



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LUCAS RABELO REBOUÇAS

CONTROLE DE UM MOTOR DE INDUÇÃO TRIFÁSICO COMO
SERVOPOSICIONADOR APLICADO A UM MANIPULADOR ROBÓTICO
UTILIZANDO CONTROLE DE CAMPO ORIENTADO

FORTALEZA

2017

LUCAS RABELO REBOUÇAS

CONTROLE DE UM MOTOR DE INDUÇÃO TRIFÁSICO COMO SERVOPOSICIONADOR
APLICADO A UM MANIPULADOR ROBÓTICO UTILIZANDO CONTROLE DE CAMPO
ORIENTADO

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica

Orientadora: Profa. Dra. Laurinda Lúcia Nogueira dos Reis

Coorientador: Prof. Dr. Antonio Barbosa de Souza Júnior

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- R24c Rebouças, Lucas Rabelo.
 Controle de um motor de indução trifásico como servoposicionador aplicado a um manipulador robótico utilizando controle por campo orientado / Lucas Rabelo Rebouças. – 2017.
 115 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2017.
 Orientação: Profa. Dra. Laurinda Lúcia Nogueira dos Reis.
 Coorientação: Prof. Dr. Antonio Barbosa de Souza Júnior.
1. Máquina de indução. 2. Controle de campo orientado. 3. Controlador PID. 4. Controlador RST. I.
 Título.

CDD 621.3

LUCAS RABELO REBOUÇAS

CONTROLE DE UM MOTOR DE INDUÇÃO TRIFÁSICO COMO SERVOPOSICIONADOR
APLICADO A UM MANIPULADOR ROBÓTICO UTILIZANDO CONTROLE DE CAMPO
ORIENTADO

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica

Aprovada em: 25 / 09 / 2017

BANCA EXAMINADORA

Profa. Dra. Laurinda L. N. dos Reis (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antonio B. de S. Júnior (Coorientador)
Instituto Federal do Ceará (IFCE)

Prof. Dr. Fabrício Gonzalez Nogueira
Departamento de Engenharia Elétrica (UFC)

Prof. Dr. Wilkley Bezerra Correia
Departamento de Engenharia Elétrica (UFC)

Prof. Dr. George André Pereira Thé
Departamento de Teleinformática (UFC)

A ela. Novembro está chegando.

AGRADECIMENTOS

À Delania, que sempre esteve ao meu lado e por entender, às vezes, o tempo que precisei me ausentar para conclusão deste trabalho.

Ao Prof. Antônio Barbosa ou, do tempo de UNIFOR, Toim da Caucaia, pela insistência em acreditar que este trabalho daria certo. Sem a sua ajuda, conselhos, disponibilidade de tempo e conversas jogadas fora, eu não estaria terminando este trabalho. Meu muito obrigado.

À Prof. Laurinda pela atenção, ensinamentos, conselhos e confiança!

Ao Prof. Éber que, mesmo distante hoje, foi o primeiro que me incentivou desde os tempos de graduação para iniciar esta etapa.

Aos professores do Departamento de Engenharia Elétrica: Fabrício, Bismark, Wilkley, Luiz Henrique, Fernando Antunes, Paulo Praça, Tobias, Ruth e etc. Obrigado pelos ensinamentos.

À COELCE, em nome de Giordane Silveira e Eduardo Formiga, pela confiança e disponibilidade para que eu pudesse terminar mais essa etapa.

Aos amigos, por todos os momentos de descontração. Sem estes momentos, não dá.

Ao CNPq pelo apoio financeiro dado ao projeto, através da chamada universal MCTI/CNPq Nº 14/2014, com o projeto número 442573/2014-16.

“Fortuna Audaces Sequitur”

(Públio Virgílio Maro, (19 a.C.))

RESUMO

Este trabalho aborda o estudo e implementação de um controle aplicado a um motor de indução trifásico (MIT) utilizado como servoposicionador aplicado ao primeiro grau de liberdade de um manipulador robótico. O MIT foi acionado utilizando-se a estratégia de controle de campo orientado indireto e utilizando controladores do tipo Proporcional-Integral (PI) e RST para as malhas de controle de corrente, velocidade e posição. Realizou-se a identificação da malha de corrente e a simulação do controlador PI aplicado a esta malha através do *software Simulink*[®], utilizando o método de Ziegler-Nichols para a sintonia dos parâmetros do controlador. Para a malha de velocidade, realizou-se o estudo entre os controladores do tipo PI e RST. Para o controlador RST utilizou-se a técnica de alocação de polos para definição dos parâmetros. Como as respostas dos controladores apresentaram valores próximos, então optou-se pelo controlador RST para ser mantido quando da implementação da malha de controle de posição. Logo após, foi desenvolvido um controlador do tipo Proporcional aplicado à malha de posição. Aplicou-se perfis de referência, tais como degraus positivos e negativos e trapezoidais, para se avaliar o comportamento do controlador. Por fim, foi realizada a implementação em bancada da técnica de controle de orientação de campo para o acionamento do MIT e das leis de controle das malhas com os parâmetros obtidos em simulação. Utilizou-se um processador digital de sinais, DSP, da *Texas Instruments*[®], modelo TMS320F28335[®], com o eixo do MIT acoplado à estrutura de um manipulador robótico, conseguindo-se resultados satisfatórios, próximos aos de simulação e dentro dos critérios de projeto estabelecidos.

Palavras-chave: Máquina de indução. Controle de campo orientado. Controlador RST. Controlador PID.

ABSTRACT

This work deals with the study and implementation of a three-phase induction motor user as servopositioner applied to the first degree of freedom of a robotic manipulator. The induction motor was modeled using the indirect oriented field control strategy and using a longPI and RST type controllers for the control loops. The current loop identification and the simulation of the PI controller applied to this loop were performed using the *Simulink*[®] software, using Ziegler-Nichols methods for tuning the controller parameters. For the velocity loop, the study was carried out between the Proporcional-Integral and RST type controllers. For the RST controller, the pole allocation technique was used to define the parameters. Thus, the responses presented close values, so the RST controller was chosen to be maintained when implementing the position control loop. Soon after, a controller of the type Proportional applied to the position loop was developed. Reference profiles were applied to evaluate the behavior of the controller. Finally, the implementation of the field orientation control technique for the activation of the induction motor and the laws of control of the meshes with the parameters obtained in simulation were carried out in bench. A digital signal processor, DSP, from *Texas Instruments*[®], model TMS320F28335[®], was used with the induction motor shaft coupled to the structure of a robotic manipulator, achieving satisfactory results, close to those of simulation and within the established design criteria.

Keywords: Induction machine. Field oriented control. RST controller. PID controller.

LISTA DE ILUSTRAÇÕES

Figura 1 – Transformação da máquina simétrica trifásica em uma máquina simétrica bifásica.	25
Figura 2 – Sistema de eixos representando a transformada de Park.	26
Figura 3 – Configuração do acionamento de campo orientado indireto para uma máquina de indução.	29
Figura 4 – Diagrama do espaço vetorial.	32
Figura 5 – Diagrama de blocos do controlador PID.	36
Figura 6 – Caracterização da resposta ao degrau no método de resposta ao degrau de (ZIEGLER; NICHOLS, 1942).	38
Figura 7 – Estrutura do controlador RST.	40
Figura 8 – Diagrama de blocos do controlador RST.	41
Figura 9 – Estrutura do manipulador montada no GPAR e os graus de liberdade existentes.	45
Figura 10 – Acionamento da estrutura através de polias e correia.	45
Figura 11 – Instalação do <i>encoder</i> na ponta do eixo do motor.	46
Figura 12 – Circuitos utilizados para controle e acionamento do manipulador.	47
Figura 13 – Diagrama simplificado da malha de corrente.	48
Figura 14 – Sinais aplicados e obtidos para identificação da malha de corrente.	49
Figura 15 – Validação do modelo obtido para a malha de corrente.	49
Figura 16 – Diagrama de blocos montado no <i>Simulink</i> [®] para simulação da malha de corrente.	51
Figura 17 – Resultado da simulação da malha de corrente para referência em $i_q^* = +1,0A$	52
Figura 18 – Resultado da simulação da malha de corrente para referência em $i_q^* = -1,0A$	52
Figura 19 – Resultado da Malha de Corrente com referência de $i_q^* = +1,0A$ e $i_d^* = +1,3A$	54
Figura 20 – Resultado da Malha de Corrente com referência de $i_q^* = +1,0A$ e $i_d^* = -1,3A$	55
Figura 21 – Resultado da Malha de Corrente com referência de $i_q^* = -1,0A$ e $i_d^* = +1,3A$	56
Figura 22 – Resultado da Malha de Corrente com referência de $i_q^* = -1,0A$ e $i_d^* = -1,3A$	56
Figura 23 – Resultado da Malha de Corrente com referência de $i_q^* = 0,0A$ e $i_d^* = +1,3A$	57
Figura 24 – Resultado da Malha de Corrente com referência de $i_q^* = 0,0A$ e $i_d^* = -1,3A$	58
Figura 25 – Diagrama simplificado da malha de velocidade.	59
Figura 26 – Sinais aplicados e obtidos para identificação da malha de velocidade.	60
Figura 27 – Validação do modelo obtido para a malha de velocidade.	61

Figura 28 – Diagrama do <i>Simulink</i> [®] para simulação dos controladores da malha de velocidade.	63
Figura 29 – Resposta da simulação com referência em degrau positivo.	63
Figura 30 – Resposta das simulações com referência em degrau positivo e negativo.	64
Figura 31 – Resposta das simulações com referência senoidal.	65
Figura 32 – Resposta das simulações com referência trapezoidal.	65
Figura 33 – Resposta dos controladores na malha de velocidade com referência em degrau positivo.	67
Figura 34 – Resposta dos controladores na malha de velocidade com referência em degrau positivo e negativo.	67
Figura 35 – Resposta dos controladores na malha de velocidade com referência senoidal.	68
Figura 36 – Resposta dos controladores na malha de velocidade com referência trapezoidal.	68
Figura 37 – Diagrama simplificado da malha de posição.	69
Figura 38 – Resposta do controlador de posição com referência em degrau sequencial.	70
Figura 39 – Resposta do controlador de posição com referência em degrau.	71
Figura 40 – Resposta do controlador de posição com referência em degrau positivo e negativo.	72
Figura 41 – Resposta do controlador de posição com referência trapezoidal.	72
Figura 42 – Motor WEG a ser utilizado.	79
Figura 43 – Inversor trifásico da <i>Semikron</i> [®]	80
Figura 44 – <i>Encoder</i> utilizado da <i>Avago</i> [®]	81
Figura 45 – Sensor de corrente utilizado da <i>LEM</i> [®]	82
Figura 46 – Processador digital de sinais utilizado.	83
Figura 47 – Placa condicionadora de sinais.	84
Figura 48 – Fonte auxiliar desenvolvida em laboratório.	85
Figura 49 – Pontos obtidos para calibração dos sensores de corrente.	113
Figura 50 – Reta estimada através do métodos dos mínimos quadrados para cada fase.	113

LISTA DE TABELAS

Tabela 2 – Parâmetros do controlador PID obtidos a partir do método de resposta ao degrau de (ZIEGLER; NICHOLS, 1942).	38
Tabela 3 – Critérios de projeto para o controlador da malha de corrente.	50
Tabela 4 – Características da resposta em simulação do controlador projetado para a malha de corrente.	53
Tabela 5 – Desempenho do controlador simulado para malha de corrente.	53
Tabela 6 – Desempenho do controlador para referência $i_q^* = +1,0A$	55
Tabela 7 – Desempenho do controlador para referência $i_q^* = -1,0A$	57
Tabela 8 – Índice de desempenho do controlador da malha de corrente.	58
Tabela 9 – Critérios de projeto para o controlador da malha de velocidade.	61
Tabela 10 – Características dos controladores de simulação para a malha de velocidade.	64
Tabela 11 – Índice de desempenho dos controladores simulados da malha de velocidade.	66
Tabela 12 – Índice de desempenho dos controladores na malha de velocidade.	69
Tabela 13 – Características da resposta do controlador da malha de posição.	71
Tabela 14 – Índices de desempenho do controlador da malha de posição.	73
Tabela 15 – Folha de dados do motor de indução a ser utilizado.	80
Tabela 16 – Folha de dados do <i>encoder</i> utilizado.	81
Tabela 17 – Características do sensor de corrente a ser utilizado.	82
Tabela 18 – Características do DSP a ser utilizado.	83
Tabela 19 – Correntes constantes aplicadas aos sensores de corrente e valores obtidos.	112

LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico/Digital
CA	Corrente Alternada
CC	Corrente Contínua
DEE	Departamento de Engenharia Elétrica
DSP	<i>Digital Signal Processors</i>
DTC	<i>Direct Torque Control</i>
GPAR	Grupo de Pesquisa de Automação e Robótica
ISO	<i>International Standard Organization</i>
MIPS	Milhões de Instruções por Segundo
MIT	Motor de Indução Trifásico
MRAC	<i>Model Reference Adaptive Control</i>
PD	Proporcional-Derivativo
PI	Proporcional-Integral
PID	Proporcional-Integral-Derivativo
PPR	Pulsos por Revolução
PRBS	<i>Pseudo-Random Binary Sequence</i>
PWM	<i>Pulse Width Modulation</i>
RMSE	<i>Root Mean Squared Error</i>
RPP	Rotacional-Prismático-Prismático
RST	Sigla para os polinômios R, S e T do controlador
SISO	<i>Single-input, single-output</i>
SMC	<i>Slide Mode Controller</i>
SVPWM	<i>Space Vector Pulse Width Modulation</i>
UFC	Universidade Federal do Ceará

LISTA DE SÍMBOLOS

α	Eixo real do plano de coordenadas
β	Eixo imaginário do plano de coordenadas
τ_{min}	Menor constante de tempo de interesse
θ_r	Posição atual do eixo do motor
θ_r^*	Referência de posição
θ_{sl}	Posição de escorregamento
μ	Média aritmética da variável
σ	Desvio padrão
ω_{sl}	Frequência de escorregamento
ω_r^*	Referência de velocidade
ω_r	Velocidade atual do eixo do motor
ω_1	Velocidade síncrona
ζ	Coefficiente de amortecimento
a	Parâmetro do método de Ziegler-Nichols
A_c	Polinômio característico de malha fechado desejado
a_1	Coefficiente do polinômio A
a_2	Coefficiente do polinômio A
am_1	Coefficiente do polinômio A_m do filtro de referência do controlador RST
am_2	Coefficiente do polinômio A_m do filtro de referência do controlador RST
A_m	Polinômio do filtro de referência do controlador RST
b_1	Coefficiente do polinômio B
b_2	Coefficiente do polinômio B
bm_0	Coefficiente do polinômio B_m do filtro de referência do controlador RST
bm_1	Coefficiente do polinômio B_m do filtro de referência do controlador RST
B_m	Polinômio do filtro de referência do controlador RST
d	Eixo direto do plano de coordenadas

e	Sinal de erro
f_1	Frequência nos terminais elétricos do estator
\hat{I}_a	Reta estimada de corrente da fase A
\hat{I}_b	Reta estimada de corrente da fase B
\hat{I}_c	Reta estimada de corrente da fase C
i_a^*	Referência de corrente da fase A
i_b^*	Referência de corrente da fase B
i_c^*	Referência de corrente da fase C
i_d	Corrente associada ao eixo direto
i_d^*	Referência de corrente associada ao eixo direto
i_q	Corrente associada ao eixo em quadratura
i_q^*	Referência de corrente associada ao eixo em quadratura
K_d	Ganho derivativo do controlador PID
K_i	Ganho integral do controlador PID
K_p	Ganho proporcional do controlador PID
L	Parâmetro do método de Ziegler-Nichols
l_{lr}	Indutância de dispersão associada ao rotor
L_{ls}	Indutância de dispersão associada ao estator
L_{ms}	Indutância de magnetização das fases relacionadas ao estator
l_r	Indutância total de cada fase do rotor
L_s	Indutância total de cada fase do estator
L'_{lr}	Indutância de dispersão referida ao estator
l_{mr}	Indutância de magnetização das fases relacionadas ao rotor
L'_r	Indutância equivalente do rotor
$M_{as,ar}$	Indutância mútua entre as fases A do estator e A do rotor
$M_{as,br}$	Indutância mútua entre as fases A do estator e B do rotor
$M_{as,cr}$	Indutância mútua entre as fases A do estator e C do rotor
$M_{bs,ar}$	Indutância mútua entre as fases B do estator e A do rotor

$M_{bs,br}$	Indutância mútua entre as fases B do estator e B do rotor
$M_{bs,cr}$	Indutância mútua entre as fases B do estator e C do rotor
$M_{cs,ar}$	Indutância mútua entre as fases C do estator e A do rotor
$M_{cs,br}$	Indutância mútua entre as fases C do estator e B do rotor
$M_{cs,cr}$	Indutância mútua entre as fases C do estator e C do rotor
M_p	Sobressinal da resposta à uma entrada em degrau
M_r	Indutância mútua entre as fases do rotor
M_s	Indutância mútua entre as fases do estator
n	Número de bits do sinal PRBS
n_a	Valor do conversor A/D da corrente da fase A
n_b	Valor do conversor A/D da corrente da fase B
n_c	Valor do conversor A/D da corrente da fase C
N_r	Número de espiras do rotor
N_s	Número de espiras do estator
q	Eixo em quadratura do plano de coordenadas
R	Polinômio R do controlador RST
r	Coefficiente do polinômio R do controlador RST
R_r	Resistência total de cada fase do rotor
S	Polinômio S do controlador RST
s_0	Coefficiente do polinômio S do controlador RST
s_1	Coefficiente do polinômio S do controlador RST
T	Polinômio T do controlador RST
T_1	Período do chaveamento do SVPWM
T_2	Período do chaveamento do SVPWM
T_a	Período da fase A
T_b	Período da fase B
T_{bits}	Tempo entre <i>bits</i> do sinal PRBS
T_c	Período da fase C

T_d	Tempo derivativo do controlador PID
T_i	Tempo integrativo do controlador PID
T_m	Torque mecânico
T_{pwm}	Período total do chaveamento do SVPWM
T_s	Período de amostragem
t_r	Tempo de subida da resposta à uma entrada em degrau
t_s	Tempo de estabilização da resposta à uma entrada em degrau
u	Sinal de controle
u_c	Sinal de referência do controlador
U_{out}	Vetor tensão de referência
v	Distúrbio aplicado ao sistema
V	Amplitude do sinal PRBS
X_a	Variável para transformação da fase A do sistema a-b-c
X_b	Variável para transformação da fase B do sistema a-b-c
X_c	Variável para transformação da fase C do sistema a-b-c
X_d	Variável para transformação do eixo direto do sistema d-q
X_q	Variável para transformação do eixo em quadratura do sistema d-q

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Revisão Bibliográfica	20
1.2	Objetivos	22
1.3	Organização do Trabalho	23
2	SISTEMA DE ACIONAMENTO COM MÁQUINA DE INDUÇÃO	24
2.1	Máquina de Indução	24
2.2	Transformadas de Clarke-Park	24
2.3	Indutâncias da máquina de indução trifásica	26
2.3.1	<i>Indutâncias mútuas entre as fases do mesmo enrolamento</i>	27
2.3.2	<i>Indutâncias mútuas entre estator e rotor</i>	28
2.4	Projeto do servoposicionador utilizando campo orientado indireto	28
2.4.1	<i>Modelagem dinâmica do campo orientado indireto para uma máquina de indução trifásica</i>	28
2.4.2	<i>Modulação por largura de pulso usando vetores espaciais (SVPWM)</i>	31
2.5	Considerações finais	33
3	CONTROLADORES APLICADOS À MÁQUINA DE INDUÇÃO	34
3.1	Controlador Proporcional-Integral-Derivativo (PID)	34
3.1.1	<i>Métodos de sintonia do controlador PID</i>	37
3.1.1.1	<i>Método de Ziegler-Nichols</i>	37
3.1.1.1.1	<i>Método da resposta ao degrau</i>	37
3.1.2	<i>PID discreto</i>	38
3.2	Controlador RST	39
3.2.1	<i>Alocação de Polos</i>	41
3.3	Considerações finais	43
4	RESULTADOS DE SIMULAÇÃO E EXPERIMENTAL	44
4.1	Bancada Experimental	44
4.2	Controlador da Malha de Corrente	47
4.2.1	<i>Identificação do modelo</i>	48
4.2.2	<i>Projeto do controlador de corrente</i>	50
4.2.3	<i>Resultados de simulação do controlador de corrente</i>	51

4.2.4	<i>Resultados experimentais do controlador de corrente</i>	54
4.3	Controlador da Malha de Velocidade	59
4.3.1	<i>Identificação do modelo</i>	59
4.3.2	<i>Projeto do controlador de velocidade</i>	61
4.3.3	<i>Resultados de simulação do controlador de velocidade</i>	62
4.3.4	<i>Resultados experimentais do controlador de velocidade</i>	66
4.4	Controlador da Malha de Posição	69
4.4.1	<i>Resultados experimentais do controlador de posição</i>	70
4.5	Considerações finais	73
5	CONCLUSÕES E TRABALHOS FUTUROS	74
5.1	Trabalhos Futuros	74
	REFERÊNCIAS	76
	ANEXOS	78
	ANEXO A – Componentes Utilizados	79
A.1	Motor de indução trifásico	79
A.2	Inversor trifásico	80
A.3	<i>Encoder</i>	81
A.4	Sensor de corrente	82
A.5	Processador digital de sinais	83
A.6	Placa condicionadora de sinais	84
A.7	Fonte auxiliar	84
	ANEXO B – Códigos para simulação no Simulink®	86
B.1	Código implementado no Controlador PI para simulação na malha de corrente no Simulink®	86
B.2	Código implementado no Controlador PI para simulação na malha de velocidade no Simulink®	87
B.3	Código implementado no Controlador RST para simulação na malha de velocidade no Simulink®	89
	ANEXO C – Código implementado no DSP	92
	ANEXO D – Código para gerar sinal PRBS	111
	ANEXO E – Método de calibração dos sensores de corrente	112

1 INTRODUÇÃO

Com o avanço da tecnologia embarcada dos microprocessadores durante as últimas décadas, há um maior interesse na construção de máquinas para a substituição da mão-de-obra humana em diversas atividades no âmbito do processo industrial, seja para uma tarefa de alto risco do ponto de vista da segurança ou apenas uma atividade repetitiva. Assim, através desse interesse e com o avanço da tecnologia eletrônica, se deu o início da ciência da Robótica, com ênfase nos robôs manipuladores.

Nos dias atuais, a automação está bastante consolidada nas mais diversas atividades industriais. A robótica, por sua vez, está em um crescimento vertiginoso dentro da indústria. Tarefas tais como manipulação de materiais, pintura, soldagem e montagem são exemplos de aplicação que os robôs executam com precisão, flexibilidade e redução de custos quando comparado ao operador humano.

Novas disciplinas da engenharia, como engenharia de manufatura, engenharia de aplicação e engenharia do conhecimento tentam lidar com a complexidade do campo de aplicação da robótica em um sistema de automação fabril (SPONG *et al.*, 2006).

O estudo e o desenvolvimento de mecanismos robóticos teve seu primeiro registro significativo na década de 1940, quando os primeiros teleoperadores foram fabricados para manipular materiais radioativos. O primeiro robô comercial controlado por computador foi introduzido pela *Unimaton Inc.*[®] no fim da década de 1950 (CABRAL, 2003). Os primeiros robôs eram controlados por computadores de processo ou minicomputadores onde todo o controle era feito de forma centralizada. O uso de microprocessadores ou microcontroladores nestas configurações era limitado para ser uma extensão do sistema central.

Com o desenvolvimento de microprocessadores mais velozes e com mais recursos foi possível descentralizar o controle do robô. O modelo IRB2000 da *Asea Brown Boveri*[®] (ABB) tem um sistema de controle formado por três processadores de 16 bits (MOTOROLA[®] 68000) chamados de computador principal, servo computador e computador de entrada e saída, e um processador digital de sinais (Texas[®] DSP TMS320C25) denominado computador axial. O servo computador implementa o controle dos seis motores de corrente alternada em malha fechada. Para isto, conta com um computador axial, que tem por função processar os sinais de controle e gerar os sinais de referência para os acionadores dos seis motores. Outra função do computador axial é processar os sinais de realimentação para o servo computador. O computador de entrada e saída fica encarregado de fazer a interface com o mundo externo e o computador principal faz

o gerenciamento de todo o sistema (CABRAL, 2003).

Atualmente os robôs manipuladores existentes na indústria, em sua maioria, são acionados por servomotores de corrente contínua que, apesar da sua facilidade de modelagem e controle deste tipo de máquina, tem uma manutenção dispendiosa devido a grande quantidade de componentes e ao contato mecânico de suas partes (KOSOW, 1996; SHIAU; LIN, 2001).

Os motores de indução trifásico (MIT) do tipo gaiola de esquilo, por sua vez, são motores de construção simples, mais baratos e de fácil manutenção, além de uma maior robustez construtiva se comparado a outros tipos de máquinas elétricas. Este tipo de máquina é amplamente utilizada em diversos processos industriais. Porém, a maior dificuldade de sua utilização para o controle de posição é a sua modelagem matemática complexa, de modo que requer um grande esforço computacional para sua implementação (LIPO; NOVOTNY, 1996).

Com o avanço da tecnologia dos processadores digitais e da eletrônica de potência, tornou-se possível a implementação de algoritmos de controle e modelagem por *software*. Consequentemente possibilitou o uso de motores de indução, mesmo com sua modelagem complexa, de forma confiável, robusta e com bom desempenho.

Nas máquinas de indução não existe um circuito físico exclusivo para o campo e isso torna o seu controle mais complexo, quando comparado com o controle da máquina de corrente contínua (BIM, 2014). Quando se manipula apenas a magnitude e a frequência das grandezas elétricas, diz-se que o controle é escalar. Por sua vez, quando a magnitude, a frequência e o fluxo magnético e, adicionalmente, a posição angular dessas grandezas é manipulada, diz-se que o controle é vetorial, também denominado por controle de campo orientado (BIM, 2014). Para o presente trabalho, ao longo do texto será utilizado o segundo termo.

Ao longo do desenvolvimento deste trabalho, foram realizados vários estudos de trabalhos anteriormente publicados, conforme apresentado a seguir.

1.1 Revisão Bibliográfica

Em (HASSE, 1969) foi proposto, pela primeira vez, o controle de velocidade do motor de indução baseado em modelo vetorial e alimentado por inversor utilizando *Pulse Width Modulation* (PWM), que representou um marco no controle das máquinas de indução do tipo gaiola de esquilo, pois o controle exibiu uma resposta dinâmica notável. Mas foi em (BLASHKE, 1972) que formulou-se a teoria geral e, dessa forma, expandiu-se a aplicação vetorial para qualquer tipo de máquina de corrente alternada e inversor, denominando-o orientação do campo.

O controle de campo orientado transforma a estrutura de controle do motor de corrente alternada semelhante àquela do motor de corrente contínua, com excitação separada e, desse modo, os acionamentos elétricos de corrente alternada passam a ter um desempenho comparável ao das máquinas de corrente contínua com excitação separada (BIM, 2014). Essa aproximação se dá através da transformação do modelo trifásico em um modelo bifásico por meio da aplicação das transformadas de Clarke-Park ao modelo dinâmico clássico do motor de indução. Como resultado obtém-se um sistema de equações baseadas em tensões e correntes que se projetam sobre eixos ortogonais que, alinhados adequadamente, confere ao modelo um comportamento dinâmico similar ao modelo da máquina de corrente contínua. Desta forma, a projeção das correntes de cada fase sobre os eixos ortogonais leva à obtenção das correntes i_d , no eixo direto, e i_q , no eixo em quadratura que são comparáveis, respectivamente, às correntes de campo e armadura do motor de corrente contínua e, portanto, variáveis teoricamente independentes ou desacopladas. Uma das dificuldades de se obter um alto desempenho no controle de campo orientado é a necessidade de uma medida, ou estimativa, precisa do escorregamento da máquina, o que permite obter a posição exata do rotor e, com isso, efetuar o desacoplamento das correntes de eixo direto e em quadratura (DINIZ, 2013).

O desenvolvimento dos processadores digitais, a partir da contínua integração da microeletrônica, na década de 1980, permitiu que o processamento de sinais e os algoritmos de controle fossem implementados por *software* e, conseqüentemente, tornou possível o uso de motores de indução de forma confiável, robusta e com um bom desempenho, tanto em regime permanente como no regime dinâmico.

Existem diversos estudos que utilizam técnicas avançadas de controle para o acionamento do motor de indução trifásico. Em (TAKAHASHI; ITOH, 1992), propõe-se o controle de posição do motor de indução através do controle não do motor em si, mas de um freio eletromagnético. O grande problema deste tipo de controle é a necessidade da adição de mais um equipamento, o que encarece o projeto. Já em (SANTANA *et al.*, 2008), usa-se controle preditivo baseado em modelo para controlar a velocidade e o fluxo do rotor. Em (BEERTEN *et al.*, 2010), utiliza-se uma estratégia preditiva aplicada ao controle direto de torque, DTC (*Direct Torque Control*), a fim de diminuir as ondulações de fluxo e torque. Em (JACOBINA *et al.*, 2003), faz-se uso de técnicas adaptativas, onde usa a estratégia do tipo modelo de referência, MRAC (*Model Reference Adaptive Control*), para o controle de velocidade do motor de indução.

Já em (LIAW *et al.*, 1993), propõe-se um controle de posição da máquina de indução

a partir do controle de campo orientado indireto aplicado à malha de corrente, que estaria totalmente desacoplada da malha mecânica. Para tanto, utiliza, além de um pré-filtro acoplado à referência de posição para efetuar o controle de velocidade, um controlador robusto na malha que tem o conjugado mecânico como saída, como na malha que utiliza o controle de posição. Em (TOLIYAT *et al.*, 1999) menciona-se que a resistência rotórica da máquina varia como uma série de fatores, tal como a temperatura e frequência, alterando os parâmetros da máquina de indução. Deste modo, propõe a sintonia da constante de tempo rotórica através da análise do fluxo estatórico e da tensão estatórica em coordenadas direta e em quadratura, a ser calculada a cada ciclo elétrico.

Existem trabalhos que abordam algoritmos de controle sem a necessidade de malhas realimentadas, tais como em (CÂMARA *et al.*, 2002), onde tem-se o uso de técnicas *encoderless/sensorless* aplicadas ao controle de uma ampla faixa de velocidade do motor de indução trifásico, porém viu-se que, para baixa velocidades, o torque é bastante reduzido. Em (FURTUNATO, 1997) é utilizada a estratégia de acionamento de campo orientado para o controle de velocidade do motor de indução, cujo objetivo é reduzir a influência da variação paramétrica do motor no controlador, empregando a técnica adaptativa de modos deslizantes, *Slide Mode Controller* (SMC).

Alguns trabalhos já foram realizados e publicados pelo GPAR (Grupo de Pesquisa de Automação e Robótica) da UFC com foco na utilização de motores de indução trifásico como servoposicionadores utilizando o controle de campo orientado indireto podem ser vistos em (DINIZ *et al.*, 2012), (DE SOUZA JUNIOR *et al.*, 2013), (REBOUÇAS *et al.*, 2015) e (SILVA *et al.*, 2015).

1.2 Objetivos

Este trabalho tem por objetivo geral implementar o controle de posição de um motor de indução trifásico aplicado ao primeiro grau de liberdade de um manipulador robótico, correspondente ao movimento giratório da base da estrutura, através do controle de campo orientado indireto, desenvolvendo-se controladores para as malhas de corrente, velocidade e posição. Para tanto, pode-se citar os seguintes objetivos específicos:

- Realizar o estudo da utilização de motores de indução trifásicos como servoposicionadores aplicados a manipuladores robóticos;
- Estudar os controladores do tipo PID e RST aplicados às máquinas de indução;

- Simular, através da ferramenta *Simulink*[®], as malhas de controle;
- Comparar os tipos de controladores utilizados para a malha de velocidade, a fim de definir o de melhor desempenho para aplicação prática;
- Montar bancada experimental, desenvolvendo os circuitos e placas de circuito impresso necessários para a realização dos ensaios;
- Implementar os controladores simulados no microprocessador digital utilizado;
- Executar ensaios experimentais dos controladores desenvolvidos e coletar os dados de respostas do sistema.

1.3 Organização do Trabalho

Este trabalho está dividido da seguinte forma: Este primeiro capítulo aborda a introdução do projeto a ser desenvolvido, procurando dar um breve histórico dos robôs manipuladores, realiza uma abordagem do acionamento de máquinas de indução trifásica e, por fim, apresenta alguns trabalhos publicados na área. O capítulo dois aborda a máquina de indução, as técnicas para transformação do modelo da máquina trifásica em bifásico, realiza um estudo sobre as indutâncias presentes na máquina e faz uma abordagem sobre o controle por orientação de campo aplicado à motores de indução. No capítulo três é elaborada uma explanação sobre os controladores de ganho constante e os métodos de Ziegler-Nichols e alocação de polos para sintonia destes controladores. Ainda no capítulo três, é apresentado o projeto do controle do motor de indução como servoposicionador. Já no capítulo quatro são apresentados os resultados obtidos em simulação, a bancada utilizada para realização do ensaios, os componentes e placas eletrônicas desenvolvidos e os resultados obtidos experimentalmente. Por fim, no capítulo cinco é realizado um fechamento global do trabalho, com as conclusões e propostas para trabalhos futuros.

2 SISTEMA DE ACIONAMENTO COM MÁQUINA DE INDUÇÃO

Neste capítulo será realizada a fundamentação teórica acerca da máquina de indução que será utilizada no controle de posição do manipulador robótico, enfatizando as indutâncias presentes. A seguir apresenta-se o conceito de controle de campo orientado indireto aplicado a motores de indução trifásicos e o projeto do servoposicionador utilizado, assim como a modelagem dinâmica para uma máquina de indução e a técnica de modulação por largura de pulso utilizando vetores espaciais.

2.1 Máquina de Indução

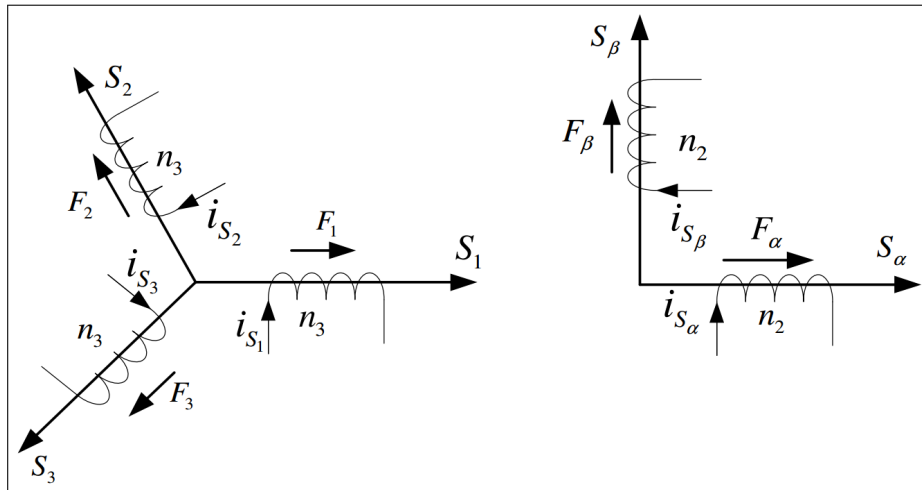
Máquinas de indução são conversores eletromecânicos rotativos de energia que funcionam baseados na produção de campos girantes no entreferro e, por essa razão, às vezes, são denominadas máquinas de campo girante (BIM, 2014).

O enrolamento polifásico do estator, ligado a uma fonte polifásica, estabelece uma distribuição de campo magnético que gira, em relação a esse enrolamento, com velocidade síncrona. Essa distribuição de campo magnético, produzida pela onda de força magnetomotriz do estator, concatena o enrolamento do rotor gerando tensões polifásicas que estabelecem correntes polifásicas nas fases. Essas correntes, por sua vez, produzem forças magnetomotrizes e, portanto, estabelecem uma distribuição de fluxo magnético que interage com o fluxo magnético de estator e, conseqüentemente, torques são desenvolvidos (BIM, 2014).

2.2 Transformadas de Clarke-Park

O primeiro passo a ser dado na obtenção de modelos mais adequados para a análise da máquina de indução é o estudo da Transformada de Clarke, ou Transformada $\alpha\beta 0$. Consiste em uma transformação linear que diagonaliza as matrizes circulantes simétricas que aparecem na formulação dos modelos da máquina trifásica. Fisicamente, a Transformada de Clarke transforma o modelo da máquina simétrica trifásica em um modelo de máquina simétrica bifásica, com mesma potência, torque, velocidade e número de polos com a introdução de algumas variáveis hipotéticas, conforme apresentado na Figura 1. O desenvolvimento matemático das equações da máquina de indução em relação às coordenadas abc e $\alpha\beta 0$ estão apresentadas em (BARBI, 1985).

Figura 1 – Transformação da máquina simétrica trifásica em uma máquina simétrica bifásica.



Fonte – (BARBI, 1985).

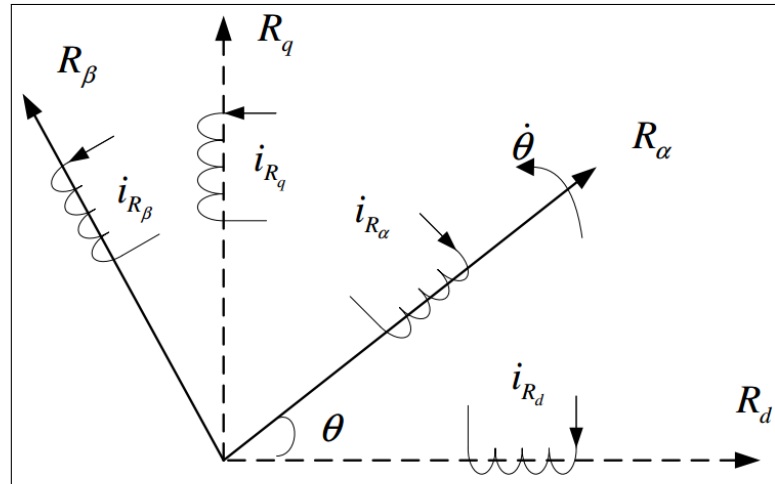
Na análise das máquinas síncronas e de indução com controle de campo orientado é utilizado o sistema de referência síncrono, cuja velocidade é a síncrona, dada por $\omega_1 = 2\pi f_1$, sendo f_1 a frequência nos terminais elétricos do estator. Esse sistema de referência é designado pelos subscritos d (eixo real) e q (eixo imaginário), denominados como eixo direto e em quadratura, respectivamente.

Para realizar a mudança do sistema de coordenadas do sistema estacionário $\alpha - \beta$ para o sistema síncrono $d - q$, é utilizada a Transformada de Park. Esta transformada possui uma grande importância no estudo da teoria de máquinas elétricas. Transforma as equações não-lineares dos modelos tradicionais em equações lineares, além da diminuição das variáveis de estado destes modelos, através da transformação de modelos trifásicos em bifásicos, facilitando assim a análise e, por conseguinte, a elaboração de controladores para estes equipamentos (BOSE, 1986). Este processo realiza a transformação do modelo da máquina trifásica, através da aplicação da transformada de Clarke, em um modelo da máquina bifásica com os enrolamentos rotóricos pseudo-estacionários (DE SOUZA JÚNIOR, 2010), como pode ser visto na Figura 2.

A Transformada de Park é obtida a partir das equações já transformadas por Clarke. Ou seja, realiza a conversão do sistema $\alpha - \beta$ advindos da transformada de Clarke, para o sistema $d - q$. A decomposição dos eixos $\alpha - \beta$ para o sistema de coordenadas $d - q$ é realizado por manipulações algébricas e é apresentado em (BIM, 2014).

Uma síntese da transformação de variáveis trifásicas, denominadas X_a , X_b e X_c , em

Figura 2 – Sistema de eixos representando a transformada de Park.



Fonte – (BARBI, 1985).

variáveis do sistema síncrono, designadas por X_d , X_q e X_0 , pode ser dada na forma de matrizes de transformação, utilizando as técnicas de Transformadas de Clarke-Park. Se θ_i é o ângulo espacial entre a fase a do enrolamento trifásico e o eixo direto do sistema síncrono, têm-se as seguintes relações matriciais (BIM, 2014):

Matriz de transformação $abc - dq0$:

$$\begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_i) & \cos\left(\theta_i - \frac{2\pi}{3}\right) & \cos\left(\theta_i + \frac{2\pi}{3}\right) \\ -\text{sen}(\theta_i) & -\text{sen}\left(\theta_i - \frac{2\pi}{3}\right) & -\text{sen}\left(\theta_i + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X_a(t) \\ X_b(t) \\ X_c(t) \end{bmatrix} \quad (2.1)$$

Matriz de transformação $dq0 - abc$:

$$\begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 1 \\ \cos\left(\theta_i - \frac{2\pi}{3}\right) & -\text{sen}\left(\theta_i - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_i + \frac{2\pi}{3}\right) & -\text{sen}\left(\theta_i + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} X_d(t) \\ X_q(t) \\ X_0(t) \end{bmatrix} \quad (2.2)$$

2.3 Indutâncias da máquina de indução trifásica

Em geral, as máquinas convencionais de indução podem ser consideradas simétricas e magneticamente lineares, e, conseqüentemente, pode-se determinar as indutâncias utilizando o princípio da superposição. O desenvolvimento matemático das indutâncias presentes na máquina de indução estão apresentadas em (BIM, 2014).

A indutância de magnetização por fase do estator, L_{ms} , é definida por:

$$L_{ms} = \left(\frac{N_s}{N_r} \right)^2 l_{mr}, \quad (2.3)$$

onde l_{mr} é a indutância associada ao fluxo magnético de entreferro do rotor por fase e N_s e N_r são o número de espiras em série por fase do estator e do rotor, respectivamente.

Ao associar a indutância de dispersão, L_{ls} (estator) e l_{lr} (rotor), as indutâncias próprias totais de cada uma das fases do estator e do rotor são dadas, respectivamente, por:

$$L_s = L_{ls} + L_{ms} \quad \text{e} \quad l_r = l_{lr} + \left(\frac{N_r}{N_s} \right)^2 L_{ms} \quad (2.4)$$

Ao substituir o rotor com N_r espiras por um equivalente com N_s espiras em série por fase do estator, tem-se a indutância própria do rotor equivalente, L'_r , dada por:

$$L'_r = L'_{lr} + L_{ms}, \quad (2.5)$$

onde L'_{lr} é a indutância de dispersão referida ao estator dada por

$$L'_{lr} = \left(\frac{N_s}{N_r} \right)^2 l_{lr} \quad (2.6)$$

2.3.1 Indutâncias mútuas entre as fases do mesmo enrolamento

O fluxo magnético concatenado com as fases bs e cs , quando apenas a fase as é excitada, define as indutâncias mútuas $L_{bs,as}$ e $L_{cs,as}$, respectivamente. Os fluxos magnéticos concatenados com as fases bs e cs são a projeção do fluxo magnético da fase as na direção dos eixos magnéticos bs e cs , isto é, é o fluxo magnético da fase as , que atravessa o entreferro, multiplicado por $\cos(\frac{2\pi}{3})$ (BIM, 2014). De forma análoga, chega-se à mesma conclusão a respeito das outras indutâncias mútuas entre as fases do estator. O mesmo raciocínio aplica-se para as mútuas entre as fases do rotor. Portanto, as indutâncias entre quaisquer fases do estator, M_s , e rotor, M_r , são dadas, respectivamente, por:

$$M_s = -\frac{L_{ms}}{2}, \quad M_r = -\frac{L_{mr}}{2} = -\left(\frac{N_r}{N_s} \right)^2 \frac{L_{ms}}{2} \quad (2.7)$$

O sinal negativo vem do fato de os eixos estarem defasados de um ângulo maior que $\frac{\pi}{2}$ radianos elétricos, o que torna negativo o fluxo magnético concatenado com um enrolamento, quando uma corrente positiva está presente no outro enrolamento.

2.3.2 Indutâncias mútuas entre estator e rotor

Na determinação das indutâncias mútuas entre as fases do estator e as fases do rotor, deve-se considerar a defasagem entre os eixos magnéticos envolvidos, tal como foi feito para se descobrir as indutâncias mútuas entre as fases do mesmo enrolamento. Desta forma, têm-se:

$$\begin{aligned} M_{as,ar} = M_{bs,br} = M_{cs,cr} &= \frac{N_r}{N_s} L_{ms} \cos(\theta_r) \quad , \\ M_{as,br} = M_{bs,cr} = M_{cs,ar} &= \frac{N_r}{N_s} L_{ms} \cos\left(\theta_r + \frac{2\pi}{3}\right) \quad \text{e} \\ M_{as,cr} = M_{bs,ar} = M_{cs,br} &= \frac{N_r}{N_s} L_{ms} \cos\left(\theta_r - \frac{2\pi}{3}\right), \end{aligned} \quad (2.8)$$

onde θ_r é a posição angular do rotor.

2.4 Projeto do servoposicionador utilizando campo orientado indireto

Como o objetivo do trabalho é realizar o controle de uma máquina de indução trifásica utilizado como servoposicionador, é necessário entender a aplicação do controle de campo orientado indireto à máquina de indução trifásica, através da modelagem dinâmica do campo orientado.

O campo orientado indireto faz uso do fato de que satisfazer a relação entre escorregamento e a corrente do estator é condição necessária e suficiente para produzir orientação de campo (LIPO; NOVOTNY, 1996).

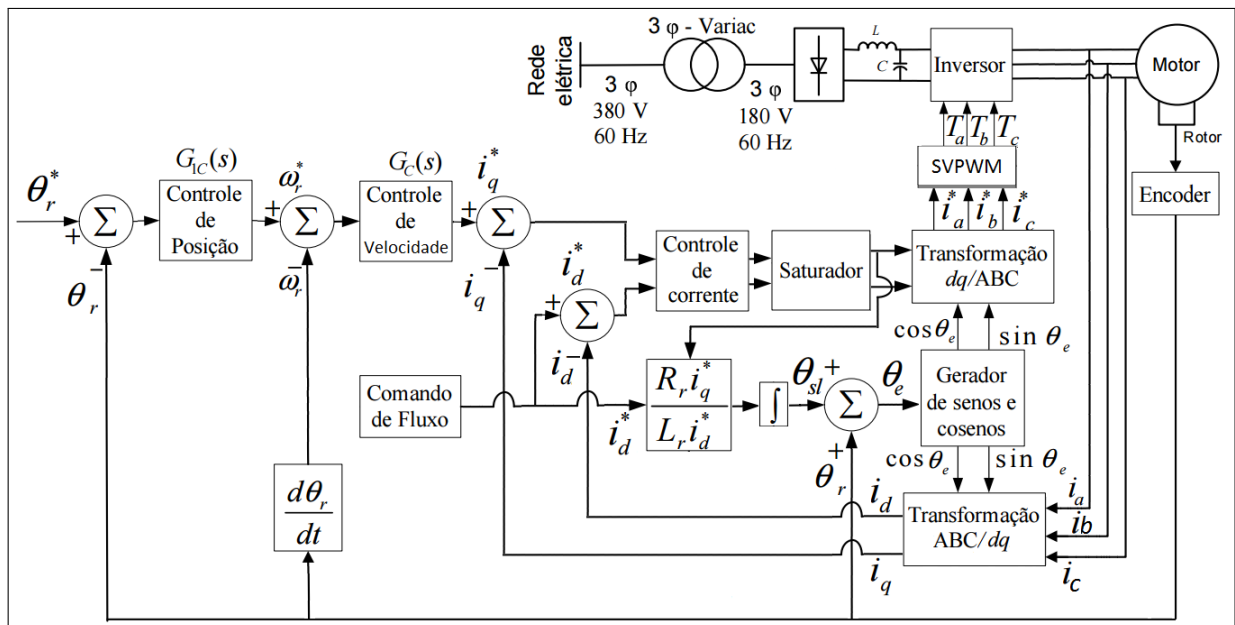
2.4.1 Modelagem dinâmica do campo orientado indireto para uma máquina de indução trifásica

O diagrama de blocos de um servoposicionador utilizando controle de campo orientado indireto e um motor de indução trifásico é mostrado na Figura 3.

O bloco *Controle de posição*, $G_{lc}(s)$, é alimentado pelo comando de posição do rotor, θ_r^* , e este envia o comando de velocidade do rotor, ω_r^* . Em seguida, tem-se o bloco *Controle de velocidade*, $G_c(s)$, o qual envia o comando da corrente de quadratura, i_q^* . Para o cálculo da corrente de comando de eixo direto, i_d^* , tem-se a realimentação da velocidade do rotor, ω_r , proveniente do bloco de filtro digital, $\frac{d\theta_r}{dt}$, que utiliza da posição do rotor, θ_r , dada pelo *encoder* fixado no eixo do motor, sendo, então, calculado no bloco *Comando de Fluxo*. Os comandos das correntes de quadratura e de eixo direto vão para o bloco *Controlador de corrente* que, após

passar pelo bloco *Saturador*, obtém-se as referências de corrente para o bloco *Transformação dq/ABC*, que, por sua vez, fornece ao bloco *SVPWM* os comandos de corrente i_a^* , i_b^* e i_c^* . O bloco *SVPWM* é responsável pelo cálculo dos períodos dos gatilhos das chaves T_a , T_b e T_c enviados ao bloco *Inversor* que, por fim, fornece as correntes necessárias para acionar o motor de indução. A tensão do barramento CC do bloco *Inversor* é fornecida por um retificador trifásico ligado a um *Variac* conectado à rede trifásica da concessionária de energia. Para se obter a posição do campo girante, soma-se a posição do rotor, θ_r , com a posição de escorregamento, θ_{sl} , a fim de alimentar o bloco *Gerador de senos e cosenos*, que por sua vez alimenta os blocos *Transformação dq/ABC* e *Transformação ABC/dq*. Este último bloco é alimentado pelas correntes i_a , i_b e i_c , através de sensores de corrente, e realiza a transformação para o sistema síncrono a fim de realimentar o controlador de corrente.

Figura 3 – Configuração do acionamento de campo orientado indireto para uma máquina de indução.



Fonte – Adaptado de (LIAW *et al.*, 1993).

A equação de estados de um motor de indução com referência girante síncrona pode ser escrita da seguinte forma (BOSE, 1986):

$$\frac{d}{dt}[A] = [B][C] + \frac{1}{\sigma L_s}[D] \quad (2.9)$$

Onde:

$$A = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix}, \quad C = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix}, \quad D = \begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix} \quad e \quad (2.10)$$

$$B = \begin{bmatrix} -\frac{R_s}{\sigma L_s} - \frac{R_r(1-\sigma)}{\sigma L_r} & \omega_e & \frac{L_m R_r}{\sigma L_s L_r^2} & \frac{P \omega_r L_m}{2 \sigma L_s L_r^2} \\ \omega_2 & -\frac{R_s}{\sigma L_s} - \frac{R_r(1-\sigma)}{\sigma L_r} & -\frac{P \omega_r L_m}{2 \sigma L_s L_r^2} & \frac{L_m R_r}{\sigma L_s L_r^2} \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & \omega_e - \frac{P}{2} \omega_r \\ 0 & \frac{L_m R_r}{L_r} & -(\omega_e - \frac{P}{2} \omega_r) & -\frac{R_r}{L_r} \end{bmatrix} \quad (2.11)$$

Sendo:

$$\sigma = 1 - \frac{L_m^2}{L_s L_r}, \quad \lambda_{qr} = L_m i_{qs} + L_r i_{dr} \quad e \quad \lambda_{dr} = L_m i_{ds} + L_r i_{qr} \quad (2.12)$$

A equação do conjugado é dada por:

$$T_e = \frac{3P L_m}{4 L_r} \left(i_{qs} \lambda_{dr} - i_{ds} \lambda_{qr} \right) \quad (2.13)$$

Em um campo orientado ideal de um motor de indução, ocorre desacoplamento entre os eixos direto e em quadratura, e o fluxo rotórico de dispersão é alinhado ao eixo direto (DE SOUZA JÚNIOR, 2010). Assim, o fluxo de dispersão e sua derivada no eixo em quadratura são nulos, ou seja:

$$\lambda_{qr} = 0 \quad e \quad \frac{d\lambda_{qr}}{dt} = 0 \quad (2.14)$$

O fluxo rotórico de dispersão pode ser calculado através da terceira linha da matriz da equação (2.9). Utilizando ainda a equação (2.14):

$$\lambda_{dr} = \frac{L_m i_{ds}}{1 + s \frac{L_r}{R_r}} \quad (2.15)$$

Fazendo a constante de tempo elétrica do sistema desprezível com relação à constante mecânica, a constante de tempo da equação (2.15) torna-se próxima a zero e a corrente i_{ds} se torna constante ($i_{ds} = i_{ds}^*$) de modo a se ter um fluxo rotórico desejado constante. Assim sendo, tem-se:

$$\lambda_{dr} = L_m i_{ds}^* \quad (2.16)$$

Utilizando as equações (2.14) e (2.16), a equação de conjugado, dada por (2.13), pode ser expressa por:

$$T_e^* = \frac{3P L_m^2}{4 L_r} i_{qs}^* i_{ds}^* \quad (2.17)$$

Sendo que i_{qs}^* denota o comando de conjugado controlado pela corrente do estator no eixo em quadratura, sendo esta controlada por $G_c(s)$, mostrado na Figura 3 (DE SOUZA JÚNIOR, 2010). No método do campo orientado indireto, a frequência precisa ser calculada em coordenadas $dq0$. Utilizando a quarta linha da equação (2.9) em conjunto com a equação (2.14), a frequência de escorregamento pode ser calculada por:

$$\omega_{sl} = \frac{L_m R_r i_{qs}^*}{L_r \lambda_{dr}} = \frac{R_r i_{qs}^*}{L_r i_{ds}^*} \quad (2.18)$$

O conjugado, T_e , a velocidade rotórica, ω_r , e a posição angular, θ_r , são relacionados por:

$$\omega_r = s\theta_r = \frac{1}{s + \frac{B}{J}} \left[T_e(s) - T_L(s) \right] \quad (2.19)$$

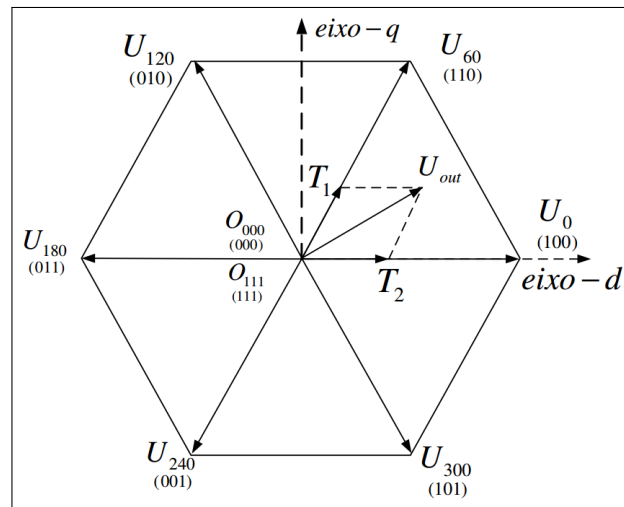
Sendo B o coeficiente de atrito viscoso e J a constante de momento de inércia.

2.4.2 Modulação por largura de pulso usando vetores espaciais (SVPWM)

Conforme apresentado na Figura 3, o chaveamento do bloco *Inversor* é realizado pelo bloco *SVPWM*, através da modulação por largura de pulso utilizando espaço vetorial, *SVPWM* (*Space Vector Pulse Width Modulation*). Este método se tornou uma técnica muito popular de modulação por largura de pulso (PWM) para aplicações em conversores CC-CA trifásicos, como o controle de máquinas de indução e motores trifásicos de magnetos permanentes (YU, 2001). O processador digital utilizado neste trabalho, modelo TMS320F228335[®] da *Texas Instrument*[®], possui suporte nativo às funções de modulação em espaço vetorial, facilitando, assim, o acionamento do inversor trifásico utilizado. A principal vantagem de se utilizar a técnica de acionamento por SVPWM, se deve ao fato da mínima geração de distorção harmônica nos enrolamentos da máquina de indução trifásica (DE SOUZA JÚNIOR, 2010).

O objetivo do SVPWM é aproximar a tensão de referência U_{out} instantaneamente através de uma combinação de chaveamentos mapeados correspondentes aos vetores de base do espaço vetorial, de modo que cada vetor corresponda a um padrão de chaveamento, conforme ilustrado na Figura 4 (YU, 2001).

Figura 4 – Diagrama do espaço vetorial.



Fonte – Adaptado de (YU, 2001).

Pode-se notar que o vetor U_{out} é uma combinação linear dos vetores espaciais base, U_x e U_{x+60} (ou U_{x-60}), que formam o setor que contém o vetor U_{out} . Os períodos T_1 e T_2 correspondem, respectivamente, à duração do chaveamento no estado correspondente aos vetores base. Desta forma, tem-se a expressão abaixo:

$$U_{out}(nT) = \frac{1}{T}(T_1U_x + T_2U_{x+60}) \quad (2.20)$$

Ou seja, para cada período do PWM, U_{out} pode ser conseguido de modo aproximado, variando os estados das chaves entre os vetores U_x e U_{x+60} (ou U_{x-60}) por períodos de duração T_1 e T_2 , respectivamente. Pelo fato de que a soma de T_1 e T_2 deve ser menor ou igual ao período total T_{pwm} , o inversor deve permanecer o período remanescente nos estados O_{000} ou O_{111} . A escolha destes vetores deve satisfazer o menor esforço de chaveamento do inversor (YU, 2001). Portanto:

$$T_{pwm}U_{out} = T_1U_x + T_2U_{x+60} + T_0(O_{000,111}), \quad (2.21)$$

onde $T_0 = T_{pwm} - T_1 - T_2$.

Portanto, de posse dos comandos de tensão, basta verificar o setor em que a resultante da tensão estática se encontra, a partir da Figura 4. Assim sendo, resolve-se a equação (2.21) para T_1 e T_2 e configura-se os temporizadores no processador digital com os valores encontrados, atribuindo os valores aos respectivos registradores do processador.

2.5 Considerações finais

Neste capítulo foram apresentadas as transformadas de Clarke-Park utilizadas para converter o modelo de uma máquina trifásica em bifásico no sistema síncrono. Em seguida foram apresentadas as indutâncias presentes na máquina de indução, tais como próprias e mútuas. Finalmente, foi apresentado o controle de campo orientado indireto e o projeto da máquina de indução como servoposicionador.

No próximo capítulo serão apresentados os controladores aplicados ao sistema de acionamento, assim como os métodos de sintonia utilizados neste trabalho.

3 CONTROLADORES APLICADOS À MÁQUINA DE INDUÇÃO

Pela simplicidade, familiaridade de implementação por operadores e desempenho, a maioria dos processos industriais apresentam os controladores Proporcional-Integral-Derivativo, (PID), em suas malhas de controle. Para malhas de controle individuais, em cascata ou de múltiplas entradas e saídas, o módulo de controle PID representa uma solução aceitável para os problemas de controle do mundo real, em relação ao seguimento de referência e rejeição de perturbação.

Esta tecnologia de controle tem sido usada durante várias décadas, sobrevivendo aos avanços tecnológicos desde a era analógica até a era digital de controle por computador. Atualmente o controlador PID pode ser usado como rotina padrão de um sistema de controle supervisão ou como um módulo de *hardware* dentro de um sistema eletrônico programável, podendo ser usado para a construção de sistemas de controle complexos.

A aplicabilidade dos controladores PID tem motivado pesquisadores e engenheiros na busca de soluções que permitam obter o melhor desempenho deste tipo de controlador. Neste capítulo apresenta-se a configuração utilizada neste trabalho, o diagrama de blocos, a lei de controle digital, a sua forma na estrutura RST e as suas características.

3.1 Controlador PID

Segundo (OGATA, 2010), mais da metade dos controladores utilizados na indústria atualmente são do tipo PID ou alguma versão modificada deste controlador. A sua grande popularidade se deve a sua simplicidade de implementação e a capacidade de satisfazer a maioria dos requisitos de projeto.

Como o nome sugere, o algoritmo do controlador PID é composto por três coeficientes: proporcional, integral e derivativo, que são ajustados para se obter a resposta dentro dos requisitos do projeto. Cada parâmetro dá uma característica à resposta do sistema, como visto abaixo:

- Proporcional:

A componente proporcional depende apenas da diferença entre o ponto de referência, $r(t)$, e a variável de processo, $y(t)$, que está sendo medida. Esta diferença é o sinal de erro, $e(t)$. O ganho proporcional K_p determina a taxa de resposta de saída para o sinal de erro. Em geral, o aumento do ganho proporcional irá aumentar a velocidade da resposta do sistema

de controle. No entanto, se o ganho proporcional é muito grande, a variável de processo poderá oscilar.

- Integral:

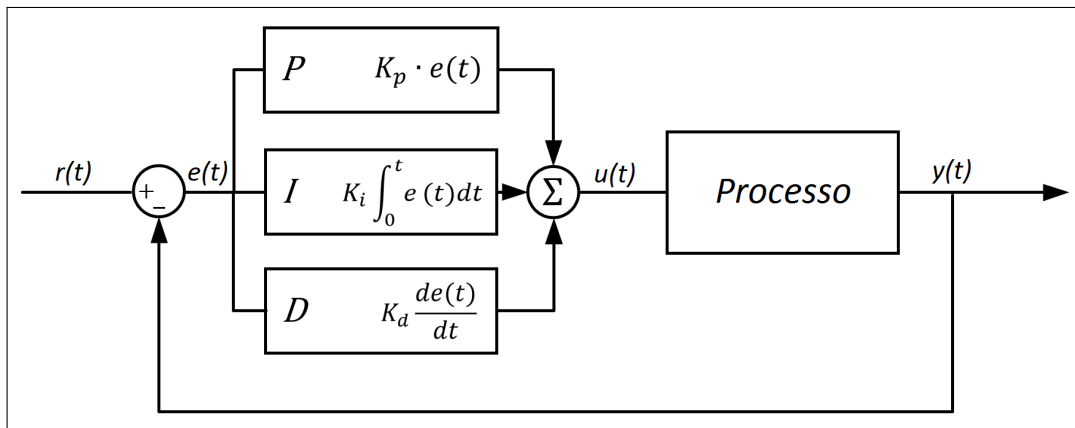
A componente integral soma o termo de erro, $e(t)$, ao longo do tempo. O resultado é que mesmo um pequeno erro fará com que a componente integral aumente lentamente. A resposta integral irá aumentar ao longo do tempo a menos que o erro seja zero, portanto, o efeito é o de conduzir o erro de regime para zero. Fazendo $T_d = 0$ na estrutura do PID, tem-se o controlador PI, onde se tem a rapidez da resposta proporcional, citada acima, e a capacidade de eliminar o erro de regime, devido ao integrador.

- Derivativo:

A componente derivada faz com que a saída diminua se a variável de processo, $y(t)$, está aumentando rapidamente. A derivada da resposta é proporcional à taxa de variação da variável de processo. Aumentar o parâmetro do tempo derivativo T_d fará com que o sistema de controle reaja mais fortemente a mudanças no parâmetro de erro aumentando a velocidade da resposta global de controle do sistema. Na prática, a maioria dos sistemas de controle utilizam o tempo derivativo T_d muito pequeno ou nulo, pois a derivada de resposta é muito sensível ao ruído no sinal da variável de processo. Se o sinal de *feedback* do sensor é ruidoso ou se a taxa de malha de controle é muito lenta, a derivada de resposta pode tornar o sistema de controle instável.

A Figura 5 apresenta o diagrama de blocos do controlador PID, onde tem-se o valor de referência, $r(t)$, o erro, $e(t)$, dado pela diferença entre o valor de referência e o sinal de saída do processo, $y(t)$. Os blocos P, I e D representam, respectivamente, a parcela proporcional, integral e derivativa do controlador, onde K_p é o ganho proporcional, K_i o integral e K_d o derivativo. O sinal de controle, $u(t)$, aplicado ao processo é o resultado da ação de todas as parcelas.

Figura 5 – Diagrama de blocos do controlador PID.



Fonte – O autor.

Através da Figura 5 tem-se a equação geral do controlador PID, dada por:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right), \quad (3.1)$$

onde T_i e T_d são os tempos integral e derivativo, respectivamente, dados por $T_i = \frac{K_p}{K_i}$ e $T_d = \frac{K_d}{K_p}$. Agindo sobre os parâmetros proporcional, integral e derivativo, consegue-se modificar as propriedades dinâmicas do controlador.

Pode-se utilizar o controlador PID por completo, com todos os termos presentes, ou somente um ou dois coeficientes. As formas mais comuns são listadas abaixo, com uma breve descrição.

- Controlador Proporcional (P):
Fornece uma relação linear (ganho K_p) entre o sinal de entrada, $e(t)$, e a saída do controlador, $u(t)$.
- Controlador Proporcional-Integral (PI):
Ação proporcional associada a uma ação integral que faz a saída do controlador aumentar em uma taxa proporcional a integral do erro $e(t)$.
- Controlador Proporcional-Derivativo (PD):
Ação proporcional associada a ação derivativa na qual a saída do controlador é diretamente proporcional à taxa de variação da sua entrada. Ou seja, com base na tendência do erro, a ação derivativa se antecipa na ação de controle.
- Controlador Proporcional-Integral-Derivativo (PID):
Incorpora as três ações descritas anteriormente.

Para este trabalho, foi utilizado o controlador do tipo PI para a malha de corrente e de velocidade. Já para a malha de posição, foi utilizado somente um controlador Proporcional, devido ao fato de que a leitura atual de velocidade é realizada através da derivada da posição fornecida pelo *encoder*, gerando um termo integral para a malha de posição. Posteriormente, será apresentado o controlador na estrutura RST, que também será utilizado na malha de velocidade, com o objetivo de comparação de respostas entre os dois tipos de controlador.

O desafio dos controladores de ganho fixo é definir os valores dos respectivos coeficientes, no caso do PID, os coeficientes proporcional, integral e derivativo, a fim de se obter a resposta mais adequada aos requisitos de projeto. Existem diversos métodos para realizar esta sintonia, porém, a técnica apresentada por (ZIEGLER; NICHOLS, 1942) constitui um marco nas aplicações de controladores do tipo PID em processos industriais.

3.1.1 Métodos de sintonia do controlador PID

Para se ter um bom desempenho do controlador PID, torna-se importante o correto ajuste dos parâmetros do controlador, ou seja, os ganhos K_p , K_i e K_d . Existem diversos métodos bastante difundidos, porém, neste trabalho será apresentado o método proposto por (ZIEGLER; NICHOLS, 1942), devido sua simplicidade de uso e boa aplicabilidade.

3.1.1.1 Método de Ziegler-Nichols

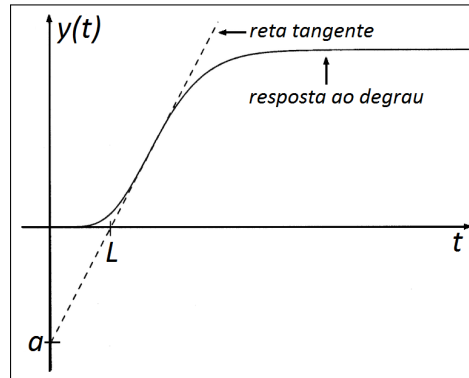
Dois métodos clássicos para a determinação dos parâmetros do controlador PID foram apresentados em (ZIEGLER; NICHOLS, 1942). Estes métodos ainda são amplamente utilizados na indústria, quer na sua forma original ou com alguma modificação. As técnicas são baseadas na determinação de algumas características da dinâmica do processo. Os parâmetros do controlador são expressos em termos das características por simples fórmulas. A sintonia dos parâmetros é feita utilizando-se a resposta ao degrau e a resposta em frequência do sistema. Para este trabalho, apresenta-se o primeiro método.

3.1.1.1.1 Método da resposta ao degrau

Este método de sintonia apresentado por (ZIEGLER; NICHOLS, 1942) é baseado na resposta ao degrau em malha aberta do sistema, que é caracterizado por dois parâmetros. Os parâmetros são determinados a partir da resposta ao degrau unitário do processo, como mostrado

na Figura 6.

Figura 6 – Caracterização da resposta ao degrau no método de resposta ao degrau de (ZIEGLER; NICHOLS, 1942).



Fonte – O autor.

Primeiramente, determina-se o ponto em que a inclinação da resposta ao degrau tem seu valor máximo. Traça-se uma reta tangente a este ponto, conforme indicado pela linha tracejada na Figura 6. A interseção entre a reta tangente e os eixos de coordenadas fornece os parâmetros a e L . Onde a é dado pelo valor que a reta tangente intercepta o eixo das ordenadas e L é dado pelo eixo das abscissas, onde a reta tangente o cruza. Os coeficientes do controlador são funções diretas destes parâmetros. A Tabela 2 apresenta os valores dos coeficientes para os tipos de controladores PID.

Tabela 2 – Parâmetros do controlador PID obtidos a partir do método de resposta ao degrau de (ZIEGLER; NICHOLS, 1942).

Controlador	K	T_i	T_d
P	$1/a$		
PI	$0,9/a$	$3L$	
PID	$1,2/a$	$2L$	$L/2$

Fonte – (ASTRÖM; HÄGGLUND, 1995)

3.1.2 PID discreto

Com o objetivo de se implementar o controlador PID, utilizou-se sua forma discreta obtida a partir da equação (3.1). Esta equação pode ser discretizada para obtenção da equação à

diferença. Para uma aproximação retangular, obtém-se:

$$u(k) = K_p \left\{ e(k) + \frac{T_s}{T_i} \sum_{i=1}^k e(i-1) + \frac{T_d}{T_s} \left[e(k) - e(k-1) \right] \right\}, \quad (3.2)$$

onde T_s é o período de amostragem.

A Equação (3.2) determina o algoritmo de controle digital do tipo PID recursivo, pois, para determinar $u(k)$, os valores passados de $e(k)$ e $u(k)$ devem ser conhecidos. Para a implementação em controladores digitais, a forma recursiva é a mais adequada (DE SOUZA JÚNIOR, 2010). Isto implica que o cálculo do controle em um instante $u(k)$ depende do valor anterior, $u(k-1)$, e dos valores de erro atual, $e(k)$, erro anterior, $e(k-1)$, e o sinal de erro anterior ao anterior, $e(k-2)$. Assim, tem-se:

$$u(k-1) = K_p \left\{ e(k-1) + \frac{T_s}{T_i} \sum_{i=1}^{k-1} e(i-1) + \frac{T_d}{T_s} \left[e(k-1) - e(k-2) \right] \right\} \quad (3.3)$$

Subtraindo a equação (3.2) de (3.3), tem-se:

$$u(k) - u(k-1) = \left(K_p + K_p \frac{T_d}{T_s} \right) e(k) + \left(K_p \frac{T_s}{T_i} - 2K_p \frac{T_d}{T_s} - K_p \right) e(k-1) + K_p \frac{T_d}{T_s} e(k-2) \quad (3.4)$$

Ou seja:

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (3.5)$$

Sendo:

$$\begin{aligned} q_0 &= K_p \left(1 + \frac{T_d}{T_s} \right), \\ q_1 &= -K_p \left(1 + 2 \frac{T_d}{T_s} - \frac{T_s}{T_i} \right) \quad \text{e} \\ q_2 &= K_p \frac{T_d}{T_s} \end{aligned} \quad (3.6)$$

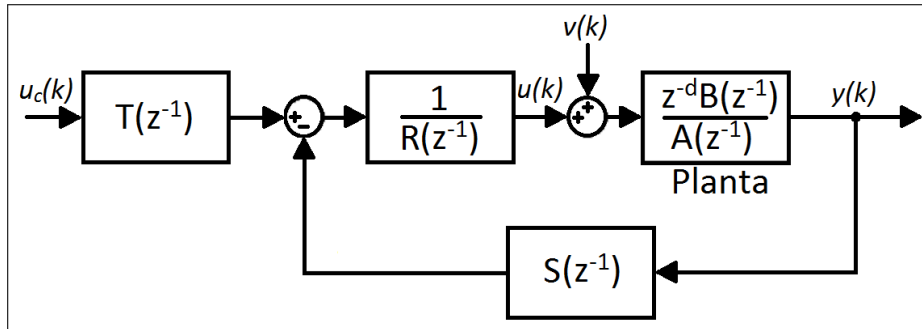
3.2 Controlador RST

Para efeitos de estudo da resposta do controlador PI na malha de velocidade da planta, foi também implementado um controlador de estrutura denominada RST a fim de se verificar qual melhor se adequa às características dinâmicas da planta. Nesta seção será realizada uma abordagem acerca desse tipo de estrutura, que leva este nome devido aos polinômios que o representam: R , S e T .

A Figura 7 ilustra a estrutura de um controlador do tipo RST. Como pode-se observar, ao contrário do que ocorre em um controlador PID, como mostrado na Figura 5, em que a entrada

do controlador é o erro entre a variável controlada e a referência, o controlador RST permite que tanto o sinal da variável controlada, $y(k)$, quanto o sinal de referência, $u_c(k)$, sejam tratados de forma independente.

Figura 7 – Estrutura do controlador RST.



Fonte – Adaptado de (CAMPOS, 2007).

A principal vantagem deste tipo de estrutura de controle se dá pelo fato de que as rejeições a distúrbios, tal como $v(k)$, e as variações nos parâmetros são tratadas de forma separada do acompanhamento das mudanças na referência, $u_c(k)$, uma vez que o polinômio da realimentação $S(z^{-1})$ é diferente de $T(z^{-1})$ (ASTRÖM; WITTERNMARK, 1990). Ou seja, diz-se que o controlador RST tem dois graus de liberdade devido ao fato de que os objetivos de regulação são assegurados pela parte $R-S$ do controlador, e o seguimento de referência é definido através do polinômio T (LANDAU *et al.*, 2011).

A equação geral de um sistema em malha fechada com um controlador RST é dada por:

$$R(z^{-1})u(k) = T(z^{-1})u_c(k) - S(z^{-1})y(k), \quad (3.7)$$

onde $y(k)$ é a variável controlada, $u_c(k)$ é a entrada de referência e $u(k)$ é o sinal de controle.

A equação (3.7) pode ser reescrita como:

$$u(k) = \frac{T(z^{-1})}{R(z^{-1})}u_c(k) - \frac{S(z^{-1})}{R(z^{-1})}y(k) \quad (3.8)$$

Onde os polinômios $R(z^{-1})$, $S(z^{-1})$ e $T(z^{-1})$ são da forma:

$$\begin{aligned} R(z^{-1}) &= r_0 + r_1z^{-1} + \dots + r_nz^{-n_r}, \\ S(z^{-1}) &= s_0 + s_1z^{-1} + \dots + s_nz^{-n_s} \quad \text{e} \\ T(z^{-1}) &= t_0 + t_1z^{-1} + \dots + t_nz^{-n_t} \end{aligned} \quad (3.9)$$

Com este tipo de controlador, é possível, antecipadamente, definir parte dos polinômios que contém o controlador, a fim de se obter a resposta em malha fechada de acordo com os requisitos de projeto. Uma das formas de sintonia desse controlador se dá através da técnica de alocação de polos, conforme seção a seguir.

3.2.1 Alocação de Polos

Um simples método para projeto do controlador é a técnica de alocação de polos. O objetivo é determinar um controlador que forneça polos de malha fechada desejados. Além disso, é necessário que o sistema siga sinais de referência de uma maneira especificada.

Assumindo-se que o processo seja SISO (*Single-input, single-output*) e representando-se o sistema na forma discreta, tem-se:

$$A(q)y(k) = B(q)[u(k) + v(k)] \quad (3.10)$$

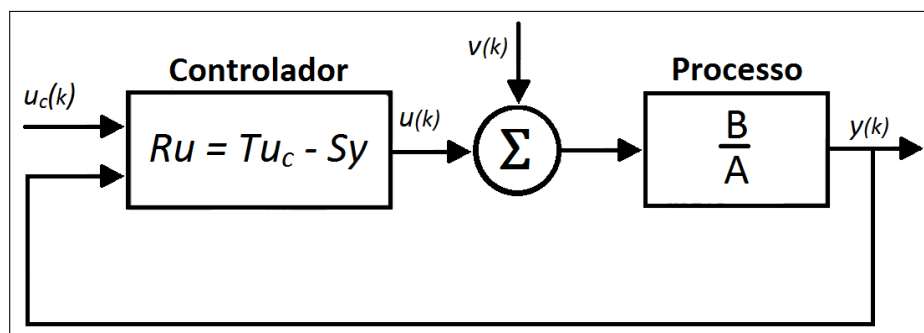
Sendo q^{-1} o operador de atraso unitário, de forma $y(k)q^{-1} = y(k-1)$, $v(k)$ o distúrbio e $A(q)$ e $B(q)$ os polinômios definidos como:

$$\begin{aligned} A(q) &= 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad \text{e} \\ B(q) &= b_0 + b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \end{aligned} \quad (3.11)$$

Onde n_a e n_b são, respectivamente, os graus dos polinômios A e B .

O controlador geral dado pela Equação (3.7) representa um *feedback* negativo com o operador $-\frac{S}{R}$ e um *feedforward* com o operador $\frac{T}{R}$. Assim sendo, tem-se dois graus de liberdade no controlador. Um diagrama de blocos de malha fechada é mostrado na Figura 8.

Figura 8 – Diagrama de blocos do controlador RST.



Fonte – (ASTRÖM; WITTERMARK, 1994).

Eliminando-se $u(k)$ entre as equações (3.7) e (3.10), tem-se as seguintes equações

em malha fechada:

$$y(k) = \frac{BT}{AR+BS}u_c(k) + \frac{BR}{AR+BS}v(k) \quad (3.12)$$

$$u(k) = \frac{AT}{AR+BS}u_c(k) + \frac{BS}{AR+BS}v(k) \quad (3.13)$$

Desta forma, o polinômio característico é então:

$$AR + BS = A_c \quad (3.14)$$

A ideia principal deste método de projeto é especificar um polinômio característico de malha fechada desejado A_c . Os polinômios R e S podem ser resolvidos através da equação (3.14). Pode-se notar que, no procedimento de projeto, considera-se o polinômio A_c como um parâmetro de projeto que é escolhido para dar as propriedades desejadas ao sistema em malha fechada (ASTRÖM; WITTERNMARK, 1994). Ou seja, a ideia básica é estimar os parâmetros da planta, a partir das medidas de entrada e saída, e utilizar a técnica de alocação de polos para obter o controlador em função do desempenho desejado de malha fechada.

O procedimento de projeto do controle por alocação de polos pode ser estabelecido pelo cálculo de $R(q^{-1})$ e $S(q^{-1})$ através da equação (3.14). Reescrevendo esta equação como

$$A(q^{-1})R(q^{-1}) + B(q^{-1})S(q^{-1}) = P_{MF}(q^{-1}) \quad (3.15)$$

O sistema em malha fechada, a partir da equação (3.12), sem distúrbio pode ser representado como

$$y(k) = \frac{B(q^{-1})T(q^{-1})}{P_{MF}(q^{-1})}u_c(k) \quad (3.16)$$

Assim, para obter-se uma resposta com erro nulo em regime para uma referência na forma degrau, o polinômio $T(q^{-1})$ deve ser relacionado como

$$T(1) = \frac{P_{MF}(1)}{B(1)} \quad (3.17)$$

O procedimento de projeto do controlador por alocação de polos admite que os parâmetros do processo são conhecidos *a priori*, caso contrário algum procedimento de identificação deve ser implementado para estimar o modelo matemático da planta.

Admitindo-se que os polos de malha fechada desejados têm constantes de tempo τ_1 e τ_2 , isto é,

$$P_{MF}(s) = (\tau_1 s + 1)(\tau_2 s + 1) \quad (3.18)$$

então, para um período de amostragem T_s , o polinômio $P_{MF}(z^{-1})$ é calculado por

$$P_{MF}(q^{-1}) = (1 - p_1 q^{-1})(1 - p_2 q^{-1}) \quad (3.19)$$

onde $p_1 = e^{-\frac{T_s}{\tau_1}}$ e $p_2 = e^{-\frac{T_s}{\tau_2}}$.

A seleção do polinômio de malha fechada $P_{MF}(q^{-1})$ determina a estabilidade do sistema, já que os polos devem estar posicionados no interior do círculo unitário no plano complexo z para que o sistema seja estável e também determina o amortecimento e a velocidade da resposta do sistema, pois estas duas propriedades são selecionadas através do posicionamento do polos de malha fechada no interior do círculo unitário.

3.3 Considerações finais

Neste capítulo foi visto o controlador PID, sua forma discreta e o método de sintonia de Ziegler-Nichols utilizado neste trabalho. Em seguida foi apresentado o controlador RST e o método de alocação de polos, onde define-se a resposta em malha fechada desejada para encontrar os parâmetros do controlador.

No próximo capítulo serão apresentados os projetos dos controladores utilizados, os resultados obtidos de simulação das malhas de corrente e velocidade. Para os resultados experimentais será inicialmente apresentada a bancada de testes montada em laboratório para realização dos ensaios e finalmente os resultados práticos dos controladores projetdos.

4 RESULTADOS DE SIMULAÇÃO E EXPERIMENTAL

Neste capítulo é apresentada a estrutura do manipulador robótico desenvolvido em laboratório para se realizar os ensaios de forma prática, assim como o projeto dos controladores e os resultados de simulação e experimentais obtidos.

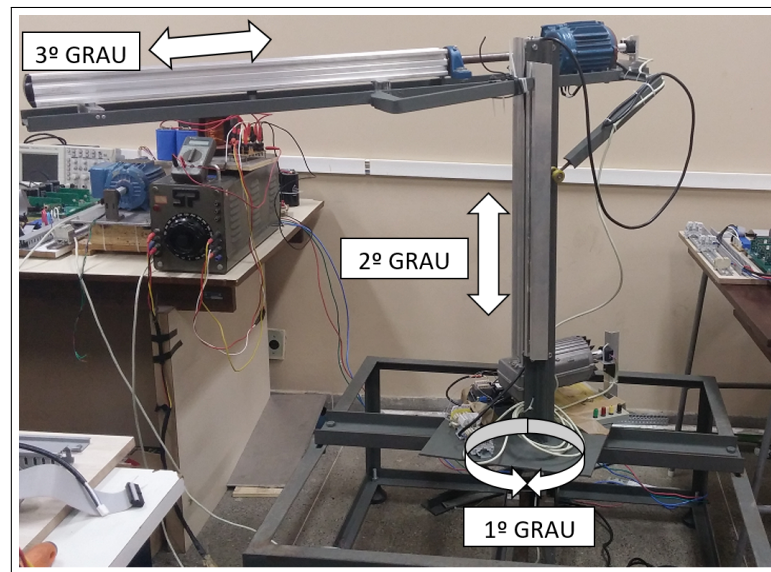
Inicialmente é mostrada a estrutura do manipulador montada em laboratório, destacando os graus de liberdade existentes, a forma de transmissão de movimento e os equipamentos e circuitos utilizados para o acionamento. Posteriormente serão discutidos os controladores utilizados, exibindo a forma de identificação e validação dos modelos obtidos, o projeto dos controladores e os resultados de simulação, através do *software Simulink*[®], e experimentais para cada malha de controle. Por fim serão apresentados os índices de desempenho dos controladores projetados.

4.1 Bancada Experimental

Para a implementação deste trabalho foi necessário realizar o desenvolvimento e construção da estrutura apresentada na Figura 9. Esta estrutura irá comportar um manipulador robótico de cinco graus de liberdade. Porém, como este trabalho é o início do projeto de desenvolvimento do manipulador acionado por motores de indução trifásico, será realizado o controle do primeiro grau de liberdade, referente à base do manipulador, para, em trabalhos futuros, estender para os demais graus. Como pode ser observado, o primeiro grau de liberdade possui um movimento rotacional em torno do eixo principal da estrutura, enquanto o segundo e terceiro possuem movimentos prismáticos, caracterizando-o como do tipo cilíndrico RPP (Rotacional-Prismático-Prismático). Os últimos dois graus dizem respeito a garra, ainda não instalada, que deverá estar conectada ao terceiro grau de liberdade.

A construção desta estrutura traz para o laboratório do Grupo de Pesquisa de Automação e Robótica, (GPAR), do Departamento de Engenharia Elétrica, (DEE), da Universidade Federal do Ceará, (UFC), um conjunto de possibilidades para o estudo da robótica aplicada a manipuladores, haja visto que pode-se realizar diversos projetos de pesquisa utilizando este manipulador desenvolvido.

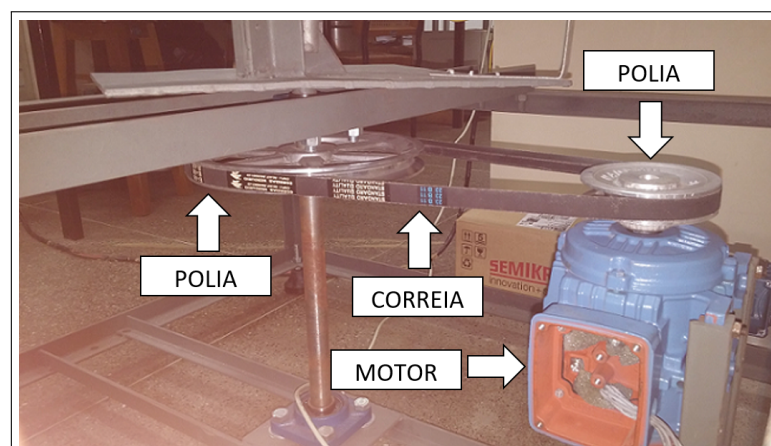
Figura 9 – Estrutura do manipulador montada no GPAR e os graus de liberdade existentes.



Fonte – O autor.

A estrutura é acionada por um motor de indução trifásico, destacado na Figura 10, do tipo gaiola de esquilo, com 0,5cv de potência nominal, tensão nominal de 380/220V, 4 polos e 1,18A de corrente nominal ligado em delta. A potência do motor foi escolhida de modo que fosse possível movimentar a estrutura do manipulador. Inicialmente se instalou um motor de 0,25cv de potência nominal, porém este não conseguiu movimentar a estrutura, fazendo com que fosse necessário um motor de maior potência. A transmissão de movimento para o primeiro grau de liberdade do manipulador se dá através do uso de correia e polias, conforme pode ser visto.

Figura 10 – Acionamento da estrutura através de polias e correia.

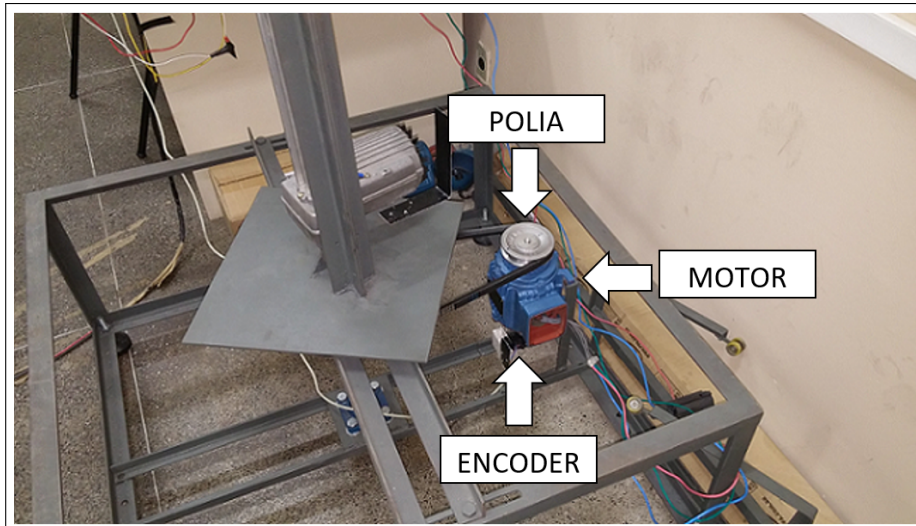


Fonte – O autor.

Para a leitura da posição do eixo do motor, condição necessária para o controle

de campo orientado indireto, conforme apresentado na seção 2.4, foi instalado um *encoder* do tipo incremental da *Avago*[®], modelo HEDS-5500, com resolução de 500 *PPR* (Pulsos por Revolução), na ponta traseira do eixo, conforme apresentado na Figura 11. Maiores detalhes sobre o *encoder* utilizado podem ser visto no Anexo A.

Figura 11 – Instalação do *encoder* na ponta do eixo do motor.



Fonte – O autor.

Para a implementação do controle digital através das técnicas vistas no capítulo 3, aquisição dos sinais de corrente e da posição do eixo do motor e implementação do controle por orientação de campo visto na seção 2.4, foi utilizado o processador digital de sinais, DSP (*Digital Signal Processors*), da *Texas Instrument*[®], modelo TMS320F28335. A principal vantagem da utilização deste modelo se deve ao fato de que, além do alto desempenho, sendo capaz de executar 150 MPIS (Milhões de Instruções por Segundo), o suporte de forma intrínseca à modulação em vetores espaciais, SVPWM, apresentada na seção 2.4.2, sem que seja necessária a implementação desta técnica no dispositivo, demandando um menor esforço de desenvolvimento do algoritmo. Outra vantagem neste modelo se deve ao fato de possuir 12 canais de modulação por largura de pulso, PWM, e dois canais de leitura de *encoders*, podendo, assim, em trabalhos futuros, ser capaz de acionar até dois motores de indução trifásicos simultaneamente, gerando economia ao projeto. Para maiores detalhes sobre o DSP utilizado neste projeto, pode-se consultar o Anexo A.

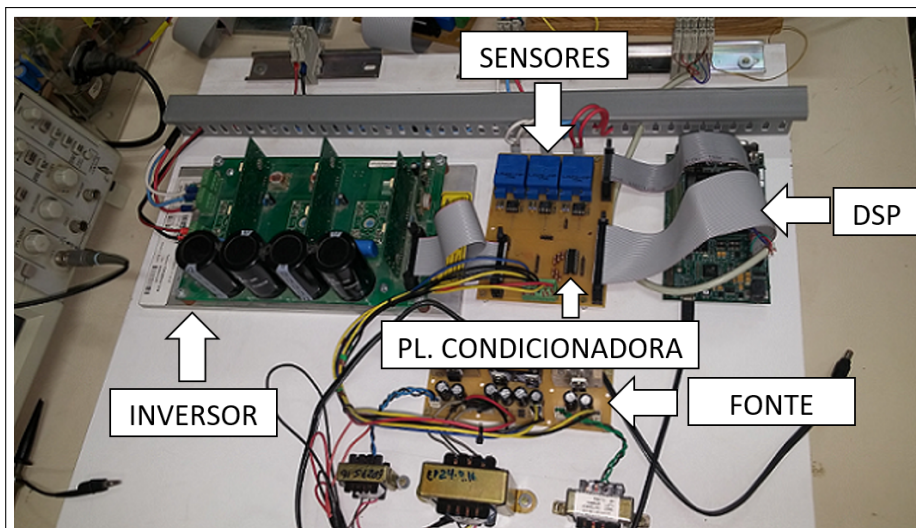
Para o condicionamento de sinais, foi desenvolvida uma placa para comportar os sensores de corrente ligados à amplificadores operacionais de modo a ser possível calibrar os limites de tensão fornecidos pelos sensores ao DSP e realizar a interface através de circuitos integrados com optoacopladores entre os sinais de controle e os sinais de potência. A placa de

circuito impresso desenvolvida pode ser vista com mais detalhes no Anexo A. O método de calibração dos sensores de corrente por ser visto no Anexo E.

Para o acionamento do motor, foi utilizado um inversor trifásico de três níveis da *Semikron*[®], modelo SKS 18G B6111 V12, operando a uma frequência de chaveamento de $10kHz$. Mais características técnicas do equipamento encontram-se no Anexo A.

Para a alimentação dos circuitos e placas, foi desenvolvido uma fonte de tensão auxiliar, capaz de fornecer 4 níveis de tensão contínua. Sendo $18V$ para chaveamento dos gatilhos do inversor trifásico, $15V$ e $-15V$ para os sensores de corrente e $5V$ para alimentação da placa de condicionamento de sinais. As placas e circuitos utilizados são apresentados na Figura 12.

Figura 12 – Circuitos utilizados para controle e acionamento do manipulador.



Fonte – O autor.

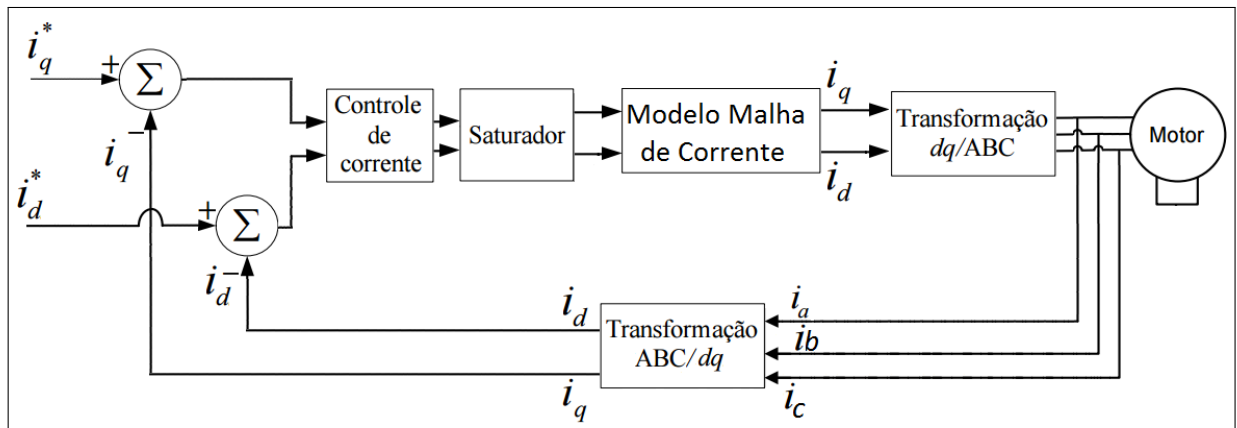
O Anexo A apresenta maiores detalhes acerca dos componentes utilizados no desenvolvimento deste trabalho.

4.2 Controlador da Malha de Corrente

Para o projeto do controlador da malha de corrente, pode-se simplificar a Figura 3, apresentada no Capítulo 2, na Figura 13, a fim de tornar mais claro o entendimento da malha de controle.

Da Figura 13 tem-se que i_q^* e i_d^* são as referências de corrente de eixo de quadratura e direto, respectivamente. O bloco *Controle de Corrente* é composto por um controlador do tipo PI que, após passar pelo bloco *Saturador*, irá atuar na malha elétrica do motor de indução

Figura 13 – Diagrama simplificado da malha de corrente.



Fonte – O autor.

através das correntes de quadratura, i_q , e de eixo direto, i_d , que também são medidas para a realimentação do controlador.

Primeiramente será realizado o ensaio de identificação e validação do modelo da malha de corrente para, posteriormente, projetar o controlador. Por fim, serão apresentados os resultados de simulação e experimentais obtidos.

4.2.1 Identificação do modelo

Realizou-se o ensaio de identificação do modelo da malha de corrente através da aplicação de um sinal do tipo PRBS (*Pseudo-Random Binary Sequence*) na entrada do sistema, i_q^* , e, através dos sensores de corrente, coletou-se as respostas obtidas a um período de amostragem de 0,02s. Vale-se ressaltar que este processo foi realizado de forma experimental, aplicando-se referências de corrente para i_q e coletando-se os resultados através dos sensores apresentados na seção 4.1.

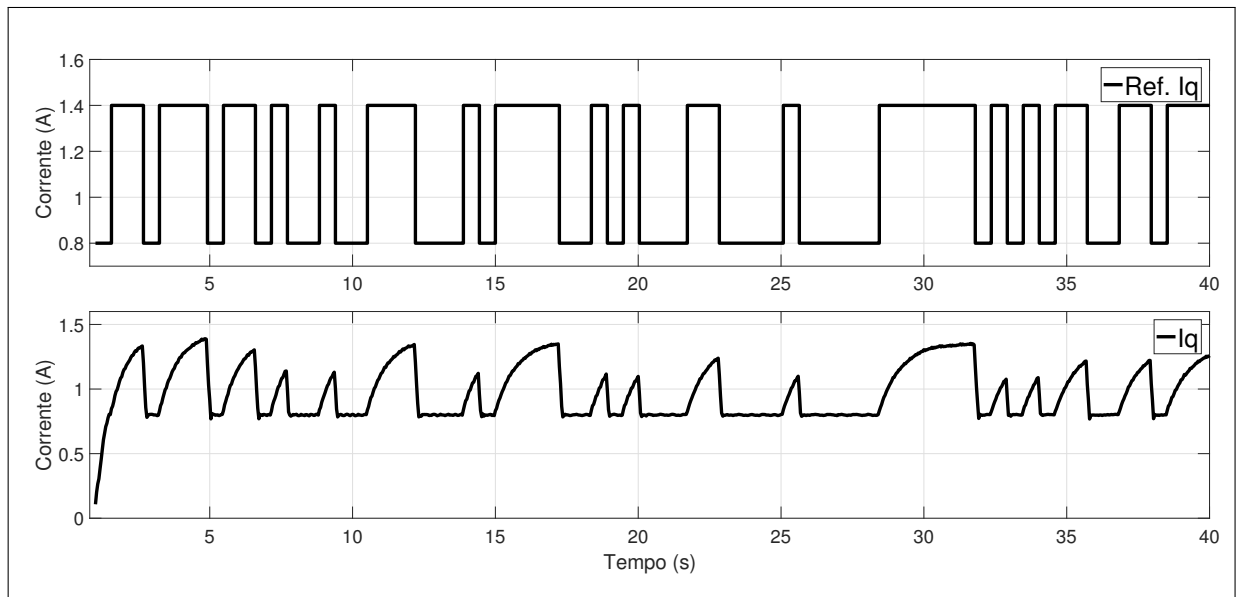
A Figura 14 apresenta na parte superior a referência do tipo PRBS aplicada a i_q^* e, na parte inferior, a resposta obtida, i_q .

Utilizando-se os dados apresentados na Figura 14 para realizar a identificação do modelo da malha de corrente, obtém-se a equação abaixo, onde relaciona a corrente de referência, i_q^* , e a corrente de saída, i_q :

$$G(s) = \frac{0,87299}{0,23234s + 1} \quad (4.1)$$

Realizando-se a discretização do modelo da malha de corrente, chega-se à equação

Figura 14 – Sinais aplicados e obtidos para identificação da malha de corrente.



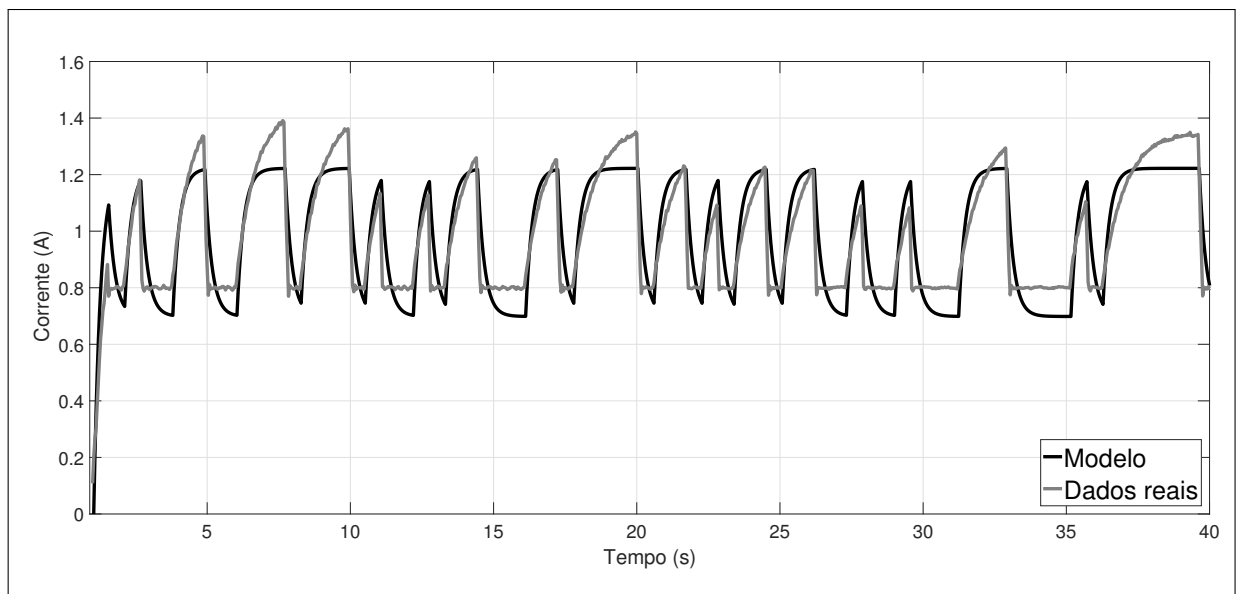
Fonte – O autor.

abaixo:

$$G(z) = \frac{0,072}{z - 0,9175} \quad (4.2)$$

A Figura 15 apresenta a validação do modelo dado pela equação (4.1) em relação à resposta real apresentada pela malha de corrente. O resultado apresentou um erro quadrático médio normalizado de 52,99%.

Figura 15 – Validação do modelo obtido para a malha de corrente.



Fonte – O autor.

Para aplicar o sinal de referência do tipo PRBS para realizar a identificação do modelo da malha de corrente, é necessário escolher adequadamente as variáveis envolvidas, tais como a amplitude do sinal, V , o número de *bits*, n , e o intervalo entre *bits*, T_{bits} . A escolha de V é normalmente limitada pela máxima excursão permitida ao sinal de excitação, sem comprometer seu funcionamento adequado nem, tampouco, levar o sistema a operar em outra faixa linear. Já o número de *bits*, n , determina a periodicidade do sinal gerado. A periodicidade do sinal PRBS não deve ser menor do que o tempo de acomodação do sistema e nem suficientemente longo. O intervalo entre *bits*, T_{bits} , deve ser compatível com a menor constante de tempo de interesse, τ_{min} . Segundo (AGUIRRE, 2007), um valor para T_{bits} que normalmente fornece bons resultados é dado por $\frac{\tau_{min}}{10} \leq T_{bits} \leq \frac{\tau_{min}}{3}$. O código fonte do projeto do sinal PRBS aplicado pode ser observado no Anexo D.

Com o modelo identificado definido pela Equação (4.1) e validado conforme apresentado na Figura 15, pode-se realizar o projeto do controlador para esta malha de controle.

4.2.2 Projeto do controlador de corrente

Para o projeto do controlador da malha de corrente, definiu-se os critérios apresentados na Tabela 3.

Tabela 3 – Critérios de projeto para o controlador da malha de corrente.

Critério	Valor
Tempo de subida (t_r)	0,80s
Tempo de estabilização (t_s)	1,00s
Sobressinal (M_p)	0,00%

Fonte – O autor.

Da Tabela 3 tem-se que o tempo de subida, t_r , é o tempo que a resposta leva até chegar à 90% do valor final, o tempo de estabilização, t_s , é o tempo em que a resposta chega à 95% do valor final e o sobressinal, M_p , é a diferença entre o valor máximo e o valor final da resposta, geralmente medido como percentagem.

Com o modelo dado pela Equação (4.1), projetou-se o controlador do tipo PI aplicando-se a técnica de resposta ao degrau apresentada por Ziegler-Nichols, vista na seção 3.1.1, para realizar a sintonia dos parâmetros do controlador. Assim obteve-se os pontos $a = 0,105$ e $L = 0,117$ e aplicando-se estes valores na Tabela 2, obtém-se os coeficientes do

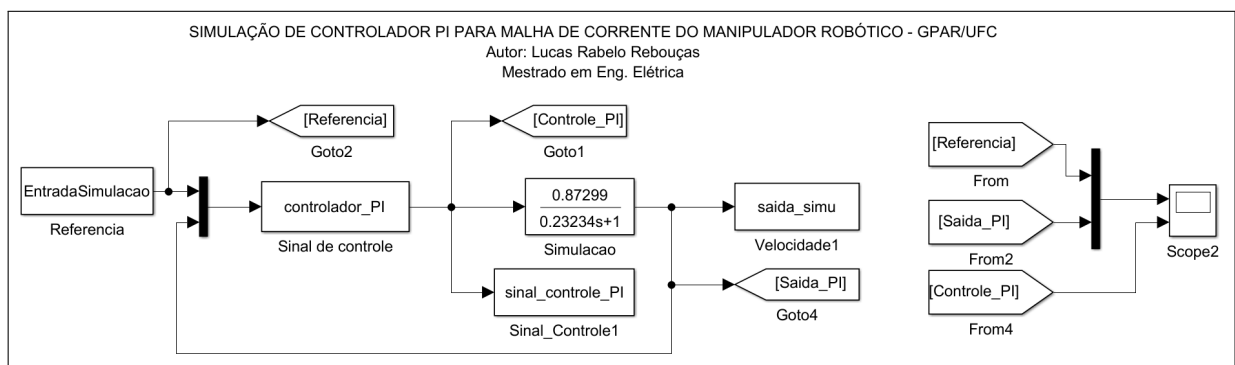
controlador PI dados por $K_p = 8,57$ e $T_i = 0,35ms$. Com estes parâmetros foram realizadas simulações computacionais para se verificar o comportamento do controlador para diversos perfis de referência.

4.2.3 Resultados de simulação do controlador de corrente

Para validar o funcionamento do controlador projetado antes da implementação prática, utilizou-se a ferramenta *Simulink*[®] para realizar a simulação da malha de controle. Foram aplicados dois perfis de referência à corrente em quadratura, i_q^* , sendo um degrau positivo de 1,0A e outro negativo de $-1,0A$. A aplicação de um degrau positivo e negativo tem o objetivo de verificar o comportamento do controlador quando se pretende, com um degrau positivo, acelerar a rotação do motor e, com o degrau negativo, realizar uma frenagem.

A Figura 16 apresenta o diagrama de blocos montado no *Simulink*[®] para realizar a simulação. O bloco *Simulação* representa o modelo identificado da malha de corrente, dado pela Equação (4.1), e no bloco *Controlador PI* é implementado o algoritmo do controlador utilizado, conforme apresentado na seção 3.1.2. Os demais blocos presentes no diagrama são para aquisição dos dados durante a execução da simulação. O algoritmo desenvolvido no bloco *Controlador PI* pode ser visto no Anexo B.

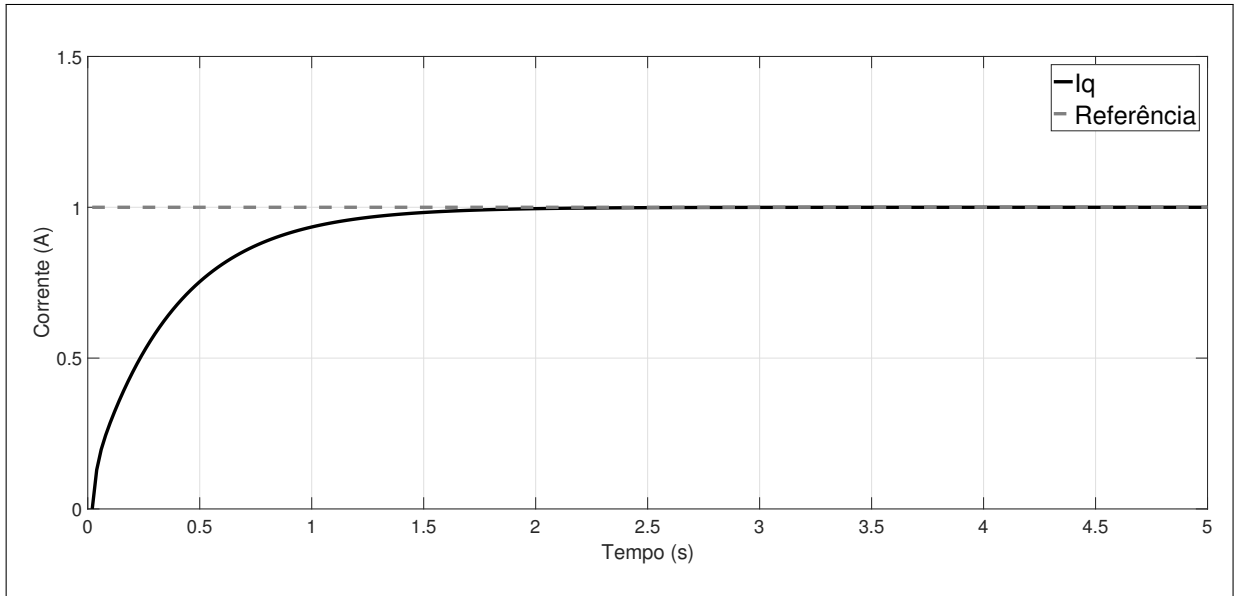
Figura 16 – Diagrama de blocos montado no *Simulink*[®] para simulação da malha de corrente.



Fonte – O autor.

A primeira referência aplicada à simulação foi em $i_q^* = +1,0A$, tendo como resposta o resultado apresentado na Figura 17.

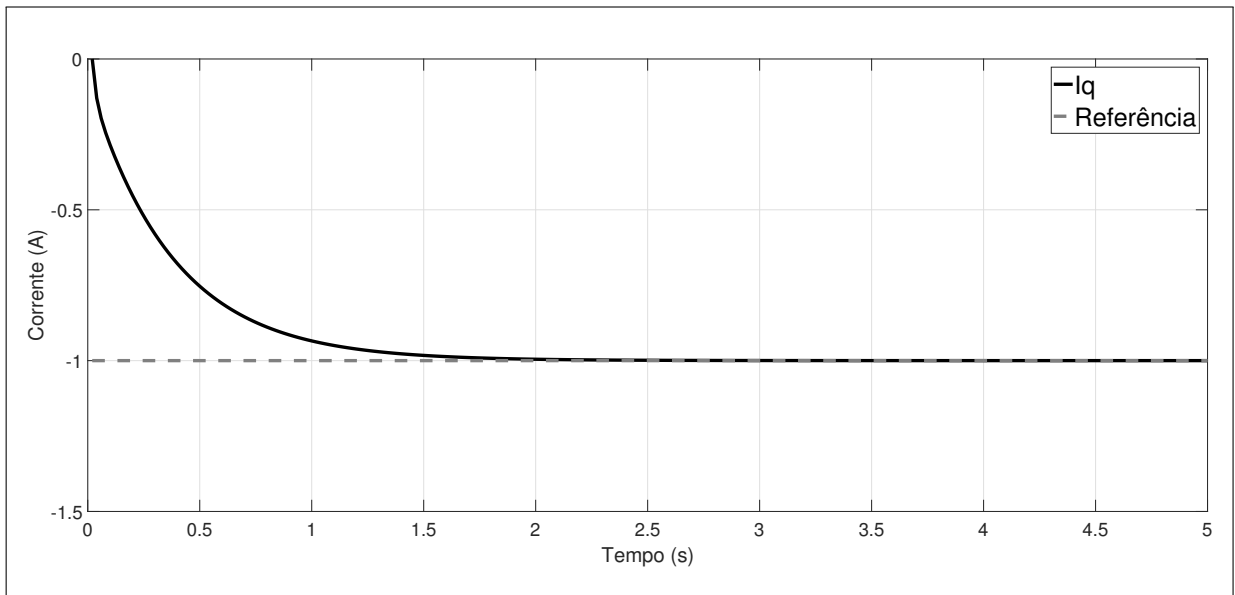
Figura 17 – Resultado da simulação da malha de corrente para referência em $i_q^* = +1,0A$.



Fonte – O autor.

Aplicando-se uma referência em degrau negativo de $-1,0A$ para a corrente em quadratura, i_q^* , obtém-se a resposta apresentada na Figura 18.

Figura 18 – Resultado da simulação da malha de corrente para referência em $i_q^* = -1,0A$.



Fonte – O autor.

Analisando-se as respostas apresentadas, tem-se como as características do controlador projetado, em relação aos critérios de projetos previamente estabelecidos, os valores apresentados na Tabela 4.

Tabela 4 – Características da resposta em simulação do controlador projetado para a malha de corrente.

Critério	Valor
Tempo de subida (t_r)	0,82s
Tempo de estabilização (t_s)	1,08s
Sobressinal (M_p)	0,00%

Fonte – O autor.

Pode-se observar que, comparado aos valores de projeto vistos na Tabela 3, o controlador projetado obteve respostas satisfatórias.

A Tabela 5 apresenta os índices de desempenho do controlador para as duas referências em degrau utilizados em simulação. Utilizou-se o critério da raiz do erro quadrático médio, RMSE (*Root Mean Squared Error*), dado por

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (i_{q_i}^* - i_{q_i})^2}{n}} \quad (4.3)$$

onde n é o número de amostras. Este critério mede o desvio médio das variações entre o resultado obtido, i_{q_i} , e a referência, $i_{q_i}^*$. Como o desvio pode ser positivo ou negativo, faz-se ao quadrado, resultando na unidade de medição ao quadrado. Aplicando-se a raiz quadrada, tem-se o resultado na unidade de medição. Desta forma, quanto menor o resultado, melhor será o desempenho do controlador.

Também foi utilizado o critério do desvio padrão, dado por

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (i_{q_i} - \mu)^2}{n}} \quad (4.4)$$

onde μ é a média de i_{q_i} e n o número de amostras. Este critério mede a dispersão estatística do resultado, indicando o quão longe os resultados obtidos se encontram do valor esperado. Quanto menor o desvio padrão encontrado, melhor será o desempenho do controlador.

Tabela 5 – Desempenho do controlador simulado para malha de corrente.

Referência	RMSE (A)	Desvio Padrão (A)
$i_q^* = +/- 1,0A$	2,8337	0,1220

Fonte – O autor.

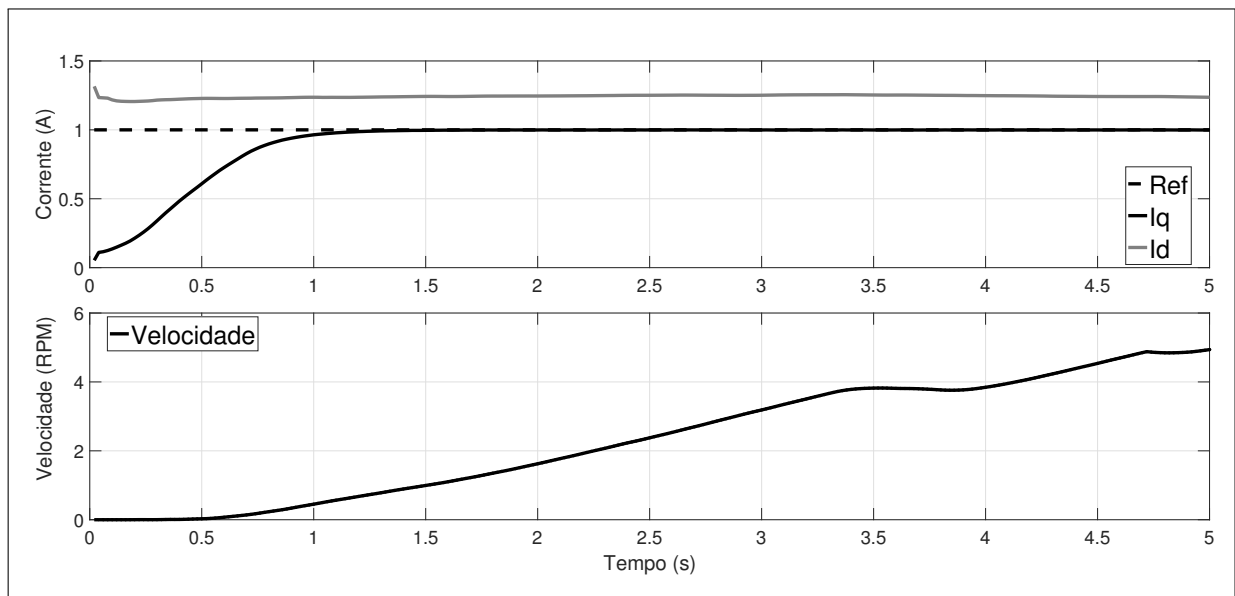
4.2.4 Resultados experimentais do controlador de corrente

Após a realização das simulações, implementou-se o controlador no processador digital. Desta maneira, para o desempenho do controlador PI aplicado à malha de corrente, foram aplicados perfis em degrau positivo de 1,0A e negativo de $-1,0A$ para a corrente em quadratura, i_q^* . Para a corrente de eixo direto, i_d^* , foram aplicados perfis em degrau positivo de 1,3A e negativo de $-1,3A$. O objetivo de se variar as correntes para degraus negativos, conforme exposto, é para avaliar o comportamento do controlador quando necessário ter uma aceleração do eixo do motor, com i_q positivo, e uma frenagem, com valores de i_q negativo.

Estes valores de degrau foram escolhidos de forma experimental, pois, ao longo dos ensaios, viu-se que são razoáveis para, no caso de i_d , gerar campo suficiente para que o motor de indução consiga movimentar a estrutura mecânica do manipulador robótico e, no caso de i_q , gerar o torque necessário.

Na Figura 19 tem-se a resposta para uma referência de $i_q^* = +1,0A$ e $i_d^* = +1,3A$. Na parte inferior da Figura, tem-se a velocidade de rotação do motor, ω_r , que devido às referências positivas, gira no sentido horário, caracterizado pelos valores positivos de velocidade.

Figura 19 – Resultado da Malha de Corrente com referência de $i_q^* = +1,0A$ e $i_d^* = +1,3A$.

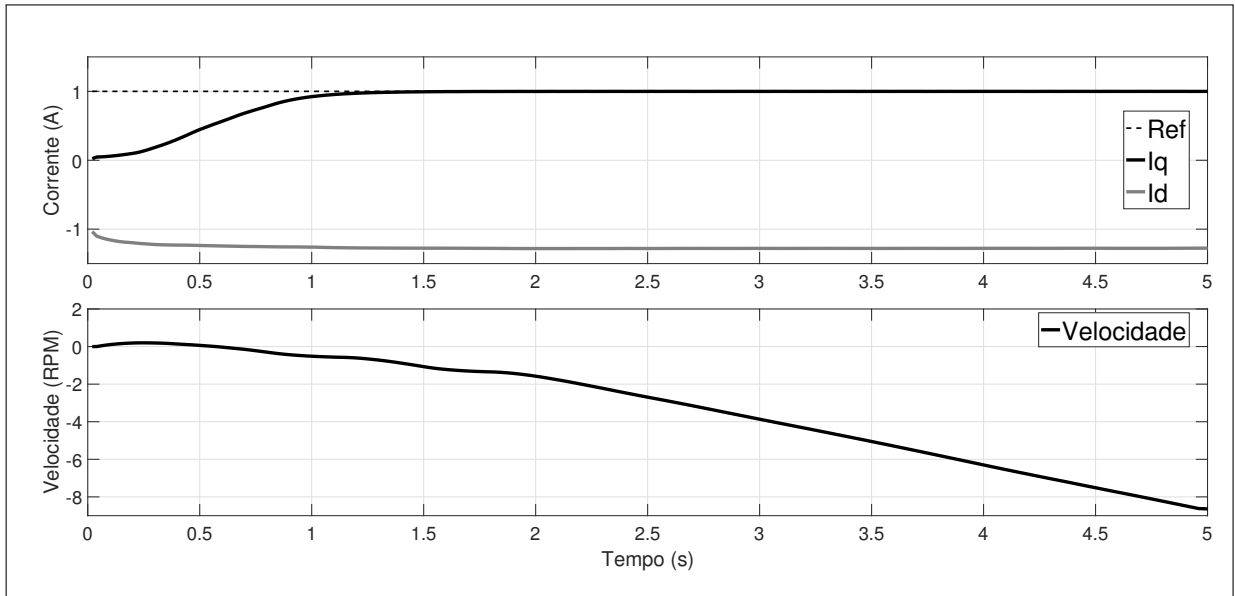


Fonte – O autor.

Mantendo-se a referência de $i_q^* = +1,0A$ e alterando-se a referência de i_d^* para ser negativa, tal que $i_d^* = -1,3A$, tem-se a resposta obtida dada pela Figura 20. Como a referência de i_d é negativa e o produto entre as correntes de eixo direto e em quadratura é vetorial, tem-se

a rotação do eixo do motor, ω_r , no sentido anti-horário, conforme é a apresentado no gráfico inferior da Figura 20, com valores negativos de velocidade.

Figura 20 – Resultado da Malha de Corrente com referência de $i_q^* = +1,0A$ e $i_d^* = -1,3A$.



Fonte – O autor.

Tendo como base as respostas do controlador apresentado nas Figuras 19 e 20, a Tabela 6 apresenta as características da resposta do controlador aplicado à corrente i_q , com relação aos critérios de desempenho do projeto.

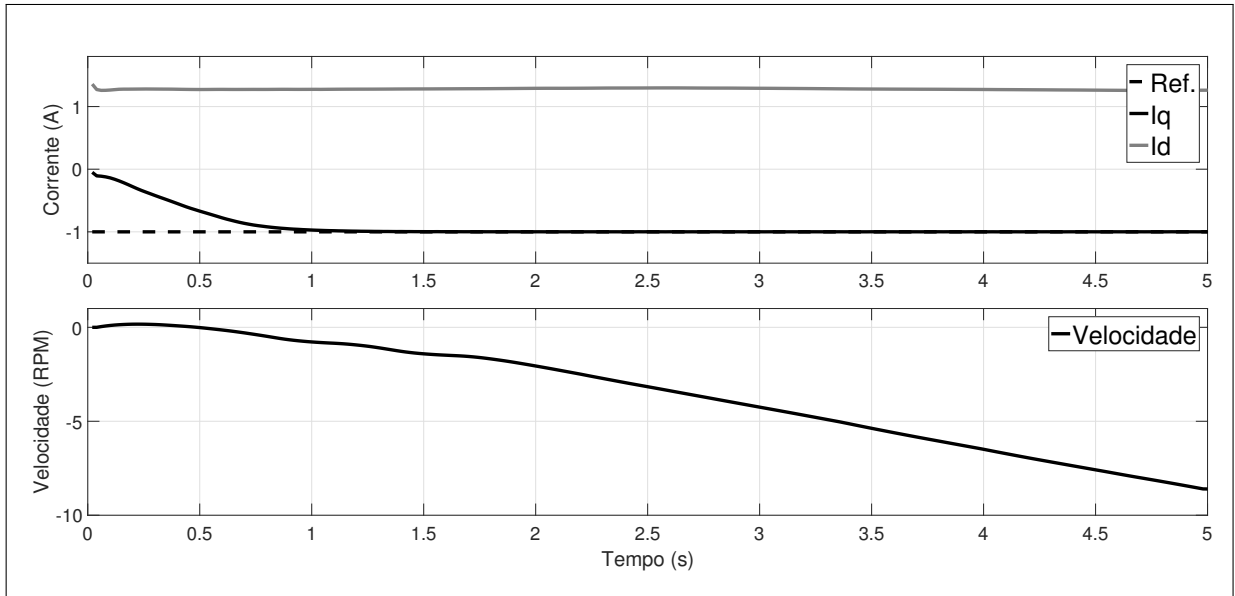
Tabela 6 – Desempenho do controlador para referência $i_q^* = +1,0A$.

Critério	Valor
Tempo de subida (t_r)	0,82s
Tempo de estabilização (t_s)	1,12s
Sobressinal (M_p)	0,00%

Fonte – O autor.

Para uma referência negativa de $-1,0A$ para i_q^* e uma referência positiva de $+1,3A$ para i_d^* , a Figura 21 apresenta o comportamento do controlador. Como no caso da Figura 20, o sentido de rotação do eixo do motor é anti-horário.

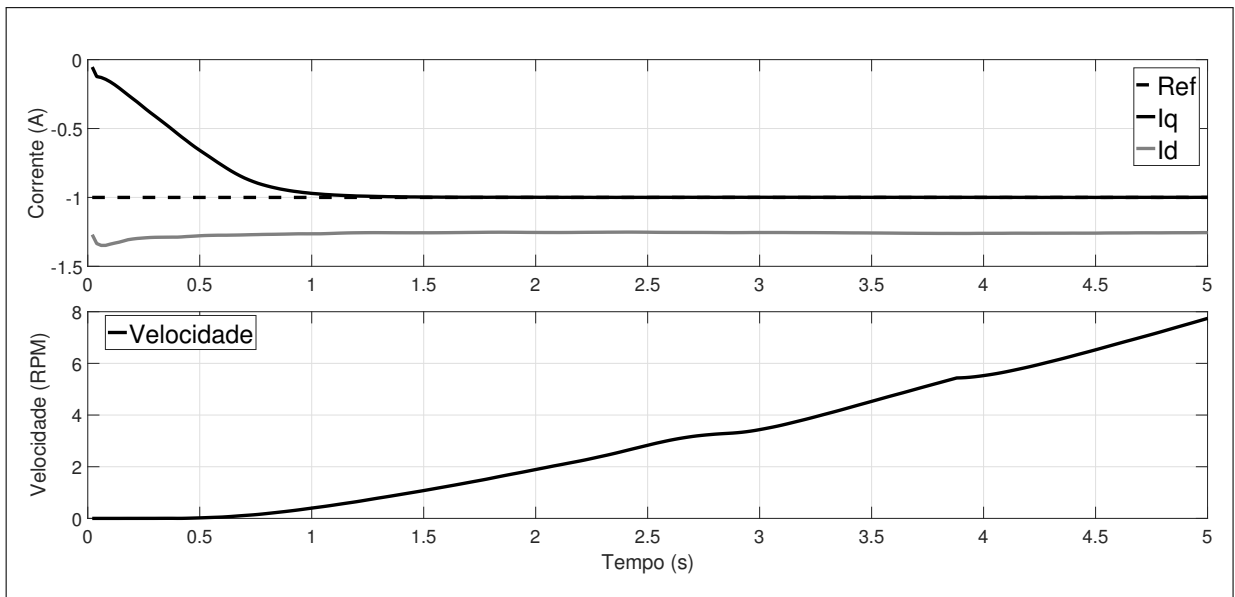
Figura 21 – Resultado da Malha de Corrente com referência de $i_q^* = -1,0A$ e $i_d^* = +1,3A$.



Fonte – O autor.

Para as duas referências negativas, $i_q^* = -1,0A$ e $i_d^* = -1,3A$, tem-se o comportamento apresentado na Figura 22. Como era de se esperar, devido ao produto vetorial entre as correntes, o sentido de rotação do eixo do motor, ω_r , é horário, caracterizado pelos valores positivos de velocidade de rotação.

Figura 22 – Resultado da Malha de Corrente com referência de $i_q^* = -1,0A$ e $i_d^* = -1,3A$.



Fonte – O autor.

A Tabela 7 apresenta as características da resposta do controlador i_q para a referência

em $-1,0A$.

Tabela 7 – Desempenho do controlador para referência $i_q^* = -1,0A$.

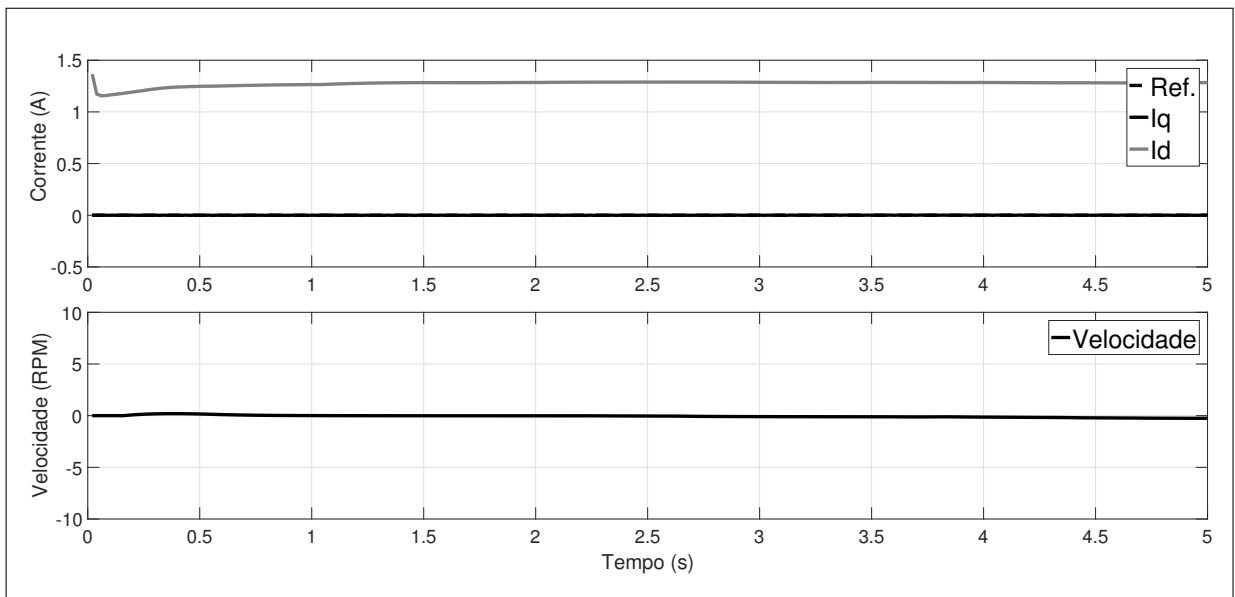
Critério	Valor
Tempo de subida (t_r)	0,84s
Tempo de estabilização (t_s)	0,96s
Sobressinal (M_p)	0,00%

Fonte – O autor.

Ao se comparar os critérios de projeto dados, Tabela 3, e os resultados obtidos, Tabelas 6 e 7, observa-se que os valores obtidos ficaram próximos aos desejados, concluindo-se que o controlador projetado atende aos critérios de projeto.

Com o objetivo de se avaliar o desempenho do controlador em corrente i_q nula, que é útil para o controle de posição da máquina de indução, foi realizado ensaio com $i_q^* = 0,0A$ e $i_d^* = +1,3A$, apresentado na Figura 23. Como a corrente em quadratura é nula, a velocidade de rotação do eixo do motor, ω_r , como esperado, será zero.

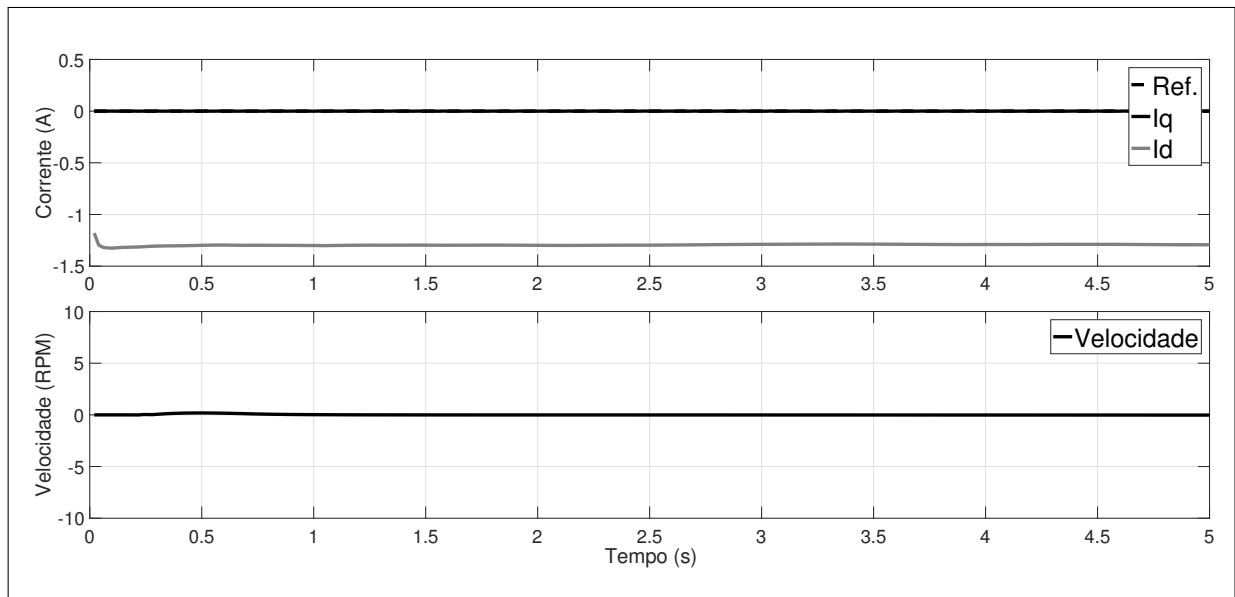
Figura 23 – Resultado da Malha de Corrente com referência de $i_q^* = 0,0A$ e $i_d^* = +1,3A$.



Fonte – O autor.

Para uma referência nula em i_q^* e uma referência negativa de $-1,3A$ em i_d^* , tem-se o resultado apresentado na Figura 24.

Figura 24 – Resultado da Malha de Corrente com referência de $i_q^* = 0,0A$ e $i_d^* = -1,3A$.



Fonte – O autor.

Os ensaios com corrente de quadratura nula tem por objetivo avaliar o comportamento do controlador quando, posteriormente, estiver ligado ao controle de posição da máquina de indução, pois, quando a posição for atingida, o controlador da malha de corrente receberá da malha de posição a referência para a corrente i_q o valor nulo.

Na Tabela 8 tem-se os índices de desempenho do controlador projetado para a malha de corrente para os perfis de referência utilizados na planta. Assim como na simulação, utilizou-se os critérios da raiz do erro quadrático médio, RMSE, e o desvio padrão para se avaliar o desempenho do controlador.

Tabela 8 – Índice de desempenho do controlador da malha de corrente.

Referência	RMSE (A)		Desvio Padrão (A)	
	Controlador i_q	Controlador i_d	Controlador i_q	Controlador i_d
$i_q^* = +1,0A$ e $i_d^* = +1,3A$	0,1158	0,0545	0,1135	0,0095
$i_q^* = +1,0A$ e $i_d^* = -1,3A$	0,1393	0,0272	0,1363	0,0173
$i_q^* = -1,0A$ e $i_d^* = +1,3A$	0,1079	0,0282	0,1058	0,0082
$i_q^* = -1,0A$ e $i_d^* = -1,3A$	0,1077	0,0459	0,1058	0,0105
$i_q^* = 0,0A$ e $i_d^* = +1,3A$	0,0008	0,0208	0,0000	0,0148
$i_q^* = 0,0A$ e $i_d^* = -1,3A$	0,0006	0,0124	0,0000	0,0078

Fonte – O autor.

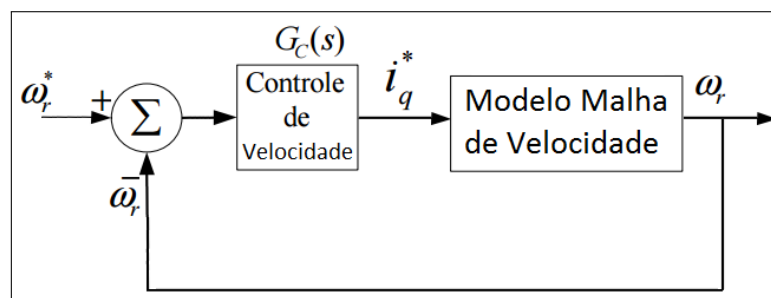
Analisando-se a Tabela 8 e observando-se os baixos erros médios quadráticos obtidos, nota-se que o desempenho dado pelo controlador projetado para a malha de corrente da máquina

de indução foi satisfatório. Assim sendo, pode-se passar a projetar o controlador da malha de controle intermediária, ou seja, a malha de velocidade.

4.3 Controlador da Malha de Velocidade

Assim como realizado para o projeto do controlador de corrente, pode-se simplificar a Figura 3 para melhor entendimento da malha de velocidade, conforme mostrado na Figura 25.

Figura 25 – Diagrama simplificado da malha de velocidade.



Fonte – O autor.

Da Figura 25 tem-se que ω_r^* é a referência de velocidade. O bloco *Controle de velocidade* é composto por um controlador que atua na malha mecânica da máquina de indução através do sinal de controle, i_q^* , que é a referência para a malha de corrente. A realimentação do controlador se dá através do *encoder* fixado na ponta do eixo do motor que, após passar por um bloco de filtro digital onde se realiza a derivada da posição, fornece o valor da velocidade atual, ω , para o controlador.

Para esta malha de controle optou-se por realizar simulações e ensaios experimentais com um controlador do tipo PI e um controlador com estrutura RST, a fim de se verificar as respostas obtidas pelos dois tipos de controlador.

Inicialmente será realizada a identificação e validação do modelo da malha de velocidade para, logo após, projetar os controladores PI e RST e apresentar os resultados de simulação e experimentais para os dois tipos de controladores.

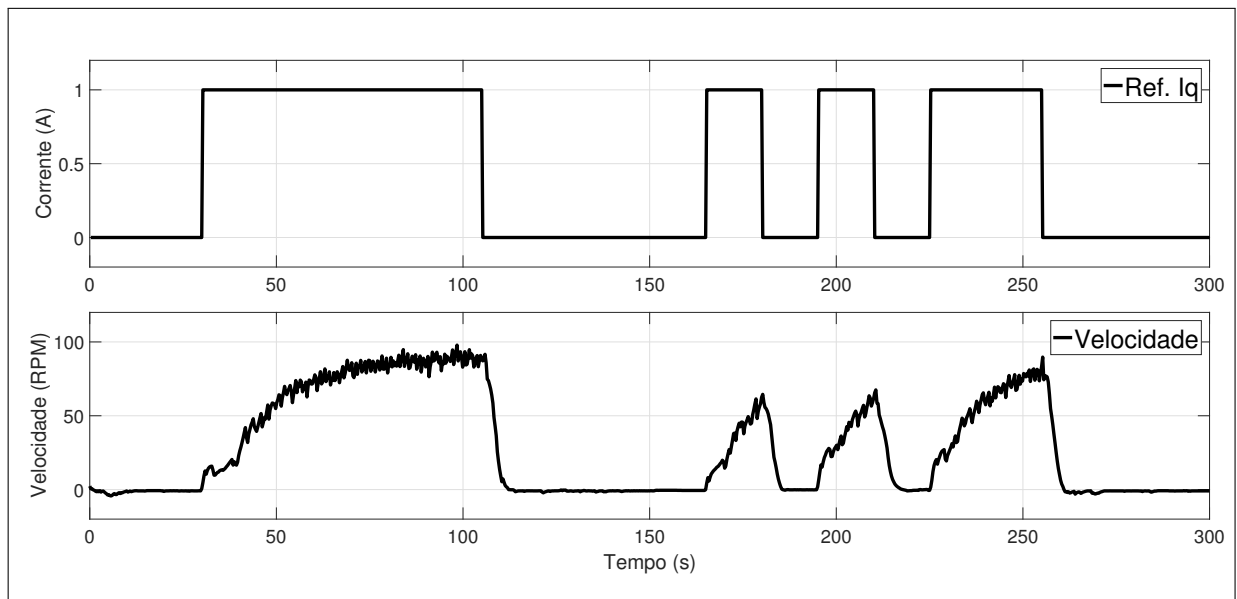
4.3.1 Identificação do modelo

Inicialmente foi realizado o ensaio de identificação do modelo da malha de velocidade através da injeção de um sinal do tipo PRBS (*Pseudo-Random Binary Sequence*) na entrada do sistema, i_q^* , e, através do *encoder* instalado na ponta do eixo do rotor, coletou-se as respostas

obtidas para um período de amostragem de 0,3s. O procedimento de identificação foi realizado de forma experimental, aplicando-se referências de corrente para i_q e coletando-se os sinais através dos sensores apresentados na seção 4.1.

A Figura 26 apresenta na parte superior as referências do tipo PRBS para i_q^* aplicada ao sistema e, na parte inferior, a velocidade, ω_r , obtida.

Figura 26 – Sinais aplicados e obtidos para identificação da malha de velocidade.



Fonte – O autor.

Utilizando os dados apresentados na Figura 26 para realizar a identificação do modelo da malha de velocidade, obteve-se a equação abaixo, onde relaciona a referência da corrente de quadratura, i_q^* , e a velocidade do eixo do motor, ω_r :

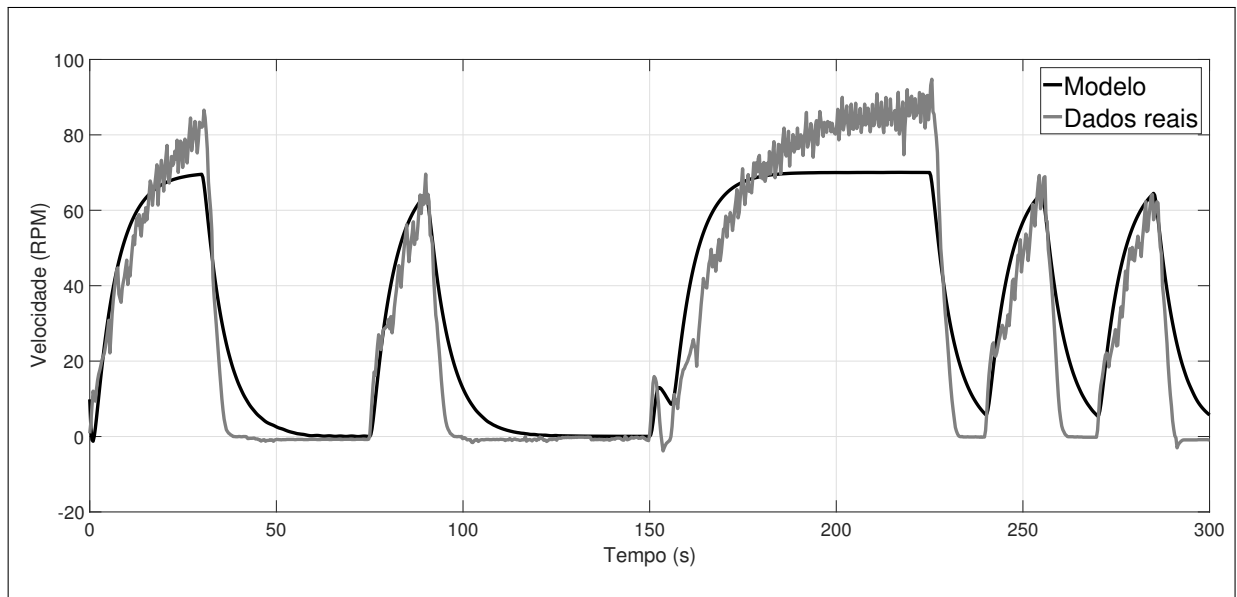
$$G(s) = \frac{73,15}{15,25s^2 + 9,706s + 1} \quad (4.5)$$

Realizando-se a discretização do modelo da malha de velocidade, chega-se à equação abaixo:

$$G(z) = \frac{0,2026z + 0,1902}{z^2 - 1,821z + 0,8262} \quad (4.6)$$

A Figura 27 apresenta a validação do modelo dado pela equação (4.5) em relação à resposta real apresentada pela malha de velocidade. O resultado apresentou um erro quadrático médio normalizado de 68,12%.

Figura 27 – Validação do modelo obtido para a malha de velocidade.



Fonte – O autor.

4.3.2 Projeto do controlador de velocidade

Para o projeto do controlador da malha de velocidade, definiu-se os critérios apresentados na Tabela 9.

Tabela 9 – Critérios de projeto para o controlador da malha de velocidade.

Critério	Valor
Tempo de subida (t_r)	2,50s
Tempo de estabilização (t_s)	2,80s
Sobressinal (M_p)	0,00%

Fonte – O autor.

Como exposto na seção 4.3, optou-se por projetar dois tipos de controlador para a malha de velocidade, sendo um do tipo PI e outro com a estrutura RST, a fim de se verificar a resposta entre os dois tipos de controladores. Para a sintonia do controlador PI aplicou-se a técnica de Ziegler-Nichols, apresentada na seção 3.1.1, obtendo-se os parâmetros $a = 42,4$ e $L = 1,92$. Aplicando-se estes valores na Tabela 2, tem-se os coeficientes do controlador dados por $K_p = 0,021$ e $T_i = 0,64ms$.

Para a obtenção dos parâmetros do controlador RST, utilizou-se a técnica de alocação de polos, apresentada na seção 3.2.1, onde, conforme apresentado, define-se um filtro de referência que atenda aos critérios de projeto apresentados na Tabela 9. Desta forma, a partir da

equação (3.14), tem-se o seguinte modelo de referência:

$$\frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} = \frac{0,6474q}{q^2 - 0,4433q + 0,0907} \quad (4.7)$$

Resolvendo-se a equação *Diophantina* dada por (3.14), tem-se:

$$(q^2 + a_1q + a_2) \cdot 1 + b_0(s_0q + s_1) = (q^2 + a_{m1}q + a_{m2}) \quad (4.8)$$

Igualando-se os coeficientes:

$$\begin{aligned} s_0 &= \frac{a_{m1} - a_1}{b_0} \\ s_1 &= \frac{a_{m2} - a_2}{b_0} \end{aligned} \quad (4.9)$$

Desta forma, conforme visto na seção 3.2, o controlador é caracterizado pelos polinômios:

$$\begin{aligned} R(q) &= q + \frac{b_1}{b_0} \\ S(q) &= s_0q + s_1 \\ T(q) &= \frac{b_{m0}q}{b_0} \end{aligned} \quad (4.10)$$

Resultando nos seguintes parâmetros para o controlador:

$$s_0 = 295,88 \quad , \quad s_1 = -273,31 \quad , \quad r = 0,9957 \quad \text{e} \quad T = 22,88 \quad (4.11)$$

Com os controladores projetados, pode-se realizar a simulação dos controladores aplicados ao modelo identificado, através do *software Simulink*[®], apresentado na próxima seção.

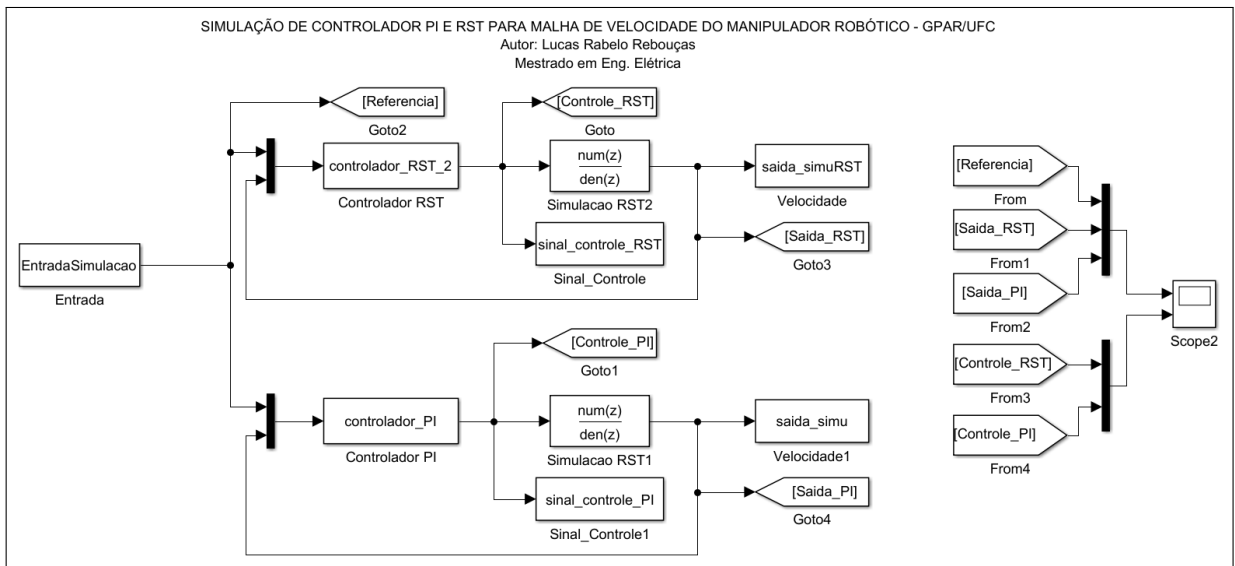
4.3.3 Resultados de simulação do controlador de velocidade

Após a realização do projeto, realizou-se a simulação dos controladores aplicados ao modelo identificado através do *software Simulink*[®], conforme apresentado na Figura 28. Os códigos implementados nos blocos *Controlador PI* e *Controlador RST* podem ser vistos no Anexo B.

Desta forma, aplicou-se perfis de referência aos controladores, tais como degrau positivo e negativo, senoidal e trapezoidal, conforme apresentado a seguir, a fim de se verificar os seus comportamentos.

Para uma referência em degrau positivo, com o comando de velocidade, ω_r^* , iniciando em 10 *rpm* e em seguida aumentado-se para 20 *rpm*, tem-se o comportamento apresentado

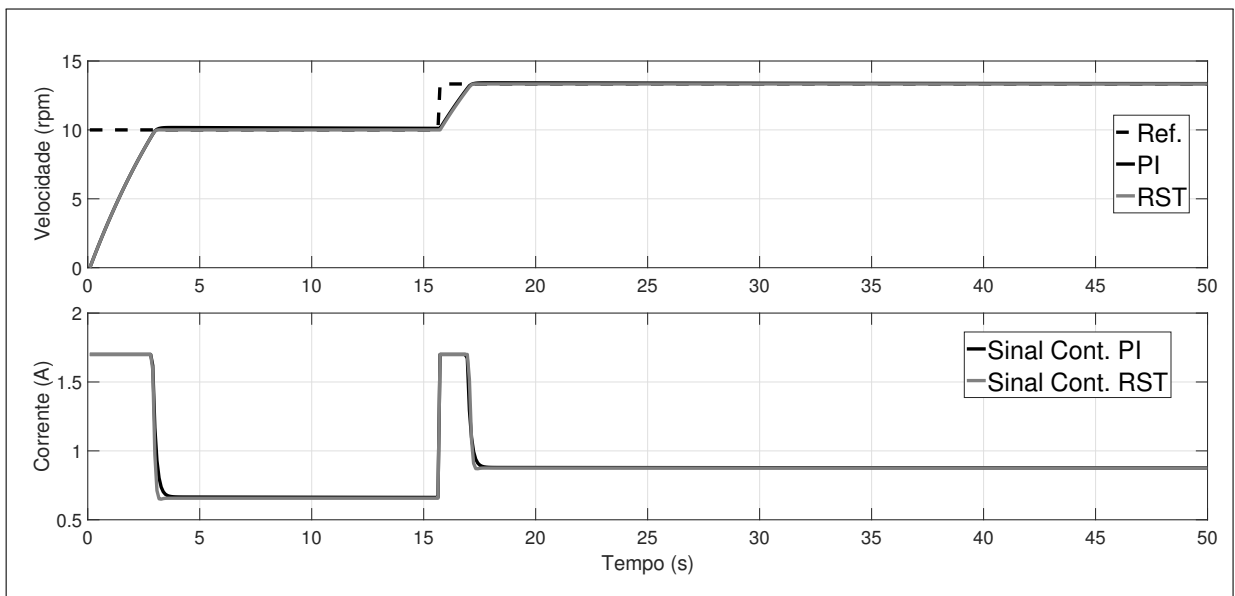
Figura 28 – Diagrama do *Simulink*[®] para simulação dos controladores da malha de velocidade.



Fonte – O autor.

na Figura 29. Nesta mesma figura, no gráfico inferior, tem-se os sinais de controle, i_q^* , correspondente às correntes, fornecidos por cada controlador, onde observa-se que, por alguns instantes, houve a saturação do sinal que, assim como na implementação prática por questões de segurança dos equipamentos, foi implementado de modo a manter as correntes em níveis seguros.

Figura 29 – Resposta da simulação com referência em degrau positivo.



Fonte – O autor.

A partir da Figura 29, pode-se retirar as características de cada controlador, conforme exibido na Tabela 10.

Tabela 10 – Características dos controladores de simulação para a malha de velocidade.

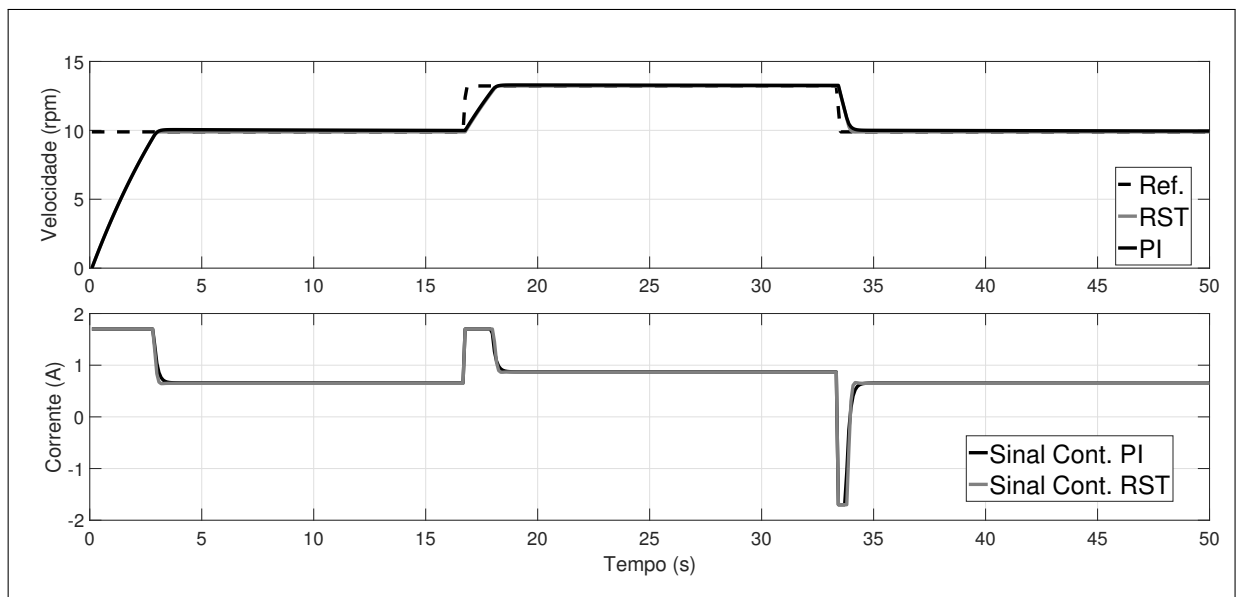
Critério	Valor	
	Controlador PI	Controlador RST
Tempo de subida (t_r)	2,62s	2,82s
Tempo de estabilização (t_s)	2,87s	2,97s
Sobressinal (M_p)	0,00%	0,00%

Fonte – O autor.

Observa-se que, segundo a Tabela 10, as características à resposta ao degrau dos controladores projetados apresentaram valores próximos aos de critério de projeto, dados pela Tabela 9.

Aplicando-se uma referência de velocidade, ω_r^* , com um degrau positivo seguido de um degrau negativo, tal como iniciando-se o comando em 10 rpm, após certo período aumenta-se para 20 rpm e, logo após, volta para 10 rpm, tem-se a resposta apresentada na Figura 30, assim como os sinais de controle, i_q^* , fornecidos pelos controladores.

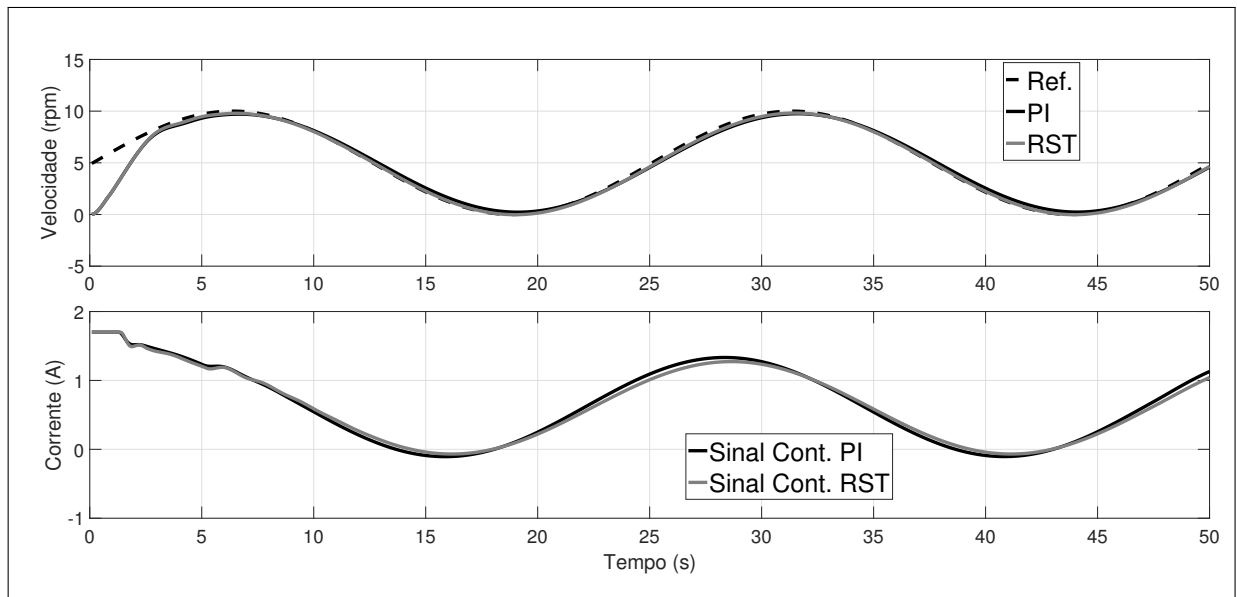
Figura 30 – Resposta das simulações com referência em degrau positivo e negativo.



Fonte – O autor.

Injetando-se uma referência de velocidade, ω_r^* , senoidal à simulação, onde a senoide tem pico positivo em 10 rpm e negativo em 0 rpm, tem-se as respostas obtidas e os sinais de controle apresentados na Figura 31.

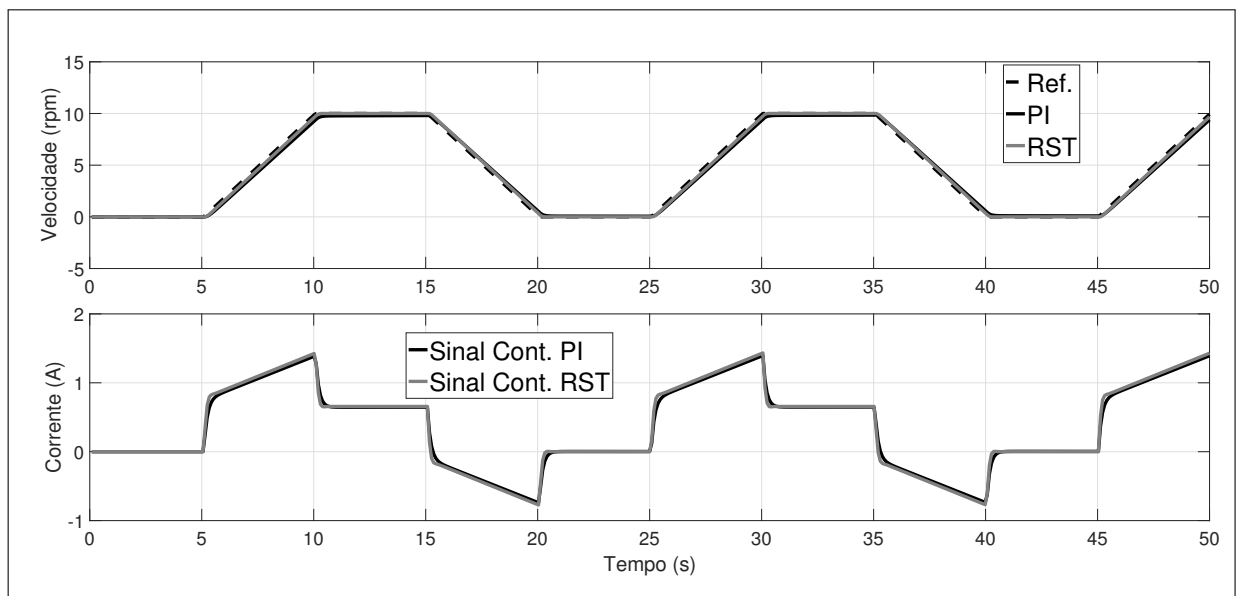
Figura 31 – Resposta das simulações com referência senoidal.



Fonte – O autor.

Utilizando-se uma referência de velocidade, ω_r^* , trapezoidal com o pico em 10 *rpm*, obteve-se as seguintes respostas, assim como os sinais de controle de cada controlador implementado, apresentadas na Figura 32.

Figura 32 – Resposta das simulações com referência trapezoidal.



Fonte – O autor.

A fim de se avaliar os dois tipos de controladores implementados em simulação, aplicou-se índices de desempenho para se verificar a performance de cada, conforme apresentado

na Tabela 11. Utilizou-se o índice estatístico da raiz do erro quadrático médio, RMSE, onde, conforme apresentado anteriormente, na seção 4.2.3, quanto menor, melhor o controlador. Também foi utilizado o desvio padrão, o qual indica uma medida de dispersão dos dados em torno da média, onde quanto menor, melhor será o desempenho do controlador.

Tabela 11 – Índice de desempenho dos controladores simulados da malha de velocidade.

Referência	RMSE (rpm)		Desvio Padrão (rpm)	
	Controlador PI	Controlador RST	Controlador PI	Controlador RST
Degrau 10 – 20rpm	32,065	1,968	1,5468	2,311
Degrau 10 – 20 – 10rpm	31,774	2,202	1,570	2,155
Senoidal	17,998	4,18	3,536	3,402
Trapezoidal	4,273	3,745	4,056	3,949

Fonte – O autor.

Como pode-se observar nas repostas apresentadas para diversos perfis de referência e os índices de desempenho exibidos na Tabela 11, os dois tipos de controlador mostraram ter comportamentos próximos, por vezes tendo o sinal de resposta e controle sobrepostos, não possibilitando notar diferenças visíveis. Desta forma, pode-se concluir que qualquer um dos dois tipos de controlador pode ser implementado na malha de velocidade quando esta estiver submetida à malha de posição.

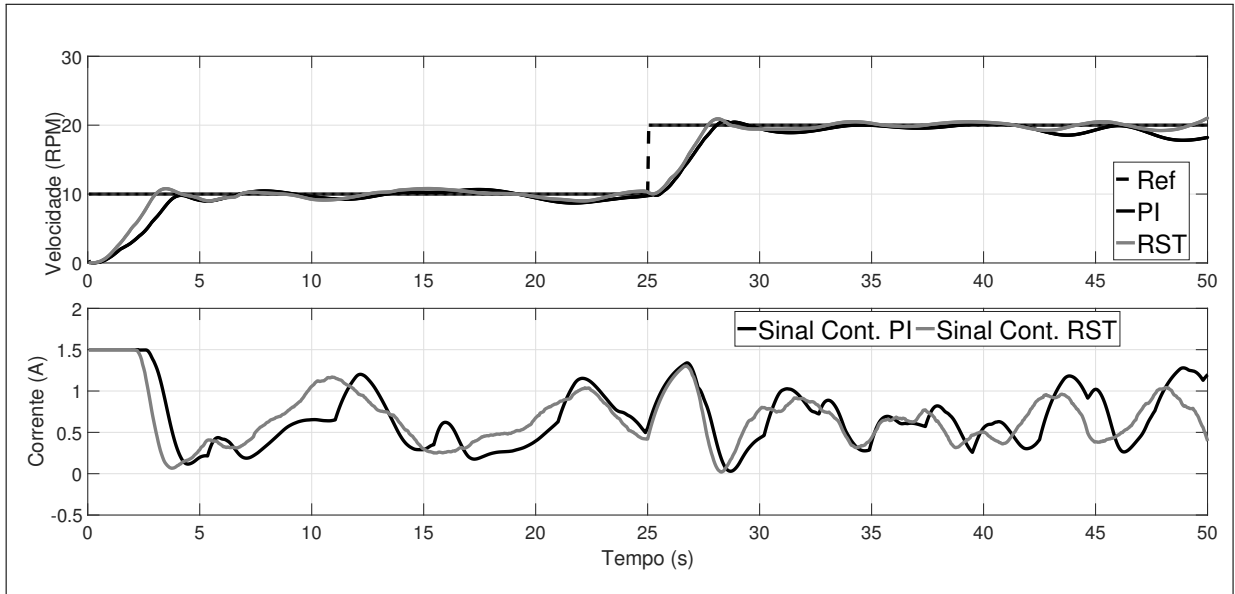
Na próxima seção serão implementados os dois controladores no processador digital utilizado, a fim de se verificar o comportamento de forma experimental.

4.3.4 Resultados experimentais do controlador de velocidade

Após a simulação computacional dos controladores, implementou-se os dois tipos no processador digital de sinais, conforme exposto nas seções 3.1 e 3.2, e aplicou-se os mesmos perfis de referência utilizados em simulação para verificar as repostas experimentais. Assim sendo, tem-se os resultados a seguir.

Para uma referência de velocidade, ω_r^* , em degraú positivo, variando de 10 rpm para 20 rpm e mantendo-se $I_d = 1,3A$, tem-se a resposta e os sinais de controle de cada controlador apresentado na Figura 33. Pode-se notar a semelhança entre as repostas obtidas e os sinais de controle de cada controlador.

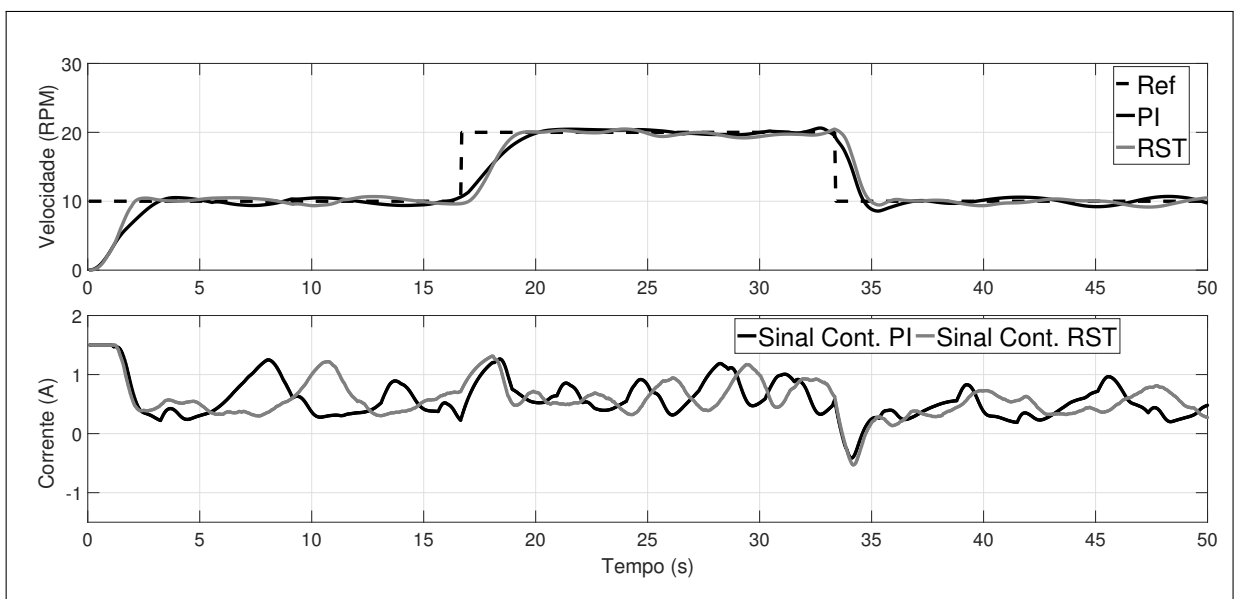
Figura 33 – Resposta dos controladores na malha de velocidade com referência em degrau positivo.



Fonte – O autor.

Já para uma referência de velocidade, ω_r^* , em degrau positivo seguido de um negativo, tal como a referência iniciando-se em 10 *rpm*, aumentado-se para 20 *rpm* e, após certo período, volta para 10 *rpm*, tem-se as respostas apresentadas na Figura 34.

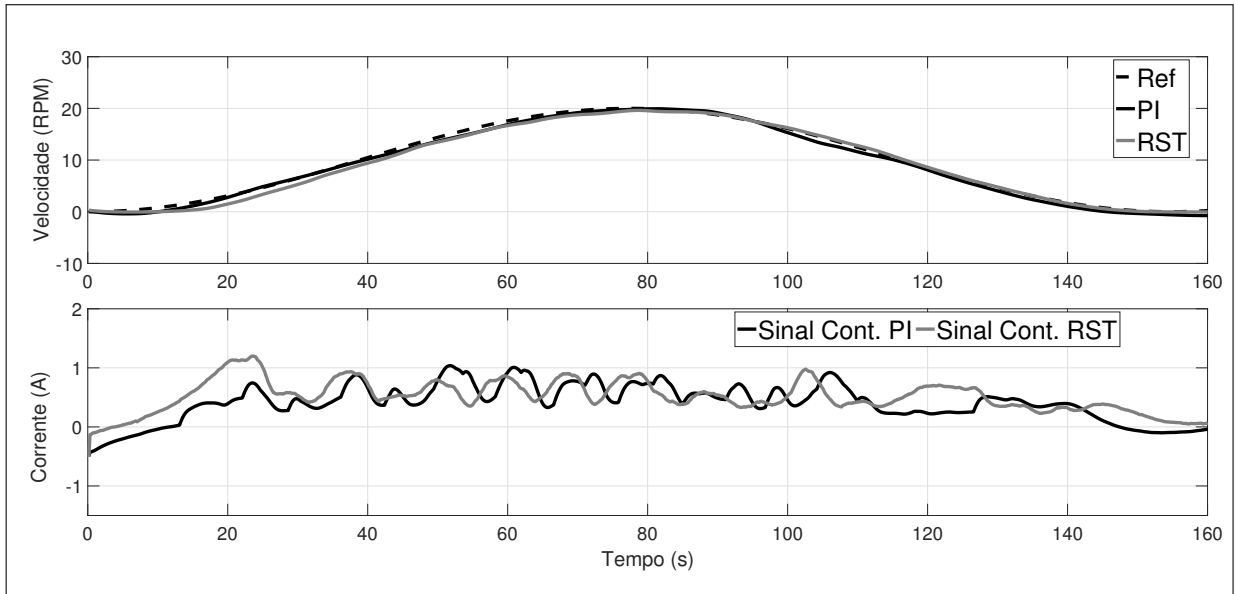
Figura 34 – Resposta dos controladores na malha de velocidade com referência em degrau positivo e negativo.



Fonte – O autor.

Aplicando-se uma referência senoidal, tal como na simulação, tem-se as respostas dos controladores mostradas na Figura 35, assim como os sinais de controle fornecido por cada tipo de controlador.

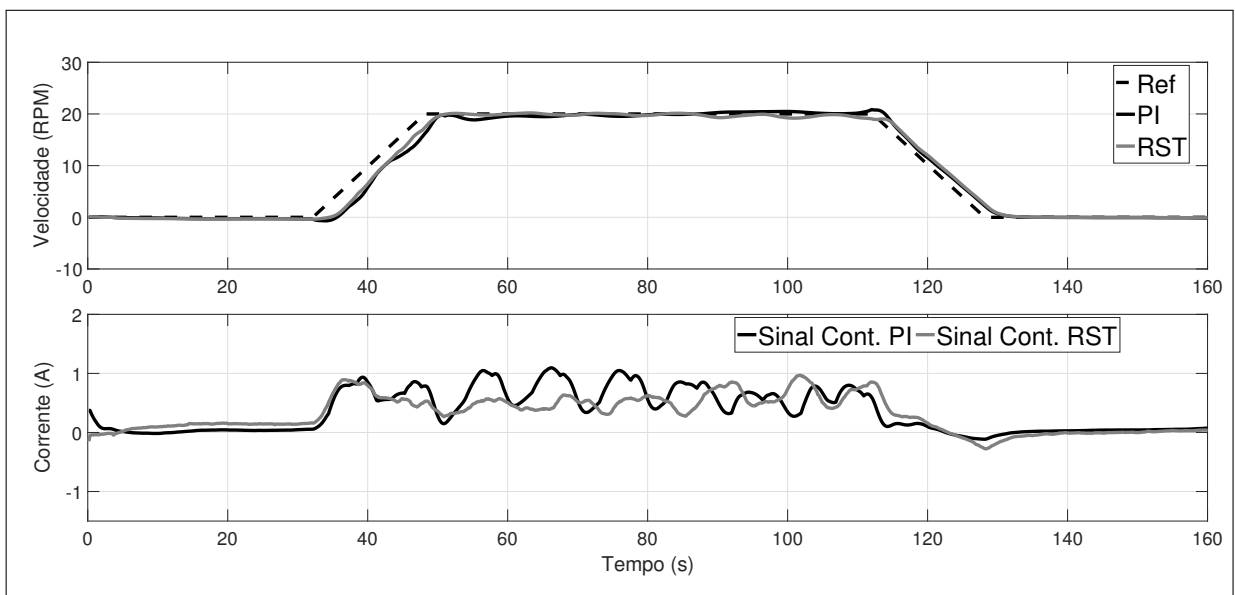
Figura 35 – Resposta dos controladores na malha de velocidade com referência senoidal.



Fonte – O autor.

Ao se aplicar uma referência trapezoidal, a fim de simular um movimento *peak and place* do manipulador, tem-se as respostas e os sinais de controle apresentados na Figura 36.

Figura 36 – Resposta dos controladores na malha de velocidade com referência trapezoidal.



Fonte – O autor.

A Tabela 12 apresenta os índices de desempenho dos controladores aplicados ao sistema real. Para verificar o desempenho do controlador projetado, foram utilizados os índices, assim como na simulação, da raiz do erro quadrático médio, RMSE, e o desvio padrão.

Tabela 12 – Índice de desempenho dos controladores na malha de velocidade.

Referência	RMSE (rpm)		Desvio Padrão (rpm)	
	Controlador PI	Controlador RST	Controlador PI	Controlador RST
Degrau 10 – 20rpm	2,69	2,38	5,53	5,55
Degrau 10 – 20 – 10rpm	2,23	2,41	5,06	4,91
Senoidal	0,59	0,73	7,28	7,20
Trapezoidal	1,36	1,17	9,34	9,27

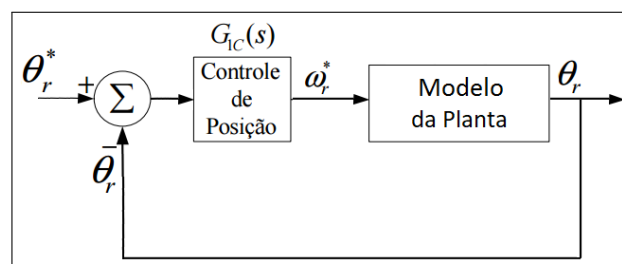
Fonte – O autor.

Pode-se observar que as repostas dos controladores são semelhantes, assim como os sinais de controle, tal como apresentado nos resultados de simulação. Também nota-se que as respostas obtidas estão próximas aos critérios de projetos estabelecidos preliminarmente e os índices de desempenho apresentados são satisfatórios, já que apresentaram baixo erro quadrático médio. Como os controladores apresentaram comportamentos semelhantes, decidiu-se por manter o controlador com estrutura RST na malha de velocidade quando da implementação do controlador da malha de posição. Mas nada impede que seja utilizado o controlador do tipo PI para esta função, sendo apenas uma decisão do autor.

4.4 Controlador da Malha de Posição

Assim como nas demais malhas, é possível realizar a simplificação da Figura 3 para tornar o entendimento da malha de posição mais claro, conforme apresentado na Figura 37.

Figura 37 – Diagrama simplificado da malha de posição.



Fonte – O autor.

Da Figura 37 tem-se que θ_r^* é a referência de posição do controlador e o bloco

Controle de posição é composto por um controlador do tipo proporcional que atua na malha mecânica da máquina de indução através do sinal de controle, ω_r^* . A realimentação do controlador se dá através do *encoder* fixado na ponta do eixo do motor, o qual fornece a posição atual do eixo do rotor, θ_r .

Objetivando-se ter uma resposta mais rápida do posicionamento do eixo do motor, utilizou-se um controlador do tipo Proporcional, onde a equação (3.1) fica reduzida somente ao termo proporcional, sendo os demais nulos.

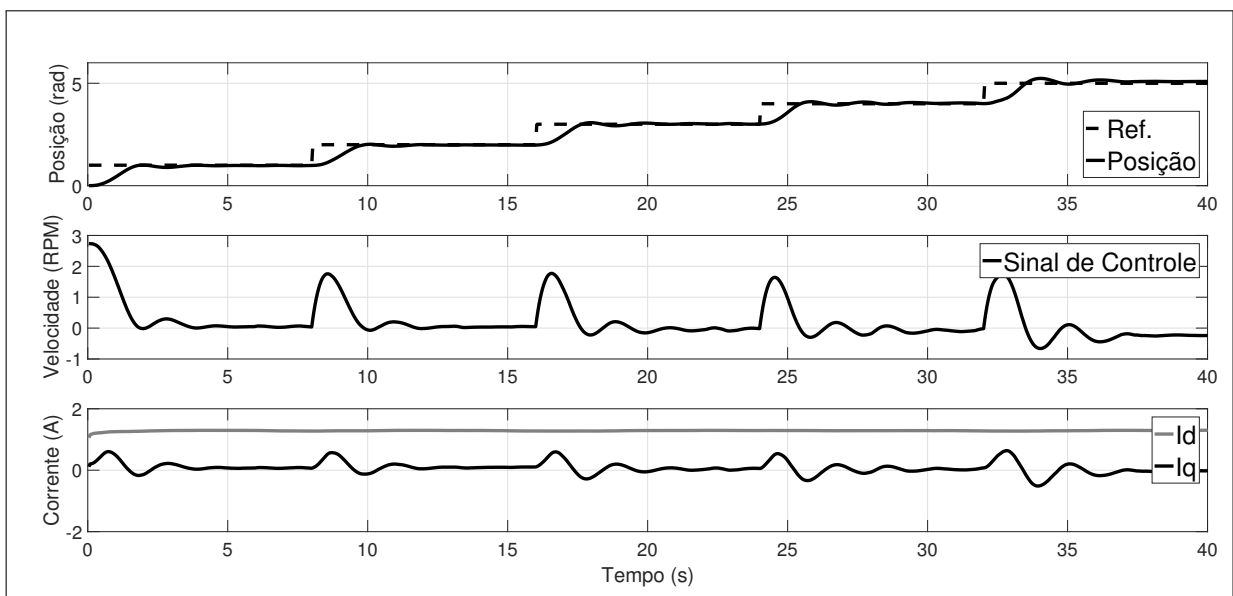
Neste caso, o valor do ganho proporcional foi obtido através de ensaios experimentais aplicando-se valores à K_p de modo a se obter uma resposta satisfatória. Desta forma, encontrou-se o valor do ganho proporcional de $K_p = 0,12$.

4.4.1 Resultados experimentais do controlador de posição

Para validar o controlador de posição aplicado ao manipulador robótico, foram aplicadas referências em degraus positivos, negativos e trapezoidal para θ_r^* , a fim de se emular situações de *pick-and-place* fabril.

Inicialmente foi aplicado uma referência em degrau sequencial, onde a referência θ_r^* inicia-se em 1 *rad* indo até 5 *rad* incrementando-se em passos de 1 *rad*, após um certo tempo. A resposta obtida, juntamente com o sinal de controle fornecido pelo controlador, ω_r^* e as correntes de quadratura, i_q , e de eixo direto, i_d , são exibidos na Figura 38.

Figura 38 – Resposta do controlador de posição com referência em degrau sequencial.



A Tabela 13 apresenta as características da resposta do controlador aplicado à malha de posição, com relação aos critérios de projeto.

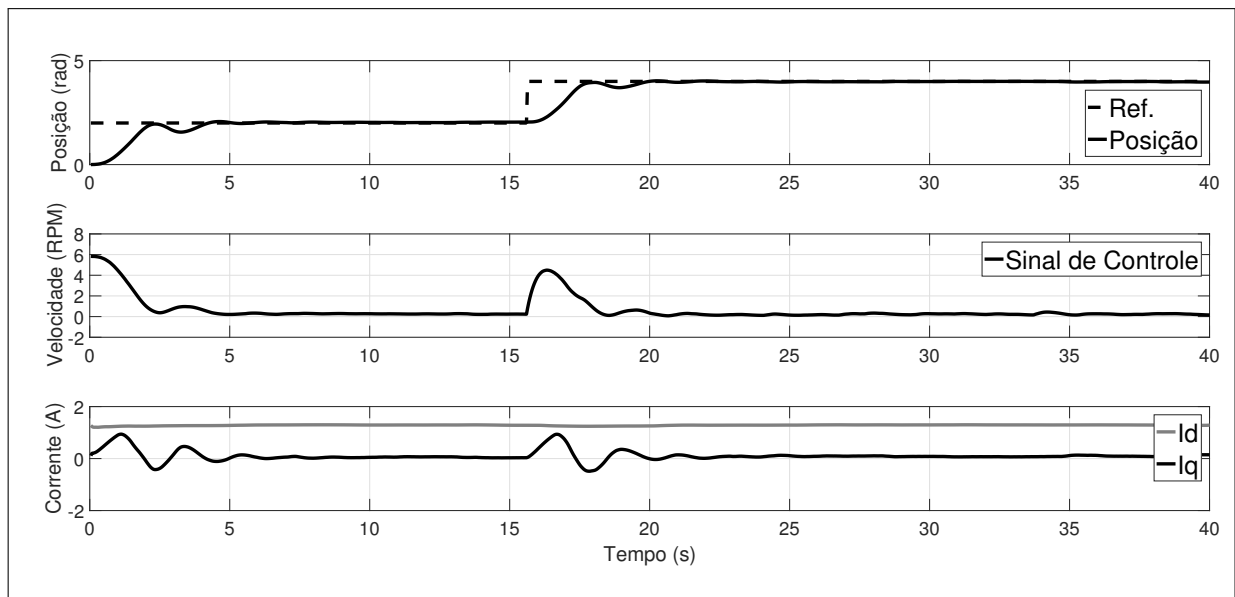
Tabela 13 – Características da resposta do controlador da malha de posição.

Critério	Valor
Tempo de subida (t_r)	1,60s
Tempo de estabilização (t_s)	1,72s
Sobressinal máximo (M_p)	4,03%

Fonte – O autor.

Logo após, realizou-se o ensaio com a referência de posição em degrau, sendo com um incremento de passo maior, de 2 rad , iniciando em 2 rad e, após algum tempo, altera-se para 4 rad . A resposta obtida é apresentada na Figura 39, assim como os sinais de controle, corrente de quadratura e de eixo direto.

Figura 39 – Resposta do controlador de posição com referência em degrau.

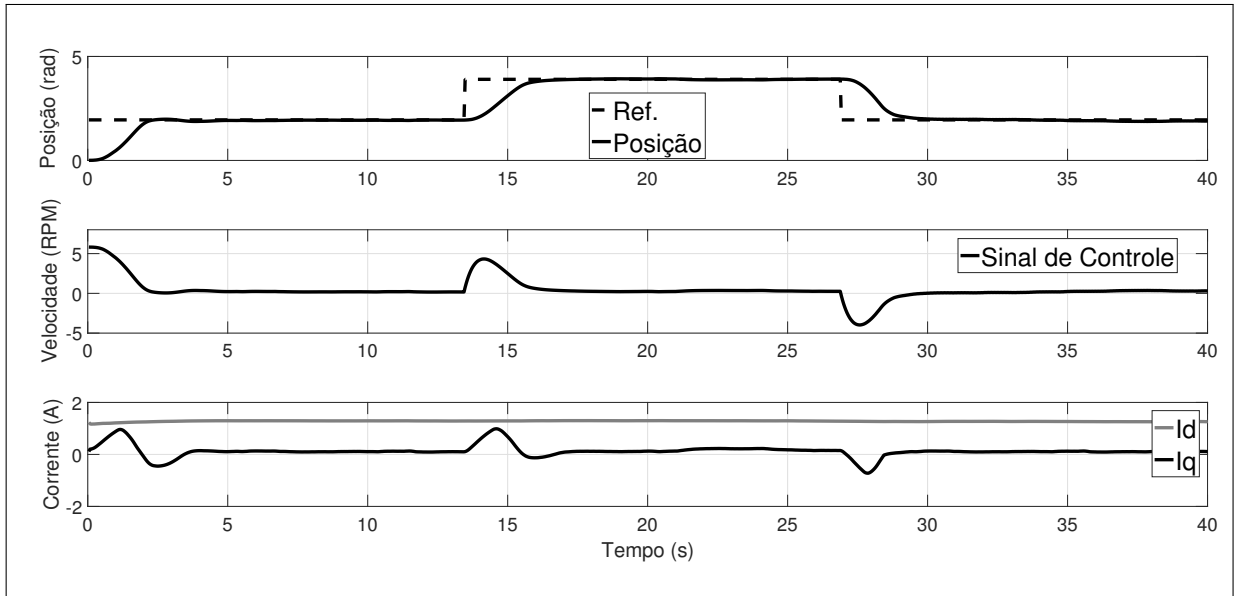


Fonte – O autor.

Para uma referência em degrau positivo seguida de um degrau negativo, tal como iniciando-se a referência θ_r^* em 2 rad , aumentado-se para 4 rad e, logo após, volta para 2 rad , tem-se a resposta apresentada na Figura 40.

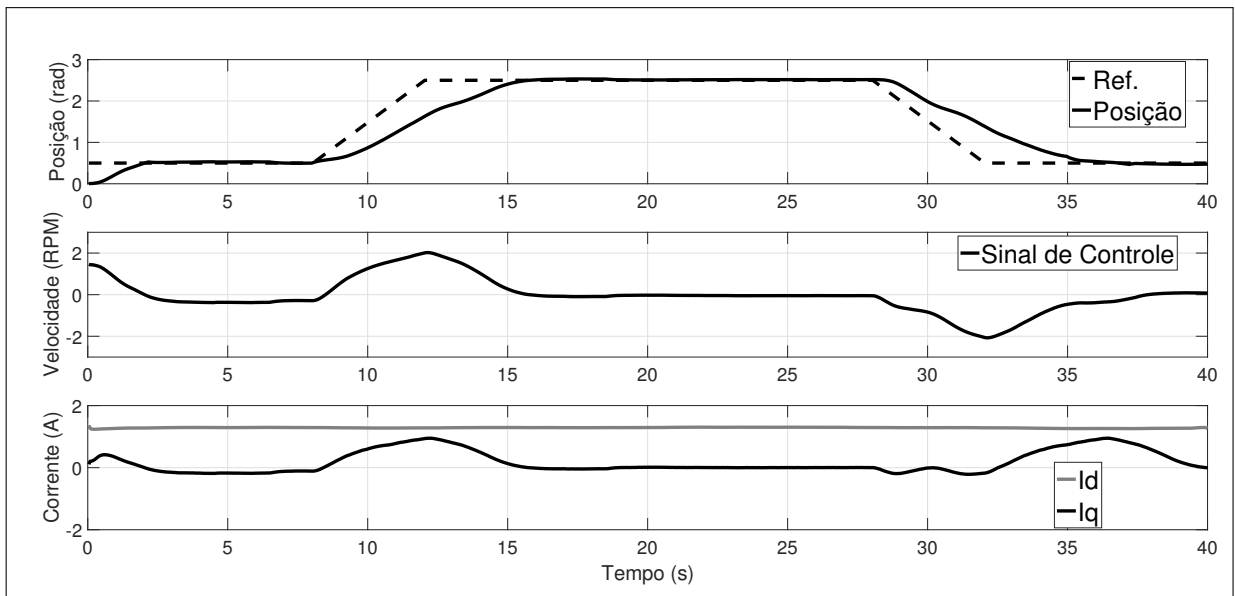
Ao se aplicar uma referência trapezoidal para θ_r^* , onde se inicia com um degrau de $0,5\text{ rad}$, sobe para $2,5\text{ rad}$ e, após certo período, volta para $0,5\text{ rad}$, tem-se o comportamento apresentado na Figura 41.

Figura 40 – Resposta do controlador de posição com referência em degrau positivo e negativo.



Fonte – O autor.

Figura 41 – Resposta do controlador de posição com referência trapezoidal.



Fonte – O autor.

Aplicando-se os índices de desempenho, conforme feito anteriormente para as outras malhas de controle, tem-se os resultados apresentados pela Tabela 14.

Como pode ser observado nos resultados obtidos, o controlador projetado para a malha de posição para a máquina de indução apresentou resultados satisfatórios, com baixo erro quadrático médio. Assim sendo, pode-se concluir que o controlador projetado de forma empírica está adequado para a utilização no primeiro grau de liberdade do manipulador robótico.

Tabela 14 – Índices de desempenho do controlador da malha de posição.

Referência	RMSE (rad)	Desvio padrão (rad)
Referência 1 – 2 – 3 – 4 – 5rad	0,31	1,48
Referência 2 – 4rad	0,46	1,09
Referência 2 – 4 – 2rad	0,54	0,98
Referência Trapezoidal	0,33	0,90

Fonte – O autor.

4.5 Considerações finais

Neste capítulo foi visto que através dos modelos identificados, das simulações realizadas e dos dados experimentais aqusitados da planta montada em laboratório, a viabilidade dos métodos propostos ao longo do trabalho e a constatação da aplicação do controle de campo orientado para o controle da máquina de indução como servoposicionador.

5 CONCLUSÕES E TRABALHOS FUTUROS

O objetivo principal deste trabalho foi o estudo e implementação do controle de um motor de indução trifásico utilizado como servoposicionador aplicado a um manipulador robótico utilizando técnicas de controle por campo orientado indireto, analisando a utilização de controladores PI e RST para as malhas de corrente, velocidade e posição.

A grande importância da utilização do acionamento de motores de indução como servoposicionadores se dá, apesar da sua complexidade de modelagem e controle, pelo fato de possuir baixo custo quando comparado a motores síncronos e de fácil manutenção, bem como à motores de corrente contínua.

Neste trabalho foi apresentado o controlador PI aplicado à malha de corrente da máquina de indução, visando dar maior robustez ao sistema, com repostas satisfatórias aos critérios de projeto, apresentando tempo de subida, utilizando critério 0 – 90%, de 0,82s e tempo de estabilização de 1,23s, sem sobressinal.

Para a malha de controle de velocidade realizou-se a implementação dos controladores digitais PI e RST, onde foram obtidas repostas bem semelhantes, com o controlador PI apresentando um tempo de subida de 8,16s, tempo de acomodação de 8,87s e sem sobressinal. Já o controlador RST apresentou um tempo de subida de 8,36s, tempo de acomodação de 8,97s e sem sobressinal. Verificando-se os índices de desempenho apresentados, nota-se que os resultados obtidos para os dois tipos de controlador são bastante semelhantes, fazendo com que as duas implementações sejam satisfatórias para o controle de velocidade da máquina de indução. Quando da implementação do controlador de posição, manteve-se o controlador RST aplicado à malha de velocidade.

Para a malha de posição, foi apresentado o controlador clássico do tipo Proporcional e realizado ensaios empíricos para a sua sintonia. Logo após foram aplicados diversos perfis de referência de posicionamento para o eixo do motor, tendo uma resposta com as características de 1,60s de tempo de subida, e 1,72s de tempo de estabilização e com um sobressinal de aproximadamente 4%.

5.1 Trabalhos Futuros

Com os estudos e implementações realizados ao longo deste trabalho, foi possível se verificar alguns pontos que podem ser propostos como trabalhos futuros relacionados ao projeto

de desenvolvimento do manipulador robótico. São eles:

Como este trabalho teve como objetivo controlar o primeiro grau de liberdade do manipulador robótico, propõe-se a extensão das técnicas utilizadas para os demais graus. Desta forma, será necessário o estudo e controle de matrizes de transferência para a conversão dos sistemas de coordenadas para cada grau do manipulador, utilizando as técnicas de Denavit-Hartenberg, cinemática direta e inversa.

Pode-se também propor a utilização de controladores adaptativos, a fim de estimar em tempo real os parâmetros do motor de indução trifásico, já que, conforme apresentado, podem variar de acordo com alguns fatores. Fazendo, assim, com que os coeficientes do controlador sejam adaptados de acordo com a estimação realizada em tempo real.

A realização de ensaios de acurácia e repetibilidade, conforme apresentado na norma ISO (*International Standard Organization*) 9283 (*Manipulating industrial robots – Performance criteria and related methods*), detalhado em (SILVA, 2016), é proposto para trabalhos futuros a fim de se obter medidas de desempenho do posicionamento do manipulador robótico de acordo com as normas internacionais.

Propõe-se também, o desenvolvimento de uma interface gráfica para controle e aquisição dos dados do manipulador, de forma a tornar a utilização do manipulador robótico mais intuitiva.

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas. Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3. ed. [S.l.]: UFMG, 2007.
- ARAÚJO, A. G. **Controle vetorial de um motor de indução trifásico aplicado em sistemas de posicionamento**. Dissertação (Mestrado) — Universidade Federal da Paraíba, João Pessoa, 2011.
- ASTRÖM, K.; HÄGGLUND, T. **PID Controllers: Theory, Design, and Tuning**. [S.l.]: Instrument Society of America, 1995.
- ASTRÖM, K.; WITTERNMARK, B. **Computer Controlled Systems: Theory and design**. [S.l.]: Prentice-Hall, 1990.
- ASTRÖM, K.; WITTERNMARK, B. **Adaptive control**. [S.l.]: Prentice-Hall, 1994.
- AVAGO TECHNOLOGIES. **HEDM-55xx/560x HEDS-55x/56xx - Quick Assemble Two and Three Channel Optical Encoders**. 2014. Disponível em: <<http://www.ti.com/lit/ds/symlink/tms320f28335.pdf>>. Acesso em: 27 mai. 2017.
- BARBI, I. **Teoria Fundamental do Motor de Indução**. [S.l.]: UFSC, 1985.
- BEERTEN, J.; VERVECKKEN, J.; DRIESEN, J. Predictive direct torque control for flux and torque ripple reduction. **IEEE Transactions on Industrial Electronics**, IEEE, v. 57, n. 1, p. 404–412, 2010.
- BIM, E. **Máquinas Elétricas e Acionamento**. 3. ed. [S.l.]: Elsevier, 2014.
- BLASHKE, F. **The principle of field orientation as applied to the new transvector closed-loop control system for rotating-field machines**. [S.l.]: Siemens Review, 1972. XXXIX.
- BOSE, B. K. **Power Electronics and AC drives**. [S.l.: s.n.], 1986. v. 2.
- CABRAL, E. **Cinemática da posição de robôs manipuladores**. Dissertação (Monografia) — Universidade de São Paulo, 2003.
- CÂMARA, H. T.; CARATI, E. G.; HEY, H. L.; PINHEIRO, H.; PINHEIRO, J. R.; GRUNDLING, H. A. Speed and position servo for induction motor using robust model reference adaptive control. v. 2, p. 1721–1727, 2002.
- CAMPOS, A. C. R. **Projeto e Análise de Controladores a partir da Identificação em Malha Fechada: Estudo de Casos**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2007.
- DE SOUZA JUNIOR, A. B.; DINIZ, E. C.; HONÓRIO, D. A.; DOS REIS, L. L. N.; BARRETO, L. H. S. C. Hybrid position control technique of induction motor drive. **Control and Cybernetics**, v. 42, n. 4, p. 754–772, 2013.
- DE SOUZA JÚNIOR, A. B. **Estudo e implementação de um servoposicionador aplicando controle vetorial indireto a um motor de indução trifásico**. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, 2010.

DINIZ, E. C. **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.** Dissertação (Doutorado) — Universidade Federal do Ceará, Fortaleza, 2013.

DINIZ, E. C.; DE SOUZA JÚNIOR, A. B.; HONÓRIO, D. A.; BARRETO, L. H. S. C.; DOS REIS, L. L. N. An elbow planar manipulator driven by induction motors using sliding mode control for current loop. **Control and Cybernetics**, v. 41, n. 2, p. 395–413, 2012.

FURTUNATO, A. F. A. **Implementação de um Controlador de Velocidade Usando Modos Deslizantes Suaves para um Motor de Indução Trifásico.** Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 1997.

HASSE, K. **Zur Dynamik Drehzahl geregelter Antriebe mit Stromrichter gespeisten Asynchron - kurzschulss lagermaschinen.** Dissertação (Ph D Dissertation) — Tech. Hochschule, Darmstadt, 1969.

HONÓRIO, D. A. **Controle de posição do motor de indução por modos deslizantes: uma análise comparativa.** Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, 2013.

JACOBINA, C. B.; RIBEIRO, L. A.; FILHO, J. B. M.; SALVADORI, F.; LIMA, A. M. N. Sistema de acionamento com motor de indução orientado indiretamente pelo campo com adaptação MRAC da velocidade. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 14, n. 1, p. 41–49, 2003.

KOSOW, I. I. **Máquinas Elétricas e Transformadores.** [S.l.]: Globo, 1996.

LANDAU, I. D.; LOZANO, R.; M'SAAD, M.; KARIMI, A. **Adaptive control: algorithms, analysis and applications.** [S.l.]: Springer Science & Business Media, 2011.

LATHI, B. P. **Sinais e Sistemas Lineares.** 2. ed. [S.l.]: Bookman, 2007.

LEM. **Current Transducer La-25-NP.** 2011. Disponível em: <<http://www.lem.com/docs/products/la\%2025-np.pdf>>. Acesso em: 21 mai. 2017.

LIAW, C.; LIN, F.; KUNG, Y. Design and implementation of a high performance induction motorservo drive. **IEE Proceedings in Electronic Power Applications**, v. 140, n. 4, p. 241–248, 1993.

LIPO, T. A.; NOVOTNY, D. W. **Vector Control and Dynamics of AC Drives.** [S.l.]: Oxford university press, 1996. v. 41.

LJUNG, L. **System Identification. Theory For The User.** 2. ed. [S.l.]: Prentince-Hall, 1999.

OGATA, K. **Modern Control Engineering.** 5th. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2010.

REBOUÇAS, L. R.; DE SOUZA JÚNIOR, A. B.; NETO, T. R. F.; DOS REIS, L. L. N.; BARRETO, L. H. S. C. Identificação de um motor de indução trifásico aplicado para posicionamento utilizando modelo de Hammerstein. **Revista Tecnologia**, Universidade de Fortaleza, v. 36, p. 7–17, 2015.

SANTANA, E. S.; BIM, E.; AMARAL, W. C. A predictive algorithm for controlling speed and rotor flux of induction motor. **IEEE Transactions on Industrial Electronics**, IEEE, v. 55, n. 12, p. 4398–4407, 2008.

SHIAU, L. G.; LIN, J. L. L. Stability of sliding-mode current control for high performance induction motor position drives. **IEE Proceedings-Electric Power Applications**, IET, v. 148, p. 69–75, 2001.

SILVA, J. L. N. **Projeto e realização de controle fuzzy para manipulador industrial tipo SCARA e sua avaliação de desempenho à luz da norma ISO 9283**. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, 2016.

SILVA, W. A.; DE SOUZA JÚNIOR, A. B.; TORRICO, B. C.; HONÓRIO, D. A.; NETO, T. R. F.; DOS REIS, L. L. N.; BARRETO, L. H. S. C. Generalized predictive control robust for position control of induction motor using field-oriented control. **Electrical Engineering**, Springer, v. 97, n. 3, p. 195–204, 2015.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. [S.l.]: wiley New York, 2006. v. 3.

TAKAHASHI, I.; ITOH, S. A simple positioning servo system by breaking control. **IEEE transactions on industry applications**, IEEE, v. 28, n. 6, p. 1314–1321, 1992.

TEXAS INSTRUMENTS. **TMS320F2833x, TMS320F2823x Digital Signal Controllers (DSCs)**. 2016. Disponível em: <http://www.mouser.com/ds/2/678/V02-1046EN_DS_HEDM-55xx_2014-11-20-952416.pdf>. Acesso em: 08 mai. 2017.

TOLIYAT, H. A.; AREFEEN, M. S.; RAHMAN K. M. FIGOLI, D. Rotor time constant updating scheme for a rotor flux-oriented induction motor drive. **IEEE Transactions on Power Electronics**, IEEE, v. 14, n. 5, p. 850–857, 1999.

WEG S.A. **W22 - Motor Elétrico Trifásico. Catálogo Técnico**. 2016. Catálogo Técnico - Mercado Brasileiro. Disponível em: <<http://ecatalog.weg.net/files/wegnet/WEG-w22-motor-trifasico-tecnico-mercado-brasil-50023622-catalogo-portugues-br.pdf>>. Acesso em: 09 mai. 2017.

WOLOVICH, W. A. **Automatic Control Systems**. 2. ed. [S.l.]: Oxford University Press, 1994.

YU, Z. Space vector PWM with TMS320C24x using hardware and software determined switching patterns. **Application Report SPRA524**, Texas Instruments, 2001.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **trans. ASME**, v. 64, n. 11, p. 759–768, 1942.

ANEXO A – COMPONENTES UTILIZADOS

Os principais componentes utilizados para o desenvolvimento desse projeto são mencionados nos tópicos a seguir.

A.1 Motor de indução trifásico

Será utilizado um motor de indução trifásico do tipo gaiola de esquilo de 0,5cv, modelo W21 *standard* da WEG[®], conforme mostrado na Figura 42, com tensão nominal de 380V em estrela e 220V em triângulo, 4 pólos e corrente nominal de 1,18A.

Figura 42 – Motor WEG a ser utilizado.



Fonte – (WEG S.A., 2016).

A Tabela 15 apresenta maiores detalhes do motor a ser utilizado.

Tabela 15 – Folha de dados do motor de indução a ser utilizado.

Parâmetro	Valor
Potência	0.50cv
Potência	0.37kW
Pólos	4
Frequência	60Hz
Tensão nominal	220/380V
Corrente nominal	1,18/2,07A
Rotação nominal	1720rpm
Resistência do estator	35,58Ω
Resistência do rotor	87,44Ω
Indutância do estator	0,16H
Indutância do rotor	0,16H
Indutância mútua estator-rotor	0,884H
Momento de inércia	0,00079kgm ²
Grau de proteção	IP55

Fonte – (WEG S.A., 2016).

A.2 Inversor trifásico

O inversor recebe os sinais de chaveamento diretamente do DSP. A partir de um circuito integrado, opto-acopladores recebem os sinais de chaveamento provenientes do DSP e repassam ao inversor de frequência, de modo a isolar o circuito de controle do de potência. O chaveamento se dá a uma frequência de 10kHz. O inversor utilizado é de três níveis da *Semikron*[®], modelo SKS 18G B6I11 V12, mostrado na Figura 43

Figura 43 – Inversor trifásico da *Semikron*[®].



Fonte – O autor.

A.3 Encoder

O *encoder* é responsável por fornecer ao DSP a posição do eixo do rotor. Fazendo com que o controlador seja capaz de calcular o escorregamento da máquina de indução, condição necessária para o controle por orientação de campo indireto.

O *encoder* utilizado é da fabricante Avago[®], modelo HEDS-5500, exibido na Figura 44.

Figura 44 – *Encoder* utilizado da Avago[®].



Fonte – (AVAGO TECHNOLOGIES, 2014).

Os dados técnicos do *encoder* são apresentados na Tabela 16.

Tabela 16 – Folha de dados do *encoder* utilizado.

Parâmetro	Valor
Temperatura de Armazenamento	-40 à 100°C
Temperatura de Operação	-40 à 100°C
Tensão de Alimentação	-0,5 à 7V
Tensão de Saída	0,5 à 7V
Corrente de Saída	-1,0 à 5mA
Resolução	500PPR
Vibração	20g, 5 à 1000Hz
Velocidade	30000rpmV
Aceleração	250000 $\frac{rad}{sec^2}$

Fonte – (AVAGO TECHNOLOGIES, 2014).

A.4 Sensor de corrente

Serão utilizados três sensores de corrente de efeito *hall* da marca LEM[®], modelo LA-25-NP, conforme apresentado na Figura 45.

Figura 45 – Sensor de corrente utilizado da LEM[®].



Fonte – (LEM, 2011).

As especificações dos sensores de corrente podem ser observadas na Tabela 17.

Tabela 17 – Características do sensor de corrente a ser utilizado.

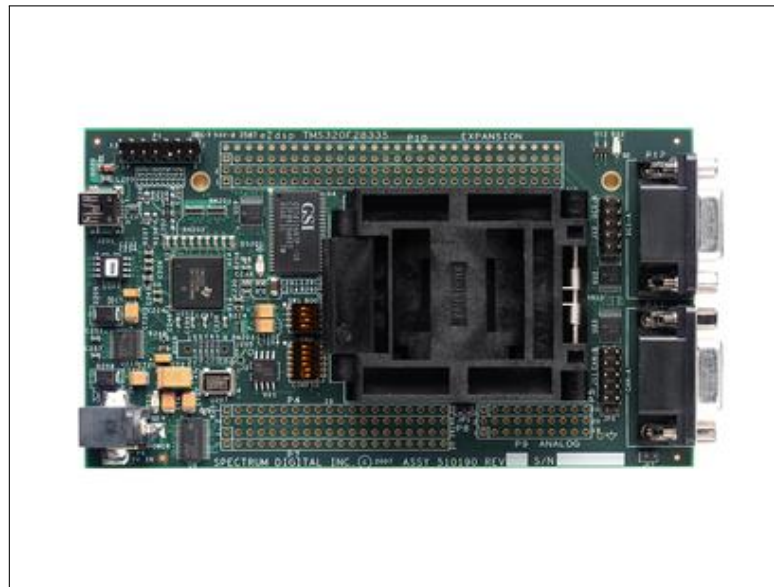
Parâmetro	Valor
Tipo de Saída	Instantânea
Tecnologia	Efeito <i>hall</i>
Medição	Corrente
Corrente nominal primária	25A
Range de medição	0 – 36A
Tensão de alimentação	15/ – 15V
Corrente nominal secundária	25mA
Precisão	+/- 0,5%
Erro de linearidade	< 0,2%
Temperatura ambiente de operação	-40... + 85°C

Fonte – (LEM, 2011).

A.5 Processador digital de sinais

Para embarcar a estratégia de controle de todas as malhas, realizar leitura dos sinais dos sensores de corrente e do encoder através de canais analógicos e comandar o chaveamento do inversor trifásico, será utilizado o processador digital de sinais modelo eZdsp F28335 da *Texas Instrument*[®], conforme mostrado na Figura 46.

Figura 46 – Processador digital de sinais utilizado.



Fonte – O autor.

Algumas características do DSP utilizado são mostradas na Tabela 18.

Tabela 18 – Características do DSP a ser utilizado.

Parâmetro	Valor
Frequência	150MHz
Milhões de instruções por segundo	150
Flash	512kB
RAM	68kB
Canais analógicos	24
Canais PWM	12

Fonte – (TEXAS INSTRUMENTS, 2016).

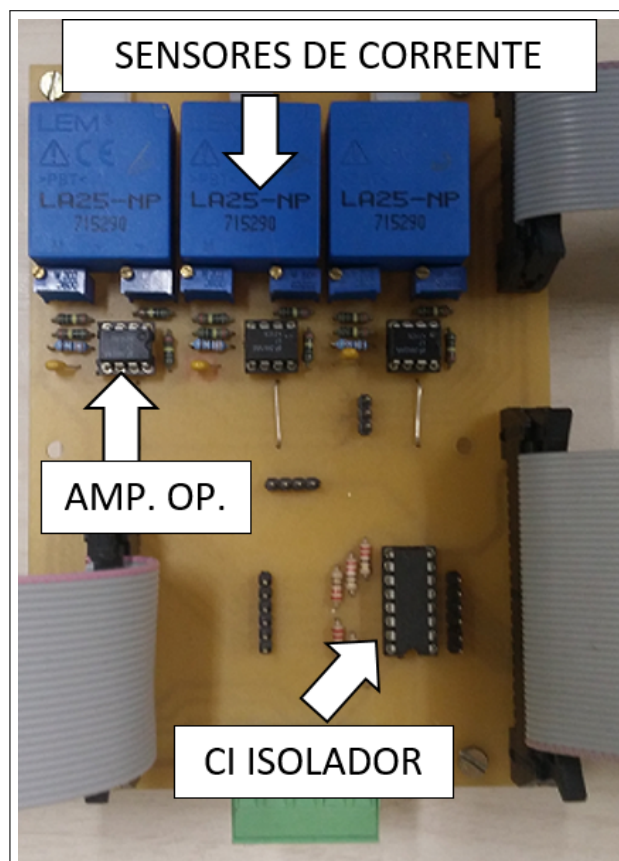
Para maiores informações acerca do DSP, verificar em (TEXAS INSTRUMENTS, 2016).

A.6 Placa condicionadora de sinais

A placa condicionadora de sinais foi concebida com o propósito de realizar a interface entre os sinais de controle e potência, através de circuito integrado com optoacopladores. Além de receber os sensores de corrente ligados à amplificadores operacionais, de modo que seja possível realizar a calibração dos sinais de saída dos sensores através de potenciômetros, com o intuito de adequar o saída à entrada do DSP.

A placa condicionadora, assim como seus componentes, podem ser vistos na Figura 47.

Figura 47 – Placa condicionadora de sinais.

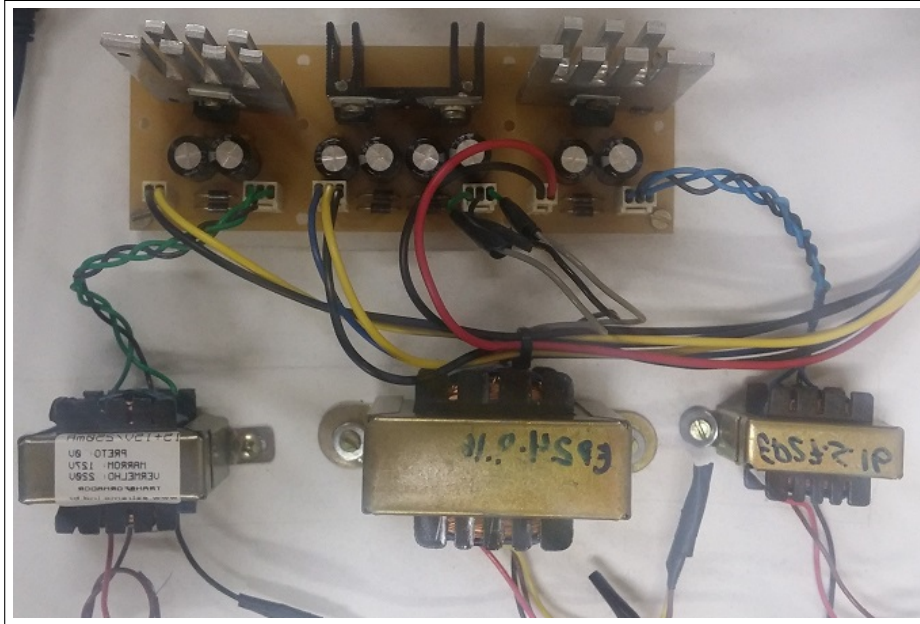


Fonte – O autor.

A.7 Fonte auxiliar

A fonte auxiliar foi desenvolvida e montada em laboratório, capaz de fornecer 4 níveis de tensão e está apresentada na Figura 48.

Figura 48 – Fonte auxiliar desenvolvida em laboratório.



Fonte – O autor.

A fonte desenvolvida é capaz de fornecer 4 níveis de tensão contínua, sendo o nível de +18V utilizado para o chaveamento dos gatilhos do inversor trifásico, apresentado na seção A.2, os níveis de +15V e -15V são utilizados para os sensores de corrente por efeito *hall*, mostrados na seção A.4, e o nível de +5V alimenta a placa de condicionamento de sinais e seus circuitos integrados, apresentada na seção A.6.

ANEXO B – CÓDIGOS PARA SIMULAÇÃO NO *SIMULINK*[®]

B.1 Código implementado no Controlador PI para simulação na malha de corrente no *Simulink*[®]

Código-fonte 1 – Código do controlador PI para simulação da malha de corrente

```

1 function [sys, x0, str, ts] = controlador_PI(t,x,u,flag,ts)
2 persistent Iqref Iqref_ant Iq Erro Erro_ant Erro_ant_ant cont cont_ant Kc Ti
   Td q0 q1 q2 Ts;
3
4 %INICIALIZACAO
5 if flag == 0
6     sizes = simsizes;
7     sizes.NumContStates = 0;
8     sizes.NumDiscStates = 0;
9     sizes.NumOutputs = 1;
10    sizes.NumInputs = 2;
11    sizes.DirFeedthrough = 1;
12    sizes.NumSampleTimes = 1;
13
14    sys = simsizes(sizes);
15
16    x0 = [ ];
17    str = [ ];
18    ts = [0.02 0];
19
20    Erro_ant = 0;
21    Erro_ant_ant = 0;
22    Iqref_ant = 0;
23    cont_ant = 0;
24
25 %CALCULA PROXIMO INSTANTE DE AMOSTRAGEM
26 elseif flag == 4
27     sys=[];
28
29 elseif flag == 3
30     %RECEBE VALORES DE REFERENCIA E FEEDBACK
31     Iqref = u(1);
32     Iq = u(2);
33
34     %PARAMETROS DO CONTROLADOR PI - FORNECIDOS PELO PIDTOOL
35     Kc = 1.36;
36     Ti = 0.38061;
37     Td = 0;

```

```

38
39 %TAXA DE AMOSTRAGEM (MESMA UTILIZADA PELO DSP)
40 Ts = 0.02;
41
42 %DISCRETIZA PARAMETROS DO PID
43 q0 = Kc*(1 + Ts/Ti + Td/Ts);
44 q1 = -Kc*(1 + 2*Td/Ts);
45 q2 = (Kc*(Td/Ts));
46
47 % CALCULO DO ERRO
48 Erro = Iqref - Iq;
49
50 % EXECUTA LEI DE CONTROLE - PID
51 cont = cont_ant + q0*Erro + q1*Erro_ant + q2*Erro_ant_ant;
52
53 %VERIFICA SATURACOES DE CORRENTE
54 if (cont > 1.8)
55     cont = 1.8;
56 elseif (cont < -1.8)
57     cont = -1.8;
58 end
59
60 %ATUALIZA VARIAVEIS
61 Iqref_ant = Iqref;
62 Erro_ant_ant = Erro_ant;
63 Erro_ant = Erro;
64 cont_ant = cont;
65
66 %SAIDA DA FUNCAO
67 out = [cont];
68 sys = out;
69 else
70     sys = [ ];
71 end
72
73 }

```

B.2 Código implementado no Controlador PI para simulação na malha de velocidade no *Simulink*[®]

Código-fonte 2 – Código do controlador PI para simulação da malha de velocidade

```

1 function [sys, x0, str, ts] = controlador_PI(t,x,u,flag,ts)

```

```

2 persistent Speedref Speedref_ant Speedf Erro Erro_ant Erro_ant_ant Iqref
   Iqref_ant Kc Ti Td q0 q1 q2 Ts;
3
4 %INICIALIZACAO
5 if flag == 0
6     sizes = simsizes;
7     sizes.NumContStates = 0;
8     sizes.NumDiscStates = 0;
9     sizes.NumOutputs = 1;
10    sizes.NumInputs = 2;
11    sizes.DirFeedthrough = 1;
12    sizes.NumSampleTimes = 1;
13
14    sys = simsizes(sizes);
15
16    x0 = [ ];
17    str = [ ];
18    ts = [0.3 0];
19
20    Erro_ant = 0;
21    Erro_ant_ant = 0;
22    Speedref_ant = 0;
23    Iqref_ant = 0;
24
25 %CALCULA PROXIMO INSTANTE DE AMOSTRAGEM
26 elseif flag == 4
27     sys=[];
28
29 elseif flag == 3
30     %RECEBE VALORES DE REFERENCIA E FEEDBACK
31     Speedref = u(1);
32     Speedf = u(2);
33
34     %PARAMETROS DO CONTROLADOR PI - FORNECIDOS PELO PIDTOOL
35     Kc = 0.3761;
36     Ti = 0.23877;
37     Td = 0;
38
39     %TAXA DE AMOSTRAGEM (MESMA UTILIZADA PELO DSP)
40     Ts = 0.3;
41
42     %DISCRETIZA PARAMETROS DO PID
43     q0 = Kc*(1 + Ts/Ti + Td/Ts);
44     q1 = -Kc*(1 + 2*Td/Ts);
45     q2 = (Kc*(Td/Ts));
46
47     % CALCULO DO ERRO

```

```

48     Erro = Speedref - Speedf;
49
50     % EXECUTA LEI DE CONTROLE - PID
51     Iqref = Iqref_ant + q0*Erro + q1*Erro_ant + q2*Erro_ant_ant;
52
53     %VERIFICA SATURACOES DE CORRENTE
54     if (Iqref > 1.8)
55         Iqref = 1.8;
56     elseif (Iqref < -1.8)
57         Iqref = -1.8;
58     end
59
60     %ATUALIZA VARIAVEIS
61     Speedref_ant = Speedref;
62     Erro_ant_ant = Erro_ant;
63     Erro_ant = Erro;
64     Iqref_ant = Iqref;
65
66     %SAIDA DA FUNCAO
67     out = [Iqref];
68     sys = out;
69 else
70     sys = [ ]; %n o faz nada
71 end

```

B.3 Código implementado no Controlador RST para simulação na malha de velocidade no *Simulink*[®]

Código-fonte 3 – Código do controlador RST para simulação da malha de velocidade

```

1 function [sys, x0, str, ts] = teste_str_motor(t,x,u,flag,ts) %<---
2 persistent Speedref Speedf Speedf_ant Speedf_ant_ant Iqref Iqref_ant
3     Iqref_ant_ant r1 r2 s0 s1 s3 T;
4
5 %inicializa o
6 if flag == 0
7     sizes = simsizes;
8     sizes.NumContStates = 0;
9     sizes.NumDiscStates = 0;
10    sizes.NumOutputs = 1;
11    sizes.NumInputs = 2;
12    sizes.DirFeedthrough = 1;

```

```

13     sizes.NumSampleTimes = 1;
14
15     sys = simsizes(sizes);
16
17     x0 = [ ];
18     str = [ ];
19     ts = [0.3 0]; %tempo de amostragem variavel
20
21     Speedf_ant = 0;
22     Speedf_ant_ant = 0;
23     Iqref_ant = 0;
24     Iqref_ant_ant = 0;
25
26 %Calcula proximo instante de amostragem
27 elseif flag == 4
28     sys=[];
29
30 elseif flag == 3
31     Speedref = u(1);
32     Speedf = u(2);
33
34 %ALOCACAO DE POLOS
35     qsi = 0.85;
36     wn = 3.2/(qsi*25);
37
38     gz = c2d(tf([wn^2],[1 2*qsi*wn wn^2]),0.3);
39     [num,den] = tfdata(gz, v );
40     bm0 = den(2)+den(3)+1;
41     am1 = den(2); am2 = den(3);
42
43 %PLANTA (Discretizacao = 0.3)
44     b0 = 0.2026; b1 = 0.1902;
45     a1 = -1.821; a2 = 0.8262;
46
47 %PARAMETROS DO CONTROLADOR
48     s0 = (am1 - a1)/b0;
49     s1 = (am2 - a2)/b0;
50     r1 = b1/b0;
51     T = bm0/b0;
52
53 %LEI DE CONTROLE - RST FIXO
54     Iqref = (Speedref)*T - (Speedf)*s0 - (Speedf_ant)*s1 - (Iqref_ant)*r1;
55
56 %SATURADOR
57     if (Iqref > 1.7)
58         Iqref = 1.7;
59     elseif (Iqref < -1.7)

```

```
60     Iqref = -1.7;
61     end
62
63     Speedf_ant_ant = Speedf_ant;
64     Speedf_ant = Speedf;
65     Iqref_ant_ant = Iqref_ant;
66     Iqref_ant = Iqref;
67
68     out = [Iqref];
69
70     sys = out;
71 else
72     sys = [ ]; %n o faz nada
73 end
```


ANEXO C – CÓDIGO IMPLEMENTADO NO DSP

Código-fonte 4 – Código implementado no DSP

```

1 #define GLOBAL_Q      18
2 long GlobalQ = GLOBAL_Q;      // Used for legacy GEL & Graph Debug.
3
4 // Include The Following Definition Files:
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include "IQmathLib.h"
8 #include "DSP2833x_Device.h"
9 #include "DSP2833x_Examples.h"
10
11 //#####
12 //                      VARI VEIS
13 //#####
14 //=====Vari veis Toin=====
15
16 #pragma DATA_SECTION(result1, ".RESULT1")
17 #pragma DATA_SECTION(result2, ".RESULT2")
18 #pragma DATA_SECTION(result3, ".RESULT3")
19 #pragma DATA_SECTION(result4, ".RESULT4")
20
21 long ilaread1, ilaread2, ilaread3, ilaread4, ilaread5, ilaread6, ilaread7, ilaread8;
22 long ilbread1, ilbread2, ilbread3, ilbread4, ilbread5, ilbread6, ilbread7, ilbread8;
23 long ila, ilb, resultado, salva, i, counter1;
24 long Newpos, Oldpos, Dpos, shift, brinca, index;
25
26 _iq18 ADCA=_IQ18(0.0); //ad actual current
27 _iq18 IA=_IQ18(0.0); //real value actual current A [A]
28 _iq18 ADCB=_IQ18(0.0); //ad actual current
29 _iq18 IB=_IQ18(0.0); //real value actual current B [A]
30 _iq18 ADCC=_IQ18(0.0); //ad actual current
31 _iq18 IC=_IQ18(0.0); //real value actual current C [A]
32 _iq18 Ialpha=_IQ18(0.0); //real value actual current Alpha [A]
33 _iq18 Ibeta=_IQ18(0.0); //real value actual current Beta [A]
34 _iq18 Id=_IQ18(0.0); //real value actual current D [A]
35 _iq18 Iq=_IQ18(0.0); //real value actual current Q [A]
36 _iq18 Id_ant=_IQ18(0.0); //real value actual current D [A] anterior
37 _iq18 Iq_ant=_IQ18(0.0); //real value actual current Q [A] anterior
38 _iq18 Umaxq=_IQ18(0.0); //Maximun voltage in UQ [V]
39 _iq18 Sine=_IQ18(0.0); //sine angle format IQ26
40 _iq18 Cosine=_IQ18(0.0); //cosine angle format IQ26
41 _iq18 Sine1=_IQ18(0.0); //sine angle format IQ19
42 _iq18 Cosine1=_IQ18(0.0); //cosine angle format IQ19

```

```

43 _iq18 Angle=_IQ18(0.0); //angle from the sensor used for FOC [rad]
44 _iq18 Angle1=_IQ18(0.0); //previous angle used for FOC [rad]
45 _iq18 Angle2=_IQ18(0.0); //actual calculated angle used for FOC [rad]
46 _iq18 Idref=_IQ18(0.0); //id current reference [A]
47 _iq18 Iqref=_IQ18(0.3); //iq current reference [A]
48 _iq18 ERRORId=_IQ18(0.0); //id current error
49 _iq18 VOLTAGEId=_IQ18(0.0); //actual calculated voltage axis D variable [V]
50 _iq18 VOLTAGEId1=_IQ18(0.0); //last calculated voltage axis D variable [V]
51 _iq18 ERRORIq=_IQ18(0.0); //iq current error
52 _iq18 VOLTAGEIq=_IQ18(0.0); //actual calculated voltage axis Q variable [V]
53 _iq18 VOLTAGEIq1=_IQ18(0.0); //last calculated voltage axis Q variable [V]
54 _iq18 UA=_IQ18(0.0); //calculated ualpha voltage [V]
55 _iq18 UB=_IQ18(0.0); //calculated ubeta voltage [V]
56 _iq18 U1=_IQ18(0.0); //calculated u1 voltage [V]
57 _iq18 U2=_IQ18(0.0); //calculated u2 voltage [V]
58 _iq18 U3=_IQ18(0.0); //calculated u3 voltage [V]
59 _iq18 umin3=_IQ18(0.0); //third harmonic minimum [V]
60 _iq18 umax3=_IQ18(0.0); //third harmonic maximum [V]
61 _iq18 uharm3=_IQ18(0.0); //third harmonic voltage [V]
62 _iq18 Ia=_IQ18(0.0); //real value actual current format IQ26 [A]
63 _iq18 Ib=_IQ18(0.0); //real value actual current format IQ26 [A]
64 _iq18 Ic=_IQ18(0.0); //real value actual current format IQ26 [A]
65 float result1[1001]; //storage 4
66 float result2[1001]; //storage 4
67 float result3[1001]; //storage 4
68 float result4[1001]; //storage 4
69 float Ia_amp;
70 float Ib_amp;
71 float Ic_amp;
72 float SpeedrefRPM;
73 float SpeedfRPM;
74 int uC1, uC2;
75 _iq18 P, P1, P2;
76 float Kp, Ti, Ts, q0, q1;
77 int loop = 0, loopX = 0;
78 _iq18 Speed=_IQ18(0.0); //instantaneous speed [m/s]
79 _iq18 Speedf=_IQ18(0.0); //filtered speed [m/s]
80 _iq18 X=_IQ18(0.0); //absolute position [mm]
81 _iq18 ERRORX=_IQ18(0.0); //absolute position error
82 _iq18 ERRORX_ant=_IQ18(0.0);
83 _iq18 Xref=_IQ18(3.0); //absolute position reference [mm]
84 _iq18 Speedref=_IQ18(0.0); //speed reference [m/s]
85 _iq18 Speedref1=_IQ18(0.0); //last calculated speed reference [m/s]
86 _iq18 ERRORSpeed=_IQ18(0.0); //speed error
87 _iq18 ERRORSpeed_ant=_IQ18(0.0); //speed error
88 _iq18 ERRORSpeed_ant_ant=_IQ18(0.0); //speed error
89 _iq18 Forceref=_IQ18(0.0); //Force reference [N]

```



```

143 void ADC_Init(void);
144 void Xintf(void);
145 void InitEqep(void);
146
147
148 //#####
149 //                MAIN CODE
150
151 //#####
152
153 void main(void)
154 {
155     ilaread1 = 0;ilaread2 = 0;ilaread3 = 0;ilaread4 = 0;ilaread5 = 0;ilaread6 =
        0;ilaread7 = 0;ilaread8 = 0;
156     ilbread1 = 0;ilbread2 = 0;ilbread3 = 0;ilbread4 = 0;ilbread5 = 0;ilbread6 =
        0;ilbread7 = 0;ilbread8 = 0;
157     ila = 0;ilb = 0;resultado = 0;salva = 0;i = 0;counter1 = 0;
158     Newpos = 0;Oldpos = 0;Dpos = 0;shift = 0;brinca = 0;index = 0;
159
160 // Step 1. Initialize System Control:
161     InitSysCtrl(); // PLL, WatchDog, enable Peripheral Clocks
162
163 // Step 2. Clear all interrupts and initialize PIE vector table:
164     DINT; // Disable CPU interrupts
165
166     InitPieCtrl();// Initialize PIE control registers to their default state
167     // The default state is all PIE interrupts disabled and flags
168     // are cleared.
169
170     IER = 0x0000; // Disable CPU interrupts and clear all CPU interrupt flags:
171     IFR = 0x0000;
172
173     InitPieVectTable();// Initialize the PIE vector table with pointers to the
        shell Interrupt
174         // Service Routines (ISR).
175     EnableInterrupts();
176
177     EALLOW;
178     PieVectTable.ADCINT = &adc_isr;
179     PieVectTable.WAKEINT = &lateCachorroFDP;
180     PieVectTable.EPWM4_INT = &epwm4_isr;
181     EDIS;
182
183     EALLOW;
184     SysCtrlRegs.SCSR = BIT1;
185     EDIS;
186

```

```

187     IER|= M_INT1;
188     IER|= M_INT3;
189
190     PieCtrlRegs.PIECTRL.bit.ENPIE = 1; //Enable PIE Block
191     PieCtrlRegs.PIEIER1.bit.INTx6 = 1;
192     PieCtrlRegs.PIEIER1.bit.INTx8 = 1; //Enable PIE interrupt pin WAKEINT
193     PieCtrlRegs.PIEIER3.bit.INTx4 = 1; //Enable EPWM INT4 in the PIE:
        Group 3 interrupt 1-3
194
195     EINT;
196     ERTM;
197 // Enable the watchdog
198     EALLOW;
199     SysCtrlRegs.WDCR = 0x00A8;
200     EDIS;
201
202     InitAdc();
203     InitEPwmGpio();
204     InitEPwm4Gpio();
205     ePwm1Inversor();
206     ePwm2Inversor();
207     ePwm3Inversor();
208     ePwm4_for_interrupt();
209     Xintf();
210     ADC_Init();
211     ServiceDog();
212     InitEQep1Gpio();
213     InitEqep();
214
215     EALLOW;
216     GpioCtrlRegs.GPADIR.bit.GPIO12 = 1;
217     EDIS;
218     GpioDataRegs.GPASET.bit.GPIO12 = 1;
219
220     WakeCount = 0;
221
222     a = _IQ18(3.0);
223
224     while(1)                //infinite loop
225     {
226         wCount++;
227         ServiceDog();
228     }
229 }
230
231 }
232

```

```

233 //#####
234 //          FUN  ES
235 //#####
236 void ADC_Init(void)
237 {
238     // Configure ADC
239     AdcRegs.ADCTRL1.bit.SEQ_CASC = 1; //16 conversoes possiveis modo cascata
240     //AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 0x07;          // Produz 8 conversoes no
                SEQ1 apenas EVA
241     AdcRegs.ADCTRL1.bit.CPS = 1; // pre scaler do clock HSCLOCK/2
242     AdcRegs.ADCMAXCONV.all = 0x0b; //--- habilitando do CONVO0 a CONV11
243
244     AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x0; // Canal ADCINA2 4a conv---SENSOR Ia
245     AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x0; // Canal ADCINA2 5a conv---SENSOR Ia
246     AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x0; // Canal ADCINA2 6a conv---SENSOR Ia
247
248     AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x3; // Canal ADCINA0 7a conv---SENSOR Ib
249     AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x3; // Canal ADCINA0 8a conv---SENSOR Ib
250     AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x3; // Canal ADCINA0 9a conv---SENSOR Ib
251
252     AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 0x5; // Canal ADCINA3 10a conv---SENSOR
                Ic
253     AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 0x5; // Canal ADCINA0 11a conv---SENSOR
                Ic
254     AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 0x5; // Canal ADCINA7 12a conv---SENSOR
                Ic
255
256     // Assumes ePWM4 clock
257     EPwm4Regs.ETSEL.bit.SOCAEN = 1;          // Enable SOC on A group
258     EPwm4Regs.ETSEL.bit.SOCASEL = 1;        // Select SOC from from CPMA on
                upcount
259     AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1 = 1;
260     AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1;
261     AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ = 0;
262     EPwm4Regs.ETFLG.bit.SOCA = 1;
263     EPwm4Regs.ETPS.bit.SOCAPRD = 1;          // Generate pulse on 1st event
264 }
265
266 void InitEqep(void)
267 {
268     EQep1Regs.QUPRD=1500000;          // Unit Timer for 100Hz at 150 MHz SYSCLKOUT
269
270     EQep1Regs.QDECCTL.bit.QSRC=00;      // QEP quadrature count mode
271
272     EQep1Regs.QEPCTL.bit.FREE_SOFT=2;
273     EQep1Regs.QEPCTL.bit.PCRM=01;      // PCRM=01 mode - QPOSCNT reset on maximum
                position

```

```

274 EQep1Regs.QEPCTL.bit.UTE=1;      // Unit Timeout Enable
275 EQep1Regs.QEPCTL.bit.QCLM=1;     // Latch on unit time out
276 EQep1Regs.QPOSMAX=0x7ff;        //0x7ff = 2047 contagens
277 EQep1Regs.QEPCTL.bit.QPEN=1;     // QEP enable
278 EQep1Regs.QDECCTL.bit.SWAP=0;    // troca o sentido da contagem;
279
280 EQep1Regs.QCAPCTL.bit.UPPS=5;    // 1/32 for unit position
281 EQep1Regs.QCAPCTL.bit.CCPS=7;    // 1/128 for CAP clock
282 EQep1Regs.QCAPCTL.bit.CEN=1;     // QEP Capture Enable
283 }
284
285 void ePwm4_for_interrupt(void)
286 {
287     //Frequencia de 10.0 kHz
288     // Setup TBCLK
289     EPwm4Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Count up
290     EPwm4Regs.TBPRD = 1874;          // Set timer period
291     EPwm4Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Disable phase loading
292     EPwm4Regs.TBPHS.half.TBPHS = 0x0000; // Phase is 0
293     EPwm4Regs.TBCTR = 0x0000;       // Clear counter
294     EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0x4; // HSPCLKDIV => /8
295     EPwm4Regs.TBCTL.bit.CLKDIV = 0x0; // CLKDIV => /1
296
297     // Setup shadow register load on ZERO
298     EPwm4Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
299     EPwm4Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
300     EPwm4Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
301     EPwm4Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;
302
303     // Set Compare values
304     EPwm4Regs.CMPA.half.CMPA = 415; // Set compare A value
305     EPwm4Regs.CMPB = 415;          // Set Compare B value
306
307     // Set actions
308     EPwm4Regs.AQCTLA.bit.ZRO = AQ_SET; // Set PWM1A on Zero
309     EPwm4Regs.AQCTLA.bit.CAU = AQ_CLEAR; // Clear PWM1A on event A,
        up count
310
311     EPwm4Regs.AQCTLB.bit.ZRO = AQ_SET; // Set PWM1B on Zero
312     EPwm4Regs.AQCTLB.bit.CBU = AQ_CLEAR; // Clear PWM1B on event B,
        up count
313
314     // Interrupt where we will change the Compare Values
315     EPwm4Regs.ETSEL.bit.INTSEL = ET_CTR_ZERO; // Select INT on Zero event
316     EPwm4Regs.ETSEL.bit.INTEN = 1;          // Enable INT
317     EPwm4Regs.ETPS.bit.INTPRD = ET_3RD;     // Generate INT on 3rd event
318 }

```



```
319
320
321 void ePwm1Inversor(void)
322 {
323     EPwm1Regs.TBPRD = 3750;                // Set timer period
324     EPwm1Regs.TBPHS.half.TBPHS = 0x0000;  // Phase is 0
325     EPwm1Regs.TBCTR = 0x0000;            // Clear counter
326
327
328     // Setup TBCLK
329     EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up
330     EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;        // Disable phase loading
331     EPwm1Regs.TBCTL.bit.HSPCLKDIV = 1;
332     EPwm1Regs.TBCTL.bit.CLKDIV = 0;
333
334     // Setup compare
335     EPwm1Regs.CMPA.half.CMPA = 156;
336
337     // Set actions
338     EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;            // Set PWM3A on Zero
339     EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
340
341     EPwm1Regs.AQCTLB.bit.CAU = AQ_CLEAR;         // Set PWM3A on Zero
342     EPwm1Regs.AQCTLB.bit.CAD = AQ_SET;
343
344     // Active high complementary PWMs - Setup the deadband
345     EPwm1Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
346     EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
347     EPwm1Regs.DBCTL.bit.IN_MODE = DBA_ALL;
348     EPwm1Regs.DBRED = 0x0050;
349     EPwm1Regs.DBFED = 0x0050;
350 }
351
352 void ePwm2Inversor(void)
353 {
354     EPwm2Regs.TBPRD = 3750;                // Set timer period
355     EPwm2Regs.TBPHS.half.TBPHS = 0x0000;  // Phase is 0
356     EPwm2Regs.TBCTR = 0x0000;            // Clear counter
357
358
359     // Setup TBCLK
360     EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up
361     EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE;        // Disable phase loading
362     EPwm2Regs.TBCTL.bit.HSPCLKDIV = 1;
363     EPwm2Regs.TBCTL.bit.CLKDIV = 0;
364
365     // Setup compare
366     EPwm2Regs.CMPA.half.CMPA = 52;
```

```

366
367 // Set actions
368 EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // Set PWM3A on Zero
369 EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
370
371
372 EPwm2Regs.AQCTLB.bit.CAU = AQ_CLEAR; // Set PWM3A on Zero
373 EPwm2Regs.AQCTLB.bit.CAD = AQ_SET;
374
375 // Active high complementary PWMs - Setup the deadband
376 EPwm2Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
377 EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
378 EPwm2Regs.DBCTL.bit.IN_MODE = DBA_ALL;
379 EPwm2Regs.DBRED = 0x0050;
380 EPwm2Regs.DBFED = 0x0050;
381 }
382
383 void ePwm3Inversor(void)
384 {
385 EPwm3Regs.TBPRD = 3750; // Set timer period
386 EPwm3Regs.TBPHS.half.TBPHS = 0x0000; // Phase is 0
387 EPwm3Regs.TBCTR = 0x0000; // Clear counter
388
389 // Setup TBCLK
390 EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up
391 EPwm3Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Disable phase loading
392 EPwm3Regs.TBCTL.bit.HSPCLKDIV = 1;
393 EPwm3Regs.TBCTL.bit.CLKDIV = 0;
394
395 // Setup compare
396 EPwm3Regs.CMPA.half.CMPA = 104;
397
398 // Set actions
399 EPwm3Regs.AQCTLA.bit.CAU = AQ_SET; // Set PWM3A on Zero
400 EPwm3Regs.AQCTLA.bit.CAD = AQ_CLEAR;
401
402 EPwm3Regs.AQCTLB.bit.CAU = AQ_CLEAR; // Set PWM3A on Zero
403 EPwm3Regs.AQCTLB.bit.CAD = AQ_SET;
404
405 // Active high complementary PWMs - Setup the deadband
406 EPwm3Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
407 EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC;
408 EPwm3Regs.DBCTL.bit.IN_MODE = DBA_ALL;
409 EPwm3Regs.DBRED = 0x0050;
410 EPwm3Regs.DBFED = 0x0050;
411 }
412

```

```

413 void Xintf(void)
414 {
415     // All Zones-----
416     // Timing for all zones based on XTIMCLK = SYSCLKOUT
417     XintfRegs.XINTCNF2.bit.XTIMCLK = 0;
418     XintfRegs.XINTCNF2.bit.WRBUFF = 3; // Buffer up to 3 writes
419     XintfRegs.XINTCNF2.bit.CLKOFF = 0; // XCLKOUT is enabled
420     XintfRegs.XINTCNF2.bit.CLKMODE = 0; // XCLKOUT = XTIMCLK
421
422     // Zone 6-----
423     // When using ready, ACTIVE must be 1 or greater
424     // Lead must always be 1 or greater
425     // Zone write timing
426     XintfRegs.XTIMING6.bit.XWRLEAD = 1;
427     XintfRegs.XTIMING6.bit.XWRACTIVE = 1;
428     XintfRegs.XTIMING6.bit.XWRTRAIL = 1;
429     // Zone read timing
430     XintfRegs.XTIMING6.bit.XRDLEAD = 1;
431     XintfRegs.XTIMING6.bit.XRDACTIVE = 2;
432     XintfRegs.XTIMING6.bit.XRDTRAIL = 0;
433     // Do not double all Zone read/write lead/active/trail timing
434     XintfRegs.XTIMING6.bit.X2TIMING = 0;
435     // Zone will not sample READY
436     XintfRegs.XTIMING6.bit.USEREADY = 0;
437     XintfRegs.XTIMING6.bit.READYMODE = 0;
438     // Size must be 1,1 - other values are reserved
439     XintfRegs.XTIMING6.bit.XSIZE = 3;
440     // Zone 7-----
441     // When using ready, ACTIVE must be 1 or greater
442     // Lead must always be 1 or greater
443     // Zone write timing
444     XintfRegs.XTIMING7.bit.XWRLEAD = 1;
445     XintfRegs.XTIMING7.bit.XWRACTIVE = 1;
446     XintfRegs.XTIMING7.bit.XWRTRAIL = 1;
447     // Zone read timing
448     XintfRegs.XTIMING7.bit.XRDLEAD = 1;
449     XintfRegs.XTIMING7.bit.XRDACTIVE = 2;
450     XintfRegs.XTIMING7.bit.XRDTRAIL = 0;
451     // don t double all Zone read/write lead/active/trail timing
452     XintfRegs.XTIMING7.bit.X2TIMING = 0;
453     // Zone will not sample XREADY signal
454     XintfRegs.XTIMING7.bit.USEREADY = 0;
455     XintfRegs.XTIMING7.bit.READYMODE = 0;
456     // Size must be 1,1 - other values are reserved
457     XintfRegs.XTIMING7.bit.XSIZE = 3;
458     //Force a pipeline flush to ensure that the write to
459     //the last register configured occurs before returning.

```

```

460     asm(" RPT #7 || NOP");
461 }
462
463
464 //#####
465 //                INTERRUPT  0
466 //#####
467
468 interrupt void adc_isr(void)
469 {
470     // Clear INT flag for this timer
471     EPwm4Regs.ETCLR.bit.INT = 1;
472
473     GpioDataRegs.GPATOGGLE.bit.GPIO12 = 1;
474
475     iCount = iCount + 1;
476
477     // Reinitialize for next ADC sequence
478     AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 1;           // Reset SEQ1
479     AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;         // Clear INT SEQ1 bit
480     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;    // Acknowledge interrupt to PIE
481
482     //EQep1Regs.QPOSCNT - POSI  0 eQep
483
484 //=====
485     IA = (AdcRegs.ADCRESULT3>>4);
486     IA = (IA+(AdcRegs.ADCRESULT4>>4));
487     IA = (IA+(AdcRegs.ADCRESULT5>>4));
488     IA = IA/3;
489     Ia = IA*262144;
490     Ia = _IQ18mpy(_IQ18(2477)-Ia,_IQ18(0.001077685));
491     Ia = -Ia;
492     Ia_amp = Ia / 262144.0;
493
494     IB = (AdcRegs.ADCRESULT0>>4);
495     IB = (IB+(AdcRegs.ADCRESULT1>>4));
496     IB = (IB+(AdcRegs.ADCRESULT2>>4));
497     IB = IB/3;
498     Ib = IB*262144;
499     Ib = _IQ18mpy(_IQ18(2595)-Ib,_IQ18(0.000981689));
500     Ib = -Ib;
501     Ib_amp = Ib / 262144.0;
502
503     IC = (AdcRegs.ADCRESULT9>>4);
504     IC = (IC+(AdcRegs.ADCRESULT10>>4));
505     IC = (IC+(AdcRegs.ADCRESULT11>>4));
506     IC = IC/3;

```

```

507 Ic = IC*262144;
508   Ic = _IQ18mpy(_IQ18(2437)-Ic,_IQ18(0.001005607));
509   Ic = -Ic;
510 Ic_amp = Ic / 262144.0;
511
512 //*****
513 //READ SPEED VALUES
514 //*****
515 rotorPosition_ant_ant = rotorPosition_ant;
516 rotorPosition_ant = rotorPosition;
517
518 rotorPosition = _IQ18mpy(_IQ18(EQep1Regs.QPOSCNT),_IQ18div(_IQ18(6.28318),
   _IQ18(2048)));
519 rotorPositionF = _IQ18toF(rotorPosition);
520
521 Dpos = rotorPosition - rotorPosition_ant; //toim
522 if (Dpos>0)
523 {
524     if (rotorPosition<rotorPosition_ant) Dpos=_IQ18(6.28)+rotorPosition -
        rotorPosition_ant;
525 }
526 if (Dpos<0)
527 {
528     if (rotorPosition>rotorPosition_ant) Dpos=_IQ18(-6.28)+rotorPosition -
        rotorPosition_ant;
529 }
530
531 Speed = _IQ18mpy(Dpos, _IQ18(10000));
532 Speedf = Speedf + _IQ18mpy((Speed - Speedf), _IQ18(0.0001));
533
534 //Velocidade lida em RPM
535 SpeedfRPM = _IQ18mpy(Speedf, _IQ18(23.333));
536
537 //*****
538 // NGULO POSI O EL TRICO
539 //*****
540 rotorPositioneetrico = _IQ18mpy(rotorPosition,_IQ18(2.0));
541
542 if(rotorPositioneetrico > _IQ18(6.28))
543 {
544     rotorPositioneetrico = rotorPositioneetrico - _IQ18(6.28);
545 }
546 if(rotorPositioneetrico < _IQ18(-6.28))
547 {
548     rotorPositioneetrico = rotorPositioneetrico + _IQ18(6.28);
549 }
550

```

```

551 //*****
552 // POSITION CONTROLLER
553 //*****
554 Xref=_IQ18(degrefPOS2_4[brinca]) ; // vetor de referencia
555
556 //pid
557     pos_ref_ant_ant=pos_ref_ant;
558     pos_ref_ant=Xref;
559     ERRORX = Xref - rotorPosition;
560
561     P = 0.63;
562     Speedref = Speedref_ant + _IQ18mpy(_IQ18(P),ERRORX) + _IQ18mpy(_IQ18(-
        P),ERRORX_ant);
563     ERRORX_ant = ERRORX;
564
565     if (Speedref > _IQ18(30)) Speedref = _IQ18(30);
566     if (Speedref < _IQ18(-30)) Speedref = _IQ18(-30);
567
568 Speedref_ant_ant=Speedref_ant;
569 Speedref_ant=Speedref;
570 Speedref1 = Speedref1 + _IQ18mpy((Speedref - Speedref1), _IQ18(0.15));
571
572 //*****
573 // SPEED CONTROLLER
574 //*****
575
576 Kp = 0.3761;
577 Ti = 0.23877;
578 Ts = 0.0001;
579 q0 = Kp*(1 + Ts/Ti + 0/Ts);
580 q1 = -Kp*(1 + 2*0/Ts);
581
582 loop++;
583 //CONTROLADOR PI
584 if (loop==200)
585 {
586     ERRORSpeed = Speedref - Speedf;
587     Iqref = Iqref_ant + _IQ18mpy(_IQ18(q0),ERRORSpeed) + _IQ18mpy(_IQ18(q1),
        ERRORSpeed_ant) + _IQ18mpy(_IQ18(0.0),ERRORSpeed_ant_ant);
588     loop=0;
589 }
590
591 if (Iqref > _IQ18(1.8)) Iqref = _IQ18(1.8);
592 if (Iqref < _IQ18(-1.8)) Iqref = _IQ18(-1.8);
593
594 Speed_ant_ant = Speed_ant;
595 Speed_ant = Speed;

```

```

596
597 Speedf_ant_ant = Speedf_ant;
598 Speedf_ant = Speedf;
599
600 Iqref_ant_ant=Iqref_ant;
601 Iqref_ant=Iqref;
602
603 ERRORSpeed_ant_ant = ERRORSpeed_ant;
604 ERRORSpeed_ant = ERRORSpeed;
605
606 Iq1 = _IQ18mpy(_IQ18(0.98),Iq1) + _IQ18mpy(_IQ18(0.02),Speedref1);
607
608 //*****
609 //   ABC TO ALPHA, BETA
610 //*****
611
612 Ialpha=Ia;
613   Ibeta=_IQ18mpy((Ia+_IQ18mpy(_IQ18(2.0),Ib)), _IQ18(0.57735)); //Ibeta=1/
        sqrt(3)*(Ia+2*Ib)
614
615 //*****
616 //   ALPHA,BETA TO D,Q
617 //*****
618
619 Sine = _IQ18sin(Angle1); //change Angle1(IQ26) to global IQ26 and
        determine the sine value
620 Cosine = _IQ18cos(Angle1); //change Angle1(IQ26) to global IQ26 and
        determine the cosine value
621
622 Id=_IQ18mpy(Ialpha, Cosine) + _IQ18mpy(Ibeta,Sine);
623 Iq=_IQ18mpy(Ibeta, Cosine) - _IQ18mpy(Ialpha,Sine);
624
625 //FILTRO DE CORRENTE ID E IQ
626 Id_ant = Id;
627
628 Iq = Iq_ant + _IQ18mpy(Iq-Iq_ant, _IQ18(0.000133));
629 Iq_ant = Iq;
630
631 //*****
632 //   ROTOR FLUX SPEED AND ANGLE
633 //*****
634
635 Im=Im+_IQ18mpy(_IQ18(0.01),(Id-Im)); //Magnetizing current Im=(1/10000)/(L2
        /R2)*(Id-Im) [A]
636   Vslip=_IQ18div(Iq,Id); //slip speed vslip=(R2/L2)*(tau[m]/pi)*(Iq/Im) [
        rad/s]
637 Vslip = _IQ18mpy(_IQ18(50), Vslip);

```

```

638 Angleslip=Angleslip + _IQ18mpy(Vslip,_IQ18(0.0001)); //Slip Angle (1/10000)
      *(2*pi/2*tau[m]) transform the position in electrical angle
639
640 if (Angleslip > _IQ18(6.283185))      Angleslip = Angleslip - _IQ18
      (6.283185);
641 if (Angleslip < _IQ18(-6.283185))    Angleslip = Angleslip + _IQ18
      (6.283185);
642
643 // Rotor flux position tobias
644 Angle2 =rotorPositioneetrico+Angleslip;
645
646 if (Angle2 > _IQ18(6.283185))      Angle2 = Angle2 - _IQ18(6.283185);
647 if (Angle2 < _IQ18(-6.283185))    Angle2 = Angle2 + _IQ18(6.283185);
648
649 //*****
650 //      CURRENT CONTROL
651 //*****
652 Idref=_IQ18(1.3);
653
654 //*****
655 //      CURRENT ERROR ID
656 //*****
657
658 ERRORId=Idref - Id;      //calculate an error
659 VOLTAGEId = _IQ18mpy(ERRORId, _IQ18(1500)) + VOLTAGEId1;
660
661 if(VOLTAGEId > _IQ18(100)) VOLTAGEId = _IQ18(100); //Vdc/sqrt(3)*(1-(3,2
      uS/100uS)) maximum output voltage
662 if(VOLTAGEId < _IQ18(-100)) VOLTAGEId = _IQ18(-100); //limits
663
664 VOLTAGEId1 = _IQ18mpy(VOLTAGEId1, _IQ18(0.96)) + _IQ18mpy(VOLTAGEId, _IQ18
      (0.04));//Synchronous (0.9787234)(0.0212766) Induction last value
665
666 //*****
667 //      CURRENT ERROR IQ
668 //*****
669
670 ERRORIq=Iqref - Iq;      //calculate an error
671 VOLTAGEIq = _IQ18mpy(ERRORIq, _IQ18(1500)) + VOLTAGEIq1;
672 Umaxq = _IQ18mpy(_IQ18sqrt(_IQ18(120) + VOLTAGEId),_IQ18sqrt(_IQ18(120) -
      VOLTAGEId));//limit the phasors inside de circle sqrt(100^2-VOLTAGEID^2)
673
674 if(VOLTAGEIq > Umaxq) VOLTAGEIq = Umaxq;
675 if(VOLTAGEIq < -Umaxq) VOLTAGEIq = -Umaxq; //limits
676
677 VOLTAGEIq1 = _IQ18mpy(VOLTAGEIq1, _IQ18(0.96)) + _IQ18mpy(VOLTAGEIq, _IQ18
      (0.04));//Synchronous (0.9787234)(0.0212766) Induction last value

```



```

678
679 //*****
680 //   D,Q TO ALPHA,BETA
681 //*****
682 Sine1 = _IQ18sin(Angle2);    //change Sine(IQ26) to IQ19
683 Cosine1 = _IQ18cos(Angle2); //change Cosine(IQ26) to IQ19
684
685 UA = _IQ18mpy(VOLTAGEId,Cosine1) - _IQ18mpy(VOLTAGEIq,Sine1);
686 UB = _IQ18mpy(VOLTAGEId,Sine1) + _IQ18mpy(VOLTAGEIq,Cosine1);
687
688 //*****
689 //   ALPHA,BETA TO A,B,C
690 //*****
691
692 U1 = UA;
693 U2 = _IQ18mpy(UA ,_IQ18(-0.5)) + _IQ18mpy(UB ,_IQ18(0.866));
694 U3 = _IQ18mpy(UA ,_IQ18(-0.5)) - _IQ18mpy(UB ,_IQ18(0.866));
695
696 //*****
697 //       THIRD HARMONIC
698 //*****
699
700 if(U1 >= U2) umax3 = U1;
701 else          umax3 = U2;
702 if(umax3 <= U3) umax3 = U3;
703
704 if(U1 <= U2) umin3 = U1;
705 else          umin3 = U2;
706 if(umin3 >= U3) umin3 = U3;
707
708 uharm3 = _IQ18mpy(umin3 + umax3, _IQ18(-0.5)); // Raumzeigermodulation
        method
709
710 //*****
711 //   COMPARE VALUES
712 //*****
713
714 EPwm1Regs.CMPA.half.CMPA = 1875 - _IQ18int(_IQ18mpy(U1 + uharm3, _IQ18
        (7.5))); // (1875)/250 (half of the timer period/(maximum voltage of
        the triangular))...TPWM/(UDC/2)
715 resultado=U1 + uharm3;
716 EPwm2Regs.CMPA.half.CMPA = 1875 - _IQ18int(_IQ18mpy(U2 + uharm3, _IQ18
        (7.5)));
717 EPwm3Regs.CMPA.half.CMPA = 1875 - _IQ18int(_IQ18mpy(U3 + uharm3, _IQ18
        (7.5)));
718
719 //*****

```

```

720 //   OVERCURRENT TEST
721 //*****
722
723 if (IA> 524288 || IA< -524288 || IB> 524288 || IB< -524288 || IC> 524288 ||
      IC < -524288 )//if the dc link is in overcurrent
724 {
725     EPwm1Regs.CMPA.half.CMPA = 0;
726     EPwm2Regs.CMPA.half.CMPA = 0;
727     EPwm3Regs.CMPA.half.CMPA = 0;
728     // EvaRegs.ACTRA.all=0;
729
730     while(1)                //infinite loop
731     {}
732 }
733
734 //*****
735 //   STORAGE VALUES
736 //*****
737
738 Angle1= Angle2;
739 if (salva>200)
740 {
741     if (brinca<1001)
742     {
743
744         result1[brinca] = _IQ18toF(Id);
745         result2[brinca] = _IQ18toF(Iq);
746         result3[brinca] = _IQ18toF(Xref);
747         result4[brinca] = _IQ18toF(rotorPosition);
748
749         brinca++;
750         salva=0;
751     }
752 }
753     salva++;
754     i++;
755
756     return;
757 }
758
759 //*****
760 //           INTERRUPT   ES   SECUND   RIAS
761 //*****
762
763 interrupt void lateCachorroFDP(void)
764 {
765     WakeCount++;

```

```
766 // Acknowledge this interrupt to get more from group 1
767 //   PieCtrlRegs.PIEACK.bit.ACK1 = 1;
768   if(WakeCount > 1)
769   {
770     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
771   }
772 //   PieCtrlRegs.PIEIFR1.bit.INTx8 = 1;
773 }
774
775 interrupt void epwm4_isr(void)
776 {
777     Loop1++;
778     // Clear INT flag for this timer
779     EPwm4Regs.ETCLR.bit.INT = 1;
780     // Acknowledge this interrupt to receive more interrupts from group 3
781     PieCtrlRegs.PIEACK.bit.ACK3 = 1;
782     return;
783 }
```

ANEXO D – CÓDIGO PARA GERAR SINAL PRBS

Código-fonte 5 – Código do *Matlab*[®] para geração do sinal PRBS.

```

1  clear; clc;
2
3  Tb = 0.28; %Tempo entre bits
4  n = 7; %Numero de bits
5  Vmais = 1.4; %Amplitude do sinal
6  Vmenos = 0.8; %Amplitude do sinal
7  N = n;
8
9  %% GERA SINAL PRBS
10 init = randi([0 1],1,n);
11 z=init;
12 n=length(init);
13 for i=(n+1):(2^n-1)
14     q = z(i-n);
15     q = xor(q,z(i-1));
16     z = [z q];
17 end
18
19 %% APLICA O TEMPO ENTRE BITS (Tb)
20 Tb = Tb/0.02;
21 k = 0;
22 for i=1:length(z)
23     for j=1:Tb
24         prbs(j+k) = z(i);
25     end
26     k = k+floor(Tb);
27 end
28
29 %% REPETE SINAL PRBS AO LONGO DE 1001 AMOSTRAS
30 prbs = [prbs prbs prbs prbs];
31 prbs = prbs(1:1001);
32
33 %% ADICIONA SINAL DE ENTRADA AO PRBS
34 for i=1:1001
35     if (prbs(i) == 1)
36         prbs(i) = Vmais;
37     else
38         prbs(i) = Vmenos;
39     end
40 end

```

ANEXO E – MÉTODO DE CALIBRAÇÃO DOS SENSORES DE CORRENTE

Para realizar a calibração dos sensores de corrente utilizou-se uma fonte em modo corrente, onde aplicou-se quatro valores constantes, tais como $-2A$, $-1A$, $0A$, $1A$ e $2A$, de modo a se verificar a tensão fornecida pelo sensor ao DSP, observando-se que, na maior corrente, não seja ultrapassado o valor de $3,3V$, valor máximo suportado pela entrada analógica do DSP, e que em $0A$ a tensão seja aproximadamente $1,65V$, a fim de que valores acima desta tensão caracterizem correntes positivas e valores abaixo, correntes negativas.

Após realizar os ajustes de tensão fornecidas pelos sensores para cada corrente aplicada, realizou-se a leitura do registrador do conversor A/D do DSP para cada valor de corrente aplicado.

Os valores de corrente aplicados, registrador do conversor A/D e os níveis de tensão fornecidos pelo sensor são mostrados na Tabela 19.

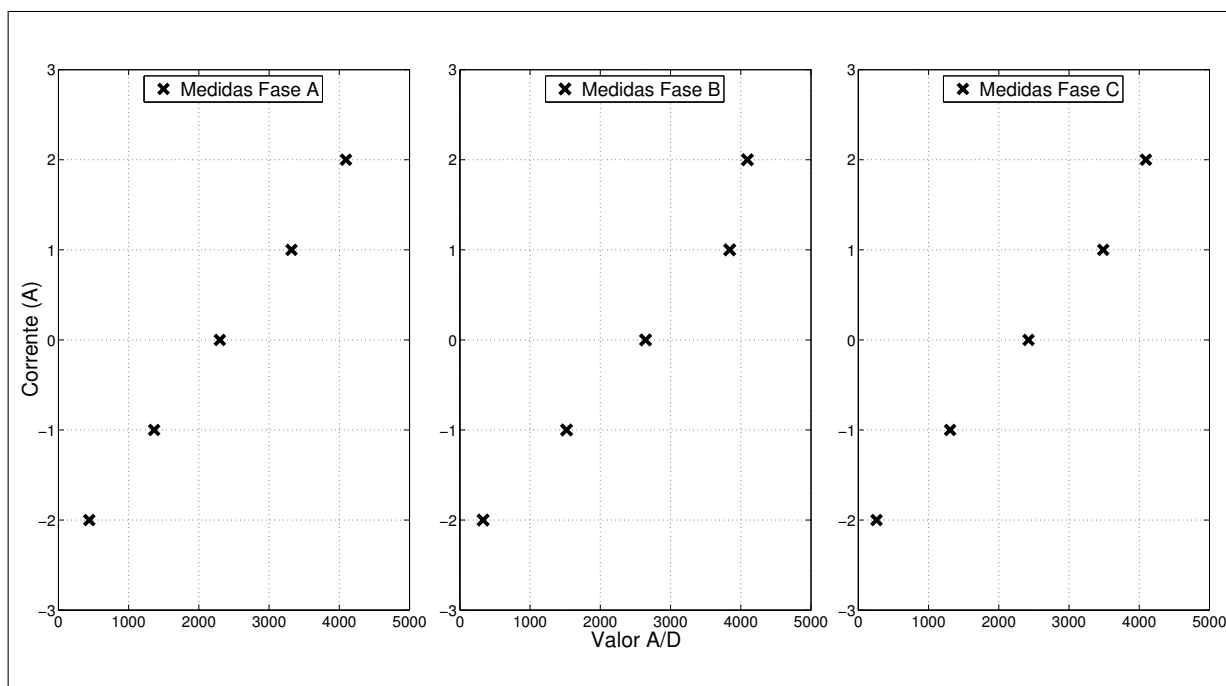
Tabela 19 – Correntes constantes aplicadas aos sensores de corrente e valores obtidos.

Corrente (A)	Fase A		Fase B		Fase C	
	Valor A/D	Tensão (V)	Valor A/D	Tensão (V)	Valor A/D	Tensão (V)
2	4095	3,07	4095	3,60	4095	3,32
1	3320	2,41	3843	2,83	3487	2,58
0	2299	1,72	2644	2,01	2424	1,78
-1	1364	1,01	1520	1,16	1307	0,96
-2	440	0,34	330	0,34	259	0,18

Fonte – O autor.

Colocando-se os dados medidos em formato de gráfico, onde tem-se no eixo y a corrente aplicada e no eixo x o valor do conversor A/D para cada fase, tem-se a Figura 49 representada abaixo.

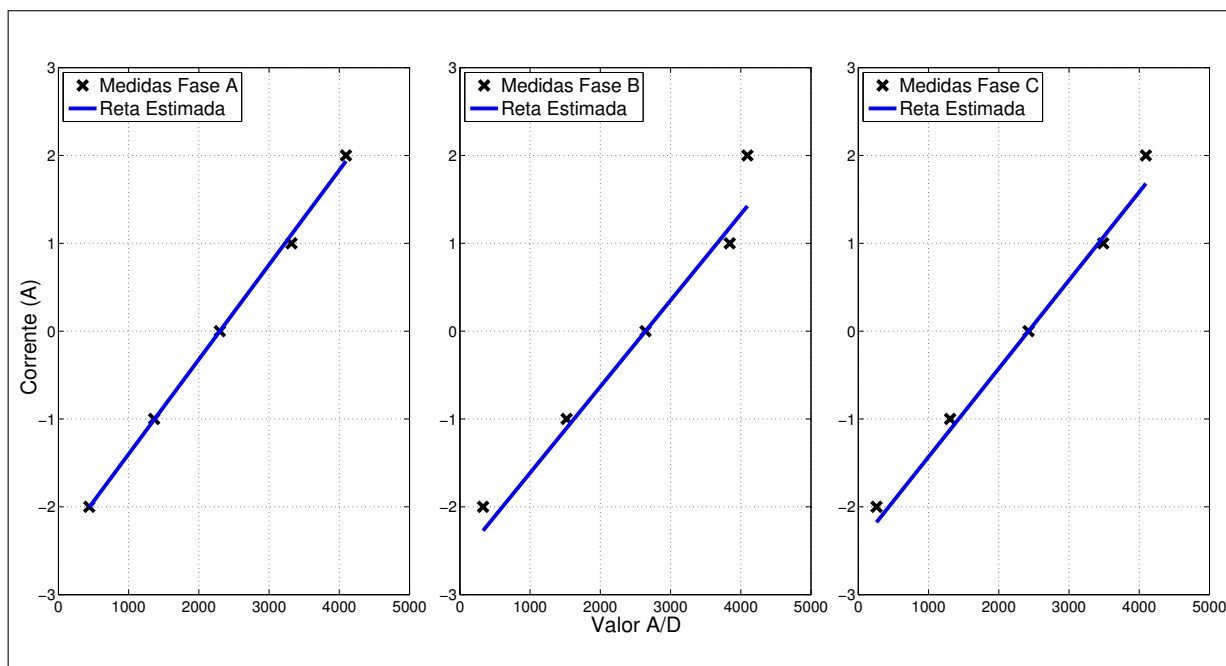
Figura 49 – Pontos obtidos para calibração dos sensores de corrente.



Fonte – O autor.

Aplicando-se o método dos mínimos quadrados para estimar uma reta, no formato $y = ax + b$, que melhor representa os pontos medidos experimentalmente, tem-se a Figura 50, onde apresenta a reta estimada para cada sensor.

Figura 50 – Reta estimada através do métodos dos mínimos quadrados para cada fase.



Fonte – O autor.

O algoritmo utilizado para a estimação da reta através do método dos mínimos quadrados é detalhado abaixo.

Código-fonte 6 – Código do *Matlab*[®] para calibração dos sensores de corrente.

```

1  clc; clear all; format long;
2
3  %Dados de corrente aplicado
4  y = [-2 -1 0 1 2] ;
5
6  %Valores do conversor A/D para cada corrente aplicada
7  xA = [440 1364 2299 3320 4095] ;
8  xB = [330 1520 2644 3843 4095] ;
9  xC = [259 1307 2424 3487 4095] ;
10
11 %y = ax+b
12 AA = [xA xA.^0];
13 AB = [xB xB.^0];
14 AC = [xC xC.^0];
15 thetaA = inv(AA * AA)*AA * y
16 thetaB = inv(AB * AB)*AB * y
17 thetaC = inv(AC * AC)*AC * y
18
19 %Valor estimado
20 yA_hat = AA*thetaA;
21 yB_hat = AB*thetaB;
22 yC_hat = AC*thetaC;
23
24 %Plot
25 subplot(1,3,1);
26 plot(xA,y, * )
27 hold on
28 plot(xA,yA_hat)
29
30 subplot(1,3,2);
31 plot(xB,y, * )
32 hold on
33 plot(xB,yB_hat)
34
35 subplot(1,3,3);
36 plot(xC,y, * )
37 hold on
38 plot(xC,yC_hat)
39
40 format short;

```

Através dos valores estimados utilizando-se o método dos mínimos quadrados, pode-se montar a equação da reta estimada para os sensores de corrente, dadas por:

$$\begin{aligned}\hat{I}_a &= 0,0010776853n_a - 2,4825558965 \\ \hat{I}_b &= 0,0009816888n_b - 2,4408711267 \\ \hat{I}_c &= 0,0010056075n_c - 2,3273779432\end{aligned}\tag{E.1}$$

onde n_a , n_b e n_c representam, respectivamente, os valores dos registradores dos conversores A/D do DSP.