UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

*Estratégias de Modelagem Dinâmica e
Simulação Computacional do
Motor de Indução Trifásico*

MARCELO MACHADO CAD

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Mestre em Engenharia Elétrica.

ORIENTADOR:

Prof. Dr. Manoel Luís de Aguiar

São Carlos
2000

DEDICATÓRIA

À Edna Borges Cortes, que foi o impulso, força e inspiração para a realização desta etapa, aos meus pais, meus grandes mestres na escola da vida e a minha afilhada Fernanda Cad.

AGRADECIMENTOS

Ao Professor Dr. Ing Manoel Luís de Aguiar pela excelente orientação fornecida e a amizade construída durante a elaboração deste trabalho.

Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior – CAPES, pela bolsa de estudo concedida.

A Nacibi Cad e Regina Maria Machado Cad, meus pais, por tudo que fizeram e representam para mim.

A Fernanda de Oliveira, Patrícia Paesani, Melisa, Dirlane, Marisi, Selma, Lillian, Rúbia, Sandra, Gislaine, Maristela, Renata Macedo, Patrícia Leite, Patrícia Mara, Marínes, Ana Luíza, Regina, Andressa, Camila, Fernando Carlos, Tibiriçá, Alex Fabiano, Josemar dos Santos, Ricardo Silveira, Fabiano e Fernando Scramim, Gilmar, Luciano Belluzzo, Randal Farago, Régis Fazio, Fábio Lima, José Roberto, Wilians, Renan Giovanini, Donato, Wilson, Fábio Costa, Edmárcio, Renato Guedes, Silvio Araújo, Azauri, Diógenes, Renato Rosa, Fabrício, Marcelo Magalhães pela amizade construída e ajuda atribuída sempre, e a meus parentes.

A Elia Matsumoto da Opencadd Computação Gráfica pela ajuda atribuída sempre que necessário.

A todos os colegas, professores e funcionários do Departamento de Engenharia Elétrica e demais departamentos da USP/São Carlos pela amizade.

E a todos que de alguma forma contribuíram para que este trabalho acontecesse.

Sumário

LISTA DE FIGURAS	vii
LISTA DE TABELAS	xii
SÍMBOLOS E NOTAÇÕES	xiii
RESUMO	xv
ABSTRACT.....	xvi
CAPÍTULO 1 - INTRODUÇÃO	1
1.1. Organização do Trabalho	3
CAPÍTULO 2 - MODELAGEM MATEMÁTICA DO MOTOR DE INDUÇÃO TRIFÁSICO.....	5
2.1. Introdução	5
2.2. Procedimentos de Modelagem do Motor de Indução Trifásico	7
2.3. Notação Matricial Trifásica	10
2.3.1. Equações de Tensão em um Circuito Resistivo-Indutivo Acoplado Magneticamente.....	10
2.3.2. Equações do Fluxo Concatenado	11
2.3.3. Transposição para o Referencial Único	12
2.3.4. Equações de Conjugado Elétrico e de Velocidade.....	13
2.4. Notação Matricial Ortogonal (Alfa Beta Zero).....	15
2.5. Notação Vetorial.....	19

CAPÍTULO 3 - MODELO VETORIAL COMPLEXO PARA O MOTOR DE INDUÇÃO	26
3.1. Introdução	26
3.2. Sistema Dinâmico Complexo de Segunda Ordem.....	26
3.3. Obtenção do Modelo do Motor de Indução como um Sistema Dinâmico Complexo	28
CAPÍTULO 4 - PROCEDIMENTOS E MÉTODOS DE RESOLUÇÃO.....	33
4.1. Introdução	33
4.2. Descrição dos Programas para Simulação	34
4.2.1. Simulação com o Programa Simnon TM	35
4.2.2. Simulação com o Programa Octave	36
4.2.3. Simulação com o Programa Matlab®	37
4.2.4. Simulação com o Programa Simulink / Matlab®	38
4.3. Preparação dos Modelos para Resolução Numérica	39
4.3.1. Notação Trifásica	39
4.3.2. Notação Ortogonal.....	44
4.3.3. Notação Vetorial	51
4.3.4. Notação Complexa	55
CAPÍTULO 5 - RESULTADOS E ANÁLISES	61
5.1. Modelo na Notação Trifásica	62
5.1.1. Simulação com o Programa Simnon TM	62
5.1.2. Simulação com o Programa Octave	65
5.1.3. Simulação com o Programa Matlab®	67
5.1.4. Simulação com o Programa Simulink / Matlab®	69
5.2. Modelo na Notação Ortogonal.....	71

5.2.1. Simulação com o Programa Simnon TM	71
5.2.2. Simulação com o Programa Octave	73
5.2.3. Simulação com o Programa Matlab®	75
5.2.4. Simulação com o Programa Simulink / Matlab®	76
5.3. Modelo na Notação Vetorial.....	77
5.3.1. Simulação com o Programa Simnon TM	77
5.3.2. Simulação com o Programa Octave	79
5.3.3. Simulação com o Programa Matlab®	80
5.3.4. Simulação com o Programa Simulink / Matlab®	82
5.4. Notação na Vetorial Complexa	83
5.4.1. Simulação com o Programa Matlab®	84
5.4.2. Simulação com o Programa Simulink / Matlab®	88
5.5. Avaliação Global dos Resultados	92
CAPÍTULO 6 - CONCLUSÕES	97
REFERÊNCIAS BIBLIOGRÁFICAS.....	100
ANEXO A	102
APÊNDICES	

Lista de Figuras

Figura 2.1 - Representação esquemática dos enrolamentos trifásicos no Motor de Indução Trifásico	8
Figura 2.2 - Circuito elétrico equivalente do motor de indução trifásico de 2 pólos com rotor em gaiola	9
Figura 2.3 - Representação dos sistemas de coordenadas trifásico e ortogonal	16
Figura 2.4 - Plano complexo e referenciais arbitrários	21
Figura 3.1 - Representação de um sistema dinâmico complexo de segunda ordem...	27
Figura 3.2 - Diagrama de blocos para variável de estado fluxo com referencial estacionário	30
Figura 3.3 - Diagrama de blocos para a variável de estado fluxo com referencial síncrono	31
Figura 3.4 - Diagrama de blocos para a variável de estado corrente com referencial estacionário	31
Figura 3.5 - Diagrama de blocos para a variável de estado corrente com referencial síncrono	31
Figura 4.1 - Diagrama de blocos para a notação trifásica com variável de estado fluxo e referencial estacionário.	41
Figura 4.2 - Diagrama de blocos para a notação trifásica com variável de estado fluxo e referencial síncrono	42
Figura 4.3 - Diagrama de blocos para a notação trifásica com variável de estado corrente e referencial estacionário	43
Figura 4.4 - Diagrama de blocos para a notação trifásica com variável de estado corrente e referencial síncrono	44

Figura 4.5 - Diagrama de blocos para a notação ortogonal com variável de estado fluxo e referencial estacionário	47
Figura 4.6 - Diagrama de blocos para a notação ortogonal com variável de estado fluxo e referencial síncrono	48
Figura 4.7 - Diagrama de blocos para a notação ortogonal com variável de estado corrente e referencial estacionário	49
Figura 4.8 - Diagrama de blocos para a notação ortogonal com variável de estado corrente e referencial síncrono	50
Figura 4.9 - Diagrama de blocos para a notação vetorial com variável de estado fluxo e referencial estacionário	53
Figura 4.10 - Diagrama de blocos para a notação vetorial com variável de estado fluxo e referencial síncrono	53
Figura 4.11 - Diagrama de blocos para a notação vetorial com variável de estado corrente e referencial estacionário	54
Figura 4.12 - Diagrama de blocos para a notação vetorial com variável de estado corrente e referencial síncrono	55
Figura 4.13 - Diagrama de blocos para a notação complexa com variável de estado fluxo e referencial estacionário	56
Figura 4.14 - Diagrama de blocos para a notação complexa com variável de estado fluxo e referencial síncrono	57
Figura 4.15 - Diagrama de blocos para a notação complexa com variável de estado corrente e referencial estacionário	59
Figura 4.16 - Diagrama de blocos para a notação complexa com variável de estado corrente e referencial síncrono	59

Figura 5.1 - Gráfico da velocidade x tempo [rad/s] e do conjugado eletromagnético x tempo [Nm/s]	63
Figura 5.2 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono	64
Figura 5.3 - Gráfico das correntes por fase [A/s] nos referenciais: a) estacionário; b) síncrono	65
Figura 5.4 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono	66
Figura 5.5 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono	68
Figura 5.6 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono	70
Figura 5.7 - Gráfico da velocidade x tempo [rad/s] e do conjugado eletromagnético x tempo [Nm/s]	71
Figura 5.8 - Gráfico dos fluxos por eixo [Wb/s] nos referenciais: a) estacionário; b) síncrono	72
Figura 5.9 - Gráfico das correntes por eixo [A/s] nos referenciais: a) estacionário; b) síncrono	73
Figura 5.10 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes [A/s] por fase nos referenciais: c) estacionário; d) síncrono	74

Figura 5.11 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono	75
Figura 5.12 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono	76
Figura 5.13 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono; e das correntes [A] nos referenciais: c) estacionário; d) síncrono	78
Figura 5.14 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono	79
Figura 5.15 - Gráfico da composição das partes real e imaginária da corrente [A] nos referenciais: a) estacionário; b) síncrono	80
Figura 5.16 - Gráfico da composição do fluxo [Wb] para os referenciais: a) estacionário; b) síncrono, e das correntes [A] nos referenciais: c) estacionário e d) síncrono, nos eixos real x imaginário	81
Figura 5.17 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono	82
Figura 5.18 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono	83
Figura 5.19 - Gráfico da velocidade x tempo e o conjugado eletromagnético x tempo [Nm/s]	84
Figura 5.20 - Gráfico do fluxo complexo [Wb] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	85

Figura 5.21 - Gráfico do comportamento transitório do fluxo [Wb] nos eixos real e imaginário x tempo para os referenciais: a) estacionário; b) síncrono	86
Figura 5.22 - Gráfico da corrente complexa [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	87
Figura 5.23 - Gráfico do comportamento transitório da corrente [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	88
Figura 5.24 - Gráfico do fluxo complexo [Wb] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	89
Figura 5.25 - Gráfico do comportamento transitório do fluxo [Wb] nos eixos real e imaginário x tempo para os referenciais: a) estacionário; b) síncrono	90
Figura 5.26 - Gráfico da corrente complexa [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	91
Figura 5.27 - Gráfico do comportamento transitório da corrente [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono	92
Figura 5.28 - Tempo médio de simulação do motor de indução trifásico usando o programa Octave	93
Figura 5.29 - Tempo médio de simulação do motor de indução trifásico usando o programa Matlab®.....	94
Figura 5.30 - Tempo médio de simulação do motor de indução trifásico usando o programa Simulink / Matlab®	94
Figura 5.31 - Tempo mínimo de simulação do motor de indução trifásico para cada programa	96

Lista de Tabelas

Tabela 01 -	Dados do motor de indução para simulação	34
Tabela 02 -	Indicativo dos programas e seus respectivos programas	35
Tabela 03 -	Tempo Mínimo de Simulação Para Cada Programa [s]	95

Símbolos e Notações

θ	defasagem angular entre o enrolamento da fase “a” de estator “a” do rotor
β	defasagem angular entre o enrolamento das fases “a” e “b” do estator, 120 graus do sistema trifásico
$\underline{I}$	vetor coluna dos fluxos do motor por fase
$\sigma = 1 - \frac{L_H^2}{L_1 L_2}$	coeficiente de dispersão global
l_1	indutância própria de estator
l_2	indutância própria de rotor
$l_{\sigma 1}$	indutância de dispersão de estator
$l_{\sigma 2}$	indutância de dispersão de rotor
L_H	indutância de magnetização
$L_1 = \frac{3}{2}l_1 + l_{s1}$	Indutância própria por fase do estator
$L_2 = \frac{3}{2}l_2 + l_{s2}$	Indutância própria por fase do rotor
m	valor máximo da indutância mútua entre enrolamentos de estator e rotor
$M = \frac{3}{2}m$	indutância própria de um enrolamento no referencial único
R_1	resistência dos enrolamentos das fases do estator
R_2	resistência dos enrolamentos das fases do rotor
ω_1	frequência das tensões de estator
ω_2	frequência das tensões de rotor
V	perda ôhmica nos enrolamentos;
k	defasagem angular do referencial genérico com relação a fase “a” de estator
W	energia magnética necessária à manutenção do campo
P	potência elétrica total fornecida
$Re\{ \}$	parte real do termo complexo

$Im\{ \}$	parte imaginária do termo complexo
J	Momento de inércia
K_D	Coefficiente de atrito viscoso
NP	Número de pares de pólos do motor de indução
s	Escorregamento $\rightarrow s = \frac{\omega_1 - \omega_2}{\omega_1}$
m_d	Conjugado eletromagnético
$\underline{u}$	vetor coluna das tensões do motor por fase
$\underline{i}$	vetor coluna das correntes do motor por fase
$\underline{I}$	matriz identidade
a_1, a_2, b_1, b_2	constantes complexas
$\underline{x}$	saída de estado do sistema complexo
c_{12}	acoplamento do sistema dinâmico complexo referente ao estator
c_{21}	acoplamento do sistema dinâmico complexo referente ao rotor
Subscrito:	
α	eixo alfa do modelo ortogonal
β	eixo beta do modelo ortogonal
0	eixo zero do modelo ortogonal
1, s	relativo a grandezas de estator, fluxo, corrente, resistência, impedância
2, r	relativo a grandezas de rotor, fluxo, corrente, resistência, impedância
a	fase “a” da rede
b	fase “b” da rede
c	fase “c” da rede
k	entidade no referencial genérico
Sobrescrito:	
$\rightarrow$	vetor
$\cdot$	derivada
$*$	conjugado complexo

Resumo

Nesse trabalho procede-se a modelagem e simulação do motor de indução trifásico considerando-se as notações trifásicas, ortogonais, vetoriais e complexas, mostrando seus equacionamentos e também o resultado das simulações. Para a simulação foram usados alguns programas de domínio da área acadêmica, comparando seus desempenhos quanto à apresentação de resultados e também tempo de processamento. Este trabalho apresenta também, um enfoque para o método de simulação do motor de indução trifásico utilizando a notação vetorial complexa, o qual é baseado na notação vetorial do motor de indução que é caracterizado por grandezas complexas. Essa técnica é obtida através de simples manipulações das equações vetoriais do modelo do motor de indução compondo uma equação de estado complexa. Com o auxílio do programa Matlab®, consegue-se simular o motor de indução trifásico sem a necessidade de separar os termos complexos em duas equações reais, relativas as partes real e imaginária. O que além de simplificar o procedimento de simulação também contribui para a construção do diagrama de blocos para poder entender melhor o comportamento do modelo estudado. São apresentadas no final do trabalho, as conclusões obtidas e, também, sugestões tanto para continuação do trabalho, quanto novas linhas de pesquisas.

Palavras-Chave: Motor de Indução, Modelagem Matemática, Simulação da Máquina Elétrica, Aproximação de Espaço de Estado, Modelagem com Fasor de Espaço.

Abstract

In this work it is carried out the modelling and simulation of the three-phase induction motor. It's considered three-phase, orthogonal, vectorial and complex notations, showing the different model equations and the result of the computational simulations. For the simulation it was used different software's of the academic area, and its results and computational performance are compared. This work gives emphasis to in new modelling procedure by using complex vector notation. This new method is based on the vectorial notation of the induction motor, which is characterized by complex entities. Through simple manipulations of complex vector equation of the dynamic induction motor equation, it is possible to compose a complex space-state equation. This complex model come be solved with Matlab® software without the separation of its complex terms in two real equations. Other advantage of the complex model is the simplifying the simulation procedure and the possibilities of the blocks diagram representation. The final conclusions and suggestions for continuation are presented in the end of work.

Keywords: Induction Motor, Mathematical Modelling, Electric Machine Simulation, Space State Approach, Space Phasor Modelling.

Capítulo 1

INTRODUÇÃO

O motor de indução é o tipo de motor elétrico mais utilizado e difundido, tanto para motorização de sistemas, quanto para processos industriais. Sua principal vantagem é a eliminação do atrito de todos contatos elétricos deslizantes e uma construção bastante simples, o que possibilitou sua construção a um custo ainda mais baixo, sendo que estas máquinas são fabricadas para uma grande variedade de aplicações, desde alguns watts até muitos megawatts (Leonhard, 1985). Além de ser robusto em termos de operação, proporcionando vantagens econômicas consideráveis tanto na aquisição, quanto na manutenção.

Mesmo com essas vantagens, os motores de indução não tinham muita importância até a alguns anos atrás, quando se levava em consideração aplicações com velocidade variável, pois todas tentativas neste sentido necessitavam de um equipamento adicional, ou então, sofriam grandes perdas de potência. Embora fossem investigados os problemas da eficácia de controlar a velocidade dos motores de indução durante décadas, todas as soluções realizáveis até alguns anos atrás eram muito complicadas e/ou caras. Uma primeira solução foi obtida com relação às técnicas de modelagem, com o propósito de se obter um conjunto de equações dinâmicas mais simples e voltadas para aplicações de controle, mas sua implementação exigia grande esforço computacional, ou os conversores de potência eram inexistentes ou de desempenho insatisfatório (Vas, 1994). Somente com o progresso recente da tecnologia de semicondutores é que puderam ser construídos, também, conversores estáticos de frequência que associados e acionados por microprocessadores de alto desempenho, possibilitaram a construção de servossistemas com motores de indução a baixo custo.

Com as técnicas de modelagem e acionamento existentes, o desempenho dos servossistemas AC com motores de indução se igualaram aos servossistemas DC. Uma vez que o custo dos motores de indução é bem inferior, os servossistemas AC se tornaram também muito mais interessantes (de Andrade, 1994).

O avanço da tecnologia também contribuiu para o avanço nas técnicas de modelagem pois com os novos processadores e programas existentes no mercado, possibilitaram-se o estudo e o aprimoramento de novas técnicas de modelagem.

Estudos recentes tem apresentado uma nova metodologia para a modelagem dinâmica e a simulação do motor de indução trifásica, baseada em grandezas complexa (Szablya & Bressane, 1973; Novotny & Wouterse, 1976; Holtz, 1995). A modelagem dinâmica pode ser estudada através da notação trifásica, ortogonal e vetorial. Tal como é conhecido e que, também, será elucidada no decorrer do trabalho, a notação trifásica tem como desvantagem o número de equações diferenciais a ser utilizado para modelagem completa, da qual resultam 8 (oito) equações diferenciais levando-se em consideração a modelagem para a velocidade e a posição angular.

O modelo ortogonal ($\alpha\beta 0$) surgiu para tentar diminuir esse número de equações, conseguindo chegar a um modelo com o mesmo número de equações, porém com um maior número de zeros na matriz, caracterizando uma matriz mais esparsa, o que facilita um pouco o cálculo em relação ao modelo da notação trifásica (Holtz, 1995). Neste tipo de análise, se o sistema for equilibrado ou sem conexão de neutro, a denominada fase "0" é eliminada resultando num sistema de apenas duas coordenadas (α, β).

Os mais promissores avanços obtidos com relação aos servossistemas AC em motores de indução resultaram a partir do surgimento da modelagem do motor utilizando técnicas vetoriais (Kovács & Rácz, 1959). Em princípio, esta técnica é definida a partir do sistema ortogonal ($\alpha\beta 0$), porém impondo-se que este plano configure um plano complexo, com um eixo real e outro imaginário. Neste caso as entidades definidas neste plano são manipuladas e processadas na notação cartesiana das entidades complexas, sem o eixo "0" da notação ($\alpha\beta 0$). A eliminação do eixo "0" proporciona uma redução de ordem, porém as manipulações algébricas necessárias para compor as equações em termos reais e imaginários, caracterizam um procedimento complicado e resultam em equações não lineares e fortemente acopladas. (Scott Wade, Matthew W. Dunnigan, Barry W. Williams, 1994).

Trabalhos recentes mostram que o uso de entidades vetoriais complexas na modelagem dinâmica, tem apresentado um resultado satisfatório e muito interessante em termos de compactação na formulação de sistemas dinâmicos, tais como os moto-

res de indução (Gataric & Garrigan, 1999). Neste trabalho, com o auxílio do *software* Matlab, serão mostrados os procedimentos de simulação da partida do motor de indução nesta notação vetorial complexa. Os resultados deste caso serão comparados com os resultados de simulação do mesmo motor, obtidos através dos outros métodos de modelagem citados. Este procedimento de modelagem vetorial complexa possui as vantagens de ser mais rápido e prático, além de facilitar a construção de diagramas de blocos, o que é muito utilizado para interpretações na área de engenharia elétrica (Dalton & Gosbell, 1989; de Aguiar & Cad, 1999c).

Além disso, no presente trabalho serão apresentadas e discutidas as formas convencionais de modelagem do motor de indução trifásico a partir dos modelos trifásicos até os modelos vetoriais, e será introduzida a modelagem vetorial complexa, bem como a devida análise destes modelos. Como forma de evidenciar outras vantagens da modelagem vetorial complexa. Serão executados procedimentos de simulação com todos os modelos a serem abordados, e os resultados e procedimentos de simulação serão comparados. Para se apresentar todos estes tópicos propostos, organizou-se o trabalho tal como descrito a seguir.

1.1 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2, apresenta-se uma descrição e desenvolvimento usual dos métodos de modelagem existentes, desde os clássicos trifásicos, passando pelo ortogonal e terminando nos atuais modelos vetoriais. Será também mostrado o equacionamento em cada um dos casos e o interesse dessas equações para simulação.

No Capítulo 3, trata-se da parte de contribuição fundamental deste trabalho, ou seja, a apresentação e análise do Modelo Vetorial Complexo. Neste ponto é apresentada toda a modelagem dinâmica baseada em entidades complexas, apresentando também as equações complexas, o diagrama de blocos e o modelo dinâmico complexo.

No Capítulo 4, apresentam-se os procedimentos utilizados para resolução de todos os modelos dinâmicos abordados no trabalho. Na resolução dos modelos foram usados alguns programas de conhecimento acadêmico com um breve descritivo dos mesmos.

No Capítulo 5, apresentam-se e discutem-se os resultados obtidos utilizando a notação trifásica, ortogonal e vetorial, utilizando as duas variáveis de estado, ou seja, fluxo e corrente, tanto no referencial estacionário, quanto no síncrono.

No Capítulo 6, apresentam-se as conclusões obtidas com o trabalho e sugere-se algumas linhas de trabalhos que poderão contribuir para a elaboração de novos estudos.

No Anexo A é apresentado uma descrição do sistema complexo de primeira ordem.

No Apêndice A são mostrados o capítulo de livro e os artigos gerados através com o estudo deste trabalho.

No Apêndice B são mostradas as listagens das rotinas desenvolvidas para a simulação das notações utilizadas no trabalho.

Capítulo 2

MODELAGEM MATEMÁTICA DO MOTOR DE INDUÇÃO TRIFÁSICO

2.1. INTRODUÇÃO

Neste Capítulo, serão apresentados os procedimentos clássicos de modelagem do motor de indução trifásico, os quais serão comparados com procedimento de modelagem vetorial complexa a ser discutido no Capítulo 3.

A evolução das técnicas de modelagem de motores de indução culminou nos atuais modelos vetoriais complexos, os quais possibilitam a representação do modelo do motor de indução através de diagramas de blocos. Na sequência são apresentados e resumidos uma série de trabalhos que contribuíram para evolução dos procedimentos de modelagem, culminando com o modelo vetorial complexo.

Os modelos primordiais relativos aos motores de indução, caracterizavam-se por serem desenvolvidos para os casos de regime permanente, avaliando portanto somente as condições nos pontos de operação ou o comportamento devido a pequenos desvios deste ponto de operação (modelagem com a técnica de pequenos deslocamentos). Estes modelos, também denominados clássicos, não permitiam a avaliação de desempenho dinâmico em grandes faixas de velocidade (Alger, 1951; Kraus, et al. 1978).

Kovács & Rácz (1959) mostraram que a formulação complexa, ou vetorial do modelo do motor de indução, é alcançada diretamente da aplicação da análise de vetor de espaço.

Szablya & Bressane (1973) analisaram a formulação complexa de sistemas dinâmicos complexos, aplicando a Transformada de Laplace para obter a função de transferência. No modelo foram utilizadas as equações fundamentais de tensão para uma máquina girante e utilizaram como referencial principal o rotor, ou seja, Transformada de Park. Foram também desenvolvidas as funções de transferência para cor-

rente, admitância, impedância e posteriormente feita a análise para a Transformada de Clark.

Novotny & Wouterse (1976) utilizaram variáveis complexas no domínio do tempo, o que proporcionou uma nova ferramenta para análise. Utilizaram também o conceito de função de transferência complexa, mostrando o comportamento desta função de transferência utilizando o método do lugar das raízes para algumas situações, como por exemplo, a máquina funcionando com baixo escorregamento, o comportamento da frequência de rotor e, também, da velocidade do rotor.

De Doncker & Novotny (1988) utilizaram a modelagem vetorial quando propuseram um controlador universal de campo orientado, com a capacidade de desacoplar o fluxo e o conjugado em um referencial de fluxo arbitrário.

Dalton & Gosbell (1989) desenvolveram a modelagem dos sistemas dinâmicos complexos, permitindo a construção de um diagrama de blocos bastante compacto, o que auxilia nas interpretações da máquina.

Yamamura (1992) introduziu a teoria do vetor espiral, baseada no comportamento transitório do motor de indução trifásico à entrada degrau, o que corresponde ao comportamento elétrico da máquina. O conceito de vetor espiral é diretamente relacionado com os conceitos de função transferência complexa, pois processam grandezas dinâmicas complexas.

Vas (1992) aplicou o vetor de espaço em máquinas e entidades elétricas; demonstrou as equações para o cálculo do conjugado eletromagnético, para a potência instantânea, para o fluxo nos modelos trifásico e ortogonal e o cálculo da corrente no modelo ortogonal. Mostra, também, o modelo de quinta ordem e depois este mesmo modelo reduzido a ordem menor.

Vas (1994) descreveu o modelo completo do motor de indução utilizando equações diferenciais complexas e utilizou diversos tipos de modelagem para controlar o motor de indução por meio de técnicas apropriadas.

Wade et al. (1994) segmentaram as equações dinâmicas complexas em partes real e imaginária, para poder simulá-las, uma vez que os programas disponíveis não manipulavam entidades complexas.

Holtz (1995) mostrou vários métodos de simulação complexa, utilizando referencial síncrono e diversos tipos de combinações de variáveis de estado, ou seja, cor-

rente de estator e fluxo de rotor, fluxo de estator e fluxo de rotor. Traça o diagrama de blocos complexo, lugar das raízes e faz a análise para as raízes complexas.

Gataric & Garrigan (1999) mostraram um estudo do motor trifásico aplicando transformada de Laplace na função de transferência complexa e mostrando seu comportamento através de gráfico de Bode, mostraram também o controle para um inversor utilizando um filtro LC e utilizando um controlador complexo.

de Aguiar & Cad (1999a; 1999b; 1999c) utilizaram a definição de sistema dinâmico complexo e mostraram como resolver um sistema de equações complexas utilizando o programa Matlab® e compararam com o resultado utilizando o desmembramento em partes real e imaginária.

de Aguiar & Cad (2000a; 2000b) estudaram e apresentaram procedimentos de modelagem e simulação do motor de indução trifásico por meio de função transferência complexa, utilizando o Matlab®/Simulink em alguns referenciais e utilizando variável de estado fluxo e corrente.

2.2. PROCEDIMENTOS DE MODELAGEM DO MOTOR DE INDUÇÃO TRIFÁSICO

A modelagem matemática é utilizada para obter uma descrição do comportamento das grandezas internas da máquina e, no caso do motor de indução trifásico, o comportamento dinâmico deve ser obtido através das equações de:

- Tensão / corrente;
- Fluxo concatenado;
- Conjugado eletromagnético;
- Movimento e posição angular.

Neste trabalho será estudadas somente a modelagem e a simulação para o caso da velocidade angular de rotor como saída.

O comportamento dinâmico deve ser obtido baseado no conhecimento da estrutura construtiva do motor, o que permitirá representá-lo por meio de um circuito elétrico equivalente e através dos fenômenos eletromagnéticos e mecânicos envolvidos neste circuito equivalente.

O motor de indução trifásico convencional contém, no caso do motor de anéis, dois enrolamentos trifásicos, um localizado no estator, sendo uma estrutura fixa e outro localizado no rotor, sendo uma estrutura girante, ambos com o mesmo número de pólos. Outra forma bem mais comum, é substituir o enrolamento do rotor por um sistema de barras paralelas, ligeiramente inclinadas em relação ao eixo mecânico, curto-circuitadas em seus extremos por dois anéis formando uma “gaiola de esquilo” e é, por isso, denominado rotor em gaiola; esses rotores podem ser diferenciados quanto à forma e/ou profundidade das barras ou ranhuras, garantindo assim diferentes características operacionais e de partida, porém tornando o acesso elétrico a ele, impraticável. Já nos motores de anéis, ou com rotor bobinado, dispõem-se de terminais no enrolamento trifásico do rotor ligado a anéis/escovas deslizantes permitindo assim uma atuação, ou medição das grandezas elétricas do mesmo, tais como parâmetros, correntes, tensões, potências, etc.

Sabe-se que um motor de indução convencional possui enrolamentos trifásicos, que é caracterizado por três bobinas, tal como mostrado na figura 2.1, denominadas fases ABC. Cada fase, por sua vez, é deslocada espacialmente no perímetro do motor de 120° elétricos.

O campo magnético no entreferro da máquina tem direção radial. As superfícies entre o estator e o rotor são lisas e a permeabilidade do ferro é admitida infinita. Considerando que os efeitos nas extremidades são desprezados, o campo magnético torna-se bi-dimensional.

FIGURA 2.1 - Representação esquemática dos enrolamentos trifásicos no Motor de Indução Trifásico.

Na figura 2.2 é vista uma representação típica da estrutura de enrolamentos do motor em forma esquemática, o qual é mais prático para se estabelecer às relações

matemática do modelo. Para utilizar o circuito elétrico da figura 2.2 para o motor de indução, são feitas as seguintes suposições:

A máquina é considerada magneticamente linear;

Os enrolamentos de fase produzem uma distribuição espacial de fmm senoidal ao longo da direção do perímetro do estator;

As fases de estator e rotor são conectados em Y, de modo que a soma das correntes instantâneas de estator e rotor seja nulas;

Efeito pelicular e perdas no ferro são desconsiderados.

FIGURA 2.2 - Circuito elétrico equivalente do motor de indução trifásico de 2 pólos com rotor em gaiola.

Como já mencionado, no presente trabalho será apresentada a modelagem do motor de indução através de diversos procedimentos clássicos, os quais serão comparados com o modelo a ser desenvolvido no Capítulo 3. Estes modelos clássicos se diferem pela notação matemática aplicada a cada um deles. A notação por sua vez está relacionada à forma de simplificações aplicada à estrutura construtiva ou de análise do motor de indução. Com base nesta classificação de modelos, distinguem-se as seguintes formas de modelagem do motor de indução:

- Notação Matricial Trifásica;
- Notação Matricial Ortogonal (Alfa Beta Zero);
- Modelo Vetorial:
 - ⇒ Separado em Parte Real e Imaginária;
 - ⇒ Complexo.

Na sequência deste Capítulo, serão revistos o modelo trifásico, o modelo ortogonal e o vetorial convencional. O modelo vetorial complexo será mais bem investigado no Capítulo 3. Em cada um dos casos a serem abordados, distingue-se ainda uma subclassificação de modelos com relação à variável de estado a ser utilizada na descrição matemática, as quais podem ser os fluxos ou as correntes do motor.

2.3. NOTAÇÃO MATRICIAL TRIFÁSICA

Na representação trifásica, obtém-se as equações diferenciais que descrevem o comportamento dinâmico das grandezas por fase, tanto de estator quanto de rotor, bem como as relações entre elas, totalizando 6 equações de tensão. A notação matricial é adotada devido ao fato de existir um número considerável de variáveis. Assim, consideram-se as tensões, correntes e fluxos no motor por fase, como sendo definidos por vetores coluna, tais como:

$$\underline{u} = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \quad \underline{i} = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}, \quad \underline{l} = \begin{bmatrix} l_a \\ l_b \\ l_c \end{bmatrix} \quad (2.1)$$

onde os subíndices a , b e c , representam cada grandeza por fase.

2.3.1. EQUAÇÕES DE TENSÃO EM UM CIRCUITO RESISTIVO-INDUTIVO ACOPLADO MAGNETICAMENTE

Com base na figura 2.2, as equações elétricas relacionam o comportamento elétrico em um circuito resistivo-indutivo acoplado magneticamente. Dessa forma as equações de tensão de estator e rotor, serão dadas por:

$$\underline{u}_{1s} = R_1 \underline{i}_{1s} + \frac{d}{dt} \underline{l}_{1s} \quad (2.2-a)$$

$$\underline{u}_{2r} = R_2 \underline{i}_{2r} + \frac{d}{dt} \underline{l}_{2r} \quad (2.2-b)$$

O duplo índice presente na equação (2.2-a) representa as grandezas fluxo e corrente de estator referida ao estator e (2.2-b) representa as grandezas fluxo e corrente de rotor referida ao rotor.

2.3.2. EQUAÇÕES DO FLUXO CONCATENADO

Os termos de fluxo presentes em (2.1) e (2.2), representam o fluxo total concatenado por fase que é composto pelas várias contribuições de fluxos devido as indutâncias próprias de estator e de rotor (l_l, l_2), pelas indutâncias de dispersão de estator e de rotor ($l_{\sigma 1}, l_{\sigma 2}$), e pela indutância mútua entre fases do enrolamento de estator e do rotor (m). Considerando-se a fase "a", a contribuição de fluxo total é:

$$\begin{aligned} \underline{I}_{a1s}(t) = & (l_1 + l_{s1})i_{a1s}(t) + l_1 \cos(\mathbf{b})i_{b1s}(t) + l_1 \cos(-\mathbf{b})i_{c1s}(t) + \\ & m \cos(\mathbf{q}(t))i_{a2r}(t) + m \cos(\mathbf{q}(t) + \mathbf{b})i_{b2r}(t) + m \cos(\mathbf{q}(t) - \mathbf{b})i_{c2r}(t) \end{aligned} \quad (2.3)$$

sendo $\mathbf{q}$ o ângulo de defasagem angular entre os enrolamentos da fase "a" de estator e "a" de rotor e $\mathbf{b}$ o ângulo de defasagem entre o enrolamento das fases "a" e "b" do estator (120° elétricos).

Em (2.3), percebe-se a presença de um triplo índice, onde o primeiro termo representa qual fase está sendo analisada, "a", "b" ou "c", o segundo termo representa se é em relação ao estator (1) ou rotor (2) e o terceiro índice mostra se o fluxo está referido ao estator "s" ou ao rotor "r". Obtêm-se as expressões para as fases "b" e "c" por analogia com a expressão da fase "a".

Em forma matricial, o vetor de fluxo concatenado de estator observado na estrutura do estator, será dado por:

$$\begin{aligned} \underline{I}_{1s} = \begin{bmatrix} \underline{I}_{a1s}(t) \\ \underline{I}_{b1s}(t) \\ \underline{I}_{c1s}(t) \end{bmatrix} = & \left\{ l_1 \begin{bmatrix} 1 & \cos(\mathbf{b}) & \cos(-\mathbf{b}) \\ \cos(-\mathbf{b}) & 1 & \cos(\mathbf{b}) \\ \cos(\mathbf{b}) & \cos(-\mathbf{b}) & 1 \end{bmatrix} + l_{s1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\} \begin{bmatrix} i_{a1s}(t) \\ i_{b1s}(t) \\ i_{c1s}(t) \end{bmatrix} + \\ & + m \begin{bmatrix} \cos \mathbf{q}(t) & \cos(\mathbf{q}(t) + \mathbf{b}) & \cos(\mathbf{q}(t) - \mathbf{b}) \\ \cos(\mathbf{q}(t) - \mathbf{b}) & \cos \mathbf{q}(t) & \cos(\mathbf{q}(t) + \mathbf{b}) \\ \cos(\mathbf{q}(t) + \mathbf{b}) & \cos(\mathbf{q}(t) - \mathbf{b}) & \cos \mathbf{q}(t) \end{bmatrix} \begin{bmatrix} i_{a2r}(t) \\ i_{b2r}(t) \\ i_{c2r}(t) \end{bmatrix} \end{aligned} \quad (2.4-a)$$

ou, omitindo a variável independente t por questão de simplificação:

$$\underline{I}_{1s} = (l_1 \underline{T}_0(0) + l_{s1} \underline{I}) \underline{i}_{1s} + m \underline{T}_0(\mathbf{q}) \underline{i}_{2r} \quad (2.4-b)$$

sendo que:

$$\underline{T}_0(\mathbf{q}) = \begin{bmatrix} \cos(\mathbf{q}) & \cos(\mathbf{q} + \mathbf{b}) & \cos(\mathbf{q} - \mathbf{b}) \\ \cos(\mathbf{q} - \mathbf{b}) & \cos(\mathbf{q}) & \cos(\mathbf{q} + \mathbf{b}) \\ \cos(\mathbf{q} + \mathbf{b}) & \cos(\mathbf{q} - \mathbf{b}) & \cos(\mathbf{q}) \end{bmatrix} \quad (2.5)$$

Da mesma forma, podem ser obtidas as expressões de fluxo concatenado nas fases do rotor visto na estrutura do rotor, cuja representação matricial final será:

$$\underline{I}_{2r} = m \underline{T}_o(-\mathbf{q})(t) \underline{i}_{1s}(t) + (l_2 \underline{T}_0(0) + l_{s2} \underline{I}) \underline{i}_{2r}(t) \quad (2.6)$$

As equações (2.4-b) e (2.6) apresentam o inconveniente, de que as grandezas relacionadas estão referenciadas a diferentes sistemas de coordenadas, com diferentes deslocamentos angulares. Para se fazer uma análise do comportamento dinâmico do motor de indução, deve-se adotar, então, um referencial único e comum para as grandezas de estator e rotor.

2.3.3. TRANSPOSIÇÃO PARA REFERENCIAL ÚNICO

Na figura 2.2, este referencial genérico é indicado em linhas tracejadas tendo uma defasagem angular k com relação à fase “a” do estator. A velocidade ou deslocamento angular deste referencial genérico é definido por:

$$\mathbf{w}_k(t) = \frac{d}{dt} k(t) \quad (2.7)$$

Usualmente adota-se o referencial genérico como sendo um daqueles que possam ser definidos no próprio motor. Desta forma adota-se um dos seguintes referenciais como sendo único:

- Referencial fixo no estator: $\mathbf{w}_k = 0$
- Referencial fixo no rotor: $\mathbf{w}_k = \mathbf{w}_{mec}$
- Referencial fixo no campo de estator: $\mathbf{w}_k = \mathbf{w}_1$

Fazendo-se a transformação adequada dos sistemas de coordenadas, e devido às relações geométricas, pode-se substituir o duplo índice pelos índices “1” para estator e “2” para rotor. Com isso, as equações da tensão e do fluxo concatenado tornam-se:

$$\underline{u}_1 = R_1 \underline{i}_1 + \frac{d}{dt} \underline{I}_1 + \mathbf{w}_k K \underline{I}_1 \quad (2.8-a)$$

$$\underline{u}_2 = R_2 \underline{i}_2 + \frac{d}{dt} \underline{L}_2 + (\underline{w}_k - \underline{w}_{mec}) \underline{K} \underline{I}_2 \quad (2.8-b)$$

$$\underline{I}_1 = L_1 \underline{i}_1 + L_H \underline{i}_2 \quad (2.9-a)$$

$$\underline{I}_2 = L_H \underline{i}_1 + L_2 \underline{i}_2 \quad (2.9-b)$$

onde

$$\underline{K} = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad (2.10-a)$$

$$L_H = \frac{3}{2} m = \frac{3}{2} l_1 = \frac{3}{2} l_2 \quad (2.10-b)$$

2.3.4. EQUAÇÕES DE CONJUGADO ELÉTRICO E DE VELOCIDADE

A maneira mais adequada para se obter a expressão do conjugado elétrico produzido no motor de indução trifásico é por meio de uma análise do balanço de energia no motor.

Considerando-se a potência elétrica total fornecida ao motor como sendo:

$$P = \underline{u}_1^T \underline{i}_1 + \underline{u}_2^T \underline{i}_2 \quad (2.11)$$

Dividindo a potência em três partes, têm-se:

$$P = V + W + m_d 2p \underline{w}_{mec} \quad (2.12)$$

sendo:

- V = perda ôhmica nos enrolamentos;
- W = energia magnética necessária à manutenção do campo;
- $m_d 2p \underline{w}_{mec}$ = potência mecânica desenvolvida pelo motor.

encontra-se que o conjugado elétrico pode ser expresso por:

$$m_d = -NP \underline{I}_1^T \underline{K} \underline{i}_1 \quad (2.13)$$

considerando que,

$$\underline{K}^T = -\underline{K} \quad (2.14)$$

e, desde que os termos de fluxo de dispersão $\underline{I}_s$ não contribuem para produção do conjugado elétrico, segue que o conjugado elétrico pode ainda ser expresso por outras formas, tais como:

$$m_d = -NP \underline{I}_H^T \underline{K} i_1 \quad (2.15-a)$$

$$m_d = NP \underline{I}_H^T \underline{K} i_2 \quad (2.15-b)$$

$$m_d = NP \underline{I}_2^T \underline{K} i_2 \quad (2.15-c)$$

Finalizando a modelagem trifásica do comportamento dinâmico do motor de indução trifásico, as equações de movimento do motor são:

$$J \frac{d}{dt} \mathbf{w}_{mec} = m_d - K_D \mathbf{w}_{mec} - m_l \quad (2.16)$$

onde m_l é o conjugado de carga.

Por conseguinte, o modelo dinâmico completo em forma matricial trifásica com referencial único, é composto por um sistema de 7 (sete) equações diferenciais que podem ser escritas em função das variáveis de estado fluxo ou corrente. Isolando-se as correntes de estator e rotor na equação (2.9), obtêm-se:

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\mathbf{s} L_1} & \frac{-L_H}{\mathbf{s} L_1 L_2} \\ \frac{-L_H}{\mathbf{s} L_1 L_2} & \frac{1}{\mathbf{s} L_2} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \quad (2.17)$$

sendo $\mathbf{s} = \left(1 - \frac{L_H^2}{L_1 L_2}\right)$ o coeficiente de dispersão global.

Substituindo as correntes das equações (2.17) diretamente nas equações (2.8), obtém-se assim o modelo em função apenas do fluxo e da tensão do motor.

$$\begin{aligned} \underline{u}_1 &= R_1 \dot{\underline{I}}_1 + \frac{d}{dt} \underline{I}_1 + \mathbf{w}_k \underline{K} \underline{I}_1 = \\ \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \end{bmatrix} &= \left(\frac{R_1}{\mathbf{s} L_1} \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \end{bmatrix} - \frac{R_1 L_H}{\mathbf{s} L_1 L_2} \begin{bmatrix} I_{2a} \\ I_{2b} \\ I_{2c} \end{bmatrix} \right) + \frac{d}{dt} \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \end{bmatrix} + \frac{\mathbf{w}_k}{\sqrt{3}} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \end{bmatrix} \end{aligned} \quad (2.18-a)$$

$$\underline{u}_2 = R_2 \dot{\underline{i}}_2 + \frac{d}{dt} \underline{L}_2 + \underline{w}_2 \underline{K} \underline{I}_2 =$$

$$\begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{2c} \end{bmatrix} = 0 = \left(-\frac{R_2 L_H}{s L_1 L_2} \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \end{bmatrix} + \frac{R_2}{s L_2} \begin{bmatrix} I_{2a} \\ I_{2b} \\ I_{2c} \end{bmatrix} \right) + \frac{d}{dt} \begin{bmatrix} I_{2a} \\ I_{2b} \\ I_{2c} \end{bmatrix} + \frac{\underline{w}_2}{\sqrt{3}} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_{2a} \\ I_{2b} \\ I_{2c} \end{bmatrix}$$

(2.18-b)

onde

$$\underline{w}_2 = \underline{w}_k - \underline{w}_{mec} \quad (2.19)$$

Por meio das equações (2.16) e (2.18) obtém-se o modelo dinâmico completo, para o motor de indução trifásico utilizando a variável de estado fluxo.

Outra forma de se expressar o mesmo modelo é utilizando a variável de estado corrente do motor. Em termos das correntes de estator e rotor, substituindo-se os termos de fluxo de estator e rotor das equações (2.9-a) e (2.9-b) diretamente em (2.8-a) e (2.8-b), obtêm-se:

$$\underline{u}_1 = R_1 \dot{\underline{i}}_1 + \frac{d}{dt} \underline{L}_1 + \underline{w}_k \underline{K} \underline{I}_1 =$$

$$\begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \end{bmatrix} = R_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + \frac{d}{dt} \left(L_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + L_H \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \right) + \frac{\underline{w}_k}{\sqrt{3}} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \left(L_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + L_H \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \right)$$

(2.20-a)

$$\underline{u}_2 = R_2 \dot{\underline{i}}_2 + \frac{d}{dt} \underline{L}_2 + \underline{w}_2 \underline{K} \underline{I}_2 =$$

$$\begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{2c} \end{bmatrix} = R_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} + \frac{d}{dt} \left(L_H \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + L_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \right) + \frac{\underline{w}_2}{\sqrt{3}} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \left(L_H \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + L_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{bmatrix} \right)$$

(2.20-b)

E acrescentando a equação (2.16), obtém-se o modelo dinâmico completo para a variável de estado corrente.

2.4. NOTAÇÃO MATRICIAL ORTOGONAL (ALFA BETA ZERO)

Com o intuito de se simplificar o modelo do motor de indução trifásico e, eventualmente, diminuir o número de variáveis das expressões matemáticas para des-

crever seu comportamento dinâmico, introduz-se o modelo ortogonal, substituindo-se o sistema trifásico de 3 (três) eixos defasados de 120° entre si, por um sistema ortogonal com 2 (dois) eixos defasados de 90° entre si.

Como consequência, o motor de indução trifásico será visto como sendo constituído apenas por duas bobinas defasadas espacialmente de 90° , nos enrolamentos de estator e de rotor. Na figura 2.3 representa-se a disposição dos sistemas trifásico e ortogonal. Incluindo-se a fase de sequência 0 (zero), bastante importante para a análise de sistemas assimétricos ou desbalanceados. Matematicamente, a fase 0 (α -ro) vem de uma condição da inversão da matriz de transformação.

FIGURA 2.3 - Representação dos sistemas de coordenadas trifásico e ortogonal.

Com base na figura 2.3, o novo sistema de eixos é denominado **a b 0** e, por conseguinte, usam-se os índices ($\alpha, \beta, 0$). O eixo “0” mencionado também é usado para representar as grandezas do sistema trifásico quando o neutro não é aterrado ou quando há fio neutro. Baseado na disposição geométrica da figura 2.3, a transformação do sistema trifásico para o sistema ortogonal será dado por:

$$\underline{u} = \frac{2}{3} \underline{A} \underline{u} \quad (2.21)$$

$$\begin{bmatrix} u_a \\ u_b \\ u_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(0^\circ) & \cos(120^\circ) & \cos(240^\circ) \\ \cos(-90^\circ) & \cos(120^\circ - 90^\circ) & \cos(240^\circ - 90^\circ) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}$$

onde $\underline{A}$ representa a respectiva matriz de transformação e o termo $2/3$ corresponde ao fator de escala para que as grandezas do sistema ortogonal tenham a mesma magni-

tude do sistema trifásico. Para se reconstruir o sistema trifásico a partir do sistema ortogonal é necessário o cálculo de matriz inversa $\underline{A}^{-1}$.

Assim como no modelo matricial trifásico, faz-se necessário obter todas equações por fase, para a tensão e fluxo e também o conjugado para se obter o modelo dinâmico completo. E como fora visto no modelo trifásico, será também utilizado o referencial único para este tipo de modelagem.

Fazendo a devida transformação de eixo trifásico para ortogonal, a partir das equações na notação trifásica em referencial único (2.8), chega-se as seguintes equações de tensão, fluxo e conjugado:

$$\underline{u}_1 = \underline{A} \underline{u}_1 = R_1 \underline{A} \underline{i}_1 + \underline{A} \frac{d}{dt} \underline{I}_1 + \underline{w}_k \underline{A} \underline{K} \underline{A}^{-1} \underline{A} \underline{I}_1 \quad (2.22-a)$$

$$\underline{u}_2 = \underline{A} \underline{u}_2 = R_2 \underline{A} \underline{i}_2 + \underline{A} \frac{d}{dt} \underline{I}_2 + (\underline{w}_k - \underline{w}_{mec}) \underline{A} \underline{K} \underline{A}^{-1} \underline{A} \underline{I}_2 \quad (2.22-b)$$

introduzindo-se a matriz $\underline{K}'$ definida por:

$$\underline{K}' = \underline{A} \underline{K} \underline{A}^{-1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.23)$$

a equação de tensão torna-se:

$$\underline{u}_1 = R_1 \underline{i}_1 + \frac{d}{dt} \underline{I}_1 + \underline{w}_k \underline{K}' \underline{I}_1 \quad (2.24-a)$$

$$\underline{u}_2 = R_2 \underline{i}_2 + \frac{d}{dt} \underline{I}_2 + (\underline{w}_k - \underline{w}_{mec}) \underline{K}' \underline{I}_2 \quad (2.24-b)$$

as equações de fluxo não se alteram permanecendo

$$\underline{I}_1 = L_1 \underline{i}_1 + L_H \underline{i}_2 \quad (2.25-a)$$

$$\underline{I}_2 = L_H \underline{i}_1 + L_2 \underline{i}_2 \quad (2.25-b)$$

e a equação do conjugado torna-se

$$m_d = -NP \underline{I}_1^T \underline{A}^T \underline{A}^{-1T} \underline{K} \underline{A}^{-1} \underline{A} \underline{i}_1 \quad (2.26)$$

onde

$$\underline{A}^T \underline{A}^{-1T} = \underline{A} \underline{A}^{-1} = \underline{I} \quad (2.27)$$

e considerando-se

$$\underline{\mathbf{I}}_1^T \underline{\mathbf{A}}^T = \underline{\mathbf{I}}_1^T \quad (2.28-a)$$

$$\underline{\mathbf{A}} \underline{\dot{\mathbf{i}}}_1 = \underline{\dot{\mathbf{i}}}_1 \quad (2.28-b)$$

$$\underline{\mathbf{A}}^{-1T} \underline{\mathbf{K}} \underline{\mathbf{A}}^{-1} = \frac{3}{2} \underline{\mathbf{K}}' \quad (2.28-c)$$

chega-se finalmente a,

$$m_d = -\frac{3}{2} NP \cdot \underline{\mathbf{I}}_1^T \underline{\mathbf{K}}' \underline{\dot{\mathbf{i}}}_1 \quad (2.29)$$

lembrando que as equações alternativas para o cálculo do conjugado, obtidas no modelo matricial trifásico, também valem no modelo matricial ortogonal.

Com isso, para se obter o modelo dinâmico completo em fluxo ou em corrente, faz-se necessário o mesmo procedimento adotado na modelagem trifásica, ou seja, partindo-se das equações (2.24-a,b), e com auxílio das equações (2.17), estas também permanecem inalteradas, chegam-se as seguintes equações:

$$\begin{aligned} \underline{u}_1 &= R_1 \underline{\dot{\mathbf{i}}}_1 + \frac{d}{dt} \underline{\mathbf{I}}_1 + \mathbf{w}_k \underline{\mathbf{K}} \underline{\mathbf{I}}_1 = \\ \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \end{bmatrix} &= \left(\frac{R_1}{\mathbf{s} L_1} \begin{bmatrix} \mathbf{I}_{1a} \\ \mathbf{I}_{1b} \\ \mathbf{I}_{10} \end{bmatrix} - \frac{R_1 L_H}{\mathbf{s} L_1 L_2} \begin{bmatrix} \mathbf{I}_{2a} \\ \mathbf{I}_{2b} \\ \mathbf{I}_{20} \end{bmatrix} \right) + \frac{d}{dt} \begin{bmatrix} \mathbf{I}_{1a} \\ \mathbf{I}_{1b} \\ \mathbf{I}_{10} \end{bmatrix} + \mathbf{w}_k \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{1a} \\ \mathbf{I}_{1b} \\ \mathbf{I}_{10} \end{bmatrix} \end{aligned} \quad (2.30-a)$$

$$\begin{aligned} \underline{u}_2 &= R_2 \underline{\dot{\mathbf{i}}}_2 + \frac{d}{dt} \underline{\mathbf{I}}_2 + \mathbf{w}_2 \underline{\mathbf{K}} \underline{\mathbf{I}}_2 = \\ \begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{20} \end{bmatrix} &= 0 = \left(-\frac{R_2 L_H}{\mathbf{s} L_1 L_2} \begin{bmatrix} \mathbf{I}_{1a} \\ \mathbf{I}_{1b} \\ \mathbf{I}_{10} \end{bmatrix} + \frac{R_2}{\mathbf{s} L_2} \begin{bmatrix} \mathbf{I}_{2a} \\ \mathbf{I}_{2b} \\ \mathbf{I}_{20} \end{bmatrix} \right) + \frac{d}{dt} \begin{bmatrix} \mathbf{I}_{2a} \\ \mathbf{I}_{2b} \\ \mathbf{I}_{20} \end{bmatrix} + \mathbf{w}_2 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{2a} \\ \mathbf{I}_{2b} \\ \mathbf{I}_{20} \end{bmatrix} \end{aligned} \quad (2.30-b)$$

onde:

$$\mathbf{w}_2 = \mathbf{w}_k - \mathbf{w}_{mec} \quad (2.31)$$

Fazendo o mesmo procedimento para corrente, utilizando as equações (2.24-a, b) e substituindo o fluxo das equações (2.25-a, b), obtêm-se as seguintes equações:

$$\underline{u}_1 = R_1 \underline{i}_1 + \frac{d}{dt} \underline{L}_1 + \underline{w}_k \underline{K} \underline{L}_1 =$$

$$\begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{10} \end{bmatrix} = R_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \end{bmatrix} + \frac{d}{dt} \left(L_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \end{bmatrix} + L_H \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{20} \end{bmatrix} \right) + \underline{w}_k \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(L_1 \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \end{bmatrix} + L_H \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{20} \end{bmatrix} \right)$$

(2.32-a)

$$\underline{u}_2 = R_2 \underline{i}_2 + \frac{d}{dt} \underline{L}_2 + \underline{w}_2 \underline{K} \underline{L}_2 =$$

$$\begin{bmatrix} u_{2a} \\ u_{2b} \\ u_{20} \end{bmatrix} = 0 = R_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{20} \end{bmatrix} + \frac{d}{dt} \left(L_H \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \end{bmatrix} + L_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{20} \end{bmatrix} \right) + \underline{w}_2 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(L_H \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \end{bmatrix} + L_2 \begin{bmatrix} i_{2a} \\ i_{2b} \\ i_{20} \end{bmatrix} \right)$$

(2.32-b)

Obs.: No caso do motor de indução ser simétrico equilibrado ou ter o neutro desconectado, os termos referentes ao eixo zero deixam de existir.

2.5. NOTAÇÃO VETORIAL

A notação vetorial, provém da analogia de uso da teoria de fasores em análise de circuitos elétricos e de corrente alternada, onde é assumido que todas as grandezas são senoidais e em regime permanente. Sua adaptação para a modelagem dinâmica do motor de indução é obtida a partir do fato que as grandezas das máquinas elétricas são consideradas periódicas. Dessa forma introduz-se o conceito de fasor de espaço, o qual é adotado para designar as grandezas elétricas do motor de indução.

A notação fasor de espaço resulta então, que o sistema ortogonal $(\mathbf{a}, \mathbf{b}, 0)$, apresentado no item 2.4, seja considerado um plano complexo e todas as grandezas representadas neste plano, serão descritas pela composição de partes real e imaginária. Sendo assim, impõe-se que todas as grandezas elétricas sejam representadas como entidades complexas. A grandeza fluxo no plano complexo, por exemplo, será representada por:

$$\vec{I} = I_a + j I_b = \frac{2}{3} (I_a + \mathbf{a} I_b + \mathbf{a}^2 I_c) = \frac{2}{3} \begin{bmatrix} 1 & \mathbf{a} & \mathbf{a}^2 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad (2.33)$$

com

$$\mathbf{a} = e^{j120^\circ} = \cos(120^\circ) + j \sin(120^\circ) = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \quad (2.34)$$

Na equação (2.33), a fase “a” do sistema trifásico coincide com o eixo real do sistema complexo, e os termos α e α^2 indicam a direção dos fluxos nas fases “b” e “c” respectivamente, num determinado instante de tempo. Sendo que α corresponde a um deslocamento espacial de 120° e α^2 um deslocamento espacial de 240° .

É admitido que o motor de indução trifásico esteja sendo excitado por tensões trifásicas simétricas e imposto que o neutro jamais seja conectado. Por esta razão, não é considerado o eixo “0”.

Os fluxos por fase são representados por:

$$I_a = \hat{I} \cos(\omega t) \quad (2.35-a)$$

$$I_b = \hat{I} \cos(\omega t + 120^\circ) \quad (2.35-b)$$

$$I_c = \hat{I} \cos(\omega t + 240^\circ) \quad (2.35-c)$$

como $\cos(x) = 1/2(e^{jx} + e^{-jx})$, chega-se a:

$$\vec{I}(t) = \hat{I}(\cos(\omega t) + j \sin(\omega t)) = \hat{I} e^{j\omega t} \quad (2.36)$$

sendo $\hat{I}$ a amplitude máxima do fluxo por fase.

A expressão (2.36), representa que o vetor de fluxo resultante tem uma amplitude constante e gira com velocidade angular constante em torno da origem do plano complexo. Os vetores de espaço para tensão e corrente são definidos de maneira análoga ao do fluxo, assim:

$$\vec{u} = u_a + j u_b = \frac{2}{3} [1 \quad \mathbf{a} \quad \mathbf{a}^2] \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (2.37-a)$$

$$\vec{i} = i_a + j i_b = \frac{2}{3} [1 \quad \mathbf{a} \quad \mathbf{a}^2] \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (2.37-b)$$

Também por analogia, $\vec{i}$ e $\vec{u}$ têm um deslocamento angular constante com amplitude constante em torno da origem do plano complexo. Uma vez que o campo girante pode ser produzido por um conjunto de dois enrolamentos deslocados espacialmente de 90° entre si e excitados por grandezas do tipo cosseno e seno, respectivamente, a notação vetorial por fasor de espaço representam as componentes $\mathbf{a}$ e $\mathbf{b}$ nos enrolamentos ortogonais.

A obtenção das grandezas de fase a partir da notação vetorial deve ser calculada pela projeção do vetor de espaço nos três eixos de fase do sistema trifásico. Para o modelo de fluxo com a fase “a” na referência, têm-se:

$$\begin{bmatrix} \mathbf{I}_a \\ \mathbf{I}_b \\ \mathbf{I}_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_a \\ \mathbf{I}_b \end{bmatrix} \quad (2.38)$$

Assim como nas modelagens anteriormente mostradas, faz-se necessário às equações de tensão, fluxo concatenado e conjugado para se obter o modelo dinâmico completo do motor de indução trifásico. A figura 2.4 mostra o plano complexo com os possíveis referenciais que podem ser adotados.

FIGURA 2.4 - Plano complexo e referenciais arbitrários.

A transformação de referenciais para o referencial $\mathbf{k}$, será dada por:

- Grandezas de Estator

$$\vec{I}_1 = \vec{I}_{1s} e^{-j k} \quad (2.39)$$

- Grandezas de rotor

$$\vec{I}_2 = \vec{I}_{2r} e^{-j(k-q)} \quad (2.40)$$

E a velocidade angular do referencial será dada por:

$$\mathbf{w}_k = \frac{d}{dt} k \quad (2.41)$$

com

$$k = \mathbf{w}_k t + k_0 \quad (2.42)$$

Introduzindo a definição vetorial (2.34) e (2.38) nas equações básicas do motor de indução trifásico dadas por (2.2) e fazendo as devidas simplificações, obtêm-se as seguintes equações de tensão:

$$\vec{u}_{1s} = R_1 \vec{i}_{1s} + \frac{d}{dt} \vec{I}_{1s} \quad (2.43-a)$$

$$\vec{u}_{2r} = R_2 \vec{i}_{2r} + \frac{d}{dt} \vec{I}_{2r} \quad (2.43-b)$$

Para as equações de fluxo, baseado em (2.4-b) e (2.6), têm-se

$$\begin{aligned} \vec{I}_{1s} = & \frac{2}{3} \left[(l_1 [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{T}_0(0) + l_{s1} [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{I}) \vec{i}_{1s} \right] + \\ & + m [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{T}_0(\mathbf{q}) \vec{i}_{2r} \end{aligned} \quad (2.44-a)$$

$$\begin{aligned} \vec{I}_{2r} = & \frac{2}{3} \left[(l_2 [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{T}_0(0) + l_{s2} [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{I}) \vec{i}_{2r} \right] + \\ & + m [1 \ \mathbf{a} \ \mathbf{a}^2] \underline{T}_0(-\mathbf{q}) \vec{i}_{1s} \end{aligned} \quad (2.44-b)$$

Considerando que

$$[1 \ \mathbf{a} \ \mathbf{a}^2] \underline{T}_0(\mathbf{q}) = \frac{3}{2} e^{j\mathbf{q}} [1 \ \mathbf{a} \ \mathbf{a}^2] \quad (2.45)$$

resulta

$$\vec{I}_{1s} = \left(\frac{3}{2} l_1 + l_{s1} \right) \vec{i}_{1s} + \frac{3}{2} m e^{j\mathbf{q}} \vec{i}_{2r} \quad (2.46-a)$$

$$\vec{I}_{2r} = \frac{3}{2} m e^{-j\mathbf{q}} \vec{i}_{1s} + \left(\frac{3}{2} l_2 + l_{s2} \right) \vec{i}_{2r} \quad (2.46-b)$$

com as devidas simplificações, a expressão (2.44) pode ser reescrita com sendo:

$$\vec{I}_{1s} = L_1 \vec{i}_{1s} + M e^{jq} \vec{i}_{2r} \quad (2.47-a)$$

$$\vec{I}_{2r} = M e^{-jq} \vec{i}_{1s} + L_2 \vec{i}_{2r} \quad (2.47-b)$$

Como nas modelagens já apresentadas, o modelo vetorial será também equacionado baseado no referencial único, para isto aplica-se as transformações (2.39) e (2.40) nas equações de tensão (2.43) e (2.47), lembrando que no referencial único será mantido apenas os índices “1” e “2” para as grandezas de rotor e estator, respectivamente. Obtendo assim:

$$\vec{u}_1 = \vec{u}_{1s} e^{-jk} = R_1 e^{-jk} \vec{i}_{1s} + e^{-jk} \frac{d}{dt} \vec{I}_{1s} \quad (2.48-a)$$

$$\vec{u}_2 = \vec{u}_{2r} e^{-j(k-q)} = R_2 e^{-j(k-q)} \vec{i}_{2r} + e^{-j(k-q)} \frac{d}{dt} \vec{I}_{2r} \quad (2.48-b)$$

Depois de se realizar o desenvolvimento matemático para a equação (2.48), obtêm-se a seguinte equação para a tensão de estator e rotor.

$$\vec{u}_1 = R_1 \vec{i}_1 + \frac{d}{dt} \vec{I}_1 + j \mathbf{w}_k \vec{I}_1 \quad (2.49-a)$$

$$\vec{u}_2 = R_2 \vec{i}_2 + \frac{d}{dt} \vec{I}_2 + j (\mathbf{w}_k - \mathbf{w}_{mec}) \vec{I}_2 \quad (2.49-b)$$

As equações de fluxo são as mesmas descritas em (2.9-a, b).

Fazendo a mesma analogia utilizada para os demais modelos, se o interesse for a variável de estado corrente, substitui-se a equação (2.17) diretamente em (2.49-a, b) e obtêm-se:

$$\vec{u}_1 = R_1 \vec{i}_1 + \frac{d}{dt} (L_1 \vec{i}_1 + L_H \vec{i}_2) + j \mathbf{w}_k (L_1 \vec{i}_1 + L_H \vec{i}_2) \quad (2.50-a)$$

$$\vec{u}_2 = R_2 \vec{i}_2 + \frac{d}{dt} (L_H \vec{i}_1 + L_2 \vec{i}_2) + j (\mathbf{w}_k - \mathbf{w}_{mec}) (L_H \vec{i}_1 + L_2 \vec{i}_2) \quad (2.50-b)$$

Para o cálculo do conjugado em notação vetorial, deve-se obter, primeiramente, a expressão da potência total no sistema ortogonal e impor as condições da notação vetorial, ou seja, que o plano ortogonal é um plano complexo e que o ponto de neutro não é conectado.

O conjugado produzido será dado por:

$$md = \frac{3}{2} NP. \text{Re} \left\{ j \vec{I}_1 \vec{i}_1^* \right\} \quad (2.51)$$

Levando-se em consideração que os termos de fluxo de dispersão não contribuem para a geração de conjugado, conclui-se que este pode ainda ser expresso pelas seguintes expressões:

$$md = \frac{3}{2} NP. \text{Im} \left\{ -\vec{I}_1 \vec{i}_1^* \right\} \quad (2.52-a)$$

$$md = \frac{3}{2} NP. \text{Im} \left\{ \vec{i}_1 \vec{I}_1^* \right\} \quad (2.52-b)$$

$$md = \frac{3}{2} NP. \text{Im} \left\{ -\vec{I}_H \vec{i}_1^* \right\} \quad (2.52-c)$$

$$md = \frac{3}{2} NP. \text{Im} \left\{ \vec{I}_H \vec{i}_2^* \right\} \quad (2.52-d)$$

$$md = \frac{3}{2} NP. \text{Im} \left\{ \vec{I}_2 \vec{i}_2^* \right\} \quad (2.52-e)$$

$$md = \frac{3}{2} NP. \text{Im} \left\{ -\vec{i}_2 \vec{I}_2^* \right\} \quad (2.52-f)$$

Algumas das vantagens da notação vetorial podem ser relacionadas como :

- Devido à representação vetorial, os vetores de corrente e fluxo proporcionam uma característica fisicamente espacial, pelo fato dessas entidades serem tratadas como variáveis complexas, tendo cada um módulo e fase, descrevendo assim, o comportamento instantâneo das mesmas;
- Conjugado eletromagnético produzido no motor de indução trifásico passará a ter uma representação visual, desde que o mesmo é proporcional ao produto das magnitudes do vetor de fluxo e de corrente e o seno do ângulo entre eles.
- As grandezas vetoriais podem ser usadas para quaisquer frequências e comportamento temporal das grandezas de fase e, portanto, permitem a análise do motor de indução trifásico quando excitado por conversores eletrônicos;

- Variações na amplitude e/ou frequência das grandezas de fase serão representadas na notação vetorial, respectivamente, por variações na amplitude e/ou velocidade angular do vetor de espaço;
- Deslocamentos de fase entre grandezas diferentes serão representadas na notação vetorial por deslocamentos angulares dos respectivos vetores de espaço.

Neste Capítulo, foram apresentados os modelos trifásicos, ortogonais e vetoriais, e como equacioná-los. Cada tipo de modelo, pode ser obtido, conforme já mencionado, utilizando como variáveis de estado fluxo ou corrente e adotando diversos tipos de referenciais.

Na literatura clássica, as possibilidades normalmente adotadas são os referenciais de estator fixo, cujo procedimento de transformação era denominado de “*Transformada de Clark*” e o referencial fixo no rotor conhecida como a “*Transformação de Park*”. No caso do referencial no rotor, o referencial gira com a velocidade angular mecânica do rotor, introduz-se uma simplificação no modelo de tal forma que as indutâncias mutuas, normalmente dependentes da posição angular, tornam-se constantes.

Os modelos matemáticos até aqui apresentados, foram e ainda são muito utilizados para os mais diversos fins, tanto em simulação quanto controle do motor de indução. Cada uma das modelagens apresentadas tem sua aplicação. Por exemplo, o modelo trifásico serve para uma simulação de uma falha de tensão em fase, mas tem a desvantagem de ser um modelo de sétima ordem. O modelo “Alfa Beta Zero” também é de sétima ordem, mas apresenta um número menor de variáveis para descrever o comportamento dinâmico. E por último, o modelo vetorial que reduz a ordem do modelo para um sistema de quinta ordem, mas que também necessita um arranjo matemático para resolução.

Como alternativa, será apresentada no próximo Capítulo uma técnica de modelagem baseada no conceito do *modelo vetorial dinâmico complexo*, que facilita a modelagem e a construção do diagrama de blocos na representação, além de propiciar uma redução de ordem de modelo, uma vez que trabalha com entidades complexas.

Capítulo 3

MODELO VETORIAL COMPLEXO PARA O MOTOR DE INDUÇÃO

3.1. INTRODUÇÃO

O comportamento dinâmico do motor de indução trifásico pode ser analisado de maneira bastante coerente pela introdução de sistemas dinâmicos com coeficientes complexos. A motivação para se utilizar a notação de sistemas dinâmicos de coeficientes complexos vem da representação do modelo vetorial do motor de indução trifásico, o qual é caracterizado por grandezas complexas e é por esta razão que se adota a nomenclatura de Modelo Vetorial Complexo. Através de simples manipulação das equações vetoriais do modelo do motor de indução trifásico é possível se compor uma equação de estado complexa, evitando-se a manipulação algébrica de separação das entidades reais e imaginárias das equações diferenciais.

Os sistemas dinâmicos com coeficientes complexos apresentam um comportamento dinâmico muito peculiar se comparados com os análogos de coeficientes reais. No Anexo A, é apresentado o conceito de sistemas dinâmicos complexos com um exemplo de um sistema de primeira ordem. No caso do motor de indução trifásico, através de manipulação algébrica, chega-se a um sistema com duas equações diferenciais complexas, dando origem a um sistema dinâmico complexo de segunda ordem.

Na sequência será apresentada a definição de um sistema dinâmico complexo de segunda ordem e os procedimentos para representar o motor de indução como um sistema dinâmico de segunda ordem.

3.2. SISTEMA DINÂMICO COMPLEXO DE SEGUNDA ORDEM

Baseado na definição de sistema dinâmico complexo de primeira ordem tal como apresentado no Anexo A, estabelece-se que um sistema equivalente e genérico de segunda ordem será descrito por:

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_1 & c_{21} \\ c_{12} & a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.1)$$

onde x_1 e x_2 são dois estados complexos, bem como os elementos a_1 , a_2 , c_{12} e c_{21} . As excitações u_1 e u_2 podem também de natureza complexa ou simplesmente real. A eq. (3.1) representa a conhecida formulação de espaço de estados, sendo que neste caso, considera-se um espaço de estados complexos.

Admitindo-se que a excitação em (3.1) seja unicamente u_1 , a representação do sistema em (3.1) na forma de diagrama de blocos resulta tal como indicado na figura 3.1.

FIGURA 3.1 - Representação de um sistema dinâmico complexo de segunda ordem.

Conforme desenvolvido e apresentado no Anexo A, para o sistema dinâmico complexo de primeira ordem, a solução para os estados x_1 e x_2 da figura 3.1 é obtido como sendo dado por:

$$x_1(t) = u_0 \left\{ \frac{a_2}{b_1 b_2} + \frac{1}{b_1 - b_2} \left[\left(1 + \frac{a_2}{b_1} \right) e^{b_1 t} - \left(1 + \frac{a_2}{b_2} \right) e^{b_2 t} \right] \right\} \quad (3.2-a)$$

$$x_2(t) = u_0 \frac{c_{12}}{b_1 b_2} \left(1 + \frac{b_2}{b_1 - b_2} e^{b_1 t} - \frac{b_1}{b_1 - b_2} e^{b_2 t} \right) \quad (3.2-b)$$

onde a_1 , a_2 , b_1 e b_2 são constantes complexas, e portanto resultando que x_1 e x_2 são também grandezas complexas. O comportamento transitório de x_1 e x_2 no plano complexo neste caso será uma composição de duas espirais amortecidas.

3.3. OBTENÇÃO DO MODELO DO MOTOR DE INDUÇÃO COMO UM SISTEMA DINÂMICO COMPLEXO.

Admitindo que as componentes da grandeza $\underline{x}(t)$ na figura 3.1 sejam os fluxos $\vec{I}_1$ e $\vec{I}_2$ de estator e rotor, do motor de indução trifásico, obtêm-se a partir da figura 3.1:

$$\begin{aligned}\dot{\vec{I}}_1 &= \vec{u}_1 - \mathbf{a}_1 \vec{I}_1 + c_{21} \vec{I}_2 \\ \dot{\vec{I}}_2 &= 0 - \mathbf{a}_2 \vec{I}_2 + c_{12} \vec{I}_1\end{aligned}\quad (3.3)$$

Reescrevendo (3.3) em forma matricial, recai-se na representação de sistemas dinâmicos no espaço de estados, tal como:

$$\begin{bmatrix} \dot{\vec{I}}_1 \\ \dot{\vec{I}}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_1 & c_{21} \\ c_{12} & -\mathbf{a}_2 \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \vec{u}_1 \quad (3.4)$$

Por outro lado, baseado nas equações de tensão do modelo vetorial em referencial genérico (2.49), o motor de indução trifásico pode ser descrito por:

$$\begin{bmatrix} \vec{u}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} + \begin{bmatrix} \dot{\vec{I}}_1 \\ \dot{\vec{I}}_2 \end{bmatrix} + \begin{bmatrix} j\mathbf{w}_k & 0 \\ 0 & j(\mathbf{w}_k - \mathbf{w}_{mec}) \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} \quad (3.5-a)$$

sendo,

$$\begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} = \begin{bmatrix} L_1 & L_H \\ L_H & L_2 \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} \quad (3.5-b)$$

A partir de (3.5-b), explicitando-se as correntes no motor de indução trifásico em função dos fluxos, chega-se a:

$$\begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\mathbf{s}L_1} & -\frac{L_H}{\mathbf{s}L_1L_2} \\ -\frac{L_H}{\mathbf{s}L_1L_2} & \frac{1}{\mathbf{s}L_2} \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} \quad (3.6)$$

onde $\mathbf{s}$ é o fator de dispersão global, dado por:

$$\mathbf{s} = 1 - \frac{L_H^2}{L_1L_2} = 1 - \frac{1}{(1+\mathbf{s}_1)(1+\mathbf{s}_2)} \quad (3.7)$$

Substituindo-se (3.6) diretamente em (3.5-a) obtêm-se:

$$\begin{bmatrix} \vec{u}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{R_1}{sL_1} & -\frac{L_H}{L_2} \frac{R_1}{sL_1} \\ -\frac{L_H}{L_1} \frac{R_2}{sL_2} & \frac{R_2}{sL_2} \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} + \begin{bmatrix} j\mathbf{w}_k & 0 \\ 0 & j(\mathbf{w}_k - \mathbf{w}_{mec}) \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} + \begin{bmatrix} \dot{\vec{I}}_1 \\ \dot{\vec{I}}_2 \end{bmatrix} \quad (3.8)$$

ou, reescrevendo na forma de espaço de estados, chega-se na equação complexa de estados para a variável de estado fluxo:

$$\begin{bmatrix} \dot{\vec{I}}_1 \\ \dot{\vec{I}}_2 \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_1}{sL_1} + j\mathbf{w}_k\right) & \frac{L_H}{L_2} \frac{R_1}{sL_1} \\ \frac{L_H}{L_1} \frac{R_2}{sL_2} & -\left(\frac{R_2}{sL_2} + j(\mathbf{w}_k - \mathbf{w}_{mec})\right) \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \vec{u}_1 \quad (3.9)$$

Finalmente comparando-se (3.9) com (3.4), obtém-se:

$$\mathbf{a}_1 = \frac{R_1}{sL_1} + j\mathbf{w}_k \quad (3.10-a)$$

$$\mathbf{a}_2 = \frac{R_2}{sL_2} + j(\mathbf{w}_k - \mathbf{w}_{mec}) \quad (3.10-b)$$

$$c_{12} = \frac{L_H}{L_1} \frac{R_2}{sL_2} \quad (3.10-c)$$

$$c_{21} = \frac{L_H}{L_2} \frac{R_1}{sL_1} \quad (3.10-d)$$

Acrescentando a equação mecânica da velocidade, chega-se então ao modelo dinâmico vetorial complexo, conforme se mostra na eq. (3.11)

$$\begin{bmatrix} \dot{\vec{I}}_1 \\ \dot{\vec{I}}_2 \\ \dot{\mathbf{w}}_{mec} \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_1 & c_{21} & 0 \\ c_{12} & -\mathbf{a}_2 & 0 \\ 0 & 0 & -K_D/J \end{bmatrix} \begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \\ \mathbf{w}_m \end{bmatrix} + \begin{bmatrix} \vec{u}_1 \\ 0 \\ m_d/J \end{bmatrix} \quad (3.11)$$

O modelo do descrito por (3.11) usa como estado os fluxos de estator e de rotor na equação elétrica. Reescrevendo-se o modelo para a variável de estado corrente com auxílio de (3.6), chega-se ao modelo matemático dado por (3.12)

$$\begin{bmatrix} \dot{\bar{i}}_1 \\ \dot{\bar{i}}_2 \\ \dot{\mathbf{w}}_{mec} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} \bar{i}_1 \\ \bar{i}_2 \\ \mathbf{w}_{mec} \end{bmatrix} + \begin{bmatrix} \frac{u_1}{s L_1} \\ -\frac{L_H u_1}{s L_1 L_2} \\ \frac{m_d}{J} \end{bmatrix} \quad (3.12)$$

onde:

$$\begin{aligned} a_{11} &= \frac{-R_1}{s L_1} - j \frac{\mathbf{w}_k - (1-s)\mathbf{w}_2}{s} \\ a_{12} &= \frac{R_2(1-s)}{s L_H} - j \left(\frac{L_H \mathbf{w}_k}{s L_1} - \frac{L_2(1-s)\mathbf{w}_2}{s L_H} \right) \\ a_{21} &= \frac{R_1(1-s)}{s L_H} - j \left(\frac{L_H \mathbf{w}_k}{s L_2} - \frac{L_1(1-s)\mathbf{w}_2}{s L_H} \right) \\ a_{22} &= \frac{-R_2}{s L_2} - j \frac{\mathbf{w}_2 - (1-s)\mathbf{w}_k}{s} \\ s &= 1 - \frac{L_H^2}{L_1 L_2} \end{aligned} \quad (3.13)$$

Observa-se com base nas equações do modelo com variáveis de estado fluxo ou corrente, que a descrição do modelo completo é bastante compacta e simples. Com esta notação, verifica-se que se pode representar o motor de indução por um diagrama de blocos também muito simples, o que facilita a análise do ponto de vista de sistema dinâmico.

Para o caso da modelagem em função da variável de estado fluxo e referencial estacionário ($\mathbf{w}_k = 0$), a representação do modelo dinâmico em diagrama de blocos, é tal como indicado na figura 3.2.

FIGURA 3.2 - Diagrama de blocos para variável de estado fluxo com referencial estacionário.

Tomando como base o exemplo anterior e mudando-se apenas o referencial, de estacionário para síncrono ($\mathbf{w}_k = \mathbf{w}_1$), o diagrama de blocos torna-se tal como o apresentado na figura 3.3:

FIGURA 3.3 - Diagrama de blocos para a variável de estado fluxo com referencial síncrono.

Agora, tomando-se o referencial estacionário e alterando a variável de estado de fluxo para corrente, o novo diagrama de blocos será tal como indicado pela figura 3.4.

FIGURA 3.4 - Diagrama de blocos para a variável de estado corrente com referencial estacionário.

Seguindo com o descritivo já apresentado no trabalho, mostrado no exemplo da figura 3.4 e adotar-se o referencial síncrono ($\mathbf{w}_k = \mathbf{w}_1$), o novo diagrama de blocos será tal como o apresentado na figura 3.5.

FIGURA 3.5 - Diagrama de blocos para a variável de estado corrente com referencial síncrono.

As expressões e diagramas apresentados anteriormente são caracterizados como funções transferências complexas e assim como na representação através de diagrama de blocos, conferem à descrição matemática uma forma bastante compacta. Esse tipo de representação do modelo do motor de indução, tem sido bastante utilizado recentemente (Novotny & Wouterse , 1976; De Doncker & Novotny, 1988; Dalton & Gosbell, 1989; Holtz, 1995; Gataric & Garrigan, 1999; de Aguiar & Cad 1999a; 1999b e 1999c).

Observa-se a partir do diagrama da figura 3.2 uma considerável simplificação da representação do motor de indução. No caso das outras formas de modelagem, designadas clássicas, é praticamente impossível a obtenção de tal representação.

No Capítulo seguinte serão apresentados os procedimentos de preparação e de simulação dos modelos até aqui analisados e, também, a comparação do modelo complexo com os demais.

Capítulo 4

PROCEDIMENTOS E MÉTODOS DE RESOLUÇÃO

4.1. INTRODUÇÃO

Nos Capítulos anteriores foi apresentado o modelo matemático para o motor de indução em diversas abordagens matemáticas e referenciais. Para se obter os resultados que possibilitem a análise de desempenho ou de estratégias de controle para estes motores, necessita-se, então, solucionar as equações diferenciais, quer sejam reais ou complexas.

Dentro da área de conhecimento da Engenharia Elétrica existem diversos pacotes de programas de ampla divulgação, que manipulam e resolvem equações ou sistema de equações diferenciais ordinárias, dentre eles Maple, MathCad, SimnonTM, Simulink / Matlab® e Octave, entre outros. Como é usual, cada um destes programas possibilita a resolução numérica das equações diferenciais utilizando uma grande diversidade de algoritmos numéricos de domínio público e de reconhecida eficiência. O procedimento de resolução utilizado para todos os casos e pacotes estudados baseia-se no uso do método de integração numérica Runge-Kutta de 4^a e 5^a ordem.

Neste trabalho serão utilizados os programas SimnonTM, Simulink / Matlab® e Octave, sendo mais utilizado o Matlab® pelas razões a serem apresentadas no decorrer do Capítulo.

Dentre as inúmeras possibilidades de análise e de simulação do motor de indução, serão investigadas aquelas onde as variáveis de estado são a corrente e o fluxo. Cada um dos casos, será simulado com o referencial no estacionário (estator fixo) e síncrono (girante). Somente estes casos totalizam 16 diferentes modelos. Outras inúmeras possibilidades podem resultar quando são usados outros referenciais ou ainda misturam-se as variáveis de estados.

Por convenção, as rotinas utilizadas, serão designadas por um nome com 5 (cinco) letras como forma de identificação das mesmas em função das variáveis de estado, do modelo e do referencial utilizado. A primeira letra determinará o tipo de

variável de estado usada no modelo, ou seja, F para fluxo ou I para Corrente; as 3 (três) letras seguintes corresponderão ao tipo de notação utilizada, ou seja, TRI para notação trifásica, ABO para notação ortogonal, VET para notação vetorial e COM para notação complexa; a quinta e última letra indicará se a rotina será usada com referencial estacionário ou girante, tal que P indica referencial estacionário e G indica o girante. A extensão de designação de cada arquivo contendo as rotinas será usada para indicar o tipo de programa em que um determinado modelo foi resolvido, ou seja, “.t” para simulação utilizando o programa SimnonTM, “.m” será utilizada tanto para o programa Matlab®, quanto para o programa Octave, diferenciando-se apenas pelo sistema operacional a ser utilizado, e a extensão “.mdl” indicará que o programa utilizado é o Simulink / Matlab®.

Nos tópicos seguintes serão apresentadas descrições resumidas dos pacotes de programas utilizados e, também, uma descrição dos procedimentos de preparação das equações de modelos que irão compor os arquivos para as respectivas rotinas e pacotes de integração numérica.

Na tabela 1 a seguir, apresentam-se os dados do motor de indução utilizados durante a simulação.

Tabela 1 – Dados do Motor de Indução para Simulação.

$R_1 = 7.56 \, \Omega$	$R_2 = 3.84 \, \Omega$	$J = 0.027 \, \text{Kg m}^2$	$K_D = 0 \, \text{Nms/rad}$
$L_1 = 0.35085 \, \text{H}$	$L_{H1} = 0.33615 \, \text{H}$	$L_2 = 0.35085 \, \text{H}$	$NP = 2$

4.2. DESCRIÇÃO DOS PROGRAMAS PARA SIMULAÇÃO

Como já mencionado, serão utilizados 4 (quatro) diferentes pacotes de programas que dispõe de rotinas de resolução numérica de equações diferenciais. Nos programas SimnonTM e Octave serão simuladas as notações: trifásica, ortogonal e vetorial, uma vez que esses pacotes não conseguem manipular entidades complexas. Já, utilizando o programa Matlab® e o pacote Simulink / Matlab®, será possível realizar a simulação de todas as notações, uma vez que o mesmo, em suas versões mais recentes, consegue manipular entidades complexas.

Na tabela 2, mostrada a seguir, apresenta-se uma síntese de todos os tipos de simulações realizadas segundo o tipo de programa, tipos de modelo, tipos de variável

de estado e referencial. Nesta tabela estão referenciadas 56 diferentes formas de simulação de um motor de indução trifásico em 4 tipos de programas. Em seguida serão abordados alguns aspectos relativos a cada um dos pacotes de programas utilizados.

Tabela 2 – Indicativo dos Programas e seus respectivos programas.

Modelo	Programa Variável de Estado	Simnon TM (extensão .t)		Octave (extensão .m)		Matlab [®] (extensão .m)		Simulink / Matlab [®] (extensão .mdl)	
		Parado	Girando	Parado	Girando	Parado	Girando	Parado	Girando
Trifásico	Fluxo	FTRIP	FTRIG	FTRIP	FTRIG	FTRIP	FTRIG	FTRIP	FTRIG
Trifásico	Corrente	ITRIP	FTRIG	ITRIP	FTRIG	ITRIP	FTRIG	ITRIP	FTRIG
Ortogonal	Fluxo	FAB0P	FAB0G	FAB0P	FAB0G	FAB0P	FAB0G	FAB0P	FAB0G
Ortogonal	Corrente	IAB0P	FAB0G	IAB0P	FAB0G	IAB0P	FAB0G	IAB0P	FAB0G
Vetorial	Fluxo	FVETP	FVETG	FVETP	FVETG	FVETP	FVETG	FVETP	FVETG
Vetorial	Corrente	IVETP	FVETG	IVETP	FVETG	IVETP	FVETG	IVETP	FVETG
Complexo	Fluxo	*	*	*	*	FCOMP	FCOMG	FCOMP	FCOMG
Complexo	Corrente	*	*	*	*	ICOMP	FCOMG	ICOMP	FCOMG

* Não aplicável para este programa.

4.2.1. SIMULAÇÃO COM O PROGRAMA SIMNONTM

Este é um programa desenvolvido pela SSPA Systems, e é muito utilizado em simulações de sistemas lineares e não-lineares e resolução de equações diferenciais, por ser de fácil manipulação e não exigir conhecimento de qualquer tipo de linguagem de programação, além resolver sistemas de equações num tempo consideravelmente rápido, pois o programa utiliza o menor número possível de passos para poder simular as equações diferenciais. O SimnonTM é um pacote utilizado em ambiente MS-Windows, sendo que a versão utilizada neste trabalho será a 1.03/Regular, embora já exista comercialmente a versão 3.0, disponível para “download” em versão demo, o qual foi utilizado para gerar os gráficos e conseguiu-se um resultado um pouco melhor.

O SimnonTM é muito recomendado para resolução de equações diferenciais na forma de equação de estado tendo diversas opções, em termos qualitativos, de rotinas de integração numérica, com controle de “passo” e definição de limite de erro (10^{-6}).

Para preparação e execução dos programas feito usando o SimnonTM, se faz necessário escrever as equações diferenciais de cada fase ou eixo a ser modelado, declarando quais são as variáveis de estado de cada equação diferencial, no arquivo

principal. Este arquivo conterá, além das equações diferenciais, os parâmetros do motor a ser modelado. Os modelos a serem simulados conforme mostrado na Tabela 2, são o trifásico, ortogonal e o vetorial, e as listagens dos programas podem ser vistas no apêndice B.

As saídas geradas utilizando o programa SimnonTM serão designadas, por convenção, de acordo com o modelo e a fase ou eixo em questão, ou seja, “f” para fluxo ou “i” para corrente, 1 e 2 para representar estator ou rotor, respectivamente e “a”, “b” e “c” ou “0” (zero) para determinar a fase ou eixo representativo. Como todos os comandos são realizados direto na linha de comando, pode-se criar, como foi feito, um programa principal que executará todos os comandos, tanto para simulação quanto para visualização dos resultados. Seus recursos de pós-processamento ainda deixam a desejar, levando-se em consideração os procedimentos necessários para criar títulos ao gráfico e aos eixos. E a resolução do gráfico. Deve-se ressaltar também, o fato de caso seja necessário utilizar o gráfico em algum processador de texto a figura será utilizada como uma figura “*bitmap*”, fato esse que também dificulta sua edição.

4.2.2. SIMULAÇÃO COM O PROGRAMA OCTAVE

Este programa utiliza o conceito de programa livre (com código aberto), ou seja, não tem um fabricante específico e conta com a colaboração dos usuários para sua atualização e expansão. Foi inicialmente desenvolvido por volta de 1988, por James B. Rawlings da Universidade de Wisconsin-Madison e John G. Ekerdt da Universidade do Texas, para uso em pós-graduação na área de Reatores Químicos e depois com a colaboração de seus usuários foi expandido para as demais áreas, obtendo grande evolução em engenharia elétrica na parte sistemas não-lineares. Seu ambiente de trabalho é o sistema operacional Linux. Assim como o SimnonTM, o Octave também utiliza uma linha de comando, porém com uma interface gráfica inferior, pois trabalha em uma janela semelhante a dos programas desenvolvido para MS-DOS, sendo que a versão aqui utilizada será a 2.0.16. Por se tratar de um programa onde os usuários o atualizam conforme suas necessidades, a versão mais atual já consegue manipular entidades complexas. Entretanto, sua rotina de integração

numérica para resolução de equações diferenciais não resulta grandezas complexas, mantendo assim o programa inadequado para o uso no modelo vetorial complexo.

Os modelos a serem simulados conforme Tabela 2 são trifásico, ortogonal e vetorial, descritos em variável de estado de fluxo e corrente e com referencial estacionário e também girando com velocidade síncrona.

A saída dos resultados, tal como indicado nos programas apresentados no apêndice B, será em forma de matriz de 7 (sete) colunas para os modelos trifásico e ortogonal, onde as 3 (três) primeiras colunas representam a variável de estado, fluxo ou corrente dependendo do modelo em questão, no estator, as outras 3 (três) de rotor sendo uma para cada fase ou eixo, e a última coluna representa a velocidade mecânica do motor de indução. Para o modelo vetorial são 5 (cinco) colunas utilizando a mesma nomenclatura que no caso ortogonal com diferença que não existirá o eixo 0 (zero). Já o conjugado eletromagnético, por não ser um estado, não tem como obter seu valor como uma saída, para isto é utilizado do artifício de gerar-se uma equação que calcule o seu valor depois que tiver sido feita a simulação. Cada linha das matrizes representa um valor de cada estado para um determinado tempo de simulação.

Os programas criados para resolução do modelo matemático do motor de indução são muito parecidos com a estrutura das rotinas desenvolvidas para o Matlab®, a ser discutido a seguir, diferenciando-se unicamente pelo procedimento de entrada de alguns parâmetros.

4.2.3. SIMULAÇÃO COM O PROGRAMA MATLAB®

Este é um programa desenvolvido pela MathWorks Corp., muito difundido no meio acadêmico e com grande número de ferramentas para aplicações em engenharia elétrica. O Matlab® em suas versões mais atualizadas, ainda possui a vantagem de manipular entidades complexas, o que serviu como incentivo para a realização deste trabalho. A versão utilizada nesta simulação é a versão 5.2.1, entretanto já existe comercialmente a mais nova versão que é a 5.3.1.

As variedades de recursos de manipulação das saídas, ajudam em muito a interpretação dos resultados, possibilitando comprovar às teorias aplicadas ao motor de indução. Assim como o programa Octave, as saídas das rotinas geradas no Matlab® também são na forma de matrizes e o procedimento para construção dos gráficos será

a mesma utilizada no Octave. Entretanto, como já foram citados, os recursos disponíveis no Matlab® são melhores. Outra vantagem que deve ser citada quanto ao uso do Matlab®, é o fato do mesmo possuir uma interface gráfica denominada *GUI* (*Graphic User Interface*) muito amigável e que torna a comunicação homem/máquina muito mais agradável.

4.2.4. SIMULAÇÃO COM O PROGRAMA SIMULINK / MATLAB®

Este é um pacote que acompanha o programa Matlab®, também desenvolvido pela MathWorks Corp., muito conhecido no meio acadêmico e com grande número de aplicações em engenharia elétrica. O Simulink tem como vantagem principal o fato de ser um programa gráfico e de fácil manipulação. O programa possui alguns blocos de funções prontos para a criação dos modelos e, também, possibilita a criação de funções e/ou rotinas auxiliares através de blocos denominados “*S-Functions*”. Para a realização da simulação da notação complexa com este pacote, fez-se necessário utilizar o recurso de criar “*S-Functions*” para se gerar os blocos de manipulação de variáveis complexas.

As variáveis de saída geradas no Simulink / Matlab®, também são na forma de matrizes. Entretanto, não obedecem ao mesmo padrão dos programas Octave e Matlab®, que têm uma sequência de saídas conforme as rotinas criadas, ou seja, no Octave e Matlab® as saídas são conforme a matriz de estado criada nos programas, já no Simulink a sequência de saída é correspondente à sequência de construção das variáveis de estado de saída.

Assim como no Matlab®, todos os modelos serão simulados, ver tabela 2, e para o desenvolvimento do modelo complexo, conforme já explicado, são criadas as rotinas FLUX1.M e FLUX2.M para a variável de estado complexa fluxo e CORR1.M e CORR2.M para a variável de estado corrente, para estator e rotor respectivamente. Essas rotinas serão utilizadas tanto nos modelos com referencial estacionário, quanto nos síncronos, uma vez que por se tratar de diagramas de blocos, basta alterar o valor da frequência de estator (ω_k) para obter-se os dois referenciais, com as equações de estado permanecendo inalteradas.

4.3. PREPARAÇÃO DOS MODELOS PARA RESOLUÇÃO NUMÉRICA

Segundo os procedimentos específicos de cada programa de resolução numérica, exige-se que o modelo descrito pelas equações diferenciais sejam reescritos na forma de equações de estado, ou seja, através de um sistema de equações diferenciais de primeira ordem organizado em forma matricial.

Para se adequar a descrição de modelos aos critérios dos pacotes de programa, será desenvolvido a seguir uma descrição da preparação das equações de cada modelo na forma de um modelo de estado. Em cada caso, as variáveis de estados poderão ser o fluxo ou a corrente no motor de indução.

4.3.1. NOTAÇÃO TRIFÁSICA

Baseado nas equações (2.18-a, b) onde a variável de estado é o fluxo de estator e rotor, isolando as derivadas de fluxo de estator e rotor, obtém-se o modelo em forma de equação de estado tal como:

$$\begin{bmatrix} \dot{I}_{1a} \\ \dot{I}_{1b} \\ \dot{I}_{1c} \\ \dot{I}_{2a} \\ \dot{I}_{2b} \\ \dot{I}_{2c} \\ \dot{W}_{mec} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{L_1} & \frac{w_k}{\sqrt{3}} & -\frac{w_k}{\sqrt{3}} & \frac{R_1 L_H}{s L_1 L_2} & 0 & 0 & 0 \\ -\frac{w_k}{\sqrt{3}} & -\frac{R_1}{L_1} & \frac{w_k}{\sqrt{3}} & 0 & \frac{R_1 L_H}{s L_1 L_2} & 0 & 0 \\ \frac{w_k}{\sqrt{3}} & -\frac{w_k}{\sqrt{3}} & -\frac{R_1}{L_1} & 0 & 0 & \frac{R_1 L_H}{s L_1 L_2} & 0 \\ \frac{R_2 L_H}{s L_1 L_2} & 0 & 0 & -\frac{R_2}{L_2} & \frac{w_2}{\sqrt{3}} & -\frac{w_2}{\sqrt{3}} & 0 \\ 0 & \frac{R_2 L_H}{s L_1 L_2} & 0 & -\frac{w_2}{\sqrt{3}} & -\frac{R_2}{L_2} & \frac{w_2}{\sqrt{3}} & 0 \\ 0 & 0 & \frac{R_2 L_H}{s L_1 L_2} & \frac{w_2}{\sqrt{3}} & -\frac{w_2}{\sqrt{3}} & -\frac{R_2}{L_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_d}{J} \end{bmatrix} \cdot \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \\ I_{2a} \\ I_{2b} \\ I_{2c} \\ W_{mec} \end{bmatrix} + \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{1c} \\ 0 \\ 0 \\ 0 \\ \frac{m_d}{J} \end{bmatrix} \quad (4.1)$$

Para se obter o modelo dinâmico completo faz necessário também o cálculo do conjugado eletromagnético obtido através da seguinte expressão:

$$m_d = \frac{NP L_H}{\sqrt{3} s L_1 L_2} [I_{1a}(I_{2c} - I_{2b}) + I_{1b}(I_{2c} - I_{2a}) + I_{1c}(I_{2b} - I_{2a})] \quad (4.2)$$

Uma outra maneira de resolver as equações (2.18-a, b) é utilizando como variável de estado a corrente, e para isso, utilizam-se as equações (2.20-a, b) isolando as derivadas de corrente de estator e rotor o modelo de estado resultante é dado por:

$$\begin{bmatrix} \dot{i}_{1a} \\ \dot{i}_{1b} \\ \dot{i}_{1c} \\ \dot{i}_{2a} \\ \dot{i}_{2b} \\ \dot{i}_{2c} \\ \dot{\mathbf{w}}_{mec} \end{bmatrix} = \begin{bmatrix} a & b & -b & c & d & -d & 0 \\ -b & a & b & -d & c & d & 0 \\ b & -b & a & d & -d & c & 0 \\ e & f & -f & g & h & -h & 0 \\ -f & e & f & -h & g & h & 0 \\ f & -f & e & h & -h & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \\ i_{2a} \\ i_{2b} \\ i_{2c} \\ \mathbf{w}_{mec} \end{bmatrix} + \begin{bmatrix} \frac{u_{1a}}{\mathbf{s} L_1} \\ \frac{u_{1b}}{\mathbf{s} L_1} \\ \frac{u_{1c}}{\mathbf{s} L_1} \\ \frac{-L_H u_{1a}}{\mathbf{s} L_1 L_2} \\ \frac{-L_H u_{1b}}{\mathbf{s} L_1 L_2} \\ \frac{-L_H u_{1c}}{\mathbf{s} L_1 L_2} \\ \frac{m_d}{J} \end{bmatrix} \quad (4.3)$$

onde:

$$\begin{aligned} a &= -\frac{R_1}{\mathbf{s} L_1}; & b &= \frac{\mathbf{w}_k - (1-\mathbf{s})\mathbf{w}_2}{\sqrt{3}\mathbf{s}}; \\ c &= \frac{R_2 L_H}{\mathbf{s} L_1 L_2}; & d &= \frac{L_H}{\sqrt{3} L_1 \mathbf{s}} (\mathbf{w}_k - \mathbf{w}_2); \\ e &= \frac{R_1 L_H}{\mathbf{s} L_1 L_2}; & f &= \frac{L_H}{\sqrt{3} L_2 \mathbf{s}} (\mathbf{w}_2 - \mathbf{w}_k); \\ g &= -\frac{R_2}{\mathbf{s} L_2}; & h &= \frac{\mathbf{w}_2 - (1-\mathbf{s})\mathbf{w}_k}{\sqrt{3}\mathbf{s}} \end{aligned}$$

O conjugado eletromagnético para o modelo trifásico, utilizando corrente como variável de estado é obtido usando a seguinte equação:

$$m_d = \frac{-NP.L_H}{\sqrt{3}} [i_{2a}(i_{1c} - i_{1b}) + i_{2b}(i_{1a} - i_{1c}) + i_{2c}(i_{1b} - i_{1a})] \quad (4.4)$$

Tanto a equação para a variável de estado fluxo (4.1), quanto a de corrente (4.3), foram apresentadas de uma maneira genérica, pois para diferenciar o referencial estacionário $\mathbf{w}_k = 0$, do referencial síncrono $\mathbf{w}_k = \mathbf{w}_1$ basta entrar com o valor de $\mathbf{w}_k$ e utilizar a tensão de entrada adequada, ou seja, no referencial estacionário a tensão de entrada terá a amplitude desejada e defasada de 120° em cada fase. Para o

referencial síncrono visto pela fase “a”, a tensão de entrada será do tipo degrau e terá amplitude máxima e as outras duas fases serão negativas e com metade da amplitude da fase “a”. Esse procedimento é utilizado para os programas SimonTM, Octave e Matlab®. Já o programa Simulink / Matlab® sendo totalmente gráfico, necessita de dois diagrama de blocos, uma vez que os blocos que representa a tensão senoidal e a tensão degrau são diferentes.

Com base nas equações (4.1) a (4.4), foram desenvolvidos os programas para simulação do modelo trifásico em cada programa. Com exceção do pacote Simulink®, que por ser um programa totalmente gráfico utiliza as equações acima, porém será criado um diagrama de blocos para sua simulação conforme apresentado a seguir:

FIGURA 4.1 - Diagrama de blocos para a notação trifásica com variável de estado fluxo e referencial estacionário.

Os blocos “ f_{1a} ”, “ f_{1b} ”, “ f_{1c} ”, “ f_{2a} ”, “ f_{2b} ”, “ f_{2c} ”, “ M_d ”, “ w_{mec} ” e “ w_2 ”, mostrados nas figuras 4.1 e 4.2 são blocos agrupados contendo as equações (4.1) e (4.2)

apresentadas anteriormente. Os índices existentes nos blocos, por convenção, correspondem a: f para designar que a variável de estado é o fluxo, 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e as letras a , b e c para representar qual fase está sendo analisada. Isto é válido também para a figura 4.2 a seguir.

FIGURA 4.2 - Diagrama de blocos para a notação trifásica com variável de estado fluxo e referencial síncrono.

Para a notação trifásica com variável de estado corrente, o diagrama de blocos a ser utilizado pelo Simulink® será tal como mostrado na figura 4.3 para o referencial estacionário e na figura 4.4 para o referencial síncrono, sendo os blocos “ $i1a$ ”, “ $i1b$ ”, “ $i1c$ ”, “ $i2a$ ”, “ $i2b$ ”, “ $i2c$ ”, “ M_d ”, “ w_{mec} ” e “ w_2 ”, blocos agrupados contendo as equações (4.3) e (4.4) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: i para designar que a variável de estado é a corrente, 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e as letras a , b e c para representar qual fase está sendo analisada.

FIGURA 4.3 - Diagrama de blocos para a notação trifásica com variável de estado corrente e referencial estacionário.

FIGURA 4.4 - Diagrama de blocos para a notação trifásica com variável de estado corrente e referencial síncrono.

4.3.2. NOTAÇÃO ORTOGONAL

Baseado nas equações (2.30-a, b) onde a variável de estado é o fluxo de estator e rotor, isolando as derivadas de fluxo de estator e rotor, obtém-se o modelo em forma de equação de estado tal como:

$$\begin{bmatrix} \dot{I}_{1a} \\ \dot{I}_{1b} \\ \dot{I}_{10} \\ \dot{I}_{2a} \\ \dot{I}_{2b} \\ \dot{I}_{20} \\ \dot{w}_{mec} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{sL_1} & w_k & 0 & \frac{R_1 L_H}{sL_1 L_2} & 0 & 0 & 0 \\ -w_k & -\frac{R_1}{sL_1} & 0 & 0 & \frac{R_1 L_H}{sL_1 L_2} & 0 & 0 \\ 0 & 0 & -\frac{R_1}{sL_1} & 0 & 0 & \frac{R_1 L_H}{sL_1 L_2} & 0 \\ \frac{R_2 L_H}{sL_1 L_2} & 0 & 0 & -\frac{R_2}{sL_2} & w_2 & 0 & 0 \\ 0 & \frac{R_2 L_H}{sL_1 L_2} & 0 & -w_2 & -\frac{R_2}{sL_2} & 0 & 0 \\ 0 & 0 & \frac{R_2 L_H}{sL_1 L_2} & 0 & 0 & -\frac{R_2}{sL_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_d}{J} \end{bmatrix} \cdot \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{10} \\ I_{2a} \\ I_{2b} \\ I_{20} \\ w_{mec} \end{bmatrix} + \begin{bmatrix} u_{1a} \\ u_{1b} \\ u_{10} \\ 0 \\ 0 \\ 0 \\ \frac{m_d}{J} \end{bmatrix} \quad (4.5)$$

O conjugado eletromagnético para o modelo ortogonal, utilizando fluxo como variável de estado é obtido usando a seguinte equação:

$$m_d = -\frac{3}{2} \frac{NP.L_H}{sL_1 L_2} [(I_{1a} I_{2b}) - (I_{2a} I_{1b})] \quad (4.6)$$

Outra maneira de se resolver o modelo ortogonal é utilizando as equações (2.32-a, b) em função da variável de estado corrente de estator e rotor, têm-se a seguinte matriz de estados:

$$\begin{bmatrix} \dot{i}_{1a} \\ \dot{i}_{1b} \\ \dot{i}_{10} \\ \dot{i}_{2a} \\ \dot{i}_{2b} \\ \dot{i}_{20} \\ \dot{\mathbf{w}}_{mec} \end{bmatrix} = \begin{bmatrix} \frac{-R_1}{sL_1} & \frac{\mathbf{w}_k - (1-s)\mathbf{w}_2}{s} & 0 & \frac{R_2 L_H}{sL_2 L_2} & \frac{L_H}{sL_1}(\mathbf{w}_k - \mathbf{w}_2) & 0 & 0 \\ \frac{(1-s)\mathbf{w}_2 - \mathbf{w}_k}{s} & \frac{-R_1}{sL_1} & 0 & \frac{L_H}{sL_1}(\mathbf{w}_2 - \mathbf{w}_k) & \frac{R_2 L_H}{sL_2 L_2} & 0 & 0 \\ 0 & 0 & \frac{-R_1}{sL_1} & 0 & 0 & \frac{R_2 L_H}{sL_2 L_2} & 0 \\ \frac{R_1 L_H}{sL_2 L_2} & \frac{L_H}{sL_2}(\mathbf{w}_2 - \mathbf{w}_k) & 0 & \frac{-R_2}{sL_2} & \frac{\mathbf{w}_2 - (1-s)\mathbf{w}_k}{s} & 0 & 0 \\ \frac{L_H}{sL_2}(\mathbf{w}_k - \mathbf{w}_2) & \frac{R_1 L_H}{sL_2 L_2} & 0 & \frac{(1-s)\mathbf{w}_k - \mathbf{w}_2}{s} & \frac{-R_2}{sL_2} & 0 & 0 \\ 0 & 0 & \frac{R_1 L_H}{sL_2 L_2} & 0 & 0 & \frac{-R_2}{sL_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_d}{J} \end{bmatrix} \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{10} \\ i_{2a} \\ i_{2b} \\ i_{20} \\ \mathbf{w}_{mec} \end{bmatrix} + \begin{bmatrix} \frac{u_{1a}}{sL_1} \\ \frac{u_{1b}}{sL_1} \\ \frac{u_{10}}{sL_1} \\ \frac{-L_H u_{2a}}{sL_1 L_2} \\ \frac{-L_H u_{2b}}{sL_1 L_2} \\ \frac{-L_H u_{20}}{sL_1 L_2} \\ \frac{m_d}{J} \end{bmatrix} \quad (4.7)$$

O conjugado eletromagnético para o modelo ortogonal, utilizando corrente como variável de estado é obtido usando a seguinte equação:

$$m_d = -\frac{3}{2} NP.L_H [(i_{1a} i_{2b}) - (i_{1b} i_{2a})] \quad (4.8)$$

Assim como na notação trifásica, tanto a equação para a variável de estado fluxo (4.5), quanto a de corrente (4.7), foram apresentadas de uma maneira genérica, pois para diferenciar o referencial estacionário $\mathbf{w}_k = 0$, do referencial síncrono $\mathbf{w}_k = \mathbf{w}_1$ basta entrar com o valor de $\mathbf{w}_k$ e utilizar a tensão de entrada adequada, ou seja, no referencial estacionário as tensões de entrada terão as amplitudes desejadas e defasadas de 90° entre si, ou seja, uma *cos* e outra *sen* para os eixos “*alfa*” e “*beta*” e nula para o eixo “*zero*”. Para o referencial síncrono visto pelo eixo “*alfa*”, a tensão de entrada será do tipo degrau e terá amplitude máxima e neste ponto, sendo nula para os outros eixos. Esse procedimento é utilizado para os programas SimonTM, Octave e Matlab®. Já o programa Simulink / Matlab® sendo totalmente gráfico,

necessita de dois diagrama de blocos, uma vez que os blocos que representa a tensão senoidal e a tensão degrau são diferentes.

Assim como foi feito para a notação trifásica, para simular a notação ortogonal utilizando o Simulink / Matlab® faz-se necessário, também, o desenvolvimento do diagrama de blocos para simulação. Os blocos “*f1alfa*”, “*f1beta*”, “*f1zero*”, “*f2alfa*”, “*f2beta*”, “*f2zero*”, “*M_d*”, “*w_{mec}*” e “*w₂*”, mostrados nas figuras 4.5 e 4.6 são blocos agrupados contendo as equações (4.5) e (4.6) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: *f* para designar que a variável de estado é o fluxo, *1* para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e as palavras *alfa*, *beta* e *zero* para representar qual eixo está sendo analisado.

FIGURA 4.5 - Diagrama de blocos para a notação ortogonal com variável de estado fluxo e referencial estacionário.

FIGURA 4.6 - Diagrama de blocos para a notação ortogonal com variável de estado fluxo e referencial síncrono.

Para a notação ortogonal com variável de estado corrente o diagrama de blocos a ser utilizado pelo Simulink / Matlab® será tal como mostrado na figura 4.7 para o referencial estacionário e na figura 4.8 para o referencial síncrono, sendo os blocos “*i1alfa*”, “*i1beta*”, “*i1zero*”, “*i2alfa*”, “*i2beta*”, “*i2zero*”, “*M_d*”, “*w_{mec}*” e “*w₂*”, blocos agrupados contendo as equações (4.7) e (4.8) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: *i* para designar que a variável de estado é a corrente, *1* para dizer que é referente ao estator e *2* para dizer que é referente ao rotor e as palavras *alfa*, *beta* e *zero* para representar qual eixo está sendo analisado.

FIGURA 4.7 - Diagrama de blocos para a notação ortogonal com variável de estado corrente e referencial estacionário.

FIGURA 4.8 - Diagrama de blocos para a notação ortogonal com variável de estado corrente e referencial síncrono.

Com base nas equações (4.5) a (4.8) e os diagramas de blocos, foram desenvolvidos os programas para simulação do modelo ortogonal em cada programa.

4.3.3. NOTAÇÃO VETORIAL

Baseado nas equações (2.49-a, b) onde a variável de estado é o fluxo de estator e rotor, isolando as derivadas de fluxo de estator e rotor, e separando os termos em suas respectivas partes real e imaginária para simulação nos programas já citados. Obtém-se o modelo em forma de equação de estado tal como:

$$\begin{bmatrix} \dot{I}_{1a} \\ \dot{I}_{1b} \\ \dot{I}_{2a} \\ \dot{I}_{2b} \\ \dot{w}_{mec} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{sL_1} & w_k & \frac{R_1 L_H}{sL_1 L_2} & 0 & 0 \\ -w_k & -\frac{R_1}{sL_1} & 0 & \frac{R_1 L_H}{sL_1 L_2} & 0 \\ \frac{R_2 L_H}{sL_1 L_2} & 0 & -\frac{R_2}{sL_2} & w_2 & 0 \\ 0 & \frac{R_2 L_H}{sL_1 L_2} & -w_2 & -\frac{R_2}{sL_2} & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} I_{1a} \\ I_{1b} \\ I_{2a} \\ I_{2b} \\ w_{mec} \end{bmatrix} + \begin{bmatrix} u_{1a} \\ u_{1b} \\ 0 \\ 0 \\ \frac{m_d}{J} \end{bmatrix} \quad (4.9)$$

O conjugado eletromagnético para o modelo vetorial, utilizando fluxo como variável de estado é obtido usando a seguinte equação:

$$m_d = -\frac{3}{2} \frac{NP.L_H}{s L_1 L_2} \left[(I_{1a} I_{2b}) - (I_{2a} I_{1b}) \right] \quad (4.10)$$

Outra maneira de se resolver o modelo vetorial é utilizando as equações (2.50-a, b) em função da variável de estado corrente de estator e rotor, e como foi feito para a variável de estado fluxo, separando-se os termos, em parte real e imaginária, têm-se a seguinte matriz de estados:

$$\begin{bmatrix} \dot{i}_{1a} \\ \dot{i}_{1b} \\ \dot{i}_{2a} \\ \dot{i}_{2b} \\ \dot{\mathbf{w}}_{mec} \end{bmatrix} = \begin{bmatrix} \frac{R_1}{sL_1} & \frac{\mathbf{w}_k - (1-s)\mathbf{w}_2}{s} & \frac{R_2 L_H}{sL_1 L_2} & \frac{L_H}{sL_1} (\mathbf{w}_k - \mathbf{w}_2) & 0 \\ \frac{(1-s)\mathbf{w}_2 - \mathbf{w}_k}{s} & \frac{R_1}{sL_1} & \frac{L_H}{sL_1} (\mathbf{w}_2 - \mathbf{w}_k) & \frac{R_2 L_H}{sL_1 L_2} & 0 \\ \frac{R_1 L_H}{sL_1 L_2} & \frac{L_H}{sL_2} (\mathbf{w}_2 - \mathbf{w}_k) & -\frac{R_2}{sL_2} & \frac{\mathbf{w}_2 - (1-s)\mathbf{w}_k}{s} & 0 \\ \frac{L_H}{sL_2} (\mathbf{w}_k - \mathbf{w}_2) & \frac{R_1 L_H}{sL_1 L_2} & \frac{(1-s)\mathbf{w}_k - \mathbf{w}_2}{s} & -\frac{R_2}{sL_2} & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{2a} \\ i_{2b} \\ \mathbf{w}_{mec} \end{bmatrix} + \begin{bmatrix} \frac{u_{1a}}{sL_1} \\ \frac{u_{1b}}{sL_1} \\ -\frac{L_H u_{1a}}{sL_1 L_2} \\ -\frac{L_H u_{1b}}{sL_1 L_2} \\ \frac{m_d}{J} \end{bmatrix} \quad (4.11)$$

O conjugado eletromagnético para o modelo vetorial, utilizando corrente como variável de estado é obtido usando a seguinte equação:

$$m_d = -\frac{3}{2} NP L_H [(i_{1a} i_{2b}) - (i_{1b} i_{2a})] \quad (4.12)$$

Do mesmo modo que nas notações trifásica e ortogonal, tanto a equação para a variável de estado fluxo (4.9), quanto a de corrente (4.11), foram apresentadas de uma maneira genérica, pois para diferenciar o referencial estacionário $\mathbf{w}_k = 0$, do referencial síncrono $\mathbf{w}_k = \mathbf{w}_1$ basta entrar com o valor de $\mathbf{w}_k$ e utilizar a tensão de entrada adequada, ou seja, no referencial estacionário as tensões de entrada terão as amplitudes desejadas, defasadas de 90° entre si, ou seja, uma *coseno* e outra *seno* para os eixos “*alfa*” e “*beta*”, respectivamente. Para o referencial síncrono visto pelo eixo “*alfa*”, a tensão de entrada será do tipo degrau e terá amplitude máxima neste eixo e a outra nula. Esse procedimento é utilizado para os programas SimonTM, Octave e Matlab®. Já o programa Simulink / Matlab® sendo totalmente gráfico, necessita de dois diagrama de blocos, uma vez que os blocos que representa a tensão senoidal e a tensão degrau são diferentes.

Seguindo o mesmo procedimento das notações citadas, para simular a notação vetorial utilizando o Simulink® faz-se necessário, também, o desenvolvimento do diagrama de blocos para simulação. Os blocos “*f1alfa*”, “*f1beta*”, “*f2alfa*”, “*f2beta*”, “*M_d*”, “*w_{mec}*” e “*w₂*”, mostrados nas figuras 4.9 e 4.10 são blocos agrupados contendo

as equações (4.9) e (4.10) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: f para designar que a variável de estado é o fluxo, 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e as palavras *alfa* e *beta* para representar qual eixo está sendo analisado.

FIGURA 4.9 - Diagrama de blocos para a notação vetorial com variável de estado fluxo e referencial estacionário.

FIGURA 4.10 - Diagrama de blocos para a notação vetorial com variável de estado fluxo e referencial síncrono.

Para a notação vetorial com variável de estado corrente o diagrama de blocos a ser utilizado pelo Simulink® será tal como mostrado na figura 4.11 para o referencial estacionário e na figura 4.12 para o referencial síncrono, sendo os blocos “ $i1_{\alpha}$ ”, “ $i1_{\beta}$ ”, “ $i2_{\alpha}$ ”, “ $i2_{\beta}$ ”, “ M_d ”, “ w_{mec} ” e “ w_2 ”, blocos agrupados contendo as equações (4.11) e (4.12) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: i para designar que a variável de estado é a corrente, 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e as palavras *alfa* e *zero* para representar qual eixo está sendo analisado.

Modelo Vetorial de Corrente com Referencial Estacionário

FIGURA 4.11 - Diagrama de blocos para a notação vetorial com variável de estado corrente e referencial estacionário.

Modelo Vetorial de Corrente com Referencial Síncrono

FIGURA 4.12 - Diagrama de blocos para a notação vetorial com variável de estado corrente e referencial síncrono.

Com base nas equações (4.9) a (4.12) e os diagramas de blocos desenvolvidos, foram desenvolvidos os programas para simulação do modelo vetorial em cada programa.

4.3.4. NOTAÇÃO VETORIAL COMPLEXA

Baseado nas equações (2.49-a, b) onde a variável de estado é o fluxo de estator e rotor, isolando as derivadas de fluxo de estator e rotor, obtém-se o modelo em forma de equação de estado tal como:

$$\begin{bmatrix} \dot{\bar{I}}_1 \\ \dot{\bar{I}}_2 \\ \dot{\bar{w}}_{mec} \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_1}{sL_1} + j\omega_k\right) & \frac{R_1 L_H}{sL_1 L_2} & 0 \\ \frac{R_2 L_H}{sL_1 L_2} & -\left(\frac{R_2}{sL_2} + j\omega_2\right) & 0 \\ 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} \bar{I}_1 \\ \bar{I}_2 \\ \bar{w}_{mec} \end{bmatrix} + \begin{bmatrix} \bar{u}_1 \\ 0 \\ \frac{m_d}{J} \end{bmatrix} \quad (4.13)$$

O conjugado eletromagnético para o modelo vetorial complexo, utilizando fluxo como variável de estado é obtido usando a seguinte equação:

$$m_d = \frac{3}{2} \frac{NP.L_H}{L_1 L_2 - L_H^2} \cdot \text{Im}\{\bar{I}_1 \bar{I}_2^*\} \quad (4.14)$$

Seguindo o mesmo procedimento das notações anteriores, para simular a rotação complexa utilizando o Simulink® faz-se necessário, também, o desenvolvimento do diagrama de blocos para simulação. Os blocos “ c_{12} ”, “ c_{21} ” são os termos da diagonal secundária da equação (4.13), os blocos “ $flux1p$ ”, “ $flux2p$ ” correspondem as “*S-Functions*” desenvolvidas para resolver o sistema complexo mostrado na equação (4.13) e os blocos “ M_d ”, “ w_{mec} ” e “ w_2 ” mostrados nas figuras 4.13 e 4.14 são blocos agrupados contendo a equação (4.13) apresentadas anteriormente. Os índices existentes nos blocos, por convenção, correspondem a: 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e a palavra *flux* para representar qual a variável de estado está sendo analisada, neste caso fluxo.

FIGURA 4.13 - Diagrama de blocos para a notação complexa com variável de estado fluxo e referencial estacionário.

Modelo Complexo de Fluxo para Referencial Síncrono

FIGURA 4.14 - Diagrama de blocos para a notação complexa com variável de estado fluxo e referencial síncrono.

Outra maneira de se resolver o modelo complexo é utilizando as equações (2.50-a, b) em função da variável de estado corrente de estator e rotor, e como foi feito para a variável de estado fluxo, tem-se a seguinte matriz de estados:

$$\begin{bmatrix} \dot{\vec{i}}_1 \\ \dot{\vec{i}}_2 \\ \dot{\vec{w}}_{mec} \end{bmatrix} = \begin{bmatrix} \frac{-R_1}{sL_1} - j \frac{\vec{w}_k - (1-s)\vec{w}_2}{s} & \frac{R_2(1-s)}{sL_H} - j \left(\frac{L_H \vec{w}_k}{sL_1} - \frac{L_2(1-s)\vec{w}_2}{sL_H} \right) & 0 \\ \frac{R_1(1-s)}{sL_H} - j \left(\frac{L_H \vec{w}_k}{sL_2} - \frac{L_1(1-s)\vec{w}_2}{sL_H} \right) & \frac{-R_2}{sL_2} - j \frac{\vec{w}_2 - (1-s)\vec{w}_k}{s} & 0 \\ 0 & 0 & -\frac{K_D}{J} \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \\ \vec{w}_{mec} \end{bmatrix} + \begin{bmatrix} \frac{\vec{u}_1}{sL_1} \\ -\frac{L_H \vec{u}_1}{sL_1 L_2} \\ \frac{m_d}{J} \end{bmatrix} \quad (4.15)$$

O conjugado eletromagnético para o modelo vetorial, utilizando corrente como variável de estado é obtido usando a seguinte equação:

$$m_d = \frac{3}{2} NP.L_H . \text{Im}\{\vec{i}_1 \vec{i}_2^*\} \quad (4.16)$$

Assim como fora feito nas notações anteriores, tanto a equação para a variável de estado fluxo (4.13), quanto a de corrente (4.15), foram apresentadas de uma maneira genérica, pois para diferenciar o referencial estacionário $\mathbf{w}_k = 0$, do referencial síncrono $\mathbf{w}_k = \mathbf{w}_1$ basta entrar com o valor de $\mathbf{w}_k$ e utilizar a tensão de entrada adequada, ou seja, no referencial estacionário a tensão de entrada terá a amplitude desejada, porém será uma entrada complexa onde a parte real é constituída do termo *cos seno* e a parte imaginária é constituída do termo *seno*. Para o referencial síncrono a tensão de entrada será do tipo degrau e terá a amplitude máxima desejada para o termo real e valor nulo para a parte imaginária. Esse procedimento é utilizado para o programa Matlab®. Já o programa Simulink / Matlab®, sendo totalmente gráfico, necessita de dois diagrama de blocos, uma vez que os blocos que representa a tensão senoidal e a tensão degrau são diferentes.

Para a notação complexa com variável de estado corrente o diagrama de blocos a ser utilizado pelo Simulink® será tal como mostrado na figura 4.15 para o referencial estacionário e na figura 4.16 para o referencial síncrono. Os blocos “ c_{12} ”, “ c_{21} ” são os termos da diagonal secundaria da equação (4.13), os blocos “*flux1p*”, “*flux2p*” correspondem as *S-Functions* desenvolvidas para resolver o sistema complexo mostrado na equação (4.13) e os blocos “ $\mathcal{M}_d$ ”, “ $\mathbf{w}_{mec}$ ” e “ $\mathbf{w}_2$ ” mostrados nas figuras 4.13 e 4.14 são blocos agrupados contendo a equação (4.13) apresentadas anteriormente. Os índices existentes nos blocos por convenção, correspondem a: 1 para dizer que é referente ao estator e 2 para dizer que é referente ao rotor e a palavra *corr* para representar qual a variável de estado está sendo analisada, neste caso corrente.

FIGURA 4.15 - Diagrama de blocos para a notação complexa com variável de estado corrente e referencial estacionário.

FIGURA 4.16 - Diagrama de blocos para a notação complexa com variável de estado corrente e referencial síncrono.

Com base nas equações (4.13) a (4.16) e os diagramas de blocos desenvolvidos, foram escritos os programas para simulação do modelo vetorial complexo em cada programa.

Para a simulação e obtenção dos resultados utilizaram-se as equações e diagramas de blocos aqui apresentados, onde se nota que a notação vetorial complexa como sendo a mais indicada, pois a notação trifásica e ortogonal possui um número de equações diferenciais maior (sete) o que necessita de um maior número de manipulações algébricas, conseqüentemente um maior tempo para a construção das equações diferenciais e as rotinas para simulação. A criação dos diagramas de blocos para uso no Simulink, também necessita de um maior cuidado e tempo para construção. A

notação vetorial também necessita de manipulações algébricas para separar os termos complexos, além do que se parte da notação vetorial complexa para obter o modelo separado em real e imaginário.

No capítulo seguinte serão apresentados os resultados para todas as simulações das notações aqui citadas, bem como gráficos comparativos do tempo de simulação em cada programa, e assim, comentado o desempenho de cada programa com relação a cada modelo.

Capítulo 5

RESULTADOS E ANÁLISES

Neste capítulo serão apresentados os resultados da simulação dos modelos apresentados nos Capítulos 2 e 3, considerando-se a variável de estado utilizada, o referencial e o tipo de modelo. Em cada caso é prevista a realização da simulação através de cada um dos pacotes de programa e/ou rotinas indicados no Capítulo 4 (ver tabela 2).

Para esta fase do trabalho, serão documentados os resultados dos casos utilizando-se como variável de estado o fluxo e a corrente, e os referenciais estacionário e referencial síncrono, em todas as notações descritas no trabalho, ou seja, a notação trifásica, a notação ortogonal, a notação vetorial e a notação complexa. Como forma de organização, os resultados serão apresentados na ordem de citação dos modelos dos capítulos 2 e 3 e em cada caso, seguindo a ordem de pacotes de programas do Capítulo 4. Será avaliado o tempo de simulação de cada programa, com exceção do programa SimnonTM pelo fato do mesmo não apresentar método de medição do tempo de simulação, o pós-processamento dos resultados incluindo a apresentação dos mesmos e procedimentos para simulação de cada notação.

Os resultados em forma gráfica, descreverão o desempenho dinâmico durante a aceleração do motor a partir do repouso sem carga, ou como designado aqui, ensaio de partida. Os resultados serão caracterizados pelos gráficos da velocidade e do conjugado eletromagnético, ressaltando que tanto a velocidade quanto o conjugado eletromagnético independem, da variável de estado ou referencial adotado, sendo assim, será apresentado apenas um gráfico mostrando o comportamento da velocidade e do conjugado eletromagnético em função do tempo. Em seqüência virão os gráficos de fluxo ou de corrente, por fase ou por eixo, de acordo com variável de estado das equações elétricas. No caso da notação vetorial será apresentado o gráfico do comportamento transitório da variável de estado utilizada e também uma composição complexa das partes real e imaginária, assim como nas notações trifásica e ortogonal. Também serão utilizadas cores para indicar o comportamento transitório, sendo a cor

vermelha para o eixo real e a cor verde representando o eixo imaginário. E no caso da notação complexa serão apresentados a comportamento complexo da variável de estado e a decomposição em termos de suas partes real e imaginária. Em cada notação, serão mostrados os gráficos para a variável de estado fluxo e corrente.

Embora sejam apresentados comentários sobre os resultados de cada caso na ordem de citação, ao final serão resumidas as características gerais dos modelos e dos pacotes de programa para resolução dos modelos.

5.1. MODELO NA NOTAÇÃO TRIFÁSICA

O modelo na notação trifásica é dado pela equação (4.1) na variável de estado fluxo e pela equação (4.3) para a variável de estado corrente tanto para o referencial estacionário ($w_k = 0$), quanto o síncrono ($w_k = w_1$). Os resultados de simulação deste caso segundo os pacotes de programa citados são tais como indicados no tópico seguinte. Os resultados serão apresentados em forma de gráficos sendo mostrados na mesma figura com cores diferentes por fase, sendo a cor vermelha para a fase “a”, a cor verde para a fase “b” e a cor azul para a fase “c”. Este procedimento será usado tanto para o estator, quanto para o rotor em todos os softwares, exceto para o programa Simulink / Matlab® que possui cores pré-definidas e serão apresentadas da seguinte forma: fase “a” na cor amarela, fase “b” na cor magenta e fase “c” na cor ciano.

5.1.1. SIMULAÇÃO COM O PROGRAMA SIMNONTM

Com as condições acima citadas, preparou-se a simulação para este caso, segundo os procedimentos descritos no Capítulo 4 e gerando-se os resultados mostrados a seguir. As devidas rotinas de preparação destes programas são apresentadas no Apêndice B.1.1.

O SimnonTM tem um tempo de simulação curto (perceptível mas não mensurável), uma vez que ele gera um número de passos mínimo para a simulação.

Na figura 5.1, apresenta-se à velocidade do motor de indução, obtido graficamente através da variável velocidade do motor ($omgm$) e também o conjugado eletromagnético durante a aceleração do motor, plotando-se a variável m_d , ambos em

função do tempo, para a variável de estado fluxo e referencial estacionário. No referencial síncrono assim como para o caso da variável de estado corrente, não serão apresentados os gráficos da velocidade e do conjugado eletromagnético, pois a mudança de referencial ou de variável de estado não influenciará o comportamento dos gráficos citados. Esses gráficos, também não serão mostrados para os demais programas, tendo em vista que os gráficos são os mesmos só com a diferença na qualidade de apresentação o que pode ser comparado nos demais gráficos. Esse procedimento será o mesmo para as outras notações.

FIGURA 5.1 - Gráfico da velocidade $\times$ tempo [rad/s]
e do conjugado eletromagnético $\times$ tempo [Nm/s].

Na figura 5.2 apresenta-se o comportamento transitório do fluxo por fase durante a aceleração do ensaio de partida para o estator e rotor utilizando os referenciais estacionário (a) e síncrono (b). No referencial estacionário impõe-se ao motor uma alimentação senoidal trifásica equilibrada e admitiu-se a fase “ $\hat{u}$ ” como referência partindo de zero. Nota-se um comportamento bastante oscilatório durante o transitório na aceleração, até próximo de 0,4 s e que o fluxo parte do valor nulo até alcançar o valor de regime. Já no referencial síncrono, impõe-se ao motor uma alimentação do tipo tensão constante com amplitude máxima para a fase “ $\hat{u}$ ” e a metade deste valor com sinal trocado para as demais fases. Esta condição corresponde ao instante em que a senóide da fase “ $\hat{u}$ ” atinge seu valor máximo e as outras fases a metade de seus valores, porém negativos. Nota-se também, que durante a parte transitória, também próximo de 0,4 s, é evidenciado as amplitudes oscilantes ou variáveis. Em ambos os casos, após o transitório os fluxos de estator e rotor atingem um nível correspondente de fluxo, com a diferença que no referencial estacionário será um valor senoidal e no referencial síncrono um valor constante.

FIGURA 5.2 - Gráfico do fluxo por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono.

Na figura 5.3 apresenta-se o comportamento transitório da corrente por fase durante a aceleração do ensaio de partida para o estator e rotor utilizando o referencial estacionário (a) e síncrono (b). Na simulação deste caso usou-se a mesma consideração usada para a variável de estado fluxo, tanto no referencial estacionário, quanto no síncrono com relação à tensão de alimentação. Nota-se novamente, para o referencial estacionário, um comportamento bastante oscilatório durante o transitório, na aceleração, até próximo de 0,4 s e que a corrente parte do valor máxima, necessária para ocorrer à partida do motor, até alcançar o valor de regime, valor esse de magnetização, que no caso do estator é maior que no rotor tendo em vista não ser usada alimentação no rotor. E, no referencial síncrono, nota-se que durante a parte transitória, também próximo de 0,4 s, é evidenciada-se as amplitudes oscilantes ou variáveis, mas que ao contrário do que ocorre com o fluxo, parte de um valor máximo durante o transitório, e atinge um nível constante, em regime, correspondente à corrente nestas condições.

FIGURA 5.3 - Gráfico das correntes por fase [A/s] nos referenciais: a) estacionário; b) síncrono.

5.1.2. SIMULAÇÃO COM O PROGRAMA OCTAVE

Com as rotinas geradas e apresentadas no apêndice B.1.2 para este programa, foram realizadas as simulações do modelo trifásico, obtendo-se os resultados abaixo mostrados.

Conforme descrito no Capítulo 4, a variável de saída y do programa Octave é na forma de matriz, para se obter as respostas gráficas desejadas faz-se necessário plotar todos os valores da variável, ou seja, a coluna inteira para se conseguir isto, usa-se o comando $y(:,n)$ onde n representa a coluna desejada.

A figura 5.4 apresenta os mesmos resultados mostrados nas figuras 5.2 e 5.3, porém usando dessa vez o programa Octave como programa simulador. Comparando os gráficos a seguir, com os apresentados pelo SimnonTM, percebe-se uma melhor definição visual além de uma maior facilidade para nomear títulos e eixo dos gráficos. As interpretações utilizadas para o programa SimnonTM também são válidas para o Octave, ou seja, os tempos de simulação e valores em transitório e regime.

FIGURA 5.4 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.1.3. SIMULAÇÃO COM O PROGRAMA MATLAB®

Assim como foi feito Octave, as rotinas geradas e apresentadas no apêndice B.1.3 para este programa, permitiram as simulações do modelo trifásico, obtendo os resultados gráficos são apresentados a seguir.

A figura 5.5, assim como no Octave, também apresenta os mesmos resultados mostrados nas figuras 5.2 e 5.3, porém usando dessa vez o programa Matlab® como programa simulador. Comparando os gráficos a seguir, com os apresentados anteriormente, percebe-se uma melhor definição visual e no caso da necessidade de transportar para um processador de texto, possibilita sua edição, tornando-se assim manipulável, enquanto que, nos outros dois programas citados as figuras importadas são em formato de figura *“bitmap”* dificultando sua edição. As interpretações já utilizadas para os outros programas, também são válidas para o Matlab®, ou seja, os tempos de simulação e valores em transitório e regime. O procedimento de apresentação será o mesmo utilizado para o Octave, ou seja, são apresentadas 4 (quatro) figuras contendo o comportamento transitório do fluxo de estator e rotor no referencial estacionário (a); o comportamento transitório do fluxo de estator e rotor no referencial síncrono (b); o comportamento transitório da corrente de estator e rotor no referencial estacionário (c) e o comportamento transitório da corrente de estator e rotor no referencial síncrono (d). Todas obtidas utilizando como saída a seis primeiras variáveis de estado das rotinas *ftrip.m* ($y(:,1)$, $y(:,2)$, $y(:,3)$, $y(:,4)$, $y(:,5)$, $y(:,6)$) para a variável de estado fluxo e *itrip.m* para a variável de estado corrente, ambos com relação ao tempo de simulação (t). As cores utilizadas em cada gráfico seguem o padrão utilizado no Octave, ou seja, vermelho para fase *“a”*, verde para a fase *“b”* e azul para a fase *“c”*.

FIGURA 5.5 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.1.4. SIMULAÇÃO COM O PROGRAMA SIMULINK

Assim como foi feito para os demais programas, foram geradas as rotinas *ftrip.mdl*, *ftrig.mdl*, *itrip.mdl* e *itrig.mdl*, tomando como base os diagramas de blocos apresentados nas figuras 4.1 até 4.4, foram realizadas as simulações do modelo trifásico, obtendo os resultados gráficos são apresentados a seguir.

A figura 5.6 também apresenta os mesmos resultados mostrados nas figuras já apresentadas, porém usando dessa vez o programa Simulink / Matlab® como programa simulador. Comparando os gráficos a seguir, com os apresentados anteriormente, nota-se uma janela gráfica diferente das demais apresentadas, esse formato apresentado pelo Simulink / Matlab® origina-se de um bloco existente dentro de sua biblioteca, denominado “*scope*”. Entretanto, este programa permite manipular as variáveis de saída desejada em ambiente Matlab® o que o torna mais maleável. Outro fato que o diferencia dos demais programas é o das cores apresentadas no gráfico serem diferentes, e apresentado conforme já descrito, ou seja, fase “*a*” na cor amarela, fase “*b*” na cor magenta e fase “*c*” na cor ciano. As interpretações já utilizadas para os outros programas, também são válidas para o Simulink / Matlab®, ou seja, os tempos de simulação e valores em transitório e regime. O procedimento de apresentação dos gráficos, será o mesmo utilizado para o Octave, ou seja, serão apresentadas 4 (quatro) figuras contendo o comportamento transitório do fluxo de estator e rotor no referencial estacionário (a); o comportamento transitório do fluxo de estator e rotor no referencial síncrono (b); o comportamento transitório da corrente de estator e rotor no referencial estacionário (c) e o comportamento transitório da corrente de estator e rotor no referencial síncrono (d). E não serão apresentados os gráficos da velocidade do motor e o conjugado eletromagnético por razões já descritas.

FIGURA 5.6 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.2. MODELO NA NOTAÇÃO ORTOGONAL

Seguindo o mesmo procedimento utilizado para a notação trifásica, a notação ortogonal é dada pela equação (4.5) para a variável de estado fluxo e pela equação (4.7) para a variável de estado corrente, tanto para o referencial estacionário ($\mathbf{w}_k = 0$), quanto para o síncrono ($\mathbf{w}_k = \mathbf{w}_1$). Os resultados de simulação deste caso segundo os pacotes de programa citados são tais como indicados no tópico seguinte. Os resultados serão apresentados em forma de gráficos sendo mostrados na mesma figura com cores diferentes, onde a cor vermelha representará o eixo “*a*”, a cor verde representará o eixo “*b*” e a cor azul representará o eixo “*0*”. Este procedimento será feito tanto para o estator, quanto para o rotor em todos os softwares, exceto para o programa Simulink / Matlab® que possui cores pré-definidas e serão apresentadas da seguinte forma: o eixo “*a*” na cor amarela, o eixo “*b*” na cor magenta e o eixo “*0*” na cor ciano.

5.2.1. SIMULAÇÃO COM O PROGRAMA SIMNONTM

Repetindo o procedimento adotado para o modelo trifásico, obtiveram-se os seguintes resultados. Na figura 5.7 é apresentado o comportamento transitório da velocidade angular e o conjugado eletromagnético, obtidos através da rotina *fab0p.t* apresentada no apêndice B.2.1 e da equação (4.6).

FIGURA 5.7 - Gráfico da velocidade $\times$ tempo [rad/s] e conjugado eletromagnético $\times$ tempo [Nm/s].

Na figura 5.8 apresenta-se o comportamento transitório do fluxo por eixo durante a aceleração do ensaio de partida para o estator e rotor utilizando os referenciais estacionário (a) e síncrono (b). No referencial estacionário impõe-se ao motor

uma alimentação senoidal ortogonal equilibrada e nula para o eixo “0”, sendo assim, todas as entidades para o eixo “0” serão nulas, e admitiu-se o eixo “a” como referência partindo de zero. Nota-se um comportamento bastante oscilatório durante o transitório na aceleração, até próximo de 0,4 s e que o fluxo parte do valor nulo até alcançar o valor de regime. Já no referencial síncrono, impôs-se ao motor uma alimentação do tipo tensão constante com amplitude máxima para o eixo “a” e a nula para os demais eixos. Esta condição corresponde ao instante em que a senóide da fase “a” atinge seu valor máximo e as outras fases valores nulos. Nota-se também, que durante a parte transitória, também próximo de 0,4 s, é evidenciado as amplitudes oscilantes ou variáveis. Em ambos os casos, após o transitório os fluxos de estator e rotor atingem um nível correspondente de fluxo, com a diferença que no referencial estacionário será um valor senoidal e no referencial síncrono um valor constante.

FIGURA 5.8 - Gráfico dos fluxos por eixo [Wb/s] nos referenciais: a) estacionário; b) síncrono.

Na figura 5.9 apresenta-se o comportamento transitório da corrente por eixo durante a aceleração do ensaio de partida para o estator e rotor utilizando o referencial estacionário (a) e síncrono (b). Na simulação deste caso usou-se a mesma consideração usada para a variável de estado fluxo, tanto no referencial estacionário, quanto

no síncrono com relação à tensão de alimentação. Nota-se novamente, para o referencial estacionário, um comportamento bastante oscilatório durante o transitório, na aceleração, até próximo de 0,4 s e que a corrente parte do valor máxima, necessária para ocorrer à partida do motor, até alcançar o valor de regime, valor esse residual, que no caso do estator é maior que no rotor tendo em vista não ser usada alimentação no rotor. E, no referencial síncrono, nota-se que durante a parte transitória, também próximo de 0,4 s, é evidenciado as amplitudes oscilantes ou variáveis, mas que ao contrário do que ocorre com o fluxo, parte de um valor máximo durante o transitório, e atinge um nível constante, em regime, correspondente à corrente.

FIGURA 5.9 - Gráfico das correntes por eixo [A/s] nos referenciais: a) estacionário; b) síncrono.

5.2.2. SIMULAÇÃO COM O PROGRAMA OCTAVE

Com as rotinas geradas e apresentadas no apêndice B.2.2 para este programa, foram realizadas as simulações do modelo ortogonal, obtendo-se os resultados mostrados a seguir. A figura 5.10 apresenta os mesmos resultados mostrados nas figuras 5.8 e 5.9, porém usando dessa vez o programa Octave como programa simulador. As

interpretações utilizadas para o programa SimnonTM também são válidas para o Octave, ou seja, os tempos de simulação e valores em transitório e regime.

FIGURA 5.10 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.2.3. SIMULAÇÃO COM O PROGRAMA MATLAB®

As rotinas geradas e apresentadas no apêndice B.2.3, foram utilizadas para realizar as simulações do modelo ortogonal, obtendo os resultados gráficos apresentados a seguir. A figura 5.11, também apresenta os mesmos resultados mostrados nas figuras 5.8 e 5.9, porém utilizando os recursos do Matlab®.

FIGURA 5.11 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.2.4. SIMULAÇÃO COM O PROGRAMA SIMULINK / MATLAB®.

As rotinas *fab0p.mdl*, *fab0g.mdl*, *iab0p.mdl* e *iab0g.mdl*, geradas a partir dos diagramas de blocos apresentados nas figuras 4.5 a 4.8, foram realizadas as simulações do modelo ortogonal, obtendo os resultados gráficos apresentados a seguir. A figura 5.12, também apresenta os mesmos resultados mostrados nas figuras 5.8 e 5.9, porém utilizando os recursos do Simulink / Matlab®.

FIGURA 5.12 - Gráfico dos fluxos por fase [Wb/s] nos referenciais: a) estacionário; b) síncrono; e das correntes por fase [A/s] nos referenciais: c) estacionário; d) síncrono.

5.3. MODELO NA NOTAÇÃO VETORIAL

Seguindo o mesmo procedimento utilizado para as notações anteriores, a notação vetorial é dada pela equação (4.13) para a variável de estado fluxo e pela equação (4.15) para a variável de estado corrente, tanto para o referencial estacionário ($\mathbf{w}_k = 0$), quanto para o síncrono ($\mathbf{w}_k = \mathbf{w}_1$). Os resultados de simulação deste caso segundo os pacotes de programa citados são tais como indicados no tópico seguinte. Os resultados serão apresentados em forma de gráficos sendo mostrados na mesma figura com cores diferentes, onde a cor vermelha representará o eixo “**a**” e a cor verde representará o eixo “**b**”. Este procedimento será feito tanto para o estator, quanto para o rotor em todos os softwares, exceto para o programa Simulink / Matlab® que possui cores pré-definidas e serão apresentadas da seguinte forma: o eixo “**a**” na cor amarela e o eixo “**b**” na cor magenta. Os gráficos para a velocidade do motor, conjugado eletromagnético e comportamento transitório nos eixos “**a**” e “**b**”, são idênticos aos gráficos apresentados na notação ortogonal, com a diferença apenas pelo fato de não existir o eixo “**0**”, condição necessária para se modelar o motor utilizando a notação vetorial. Sendo assim, serão mostrados apenas os gráficos da composição das partes real e imaginária do fluxo e da corrente em cada referencial.

5.3.1. SIMULAÇÃO COM O PROGRAMA SIMNONTM

Utilizando as rotinas geradas e apresentadas no apêndice B.3.1, e repetindo-se o procedimento adotado para as notações anteriores para a simulação, obtiveram-se os resultados apresentados a seguir.

Na figura 5.13 são apresentados 4 (quatro) gráficos correspondentes à composição das partes real e imaginária do fluxo de estator e rotor nos referenciais estacionário (a); síncrono (b); e da corrente de estator e rotor nos referenciais estacionário (c) e síncrono (d). Nota-se que a variável de estado fluxo parte da origem no instante inicial, realiza uma trajetória espiral e atinge o valor de regime no instante t_{final} , valor esse correspondente ao tempo final de simulação, sendo que tanto para o referencial estacionário, quanto o referencial síncrono, após o transitório os fluxos de estator e rotor atingem um nível de fluxo com a diferença que no referencial estacio-

nário terá um comportamento circular e no referencial estacionário converge para um ponto. Já a corrente tem o mesmo desempenho, com a diferença apenas pelo fato que de parte de um valor nulo realiza a trajetória espiral e retorna a um valor de magnetização.

FIGURA 5.13 - Gráfico da composição das partes real e imaginária para o fluxo [Wb]: (a) estacionário; b) síncrono; e para as correntes [A] nos referenciais: c) estacionário; d) síncrono.

5.3.2. SIMULAÇÃO COM O PROGRAMA OCTAVE

Com as rotinas geradas e apresentadas no apêndice B.2.2 para este programa, foram realizadas as simulações do modelo vetorial, obtendo-se os resultados mostrados a seguir. A figura 5.14 apresenta os mesmos resultados mostrados na figura 5.13-a e b, ou seja, a composição das partes real e imaginária da corrente nos referenciais estacionário (a) e síncrono (b), porém usando dessa vez o programa Octave como programa simulador. As interpretações utilizadas para o programa SimnonTM também são válidas para o Octave, ou seja, os tempos de simulação e valores em transiente e regime. Deve ser ressaltado apenas como um dos critérios de avaliação de desempenho do programa é o número de pontos gerados em cada gráfico, onde o programa SimnonTM, mostrou um gráfico não tão definido por usar menos pontos e no Octave o gráfico já ficou mais circular, entretanto, o tempo de simulação foi muito maior.

FIGURA 5.14 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono.

A figura 5.15 apresenta os mesmos resultados mostrados na figura 5.13-c e d, ou seja, a composição das partes real e imaginária da corrente nos referenciais estacionário (a) e síncrono (b), as interpretações utilizadas para o programa SimnonTM também são válidas para o Octave, ou seja, os tempos de simulação e valores em transitório e regime.

FIGURA 5.15 - Gráfico da composição das partes real e imaginária da corrente [A] nos referenciais: a) estacionário; b) síncrono.

5.3.3. SIMULAÇÃO COM O PROGRAMA MATLAB

As rotinas geradas e apresentadas no apêndice B.3.3, foram utilizadas para realizar as simulações do modelo vetorial, obtendo os resultados gráficos apresentados a seguir. A figura 5.16, também apresenta os mesmos resultados mostrados nas figuras 5.14 e 5.15, porém utilizando os recursos do Simulink / Matlab®. Nota-se que assim como no Octave a figura está mais definida devido a um número de maior de pontos, entretanto no Matlab® a simulação foi mais rápida conforme será mostrado mais adiante.

FIGURA 5.16 - Gráfico da composição do fluxo [Wb] para os referenciais: a) estacionário; b) síncrono e da composição das correntes [A] nos referenciais: c) estacionário e d) síncrono, nos eixos real x imaginário.

5.3.4. SIMULAÇÃO COM O PROGRAMA SIMULINK / MATLAB®

Seguindo o mesmo procedimento feito para as notações trifásica e ortogonal, será simulada a notação vetorial utilizando o programa Simulink/ Matlab®, com as variáveis de estado o fluxo e a corrente, nos referenciais estacionário e síncrono. Utilizando as rotinas geradas a partir dos diagramas de blocos apresentados nas figuras 4.9 e 4.10 para o fluxo, e figuras 4.11 e 4.12 para a corrente nos referenciais estacionário e síncrono. Obtendo os resultados gráficos apresentados a seguir. A figura 5.17, também apresenta os mesmos resultados mostrados nas figuras 5.16-a e b, porém utilizando os recursos do Simulink / Matlab®.

FIGURA 5.17 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono.

A figura 5.18 apresenta os mesmos resultados mostrados nas figuras 5.16-c e d, ou seja, a composição das partes real e imaginária da corrente nos referenciais estacionário (a) e síncrono (b), as interpretações utilizadas para os programas já mostrados também são válidas para o Simulink, ou seja, os tempos de simulação e valores em transitório e regime.

FIGURA 5.18 - Gráfico da composição das partes real e imaginária do fluxo [Wb] nos referenciais: a) estacionário; b) síncrono.

5.4. MODELO NA NOTAÇÃO VETORIAL COMPLEXA

Novamente, seguindo o mesmo procedimento utilizado para as notações anteriores, será simulado o modelo complexo com variável de estado fluxo dado pela equação (4.5) e pela equação (4.7) para a variável de estado corrente tanto para o referencial estacionário ($w_k = 0$), quanto para o síncrono ($w_k = w_1$), para tal simulação serão usados os softwares Matlab® e Simulink / Matlab®, uma vez que os demais softwares utilizados não conseguem manipular entidades complexas, conforme descrito no capítulo anterior. Os resultados de simulação deste caso segundo os pacotes de programas citados são tais como indicados no tópico seguinte. Os resultados serão apresentados em forma de gráficos sendo mostrados na mesma figura com cores diferentes, onde a cor vermelha representará o eixo “Real” e a cor verde representará o eixo “Imaginário”. Este procedimento será feito tanto para o estator, quanto para o rotor, porém para o programa Simulink / Matlab® que possui cores pré-

definidas serão apresentadas da seguinte forma: o eixo “*Real*” na cor amarela, o eixo “*Imaginário*” na cor magenta.

5.4.1. SIMULAÇÃO COM O PROGRAMA MATLAB

Utilizou-se a rotina *fcomp.m* gerada e apresentada no apêndice B.4.1, para a simulação. Na figura 5.19 é apresentado o comportamento transitório da velocidade e o conjugado eletromagnético, para tal procedimento utilizou-se a variável de saída $y(:,3)$ e o valor do md obtido com a equação (4.14), mostrados na rotina *mfcomp.m*.

FIGURA 5.19 - Gráfico da velocidade x tempo [rad/s] e o conjugado eletromagnético x tempo [Nm/s].

Nas figuras 5.20-a e b, apresentam-se 4 (quatro) gráficos, referentes ao comportamento complexo da variável de estado fluxo tanto de estator, quanto de rotor nos referenciais (a) estacionário; (b) síncrono. Nota-se que a variável de estado fluxo parte da origem no instante inicial, realiza uma trajetória espiral e atinge o valor de regime no instante t_{final} , correspondente ao tempo final de simulação, sendo que tanto para o referencial estacionário, quanto o referencial síncrono, após o transitório os fluxos de estator e rotor atingem um nível correspondente ao fluxo nestas condições, com a diferença que no referencial estacionário terá um comportamento circular e no referencial estacionário converge para um ponto.

FIGURA 5.20 - Gráfico do fluxo complexo [Wb] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

Já as figuras 5.21-a e b, ilustram o comportamento transitório das partes real e imaginária do fluxo no estator e no rotor, para os referenciais estacionário (a) e síncrono (b). No referencial estacionário impôs-se ao motor uma alimentação senoidal vetorial equilibrada e admitiu-se o eixo “real” como referência partindo de zero. Nota-se um comportamento bastante oscilatório durante o transitório na aceleração, até próximo de 0,4 s e que o fluxo parte do valor nulo até alcançar o valor de regime. Já no referencial síncrono, impôs-se ao motor uma alimentação do tipo tensão constante com amplitude máxima para o eixo “real” e a nulo para o eixo “imaginário”. Esta condição corresponde ao instante em que a senóide do eixo “real” atinge seu valor máximo e valor nulo para o eixo “imaginário”. Nota-se também, que durante a parte transitória, também próximo de 0,4 s, são evidenciadas as amplitudes oscilantes ou variáveis. Em ambos os casos, após o regime os fluxos de estator e rotor atingem um nível correspondente ao fluxo nestas condições, com a diferença que no referencial estacionário será um valor senoidal e no referencial síncrono um valor constante.

FIGURA 5.21 - Gráfico do comportamento transitório do fluxo [Wb] nos eixos real e imaginário x tempo para os referenciais: a) estacionário; b) síncrono.

Nas figuras 5.22-a e b, apresentam-se 4 (quatro) gráficos, referente ao comportamento complexo da variável de estado corrente tanto de estator, quanto de rotor nos referenciais (a) estacionário; (b) síncrono. Nota-se que a variável de estado corrente parte de origem no instante inicial, realiza uma trajetória espiral e atinge o valor de regime no instante t_{final} , sendo no referencial estacionário terá um comportamento circular e retornando a um valor de magnetização e no referencial estacionário converge para um ponto próximo da origem.

FIGURA 5.22 - Gráfico da corrente complexa [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

Já as figuras 5.23-a e b, ilustram o comportamento transitório das partes real e imaginária da corrente no estator e no rotor, para os referenciais estacionário (a) e síncrono (b). No referencial estacionário impôs-se ao motor uma alimentação senoidal vetorial equilibrada e admitiu-se o eixo “*real*” como referência partindo de zero. Nota-se um comportamento bastante oscilatório durante o transitório na aceleração, até próximo de 0,4 s e que a corrente parte do valor máximo até alcançar o valor de regime. Já no referencial síncrono, impôs-se ao motor uma alimentação do tipo tensão constante com amplitude máxima para o eixo “*real*” e a nulo para o eixo “*imaginário*”. Esta condição corresponde ao instante em que a senóide do eixo “*real*” atinge seu valor máximo e valor nulo para o eixo “*imaginário*”. Nota-se também, que durante a parte transitória, também próximo de 0,4 s, são evidenciadas as amplitudes oscilantes ou variáveis. Em ambos os casos, após o regime as correntes de estator e rotor atingem um nível correspondente ao fluxo nestas condições, com a diferença

que no referencial estacionário será um valor senoidal e no referencial síncrono um valor constante.

FIGURA 5.23 - Gráfico do comportamento transitório da corrente [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

5.4.2. SIMULAÇÃO COM O PROGRAMA SIMULINK / MATLAB®

Utilizaram-se as rotinas *fcomp.mdl*, *fcomg.mdl*, *icomg.mdl* e *icomg.mdl*, geradas a partir dos diagramas de blocos mostrados nas figuras 4.13 a 4.16, para a simulação. Conforme descrito no capítulo anterior o ambiente Simulink / Matlab®, a versão 5.2.1 não manipula entidades complexas. Sendo por isso necessário o uso das já citadas “S-Functions”. No caso do modelo de fluxo foram criadas as rotinas *flux1p.m* e *flux2p.m* para o referencial estacionário e *flux1g.m* e *flux2g.m* para o referencial síncrono, no estator e no rotor, essas rotinas foram apresentadas no apêndice B.4.2.

Na figura 5.24 apresenta-se o comportamento complexo do fluxo nos referenciais estacionário (a) e síncrono (b), nota-se que os resultados são os mesmos mostrados na figura 5.20, porém utilizando os recursos disponíveis no Simulink. Nota-se

que a variável de estado fluxo parte da origem no instante inicial, realiza uma trajetória espiral e atinge o valor de regime no instante t_{final} , sendo que tanto para o referencial estacionário, quanto o referencial síncrono, após o transitório os fluxos de estator e rotor atingem um nível correspondente ao fluxo nestas condições, com a diferença que no referencial estacionário terá um comportamento circular e no referencial estacionário converge para um ponto. Nota-se também, que a resolução dos gráficos apresentados, é inferior ao Matlab®.

FIGURA 5.24 - Gráfico do fluxo complexo [Wb] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

Na figura 5.25, apresenta-se o comportamento transitório do fluxo de estator nos eixos real e imaginário para os referenciais estacionário (a) e síncrono (b), os gráficos apresentados são idênticos aos gráficos da figura 5.21 do Matlab®, sendo assim, todas as considerações feitas para esta figura, com relação a tempo de estabilização, comportamento transitório e em regime, resposta para a tensão de entrada aplicada e referencial adotado, também são válidas para o Simulink.

FIGURA 5.25 - Gráfico do comportamento transitório do fluxo [Wb] nos eixos real e imaginário x tempo para os referenciais: a) estacionário; b) síncrono.

Na figura 5.26, apresenta-se o comportamento complexo da corrente de estator nos eixos real e imaginário para os referenciais estacionário (a) e síncrono (b), os gráficos apresentados são idênticos aos gráficos da figura 5.22 do Matlab®. Nota-se que a variável de estado corrente parte de origem no instante inicial, realiza uma trajetória espiral e atinge o valor de regime no instante t_{final} , valor esse correspondente ao tempo final de simulação, sendo no referencial estacionário terá um comportamento circular e retornando a um valor de magnetização e no referencial estacionário converge para um ponto próximo da origem. Nota-se também, que a definição da figura 5.26 é inferior a figura 5.22 apresentada pelo Matlab®.

FIGURA 5.26 - Gráfico da corrente complexa [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

Na figura 5.27, apresenta-se o comportamento transitório da corrente de estator nos eixos real e imaginário para os referenciais estacionário (a) e síncrono (b), os gráficos apresentados são idênticos aos gráficos da figura 5.21 do Matlab®. Sendo que, as mesmas considerações para a tensão de entrada aplicadas no Matlab®, são válidas aqui. Nota-se um comportamento bastante oscilatório durante o transitório na aceleração, até próximo de 0,4 s e que a corrente parte do valor máximo até alcançar o valor de regime. Já no referencial síncrono, impôs-se ao motor uma alimentação do tipo tensão constante com amplitude máxima para o eixo “*real*” e a nulo para o eixo “*imaginário*”. Esta condição corresponde ao instante em que a senóide do eixo “*real*” atinge seu valor máximo e valor nulo para o eixo “*imaginário*”. Nota-se também, que durante a parte transitória, também próximo de 0,4 s, são evidenciadas as amplitudes oscilantes ou variáveis. Em ambos os casos, após o regime as correntes de estator e rotor atingem um nível correspondente ao fluxo, com a diferença que no referencial estacionário será um valor senoidal e no referencial síncrono um valor constante.

FIGURA 5.27 - Gráfico do comportamento transitório da corrente [A] nos eixos real e imaginário para os referenciais: a) estacionário; b) síncrono.

5.5. AVALIAÇÃO GLOBAL DOS RESULTADOS

Para avaliar o desempenho de cada programa com relação ao tempo de simulação, foram feitas 20 (vinte) simulações do motor de indução trifásico sem carga, para os programas Octave, Matlab® e Simulink / Matlab®, e serão apresentados os tempos mínimo, máximo e médio de simulação. O SimnonTM, segundo informações do próprio fabricante (SSPA Systems) não tem como avaliar o tempo de simulação, mas visivelmente aparenta ser o mais rápido. Os gráficos a seguir, demonstram o comportamento de cada um dos programas citados, comparados individualmente e entre si.

Para as simulações foi utilizado dos programas SimnonTM, Matlab® e Simulink / Matlab® foi utilizado um microcomputador Pentium II 300 MHz, com 128 MB de memória RAM em ambiente Windows®, já para as simulações utilizando o programa Octave, utilizou-se um microcomputador Pentium II dual 450 MHz rodan-

do em sistema Linux, utilizando processamento paralelo, com 128 MB de memória, porém foi simulado via rede.

A figura 5.28 mostra o tempo de médio de simulação para o programa Octave, conforme descrito, esse programa não consegue resolver equações diferenciais complexas. Embora, conforme já descrito por tratar-se de um programa que conta com a colaboração de usuários, já consegue manipular termos complexos, sendo necessário uma adaptação de suas rotinas de integração numérica, para isso, fato esse que não o objetivo do trabalho.

FIGURA 5.28 - Tempo médio de simulação do motor de indução trifásico usando o programa Octave.

A figura 5.29, a seguir, mostra o tempo médio de simulação utilizando o programa Matlab®.

FIGURA 5.29 - Tempo médio de simulação do motor de indução trifásico usando o programa Matlab®.

Já a figura 5.30 mostra o desempenho da simulação do motor de indução para o programa Simulink / Matlab®.

FIGURA 5.30 - Tempo médio de simulação do motor de indução trifásico usando o programa Simulink / Matlab®.

Para efeito de comparação, foi criada a tabela 3, a seguir, contendo o tempo mínimo de simulação de cada programa, para as rotinas mostradas na tabela 2.

Tabela 3 – Tempo Mínimo de Simulação Para Cada Programa [s].

<i>Rotinas</i>	Octave	Matlab®	Simulink / Matlab®
FTRIP	15,63	2,74	0,49
FTRIG	5,57	1,38	0,27
FAB0P	13,46	2,47	0,28
FAB0G	4,85	1,26	0,16
FVETP	12,13	2,09	8,41
FVETG	4,32	1,09	4,33
FCOMP	*	1,92	16,42
FCOMG	*	0,87	7,58
ITRIP	25,82	6,26	0,49
ITRIG	8,16	1,81	0,27
IAB0P	21,31	5,60	0,33
IAB0G	6,63	1,31	0,16
IVETP	19,13	4,77	8,46
IVETG	6,02	1,15	4,34
ICOMP	*	3,62	16,31
ICOMG	*	0,87	7,58

* O programa Octave não resolve as rotinas complexas.

Baseado nos dados apresentados na Tabela 3, construiu-se o gráfico da figura 5.31, para uma melhor visualização e análise comparativa dos mesmos.

FIGURA 5.31 - Tempo mínimo de simulação do motor de indução trifásico para cada programa.

Através dos resultados apresentados, verifica-se que todos os programas no que diz respeito à resolução de equações diferenciais apresentam um bom desempenho, diferenciando em alguns detalhes, a serem descritos a seguir:

- O SimnonTM possui uma desvantagem muito considerável em relação ao pós-processamento das saídas, ou seja, sua parte gráfica não possibilita uma melhor apresentação dos mesmos, além de não manipular entidades complexas, parte fundamental deste trabalho. Como vantagem apresenta um tempo de simulação bastante curto, embora não tenha como ser avaliado.
- O Octave apresenta como desvantagens não manipular entidades complexas, à parte de interface com o usuário não é muito amigável (semelhante ao MS-DOS). O pós-processamento de resultados que também não possibilita uma comparação muito clara e como pode ser visto nas figuras 5.28 e 5.31 é o programa mais lento, para qualquer tipo de notação simulada, dentre os apresentados.
- Simulink / Matlab® é muito fácil de utilizar, por ser totalmente gráfico apresenta resultados de uma forma bem visível, entretanto, quando se faz necessário comentar os resultados em qualquer processador de texto, tor-

na-se inviável conforme comentado no decorrer deste Capítulo. As figuras não são manipuláveis. Em relação ao tempo de simulação, levando-se em consideração o modelo já pronto (desenhado), este programa tem um desempenho bom para as notações trifásica e ortogonal, conforme mostrado nas figuras 5.30 e 5.31, mas para as notações vetorial e vetorial complexa é inferior ao Matlab®. Porém, leva-se muito tempo para construir o modelo do motor de indução em qualquer uma das notações apresentadas.

- O Matlab® dentre os quatro programas utilizados, foi o que atendeu plenamente a todas as notações aqui apresentadas, tem um pós-processamento de resultados muito amigável e fácil de interpretar. Com relação ao tempo de simulação, este programa apresenta um desempenho satisfatório para as notações trifásica e ortogonal e é o mais indicado para as notações vetorial e vetorial complexa, conforme mostrado nas figuras 5.29 e 5.31.

É notável que dentre as quatro notações apresentadas para o modelar o motor de indução trifásico, a mais indicada é a notação vetorial complexa. Além de não necessitar de manipulações algébricas para a sua construção, fato este bastante considerável em relação às outras notações, basta usar a equação de estado da máquina. É a que apresenta o menor número de equações diferenciais e tem um tempo de simulação muito rápido, conforme mostrado na figura 5.31. Ainda, para o motor de indução trifásico esta notação é a mais indicada, pois demonstra o comportamento em tempo real da máquina tanto no transitório, quanto em regime permanente.

Capítulo 6

CONCLUSÕES

Conforme visto nos capítulos anteriores, mostrou-se todos os tipos de modelagens e apresentaram-se os resultados para a variável de estado fluxo e corrente nos referenciais estacionário e síncrono. Discutiu-se com relação aos programas utilizados e podem-se destacar alguns pontos importantes.

No uso do SimnonTM, conforme já comentado, o tempo de simulação é rápido (embora não foi utilizado nenhum procedimento de cronometragem específico). Entretanto, no que diz respeito à definição e qualidade de resultados, este programa apresenta um nível bem inferior. As janelas gráficas de saída também possuem deficiência, por exemplo, caso o interesse seja mostrar mais de um gráfico por janela, os gráficos ficam muito juntos e os nomes dos eixos não ficam em uma posição muito boa. O SimnonTM não manipula entidades complexas, o que torna o software inadequado para utilização no modelo vetorial complexo.

A grande vantagem do programa Octave é o fato de ser um programa “free” e disponível na Internet. Sua parte gráfica também necessita de uma melhora, caso o interesse seja plotar vários gráficos em uma mesma janela, os gráficos não permitem uma análise precisa. Como o Octave é utilizado por diversas pessoas, ele é atualizado a medida que um usuário sente a necessidade de alguma ferramenta, e já se consegue manipular algumas entidades complexas, mas ainda faz-se necessário desenvolver uma rotina de integração complexa, o que não é objeto deste trabalho. Outra desvantagem é o tempo de simulação dos modelos, os quais são superiores ao tempo de simulação do SimnonTM e do Matlab®.

O Matlab® conforme comentado no trabalho, demonstrou ser o mais indicado *software*, dentre os utilizados, mesmo tendo um custo considerado. O Matlab® possui um tempo de simulação razoavelmente rápido, conforme dados levantados e apresentado no capítulo anterior, mostrou-se bem mais rápido que o Octave. Outra vantagem é o fato de manipular grandezas complexas, além de apresentar uma parte de pós-processamento de resultados muito boa, tais que os gráficos possam ser orga-

nizados de uma forma clara. Possui um ambiente gráfico que possibilita a criação de programas totalmente interagidos com o usuário.

Com relação ao Simulink / Matlab®, ele tem como principal vantagem ser totalmente gráfico e com isso o diagrama de blocos torna-se bem visível e de fácil interpretação, tem um tempo de simulação curto, mesmo comparado com o Matlab®. Suas desvantagens são não manipular grandezas complexas, sendo necessário o uso das já citadas “*S-Functions*” e, também, sua parte de pós-processamento de resultados, assim como o SimnonTM e o Octave, não é muito amigável. Por exemplo, visualizar vários gráficos em uma mesma janela. Entretanto, o Simulink disponibiliza as variáveis desejadas no ambiente Matlab, tornando-as assim, mais fácil de manipular.

Com relação às formas de modelagem para o modelo dinâmico completo, pode-se dizer que o modelo trifásico tem como desvantagem o número de equações diferenciais, já que é um modelo de sétima ordem (possui 7 equações diferenciais), porém, possibilita por exemplo simular uma falta de fase, além de mostra o comportamento real por fase da variável de estado desejada.

Já o modelo ortogonal, mesmo tendo o mesmo número de equações diferenciais que o modelo trifásico, possui um maior número de zeros na matriz o que torna sua resolução um pouco mais fácil, mas necessita de transformações para conseguir o valor por fase de cada entidade elétrica.

O modelo vetorial é o modelo ortogonal, desprezando-se o eixo “0” tornando-se assim, um modelo de quinta ordem, ou seja, 5 (cinco) equações diferenciais. Ele possibilita a análise do comportamento transitório da máquina, uma vez que é todo baseado em vetor de espaço, mostrando cada entidade elétrica em seu módulo e fase, ou seja, mostrando a característica no tempo e espaço. Fato esse, inexistente, nas modelagens anteriores. Como desvantagem, cita-se o fato da necessidade da manipulação algébrica, ou seja, separar a equação complexa em termos de suas partes real e imaginária.

O modelo vetorial complexo mostra-se como uma interessante forma de modelar o motor de indução, comparadas às demais modelagens. Primeiro, por ser vetorial, possibilita a análise tanto em regime permanente, quanto durante o regime transitório do motor de indução. Segundo, a precisão obtida nos resultados, pois se comparando todas as notações pode-se notar que os resultados obtidos para a notação

vetorial foram os mesmos que os obtidos para as demais notações, fazendo as devidas transposições de referencial. Terceiro, ao manipular grandezas complexas, facilita-se conforme mostrado, a construção do diagrama de blocos, que é uma ferramenta muito importante na análise dinâmica do motor de indução. E finalizando, o modelo vetorial complexo apresenta um tempo de simulação rápido se comparado aos demais, conforme já citado no capítulo anterior.

Como colaboração este trabalho apresentou algumas notações para modelar o motor de indução trifásico, utilizou alguns softwares existentes no mercado para obter os resultados da simulação. Foi dado um enfoque na notação vetorial complexa, as razões já foram citadas no decorrer do trabalho.

O intuito do trabalho foi mostrar algumas modelagens existentes, preocupando-se apenas com a modelagem do motor de indução trifásico, e considerando-se o conversor como um bloco fechado, sugere-se como continuação do trabalho estudar a análise vetorial para as técnicas de controle, ou seja, como seria e o que representa um lugar das raízes complexo, qual o diagrama de bode complexo. E estudar também qual seria o comportamento se o conversor utilizado fosse complexo e qual a possibilidade de sua implementação, bem como a possibilidade de implementar um controlador complexo.

Referências Bibliográficas

- Alger, Philip L., (1975). *The Nature of Polyphase Induction Machines*. John Wiley & Sons, Inc., New York Chapman & Hall, Ltd., London.
- Dalton, P. M.; Gosbell, V.J., (1989). A Study of Induction Motor Current Control Using the Complex Number Representation, *Proceeding Conference Rec. of IEEE Industry Application Society Meeting*, Part I, pp. 354-361.
- de Aguiar, Manoel L.; Cad, Marcelo M., (1999a). Induction Motor Modelling and Simulation Using Complex Dynamic Systems, *IASTED – International Conference on Modelling and Simulation*, May, 05-08, Philadelphia PA, USA, pp. 15-18.
- de Aguiar, Manoel L.; Cad, Marcelo M., (1999b). Solving Complex Dynamic Systems with Matlab in Electrical Engineering Problems, *International Conference on Promotion and Enhancement of Computational Methods in Engineering and Science*, VII – EPMESC, Aug, 02-05, Macao - CD-ROM.
- de Aguiar, Manoel L.; Cad, Marcelo M., (1999c). Resolvendo Sistemas Dinâmicos Complexos em Problemas de Engenharia Elétrica com o Programa Matlab, *XX Congresso Ibero Latino-Americano de Métodos Computacionais para Engenharia - CILAMCE*, 3-5/nov, São Paulo/SP – CD-ROM.
- de Aguiar, M. L.; Cad, M. M., (2000a). The Concept of Complex Transfer Functions Applied to the Modeling of Induction Motors, *Winter Meeting 2000 of the IEEE Power Engineering Society*, January 23-27, Singapore – CD-ROM.
- de Aguiar, M. L.; Cad, M. M., (2000b). Modeling and Simulation of Induction Motors with Complex Transfer Functions, *International Power Electronics Conference – IPEC – Tokyo 2000*, April 3-7, Tokyo, Japan, pp. 1902-1906.
- de Andrade, Darizon Alves (1994). *Dynamic Control of Inverter-fed Cage Induction Motors*. Leeds. 212p. Tese (Doutorado) – Eletronic & Electrical Engineering Department, University of Leeds.
- De Doncker, R. W.; Novotny, D.W. (1998). The Universal Field Oriented Controller, *Conf. Rec. IEEE/IAS Ann. Mtg.*, pp. 450-456.
- Holtz, Joachim (1995). The Representation of AC Machine Dynamics by Complex Signal Flow Graphs, *IEEE Transactions On Industrial Electronics*, vol. 42, nº 3, pp 263-271.

- Kovács, K. P.; Rácz, P., (1959). *Transient Vorgänge in Wechselstrommaschinen*, Budapest, Verlag der Ungar. Akad. D. Wissensch.
- Krause, Paul C.; (1995). *Analysis of Electrical Machinery*. Piscataway: IEEE Press.
- Leonhard, W., (1985). *Control of Electrical Drives*. Springer-Verlag.
- Novotny, D. W.; Wouterse, J. H., (1976). Induction Machine Transfer Function and Dynamic Response by Means of Complex Time Variables, *IEEE Transactions on Power Apparatus Systems*, vol. PAS 95, n° 4, pp. 1325-1335.
- Matlab® with Simulink™ User's Guide*. Mathworks, Inc., 1993.
- University of Wisconsin-Madison, Department of Chemical Engineering. (1999). *Octave v. 2.0.16: user manual*. http://www.che.wisc.edu/octave/doc/octave_toc.html (20 dec.).
- S. Gataric; N. R. Garrigan (1999). Modeling and Design of Three-Phase Systems Using Complex Transfer Functions, *PESC 99*.
- S. Wade, M. W. Dunnigan, B. W. Williams (1994). Simulation of Induction Machine Vector Control and Parameter Identification, *IEE – Power Electronics and Variable-Speed Drives*, 26-28 October, Conference Publication n.° 399.
- Simmon™* (1993). *User's Guide for MS-DOS Computers*. SSPA Systems.
- Szablya, J. F.; Bressane, J. M. (1973). Transfer Function of AC Machines, *IEEE Transactions on Power Apparatus Systems*, vol. PAS 92, n.° 1, Jan/Feb, pp. 177-186.
- The MATLAB Compiler user's guide, in *MathWorks Handbook*, MathWorks, 1994.
- Using Simulink, in *MathWorks Handbook*, MathWorks, 1996.
- W. Leonhard (1985), *Control of Electrical Drivers*, Springer Verlag. Cap. 10: Symmetrical Three-phase AC Induction Machine.
- Vas, P. (1992). *Electrical Machines and Drives – A Space-Vector Theory Approach*. Oxford University Press.
- Vas, P. (1994). *Vector Control of AC Machines*. Oxford University Press.
- Yamamura, S. (1992). *Spiral Vector Theory of AC Circuits and Machines*. Oxford University Press.

Anexo A

SISTEMAS DINÂMICOS COMPLEXOS

Uma maneira bastante coerente de se avaliar o comportamento dinâmico do motor de indução trifásico, é através da análise de sistemas dinâmicos com coeficientes complexos. Baseado no modelo vetorial e com uma simples manipulação das equações vetoriais do modelo do motor de indução trifásico, compõe-se à equação de estado complexa, evitando assim, a manipulação algébrica de separar os termos em partes real e imaginária das equações diferenciais.

Considerando-se, por exemplo, um sistema de primeira ordem representado pela seguinte equação diferencial:

$$\dot{x}(t) + a x(t) = u(t) \quad (\text{A.1-a})$$

com

$$a = d + jw \quad (\text{A.1-b})$$

O processo representado por (A.1), produzirá uma saída do tipo complexa, independentemente se a entrada $u(t)$ tenha um valor puramente real. A figura A.1 transpõe a equação diferencial para o diagrama de blocos.

Figura A.1 - Representação de um sistema dinâmico a coeficientes complexos.

Como forma de encontrar a saída $x(t)$ em função dos parâmetros e da entrada $u(t)$, aplica-se à transformada de Laplace em (A.1) e chega-se à:

$$G(s) = \frac{X}{U}(s) = \frac{1}{s + a} \quad (\text{A.2})$$

Que demonstra um sistema dinâmico de primeira ordem, e admitindo uma entrada do tipo degrau e puramente real com amplitude u_0 , a solução para (A.1) será:

$$x(t) = L^{-1} \left\{ \frac{u_0}{s(s+a)} \right\} = \frac{u_0}{a} (1 - e^{-at}) = \frac{u_0}{d + jw} (1 - e^{-(d+jw)t}) \quad (A.3)$$

Tendo-se a variável ‘ t ’ como parâmetro, a saída $x(t)$ apresentará uma trajetória do tipo espiral partindo de $x(t)=0$ para $t=0$ e finalizando em $x(t)=u_0/a$ no plano complexo, para $t \rightarrow \infty$. Como mostra a figura A.2.

Figura A.2 - Comportamento transitório de $x(t)$ para uma excitação $u(t)$ real, (vide (A.1-a)).

Modelo Vetorial Separado em Real e Imaginário

Como solução para o modelo vetorial complexo, um artifício muito utilizado nos pacotes de programas existentes, é o de separar o sistema em complexo em termos de partes real e imaginária.

$$\begin{aligned} x(t) &= x_a(t) + jx_b(t) = \\ &= \left(\frac{u_0}{d^2 + w^2} \right) \left[(d - e^{-dt} (d \cos(wt) - w \sin(wt))) - \right. \\ &\quad \left. - j(w - e^{-dt} (d \sin(wt) + w \cos(wt))) \right] \end{aligned} \quad (A.4)$$

Através de (A.4) pode-se fazer algumas considerações:

→ Sendo $d=0$, o sistema torna-se puramente imaginário e as componentes a e b de $x(t)$ tem um comportamento temporal oscilante sem sub-amortecimento, com isso, a espiral do plano complexo degenera-se em um círculo;

- Sendo $\mathbf{w}=0$, o sistema é um sistema dinâmico puramente real de primeira ordem, e no plano complexo o comportamento transitório irá situar-se sobre o eixo real;
- E caso $\mathbf{d} \neq 0$ e $\mathbf{w} \neq 0$, o sistema dinâmico de primeira ordem apresenta um comportamento de segunda ordem real sub-amortecido tanto na parte real quanto na parte imaginária; e no plano complexo representará um espiral.

Partindo-se da equação (A.1) separada em parte real e imaginária e admitindo $u(t)$ real, tem-se:

$$\dot{x}_a(t) + j\dot{x}_b(t) = (\mathbf{d} + j\mathbf{w})(x_a(t) + jx_b(t)) = u_0 + j0 \quad (\text{A.5})$$

$$[\dot{x}_a(t) + \mathbf{d}x_a(t) - \mathbf{w}x_b(t)] + j[\dot{x}_b(t) + \mathbf{d}x_b(t) - \mathbf{w}x_a(t)] = u_0$$

ou, colecionando-se as partes real e imaginária, tem-se:

$$\begin{cases} \dot{x}_a(t) = u_0 - \mathbf{d}x_a(t) + \mathbf{w}x_b(t) \\ \dot{x}_b(t) = -\mathbf{d}x_b(t) - \mathbf{w}x_a(t) \end{cases} \quad (\text{A.6})$$

Colocando-se na forma de espaço de estado, para se proceder à solução numérica, chega-se a:

$$\begin{bmatrix} \dot{x}_a \\ \dot{x}_b \end{bmatrix} = \begin{bmatrix} -\mathbf{d} & \mathbf{w} \\ -\mathbf{w} & -\mathbf{d} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} u_0 \\ 0 \end{bmatrix} \quad (\text{A.7})$$

Aplicando-se a Transformada de Laplace em (A.6) obtêm-se:

$$\begin{aligned} X_a(s) &= \frac{1}{s + \mathbf{d}} u_0 + \frac{\mathbf{w}}{s + \mathbf{d}} X_b(s) \\ X_b(s) &= -\frac{\mathbf{w}}{s + \mathbf{d}} X_a(s) \end{aligned} \quad (\text{A.8})$$

Pelo desmembramento da equação diferencial complexa de primeira ordem (A.1), obtêm-se um sistema de duas equações diferenciais de primeira ordem acopladas. As duas equações compreendendo as partes reais e imaginárias são então solucionadas como um sistema de equações diferenciais reais. A correspondente representação em diagrama de blocos para o sistema de equações em (6) ou (7), é obtido como indicado na figura A.3.

Figura A.3 - Representação de (6) em diagrama de blocos.

Comparando-se a figura A.1, que representa a notação vetorial complexa, com a figura A.3, que representa o modelo vetorial separado em partes real e imaginária, pode-se observar a presença de um duplo acoplamento e também, a complexidade na construção do diagrama de blocos da figura A.3, enquanto que na figura A.1, é um diagrama de blocos mais compacto e fácil de construir.

A seguir, serão apresentadas as publicações resultantes do estudo deste trabalho.

Apêndice A

Neste tópico serão mostrados os artigos e o capítulo de livro, resultantes do estudo deste trabalho e que serviram como um incentivo maior para a realização do mesmo. Os artigos serão mostrados na sua ordem cronológica:

1. M. L. de Aguiar – M. M. Cad, “*Solving Complex Dynamic Systems with Matlab in Electrical Engineering Problems*”, International Conference on Promotion and Enhancement of Computational Methods in Engineering and Science, VII – EPMESC, Aug., 02-05 1999, MACAO – CD-ROM.
2. M. L. de Aguiar – M. M. Cad, “*Induction Motor Modelling and Simulation Using Complex Dynamic Systems*”, Proceedings of the IASTED International Conference Modeling and Simulation (MS’ 99), May 5-8, 1999, Philadelphia, Pennsylvania – USA – CD-ROM.
3. M. L. de Aguiar – M. M. Cad, “*Resolvendo Sistemas Dinâmicos Complexos em Problemas de Engenharia Elétrica com o Programa Matlab*”, XX Congresso Ibero Latino-Americano de Métodos Computacionais para Engenharia, CILAMCE, 3-5/nov/99, São Paulo/SP – CD-ROM.
4. M. L. de Aguiar – M. M. Cad, “*Resolvendo Sistemas Dinâmicos Complexos em Problemas de Engenharia Elétrica com o Programa Matlab*”, I Seminário Nacional de Controle e Automação, SNAC, 10-12/nov/99, Salvador/BA – CD-ROM.
5. M. L. de Aguiar – M. M. Cad, “*The Concept of Complex Transfer Functions Applied to the Modeling of Induction Motors*”, Winter Meeting 2000 of the IEEE Power Engineering Society, January 23-27, 2000, Singapore – CD-ROM.
6. M. L. de Aguiar – M. M. Cad, “*Modeling and Simulation of Induction Motors with Complex Transfer Functions*”, International Power Electronics Conference – IPEC-Tokyo 2000 April 3-7, 2000, Tokyo, Japan, pp. 1902-1906.

O trabalho “*Solving Complex Dynamic Systems with Matlab in Electrical Engineering Problems*”, faz parte do livro Computational Methods in Engineering & Science, publicado como capítulo do volume I, no tópico Mathematical Applications, páginas 167 a 175, editado por E. Arantes e Oliveira, J. Bento, E. Pereira. c/o Mrs Lurdes Farrusco, Instituto Superior Técnico, 1049-001 Lisboa, Portugal.

Apêndice B

São aqui apresentadas as rotinas utilizadas pelos programas conforme preparação descrita no capítulo 4. As rotinas serão apresentadas na sequência dos programas apresentada no capítulo 4, ou seja, SimnonTM, Octave, Matlab® e Simulink / Matlab®. E também de acordo com o modelo a ser usado, ou seja, trifásico, ortogonal, vetorial e complexo, e também, de acordo com a variável de estado a ser usada, ou seja, fluxo e corrente utilizando os referenciais fixo no estator ($w_k = 0$), e fixo no campo de estator ($w_k = w_1$).

B.1) Notação Trifásica

Conforme descrito no capítulo 4 apresentam-se aqui as rotinas construídas para simulação dos modelos utilizando como variável de estado o fluxo e referencial fixo no estator para o modelo trifásico.

B.1.1) Rotinas a Serem Usadas no Programa SimnonTM

Rotina ftrip.t: Modelo de fluxo trifásico com referencial fixo no estator.

CONTINUOUS SYSTEM FTRIP

" Version: 1.0
" Abstract: Modelo de Fluxo Trifásico
" Description: Referencial parado (wl=0)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

" Inputs and outputs:

" States, derivates and time:
STATE f1a f1b f1c f2a f2b f2c omgm
DER df1a df1b df1c df2a df2b df2c domgm
TIME t

" Initializations:

s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=wl/sqrt(3)
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2/sqrt(3)

```

m=-kd/J
u1a=311.1270*sin(w1*t)
u1b=311.1270*sin(w1*t-2.094)
u1c=311.1270*sin(w1*t+2.094)

" Equations:
w2=wl-abs(NP*omgm)
md=NP*lh/(sqrt(3)*s*I1*I2)*(f1a*(f2c-f2b)+f1b*(f2a-f2c)+f1c*(f2b-f2a))
df1a=a*f1a+b*f1b-b*f1c+c*f2a+u1a
df1b=b*f1a+a*f1b+b*f1c+c*f2b+u1b
df1c=b*f1a-b*f1b+a*f1c+c*f2c+u1c
df2a=d*f1a+e*f2a+f*f2b-f*f2c
df2b=d*f1b-f*f2a+e*f2b+f*f2c
df2c=d*f1c+f*f2a-f*f2b+e*f2c
domgm=m*omgm+md/J

" Parameter values:
wl:0
w1:376.99
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2
END

```

Rotina mftrip.t: Macro MFTRIP.T para executar o programa ftrip.t

```

MACRO MFTRIP
" Version: 1.0
" Abstract: Macro para iniciar ftrip
" Description: Simular e Plotar ftrip
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst ftrip
store f1a f1b f1c f2a f2b f2c omgm md
simu 0 0.5/f1
export f1 < f1 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp ftrip1
newplot
ashow f1a f1b f1c
text 'Fluxo de Estator nas Fases a b c'
hcopy bmp ftrip2

```

```

newplot
ashow f2a f2b f2c
text 'Fluxo de Rotor nas Fases a b c'
hcopy bmp ftrip3
END

```

Rotina ftrig.t: Modelo de fluxo trifásico com referencial síncrono.

CONTINUOUS SYSTEM FTRIG

```

" Version: 1.0
" Abstract: Modelo de Fluxo Trifásico
" Description: Referencial Girando (wl=w1)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

```

" Inputs and outputs:

```

" States, derivates and time:
STATE f1a f1b f1c f2a f2b f2c omgm
DER df1a df1b df1c df2a df2b df2c domgm
TIME t

```

" Initializations:

```

s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=wl/sqrt(3)
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2/sqrt(3)
m=-kd/J
u1a=U
u1b=-U/2
u1c=-U/2

```

" Equations:

```

w2=wl-abs(NP*omgm)
md=NP*lh/(sqrt(3)*s*l1*l2)*(f1a*(f2c-f2b)+f1b*(f2a-f2c)+f1c*(f2b-f2a))
df1a=a*f1a+b*f1b-b*f1c+c*f2a+u1a
df1b=b*f1a+a*f1b+b*f1c+c*f2b+u1b
df1c=b*f1a-b*f1b+a*f1c+c*f2c+u1c
df2a=d*f1a+e*f2a+f*f2b-f*f2c
df2b=d*f1b-f*f2a+e*f2b+f*f2c
df2c=d*f1c+f*f2a-f*f2b+e*f2c
domgm=m*omgm+md/J

```

" Parameter values:

```

wl:376.9911
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0

```

J:0.027

NP:2

END

Rotina mftrig.t: Macro MFTRIG.T para executar o programa ftrig.t

MACRO MFTRIG

" Version: 1.0

" Abstract: Macro para iniciar ftrig

" Description: Simular e plotar ftrig

" Revision: 1.0

" Author: Marcelo M. Cad

" Created: 06/11/99

" Enter commands here:

```
algor dopri45r
error 1e-6
syst ftrig
store f1a f1b f1c f2a f2b f2c omgm md
simu 0 0.5/f2
export f2 < f2 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp ftrig1
newplot
ashow f1a f1b f1c
text 'Fluxo de Estator nas Fases a b c'
hcopy bmp ftrig2
newplot
ashow f2a f2b f2c
text 'Fluxo de Rotor nas Fases a b c'
hcopy bmp ftrig3
END
```

Rotina itrip.t: Modelo de corrente trifásico com referencial fixo no estator.

CONTINUOUS SYSTEM ITRIP

" Version: 1.0

" Abstract: Modelo de Corrente Trifásico

" Description: Referencial Parado (wl=0)

" Revision: 1.0

" Author: Marcelo M. Cad

" Created: 05/11/99

" Inputs and outputs:

" States, derivates and time:

STATE i1a i1b i1c i2a i2b i2c omgm

DER di1a di1b di1c di2a di2b di2c domgm

TIME t

" Initializations:

s=1-lh^2/11^2


```

a=-r1/(s*I1)
b=(w1-((1-s)*w2))/(s*sqrt(3))
c=(r2*Ih)/(s*I1*I2)
d=(Ih*(w1-w2))/(I1*s*sqrt(3))
e=(r1*Ih)/(s*I1*I2)
f=(Ih*(w2-w1))/(s*sqrt(3)*I2)
g=-r2/(s*I2)
h=(w2-(w1*(1-s)))/(s*sqrt(3))
m=-kd/J
v1=1/(s*I1)
v2=-Ih/(s*I1*I2)

u1a=311.1270*sin(w1*t)
u1b=311.1270*sin(w1*t-2.094)
u1c=311.1270*sin(w1*t+2.094)

" Equations:
w2=w1-abs(NP*omgm)
md=NP*Ih/sqrt(3)*(i2a*(i1c-i1b)+i2b*(i1a-i1c)+i2c*(i1b-i1a))
di1a=a*i1a+b*i1b-b*i1c+c*i2a+d*i2b-d*i2c+v1*u1a
di1b=-b*i1a+a*i1b+b*i1c-d*i2a+c*i2b+d*i2c+v1*u1b
di1c=b*i1a-b*i1b+a*i1c+d*i2a-d*i2b+c*i2c+v1*u1c
di2a=e*i1a+f*i1b-f*i1c+g*i2a+h*i2b-h*i2c+v2*u1a
di2b=-f*i1a+e*i1b+f*i1c-h*i2a+g*i2b+h*i2c+v2*u1b
di2c=f*i1a-f*i1b+e*i1c+h*i2a-h*i2b+g*i2c+v2*u1c
domgm=m*omgm+md/J

" Parameter values:
w1:0
w1:376.99
U:311.13
r1:7.56
r2:3.84
I1:0.35085
I2:0.35085
Ih:0.33615
kd:0
J:0.027
NP:2

END

```

Rotina mitrip.t: Macro MITRIP.T para executar o programa itrip.t

```

MACRO MITRIP
" Version: 1.0
" Abstract: Macro para iniciar itrip
" Description: Simular e plotar itrip
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst itrip
store i1a i1b i1c i2a i2b i2c omgm md

```

```

simu 0 0.5/f7
export f7 < f7 / 0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp itrip1
newplot
ashow i1a i1b i1c
text 'Corrente de Estator nas Fases a b c'
hcopy bmp itrip2
newplot
ashow i2a i2b i2c
text 'Corrente de Rotor nas Fases a b c'
hcopy bmp itrip3
END

```

Rotina itrig.t: Modelo de corrente trifásico com referencial síncrono.

CONTINUOUS SYSTEM ITRIG

```

" Version: 1.0
" Abstract: Modelo de Corrente Trifásico
" Description: Referencial Girando (wl=w1)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

```

" Inputs and outputs:

```

" States, derivates and time:
STATE i1a i1b i1c i2a i2b i2c omgm
DER di1a di1b di1c di2a di2b di2c domgm
TIME t

```

" Initializations:

```

s=1-lh^2/l1^2
a=-r1/(s*l1)
b=(wl-((1-s)*w2))/(s*sqrt(3))
c=(r2*lh)/(s*l1*l2)
d=(lh*(wl-w2))/(l1*s*sqrt(3))
e=(r1*lh)/(s*l1*l2)
f=(lh*(w2-w1))/(s*sqrt(3)*l2)
g=-r2/(s*l2)
h=(w2-(wl*(1-s)))/(s*sqrt(3))
m=-kd/J
v1=1/(s*l1)
v2=-lh/(s*l1*l2)

```

```

u1a=U
u1b=-U/2
u1c=-U/2

```

" Equations:

```

w2=wl-abs(NP*omgm)
md=NP*lh/sqrt(3)*(i2a*(i1c-i1b)+i2b*(i1a-i1c)+i2c*(i1b-i1a))
di1a=a*i1a+b*i1b-b*i1c+c*i2a+d*i2b-d*i2c+v1*u1a
di1b=-b*i1a+a*i1b+b*i1c-d*i2a+c*i2b+d*i2c+v1*u1b

```

```

di1c=b*i1a-b*i1b+a*i1c+d*i2a-d*i2b+c*i2c+v1*u1c
di2a=e*i1a+f*i1b-f*i1c+g*i2a+h*i2b-h*i2c+v2*u1a
di2b=-f*i1a+e*i1b+f*i1c-h*i2a+g*i2b+h*i2c+v2*u1b
di2c=f*i1a-f*i1b+e*i1c+h*i2a-h*i2b+g*i2c+v2*u1c
domgm=m*omgm+md/J

```

" Parameter values:

wl:376.9911

w1:376.9911

U:311.13

r1:7.56

r2:3.84

l1:0.35085

l2:0.35085

lh:0.33615

kd:0

J:0.027

NP:2

END

Rotina mitrig.t: Macro MITRIG.T para executar o programa itrig.t

MACRO MITRIG

" Version: 1.0

" Abstract: Macro para iniciar itrig

" Description: Simular e plotar itrig

" Revision: 1.0

" Author: Marcelo M. Cad

" Created: 06/11/99

" Enter commands here:

```

algor dopri45r
error 1e-6
syst itrig
store i1a i1b i1c i2a i2b i2c omgm md
simu 0 0.5/f8
export f8 < f8 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp itrig1
newplot
ashow i1a i1b i1c
text 'Corrente de Estator nas Fases a b c'
hcopy bmp itrig2
newplot
ashow i2a i2b i2c
text 'Corrente de Rotor nas Fases a b c'
hcopy bmp itrig3
END

```

B.1.2) Rotinas a Serem Usadas no Programa Octave

Rotina ftrip.m: Modelo de fluxo trifásico com referencial fixo no estator

```
%ftrip.m - rotina para modelo de fluxo trifasico com referencial parado
function yp = ftrip (y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
kd=0; M=0.027;NP=2;w1=0;
u1a=311.1270*sin(w1*t);u1b=311.1270*sin(w1*t-2*pi/3);
u1c=311.1270*sin(w1*t+2*pi/3);

w2=w1-abs(NP*y(7));
md=NP*lh/(sqrt(3)*s*l1*l2)*(y(1)*(y(6)-y(5))+y(2)*(y(4)-y(6))+y(3)*(y(5)-y(4)));

a=-r1/(s*l1);
b=w1/sqrt(3);
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
E=-r2/(s*l2);
f=w2/sqrt(3);
m=-kd/M;

yp=[a b -b c 0 0 0; ...
    -b a b 0 c 0 0; ...
    b -b a 0 0 c 0; ...
    d 0 0 E f -f 0; ...
    0 d 0 -f E f 0; ...
    0 0 d f -f E 0; ...
    0 0 0 0 0 m]*y + ...
    [u1a;u1b;u1c;0;0;0;md/M];

clear a b c d E f m
```

Rotina mftrip.m: Macro MFTRIP.M para executar o programa ftrip.m

```
%mftrip.m - rotina para execução do programa ftrip.m
%parâmetros para calcular o md
l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;NP=2;
%definição do tempo de simulação
t=linspace(0,0.5,1500);
%definição do vetor de condições iniciais
x0=[0;0;0;0;0;0;0];
%definição do tempo para cálculo do tempo de simulação
t0=clock;
%rotina para simulação
y=lsode("ftrip",x0,t);
%construção dos gráficos
w=figure(1);
%calculo do md
md=3*NP*lh/(2*s*l1*l2)*(y(:,1).*y(:,5)-y(:,2).*y(:,4));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
```

```

title('Fluxo de Estator nas Fases A B C');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;
subplot(3,1,1);plot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Fluxo de Rotor nas Fases A B C');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;
subplot(3,1,2);plot(t,y(:,4),t,y(:,5),t,y(:,6));
% cálculo final do tempo de simulacao
tso_ftrip = etime(clock(),t0)
clear w z z1

```

Rotina ftrig.m: Modelo de fluxo trifásico com referencial síncrono

```

% ftrig.m - rotina para modelo de fluxo trifasico com referencial girando
function yp = ftrig (y,t)
% parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
kd=0; M=0.027;NP=2;w1=376.9911;u1a=U;u1b=-U/2;u1c=u1b;

w2=w1-abs(NP*y(7));
md=NP*lh/(sqrt(3)*s*l1*l2)*(y(1)*(y(6)-y(5))+y(2)*(y(4)-y(6))+y(3)*(y(5)-y(4)));

a=-r1/(s*l1);
b=w1/sqrt(3);
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
E=-r2/(s*l2);
f=w2/sqrt(3);
m=-kd/M;

yp=[a b -b c 0 0 0; ...
    -b a b 0 c 0 0; ...
    b -b a 0 0 c 0; ...
    d 0 0 E f -f 0; ...
    0 d 0 -f E f 0; ...
    0 0 d f -f E 0; ...
    0 0 0 0 0 0 m]*y + ...
    [u1a;u1b;u1c;0;0;0;md/M];

clear a b c d E f m

```

Rotina mftrig.m: Macro MFTRIG.M para executar o programa ftrig.m

```

% mftrig.m - rotina para execução do programa ftrig.m
% parâmetros para calcular o md
l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;NP=2;
% definição do tempo de simulação
t=linspace(0,0.5,1500);
% definição do vetor de condições iniciais
x0=[0;0;0;0;0;0;0];
% definição do tempo para cálculo do tempo de simulação
t0=clock;
% rotina para simulação
y=lsode("ftrig",x0,t);
% construção dos gráficos
w=figure(1);
% cálculo do md
md=-3*NP*lh/(2*s*l1*l2)*(y(:,1).*y(:,5)-y(:,2).*y(:,4));
subplot(1,2,1)

```

```

title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Fluxo de Estator nas Fases A B C');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;
mplot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Fluxo de Rotor nas Fases A B C');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;
mplot(t,y(:,4),t,y(:,5),t,y(:,6));
% cálculo final do tempo de simulacao
tso_ftrig = etime(clock(),t0)
clear w z z1

```

Rotina itrip.m: Modelo de corrente trifásico com referencial fixo no estator

```

%itrip.m - rotina para modelo de corrente trifasico com referencial parado
function yp = itrip(y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;kd=0;M=0.027;NP=2;
wl=0;u1a=311.1270*sin(w1*t+pi/2);u1b=311.1270*sin(w1*t-2*pi/3+pi/2);
u1c=311.1270*sin(w1*t+2*pi/3+pi/2);

w2=w1-abs(NP*y(7));
md=NP*lh/sqrt(3)*(y(4)*(y(3)-y(2))+y(5)*(y(1)-y(3))+y(6)*(y(2)-y(1)));

a=-r1/(s*l1);
b=(w1-((1-s)*w2))/(s*sqrt(3));
c=(r2*lh)/(s*l1*l2);
d=(lh*(w1-w2))/(l1*s*sqrt(3));
E=(r1*lh)/(s*l1*l2);
f=(lh*(w2-w1))/(s*sqrt(3)*l2);
g=-r2/(s*l2);
h=(w2-(w1*(1-s)))/(s*sqrt(3));
m=-kd/M;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b -b c d -d 0; ...
-b a b -d c d 0; ...
b -b a d -d c 0; ...
E f -f g h -h 0; ...
-f E f -h g h 0; ...
f -f E h -h g 0; ...
0 0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u1c;v2*u1a;v2*u1b;v2*u1c;md/M];

clear a b c d E f g h m v1 v2

```

Rotina mitrip.m: Macro MITRIP.M para executar o programa itrip.m

```

clear all
%mitrip.m - rotina para execução do programa itrip.m
%parâmetros para calcular o md
l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;NP=2;
%definição do tempo de simulação
t=linspace(0,0.5,1500);
%definição do vetor de condições iniciais

```

```

x0=[0;0;0;0;0;0;0];
%definição do tempo para cálculo do tempo de simulação
t0=clock;
%rotina para simulação
y=lsode("itrip",x0,t);
%construção das saídas
w=figure(1);
md=NP*lh/sqrt(3)*(y(:,4).*(y(:,3)-y(:,2))+y(:,5).*(y(:,1)-y(:,3))+y(:,6).*(y(:,2)-y(:,1)));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel.[rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Comportamento Transitório da Corrente de Estator nas Fases A B C');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;mplot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Comportamento Transitório da Corrente de Rotor nas Fases A B C');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;mplot(t,y(:,4),t,y(:,5),t,y(:,6));
%calculo final do tempo de simulação
tso_itrip = etime(clock(),t0)
clear w z z1

```

Rotina itrig.m: Modelo de corrente trifásico com referencial síncrono

```

%itrig.m - rotina para modelo de corrente trifasico com referencial girando
function yp = itrig(y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;kd=0;M=0.027;NP=2;
wl=376.9911;u1a=U;u1b=-U/2;u1c=u1b;

w2=wl-abs(NP*y(7));
md=NP*lh/sqrt(3)*(y(4)*(y(3)-y(2))+y(5)*(y(1)-y(3))+y(6)*(y(2)-y(1)));

a=-r1/(s*l1);
b=(w1-((1-s)*w2))/(s*sqrt(3));
c=(r2*lh)/(s*l1*l2);
d=(lh*(w1-w2))/(l1*s*sqrt(3));
E=(r1*lh)/(s*l1*l2);
f=(lh*(w2-w1))/(s*sqrt(3)*l2);
g=-r2/(s*l2);
h=(w2-(w1*(1-s)))/(s*sqrt(3));
m=-kd/M;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b -b c d -d 0; ...
-b a b -d c d 0; ...
b -b a d -d c 0; ...
E f -f g h -h 0; ...
-f E f -h g h 0; ...
f -f E h -h g 0; ...
0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u1c;v2*u1a;v2*u1b;v2*u1c;md/M];

clear a b c d E f g h m v1 v2

```

Rotina mitrig.m: Macro MITRIG.M para executar o programa itrig.m

```
clear all
%mitrig.m - rotina para execução do programa itrig.m
%parâmetros para calcular o md
l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820; NP=2;
%definição do tempo de simulação
t=linspace(0,0.5,1500);
%definição do vetor de condições iniciais
x0=[0;0;0;0;0;0;0];
%definição do tempo para cálculo do tempo de simulação
t0=clock;
%rotina para simulação
y=lsode("itrig",x0,t);
%construção das saídas
w=figure(1);
md=NP*lh/sqrt(3)*(y(:,4).*(y(:,3)-y(:,2))+y(:,5).*(y(:,1)-y(:,3))+y(:,6).*(y(:,2)-y(:,1)));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel.[rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Comportamento Transitório da Corrente de Estator nas Fases A B C');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;subplot(1,3,1);plot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Comportamento Transitório da Corrente de Rotor nas Fases A B C');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;subplot(1,3,1);plot(t,y(:,4),t,y(:,5),t,y(:,6));
%calculo final do tempo de simulação
tso_itrig = etime(clock(),t0)
clear w z z1
```

B.1.3) Rotinas a Serem Usadas no Programa Matlab

Rotina ftrip.m: Modelo de fluxo trifásico com referencial fixo no estator

```
%ftrip.m - rotina para modelo de fluxo trifasico com referencial fixo no estator
function yp = ftrip(t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;w1=0;
u1a=311.1270*sin(w1*t);u1b=311.1270*sin(w1*t-2*pi/3);
u1c=311.1270*sin(w1*t+2*pi/3);

w2=w1-abs(NP*y(7));
md=NP*lh/(sqrt(3)*s*l1*l2)*(y(1)*(y(6)-y(5))+y(2)*(y(4)-y(6))+y(3)*(y(5)-y(4)));

a=-r1/(s*l1);
b=w1/sqrt(3);
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
e=-r2/(s*l2);
f=w2/sqrt(3);
m=-kd/J;

yp=[a b -b c 0 0 0; ...
    -b a b 0 c 0 0; ...
```



```

b -b a 0 0 c 0; ...
d 0 0 e f -f 0; ...
0 d 0 -f e f 0; ...
0 0 d f -f e 0; ...
0 0 0 0 0 0 m]* y + ...
[u1a;u1b;u1c;0;0;0;md/J];

```

```
clear a b c d e f m
```

Rotina ftrig.m: Modelo de fluxo trifásico com referencial síncrono

```

%yftrig.m - rotina para modelo de fluxo trifasico com referencial síncrono
function yp = ftrig(t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027; NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;u1a=U;u1b=-U/2;u1c=u1b;

w2=wl-abs(NP*y(7));
md=NP*lh/(sqrt(3)*s*l1*l2)*(y(1)*(y(6)-y(5))+y(2)*(y(4)-y(6))+y(3)*(y(5)-y(4)));

a=-r1/(s*l1);
b=wl/sqrt(3);
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
e=-r2/(s*l2);
f=w2/sqrt(3);
m=-kd/J;

yp=[a b -b c 0 0 0; ...
-b a b 0 c 0 0; ...
b -b a 0 0 c 0; ...
d 0 0 e f -f 0; ...
0 d 0 -f e f 0; ...
0 0 d f -f e 0; ...
0 0 0 0 0 0 m]* y + ...
[u1a;u1b;u1c;0;0;0;md/J];

```

```
clear a b c d e f m
```

Rotina itrip.m: Modelo de corrente trifásico com referencial fixo no estator

```

%itrip.m - rotina para modelo de corrente trifasico
% com referencial fixo no estator
function yp = itrip(t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=0;u1a=311.1270*sin(w1*t+pi/2);u1b=311.1270*sin(w1*t-2*pi/3+pi/2);
u1c=311.1270*sin(w1*t+2*pi/3+pi/2);

w2=wl-abs(NP*y(7));
md=NP*lh/sqrt(3)*(y(4)*(y(3)-y(2))+y(5)*(y(1)-y(3))+y(6)*(y(2)-y(1)));

a=-r1/(s*l1);
b=(wl-((1-s)*w2))/(s*sqrt(3));
c=(r2*lh)/(s*l1*l2);

```

```

d=(lh*(w1-w2))/(l1*s*sqrt(3));
e=(r1*lh)/(s*l1*l2);
f=(lh*(w2-w1))/(s*sqrt(3)*l2);
g=-r2/(s*l2);
h=(w2-(w1*(1-s)))/(s*sqrt(3));
m=-kd/J;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b -b c d -d 0; ...
    -b a b -d c d 0; ...
    b -b a d -d c 0; ...
    e f -f g h -h 0; ...
    -f e f -h g h 0; ...
    f -f e h -h g 0; ...
    0 0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u1c;v2*u1a;v2*u1b;v2*u1c;md/J];

clear a b c d e f g h m v1 v2

```

Rotina itrigr.m: Modelo de corrente trifásico com referencial síncrono

```

%itrigr.m - rotina para modelo de corrente trifasico
% com referencial síncrono
function yp = itrigr(t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
w1=376.9911;u1a=U;u1b=-U/2;u1c=u1b;

w2=w1-abs(NP*y(7));
md=NP*lh/sqrt(3)*(y(4)*(y(3)-y(2))+y(5)*(y(1)-y(3))+y(6)*(y(2)-y(1)));

a=-r1/(s*l1);
b=(w1-((1-s)*w2))/(s*sqrt(3));
c=(r2*lh)/(s*l1*l2);
d=(lh*(w1-w2))/(l1*s*sqrt(3));
e=(r1*lh)/(s*l1*l2);
f=(lh*(w2-w1))/(s*sqrt(3)*l2);
g=-r2/(s*l2);
h=(w2-(w1*(1-s)))/(s*sqrt(3));
m=-kd/J;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b -b c d -d 0; ...
    -b a b -d c d 0; ...
    b -b a d -d c 0; ...
    e f -f g h -h 0; ...
    -f e f -h g h 0; ...
    f -f e h -h g 0; ...
    0 0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u1c;v2*u1a;v2*u1b;v2*u1c;md/J];

clear a b c d e f g h m v1 v2

```

B.2) Notação Ortogonal

Conforme descrito no capítulo 4 apresentam-se aqui as rotinas construídas para a simulação dos modelos utilizando como variável de estado o fluxo e referencial fixo no estator para o modelo trifásico.

B.2.1) Rotinas a Serem Usadas no Programa SimnonTM

Rotina fab0p.t: Modelo de fluxo ortogonal com referencial fixo no estator.

CONTINUOUS SYSTEM FAB0P

" Version: 1.0
" Abstract: Modelo de Fluxo Alfa Beta Zero
" Description: Referencial Parado (wl=0)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

" Inputs and outputs:

" States, derivates and time:
STATE f1a f1b f10 f2a f2b f20 omgm
DER df1a df1b df10 df2a df2b df20 domgm
TIME t

" Initializations:

s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=wl
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2
m=-kd/J

u1a=U*cos(w1*t)
u1b=U*sin(w1*t)
u10=0

" Equations:

w2=wl-abs(NP*omgm)
md=-3*NP*lh/(2*s*l1*l2)*(f1a*f2b-f1b*f2a)
df1a=a*f1a+b*f1b+c*f2a+u1a
df1b=b*f1a+a*f1b+c*f2b+u1b
df10=a*f10+c*f20+u10
df2a=d*f1a+e*f2a+f*f2b
df2b=d*f1b-f*f2a+e*f2b
df20=d*f10+e*f20
domgm=m*omgm+md/J

" Parameter values:

wl:0
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

Rotina mfab0p.t: Macro MFAB0P.T para executar o programa fab0p.t

MACRO MFAB0P

" Version: 1.0
" Abstract: Macro para iniciar fab0p
" Description: Simular e plotar fab0p
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst fab0p
store f1a f1b f10 f2a f2b f20 omgm md
simu 0 0.5/f3
export f3 < f3 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp fab0p1
newplot
ashow f1a f1b f10
text 'Fluxo de Estator nos Eixos Alfa Beta Zero'
hcopy bmp fab0p2
newplot
ashow f2a f2b f20
text 'Fluxo de Rotor nos Eixos Alfa Beta Zero'
hcopy bmp fab0p3
END

Rotina fab0g.t: Modelo de fluxo ortogonal com referencial síncrono.

CONTINUOUS SYSTEM FAB0G

" Version: 1.0
" Abstract: Modelo de Fluxo Alfa Beta Zero
" Description: Referencial Girando (wl=w1)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

```

" Inputs and outputs:

" States, derivates and time:
STATE f1a f1b f10 f2a f2b f20 omgm
DER df1a df1b df10 df2a df2b df20 domgm
TIME t

" Initializations:
s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=w1
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2
m=-kd/J

u1a=U
u1b=0
u10=0

" Equations:
w2=w1-abs(NP*omgm)
md=-3*NP*lh/(2*s*l1*l2)*(f1a*f2b-f1b*f2a)
df1a=a*f1a+b*f1b+c*f2a+u1a
df1b=b*f1a+a*f1b+c*f2b+u1b
df10=a*f10+c*f20+u10
df2a=d*f1a+e*f2a+f*f2b
df2b=d*f1b-f*f2a+e*f2b
df20=d*f10+e*f20
domgm=m*omgm+md/J

" Parameter values:
wl:376.9911
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

```

Rotina mfab0g.t: Macro MFAB0G.T para executar o programa fab0g.t

```

MACRO MFAB0G
" Version: 1.0
" Abstract: Macro para iniciar fab0g
" Description: Simular e plotar fab0g
" Revision: 1.0
" Author: Marcelo M. Cad

```

" Created: 06/11/99

" Enter commands here:

```
algor dopri45r
error 1e-6
syst fab0g
store f1a f1b f10 f2a f2b f20 omgm md
simu 0 0.5/f4
export f4 < f4 /0
split 1 2
ashow omgm
text 'Evolucao do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp fab0g1
newplot
ashow f1a f1b f10
text 'Fluxo de Estator nos Eixos Alfa Beta Zero'
hcopy bmp fab0g2
newplot
ashow f2a f2b f20
text 'Fluxo de Rotor nos Eixos Alfa Beta Zero'
hcopy bmp fab0g3
END
```

Rotina iab0p.t: Modelo de corrente ortogonal com referencial fixo no estator.

CONTINUOUS SYSTEM IAB0P

" Version: 1.0
" Abstract: Modelo de Corrente Alfa Beta Zero
" Description: Referencial Parado (wl=0)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

" Inputs and outputs:

" States, derivates and time:
STATE i1a i1b i10 i2a i2b i20 omgm
DER di1a di1b di10 di2a di2b di20 domgm
TIME t

" Initializations:

```
s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=1/s*(wl-w2*(1-s))
c=r2*lh/(s*l1*l2)
d=lh/(s*l1)*(wl-w2)
e=r1*lh/(s*l1*l2)
f=lh/(s*l2)*(wl-w2)
g=-r2/(s*l2)
h=1/s*(wl*(1-s)-w2)
m=-kd/J
v1=1/(s*l1)
v2=-lh/(s*l1*l2)
```

```

u1a=U*cos(w1*t)
u1b=U*sin(w1*t)
u10=0

" Equations:
w2=wl-abs(NP*omgm)
md=-3*NP*lh/2*(i1a*i2b-i1b*i2a)
di1a=a*i1a+b*i1b+c*i2a+d*i2b+v1*u1a
di1b=-b*i1a+a*i1b-d*i2a+c*i2b+v1*u1b
di10=a*i10+c*i20+v1*u10
di2a=e*i1a-f*i1b+g*i2a-h*i2b+v2*u1a
di2b=f*i1a+e*i1b+h*i2a+g*i2b+v2*u1b
di20=e*i10+g*i20+v2*u10
domgm=m*omgm+md/J

" Parameter values:
wl:0
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

```

Rotina miab0p.t: Macro MIAB0P.T para executar o programa iab0p.t

```

MACRO MIAB0P
" Version: 1.0
" Abstract: Macro para iniciar iab0p
" Description: Simular e plotar iab0p
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst iab0p
store i1a i1b i10 i2a i2b i20 omgm md
simu 0 0.5/f9
export f9 < f9 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp iab0p1
newplot
ashow i1a i1b i10

```

```

text 'Corrente de Estator nos Eixos Alfa Beta Zero'
hcopy bmp iab0p2
newplot
ashow i2a i2b i20
text 'Corrente de Rotor nos Eixos Alfa Beta Zero'
hcopy bmp iab0p3
END

```

Rotina iab0g.t: Modelo de corrente ortogonal com referencial síncrono.

CONTINUOUS SYSTEM IAB0G

```

" Version: 1.0
" Abstract: Modelo de Corrente Alfa Beta Zero
" Description: Referencial Girando (wl=w1)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

```

" Inputs and outputs:

```

" States, derivates and time:
STATE i1a i1b i10 i2a i2b i20 omgm
DER di1a di1b di10 di2a di2b di20 domgm
TIME t

```

" Initializations:

```

s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=1/s*(wl-w2*(1-s))
c=r2*lh/(s*l1*l2)
d=lh/(s*l1)*(wl-w2)
e=r1*lh/(s*l1*l2)
f=lh/(s*l2)*(wl-w2)
g=-r2/(s*l2)
h=1/s*(wl*(1-s)-w2)
m=-kd/J
v1=1/(s*l1)
v2=-lh/(s*l1*l2)

```

```

u1a=U
u1b=0
u10=0

```

" Equations:

```

w2=wl-abs(NP*omgm)
md=-3*NP*lh/2*(i1a*i2b-i1b*i2a)
di1a=a*i1a+b*i1b+c*i2a+d*i2b+v1*u1a
di1b=-b*i1a+a*i1b-d*i2a+c*i2b+v1*u1b
di10=a*i10+c*i20+v1*u10
di2a=e*i1a-f*i1b+g*i2a-h*i2b+v2*u1a
di2b=f*i1a+e*i1b+h*i2a+g*i2b+v2*u1b
di20=e*i10+g*i20+v2*u10
domgm=m*omgm+md/J

```

" Parameter values:

```

wl:376.9911

```



```

w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

```

END

Rotina miab0g.t: Macro MIAB0G.T para executar o programa iab0g.t

```

MACRO MIAB0G
" Version:    1.0
" Abstract:   Macro para iniciar iab0g
" Description: Simular e plotar iab0g
" Revision:   1.0
" Author:     Marcelo M. Cad
" Created:    06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst iab0g
store i1a i1b i10 i2a i2b i20 omgm md
simu 0 0.5/f10
export f10 < f10 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp iab0g1
newplot
ashow i1a i1b i10
text 'Corrente de Estator nos Eixos Alfa Beta Zero'
hcopy bmp iab0g2
newplot
ashow i2a i2b i20
text 'Corrente de Rotor nos Eixos Alfa Beta Zero'
hcopy bmp iab0g3
END

```

B.2.2) Rotinas a Serem Usadas no Programa Octave

Rotina fab0p.m: Modelo de fluxo ortogonal com referencial fixo no estator.

```

%fab0p.m - rotina para modelo de fluxo alfabetazero com referencial parado
function yp = fab0p (y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;

```

```

r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
kd=0; M=0.027; NP=2; w1=0; u1a=U*cos(w1*t); u1b=U*sin(w1*t); u10=0;

```

```

w2=w1-abs(NP*y(7));
md=3*NP*lh/(2*s*l1*l2)*(y(1)*y(5)-y(2)*y(4));

```

```

a=-r1/(s*l1);
b=w1;
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
E=-r2/(s*l2);
f=w2;
m=-kd/M;

```

```

yp=[a b 0 c 0 0 0; ...
    -b a 0 0 c 0 0; ...
      0 0 a 0      0 c 0; ...
      d 0 0 E f 0 0; ...
      0 d 0 -f E 0 0; ...
      0 0 d 0 0 E 0; ...
      0 0 0 0 0 0 m]* y + ...
[u1a;u1b;u10;0;0;0;md/M];

```

```

clear a b c d E f m

```

Rotina mfab0p.m: Macro MFAB0P.M para executar o programa fab0p.m

```

clear all
%mfab0p.m - rotina para execução do programa fab0p.m
%parametros para calcular o md
l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;NP=2;
%definicao do tempo de simulacao
t=linspace(0,0.5,1500);
%definicao do vetor de condicoes iniciais
x0=[0;0;0;0;0;0;0];
%definicao do tempo para calculo do tempo de simulacao
t0=clock;
%rotina para simulacao
y=lsode("fab0p",x0,t);
%construcao dos graficos
w=figure(1);
%calculo do md
md=3*NP*lh/(2*s*l1*l2)*(y(:,1).*y(:,5)-y(:,2).*y(:,4));
subplot(1,2,1)
title('Evolução do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Fluxo de Estator nos Eixos Alfa Beta Zero');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;mplot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Fluxo de Rotor nos Eixos Alfa Beta Zero');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;mplot(t,y(:,4),t,y(:,5),t,y(:,6));
%calculo do tempo de simulacao
tso_fab0p = etime(clock(),t0)
clear w z z1 z2 wz wz1 wz2

```

Rotina fab0g.m: Modelo de fluxo ortogonal com referencial síncrono.

%fab0g.m - rotina para modelo de fluxo alfabetazero com referencial parado

function yp = fab0g (y,t)

%parametros da maquina

w1=120*pi;w2=0;md=0;U=311.1270;

r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;

kd=0; M=0.027;NP=2;w1=376.9911;u1a=U;u1b=0;u10=0;

w2=w1-abs(NP*y(7));

md=3*NP*lh/(2*s*l1*l2)*(y(1)*y(5)-y(2)*y(4));

a=-r1/(s*l1);

b=w1;

c=r1*lh/(s*l1*l2);

d=r2*lh/(s*l1*l2);

E=-r2/(s*l2);

f=w2;

m=-kd/M;

yp=[a b 0 c 0 0 0; ...

-b a 0 0 c 0 0; ...

0 0 a 0 0 c 0; ...

d 0 0 E f 0 0; ...

0 d 0 -f E 0 0; ...

0 0 d 0 0 E 0; ...

0 0 0 0 0 m]* y + ...

[u1a;u1b;u10;0;0;0;md/M];

clear a b c d E f m

Rotina mfab0g.m: Macro MFAB0G.M para executar o programa fab0g.m

%mfab0g.m - rotina para execução do programa fab0g.m

%parâmetros para calcular o md

l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;NP=2;

%definição do tempo de simulação

t=linspace(0,0.5,1500);

%definição do vetor de condicoes iniciais

x0=[0;0;0;0;0;0;0];

%definição do tempo para cálculo do tempo de simulação

t0=clock;

%rotina para simulação

y=lsode("fab0g",x0,t);

%construcao dos graficos

w=figure(1);

%cálculo do md

md=3*NP*lh/(2*s*l1*l2)*(y(:,1).*y(:,5)-y(:,2).*y(:,4));

subplot(1,2,1)

title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));

subplot(1,2,2)

title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);

z=figure(2);

title('Fluxo de Estator nos Eixos Alfa Beta Zero');ylabel('Fluxo [Wb]');xlabel('Tempo

[s]');grid;plot(t,y(:,1),t,y(:,2),t,y(:,3));

z1=figure(3);

```

title('Fluxo de Rotor nos Eixos Alfa Beta Zero');ylabel('Fluxo [Wb]');xlabel('Tempo [s]');grid;
subplot(4,1,1);plot(t,y(:,4),t,y(:,5),t,y(:,6));
%cálculo do tempo de simulação
tso_fab0g = etime(clock(),t0)
clear w z z1 z2 wz wz1 wz2

```

Rotina iab0p.m: Modelo de corrente ortogonal com referencial fixo no estator.

```

%iab0p.m - rotina para modelo de corrente ortogonal com referencial parado
function yp = iab0p (y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;kd=0;M=0.027;NP=2;
wl=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);u10=0;

w2=w1-abs(NP*y(7));
md=3*NP*lh/2*(y(1)*y(5)-y(2)*y(4));

a=-r1/(s*l1);
b=1/s*(w1-w2*(1-s));
c=r2*lh/(s*l1*l2);
d=lh/(s*l1)*(w1-w2);
E=r1*lh/(s*l1*l2);
f=lh/(s*l2)*(w1-w2);
g=-r2/(s*l2);
h=1/s*(w1*(1-s)-w2);
m=-kd/M;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b 0 c d 0 0; ...
    -b a 0 -d c 0 0; ...
    0 0 a 0 0 c 0; ...
    E -f 0 g -h 0 0; ...
    f E 0 h g 0 0; ...
    0 0 E 0 0 g 0; ...
    0 0 0 0 0 m]* y + ...
[v1*u1a;v1*u1b;v1*u10;v2*u1a;v2*u1b;v2*u10;md/M];

clear a b c d E f g h m v1 v2

```

Rotina miab0p.m: Macro MIAB0P.M para executar o programa iab0p.m

```

clear all
%miab0p.m - rotina para execução do programa iab0p.m
%parâmetros para calcular o md
l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;NP=2;
%definição do tempo de simulação
t=linspace(0,0.5,1500);
%definição do vetor de condicoes iniciais
x0=[0;0;0;0;0;0;0];
%definição do tempo para calculo do tempo de simulação
t0=clock;
%rotina para simulação
y=lsode("iab0p",x0,t);
%construção dos gráficos
w=figure(1);

```

```

%cálculo do md
md=3*NP*lh/2*(y(:,1).*y(:,5)-y(:,2).*y(:,4));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Corrente de Estator nos Eixos Alfa Beta Zero');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;
mplot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Corrente de Rotor nos Eixos Alfa Beta Zero');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;
mplot(t,y(:,4),t,y(:,5),t,y(:,6));
%cálculo do tempo de simulação
tso_iab0p = etime(clock(),t0)
clear w z z1

```

Rotina iab0g.m: Modelo de fluxo ortogonal com referencial síncrono.

```

%iortg.m - rotina para modelo de corrente ortogonal com referencial girando
function yp = iab0g (y,t)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;kd=0;M=0.027;NP=2;
wl=376.9911;u1a=U;u1b=0;u10=0;

w2=w1-abs(NP*y(7));
md=3*NP*lh/2*(y(1)*y(5)-y(2)*y(4));

a=-r1/(s*l1);
b=1/s*(wl-w2*(1-s));
c=r2*lh/(s*l1*l2);
d=lh/(s*l1)*(wl-w2);
E=r1*lh/(s*l1*l2);
f=lh/(s*l2)*(wl-w2);
g=-r2/(s*l2);
h=1/s*(wl*(1-s)-w2);
m=-kd/M;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b 0 c d 0 0; ...
    -b a 0 -d c 0 0; ...
    0 0 a 0 0 c 0; ...
    E -f 0 g -h 0 0; ...
    f E 0 h g 0 0; ...
    0 0 E 0 0 g 0; ...
    0 0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u10;v2*u1a;v2*u1b;v2*u10;md/M];

```

```
clear a b c d E f g h m v1 v2
```

Rotina miab0g.m: Macro MIAB0G.M para executar o programa iab0g.m

```

clear all
%miab0g.m - rotina para execução do programa iab0g.m
%parâmetros para calcular o md
l1=0.35085;l2=0.35085;lh=0.33615;s=0.0820;NP=2;

```

```

%definição do tempo de simulação
t=linspace(0,0.5,1500);
%definição do vetor de condicoes iniciais
x0=[0;0;0;0;0;0;0];
%definição do tempo para calculo do tempo de simulação
t0=clock;
%rotina para simulação
y=lsode("iab0g",x0,t);
%construção dos gráficos
w=figure(1);
%cálculo do md
md=3*NP*lh/2*(y(:,1).*y(:,5)-y(:,2).*y(:,4));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Vel. [rad/s]');grid;plot(t,y(:,7));
subplot(1,2,2)
title('Torque Eletromagnético');ylabel('Torque [Nm]');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Corrente de Estator nos Eixos Alfa Beta Zero');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;subplot(t,y(:,1),t,y(:,2),t,y(:,3));
z1=figure(3);
title('Corrente de Rotor nos Eixos Alfa Beta Zero');ylabel('Corrente [A]');xlabel('Tempo [s]');grid;subplot(t,y(:,4),t,y(:,5),t,y(:,6));
%cálculo do tempo de simulação
tso_iab0g = etime(clock(),t0)
clear w z z1

```

B.2.3) Rotinas a Serem Usadas no Programa Matlab

Rotina fab0p.m: Modelo de fluxo ortogonal com referencial fixo no estator.

```

%fab0p.m - rotina para modelo de fluxo alfabetazero com referencial
%fixo no estator
function yp = fab0p (t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=0;u1a=U*cos(w1*t);
u1b=U*sin(w1*t);u10=0;

w2=w1-abs(NP*y(7));
md=3*NP*lh/(2*s*l1*l2)*(y(1)*y(5)-y(2)*y(4));

a=-r1/(s*l1);
b=w1;
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
e=-r2/(s*l2);
f=w2;
m=-kd/J;

yp=[a b 0 c 0 0 0; ...
-b a 0 0 c 0 0; ...
0 0 a 0 0 c 0; ...
d 0 0 e f 0 0; ...
0 d 0 -f e 0 0; ...
0 0 d 0 0 e 0; ...

```

```

0 0 0 0 0 0 m]* y + ...
[u1a;u1b;u10;0;0;0;md/J];

```

```
clear a b c d e f m
```

Rotina fab0g.m: Modelo de fluxo ortogonal com referencial síncrono.

```
%fab0g.m - rotina para modelo de fluxo alfabetazero com referencial síncrono
```

```
function yp = yab0 (t,y)
```

```
%parametros da maquina
```

```
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
```

```
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
```

```
wl=376.9911;u1a=U;u1b=0;u10=0;
```

```
w2=wl-abs(NP*y(7));
```

```
md=3*NP*lh/(2*s*l1*l2)*(y(1)*y(5)-y(2)*y(4));
```

```
a=-r1/(s*l1);
```

```
b=wl;
```

```
c=r1*lh/(s*l1*l2);
```

```
d=r2*lh/(s*l1*l2);
```

```
e=-r2/(s*l2);
```

```
f=w2;
```

```
m=-kd/J;
```

```
yp=[a b 0 c 0 0 0; ...
```

```
-b a 0 0 c 0 0; ...
```

```
0 0 a 0 0 c 0; ...
```

```
d 0 0 e f 0 0; ...
```

```
0 d 0 -f e 0 0; ...
```

```
0 0 d 0 0 e 0; ...
```

```
0 0 0 0 0 0 m]* y + ...
```

```
[u1a;u1b;u10;0;0;0;md/J];
```

```
clear a b c d e f m
```

Rotina iab0p.m: Modelo de corrente ortogonal com referencial fixo no estator.

```
%iab0p.m - rotina para modelo de corrente alfabetazero
```

```
% com referencial fixo no estator
```

```
function yp = iab0p (t,y)
```

```
%parametros da maquina
```

```
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
```

```
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
```

```
wl=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);u10=0;
```

```
w2=wl-abs(NP*y(7));
```

```
md=3*NP*lh/2*(y(1)*y(5)-y(2)*y(4));
```

```
a=-r1/(s*l1);
```

```
b=1/s*(wl-w2*(1-s));
```

```
c=r2*lh/(s*l1*l2);
```

```
d=lh/(s*l1)*(wl-w2);
```

```
e=r1*lh/(s*l1*l2);
```

```
f=lh/(s*l2)*(wl-w2);
```

```
g=-r2/(s*l2);
```

```
h=1/s*(wl*(1-s)-w2);
```

```
m=-kd/J;
```

```

v1=1/(s*I1);
v2=-lh/(s*I1*I2);

yp=[a b 0 c d 0 0; ...
    -b a 0 -d c 0 0; ...
     0 0 a 0 0 c 0; ...
    e -f 0 g -h 0 0; ...
    f e 0 h g 0 0; ...
     0 0 e 0 0 g 0; ...
    0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u10;v2*u1a;v2*u1b;v2*u10;md/J];

```

```
clear a b c d e f g h m v1 v2
```

Rotina iab0g.m: Modelo de corrente ortogonal com referencial síncrono.

%iab0g.m - rotina para modelo de corrente alfabetazero com referencial síncrono

```

function yp = iab0g (t,y)
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;u1a=U;u1b=0;u10=0;

```

```

w2=wl-abs(NP*y(7));
md=3*NP*lh/2*(y(1)*y(5)-y(2)*y(4));

```

```

a=-r1/(s*I1);
b=1/s*(wl-w2*(1-s));
c=r2*lh/(s*I1*I2);
d=lh/(s*I1)*(wl-w2);
e=r1*lh/(s*I1*I2);
f=lh/(s*I2)*(wl-w2);
g=-r2/(s*I2);
h=1/s*(wl*(1-s)-w2);
m=-kd/J;
v1=1/(s*I1);
v2=-lh/(s*I1*I2);

```

```

yp=[a b 0 c d 0 0; ...
    -b a 0 -d c 0 0; ...
     0 0 a 0 0 c 0; ...
    e -f 0 g -h 0 0; ...
    f e 0 h g 0 0; ...
     0 0 e 0 0 g 0; ...
    0 0 0 0 0 m]*y + ...
[v1*u1a;v1*u1b;v1*u10;v2*u1a;v2*u1b;v2*u10;md/J];

```

```
clear a b c d e f g h m v1 v2
```


B.3) Notação Vetorial

Conforme descrito no capítulo 4 apresentam-se aqui as rotinas construídas para simulação dos modelos utilizando como variável de estado o fluxo e referencial fixo no estator para o modelo vetorial.

B.3.1) Rotinas a Serem Usadas no Programa SimnonTM

Rotina fvetsp.t: Modelo de fluxo vetorial com referencial fixo no estator.

```
CONTINUOUS SYSTEM FVETP
" Version:      1.0
" Abstract:     Modelo de Fluxo Vetorial Separado
" Description:   Referencial Parado (wl=0)
" Revision:     1.0
" Author:       Marcelo M. Cad
" Created:      05/11/99

" Inputs and outputs:

" States, derivates and time:
STATE f1a f1b f2a f2b omgm
DER df1a df1b df2a df2b domgm
TIME t

" Initializations:
s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=w1
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2
m=-kd/J

u1a=U*cos(w1*t)
u1b=U*sin(w1*t)

" Equations:
w2=w1-abs(NP*omgm)
md=-3*NP*lh/(2*s*l1*l2)*(f1a*f2b-f1b*f2a)
df1a=a*f1a+b*f1b+c*f2a+u1a
df1b=b*f1a+a*f1b+c*f2b+u1b
df2a=d*f1a+e*f2a+f*f2b
df2b=d*f1b-f*f2a+e*f2b
domgm=m*omgm+md/J

" Parameter values:
wl:0
w1:376.9911
U:311.13
r1:7.56
```

r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

Rotina mfvetp.t: Macro MFVETP.T para executar o programa fvetp.t

MACRO MFVETP

" Version: 1.0
" Abstract: Macro para iniciar fvetp
" Description: Simular e plotar fvetp
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst fvetp
store f1a f1b f2a f2b omgm md
simu 0 0.5/f5
export f5 < f5 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp fvetp1
newplot
ashow f1a f1b
text 'Comportamento Transitório do Fluxo de Estator Real e Imaginario'
hcopy bmp fvetp2
newplot
ashow f2a f2b
text 'Comportamento Transitório do Fluxo de Rotor Real e Imaginario'
hcopy bmp fvetp3
newplot
ashow f1b (f1a)
text 'Composicao Complexa do Fluxo de Estator'
hcopy bmp fvetp4
newplot
ashow f2b (f2a)
text 'Composicao Complexa do Fluxo de Rotor'
hcopy bmp fvetp5
END

Rotina fvetg.t: Modelo de fluxo vetorial com referencial síncrono.

CONTINUOUS SYSTEM FVETG

" Version: 1.0
" Abstract: Modelo de Fluxo Vetorial Separado

```

" Description: Referencial Girando (wl=w1)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99

" Inputs and outputs:

" States, derivates and time:
STATE f1a f1b f2a f2b omgm
DER df1a df1b df2a df2b domgm
TIME t

" Initializations:
s=1-lh^2/l1^2
a=-r1/(s*l1)
b=w1
c=r1*lh/(s*l1*l2)
d=r2*lh/(s*l1*l2)
e=-r2/(s*l2)
f=w2
m=-kd/J
u1a=U
u1b=0

" Equations:
w2=w1-abs(NP*omgm)
md=-3*NP*lh/(2*s*l1*l2)*(f1a*f2b-f1b*f2a)
df1a=a*f1a+b*f1b+c*f2a+u1a
df1b=b*f1a+a*f1b+c*f2b+u1b
df2a=d*f1a+e*f2a+f*f2b
df2b=d*f1b-f*f2a+e*f2b
domgm=m*omgm+md/J

" Parameter values:
wl:376.9911
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

```

Rotina mfvetg.t: Macro MFVETG.T para executar o programa fvetg.t

```

MACRO MFVETG
" Version: 1.0
" Abstract: Macro para iniciar fvetg
" Description: Simular e plotar fvetg
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

```

" Enter commands here:

```
algor dopri45r
error 1e-6
syst fvetg
store f1a f1b f2a f2b omgm md
simu 0 0.5/f6
export f6 <f6 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md
text 'Torque Eletromagnetico'
hcopy bmp fvetg1
newplot
ashow f1a f1b
text 'Comportamento Transitório do Fluxo de Estator Real e Imaginario'
hcopy bmp fvetg2
newplot
ashow f2a f2b
text 'Comportamento Transitório do Fluxo de Rotor Real e Imaginario'
hcopy bmp fvetg3
newplot
ashow f1b (f1a)
text 'Composicao Complexa do Fluxo de Estator'
hcopy bmp fvetg4
newplot
ashow f2b (f2a)
text 'Composicao Complexa do Fluxo de Rotor'
hcopy bmp fvetg5
END
```

Rotina ivetp.t: Modelo de corrente vetorial com referencial fixo no estator.

CONTINUOUS SYSTEM IVSP

```
" Version: 1.0
" Abstract: Modelo de Corrente Vetorial Separado
" Description: Referencial Parado (wl=0)
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 05/11/99
```

" Inputs and outputs:

```
" States, derivates and time:
STATE i1a i1b i2a i2b omgm
DER di1a di1b di2a di2b domgm
TIME t
```

" Initializations:

```
s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=1/s*(wl-(1-s)*w2)
c=r2*(1-s)/(s*lh)
d=1/s*(lh/l1*wl-l2/lh*(1-s)*w2)
```

```

e=r1*(1-s)/(s*lh)
f=1/s*(1/lh*(1-s)*w1-lh/l2*w2)
g=-r2/(s*l2)
h=1/s*(w2-(1-s)*w1)
m=-kd/J
v1=1/(s*l1)
v2=-lh/(s*l1*l2)

u1a=U*cos(w1*t)
u1b=U*sin(w1*t)

" Equations:
w2=w1-abs(NP*omgm)
md=-3*NP*lh/2*(i1a*i2b-i1b*i2a)
di1a=a*i1a+b*i1b+c*i2a+d*i2b+v1*u1a
di1b=-b*i1a+a*i1b-d*i2a+c*i2b+v1*u1b
di2a=e*i1a-f*i1b+g*i2a+h*i2b+v2*u1a
di2b=f*i1a+e*i1b-h*i2a+g*i2b+v2*u1b
domgm=m*omgm+md/J

" Parameter values:
w1:0
w1:376.9911
U:311.13
r1:7.56
r2:3.84
l1:0.35085
l2:0.35085
lh:0.33615
kd:0
J:0.027
NP:2

END

```

Rotina mivetp.t: Macro MIVETP.T para executar o programa ivetp.t

```

MACRO MIVETP
" Version: 1.0
" Abstract: Macro para iniciar ivetp
" Description: Simular e plotar ivetp
" Revision: 1.0
" Author: Marcelo M. Cad
" Created: 06/11/99

" Enter commands here:

algor dopri45r
error 1e-6
syst ivetp
store i1a i1b i2a i2b omgm md
simu 0 0.5/f11
export f11 < f11 /0
split 1 2
ashow omgm
text 'Aceleração do Motor'
ashow md

```

```

text 'Torque Eletromagnetico'
hcopy bmp ivetp1
newplot
ashow i1a i1b
text 'Comportamento Transitório da Corrente de Estator Real e Imaginario'
hcopy bmp ivetp2
newplot
ashow i2a i2b
text 'Comportamento Transitório da Corrente de Rotor Real e Imaginario'
hcopy bmp ivetp3
newplot
ashow i1b (i1a)
text 'Composicao Complexa da Corrente de Estator'
hcopy bmp ivetp4
newplot
ashow i2b (i2a)
text 'Composicao Complexa da Corrente de Rotor'
hcopy bmp ivetp5
END

```

Rotina ivetg.t: Modelo de corrente vetorial com referencial síncrono.

CONTINUOUS SYSTEM IVSG

```

" Version:    1.0
" Abstract:   Modelo de Corrente Vetorial Separado
" Description: Referencial Girando (wl=w1)
" Revision:   1.0
" Author:     Marcelo M. Cad
" Created:    05/11/99

```

" Inputs and outputs:

```

" States, derivates and time:
STATE i1a i1b i2a i2b omgm
DER di1a di1b di2a di2b domgm
TIME t

```

```

" Initializations:
s=1-lh^2/l1^2
T1=l1/r1
T2=l2/r2
a=-r1/(s*l1)
b=1/s*(wl-(1-s)*w2)
c=r2*(1-s)/(s*lh)
d=1/s*(lh/l1*wl-l2/lh*(1-s)*w2)
e=r1*(1-s)/(s*lh)
f=1/s*(l1/lh*(1-s)*wl-lh/l2*w2)
g=-r2/(s*l2)
h=1/s*(w2-(1-s)*wl)
m=-kd/J
v1=1/(s*l1)
v2=-lh/(s*l1*l2)

```

```

u1a=U
u1b=0

```

```

" Equations:
w2=wl-abs(NP*omgm)

```

```

md=-3*NP*lh/2*(i1a*i2b-i1b*i2a)
di1a=a*i1a+b*i1b+c*i2a+d*i2b+v1*u1a
di1b=-b*i1a+a*i1b-d*i2a+c*i2b+v1*u1b
di2a=e*i1a-f*i1b+g*i2a+h*i2b+v2*u1a
di2b=f*i1a+e*i1b-h*i2a+g*i2b+v2*u1b
domgm=m*omgm+md/J

```

" Parameter values:

wl:376.9911

w1:376.9911

U:311.13

r1:7.56

r2:3.84

l1:0.35085

l2:0.35085

lh:0.33615

kd:0

J:0.027

NP:2

END

Rotina mivetg.t: Macro MIVETG.T para executar o programa fietg.t

MACRO MIVETG

" Version: 1.0

" Abstract: Macro para iniciar ivetg

" Description: Simular e plotar ivetg

" Revision: 1.0

" Author: Marcelo M. Cad

" Created: 06/11/99

" Enter commands here:

algor dopri45r

error 1e-6

syst ivetg

store i1a i1b i2a i2b omgm md

simu 0 0.5/f12

export f12 < f12 /0

split 1 2

ashow omgm

text 'Aceleração do Motor'

ashow md

text 'Torque Eletromagnetico'

hcopy bmp ivetg1

newplot

ashow i1a i1b

text 'Comportamento Transitório da Corrente de Estator Real e Imaginario'

hcopy bmp ivetg2

newplot

ashow i2a i2b

text 'Comportamento Transitório da Corrente de Rotor Real e Imaginario'

hcopy bmp ivetg3

newplot

ashow i1b (i1a)

text 'Composicao Complexa da Corrente de Estator'

```

hcopy bmp ivetg4
newplot
ashow i2b (i2a)
text 'Composicao Complexa da Corrente de Rotor'
hcopy bmp ivetg5
END

```

B.3.2) Rotinas a Serem Usadas no Programa Octave

Rotina fvetp.m: Modelo de fluxo vetorial com referencial fixo no estator.

```

% fvetp.m - rotina para calculo do modelo de fluxo vetorial com referencial fixo no estator.
function yp = fvetp (y,t);
% dados da máquina
w1=120*pi;w2=0;md=0;U=311.1270;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
kd=0; M=0.027;NP=2;T1=0.0464; T2=0.0914;w1=0;u1a=U*cos(w1*t);
u1b=U*sin(w1*t);
% equações para ode
w2=w1-abs(NP*y(5));
md=-3*NP*lh/(2*s*l1*l2)*(y(1)*y(4)-y(2)*y(3));

a=-r1/(s*l1);
b=w1;
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
E=-r2/(s*l2);
f=w2;
m=-kd/M;

yp=[a b c 0 0;...
    -b a 0 c 0;...
    d 0 E f 0;...
    0 d -f E 0;...
    0 0 0 0 m]*y+[u1a;u1b;0;0;md/M];

clear a b c d E f m

```

Rotina mfvet.m: Macro MFVETP.M para executar o programa fvetp.m

```

% mfvetp.m - rotina para execução do programa fvetp.m
% parâmetros para calcular o md
l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;NP=2;
% definição do tempo de simulação
t=linspace(0,0.5,1500);
% definição do vetor de condições iniciais
x0=[0;0;0;0;0];
% definição do tempo para calculo do tempo de simulação
t0=clock;
% rotina para simulação
y=lsode("fvetp",x0,t);
% construção das saídas
w=figure(1);
md=-3*NP*lh/(2*s*l1*l2)*(y(:,1).*y(:,4)-y(:,2).*y(:,3));
subplot(1,2,1)
title('Aceleração do Motor');xlabel('Tempo [s]');ylabel('Velocidade [rad/s]');grid;plot(t,y(:,5));

```



```

subplot(1,2,2)
title('Conjugado da Aceleração');ylabel('Torque');xlabel('Tempo [s]');grid;plot(t,md);
z=figure(2);
title('Fluxo de Estator Real');ylabel('Fluxo');xlabel('Tempo [s]');grid;plot(t,y(:,1));
z1=figure(3);
title('Fluxo de Estator Imaginário');ylabel('Fluxo');xlabel('Tempo [s]');grid;plot(t,y(:,2));
z2=figure(4);
title('Composicao Complexa do Fluxo de Estator');ylabel('Imaginario');xlabel('Real');grid;plot(y(:,1),y(:,2));
wz=figure(5);
title('Fluxo de Rotor Real');ylabel('Fluxo');xlabel('Tempo [s]');grid;plot(t,y(:,3));
wz1=figure(6);
title('Fluxo de Rotor Imaginário');ylabel('Fluxo');xlabel('Tempo [s]');grid;plot(t,y(:,4));
wz2=figure(7);
title('Composicao Complexa do Fluxo de Rotor');ylabel('Imaginario');xlabel('Real');grid;plot(y(:,3),y(:,4));
%calculo final do tempo de simulação
tso_fvetp = etime(clock(),t0)
clear w z z1 z2 wz wz1 wz2

```

B.3.3) Rotinas para uso no Programa Matlab

Rotina fvetp.m: Modelo de fluxo vetorial com referencial fixo no estator.

```

% fvetp.m - rotina para calculo do modelo de fluxo vetorial separado em parte real e imaginaria
% com referencial fixo no estator
function yp = fvetp (t,y);
% dados da máquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
w1=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);
% equações para ode
w2=w1-abs(NP*y(5));
md=-3*NP*lh/(2*s*l1*l2)*(y(1)*y(4)-y(2)*y(3));

a=-r1/(s*l1);
b=w1;
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
e=-r2/(s*l2);
f=w2;
m=-kd/J;

yp=[a b c 0 0;...
    -b a 0 c 0;...
    d 0 e f 0;...
    0 d -f e 0;...
    0 0 0 0 m]*y+[u1a;u1b;0;0;md/J];

clear a b c d e f m

```

Rotina fvetg.m: Modelo de fluxo vetorial com referencial síncrono.

```

% fvetg.m - rotina para calculo do modelo de fluxo vetorial separado em parte real e imaginaria
% com referencial síncrono
function yp = fvetg (t,y);
% dados da máquina

```

```

w1=120*pi;w2=0;md=0;U=311.1270;kd=0;J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;u1a=U;u1b=0;
%equações para ode
w2=wl-abs(NP*y(5));
md=-3*NP*lh/(2*s*l1*l2)*(y(1)*y(4)-y(2)*y(3));

a=-r1/(s*l1);
b=wl;
c=r1*lh/(s*l1*l2);
d=r2*lh/(s*l1*l2);
e=-r2/(s*l2);
f=w2;
m=-kd/J;

yp=[a b c 0 0;...
    -b a 0 c 0;...
    d 0 e f 0;...
    0 d -f e 0;...
    0 0 0 0 m]*y+[u1a;u1b;0;0;md/J];

clear a b c d e f m

```

Rotina ivetp.m: Modelo de corrente vetorial com referencial fixo no estator.

```

%ivetp.m - rotina para calculo do modelo de corrente vetorial separado
% em parte real e imaginaria com referencial fixo no estator
function yp = ivetp(t,y);
%dados da máquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);
%equações para ode
w2=wl-abs(NP*y(5));
md=3/2*NP*lh*(y(2)*y(3)-y(1)*y(4));

a=-r1/(s*l1);
b=1/s*(wl-(1-s)*w2);
c=r2*(1-s)/(s*lh);
d=1/s*(lh/l1*w1-l2/lh*(1-s)*w2);
e=r1*(1-s)/(s*lh);
f=1/s*(l1/lh*(1-s)*wl-lh/l2*w2);
g=-r2/(s*l2);
h=1/s*(w2-(1-s)*wl);
m=-kd/J;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b c d 0;...
    -b a -d c 0;...
    e -f g h 0;...
    f e -h g 0;...
    0 0 0 0 m]*y+[v1*u1a;v1*u1b;v2*u1a;v2*u1b;md/J];

clear a b c d e f g h m v1 v2

```

Rotina ivetg.m: Modelo de corrente vetorial com referencial síncrono.

```
%ivetg.m - rotina para calculo do modelo de corrente vetorial separado
%      em parte real e imaginaria com referencial síncrono
function yp = ivetg (t,y);
%dados da máquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;u1a=U;u1b=0;
%equações para ode
w2=wl-abs(NP*y(5));
md=3/2*NP*lh*(y(2)*y(3)-y(1)*y(4));

a=-r1/(s*l1);
b=1/s*(wl-(1-s)*w2);
c=r2*(1-s)/(s*lh);
d=1/s*(lh/l1*w1-l2/lh*(1-s)*w2);
e=r1*(1-s)/(s*lh);
f=1/s*(l1/lh*(1-s)*w1-lh/l2*w2);
g=-r2/(s*l2);
h=1/s*(w2-(1-s)*wl);
m=-kd/J;
v1=1/(s*l1);
v2=-lh/(s*l1*l2);

yp=[a b c d 0;...
    -b a -d c 0;...
    e -f g h 0;...
    f e -h g 0;...
    0 0 0 0 m]*y+[v1*u1a;v1*u1b;v2*u1a;v2*u1b;md/J];

clear a b c d e f g h m v1 v2
```

B.4) Notação Vetorial Complexa

Conforme descrito no capítulo 4 apresentam-se aqui as rotinas construídas para simulação dos modelos utilizando como variável de estado o fluxo e referencial fixo no estator para o modelo vetorial complexo, utilizando-se apenas os Programas Matlab® e Simulink® / Matlab®.

B.4.1) Rotinas para uso no Programa Matlab®

Rotina fcomp.m: Modelo de fluxo vetorial complexo com referencial fixo no estator.

```
%fcomp.m - rotina para modelo de fluxo vetorial complexo com referencial
%fixo no estator
function yp = fcomp (t,y);
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);U=u1a+i*u1b;
```

```

w2=w1-abs(NP*y(3));
md=3/2*NP*imag((-lh*y(1)/(11*12-lh^2)+11*y(2)/(11*12-lh^2))*conj(y(2)));

a=-(r1*12/(11*12-lh^2)+i*w1);
b=r1*lh/(11*12-lh^2);
c=r2*lh/(11*12-lh^2);
d=-(r2*11/(11*12-lh^2)+i*w2);
m=-kd/J;

yp=[a b 0;...
    c d 0;...
    0 0 m]*y + [U;0;md/J];

clear a b c d m

```

Rotina fcomg.m: Modelo de fluxo vetorial complexo com referencial síncrono.

```

%fcomg.m - rotina para modelo de fluxo vetorial complexo com referencial síncrono
function yp = fcomg (t,y);
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;U=311.1270;

w2=w1-abs(NP*y(3));
md=3/2*NP*imag((-lh*y(1)/(11*12-lh^2)+11*y(2)/(11*12-lh^2))*conj(y(2)));

a=-(r1*12/(11*12-lh^2)+i*w1);
b=r1*lh/(11*12-lh^2);
c=r2*lh/(11*12-lh^2);
d=-(r2*11/(11*12-lh^2)+i*w2);
m=-kd/J;

yp=[a b 0;...
    c d 0;...
    0 0 m]*y + [U;0;md/J];

clear a b c d m

```

Rotina icomp.m: Modelo de corrente vetorial complexo com referencial fixo no estator.

```

%icomp.m - rotina para modelo de corrente vetorial complexo
% com referencial fixo no estator
function yp = icomp (t,y);
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=0;u1a=U*cos(w1*t);u1b=U*sin(w1*t);U=u1a+i*u1b;

w2=w1-abs(NP*y(3));
md=3/2*NP*lh*imag(y(1)*conj(y(2)));
a=-(1/s)*(r1/l1+i*(w1-w2*(1-s)));
b=lh/(l1*s)*(r2/l2+i*(w2-w1));
c=lh/(l2*s)*(r1/l1+i*(w1-w2));
d=-(1/s)*(r2/l2+i*(w2-w1*(1-s)));
m=-kd/J;
v1=1/(l1*s);
v2=-lh/(s*l1*l2);

```

```

yp=[a b 0;...
    c d 0;...
    0 0 m]*y + [v1*U;v2*U;md/J];
clear a b c d m v1 v2

```

Rotina icomg.m: Modelo de corrente vetorial complexo com referencial síncrono.

```

%icomg.m - rotina para modelo de corrente vetorial complexo
% com referencial síncrono
function yp = icomg (t,y);
%parametros da maquina
w1=120*pi;w2=0;md=0;U=311.1270;kd=0; J=0.027;NP=2;
r1=7.56; r2=3.84; l1=0.35085; l2=0.35085; lh=0.33615; s=0.0820;
wl=376.9911;U=311.1270;

w2=w1-abs(NP*y(3));
md=3/2*NP*lh*imag(y(1)*conj(y(2)));
a=-(1/s)*(r1/l1+i*(w1-w2*(1-s)));
b=lh/(l1*s)*(r2/l2+i*(w2-w1));
c=lh/(l2*s)*(r1/l1+i*(w1-w2));
d=-(1/s)*(r2/l2+i*(w2-w1*(1-s)));
m=-kd/J;
v1=1/(l1*s);
v2=-lh/(s*l1*l2);
yp=[a b 0;...
    c d 0;...
    0 0 m]*y + [v1*U;v2*U;md/J];
clear a b c d m v1 v2

```

B.4.2) Rotinas para uso no Programa Simulink® / Matlab®

As rotinas criadas para uso no Simulink® / Matlab®, seguem a estrutura da Tabela 2, e como foi descrito no decorrer do trabalho, são todas em formato de diagrama de blocos, e foram apresentadas no capítulo 4, a seguir seguem as *S-Functions* criadas para execução da notação complexa.

Rotina flux1p.m: Modelo de fluxo complexo com referencial fixo no estator.

```

function [sys,x0,str,ts] = flux1p (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, w1, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;w1=0;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;

```

```

sizes.NumDiscStates=0;
sizes.NumOutputs=2;
sizes.NumInputs=4;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
sys=simsizes(sizes);
x0=[0 0];
str=[];
ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= u(1) - r1/(s*11)*x(1) + w1*x(2) + u(3);
    xi= u(2) - w1*x(1) - r1/(s*11)*x(2) + u(4);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina flux2p.m: Modelo de fluxo complexo com referencial fixo no estator.

```

function [sys,x0,str,ts] = flux2p (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, w1, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=3;
    sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1;
    sys=simsizes(sizes);
    x0=[0 0];
    str=[];
    ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= u(1) - r2/(s*l2)*x(1) + u(3)*x(2);
    xi= u(2) - r2/(s*l2)*x(2) - u(3)*x(1);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

```

```

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina flux1g.m: Modelo de fluxo complexo com referencial síncrono.

```

function [sys,x0,str,ts] = flux1g (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, wl, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;wl=120*pi;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=4;
    sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1;
    sys=simsizes(sizes);
    x0=[0 0];
    str=[];
    ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= u(1) - r1/(s*l1)*x(1) + wl*x(2) + u(3);
    xi= u(2) - wl*x(1) - r1/(s*l1)*x(2) + u(4);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina flux2g.m: Modelo de fluxo complexo com referencial síncrono.

```

function [sys,x0,str,ts] = flux2g (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, wl, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

```

```
r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;
```

```
switch flag
```

```
case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDis cStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=3;
    sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1;
    sys=simsizes(sizes);
    x0=[0 0];
    str=[];
    ts=[-1 0];
```

```
case 1 %Se flag=1, retorna número de estados
    xr= u(1) - r2/(s*l2)*x(1) + u(3)*x(2);
    xi= u(2) - r2/(s*l2)*x(2) - u(3)*x(1);
    sys=[xr xi];
```

```
case 3 %Se flag=3, retorna o número de saídas
    sys=x;
```

```
case { 2, 4, 9 } % estados nao utilizados
    sys=[];
```

```
otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;
```

Rotina corr1p.m: Modelo de corrente complexo com referencial fixo no estator.

```
function [sys,x0] = corr1p (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, w1, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3
```

```
r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;
kd=0; J=0.027;zp=2;T1=l1/r1;T2=l2/r2;w1=120*pi;w2=0;
```

```
switch flag
```

```
case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.Nu mOutputs=2;
    sizes.NumInputs=6;
    sizes.DirFeedthrough=0;
    sizes.NumSampleTimes=0;
    auxiliar=simsizes(sizes);
    sys=[2 0 2 6 0 0];
    x0=[0 0];
```



```

str=[];
ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= 1/(s*I1)*u(1) - r1/(s*I1)*x(1) + (1/s*u(3)-(1-s)/s*u(4))*x(2) + r2*lh/(s*I1*I2)*u(5) +
    lh/(s*I1)*(u(3)-u(4))*u(6);
    xi= 1/(s*I1)*u(2) - (1/s*u(3)-(1-s)/s*u(4))*x(1) - r1/(s*I1)*x(2) - lh/(s*I1)*(u(3)-u(4))*u(5) +
    r2*lh/(s*I1*I2)*u(6);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina corr2p.m: Modelo de corrente complexo com referencial fixo no estator.

```

function [sys,x0] = corr2p (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, w1, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;
kd=0; J=0.027;zp=2;T1=l1/r1;T2=l2/r2;w1=120*pi;w2=0;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=6;
    sizes.DirFeedthrough=0;
    sizes.NumSampleTimes=0;
    auxiliar=simsizes(sizes);
    sys=[2 0 2 6 0 0];
    x0=[0 0];
    str=[];
    ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= - lh/(s*I1*I2)*u(1) + r1*lh/(s*I1*I2)*u(3) - lh/(s*I2)*(u(5)-u(6))*u(4) - r2/(s*I2)*x(1) +
    (1/s*u(6)-(1-s)/s*u(5))*x(2);
    xi= - lh/(s*I1*I2)*u(2) + lh/(s*I2)*(u(5)-u(6))*u(3) + r1*lh/(s*I1*I2)*u(4) - (1/s*u(6)-(1-
    s)/s*u(5))*x(1) - r2/(s*I2)*x(2);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

```

```

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina corr1g.m: Modelo de corrente complexo com referencial síncrono.

```

function [sys,x0] = corr1p (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, wl, md, Lh, s
%entradas: u1, wmec, w1, w2
%saídas: y1, y2, y3

r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;
kd=0; J=0.027;zp=2;T1=l1/r1;T2=l2/r2;w1=120*pi;w2=0;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=6;
    sizes.DirFeedthrough=0;
    sizes.NumSampleTimes=0;
    auxiliar=simsizes(sizes);
    sys=[2 0 2 6 0 0];
    x0=[0 0];
    str=[];
    ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= 1/(s*l1)*u(1) - r1/(s*l1)*x(1) + (1/s*u(3)-(1-s)/s*u(4))*x(2) + r2*lh/(s*l1*l2)*u(5) +
    lh/(s*l1)*(u(3)-u(4))*u(6);
    xi= 1/(s*l1)*u(2) - (1/s*u(3)-(1-s)/s*u(4))*x(1) - r1/(s*l1)*x(2) - lh/(s*l1)*(u(3)-u(4))*u(5) +
    r2*lh/(s*l1*l2)*u(6);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```

Rotina corr2g.m: Modelo de corrente complexo com referencial síncrono.

```

function [sys,x0] = corr2g (t,x,u,flag)
%Laplace das equações diferenciais do motor de indução
%Variáveis usadas são L1, L2, R1, R2, wl, md, Lh, s
%entradas: u1, wmec, w1, w2

```

```

%saídas: y1, y2, y3

r1=7.56;r2=3.84;l1=0.35085;l2=0.35085;lh=0.33615;s=1-lh^2/l1^2;
kd=0; J=0.027;zp=2;T1=l1/r1;T2=l2/r2;w1=120*pi;w2=0;

switch flag

case 0 % Inicialização
    sizes=simsizes;
    sizes.NumContStates=2;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=2;
    sizes.NumInputs=6;
    sizes.DirFeedthrough=0;
    sizes.NumSampleTimes=0;
    auxiliar=simsizes(sizes);
    sys=[2 0 2 6 0 0];
    x0=[0 0];
    str=[];
    ts=[-1 0];

case 1 %Se flag=1, retorna número de estados
    xr= - lh/(s*l1*l2)*u(1) + r1*lh/(s*l1*l2)*u(3) - lh/(s*l2)*(u(5)-u(6))*u(4) - r2/(s*l2)*x(1) +
    (1/s*u(6)-(1-s)/s*u(5))*x(2);
    xi= - lh/(s*l1*l2)*u(2) + lh/(s*l2)*(u(5)-u(6))*u(3) + r1*lh/(s*l1*l2)*u(4) - (1/s*u(6)-(1-
    s)/s*u(5))*x(1) - r2/(s*l2)*x(2);
    sys=[xr xi];

case 3 %Se flag=3, retorna o número de saídas
    sys=x;

case { 2, 4, 9 } % estados nao utilizados
    sys=[];

otherwise % erro
    error(['Flag invalido = ',num2str(flag)]);
end;

```