ceará (UFC)

A Deus.

Aos meus pais, Regina e Meirizardo.

## **AGRADECIMENTOS**

Primeiramente a Deus, por tornar nossas conquistas expressão fiel da Sua vontade, és para mim guarda, guia e adjuntor.

À minha família querida, por todo amor, suporte e sacrifício que fizeram para que eu chegasse até aqui. Em especial aos meus pais Meirizado e Regina, que são os grandes incentivadores e responsáveis por minhas conquistas e ao meu irmão Samuel por ser meu melhor amigo e companheiro todos esses anos.

Ao Professor Marcus Rogério, pela confiança, paciência e orientação durante a realização deste e de tantos outros trabalhos durante toda a minha passagem pela graduação.

Aos professores Isaac Machado, Vandilberto Pinto e Leonardo Tabosa pelo suporte durante todo o desenvolvimento deste trabalho no Laboratório de Máquinas Elétricas.

À todos os professores do curso de Engenharia Elétrica que fizeram parte essencial dessa jornada e que contribuíram para o meu desenvolvimeto pessoal e acadêmico, em especial os professores Eber Diniz, Rômulo Nunes, participantes de banca examinadora, e André Lima, pelos ensinamentos e desafios propostos.

À minha querida namorada Débora Aguiar que de todas as formas está disposta a me ajudar e contribuir para o meu sucesso. É um prazer inenarrável tê-la ao meu lado.

Aos meus grandes amigos feitos durante a faculdade os quais deixaram meus dias mais alegres e divertidos, em destaque a Johnantan Santos, William de Sousa e Artur Melo. Amizades para toda vida.

Aos colegas e orientadores do estágio na Coelce, especialmente a Glicélia Mendonça e ao Paulo Ricardo, pessoas que tive a oportunidade e o prazer de conhecer.

*“Mas aqueles que esperam no Senhor renovam as suas forças. Voam alto como águias, correm e não ficam exaustos, andam e não se cansam.”*

*Isaías 40:31*

## RESUMO

Neste trabalho, um sistema de controle vetorial do motor de indução é desenvolvido baseado em um conversor de topologia *back-to-back* e processado por um DSP (Processador Digital de Sinais). Pelo seu menor custo de fabricação e de manutenção e pelo avanço da eletrônica de potência, os motores de indução são a força de trabalho da indústria no mundo todo, nisso se dá a importância desse trabalho, controlar velocidade e posição desses motores possibilitando as mais variadas aplicações. A representação em quadratura da máquina de indução é utilizada para se chegar em um modelo mais simples e mais facilmente controlável e computável do que o modelo trifásico. Nesta representação, e fazendo algumas condições, nos permite alinhar o fluxo do rotor de forma que seja possível controlar torque e fluxo no motor de forma independente, por isso chamado de controle por campo orientado. Esta estratégia de controle é aplicada na malha de corrente do motor de indução. Resultados de simulação são apresentados para diversas referências de velocidade e posição, com entrada e saída de carga, analisando diversas grandezas relacionadas e comprovando o controle. Os mesmos ensaios realizados em simulação, também foram desenvolvidos experimentalmente e analisados. O controlador Proporcional-Integral (PI) foi utilizado nas malhas de posição, velocidade e corrente e dimensionado por alocação de polos. É feita uma comparação entre dois métodos de controle de corrente, histerese e por PWM senoidal com controladores PI. O DSP nos forneceu a capacidade de processamento necessária para realizar esse tipo de controle na prática, com ele foi possível chegar a um tempo de amostragem de 5  $\mu$ s.

**Palavras-chave:** Motor de indução. Controle vetorial. DSP. Inversor trifásico. Histerese. PWM. PI.



## ABSTRACT

In this work, a vector control system of induction motor is developed based on a converter back-to-back topology and processed by a DSP (Digital Signal Processor). For its lower manufacturing costs and maintenance and the power electronics advance, induction motors are the industry's workforce worldwide, it is given the importance of this work, control speed and position of these engines enabling the most varied applications. The quadrature representation of the induction machine is used to arrive at a simplified and more easily controllable and computable than the three-phase model. In this representation, and doing some conditions, allows us to align the rotor flux so that you can control torque and flux in independently way, so called field-oriented control. This control strategy is applied to the mesh induction motor current. Simulation results are presented for different speed references and position, with addition and load removal, analyzing various related quantities and verifying control. The same trials in simulation, have also been developed experimentally and analyzed. Proportional-Integral (PI) controller was used in the mesh position, speed and current and scaled by pole placement. A comparison between two current control methods, hysteresis and sinusoidal PWM with PI controllers is made. The DSP has provided us the processing power needed to perform this type of control in practice, it was possible to reach a sampling time of 5  $\mu$ s.

**Keywords:** Induction motor. Vector control. DSP. Three-phase inverter. Hysteresis. PWM. IP.

## LISTA DE FIGURAS

Figura 2.1 – Características construtivas do estator (a) e do rotor (b) do motor de indução.....	25
Figura 2.2 – Modelo de regime permanente da máquina de indução.....	26
Figura 2.3 – Representação da máquina simétrica trifásica.....	27
Figura 2.4 – Relação entre as grandezas $dq$ e $abc$ na notação para motor.....	31
Figura 2.5 – Representação dos eixos $dq$ e $\alpha\beta$ .....	32
Figura 2.6 – Desacoplamentos das correntes de eixo direto e em quadratura da corrente estatórica.....	35
Figura 2.7 – Diagrama esquemático do controle por campo orientado indireto.....	37
Figura 2.8 – Diagrama esquemático do conversor utilizado para controlar a corrente..	38
Figura 2.9 – Modelo de controlador de corrente para motor trifásico.....	39
Figura 2.10 – Controlador com comparador de histerese.....	40
Figura 2.11 – Controlador com comparador de histerese monofásico simulado no PSIM.....	40
Figura 2.12 – Corrente de referência e corrente na carga controlada por histerese.....	41
Figura 2.13 – Controlador PI síncrono.....	42
Figura 3.1 – Acoplamento motor e freio de Foucault.....	43
Figura 3.2 – Gráfico da potência consumida pelo motor a vazio em função da tensão de alimentação.....	45
Figura 3.3 – Gráfico da desaceleração do motor, $\omega(t)$ , obtido com o DSP e o CCS.....	51
Figura 3.4 – Resposta do torque desenvolvido pelo motor em função da velocidade do rotor.....	52
Figura 3.5 – Resposta da potência do motor em função da velocidade do rotor.....	53
Figura 3.6 – Corrente do estator em função da velocidade do rotor.....	53
Figura 3.7 – Eficiência do motor em função da velocidade do rotor.....	53

Figura 3.8 – Tensão de alimentação.....	54
Figura 3.9 – Corrente no estator.....	54
Figura 3.10 – Torque desenvolvido.....	55
Figura 3.11 – Velocidade do rotor.....	55
Figura 4.1 – Sistema de controle a malha fechada de um controlador.....	57
Figura 4.2 – Resposta da malha de velocidade a um degrau unitário com o controlador (vermelho) e sem controlador (azul tracejada).....	66
Figura 4.3 – Lugar das raízes da malha de velocidade.....	66
Figura 4.4 – Resposta da malha de corrente a um degrau unitário com o controlador (vermelho) e sem controlador (azul tracejada).....	68
Figura 4.5 – Lugar das raízes da malha de corrente.....	68
Figura 4.6 – Resposta da malha de posição a um degrau unitário com o controlador (vermelho) e sem controlador(azul tracejada).....	70
Figura 4.7 – Lugar das raízes da malha de posição.....	70
Figura 5.1 – Diagrama esquemático do controle de velocidade por campo orientado indireto.....	72
Figura 5.2 – Diagrama esquemático do controle de posição por campo orientado indireto.....	72
Figura 5.3 – Bloco do Comando de torque.....	73
Figura 5.4 – Bloco do comando de fluxo.....	73
Figura 5.5 – Bloco do Estimador da posição do fluxo do rotor.....	74
Figura 5.6 – Velocidade de referência no teste a vazio.....	74
Figura 5.7 – Carga mecânica no teste a vazio.....	74
Figura 5.8 – Velocidade angular do rotor no teste a vazio.....	75
Figura 5.9 – Tensão de fase do estator no teste a vazio.....	75
Figura 5.10 – Corrente do estator no teste a vazio.....	76
Figura 5.11 – Conjugado eletromagnético desenvolvido no teste a vazio.....	76

Figura 5.12 – Fluxo do rotor no teste a vazio.....	77
Figura 5.13 – Velocidade de referência no teste com carga.....	77
Figura 5.14 – Carga mecânica no teste com carga.....	78
Figura 5.15 – Velocidade angular do rotor no teste com carga.....	78
Figura 5.16 – Tensão de fase do estator no teste com carga.....	79
Figura 5.17 – Corrente do estator no teste com carga.....	79
Figura 5.18 – Conjugado eletromagnético desenvolvido no teste com carga.....	80
Figura 5.19 – Fluxo do rotor no teste com carga.....	80
Figura 5.20 – Posição de referência (tracejado) e posição do rotor (contínua).....	81
Figura 5.21 – Carga mecânica no teste de controle de posição.....	81
Figura 5.22 – Corrente do estator no teste de controle de posição.....	81
Figura 6.1 – Bancada <i>back-to-back</i> utilizada na implementação prática.....	82
Figura 6.2 – Pré-carga: retificador monofásico ligado ao barramento CC.....	83
Figura 6.3 – Esquemático do conversor 1.....	83
Figura 6.4 – <i>Encoder</i> (b) e os dois sinais gerados por ele (a).....	84
Figura 6.5 – DSP TI F28335.....	85
Figura 6.6 – Fluxograma do programa.....	86
Figura 6.7 – Velocidade do rotor (wbm), velocidade de referência (wbm*) e conjugado eletromagnético de referência (Tem*) no teste a vazio.....	88
Figura 6.8 – Correntes de referência $id_{se}^*$ ( $id_{se}^*$ ) e $iq_{se}^*$ ( $iq_{se}^*$ ) no teste a vazio.....	88
Figura 6.9 – Correntes trifásicas de linha no teste a vazio.....	89
Figura 6.10 – Velocidade do rotor (wbm), velocidade de referência (wbm*) e conjugado eletromagnético de referência (Tem*) no teste com carga.....	90
Figura 6.11 – Correntes trifásicas de linha no teste com carga.....	91
Figura 6.12 – Posição do eixo do motor ( $\theta$ ) e posição de referência ( $\theta^*$ ).....	92

Figura 6.13 – Velocidade do rotor (wbm) durante o controle de posição.....	92
Figura 6.14 – Correntes trifásicas de linha no teste de controle de posição.....	93

## LISTA DE TABELAS

Tabela 3.1	– Dados de placa do motor.....	44
Tabela 3.2	– Dados obtidos no ensaio a vazio.....	44
Tabela 3.3	• Dados obtidos no ensaio a vazio variando a tensão.....	45
Tabela 3.4	• Dados obtidos no ensaio de rotor bloqueado.....	46
Tabela 3.5	• Resumo dos parâmetros elétricos do motor.....	48
Tabela 3.6	• Tempo de desaceleração do rotor.....	49
Tabela 3.7	• Resumo dos parâmetros mecânicos do conjunto motor mais disco do freio.....	51
Tabela 4.1	• Valores dos ganhos do controlador PI da malha de velocidade.....	65
Tabela 4.2	• Valores dos ganhos do controlador PI da malha de corrente.....	67
Tabela 4.3	• Valores dos ganhos do controlador PI da malha de posição.....	69

## **LISTA DE ABREVIATURAS E SIGLAS**

CA	Corrente Alternada
CC	Corrente Contínua
DSP	Digital Signal Processor (Processador Digital de Sinais)
FTMA	Função de Transferência de Malha Aberta
FTMF	Função de Transferência de Malha Fechada
GPIO	General Purpose Input Output (Entrada e Saída de Propósitos Gerais)
MIT	Motor de Indução Trifásico
PI	Controlador Proporcional e Integral
PWM	Pulse Width Modulation (Modulação por Largura de Pulso)
SPWM	Sin Pulse Width Modulation (Modulação por Largura de Pulso Senoidal)
SVPWM	Space Vector Pulse Width Modulation (Modulação por Largura de Pulso por Espaços Vetoriais)

## LISTA DE SÍMBOLOS

$\Phi_r$	Fluxo produzido pelo rotor
$\Phi_s$	Fluxo produzido pelo estator
$\Phi_{total}$	Fluxo total no entreferro
$I_a^*$	Comando de corrente da fase A
$I_b^*$	Comando de corrente da fase B
$I_{bl}$	Corrente no ensaio de rotor bloqueado
$I_c^*$	Comando de corrente da fase C
$I_r$	Corrente no rotor referenciada ao estator
$I_s$	Corrente no estator
$I_{vz}$	Corrente no ensaio a vazio
$I_\phi$	Corrente de magnetização
$K_i$	Ganho integral do controlador PI
$Ki_z$	Ganho integral do controlador PI discreto
$K_p$	Ganho proporcional do controlador PI
$Kp_z$	Ganho proporcional do controlador PI discreto
$L_{lr}$	Indutância Rotórica de dispersão por fase referenciada ao estator
$L_{ls}$	Indutância Estatórica de dispersão por fase
$L_m$	Indutância de magnetização por fase
$P_{bl}$	Potência no ensaio de rotor bloqueado
$P_{vz}$	Potência no ensaio a vazio
$R_c$	Resistência que representa as perdas no núcleo magnético
$R_r$	Resistência Rotórica por fase referenciada ao estator
$R_s$	Resistência Estatórica por fase
$T_a$	Tempo de amostragem
$T_c$	Torque de carga
$T_e$	Conjugado eletromagnético
$V_{ag}$	Tensão entre fase e o centro estrela
$V_{ar}$	Tensão na fase A do rotor
$V_{as}$	Tensão na fase A do estator
$V_{bl}$	Tensão no ensaio de rotor bloqueado
$V_{br}$	Tensão na fase B do rotor



$V_{bs}$	Tensão na fase B do estator
$V_{cr}$	Tensão na fase C do rotor
$V_{cs}$	Tensão na fase C do estator
$V_{vz}$	Tensão no ensaio a vazio
$Z_{th}$	Impedância quivalente de Thevenin
$i_{ar}$	Corrente na fase A do rotor
$i_{as}$	Corrente na fase A do estator
$i_{br}$	Corrente na fase B do rotor
$i_{bs}$	Corrente na fase B do estator
$i_{cr}$	Corrente na fase C do rotor
$i_{cs}$	Corrente na fase C do estator
$i_{dr}^e$	Corrente do rotor no eixo $d$ no sistema de referência síncrono
$i_{dr}^s$	Corrente do rotor no eixo $d$ no sistema de referência estacionário
$i_{ds}^e$	Corrente do estator no eixo $d$ no sistema de referência síncrono
$i_{ds}^s$	Corrente do estator no eixo $d$ no sistema de referência estacionário
$i_{qr}^e$	Corrente do rotor no eixo $q$ no sistema de referência síncrono
$i_{qr}^s$	Corrente do rotor no eixo $q$ no sistema de referência estacionário
$i_{qs}^e$	Corrente do estator no eixo $q$ no sistema de referência síncrono
$i_{qs}^s$	Corrente do estator no eixo $q$ no sistema de referência estacionário
$n_s$	Velocidade síncrona em RPM
$t_s$	Tempo de acomodação
$v_{dr}^e$	Tensão do rotor no eixo $d$ no sistema de referência síncrono
$v_{dr}^s$	Tensão do rotor no eixo $d$ no sistema de referência estacionário
$v_{ds}^e$	Tensão do estator no eixo $d$ no sistema de referência síncrono
$v_{ds}^s$	Tensão do estator no eixo $d$ no sistema de referência estacionário
$v_{qr}^e$	Tensão do rotor no eixo $q$ no sistema de referência síncrono
$v_{qr}^s$	Tensão do rotor no eixo $q$ no sistema de referência estacionário
$v_{qs}^e$	Tensão do estator no eixo $q$ no sistema de referência síncrono
$v_{qs}^s$	Tensão do estator no eixo $q$ no sistema de referência estacionário
$\delta_r$	Constante de tempo rotórica
$\theta_e$	Posição angular do fluxo do rotor
$\theta_r$	Posição angular do eixo do motor

$\lambda_{ar}$	Fluxo na fase A do rotor
$\lambda_{as}$	Fluxo na fase A do estator
$\lambda_{br}$	Fluxo na fase B do rotor
$\lambda_{bs}$	Fluxo na fase B do estator
$\lambda_{cr}$	Fluxo na fase C do rotor
$\lambda_{cs}$	Fluxo na fase C do estator
$\lambda_{dr}^e$	Fluxo do rotor no eixo $d$ no sistema de referência síncrono
$\lambda_{dr}^s$	Fluxo do rotor no eixo $d$ no sistema de referência estacionário
$\lambda_{ds}^e$	Fluxo do estator no eixo $d$ no sistema de referência síncrono
$\lambda_{ds}^s$	Fluxo do estator no eixo $d$ no sistema de referência estacionário
$\lambda_{qr}^e$	Fluxo do rotor no eixo $q$ no sistema de referência síncrono
$\lambda_{qr}^s$	Fluxo do rotor no eixo $q$ no sistema de referência estacionário
$\lambda_{qs}^e$	Fluxo do estator no eixo $q$ no sistema de referência síncrono
$\lambda_{qs}^s$	Fluxo do estator no eixo $q$ no sistema de referência estacionário
$\omega_e$	Velocidade de referência para a transformada de Park
$\omega_m$	Velocidade angular mecânica do rotor
$\omega_n$	Frequência natural de oscilação do Sistema
$\omega_r$	Velocidade angular elétrica do rotor
$\omega_s$	Velocidade síncrona em rad/s
$B$	Coefficiente de amortecimento do sistema
$J$	Momento de inércia do sistema
$K$	Ganho do sistema
$P$	Número de polos
$f$	Frequência da rede
$n$	Velocidade do rotor em RPM
$p$	Derivada em relação ao tempo
$s$	Escorregamento
$\beta$	Ganho da malha do sistema
$\xi$	Coefficiente de amortecimento do Sistema
$\sigma$	Coefficiente de dispersão
$\tau$	Constante de tempo da malha do sistema

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>20</b>
<b>1.1</b>	<b>Finalidade e objetivos</b> .....	<b>21</b>
<b>1.2</b>	<b>Escopo e metodologia</b> .....	<b>21</b>
<b>1.3</b>	<b>Organização do trabalho</b> .....	<b>22</b>
<b>2</b>	<b>MODELO MATEMÁTICO DA MÁQUINA DE INDUÇÃO E CONTROLE DE CAMPO ORIENTADO INDIRETO</b> .....	<b>24</b>
<b>2.1</b>	<b>Características construtivas</b> .....	<b>24</b>
<b>2.2</b>	<b>Modelo de regime permanente</b> .....	<b>25</b>
<b>2.3</b>	<b>Estudo da máquina de indução simétrica trifásica</b> .....	<b>27</b>
<b>2.4</b>	<b>Equação mecânica</b> .....	<b>30</b>
<b>2.5</b>	<b>Transformada de Clark</b> .....	<b>30</b>
<b>2.6</b>	<b>Transformada de Park e sistemas dereferência</b> .....	<b>32</b>
<b>2.7</b>	<b>Controle por campo orientado indireto</b> .....	<b>34</b>
<b>2.8</b>	<b>Controladores de corrente</b> .....	<b>37</b>
<b>2.8.1</b>	<i>Controladores com Comparadores de Histerese</i> .....	<b>39</b>
<b>2.8.2</b>	<i>Controladores PI síncrono</i> .....	<b>41</b>
<b>3</b>	<b>AQUISIÇÃO DOS PARÂMETROS DO MOTOR</b> .....	<b>43</b>
<b>3.1</b>	<b>Parâmetros elétricos</b> .....	<b>44</b>
<b>3.2</b>	<b>Parâmetros mecânicos</b> .....	<b>48</b>
<b>3.3</b>	<b>Curvas do motor</b> .....	<b>52</b>
<b>4</b>	<b>PROJETO DOS CONTROLADORES PI</b> .....	<b>56</b>
<b>4.1</b>	<b>Controladores PI</b> .....	<b>56</b>
<b>4.2</b>	<b>Planta da malha de velocidade</b> .....	<b>58</b>
<b>4.3</b>	<b>Cálculo dos parâmetros do controlador de velocidade</b> .....	<b>59</b>
<b>4.4</b>	<b>Planta da malha de corrente</b> .....	<b>61</b>
<b>4.5</b>	<b>Cálculo dos parâmetros do controlador de corrente</b> .....	<b>62</b>
<b>4.6</b>	<b>Planta e cálculo dos parâmetros do controlador de posição</b> .....	<b>63</b>
<b>4.7</b>	<b>Resultados dos ganhos dos controladores</b> .....	<b>64</b>
<b>4.7.1</b>	<i>Sintonia do controlador PI da malha de velocidade</i> .....	<b>64</b>
<b>4.7.2</b>	<i>Sintonia do controlador PI da malha de corrente</i> .....	<b>67</b>
<b>4.7.3</b>	<i>Sintonia do controlador PI da malha de posição</i> .....	<b>69</b>

<b>5</b>	<b>RESULTADOS DA SIMULAÇÃO COMPUTACIONAL .....</b>	<b>71</b>
<b>5.1</b>	<b>Teste de controle de velocidade a vazio .....</b>	<b>74</b>
<b>5.2</b>	<b>Teste de controle de velocidade com carga .....</b>	<b>77</b>
<b>5.3</b>	<b>Teste de controle de posição .....</b>	<b>80</b>
<b>6</b>	<b>APLICAÇÃO PRÁTICA .....</b>	<b>82</b>
<b>6.1</b>	<b>Programação do DSP .....</b>	<b>84</b>
<b>6.2</b>	<b>Resultados experimentais .....</b>	<b>87</b>
<b>6.2.1</b>	<i>Teste de controle de velocidade a vazio .....</i>	<i>87</i>
<b>6.2.2</b>	<i>Teste de controle de velocidade com carga .....</i>	<i>89</i>
<b>6.2.3</b>	<i>Teste de controle de posição .....</i>	<i>91</i>
<b>6.3</b>	<b>Considerações finais .....</b>	<b>93</b>
<b>7</b>	<b>CONCLUSÕES .....</b>	<b>94</b>
<b>7.1</b>	<b>Resultados do presente trabalho .....</b>	<b>94</b>
<b>7.2</b>	<b>Trabalhos futuros .....</b>	<b>94</b>
	<b>REFERÊNCIAS .....</b>	<b>96</b>
	<b>APÊNDICE A – CÓDIGO FONTE DO DSP PARA CONTROLE DE VELOCIDADE DO MOTOR DE INDUÇÃO POR ORIENTAÇÃO DE CAMPO .....</b>	<b>98</b>
	<b>APÊNDICE B – CÓDIGO FONTE DO DSP PARA CONTROLE DE POSIÇÃO DO MOTOR DE INDUÇÃO POR ORIENTAÇÃO DE CAMPO .....</b>	<b>107</b>
	<b>APÊNDICE C – CÓDIGO FONTE DO DSP PARA ACIONAMENTO DO MOTOR DE INDUÇÃO POR CONTROLE V/f .....</b>	<b>116</b>

## 1 INTRODUÇÃO

As máquinas elétricas têm desempenhado papel fundamental na indústria e no desenvolvimento da sociedade humana, elas constituem um eficiente meio de conversão entre energia mecânica e elétrica, nos dois sentidos. Os motores de indução do tipo gaiola de esquilo representam a maioria dos motores comercializados no Brasil, sendo responsável de parte considerável da energia elétrica consumida, evidenciando a importância de estudos constantes sobre a eficiência deste equipamento. Estes motores possuem um custo reduzido de aquisição e manutenção em comparação com motores síncronos ou de corrente contínua de potência equivalente, sendo encontrados numa grande faixa de potência, desde alguns *watts* até milhares de *cv*.

O processo de automatização industrial vem exigindo cada vez mais a aplicação de motores elétricos em sistemas de acionamento, exigindo algum tipo de controle de torque, aceleração, velocidade ou posição. Pela sua maior facilidade de controle e modelagem mais simples, por um certo tempo, os motores de corrente contínua eram os mais difundidos na indústria, isso se deve ao fato de este tipo de motor já possuir o desacoplamento físico da malha do fluxo de campo e da malha de conjugado eletromecânico. Entretanto, com o avanço da eletrônica de potência, especialmente o surgimento do IGBT em meados da década de 80, e dos microprocessadores, bem como seu barateamento, permitiu que os inversores ficassem mais baratos e capazes de computar a complexa modelagem (modelo trifásico e não linear) e o controle dos motores de indução passou a ser viável.

O modelo do motor em quadratura é utilizado no controle vetorial e obtido pela transformada de Clark, onde o sistema trifásico é decomposto em um eixo em quadratura, transformação trifásico-bifásico. Como os eixos em quadratura estão estáticos, é dito que motor está representa em um eixo de coordenadas estacionárias (eixo  $\alpha\beta$ ). Fazendo esses eixos girarem na mesma velocidade dos vetores do fluxo do motor, velocidade síncrona, e decompondo esses vetores nesses eixos girantes (eixo  $dq$ , direto e em quadratura), vemos agora as grandezas do motor (seja corrente, tensão ou fluxo) como constantes, essa é a chamada transformada de Park.

O controle por campo orientado consiste em concentrar todo o fluxo de campo (do rotor) no eixo direto, desta forma, é possível obter o desacoplamento das malhas de fluxo e de torque, de forma semelhante que as máquinas de corrente contínua. Como o sistema de controle por campo orientado utilizado será o indireto e não efetua a medição direta e sim estima a posição do fluxo no entreferro. Partindo das considerações de que o fluxo está alinhado no eixo direto, o sistema indiretamente impõe que o fluxo se alinhe, pois todos os cálculos foram

realizados com essa condição. Por esta razão, esta estratégia é chamada de controle por campo orientado indireto (ONG, 1997).

### **1.1 Finalidade e objetivos**

O fim do presente trabalho é chegar a um sistema prático que torne o motor de indução trifásico aplicável em situações em que se deseja uma estabilidade, seja de velocidade ou posição, aliando teorias de máquinas elétricas, eletrônica de potência, controle de sistemas dinâmicos, acionamentos de máquinas, controle discreto, lógica de programação e processadores. O motor deve seguir uma posição ou velocidade de referência, mesmo com variações de carga mecânica em seu eixo, e sua partida deve ser suave, com uma rampa de velocidade para impedir as altas correntes de partida. O sistema de controle vetorial conterà três partes principais: transformações, controladores e o estimador de campo.

### **1.2 Escopo e metodologia**

Este projeto abrange várias áreas acadêmicas, como a arquitetura de processadores, programação, depuração, circuitos analógicos e digitais, modelagem da máquina de indução e teoria de controle vetorial com a representação das grandezas trifásicas no plano complexo, eletrônica de potência, controle de sistemas dinâmicos e controle discreto.

A bancada disponível no laboratório é constituída por um motor de indução trifásico gaiola de esquilo, um freio de Foucault, um *encoder* incremental, um conversor de topologia *back-to-back*, uma placa (JTAG DSP *simulator*) de envio e aquisição de dados conectada a um computador. No conversor será utilizado um lado como retificador e o outro como inversor, três medidores de corrente e a placa de controle que possui um DSP para processamento dos sinais e realização do controle. No computador será utilizado o Code Composer Studio 6.1 para desenvolvimento do código para o DSP e o ambiente “*Debug*”, deste mesmo programa, que trabalha em tempo real com o DSP.

Em primeiro plano, o sistema de controle vetorial foi projetado para o tempo contínuo e verificado com a simulação usando Matlab/Simulink. Antes do sistema de controle ser implementado, o motor foi ensaiado para obtenção dos parâmetros. Em seguida, o conversor adquirido pela Universidade Federal do Ceará foi ensaiado e depurado, onde foi levantada toda a relação do DSP com os conversores e seus componentes. Foi elaborado o procedimento para

utilização do equipamento. Os periféricos do DSP que serão utilizados no sistema de controle foram testados e calibrados, estes são: PWM, Timer0, Timer1, QEP (*Encoder*), Conversor Analógico-Digital (ADC), GPIO (*General Purpose Input Output*) e bibliotecas matemáticas. Para verificar todos os periféricos e a pinagem do DSP, um controlador de malha aberta volts por hertz (V/f) é implementado pela primeira vez no conversor. Verificou-se a implementação das transformações de coordenadas e a estimativa de fluxo, após a depuração e verificação, a aplicação do regulador de corrente vetorial é acrescentada, em seguida, o regulador de velocidade e, finalmente, o regulador de posição.

Foram desenvolvidos 4 algoritmos, o primeiro configura e habilita todos os periféricos do DSP, fica como um ponto de partida para projetos futuros. O segundo é um *softstart*, parte o motor com uma rampa de velocidade mantendo a relação tensão/velocidade constante, ou seja, mantendo o torque constante. O terceiro algoritmo executa o controle de velocidade do motor por campo orientado indireto utilizando o controle de corrente por histerese. O quarto controla a posição do eixo do rotor, em relação a posição inicial, utilizando histerese para controlar a corrente.

### **1.3 Organização do trabalho**

No capítulo 2 é descrito a modelagem matemática de regime permanente e a instantânea (transitória) das máquinas trifásicas de indução. O modelo em regime permanente é importante para realização de ensaios e obtenção dos parâmetros do motor, já o modelo instantâneo é importante para o estudo de transitórios no motor e para o controle vetorial. É ainda descrito as transformações para se chegar no modelo em quadratura do motor, nas referências estacionária e síncrona, e também é descrito o controle por campo orientado indireto.

No capítulo 3 é mostrado o desenvolvimento dos ensaios da máquina, são realizados 4 ensaios, três deles o motor estando a vazio e um estando com o rotor bloqueado. O objetivo deste capítulo é a obtenção dos parâmetros elétricos e mecânicos do motor.

No capítulo 4 é descrito o projeto dos controladores PI de posição, velocidade e corrente e mostrado, através de métodos da teoria de controle de sistemas dinâmicos, garantindo a estabilidade do sistema.

No capítulo 5 é desenvolvido a simulação no *software* Matlab/Simulink e mostrado os resultados para controle de velocidade e de posição para diferentes ciclos de trabalho.

No capítulo 6 é descrito o procedimento executado para programação do DSP TMS320F28335 na plataforma CCS (Code Composer Studio), é mostrado como cada periférico

deste processador foi utilizado na implementação do controle da máquina nos algoritmos desenvolvidos. Também são mostrados os resultados experimentais obtidos, com a aplicação dos algoritmos de controle vetorial desenvolvidos e do conversor *back-to-back*.

O capítulo 7 apresenta as conclusões e os trabalhos futuros.



## 2 MODELO MATEMÁTICO DA MÁQUINA DE INDUÇÃO E CONTROLE DE CAMPO ORIENTADO INDIRETO

A seguir será explanado toda a modelagem matemática da máquina de indução, o modelo transitório e o modelo de regime permanente, bem como do controle por campo orientado indireto.

### 2.1 Características construtivas

O motor de indução trifásico é composto por um enrolamento de armadura no estator (parte estática do motor) e um enrolamento de campo no rotor (parte girante do motor), geralmente os enrolamentos do rotor estão curto circuitados e a alimentação é ofertada nos enrolamentos do estator. Este tipo de motor também é conhecido como motor assíncrono, pois seu rotor não gira na mesma velocidade da oscilação da rede em que está conectado, isso ocorre exatamente pelo princípio de funcionamento deste tipo de motor, pois é justamente devido a esta diferença de velocidade que existe uma variação do fluxo concatenado que cruza os enrolamentos do rotor, induzindo uma tensão nestes enrolamentos do rotor, que produzirá uma corrente com um conseqüente fluxo magnético que tende a se alinhar com o fluxo gerado pelo estator, desta forma, produzindo torque. Essa diferença entre a velocidade síncrona ( $n_s$ ) e a velocidade do rotor ( $n$ ) chama-se escorregamento ( $s$ ) (Fitzgerald, 2006).

$$s = \frac{n_s - n}{n_s} \quad (2.1)$$

Onde,

$$\omega_s = \frac{4\pi f}{P} \quad (2.2)$$

$$n_s = \frac{\pi}{30} \omega_s \quad (2.3)$$

O motor de indução trifásico, como o próprio nome sugere, é alimentado por três correntes alternadas defasadas de 120 graus entre si. Cada fase do motor produz um campo magnético pulsante. A soma dos campos pulsantes de cada fase resulta em um campo que se move ao longo do estator (campo girante). Este campo que se desloca em relação ao rotor induz tensões no mesmo, lei de Faraday, produzindo correntes no rotor.

As correntes induzidas no rotor possuem polaridade opostas as correntes do estator,

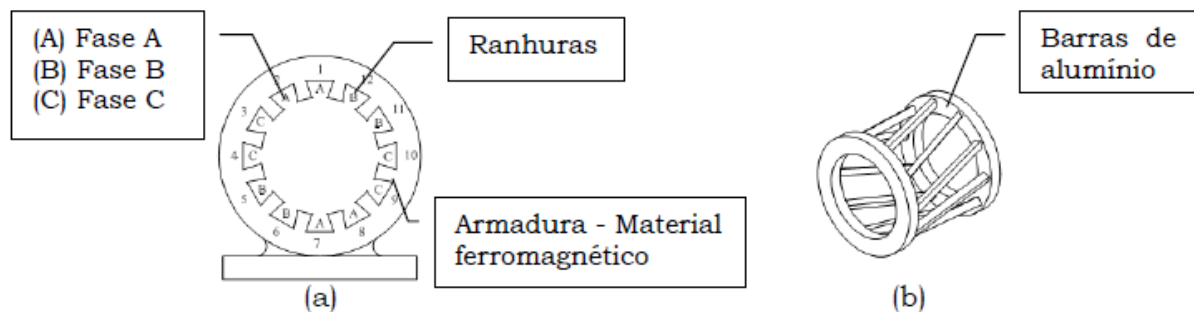
lei de Lenz, criando-se também um campo magnético girante no rotor, que tentará alinhar-se ao campo girante do estator.

A velocidade síncrona ( $\omega_s$ ) está relacionada diretamente com a frequência de alimentação da rede ( $f$ ) e o número de pólos ( $P$ ) do estator (2.2).

A velocidade do rotor não deve igualar-se a do campo girante do estator. Caso essas velocidades fossem iguais, não haveria movimento relativo entre o condutor e o campo, logo não haveria indução de correntes e não se produziria fluxo magnético no rotor. Assim, o rotor deve deslizar em velocidade menor para que se produza conjugado. Isso resulta numa diferença entre a velocidade do campo girante (velocidade síncrona) e a velocidade na qual o rotor gira como resultado da interação entre seu campo e o campo girante do estator.

A maioria dos motores de indução possuem o rotor do tipo gaiola de esquilo, constituído de barras condutoras alojadas em um núcleo de material ferromagnético curto circuitadas nas suas extremidades. Estas máquinas são bastante robustas quando comparadas com outras máquinas, são de construção simples, possuem menor custo de aquisição, demandam menos manutenção, existem numa maior faixa de potência (de alguns *watts* até modelos de 10000 HP) e atingem maiores velocidade (com modelos de até 5000 RPM) (Ong, 1997).

Figura 2.1 – Características construtivas do estator (a) e do rotor (b) do motor de indução

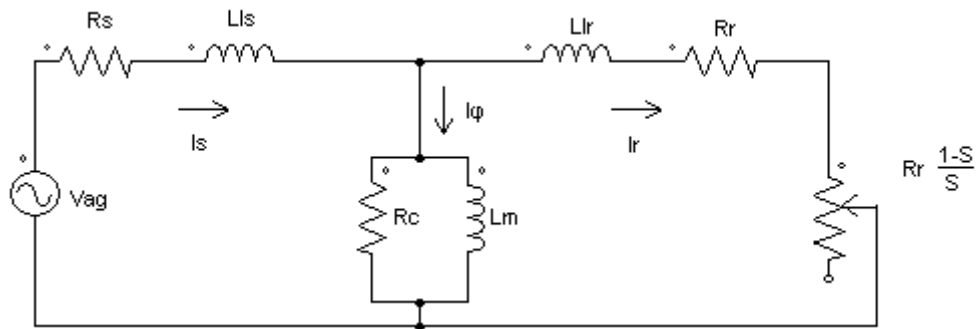


Fonte: Fitzgerald (2006).

## 2.2 Modelo de regime permanente

Neste modelo das máquinas de indução, os transitórios elétricos são desconsiderados, ou seja, este modelo representa a máquina em seu estado de estabilidade. O modelo representa o equivalente em estrela por fase do motor e é muito utilizado como base de cálculo de parâmetros a partir de ensaios. Considera-se que os enrolamentos do motor são simétricos, as tensões e correntes são puramente senoidais, o comportamento magnético do motor é linear e suas grandezas são referidas ao estator.

Figura 2.2 – Modelo de regime permanente da máquina de indução



Fonte: Própria.

Do ponto de vista desta modelagem, a máquina de indução se assemelha a um transformador com o secundário curto circuitado, onde o primário é o estator e o secundário é o rotor. Temos representando as perdas ôhmicas estáticas e rotóricas,  $R_s$  e  $R_r$ , respectivamente, as indutâncias de dispersão,  $L_{ls}$  e  $L_{lr}$ , e a indutância de magnetização  $L_m$ . A resistência  $R_c$  representa as perdas no núcleo, principalmente devido a correntes parasitas. O elemento  $R_r (1 - s)/s$  representa a potência ativa dissipada equivalente a potência mecânica do motor, esta potência é formada pela potência no eixo, por perdas rotacionais e por atrito (Fitzgerald, 2006).

Os parâmetros do circuito equivalente podem ser obtidos através de ensaios na máquina, a partir de contato direto com o fabricante ou estimados através de um modelo matemático baseado nas informações das curvas do fabricante.

Dois tipos de ensaios são realizados na máquina de indução para obter seus parâmetros, o ensaio a vazio e o ensaio de rotor bloqueado.

O ensaio a vazio fornece informações sobre a corrente de excitação e as perdas a vazio. Aplica-se a tensão nominal aos terminais do motor, estando o mesmo sem carga no eixo (a vazio), logo mede-se os valores de tensão ( $V_{vz}$ ), corrente de linha ( $I_{vz}$ ) e a potência elétrica polifásica total de entrada ( $P_{vz}$ ).

Como no caso do transformador no ensaio de curto circuito, o ensaio de rotor bloqueado fornece informações sobre as reatâncias de dispersão da máquina. O rotor é bloqueado, de forma que não possa girar ( $s = 1$ ), e uma tensão reduzida é aplicada ao estator, tal que faça circular pelo menos a corrente nominal. Assim, mede-se os valores de tensão ( $V_{BL}$ ), corrente de linha ( $I_{BL}$ ), potência elétrica polifásica total de entrada ( $P_{BL}$ ) e frequência do ensaio de rotor bloqueado ( $f_{BL}$ ).

Desprezando a resistência  $R_c$ , que representa as perdas no núcleo, a impedância equivalente do circuito de Thevenin em referência ao estator é:

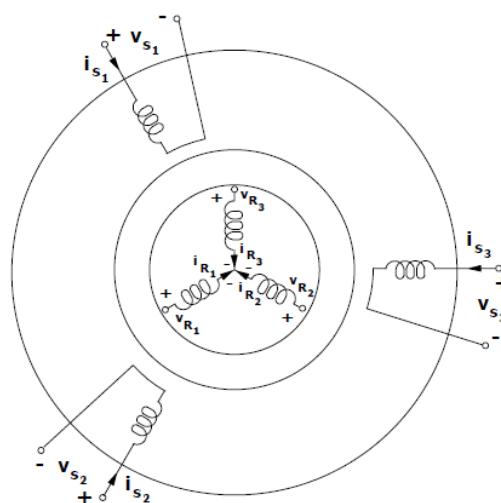
$$Z_{th} = R_s + jX_{ls} + \frac{jX_m(R_r/s + jX_{lr})}{R_r/s + j(X_m + X_{lr})} \quad (2.4)$$

### 2.3 Estudo da máquina de indução simétrica trifásica

De fato, a máquina de indução polifásica é simétrica, tanto a de rotor bobinado quanto a de gaiola de esquilo, em ambas os enrolamentos do estator são igualmente defasados, bem como os do rotor, e possuem estruturas simétricas cilíndricas tanto no rotor quanto no estator. A diferença é que na máquina de rotor bobinado, o rotor possui a mesma quantidade de fases que o estator, e na máquina com o rotor gaiola de esquilo, acontece que cada barra da gaiola constitui uma fase (Barbi, 1986).

O modelo desenvolvido a seguir, representa uma máquina de indução trifásica de rotor bobinado, com os enrolamentos do rotor curto circuitados. Como foi dito, a maioria dos motores utilizados na indústria são de rotor gaiola de esquilo, bem como o motor cujo objetivo deste trabalho é controlar que também é deste tipo, e é possível utilizar este modelo que será descrito a seguir pois quando são realizados os ensaios na máquina de rotor gaiola de esquilo, utiliza-se o modelo que representa o rotor com a mesma quantidade de fases que o estator (modelo de regime permanente), logo, o que se obtém são parâmetros do equivalente rotor trifásico deste rotor gaiola de esquilo.

Figura 2.3 – Representação da máquina simétrica trifásica



Fonte: Barbi (1986).

A seguintes considerações são feitas para o modelo a ser desenvolvido:

1. A saturação magnética é negligenciada. É considerada válida a superposição

dos fluxos e todas as indutâncias são consideradas independentes da magnitude das correntes.

2. A distribuição espacial das forças magnetomotrizes e dos fluxos no entreferro é considerada senoidal e simétrica em relação ao eixo magnético dos enrolamentos.
3. As distorções no fluxo introduzidas pelo efeito das ranhuras são ignoradas. Os enrolamentos são considerados constituídos de uma distribuição senoidal de condutores de diâmetro desprezível.
4. As perdas por histerese e por correntes parasitas no material magnético são desprezadas.
5. Harmônicas nas tensões e correntes são desprezíveis.

As consequências destas considerações são:

1. Os fluxos podem ser superpostos, ou seja, o fluxo total é igual ao somatório dos fluxos produzidos pelos enrolamentos do estator e do rotor;

$$\Phi_{total} = \sum \Phi_r + \sum \Phi_s \quad (2.5)$$

2. As indutâncias próprias e resistências dos enrolamentos do rotor e do estator são constantes e iguais entre as fases;
3. Como consequência do defasamento igual entre as fases, a indutância mútua entre enrolamentos de estator são constantes, bem como entre os enrolamentos do rotor;
4. As indutâncias mútuas entre rotor e estator são funções senoidais do deslocamento angular.

As equações de tensão resultante do acoplamento magnético dos circuitos do rotor e do estator podem ser escritas como segue (Ong, 1997):

$$v_{as} = i_{as}R_s + \frac{d\lambda_{as}}{dt} \quad V \quad (2.6)$$

$$v_{bs} = i_{bs}R_s + \frac{d\lambda_{bs}}{dt} \quad V \quad (2.7)$$

$$v_{cs} = i_{cs}R_s + \frac{d\lambda_{cs}}{dt} \quad V \quad (2.8)$$

Agora as tensões nas fases do rotor:

$$v_{ar} = i_{ar}R_r + \frac{d\lambda_{ar}}{dt} \quad V \quad (2.9)$$

$$v_{br} = i_{br}R_r + \frac{d\lambda_{br}}{dt} \quad V \quad (2.10)$$

$$v_{cr} = i_{cr}R_r + \frac{d\lambda_{cr}}{dt} \quad V \quad (2.11)$$

Onde os subscritos  $a$ ,  $b$  e  $c$  indicam a fase,  $s$  indica que se trata do estator e  $r$  que se trata do rotor.

Em notação matricial, os fluxos de dispersão dos enrolamentos do estator e do rotor em termos das indutâncias dos enrolamentos e das correntes, pode ser escrito:

$$\begin{bmatrix} \lambda_s^{abc} \\ \lambda_r^{abc} \end{bmatrix} = \begin{bmatrix} L_{ss}^{abc} & L_{sr}^{abc} \\ L_{rs}^{abc} & L_{rr}^{abc} \end{bmatrix} \begin{bmatrix} i_s^{abc} \\ i_r^{abc} \end{bmatrix} \quad Wb \quad (2.12)$$

Onde,

$$\begin{aligned} \lambda_s^{abc} &= [\lambda_{as} \ \lambda_{bs} \ \lambda_{cs}]^t \\ \lambda_r^{abc} &= [\lambda_{ar} \ \lambda_{br} \ \lambda_{cr}]^t \\ i_s^{abc} &= [i_{as} \ i_{bs} \ i_{cs}]^t \\ i_r^{abc} &= [i_{ar} \ i_{br} \ i_{cr}]^t \end{aligned} \quad (2.13)$$

E as matrizes de indutâncias estator-estator e rotor-rotor são:

$$L_{ss}^{abc} = \begin{bmatrix} L_{ls} + L_{ss} & L_{sm} & L_{sm} \\ L_{sm} & L_{ls} + L_{ss} & L_{sm} \\ L_{sm} & L_{sm} & L_{ls} + L_{ss} \end{bmatrix} \quad H \quad (2.14)$$

$$L_{rr}^{abc} = \begin{bmatrix} L_{lr} + L_{rr} & L_{rm} & L_{rm} \\ L_{rm} & L_{lr} + L_{rr} & L_{rm} \\ L_{rm} & L_{rm} & L_{lr} + L_{rr} \end{bmatrix} \quad H \quad (2.15)$$

Onde  $L_{ss}$  e  $L_{rr}$  são indutâncias próprias,  $L_{ls}$  e  $L_{lr}$  são indutâncias de dispersão e  $L_{sm}$  e  $L_{rm}$  são indutâncias mútuas entre os enrolamentos iguais.

$$L_{sr}^{abc} = [L_{rs}^{abc}]^t = L_{sr} \begin{bmatrix} \cos(\theta_r) & \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) \\ \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r) & \cos(\theta_r + \frac{2\pi}{3}) \\ \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r) \end{bmatrix} \quad H \quad (2.16)$$

Nota-se que o modelo idealizado da máquina é representado por seis equações diferenciais, uma para cada enrolamento. Essas equações são acopladas nas outras através das indutâncias mútuas entre os enrolamentos.

Transformações matemáticas como a de Clark e a de Park podem facilitar a computação da solução transitória do sistema transformando as equações diferenciais com as indutâncias variando no tempo em equações diferenciais com as indutâncias constantes.

## 2.4 Equação mecânica

A equação mecânica relaciona a carga com as grandezas elétricas. Ela descreve, em termos do torque eletromagnético e dos parâmetros da carga, a evolução da velocidade mecânica ou a posição do eixo. Em função da velocidade, podemos escrever (Reginatto, 1993)

$$J \frac{d\omega_m}{dt} + B\omega_m + Tc = Te \quad (2.17)$$

Onde

- $J$  é o momento de inércia do motor mais carga;
- $B$  é o coeficiente de amortecimento do motor mais carga;
- $Tc$  é o torque da carga;
- $Te$  é o torque eletromagnético.

Escrevendo em termos da velocidade angular elétrica do rotor,  $\omega_r$ , temos

$$\omega_r = \frac{P}{2} \omega_m \quad (2.18)$$

$$J \frac{d\omega_m}{dt} + B\omega_m = \frac{P}{2} (Te - Tc) \quad (2.19)$$

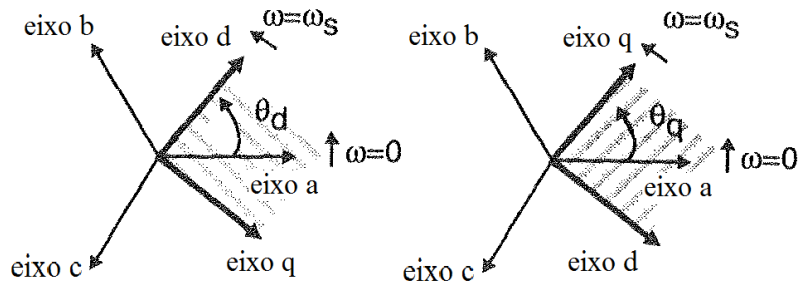
## 2.5 Transformada de Clark

Para obtenção de um modelo que forneça uma melhor análise da máquina de indução, o primeiro passo é estudar a Transformada de Clark ou transformada  $\alpha\beta$ . Essa transformada linear propõe a decomposição do sistema em dois eixos estabelecidos em quadratura, eixo  $\alpha$  e eixo  $\beta$ .

Considere a Figura 2.4, a máquina trifásica é representada fasorialmente por três vetores,  $a$ ,  $b$  e  $c$ , de mesma amplitude e com uma defasagem de  $120^\circ$  entre cada uma das fases. Sendo representado o estator, estes vetores giram em sentido anti-horário na velocidade síncrona,  $\omega_s$ . Para que a transformada seja bidirecional, a matriz que representa esta operação precisa possuir inversa, ou seja, precisa ser quadrada, desta forma é adicionada uma terceira

componente chamada de componente homopolar ou componente de sequência zero. A componente  $0$  representa a soma das três fases de um sistema trifásico, logo, em um sistema equilibrado, como no caso do motor de indução, esta componente é nula.

Figura 2.4 – Relação entre as grandezas  $dq$  e  $abc$  na notação para motor



Fonte: Ong (1997).

A transformada de Clark, bem como a matriz de transformação  $T_{\alpha\beta 0}$ , estão representados abaixo.

$$[f_{\alpha\beta 0}] = [T_{\alpha\beta 0}][f_{abc}] \quad (2.20)$$

$$[T_{\alpha\beta 0}] = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \quad (2.21)$$

Onde  $f$  é tensão, corrente ou fluxo. A operação inversa é dada por

$$[f_{abc}] = [T_{\alpha\beta 0}]^{-1}[f_{\alpha\beta 0}] \quad (2.22)$$

$$[T_{\alpha\beta 0}]^{-1} = \begin{bmatrix} 0 & 1 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \quad (2.23)$$

Esta é a transformação utilizada para obtenção da máquina bifásica considerando invariante as amplitudes das grandezas elétricas, o mesmo não pode ser dito para a potência.



## 2.6 Transformada de Park e sistemas de referência

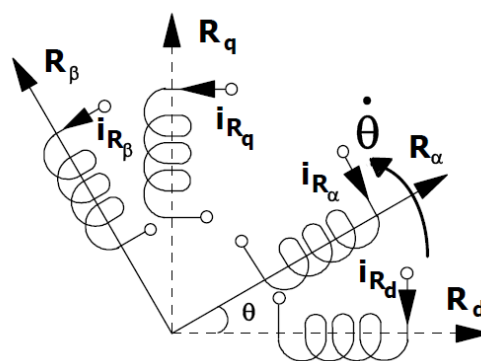
Fisicamente, a transformação de Park transforma a máquina simétrica trifásica numa máquina simétrica bifásica, mantendo conjugado, potência mecânica e velocidade. O sistema trifásico é decomposto em dois eixos defasados em  $90^\circ$ , o eixo direto ( $d$ ) e o eixo em quadratura ( $q$ ) que giram em uma determinada velocidade.

Como na máquina de indução o campo magnético do estator e os enrolamentos do rotor giram em velocidades diferentes, é necessário se estabelecer um sistema de referência utilizado para a representação das variáveis da máquina em uma mesma velocidade. A velocidade do eixo  $dq$  pode ser nula, igual a velocidade do rotor ou igual a igual a velocidade síncrona, de forma mais comum.

- $\omega = 0$  é o sistema de coordenadas fixas ou estacionárias no estator ou sistema de coordenadas  $\alpha\beta$ ;
- $\omega = \omega_r$  é o sistema de coordenadas fixo no rotor;
- $\omega = \omega_s$  é o sistema de coordenadas síncrono, por girar com a frequência síncrona das tensões de alimentação. Durante este trabalho, quando falado em eixo  $dq$  entenda como sistema de coordenadas síncrono.

Onde  $\omega = \dot{\theta}$  na Figura 2.5 e é velocidade com que giram os eixos  $dq$  da transformada de Park.

Figura 2.5 – Representação dos eixos  $dq$  e  $\alpha\beta$



Fonte: Barbi (1986).

A Transformada de Park com referência arbitrária está equacionada abaixo

$$\begin{bmatrix} f_q^x \\ f_d^x \\ f_0^x \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_x) & -\text{sen}(\theta_x) & 0 \\ \text{sen}(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \\ f_0 \end{bmatrix} \quad (2.24)$$

Onde:

- $\mathbf{x}$  é a nova base de referência (estática, rotor ou estator);
- $\theta_x$  representa o ângulo formado entre os fasores das fases do estator.

Este novo sistema de coordenadas é conhecido como sistemas de coordenadas arbitrárias, pois agora o eixo em quadratura, que antes era fixo, pode girar na velocidade do rotor ou na velocidade síncrona. Utilizando o eixo de referência girando na velocidade síncrona, as variáveis de eixo direto e em quadratura são vistas como constantes.

O sobrescrito  $e$  nas variáveis do motor, indicam que esta variável está na referência síncrona (no eixo dq0), já o sobrescrito  $s$  indica que a variável está na referência estacionária (no eixo  $\alpha\beta$ 0).

O modelo do motor de indução no sistema de coordenadas síncrono ( $\theta_x = \theta_e$ ) pode ser apresentado na forma (Reginatto, 1993)

$$v_{qs}^e = R_s i_{qs}^e + p \lambda_{qs}^e + \omega_e \lambda_{ds}^e \quad (2.25)$$

$$v_{ds}^e = R_s i_{ds}^e + p \lambda_{ds}^e - \omega_e \lambda_{qs}^e \quad (2.26)$$

$$0 = R_r i_{qr}^e + p \lambda_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e \quad (2.27)$$

$$0 = R_r i_{dr}^e + p \lambda_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e \quad (2.28)$$

$$\lambda_{qs}^e = L_s i_{qs}^e + L_m i_{qr}^e \quad (2.29)$$

$$\lambda_{ds}^e = L_s i_{ds}^e + L_m i_{dr}^e \quad (2.30)$$

$$\lambda_{qr}^e = L_r i_{qr}^e + L_m i_{qs}^e \quad (2.31)$$

$$\lambda_{dr}^e = L_r i_{dr}^e + L_m i_{ds}^e \quad (2.32)$$

$$T_e = \frac{3}{2} \frac{P L_m}{L_r} (\lambda_{dr}^e i_{qs}^e - \lambda_{qr}^e i_{ds}^e) \quad (2.33)$$

$$J p \omega_r = -B \omega_r + \frac{P}{2} (T_e - T_c) \quad (2.34)$$

No sistema de coordenadas estacionário ( $\theta_x = 0$ ), o modelo do motor de indução

fica representado por (Reginatto, 1993)

$$v_{qs}^s = R_s i_{qs}^s + p \lambda_{qs}^s \quad (2.35)$$

$$v_{ds}^s = R_s i_{ds}^s + p \lambda_{ds}^s \quad (2.36)$$

$$0 = R_r i_{qr}^s + p \lambda_{qr}^s - \omega_r \lambda_{dr}^s \quad (2.37)$$

$$0 = R_r i_{dr}^s + p \lambda_{dr}^s + \omega_r \lambda_{qr}^s \quad (2.38)$$

$$\lambda_{qs}^s = L_s i_{qs}^s + L_m i_{qr}^s \quad (2.39)$$

$$\lambda_{ds}^s = L_s i_{ds}^s + L_m i_{dr}^s \quad (2.40)$$

$$\lambda_{qr}^s = L_r i_{qr}^s + L_m i_{qs}^s \quad (2.41)$$

$$\lambda_{dr}^s = L_r i_{dr}^s + L_m i_{ds}^s \quad (2.42)$$

$$T_e = \frac{3}{2} \frac{P}{2} \frac{L_m}{L_r} (\lambda_{dr}^s i_{qs}^s - \lambda_{qr}^s i_{ds}^s) \quad (2.43)$$

$$J p \omega_r = -B \omega_r + \frac{P}{2} (T_e - T_c) \quad (2.44)$$

## 2.7 Controle por campo orientado indireto

A vantagem do controle vetorial indireto com relação ao controle vetorial direto é que o mesmo necessita apenas da posição do eixo para estimar o fluxo do rotor, evitando assim a necessidade de instalação de sensores de fluxo no entreferro da máquina, utilizando apenas *encoder* para medição da posição, e, desta forma, diminuindo o custo de implementação do sistema.

A Orientação do fluxo de rotor consiste em alinhar o vetor fluxo de rotor ao longo do eixo direto de tal forma que se possa escrever

$$\lambda_{dr}^e = \lambda_r^e \quad (2.45)$$

$$\lambda_{qr}^e = p \lambda_{qr}^e = 0 \quad (2.46)$$

O fato de a componente de quadratura do fluxo de rotor ser nula estabelece uma relação direta entre as componentes de quadratura das correntes estáticas e rotóricas.

Desenvolvendo as equações do modelo do motor de indução no sistema de coordenadas síncrono, mostradas no item 2.4.1, considerando o fluxo de quadratura do rotor igual a zero, temos (Reginatto, 1993):

$$p\lambda_{dr}^e + \frac{R_r}{L_r}\lambda_{dr}^e = \frac{L_m R_r}{L_r} i_{ds}^e \quad (2.47)$$

$$\lambda_{dr}^e = \frac{L_m R_r}{pL_r + R_r} i_{ds}^e \quad (2.48)$$

$$T_e = \frac{3P}{2} \frac{L_m}{L_r} \lambda_{dr}^e i_{qs}^e \quad (2.49)$$

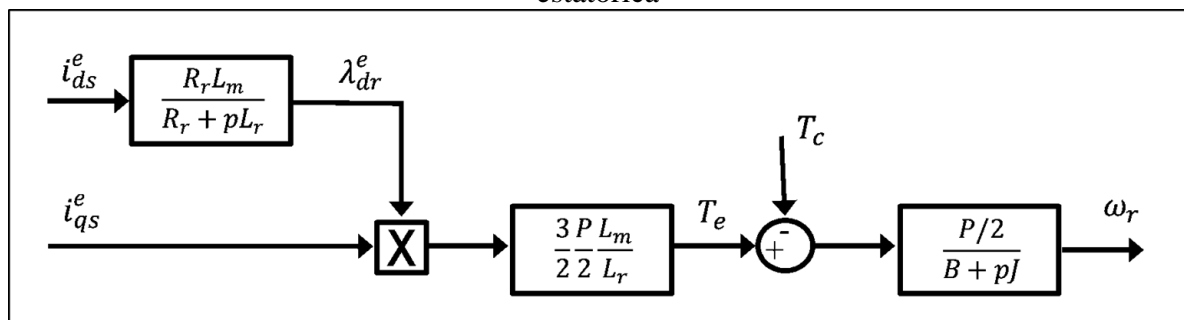
$$Jp\omega_r + B\omega_r = \frac{P}{2}(T_e - T_c) \quad (2.50)$$

As equações acima determinam o comportamento do torque e do fluxo do motor quando o fluxo de rotor está orientado ao longo do eixo direto. Estas equações mostram que:

- O fluxo  $\lambda_{dr}^e = \lambda_r^e$ , depende unicamente da componente  $i_{ds}^e$  da corrente de estator;
- Se o fluxo  $\lambda_{dr}^e$  for constante, o torque eletromagnético é proporcional a componente  $i_{qs}^e$ .

Isto significa que as componentes  $i_{qs}^e$ , e  $i_{ds}^e$  da corrente de estator foram desacopladas nas componentes que produzem fluxo e torque no motor, respectivamente.

Figura 2.6 – Desacoplamentos das correntes de eixo direto e em quadratura da corrente estática



Fonte: Própria.

Nestas condições, o controle do motor de indução é similar ao controle de um motor CC, pois a resposta do torque é direta em relação à corrente de estator de quadratura na

referência síncrona. Para isso é necessário manter o fluxo constante.

Quando o fluxo de rotor está ao longo do eixo direto, a seguinte expressão fornece a velocidade angular de escorregamento.

$$\omega_2 = \omega_e - \omega_r = \frac{R_r i_{qs}^e}{L_r i_{ds}^e} \quad (2.51)$$

$$\theta_r = \int \omega_r(t) dt \quad (2.52)$$

$$\theta_e = \theta_r + \int \omega_2(t) dt \quad (2.53)$$

Temos ainda,

$$\delta_r = \frac{L_r}{R_r} \quad (2.54)$$

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \quad (2.55)$$

$$L_s = L_{ls} + L_m \quad (2.56)$$

$$L_r = L_{lr} + L_m \quad (2.57)$$

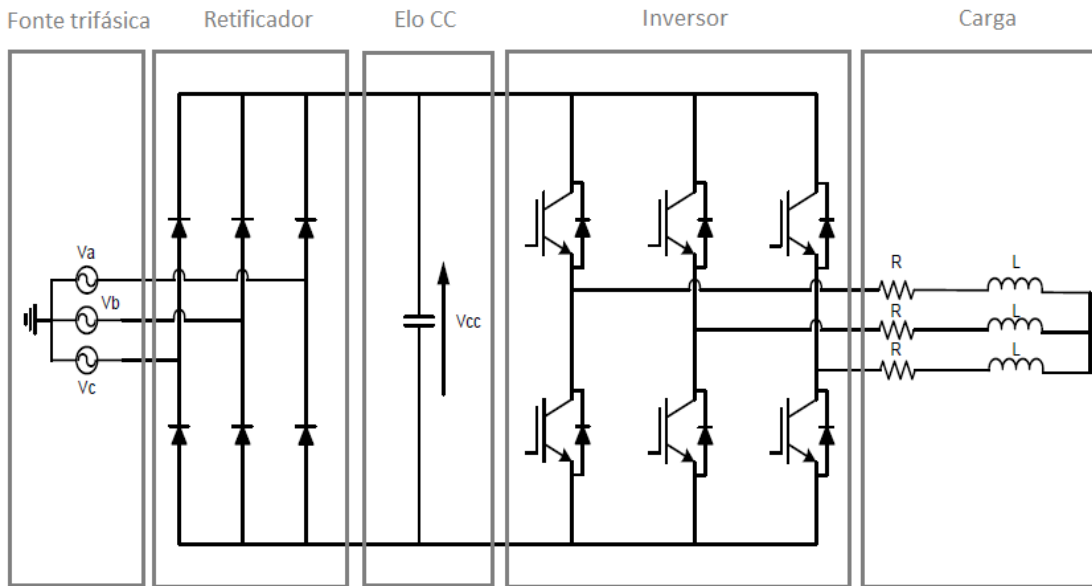
Onde,  $\delta_r$  é a constante de tempo rotórica e  $\sigma$  é o coeficiente de dispersão.

Na Figura 2.7 está representado o processo do controle vetorial pelo método de indireto da orientação de campo. Classifica-se como método indireto pois não existe aquisição do fluxo do rotor, a posição do fluxo do rotor é estimada, no bloco Posição do Fluxo, através da Equações 2.51 e 2.53. No Capítulo 5 será comentado com detalhes o funcionamento de todo o processo do controle vetorial, bem como cada bloco.

Para realizar o controle da posição, mais um controlador PI é adicionado antes do controlador de velocidade. Um sinal de posição de referência é estabelecido e com a integração da velocidade mecânica obtemos a posição do rotor.



Figura 2.8 – Diagrama esquemático do conversor utilizado para controlar a corrente



Fonte: Própria.

Esta topologia de inversor é chamada de CSI, do inglês *Source Voltage Inversor* (Inversor com fonte de tensão). Apesar do inversor ser da configuração de fonte de tensão, a indutância presente na carga, no caso o motor, permite a controlabilidade de corrente, justamente pela inercia que este elemento fornece, não permitindo que a corrente varie imediatamente como no caso de uma carga apenas resistiva.

Os sinais de comando das chaves de potência do inversor são na forma de modulação por largura de pulso, PMW (*"Pulse Width Modulation"*). Estes sinais são formados por uma cadeia de pulsos binários representando os estados que as chaves podem assumir (bloqueio ou condução). O estado da chave irá determinar qual dos dois terminais do Elo CC (positivo ou negativo) será conectado à carga, controlando o fluxo de potência da fonte à carga.

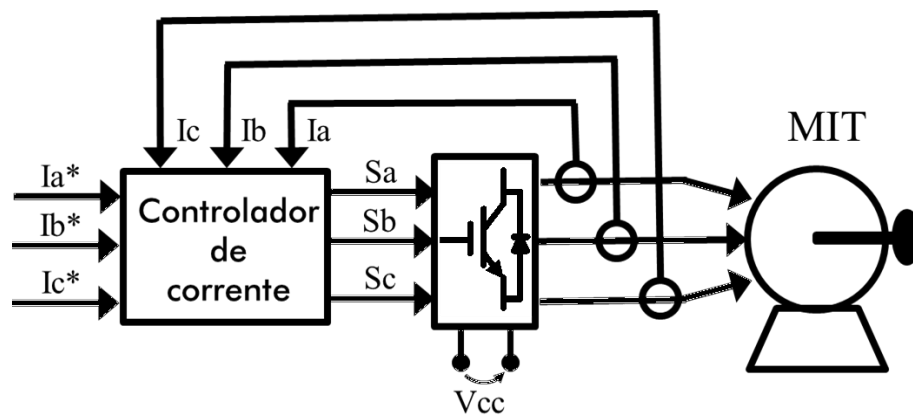
Portanto, o objetivo dos controladores de corrente é forçar que um sinal de corrente de referência seja imposto na carga. A ação do chaveamento é baseada na comparação entre o sinal de corrente medido na carga e o de referência. Um chaveamento rápido e de forma a minimizar os níveis de harmônicos implicam em um acionamento com boa resposta e baixos torques pulsantes.

Aqui serão tratados dois tipos de controladores de corrente, os comparadores por histerese e os baseados na modulação PWM. Um modelo básico para controle de corrente está representado na Figura 2.9, e tem os seguintes componentes (Kazmierkowski, 1998):

- Sinais de referência: Sinal de corrente desejado na carga ( $I_a^*$ ,  $I_b^*$  e  $I_c^*$ );

- Controlador: Realiza os cálculos necessários ao controle em função dos sinais de referência e a corrente na carga e retorna o comando das chaves do inversor (Controlador de corrente);
- Corrente na carga: Por meio de transdutores de corrente baseados no *Efeito Hall*, por exemplo, os sinais de corrente na carga são obtidos e enviados ao controlador ( $I_a$ ,  $I_b$  e  $I_c$ );
- Comando das chaves: São resultado dos cálculos realizados no controlador e de forma a impor a corrente de forma desejada na carga com o mínimo de níveis harmônicos ( $S_a$ ,  $S_b$  e  $S_c$ );
- Inversor: Conjunto de chaves que, de acordo com o comando das chaves, irá determinar a tensão na carga;
- Fonte de corrente contínua: Tensão proveniente de uma fonte CC ou da retificação da tensão da rede elétrica ( $V_{cc}$ );
- Carga: Elemento o qual deseja-se impor a corrente (Motor de Indução Trifásico).

Figura 2.9 – Modelo de controlador de corrente para motor trifásico



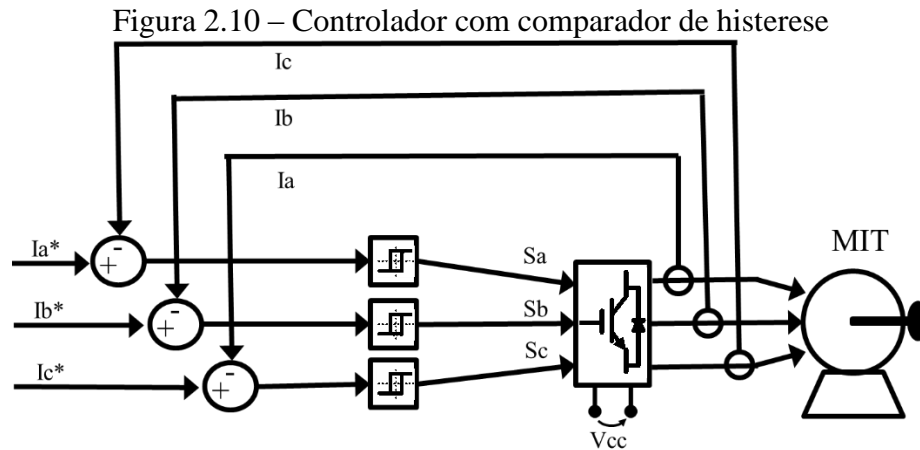
Fonte: Própria.

### 2.8.1 Controladores com Comparadores de Histerese

As principais vantagens dos controladores com comparadores de histerese são: simplicidade na implementação, robustez, ótimo desempenho dinâmico e a independência dos parâmetros da carga. As desvantagens desse tipo de controle são: limitação pela frequência de chaveamento, frequência de chaveamento variável e dependente da constante de tempo da carga e da tensão da fonte.



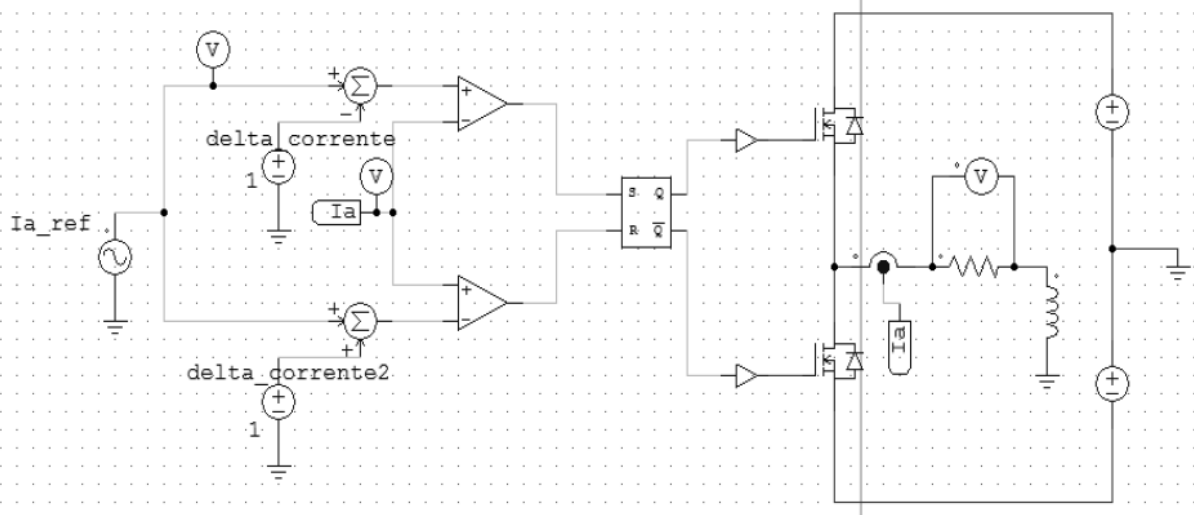
Seu algoritmo de controle é formado por três comparadores, onde cada um possui uma unidade de memória e é responsável por uma perna do inversor. É estipulado uma margem de variação superior e inferior ao sinal de referência, se ultrapassados esses valores as chaves são comutadas, senão o estado anterior permanece.



Fonte: Própria.

Na Figura 2.11 temos a simulação do controlador por comparador de histerese monofásico feito no *software* PSIM 9.0, o controlador foi implementado por dois comparadores analógicos e um *flip-flop* tipo Set-Reset.

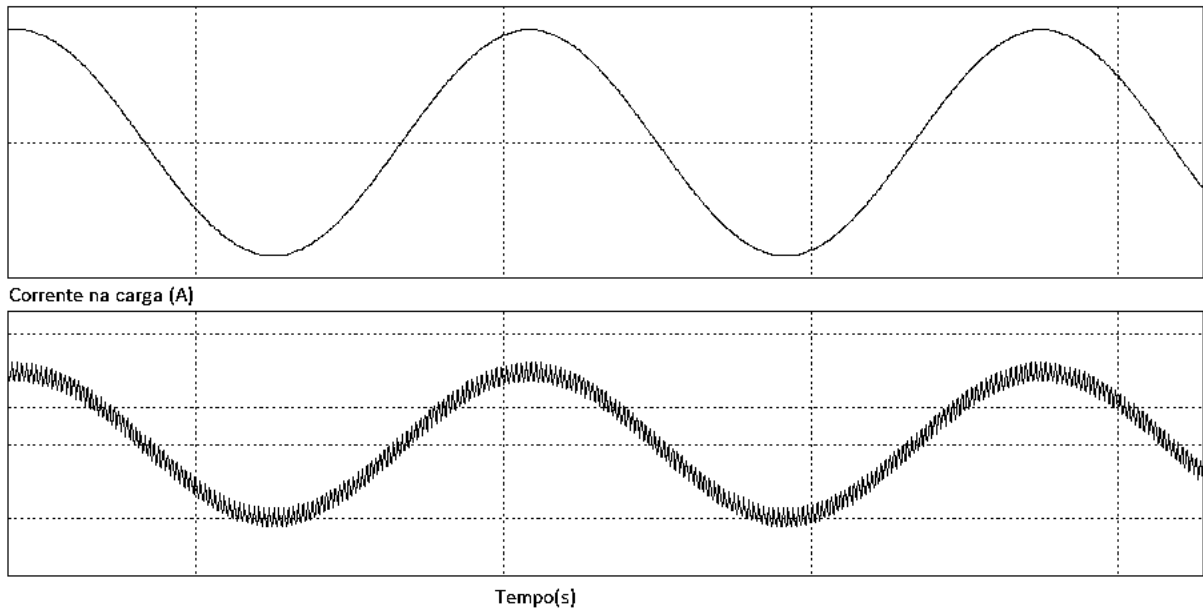
Figura 2.11 – Controlador com comparador de histerese monofásico simulado no PSIM



Fonte: Própria.

Na Figura 2.12 temos os resultados da simulação com a corrente de referência e a corrente medida na carga.

Figura 2.12 – Corrente de referência e corrente na carga controlada por histerese



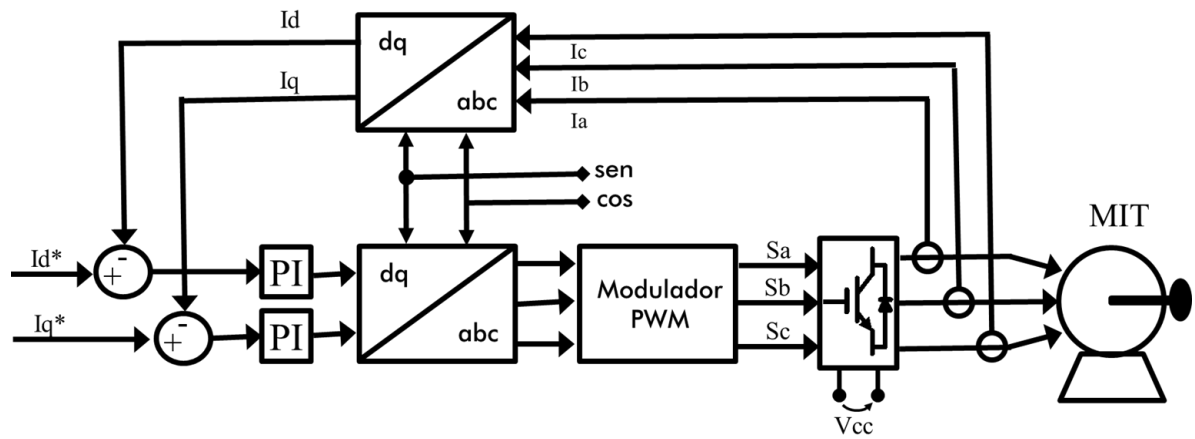
Fonte: Própria.

### 2.8.2 Controladores PI síncrono

O controlador PI síncrono utiliza dois compensadores PI com componentes vetoriais no sistema de referência síncrono, girante na frequência da rede de alimentação (eixos  $dq$ ).

Para realizar as transformadas de eixos de referência, é necessário o conhecimento do ângulo das correntes (seno e cosseno do ângulo). O ângulo pode ser obtido por meio de um circuito PLL (“*Phase Locked Loop*”), muito utilizado para o sincronismo de sinais. Outra forma é utilizar a posição do fluxo que foi estimada no controle vetorial.

Figura 2.13 – Controlador PI síncrono

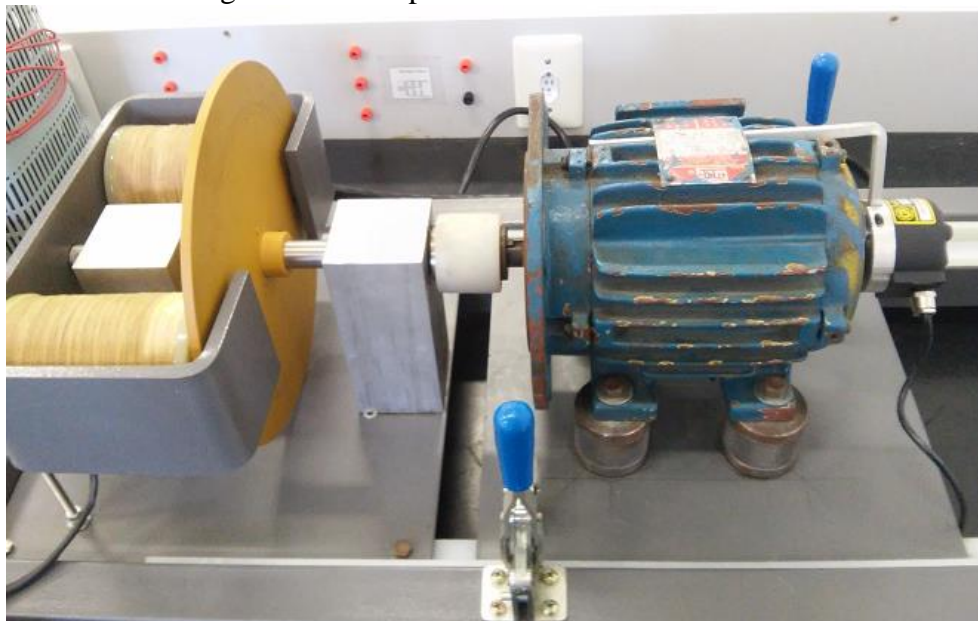


Fonte: Própria.

### 3 AQUISIÇÃO DOS PARÂMETROS DO MOTOR

Para o bom e correto funcionamento do controle vetorial é imprescindível ter uma boa faixa de acerto nos parâmetros da máquina e do conjunto na qual está acoplada. Portanto, nos ensaios realizados, o motor estava acoplado no *kit* freio de Foucault, com este desligado, de forma que os parâmetros já estejam em função dessa massa acoplada na condição a vazio, para que a corrente  $i_{ds}^e$  que será encontrada e mantida constante no acionamento seja suficiente para manter o motor a vazio e a corrente  $i_{qs}^e$  inteire o conjugado proveniente do aumento da carga quando o freio for ativado. Na Figura 3.1 pode-se ver o sistema montado.

Figura 3.1 – Acoplamento motor e freio de Foucault



Fonte: Própria.

Foram realizados quatro ensaios, os dois primeiros para obtenção dos parâmetros elétricos e os dois restantes para obtenção de parâmetros mecânicos. Como o motor está acoplado em um sistema que influencia seu desempenho significativamente, optou-se pela realização dos ensaios para obtenção de grandezas mecânicas ao invés de utilizar os dados fornecidos pelo fabricante.

Para a realização dos ensaios, foi utilizado um motor de indução trifásico gaiola de esquilo ligado em estrela, a tensão das suas bobinas é 220 V. Na Tabela 3.1 abaixo temos os dados da placa do motor.

Tabela 3.1 – Dados de placa do motor

<b>Tensão Nominal</b>	220V ( $\Delta$ ) / 380V (Y)
<b>Corrente Nominal</b>	3,8 A ( $\Delta$ ) / 2,2 A (Y)
<b>Frequência</b>	60Hz
<b>Potência</b>	1CV = 750W
<b>Fator de Potência</b>	0,74
<b>Velocidade Nominal</b>	1730 RPM
<b>Rendimento</b>	69%
<b>IP/IN</b>	5,5
<b>Classe de Isolamento</b>	IP55
<b>Categoria</b>	N

Fonte: própria.

### 3.1 Parâmetros elétricos

Para obtenção dos parâmetros elétricos foram realizados dois ensaios, ensaio a vazio e ensaio de rotor bloqueado.

Com o motor acoplado apenas ao disco do freio de Foucault mas sem nenhuma carga adicionada, o motor foi acionado com tensão nominal e as seguintes leituras foram realizadas, conforme Tabela 3.2.

Tabela 3.2 - Dados obtidos no ensaio a vazio

<b>Ensaio a Vazio</b>	
Resistencia do Estator	5,35 $\Omega$ /fase
Tensão (V)	220
Corrente (A)	1,3
Potência à Vazio (W)	430
Velocidade (RPM)	1795

Fonte: própria.

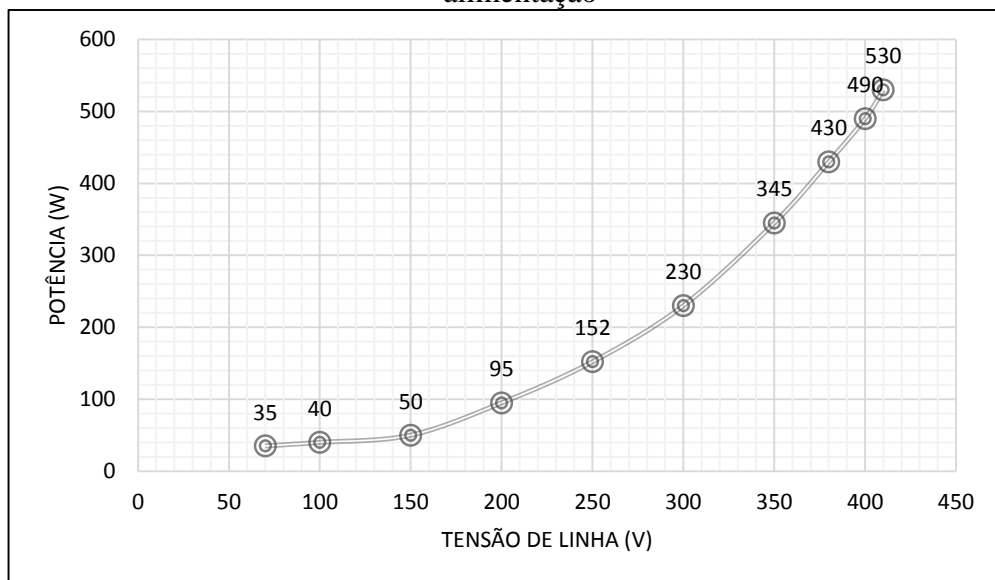
A fim de se obter as perdas rotacionais, a tensão foi elevada até 108% e reduzida até 18% da tensão nominal, com o motor ainda a vazio. Para cada valor de tensão aplicada, foi medido corrente de linha, potência e velocidade. A ideia é plotar a curva da potência consumida em função da tensão de alimentação, e extrapolar o valor da potência para tensão zero, esta potência é equivalente às perdas rotacionais da máquina. Os dados obtidos estão na Tabela 3.3 e a plotagem na Figura 3.2.

Tabela 3.3 - Dados obtidos no ensaio a vazio variando a tensão

Tensão de linha (V)	Corrente (I)	Potência (W)	Velocidade (rpm)
410	1,5	530	1796
400	1,4	490	1795
380	1,25	430	1795
350	1,11	345	1795
300	0,9	230	1795
250	0,76	152	1793
200	0,62	95	1791
150	0,5	50	1783
100	0,41	40	1762
70	0,43	35	1706

Fonte: própria.

Figura 3.2 – Gráfico da potência consumida pelo motor a vazio em função da tensão de alimentação



Fonte: própria.

Na Figura 3.2 nota-se que a potência consumida tende a 35 W quando a tensão se aproxima de zero, portanto as perdas rotacionais são consideradas com este valor.

$$P_{rot} = 35 \text{ W}$$

O ensaio de rotor bloqueado consiste em aumentar a tensão de alimentação de zero até que a corrente alcance seu valor nominal com o eixo do motor impedido de girar (escorregamento igual a 1). Nestas condições é medido a potência, a tensão e a corrente. Os valores obtidos estão mostrados na Tabela 3.4.

Tabela 3.4 - Dados obtidos no ensaio de rotor bloqueado

<b>Ensaio de Rotor Bloqueado</b>	
Tensão (V)	51
Corrente (A)	1,3
Perdas Rotor Bloqueado (W)	75

Fonte: própria.

Com base no modelo da Figura 2.2 e com os dados obtidos nos ensaios, podemos encontrar os parâmetros da máquina.

Desconsiderando a resistência  $R_c$  no ramo de magnetização, do ensaio a vazio tem-se que a impedância a vazio é dada por

$$Z_{vz} = \frac{V_{vz}}{i_{vz}} = \frac{51}{1,3} = 39,231 \Omega \quad (3.1)$$

e a resistência e a reatância a vazio são dadas por

$$R_{vz} = \frac{P_{vz}}{3 \cdot i_{vz}^2} = \frac{75}{3 \cdot 1,3^2} = 11,793 \Omega \quad (3.2)$$

$$X_{vz} = \sqrt{Z_{vz}^2 - R_{vz}^2} = \sqrt{39,231^2 - 11,793^2} = 37,544 \Omega \quad (3.4)$$

$$X_{vz} = X_m + X_{ls} \quad (3.5)$$

Como o motor é de categoria N, a relação entre as reatâncias de dispersão estatóricas e rotóricas é

$$X_{ls} = 1,67 \cdot X_{lr} \quad (3.6)$$

Do ensaio de rotor bloqueado, pode-se fazer o mesmo procedimento.

$$Z_{rb} = \frac{V_{rb}}{i_{rb}} = \frac{51}{1,3} = 39,231 \Omega \quad (3.7)$$

$$R_{rb} = \frac{P_{rb}}{3 \cdot i_{rb}^2} = \frac{75}{3 \cdot 1,3^2} = 11,793 \Omega \quad (3.8)$$

$$X_{rb} = \sqrt{Z_{rb}^2 - R_{rb}^2} = \sqrt{39,231^2 - 11,793^2} = 37,544 \Omega \quad (3.9)$$

Para o um cálculo mais preciso do valor da reatância de dispersão do rotor é válida a equação

$$X_{lr} = (X_{rb} - X_{ls}) \frac{(X_{vz} - X_{ls})}{(X_{vz} - X_{rb})} \quad (3.10)$$

Logo, tem-se que o valor da reatância de dispersão do rotor é

$$X_{lr} = (36,335 - 1,67X_{lr}) \frac{(146,444 - 1,67X_{lr})}{(146,444 - 36,335)}$$

$$2,789X_{lr}^2 - 415,35X_{lr} + 5321 = 0$$

$$X_{lr} = 14,157 \Omega$$

De acordo com a Equação 3.6, o valor da reatância de dispersão do estator é

$$X_{ls} = 1,67X_{lr} = 1,67 \cdot 14,157 = 23,642 \Omega$$

Substituindo na Equação 3.5, o valor da reatância de magnetização é

$$X_m = X_{vz} - X_{ls} = 146,444 - 23,642 = 122,803 \Omega$$

Para um cálculo mais preciso do valor da resistência do rotor é válida a equação

$$R_r = (R_{rb} - R_s) \left( \frac{X_m + X_{lr}}{X_m} \right)^2 \quad (3.11)$$

Substituindo os valores anteriormente calculados, tem-se que o valor da resistência do rotor será

$$R_r = (14,793 - 5,35) \left( \frac{122,803 + 14,157}{122,803} \right)^2$$

$$R_r = 11,746 \Omega$$

Resolvendo as Equações 2.54 e 2.56,

$$\delta_r = 0,0309$$

$$\sigma = 0,248$$

Por fim, substituindo os valores encontrados nas Equações 2.55 e 2.57, obtemos

$$L_s = \frac{X_s}{\omega_b} = \frac{122,803 + 23,642}{2\pi \cdot 60} = 0,388 H$$

$$L_r = \frac{X_r}{\omega_b} = \frac{122,803 + 14,157}{2\pi \cdot 60} = 0,363 H$$

Portanto, os parâmetros elétricos do motor estão reunidos na Tabela 3.5.



Tabela 3.5 - Resumo dos parâmetros elétricos do motor

Parâmetro	Símbolo	Valor
<b>Resistência estatórica</b>	$R_s$ ( $\Omega$ )	5,350
<b>Resistência rotórica</b>	$R_r$ ( $\Omega$ )	11,746
<b>Reatância de dispersão estatórica</b>	$X_{ls}$ ( $\Omega$ )	23,642
<b>Reatância de dispersão rotórica</b>	$X_{lr}$ ( $\Omega$ )	14,157
<b>Reatância de magnetização</b>	$X_m$ ( $\Omega$ )	122,803
<b>Indutância de magnetização</b>	$L_m$ (H)	0,326
<b>Indutância própria estatórica</b>	$L_s$ (H)	0,388
<b>Indutância própria rotórica</b>	$L_r$ (H)	0,363
<b>Constante de tempo rotórica</b>	$\delta_r$ (s)	0,031
<b>Coefficiente de dispersão</b>	$\sigma$	0,248

Fonte: própria.

### 3.2 Parâmetros mecânicos

Se o motor for operado a vazio, em velocidade nominal, e se então for repentinamente desligado da fonte, o decaimento da velocidade do motor será determinado pelas perdas rotacionais como (Fitzgerald, 2006):

$$J \frac{d\omega_m}{dt} = -T_{rot} = -\frac{P_{rot}}{\omega_m} \quad (3.12)$$

Assim, como as perdas rotacionais  $P_{rot}$  são conhecidas para a velocidade a vazio (1795 RPM ou 187,972 rad/s), a inércia do conjunto rotor mais disco do freio  $J$  pode ser obtida à partir do decaimento da velocidade.

$$J = -\frac{P_{rot}}{\omega_m} \frac{\Delta t}{\Delta \omega_m} \quad (3.13)$$

Para maior acurácia, foram realizadas quatro medições do tempo que a velocidade do rotor levava para ir da velocidade nominal até zero quando a fonte de alimentação é retirada.

Tabela 3.6 – Tempo de desaceleração do rotor

Tempo (segundos)	
$\Delta t_1$	13,15
$\Delta t_2$	12,73
$\Delta t_3$	13,07
$\Delta t_4$	12,98
$\Delta t_{\text{médio}}$	<b>12,98</b>

Fonte: própria.

Da Equação 3.13,

$$J = -\frac{35}{187,97} \frac{(12,98 - 0)}{(0 - 187,97)} = 0,013 \text{ kg} \cdot \text{m}^3$$

A equação mecânica do motor de indução/carga é:

$$\frac{d\omega_m}{dt} = \frac{1}{J} (T_e - T_L - B\omega_m) \quad (3.14)$$

Onde,  $T_e$  é o conjugado eletromagnético gerado pela máquina,  $T_L$  o torque de carga presente no seu eixo e  $B$  o coeficiente de atrito viscoso.

Considerando o motor a vazio, o torque de carga  $T_L$  é nulo e o conjugado eletromagnético pode ser encontrado, de acordo com o modelo de regime permanente da máquina de indução, da seguinte forma

$$T_e = T_{mec} = \frac{P_{mec}}{\omega_m} = \frac{P_g}{\omega_s} = \frac{n_{fases} I_r'^2 (R_r/s)}{\omega_s} \quad (3.15)$$

Onde  $P_g$  é a potência no entreferro,  $\omega_s$  a velocidade angular síncrona,  $n_{fases}$  o número de fases do motor e  $I_r'$  a corrente do rotor referenciada ao estator.

A impedância equivalente do circuito é (onde o operador // significa *em paralelo*):

$$Z_{eq} = R_s + jX_{ls} + jX_m // (R_r/s_{vz} + jX_{lr}) \quad (3.16)$$

$$Z_{eq} = 8,913 + j146,329 \Omega$$

$$I_s = \frac{V_{vz}}{Z_{eq}} = 0,091 - j1,498 \text{ A}$$

A tensão no lado do rotor  $V_r$ , referenciado ao estator, é

$$V_r = V_{vz} - I_s(R_s + jX_{ls}) = 184,099 + j5,857 V$$

A corrente de magnetização  $I_\phi$  é calculada da seguinte forma

$$I_\phi = \frac{V_r}{jX_m} = 0,048 - j1,499 A$$

A corrente do rotor referenciada ao estator  $I'_r$  é

$$I'_r = I_s - I_\phi = 0,044 + j1,239 \times 10^{-3} A$$

O escorregamento do motor a vazio é:

$$s_{vz} = \frac{n_s - n_m}{n_s} = \frac{1800 - 1795}{1800} = 0,00278$$

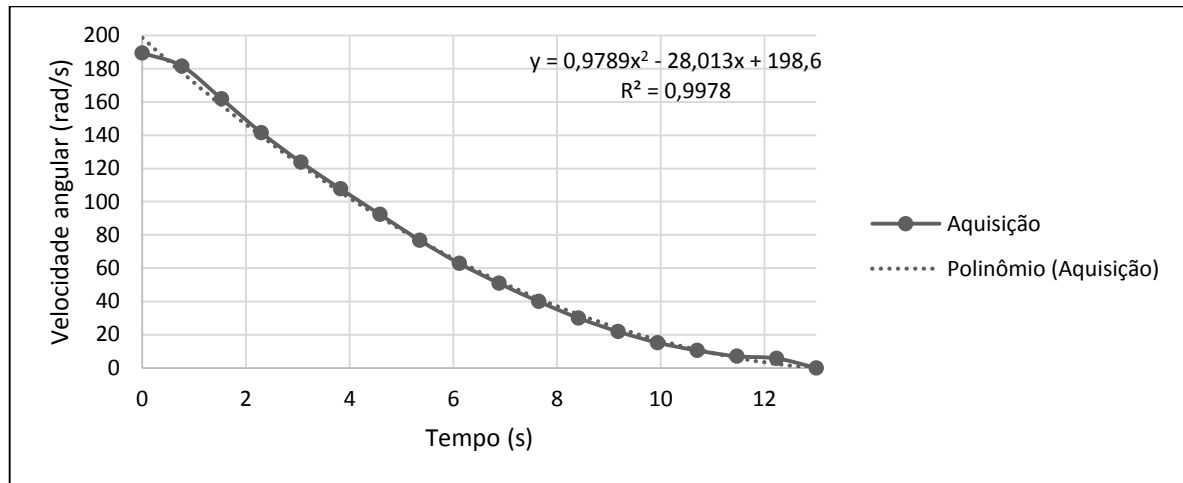
Substituindo em (3.15)

$$T_e = 0,128 N \cdot m$$

Podemos resolver (3.14) para encontrar  $B$ , obtendo a curva de desaceleração do motor. Este ensaio não é simples, pois exige um equipamento que obtenha com exatidão vários pontos durante a desaceleração do motor. O equipamento utilizado foi um *encoder* incremental bidirecional de 2500 pulsos por volta, os sinais chegam até o Processador Digital de Sinais (DSP) e este realiza as operações necessárias para retornar o valor da velocidade e se comunica com o computador por meio da JTAG e do *software* Code Composer Studio 6.1 (CCS 6.1). Maiores detalhes do *encoder* e da aquisição da velocidade com o DSP estão mostrados no Capítulo 6.

Na Figura 3.3 abaixo, temos os pontos obtidos da medição da velocidade durante a retirada da alimentação do motor. Uma linha de tendência é traçada para obter a curva aproximada de desaceleração.

Figura 3.3 – Gráfico da desaceleração do motor,  $\omega(t)$ , obtido com o DSP e o CCS



Fonte: própria.

A curva Aquisição pode ser aproximada pelo Polinômio(Aquisição), portanto pode-se escrever

$$\omega_m(t) = 0,9789t^2 - 28,013t + 198,6 \quad (3.17)$$

Derivando (3.17),

$$\frac{d\omega_m(t)}{dt} = 1,957t - 28,013 \quad (3.18)$$

Substituindo tudo em (3.14) e considerando  $t$  igual a 0, o coeficiente de atrito viscoso  $B$  é encontrado.

$$B = 2,598 \times 10^{-3} \text{ N} \cdot \text{m} \cdot \text{s}$$

Portanto, os parâmetros mecânicos do conjunto motor mais disco do freio estão reunidos na Tabela 3.7.

Tabela 3.7 - Resumo dos parâmetros mecânicos do conjunto motor mais disco do freio

Parâmetro	Símbolo	Valor
<b>Momento de inércia</b>	$J \text{ (kg} \cdot \text{m}^3)$	0,013
<b>Coefficiente de atrito viscoso</b>	$B \text{ (N} \cdot \text{m} \cdot \text{s)}$	$2,598 \times 10^{-3}$

Fonte: própria.

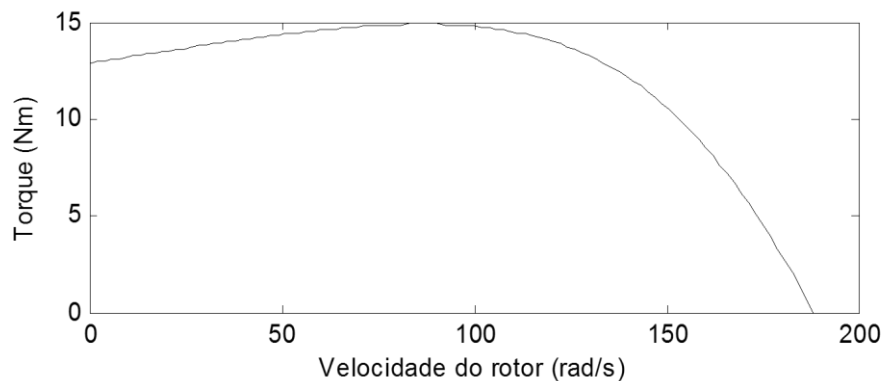
### 3.3 Curvas do motor

Nesta seção, foi implementado uma simulação do modelo  $dq0$  de referência estacionária do motor de indução. A simulação foi usada para examinar as características da máquina de indução trifásica operando como motor e alimentada com tensões senoidais. Para isto, utilizou-se a ferramenta computacional *Simulink*<sup>®</sup> de tal modo a visualizar o comportamento da máquina quando alimentado por uma fonte sem controle.

As curvas das Figuras de 3.4 a 3.7 foram obtidas a partir do modelo de regime permanente na máquina de indução, portanto, correspondem ao comportamento do motor em função da velocidade do motor quando em regime permanente, ou seja, após o término de todos os transitórios.

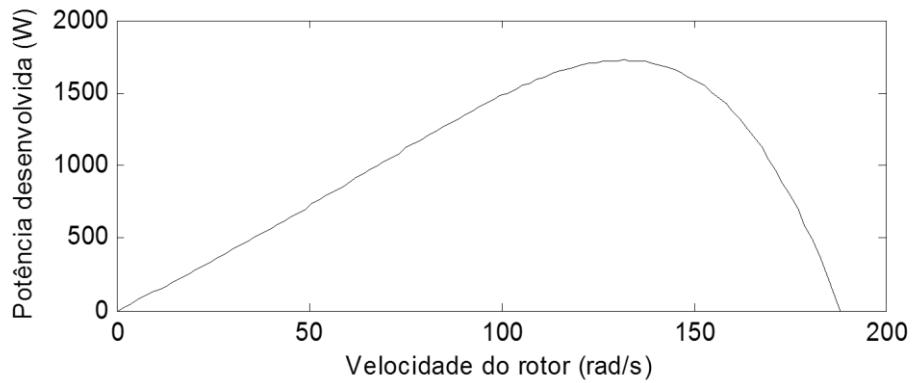
Primeiramente, iremos analisar o comportamento do motor em função da velocidade do rotor com tensão nominal aplicada e sem carga mecânica aplicada. Analisando a curva plotada, Figura 3.4, do torque desenvolvido na máquina, nota-se que o conjugado de partida ( $s = 1$ ) é de 12,9 Nm e o máximo conjugado (14,9 Nm) é desenvolvido para o escorregamento de 0,54.

Figura 3.4 – Resposta do torque desenvolvido pelo motor em função da velocidade do rotor



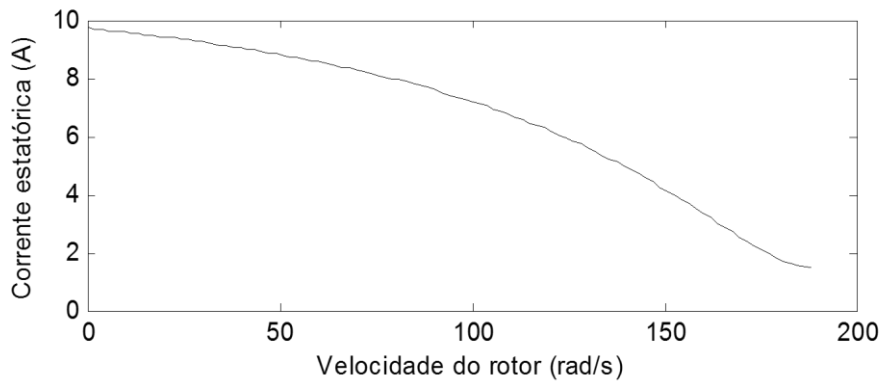
Fonte: própria.

Figura 3.5 – Resposta da potência do motor em função da velocidade do rotor



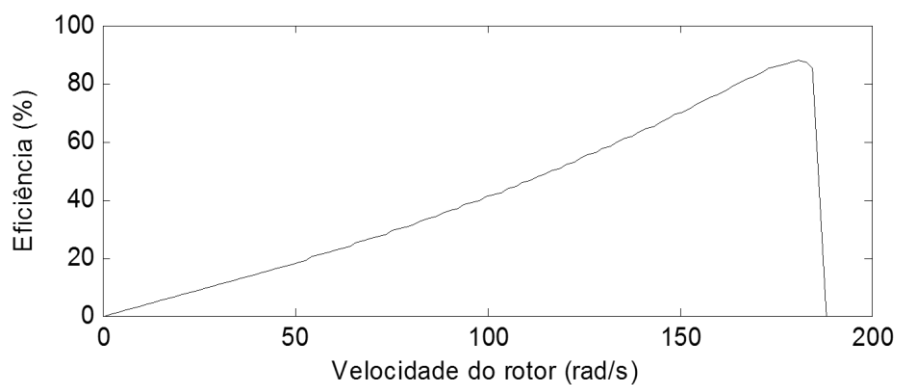
Fonte: própria.

Figura 3.6 – Corrente do estator em função da velocidade do rotor



Fonte: própria.

Figura 3.7 – Eficiência do motor em função da velocidade do rotor



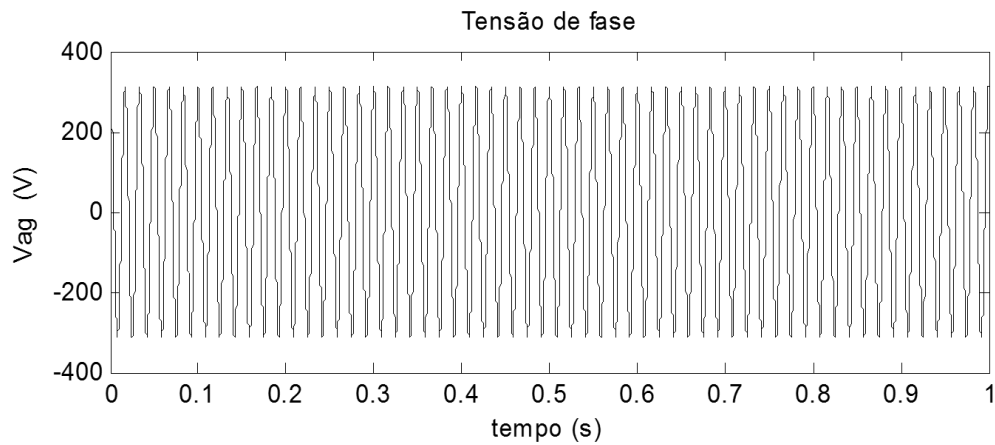
Fonte: própria.

Como pode ser visto na Figura 3.7, o ponto de máxima eficiência (87,8%) deste motor, de acordo com essas simulações, é para a velocidade do rotor de 181 rad/s, o que significa um escorregamento de 0,024.

A seguir temos a resposta transitória de  $i_{as}$ ,  $T_{em}$  e  $\omega_r$  durante a partida e com posterior aplicação de degraus de carga mecânica. Estes resultados são obtidos a partir da simulação do modelo  $qd0$  com referência estacionária do motor de indução.

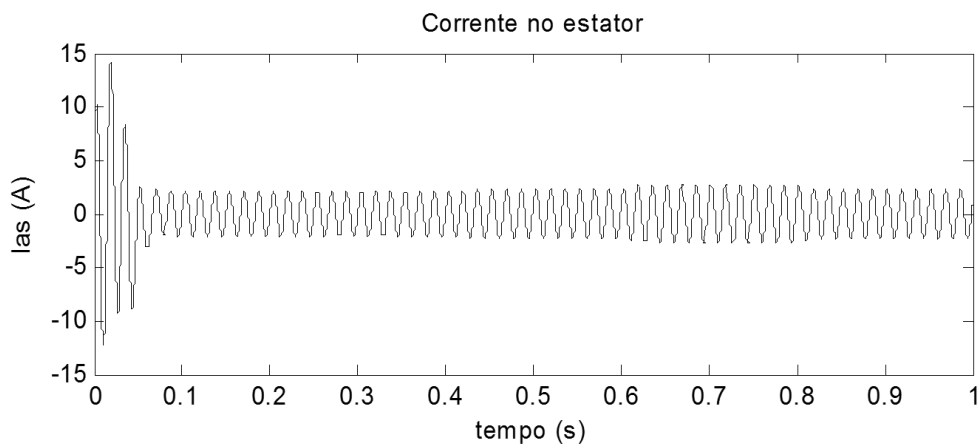
Aos 0,4 segundos um degrau de carga no valor de 50% do conjugado nominal é adicionado. No instante 0,6 s mais um degrau de carga é adicionado, chegando a 100% do nominal. E em 0,8 s é retirado 50% da carga.

Figura 3.8 – Tensão de alimentação



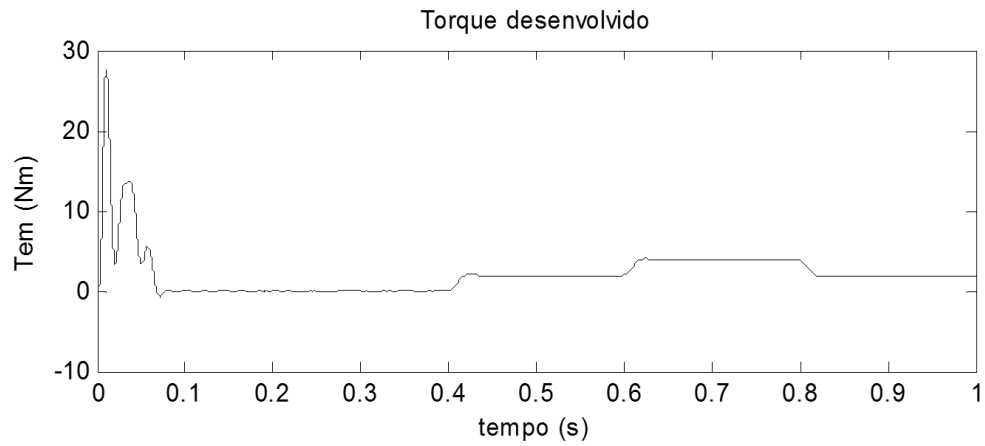
Fonte: própria.

Figura 3.9 – Corrente no estator



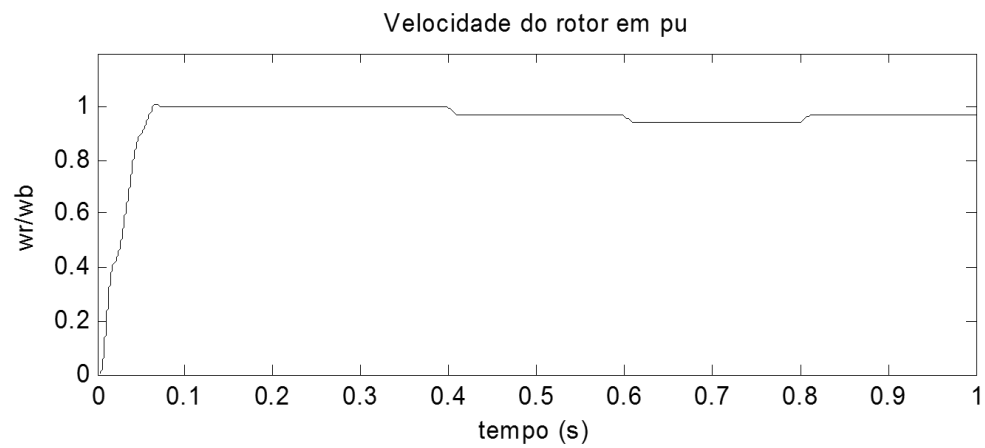
Fonte: própria.

Figura 3.10 – Torque desenvolvido



Fonte: própria.

Figura 3.11 – Velocidade do rotor



Fonte: própria.



## 4 PROJETO DOS CONTROLADORES PI

Com base nos modelos desenvolvidos nos capítulos anteriores e nos parâmetros obtidos, neste capítulo será mostrado o desenvolvimento de uma estratégia de controle que garanta o melhor desempenho do sistema. O controlador escolhido é do tipo PI (Proporcional e Integral) e para atender aos critérios de desempenho, é preciso fazer ajustes nos parâmetros deste controlador.

No controle vetorial desenvolvido (Figura 2.7), é preciso um controlador PI da malha de velocidade e, no caso de utilização do controlador PI síncrono para a corrente, é necessário outro PI para a malha de corrente. A partir das funções de transferência obtidas pela malha de corrente e velocidade, será usada a técnica de alocação de polos para que o sistema satisfaça às especificações de desempenho. No caso do controle de posição, um outro PI anterior ao controlador de velocidade é adicionado.

### 4.1 Controladores PI

Uma forma de controle bastante difundida na indústria é chamada de controlador PI, uma derivação do controlador de três termos ou controlador PID onde o termo derivativo é desprezado. Este controlador tem uma função de transferência (Dorf, 1998)

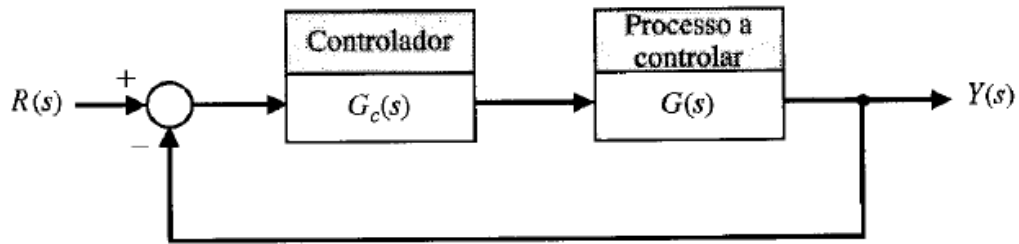
$$G_c(s) = K_p + \frac{K_i}{s} \quad (4.1)$$

O controlador oferece um termo proporcional e um termo integral. A equação para a saída no domínio do tempo é

$$u(t) = K_p e(t) + K_i \int e(t) dt \quad (4.2)$$

Sendo  $K_p$  a variável que determina o ganho proporcional,  $K_i$  a que determina o ganho integral,  $e(t)$  é o erro dado pela diferença entre o sinal de referência  $r(t)$  e o de saída  $y(t)$ , e  $u(t)$  representa o sinal de controle da planta. Os parâmetros que determinam os ganhos são ajustáveis. A Figura 4.1 a seguir representa o diagrama de blocos de um controlador.

Figura 4.1 – Sistema de controle a malha fechada de um controlador



Fonte: Dorf (1998).

A função de transferência de malha aberta (FTMA), de acordo com a Figura 4.1, é igual ao produto da função de transferência do controlador com a função de transferência da planta

$$FTMA = G_c(s) \cdot G(s) \quad (4.3)$$

Realizando a realimentação negativa, a função de transferência de malha fechada (FTMF) fica

$$FTMF = \frac{FTMA}{1 + FTMA} \quad (4.4)$$

A função de transferência de sistemas de segunda ordem é mostrada em (4.5), onde  $\xi$  representa o coeficiente de amortecimento do sistema,  $\omega_n$  a frequência natural de oscilação e  $K$  o ganho.

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (4.5)$$

O comportamento de sistemas de segunda ordem pode ser definido, em função do coeficiente de amortecimento, da seguinte forma:

- Sistema sobre amortecido ( $\xi > 1$ ): O sistema apresenta dois polos reais e distintos. Quanto maior for  $\xi$ , maior o sistema se aproximará do comportamento de um sistema de primeira ordem.
- Sistema criticamente amortecido ( $\xi = 1$ ): O sistema apresenta dois polos reais e iguais. Nesse caso o sistema não possui oscilação.
- Sistema subamortecido ( $0 < \xi < 1$ ): O sistema apresenta dois polos complexos conjugados. Nesse caso o sistema possui oscilações.

Para a implementação computacional do controlador PI é necessário fazer uma modificação, utiliza-se o modelo discreto. Os ganhos discretos  $Kp_z$  e  $Ki_z$  são dados por (Dorf, 1998)

$$Kp_z = K_p - \frac{K_i T_a}{2} \quad (4.6)$$

$$Ki_z = K_i T_a \quad (4.7)$$

Onde  $T_a$  é o tempo de amostragem e a equação da saída para a aplicação do controlador discreto é definida a seguir.

$$u(k) = u(k - 1) + Kp_z[e(k) - e(k - 1)] + Ki_z \cdot e(k) \quad (4.8)$$

#### 4.2 Planta da malha de velocidade

O modelo da malha de velocidade é equacionado a partir da equação mecânica (2.17) do sistema mecânico formado pelo conjunto motor mais carga mecânica. Repetindo (2.17) e explicitando  $d\omega_m/dt$ , temos

$$\frac{d\omega_m}{dt} = \frac{1}{J} (T_e - T_c - B\omega_m) \quad (4.9)$$

Considerando o torque de carga  $T_c$  uma perturbação, aplica-se a transformada de Laplace em (4.9) para análise no domínio da frequência. Onde  $\Omega_m$  é a transformada de Laplace de  $\omega_m$ .

$$s\Omega_m(s) = \frac{1}{J} [T_e - B\Omega_m(s)] \quad (4.10)$$

Sendo  $T_e$  a entrada do sistema e  $\Omega_m$  a saída, a função de transferência da malha de velocidade fica

$$\frac{\Omega_m(s)}{T_e(s)} = \frac{\frac{1}{J}}{s + \frac{B}{J}}$$

Dividindo o numerador e o denominador por  $B/J$ , tem-se a função de transferência

$$G_{\omega}(s) = \frac{\Omega_m(s)}{T_e(s)} = \frac{\frac{J}{B}}{s\frac{1}{B} + 1} = \frac{\beta_{\omega}}{\tau_{\omega}s + 1} \quad (4.11)$$

$$\beta_{\omega} = \frac{J}{B} \quad (4.12)$$

$$\tau_{\omega} = \frac{1}{B} \quad (4.13)$$

Onde  $\beta_{\omega}$  representa o ganho e  $\tau_{\omega}$  a constante de tempo da malha de velocidade.

Substituindo os parâmetros da Tabela 3.7 em (4.11), a dinâmica da malha de velocidade é representada a seguir

$$G_{\omega}(s) = \frac{\Omega_m(s)}{T_e(s)} = \frac{384,85}{4,956s + 1} \quad (4.14)$$

A corrente no eixo de referência q,  $i_q^s$ , é dependente da saída do controlador de velocidade, para (4.11) ser válida, pressupomos que o controle de corrente é ideal e a corrente de referência é perfeitamente aplicada ao motor.

### 4.3 Cálculo dos parâmetros do controlador de velocidade

A função de transferência de malha aberta (FTMA), de acordo com (4.3), é

$$FTMA_{\omega} = \left( Kp_{\omega} + \frac{Ki_{\omega}}{s} \right) \left( \frac{\beta_{\omega}}{\tau_{\omega}s + 1} \right) \quad (4.15)$$

Realizando a realimentação negativa, a função de transferência de malha fechada (FTMF) fica

$$FTMF_{\omega} = \frac{FTMA_{\omega}}{1 + FTMA_{\omega}} = \frac{\frac{Kp_{\omega}\beta_{\omega}}{\tau_{\omega}}s + \frac{Ki_{\omega}\beta_{\omega}}{\tau_{\omega}}}{s^2 + s\left(\frac{1 + Kp_{\omega}\beta_{\omega}}{\tau_{\omega}}\right) + \frac{Ki_{\omega}\beta_{\omega}}{\tau_{\omega}}} \quad (4.16)$$

Pela função de transferência de malha fechada do sistema, nota-se que trata-se de um sistema de segunda ordem. Desta forma, os parâmetros do PI são estabelecidos a partir da função de transferência do sistema de segunda ordem (4.5) e seus critérios de desempenho. A presença de um zero em (4.16) provocará um efeito que irá diferir do esperado com base em

(4.5), a presença deste zero significará um *overshoot* (ultrapassagem) proporcional a respostas rápidas, porém não irá afetar no desempenho do tempo de acomodação.

O tempo de acomodação  $t_s$  é o tempo necessário para que a curva de resposta alcance valores e permaneça dentro de uma faixa em torno do valor da referência. Adotando o critério de 2% de erro do valor final, o tempo de acomodação do sistema de segunda ordem pode ser aproximado por

$$t_s = \frac{4}{\xi \omega_n} \quad (4.17)$$

Comparando a equação característica, polinômio do denominador, de (4.5) e (4.16) obtemos a igualdade

$$s^2 + s \left( \frac{1 + Kp_\omega \beta_\omega}{\tau_\omega} \right) + \frac{Ki_\omega \beta_\omega}{\tau_\omega} = s^2 + 2\xi \omega_n s + \omega_n^2 \quad (4.18)$$

Mediante essa igualdade, os ganhos do controlador são definidos

$$Kp_\omega = \frac{2\xi \omega_n \tau_\omega - 1}{\beta_\omega} \quad (4.19)$$

$$Ki_\omega = \frac{\tau_\omega \omega_n^2}{\beta_\omega} \quad (4.20)$$

Isolando  $\omega_n$  em (4.17) e substituindo em (4.19) e (4.20), obtemos os ganhos proporcional e integral do controlador de velocidade

$$Kp_\omega = \frac{8\tau_\omega - t_s}{t_s \beta_\omega} \quad (4.21)$$

$$Ki_\omega = \frac{16\tau_\omega}{t_s^2 \xi^2 \beta_\omega} \quad (4.22)$$

De acordo com (4.21) e (4.22) vemos que os ganhos do controlador de velocidade dependem dos parâmetros mecânicos da máquina de indução, do coeficiente de amortecimento e do tempo de acomodação. Os parâmetros da máquina são constantes e o coeficiente de amortecimento e o tempo de acomodação são escolhidos mediante critério utilizado.

#### 4.4 Planta da malha de corrente

Agora definiremos a planta da malha de corrente a partir das equações dinâmicas da máquina de indução no eixo de referência  $dq$  como mostrado no Capítulo 2. Mediante manipulação de (2.25) a (2.32) obtemos (4.23) e (4.24), nota-se que existe uma relação dinâmica entre a corrente de quadratura estatórica e sua tensão, o mesmo acontece para a corrente de eixo direto e sua tensão (Barbi, 1986).

$$p i_{qs}^e = -\omega_e i_{ds}^e - \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right) i_{qs}^e - \frac{L_m}{L_s L_r \sigma} \omega_r \lambda_{dr}^e + \frac{L_m}{L_s L_r \sigma \tau_r} \lambda_{qr}^e + \frac{1}{L_s \sigma} v_{qs}^e \quad (4.23)$$

$$p i_{ds}^e = - \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right) i_{ds}^e + \omega_e i_{qs}^e + \frac{L_m}{L_s L_r \sigma \tau_r} \lambda_{dr}^e + \frac{L_m}{L_s L_r \sigma} \omega_r \lambda_{qr}^e + \frac{1}{L_s \sigma} v_{ds}^e \quad (4.24)$$

Ponderando as considerações presente no controle por orientação do fluxo, o termo em quadratura do fluxo do rotor é nulo. As variações lentas de velocidade  $\omega_r$  e do fluxo  $\lambda_{dr}^e$  têm o comportamento minimizado pela ação do integrador, podendo assim ser desprezados. Mediante essas considerações, a nova equação que representa a dinâmica da malha de corrente é mostrada a seguir.

$$\frac{d i_{qs}^e}{dt} = - \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right) i_{qs}^e + \frac{1}{L_s \sigma} v_{qs}^e \quad (4.25)$$

Aplicando a transformada de Laplace em (4.25)

$$s I_{qs}^e(s) = - \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right) I_{qs}^e(s) + \frac{1}{L_s \sigma} V_{qs}^e(s) \quad (4.26)$$

Isolando a entrada  $V_{qs}^e(s)$  e a saída  $I_{qs}^e(s)$  obtém-se a função de transferência

$$\frac{I_{qs}^e(s)}{V_{qs}^e(s)} = \frac{\frac{1}{L_s \sigma}}{s + \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right)} \quad (4.27)$$

É notável que a malha de corrente é de primeira ordem. Substituindo o numerador e o denominador por  $\left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right)$ , temos

$$G_{I_{qs}^e}(s) = \frac{I_{qs}^e(s)}{V_{qs}^e(s)} = \frac{\frac{1}{L_s \sigma} / \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right)}{s / \left( \frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r} \right) + 1} = \frac{\beta_{I_{qs}^e}}{\tau_{I_{qs}^e} s + 1} \quad (4.28)$$

Onde

$$\beta_{I_{qs}^e} = \frac{\frac{1}{L_s \sigma}}{\frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r}} \quad (4.29)$$

$$\tau_{I_{qs}^e} = \frac{1}{\frac{R_s}{L_s \sigma} + \frac{1 - \sigma}{\sigma \tau_r}} \quad (4.30)$$

De acordo com os parâmetros elétricos do motor de indução utilizado, Tabela 3.5, a função de transferência da malha de corrente fica

$$G_{I_{qs}^e}(s) = \frac{0,068}{6,515 \times 10^{-3} s + 1} \quad (4.31)$$

O mesmo pode ser feito com a malha da corrente de eixo direto  $i_{ds}^e$  de acordo com (4.24), a partir disso, a função de transferência para a malha de corrente de eixo direto pode ser representado por

$$G_{I_{ds}^e}(s) = \frac{I_{ds}^e(s)}{V_{ds}^e(s)} = \frac{0,068}{6,515 \times 10^{-3} s + 1} \quad (4.32)$$

Mediante os valores obtidos, é possível ver que as considerações feitas estavam corretas, pois a constante de tempo da malha de velocidade é da ordem de 1000 vezes maior do que a constante de tempo das malhas de corrente.

#### 4.5 Cálculo dos parâmetros do controlador de corrente

Para o cálculo dos ganhos do controlador de corrente, é repetido o mesmo procedimento da seção 4.3. Como a função de transferência da malha de corrente, assim como a da malha de velocidade, é de primeira ordem, assim como (4.21) e (4.22) podemos escrever os ganhos do controlador de corrente como

$$Kp_{I_{qs}^e} = \frac{8\tau_{I_{qs}^e} - t_s}{t_s \beta_{I_{qs}^e}} \quad (4.33)$$

$$Ki_{I_{qs}^e} = \frac{16\tau_{I_{qs}^e}}{t_s^2 \xi^2 \beta_{I_{qs}^e}} \quad (4.34)$$

#### 4.6 Planta e cálculo dos parâmetros da malha de posição

Por fim, agora será mostrado o procedimento para obtenção da função de transferência da posição angular do rotor considerando como entrada a velocidade angular do rotor. Considera-se ainda que os controles posteriores, de velocidade e de corrente, são ideais.

É sabido que

$$\omega = \frac{d\theta}{dt} \quad (4.35)$$

Aplicando a transformada de Laplace e dividindo entrada  $\omega$  pela saída  $\theta$ , temos

$$G_\theta(s) = \frac{\Theta(s)}{\Omega(s)} = \frac{1}{s} \quad (4.36)$$

Onde  $\Theta(s)$  é a transformada de Laplace da posição  $\theta$  e  $\Omega(s)$  é a da velocidade  $\omega$ .

A função de transferência de malha aberta (FTMA), de acordo com (4.3), é

$$FTMA_\theta = \left( Kp_\theta + \frac{Ki_\theta}{s} \right) \left( \frac{1}{s} \right) \quad (4.37)$$

Realizando a realimentação negativa, a função de transferência de malha fechada (FTMF) fica

$$FTMF_\theta = \frac{FTMA_\theta}{1 + FTMA_\theta} = \frac{Kp_\theta s + Ki_\theta}{s^2 + sKp_\theta + Ki_\theta} \quad (4.38)$$

Portanto, trata-se de um sistema de segunda ordem. Comparando a equação característica, polinômio do denominador, de (4.5) e (4.38) obtemos a igualdade

$$s^2 + sKp_\theta + Ki_\theta = s^2 + 2\xi\omega_n s + \omega_n^2 \quad (4.39)$$

Mediante essa igualdade, os ganhos do controlador são definidos

$$Kp_\theta = 2\xi\omega_n = \frac{8}{t_s} \quad (4.40)$$



$$Ki_{\theta} = \frac{16}{t_s^2 \xi^2} \quad (4.41)$$

#### 4.7 Resultados dos ganhos dos controladores

Utilizando os parâmetros elétricos e mecânicos da máquina de indução obtidos no Capítulo 3 e os modelos desenvolvidos nas seções anteriores, esta seção mostra os resultados dos ganhos dos controladores PI na malha de velocidade, posição e corrente. Tais ganhos serão utilizados na simulação do acionamento por controle vetorial da máquina de indução e na implementação prática.

Como visto no início deste capítulo, quanto maior o coeficiente de amortecimento  $\xi$ , mais o comportamento do sistema vai se assemelhar ao comportamento de um sistema de primeira ordem. Este comportamento é desejável pois não possui ultrapassagem da referência (*overshoot*), não possui oscilações e chega rapidamente à referência, o que é imprescindível em sistemas de posicionamento ou onde se deseja velocidade constante.

A sintonia dos controladores deve levar em conta as particularidades de cada malha, por isso, não é sabido utilizar os mesmos critérios de sintonia para malhas que representam fenômenos diferentes. Conforme cada malha, será utilizado um coeficiente de amortecimento maior que a unidade, comportamento sobre amortecido, e o tempo de acomodação igual a uma fração da respectiva constante de tempo da malha.

##### 4.7.1 Sintonia do controlador PI da malha de velocidade

Como critério de desempenho, é considerado que o tempo de acomodação  $t_s$  para a malha de velocidade é igual a uma fração da sua respectiva constante de tempo,

$$t_s = \frac{\tau_{\omega}}{124},$$

É usado um coeficiente de amortecimento  $\xi = 2$ . Como já comentado, devido ao fato de a constante de tempo da malha mecânica ser 1000 vezes mais lenta que a constante de tempo elétrica e a corrente no motor ser considerada igual a corrente de referência, isso nos permite analisar o sistema de forma desacoplada.

Para obtenção dos ganhos discretos de cada controlador usado para implementação prática dos controles via DSP, utilizou-se (4.6) e (4.7) sendo o tempo de amostragem  $T_a$  configurado no processador igual a 5  $\mu$ s (microsegundos) nos testes de velocidade.

Substituindo em (4.21) e (4.22), obtém-se os ganhos do controlador de velocidade. Na Tabela 4.1 temos o resumo dos parâmetros desta malha.

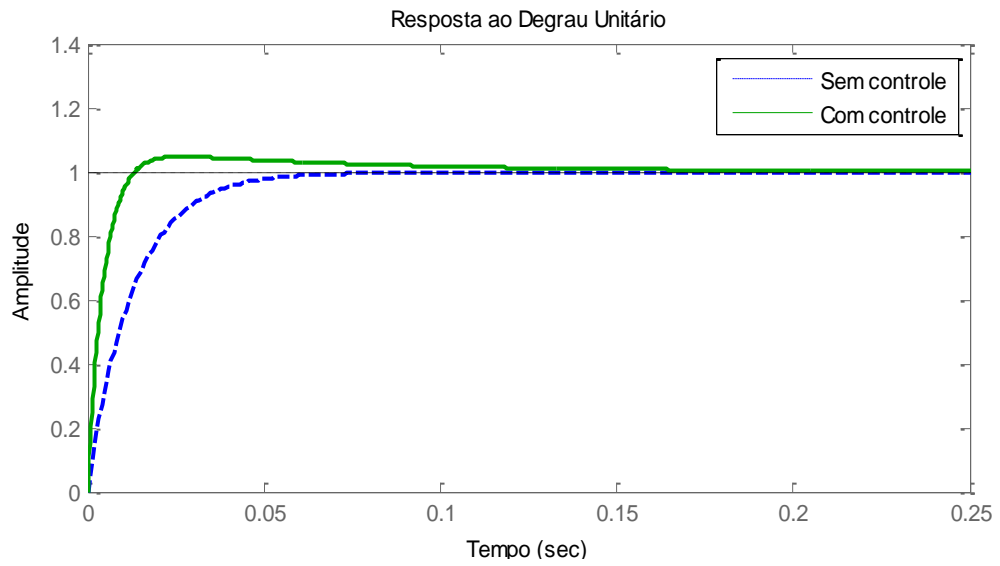
Tabela 4.1 – Valores dos ganhos do controlador PI da malha de velocidade

<b>Parâmetro</b>	<b>Valor</b>
$\tau_\omega$	4,956 s
$\beta_\omega$	384,85
$\xi$	2
$t_s$	0,04 s
$Kp_\omega$	2,575
$Ki_\omega$	32,247
$T_a$	$5 \cdot 10^{-6}$ s
$Kpz_\omega$	2,575
$Kiz_\omega$	$1,612 \cdot 10^{-4}$

Fonte: Própria.

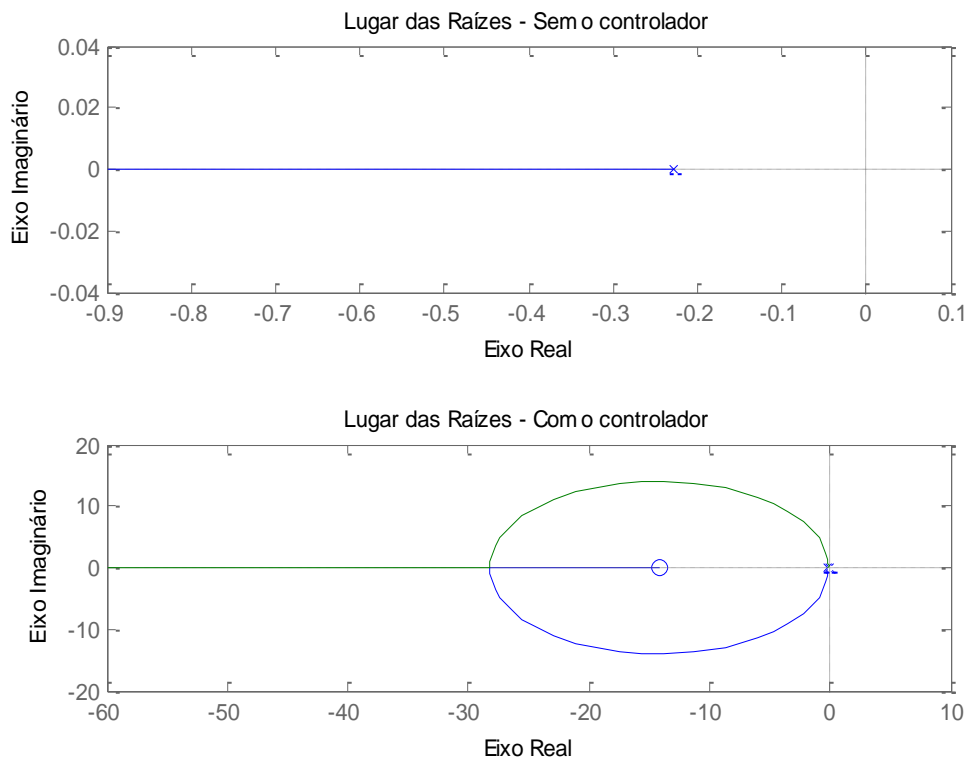
Na Figura 4.2 pode-se observar o comparativo entre a resposta a uma entrada em forma de degrau unitário no sistema com e sem o controlador. Nota-se que o controlador dá um avanço de fase e, mediante a presença do termo integrador, fará o sistema seguir a referência em caso de distúrbios na carga.

Figura 4.2 – Resposta da malha de velocidade a um degrau unitário com o controlador (vermelho) e sem controlador (azul tracejada)



Fonte: Própria.

Figura 4.3 – Lugar das raízes da malha de velocidade



Fonte: Própria.

#### 4.7.2 Sintonia do controlador PI da malha de corrente

Como critério de desempenho, é considerado que o tempo de acomodação  $t_s$  para a malha de corrente é igual a uma fração da sua respectiva constante de tempo,

$$t_s = \frac{\tau_i}{2},$$

É usado um coeficiente de amortecimento  $\xi = 2$ . Considera-se que a tensão no motor aplicada pelo inversor é idealmente igual a tensão de saída deste controlador, isso nos permite analisar o sistema sem considerar o chaveamento.

Para obtenção dos ganhos discretos usado para implementação prática dos controles via DSP, utilizou-se (4.6) e (4.7) sendo o tempo de amostragem  $T_a$  configurado no processador igual a 10  $\mu$ s (microsegundos).

Substituindo em (4.33) e (4.34), obtém-se os ganhos do controlador de corrente. Na Tabela 4.2 temos o resumo dos parâmetros desta malha.

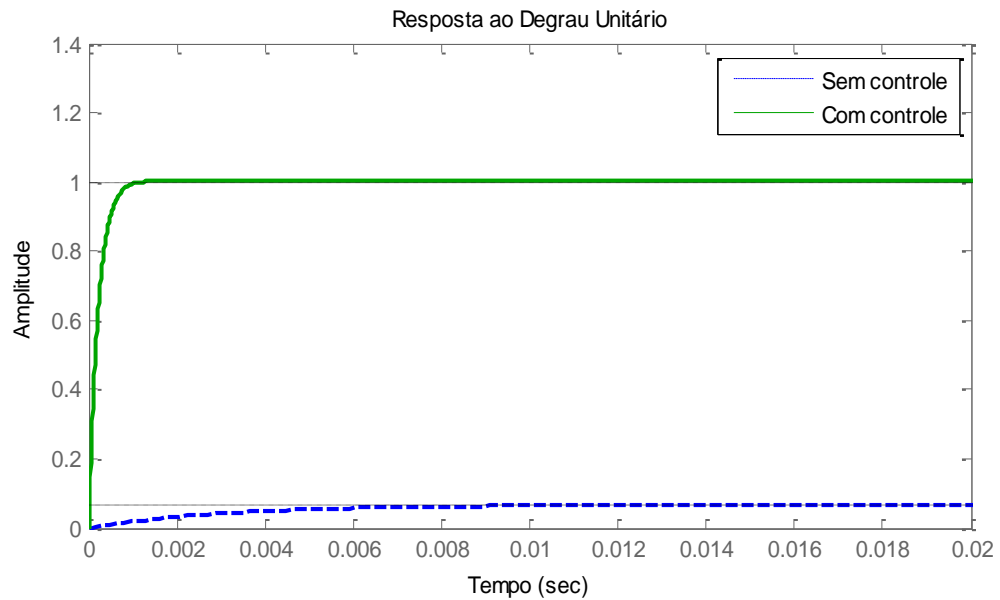
Tabela 4.2 – Valores dos ganhos do controlador PI da malha de corrente

<b>Parâmetro</b>	<b>Valor</b>
$\tau_i$	$6,525 \cdot 10^{-3}$ s
$\beta_i$	0,068
$\xi$	2
$t_s$	$3,258 \cdot 10^{-3}$ s
$Kp_i$	221,893
$Ki_i$	$3,633 \cdot 10^4$
$T_a$	$1 \cdot 10^{-5}$ s
$Kpz_i$	221,712
$Kiz_i$	0,363

Fonte: Própria.

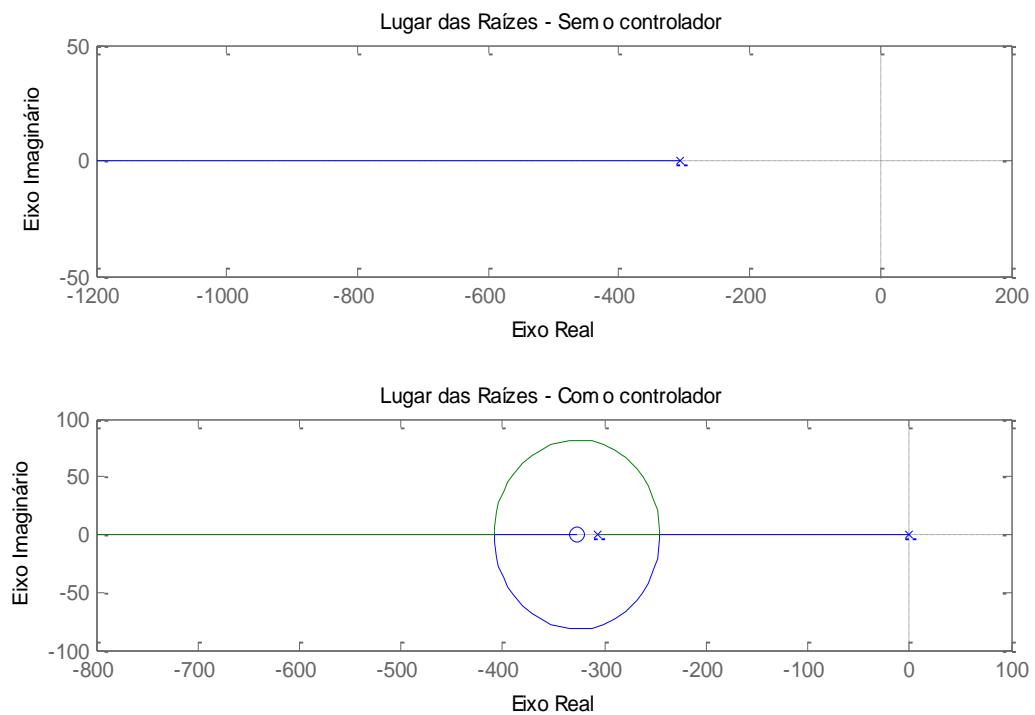
Na Figura 4.4 pode-se observar o comparativo entre a resposta a uma entrada em forma de degrau unitário no sistema com e sem o controlador. Nota-se que o controlador dá um avanço de fase e, mediante a presença do termo integrador, faz o sistema seguir a referência.

Figura 4.4 – Resposta da malha de corrente a um degrau unitário com o controlador (vermelho) e sem controlador (azul tracejada)



Fonte: Própria.

Figura 4.5 – Lugar das raízes da malha de corrente



Fonte: Própria.

### 4.7.3 Sintonia do controlador PI da malha de posição

Como critério de desempenho, é considerado que o tempo de acomodação  $t_s$  para a malha de posição é igual a uma fração da sua respectiva constante de tempo,

$$t_s = \frac{\tau_\theta}{8},$$

É usado um coeficiente de amortecimento  $\xi = 8$ . Considera-se ideal os controladores de velocidade e de corrente.

Para obtenção dos ganhos discretos usado para implementação prática dos controles via DSP, utilizou-se (4.6) e (4.7) sendo o tempo de amostragem  $T_a$  configurado no processador igual a 10  $\mu$ s (microssegundos).

Substituindo em (4.40) e (4.41), obtém-se os ganhos do controlador de posição. Na Tabela 4.3 temos o resumo dos parâmetros desta malha.

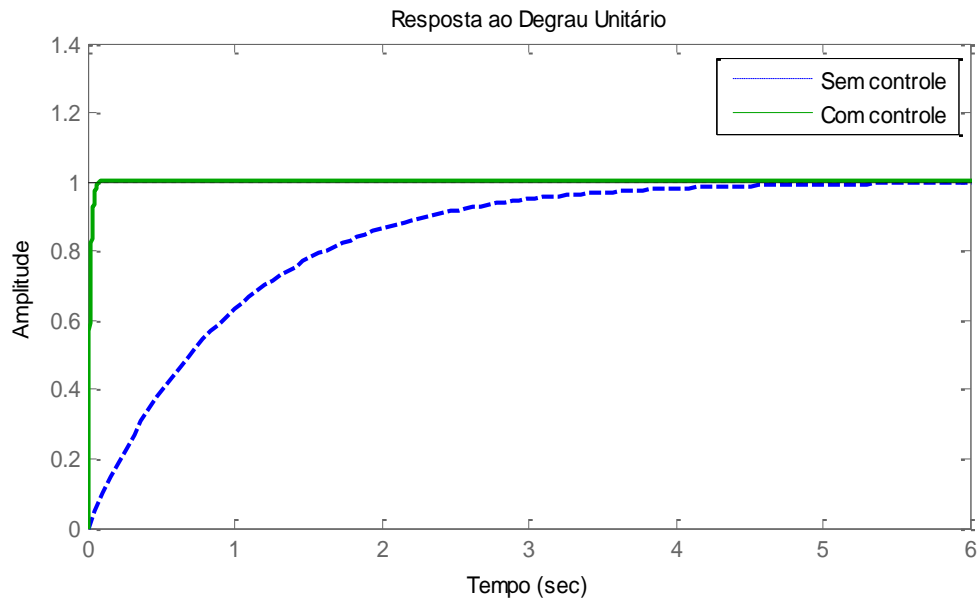
Tabela 4.3 – Valores dos ganhos do controlador PI da malha de posição

Parâmetro	Valor
$\tau_\theta$	1 s
$\beta_\theta$	1
$\xi$	8
$t_s$	0,125 s
$Kp_\theta$	64
$Ki_\theta$	16
$T_a$	$1 \cdot 10^{-5}$ s
$Kpz_\theta$	64
$Kiz_\theta$	$1,6 \cdot 10^{-4}$

Fonte: Própria.

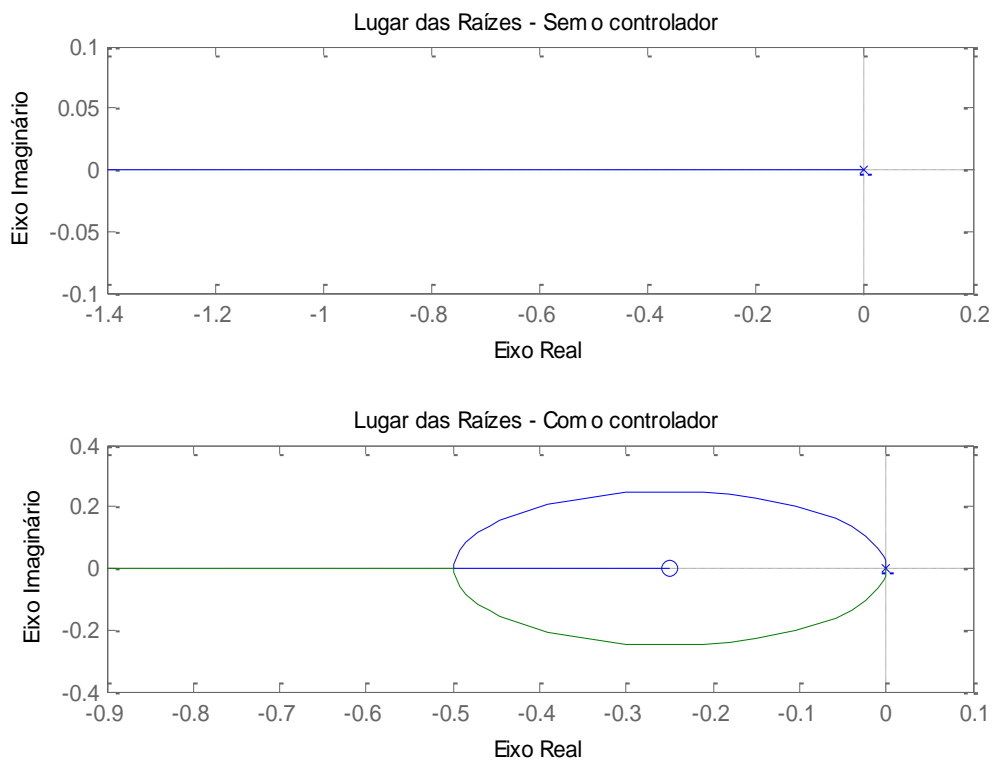
Na Figura 4.6 pode-se observar o comparativo entre a resposta a uma entrada em forma de degrau unitário no sistema com e sem o controlador. Nota-se que o controlador dá um avanço de fase e, mediante a presença do termo integrador, fará o sistema seguir a referência em caso de distúrbios na carga.

Figura 4.6 – Resposta da malha de posição a um degrau unitário com o controlador (vermelho) e sem controlador (azul tracejada)



Fonte: Própria.

Figura 4.7 – Lugar das raízes da malha de posição



Fonte: Própria.

## 5 RESULTADOS DA SIMULAÇÃO COMPUTACIONAL

Este capítulo apresenta a aplicação da técnica de controle vetorial para o controle de velocidade ou posição de um motor de indução. Pretende-se analisar o desempenho dinâmico do acionamento da máquina de indução com as estratégias e controladores utilizados através de simulações. Será mostrado o comportamento do controle com base nos controladores dimensionados a partir do método de alocação de polos. As simulações foram realizadas com a utilização da ferramenta MATLAB R2010a e *Simulink*<sup>®</sup>.

Para tanto, são definidos ciclos de trabalho constituindo uma sequência de alterações de referência de velocidade, referência de posição e de torque de carga no motor. Ao longo desse ciclo são levantadas curvas de grandezas necessárias para a análise do desempenho do controle e comprovar sua funcionalidade.

Três ensaios serão realizados, cada um com seu respectivo ciclo de trabalho. Os dois primeiros ensaios são respectivos ao controle de velocidade da máquina e o terceiro ao controle de posição. Os três ensaios e seus respectivos ciclos de trabalho são:

1. Teste de velocidade a vazio: com o motor sem carga mecânica em seu eixo, o eixo do rotor deve seguir uma velocidade de referência que cresce em rampa, estabiliza, e em seguida, decresce, inverte o sentido, estabiliza novamente e, por fim, diminui em rampa até parar.
2. Teste de velocidade com carga: ainda sem carga mecânica, o motor deve seguir uma velocidade que vai de zero ao valor nominal, em rampa, e nesta velocidade deve permanecer. Após alguns instantes é adicionado 100% da carga nominal no eixo da máquina, em seguida cargas mecânicas no eixo são retiradas e adicionadas em taxas de 100 e 50% da carga nominal.
3. Teste de posicionamento: o eixo do rotor deve seguir o sinal de referência de posição que faz um degrau de 0,25 radianos em relação a sua posição inicial e deve permanecer nessa posição, cargas mecânicas no eixo são adicionadas e retiradas em taxas de 100 e 50% da carga nominal.

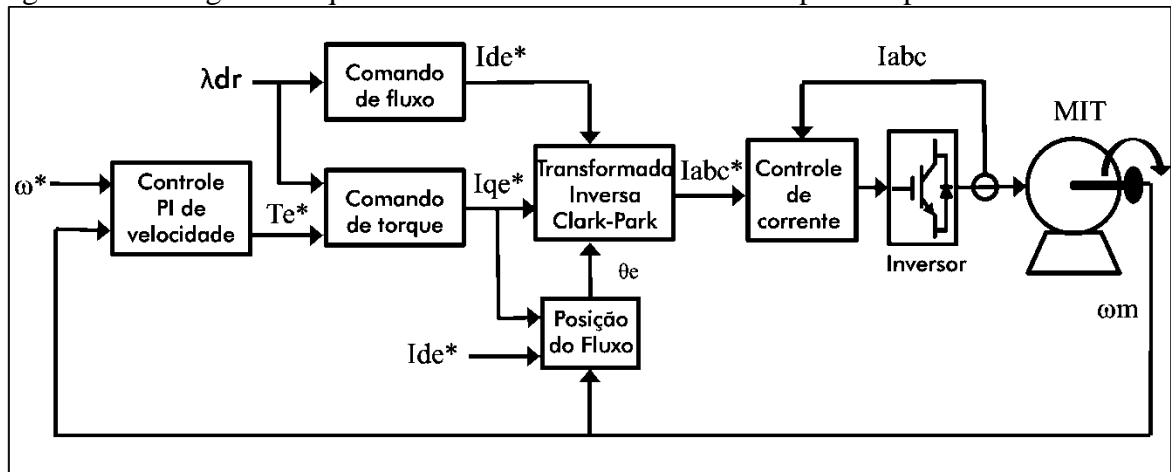
A simulação é realizada considerando um nível de fluxo igual ao fluxo nominal, a componente  $i_{ds}^e$  é ajustada para estabelecer tal nível de fluxo quando em alinhamento. Utilizou-se o controlador PI síncrono para a corrente e considerou-se o motor ligado em delta, portanto, algumas transformações entre variáveis de linha e de fase foram necessárias. Devemos examinar quão bem o controle manteve o fluxo constante durante as variações de carga, e observar a resposta dinâmica deste método de controle.



Para não estender muito o tempo de simulação, os chaveamentos do PWM no inversor foram omitidos, e apenas a componente fundamental da tensão de saída do PWM foi considerada.

O diagrama de blocos do sistema utilizando controle vetorial de velocidade e um motor de indução trifásico, equivalente ao da Figura 2.6, é mostrado, superficialmente, na Figura 5.1.

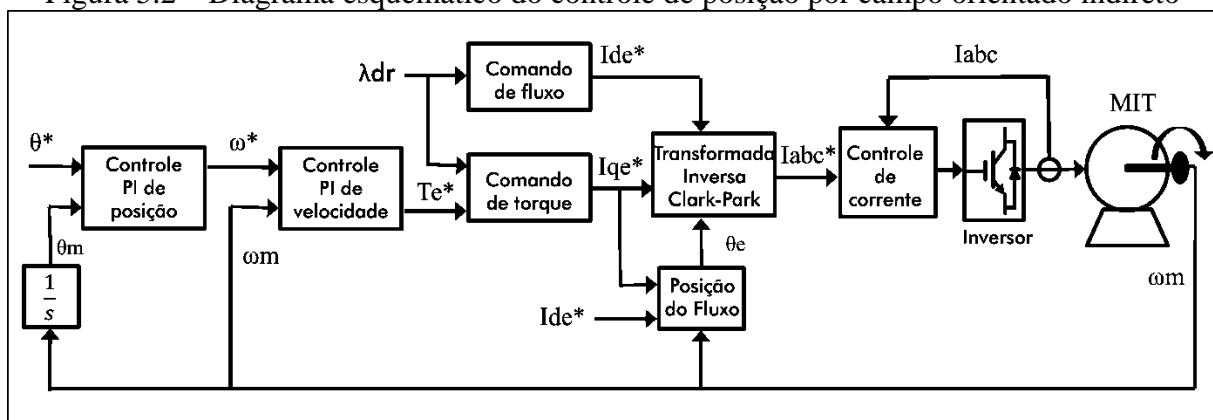
Figura 5.1 – Diagrama esquemático do controle de velocidade por campo orientado indireto



Fonte: Própria.

Para o controle de posição do eixo do motor, Figura 5.2, um controlador PI é adicionado antes do controlador PI de velocidade. Um sinal de referência de posição é definido e realimentado pela integração do sinal de velocidade, a saída deste controlador é a velocidade de referência  $\omega^*$ .

Figura 5.2 – Diagrama esquemático do controle de posição por campo orientado indireto

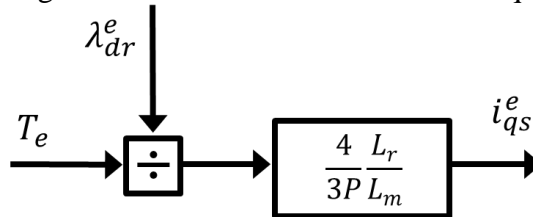


Fonte: Própria.

Na esquerda do diagrama, o controlador PI de posição converte o erro de posição na velocidade de referência, e o controlador PI de velocidade converte o erro de velocidade no

conjugado de referência. O bloco de Comando de torque é definido pela equação (2.49) e pode ser visto na Figura 5.3, assim como o de comando de fluxo é definido por (2.48) e está na Figura 5.4.

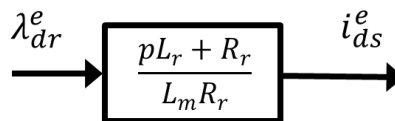
Figura 5.3 – Bloco do Comando de torque



Fonte: Própria.

Como o fluxo é mantido constante, o termo em quadratura da corrente do estator fica dependente apenas do conjugado. Na Figura 5.4 temos o bloco de comando do fluxo, onde  $p$  é  $d/dt$ , como o fluxo é constante, a corrente de eixo direto do estator pode ser calculada apenas por  $i_{ds}^e = \lambda_{dr}^e / L_m$ .

Figura 5.4 – Bloco do comando de fluxo



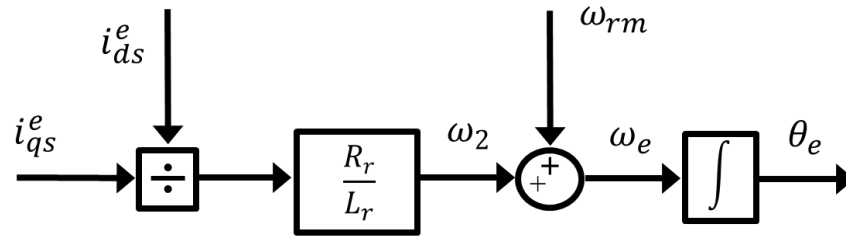
Fonte: Própria.

Do ensaio a vazio, obteve-se que a corrente a vazio é de 1,3 A (valor eficaz). Passando para a forma vetorial trifásica e efetuando as transformadas de Clark e Park, concluímos que a corrente de eixo direto a vazio, responsável pelo fluxo, é 1,8 A. Podemos encontrar o fluxo do motor que deve ser mantido constante aplicando (2.48).

$$\lambda_r^e = \lambda_{dr}^e = \frac{L_m R_r}{pL_r + R_r} i_{ds}^e = L_m i_{ds}^e = 0,586 \text{ Wb}$$

A estimação do fluxo é feita pela integração da velocidade angular do fluxo, obtido a partir da adição da velocidade de escorregamento, adquirido por (2.51), com a velocidade mecânica em radianos elétricos por segundo.

Figura 5.5 – Bloco do Estimador da posição do fluxo do rotor

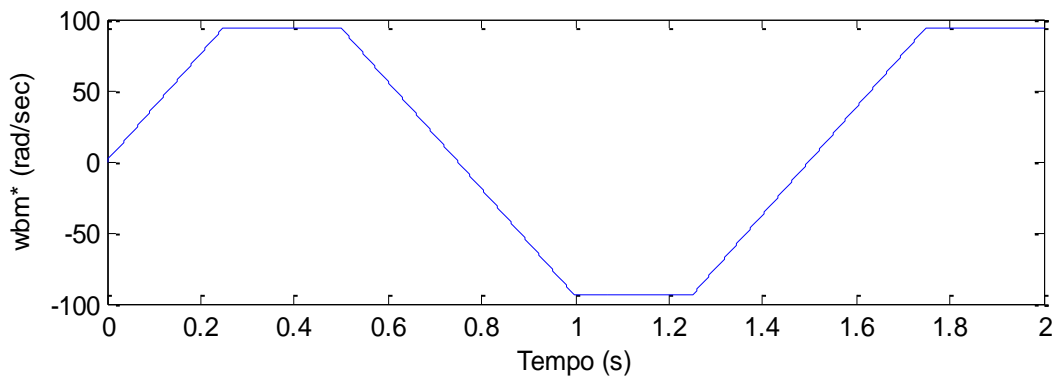


Fonte: Própria.

### 5.1 Teste de controle de velocidade a vazio

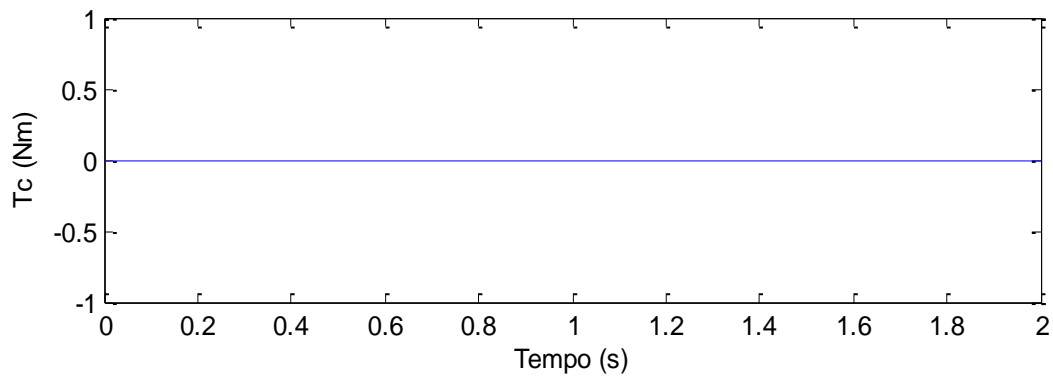
De acordo com esquema da Figura 5.1, nesta seção será realizado a simulação do controle de velocidade no primeiro ciclo de trabalho. As Figuras 5.6 e 5.7 definem este ciclo de trabalho, não há carga no eixo do motor e a velocidade varia em rampa numa aceleração de  $377 \text{ rad/s}^2$  até a metade da velocidade nominal.

Figura 5.6 – Velocidade de referência no teste a vazio



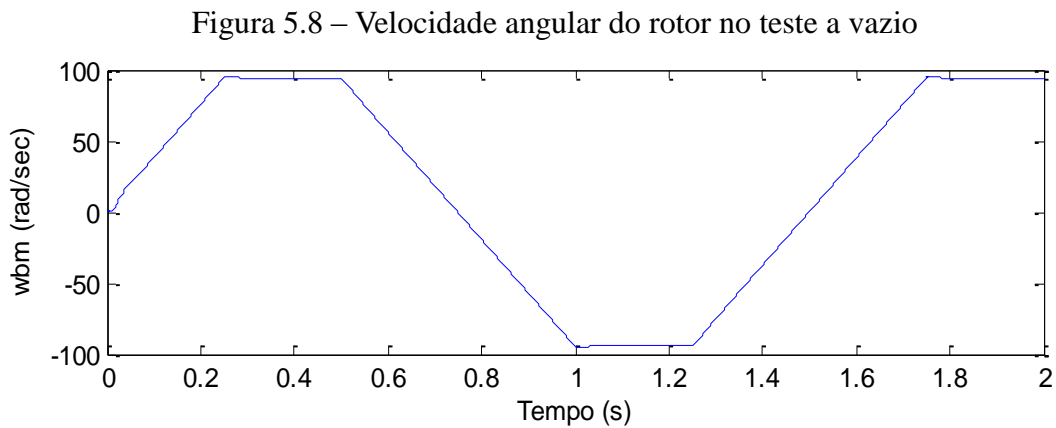
Fonte: Própria.

Figura 5.7 – Carga mecânica no teste a vazio



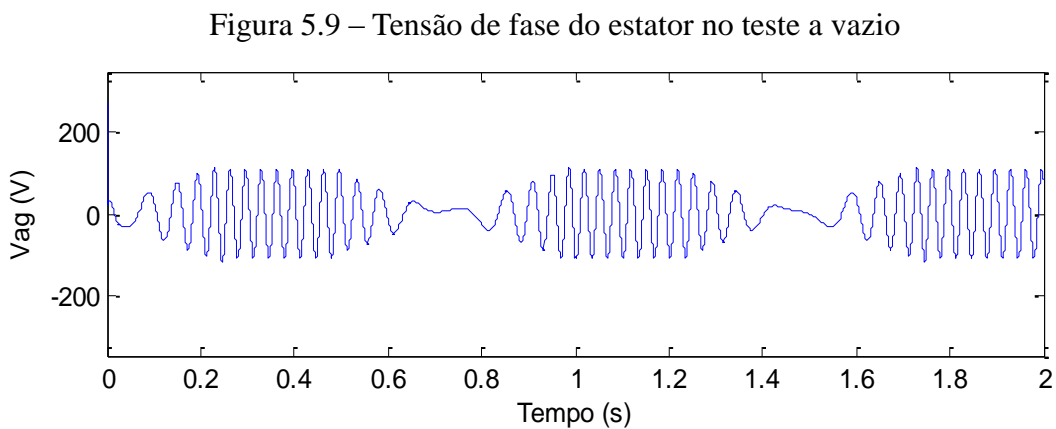
Fonte: Própria.

A Figura 5.8 mostra como a dinâmica do sistema se comportou durante o acionamento do motor de indução.



Fonte: Própria.

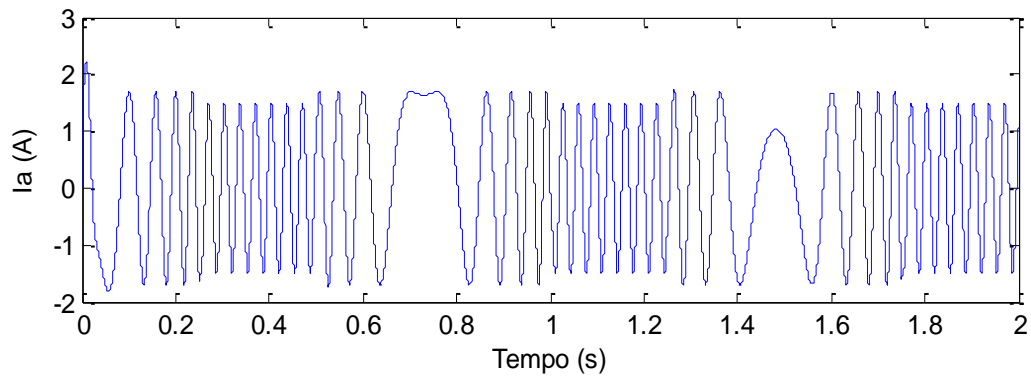
Observa-se que o controle se comportou da forma esperada, a velocidade seguiu a referência com rápida acomodação, comprovando os ganhos dos controladores PI calculados. Nas Figuras 5.9 e 5.10 temos o comportamento da tensão e da corrente de fase do estator.



Fonte: Própria.

Aplicando o controle vetorial juntamente com uma velocidade de referência em rampa na saída, a corrente de partida do motor é minimizada, aumentando a vida útil do motor. As correntes do motor na partida direta e o trabalho em sobrecarga são os maiores fatores responsáveis pela perda de vida útil do motor.

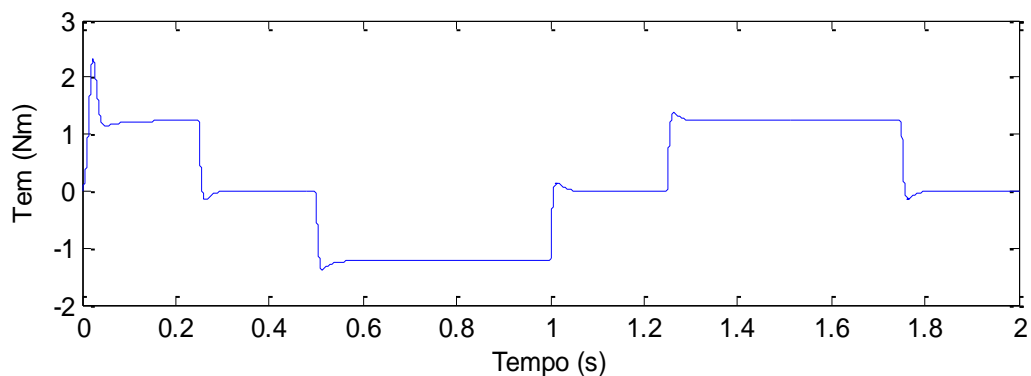
Figura 5.10 – Corrente do estator no teste a vazio



Fonte: Própria.

Como mostrado na Figura 5.11, nos períodos de aceleração é desenvolvido conjugado eletromagnético, ou seja, a componente em quadratura da corrente estatórica é não nula devido ao erro de velocidade no controlador PI de velocidade.

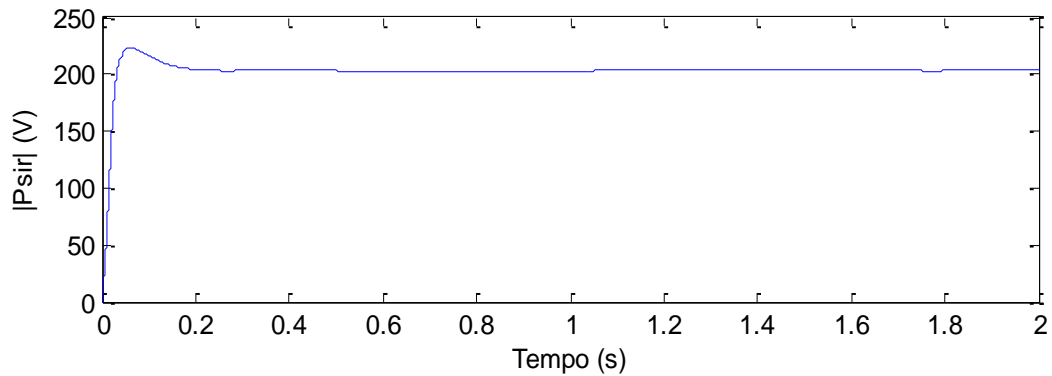
Figura 5.11 – Conjugado eletromagnético desenvolvido no teste a vazio



Fonte: Própria.

Observa-se na Figura 5.12 como o fluxo do rotor,  $\sqrt{\psi_{qr}^2 + \psi_{dr}^2}$ , demora cerca de 0,2 segundos para se estabelecer e ali permanece constante, assim como a componente de eixo direto da corrente do estator. Nota-se também que no início do segundo ciclo da rampa de subida de velocidade, o fluxo do rotor não passa pelo mesmo transitório que acontece no início do primeiro ciclo, devido ao fato de o fluxo do motor já estar completamente estabelecido e o controle forçar que o fluxo permaneça constante apesar de a velocidade do motor ter reduzido a zero.

Figura 5.12 – Fluxo do rotor no teste a vazio



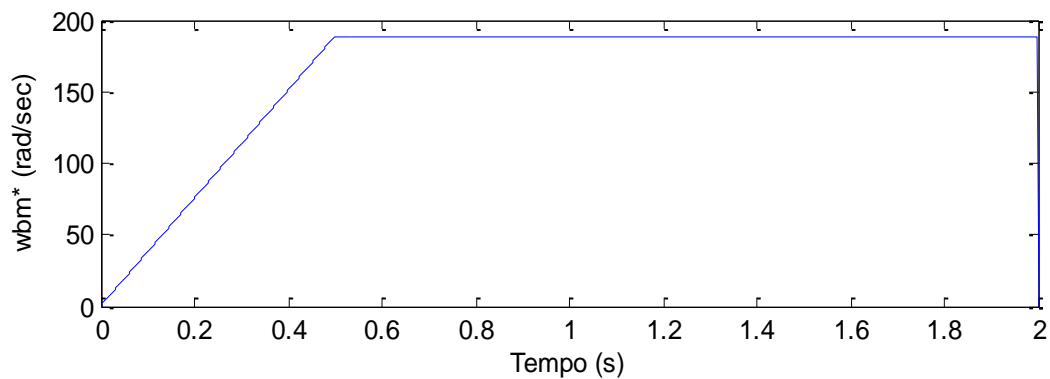
Fonte: Própria.

Como observado no resultado das simulações, a corrente de eixo direto assume o fluxo da máquina, enquanto que a componente em quadratura assume o conjugado eletromagnético, necessário nas acelerações e mudanças de cargas. Se considerarmos acionar o motor de indução com a corrente de quadratura nula, somente com a corrente de eixo direto, veremos que a máquina não parte, apenas estabelece o fluxo no entreferro mas não possui torque para partir.

## 5.2 Teste de controle de velocidade com carga

Neste segundo teste de velocidade, será considerado o ciclo de trabalho onde a velocidade de referência é conforme mostrado na Figura 5.13 e a carga mecânica conforme a Figura 5.14. A velocidade aumenta com aceleração de  $377 \text{ rad/s}^2$ , atingindo  $188,5 \text{ rad/s}$  (1800 RPM) em 0,5 segundos.

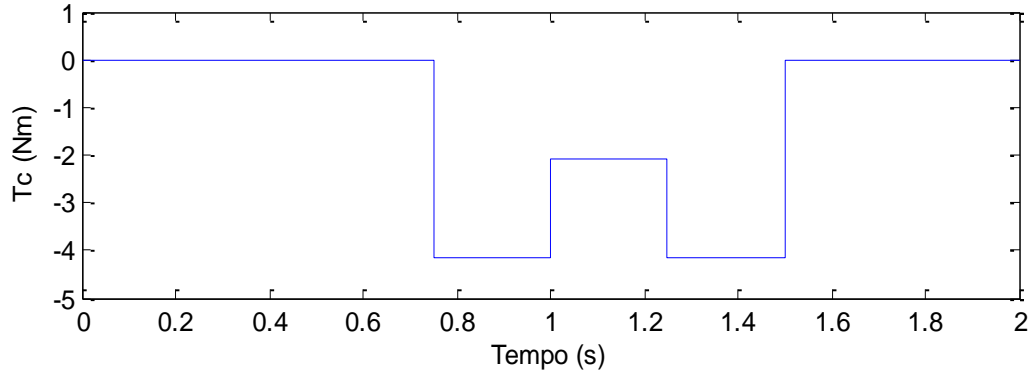
Figura 5.13 – Velocidade de referência no teste com carga



Fonte: Própria.

O comportamento do controle de velocidade é posto à prova com as variações de carga definidas na Figura 5.14. Cargas mecânicas aplicadas no eixo se opõem ao sentido de rotação do motor e, portanto, possuem o sinal negativo.

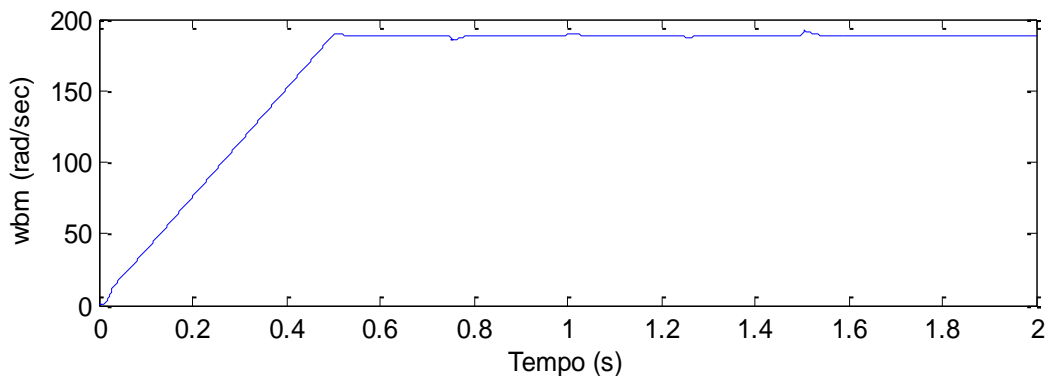
Figura 5.14 – Carga mecânica no teste com carga



Fonte: Própria.

Foi visto na Figura 3.11 que o motor de indução quando acionado por partida direta, sem controle, não alcança a velocidade padrão para uma máquina de quatro polos, 1800 RPM, devido ao escorregamento, e, quando há adição de carga, a diferença fica ainda maior. Como resultado principal desta simulação, a velocidade do rotor é mostrada na Figura 5.15. Observa-se que a entrada de carga promove uma desaceleração na máquina, e o erro de velocidade ocasiona o aumento da componente  $i_{qs}^e$ , que acarreta também aumentando a velocidade de escorregamento  $\omega_2$ , portanto, o controle trata a adição de carga com o aumento do módulo e da frequência da corrente (Figura 5.17).

Figura 5.15 – Velocidade angular do rotor no teste com carga

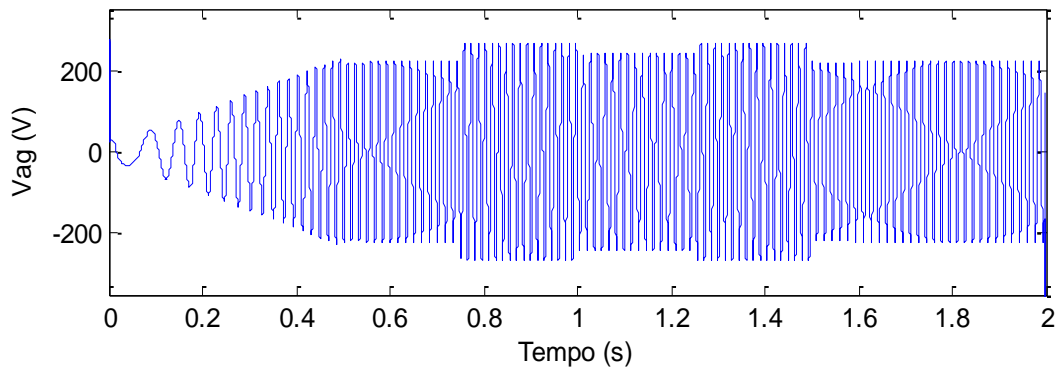


Fonte: Própria.

De forma semelhante ao controle escalar, onde a relação tensão por velocidade é mantida constante para também manter o torque constante, na Figura 5.16 vemos que o que

acontece no controle vetorial é semelhante. Como a velocidade de referência aumenta em rampa, o módulo da tensão e a sua frequência seguem o mesmo comportamento.

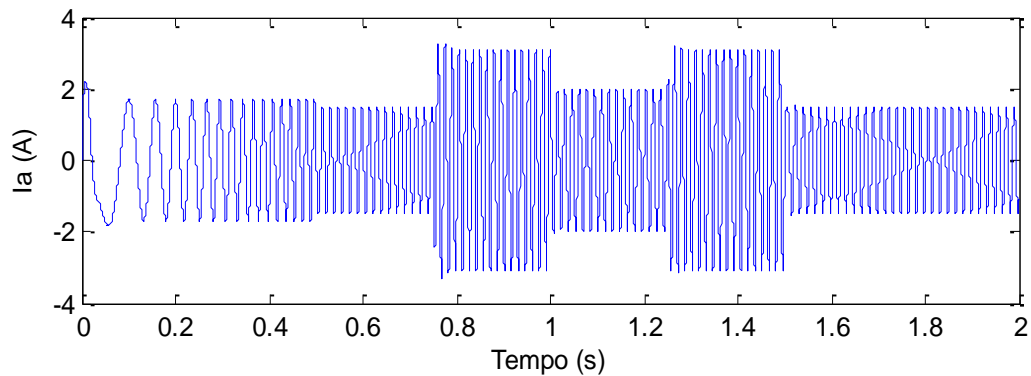
Figura 5.16 – Tensão de fase do estator no teste com carga



Fonte: Própria.

Comparando os resultados do acionamento do motor com partida direta e através do controle vetorial, nota-se que na partida direta a corrente de partida chegava a 15 A (Figura 3.9) enquanto que a corrente obtida através do controle vetorial possui um pico na partida de apenas 2 A.

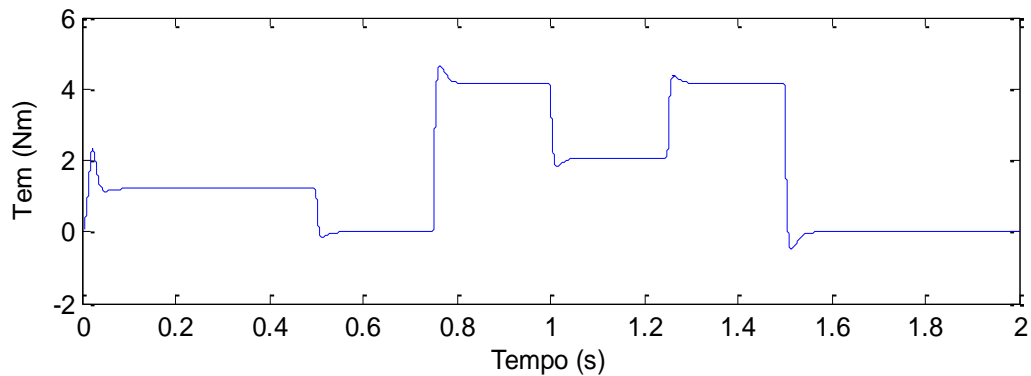
Figura 5.17 – Corrente do estator no teste com carga



Fonte: Própria.



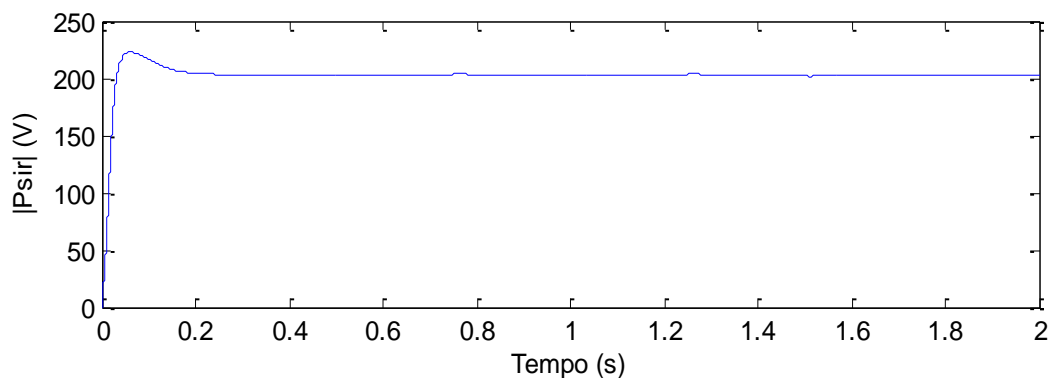
Figura 5.18 – Conjugado eletromagnético desenvolvido no teste com carga



Fonte: Própria.

Confrontando o resultado do fluxo obtido no acionamento do teste a vazio (Figura 5.12) e do teste com carga (Figura 5.19), é notável que apesar da variação de carga aplicada ao eixo do motor, o fluxo ainda se manteve, exatamente como no ensaio a vazio. Isto ocorre pois a corrente de eixo direto, responsável pelo fluxo, é mantida constante e as variações de carga são compensadas pela componente em quadratura.

Figura 5.19 – Fluxo do rotor no teste com carga

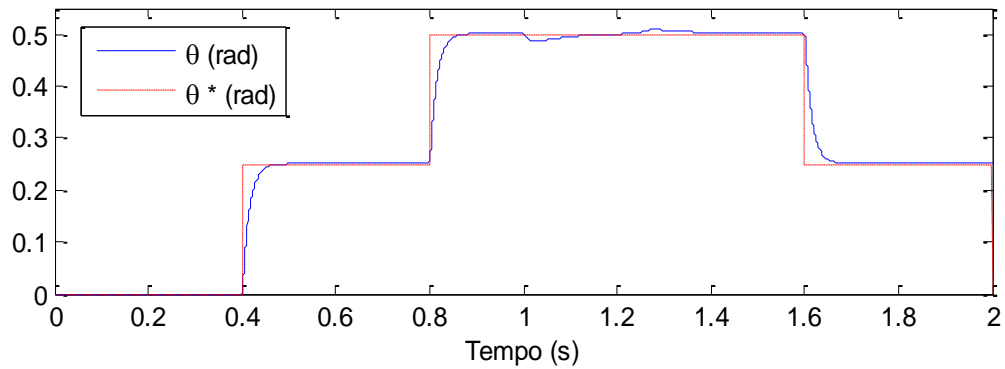


Fonte: Própria.

### 5.3 Teste de controle de posição

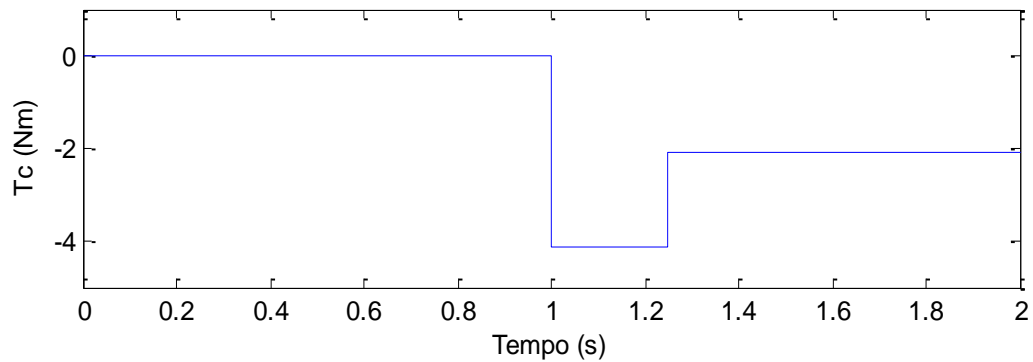
O terceiro teste trata do controle da posição angular do rotor do motor de indução conforme o diagrama de blocos mostrado na Figura 5.2 e os ganhos dos controladores PI obtidos no capítulo 4. O ciclo de trabalho abrange a posição de referência,  $\theta^*$ , mostrada na Figura 5.20 (tracejado) e a carga aplicada ao eixo mostrada na Figura 5.21. Na posição de referência são aplicados degraus de 0,25 radianos (equivalente a  $14,32^\circ$ ) e obtém-se os resultados apresentados na Figura 5.20 (traçado contínuo).

Figura 5.20 – Posição de referência (tracejado) e posição do rotor (contínua)



Fonte: Própria.

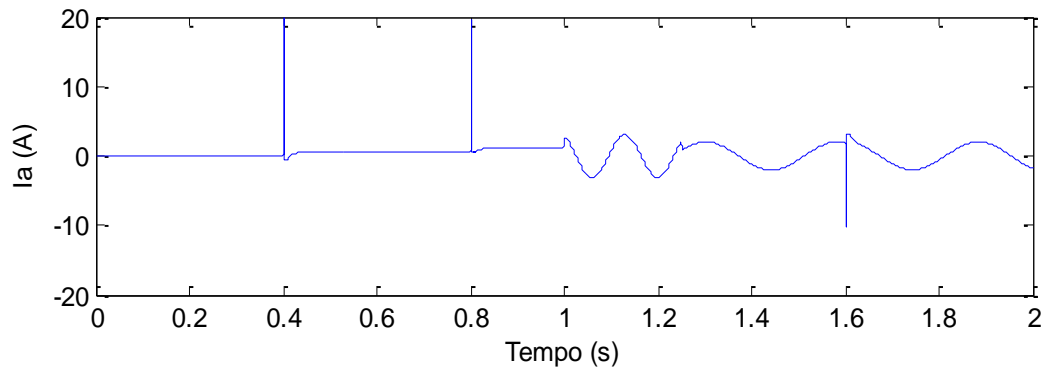
Figura 5.21 – Carga mecânica no teste de controle de posição



Fonte: Própria.

Para se manter a posição constante, quando não há torque contrário, a corrente no motor de indução é mantida constante (Figura 4.22), porém, quando existe torque externo aplicado ao eixo, uma corrente alternada é aplicada no estator para compensar e manter a posição angular constante.

Figura 5.22 – Corrente do estator no teste de controle de posição

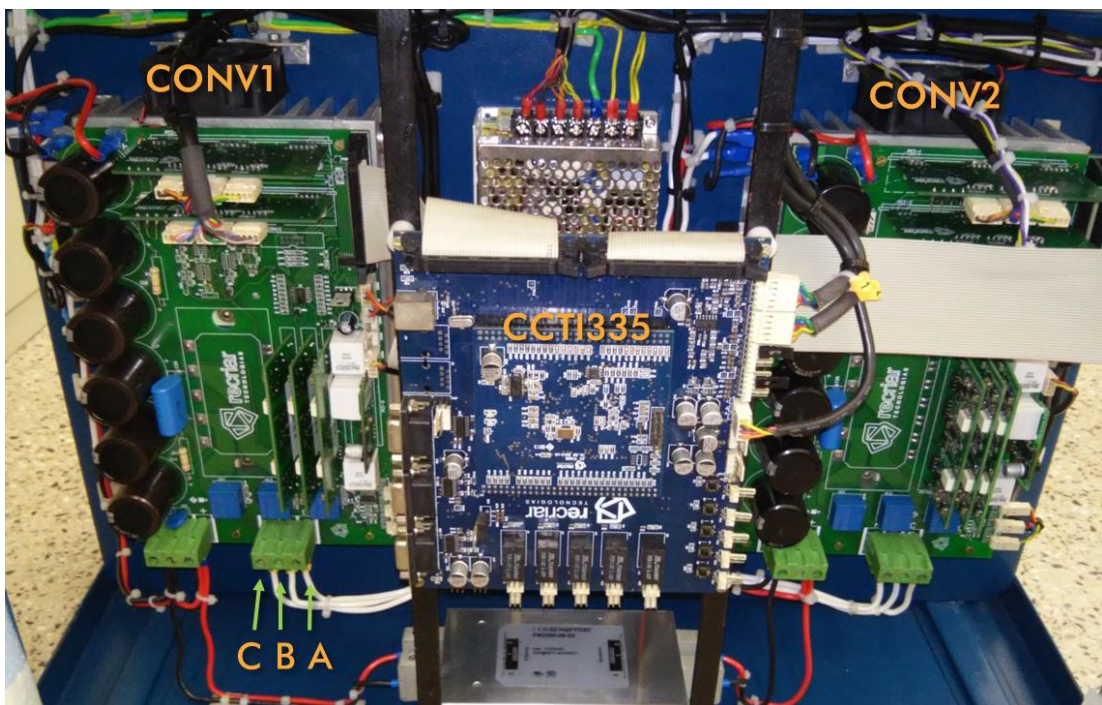


Fonte: Própria.

## 6 APLICAÇÃO PRÁTICA

Para aplicar experimentalmente o controle vetorial, utilizou-se a bancada apresentada na Figura 6.1. Nesta bancada há uma placa de expansão *back-to-back*, são dois inversores trifásicos com o barramento CC compartilhado. As chaves dos inversores são transistores IGBT de 15 A e o barramento CC é limitado a 450 V. A placa de controle utilizada é o Cartão de Controle CCTI335 da *Texas Instruments* e é comandada pelo processador de sinais DSP TI F28335.

Figura 6.1 – Bancada *back-to-back* utilizada na implementação prática

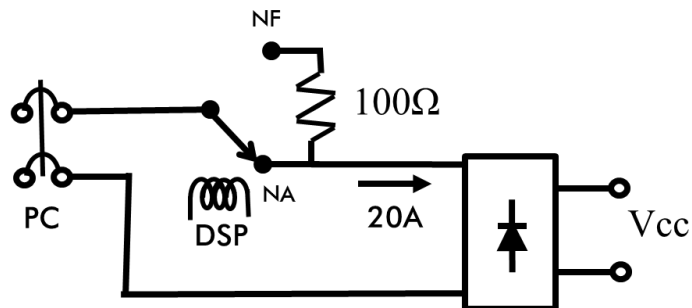


Fonte: Própria.

A topologia de conversor *back-to-back* é utilizada quando é necessário o fluxo de potência nos dois sentidos, da rede para a máquina e da máquina para a rede. Como na nossa aplicação a máquina de indução é utilizada apenas como motor, o conversor 2 (CONV2) foi utilizado apenas como retificador trifásico, recebendo as tensões trifásicas da rede e entregando tensão contínua no barramento CC. Na seção 2.8, que trata de controladores de corrente, foi apresentado o esquemático da parte de potência desse conversor. Associado a cada conversor existe uma unidade retificadora monofásica chamada pré-carga (Figura 6.2), é utilizada para cargas menores ou para carregar o banco de capacitores do barramento CC. Na pré-carga existe um resistor de  $100\Omega$  para limitar a corrente enquanto os capacitores são carregados, como o resistor não é de alta potência e representa uma perda, quando houver fluxo de corrente

relacionado ao acionamento do inversor, um relé deve ser acionado via DSP para fazer um *bypass* do resistor.

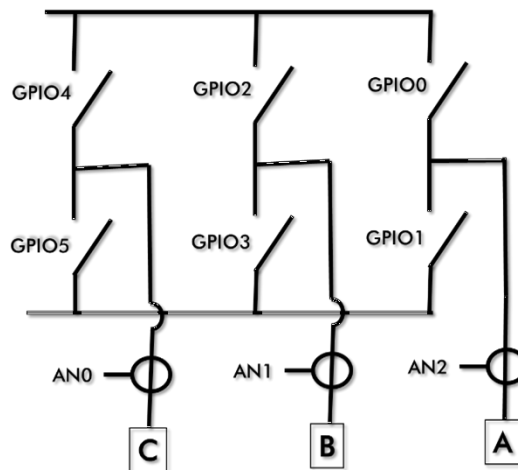
Figura 6.2 – Pré-carga: retificador monofásico ligado ao barramento CC



Fonte: Própria.

Na Figura 6.3, temos o esquemático do conversor 1 (CONV1), indicando qual pino do DSP está relacionada a cada chave nos braços do inversor. Em cada fase existe um transdutor de corrente por efeito *Hall* da LEM, os canais analógicos do DSP pertinentes a esses sensores também estão apresentados.

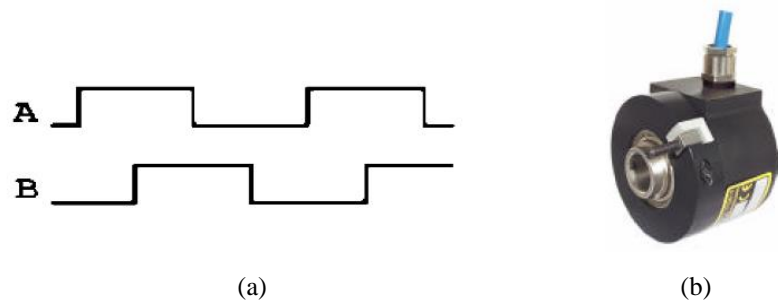
Figura 6.3 – Esquemático do conversor 1



Fonte: Própria.

Para medição da velocidade do rotor, utilizou-se um *encoder* incremental. Este tipo de equipamento é instalado no eixo traseiro da máquina e preso por uma flange (Figura 3.1). No modelo utilizado, *encoder* bidirecional, são gerados dois sinais, o segundo defasado em  $90^\circ$  do primeiro quando girado no sentido anti-horário e o primeiro defasado em  $90^\circ$  do segundo quando girado no sentido horário, desta forma, é possível encontrar o sentido de rotação. São produzidos 2500 pulsos a cada volta do eixo do *encoder*, o que significa que a velocidade pode ser medida com precisão até em baixas rotações, como no caso do controle de posição.

Figura 6.4 – *Encoder* (b) e os dois sinais gerados por ele (a)



Fonte: Datasheet: Incremental Encoder 59C.

## 6.1 Programação do DSP

Como já mencionado, é utilizado um processador digital de sinais da *Texas Instruments*, o F28335 (Figura 6.5). Este DSP opera com (User's Guide: TMS320C28x CPU and Instruction Set Reference Guide):

- Um ciclo de *clock* de 150 MHz;
- Ponto flutuante de 32 bits em *hardware*;
- Arquitetura de barramento Harvard: um barramento para a memória de dados e outro para a memória de instruções;
- *Pipelining*: as instruções não possuem tempo de execução determinístico, pois as micro-operações das instruções são realizadas uma atrás da outra, independente de a instrução anterior tiver sido concluída ou não;
- RISC (Computador com Conjunto Reduzido de Instruções);
- Memória *flash* 16x256K e memória RAM 16x34K integrado;
- Oscilador integrado;
- 58 Interrupções de periféricos;
- 88 Pinos de entrada e saída com multiplexação programável e individual GPIO;
- 18 Saídas PWM (Temporizadores de 16 bits);
- 6 Entradas com *Event Capture* (Temporizadores de 32 bits);
- 2 Interfaces para *Quadrature Encoder* (Temporizadores de 32 bits);
- 3 Temporizadores de 32 bits;
- Porta serial;
- 16 Canais para conversão Analógico-Digital de 12 bits (taxa de conversão de 80ns);
- Depuração em tempo real via *hardware*.

Figura 6.5 – DSP TI F28335



Fonte: Própria.

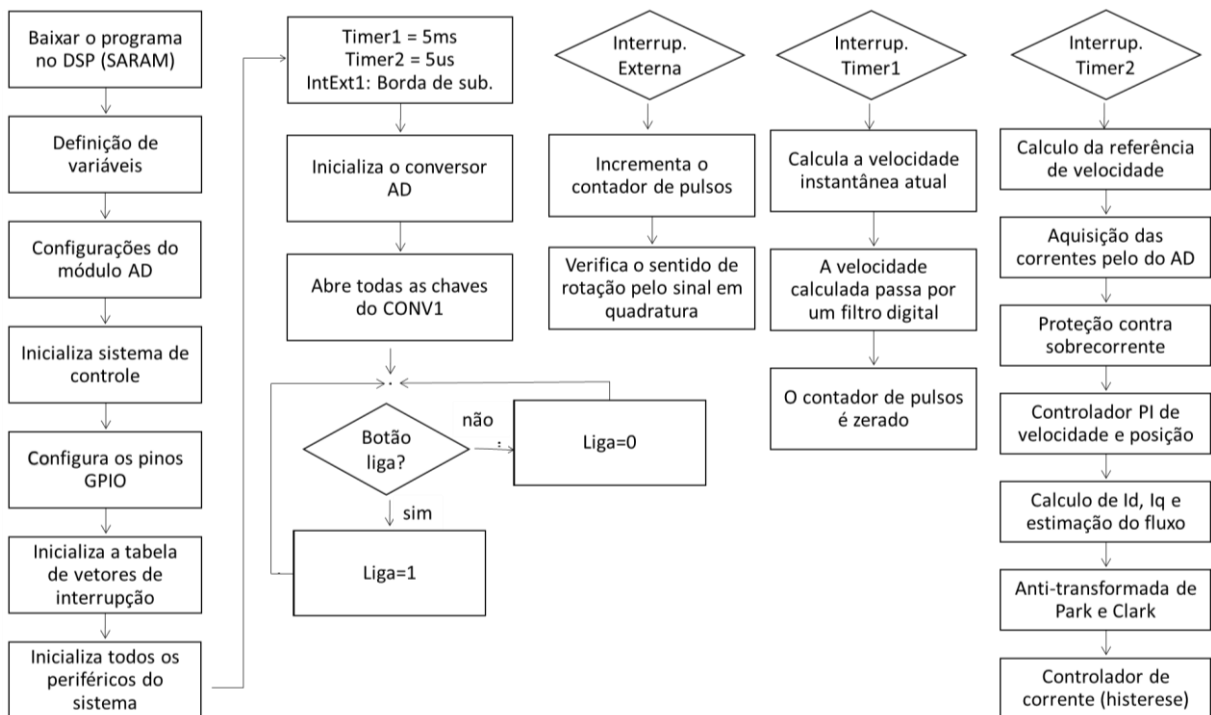
A *Texas Instruments* (TI) disponibiliza um pacote integrado para propósitos de controle industrial chamada *controlSUITE*<sup>TM</sup>. Este pacote consiste de arquivos de cabeçalho, arquivos fonte, bibliotecas, arquivos *linker command* (são arquivos que definem se o programa será gravado na memória *flash* ou na RAM) para desenvolvimento de um programa executável. Além do mais, também fornece vários códigos de exemplos, que permitem que um novo usuário possa aprender rapidamente. O *Code Composer Studio*<sup>TM</sup> (CCS) é o ambiente de desenvolvimento ou IDE (*Integrated Development Enviroment*) para os microcontroladores da TI.

Um breve controle escalar ou V/f de malha aberta foi também implementado nos primeiros estágios da programação para testar a funcionalidade de cada periférico do DSP. Nesta implementação foram utilizados os módulos de ePWM, IQMath, ADC, Timer0, Timer1 e XINT1, onde foi possível implementar o PWM senoidal, a medição de velocidade pelo *encoder* e a calibragem dos canais analógicos para medição correta das correntes. Para uma descrição mais detalhada do *hardware* usado no DSP F28335, seus periféricos e módulos, sugiro a leitura de Cheng (2011) e Data Manual: Digital Signal Controllers (2012).

O sistema de controle vetorial completo do motor de indução é basicamente desenvolvido de acordo com o diagrama de blocos mostrado na Figura 5.1. Na Figura 6.6 é apresentado o fluxograma detalhado do programa implementado para efetuar o controle vetorial. Como pode ser observado, o programa inicia com a declaração de variáveis, inicialização e configuração do *hardware* do DSP, o inversor inicia com todas as suas chaves abertas. Sequencialmente, o programa entra no *loop* principal, onde vigia os botões de ligar e desligar e espera pelas interrupções. As interrupções são eventos que são disparados via *hardware* (mais

comumente) e levam a execução do programa para outro bloco chamado ISR (*Interrupt Service Routine*), ou vetor de interrupção, faz as operações necessárias e retorna o local onde estava anteriormente. Como visto no fluxograma, o programa possui três interrupções, por ordem de maior prioridade, a primeira ocorre quando há uma borda subida no sinal proveniente do *encoder*, a segunda ocorre quando o contador do *timer1* alcança 5 ms (milissegundos) e a terceira quando o contador do *timer2* alcança 5 us (microssegundos). Todas as interrupções vêm do *loop* principal e para lá retornam. Como no controle vetorial a aquisição e o controle de corrente ocorrem na interrupção do *timer2*, a tempo de amostragem do programa é de 5 us, implicando na frequência amostral de 200 kHz.

Figura 6.6 – Fluxograma do programa



Fonte: Própria.

Para aumento na performance do programa, utilizou-se a biblioteca matemática FPUfastRTS que, além de trabalhar com *Floating Point* (números reais), calcula o seno e o cosseno pela interpolação entre os 200 valores preestabelecidos em uma tabela. Sendo que a biblioteca matemática padrão é a IQmath que trabalha com *Fixed Point* (números inteiros) e calcula funções trigonométricas através de expansão por séries de Taylor. Utilizando a FPUfastRTS foi possível mais do que dobrar a frequência de amostragem e chegar até 200 kHz.

## 6.2 Resultados experimentais

Nesta seção será mostrado o comportamento do motor de indução trifásico de 1 cv, 220V, gaiola de esquilo, ligado em delta e acoplado a um freio de Foucault, aplicando controle vetorial por campo orientado mediante os ciclos de trabalhos conforme simulado no Capítulo 5. Os resultados experimentais para o controle de velocidade e de posição do eixo do motor são apresentados a seguir, sendo que primeiramente foi implementado o controle escalar (V/f) em malha aberta para testes iniciais do *hardware* do DSP e do conversor. Os resultados do controle V/f foram omitidos mas o código fonte do DSP para este acionamento pode ser visto no Apêndice C.

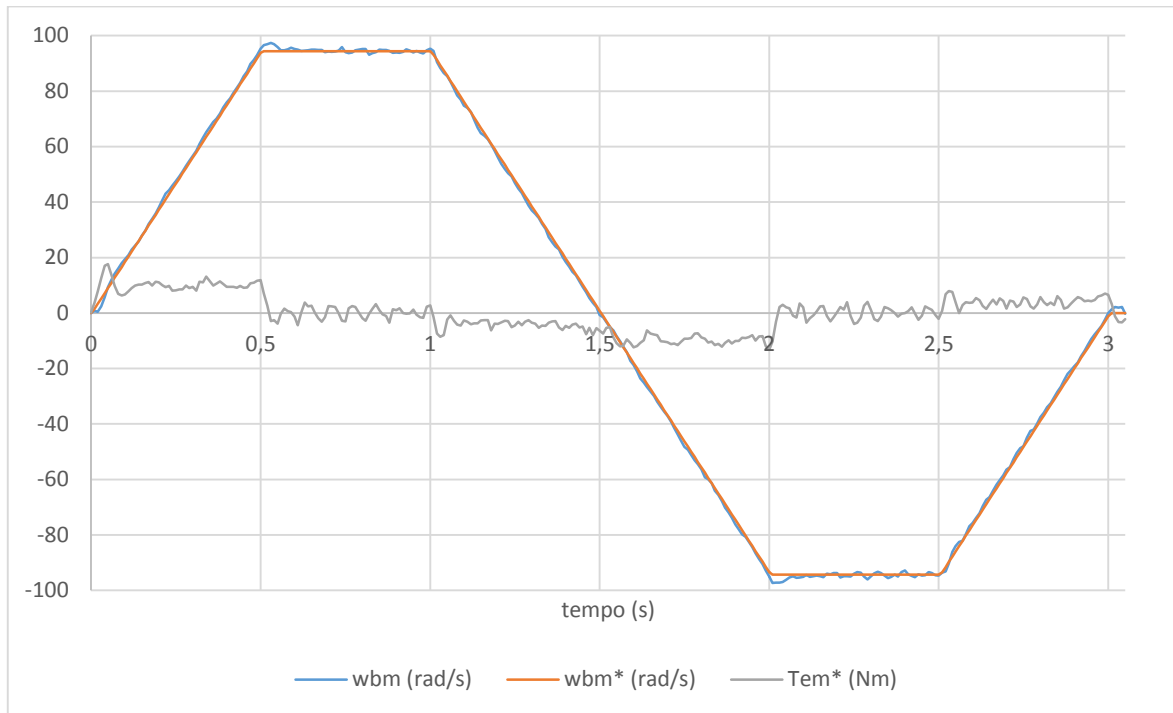
### 6.2.1 Teste de controle de velocidade a vazio

O resultado experimental para o controle de velocidade do eixo do motor de indução utilizando controle vetorial segundo o ciclo de trabalho do teste a vazio com inversão de sentido (simulado na seção 5.1) é apresentado nas Figuras 6.7, 6.8 e 6.9. Os resultados das duas primeiras destas figuras foram obtidos a partir da exportação dos pontos no CCS no modo *real time debug* numa taxa amostral de 200 Hz, o resultado da Fig. 6.9 foi obtido com três pontas de prova de corrente em um osciloscópio. O código fonte do DSP para este acionamento pode ser visto no Apêndice A.

Na Figura 6.7 vemos que a velocidade do rotor seguiu bem a referência nos dois sentidos. Os maiores erros ocorreram quando havia variação da aceleração, chegando a uma diferença de 3,016 rad/s (3,20% no instante 0,53 segundos) e o máximo tempo de acomodação pelo critério de 2% foi de 0,05 segundos. A média de erro durante todo o acionamento foi em torno de 0,83%, o que é um excelente resultado. Vemos também o conjugado eletromagnético de referência, saída do controlador PI de velocidade, que é máximo nos momentos de aceleração positiva e possui valor médio nulo nos momentos de velocidade constante.



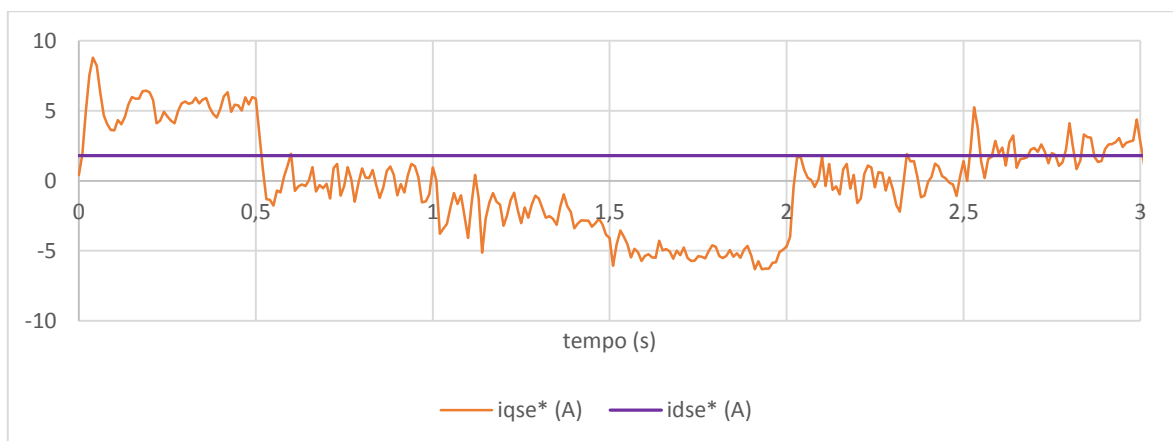
Figura 6.7 – Velocidade do rotor ( $w_{bm}$ ), velocidade de referência ( $w_{bm}^*$ ) e conjugado eletromagnético de referência ( $Tem^*$ ) no teste a vazio



Fonte: Própria.

Pode-se ver como se comportaram as correntes de referência de eixo direto e em quadratura durante o acionamento na Figura 6.8. São de referência pois são provenientes do bloco de controle vetorial e são valores que devem ser impostos ao motor, e não que foram medidos do motor. Vemos que a corrente de eixo direto ( $i_{ds}^e$ ) responsável pelo fluxo permanece constante enquanto que a corrente em quadratura assume valores para compensar o erro de velocidade com conjugado eletromagnético.

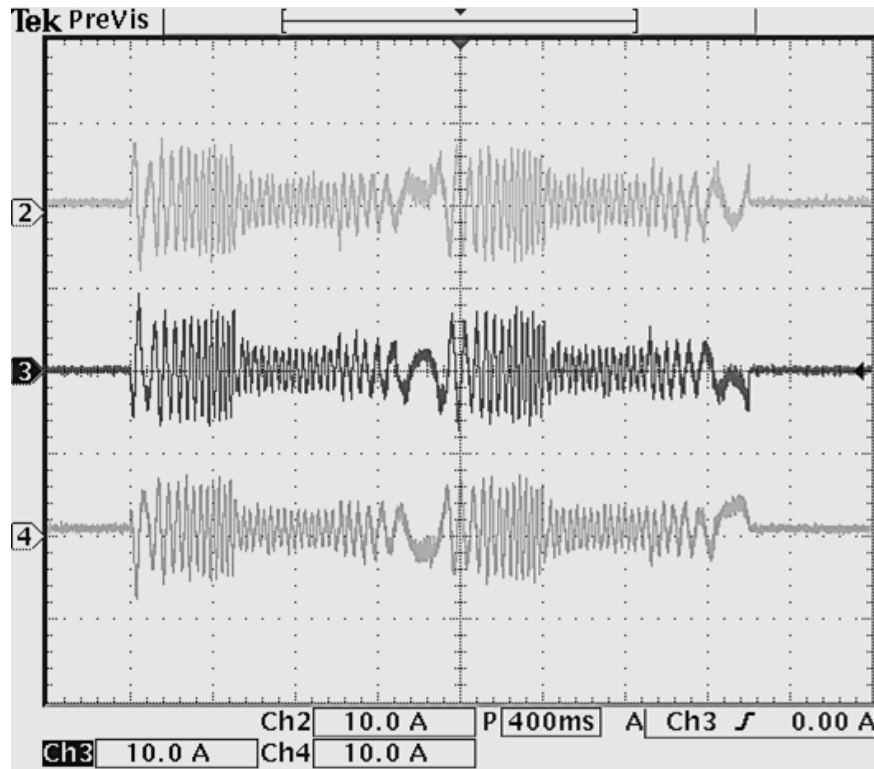
Figura 6.8 – Correntes de referência  $i_{ds}^{e*}$  ( $i_{ds}^*$ ) e  $i_{qs}^{e*}$  ( $i_{qs}^*$ ) no teste a vazio



Fonte: Própria.

As três correntes de linha do estator medidas são mostradas na Figura 6.9, pode ser observado amplitude e a frequência das correntes.

Figura 6.9 – Correntes trifásicas de linha no teste a vazio



Fonte: Própria.

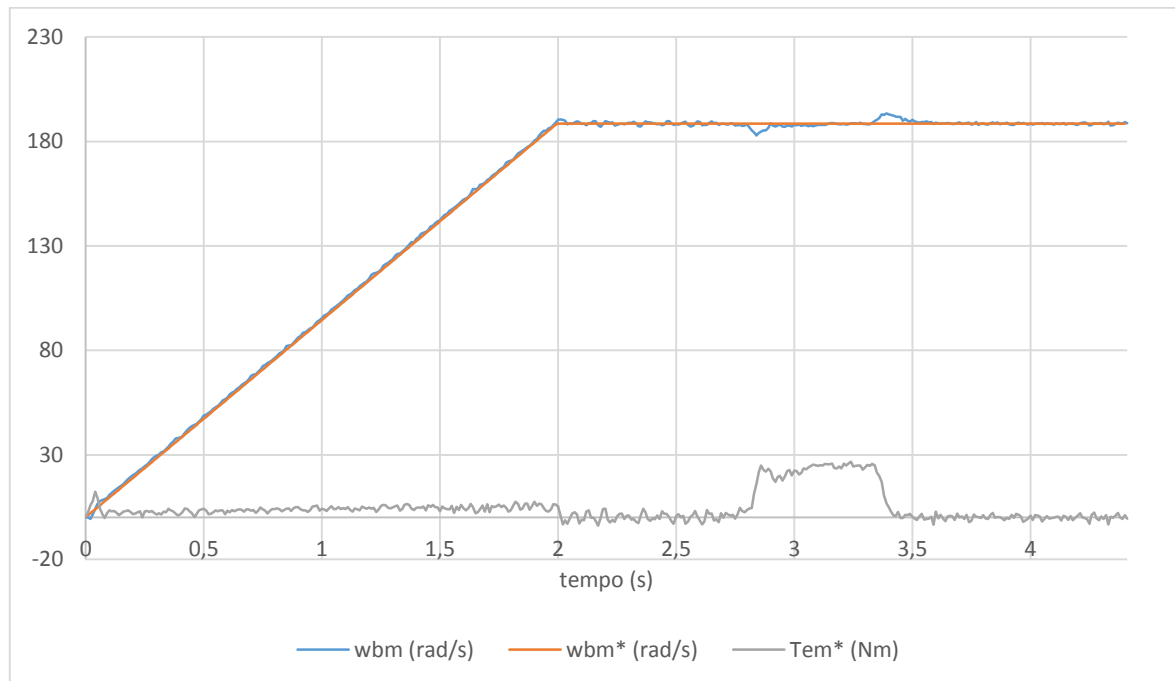
### 6.2.2 Teste de controle de velocidade com carga

O resultado experimental para o controle de velocidade do eixo do motor de indução utilizando controle vetorial segundo o ciclo de trabalho do teste com aplicação de carga (simulado na seção 5.2) é apresentado nas Figuras 6.10 e 6.11. Os resultados da primeira destas figuras foram obtidos a partir da exportação dos pontos no CCS no modo *real time debug* numa taxa amostral de 200 Hz, o resultado da Fig. 6.11 foi obtido com três pontas de prova de corrente em um osciloscópio. Para aplicar carga mecânica no eixo do motor foi utilizado um freio de Foucault de 7 Nm ajustado em 57% para atingir 4 Nm, o conjugado nominal do motor utilizado. O código fonte do DSP para este acionamento, assim como o anterior, pode ser visto no Apêndice A, a diferença está apenas na definição da velocidade de referência.

Na Figura 6.10 vemos que a velocidade do rotor seguiu bem a referência durante a rampa de partida, a entrada e a saída de carga. Os maiores erros ocorreram quando havia entrada e saída de carga, chegando a uma diferença de 4,90 rad/s (2,60%) no instante da retirada da

carga, o tempo de acomodação considerando o critério de 2% foi de 0,07 s (70 milésimos de segundo). A média de erro durante todo o acionamento foi de 0,92%. Vemos também o conjugado eletromagnético de referência, resultado do controle vetorial, que é máximo nos momentos em que a carga está aplicada, possui valor médio nulo nos momentos de velocidade de referência constante e sem carga. Durante a partida existe conjugado eletromagnético positivo devido a aceleração.

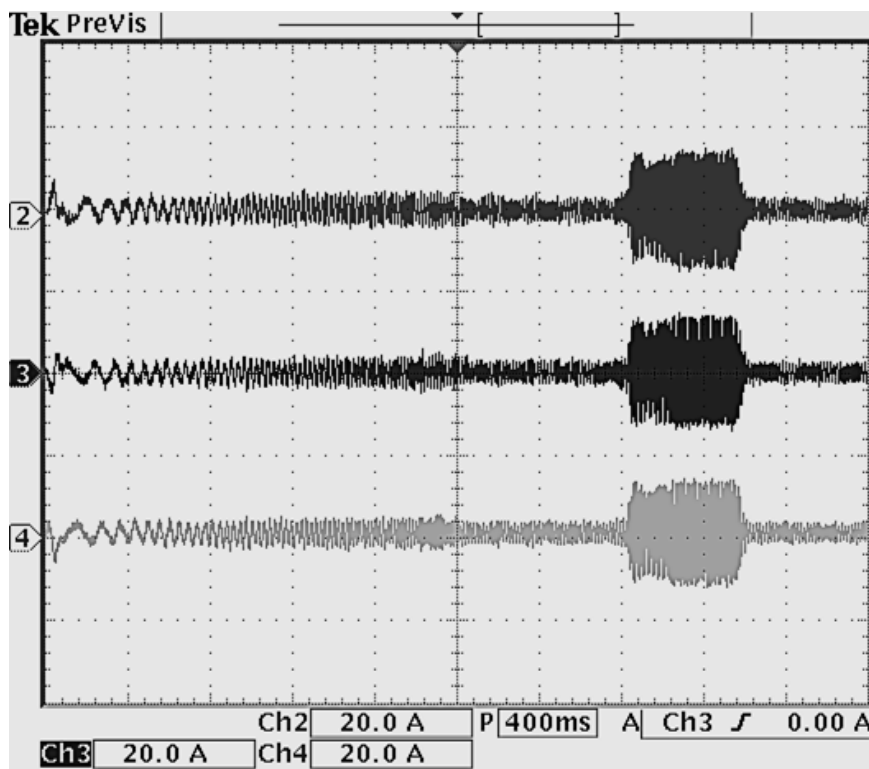
Figura 6.10 – Velocidade do rotor (w<sub>bm</sub>), velocidade de referência (w<sub>bm</sub><sup>\*</sup>) e conjugado eletromagnético de referência (T<sub>em</sub><sup>\*</sup>) no teste com carga



Fonte: Própria.

As três correntes de linha do estator medidas são mostradas na Figura 6.11, pode ser observado amplitude e a frequência das correntes durante a partida e variações de carga.

Figura 6.11 – Correntes trifásicas de linha no teste com carga



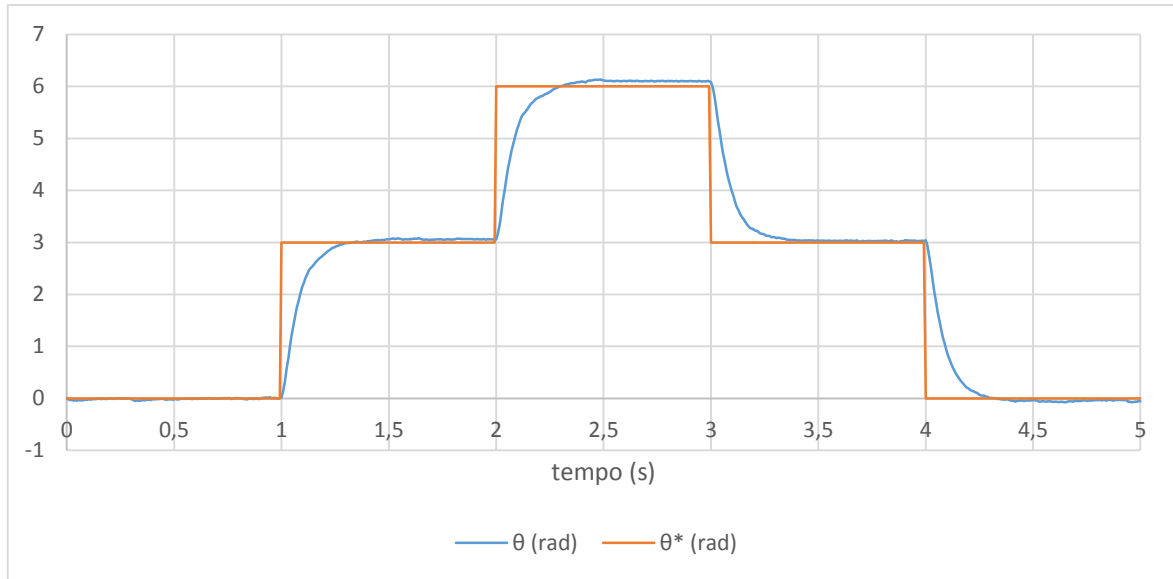
Fonte: Própria.

### 6.2.3 Teste de controle de posição

O resultado experimental para o controle da posição do eixo do motor de indução utilizando controle vetorial segundo o ciclo de trabalho do teste simulado na seção 5.3 é apresentado nas Figuras 6.12, 6.13 e 6.14. Os resultados das duas primeiras destas figuras foram obtidos a partir da exportação dos pontos no CCS no modo *real time debug* numa taxa amostral de 250 Hz. A posição é obtida a partir da integração numérica da velocidade. O código fonte do DSP para este acionamento pode ser visto no Apêndice B.

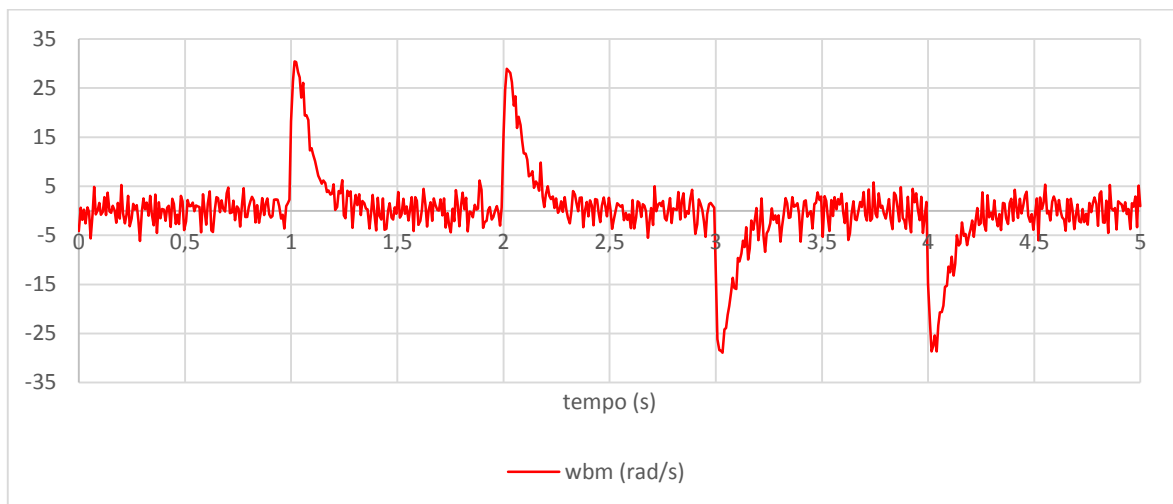
Na Figura 6.12 vemos que a posição do rotor seguiu bem a referência com degraus positivos e negativos. O erro de estado permanente foi de 1,69% (0,100 rad) e o tempo de acomodação para o critério de 2% é de 0,272 segundos. O comportamento da velocidade durante o acionamento com controle de posição pode ser visto na Figura 6.13.

Figura 6.12 – Posição do eixo do motor ( $\theta$ ) e posição de referência ( $\theta^*$ )



Fonte: Própria.

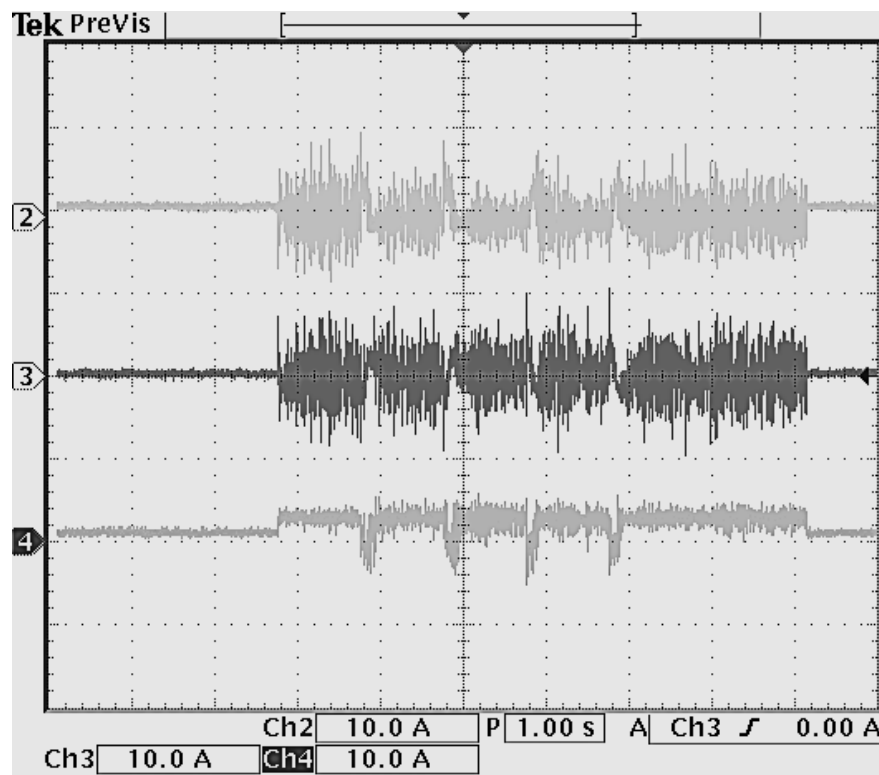
Figura 6.13 – Velocidade do rotor ( $\omega_{bm}$ ) durante o controle de posição



Fonte: Própria.

As três correntes de linha do estator medidas são mostradas na Figura 6.14, obtido com três pontas de prova de corrente em um osciloscópio.

Figura 6.14 – Correntes trifásicas de linha no teste de controle de posição



Fonte: Própria.

### 6.3 Considerações finais

Neste capítulo apresentou-se o desenvolvimento experimentalmente para aplicação do controle vetorial utilizando o conversor descrito e o DSP TI F28335. Primeiramente foi controlado a velocidade com o motor a vazio, utilizando referências de velocidade em rampa e com inversão de sentido, em seguida foi aplicado carga mecânica ao eixo do motor e observado seu comportando mediante ação do controle, e por último foi controlado a posição do eixo do motor. Comparando os resultados experimentais com os obtidos nas simulações computacionais, vemos que o motor se comportou de forma bem semelhante ao previsto e teve erros menores que 2%, validando assim as teorias e modelagens aplicadas, bem como os parâmetros da máquina obtidos e os ganhos dos controladores dimensionados.

## 7 CONCLUSÕES

### 7.1 Resultados do presente trabalho

O principal objetivo deste trabalho foi o desenvolvimento de uma estratégia de controle do fluxo e do conjugado do motor de indução trifásico através de comando de correntes, resultando no posicionamento do eixo do motor ou no controle de velocidade. As aplicações do controle de posição e velocidade são as mais variadas possíveis na indústria, em automóveis, na robótica e em outros setores.

O lance de ser avaliado a utilização deste tipo de motor que exige análise tão mais complexa que outros tipos de motores como o de corrente contínua e o síncrono de ímãs permanentes, reside no fato de que seu custo de aquisição e de manutenção é bem inferior, principalmente quando se considera o motor de indução de gaiola de esquilo. Os parâmetros elétricos e mecânicos obtidos através de ensaios e os ganhos dos controladores projetados por alocação de polos se mostraram bastante válidos mediante a análise dos resultados experimentais e computacionais. Quando comparado os resultados da simulação e os experimentais, alguns problemas aparecem. O erro de regime permanente, experimental, no controle de velocidade oscilou em torno de  $\pm 0,48\%$  e no controle de posição chegou a  $1,69\%$  enquanto que nas simulações esse tipo de erro não se mostra. Isto é provavelmente causado por parâmetros incorretos da máquina e pelo ruído produzido pelo chaveamento em alta frequência dos IGBT no conversor, influenciando em medições erradas nos canais analógicos (valores das correntes) e nos sinais do *encoder*. Certamente, algumas pequenas diferenças entre experimentos e simulações são toleradas, com esses fatores práticos incluídos.

### 7.2 Trabalhos futuros

Ainda há bastante potencial para melhoria de performance do presente sistema. Uma vez que a frequência de chaveamento, quando utilizado o controle de corrente por histerese, é variável e superior quando comparado com o controlador PI síncrono (SPWM), este último pode ser implementado, juntamente com aplicação de filtros contra ruídos nos canais analógicos, para melhoria de performance e diminuição do tempo de amostragem do controle, possibilitando assim o controle de dois ou mais motores simultaneamente utilizando o mesmo DSP. Ou ainda, *Space Vector PWM* (SVPWM) pode ser introduzido neste controle como substituto da histerese e do SPWM, possibilitando a redução de distorção harmônica e um uso

mais eficiente da fonte de alimentação. Pretende-se aplicar a lei de controle desenvolvida para motores de indução em veículos elétricos do tipo lagarta com estratégias de frenagem regenerativa.



## REFERÊNCIAS

FITZGERALD, A. E., **Máquinas Elétricas com Introdução à Eletrônica de Potência**. 6º ed. Bookman, 2006.

ONG, Chee-mun. **Dynamic Simulation of Electric Machinery Using Matlab®/Simulink**. Prentice Hall, 1997.

BARBI, I. **Teoria Fundamental do Motor de Indução**. Ed. UFSC, 1986.

REGINATTO, Romeu. **Controle por Campo Orientado do Motor de Indução com Adaptação de Parâmetros via MRAC**. Dissertação de mestrado. Universidade Federal de Santa Catarina, Florianópolis, 1993.

DINIZ, Eber de Castro. **Utilização de Motores de Indução Trifásicos em Manipulador Articulado com 2 Graus de Liberdade Considerando Estratégias de Controle Vetorial de Campo Orientado Indireto e Modo de Controle Deslizante**. Tese de Doutorado. Departamento de Engenharia Elétrica. Universidade Federal do Ceará, Fortaleza, 2006.

“**MATLAB for Windows User’s Guide**”, The Math Works Inc., 1991.

KAZMIERKOWSKI, M. P.; MALESANI, L., **Current Control Techniques for Three-Phase Voltage-Source PWM Converters: A Survey**. IEEE Trans. Power Electron., vol 45, pp. 691-703, Outubro 1998.

DORF, R. C.; BISHOP, R. H.. **Sistemas de Controle Modernos**. 8ª ed. Rio de Janeiro: LTC, 2001.

Elcis. Datasheet: **Incremental Encoder 59C**.

CHENG, Qian; YUAN, Lei. **Vector Control of an Induction Motor Based on DSP**. Dissertação de Mestrado. Department of Science and Enviroment. Chalmers University of Technology, Goteborg, Suécia, 2011.

Texas Instruments. Data Manual: **Digital Signal Controllers (DSCs)**. Literature Number: SPRS439M. 2012.

Texas Instruments. User’s Guide: **TMS320C28x CPU and Instruction Set Reference Guide**. Literature Number: SPRU430F. 2015.

Texas Instruments. User’s Guide: **C2833x/C2823x C/C++ Header Files and Peripheral Examples Quick Start**. Literature Number: SPRCA73. 2010.

Texas Instruments. User’s Guide: **Programing TMS320x28xx and 28xxx Peripherals in C/C++**. Literature Number: SPRAA85. 2009.

Texas Instruments. Reference Guide: **TMS321x281x DSP System Control and Interrupts**. Literature Number: SPRU078B. 2004.

Ogata, K. **Engenharia de Controle Moderno**. São Paulo: Prentice Hall, 2005.

SOARES, Marcelo Nesci. **Projeto de Controladores PI para Acionamento Vetorial de Máquina de Indução**. Monografia de Curso de Engenharia Elétrica. Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

DE ARAÚJO, Amanda Guerra. **Controle Vetorial de um Motor de Indução Trifásico Aplicado em Sistemas de Posicionamento**. Dissertação de mestrado. Universidade Federal da Paraíba, João Pessoa, 2011.

UNIVERSIDADE FEDERAL DO CEARÁ. Biblioteca Universitária. **Guia de normalização de trabalhos acadêmicos da Universidade Federal do Ceará**. Fortaleza, 2013.

## APÊNDICE A – CÓDIGO FONTE DO DSP PARA CONTROLE DE VELOCIDADE DO MOTOR DE INDUÇÃO POR ORIENTAÇÃO DE CAMPO

```

#include "DSP28x_Project.h" // Device Headerfile and Examples Include File
#include "Example_freqcal.h" // Example specific include file
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "C28x_FPU_FastRTS.h"

//Definição dos botões
#define P22 (GpioDataRegs.GPBDAT.bit.GPIO47)
#define P23 (GpioDataRegs.GPCDAT.bit.GPIO80)
#define P24 (GpioDataRegs.GPCDAT.bit.GPIO81)
#define P25 (GpioDataRegs.GPCDAT.bit.GPIO82)
#define P26 (GpioDataRegs.GPCDAT.bit.GPIO83)

//
#define PIH2          1.570796327
#define PIH           3.141592654

//Vetores das interrupções
interrupt void cpu_timer2_isr(void);
interrupt void cpu_timer1_isr(void);
interrupt void xint1_isr(void);

//Parametros do motor
float Rs=5.35, Rr=9.443, Ls=0.389, Lr=0.377, Lm=0.360, J=0.013;
float wref=1800*PIH/30, Tem, P=4;
float lambdadr_e=0.648; //469.002/(2*PIH*60)*0.635; *1.4
float wrm=0, thetar=0; //frequencia eletrica de base
float ids_e, iqs_e, iqs_s, ids_s;
float Iqse, Idse, Iqss, Idss;
float sin_rho, cos_rho, rho=0;
float Tn=4.14;
float Kw2, Kiq;

float k=1, t1=0, t0=0;
Uint32 cont=0, tempo=0, tv=0;
//Declaração de variáveis
int ligal=0, rot=1, enctemp=1;
float velocidade[3]={0,0,0}, Ia, Ib, Ic;
float AdcResult0[3]={0,0,0}, AdcResult1[3]={0,0,0}, AdcResult2[3]={0,0,0};
float margem=0.005, w=0, w2=0;
float iaref=0, ibref=0, icref=0;
float iasref=0, ibsref=0, icsref=0;
float Iarefmax, Iarefmin, Ibrefmax, Ibrefmin, Icrefmax, Icrefmin;
//variaveis do pi
float u[2]={0,0}, ui[2]={0,0}, up, erro[2]={0,0};
float Kp=2.921, Ki=2.075e-4;

// ----- BUFFERS PARA GRAFICOS ----- //
float plot1[470];
float plot2[470];
float plot3[470];
long int plot4[470];
unsigned long int contaPlot = 0, samplerPlot = 0;

```

```

//Configurações do módulo AD
EALLOW;
// Default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3) = 25.0
MHz
EDIS;
#define ADC_CKPS 0x1 // ADC module clock = HSPCLK/2*ADC_CKPS = 25.0MHz/(1*2)
= 12.5MHz
#define ADC_SHCLK 0xf // S/H width in ADC module periods =
16 ADC clocks
#define AVG 1000 // Average sample limit
#define ZOFFSET 0x00 // Average Zero offset
#define BUF_SIZE 2048 // Sample buffer size

void main(void)
{

// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the DSP2833x_SysCtrl.c file.
InitSysCtrl();

// Specific clock setting for this project:
EALLOW;
SysCtrlRegs.HISPCP.all = ADC_MODCLK; // HSPCLK = SYSCLKOUT/ADC_MODCLK
EDIS;

// Step 2. Initialize GPIO:
// This example function is found in the DSP2833x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.

// configGPIO();
InitGpio();
EALLOW;
GpioCtrlRegs.GPAPUD.all = 0xFFFF; // Pullup's disabled GPIO00-GPIO31
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO4 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO8 = 1; //define como saída
GpioCtrlRegs.GPBDIR.bit.GPIO32 = 1; //define como saída COOLER CONV1
GpioCtrlRegs.GPBDIR.bit.GPIO49 = 1; //define como saída RELÉ DO RESISTOR DA
PRE-CARGA
GpioCtrlRegs.GPBDIR.bit.GPIO50 = 1; //conv7
//GPIO DO CONV2 (PRE-CARGA)
GpioCtrlRegs.GPBDIR.bit.GPIO56 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO57 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO53 = 1;
EDIS;

// GPIO20 is XINT1, GPIO21 is XINT2
EALLOW;
GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 20; // Xint1 is GPIO20
EDIS;

// Only init the GPIO for EQep1 and EPwm1 in this case

```

```

// This function is found in DSP2833x_EQep.c

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

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the DSP2833x_PieCtrl.c file.
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 in this example. This is useful for debug purposes.
// The shell ISR routines are found in DSP2833x_DefaultIsr.c.
// This function is found in DSP2833x_PieVect.c.
InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
EALLOW; // This is needed to write to EALLOW protected registers
// PieVectTable.EPwm1_INT= &prdTick;
PieVectTable.TINT2 = &cpu_timer2_isr;
PieVectTable.XINT13 = &cpu_timer1_isr;
PieVectTable.XINT1 = &xint1_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize all the Device Peripherals:
// Example specific ePWM setup. This function is found
// in Example_EPwmSetup.c
// EPwmSetup();
InitAdc(); // For this example, init the ADC
// configADC();
////////////////////////////////////
// Specific ADC setup for this example:
AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;
AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS;
AdcRegs.ADCTRL1.bit.SEQ_CASC = 1; // 1 Cascaded mode
AdcRegs.ADCMAXCONV.all = 7;
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // ADCINA0 1º Conv. SEQ1 /* a conv00
eh o pino zero */
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 1; // ADCINA1 2º Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 2; // ADCINA2 3º Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 3; // ADCINA3 4º Conv. SEQ1
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 4; // ADCINA4 5º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 5; // ADCINA5 6º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 6; // ADCINA6 7º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 7; // ADCINA7 8º Conv. SEQ2
AdcRegs.ADCTRL1.bit.CONT_RUN = 1; // Setup continuous run
////////////////////////////////////

InitCpuTimers(); // For this example, only initialize the Cpu Timers

```

```

ConfigCpuTimer(&CpuTimer1, 150, 5000); //Interrompe a cada 5 ms
ConfigCpuTimer(&CpuTimer2, 150, 5); // 10 us

CpuTimer2Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit =
0
CpuTimer1Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit =
0

// Step 5. User specific code, enable interrupts:
// Enable CPU INT1 which is connected to CPU-Timer 0:
IER |= M_INT1; // Enable CPU Interrupt 1, adc, timer0, intext1
// IER |= M_INT3;
IER |= M_INT13; //timer1
IER |= M_INT14; //timer2

// Enable TINT0 in the PIE: Group 3 interrupt 1
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE block
PieCtrlRegs.PIEIER1.bit.INTx4 = 1; // Enable PIE Gropu 1 INT4 int ext 1

PieCtrlRegs.PIEIER1.bit.INTx7 = 1; //timer2 epwm
PieCtrlRegs.PIEIER3.bit.INTx1 = 1; //timer1

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

// Configure XINT1
XIntruptRegs.XINT1CR.bit.POLARITY = 1; // Falling(0) edge interrupt

// Enable XINT1 and XINT2
XIntruptRegs.XINT1CR.bit.ENABLE = 1; // Enable Xint1

// Start SEQ1 adc
AdcRegs.ADCTRL2.all = 0x2000;

GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;

//constante de iq
Kiq=(4/(3*P))*Lr/(Lm*lambda_r_e);
//definição do valor constante de id
ids_e=1.8;
//constante de w2
Kw2=(Rr/Lr)*(1/ids_e);

for(;;)
{
    // Conversor 01
    if (P22)
        liga1 = 1; // Liga
    if (P23)
        liga1 = 0; // Desliga
    // Conversor 02
    if (P25)

```

```

        GpioDataRegs.GPASET.bit.GPIO15 = 1;           // Liga
    }
    if (P26)
        GpioDataRegs.GPACLEAR.bit.GPIO15 = 1;       // Desliga
}

interrupt void cpu_timer2_isr(void) {
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
    GpioDataRegs.GPASET.bit.GPIO8 = 1;
    t0=(CpuTimer2.PeriodInUSec)*1e-6; //time between interruptions
    tempo=tempo+t0;

    //Calcula wrm e thetar
    // thetar=thetar+wrm*t0; //angulo mecanico
    // if (thetar>=2*PIH) {thetar=thetar-2*PIH;}
    // if (thetar<=-2*PIH) {thetar=thetar+2*PIH;}

    //proteção contra sobre velocidades
    if (wrm*rot>200) liga1=0;

    if (liga1) {

        GpioDataRegs.GPBSET.bit.GPIO56 = 1; //PRE-CARGA DO CONV2
        GpioDataRegs.GPBSET.bit.GPIO57 = 1;
        GpioDataRegs.GPBSET.bit.GPIO53 = 1;

        GpioDataRegs.GPASET.bit.GPIO14 = 1; //led
        GpioDataRegs.GPBSET.bit.GPIO32 = 1; //cooler
        GpioDataRegs.GPBSET.bit.GPIO49 = 1; //R pre-carga
        GpioDataRegs.GPBSET.bit.GPIO50 = 1; //conv7

        tempo++;

    }

    // WREF - ENSAIO A VAZIO
    /*
        if (tempo<=100000) w=w+(942.5e-6);
        else if (tempo<=200000) w=w;
        else if (tempo<=400000) w=w-(942.5e-6);
        else if (tempo<=500000) w=w;
        else if (tempo<=600000) w=w+(942.5e-6);
        else {w=0; if((wrm*rot)<=0.005) liga1=0;}
    */
    // WREF - ENSAIO COM CARGA
    if (w<wref) {w=w+(471.25e-6);} //rampa de velocidade (partida em 2s)
    (377/P)/(5/t0)
    else w=wref;

    // Ia Rotor
    // AdcResult0[0]=AdcResult0[2];
    // AdcResult0[1]=(AdcRegs.ADCRESULT2 >> 4);
    // AdcResult0[2]=0.2*AdcResult0[1] + 0.8*AdcResult0[0];
    // Ib Rotor
    // AdcResult1[0]=AdcResult1[2];
    // AdcResult1[1]=(AdcRegs.ADCRESULT1 >> 4);
    // AdcResult1[2]=0.2*AdcResult1[1] + 0.8*AdcResult1[0];
    // Ic Rotor
    // AdcResult2[0]=AdcResult2[2];

```

```

    AdcResult2[1]=(AdcRegs.ADCRESULT0 >> 4);
//    AdcResult2[2]=0.2*AdcResult2[1] + 0.8*AdcResult2[0];

    Ia=(AdcResult0[1]*0.02402-49.51);//((AdcResult0[1]/4095.0*3.0-
1.495)*0.49/0.023-0.33)*1.53846;
    Ib=(AdcResult1[1]*0.02511-51.70);//((AdcResult1[1]/4095.0*3.0-
1.495)*0.49/0.022-0.31);
    Ic=(AdcResult2[1]*0.02877-58.94);//((AdcResult2[1]/4095.0*3.0-
1.495)*0.97/0.038-0.15);

    //PROTEÇÃO contra sobrecorrente
    if (Ia>50) liga1=0;
    if (Ib>50) liga1=0;
    if (Ic>50) liga1=0;

//Controlador PI de velocidade
    u[0]=u[1];
    if (u[0]>10*Tn) u[0]=15; //Saturação
    if (u[0]<-10*Tn) u[0]=-15;
    erro[0]=erro[1];
    erro[1]=w-wrm;
    u[1]=u[0]+Kp*(erro[1]-erro[0])+Ki*erro[1];

    Tem=u[1];

    //Controle por campo orientado
    iqs_e=Kiq*Tem;
//    ids_e=(lambdadr_e)/(Lm);
//    iqs_e=Tem;
    w2=Kw2*iqs_e;
    rho=rho+(w2+wrm*P*0.5)*t0;
    if (rho>=2*PIH) rho=rho-2*PIH;
    if (rho<=-2*PIH) rho=rho+2*PIH;
    sin_rho=sin(rho);
    cos_rho=cos(rho);

//    sincos(rho, sin_rho, cos_rho);

    //transformada inversa de park
    iqs_s=iqs_e*cos_rho+ids_e*sin_rho;
    ids_s=-iqs_e*sin_rho+ids_e*cos_rho;
    //transformada inversa de clark
    iasref=iqs_s;
    ibsref=-(iqs_s+1.732*ids_s)*0.5;
    icsref=-(iqs_s-1.732*ids_s)*0.5;

    iaref=iasref-icsref;
    ibref=ibsref-iasref;
    icref=icsref-ibsref;

//    iaref=5*sin(wt);
//    ibref=5*sin(wt-2*PIH/3);
//    icref=5*sin(wt+2*PIH/3);

    //definição das margens superior e inferior da histerese
    Iarefmax=iaref+margem;
    Iarefmin=iaref-margem;
    Ibrefmax=ibref+margem;
    Ibrefmin=ibref-margem;

```



```

Icrefmax=icref+margem;
Icrefmin=icref-margem;

//HISTERESE
if (Ia>Iarefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
    GpioDataRegs.GPASET.bit.GPIO1 = 1;
}
if (Ia<Iarefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
    GpioDataRegs.GPASET.bit.GPIO0 = 1;
}
if (Ib>Ibrefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
    GpioDataRegs.GPASET.bit.GPIO3 = 1;
}
if (Ib<Ibrefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPASET.bit.GPIO2 = 1;
}
if (Ic>Icrefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
    GpioDataRegs.GPASET.bit.GPIO5 = 1;
}
if (Ic<Icrefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
    GpioDataRegs.GPASET.bit.GPIO4 = 1;
}
}
if (liga1==0) { //desliga conversor
    w=0; //zera a rampa
    GpioDataRegs.GPACLEAR.bit.GPIO14 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO49 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO32 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO50 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;

    GpioDataRegs.GPBCLEAR.bit.GPIO56 = 1; //DESLIGA PRE-CARGA DO CONV2
    GpioDataRegs.GPBCLEAR.bit.GPIO57 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO53 = 1;

    GpioDataRegs.GPACLEAR.bit.GPIO15 = 1;

    rho=0;
    thetar=0;
    ui[0]=0;
    w=0;
    tempo=0;
    u[1]=0;
    erro[1]=0;
    tv=0;
}
GpioDataRegs.GPACLEAR.bit.GPIO8 = 1;

```

```

    // Acknowledge this interrupt to receive more interrupts from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void cpu_timer1_isr(void) {

    t1=(CpuTimer1.PeriodInUsec); //time between interruptions em us
    //medição da velocidade

    tv++;

    if (k>cont*0.5)    rot=-1;
    else rot=1;

    velocidade[1]=rot*24000.0*(cont)/t1; //borda de subida
    60/2500*1000000*(cont)/tv (RPM)
    velocidade[0]=velocidade[2];
    velocidade[2]=0.4*velocidade[1] + 0.6*velocidade[0];
    wrm=velocidade[2]*PIH/30;
    cont=0;
    k=0;

    //ABC para qd0 estacionario
    //Iqss=(2/3)*(Ia-(Ib+Ic)/2);
    //Idss=(Ic-Ib)/sqrt(3);

    //I0ss=(Ia+Ib+Ic)/3;
    //qd0 estacionario para referencia sincrona
    //Iqse=Iqss*cos_rho-Idss*sin_rho;
    //Idse=Iqss*sin_rho+Idss*cos_rho;

    //atualização dos graficos
    if ((samplerPlot++) == 1) {
        samplerPlot = 0;

        plot1[contaPlot] = wrm;
        plot2[contaPlot] = w;
        plot3[contaPlot] = Tem;
        plot4[contaPlot] = tv;

        if (++contaPlot== 470) {
            contaPlot = 0;
        }
    }

    CpuTimer1.InterruptCount++;
    // Acknowledge this interrupt to receive more interrupts from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void xint1_isr(void)
{
    GpioDataRegs.GPASET.bit.GPIO6 = 1;
    if (enctemp==1) {

        cont++;
        // XIntruptRegs.XINT1CR.bit.POLARITY = 0;    // Falling(0) edge interrupt
        // enctemp=0;
    }
}

```

```
        if (GpioDataRegs.GPADAT.bit.GPIO21) k++;    //testa sinal em quadratura
para definição do sentido
    }
    //else {
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    //    XIntruptRegs.XINT1CR.bit.POLARITY = 1;    // Rising(1) edge interrupt
    //    enctemp=1;
    //}
        // Acknowledge this interrupt to get more from group 1
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
    }
```

## APÊNDICE B – CÓDIGO FONTE DO DSP PARA CONTROLE DE POSIÇÃO DO MOTOR DE INDUÇÃO POR ORIENTAÇÃO DE CAMPO

```

#include "DSP28x_Project.h" // Device Headerfile and Examples Include File
#include "Example_freqcal.h" // Example specific include file
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "C28x_FPU_FastRTS.h"

//Definição dos botões
#define P22 (GpioDataRegs.GPBDAT.bit.GPIO47)
#define P23 (GpioDataRegs.GPCDAT.bit.GPIO80)
#define P24 (GpioDataRegs.GPCDAT.bit.GPIO81)
#define P25 (GpioDataRegs.GPCDAT.bit.GPIO82)
#define P26 (GpioDataRegs.GPCDAT.bit.GPIO83)

//
#define PIH2          1.570796327
#define PIH          3.141592654

//Vetores das interrupções
interrupt void cpu_timer2_isr(void);
interrupt void cpu_timer1_isr(void);
interrupt void xint1_isr(void);

//Parametros do motor
float Rs=5.35, Rr=9.443, Ls=0.389, Lr=0.377, Lm=0.360, J=0.013;
float wref=1800*PIH/30, Tem, P=4;
float lambdadr_e=0.648; //469.002/(2*PIH*60)*0.635; *1.4
float wrm=0, thetar=0; //frequencia eletrica de base
float ids_e, iqs_e, iqs_s, ids_s;
float Iqse, Idse, Iqss, Idss;
float sin_rho, cos_rho, rho=0;
float Tn=4.14;
float Kw2, Kiq;

float k=1, t1=0, t0=0;
Uint32 cont=0, tempo=0, tv=0;
//Declaração de variáveis
int liga1=0, rot=1, enctemp=1;
float velocidade[3]={0,0,0}, Ia, Ib, Ic;
float AdcResult0[3]={0,0,0}, AdcResult1[3]={0,0,0}, AdcResult2[3]={0,0,0};
float margem=0.005, w=0, w2=0;
float iaref=0, ibref=0, icref=0;
float iasref=0, ibsref=0, icsref=0;
float Iarefmax, Iarefmin, Ibrefmax, Ibrefmin, Icrefmax, Icrefmin;
//variaveis do pi velocidade
float u[2]={0,0}, ui[2]={0,0}, erro[2]={0,0};
float Kp=2.921, Ki=4.149e-4;
//variaveis do pi posição
float posref=0;
float upos[2]={0,0}, uipos[2]={0,0}, erropos[2]={0,0};
float Kppos=64, Kipos=1.6e-4;

// ----- BUFFERS PARA GRAFICOS ----- //
float plot1[470];

```

```

float plot2[470];
float plot3[470];
long int plot4[470];
unsigned long int contaPlot = 0, samplerPlot = 0;
int divfreq=0;

//Configurações do módulo AD
EALLOW;
// Default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3) = 25.0
MHz
EDIS;
#define ADC_CKPS 0x1 // ADC module clock = HSPCLK/2*ADC_CKPS = 25.0MHz/(1*2)
= 12.5MHz
#define ADC_SHCLK 0xf // S/H width in ADC module periods =
16 ADC clocks
#define AVG 1000 // Average sample limit
#define ZOFFSET 0x00 // Average Zero offset
#define BUF_SIZE 2048 // Sample buffer size

void main(void)
{

// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the DSP2833x_SysCtrl.c file.
InitSysCtrl();

// Specific clock setting for this project:
EALLOW;
SysCtrlRegs.HISPCP.all = ADC_MODCLK; // HSPCLK = SYSCLKOUT/ADC_MODCLK
EDIS;

// Step 2. Initialize GPIO:
// This example function is found in the DSP2833x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.

// configGPIO();
InitGpio();
EALLOW;
GpioCtrlRegs.GPAPUD.all = 0xFFFF; // Pullup's disabled GPIO00-GPIO31
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO4 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1; //define como saída
GpioCtrlRegs.GPADIR.bit.GPIO8 = 1; //define como saída
GpioCtrlRegs.GPBDIR.bit.GPIO32 = 1; //define como saída COOLER CONV1
GpioCtrlRegs.GPBDIR.bit.GPIO49 = 1; //define como saída RELÉ DO RESISTOR DA
PRE-CARGA
GpioCtrlRegs.GPBDIR.bit.GPIO50 = 1; //conv7
//GPIO DO CONV2 (PRE-CARGA)
GpioCtrlRegs.GPBDIR.bit.GPIO56 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO57 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO53 = 1;
EDIS;

```

```

// GPIO20 is XINT1, GPIO21 is XINT2
EALLOW;
GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 20; // Xint1 is GPIO20
EDIS;

// Only init the GPIO for EQep1 and EPwm1 in this case
// This function is found in DSP2833x_EQep.c

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

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the DSP2833x_PieCtrl.c file.
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 in this example. This is useful for debug purposes.
// The shell ISR routines are found in DSP2833x_DefaultIsr.c.
// This function is found in DSP2833x_PieVect.c.
InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
EALLOW; // This is needed to write to EALLOW protected registers
// PieVectTable.EPWM1_INT= &prdTick;
PieVectTable.TINT2 = &cpu_timer2_isr;
PieVectTable.XINT13 = &cpu_timer1_isr;
PieVectTable.XINT1 = &xint1_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize all the Device Peripherals:
// Example specific ePWM setup. This function is found
// in Example_EPwmSetup.c
// EPwmSetup();
InitAdc(); // For this example, init the ADC
// configADC();
////////////////////////////////////
// Specific ADC setup for this example:
AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;
AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS;
AdcRegs.ADCTRL1.bit.SEQ_CASC = 1; // 1 Cascaded mode
AdcRegs.ADCMAXCONV.all = 7;
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // ADCINA0 1o Conv. SEQ1 /* a conv00
eh o pino zero */
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 1; // ADCINA1 2o Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 2; // ADCINA2 3o Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 3; // ADCINA3 4o Conv. SEQ1
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 4; // ADCINA4 5o Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 5; // ADCINA5 6o Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 6; // ADCINA6 7o Conv. SEQ2

```

```

    AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 7; // ADCINA7 8ª Conv. SEQ2
    AdcRegs.ADCTRL1.bit.CONT_RUN = 1; // Setup continuous run
    //////////////////////////////////////

    InitCpuTimers(); // For this example, only initialize the Cpu Timers

    ConfigCpuTimer(&CpuTimer1, 150, 1000); //Interrompe a cada 5 ms
    ConfigCpuTimer(&CpuTimer2, 150, 10); // 10 us

    CpuTimer2Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit =
0
    CpuTimer1Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit =
0

// Step 5. User specific code, enable interrupts:
// Enable CPU INT1 which is connected to CPU-Timer 0:
IER |= M_INT1; // Enable CPU Interrupt 1, adc, timer0, intext1
// IER |= M_INT3;
IER |= M_INT13; //timer1
IER |= M_INT14; //timer2

// Enable TINT0 in the PIE: Group 3 interrupt 1
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE block
PieCtrlRegs.PIEIER1.bit.INTx4 = 1; // Enable PIE Gropu 1 INT4 int ext 1

PieCtrlRegs.PIEIER1.bit.INTx7 = 1; //timer2 epwm
PieCtrlRegs.PIEIER3.bit.INTx1 = 1; //timer1

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

// Configure XINT1
XIntruptRegs.XINT1CR.bit.POLARITY = 1; // Falling(0) edge interrupt

// Enable XINT1 and XINT2
XIntruptRegs.XINT1CR.bit.ENABLE = 1; // Enable Xint1

// Start SEQ1 adc
AdcRegs.ADCTRL2.all = 0x2000;

    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;

//constante de iq
    Kiq=(4/(3*P))*Lr/(Lm*lambda_r_e);
//definição do valor constante de id
    ids_e=1.8;
//constante de w2
    Kw2=(Rr/Lr)*(1/ids_e);

    for(;;)
    {

        // Conversor 01

```

```

        if (P22)
            liga1 = 1;          // Liga
        if (P23)
            liga1 = 0;          // Desliga
        // Conversor 02
        if (P25)
            GpioDataRegs.GPASET.bit.GPIO15 = 1;          // Liga

        if (P26)
            GpioDataRegs.GPACLEAR.bit.GPIO15 = 1;        // Desliga
    }
}

```

```

interrupt void cpu_timer2_isr(void) {
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
    GpioDataRegs.GPASET.bit.GPIO8 = 1;
    t0=(CpuTimer2.PeriodInUsec)*1e-6; //time between interruptions

    tempo++;

    //Calcula wrm e thetar
    thetar=tetar+wrm*t0;          //angulo mecanico
    //if (thetar>=2*PIH) {thetar=thetar-2*PIH;}
    //if (thetar<=-2*PIH) {thetar=thetar+2*PIH;}

    //proteção contra sobre velocidades
    if (wrm*rot>200)          liga1=0;

    if (liga1) {

        GpioDataRegs.GPBSET.bit.GPIO56 = 1; //PRE-CARGA DO CONV2
        GpioDataRegs.GPBSET.bit.GPIO57 = 1;
        GpioDataRegs.GPBSET.bit.GPIO53 = 1;

        GpioDataRegs.GPASET.bit.GPIO14 = 1; //led
        GpioDataRegs.GPBSET.bit.GPIO32 = 1; //cooler
        GpioDataRegs.GPBSET.bit.GPIO49 = 1; //R pre-carga
        GpioDataRegs.GPBSET.bit.GPIO50 = 1; //conv7

        tempo=tempo+t0;

        //REFERENCIA DE POSIÇÃO
        /**
        if (tempo<=100000) posref=0;
        else if (tempo<=200000) posref=3;
        else if (tempo<=300000) posref=6;
        else if (tempo<=400000) posref=3;
        else posref=0;

        /**/
        // if (tempo>=1) posref=4;

        // Ia Rotor
        // AdcResult0[0]=AdcResult0[2];
        AdcResult0[1]=(AdcRegs.ADCRESULT2 >> 4);
        // AdcResult0[2]=0.2*AdcResult0[1] + 0.8*AdcResult0[0];
        // Ib Rotor
        // AdcResult1[0]=AdcResult1[2];
        AdcResult1[1]=(AdcRegs.ADCRESULT1 >> 4);
        // AdcResult1[2]=0.2*AdcResult1[1] + 0.8*AdcResult1[0];

```



```

// Ic Rotor
// AdcResult2[0]=AdcResult2[2];
// AdcResult2[1]=(AdcRegs.ADCRESULT0 >> 4);
// AdcResult2[2]=0.2*AdcResult2[1] + 0.8*AdcResult2[0];

Ia=(AdcResult0[1]*0.02402-49.51);//((AdcResult0[1]/4095.0*3.0-
1.495)*0.49/0.023-0.33)*1.53846;
Ib=(AdcResult1[1]*0.02511-51.70);//((AdcResult1[1]/4095.0*3.0-
1.495)*0.49/0.022-0.31;
Ic=(AdcResult2[1]*0.02877-58.94);//((AdcResult2[1]/4095.0*3.0-
1.495)*0.97/0.038-0.15;

//PROTEÇÃO contra sobrecorrente
if (Ia>50) liga1=0;
if (Ib>50) liga1=0;
if (Ic>50) liga1=0;

//Controlador PI de posição
upos[0]=upos[1];
erropos[0]=erropos[1];
erropos[1]=posref-thetar;
upos[1]=upos[0]+Kppos*(erropos[1]-erropos[0])+Kipos*erropos[1];
w=upos[1];

//Controlador PI de velocidade
u[0]=u[1];
if (u[0]>15) u[0]=15; //Saturação
if (u[0]<-15) u[0]=-15;
erro[0]=erro[1];
erro[1]=w-wrm;
u[1]=u[0]+Kp*(erro[1]-erro[0])+Ki*erro[1];
Tem=u[1];

//Controle por campo orientado
iqs_e=Kiq*Tem;
// ids_e=(lambdadr_e)/(Lm);
// iqs_e=Tem;
w2=Kw2*iqs_e;
rho=rho+(w2+wrm*P*0.5)*t0;
if (rho>=2*PIH) rho=rho-2*PIH;
if (rho<=-2*PIH) rho=rho+2*PIH;
sin_rho=sin(rho);
cos_rho=cos(rho);

// sincos(rho, sin_rho, cos_rho);

//transformada inversa de park
iqs_s=iqs_e*cos_rho+ids_e*sin_rho;
ids_s=-iqs_e*sin_rho+ids_e*cos_rho;
//transformada inversa de clark
iasref=iqs_s;
ibsref=-(iqs_s+1.732*ids_s)*0.5;
icsref=-(iqs_s-1.732*ids_s)*0.5;

iaref=iasref-icsref;
ibref=ibsref-iasref;
icref=icsref-ibsref;

// iaref=5*sin(wt);

```

```

//      ibref=5*sin(wt-2*PIH/3);
//      icref=5*sin(wt+2*PIH/3);

//definição das margens superior e inferior da histerese
Iarefmax=iaref+margem;
Iarefmin=iaref-margem;
Ibrefmax=ibref+margem;
Ibrefmin=ibref-margem;
Icrefmax=icref+margem;
Icrefmin=icref-margem;

//HISTERESE
if (Ia>Iarefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
    GpioDataRegs.GPASET.bit.GPIO1 = 1;
}
if (Ia<Iarefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
    GpioDataRegs.GPASET.bit.GPIO0 = 1;
}
if (Ib>Ibrefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
    GpioDataRegs.GPASET.bit.GPIO3 = 1;
}
if (Ib<Ibrefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPASET.bit.GPIO2 = 1;
}
if (Ic>Icrefmax) {
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
    GpioDataRegs.GPASET.bit.GPIO5 = 1;
}
if (Ic<Icrefmin) {
    GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
    GpioDataRegs.GPASET.bit.GPIO4 = 1;
}
}
if (liga1==0) { //desliga conversor
    w=7; //zera a rampa
    GpioDataRegs.GPACLEAR.bit.GPIO14 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO49 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO32 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO50 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO0 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;

    GpioDataRegs.GPBCLEAR.bit.GPIO56 = 1; //DESLIGA PRE-CARGA DO CONV2
    GpioDataRegs.GPBCLEAR.bit.GPIO57 = 1;
    GpioDataRegs.GPBCLEAR.bit.GPIO53 = 1;

    GpioDataRegs.GPACLEAR.bit.GPIO15 = 1;

    posref=0;
    rho=0;
}

```

```

    thetar=0;
    w=0;
    tempo=0;
    ui[0]=0;
    u[1]=0;
    erro[1]=0;
    uipos[0]=0;
    upos[1]=0;
    erropos[1]=0;
}
GpioDataRegs.GPACLEAR.bit.GPIO8 = 1;
// Acknowledge this interrupt to receive more interrupts from group 1
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

```

```

interrupt void cpu_timer1_isr(void) {

    t1=(CpuTimer1.PeriodInUsec); //time between interruptions em us
    //medição da velocidade

    if (k>cont*0.5)    rot=-1;
        else rot=1;

        velocidade[1]=rot*24000.0*(cont)/t1; //borda de subida
60/2500*1000000*(cont)/tv (RPM)
        velocidade[0]=velocidade[2];
        velocidade[2]=0.4*velocidade[1] + 0.6*velocidade[0];
        wrm=velocidade[2]*PIH/30;
        cont=0;
        k=0;

    //ABC para qd0 estacionario
    //Iqss=(2/3)*(Ia-(Ib+Ic)/2);
    //Idss=(Ic-Ib)/sqrt(3);

    //I0ss=(Ia+Ib+Ic)/3;
    //qd0 estacionario para referencia sincrona
    //Iqse=Iqss*cos_rho-Idss*sin_rho;1
    //Idse=Iqss*sin_rho+Idss*cos_rho;

    if (divfreq>=3) {
    tv++;
    //atualização dos graficos
    if ((samplerPlot++) == 1) {
        samplerPlot = 0;

        plot1[contaPlot] = wrm;
        plot2[contaPlot] = posref;
        plot3[contaPlot] = iqs_e;
        plot4[contaPlot] = tv;

        if (++contaPlot == 470) {
            contaPlot = 0;
        }
    }
    divfreq=0;
}
else divfreq++;

```

```

    CpuTimer1.InterruptCount++;
    // Acknowledge this interrupt to receive more interrupts from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void xint1_isr(void)
{
    GpioDataRegs.GPASET.bit.GPIO6 = 1;
    if (enctemp==1) {

        cont++;
        // XIntruptRegs.XINT1CR.bit.POLARITY = 0;    // Falling(0) edge interrupt
        // enctemp=0;
        if (GpioDataRegs.GPADAT.bit.GPIO21) k++;    //testa sinal em quadratura
        para definição do sentido
    }
    //else {
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        // XIntruptRegs.XINT1CR.bit.POLARITY = 1;    // Rising(1) edge interrupt
        // enctemp=1;
        //}
        // Acknowledge this interrupt to get more from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

```

## APÊNDICE C – CÓDIGO FONTE DO DSP PARA ACIONAMENTO DO MOTOR DE INDUÇÃO POR CONTROLE V/f

```

#include "DSP28x_Project.h" // Device Headerfile and Examples Include File
#include "Example_freqcal.h" // Example specific include file
#include "IQmathLib.h"
//
#define PI 3.14156
_iq in1, out1;

#define P22 (GpioDataRegs.GPBDAT.bit.GPIO47)
#define P23 (GpioDataRegs.GPCDAT.bit.GPIO80)
#define P24 (GpioDataRegs.GPCDAT.bit.GPIO81)
#define P25 (GpioDataRegs.GPCDAT.bit.GPIO82)
#define P26 (GpioDataRegs.GPCDAT.bit.GPIO83)

void EPwmSetup(void);
interrupt void cpu_timer0_isr(void);
__interrupt void prdTick(void);
interrupt void cpu_timer1_isr(void);
interrupt void xint1_isr(void);
interrupt void adc_isr(void);

float velocidade[3]={0,0,0}, velbaixa[3]={0,0,0}, DUTY1, DUTY2, DUTY3, SP1=0.0;
float thetaA=0, thetaB=4*PI/3, thetaC=2*PI/3;
float t=0, w=7, sen=0, kA=1, kB=1, kC=1, ma;
float CPU_CLK1=150000000, PWM_CLK1=10000;
int liga1=0, rot=1;

float Ia, Ib, Ic;
float AdcResult0[3]={0,0,0}, AdcResult1[3]={0,0,0}, AdcResult2[3]={0,0,0};

float tv=0;
int desligar=0;
float tempo=0, k=1, tr1=0, tr2=0;
int cont=0, enctemp=1, aux=0;

FREQCAL freq=FREQCAL_DEFAULTS;

EALLOW;
// Default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3) = 25.0
MHz
EDIS;

#define ADC_CKPS 0x1 // ADC module clock = HSPCLK/2*ADC_CKPS = 25.0MHz/(1*2)
= 12.5MHz
#define ADC_SHCLK 0xf // S/H width in ADC module periods =
16 ADC clocks
#define AVG 1000 // Average sample limit
#define ZOFFSET 0x00 // Average Zero offset
#define BUF_SIZE 2048 // Sample buffer size

void main(void)
{
float AdcResult[16] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
// Step 1. Initialize System Control:

```

```

// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the DSP2833x_SysCtrl.c file.
InitSysCtrl();

SP1=CPU_CLK1/(2*PWM_CLK1);

// Specific clock setting for this example:
EALLOW;
SysCtrlRegs.HISPCP.all = ADC_MODCLK;    // HSPCLK = SYSCLKOUT/ADC_MODCLK
EDIS;

// Step 2. Initialize GPIO:
// This example function is found in the DSP2833x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.
// InitGpio(); // Skipped for this example

// Only init the GPIO for EQep1 and EPwm1 in this case
// This function is found in DSP2833x_EQep.c
InitGpio();
// InitEQep1Gpio();
    InitEPwm1Gpio();
    InitEPwm2Gpio();
    InitEPwm3Gpio();

EALLOW;
GpioCtrlRegs.GPBDIR.bit.GPIO32 = 1; //define como saída COOLER CONV1
GpioCtrlRegs.GPBDIR.bit.GPIO49 = 1; //define como saída RELÉ DO RESISTOR DA
PRE-CARGA
//GPIO DO CONV2 (PRE-CARGA)
GpioCtrlRegs.GPBDIR.bit.GPIO56 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO57 = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO53 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1; //define como saída
EDIS;
// GPIO20 is XINT1, GPIO21 is XINT2
EALLOW;
GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 20; // Xint1 is GPIO20
EDIS;

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

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the DSP2833x_PieCtrl.c file.
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 in this example. This is useful for debug purposes.
// The shell ISR routines are found in DSP2833x_DefaultIsr.c.
// This function is found in DSP2833x_PieVect.c.

```

```

InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
EALLOW; // This is needed to write to EALLOW protected registers
PieVectTable.EPwm1_INT= &prdTick;
PieVectTable.TINT0 = &cpu_timer0_isr;
PieVectTable.XINT13 = &cpu_timer1_isr;
PieVectTable.ADCINT = &adc_isr;
PieVectTable.XINT1 = &xint1_isr;
// PieVectTable.XINT2 = &xint2_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize all the Device Peripherals:
// Example specific ePWM setup. This function is found
// in Example_EPwmSetup.c
EPwmSetup();

InitAdc(); // For this example, init the ADC

////////////////////////////////////

// Specific ADC setup for this example:
AdcRegs.ADCCTRL1.bit.ACQ_PS = ADC_SHCLK;
AdcRegs.ADCCTRL3.bit.ADCCLKPS = ADC_CKPS;
AdcRegs.ADCCTRL1.bit.SEQ_CASC = 1; // 1 Cascaded mode
AdcRegs.ADCMAXCONV.all = 7;
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // ADCINA0 1º Conv. SEQ1 /* a conv00
eh o pino zero */
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 1; // ADCINA1 2º Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 2; // ADCINA2 3º Conv. SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 3; // ADCINA3 4º Conv. SEQ1
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 4; // ADCINA4 5º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 5; // ADCINA5 6º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 6; // ADCINA6 7º Conv. SEQ2
AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 7; // ADCINA7 8º Conv. SEQ2
AdcRegs.ADCCTRL1.bit.CONT_RUN = 1; // Setup continuous run
////////////////////////////////////

InitCpuTimers(); // For this example, only initialize the Cpu Timers

ConfigCpuTimer(&CpuTimer1, 150, 10);
ConfigCpuTimer(&CpuTimer0, 150, 5000);

CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit =
0
CpuTimer1Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit =
0

// Step 5. User specific code, enable interrupts:
// Enable CPU INT1 which is connected to CPU-Timer 0:
IER |= M_INT1; // Enable CPU Interrupt 1, adc, timer0
IER |= M_INT3;
IER |= M_INT13; //timer1
IER |= M_INT1; // Enable CPU int1

// Enable TINT0 in the PIE: Group 3 interrupt 1
PieCtrlRegs.PIEIER3.bit.INTx1 = 1; //timer1
PieCtrlRegs.PIEIER1.bit.INTx6 = 1; //adc

```

```

PieCtrlRegs.PIEIER1.bit.INTx7 = 1; //timer0 epwm
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE block
PieCtrlRegs.PIEIER1.bit.INTx4 = 1; // Enable PIE Gropu 1 INT4

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

// Configure XINT1
XIntruptRegs.XINT1CR.bit.POLARITY = 1; // Falling(0) edge interrupt

// Enable XINT1 and XINT2
XIntruptRegs.XINT1CR.bit.ENABLE = 1; // Enable Xint1

freq.init(&freq); // Initializes eQEP for frequency calculation in
// FREQCAL_Init(void)function in Example_EPwmSetup.c

// Start SEQ1
AdcRegs.ADCTRL2.all = 0x2000;

EPwm1Regs.CMPA.half.CMPA=SP1;
EPwm2Regs.CMPA.half.CMPA=SP1;
EPwm3Regs.CMPA.half.CMPA=SP1;

for(;;)
{
    // Conversor 01
    if (P22)
        liga1 = 1; // Liga
    if (P23)
        liga1 = 0; // Desliga
    // Conversor 02
    if (P25)
        GpioDataRegs.GPASET.bit.GPIO15 = 1; // Liga

    if (P26)
        GpioDataRegs.GPACLEAR.bit.GPIO15 = 1; // Desliga

    /*
    k=0;
    aux=0;
    while (k<=10) {
        k++;
        if (GpioDataRegs.GPADAT.bit.GPIO20==0) {
            aux++;
        }
    }
    if (aux>=k*0.5) encntemp=0;
    else encntemp=1;
    */
    //if (GpioDataRegs.GPADAT.bit.GPIO20) tr2=0; //filtro
    //else tr1=0;

    AdcResult[0] = (AdcRegs.ADCRESULT0 >> 4); // Ia Rotor
    AdcResult[1] = (AdcRegs.ADCRESULT1 >> 4); // Ib Rotor
    AdcResult[2] = (AdcRegs.ADCRESULT2 >> 4); // Ic Rotor

}

```



```

}

interrupt void adc_isr(void) {
    // ----- CONVERSOR AD -----
    //
    // Uint32 AdcResult[16] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    // AdcResult[0] = (AdcRegs.ADCRESULT0 >> 4);           // Ia Rotor
    // AdcResult[1] = (AdcRegs.ADCRESULT1 >> 4);           // Ib Rotor
    // AdcResult[2] = (AdcRegs.ADCRESULT2 >> 4);           // Ic Rotor
    // Reinitialize for next ADC sequence
    AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 1;           // Reset SEQ1
    AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;         // Clear INT SEQ1 bit
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;    // Acknowledge interrupt to PIE
    return;
}

interrupt void cpu_timer1_isr(void) {

    // Ia Rotor
    AdcResult0[0]=AdcResult0[2];
    AdcResult0[1]=(AdcRegs.ADCRESULT2 >> 4);
    AdcResult0[2]=0.3*AdcResult0[1] + 0.7*AdcResult0[0];
    // Ib Rotor
    AdcResult1[0]=AdcResult1[2];
    AdcResult1[1]=(AdcRegs.ADCRESULT1 >> 4);
    AdcResult1[2]=0.3*AdcResult1[1] + 0.7*AdcResult1[0];
    // Ic Rotor
    AdcResult2[0]=AdcResult2[2];
    AdcResult2[1]=(AdcRegs.ADCRESULT0 >> 4);
    AdcResult2[2]=0.3*AdcResult2[1] + 0.7*AdcResult2[0];

    Ia=((AdcResult0[2]*0.01561-31.85)-0.33)*1.53846; //(AdcResult0[1]/4095.0*3.0-
1.495)*0.49/0.023-0.33;
    Ib=((AdcResult1[2]*0.01632-33.298)-
0.31)*1.53846; //(AdcResult1[1]/4095.0*3.0-1.495)*0.49/0.022-0.31;
    Ic=((AdcResult2[2]*0.01870-38.1618)-
0.15)*1.53846; //(AdcResult2[1]/4095.0*3.0-1.495)*0.97/0.038-0.15;

    //PROTEÇÃO contra sobrecorrente
    // if (Ia>10) liga1=0;
    // if (Ib>10) liga1=0;
    // if (Ic>10) liga1=0;

    //velocidade = freq.freqhz_pr*60/50;           // function in
Example_EPwmSetup.c
    CpuTimer1.InterruptCount++;
    // GpioDataRegs.GPATOGGLE.bit.GPIO14 = 1; // Toggle GPIO14 once per 500
milliseconds
    GpioDataRegs.GPATOGGLE.bit.GPIO15 = 1; // Toggle GPIO14 once per 500
milliseconds

    // Acknowledge this interrupt to receive more interrupts from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void cpu_timer0_isr(void) {
    //GpioDataRegs.GPATOGGLE.bit.GPIO6 = 1;

```

```

t=(CpuTimer0.PeriodInUsec); //time between interruptions em us
//medição da velocidade
tv=tv+t; //tempo passado
tr1=tr1+t; //tempo alto
tr2=tr2+t; //tempo baixo
/*
if ((tr1+tr2)>10000) {velocidade[2]=0;}

if (GpioDataRegs.GPADAT.bit.GPIO20==0) {
    tr1=0;
    if (tr2>1)

        if (enctemp==1)
        {enctemp = 0;
        //GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        //cont++;
        }
    }
*/
if (tv>=tv) {
    if (k>cont*0.5) rot=1;
    else rot=-1;

    velocidade[1]=rot*24000.0*(cont)/tv; //borda de subida
60/2500*1000000*(cont)/tv (RPM)
    velocidade[0]=velocidade[2];
    velocidade[2]=0.2*velocidade[1] + 0.8*velocidade[0];
    tv=0;
    cont=0;
    k=0;
    //GpioDataRegs.GPATOGGLE.bit.GPIO6 = 1;
}

CpuTimer0.InterruptCount++;
//GpioDataRegs.GPATOGGLE.bit.GPIO15 = 1; // Toggle GPIO14 once per 500
milliseconds
//GpioDataRegs.GPATOGGLE.bit.GPIO14 = 1; // Toggle GPIO14 once per 500
milliseconds
// Acknowledge this interrupt to receive more interrupts from group 1
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

__interrupt void prdTick(void) // Interrupts once per ePWM period
{

    if (ligal) {
        GpioDataRegs.GPASET.bit.GPIO56 = 1; //PRE-CARGA DO CONV2
        GpioDataRegs.GPASET.bit.GPIO57 = 1;
        GpioDataRegs.GPASET.bit.GPIO53 = 1;

        GpioDataRegs.GPASET.bit.GPIO14 = 1; //led
        GpioDataRegs.GPASET.bit.GPIO32 = 1; //cooler
        GpioDataRegs.GPASET.bit.GPIO49 = 1; //R pre-carga

        t=1/PWM_CLK1;
        //if (t>=(16.667e-3)/2) {t=0;}
        if (desligar) {
            if (w>=7) w=w-74*t;

```

```

        else {w=7; liga1=0; desligar=0;}
    }
    else if (w<377) w=w+74*t; //rampa de velocidade (partida em 5s) 370/(5/t)

    thetaA = thetaA+w*t;
    thetaB = thetaB+w*t;
    thetaC = thetaC+w*t;

    ma=w*2.65252e-3; //w/377

    if (thetaA>=PI) {thetaA=thetaA-PI; kA=kA*-1;}
    if (thetaB>=PI) {thetaB=thetaB-PI; kB=kB*-1;}
    if (thetaC>=PI) {thetaC=thetaC-PI; kC=kC*-1;}

    in1=_IQ(thetaA);
    out1=_IQsin(in1);
    sen=(out1+0.0)/(536870912.0); //2^29 GLOBAL_Q=29
    DUTY1 = (kA*ma*sen+1)/2;

    in1=_IQ(thetaB);
    out1=_IQsin(in1);
    sen=(out1+0.0)/(536870912.0); //2^29 GLOBAL_Q=29
    DUTY2 = (kB*ma*sen+1)/2;

    in1=_IQ(thetaC);
    out1=_IQsin(in1);
    sen=(out1+0.0)/(536870912.0); //2^29 GLOBAL_Q=29
    DUTY3 = (kC*ma*sen+1)/2;

    //GpioDataRegs.GPATOGGLE.bit.GPIO2 = 1;
    EPwm1Regs.CMPA.half.CMPA =SP1*DUTY1;
    EPwm2Regs.CMPA.half.CMPA =SP1*DUTY2;
    EPwm3Regs.CMPA.half.CMPA =SP1*DUTY3;
}
/*
    if (liga1==0)
        if(w>7) {
            liga1=1;
            desligar=1;
        }
*/
    if (liga1==0) {
        EPwm1Regs.CMPA.half.CMPA=SP1;
        EPwm2Regs.CMPA.half.CMPA=SP1;
        EPwm3Regs.CMPA.half.CMPA=SP1;

        GpioDataRegs.GPACLEAR.bit.GPIO14 = 1;
        GpioDataRegs.GPBCLEAR.bit.GPIO49 = 1;
        GpioDataRegs.GPBCLEAR.bit.GPIO32 = 1;
        GpioDataRegs.GPBCLEAR.bit.GPIO56 = 1; //DESLIGA PRE-CARGA DO CONV2
        GpioDataRegs.GPBCLEAR.bit.GPIO57 = 1;
        GpioDataRegs.GPBCLEAR.bit.GPIO53 = 1;

        w=7;
    }

    // Acknowledge this interrupt to receive more interrupts from group 3
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
    EPwm1Regs.ETCLR.bit.INT=1;

```

```

}

interrupt void xint1_isr(void)
{
    //GpioDataRegs.GPATOGGLE.bit.GPIO6 = 1;
    if (enctemp==1) {
        GpioDataRegs.GPASET.bit.GPIO6 = 1;
        cont++;
        XIntruptRegs.XINT1CR.bit.POLARITY = 0;    // Falling(0) edge
interrupt
        enctemp=0;
        if (GpioDataRegs.GPADAT.bit.GPIO21) k++;
    }
    else {
        XIntruptRegs.XINT1CR.bit.POLARITY = 1;    // Falling(0) edge
interrupt
        enctemp=1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    }

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

```