

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS - GRADUAÇÃO EM ENGENHARIA ELÉTRICA

SUELLEN CUNHA DA SILVA

**BIBLIOTECA E IMPLEMENTAÇÃO DE MODELOS PARA REATORES
NATURALMENTE SATURADOS NA LINGUAGEM MODELS DO PROGRAMA
ATP**

Recife
2016

Suellen Cunha da Silva

Biblioteca e Implementação de Modelos para Reatores
Naturalmente Saturados na Linguagem MODELS do
Programa ATP

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de Mestre em Engenharia Elétrica. Área de Concentração: Processamento de Energia.

Orientador: Manoel Afonso de Carvalho Júnior, *PhD*.

Recife

2016

Catálogo na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

S586i Silva, Suellen Cunha da.
Biblioteca e implementação de modelos para reatores naturalmente saturados na linguagem MODELS do programa ATP / Suellen Cunha da Silva. - 2016.
194 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Manoel Afonso de Carvalho Júnior.
Coorientador: Prof. Dr. Geraldo Leite Torres.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG.
Programa de Pós-Graduação em Engenharia Elétrica, 2016.
Inclui Referências e Apêndice.

1. Engenharia Elétrica. 2. Reatores naturalmente saturados. 3. Linguagem MODELS. 4. Programa ATP. 5. Circuito Equivalente Magnético Unificado. I. Carvalho Júnior, Manoel Afonso de. (Orientador). II. Torres, Geraldo Leite. (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.)

BCTG/2016-324



Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

SUELLEN CUNHA DA SILVA

TÍTULO

**“BIBLIOTECA E IMPLEMENTAÇÃO DE MODELOS
PARA REATORES NATURALMENTE SATURADOS NA
LINGUAGEM *MODELS* DO PROGRAMA ATP”**

A comissão examinadora composta pelos professores: MANOEL AFONSO DE CARVALHO JÚNIOR, DEE/UFPE; RONALDO RIBEIRO BARBOSA DE AQUINO, DEE/UFPE e LUIZ ANTÔNIO MAGNATA DA FONTE, DEE/UFPE, sob a presidência do primeiro, consideram a candidata **SUELLEN CUNHA DA SILVA *APROVADA***.

Recife, 29 de agosto de 2016.

MARCELO CABRAL CAVALCANTI
Coordenador do PPGEE

MANOEL AFONSO DE CARVALHO JÚNIOR
Orientador e Membro Titular Interno

LUIZ ANTÔNIO MAGNATA DA FONTE
Membro Titular Externo

RONALDO RIBEIRO BARBOSA DE AQUINO
Membro Titular Interno

Dedico a Valdenira Cunha (in memoriam), minha avó.

Agradecimentos

Primeiramente agradeço a Deus, pois me abençoou durante toda a trajetória de estudos me proporcionando saúde, sabedoria e paciência. Ao meu marido Bruno Silva, que me incentivou e ajudou a durante estes dois anos, com tamanha persistência e sempre otimista. Aos meus pais e amigos do LDSP que me incentivaram e me ajudaram. Ao professor Manoel Afonso tenho profunda gratidão e admiração, pois me deu todo o suporte e dedicação sem exitar.

Resumo

Atualmente, a expansão da demanda energética acontece por dois motivos principais: o aumento da população mundial e do consumo energético per capita. Um dos desafios no contexto de distribuição de energia está relacionado a regulação dos níveis de tensão das barras de transmissão no sistema de distribuição de energia. Para correção de comportamentos indesejáveis (e.g., flutuações nos níveis de tensão), reatores naturalmente saturados (RNSs) são empregados na compensação de reativos ajustando os níveis de tensão do sistema elétrico de potência. Além da utilização de protótipos para validação de cenários do sistema elétrico, modelos computacionais de RNSs podem facilitar a modelagem de reatores utilizando ferramentas de softwares. Tais modelos auxiliam o projeto destes equipamentos e diminuem significativamente seu custo de prototipação. Este trabalho apresenta um esquema de modelagem e uma biblioteca de simulação de RNSs utilizando modelos de circuito magnético unificado (UMEC) na linguagem MODELS do software *Alternative Transient Program* (ATP). A biblioteca implementa características comuns a diferentes tipos de RNSs tornando simples a criação de novos modelos sem a necessidade de reescrita de todo o código para cada componente. Dessa forma, engenheiros podem simular o comportamento de RNSs com diferentes configurações de uma maneira simples e efetiva. Neste trabalho, diversos estudos de caso são mostrados com o intuito de se apresentar a viabilidade da solução apresentada.

Palavras-chave: Reatores Naturalmente Saturados, Linguagem MODELS, Programa ATP, Circuito Equivalente Magnético Unificado (UMEC).

Abstract

Currently, the energetic demand increase occurs due to two main reasons: the rise of world population and expansion of per-capita consumption. A key challenge in the context of power distribution is related to regulating the voltage levels of transmission bars in power distribution systems. In order to fix undesirable behavior (e.g., fluctuations in voltage levels), naturally saturated reactors (RNSs) are employed in the compensation of reactive power and adjusting the electrical system voltage levels. The complexity and high cost of creating actual prototypes for validating electrical system scenarios requires the creation of computer models of RNSs. Such models help the design of equipment and significantly reduce their cost of prototyping. This paper presents a modeling scheme and an RNS simulation library using Unified Magnetic Equivalent Circuit (UMEC) in MODELS language of Alternative Transient Program (ATP) software. The library implements common features for different types of RNSs facilitating the creation of new RNSs without the need write the all the source code from scratch for each component. Thus, engineers can simulate the behavior of RNSS with different configurations in a more simple and effective way. In this work, several case studies are shown to demonstrate the viability of the proposed solution.

Keywords: Naturally saturated reactors, MODELS language, ATP program, Unified Magnetic Equivalent Circuit (UMEC).

Lista de ilustrações

Figura 1 – Característica de saturação intrínseca ao material magnético.	15
Figura 2 – Arquivo PlotXY	20
Figura 3 – Janela principal	21
Figura 4 – Menu suspenso de componentes do programa ATPdraw	22
Figura 5 – Circuito básico no ATPdraw	23
Figura 6 – Visualizador de resultados do ATPdraw	23
Figura 7 – Editor de texto do ATPdraw	24
Figura 8 – Circuito equivalente de transformador de dois enrolamentos	27
Figura 9 – Transformador equivalente depois da discretização	29
Figura 10 – Transformador não linear	30
Figura 11 – caminhos de fluxos magnéticos no núcleo	31
Figura 12 – Circuito equivalente (UMEC)	31
Figura 13 – Equivalente de Norton do modelo UMEC	32
Figura 14 – a) caminhos de fluxos magnéticos no núcleo. b) circuito equivalente (UMEC)	35
Figura 15 – Curva do indutor não linear $\lambda \times I$	36
Figura 16 – Circuito simulador	36
Figura 17 – Algoritmo da Curva $B \times H$ utilizando método de aproximação linear por partes	38
Figura 18 – Algoritmo do cálculo da corrente do UMEC para RNS	41
Figura 19 – Caminhos de fluxos magnéticos no núcleo.	42
Figura 20 – Circuito equivalente (UMEC)	42
Figura 21 – Caminhos de fluxos magnéticos no núcleo.	44
Figura 22 – Circuito UMEC do reator com dois enrolamentos	44
Figura 23 – Caminhos de fluxos magnéticos no núcleo.	47
Figura 24 – Circuito UMEC do reator com 6 enrolamentos.	47
Figura 25 – Caminhos de fluxos magnéticos no núcleo.	51
Figura 26 – Circuito UMEC do reator de 9 enrolamentos.	52
Figura 27 – Caminhos de fluxos magnéticos no núcleo.	55
Figura 28 – Circuito UMEC do reator em monobloco com 12 enrolamentos.	55
Figura 29 – Biblioteca RLib.	75
Figura 30 – Fluxo de chamadas da biblioteca RLib.	76
Figura 31 – Integração entre a biblioteca RLib e diferentes implementações de reatores naturalmente saturados.	77
Figura 32 – Dados de Entrada do RNS1.	78
Figura 33 – Curva BxH do reator monofásico saturado sem perdas.	79
Figura 34 – Corrente de saída do reator monofásico saturado sem perdas.	79
Figura 35 – Fluxos magnéticos do reator monofásico saturado sem perdas.	79

Figura 36 – Circuito indutor não linear no ATPdraw.	80
Figura 37 – Corrente de saída do modelo <i>UMEC</i> do reator monofásico sem perdas X Corrente de saída representado pelo Indutor não linear do <i>TYPE98</i>	80
Figura 38 – Circuito indutor não linear do ATPdraw com perdas.	81
Figura 39 – Fluxos dispersos do reator com perdas e Fluxos dispersos do reator sem perdas.	81
Figura 40 – Correntes de reator com perdas e Correntes de reator sem perdas.	81
Figura 41 – Dados de Entrada do RNS2.	82
Figura 42 – parâmetros de testes utilizados no sistema.	83
Figura 43 – Reator monofásico em testes.	83
Figura 44 – Circuito do transformador saturável do atpdraw.	83
Figura 45 – Comparativo de correntes primárias dos ensaios de curto-circuito do RNS e do transformador saturável do atpdraw.	84
Figura 46 – Comparativo de correntes secundárias dos ensaios de curto-circuito do RNS e do transformador saturável do atpdraw.	84
Figura 47 – Comparativo de correntes primárias dos ensaios de plena carga do RNS e do transformador saturável do atpdraw.	84
Figura 48 – Comparativo de correntes psecundárias dos ensaios de plena carga do RNS e do transformador saturável do atpdraw.	85
Figura 49 – Comparativo de correntes primárias dos ensaios em vazio do RNS e do transformador saturável do atpdraw.	85
Figura 50 – Comparativo de correntes secundárias dos ensaios em vazio do RNS e do transformador saturável do atpdraw.	85
Figura 51 – Parâmetros de entrada do reator de 6 enrolamentos.	86
Figura 52 – Corrente trifásica nos enrolamentos primário e secundário do reator trifásico.	86
Figura 53 – Tensão trifásica nos enrolamentos primário e secundário do reator trifásico.	87
Figura 54 – Corrente na fase A no enrolamento primário do RNS3YD.	87
Figura 55 – Laço de histerese do RNS3YD.	88
Figura 56 – transformador saturável do atpdraw.	88
Figura 57 – Corrente de fase A dos RNS3YD X Transformador saturável do ATPdraw. . .	88
Figura 58 – Parâmetros de entrada do reator RNS9.	89
Figura 59 – Corrente de fase A dos reatores trifásicos RNS9 e o RNS3YD.	89
Figura 60 – Reator com ligação paralela dos enrolamentos.	90
Figura 61 – Corrente da fase A do RNS em Monobloco.	91
Figura 62 – Tensões nos enrolamentos de entrada do RNS em Monobloco.	91

Lista de abreviaturas e siglas

ATP	Alternative Transient Program
RNS	Reator Naturalmente Saturado
UMEC	Circuito Equivalente Magnético Unificado
Rlib	Nome da Biblioteca
EMTP-ATP	Electromagnetic Transients Program
ATPDraw	Pré-processador Gráfico
TACS	Total Access Communication System
MODELS	Linguagem de Programação
RlibAux	Módulos Auxiliares da Biblioteca
RNSMatrix	Módulo Principal para Operação de Matriz
RNS1	Reator Monofásico com um Enrolamento
RNS2	Reator Monofásico com dois Enrolamentos
RNS3YD	Reator Trifásico com seis Enrolamentos
RNS9	Reator Trifásico com nove Enrolamentos

Lista de símbolos

V_s	Tensão na Fonte
B	Campo Magnético
H	Densidade Magnética
I	Corrente do Material Magnético
I_ϕ	Corrente de Saturação
λ	Fluxo Magnético
V_n	Tensão Nominal
I_n	Tensão Nominal
V_1	Tensão no Enrolamento Primário
V_2	Tensão no Enrolamento Secundário
L_1	Indutância no Enrolamento Primário
L_2	Indutância no Enrolamento Secundário
$L_{disperso}$	Indutância Dispersa
ϕ	Fluxo Magnético
Y	Admitância
P	Permeância
I_{ns}	Corrente da fonte de Corrente
P_d	Permeância Dispersa
P_r	Permeância de Retorno
N	Número de Espiras

Sumário

1	INTRODUÇÃO	14
1.1	Considerações Iniciais	14
1.2	Objetivos	15
1.3	Trabalhos Relacionados	16
1.4	Apresentação do trabalho	18
2	REFERENCIAL TEÓRICO	19
2.1	ATP - Alternative Transient Program	19
2.1.1	ATPDraw	20
2.1.2	MODELS	22
2.2	Modelo Básico de Transformadores	27
2.2.1	Modelagem de transformadores não lineares	29
2.3	Circuito Equivalente Magnético Unificado (UMEC)	30
2.3.1	Análise de transformador UMEC	31
2.4	Aproximação linear por partes	35
3	MODELAGEM DE REATORES NATURALMENTE SATURADOS UTILIZANDO O UMEC NO ATP	39
3.1	Modelo UMEC para reator monofásico naturalmente saturado com 1 enrolamento	40
3.2	Modelo UMEC para reator monofásico naturalmente saturado com 2 enrolamentos	44
3.3	Modelo UMEC para reator trifásico naturalmente saturado com 6 enrolamentos e dois retornos	47
3.4	Modelo UMEC para reator trifásico naturalmente saturado com 9 enrolamentos e dois retornos	51
3.5	Modelo UMEC para reator trifásico com configuração em monobloco naturalmente saturado com 12 enrolamentos e dois retornos	55
4	BIBLIOTECA RLIB	60
4.1	Considerações Importantes Sobre a Linguagem Models	60
4.1.1	Implementação de subrotinas.	60
4.1.2	Representação de matrizes retangulares.	63
4.2	Visão Geral da Arquitetura da Biblioteca RLib	65
4.2.1	Pacote para operações sobre matrizes - RLibAux	65

4.2.2	Comportamento da Biblioteca RLIB	71
4.2.3	Integração de Modelos de RNS com a Biblioteca RLIB	74
5	ESTUDOS DE CASO	78
5.1	RNS monofásico com um enrolamento	78
5.2	RNS monofásico com dois enrolamentos	82
5.3	RNS trifásico com seis enrolamentos	86
5.4	RNS trifásico com nove enrolamentos	87
5.5	RNS trifásico em Monobloco com Doze Enrolamentos	89
6	CONCLUSÃO	92
6.1	Sugestões para trabalhos futuros	93
	REFERÊNCIAS	94
	CÓDIGO FONTE DA BIBLIOTECA RLIB	96
	Código Fonte do Modelo para Inversão de Matrizes	96
	Código Fonte da Biblioteca RLib	98
	Código de Reatores Naturalmente Saturados	110

1 Introdução

Este capítulo apresenta uma breve introdução do trabalho proposto. Entre os conceitos apresentados temos uma apresentação da motivação do trabalho, o objetivo do mesmo e os trabalhos relacionados.

1.1 Considerações Iniciais

O crescimento da demanda de energia nos sistemas elétricos de potência tem exigido das concessionárias a implantação de novos recursos de transmissão para atendimento da demanda de forma satisfatória. Uma particularidade dos longos sistemas de transmissão de alta tensão é a precária regulação oferecida nas barras de consumo. Na operação em plena carga, ocorre uma redução, por vezes, extrema da tensão. Por outro lado, o sistema na operação em vazio ou com pouca carga, e devido ao grande comprimento da linha e a capacitância associada apresenta elevações na tensão terminal, muitas vezes, intolerável.

Para inibir este comportamento indesejado, medidas corretivas devem ser utilizadas para restaurar os padrões exigidos e, neste campo, a regulação de tensão é um tema de grande relevância. Uma alternativa comumente empregada pelas empresas de energia é a compensação de reativos, em que utilizam reatores e capacitores conectados em derivação no terminal da carga, permitindo um controle de tensão local (FONTE, 1997).

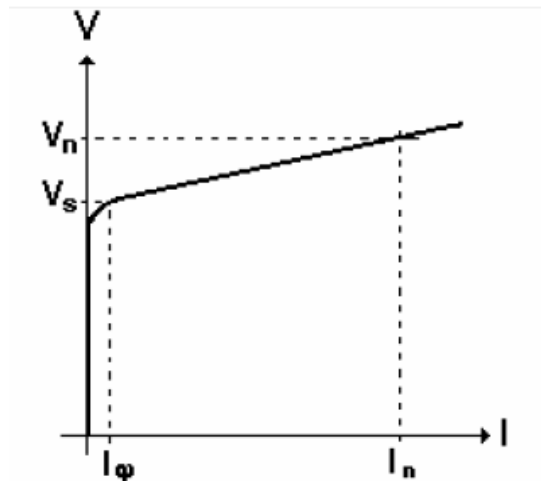
A atuação dos reatores é de proibir que a tensão se eleve a valores inadmissíveis, entretanto, os capacitores atuam de forma a inibir a tensão de baixar além do permitido. Mesmo sendo uma prática comum este tipo de regulação de tensão, apresenta desvantagens, tais como:

- A regulação de tensão por meio desse procedimento utiliza um bloco desses elementos por vez.
- Sempre que o carregamento do sistema experimentar mudanças consideráveis, este tipo de compensação terá a necessidade de executar manobras de injeção ou remoção de equipamentos. Assim, quando a carga suprida pela linha sofrer uma elevação, os reatores deverão ser gradativamente retirados de operação para conter a redução da tensão, enquanto os capacitores devem seguir uma manobra similar quando houver redução do consumo energético no sistema.

Para resolver os inconvenientes mostrados anteriormente, uma solução viável corresponde ao uso de compensadores eletrônicos (FONTE, 1997). Embora tecnicamente atrativos, os elevados custos de aquisição e manutenção desses equipamentos se mostra uma solução de difícil aceitação.

Uma outra alternativa para a compensação de reativos, seria a utilização de Reatores Naturalmente Saturados (RNSs) que são mais baratos, possuem manutenção de baixo custo, elevada confiabilidade e alta resistência a intempéries (BRASIL, 1996). Seu princípio de funcionamento é mostrado na Figura 1 que apresenta as características de tensão (V) e corrente (I) do RNS.

Figura 1 – Característica de saturação intrínseca ao material magnético.



Fonte: (ARAUJO; NEVES, 2005)

Quando o reator está submetido a condição de plena carga, ele funcionará na vizinhança do ponto (V_s, I_ϕ) , demandando, dessa forma, uma potência bem pequena do sistema. Porém, quando a carga do sistema diminui, a tensão experimenta uma elevação correspondente, o que desloca o ponto de operação do reator para as proximidades do ponto (V_n, I_n) . Isso eleva o seu consumo numa proporção ditada pela inclinação da sua curva característica, a qual, em princípio, poderá ser especificada para satisfazer o nível de compensação desejada. Segundo (MAIA et al., 2012), tal comportamento, além de ensejar um controle contínuo da tensão dentro de patamares determinados, disponibiliza a capacidade da linha de transmissão para o atendimento das cargas.

1.2 Objetivos

Os principais equipamentos que são utilizados para regulação de tensão, compensadores eletrônicos e reatores saturados, possuem complexidade em fazer simulações computacionais para sua representação. Um dos motivos deve-se a sua natureza não linear. Uma ferramenta computacional muito utilizada para simulações de transitórios no sistema elétrico é a ferramenta *Alternative Transient Program* (ATP). Porém, para representar modelos de reatores naturalmente saturados no ATP, usuários necessitam realizar uma modelagem indireta por meio de outros componentes, tais como, o indutor não linear para representar reatores monofásicos ou transformador saturável para representar reatores trifásicos.

Buscando uma alternativa para representar reatores saturados na ferramenta ATP. Este trabalho tem por finalidade a criação de uma biblioteca construída na linguagem MODELS do programa ATP que represente modelos de reatores a núcleo saturados, que suporte as características magnéticas, quantidades de enrolamentos e diferentes quantidades de núcleos. A linguagem MODELS é uma ferramenta de simulação de componentes criados pelo próprio usuário. Este tipo de linguagem gera bastante interesse nos usuários pois, proporciona uma flexibilidade de utilização de uma linguagem de programação de alto nível sem precisar interagir de forma direta com o código fonte da ferramenta ATP. Assim, usuários ficam livres para criar bibliotecas de modelos e sub modelos em função de cada uma das aplicações.

Este trabalho é uma extensão de (DELFIN, 2014) que propôs uma modelagem de reatores naturalmente saturados utilizando circuito UMEC na linguagem MODELS no ATP. Apesar do pioneirismo do trabalho desenvolvido pela autora, os modelos apresentados possuem características que dificultam a reutilização e entendimento dos códigos propostos. Dessa forma, a criação e simulação de um novo modelo de RNS, tomando como base os modelos apresentados no trabalho, representa uma tarefa de difícil execução.

Por conta dessa falta de padronização e utilização de subrotinas, os códigos empregados se tornaram extensos, de difícil legibilidade e manutenibilidade. Para contornar as questões destacadas acima, uma biblioteca é proposta para facilitar a implementação de reatores a núcleo saturados de diferentes configurações. Esta biblioteca permite a criação de modelos de RNS na linguagem MODELS de forma simplificada.

Especificamente os objetivos deste trabalho são:

- Desenvolvimento de uma biblioteca responsável para facilitar a criação de modelos de reatores naturalmente saturados na linguagem MODELS do software ATP.
- Desenvolvimento de múltiplas configurações de reatores á núcleo saturado, variando-se o número de enrolamentos e o arranjo utilizado.
- Simulações dos modelos construídos.
- Comparações dos resultados obtidos nas simulações com componentes similares utilizados no programa ATP.

1.3 Trabalhos Relacionados

Nos últimos anos, alguns autores têm dedicado esforços para estudar questões sobre reatores saturados. Em (BRASIL, 1996) foi desenvolvido um modelo de reator saturado no programa de transitórios eletromagnéticos EMTP-ATP (versão anterior do ATP), que verifica a viabilidade da utilização do mesmo em simulações com regime transitórios e permanentes. Na

verdade, a diferença entre o EMTP e o ATP se dá mais pela interface mais amigável da segunda. Logo, as duas versões realizam os mesmos tipos de simulações.

Magnata (FONTE, 1997) propôs um aprofundamento teórico relacionado a reatores saturados estudando seu comportamento, tipos de ligações e como atenuar os harmônicos gerados pelos reatores.

Em (FONTE, 2004), o autor descreve o modelo matemático dos reatores saturados, projeta e constrói protótipos de reatores com saturação natural em laboratório acadêmico. Um dos focos do trabalho está na análise do desempenho dos protótipos e conferência dos seus resultados com estudos realizados anteriormente pelo autor.

(BARBOSA, 2009) analisa o comportamento de compensadores estáticos utilizando reatores saturados. Os estudos realizados evidenciaram que nas condições impostas, o tempo de resposta do reator se mostrou compatível com as referências encontradas na literatura. O reator saturado demonstrou claramente sua propriedade de regulação de tensão, bem como apresentou geração de harmônicos quando na condição de saturação.

O trabalho apresentado em (PACHECO et al., 2007) tem por finalidade projetar e analisar o desempenho de um reator saturado de 570 kVAr e 13,8 kV utilizado para regulação de tensão. Quando comparados com outros equipamentos, segundo autores, o novo produto demonstra expressivos ganhos, principalmente em relação às perdas, rendimentos, temperaturas de operação.

(BARBOSA et al., 2010) propõe uma estratégia para a modelagem computacional de reatores a núcleo saturado utilizando a plataforma ATP. Visando o estabelecimento de mecanismos para validação dos desempenhos computacionais obtidos, estes são correlacionados com medições em um protótipo de 600 kVAr e 13,8 kV. Os autores mostraram que as características operativas e representativas das correlações entre as potências reativas e correntes absorvidas pelo equipamento, diante de distintos níveis de tensões aplicadas, ratificaram o bom desempenho do modelo computacional.

A dissertação apresentada em (FREITAS, 2010) descreve arquiteturas e funcionalidades inéditas para dispositivos que empregam reatores naturalmente saturados como elemento de controle. Os resultados obtidos pelos experimentos da aplicação de reatores em série funcionando como limitador de corrente de curto circuito comprovam que este dispositivo é extremamente eficaz na contenção de curto circuitos, podendo promover variados níveis de redução.

Neves et al. (HERMANN, 1993) mostra um algoritmo para calcular as características de saturação de transformadores de núcleo de ferro, baseado nas curvas convencionais ($V \times I$) fornecidas. Um método direto para o cálculo de curvas de saturação do núcleo de ferro ($\lambda \times I$) foi apresentado e é fundamento em bases de testes de transformador.

Diferente dos anteriores, esse trabalho propõe uma modelagem para representação de reatores naturalmente saturados de diferentes configurações utilizando a modelagem UMEC. Além disso, foram desenvolvidas as respectivas implementações na linguagem MODELS do

programa ATP. Para auxiliar o processo de modelagem e avaliações das matrizes relacionada aos modelos propostos, foi desenvolvida uma biblioteca denominada Rlib que permite a criação de novas implementações de reatores a núcleo saturado.

Segundo nossos melhores conhecimentos e intenções, esta é a primeira vez que se tentam fazer uma abordagem unificada que reúne modelos do tipo UMEC para RNS, suas implementações na linguagem MODELS e uma biblioteca que permite criação de novos modelos.

1.4 Apresentação do trabalho

A estrutura do documento é dividida em sete capítulos, conforme exposto a seguir:

- Capítulo 2 - Apresenta os princípios básicos do programa ATP, relaciona alguns conceitos sobre modelos de transformadores, apresenta o modelo UMEC, explica como é realizada a inversão de matrizes na linguagem MODELS do ATP. Por fim, apresenta a teoria da linearização por trechos utilizada nesse trabalho.
- Capítulo 3 - Descreve a modelagem matemática de cinco reatores saturados. Iniciando os estudos com reatores monofásicos e finalizando com a modelagem de um reator trifásico de doze enrolamentos com configuração em monobloco.
- Capítulo 4 - Trata-se da criação de uma biblioteca na linguagem MODELS que possui módulos para cálculos envolvendo matrizes retangulares e um módulo específico para o cálculo das interações eletromagnéticas dos reatores naturalmente saturados.
- Capítulo 5 - Apresenta as simulações dos reatores saturados modelados neste trabalho, igualmente compara-se os resultados com modelos utilizados no programa ATP.
- Capítulo 6 - Consiste de algumas considerações finais acerca do trabalho, além de sugestões relevantes para a continuação de pesquisas relacionadas à modelagem de reatores saturados.

2 Referencial Teórico

2.1 ATP - Alternative Transient Program

A utilização de softwares de simulação para análises de sistemas elétricos de potência constitui um fator de grande importância na formação de engenheiros elétricos, dado que permite o estudo antecipado do comportamento de sistemas reais. Porém, deve-se considerar que softwares de simulação são uma ferramenta complementar aos estudos teóricos e na atuação prática. Assim, faz-se necessário uma correta interpretação e análise dos resultados de cada simulação.

O ATP é um dos sistemas mais utilizados para simulação digital de fenômenos transitórios eletromagnéticos, bem como de natureza eletromecânica em sistemas elétricos de potência. Com este programa, redes complexas e sistemas de estrutura arbitrária de controle podem ser simulados, e suas respectivas métricas são calculadas em tempos discretos. O ATP tem uma ampla capacidade de modelagem, além de fazer cálculos de transientes.

O aplicativo não é do domínio público, porém qualquer pessoa que não tenha o propósito de utilizá-lo para fins comerciais pode instalar e utilizar o programa sem nenhum custo. O ATP analisa variáveis de interesse em redes de energia elétrica em função do tempo, geralmente iniciadas por alguns distúrbios. Para resolver equações diferenciais no domínio do tempo, o ATP utiliza basicamente a regra de integração trapezoidal (KRAUSE; AMBLER; FOX, 1993).

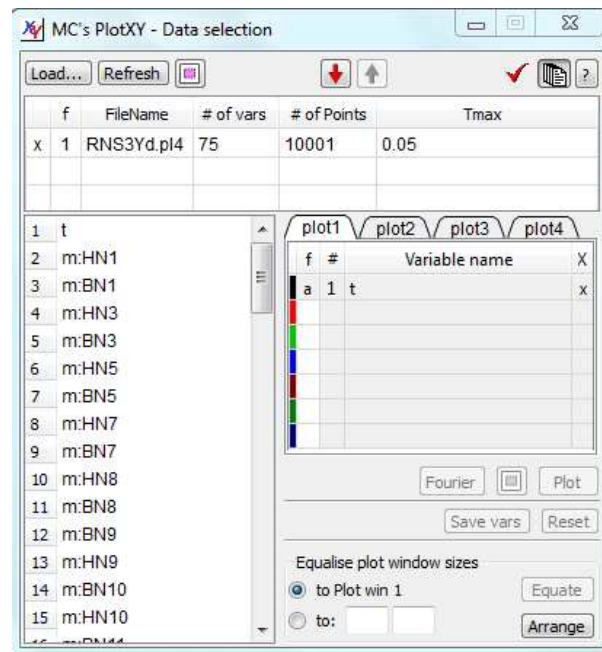
Este programa realiza análises de transitórios, frequências, harmônicos e estatísticas. A ferramenta apresenta uma biblioteca de componentes elétricos prontos para serem utilizados pelo usuários. O principais componentes da biblioteca são:

- Componentes de resistências, capacitâncias e indutâncias concentradas.
- Modelos de linhas de transmissão aéreas ou cabos.
- Componentes não-lineares: transformadores, incluindo saturação e histerese, pára-raios, indutâncias e resistências.
- Fontes ideais de corrente e de tensão - sinusoidais, exponenciais, degrau, tipo rampa.
- Máquinas rotativas: máquina síncrona trifásicas, modelo de máquina universal.
- Válvulas (diodos, tiristores, os triacs), TACS/MODELS e chaves controladas.

O arquivo de entrada de dados do software ATP geralmente possui a terminação .ATP (ou .DAT), exceto arquivos criados pelo ATPDraw (Seção 2.1.1) que possuem extensão .ADP. Os

arquivos de saída do ATP possuem a terminação .LIS para os resultados impressos e a terminação .PL4 para a visualização dos gráficos. Através de uma tabela de valores de arquivos .PL4, gráficos podem ser produzidos através do programa PlotXY (presente no software ATP) Figura 2. No PlotXY, o usuário pode escolher métricas (e.g., BN1, ou HN1) e plotar a variação dessa métrica ao longo do tempo t ou em relação a outra variável (lista de variáveis no lado esquerdo da Figura 2).

Figura 2 – Arquivo PlotXY



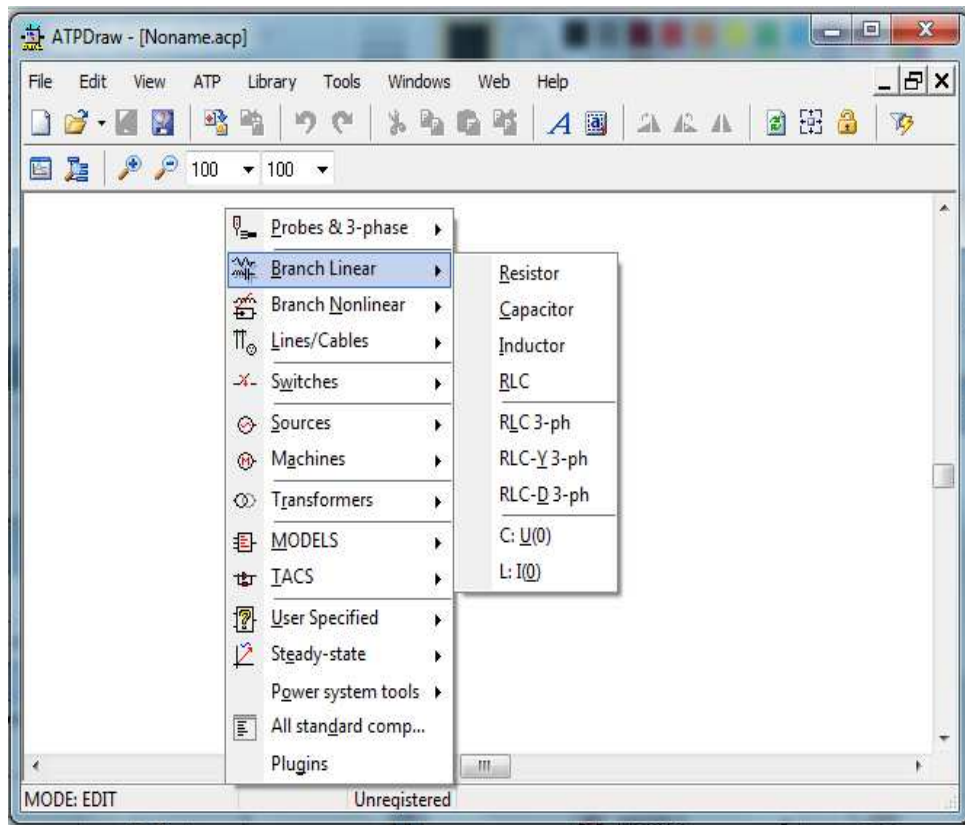
Fonte: Próprio Autor

2.1.1 ATPDraw

Como dito anteriormente, o ATP utiliza arquivos de entrada do tipo .ATP ou .DAT. Esses arquivos são textuais e não são simples de serem manipulados. Para facilitar a criação/manipulação de esquemas do ATP, a ferramenta gráfica ATPDraw (PRIKLER; HOILDALEN, 2002) foi criada. O aplicativo apresenta a maioria dos componentes elétricos que são utilizados na modelagem de circuitos de potência do ATP e foi desenvolvido para simplificar a modelagem de circuitos elétricos de potência no ATP. O circuito é armazenado no disco em um único arquivo de projeto, que inclui todos os objetos de simulação e opções necessárias para simular o sistema. Na inexistência de componentes já prontos na biblioteca, o usuário através da linguagem Models pode criar seus próprios modelos.

O ATPDraw admite sistemas em circuitos monofásicos ou trifásicos e possui as funcionalidades do sistema operacional Windows: copiar, colar, girar, importar, exportar, agrupar, desfazer, entre outros. A janela principal do programa é visualizada na Figura 3. O menu principal do ATPdraw é composto por:

Figura 3 – Janela principal



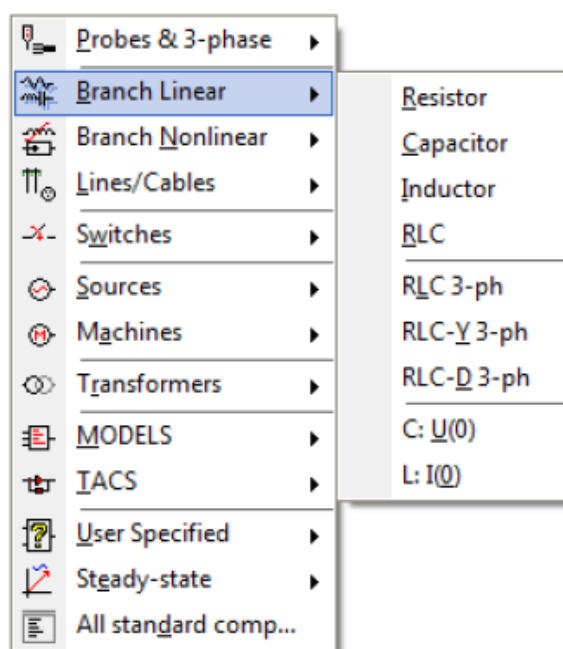
Fonte: Próprio Autor

- File - permite criar novos circuitos elétricos, salvar ou abrir um arquivos criado anteriormente, fechar arquivos, importar elementos, fechar o programa.
- Edit - contém as funções básicas de edição, por exemplo: copiar/colar, desfazer/refazer, duplicar, escolher, apagar, adicionar texto, etc.
- View - configura a visualização das janelas, zoom, tipo de letra dos componentes, atualização de mudanças no circuito, e opções para personalizar as janelas de desenho.
- ATP - executa o programa, cria nomes para todos os nós do sistema simulado, gera ou edita arquivos .ATP, especifica qual a configuração desejada para simulação do sistema criado.
- Library - permite criar e/ou modificar componentes já existentes ou aqueles que foram criados pelo usuário.
- Tools - ferramentas de edição de texto, edição de imagens e pode-se configurar a interface do ATPDraw.
- Windows - O usuário pode selecionar a janela correspondente a cada circuito e ativar ou desativar uma janela do sistema.

- Help - O usuário pode solicitar informação de ajuda do ATPDraw.

ATPDraw suporta modelagem hierárquica de componentes, substituindo um grupo de objetos por um único ícone em um número quase ilimitado de camadas. Suporta até 10.000 componentes, cada um com no máximo 64 dados e 32 nós (PRIKLER; HOILDALEN, 2002). A Figura 4 detalha alguns componentes que podem ser utilizados no programa, tais como: Componentes Standard (Probes & 3-phase), Ramais Lineares (Branch Linear), Ramais Não-Lineares (Branch Nonlinear), Linhas e Cabos (Lines/Cables), Interruptores (Switches), Fontes (Sources), Máquinas (Machines), Transformadores (Transformer), entre outros.

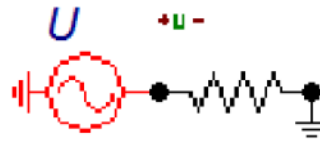
Figura 4 – Menu suspenso de componentes do programa ATPdraw



Fonte: Próprio Autor

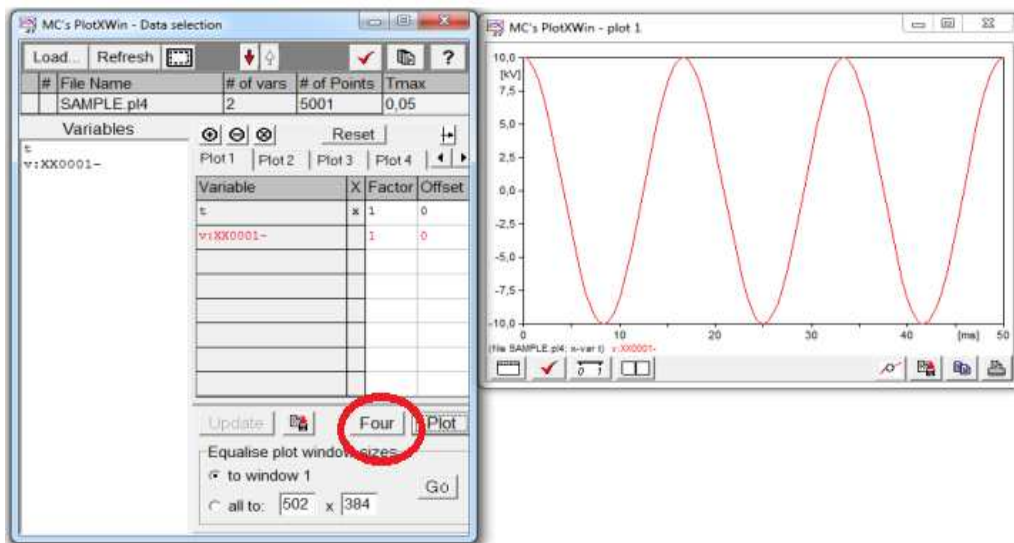
Na Figura 5, é mostrado um exemplo básico de circuito criado no ATPDraw, seus gráficos são plotados no programa PlotXY (Figura 6) que permite representar até 8 curvas no mesmo gráfico, também é possível representar na mesmo gráfico curvas com métricas de até três arquivos diferentes. As curvas são apresentadas em diferentes cores para uma melhor visualização dos dados. No programa PlotXY, também é possível analisar os harmônicos, basta acionar a opção FOUR (circulado em vermelho na Figura 6) que os gráficos de Fourier são mostrados. Por fim, a Figura 7 apresenta o código fonte respectivo ao modelo da Figura 5. Para mais detalhes relacionados as ferramentas ATP e ATPDraw o leitor é referido as referências (KRAUSE; AMBLER; FOX, 1993) e (PRIKLER; HOILDALEN, 2002).

Figura 5 – Circuito básico no ATPdraw



Fonte: Próprio Autor

Figura 6 – Visualizador de resultados do ATPdraw



Fonte: Próprio Autor

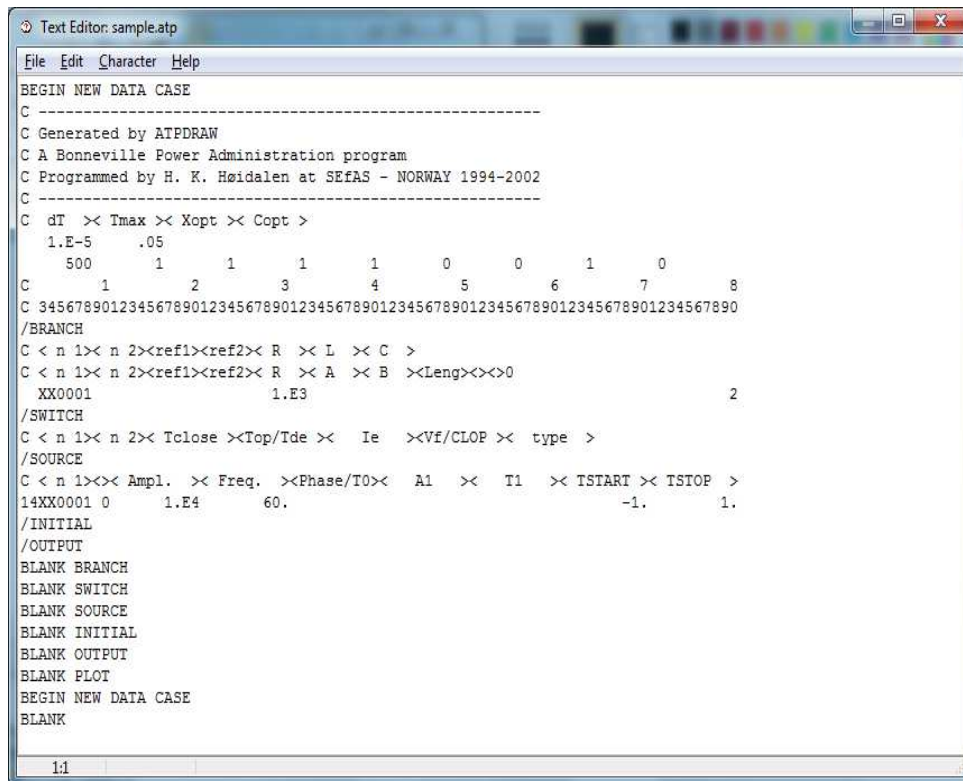
2.1.2 MODELS

A MODELS é uma linguagem de programação criada para simular sistemas no domínio do tempo e descrever comportamentos dinâmicos de sistemas físicos de alta complexidade (DUBÉ, 1996). No programa ATP, esta linguagem permite o controle da operação dos componentes elétricos do sistema simulado, tal como a criação de novos modelos. Ela tem a capacidade de interligar o ATP com programas externos.

A linguagem MODELS possui como características:

- Um extenso campo de variáveis, que podem ser numéricas, lógicas, escalares ou matriciais.
- Uma sintaxe regular para o uso de variáveis, matrizes, expressões e funções.
- A possibilidade de especificar operações que só são executadas quando estiverem reunidas algumas condições, como por exemplo quando forem detectados erros.
- A linguagem gerencia automaticamente o armazenamento e recuperação de valores históricos das variáveis de um modelo, conforme necessário durante a execução da

Figura 7 – Editor de texto do ATPdraw



```

Text Editor: sample.atp
File Edit Character Help
BEGIN NEW DATA CASE
C -----
C Generated by ATPDRAW
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEFAS - NORWAY 1994-2002
C -----
C dT >< Tmax >< Xopt >< Copt >
  1.E-5      .05
    500      1      1      1      1      0      0      1      0
C      1      2      3      4      5      6      7      8
C 34567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R >< L >< C >
C < n 1>< n 2><ref1><ref2>< R >< A >< B ><Leng><><0
  XX0001      1.E3      2
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde >< Ie ><VF/CLOP >< type >
/SOURCE
C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< ISTART >< TSTOP >
14XX0001 0      1.E4      60.      -1.      1.
/INITIAL
/OUTPUT
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
1:1

```

Fonte: Próprio Autor

simulação.

- Instâncias individuais dos modelos de um sistema automaticamente atualizam o seu funcionamento de acordo com o aumento do valor do tempo de simulação.
- Uma série de funções pré-definidas com base no tempo são disponíveis, tais como, integrais, derivadas, equações diferenciais, operadores de Laplace e funções de transferência Z.

A MODELS é uma ferramenta para o desenvolvimento de modelos de componentes de circuito e de controle que não pode ser construída facilmente com o conjunto de componentes existentes disponíveis na ATP e TACS. Ela possui a flexibilidade de uma linguagem de programação completa sem ter que interagir com ATP ao nível de programação. Proporciona uma interface de programa no ATP definido no nível de modelagem em termos de tensões, correntes e sinais de controle, em oposição a uma interface que seria definido ao nível de programação em termos de variáveis, blocos comuns, e sub-rotinas do ATP (DUBÉ, 1996).

Esta linguagem faz com que seja possível ligar programas externos com o ATP para a modelagem de componentes, o acesso a medições, ou interação com equipamentos, sem a necessidade de um conhecimento de programação da operação interna de ATP e sem necessidade

de qualquer modificação para ser feito com o código fonte de ATP (DUBÉ, 1996).

Tal como a MODELS, a linguagem de programação TACS também simula sistemas de controle utilizando algoritmos. Porém a TACS, não fornecem um mecanismo para resolver conjuntos de equações simultâneas não lineares. Qualquer loop envolvendo um dispositivo não-linear é cortado automaticamente para que possa caber na solução de matriz linear da TACS. Isso implica em atrasos de uma iteração nas saídas dos dispositivos não-lineares. Dependendo do tamanho do passo do tempo e da natureza do sistema simulado, as consequências desta abordagem podem variar a saída de forma satisfatória deixando o resultado numericamente instável, por este motivo esta linguagem entrou em desuso.

A linguagem MODELS é bastante semelhante a linguagem FORTRAN E PASCAL. Suas instruções podem ser vistas no exemplo Código Fonte 2.1:

```

1
2 MODEL proc_example
3   DATA inival {dflt:0}
4   INPUT x
5   VAR y
6
7   INIT -- this is executed when the instance is created or first USED
8     integral(x) := inival
9   ENDINIT
10
11  EXEC -- this is the default operation procedure,
12    -- executed when the instance is used from a USE statement
13    y := integral(x) -- assigns value of integral to y
14    self.write_y -- calls the procedure 'write_y'
15  ENDEXEC
16
17  PROC write_y
18    write('In model proc_example, y=', y)
19  ENDPROC
20
21  PROC reset_integral(resetval) -- uses one input argument
22    integral(x) := resetval -- resets the value of the integral
23  ENDPROC
24
25  PROC integral_value
26    integral_value := integral(x) -- returns the value of the integral
27  ENDPROC
28
29 ENDMODEL

```

Código Fonte 2.1 – Exemplo de utilização de submodelos localmente definidos

O parâmetro DATA são valores constantes de um modelo, como parâmetros e dimensões da matriz, definidos quando uma nova instância de um modelo é criada em uma instrução USE. DATA é usado para a realização de valores que são definidos no exterior do modelo, e que não

variam durante a simulação. Os nomes dos elementos de DATA são visíveis dentro do modelo, onde eles são declarados. O seu valor é atribuído no procedimento USE, uma vez atribuído, valores de dados não variam no tempo. O seu valor é constante ao longo do comprimento total de uma simulação, um valor padrão também pode ser opcionalmente atribuído dentro do modelo, na declaração. Quando existe um valor padrão para um elemento de dados, que a atribuição de elemento de um valor na instrução USE é opcional.

```

1
2 DATA n                -- uma dimensão de matriz utilizada no modelo
3 DATA freq {dflt: 60} -- uma frequência em Hz

```

Código Fonte 2.2 – Exemplo de utilização do elemento DATA

O parâmetro INPUT são entradas usadas para a realização de valores que são definidos no exterior do modelo, e que varia durante a simulação. Os nomes dos elementos INPUT são visíveis no interior do modelo onde eles são declarados, e também nas demonstrações USE do modelo. Seu valor é atribuído nas demonstrações USE. Um valor padrão também pode ser opcionalmente atribuído dentro do modelo, na declaração. Quando existe um valor padrão para um elemento de entrada, atribuir esse elemento um valor na instrução USE é opcional. Valores de entrada são automaticamente recalculados na declaração USE cada vez que a instância USE é chamado.

```

1
2 DATA n                -- numeros de terminais
3 INPUT vterm[1..n]     -- valor das tensões nos terminais
4   vref {dflt: 0}     -- valor de tensão de referência igual a zero, se não for usado.

```

Código Fonte 2.3 – Exemplo de utilização do elemento INPUT

Os nomes das variáveis VAR só são visíveis dentro do modelo em que elas são declaradas, exceto quando eles são declarados como OUTPUT. Seu valor pode ser atribuído em qualquer declaração dentro do modelo. Uma vez atribuído um valor, uma variável mantém esse valor durante o simulação, até que seja atribuído um novo valor. Continuando com o exemplo anterior, pode-se controlar as tensões de entrada para amplitude máxima para cada terminal, digamos, e o máximo para todos terminais. Definindo as variáveis vtmax e vmax:

```

1
2 DATA n                -- numeros de terminais
3 INPUT vterm[1..n]     -- valor das tensões nos terminais
4   vref {dflt: 0}     -- valor de tensão de referência igual a zero, se não for usado.
5 VAR vtmax[1..n]      -- Valor maximo absoluto de cada tensão terminal
6   vmax                -- Valor maximo absoluto de todas as tensões terminais

```

Código Fonte 2.4 – Exemplo de utilização do elemento VAR

As variáveis podem ser declaradas como saída de um modelo. Os nomes das variáveis declarados como OUTPUT são visíveis para os parâmetros USE que definem os casos do modelo.

Nenhuma outra variável é visível fora do modelo. Continuando o exemplo, os valores de `vtmax` e `vmax` podem ser especificada como a saída do modelo, como mostrado a seguir:

```

1
2 DATA n                -- numeros de terminais
3 INPUT vterm[1..n]    -- valor das tensões nos terminais
4     vref {dflt: 0}    -- valor de tensão de referência igual a zero, se não for usado.
5 VAR vtmax[1..n]     -- Valor maximo absoluto de cada tensão terminal
6     vmax              -- Valor maximo absoluto de todas as tensões terminais
7 OUTPUT vtmax[1..n], vmax  -- pode ser usado como saída do modelo

```

Código Fonte 2.5 – Exemplo de utilização do elemento OUTPUT

A instrução `EXEC` é o procedimento operacional principal do modelo, chamado quando é necessário atualizar uma instância do modelo para um novo tempo de simulação. Ele é chamado automaticamente cada vez que uma instrução `USE` é executado. Descreve como o modelo funciona como incremento do tempo. Finalizando a execução com a instrução `ENDEXEC`.

```

1
2 INPUT voltage, current
3 VAR flux, energy, power, gen_1
4 INIT
5     integral(voltage) := 0
6     integral(power)   := 0
7 ENDINIT
8 EXEC
9     flux := integral da tensão
10    energy := integral da potência
11    power := voltage * current
12 ENDEXEC
13 ENDMODEL

```

Código Fonte 2.6 – Exemplo de utilização do elemento EXEC

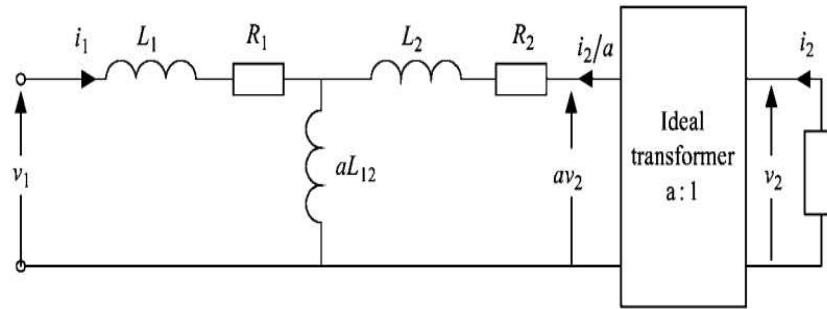
Para mais detalhes sobre a linguagem `MODELS`, o leitor é referido a (DUBÉ, 1996).

2.2 Modelo Básico de Transformadores

O circuito equivalente básico de um modelo de transformador é mostrado na Figura 8. As tensões através destas bobinas é expressa na Equação 1.

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \quad (1)$$

Figura 8 – Circuito equivalente de transformador de dois enrolamentos



Fonte: (SWIFT, 1971)

Descrevendo a Equação 1, L_{11} e L_{22} representam as indutâncias próprias dos enrolamentos primário e secundário do transformador. L_{12} e L_{21} são as indutâncias mutuas entre os enrolamentos. As correntes que passam nos enrolamentos primários i_1 e secundários i_2 são expressadas na Equação 2.

$$\frac{d}{dt} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \frac{1}{L_{12}L_{21} - L_{11}L_{22}} \begin{bmatrix} -L_{22} & L_{12} \\ L_{21} & -L_{11} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (2)$$

Como o acoplamento mútuo é bilateral, L_{12} e L_{21} são idênticos. Usando a relação de transformação $a = v_1/v_2$ da Equação 1, chega-se a:

$$L_1 = L_{11} - aL_{12} \quad (3)$$

$$L_2 = a^2L_{22} - aL_{12} \quad (4)$$

Separando as Equações 3 e 4 de acordo com seus componentes, temos:

$$\frac{di_1}{dt} = \frac{L_{22}}{L_{11}L_{22} - L_{12}L_{21}}v_1 - \frac{L_{21}}{L_{11}L_{22} - L_{12}L_{21}}v_2 \quad (5)$$

$$\frac{di_2}{dt} = \frac{-L_{12}}{L_{11}L_{22} - L_{12}L_{21}}v_1 + \frac{L_{11}}{L_{11}L_{22} - L_{12}L_{21}}v_2 \quad (6)$$

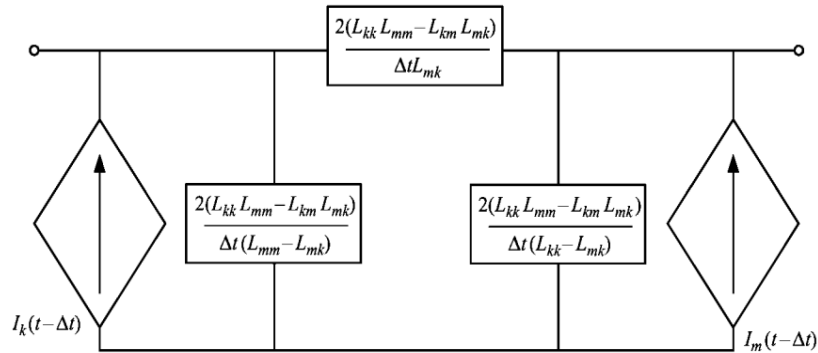
Resolvendo a Equação 5 utilizando a regra de integração trapezoidal (COGO; OLIVEIRA, 1983), têm - se que a corrente que passa no enrolamento primário é:

$$i_1(t) = \frac{L_{22}}{L_{11}L_{22} - L_{12}L_{21}} \int_0^t v_1 dt - \frac{L_{21}}{L_{11}L_{22} - L_{12}L_{21}} \int_0^t v_2 dt \quad (7)$$

$$\begin{aligned}
 &= i_1(t - \Delta t) + \frac{L_{22}}{L_{11}L_{22} - L_{12}L_{21}} \int_{t-\Delta t}^t v_1 dt - \frac{L_{21}}{L_{11}L_{22} - L_{12}L_{21}} \int_{t-\Delta t}^t v_2 dt \\
 &= i_1(t - \Delta t) + \frac{L_{22}\Delta t}{2(L_{11}L_{22} - L_{12}L_{21})} (v_1(t - \Delta t) + v_1(t)) - \frac{L_{21}\Delta t}{2(L_{11}L_{22} - L_{12}L_{21})} (v_2(t - \Delta t) + v_2(t))
 \end{aligned}$$

De forma similar, pode-se estender os conceitos para determinar a corrente $i_2(t)$. A modelagem dessas equações pode ser vista na Figura 9. Deve-se observar que a discretização desses modelos usando a regra trapezoidal não permite completa isolamento entre os terminais da fonte. Se a fonte d.c é aplicada no enrolamento primário uma pequena parte segue para o enrolamento secundário.

Figura 9 – Transformador equivalente depois da discretização



Fonte: (WATSON JOS ARRILLAGA, 2007)

Através de testes de curto-circuito e circuito aberto as informações de aL_{12}, L_1 e L_2 são adquiridas. Em programas de simulações esses testes são realizados internamente e o usuário entra apenas com os valores de reatância de dispersão e magnetizante. A matriz de indutância fornece informações sobre as correntes magnetizantes e também, a indutância de dispersão.

A indutância de dispersão é dada pela Equação 8:

$$L_{disperso} = L_{11} - L_{21}^2 / L_{22} \quad (8)$$

Os valores da matriz de indutância devem ser especificados com muita precisão para reduzir os erros causados pela subtração de dois números.

2.2.1 Modelagem de transformadores não lineares

Os efeitos da não linearidade magnética e das perdas nos núcleos dos transformadores são representados pela fonte de corrente *shunt* e resistência equivalente no enrolamento. A

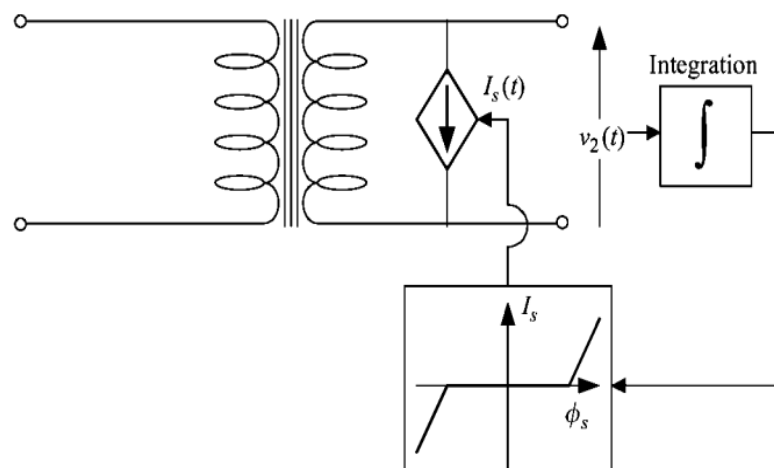
aproximação monofásica não incorpora acoplamento magnético entre as fases e a injeção de corrente magnetizante é calculada em cada passo de tempo independente das outras fases.

Para evitar uma retriangulação da matriz indutância (HERMANN, 1993) toda vez que as indutâncias mudarem, a representação da fonte de corrente é utilizada. No momento da inicialização é desejável inibir a saturação, através de um limite de fluxo derivado da integração da tensão, permitindo assim, que o estado estacionário chegue rapidamente. Depois, este limite é retirado permitindo que o fluxo vá para a região de saturação. A Figura 10 mostra a saturação no enrolamento mutuamente acoplado.

O programa de simulação EMTP utiliza o princípio da dualidade para obter informações da corrente de magnetização e configurações dos núcleos de transformadores de múltiplas pernas. Porém, este método envolve grande quantidades de componentes, tornando sua implementação complexa.

Para reduzir a complexidade exigida pelo método da dualidade um modelo baseado no equivalente de Norton, é derivado diretamente de análises de circuitos magnéticos. O chamado modelo UMEC(circuito equivalente magnético unificado). Esclarecendo que ambos os métodos apoiam-se diretamente na arquitetura dos circuitos magnéticos. Porém, o modelo UMEC utiliza um processamento próprio.

Figura 10 – Transformador não linear



Fonte: (WATSON JOS ARRILLAGA, 2007)

2.3 Circuito Equivalente Magnético Unificado (UMEC)

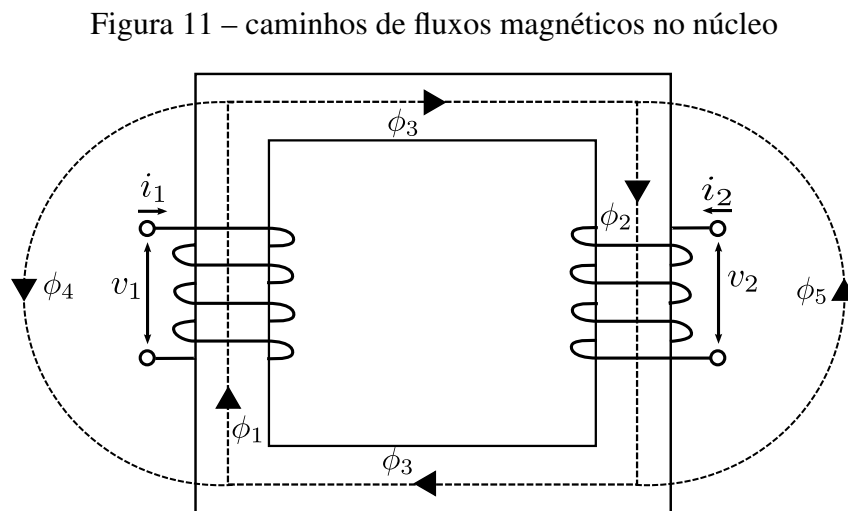
A maioria dos programas de simulações utilizam modelos de transformadores convencionais, baseados nos seus componentes(resistores e indutores). Nesses modelos seus parâmetros são obtidos de forma fácil, através de testes de curto-circuito e circuito aberto. Porém, para transformadores de múltiplos núcleos, onde o fluxo se distribui de forma desigual este modelo

não representa o transformador de forma adequada.

Diferente do modelo citado, o modelo de transformadores de Circuito Equivalente Magnético Unificado (UMEC) baseia-se na geometria do núcleo ferromagnético. Ele emprega os efeitos de acoplamento magnético entre os enrolamentos (de mesma fase e fases diferentes), e os efeitos da saturação. Dessa forma, gera uma matriz de indutância e cria um circuito equivalente mais adequado. Caracterizando melhor o comportamento do transformador em um sistema elétrico (HORITA,).

2.3.1 Análise de transformador UMEC

A Figura 11 representa o circuito magnético de um transformador monofásico de dois enrolamentos e a Figura 12 mostra o respectivo circuito equivalente magnético unificado (UMEC).



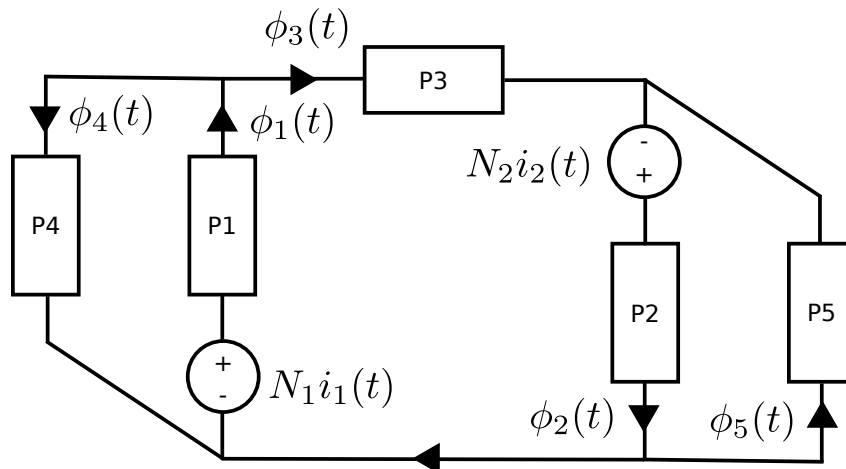
Fonte: Própria

Na Figura 11 as tensões relativas $v_1(t)$ e $v_2(t)$ dos enrolamentos primário e secundário são usadas para calcular os fluxos $\Phi_1(t)$ e $\Phi_2(t)$ em cada enrolamento.

O fluxo Φ_1 relacionado ao enrolamento primário é dividido em fluxo no jugo (Φ_3) e fluxo de dispersão (Φ_4). De forma similar, o enrolamento 2 possui o fluxo que o percorre (Φ_2), fluxo que percorre o jugo (Φ_3) e o fluxo disperso (Φ_5). Exibindo dessa forma que os fluxos não se comportam de forma uniforme nos núcleos. Os jugos superiores e inferiores são considerados iguais e possuem comprimentos L_y e área de seção transversal A_y . L_w representa o comprimento das colunas onde estão os enrolamentos e A_w a área de seção transversal.

Na Figura 12 observa-se as fontes de forças magnetomotrizes $N_1 i_1(t)$ e $N_2 i_2(t)$ que representam os enrolamentos primário e secundário. P_1 e P_2 representam as permeâncias das colunas onde se encontram os enrolamentos. P_3 representa a permeância nos jugos do

Figura 12 – Circuito equivalente (UMEC)

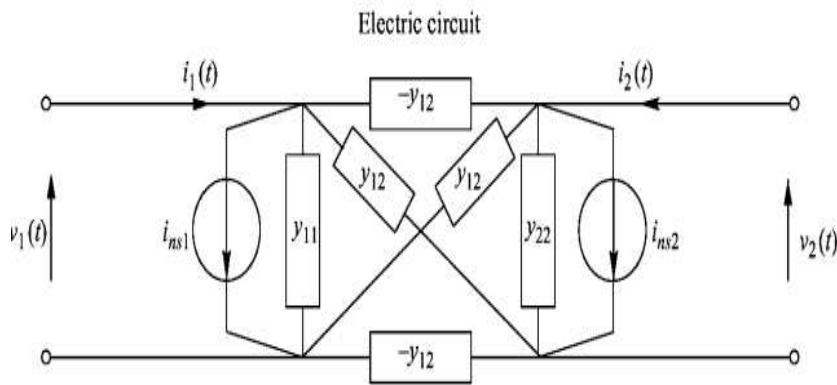


Fonte: Própria

transformador, P_4 e P_5 equivalem as permeâncias dispersas em cada enrolamento. A permeância no núcleo é calculada através da curva $B \times H$.

O circuito magnético da Figura 12 pode ser representado pelo equivalente de Norton mostrado na Figura 13 que é adequado para implementação de transientes eletromagnéticos. A força magnetomotriz em cada ramo da Figura 12 do circuito pode ser escrito em forma de vetor como na Equação 9.

Figura 13 – Equivalente de Norton do modelo UMEC



Fonte: Própria

$$\tilde{\theta} = [N]i - [R]\tilde{\phi} \tag{9}$$

A representação do fluxo em um ramo do reator na forma de vetor é demonstrado na Equação 10. Sendo $[P]$ a matriz diagonal contendo as permeâncias dos ramos e $[N]$ a matriz diagonal contendo os números de enrolamentos.

$$\tilde{\Phi} = [P]([N]i - \tilde{\theta}) \tag{10}$$

Para determinar a matriz de incidência nodal $[A]$, temos que em cada nó os fluxos devem somar zero:

$$[A]^T \tilde{\phi} = \tilde{0} \quad (11)$$

Aplicando a matriz de ligação de ramos e nós do circuito, temos:

$$[A] \tilde{\theta}_{nodal} = \tilde{\theta} \quad (12)$$

Multiplicando a Equação 10 por $[A]^T$ e substituindo nas Equações 11 e 12, temos:

$$\tilde{0} = [A]^T [P] [N] \tilde{i} - [A]^T [P] [A] \tilde{\theta}_{nodal} \quad (13)$$

Resolvendo a Equação 13 e multiplicando ambos os lados por $[A]$:

$$[A] \theta_{nodal} = ([A]^T [P] [A])^{-1} [A]^T [P] [N] i \quad (14)$$

Substituindo a Equação 13 na Equação 11 e em seguida na Equação 10, tem-se:

$$\tilde{\phi} = [M] [P] [N] \tilde{i} \quad (15)$$

onde,

$$[M] = ([I] - [P] [A] ([A]^T [P] [A])^{-1} [A]^T) [P] \quad (16)$$

A matriz $[M]$ é não singular. O ramo divide-se em dois grupos, um com a força magnetomotriz onde está o enrolamento e o outro com relutância pura, logo:

$$\begin{bmatrix} \Phi_s \\ \Phi_r \end{bmatrix} = \begin{bmatrix} [M_{ss}] & [M_{sr}] \\ [M_{rs}] & [M_{rr}] \end{bmatrix} \begin{bmatrix} [P_s] & [0] \\ [0] & [P_r] \end{bmatrix} \begin{bmatrix} [N_s] i_s \\ [0] \end{bmatrix} \quad (17)$$

O fluxo no braço do enrolamento $\tilde{\Phi}_s(t - \Delta t)$ é calculado através da corrente do enrolamento utilizando a parte superior da Equação 17:

$$\tilde{\Phi}_s = [M_{ss}^*] [P_s^*] [N_s] \tilde{i}_s \quad (18)$$

Φ_s é o vetor que contém os fluxos dos enrolamentos Φ_1 e Φ_2 . i_s é o vetor que contém a corrente nos enrolamentos i_1 e i_2 . $[P_s]$ representa o ramo da permeância atual. $[M_{ss}]$ e $[N_s]$ representam a matriz de permeância quadrática e matriz diagonal das espiras nos enrolamentos.

O fluxo nos jugos e os fluxos dispersos $\Phi_r(t - \Delta t)$ são calculados utilizando a parte inferior da Equação 17:

$$\Phi_r = [M_{rs}^*][P_s^*][N_s]i_s \quad (19)$$

Através da lei de Faraday, pode-se obter a tensão no enrolamento visto na Equação 20.

$$v = N \frac{d\Phi}{dt} \quad (20)$$

Isolando o fluxo na Equação 20 e aplicando o método da integração trapezoidal (COGO; OLIVEIRA, 1983) a Equação 20 fica:

$$\tilde{\Phi}_s(t) = \tilde{\Phi}_s(t - \Delta t) + \frac{dt}{2}[N_s]^{-1}[\tilde{v}_s(t) + \tilde{v}_s(t - \Delta t)] \quad (21)$$

A integração trapezoidal é utilizada para relacionar a tensão no enrolamento do transformador com o fluxo magnético no enrolamento. Esta derivação é adequada para programas de simulações eletromagnéticas, onde, Δt é o passo de tempo da simulação e v_s é o vetor da tensão v_1 e v_2 no enrolamento. Cada tensão no enrolamento é usada para calcular o fluxo correspondente ao braço onde está o enrolamento.

Combinando as Equações 21 e 18, obtêm-se o equivalente de Norton:

$$\tilde{i}_s(t) = [Y_{ss}^*]\tilde{v}_s(t) + i_{ns}^*(t) \quad (22)$$

onde,

$$[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1} \quad (23)$$

e

$$i_{ns}^*(t) = ([M_{ss}^*][P_s^*][N_s])^{-1} \left(\frac{dt}{2} [N_s]^{-1} \tilde{v}_s(t - \Delta t) + \tilde{\Phi}_s(t - \Delta t) \right) \quad (24)$$

logo, para o UMEC da Figura 14 a Equação 22 representa:

$$\begin{bmatrix} i_1(t) \\ i_2(t) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} i_{ns1}(t) \\ i_{ns2}(t) \end{bmatrix} \quad (25)$$

Representado pelo circuito equivalente de Norton do modelo UMEC. O circuito equivalente de Norton é ideal para simulações dinâmicas e a matriz admitância simétrica $[Y_{ss}^*]$ é não diagonal, assim inclui acoplamento mútuo. Se necessário, as perdas nos enrolamentos podem ser representados pela adição de resistências em série nos terminais do equivalente de Norton.

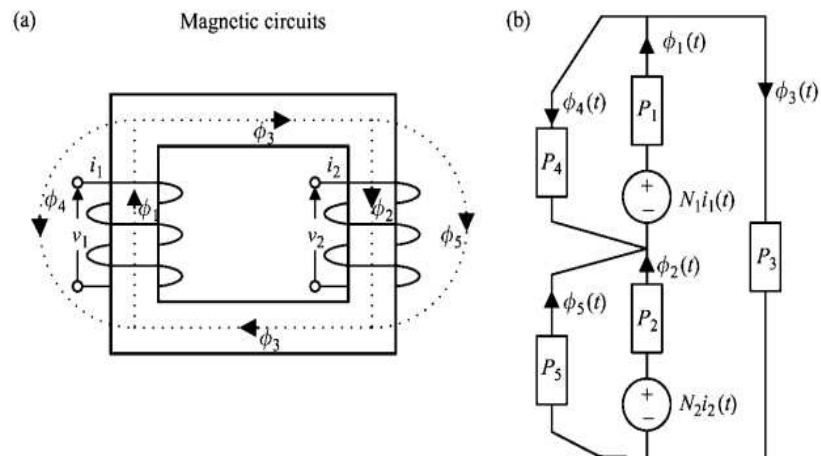


Figura 14 – a) caminhos de fluxos magnéticos no núcleo. b) circuito equivalente (UMEC)
 Fonte: Própria

2.4 Aproximação linear por partes

Constantemente, simulações de fenômenos transitórios solicitam a representação de elementos não lineares, tais como, indutâncias de núcleos de transformadores e reatores. Esses elementos são usualmente tratados com a utilização de aproximação linear por partes.

O indutor não linear, usado no programa ATP para representar reator naturalmente saturado, tem seu núcleo de forma fechada, tal como um transformador com secundário sem enrolamentos. Nesse programa a curva $\lambda \times I$ do indutor não linear (Figura 15) é produzida através de chaveamento de componentes. L_1 e L_2 mostrados na Figura 16 são inclinações do primeiro e segundo segmentos, respectivamente.

Idealmente , a chave S_1 fecha em $\lambda = \Lambda_1$ e abre quando a condição $\lambda < \Lambda_1$ for atingida. Com S_1 aberta, a indutância L_1 reproduz a região linear de 0 a Λ_1 no eixo λ ; com a chave fechada, o segmento em que $\lambda < \Lambda_1$ é reproduzido. A reprodução do segundo segmento não é trivial. Observa-se que, quando S_1 fecha, embora a tensão nos terminais de L_1 e L_2 seja a mesma, o fluxo, em cada elemento , é diferente. No momento de fechamento da chave S_1 , o fluxo em L_2 é nulo.

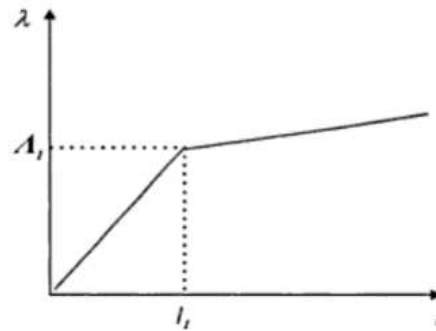
O fluxo total enlaçado na indutância não-linear é a integral da tensão entre seus terminais:

$$\lambda(t) = \int_0^t v(t)dt \tag{26}$$

Assumindo que t_a é o instante de acionamento da chave, essa equação é a soma de duas parcelas:

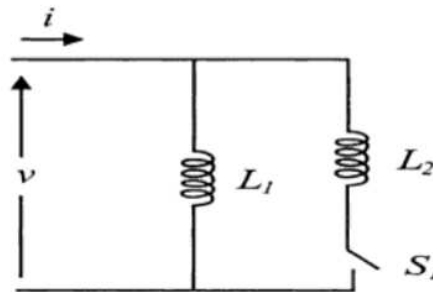
$$\lambda(t) = \int_0^{t_a} v(t)dt + \int_{t_a}^t v(t)dt \tag{27}$$

Figura 15 – Curva do indutor não linear $\lambda \times I$



Fonte: (ARAUJO; NEVES, 2005)

Figura 16 – Circuito simulador



Fonte: (ARAUJO; NEVES, 2005)

A primeira parcela corresponde a Λ_1 , a segunda ao fluxo enlaçado λ_{L_2} quando L_2 é conectado. Assim,

$$\lambda(t) = \Lambda_1 + \lambda_{L_2} \tag{28}$$

No circuito da Figura 16, com a chave fechada, a corrente i no primeiro trecho é dada por:

$$i(t) = \lambda(t)/L_1 + \lambda_{L_2}(t)/L_2 \tag{29}$$

Substituindo λ_{L_2} da equação 28 na equação 29, tem-se:

$$i(t) = \lambda(t)/L_1 + (\lambda_{L_2}(t) - \Lambda_1)/L_2 \tag{30}$$

Considerando as características $\lambda \times i$, para $\lambda > \Lambda_1$:

$$i(t) = i_1 + (\lambda(t) - \Lambda_1)/L'_2 = \Lambda(t)/L_1 + (\lambda(t) - \Lambda_1)/L'_2 \tag{31}$$

Como,

$$i(t) = \lambda(t)/L_1 + (\lambda(t) - \lambda_1)/L_2' = \lambda(t)/L_1 + (\lambda(t) - \lambda_1)/L_2 \quad (32)$$

Desse modo $L_2 = L_2' // L_1$, temos que a inclinação da curva $\lambda \times i$ no segundo segmento é:

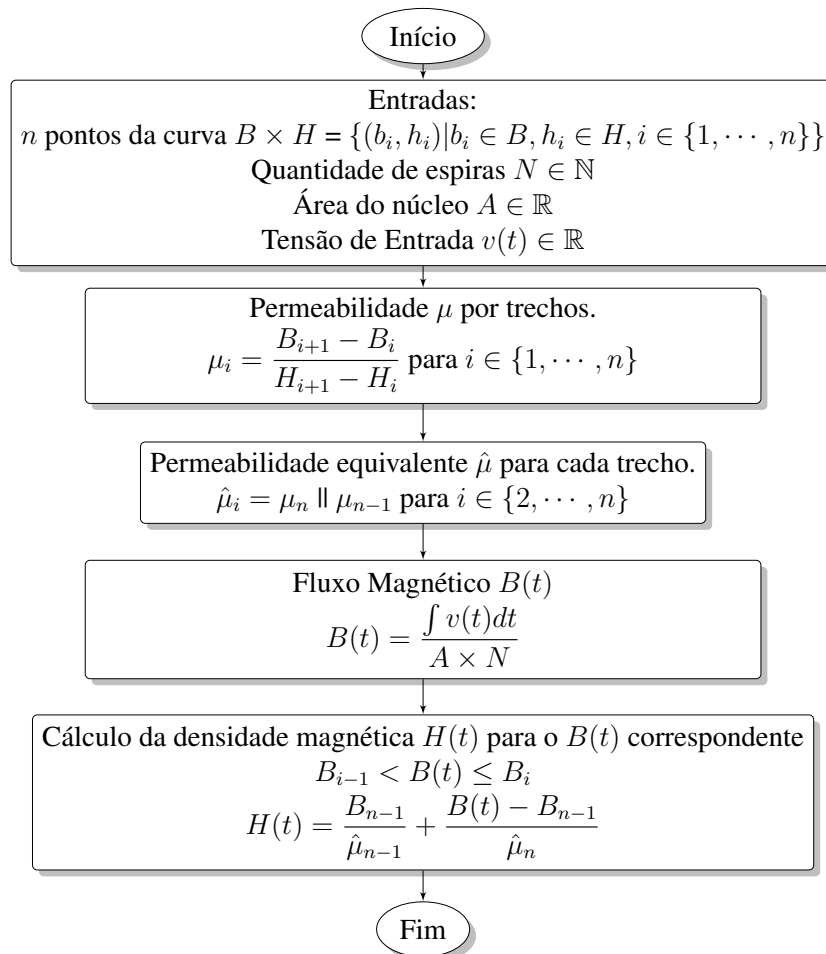
$$1/L_2' = 1/L_1 + 1/L_2 \quad (33)$$

Da mesma forma é feito para diversos segmentos da curva $\lambda \times i$.

Como demonstrado nas equações anteriores o indutor não linear é representado pelo método de aproximação linear por trechos na curva $\lambda \times i$. Através dele, simula-se RNS monofásicos no programa ATP.

Para a modelagem do UMEC usa-se parâmetros da curva $B \times H$ e seu algoritmo é proposto pelo método da aproximação linear por trechos (Figura17).

Figura 17 – Algoritmo da Curva $B \times H$ utilizando método de aproximação linear por partes



Fonte: Próprio Autor

3 Modelagem de Reatores naturalmente saturados utilizando o UMEC no ATP

Para simulações de máquinas elétricas, tais como, transformadores e reatores naturalmente saturados, é de extrema importância um bom entendimento de suas características eletromagnéticas, especificamente nos enrolamentos e nos núcleos. Neste capítulo será enfatizado as propriedades magnéticas destes equipamentos, seus comportamentos dinâmicos e os efeitos de saturação.

Primeiramente será descrito o comportamento dinâmico dos transformadores monofásicos, posteriormente os trifásicos e por fim, os efeitos de saturação. Nas primeiras simulações realizadas nos programas de transitórios eletromagnéticos foi assumido que os fluxos eram uniformes através das pernas e dos jugos no núcleo dos transformadores. Os fluxos dispersos nos enrolamentos individuais foram combinados e a corrente magnetizante foi inserida ao lado da reatância resultante série de dispersão. Um modelo para representar um transformador com vários núcleos também foi criado, baseado em circuito equivalente magnético unificado(UMEC) implementado no programa ATP.

O reator naturalmente saturado calcula a permeabilidade magnética em cada passo de tempo através da curva $B \times H$. As permeâncias do UMEC são calculadas pelas permeâncias no ramo mostrada na Equação 1, permeâncias de dispersão é demonstrada pela Equação 2 e permeâncias no retorno é calculada pela Equação 3.

$$P = \frac{\mu \times A_w}{L_w} \quad (1)$$

$$P_d = \frac{L_d}{N^2} \quad (2)$$

$$P_r = \frac{L_s}{N_1 \times N_2} \quad (3)$$

Os fluxos magnéticos dispostos no modelo UMEC é dividido em:

* Fluxo Magnético do ramo principal.

$$\phi = B \times A \quad (4)$$

* Fluxo Magnético de dispersão.

$$\phi_d = P_d(H \times L + \frac{\phi}{P}) \quad (5)$$

* Fluxo Magnético de retorno.

$$\phi_r = P_r \frac{\phi_{d1}}{P_{d1}} + \frac{\phi_{d2}}{P_{d2}} \quad (6)$$

* Fluxo Magnético no Jugo.

$$\phi_y = \phi - \phi_d \quad (7)$$

As equações que vão da Equação 1 a 7 são utilizadas para montar o algoritmo do UMEC desenvolvido na linguagem MODELS do programa ATP. Este algoritmo completo é mostrado na Figura 18.

3.1 Modelo UMEC para reator monofásico naturalmente saturado com 1 enrolamento

A modelagem deste reator foi desenvolvida com base na Figura 19 que representa o circuito magnético. E através dele, é desenvolvido o modelo UMEC (Figura 20). O modelo UMEC é constituído de nós para representar entradas e saídas de fluxos magnéticos, a fonte de força magnetomotriz $N_1 I_1$ representa cada enrolamento individualmente, as tensões dos enrolamentos são usadas para calcular os fluxos do membro onde se encontra o enrolamento.

As matrizes utilizadas para a modelagem matemática do reator em questão são mostradas a partir da Equação 8 que representa a Matriz de Incidência Nodal, finalizando com a Equação 21 que demonstra correntes nos enrolamentos.

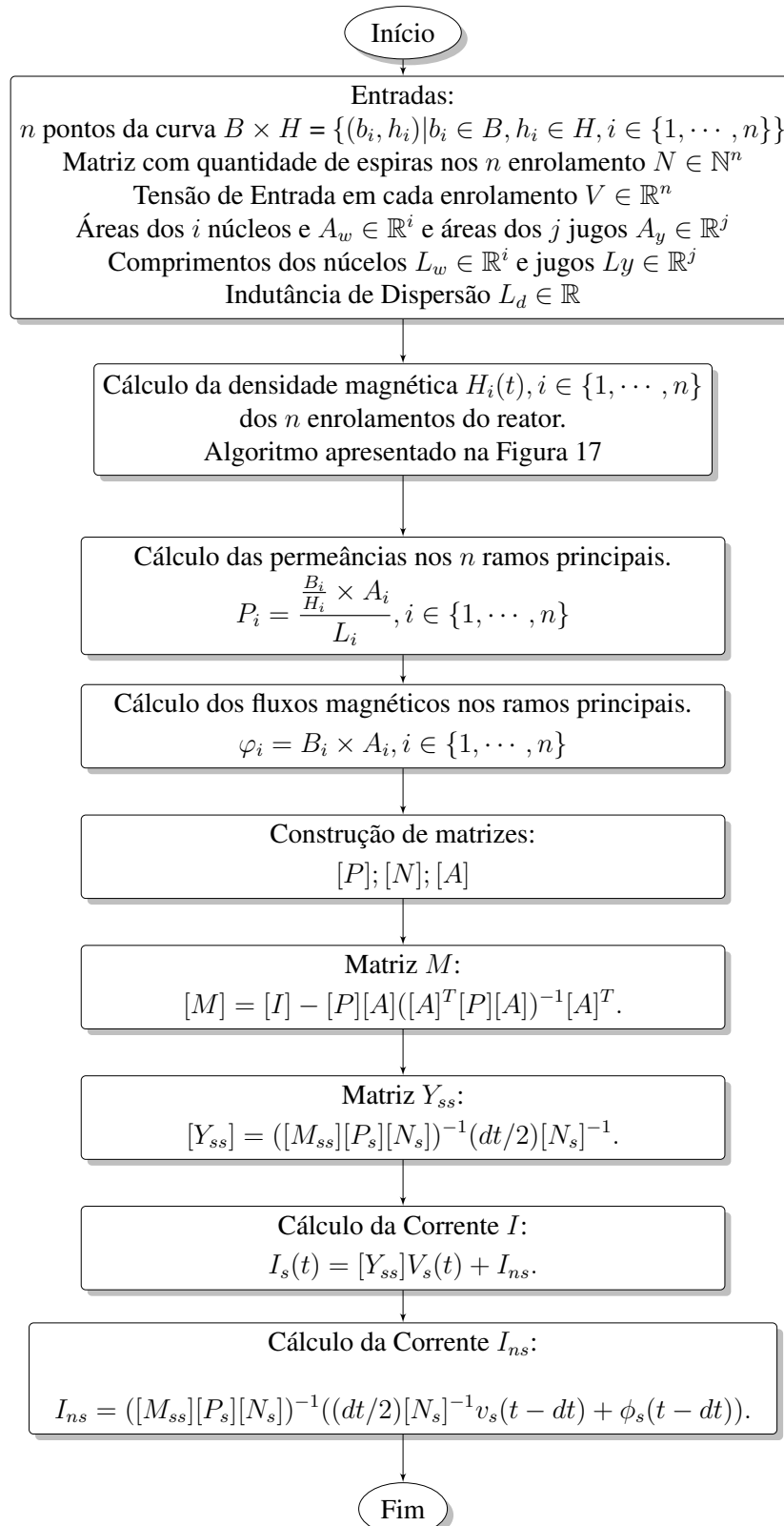
* Matriz [A] é obtida através dos nós do modelo UMEC, assim:

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \end{bmatrix} \quad (8)$$

* Matriz Permeância [P] diagonal é :

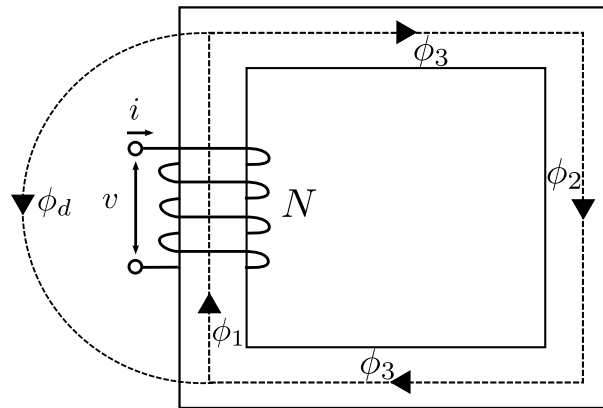
$$P = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix} \quad (9)$$

Figura 18 – Algoritmo do cálculo da corrente do UMEC para RNS



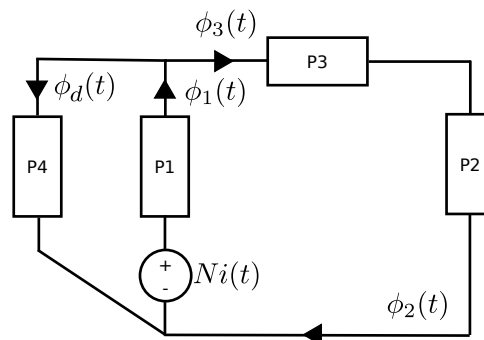
Fonte: Próprio Autor

Figura 19 – Caminhos de fluxos magnéticos no núcleo.



Fonte: Próprio Autor

Figura 20 – Circuito equivalente (UMEC)



Fonte: Próprio Autor

* Matriz diagonal das espiras [N]:

$$N = \begin{bmatrix} N_1 \end{bmatrix} \tag{10}$$

* Matriz das tensões nos enrolamentos:

$$V = \begin{bmatrix} V_1 \end{bmatrix} \tag{11}$$

* Matriz dos fluxos nos enrolamentos:

$$\Phi = \begin{bmatrix} \Phi_1 \end{bmatrix} \tag{12}$$

* Matriz identidade [I] =

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

* Matriz relacionada a fonte de corrente de cada enrolamento $[I_{ns}]$:

$$I_{ns} = [I_{ns1}] \quad (14)$$

* Matriz $[M]$ quadrada é:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \quad (15)$$

como $[M] = [I] - [P][A]([A]^T[P][A])^{-1}[A]^T$, temos:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \end{bmatrix} ([A]^T \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \end{bmatrix}) \quad (16)$$

* A submatriz $[M_{ss}]$ pode ser obtida:

$$M_{ss} = [M_{11}] \quad (17)$$

* Matriz $[P_{ss}]$ relacionada ao enrolamento 1 é :

$$P_s = [P_1] \quad (18)$$

* A matriz admitância pode ser entendida como: $[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1}$, logo:

$$Y_{ss} = [Y_{11}] \quad (19)$$

* Reescrevendo a matriz I_{ns} , temos:

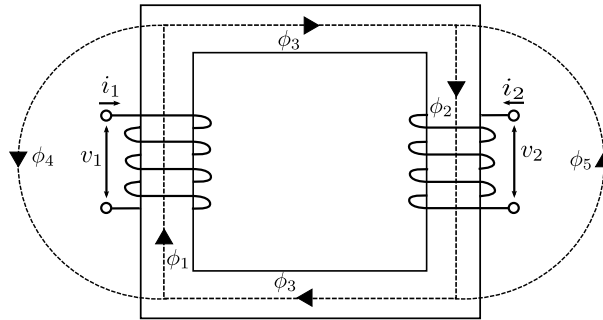
$$[I_{ns1}] = ([M_{11}] [P_1] [N_1])^{-1} (\frac{\Delta t}{2} [N_1]^{-1} [V_1] + [\Phi_1]) \quad (20)$$

* Logo, a corrente no enrolamento 1 é :

$$[I_1] = [Y_{11}] [V_1] + [I_{ns1}] \quad (21)$$

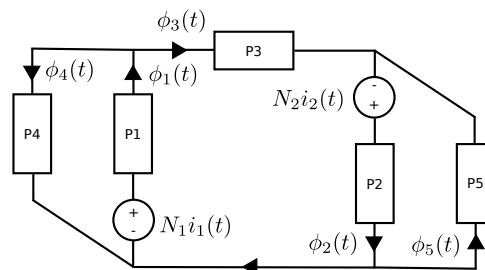
3.2 Modelo UMEC para reator monofásico naturalmente saturado com 2 enrolamentos

Figura 21 – Caminhos de fluxos magnéticos no núcleo.



Fonte: Próprio Autor

Figura 22 – Circuito UMEC do reator com dois enrolamentos



Fonte: Próprio Autor

A modelagem deste reator leva em consideração o circuito magnético da Figura 21 e o modelo UMEC mostrado na Figura 22.

Da mesma forma que foi calculada as matrizes para o reator anterior, aplica-se para o reator saturado com dois enrolamentos. Iniciando os estudos com a matriz de Incidência Nodal na Equação 22 e finalizando com a fórmula das correntes para os dois enrolamentos (Equação 35), temos:

* Matriz [A] é obtida através dos nós 1 e 2 do modelo UMEC:

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \quad (22)$$

* De acordo com o modelo UMEC a Matriz Permeância [P] é construída:

$$P = \begin{bmatrix} P_1 & 0 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 & 0 \\ 0 & 0 & P_3 & 0 & 0 \\ 0 & 0 & 0 & P_4 & 0 \\ 0 & 0 & 0 & 0 & P_5 \end{bmatrix} \quad (23)$$

* Matriz diagonal [N] dos dois enrolamentos:

$$N = \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix} \quad (24)$$

* Matriz das tensões dos enrolamentos:

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (25)$$

* Matriz dos fluxos magnéticos dos enrolamentos:

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} \quad (26)$$

* Matriz identidade [I] =

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

* Matriz relacionada a fonte de corrente de cada enrolamento [I_{ns}]:

$$I_{ns} = \begin{bmatrix} I_{ns1} \\ I_{ns2} \end{bmatrix} \quad (28)$$

* Matriz [M] quadrada :

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} \end{bmatrix} \quad (29)$$

como $[M] = [I] - [P][A]([A]^T[P][A])^{-1}[A]^T$, temos:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} P_1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_5 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \left([A]^T \begin{bmatrix} P_1 & 0 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 & 0 \\ 0 & 0 & P_3 & 0 & 0 \\ 0 & 0 & 0 & P_4 & 0 \\ 0 & 0 & 0 & 0 & P_5 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \right)^{-1} [A]^T \quad (30)$$

* A submatriz $[M_{ss}]$ pode ser obtida:

$$M_{ss} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (31)$$

* Matriz $[P_{ss}]$ relacionada aos enrolamentos 1 e 2 é :

$$P_s = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \quad (32)$$

* A matriz admitância pode ser entendida como: $[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1}$, logo:

$$Y_{ss} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \quad (33)$$

* Reescrevendo a matriz I_{ns} , temos:

$$\begin{bmatrix} I_{ns1} \\ I_{ns2} \end{bmatrix} = \left(\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix} \right)^{-1} \left(\frac{\Delta t}{2} \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix}^{-1} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} \right) \quad (34)$$

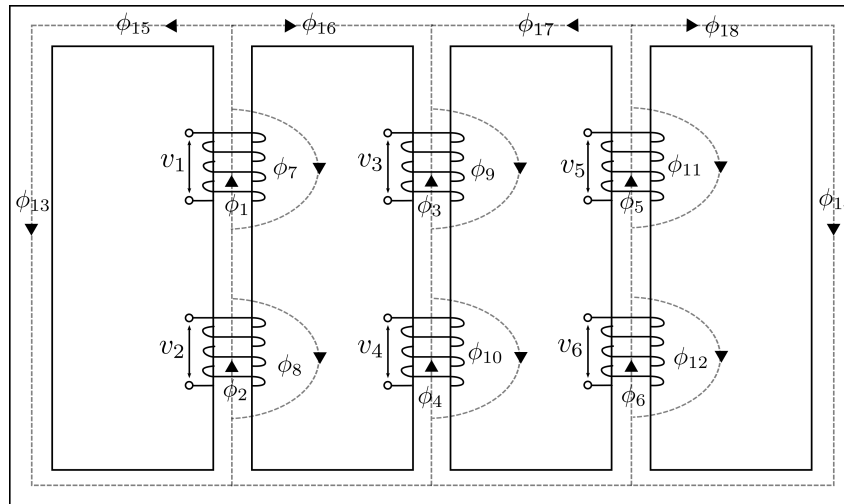
* Logo, as correntes nos enrolamentos 1 e 2 do reator são:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} I_{ns1} \\ I_{ns2} \end{bmatrix} \quad (35)$$

3.3 Modelo UMEC para reator trifásico naturalmente saturado com 6 enrolamentos e dois retornos

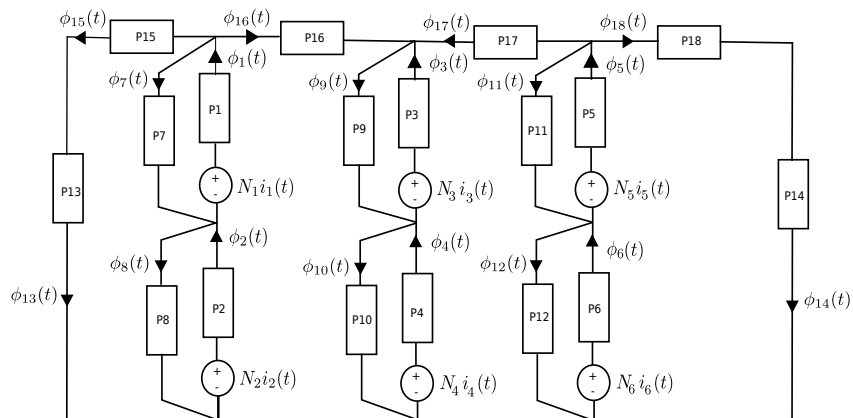
Replicando os conceitos utilizados nas outras modelagens segue os circuito magnético (Figura 23) e o UMEC do RNS com 6 enrolamentos e 2 retornos.

Figura 23 – Caminhos de fluxos magnéticos no núcleo.



Fonte: Próprio Autor

Figura 24 – Circuito UMEC do reator com 6 enrolamentos.



Fonte: Próprio Autor

As matrizes necessárias para a modelagem do RNS trifásico com 6 enrolamentos são mostradas a seguir:

* Matriz [A] é obtida através dos nós do modelo UMEC, logo:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad (36)$$

* Matriz Permeância [P] diagonal é obtida de acordo com o modelo UMEC:

$$\begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{18} \end{bmatrix}$$

* Matriz diagonal das espiras [N]:

$$N = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & N_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & N_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & N_6 \end{bmatrix} \quad (37)$$

* Matriz das tensões nos enrolamentos:

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} \quad (38)$$

* Matriz dos fluxos nos enrolamentos:

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \end{bmatrix} \quad (39)$$

* Matriz identidade [I]:

$$I_{18} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (40)$$

* Matriz relacionada a fonte de corrente de cada enrolamento [I_{ns}]:

$$I_{ns} = \begin{bmatrix} I_{ns1} \\ I_{ns2} \\ I_{ns3} \\ I_{ns4} \\ I_{ns5} \\ I_{ns6} \end{bmatrix} \quad (41)$$

* Matriz [M] quadrada é dada por:

$$[M] = [I] - [P][A]([A]^T[P][A])^{-1}[A]^T, \text{ logo}$$

* A submatriz $[M_{ss}]$ pode ser obtida:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix} \quad (42)$$

* Matriz [Pss] relacionada aos enrolamentos 1 ao 6 é :

$$P_s = \begin{bmatrix} P_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & P_6 \end{bmatrix} \quad (43)$$

* A matriz admitância pode ser entendida como: $[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1}$,
logo:

$$Y_{ss} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} & Y_{45} & Y_{46} \\ Y_{51} & Y_{52} & Y_{53} & Y_{54} & Y_{55} & Y_{56} \\ Y_{61} & Y_{62} & Y_{63} & Y_{64} & Y_{65} & Y_{66} \end{bmatrix} \quad (44)$$

* Reescrevendo a matriz I_{ns} , temos:

$$\begin{bmatrix} I_{ns1} \\ \vdots \\ \vdots \\ I_{ns6} \end{bmatrix} = \left([M_{ss}] \begin{bmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_6 \end{bmatrix} \begin{bmatrix} N_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & N_6 \end{bmatrix} \right)^{-1} \left(\frac{\Delta t}{2} [N_s]^{-1} \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ V_6 \end{bmatrix} + \begin{bmatrix} \Phi_1 \\ \vdots \\ \vdots \\ \Phi_6 \end{bmatrix} \right) \quad (45)$$

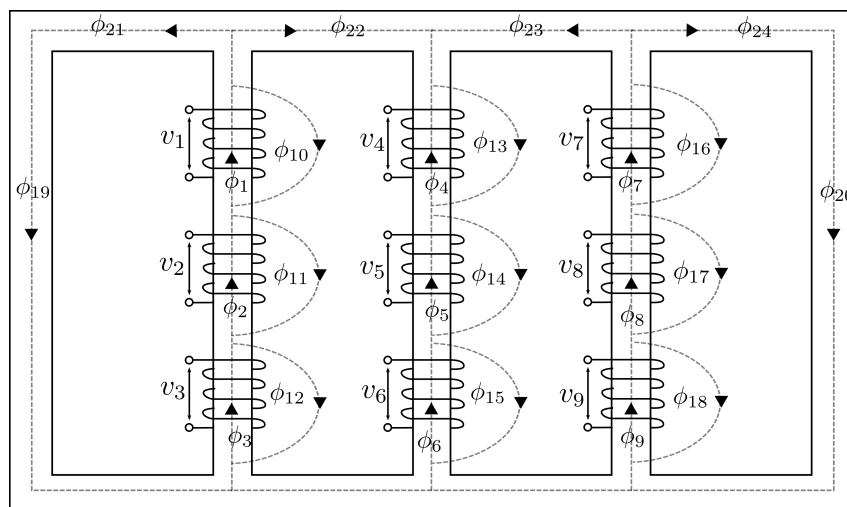
* Conseqüentemente, as correntes nos 6 enrolamentos são:

$$\begin{bmatrix} I_1 \\ \vdots \\ \vdots \\ I_6 \end{bmatrix} = \begin{bmatrix} Y_{1,1} & \dots & Y_{1,6} \\ \vdots & \ddots & \vdots \\ \vdots & \dots & \vdots \\ Y_{6,1} & \dots & Y_{6,6} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ V_6 \end{bmatrix} + \begin{bmatrix} I_{ns1} \\ \vdots \\ \vdots \\ I_{ns6} \end{bmatrix} \quad (46)$$

3.4 Modelo UMEC para reator trifásico naturalmente saturado com 9 enrolamentos e dois retornos

O circuito magnético (Figura 25) e o UMEC (Figura 26) do reator trifásico naturalmente saturado com 9 enrolamentos e 2 retornos. É mostrado na figura abaixo:

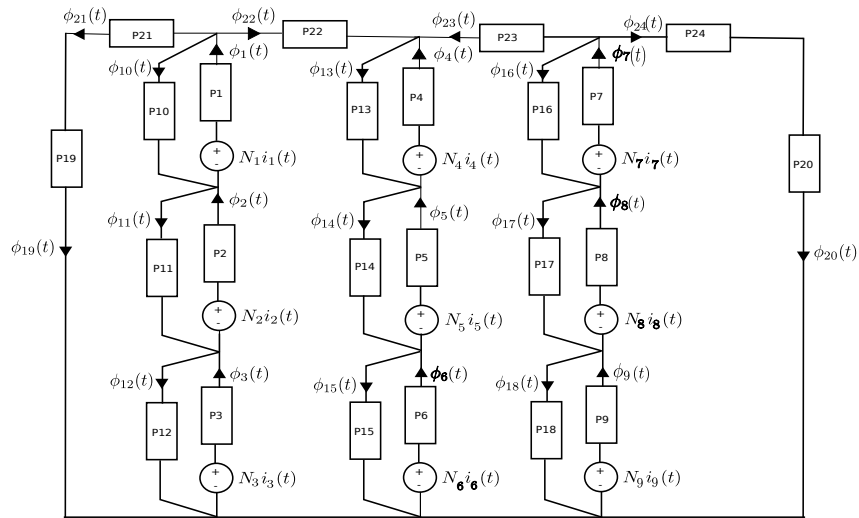
Figura 25 – Caminhos de fluxos magnéticos no núcleo.



Fonte: Próprio Autor

As matrizes necessárias para a modelagem do RNS trifásico com nove enrolamentos são mostradas:

Figura 26 – Circuito UMEC do reator de 9 enrolamentos.



Fonte: Próprio Autor

* Matriz [A] é obtida através dos nós do modelo UMEC, logo:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (47)$$

* Matriz Permeância [P] diagonal é obtida de acordo com o modelo UMEC e equivale a

uma matriz de tamanho 24.

$$P = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{24} \end{bmatrix} \quad (48)$$

* Matriz das tensões nos enrolamentos:

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \end{bmatrix} \quad (49)$$

* Matriz dos fluxos nos enrolamentos:

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \end{bmatrix} \quad (50)$$

* Matriz identidade [I] :

$$I_{24} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (51)$$

* Matriz relacionada a fonte de corrente de cada enrolamento $[I_{ns}]$:

$$I_{ns} = \begin{bmatrix} I_{ns1} \\ I_{ns2} \\ I_{ns3} \\ I_{ns4} \\ I_{ns5} \\ I_{ns6} \\ I_{ns7} \\ I_{ns8} \\ I_{ns9} \end{bmatrix} \quad (52)$$

* Matriz $[M]$ é dada por : $[M] = [I] - [P][A]([A]^T[P][A])^{-1}[A]^T$, logo:

* A submatriz $[M_{ss}]$ pode ser obtida:

$$M_{ss} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & m_{17} & m_{18} & m_{19} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & m_{27} & m_{28} & m_{29} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & m_{37} & m_{38} & m_{39} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & m_{47} & m_{48} & m_{49} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} & m_{57} & m_{58} & m_{59} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} & m_{67} & m_{68} & m_{69} \\ m_{71} & m_{72} & m_{73} & m_{74} & m_{75} & m_{76} & m_{77} & m_{78} & m_{79} \\ m_{81} & m_{82} & m_{83} & m_{84} & m_{85} & m_{86} & m_{87} & m_{88} & m_{89} \\ m_{91} & m_{92} & m_{93} & m_{94} & m_{95} & m_{96} & m_{97} & m_{98} & m_{99} \end{bmatrix} \quad (53)$$

* Matriz $[P_s]$ relacionada aos enrolamentos 1 ao 9 é :

$$P_s = \begin{bmatrix} P_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & P_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & P_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_9 \end{bmatrix} \quad (54)$$

* A matriz admitância pode ser entendida como: $[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1}$

* Reescrevendo a matriz I_{ns} , temos:

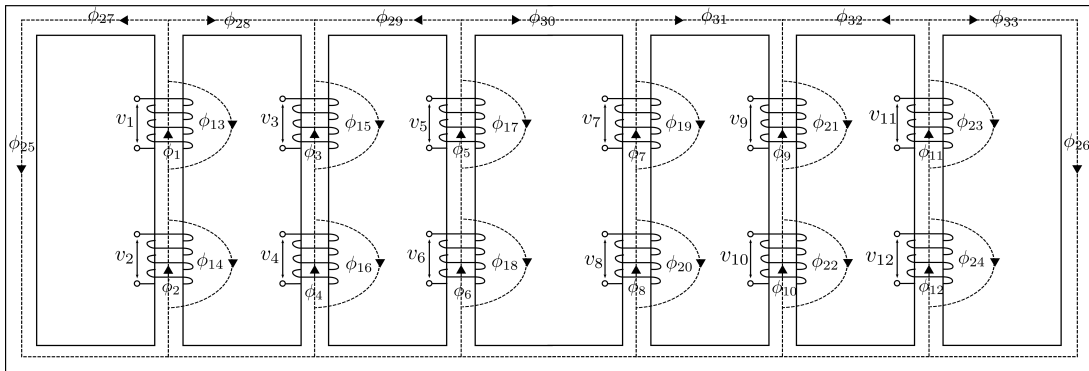
$$[I_{ns}] = ([M_{ss}] [P_{ss}] [N_{ss}])^{-1} (\frac{\Delta t}{2} [N_{ss}]^{-1} [V] + [\Phi]) \quad (55)$$

* Assim, as correntes nos 9 enrolamentos são:

$$\begin{bmatrix} I_1 \\ \vdots \\ \vdots \\ I_9 \end{bmatrix} = \begin{bmatrix} Y_{1,1} & \cdots & Y_{1,9} \\ \vdots & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ Y_{9,1} & \cdots & Y_{9,9} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ V_9 \end{bmatrix} + \begin{bmatrix} I_{ns1} \\ \vdots \\ \vdots \\ I_{ns9} \end{bmatrix} \quad (56)$$

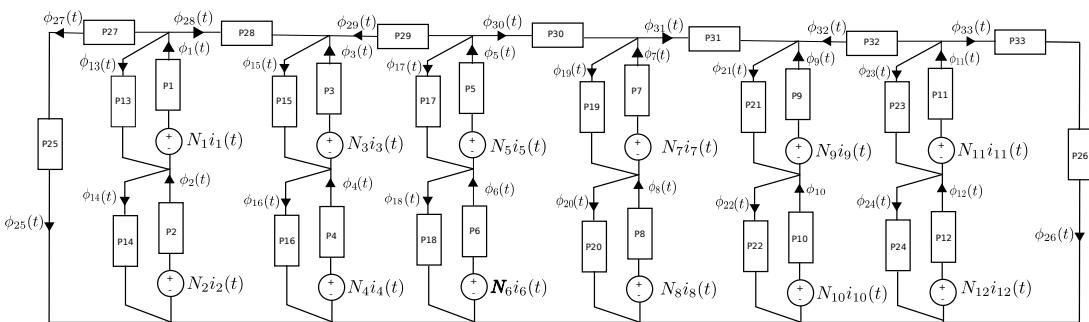
3.5 Modelo UMEC para reator trifásico com configuração em monobloco naturalmente saturado com 12 enrolamentos e dois retornos

Figura 27 – Caminhos de fluxos magnéticos no núcleo.



Fonte: Próprio Autor

Figura 28 – Circuito UMEC do reator em monobloco com 12 enrolamentos.



Fonte: Próprio Autor

O circuito magnético (Figura 27) e o UMEC (Figura 28) do reator trifásico naturalmente saturado com 12 enrolamentos e 2 retornos são mostrados. As matrizes necessárias para a modelagem do reator saturado trifásico com configuração monobloco são mostradas a seguir:

* Matriz [A] é obtida através do modelo UMEC, logo:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad (57)$$

* Matriz Permeância [P] diagonal é obtida de acordo com o modelo UMEC e equivale a uma matriz de tamanho 33.

$$P = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{33} \end{bmatrix} \quad (58)$$

* Matriz das tensões nos enrolamentos:

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \\ V_{10} \\ V_{11} \\ V_{12} \end{bmatrix} \quad (59)$$

* Matriz dos fluxos nos enrolamentos:

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \end{bmatrix} \quad (60)$$

* Matriz identidade [I] :

$$I_{33} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (61)$$

* Matriz relacionada a fonte de corrente de cada enrolamento $[I_{ns}]$:

$$I_{ns} = \begin{bmatrix} I_{ns1} \\ I_{ns2} \\ I_{ns3} \\ I_{ns4} \\ I_{ns5} \\ I_{ns6} \\ I_{ns7} \\ I_{ns8} \\ I_{ns9} \\ I_{ns10} \\ I_{ns11} \\ I_{ns12} \end{bmatrix} \quad (62)$$

* Matriz $[M]$ é dada por : $[M] = [I] - [P][A]([A]^T[P][A])^{-1}[A]^T$, logo:

* A submatriz $[M_{ss}]$ pode ser obtida:

$$M_{ss} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,12} \\ \vdots & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ m_{12,1} & \cdots & m_{12,12} \end{bmatrix} \quad (63)$$

* Matriz [Pss] relacionada aos enrolamentos 1 ao 12 é :

$$P_s = \begin{bmatrix} P_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & P_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & P_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{12} \end{bmatrix} \quad (64)$$

* A matriz admitância pode ser entendida como: $[Y_{ss}^*] = ([M_{ss}^*][P_s^*][N_s])^{-1} \frac{dt}{2} [N_s]^{-1}$

* Reescrevendo a matriz I_{ns} , temos:

$$[I_{ns}] = ([M_{ss}] [P_{ss}] [N_{ss}])^{-1} \left(\frac{\Delta t}{2} [N_{ss}]^{-1} [V] + [\Phi] \right) \quad (65)$$

* Logo, as correntes nos 12 enrolamentos é definida como:

$$\begin{bmatrix} I_1 \\ \vdots \\ \vdots \\ I_{12} \end{bmatrix} = \begin{bmatrix} Y_{1,1} & \cdots & Y_{1,12} \\ \vdots & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ Y_{12,1} & \cdots & Y_{12,12} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ V_{12} \end{bmatrix} + \begin{bmatrix} I_{ns1} \\ \vdots \\ \vdots \\ I_{ns12} \end{bmatrix} \quad (66)$$

4 Biblioteca RLib

Este capítulo apresenta uma descrição da biblioteca desenvolvida para avaliação de reatores naturalmente saturados utilizando a linguagem Models do ATP (RLib). A ideia de se utilizar uma biblioteca para reatores naturalmente saturados veio da necessidade de ter um componente unificado que permitisse a implementação de diferentes reatores naturalmente saturados sem a necessidade de se recriar todo o código para cada componente. Dessa forma, cientistas e engenheiros podem simular o comportamento de RNSs com diferentes configurações de uma maneira mais simples que copiar o código de reatores previamente implementados e adicionar novas funcionalidades. Esse tipo de implementação é mais suscetível a erros.

Com a utilização do componente proposto, o trabalho de elaboração de um modelo de simulação é facilitado e o tempo de desenvolvimento do projeto do reator é diminuído de forma significativa. Este capítulo é dividido em três seções principais listadas a seguir: (i) arquitetura do biblioteca RLib; (ii) comportamento da biblioteca RLib; (iii) integração com modelos de reatores naturalmente saturados. É importante destacar que nesse capítulo além de uma visão geral de como é o comportamento e estrutura da biblioteca RLib, serão mostrados também detalhes de implementação de como utilizar a biblioteca. Dessa forma, analistas de RNSs poderão criar modelos de reatores utilizando os exemplos aqui mostrados como base.

4.1 Considerações Importantes Sobre a Linguagem Models

Esta seção apresenta algumas considerações que devem ser mencionadas para deixar mais clara a forma como alguns componentes da linguagem Model foram utilizados. Por exemplo, na linguagens modelos a representação de matrizes retangulares não é claramente definida. Portanto, soluções alternativas tiveram que ser implementadas para que esse tipo de comportamento fosse codificado. Os dois principais elementos destacados nesse capítulo se referem a utilização de subrotinas e implementação de matrizes retangulares.

4.1.1 Implementação de subrotinas.

A linguagem models apesar de poderosa, difere de linguagens de programação convencionais como C, Java ou Python. Uma vez que tem o foco na representação e interligação de componentes elétricos, a linguagem não possui alguns elementos que facilitam a representação de tipos de dados de alto nível como classes, estrutura de dados ou subrotinas. Ao invés disso, a linguagem permite a criação de submodelos hierárquicos que simulam o comportamento de subrotinas. Mais especificamente, a linguagem models permite as seguintes formas de se escrever subrotinas ou procedimentos (DUBÉ, 1996):

- **Função declarativas.** Uma função declarativa possui as mesmas características de uma expressão. Para um certo conjunto de argumentos, uma função da linguagem Models calcula a expressão e retorna um resultado. Contudo, toda expressão deve ser atômica e variáveis intermediárias não podem ser criadas. Dessa forma, funções da linguagem Models diferem de funções em linguagens de mais alto nível, como C++ por exemplo, dado que expressões não podem ser encadeadas. Na linguagem models, uma expressão por função é admitida.

O trecho de código 4.1 apresenta a forma como funções são definidas na linguagem Models. A palavra chave *FUNCTION* inicia a função. Em seguida, o nome da função declarativa deve ser declarada, sua lista de argumentos (*argname1*, *argname2*, ...) e por fim uma expressão (*expression*) que calcula e retorna um resultado baseado nos parâmetros informados.

```
1 FUNCTION somename(argname1, argname2, ... ) := expression
```

Código Fonte 4.1 – Template para criação de funções declarativas.

Um exemplo da utilização de uma função declarativa pode ser vista no Código Fonte 4.2. Nesse caso, três variáveis são somadas e o resultado é retornado. Observe que apenas um expressão pode ser utilizada para o cálculo do resultado. Portanto, algoritmos mais complexos que requerem a utilização de uma ou mais variáveis não podem ser criados apenas utilizando-se funções declarativas.

```
1 -- Calcula a soma de param1 + param2 + param3
2 FUNCTION simple_sum(param1, param2, param3) := a+b+c
3 ...
4 -- result guarda o valor 6.
5 result := simple_sum(1, 2, 3)
```

Código Fonte 4.2 – Exemplo de utilização de funções declarativas

- **Função externa.** Uma função externa corresponde a uma rotina escrita em outra linguagem de programação e adicionada a linguagem models no processo de montagem do código fonte (*linking process*). Apesar de apresentar uma solução viável para construir subrotinas complexas, o código fonte do ATP precisa ser recompilado juntamente com o código da rotina externa para que essa possa ser utilizada o que torna o processo como um todo bastante trabalhoso. Para detalhes de como utilizar funções externas o leitor é direcionado a referência (DUBÉ, 1996)
- **Submodelos da Models.** Submodelos da linguagem Models, ou simplesmente modelos, são componentes da linguagem que permitem guardar valores e executar procedimentos. Portanto, modelos intermediários são utilizados para representar o comportamento de procedimentos na linguagem. Ao invés de criarmos um procedimento e utilizarmos este para

a execução de uma subrotina, um submodelo é criado e este é usado para representar/executar a funcionalidade desejada. Existem duas formas de se utilizar um submodelo dentro de outro modelo: (i) utilizando um modelo localmente definido, ou (ii) empregando-se um modelo externo. No modelo localmente definido, a sua descrição é feita dentro do modelo que o chama. Por outro lado, o modelo externo é declarado fora do módulo que o utiliza. O Código Fonte 4.3 apresenta a forma como um submodelo é criado e utilizado. Observe que um modelo de mais alto nível é criado na linha 1 (*modelo_principal*) e dentro deste um submodelo é criado (*sub_modelo*). A forma de utilização do submodelo pelo modelo principal é mostrada na linha 17.

```

1  MODEL modelo_principal
2  -- O modelo de alto nível é definido da forma habitual
3  CONST ...
4  DATA ...
5  INPUT ...
6  VAR ...
7  OUTPUT ...
8      MODEL sub_modelo
9      -- O modelo intermediário é criado e utilizado no modelo principal
10     CONST ...
11     DATA ...
12     ...
13     ENDMODEL
14     ...
15     EXEC
16     ...
17     USE sub_modelo AS submod_1
18     -- Forma de utilização do submodelo
19     ...
20     ENDUSE
21     ...
22     ENDEXEC
23     ENDMODEL

```

Código Fonte 4.3 – Exemplo de utilização de submodelos localmente definidos

A outra forma de utilização de um submodelo corresponde a um modelo externo. Nesse caso, ao invés de se definir o submodelo dentro do modelo que o utiliza, o modelo externo é definido fora do modelo chamador utilizando-se a palavra chave *EXTERNAL*. O Código Fonte 4.4 apresenta a forma de utilização de um modelo externo na linguagem models. Nesse exemplo o modelo_utilizador (linhas 1 a 5) executa o modelo_utilizado (linhas 16). Para que o modelo_utilizador reconheça o modelo_utilizado como um submodelo é necessário que isso seja definido com o uso da palavra chave *EXTERNAL* na linha 12.

```

1  MODEL modelo_utilizado
2  CONST ...
3  DATA ...
4  ...
5  ENDMODEL
6

```

```

7  MODEL modelo_utilizador
8  -- Esse modelo vai utilizar o modelo_utilizado
9  CONST ...
10 DATA ...
11 ...
12 MODEL modelo_utilizado EXTERNAL
13 ...
14 EXEC
15 ...
16 USE modelo_utilizado AS mod1
17 -- O modelo_utilizado está sendo usado nesse ponto.
18 ...
19 ENDUSE
20 ...
21 ENDEXEC
22 ENDMODEL

```

Código Fonte 4.4 – Forma de utilização de modelos externos.

A biblioteca RLib utiliza submodelos internos e externos para declaração de procedimentos e subrotinas fazendo com que a solução proposta seja totalmente compatível com a linguagem model da ATP e não dependente de nenhuma outra linguagem de programação como no caso da utilização de funções externas.

4.1.2 Representação de matrizes retangulares.

Linguagens de programação de alto nível como C++, Java ou Python apresentam suporte nativo para representação e manipulação de matrizes retangulares. Por exemplo, na linguagem java, a expressão

```
1 int matrix[][] = new int[20][20];
```

cria uma matriz de inteiros chamada matrix com 20 linhas e 20 colunas. Contudo, na linguagem models apenas matrizes de uma dimensão (matrizes do tipo linha) são suportados (DUBÉ, 1996). Assim, uma solução alternativa é proposta para contornar essa restrição da linguagem e ainda assim representar matrizes com mais de uma dimensão. Nesse caso, para representa uma matrix $A_{n \times m}$ utilizamos um vetor de B de dimensão $p = n \times m$. Para tal, cada elemento de $a_{i,j} \in A$ é mapeado em um elemento $b \in B_p$. Esse mapeamento é representado a seguir:

$$b_{(i-1) \times m + j} \in \mathbb{R}^p = a_{i,j} \in \mathbb{R}^{n \times m}$$

Esse mapeamento representa as n linhas da matriz A de forma justaposta no vetor B . Considere esse exemplo para demonstrar como uma matriz retangular $C_{n \times m}$, definida a seguir, pode ser mapeada em um vetor D com dimensão $p = n \times m$.

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,m} \end{bmatrix}.$$

Utilizando-se o mapeamento definido acima podemos representar o vetor D na forma a seguir.

$$D = (d_1, d_2, \dots, d_{n \times m}) = (c_{1,1}, \dots, c_{1,m}, c_{2,1}, \dots, c_{2,m}, \dots, c_{n,1}, \dots, c_{n,m})$$

Portanto, sempre que quisermos representar uma matriz retangular de n linhas e m colunas na linguagem `models`, utilizaremos um vetor com tamanho $n \times m$ e acessaremos o elemento da linha n e coluna m da forma apresentada no Código Fonte 4.5. Na linha 2, um vetor (chamado *matrix*) de tamanho 200 (10×20), o número de linhas da matriz (*linhas*) e o número de colunas (*colunas*) são representados. Em seguida essas variáveis são inicializadas na linha 6. Nas linhas 11 e 14, os elementos das posições (5, 4) e (6, 9) da matriz são atualizadas com valores 12 e 13 respectivamente. Por fim, o elemento da posição (9, 10) recebe a soma das dos dois elementos mencionados anteriormente.

```

1 MODEL matrixRepresentation
2   VAR matrix[1..200], linhas, colunas
3   INIT
4     -- Cria uma matrix de 20 linhas e 10 colunas e inicializa todos os
5     -- elementos com valor '0'
6     linhas := 20, colunas:= 10, matrix[1..200]:=0
7   ENDINIT
8   EXEC
9
10  -- Guarda o valor 12 no elemento da linha 5 e coluna 4 (matrix[5][4] = 12).
11  matrix[(5 - 1)*colunas + 4] = 12
12
13  -- Guarda o valor 13 no elemento da linha 6 e coluna 9 (matrix[6][9] = 13).
14  matrix[(6 - 1)*colunas + 9] = 13
15
16  -- Soma o elemento da posição (5, 4) com o element (6, 9) e guarda no elemento (9, 10)
17  matrix[(9 - 1)*colunas + 10] = matrix[(5 - 1)*colunas + 4] + matrix[(6 - 1)*colunas + 9]
18
19  -- O elemento (9, 10) da matrix contem o valor 25
20  ENDEXEC
21 ENDMODEL

```

Código Fonte 4.5 – Forma alternativa de representação de matrizes retangulares na linguagem `models`

4.2 Visão Geral da Arquitetura da Biblioteca RLib

O diagrama de classes (LOBO,) que representa a biblioteca RLib é mostrada na Figura 29. Na Figura são listados sete módulos auxiliares (RLibAux) e um módulo principal (RNSMatrix). Esses oito módulos juntos formam a biblioteca RLib. Na linguagem Models, um módulo é também chamado de modelo. Portanto, nesse documento utilizaremos ambas as expressões (módulo e modelo) para designar uma entidade do sistema que possui um conjunto de atributos e operações. Nesta seção, inicialmente será apresentada a biblioteca RLibAux que é composta por modelos criados para realizar operações sobre matrizes (Seção 4.2.1). Em seguida o comportamento da biblioteca RLib propriamente dita é descrita (Seção 4.2.2).

4.2.1 Pacote para operações sobre matrizes - RLibAux

Os modelos da biblioteca RLibAux (Figure 29) são mostrados a seguir:

1. *sumMatrices*. Este módulo é responsável por realizar a soma de matrizes (Código Fonte 4.6). O módulo recebe duas matrizes (e.g., mmA e mmB) como parâmetro e retorna uma nova matriz correspondente a soma das entradas ($mmOut = mmA + mmB$). Observe que a forma de representação das duas matrizes está de acordo com o que foi apresentado na Seção 4.1.2. Dois *loops* encadeados do tipo **FOR** (linhas 17 e 18) são criados para varrer os elementos das matrizes mmA e mmB e armazenar a respectiva soma na matriz mmC . Observe que na linha 19 os elementos da mmA e mmB são somados e o resultado é guardado na matrix $mmOut$.

Para o modelo apresentado, o número máximo de elementos em cada matriz de entrada e na matriz de saída é 900 (linhas 2, 3 e 6). Contudo, é possível aumentar essa capacidade alterando-se o tamanho das matrizes de entrada e saída. É importante salientar que quanto maior a capacidade de representação de cada matriz maior será a quantidade de memória consumida pela biblioteca.

```

1 MODEL sumMatrices
2   INPUT linesA, columnsA, mmA[1..900]
3   linesB, columnsB, mmB[1..900]
4   VAR i, j, c, mmOut[1..900], linesOut, columnsOut
5   OUTPUT
6   mmOut[1..900], linesOut, columnsOut
7   INIT
8   i := 0, j:= 0, c:= 0, mmOut[1..900]:=0
9   ENDINIT
10  EXEC
11
12  linesOut := linesA, columnsOut := columnsA
13
14  -----
15  -- SUM THE MATRICES
16  -----

```

```

17   FOR i:=1 TO linesA DO
18     FOR j:=1 TO columnsA DO
19       mmOut[(i-1)*columnsA + j] := mA[(i-1)*columnsA + j]+mB[(i-1)*columnsB + j]
20     ENDFOR
21   ENDFOR
22   ENDEXEC
23 ENDMODEL

```

Código Fonte 4.6 – Módulo da biblioteca RLib para soma de matrizes

2. *subtractMatrices*. Análogo ao *sumMatrices*, a diferença como o próprio nome indica é que ao invés de somarmos duas matrizes, este modelo subtrai duas matrizes de entrada mmA e mmB e retorna como resultado a matriz $mmOut = mmA - mmB$. A estrutura do módulo é apresentada no Código Fonte 4.7. Observe que a subtração das matrizes é efetivamente realizada na linha 19, na qual para cada linha e coluna da matriz de saída o elemento resultante vem da operação $mmA - mmB$ da linha e coluna correspondente.

```

1  MODEL subtractMatrices
2  INPUT linesA, columnsA, mA[1..900]
3  linesB, columnsB, mB[1..900]
4  VAR i, j, c, mmOut[1..900], linesOut, columnsOut
5  OUTPUT
6  mmOut[1..900], linesOut, columnsOut
7  INIT
8  i := 0, j:= 0, c:= 0, mmOut[1..900]:=0
9  ENDINIT
10 EXEC
11
12 linesOut := linesA, columnsOut := columnsA
13
14 -----
15 -- SUBTRAC THE MATRICES
16 -----
17 FOR i:=1 TO linesA DO
18   FOR j:=1 TO columnsA DO
19     mmOut[(i-1)*columnsA + j] := mA[(i-1)*columnsA + j]-mB[(i-1)*columnsB + j]
20   ENDFOR
21 ENDFOR
22 ENDEXEC
23 ENDMODEL

```

Código Fonte 4.7 – Módulo da biblioteca RLib para subtração de matrizes

3. *transposeMatrices*. Nesse módulo (Código Fonte 4.8), uma matriz de entrada mmA é passada como parâmetro (linha 2) e a saída (linha 5) corresponde a matriz transposta da entrada ($mmOut = mmA^T$). O número de linhas da matriz de saída corresponde ao número de colunas da matriz de entrada e o número de colunas da matriz de saída recebe o número de linhas da matriz de entrada (linhas 11 e 12). Dois *loops* do tipo **FOR** são criados para varrer as linhas e colunas da matriz de entrada (linhas 17 e 18) e cada elemento da linha i e coluna j da matriz de entrada é copiado para o elemento da linha i e

coluna j da matriz de saída (linha 19).

```

1 MODEL transposeMatrix
2 INPUT linesA, columnsA, mA[1..900]
3 VAR i, j, mmOut[1..900], linesOut, columnsOut
4 OUTPUT
5 mmOut[1..900], linesOut, columnsOut
6 INIT
7 i := 0, j:= 0, mmOut[1..900]:=0
8 ENDINIT
9 EXEC
10
11 linesOut := columnsA
12 columnsOut := linesA
13
14 -----
15 -- TRANSPOSE THE MATRIX
16 -----
17 FOR i:=1 TO linesA DO
18   FOR j:=1 TO columnsA DO
19     mmOut[(j-1)*columnsOut + i] := mA[(i-1)*columnsA + j]
20   ENDFOR
21 ENDFOR
22 ENDEXEC
23 ENDMODEL

```

Código Fonte 4.8 – Módulo da biblioteca RLib para transposição de atrizes

4. *changeMatrixSize*. Este modelo realiza a criação de uma submatriz a partir de uma matriz de entrada. Dada uma matriz $A_{n \times m}$ e dois números inteiros $k \leq n$ e $j \leq m$. Chamamos a submatriz $B_{k \times j}$ de $A_{n \times m}$, uma matriz formada por elementos k linhas e j colunas de A (situadas nas posições correspondentes em A) (LIMA, 2014). No caso específico da biblioteca RLib, apenas as últimas $t = n - k$ linhas e $u = m - j$ colunas são removidas da matriz original A .

Considere o exemplo a seguir. Seja a matriz $C_{4 \times 4}$ dada por

$$C = \begin{bmatrix} 1 & 2 & 7 & 0 \\ 4 & 5 & 9 & 1 \\ 3 & 5 & 10 & 1 \\ 10 & 3 & 6 & 4 \end{bmatrix}.$$

A matrix D de dimensão 2×2 é uma submatriz de C dada por

$$D = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}.$$

É importante salientar que como a representação de matrizes retangulares é feita de forma indireta por meio de um vetor (Seção 4.1.2), a simples mudança das variáveis que representam os números de linhas e colunas da matriz original não é suficiente para realizar tal

operação. Nesse caso, é necessário que se transporte cada elemento da matriz de origem para a nova matriz reduzida. O Código Fonte 4.9 apresenta o modelo que obtém uma submatriz ($mmOut_{linesOut \times columnsOut}$) a partir de uma matriz de entrada $mmA_{linesA \times columnsA}$. Na linha 2, são passados os parâmetros de entrada do modelo linhas e colunas da matriz original mmA e quantidade de linhas e colunas da matriz de saída $mmOut$. Nas linhas 12 e 13, a quantidade linhas e colunas da matriz de saída é atualizada e nas linhas 18 e 19 dois *loops* do tipo **FOR** são criados para varrer as linhas e colunas da matriz de saída. A cópia dos valores da matriz de entrada para a matriz de saída é realizada na linha 20. Observe, que cada elemento da mmA é copiado para matriz $mmOut$ só nas linhas e colunas da matriz de saída.

```

1 MODEL changeMatrixSize
2 INPUT linesA, columnsA, mmA[1..900],
3     newLines, newColumns
4 VAR i, j, mmOut[1..900], linesOut, columnsOut
5 OUTPUT
6     mmOut[1..900], linesOut, columnsOut
7 INIT
8     i := 0, j := 0, mmOut[1..900] := 0
9 ENDINIT
10 EXEC
11
12 linesOut := newLines
13 columnsOut := newColumns
14
15 -----
16 -- CHANGE MATRIX SIZE
17 -----
18 FOR i:=1 TO newLines DO
19     FOR j:=1 TO newColumns DO
20         mmOut[(i-1)*columnsOut + j] := mmA[(i-1)*columnsA + j]
21     ENDFOR
22 ENDFOR
23 ENDEXEC
24 ENDMODEL

```

Código Fonte 4.9 – Módulo da biblioteca RLib para modificar as dimensões de uma matriz

5. *multiplyMatrixScalar*. Este módulo (Código Fonte 4.10) realiza a multiplicação de uma matriz de entrada mmA por um escalar *scalar* (linhas 2 e 3) e retorna uma matriz de saída $mmOut = mmA.scalar$. Como nos módulos apresentados anteriormente, dois *loops* do tipo **FOR** (linhas 18 e 19) para varrer as linhas e colunas da matriz de saída e guardar o resultado da multiplicação de cada elemento da matriz de entrada pelo valor escalar (linha 20).

```

1 MODEL multiplyMatrixScalar
2 INPUT linesA, columnsA, mmA[1..900],
3     scalar
4 VAR i, j, mmOut[1..900], linesOut, columnsOut

```

```

5  OUTPUT
6  mmOut[1..900], linesOut, columnsOut
7  INIT
8  i := 0, j:= 0, mmOut[1..900]:=0
9  ENDINIT
10 EXEC
11
12 linesOut := linesA
13 columnsOut := columnsA
14
15 -----
16 -- MULTIPLY THE MATRICES
17 -----
18 FOR i:=1 TO linesA DO
19   FOR j:=1 TO columnsA DO
20     mmOut[(i-1)*columnsOut + j] := mmA[(i-1)*columnsA + j]*scalar
21   ENDFOR
22 ENDFOR
23 ENDEXEC
24 ENDMODEL

```

Código Fonte 4.10 – Módulo da biblioteca RLib para multiplicação de uma matriz por escalar

6. *multiplyMatrices*. O módulo que realiza multiplicação de matrizes é aprensetado no Código Fonte 4.11. Este módulo recebe duas matrizes como entrada mmA e mmB (linhas 2 e 3) e retorna como resultado o produto das duas matrizes $mmOut = mmA mmB$. O matriz de saída possui o número de linhas da matriz mmA e o número de colunas da matriz mmB (linhas 12 e 13). Dois *loops* do tipo **FOR** são criados para varrer as linhas e colunas da matriz de saída (linhas 18 e 19). Cada entrada da matriz de saída corresponde ao produto interno da respectiva linha da matriz mmA pela coluna da matriz mmB (linhas 21, 22 e 23).

```

1  MODEL multiplyMatrices
2  INPUT linesA, columnsA, mmA[1..900]
3     linesB, columnsB, mmB[1..900]
4  VAR i, j, c, k, mmOut[1..900], linesOut, columnsOut
5  OUTPUT
6  mmOut[1..900], linesOut, columnsOut
7  INIT
8  i := 0, j:= 0, c:= 0, k:=0, mmOut[1..900]:=0
9  ENDINIT
10 EXEC
11
12 linesOut := linesA
13 columnsOut := columnsB
14
15 -----
16 -- MULTIPLY THE MATRICES
17 -----
18 FOR i:=1 TO linesA DO
19   FOR j:=1 TO columnsB DO
20     c := 0
21     FOR k:=1 TO linesB DO

```

```
22         c := c + mmA[(i-1)*columnsA + k]*mmB[(k-1)*columnsB + j]
23     ENDFOR
24     mmOut[(i-1)*columnsB + j] := c
25 ENDFOR
26 ENDFOR
27 ENDEXEC
28 ENDMODEL
```

Código Fonte 4.11 – Módulo da biblioteca RLib para multiplicar duas matrizes

7. *invertMatrix*. O Código Fonte A.1 apresenta o módulo que realiza inversão de matrizes e cálculo de determinantes. Uma forma bastante difundida para o cálculo da inversa de matrizes corresponde a técnica baseada na matriz adjunta (transposta da matriz de cofatores) (STRANG, 2003). Apesar de ser facilmente codificada, a implementação padrão do algoritmo requer a utilização de procedimentos recursivos para o cálculo do determinante das matrizes dos cofatores (PRESS, 2007). Como a linguagem Models não oferece suporte a recursividade, um algoritmo iterativo da (decomposição LU) foi utilizado para cálculo da inversa de uma matriz.

Sendo assim, utilizamos o método da decomposição LU para representar a matriz de entrada A como o produto de duas matrizes triangulares L (inferior) e U (superior) (PRESS, 2007). O método da decomposição LU foi inicialmente proposto pelo matemático inglês Allan Turing em 1948 (TURING, 1948). Além das matrizes L e U , foi utilizada uma matriz de permutação P que é utilizada para diminuir os erros de arredondamento nos cálculos utilizados (TREFETHEN; BAU, 1997). A matrix P possui as mesmas dimensões da matriz a ser invertida e cada linha possui apenas um zero. Tal matriz serve para trocar a ordem das linhas da matriz de entrada de forma que linhas que apresentem números com valores absolutos maiores apareçam nas primeiras posições. Dada uma matriz de entrada A a ser invertida. O primeiro passo do algoritmo consiste em criar uma matriz de pivotamento que troque a ordem das linhas da matriz de entrada conforme descrito acima (linhas 36 a 60 do Código Fonte A.1).

Em seguida, a matriz de pivotamento P é multiplicada pela matriz de entrada A (linhas 65 a 72). O próximo passo consiste em avaliarmos as matrizes L e U que representem a matriz PA (linhas 98 a 119).

$$PA = LU$$

O método Crout (STRANG, 2003; PRESS, 2007) para decomposição LU foi adotado e é apresentado a seguir. Dadas uma matriz $B = LU$ onde L corresponde a uma matriz triangular inferior e U uma matriz triangular superior representadas a seguir.

$$\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{2,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{n,n} \end{bmatrix}$$

Os elementos das matrizes U e L são dados pelas Equações 1 e 2.

$$u_{i,j} = b_{i,j} - \sum_{k=1}^{i-1} u_{k,j} l_{i,k}, \quad i \leq j \quad (1)$$

$$l_{i,j} = \frac{1}{u_{j,j}} (b_{i,j} - \sum_{k=1}^{j-1} u_{k,j} l_{i,k}), \quad i > j \quad (2)$$

A partir desse ponto, pode-se utilizar a seguinte estratégia para se calcular a matriz inversa de A . Sabe-se que

$$AA^{-1} = I$$

onde I é a matriz identidade. Portanto, a seguinte expressão também é verdadeira.

$$PAA^{-1} = PI = P$$

dado que $PA = LU$, então

$$LUA^{-1} = P$$

Seja $UA^{-1} = Z$, então $LZ = P$. Dado que P e L são conhecidos, a matriz Z pode ser encontrada facilmente utilizando-se a técnica de substituição direta (STRANG, 2003). Uma vez que a matriz Z seja conhecida, pode-se calcular a matriz A^{-1} (referenciada como matriz X no código fonte) pela técnica de substituição reversa (STRANG, 2003). Estes passos são codificados nas linhas 132 a 143 e nas linhas 148 a 159.

4.2.2 Comportamento da Biblioteca RLIB

A implementação da biblioteca RLib está contida no Código Fonte A.2. Os módulos para operações com matrizes (Pacote RLibAux) estão localizados dentro do mesmo arquivo da biblioteca RLib e estão definidos entre as linhas 34 e 361 como submodelos localmente definidos (Seção 4.1.1). O algoritmo para determinação dos elementos do modelos UMEC são apresentados na Seção 2.3.1. As entradas da biblioteca RLib são (linhas 2 a 9):

- **Passo de Simulação - *deltaTime*** - Δt Passo discreto do processo de simulação (definido no arquivo ATP).
- **Matriz de Incidência Nodal - *MatrixA***. Corresponde a uma matriz de conexão de nós e ramos do circuito magnético.
- **Matriz de Permeâncias - *MP***. A matriz de permeâncias corresponde a uma matrix diagonal com as permeâncias de todos os ramos do modelo UMEC.
- **Matriz de Número de Espiras em Cada Enrolamento - *MatrixN***. Matriz diagonal com o número de espiras de cada enrolamento do reator.
- **Matriz de Tensões nos Enrolamentos - *MatrixV***. Matriz coluna com as tensões em cada um dos enrolamentos do reator.
- **Matriz de Fluxo Magnético nos Enrolamentos - *MatrixO***. Matriz coluna com o fluxo magnético para cada enrolamento do reator.
- **Matriz das Correntes da Fonte de Tensão Equivalente de entrada - *MatrixInsIn***. Matriz coluna com as correntes da fonte de tensão equivalente para cada enrolamento no tempo $t - \Delta t$.

As saídas da biblioteca RLib são apresentadas a seguir (linhas 31 e 32 do Código Fonte A.2):

- **Matriz das Correntes da Fonte de Tensão Equivalente de saída - *MatrixIns***. Matriz coluna com as correntes da fonte de tensão equivalente para cada enrolamento no tempo corrent t .
- **Matriz das Correntes de Saída - *MatrixIs***. Matriz coluna com a corrente de saída em cada enrolamento.

A Figura 30 mostra o diagrama de sequência (LOBO,) do funcionamento das chamadas da biblioteca RLib e como as entidades da biblioteca são utilizadas. Na explicação que segue, utilizaremos a figura para explicar a dinâmica de eventos do Código Fonte A.2.

- **Inicialização**. No início da execução do módulo, as dimensões da matriz de permeância (*MP*) são utilizadas para para criar uma matriz identidade de mesmo tamanho (linhas 373 até 379 - Código Fonte A.2). Esse passo não é mostrado na Figura 30.
- **Passo 1**. A matrix de permeâncias é multiplicada pela matriz de incidência nodal utilizando o modelo multiplyMatrices (linhas 389 até 401). Observe na Figura 30 que o Modelo RLib utiliza o modelo multiplyMatices passando os respectivos argumentos. O retorno dessa operação ($P \times A$) é passado para o módulo principal da biblioteca RLib.

- **Passo 2.** Nesse passo, a matriz de incidência nodal é transposta por meio do modelo `transposeMatrix`. Essa operação é realizada nas linhas 403 a 412 do Código Fonte A.2. Após isso, a matriz A transposta (A^T) é retornada para o módulo principal.
- **Passo 3.** A matriz de incidência nodal transposta (A^T) é multiplicada pela matriz $P \times A$ utilizando o modelo `multiplyMatrices` (linhas 414 a 426 do Código Fonte A.2). O retorno dessa operação $A^T \times (P \times A)$ é passado para o módulo principal (RLib).
- **Passo 4.** A matriz $A^T(PA)$ é invertida utilizando o modelo `invertMatrix` e o resultado $T = (A^T \times (P \times A))^{-1}$ é retornado para o módulo principal (linhas 428 até 437).
- **Passo 5.** Caso a matriz T avaliada no Passo 4 seja inversível, o modelo `multiplyMatrix` é utilizado para encontrar o produto das matrizes $P \times A$ e T . O resultado dessa operação $Q_1 = (P \times A) \times T$ é retornado para o módulo principal. Este passo é executado nas linhas 441 até 453 do Código Fonte A.2.
- **Passo 6.** Nesse ponto, a matriz $Q_2 = Q_1 \times A^T$ é avaliada por meio do modelo `multiplyMatrices` (linhas 455 até 467).
- **Passo 7.** A matriz identidade I criada na **Inicialização** é passada junto com Q_2 para o módulo `subtractMatrices`. O retorno dessa operação é matriz $Q = I - Q_2$ calculada nas linhas 469 até 481.
- **Passo 8.** Uma submatriz de Q é calculada (através do módulo `changeMatrixSize`) para que se obtenha uma matriz quadrada Q_n com número de linhas e colunas dada pelo número de núcleos do reator (linhas 483 até 494).
- **Passo 9.** Similarmente ao que foi feito com a matriz Q_n , uma nova matriz P_n é encontrada pela mudança das dimensões da matriz de permeâncias. O módulo `changeMatrixSize` é utilizado e o número de linhas e colunas da matriz P_n corresponde ao número de núcleos do reator (linhas 496 até 507).
- **Passo 10.** Nesse passo, a inversa da matriz do número de espiras em cada enrolamento ($MatrixN$) é encontrada pelo módulo `invertMatrix` $N^{-1} = MatrixN^{-1}$. Essa operação é efetuada nas linhas 509 até 518 do Código Fonte A.2.
- **Passo 11.** Caso $MatrixN$ seja inversível, o modelo `multiplyMatrixScalar` é utilizado para multiplicar a matriz N^{-1} pela metade do intervalo definido na simulação do ATP δ_t . O resultado $B = N^{-1} \cdot \delta_t / 2$ é retornado para o módulo principal. Esse passo é executado nas linhas 524 até 534.
- **Passo 12.** Nesse passo, as matrizes Q_n e P_n são multiplicados utilizando-se o módulo `multiplyMatrices`. O resultado $Q_n \times P_n$ é retornado para o módulo principal. As linhas 536 até 548 do Código Fonte A.2 implementam essa multiplicação.

- **Passo 13.** A matriz $Q_n \times P_n$ é multiplicada a matriz do número de espiras em cada enrolamento (*MatrixN* ou simplesmente *N*). O resultado $QPN = (Q_n \times P_n) \times N$ é retornado para a módulo principal (linhas 550 até 562).
- **Passo 14.** A matriz QPN é invertida pelo módulo *invertMatrices* (linhas 564 até 573) e o resultado QPN^{-1} é retornado para o módulo principal.
- **Passo 15.** Se a matriz QPN admitir inversa, a matriz QPN^{-1} é multiplicada é multiplicada a matriz B (Passo 11) e o resultado $Y_{ss} = QPN^{-1} \times B$ é retornado ao módulo principal (linhas 577 até 589).
- **Passo 16.** O módulo *multiplyMatrices* é utilizado para multiplicar as matrizes B e V (linhas 591 até 603). O resultado da operação $BV = B \times V$ é retornado para o módulo RLib.
- **Passo 17.** As matrizes BV e O são somadas pelo modelo *sumMatrices*. O resultado $BVO = BV + O$ é retornado para o módulo principal. As linhas 605 até 617 do Código Fonte A.2.
- **Passo 18.** Nesse passo, a matriz I_{ns} é calculada pela multiplicação de QPN^{-1} por BVO . É importante que o resultado $I_{ns} = QPN^{-1} \times BVO$ é uma das saídas da biblioteca RLib. O Código Fonte A.2 apresenta esse cálculo nas linhas 619 até 631.
- **Passo 19.** A matriz Y_{ss} é multiplicada por V e o resultado $YV = Y_{ss} \times V$ é retornado para o módulo principal. O trecho das linhas 633 até 645 do Código Fonte A.2 implementa essa multiplicação de matrizes.
- **Passo 20.** Nesse passo, a segunda e última saída do algoritmo RLib é calculado pela soma da matriz YV e I_{ns} . O cálculo da matriz $I_s = YV + I_{ns}(\delta(t - 1))$ é realizado nas linhas 647 até 659 do Código Fonte A.2 (a partir da linha 172).

4.2.3 Integração de Modelos de RNS com a Biblioteca RLIB

A integração entre modelos do RNS e a biblioteca RLIB é apresentada na Figura 31. Para utilização da biblioteca é necessário que se prepare as matrizes de entrada do modelo. Em seguida a biblioteca deve ser chamada e o resultado salvo nas respectivas matrizes de saída. A forma de utilização da biblioteca pode ser observada no Código Fonte A.3.

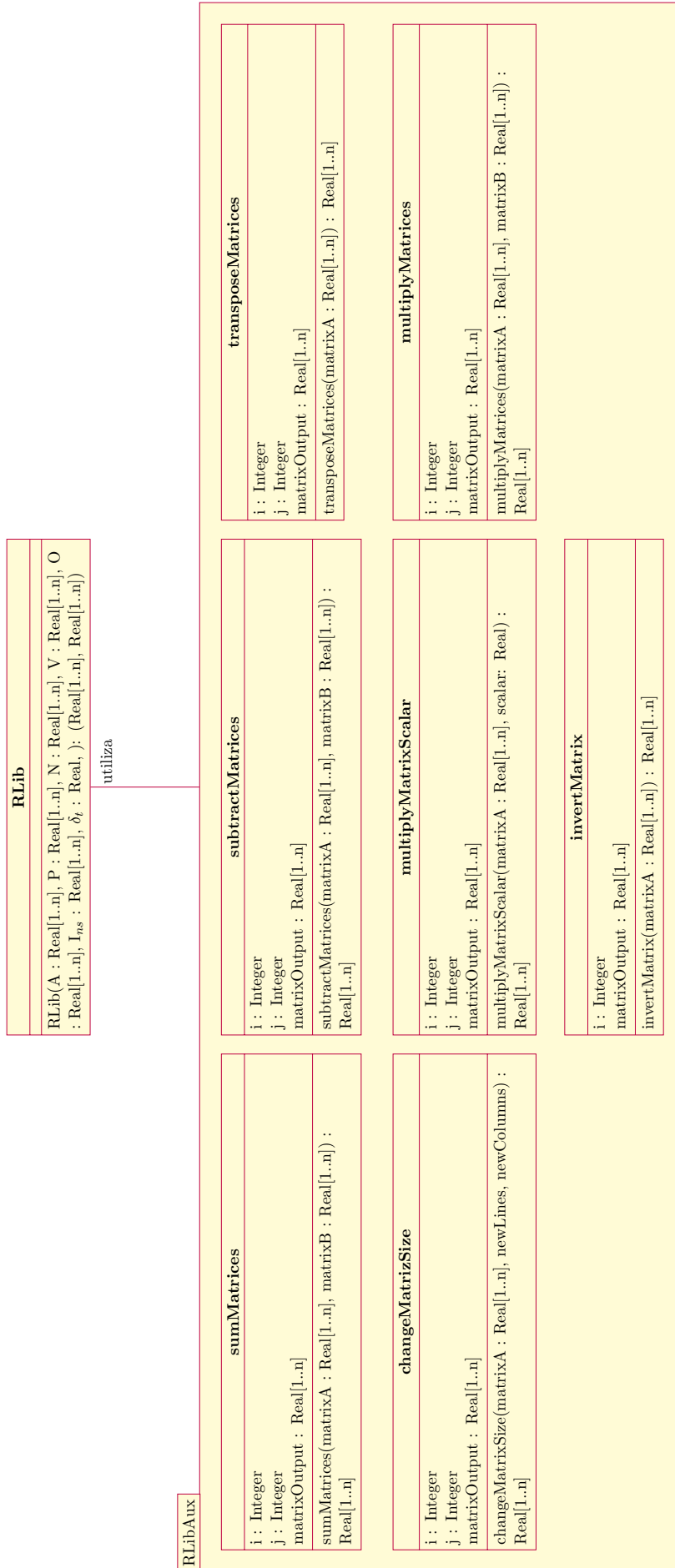


Figura 29 – Biblioteca RLib.

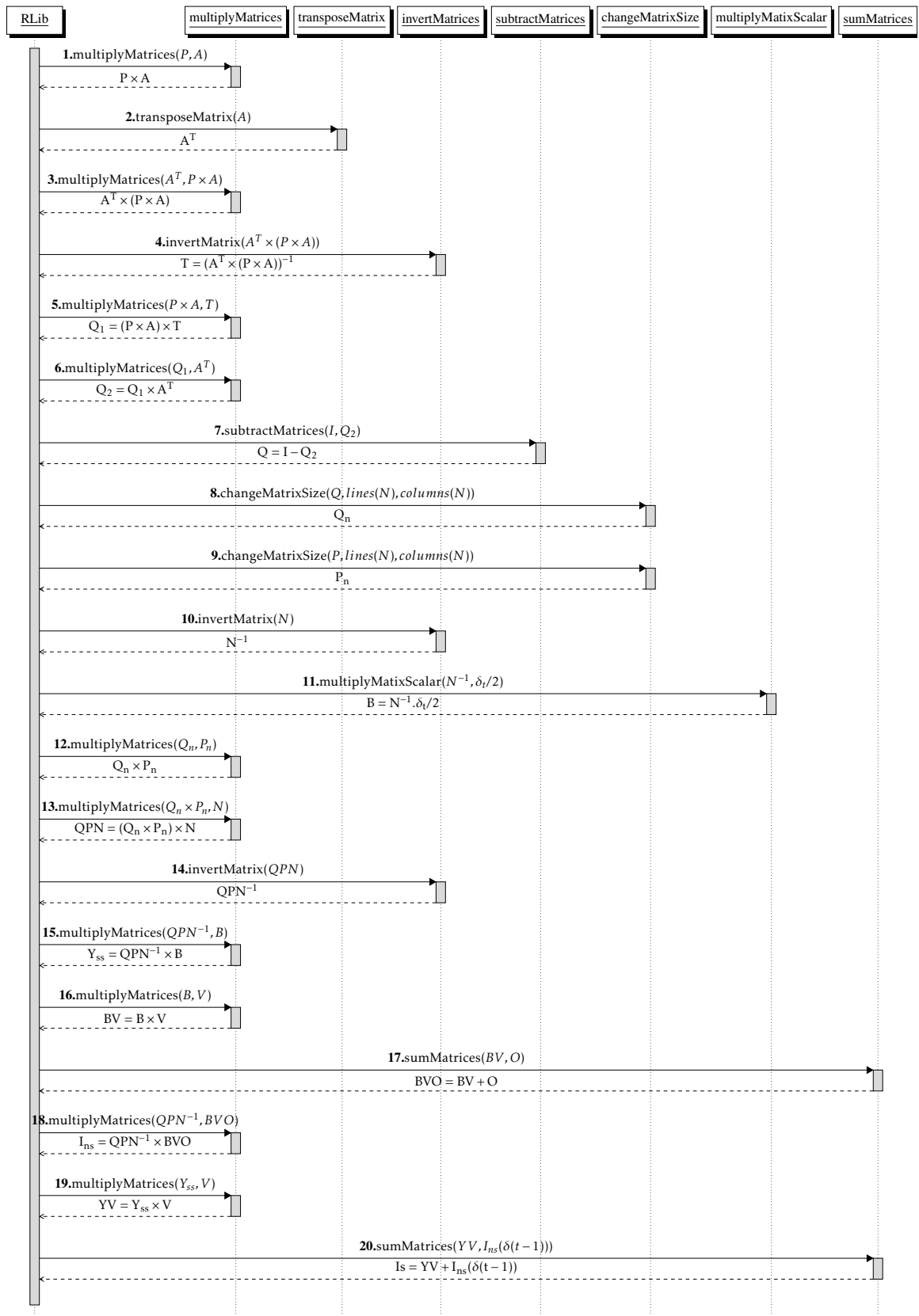


Figura 30 – Fluxo de chamadas da biblioteca RLib.

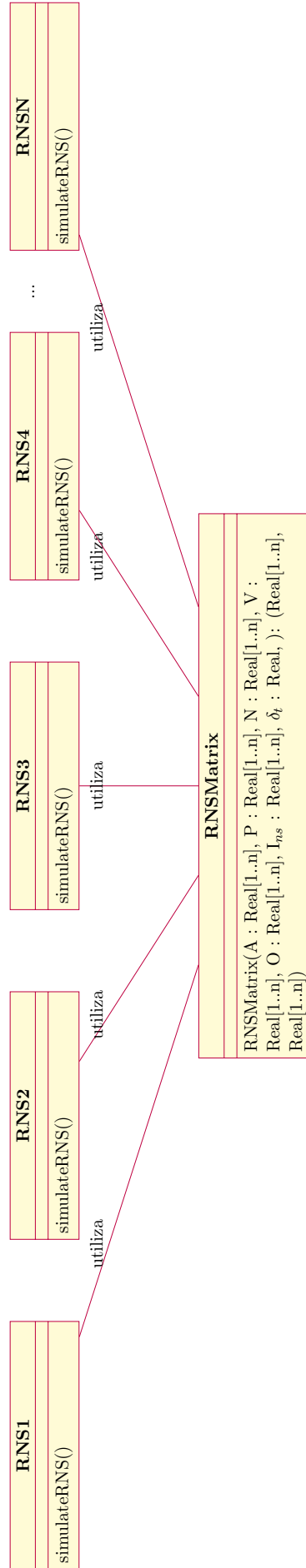


Figura 31 – Integração entre a biblioteca RLib e diferentes implementações de reatores naturalmente saturados.

5 Estudos de Caso

Neste capítulo, serão mostradas simulações dos reatores descritos no capítulo de modelagem. Os modelos criados foram desenvolvidos utilizando a linguagem MODEL no programa de transitórios ATP.

5.1 RNS monofásico com um enrolamento

Para o reator em questão, foram realizados dois experimentos. Primeiramente, foi analisado um reator com comportamento ideal. Em seguida, foram consideradas perdas nos enrolamentos e fluxos dispersos na simulação do reator.

O reator saturado é composto pelos pontos da curva de magnetização e seus valores são mostrados em anexo no Código Fonte A.3. Este algoritmo mostra como foi construída a modelagem do reator naturalmente saturado sem as perdas, utilizando a linguagem MODEL no programa de transitórios ATP.

Os dados de entrada de reatores naturalmente saturados são mostrados na Figura 32.

Figura 32 – Dados de Entrada do RNS1.

Dados de entrada	
Área do enrolamento (arEnro)	0.454 (m ²)
Área do jugo (arJugo)	0.454 (m ²)
Área de retorno (arReto)	0.454 (m ²)
Comprimento do enrolamento	3.59 (m)
Comprimento do jugo (compJg)	2.66 (m)
Comprimento de retorno (compRt)	3.59 (m)
Número de espiras (nuEsp)	65
Indutância de dispersão(indDis)	0 (mH)
Tensão de Entrada (V1)	22KV

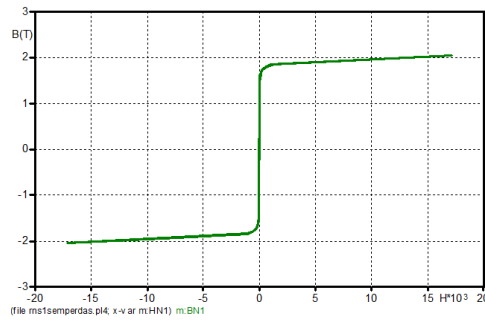
Fonte: Próprio Autor

A curva $B \times H$ utilizada como entrada para o RNS monofásico com um enrolamento (com e sem perdas) é obtida a partir da descrição do material ferromagnético utilizado no experimento, mostrada na Figura 33.

A Figura 34 apresenta o comportamento da corrente de saída do RNS sem perdas. Os fluxos magnéticos do reator são mostrados na Figura 35, e são identificados por fluxos nos enrolamentos (O1), fluxo no jugo (O2) e fluxo de retorno (O3). Como esperado, o fluxo disperso (representado por O4 e distribuído na cor rosa da figura) tem valor zero, pois este modelo não considera perdas por fluxos dispersos.

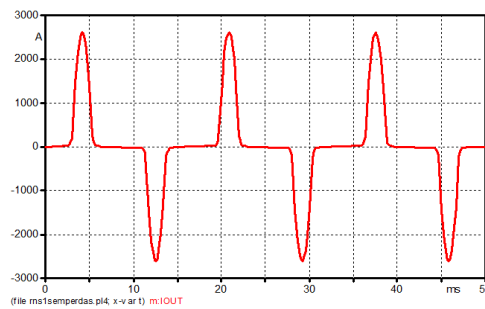
Para âmbito de comparação de resultados utilizou-se o componente Indutor não linear (referencial teórico 2.4) representado pelo elemento *TYPE98* da ferramenta *ATP* sem perdas de

Figura 33 – Curva BxH do reator monofásico saturado sem perdas.



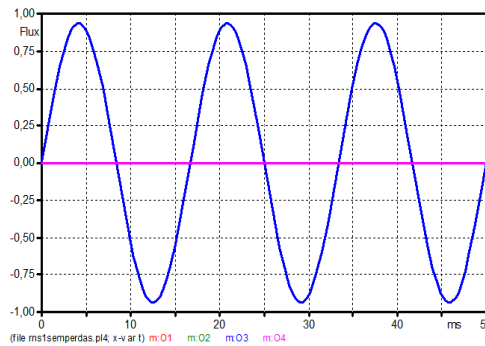
Fonte: Próprio Autor

Figura 34 – Corrente de saída do reator monofásico saturado sem perdas.



Fonte: Próprio Autor

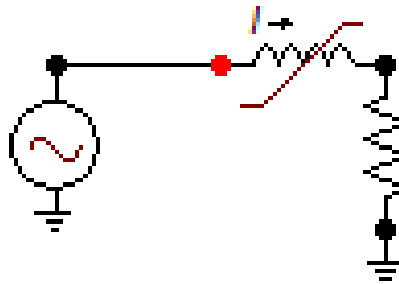
Figura 35 – Fluxos magnéticos do reator monofásico saturado sem perdas.



Fonte: Próprio Autor

resistência e indutância de dispersão. Este componente utiliza como entrada apenas a curva $\lambda \times i$ e seu respectivo modelo é mostrado no circuito da Figura 36.

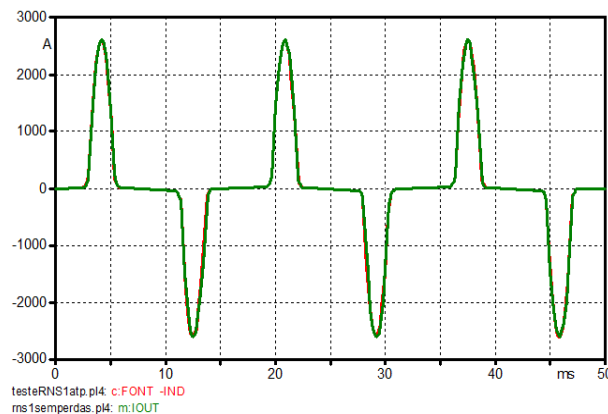
Figura 36 – Circuito indutor não linear no ATPdraw.



Fonte: Próprio Autor

A comparação entre as correntes de saída do modelo RNS monofásico (apresentado na cor verde) e o indutor não linear (apresentado na cor vermelha) é visualizada na Figura 37. Pode-se observar que a diferença entre as duas correntes é quase imperceptível. Portanto, assume-se que os resultados são equivalentes.

Figura 37 – Corrente de saída do modelo *UMEC* do reator monofásico sem perdas X Corrente de saída representado pelo Indutor não linear do *TYPE98*.

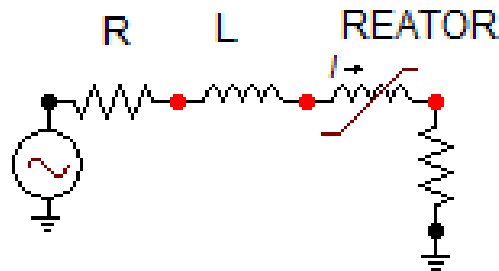


Fonte: Próprio Autor

Outra análise realizada para o modelo em questão foi a inclusão das perdas nos enrolamentos (perdas por efeito joule) e as perdas por fluxos dispersos descritos no algoritmo do Código Fonte A.4, através da inclusão do resistor em série nos terminais de entrada dos modelos.

O circuito desenvolvido no ATPdraw para comparação dos resultados do RNS com perdas contém uma resistência e uma indutância em série justamente para caracterizar as perdas (Figura 38). As simulações realizadas no modelo *UMEC* do RNS com perdas são mostradas a seguir. A inclusão das perdas no reator não afetam os fluxos magnéticos do ramo principal, jugo e retorno. Pois, a alteração no código do reator se deu em termos de fluxos dispersos. A Figura 39

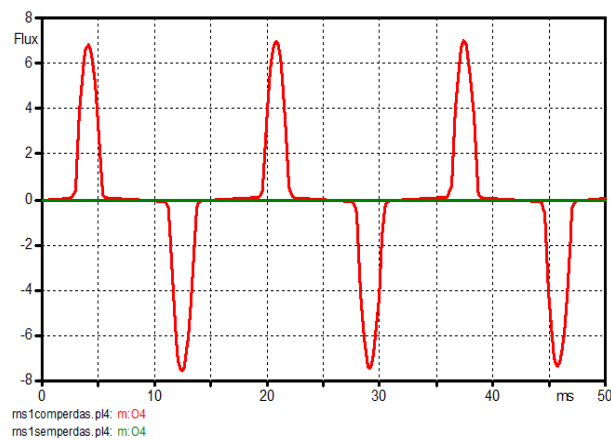
Figura 38 – Circuito indutor não linear do ATPdraw com perdas.



Fonte: Próprio Autor

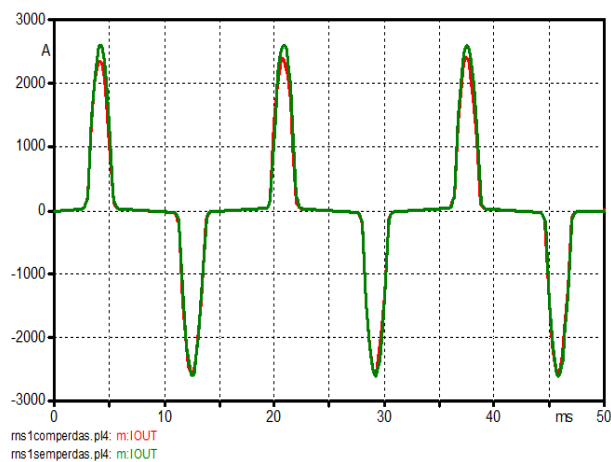
representa os fluxos dispersos de ambas as configurações na qual a linha verde zerada demonstra o reator sem perdas e a curva vermelha o reator com perdas de fluxos dispersos, sendo este comportamento completamente esperado.

Figura 39 – Fluxos dispersos do reator com perdas e Fluxos dispersos do reator sem perdas.



Fonte: Próprio Autor

Figura 40 – Correntes de reator com perdas e Correntes de reator sem perdas.



Fonte: Próprio Autor

A Figura 40 mostra o comparativo das correntes do reator com perdas (destacado em vermelho) e sem perdas (destacado na cor verde). E como esperado, o reator com perdas tem valor menor de corrente, pois tem menor tensão induzida no RNS e logo menor corrente. Como a curva $V \times I$ é acentuada, pequenas variações de tensões importam em grandes variações de corrente.

5.2 RNS monofásico com dois enrolamentos

Para simulação do reator saturado com dois enrolamentos (Figura 21) foram realizadas três análises listadas a seguir:

- Reator com ensaio de curto-circuito em 10% da tensão nominal;
- Reator com ensaio em vazio;
- Reator com ensaio em plena carga

No passo de tempo $1e-5$ segundos aplicado em cada caso. As modelagens realizadas para o reator são apresentadas no capítulo anterior e os respectivos códigos fontes (A.5, A.6 e A.7) são mostrados em anexo. Nos dados de entrada Figura 41 foram adicionados as indutâncias de dispersão dos enrolamentos primário e secundário, as tensões nos enrolamentos e as espiras dos mesmos. Os parâmetros utilizados para os ensaios são mostrados na Figura 42 e na Figura 43 o circuito montado no atpdraw, sendo V1 a tensão da fonte, R1 a resistência da fonte e RL a resistência na carga.

Figura 41 – Dados de Entrada do RNS2.

Dados de Entrada	
Área do enrolamento (arEnro)	0,454(m2)
Área do Jugo (arJugo)	0,454(m2)
Área de retorno (arReto)	0,454(m2)
Comprimento do enrolamento	3,59 (m)
Comprimento do Jugo	2,66 (m)
Comprimento do retorno	3,59 (m)
Numeros de espiras (N1)	65
Numeros de espiras (N2)	450
Indutância de dispersão (inDis1)	0,247167 (mH)
Indutância de dispersão (inDis2)	11,865 (mH)
tensão (V1)	16 KV
tensão (V2)	110KV

Fonte: Próprio Autor

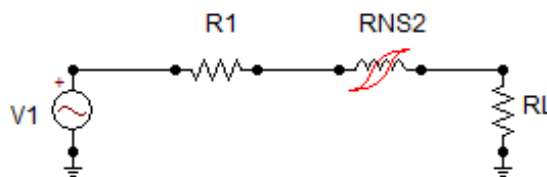
Os resultados dos modelos UMECs foram comparados com o componente transformador saturável do atpdraw (Figura 44). As formas de ondas das correntes dos enrolamentos primários

Figura 42 – parâmetros de testes utilizados no sistema.

ENSAIO	V1(KV)	RL(Ω)	R1(Ω)
Curto-circuito	1,6	0,0075	0,008
Plena Carga	16	64,5	0,008
Vazio	16	1,00E+06	0,107

Fonte: Próprio Autor

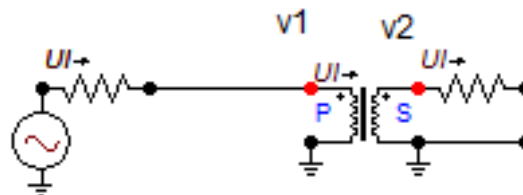
Figura 43 – Reator monofásico em testes.



Fonte: Próprio Autor

e secundários realizadas através dos ensaios mencionados anteriormente, são mostradas a seguir. Nas Figuras 45 e 46 são visualizadas os gráficos dos ensaios de curto-circuito realizados pelo RNS (na cor vermelha) e pelo transformador saturável (na cor verde) do atpdraw. Observa-se que ambos se comportam de forma equivalentes.

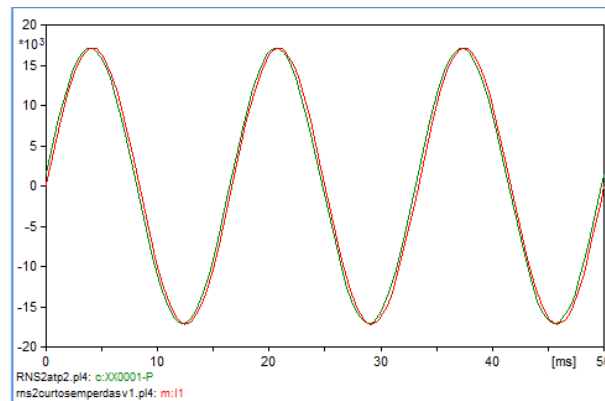
Figura 44 – Circuito do transformador saturável do atpdraw.



Fonte: Próprio Autor

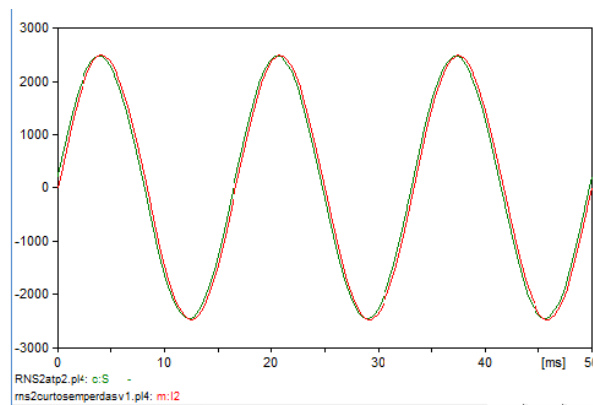
Nas Figuras 47 e 48 são visualizadas os gráficos dos ensaios em plena carga realizados pelo RNS e pelo transformador saturável do atpdraw. E por fim, as simulações comparativas dos ensaios em vazio realizados pelo RNS (na cor vermelha) e pelo transformador saturável (na cor verde) do atpdraw são mostradas nas Figuras 49 e 50.

Figura 45 – Comparativo de correntes primárias dos ensaios de curto-circuito do RNS e do transformador saturável do atpdraw.



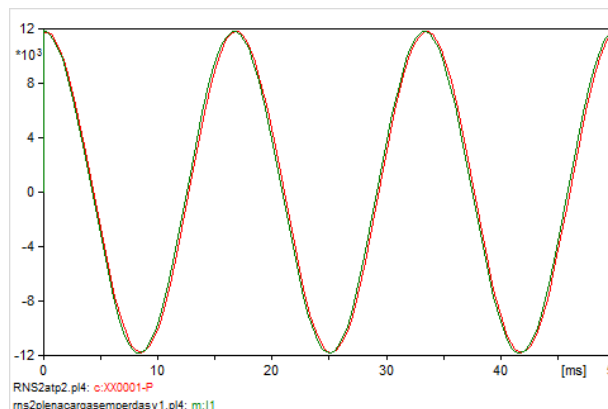
Fonte: Próprio Autor

Figura 46 – Comparativo de correntes secundárias dos ensaios de curto-circuito do RNS e do transformador saturável do atpdraw.



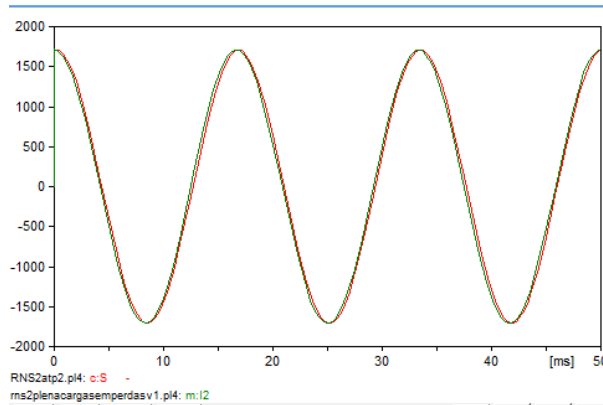
Fonte: Próprio Autor

Figura 47 – Comparativo de correntes primárias dos ensaios de plena carga do RNS e do transformador saturável do atpdraw.



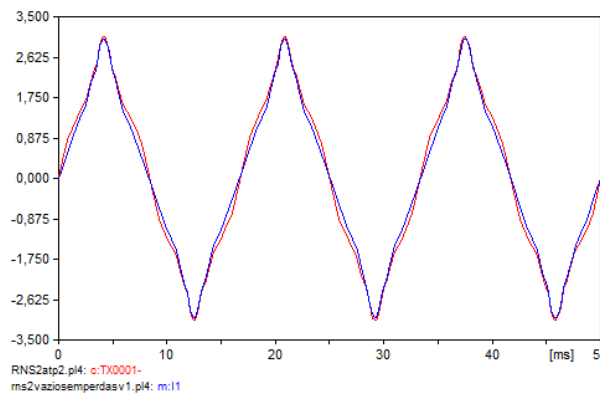
Fonte: Próprio Autor

Figura 48 – Comparativo de correntes psecundárias dos ensaios de plena carga do RNS e do transformador saturável do atpdraw.



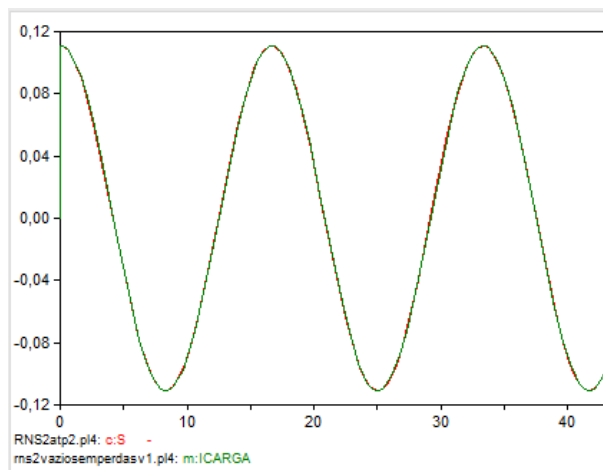
Fonte: Próprio Autor

Figura 49 – Comparativo de correntes primárias dos ensaios em vazio do RNS e do transformador saturável do atpdraw.



Fonte: Próprio Autor

Figura 50 – Comparativo de correntes secundárias dos ensaios em vazio do RNS e do transformador saturável do atpdraw.



Fonte: Próprio Autor

De acordo com os resultados mostrados nas figuras, fica claro que os comportamentos do RNS e do componente transformador saturável são equivalentes. Demonstrando que o RNS modelado na MODELS possui alto grau de confiabilidade e equivalência nos seus resultados.

5.3 RNS trifásico com seis enrolamentos

O reator trifásico de três núcleos e seis enrolamentos (Figura 23) possui ligação estrela-triângulo sendo modelado com 48 pontos da curva BxH. Primeiro, foi desenvolvido um reator sem perdas de histerese, depois, o laço de histerese foi adicionado, pois este está diretamente relacionado com a largura da curva BxH. A área do ciclo de histerese é representada durante a região de não saturação, e a relutância neste intervalo é praticamente constante. A curva de histerese foi inserida no modelo para ser possível medir perdas no núcleo. O código da modelagem deste reator pode ser visto no anexo em Código Fonte A.9. Os parâmetros de entrada para simulação desse reator é mostrado na Figura 51. As formas de onda das correntes são vistas na Figura 52.

Figura 51 – Parâmetros de entrada do reator de 6 enrolamentos.

Dados de Entrada	
Área do enrolamento (arEnro)	0.0158 (m ²)
Área do Jugo (arJugo)	0.0158 (m ²)
Área de retorno (arReto)	0.0158 (m ²)
Comprimento do enrolamento	3,59 (m)
Comprimento do Jugo	2,66 (m)
Comprimento do retorno	3,59 (m)
Numeros de espiras (N1)	195
Numeros de espiras (N2)	98
Indutância de dispersão (inDis1)	3.88E-4 (mH)
Indutância de dispersão (inDis2)	0.00678 (mH)
tensão (A,B,C)	2449.5 KV

Fonte: Próprio Autor

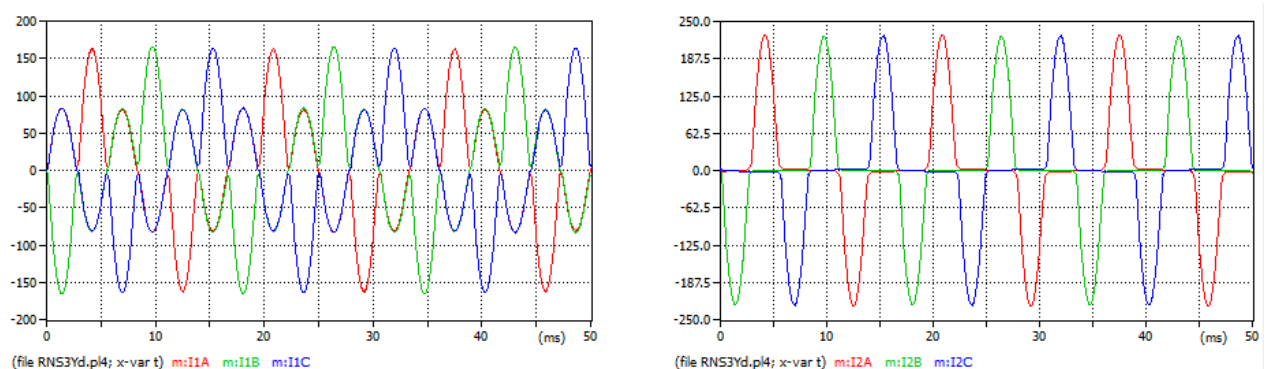
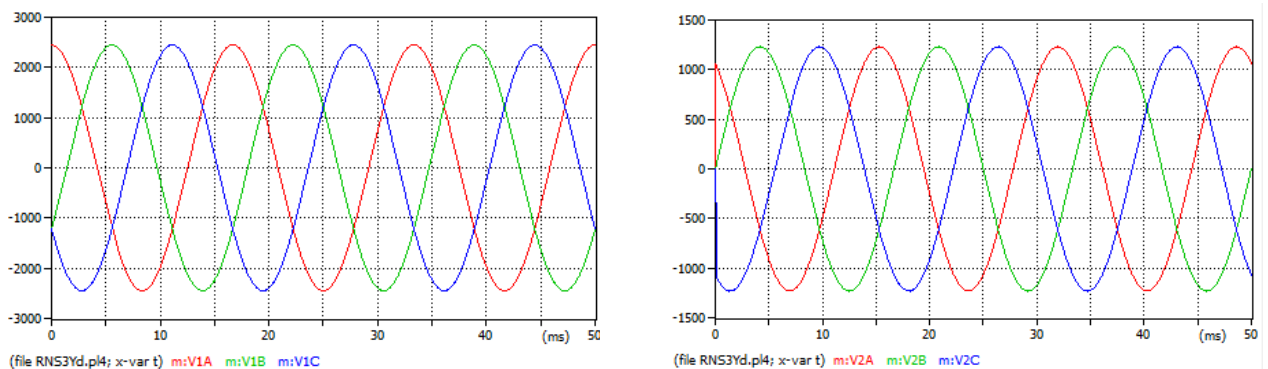


Figura 52 – Corrente trifásica nos enrolamentos primário e secundário do reator trifásico.
Fonte: Próprio Autor

As tensões trifásicas nos enrolamentos primários e secundários do reator de seis enrolamentos são mostradas na Figura 53. A corrente de fase A é vista na Figura 54 e o laço de histerese na Figura 55. Para âmbito de comparação com este reator, foi utilizado o transformador saturável do atpdraw representado na Figura 56.

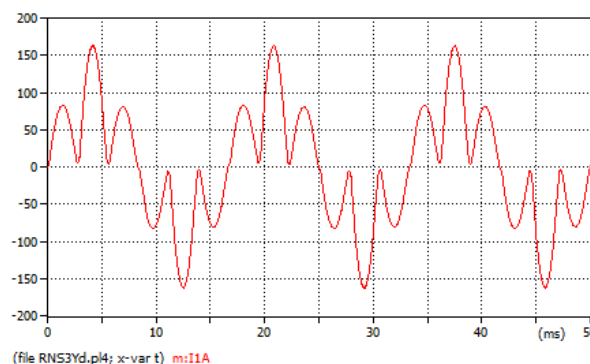
A base que norteia a utilização de um recurso pré-existente na plataforma ATP, como já falado, se apoia no fato que a estrutura construtiva de um reator saturado apresenta grande similaridade com a de transformadores.

Figura 53 – Tensão trifásica nos enrolamentos primário e secundário do reator trifásico.



Fonte: Próprio Autor

Figura 54 – Corrente na fase A no enrolamento primário do RNS3YD.



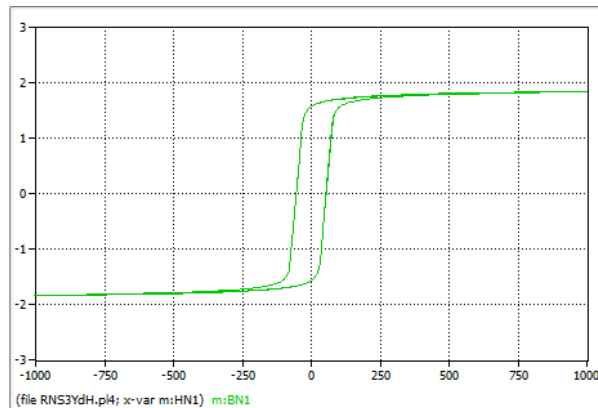
Fonte: Próprio Autor

A Figura 57 apresenta as curvas das correntes nas fases A, B e C do reator trifásico RNS3YD com seis enrolamentos e as correntes do transformador saturável. Os resultados mostraram uma diferença mínima de 4,14% das correntes na fase A, 4,06% nas correntes de fase B e 3,85% nas correntes de fase C. Isso deu a presença do retorno no UMEC. Demonstrando que o modelo desenvolvido neste trabalho é equivalente e possui resultados satisfatórios.

5.4 RNS trifásico com nove enrolamentos

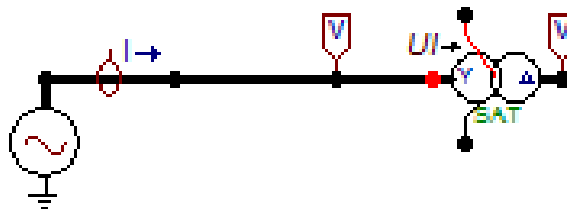
O reator naturalmente saturado com três núcleos e nove enrolamentos (Figura 25) foi configurado na ligação zigzag-delta como mostra o anexo de Código Fonte A.11. Também

Figura 55 – Laço de histerese do RNS3YD.



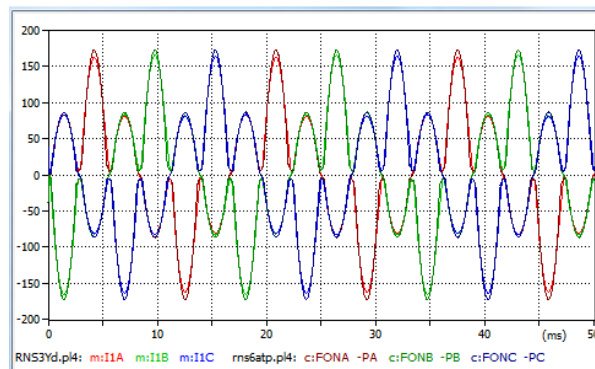
Fonte: Próprio Autor

Figura 56 – transformador saturável do atpdraw.



Fonte: Próprio Autor

Figura 57 – Corrente de fase A dos RNS3YD X Transformador saturável do ATPdraw.



Fonte: Próprio Autor

foi considerado os efeitos da histerese neste reator e possui os mesmos 48 pontos da curva BxH utilizado na configuração anterior. Os dados de entrada utilizados para a modelagem deste reator é mostrado na Figura 58.

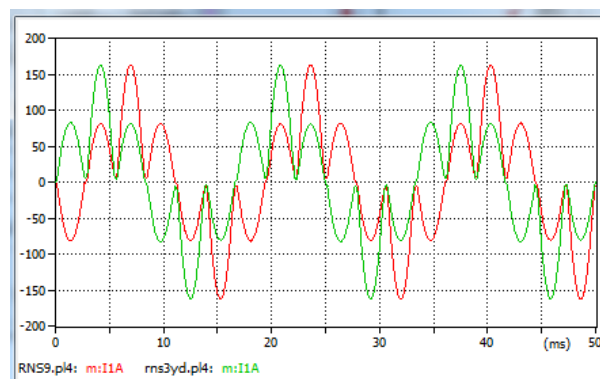
O reator em questão contém a mesma quantidade de núcleos do reator de seis enrolamentos, acrescentando um enrolamento para cada fase, totalizando 9 enrolamentos. O acréscimo de enrolamentos dentro do mesmo núcleo, não modificam amplitudes de corrente ou tensão.

Figura 58 – Parâmetros de entrada do reator RNS9.

Dados de Entrada	
Área do enrolamento (arEnro)	0.0158 (m ²)
Área do Jugo (arJugo)	0.0158 (m ²)
Área de retorno (arReto)	0.0158 (m ²)
Comprimento do enrolamento 1	0.29 (m)
Comprimento do enrolamento 2	0.154 (m)
Comprimento do enrolamento 3	0.27 (m)
Comprimento do Jugo	0.3608 (m)
Comprimento do retorno	0,8 (m)
Numeros de espiras (N1)	145
Numeros de espiras (N2)	77
Numeros de espiras (N3)	98
Indutância de dispersão (inDis1)	5.119E-5 (mH)
Indutância de dispersão (inDis2)	1.279E-4 (mH)
Indutância de dispersão (inDis3)	0.0108 (mH)
tensão (A,B,C)	2449.5 KV

Fonte: Próprio Autor

Figura 59 – Corrente de fase A dos reatores trifásicos RNS9 e o RNS3YD.



Fonte: Próprio Autor

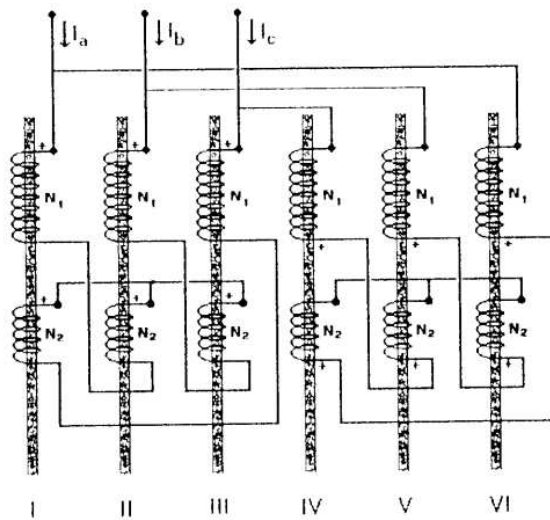
As modificações se dão nas defasagens angulares, pois para o reator de 6 enrolamentos as forças eletromotrizes são defasadas em $\pi/6$ uma das outras. Para o reator de 9 enrolamentos esta defasagem é $\pi/9$. A diferença angular entre os dois reatores representando a cor vermelha o reator com nove enrolamentos e na cor verde o de seis enrolamentos é visualizada na Figura 59. Como esperado, o reator proposto se desenvolveu como deveria com as respectivas defasagens e mesma amplitude. Concluindo que o modelo é válido e se comporta de forma satisfatória.

5.5 RNS trifásico em Monobloco com Doze Enrolamentos

O reator proposto possui configuração em Monobloco contendo 6 núcleos (I,II,III,IV,V,VI) e 12 enrolamentos (Figura 27). Nesta montagem existe duas unidades trifásicas independentes: A primeira compreende os núcleos I, II e III a segunda compreende os outros

três núcleos, operando em paralelo (Figura 60).

Figura 60 – Reator com ligação paralela dos enrolamentos.



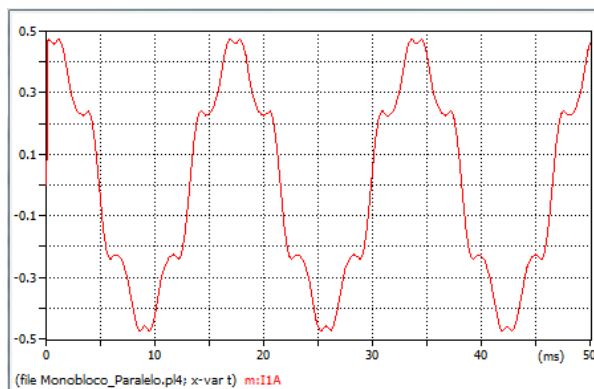
Fonte: (FONTE, 1997)

Nesta configuração as FMM's possui uma defasagem de $\pi/6$ radianos, isto acarretará a ocorrência de saturação de um núcleo para cada unidade trifásica simultaneamente. Para atender a exigência, cada núcleo permanece saturado por um período de $\pi/3$ radianos.

A cada semiciclo da tensão de alimentação, a saturação atinge uma vez qualquer um desses núcleos, nesse momento um pulso de corrente percorre o enrolamento montado no mesmo. Esta mesma corrente circulará em outro enrolamento de mesma fase pertencente ao núcleo que deveria operar isento de saturação. Seguindo esta linha de raciocínio, todas as fases terão correntes em formas de pulso.

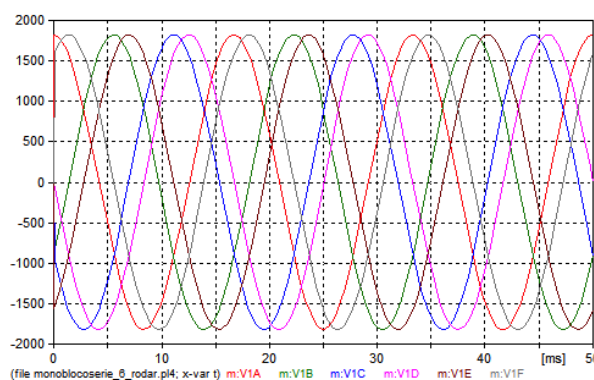
As correntes obtidas através da soma das contribuições das duas unidades trifásicas para a fase A, assinala a Figura 61. As tensões nos enrolamentos primários (V1A,V1B,V1C,V1D,V1E e V1F) do reator é mostrado na Figura 62. Observando as figuras, conclui-se que seus comportamentos estão conforme esperado para este tipo de reator.

Figura 61 – Corrente da fase A do RNS em Monobloco.



Fonte: Própria

Figura 62 – Tensões nos enrolamentos de entrada do RNS em Monobloco.



Fonte: Própria

6 Conclusão

O programa computacional ATP muito utilizado na engenharia e academia para simulações de transitórios eletromagnéticos em sistemas elétricos é uma ferramenta importante para análises de componentes não lineares, tais como, reatores saturados. O ATP possui uma biblioteca de componentes que podem ser utilizados diretamente pelo usuário para modelagem de circuitos elétricos. Contudo, é impossível representar todos os componentes elétricos existentes. Portanto, o programa ATP permite também que os usuários criem seus próprios modelos utilizando a linguagem MODELS na qual pode-se especificar e modificar os valores de parâmetros (numéricos e/ou analógicos) para controlar operação de componentes elétricos do sistema simulado. Na linguagem MODELS, é possível implementar grandezas magnéticas essenciais para análises de diversos componentes, tais como, fluxos magnéticos, permeâncias, intensidade de campo magnético, densidade magnética.

O processo de compensação de reativos é muito utilizado nos dias de hoje para encontrar a conformidade e qualidade dos padrões de tensão no sistema elétrico de potência. Algumas soluções, como compensadores eletrônicos são utilizados, porém, com custo bastante elevado e pouco tempo de vida. Uma outra alternativa para a compensação de reativos, seria a utilização de reatores naturalmente saturados, eles são mais baratos, possuem manutenção de baixo custo, elevada confiabilidade e equipamento de alta resistência a intempéries. No entanto, por se tratar de um componentes altamente especializado este componente não representa um componente pré-existente na biblioteca do ATP.

No intuito de se criar um modelo para reatores naturalmente saturados para o ATP, este trabalho propôs uma biblioteca de suporte a criação de reatores naturalmente saturados utilizando modelos de circuito equivalente magnético unificado (UMEC). Este modelo de circuito calcula o fluxo magnético no ramo em cada etapa de tempo, que determina a admitância equivalente que é adicionado com uma derivação entre os terminais do circuito equivalente de Norton para representar as perdas no núcleo magnético.

As contribuições do trabalho são listadas a seguir. Foi apresentado uma visão geral e importância das ferramentas ATP e linguagem MODELS. Além disso, foram destacados conceitos importantes relacionados aos reatores naturalmente saturados e como trabalhar com sua natureza não linear (método da linearização por trechos). A descrição matemática utilizando modelagem UMEC foi apresentada para diferentes tipos de RNSs. Uma vez apresentada a modelagem matemática dos reatores, foi descrito o código da biblioteca proposta e sua ligação com o referencial teórico. Por fim, um conjunto de estudos de caso foram apresentados para demonstrar a viabilidade da solução proposta.

6.1 Sugestões para trabalhos futuros

Com a crescente demanda de aplicações de reatores saturados para compensação de linhas de transmissão em grandes eixos de transferências de potência, esse assunto é muito estudado na busca de resultados mais eficientes. Algumas sugestões para trabalhos futuros são indicadas a seguir:

- Expandir a modelagem de UMECs utilizando a linguagem MODELS para projetar reatores com números genéricos de núcleos, diversos enrolamentos e diferentes configurações. Beneficiando-se da agilidade que a biblioteca desenvolvida neste trabalho proporciona para os projetos.
- Realizar uma análise ampla dos reatores, observando o comportamento das forças eletromotrizes e seus respectivos harmônicos.
- Aperfeiçoar a biblioteca desenvolvida na linguagem MODELS, para ampliação de simulação dos reatores saturados.
- Adaptar a biblioteca proposta para a linguagem C e comparar aspectos de performance na simulação de reatores entre a biblioteca proposta e sua respectiva versão escrita na linguagem C.

Referências

- ARAUJO, A. E. D.; NEVES, W. L. *Cálculos de transitórios eletromagnéticos em sistemas de energia*. [S.l.]: Editora UFMG, 2005. Citado 2 vezes nas páginas 15 e 36.
- BARBOSA, J. *Modelagem e Análise de Desempenho Transitório e Dinâmico de Compensadores Estáticos de Reativos Utilizando Reatores Saturados*. 2009. Citado na página 17.
- BARBOSA, J. et al. Proposta e validação de modelagem de reatores a núcleo saturado no simulador atp. 2010. Citado na página 17.
- BRASIL, D. d. O. C. do. *Aplicação de reatores saturados em sistemas de transmissão*. 1996. Citado 2 vezes nas páginas 15 e 16.
- COGO, J.; OLIVEIRA, A. de. Aplicação do método de integração trapezoidal em sistemas elétricos. *Seminário de Pesquisa EFEI*, 1983. Citado 2 vezes nas páginas 28 e 34.
- DELFIN, C. G. *Modelagem de reatores naturalmente saturados usando circuito equivalente magnético unificado em rotina da MODELS no ATP*. Tese (Doutorado) — Dissertação de Mestrado. Universidade Federal de Pernambuco, 2014. Citado na página 16.
- DUBÉ, L. Users guide to models in atp. *Bonneville Power Administration*, 1996. Citado 6 vezes nas páginas 22, 24, 27, 60, 61 e 63.
- FONTE, L. da. *Desenvolvimento de Reatores com Saturação Natural para Aplicação em Sistemas de Potência, Dr*. Tese (Doutorado) — UFPE, Recife, Brazil, 2004. Citado na página 17.
- FONTE, L. M. D. *Reatores Saturados: Alguns Aspectos Operacionais e de Projeto*. 1997. Citado 3 vezes nas páginas 14, 17 e 90.
- FREITAS, F. E. F. Aplicação série de reatores naturalmente saturados em sistemas de potência. Universidade Federal de Pernambuco, 2010. Citado na página 17.
- HERMANN, W. D. On modelling iron core nonlinearities. *IEEE Transactions on Power Systems*, v. 8, n. 2, 1993. Citado 2 vezes nas páginas 17 e 29.
- HORITA, M. A. B. *Reatores controlados por saturação para compensação de reativos*. Tese (Doutorado) — Universidade de São Paulo. Citado na página 31.
- KRAUSE, P.; AMBLER, S.; FOX, J. *ATP user manual*. [S.l.], 1993. Citado 2 vezes nas páginas 19 e 22.
- LIMA, T. *Lições de álgebra linear, 2ª Edição*. [s.n.], 2014. (Ensino). ISBN 9789892608549. Disponível em: <<https://books.google.com.br/books?id=qXaMCwAAQBAJ>>. Citado na página 67.
- LOBO, E. *Guia prático de engenharia de software*. Universo dos Livros Editora. ISBN 9788578730369. Disponível em: <<https://books.google.com.br/books?id=QMkSI0jtLV8C>>. Citado 2 vezes nas páginas 65 e 72.

MAIA, M. J. et al. Aplicação no sistema chesf da tecnologia de reatores saturados para limitação de corrente de curto-circuito: Simulações e ensaios. *IV Simpósio Brasileiro de Sistemas Elétricos*, 2012. Citado na página 15.

PACHECO, J. et al. Projeto e análise desempenho de um reator saturado de 570 kvar e 13, 8 kv para regulação de tensão. In: *VII Conferência Brasileira sobre Qualidade da Energia Elétrica-CBQEE, Santos-SP, Brasil*. [S.l.: s.n.], 2007. Citado na página 17.

PRESS, W. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN 9780521880688. Disponível em: <<https://books.google.com.br/books?id=1aAOdzK3FegC>>. Citado na página 70.

PRIKLER, L.; HOILDALEN, H. Atpdraw users' manual. *SINTEF Energy Research*, 2002. Citado 2 vezes nas páginas 20 e 22.

STRANG, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2003. ISBN 9780961408893. Disponível em: <<https://books.google.com.br/books?id=Gv4pCVyoUVYC>>. Citado 2 vezes nas páginas 70 e 71.

SWIFT, G. W. Power transformer core behavior under transient conditions. *IEEE Transactions on Power Apparatus and Systems*, IEEE, n. 5, p. 2206–2210, 1971. Citado na página 27.

TREFETHEN, L.; BAU, D. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1997. ISBN 9780898719574. Disponível em: <<https://books.google.com.br/books?id=JaPtxOytY7kC>>. Citado na página 70.

TURING, A. M. Rounding-off errors in matrix processes. *The Quarterly Journal of Mechanics and Applied Mathematics*, Oxford Univ Press, v. 1, n. 1, p. 287–308, 1948. Citado na página 70.

WATSON JOS ARRILLAGA, L. U. K. N. *Power Systems Electromagnetic Transients Simulation*. [S.l.]: Published by The Institution of Engineering and Technology, 2007. Citado 2 vezes nas páginas 29 e 30.

A Código Fonte da biblioteca RLIB

A.1 Código Fonte do Modelo para Inversão de Matrizes

```

1 MODEL invertMatrix
2   INPUT matrixInput[1..900], matrixDimension
3   VAR y[1..900], size, P[1..900], matrixI[1..900], i, j, k
4   matrixPivot[1..900], max_i, elem1, elem2, swap1, swap2, c
5   matrixPA[1..900], L[1..900], U[1..900], s, determinant,
6   Z[1..900], X[1..900], linesOut, columnsOut
7
8   OUTPUT
9     X[1..900], linesOut, columnsOut, determinant
10  EXEC
11
12  linesOut := matrixDimension
13  columnsOut := matrixDimension
14
15  y[1..900] := matrixInput[1..900]
16  size:=matrixDimension
17
18  matrixI[1..900] := 0
19  matrixPA[1..900] := 0
20  Z[1..900] := 0
21  X[1..900] := 0
22  L[1..900] := 0
23  U[1..900] := 0
24
25
26  -----
27  -- CREATE IDENTITY MATRIX
28  -----
29  FOR i:=1 TO size DO
30    matrixI[(i-1)*size + i]:=1
31  ENDFOR
32
33  -----
34  -- CREATE PIVOT MATRIX
35  -----
36  matrixPivot[1..900] := matrixI[1..900]
37
38  FOR j :=1 TO size DO
39    max_i := j
40
41    FOR i := j TO size DO
42      elem1 := abs(y[(i-1)*size + j])
43      elem2 := abs(y[(max_i-1)*size + j])
44
45      IF elem1 > elem2 THEN
46        max_i := i
47      ENDIF
48    ENDFOR
49
50    i := j
51

```

```

52  IF i <> max_i THEN
53  FOR k :=1 TO size DO
54      swap1 := matrixPivot[(i-1)*size + k]
55      swap2 := matrixPivot[(max_i-1)*size + k]
56      matrixPivot[(max_i-1)*size + k] := swap1
57      matrixPivot[(i-1)*size+ k] := swap2
58  ENDFOR
59  ENDIF
60 ENDFOR
61
62 -----
63 -- MULTIPLY PIVOT BY THE INPUT MATRIX
64 -----
65 FOR i:=1 TO size DO
66     FOR j:=1 TO size DO
67         c := 0
68         FOR k:=1 TO size DO
69             c := c + matrixPivot[(i-1)*size + k]*y[(k-1)*size + j]
70         ENDFOR
71         matrixPA[(i-1)*size + j] := c
72     ENDFOR
73 ENDFOR
74
75 -----
76 -- PREPARE MATRICES L AND U
77 -----
78 FOR i:=1 TO size DO
79     FOR j:=1 TO size DO
80         IF i = j THEN L[(i-1)*size + j]:=1 ELSE L[(i-1)*size + j]:=0
81     ENDFOR
82 ENDFOR
83 ENDFOR
84
85 -----
86 -- PREPARE MATRICES L AND U
87 -----
88 FOR i:=1 TO size DO
89     FOR j:=1 TO size DO
90         IF i = j THEN U[(i-1)*size + j]:=1 ELSE U[(i-1)*size + j]:=0
91     ENDFOR
92 ENDFOR
93 ENDFOR
94
95 -----
96 -- Calculate L AND U
97 -----
98 FOR j := 1 TO size DO
99     FOR i :=1 TO size DO
100        IF i <= j THEN
101            s := 0
102            IF j > 1 THEN
103                FOR k:=1 TO j - 1 DO
104                    s := s + U[(k-1)*size + j]*L[(i-1)*size + k]
105                ENDFOR
106            ENDFOR
107            U[(i-1)*size + j] := matrixPA[(i-1)*size + j] - s
108        ENDFOR
109        IF i > j THEN
110            s := 0
111            IF i > 1 THEN

```

```

112     FOR k:=1 TO i - 1 DO
113         s := s + U[(k-1)*size + j]*L[(i-1)*size + k]
114     ENDFOR
115     ENDIF
116     L[(i-1)*size + j] := (matrixPA[(i-1)*size + j] - s)/U[(j-1)*size + j]
117     ENDIF
118     ENDFOR
119 ENDFOR
120
121
122 -----
123 -- Find the determinant
124 -----
125 determinant := 1
126 FOR i:=1 TO size DO
127     determinant := determinant * L[(i-1)*size + i]
128     determinant := determinant * U[(i-1)*size + i]
129 ENDFOR
130
131 IF determinant <> 0 THEN
132     -----
133     -- Calculating the matrix Z
134     -----
135     FOR i:=1 TO size DO
136         FOR j:=1 TO size DO
137             s := 0
138             FOR k:=1 TO size DO
139                 s := s + L[(i-1)*size + k]*Z[(k-1)*size + j]
140             ENDFOR
141             Z[(i-1)*size + j] := (matrixPivot[(i-1)*size + j] - s)/L[(i-1)*size + i];
142         ENDFOR
143     ENDFOR
144
145     -----
146     -- Calculating the matrix X
147     -----
148     FOR j:=1 TO size DO
149         X[(size-1)*size + j] := Z[(size-1)*size + j]/U[(size-1)*size + size];
150         IF size > 1 THEN
151             FOR i:=size-1 TO 1 BY (-1) DO
152                 s := 0
153                 FOR k:=i+1 TO size DO
154                     s := s + U[(i-1)*size + k]*X[(k-1)*size + j]
155                 ENDFOR
156                 X[(i-1)*size + j] := (Z[(i-1)*size + j] - s)/U[(i-1)*size + i];
157             ENDFOR
158         ENDIF
159     ENDFOR
160 ENDIF
161 ENDEXEC
162 ENDMODEL

```

Código Fonte A.1 – Módulo da biblioteca RLib para inversão de matrizes

A.2 Código Fonte da Biblioteca RLib

```

2  INPUT deltaTime,
3  lMatrixA, cMatrixA, MatrixA[1..900],
4  lMP, cMP, MP[1..900]
5  lMatrixN, cMatrixN, MatrixN[1..900]
6  lMatrixV, cMatrixV, MatrixV[1..900]
7  lMatrixO, cMatrixO, MatrixO[1..900],
8  lMatrixInsIn, cMatrixInsIn, MatrixInsIn[1..900]
9  VAR mSize, dt, determnt, aux,
10 lMatrixP, cMatrixP, MatrixP[1..900]
11 lMatrixI, cMatrixI, MatrixI[1..900]
12 lMatrixPA, cMatrixPA, MatrixPA[1..900]
13 lMatrixTA, cMatrixTA, MatrixTA[1..900]
14 lMatrixTAPA, cMatrixTAPA, MatrixTAPA[1..900]
15 lMatrixITAPA, cMatrixITAPA, MatrixITAPA[1..900]
16 lMatrixQ1, cMatrixQ1, MatrixQ1[1..900]
17 lMatrixQ2, cMatrixQ2, MatrixQ2[1..900]
18 lMatrixQ, cMatrixQ, MatrixQ[1..900]
19 lMatrixINss, cMatrixINss, MatrixINss[1..900]
20 lMatrixTemp1, cMatrixTemp1, MatrixTemp1[1..900]
21 lMatrixQP, cMatrixQP, MatrixQP[1..900]
22 lMatrixQPN, cMatrixQPN, MatrixQPN[1..900]
23 lMatrixTemp2, cMatrixTemp2, MatrixTemp2[1..900]
24 lMatrixYss, cMatrixYss, MatrixYss[1..900]
25 lMatrixT1V, cMatrixT1V, MatrixT1V[1..900]
26 lMatrixT1VF, cMatrixT1VF, MatrixT1VF[1..900]
27 lMatrixIns, cMatrixIns, MatrixIns[1..900]
28 lMatrixYssV, cMatrixYssV, MatrixYssV[1..900]
29 lMatrixIs, cMatrixIs, MatrixIs[1..900]
30 OUTPUT
31 lMatrixIns, cMatrixIns, MatrixIns[1..900]
32 lMatrixIs, cMatrixIs, MatrixIs[1..900]
33
34 MODEL sumMatrices
35   INPUT linesA, columnsA, mmA[1..900]
36   linesB, columnsB, mmB[1..900]
37   VAR i, j, c, mmOut[1..900], linesOut, columnsOut
38   OUTPUT
39     mmOut[1..900], linesOut, columnsOut
40   INIT
41     i := 0, j := 0, c := 0, mmOut[1..900] := 0
42   ENDINIT
43   EXEC
44
45   linesOut := linesA, columnsOut := columnsA
46
47   -----
48   -- SUM THE MATRICES
49   -----
50   FOR i:=1 TO linesA DO
51     FOR j:=1 TO columnsA DO
52       mmOut[(i-1)*columnsA + j] := mmA[(i-1)*columnsA + j]+mmB[(i-1)*columnsB + j]
53     ENDFOR
54   ENDFOR
55   ENDEXEC
56 ENDMODEL
57
58 MODEL subtractMatrices
59   INPUT linesA, columnsA, mmA[1..900]
60   linesB, columnsB, mmB[1..900]
61   VAR i, j, c, mmOut[1..900], linesOut, columnsOut

```

```

62  OUTPUT
63  mmOut[1..900], linesOut, columnsOut
64  INIT
65  i := 0, j:= 0, c:= 0, mmOut[1..900]:=0
66  ENDINIT
67  EXEC
68
69  linesOut := linesA, columnsOut := columnsA
70
71  -----
72  -- SUM THE MATRICES
73  -----
74  FOR i:=1 TO linesA DO
75  FOR j:=1 TO columnsA DO
76  mmOut[(i-1)*columnsA + j] := mmA[(i-1)*columnsA + j]-mmB[(i-1)*columnsB + j]
77  ENDFOR
78  ENDFOR
79  ENDEXEC
80  ENDMODEL
81
82  MODEL transposeMatrix
83  INPUT linesA, columnsA, mmA[1..900]
84  VAR i, j, mmOut[1..900], linesOut, columnsOut
85  OUTPUT
86  mmOut[1..900], linesOut, columnsOut
87  INIT
88  i := 0, j:= 0, mmOut[1..900]:=0
89  ENDINIT
90  EXEC
91
92  linesOut := columnsA
93  columnsOut := linesA
94
95  -----
96  -- MULTIPLY THE MATRICES
97  -----
98  FOR i:=1 TO linesA DO
99  FOR j:=1 TO columnsA DO
100 mmOut[(j-1)*columnsOut + i] := mmA[(i-1)*columnsA + j]
101 ENDFOR
102 ENDFOR
103 ENDEXEC
104 ENDMODEL
105
106 MODEL changeMatrixSize
107 INPUT linesA, columnsA, mmA[1..900],
108 newLines, newColumns
109 VAR i, j, mmOut[1..900], linesOut, columnsOut
110 OUTPUT
111 mmOut[1..900], linesOut, columnsOut
112 INIT
113 i := 0, j:= 0, mmOut[1..900]:=0
114 ENDINIT
115 EXEC
116
117 linesOut := newLines
118 columnsOut := newColumns
119
120 -----
121 -- MULTIPLY THE MATRICES

```

```

122 -----
123 FOR i:=1 TO newLines DO
124   FOR j:=1 TO newColumns DO
125     mmOut[(i-1)*columnsOut + j] := mA[(i-1)*columnsA + j]
126   ENDFOR
127 ENDFOR
128 ENDEXEC
129 ENDMODEL
130
131 MODEL multiplyMatrixScalar
132   INPUT linesA, columnsA, mA[1..900],
133     scalar
134   VAR i, j, mmOut[1..900], linesOut, columnsOut
135   OUTPUT
136     mmOut[1..900], linesOut, columnsOut
137   INIT
138     i := 0, j:= 0, mmOut[1..900]:=0
139   ENDINIT
140   EXEC
141
142   linesOut := linesA
143   columnsOut := columnsA
144
145 -----
146 -- MULTIPLY THE MATRICES
147 -----
148 FOR i:=1 TO linesA DO
149   FOR j:=1 TO columnsA DO
150     mmOut[(i-1)*columnsOut + j] := mA[(i-1)*columnsA + j]*scalar
151   ENDFOR
152 ENDFOR
153 ENDEXEC
154 ENDMODEL
155
156 MODEL printMatrix
157   INPUT linesA, columnsA, mA[1..900]
158   VAR i, j
159   EXEC
160     write("*****")
161     FOR i:=1 TO linesA DO
162       FOR j:=1 TO columnsA DO
163         write(mA[(i-1)*columnsA + j])
164       ENDFOR
165       write("-----")
166     ENDFOR
167   ENDEXEC
168 ENDMODEL
169
170
171 MODEL multiplyMatrices
172   INPUT linesA, columnsA, mA[1..900]
173     linesB, columnsB, mB[1..900]
174   VAR i, j, c, k, mmOut[1..900], linesOut, columnsOut
175   OUTPUT
176     mmOut[1..900], linesOut, columnsOut
177   INIT
178     i := 0, j:= 0, c:= 0, k:=0, mmOut[1..900]:=0
179   ENDINIT
180   EXEC
181

```

```

182 linesOut := linesA
183 columnsOut := columnsB
184
185 -----
186 -- MULTIPLY THE MATRICES
187 -----
188 FOR i:=1 TO linesA DO
189     FOR j:=1 TO columnsB DO
190         c := 0
191         FOR k:=1 TO linesB DO
192             c := c + mmA[(i-1)*columnsA + k]*mmB[(k-1)*columnsB + j]
193         ENDFOR
194         mmOut[(i-1)*columnsB + j] := c
195     ENDFOR
196 ENDFOR
197 ENDEXEC
198 ENDMODEL
199
200 MODEL invertMatrix
201     INPUT matrixInput[1..900], matrixDimension
202     VAR y[1..900], size, P[1..900], matrixI[1..900], i, j, k
203     matrixPivot[1..900], max_i, elem1, elem2, swap1, swap2, c
204     matrixPA[1..900], L[1..900], U[1..900], s, determinant,
205     Z[1..900], X[1..900], linesOut, columnsOut
206
207     OUTPUT
208         X[1..900], linesOut, columnsOut, determinant
209     EXEC
210
211     linesOut := matrixDimension
212     columnsOut := matrixDimension
213
214     y[1..900] := matrixInput[1..900]
215     size:=matrixDimension
216
217     matrixI[1..900] := 0
218     matrixPA[1..900] := 0
219     Z[1..900] := 0
220     X[1..900] := 0
221     L[1..900] := 0
222     U[1..900] := 0
223
224
225 -----
226 -- CREATE IDENTITY MATRIX
227 -----
228 FOR i:=1 TO size DO
229     matrixI[(i-1)*size + i]:=1
230 ENDFOR
231
232 -----
233 -- CREATE PIVOT MATRIX
234 -----
235 matrixPivot[1..900] := matrixI[1..900]
236
237 FOR j :=1 TO size DO
238     max_i := j
239
240     FOR i := j TO size DO
241         elem1 := abs(y[(i-1)*size + j])

```

```

242     elem2 := abs(y[(max_i-1)*size + j])
243
244     IF elem1 > elem2 THEN
245         max_i := i
246     ENDIF
247 ENDFOR
248
249 i := j
250
251 IF i <> max_i THEN
252     FOR k :=1 TO size DO
253         swap1 := matrixPivot[(i-1)*size + k]
254         swap2 := matrixPivot[(max_i-1)*size + k]
255         matrixPivot[(max_i-1)*size + k] := swap1
256         matrixPivot[(i-1)*size+ k] := swap2
257     ENDFOR
258 ENDIF
259 ENDFOR
260
261 -----
262 -- MULTIPLY PIVOT BY THE INPUT MATRIX
263 -----
264 FOR i:=1 TO size DO
265     FOR j:=1 TO size DO
266         c := 0
267         FOR k:=1 TO size DO
268             c := c + matrixPivot[(i-1)*size + k]*y[(k-1)*size + j]
269         ENDFOR
270         matrixPA[(i-1)*size + j] := c
271     ENDFOR
272 ENDFOR
273
274 -----
275 -- PREPARE MATRICES L AND U
276 -----
277 FOR i:=1 TO size DO
278     FOR j:=1 TO size DO
279         IF i = j THEN L[(i-1)*size + j]:=1 ELSE L[(i-1)*size + j]:=0
280     ENDIF
281 ENDFOR
282 ENDFOR
283
284 -----
285 -- PREPARE MATRICES L AND U
286 -----
287 FOR i:=1 TO size DO
288     FOR j:=1 TO size DO
289         IF i = j THEN L[(i-1)*size + j]:=1 ELSE L[(i-1)*size + j]:=0
290     ENDIF
291 ENDFOR
292 ENDFOR
293
294 -----
295 -- Calculate L AND U
296 -----
297 FOR j := 1 TO size DO
298     FOR i :=1 TO size DO
299         IF i <= j THEN
300             s := 0
301             IF j > 1 THEN

```



```

302     FOR k:=1 TO j - 1 DO
303         s := s + U[(k-1)*size + j]*L[(i-1)*size + k]
304     ENDFOR
305     ENDIF
306     U[(i-1)*size + j] := matrixPA[(i-1)*size + j] - s
307     ENDIF
308     IF i > j THEN
309         s := 0
310         IF i > 1 THEN
311             FOR k:=1 TO i - 1 DO
312                 s := s + U[(k-1)*size + j]*L[(i-1)*size + k]
313             ENDFOR
314             ENDIF
315             L[(i-1)*size + j] := (matrixPA[(i-1)*size + j] - s)/U[(j-1)*size + j]
316         ENDIF
317     ENDFOR
318 ENDFOR
319
320 -----
321 -- Find the determinant
322 -----
323
324 determinant := 1
325 FOR i:=1 TO size DO
326     determinant := determinant * L[(i-1)*size + i]
327     determinant := determinant * U[(i-1)*size + i]
328 ENDFOR
329
330 IF determinant <> 0 THEN
331     -----
332     -- Calculating the matrix Z
333     -----
334     FOR i:=1 TO size DO
335         FOR j:=1 TO size DO
336             s := 0
337             FOR k:=1 TO size DO
338                 s := s + L[(i-1)*size + k]*Z[(k-1)*size + j]
339             ENDFOR
340             Z[(i-1)*size + j] := (matrixPivot[(i-1)*size + j] - s)/L[(i-1)*size + i];
341         ENDFOR
342     ENDFOR
343
344 -----
345 -- Calculating the matrix X
346 -----
347     FOR j:=1 TO size DO
348         X[(size-1)*size + j] := Z[(size-1)*size + j]/U[(size-1)*size + size];
349         IF size > 1 THEN
350             FOR i:=size-1 TO 1 BY (-1) DO
351                 s := 0
352                 FOR k:=i+1 TO size DO
353                     s := s + U[(i-1)*size + k]*X[(k-1)*size + j]
354                 ENDFOR
355                 X[(i-1)*size + j] := (Z[(i-1)*size + j] - s)/U[(i-1)*size + i];
356             ENDFOR
357         ENDIF
358     ENDFOR
359 ENDIF
360 ENDEXEC
361 ENDMODEL

```

```
362
363
364 EXEC
365
366 mSize:=900
367 dt := deltaTime
368
369 lMatrixP := lMP
370 cMatrixP := cMP
371 MatrixP[1..mSize] := MP[1..mSize]
372
373 lMatrixI := lMatrixP
374 cMatrixI := lMatrixI
375 MatrixI[1..mSize]:=0
376
377 FOR i:=1 TO cMatrixI DO
378     MatrixI[(i-1)*cMatrixI + i]:=1
379 ENDFOR
380
381 lMatrixIns := lMatrixO
382 cMatrixIns := cMatrixO
383 MatrixIns[1..900] := 0
384
385 lMatrixIs := lMatrixO
386 cMatrixIs := cMatrixO
387 MatrixIs[1..900] := 0
388
389 USE multiplyMatrices as mm2
390 INPUT
391 linesA := lMatrixP,
392 columnsA := cMatrixP,
393 mmA[1..mSize] := MatrixP[1..mSize]
394 linesB := lMatrixA,
395 columnsB := cMatrixA,
396 mmB[1..mSize] := MatrixA[1..mSize]
397 OUTPUT
398 lMatrixPA := linesOut,
399 cMatrixPA:= columnsOut,
400 MatrixPA[1..mSize] := mmOut[1..mSize]
401 ENDUSE
402
403 USE transposeMatrix as tml
404 INPUT
405 linesA := lMatrixA,
406 columnsA := cMatrixA,
407 mmA[1..mSize] := MatrixA[1..mSize]
408 OUTPUT
409 lMatrixTA := linesOut,
410 cMatrixTA:= columnsOut,
411 MatrixTA[1..mSize] := mmOut[1..mSize]
412 ENDUSE
413
414 USE multiplyMatrices as mm2
415 INPUT
416 linesA := lMatrixTA,
417 columnsA := cMatrixTA,
418 mmA[1..mSize] := MatrixTA[1..mSize]
419 linesB := lMatrixPA,
420 columnsB := cMatrixPA,
421 mmB[1..mSize] := MatrixPA[1..mSize]
```

```

422  OUTPUT
423  lMatrixTAPA := linesOut,
424  cMatrixTAPA:= columnsOut,
425  MatrixTAPA[1..mSize] := mmOut[1..mSize]
426  ENDUSE
427
428  USE invertMatrix as im1
429  INPUT
430  matrixInput[1..mSize] := MatrixTAPA[1..mSize],
431  matrixDimension := lMatrixTAPA
432  OUTPUT
433  lMatrixITAPA := linesOut,
434  cMatrixITAPA:= columnsOut,
435  MatrixITAPA[1..mSize] := X[1..mSize],
436  determnt := determinant
437  ENDUSE
438
439  IF determnt <> 0 THEN
440    --write("Calculou! 1", lMatrixITAPA, cMatrixITAPA)
441    USE multiplyMatrices as mm3
442    INPUT
443    linesA := lMatrixPA,
444    columnsA := cMatrixPA,
445    mmA[1..mSize] := MatrixPA[1..mSize]
446    linesB := lMatrixITAPA,
447    columnsB := cMatrixITAPA,
448    mmB[1..mSize] := MatrixITAPA[1..mSize]
449    OUTPUT
450    lMatrixQ1 := linesOut,
451    cMatrixQ1:= columnsOut,
452    MatrixQ1[1..mSize] := mmOut[1..mSize]
453    ENDUSE
454
455    USE multiplyMatrices as mm4
456    INPUT
457    linesA := lMatrixQ1,
458    columnsA := cMatrixQ1,
459    mmA[1..mSize] := MatrixQ1[1..mSize]
460    linesB := lMatrixTA,
461    columnsB := cMatrixTA,
462    mmB[1..mSize] := MatrixTA[1..mSize]
463    OUTPUT
464    lMatrixQ2 := linesOut,
465    cMatrixQ2:= columnsOut,
466    MatrixQ2[1..mSize] := mmOut[1..mSize]
467    ENDUSE
468
469    USE subtractMatrices as sm1
470    INPUT
471    linesA := lMatrixI,
472    columnsA := cMatrixI,
473    mmA[1..mSize] := MatrixI[1..mSize]
474    linesB := lMatrixQ2,
475    columnsB := cMatrixQ2,
476    mmB[1..mSize] := MatrixQ2[1..mSize]
477    OUTPUT
478    lMatrixQ := linesOut,
479    cMatrixQ:= columnsOut,
480    MatrixQ[1..mSize] := mmOut[1..mSize]
481    ENDUSE

```

```

482
483     USE changeMatrixSize as cms1
484 INPUT
485     linesA := lMatrixQ,
486     columnsA := cMatrixQ,
487     mmA[1..mSize] := MatrixQ[1..mSize]
488     newLines := lMatrixN,
489     newColumns := lMatrixN,
490 OUTPUT
491     lMatrixQ := linesOut,
492     cMatrixQ:= columnsOut,
493     MatrixQ[1..mSize] := mmOut[1..mSize]
494     ENDUSE
495
496 USE changeMatrixSize as cms2
497 INPUT
498     linesA := lMatrixP,
499     columnsA := cMatrixP,
500     mmA[1..mSize] := MatrixP[1..mSize]
501     newLines := lMatrixN,
502     newColumns := lMatrixN,
503 OUTPUT
504     lMatrixP := linesOut,
505     cMatrixP:= columnsOut,
506     MatrixP[1..mSize] := mmOut[1..mSize]
507     ENDUSE
508
509 USE invertMatrix as im2
510 INPUT
511     matrixInput[1..mSize] := MatrixN[1..mSize],
512     matrixDimension := lMatrixN
513 OUTPUT
514     lMatrixINss := linesOut,
515     cMatrixINss:= columnsOut,
516     MatrixINss[1..mSize] := X[1..mSize],
517     determnt := determinant
518     ENDUSE
519
520 IF determnt <> 0 THEN
521     --write("Calculou! 2", lMatrixINss, cMatrixINss)
522     aux := dt/2
523
524 USE multiplyMatrixScalar as me1
525 INPUT
526     linesA := lMatrixINss,
527     columnsA := cMatrixINss,
528     mmA[1..mSize] := MatrixINss[1..mSize]
529     scalar := aux
530 OUTPUT
531     lMatrixTemp1 := linesOut,
532     cMatrixTemp1:= columnsOut,
533     MatrixTemp1[1..mSize] := mmOut[1..mSize]
534     ENDUSE
535
536 USE multiplyMatrices as mm5
537 INPUT
538     linesA := lMatrixQ,
539     columnsA := cMatrixQ,
540     mmA[1..mSize] := MatrixQ[1..mSize]
541     linesB := lMatrixP,

```

```

542     columnsB := cMatrixP,
543     mmB[1..mSize] := MatrixP[1..mSize]
544     OUTPUT
545     lMatrixQP := linesOut,
546     cMatrixQP:= columnsOut,
547     MatrixQP[1..mSize] := mmOut[1..mSize]
548     ENDUSE
549
550     USE multiplyMatrices as mm6
551     INPUT
552     linesA := lMatrixQP,
553     columnsA := cMatrixQP,
554     mA[1..mSize] := MatrixQP[1..mSize]
555     linesB := lMatrixN,
556     columnsB := cMatrixN,
557     mmB[1..mSize] := MatrixN[1..mSize]
558     OUTPUT
559     lMatrixQPN := linesOut,
560     cMatrixQPN:= columnsOut,
561     MatrixQPN[1..mSize] := mmOut[1..mSize]
562     ENDUSE
563
564     USE invertMatrix as im3
565     INPUT
566     matrixInput[1..mSize] := MatrixQPN[1..mSize],
567     matrixDimension := lMatrixQPN
568     OUTPUT
569     lMatrixTemp2 := linesOut,
570     cMatrixTemp2:= columnsOut,
571     MatrixTemp2[1..mSize] := X[1..mSize],
572     determnt := determinant
573     ENDUSE
574
575     IF determnt <> 0 THEN
576         --write("Calculou!")
577         USE multiplyMatrices as mm7
578         INPUT
579         linesA := lMatrixTemp2,
580         columnsA := cMatrixTemp2,
581         mA[1..mSize] := MatrixTemp2[1..mSize]
582         linesB := lMatrixTemp1,
583         columnsB := cMatrixTemp1,
584         mmB[1..mSize] := MatrixTemp1[1..mSize]
585         OUTPUT
586         lMatrixYss := linesOut,
587         cMatrixYss:= columnsOut,
588         MatrixYss[1..mSize] := mmOut[1..mSize]
589         ENDUSE
590
591         USE multiplyMatrices as mm8
592         INPUT
593         linesA := lMatrixTemp1,
594         columnsA := cMatrixTemp1,
595         mA[1..mSize] := MatrixTemp1[1..mSize]
596         linesB := lMatrixV,
597         columnsB := cMatrixV,
598         mmB[1..mSize] := MatrixV[1..mSize]
599         OUTPUT
600         lMatrixT1V := linesOut,
601         cMatrixT1V:= columnsOut,

```

```
602     MatrixT1V[1..mSize] := mmOut[1..mSize]
603     ENDUSE
604
605     USE sumMatrices as sm1
606     INPUT
607     linesA := lMatrixT1V,
608     columnsA := cMatrixT1V,
609     mmA[1..mSize] := MatrixT1V[1..mSize]
610     linesB := lMatrixO,
611     columnsB := cMatrixO,
612     mmB[1..mSize] := MatrixO[1..mSize]
613     OUTPUT
614     lMatrixT1VF := linesOut,
615     cMatrixT1VF:= columnsOut,
616     MatrixT1VF[1..mSize] := mmOut[1..mSize]
617     ENDUSE
618
619     USE multiplyMatrices as mm8
620     INPUT
621     linesA := lMatrixTemp2,
622     columnsA := cMatrixTemp2,
623     mmA[1..mSize] := MatrixTemp2[1..mSize]
624     linesB := lMatrixT1VF,
625     columnsB := cMatrixT1VF,
626     mmB[1..mSize] := MatrixT1VF[1..mSize]
627     OUTPUT
628     lMatrixIns := linesOut,
629     cMatrixIns:= columnsOut,
630     MatrixIns[1..mSize] := mmOut[1..mSize]
631     ENDUSE
632
633     USE multiplyMatrices as mm9
634     INPUT
635     linesA := lMatrixYss,
636     columnsA := cMatrixYss,
637     mmA[1..mSize] := MatrixYss[1..mSize]
638     linesB := lMatrixV,
639     columnsB := cMatrixV,
640     mmB[1..mSize] := MatrixV[1..mSize]
641     OUTPUT
642     lMatrixYssV := linesOut,
643     cMatrixYssV:= columnsOut,
644     MatrixYssV[1..mSize] := mmOut[1..mSize]
645     ENDUSE
646
647     USE sumMatrices as sm2
648     INPUT
649     linesA := lMatrixYssV,
650     columnsA := cMatrixYssV,
651     mmA[1..mSize] := MatrixYssV[1..mSize]
652     linesB := lMatrixInsIn,
653     columnsB := cMatrixInsIn,
654     mmB[1..mSize] := MatrixInsIn[1..mSize]
655     OUTPUT
656     lMatrixIs := linesOut,
657     cMatrixIs:= columnsOut,
658     MatrixIs[1..mSize] := mmOut[1..mSize]
659     ENDUSE
660     ENDIF
661     ENDIF
```

```

662  ENDIF
663  ENDEXEC
664  ENDMODEL

```

Código Fonte A.2 – Módulo da biblioteca RLib

A.3 Codigo de Reatores Naturalmente Saturados

```

1  BEGIN NEW DATA CASE
2  C -----
3  C Generated by ATPDRAW novembro, sábado 7, 2015
4  C A Bonneville Power Administration program
5  C by H. K. Hidalen at SEFAS/NTNU - NORWAY 1994-2009
6  C -----
7  POWER FREQUENCY                60.
8  C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9    1.E-5      .05
10   500      1      1      1      1      0      0      1      0
11 /MODELS
12 MODELS
13 INPUT
14 MM0001 {v(XX0001)}
15 MM0002 {v(XX0002)}
16 OUTPUT
17   XX0002
18 MODEL RNS1
19 INPUT inpuV1      -- Tensao de Entrada V1
20       inpuV2      -- Tensao de Entrada V2
21 DATA arEnro      -- Area Enrolamento
22       arJugo      -- Area do Jugo
23       arReto      -- Area de Retorno
24       nuEspir     -- Numero de Espiras
25       compEr      -- Comprimento do Braco de Enrolamento
26       compJg      -- Comprimento do Braco do Jugo
27       compRt      -- Comprimento do Braco de Retorno
28       indDis      -- Indutancia de dispersao
29 VAR B[1..10]      -- Campo magnetico dos nove trechos
30     H[1..10]      -- Densidade magnetica para os nove trechos
31     BN[1..3]      -- Campo magnetico para o ramo principal, jugo e retorno
32     HN[1..3]      -- Densidade Magnetica para o ramo principal, jugo e retorno
33     BN1, BN2, BN3 -- Valores finais para campo magnetico para o ramo principal, jugo e retorno
34     HN1, HN2, HN3 -- Valores finais para densidade Magnetica para o ramo principal, jugo e retorno
35     l[1..9]       -- Indutancias para os nove trechos
36     lp[2..9]      -- Indutancias linhas (l') para os nove trechos
37     x, y          -- Variaveis intermediarias para facilitar o calculo de H e B
38     P1,P2,P3,P4   -- Permeancias do UMEC
39     O1,O2,O3,O4   -- Fluxos do UMEC (Ramo Principal, Jugo, Jugo e dispersao)
40     m2a,m2b,m2c,dm2 -- Variaveis intermediarias das matrizes
41     m311,m312,m321 -- Variaveis intermediarias das matrizes
42     m322,m511,m512 -- Variaveis intermediarias das matrizes
43     m11,y211,y311,y11 -- Variaveis intermediarias das matrizes
44     Ins1          -- Corrente da fonte de corrente do Equivalente de Norton
45     Iout          -- Corrente de saida
46     v11          -- Tensao acumulada na entrada 1
47     v12          -- Tensao acumulada na entrada 2
48     v1           -- Diferenca de v11 - v12
49     -- Essa parte corresponde as entidades da biblioteca RLib

```

```

50     m, dt, determnt, aux,
51     lMtrxA, cMtrxA, MtrxA[1..900]
52     lMtrxP, cMtrxP, MtrxP[1..900]
53     lMtrxN, cMtrxN, MtrxN[1..900]
54     lMtrxV, cMtrxV, MtrxV[1..900]
55     lMtrxO, cMtrxO, MtrxO[1..900]
56     lMtrxI, cMtrxI, MtrxI[1..900]
57     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
58     lMtrxIns, cMtrxIns, MtrxIns[1..900]
59     lMtrxIs, cMtrxIs, MtrxIs[1..900]
60 $INCLUDE ./RLib.m
61 OUTPUT Iout
62 INIT
63     BN1:=0.01, BN2:=0.01, BN3:=0.01 -- Inicializacao dos valores do campo magnetico
64     HN1:=0.01, HN2:=0.01, HN3:=0.01 -- Inicializacao dos valores da densidade magnetica
65     HN[1..3]:=0.01, BN[1..3]:=0.01 -- Inicializacao dos valores do campo magnetico e densidade magnetica
66     integral (inpuV1) :=0
67     integral (inpuV2) :=0
68     histdef (integral (inpuV1)) :=0
69     histdef (integral (inpuV2)) :=0
70     Ins1:=0
71     Iout:=0
72     m:=900
73     lMtrxP := 4,
74     cMtrxP := 4,
75     MtrxP[1..m] :=0
76 ENDINIT
77 EXEC
78     H[1]:=0 -- Definicao da curva BxH do material ferro magnetico
79     B[1]:=0
80     H[2]:=20.292
81     B[2]:=0.9
82     H[3]:=24.987
83     B[3]:=1.2
84     H[4]:=37.242
85     B[4]:=1.45
86     H[5]:=58.887
87     B[5]:=1.58
88     H[6]:=119.366
89     B[6]:=1.68
90     H[7]:=226.796
91     B[7]:=1.74
92     H[8]:=612.747
93     B[8]:=1.81
94     H[9]:=1193.662
95     B[9]:=1.86
96     H[10]:=10000
97     B[10]:=1.97
98
99     -- Metodo de aproximacao linear por trechos
100    -- Esse loop calcula a indutancia de cada trecho
101    FOR i:=1 TO 9 DO
102        l[i] := (B[i+1]-B[i]) / (H[i+1]-H[i])
103    ENDFOR
104
105    -- Metodo de aproximacao linear por trechos parte 2
106    -- Segundo loop para calcular o l' para cada trecho
107    -- Utilizar Lp' para criar indutores equivalentes que facam
108    -- o mesmo efeito na indutancia total.
109

```



```

110  FOR i:=2 TO 9 DO
111      lp[i]:=1/((1/l[i])-(1/l[i-1]))
112  ENDFOR
113
114  -- Calculo do campo magnetico no ramo principal
115  vl1:=integral(inpuV1)
116  vl2:=integral(inpuV2)
117  BN[1]:=(vl1-vl2)/(arEnro*nuEspis*(0.96))
118
119  IF BN[1]>=0 THEN          -- Calculo de H para os trechos do sistema
120      HN[1]:=BN[1]/l[1]
121      FOR i:=2 TO 9 DO
122          IF BN[1]>B[i] THEN
123              HN[1]:=HN[1]+((BN[1]-B[i])/lp[i])
124          ENDIF
125      ENDFOR
126  ELSE
127      HN[1]:=BN[1]/l[1]
128      FOR i:=2 TO 9 DO
129          IF BN[1]<=-B[i] THEN
130              HN[1]:=HN[1]+((BN[1]+B[i])/lp[i])
131          ENDIF
132      ENDFOR
133  ENDIF
134
135  P1:=(BN[1]/HN[1])*arEnro/compEr      -- Permeancia do ramo principal
136  O1:=BN[1]*arEnro                    -- Fluxo no ramo principal
137  P4:=2*indDis/(nuEspis*nuEspis)     -- Permeancia de dispersao
138  O4:=P4*(HN[1]*compEr+(O1/P1))      -- Fluxo de dispersao
139  O3:=O1-O4                           -- Fluxo no Jugo
140  O2:=O3                               -- Fluxo no retorno
141  BN[3]:=O3/arJugo                    -- Campo magnetico no Jugo
142  BN[2]:=O2/arReto                    -- Campo magnetico de retorno
143
144  FOR i:=2 TO 3 DO
145      IF BN[i]>=0 THEN                -- Calculo de H para os trechos do sistema
146          HN[i]:=BN[i]/l[1]
147          FOR j:=2 TO 9 DO
148              IF BN[i]>B[j] THEN
149                  HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
150              ENDIF
151          ENDFOR
152      ELSE
153          HN[i]:=BN[i]/l[1]
154          FOR j:=2 TO 9 DO
155              IF BN[i]<=-B[j] THEN
156                  HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
157              ENDIF
158          ENDFOR
159      ENDIF
160  ENDFOR
161
162  P3:=(BN[3]/HN[3])*arJugo/compJg
163  P2:=(BN[2]/HN[2])*arReto/compRt
164
165  HN1:=HN[1]
166  BN1:=BN[1]
167  HN2:=HN[2]
168  BN2:=BN[2]
169  HN3:=HN[3]

```

```

170 BN3:=BN[3]
171
172 -- Essa parte corresponde ao calculo das matrizes RLib
173 m:=900
174 lMtrxA := 4
175 cMtrxA := 2
176 MtrxA[1..m]:=0
177 MtrxA[1..cMtrxA] :=[1, -1]
178 MtrxA[cMtrxA+1..2*cMtrxA] :=[0, 1]
179 MtrxA[2*cMtrxA+1..3*cMtrxA] :=[-1, 0]
180 MtrxA[3*cMtrxA+1..4*cMtrxA] :=[-1, 1]
181
182 lMtrxN := 1
183 cMtrxN := 1
184 MtrxN[1..m]:=0
185 MtrxN[0*cMtrxN+1] := nuEspI
186
187 lMtrxV := 1
188 cMtrxV := 1
189 MtrxV[1..m]:= 0
190 MtrxV[0*cMtrxV+1] := (inpuV1-inpuV2)
191
192 lMtrxO := 1
193 cMtrxO := 1
194 MtrxO[1..m]:=0
195 MtrxO[0*cMtrxO+1] := O1
196
197
198 lMtrxInsIn := 1
199 cMtrxInsIn := 1
200 MtrxInsIn[1..m]:=0
201 MtrxInsIn[0*cMtrxInsIn+1] := Ins1
202
203 dt := timestep
204
205 MtrxP[0*cMtrxP + 1] := P1
206 MtrxP[1*cMtrxP + 2] := P2
207 MtrxP[2*cMtrxP + 3] := P3
208 MtrxP[3*cMtrxP + 4] := P4
209
210 USE RNSMatrix as RNSM1
211 INPUT
212     deltaTime := dt,
213
214     lMatrixA := lMtrxA,
215     cMatrixA := cMtrxA,
216     MatrixA[1..m] := MtrxA[1..m]
217
218     lMP := lMtrxP,
219     cMP := cMtrxP,
220     MP[1..900] := MtrxP[1..m]
221
222     lMatrixN := lMtrxN,
223     cMatrixN := cMtrxN,
224     MatrixN[1..m] := MtrxN[1..m]
225
226     lMatrixV := lMtrxV,
227     cMatrixV := cMtrxV,
228     MatrixV[1..m] := MtrxV[1..m]
229

```



```

290 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
291 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
292   XX0001           1.           0
293   XX0002           .001         1
294 /SOURCE
295 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
296 14XX0001   2.2E4   60.           -1.   100.
297 60XX0002-1
298 /OUTPUT
299 BLANK MODELS
300 BLANK BRANCH
301 BLANK SWITCH
302 BLANK SOURCE
303 BLANK OUTPUT
304 BLANK PLOT
305 BEGIN NEW DATA CASE
306 BLANK

```

Código Fonte A.3 – Código ATP para reator naturalmente saturado com 1 enrolamento sem perdas

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW  abril, sexta-feira 12, 2013
4 C A Bonneville Power Administration program
5 C by H. K. Hidalen at SEFAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY           60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9   1.E-5   .05
10   500     1     1     1     1     0     0     1     0
11 /MODELS
12 MODELS
13 INPUT
14 MM0001 {v(XX0001)}
15 MM0002 {v(XX0002)}
16 OUTPUT
17   XX0002
18 MODEL RNS1
19 INPUT inpuV1           -- Tensao de Entrada V1
20       inpuV2           -- Tensao de Entrada V2
21 DATA arEnro           -- Area Enrolamento
22       arJugo           -- Area do Jugo
23       arReto           -- Area de Retorno
24       nuEspI           -- Numero de Espiras
25       compEr           -- Comprimento do Braco de Enrolamento
26       compJg           -- Comprimento do Braco do Jugo
27       compRt           -- Comprimento do Braco de Retorno
28       indDis           -- Indutancia de dispersao
29 VAR B[1..10]           -- Campo magnetico dos nove trechos
30       H[1..10]         -- Densidade magnetica para os nove trechos
31       BN[1..3]         -- Campo magnetico para o ramo principal, jugo e retorno
32       HN[1..3]         -- Densidade Magnetica para o ramo principal, jugo e retorno
33       BN1, BN2, BN3    -- Valores finais para campo magnetico para o ramo principal, jugo e retorno
34       HN1, HN2, HN3    -- Valores finais para densidade Magnetica para o ramo principal, jugo e retorno
35       l[1..9]          -- Indutancias para os nove trechos
36       lp[2..9]         -- Indutancias linhas (l') para os nove trechos
37       x, y             -- Variaveis intermediarias para facilitar o calculo de H e B
38       P1,P2,P3,P4      -- Permeancias do UMEC

```

```

39      O1,O2,O3,O4      -- Fluxos do UMEC (Ramo Principal, Jugo, Jugo e dispersao)
40      m2a,m2b,m2c,dm2 -- Variaveis intermediarias das matrizes
41      m311,m312,m321  -- Variaveis intermediarias das matrizes
42      m322,m511,m512  -- Variaveis intermediarias das matrizes
43      m11,y211,y311,y11 -- Variaveis intermediarias das matrizes
44      Ins1            -- Corrente da fonte de corrente do Equivalente de Norton
45      Iout           -- Corrente de saida
46      v11           -- Tensao acumulada na entrada 1
47      v12           -- Tensao acumulada na entrada 2
48      v1            -- Diferenca de v11 - v12
49      v1            -- Tensao de entrada no Reator = inpuV1 - ResistenciaInterna*Iout
50      m, dt, determnt, aux,
51      lMtrxA, cMtrxA, MtrxA[1..900]
52      lMtrxP, cMtrxP, MtrxP[1..900]
53      lMtrxN, cMtrxN, MtrxN[1..900]
54      lMtrxV, cMtrxV, MtrxV[1..900]
55      lMtrxO, cMtrxO, MtrxO[1..900]
56      lMtrxI, cMtrxI, MtrxI[1..900]
57      lMtrxIns, cMtrxIns, MtrxIns[1..900]
58      lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
59      lMtrxIs, cMtrxIs, MtrxIs[1..900]
60 $INCLUDE ./RLib.m
61 OUTPUT Iout
62 INIT
63      BN1:=0.01, BN2:=0.01, BN3:=0.01 -- Inicializacao dos valores do campo magnetico
64      HN1:=0.01, HN2:=0.01, HN3:=0.01 -- Inicializacao dos valores da densidade magnetica
65      HN[1..3]:=0.01, BN[1..3]:=0.01 -- Inicializacao dos valores do campo magnetico e densidade magnetica
66      integral(v1):=0
67      integral(inpuV2):=0
68      histdef(integral(v1)):=0
69      histdef(integral(inpuV2)):=0
70      Ins1:=0
71      Iout:=0
72      v1:=0
73 ENDINIT
74 EXEC
75      H[1]:=0      -- Definicao da curva BxH do material ferro magnetico
76      B[1]:=0
77      H[2]:=20.292
78      B[2]:=0.9
79      H[3]:=24.987
80      B[3]:=1.2
81      H[4]:=37.242
82      B[4]:=1.45
83      H[5]:=58.887
84      B[5]:=1.58
85      H[6]:=119.366
86      B[6]:=1.68
87      H[7]:=226.796
88      B[7]:=1.74
89      H[8]:=612.747
90      B[8]:=1.81
91      H[9]:=1193.662
92      B[9]:=1.86
93      H[10]:=10000
94      B[10]:=1.97
95
96      -- Metodo de aproximacao linear por trechos
97      -- Esse loop calcula a indutancia de cada trecho
98      FOR i:=1 TO 9 DO

```

```

99         l[i]:= (B[i+1]-B[i]) / (H[i+1]-H[i])
100     ENDFOR
101
102     -- Metodo de aproximacao linear por trechos parte 2
103     -- Segundo loop para calcular o l' para cada trecho
104     -- Utilizar lp' para criar indutores equivalentes que facam
105     -- o mesmo efeito na indutancia total.
106
107     FOR i:=2 TO 9 DO
108         lp[i]:= 1/((1/l[i])-(1/l[i-1]))
109     ENDFOR
110
111     -- Calculo do campo magnetico no ramo principal
112     v1:=inpuV1-0.17*Iout
113     v11:=integral(v1)
114     v12:=integral(inpuV2)
115     BN[1]:= (v11-v12) / (arEnro*nuEsp1*(0.96))
116
117     IF BN[1]>=0 THEN -- Calculo de H para os trechos do sistema
118         HN[1]:=BN[1]/l[1]
119         FOR i:=2 TO 9 DO
120             IF BN[1]>B[i] THEN
121                 HN[1]:=HN[1]+((BN[1]-B[i])/lp[i])
122             ENDIF
123         ENDFOR
124     ELSE
125         HN[1]:=BN[1]/l[1]
126         FOR i:=2 TO 9 DO
127             IF BN[1]<=B[i] THEN
128                 HN[1]:=HN[1]+((BN[1]+B[i])/lp[i])
129             ENDIF
130         ENDFOR
131     ENDIF
132
133     P1:= (BN[1]/HN[1])*arEnro/compEr -- Permeancia do ramo principal
134     O1:=BN[1]*arEnro -- Fluxo no ramo principal
135     P4:=2*indDis/(nuEsp1*nuEsp1) -- Permeancia de dispersao
136     O4:=P4*(HN[1]*compEr+(O1/P1)) -- Fluxo de dispersao
137     O3:=O1-O4 -- Fluxo no Jugo
138     O2:=O3 -- Fluxo no retorno
139     BN[3]:=O3/arJugo -- Campo magnetico no Jugo
140     BN[2]:=O2/arReto -- Campo magnetico de retorno
141
142     FOR i:=2 TO 3 DO
143         IF BN[i]>=0 THEN -- Calculo de H para os trechos do sistema
144             HN[i]:=BN[i]/l[1]
145             FOR j:=2 TO 9 DO
146                 IF BN[i]>B[j] THEN
147                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
148                 ENDIF
149             ENDFOR
150         ELSE
151             HN[i]:=BN[i]/l[1]
152             FOR j:=2 TO 9 DO
153                 IF BN[i]<=B[j] THEN
154                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
155                 ENDIF
156             ENDFOR
157         ENDIF
158     ENDFOR

```

```

159
160 P3:=(BN[3]/HN[3])*arJugo/compJg
161 P2:=(BN[2]/HN[2])*arReto/compRt
162 HN1:=HN[1]
163 BN1:=BN[1]
164 HN2:=HN[2]
165 BN2:=BN[2]
166 HN3:=HN[3]
167 BN3:=BN[3]
168
169 --Calculos das matrizes
170 m:=900
171 lMtrxA := 4
172 cMtrxA := 2
173 MtrxA[1..m]:=0
174 MtrxA[1..cMtrxA] :=[1, -1]
175 MtrxA[cMtrxA+1..2*cMtrxA] :=[0, 1]
176 MtrxA[2*cMtrxA+1..3*cMtrxA] :=[-1, 0]
177 MtrxA[3*cMtrxA+1..4*cMtrxA] :=[-1, 1]
178
179 lMtrxN := 1
180 cMtrxN := 1
181 MtrxN[1..m]:=0
182 MtrxN[0*cMtrxN+1] := nuEspI
183
184 lMtrxV := 1
185 cMtrxV := 1
186 MtrxV[1..m]:= 0
187 MtrxV[0*cMtrxV+1] := (v1-inpuV2)
188
189 lMtrxO := 1
190 cMtrxO := 1
191 MtrxO[1..m]:=0
192 MtrxO[0*cMtrxO+1] := O1
193
194
195 lMtrxInsIn := 1
196 cMtrxInsIn := 1
197 MtrxInsIn[1..m]:=0
198 MtrxInsIn[0*cMtrxInsIn+1] := Ins1
199
200 dt := timestep
201
202 lMtrxP := 4,
203 cMtrxP := 4,
204 MtrxP[1..m]:=0
205 MtrxP[0*cMtrxP + 1] := P1
206 MtrxP[1*cMtrxP + 2] := P2
207 MtrxP[2*cMtrxP + 3] := P3
208 MtrxP[3*cMtrxP + 4] := P4
209
210 USE RNSMatrix as RNSM1
211 INPUT
212     deltaTime := dt,
213
214     lMatrixA := lMtrxA,
215     cMatrixA := cMtrxA,
216     MatrixA[1..m] := MtrxA[1..m]
217
218     lMP := lMtrxP,

```

```

219      cMP := cMtrxP,
220      MP[1..900] := MtrxP[1..m]
221
222      lMatrixN := lMtrxN,
223      cMatrixN := cMtrxN,
224      MatrixN[1..m] := MtrxN[1..m]
225
226      lMatrixV := lMtrxV,
227      cMatrixV := cMtrxV,
228      MatrixV[1..m] := MtrxV[1..m]
229
230      lMatrixO := lMtrxO,
231      cMatrixO := cMtrxO,
232      MatrixO[1..m] := MtrxO[1..m]
233
234      lMatrixInsIn := lMtrxInsIn,
235      cMatrixInsIn := cMtrxInsIn,
236      MatrixInsIn[1..m] := MtrxInsIn[1..m]
237 OUTPUT
238      MtrxIns[1..m] := MatrixIns[1..m],
239      MtrxIs[1..m] := MatrixIs[1..m]
240 ENDUSE
241      Ins1 := MtrxIns[1]
242      Iout := MtrxIs[1]
243
244 ENDEXEC
245 ENDMODEL
246 RECORD
247      RNS1.HN1 AS HN1
248      RNS1.HN2 AS HN2
249      RNS1.HN3 AS HN3
250      RNS1.BN1 AS BN1
251      RNS1.BN2 AS BN2
252      RNS1.BN3 AS BN3
253      RNS1.P1 AS P1
254      RNS1.P2 AS P2
255      RNS1.P3 AS P3
256      RNS1.P4 AS P4
257      RNS1.O1 AS O1
258      RNS1.O2 AS O2
259      RNS1.O3 AS O3
260      RNS1.O4 AS O4
261      RNS1.m11 AS m11
262      RNS1.y11 AS y11
263      RNS1.v1 AS v1
264      RNS1.inpuV1 AS inpuV1
265      RNS1.inpuV2 AS inpuV2
266      RNS1.v1 AS v1
267      RNS1.Ins1 AS Ins1
268      RNS1.Iout AS Iout
269      RNS1.v11 AS v11
270      RNS1.v12 AS v12
271 USE RNS1 AS RNS1
272 INPUT
273      inpuV1:= MM0001
274      inpuV2:= MM0002
275 DATA
276      arEnro:= 0.454
277      arJugo:= 0.454
278      arReto:= 0.454

```



```

279 nuEsp1:=      65.
280 compEr:=      3.59
281 compJg:=      2.66
282 compRt:=      3.59
283 indDis:= 1.236E-4
284 OUTPUT
285 XX0002:=Iout
286 ENDUSE
287 ENDMODELS
288 C      1      2      3      4      5      6      7      8
289 C 34567890123456789012345678901234567890123456789012345678901234567890
290 /BRANCH
291 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
292 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
293 XX0001      1.      0
294 XX0002      .001      1
295 /SOURCE
296 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
297 14XX0001      2.2E4      60.      -1.      100.
298 60XX0002-1      1.E3
299 /OUTPUT
300 BLANK MODELS
301 BLANK BRANCH
302 BLANK SWITCH
303 BLANK SOURCE
304 BLANK OUTPUT
305 BLANK PLOT
306 BEGIN NEW DATA CASE
307 BLANK

```

Código Fonte A.4 – Código ATP para reator naturalmente saturado com 1 enrolamento e perdas

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW novembro, sabado 7, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hoidalen at SEfAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY      60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9 1.E-5      .05
10 500      1      1      1      1      0      0      1      0
11 /MODELS
12 MODELS
13 INPUT
14 MM0001 {v{XX0001}}
15 MM0002 {i{XX0003}}
16 OUTPUT
17 XX0002
18 MODEL RNS2
19 INPUT inputV      -- Tensao de Entrada
20      iCarga      -- Corrente na Carga
21 DATA arEnro      -- Area Enrolamento
22      arJugo      -- Area do Jugo
23      nuEsp1      -- Numero de Espiras enrolamento 1
24      nuEsp2      -- Numero de Espiras enrolamento 2
25      compEr      -- Comprimento do Braco de Enrolamento
26      compJg      -- Comprimento do Braco do Jugo
27      Ld1      -- Indutancia de dispersao no enrolamento 1
28      Ld2      -- Indutancia de dispersao no enrolamento 2

```

```

29     rFonte                -- Resistencia da fonte
30 VAR B[1..10]             -- Campo magnetico dos nove trechos
31     H[1..10]             -- Densidade magnetica para os nove trechos
32     HN[1..3]
33     BN[1..3]
34     HN1
35     BN1
36     HN2
37     BN2
38     BN3
39     HN3
40     l[1..9]              -- Indutancias para os nove trechos
41     lp[2..9]            -- Indutancias linhas (l') para os nove trechos
42     x, y, z, w, xl, yl
43     d1
44     d2
45     P1
46     P2
47     P3
48     P4
49     P5
50     O1
51     O2
52     O3
53     O4
54     O5
55     ins1
56     ins2
57     i1
58     i2
59     v2
60     dy1
61     vl1
62     vl2
63     io1
64     Zc
65     Xc
66     Pc
67     iLp
68     v2p
69     v1p
70     io1c
71     ZL
72     e2
73     e1
74     v1
75     m, dt, determnt, aux,
76     lMtrxA, cMtrxA, MtrxA[1..900]
77     lMtrxP, cMtrxP, MtrxP[1..900]
78     lMtrxN, cMtrxN, MtrxN[1..900]
79     lMtrxV, cMtrxV, MtrxV[1..900]
80     lMtrxO, cMtrxO, MtrxO[1..900]
81     lMtrxI, cMtrxI, MtrxI[1..900]
82     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
83     lMtrxIns, cMtrxIns, MtrxIns[1..900]
84     lMtrxIs, cMtrxIs, MtrxIs[1..900]
85 OUTPUT v2
86 $INCLUDE ./RLib.m
87 INIT
88     HN1:=0

```

```
89  BN1:=0
90  HN2:=0
91  BN2:=0
92  HN3:=0
93  BN3:=0
94  HN[1..3]:=0
95  BN[1..3]:=0
96  ins1:=0
97  ins2:=0
98  i1:=0
99  i2:=0.01
100 v2:=0.01
101 e2:=0
102 e1:=0
103 O2:=0
104 P2:=0.01
105 P3:=0.01
106 io1:=0
107 ZL:=0
108 v1:=0
109 integral (v1) :=0
110 integral (v2) :=0
111 histdef (integral (v1)) :=0
112 histdef (integral (v2)) :=0
113 iLp:=0.1
114 v2p:=0.1
115 v1p:=0.1
116 io1c:=0
117 Zc:=0
118 ENDINIT
119 EXEC
120   H[1]:=11.937
121   B[1]:=0.4
122   H[2]:=20.292
123   B[2]:=0.9
124   H[3]:=24.987
125   B[3]:=1.2
126   H[4]:=37.242
127   B[4]:=1.45
128   H[5]:=58.887
129   B[5]:=1.58
130   H[6]:=119.366
131   B[6]:=1.68
132   H[7]:=226.796
133   B[7]:=1.74
134   H[8]:=612.747
135   B[8]:=1.81
136   H[9]:=1193.662
137   B[9]:=1.86
138   H[10]:=10000
139   B[10]:=1.97
140
141   -- Metodo de aproximacao linear por trechos
142   -- Esse loop calcula a indutancia de cada trecho
143   FOR i:=1 TO 9 DO
144     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
145   ENDFOR
146
147   -- Metodo de aproximacao linear por trechos parte 2
148   -- Segundo loop para calcular o l' para cada trecho
```

```

149 -- Utilizar Lp' para criar indutores equivalentes que facam
150 -- o mesmo efeito na indutancia total.
151
152 FOR i:=2 TO 9 DO
153     lp[i]:=1/((1/l[i])-(1/l[i-1]))
154 ENDFOR
155
156 d1:=arEnro*nuEsp1*(0.96)
157 d2:=arEnro*nuEsp2*(0.96)
158 P4:=Ld1/(nuEsp1*nuEsp1)
159 P5:=Ld2/(nuEsp2*nuEsp2)
160 v1:=inputV-rFonte*i1
161 v2:=(nuEsp2/nuEsp1)*v1
162
163 IF t=1e-5 THEN
164     v1p:=v1/cos(376.99*t)
165 ENDIF
166 IF t=1e-5 THEN
167     iLp:=iCarga/cos(376.99*t)
168 ENDIF
169 IF t=1e-5 THEN
170     v2p:=v2/cos(376.99*t)
171 ENDIF
172 ZL:=v2p/iLp
173
174 BN[1]:=integral(v1)
175 BN[1]:=BN[1]/d1
176 BN[2]:=integral(v2)
177 BN[2]:=BN[2]/d2
178
179     FOR i:=1 TO 2 DO
180         IF BN[i]>=0 THEN -- Calculo de H para os trechos do sistema
181             HN[i]:=BN[i]/l[1]
182             FOR j:=2 TO 9 DO
183                 IF BN[i]>B[j] THEN
184                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
185                 ENDIF
186             ENDFOR
187         ELSE
188             HN[i]:=BN[i]/l[1]
189             FOR j:=2 TO 9 DO
190                 IF BN[i]<-B[j] THEN
191                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
192                 ENDIF
193             ENDFOR
194         ENDIF
195     ENDFOR
196
197
198     P1:=(BN[1]/HN[1])*arEnro/compEr
199     P2:=(BN[2]/HN[2])*arEnro/compEr
200     O1:=BN[1]*arEnro
201     O2:=BN[2]*arEnro
202     O4:=P4*(HN[1]*compEr+(O1/P1))
203     O5:=P5*(HN[2]*compEr+(O2/P2))
204     O3:=O1-O4
205     BN[3]:=O3/arJugo
206
207     IF BN[3]>=0 THEN -- Calculo de H para os trechos do sistema
208         HN[3]:=BN[3]/l[1]

```

```

209         FOR i:=2 TO 9 DO
210             IF BN[3]>B[i] THEN
211                 HN[3]:=HN[3]+((BN[3]-B[i])/lp[i])
212             ENDIF
213         ENDFOR
214     ELSE
215         HN[3]:=BN[3]/l[1]
216         FOR i:=2 TO 9 DO
217             IF BN[3]<-B[i] THEN
218                 HN[3]:=HN[3]+((BN[3]+B[i])/lp[i])
219             ENDIF
220         ENDFOR
221     ENDIF
222
223 IF HN[3] <> 0 THEN
224     P3:=(BN[3]/HN[3])*arJugo/compJg
225
226     e2:=i2*ZL
227     e1:=v1
228
229     HN1:=HN[1]
230     BN1:=BN[1]
231     HN2:=HN[2]
232     BN2:=BN[2]
233     HN3:=HN[3]
234     BN3:=BN[3]
235
236     -- Essa parte corresponde ao calculo das matrizes RLib
237     m:=900
238     lMtrxA := 5
239     cMtrxA := 2
240     MtrxA[1..m]:=0
241     MtrxA[1..cMtrxA] :=[1, -1]
242     MtrxA[cMtrxA+1..2*cMtrxA] :=[0, 1]
243     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[-1, 0]
244     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[-1, 1]
245     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[0, -1]
246
247     lMtrxN := 2
248     cMtrxN := 2
249     MtrxN[1..m]:=0
250     MtrxN[0*cMtrxN+1] := nuEsp1
251     MtrxN[1*cMtrxN+2] := nuEsp2
252
253     lMtrxV := 2
254     cMtrxV := 1
255     MtrxV[1..m]:= 0
256     MtrxV[0*cMtrxV+1] := (e1)
257     MtrxV[1*cMtrxV+1] := (e2)
258
259     lMtrxO := 2
260     cMtrxO := 1
261     MtrxO[1..m]:=0
262     MtrxO[0*cMtrxO+1] := O1
263     MtrxO[1*cMtrxO+1] := O2
264
265     lMtrxInsIn := 2
266     cMtrxInsIn := 1
267     MtrxInsIn[1..m]:=0
268     MtrxInsIn[0*cMtrxInsIn+1] := Ins1

```

```

269     MtrxInsIn[1*cMtrxInsIn+1] := Ins2
270
271     dt := timestep
272
273     lMtrxP := 5,
274     cMtrxP := 5,
275     MtrxP[1..m]:=0
276     MtrxP[(1-1)*cMtrxP + 1] := P1
277     MtrxP[(2-1)*cMtrxP + 2] := P2
278     MtrxP[(3-1)*cMtrxP + 3] := P3
279     MtrxP[(4-1)*cMtrxP + 4] := P4
280     MtrxP[(5-1)*cMtrxP + 5] := P5
281
282     USE RNSMatrix as RNSM1
283     INPUT
284     deltaTime := dt,
285
286     lMatrixA := lMtrxA,
287     cMatrixA := cMtrxA,
288     MatrixA[1..m] := MtrxA[1..m]
289
290     lMP := lMtrxP,
291     cMP := cMtrxP,
292     MP[1..900] := MtrxP[1..m]
293
294     lMatrixN := lMtrxN,
295     cMatrixN := cMtrxN,
296     MatrixN[1..m] := MtrxN[1..m]
297
298     lMatrixV := lMtrxV,
299     cMatrixV := cMtrxV,
300     MatrixV[1..m] := MtrxV[1..m]
301
302     lMatrixO := lMtrxO,
303     cMatrixO := cMtrxO,
304     MatrixO[1..m] := MtrxO[1..m]
305
306     lMatrixInsIn := lMtrxInsIn,
307     cMatrixInsIn := cMtrxInsIn,
308     MatrixInsIn[1..m] := MtrxInsIn[1..m]
309     OUTPUT
310     MtrxIns[1..m] := MatrixIns[1..m],
311     MtrxIs[1..m] := MatrixIs[1..m]
312     ENDUSE
313     Ins1 := MtrxIns[1]
314     Ins2 := MtrxIns[2]
315     i1 := MtrxIs[1]
316     i2 := MtrxIs[2]
317
318     x:=((1/P1)*((1/P3)+(1/P4)+(1/P5))+((1/P4)*((1/P3)+(1/P5))))
319     Pc:=1/(x/((1/P3)+(1/P4)+(1/P5)))
320     Xc:=376.99*nuEsp1*nuEsp1*Pc
321     Zc:=Xc/2 -- Zc corresponde a impedancia do transformador
322     i1:=(i1+(vlp/Zc))*sin(376.99*t)
323     i2:= i2 + i1*(nuEsp1/nuEsp2)
324
325     ENDIF
326     ENDEXEC
327     ENDMODEL
328     RECORD

```

```

329 RNS2.HN1 AS HN1
330 RNS2.HN2 AS HN2
331 RNS2.HN3 AS HN3
332 RNS2.BN1 AS BN1
333 RNS2.BN2 AS BN2
334 RNS2.BN3 AS BN3
335 RNS2.INS1 AS INS1
336 RNS2.INS2 AS INS2
337 RNS2.P1 AS P1
338 RNS2.P2 AS P2
339 RNS2.P3 AS P3
340 RNS2.P4 AS P4
341 RNS2.P5 AS P5
342 RNS2.O1 AS O1
343 RNS2.O2 AS O2
344 RNS2.O3 AS O3
345 RNS2.O4 AS O4
346 RNS2.O5 AS O5
347 RNS2.v1 AS v1
348 RNS2.v2 AS v2
349 RNS2.e2 AS e2
350 RNS2.e1 AS e1
351 RNS2.v11 AS v11
352 RNS2.v12 AS v12
353 RNS2.i1 AS i1
354 RNS2.i2 AS i2
355 RNS2.io1 AS io1
356 RNS2.io1c AS io1c
357 RNS2.iCarga AS iCarga
358 RNS2.ZL AS ZL
359 RNS2.Zc AS Zc
360 USE RNS2 AS RNS2
361 INPUT
362 inputV:= MM0001
363 iCarga:= MM0002
364 DATA
365 arEnro:= 0.454
366 arJugo:= 0.454
367 nuEsp1:= 65.
368 nuEsp2:= 450.
369 compEr:= 3.59
370 compJg:= 2.66
371 Ld1:= 2.472E-4
372 Ld2:= 0.011846
373 rFonte:= 0.008
374 OUTPUT
375 XX0002:=v2
376 ENDUSE
377 ENDMODELS
378 C 1 2 3 4 5 6 7 8
379 C 34567890123456789012345678901234567890123456789012345678901234567890
380 /BRANCH
381 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
382 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
383 XX0001 .01 0
384 XX0002XX0003 .0075 3
385 /SWITCH
386 C < n 1>< n 2>< Tclose ><Top/Tde >< Ie ><Vf/CLOP >< type >
387 XX0003 -1. 1.E3 1
388 /SOURCE

```

```

389 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
390 14XX0001 1600. 60. -1. 100.
391 60XX0002 0 1.E3
392 /OUTPUT
393 BLANK MODELS
394 BLANK BRANCH
395 BLANK SWITCH
396 BLANK SOURCE
397 BLANK OUTPUT
398 BLANK PLOT
399 BEGIN NEW DATA CASE
400 BLANK

```

Código Fonte A.5 – Código ATP para reator naturalmente saturado com 2 enrolamentos em curto e sem perdas

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW novembro, sabado 7, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hidalen at SEFAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY 60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9 1.E-5 .05
10 500 1 1 1 1 0 0 1 0
11 /MODELS
12 MODELS
13 INPUT
14 MM0001 {v(XX0001)}
15 MM0002 {i(XX0003)}
16 OUTPUT
17 XX0002
18 MODEL RNS2
19 INPUT inputV -- Tensao de Entrada
20 iCarga -- Corrente na Carga
21 DATA arEnro -- Area Enrolamento
22 arJugo -- Area do Jugo
23 nuEsp1 -- Numero de Espiras enrolamento 1
24 nuEsp2 -- Numero de Espiras enrolamento 2
25 compEr -- Comprimento do Braco de Enrolamento
26 compJg -- Comprimento do Braco do Jugo
27 Ld1 -- Indutancia de dispersao no enrolamento 1
28 Ld2 -- Indutancia de dispersao no enrolamento 2
29 rFonte -- Resistencia da fonte
30 VAR B[1..10] -- Campo magnetico dos nove trechos
31 H[1..10] -- Densidade magnetica para os nove trechos
32 HN[1..3]
33 BN[1..3]
34 HN1
35 BN1
36 HN2
37 BN2
38 BN3
39 HN3
40 l[1..9] -- Indutancias para os nove trechos
41 lp[2..9] -- Indutancias linhas (l') para os nove trechos
42 x, y, z, w, xl, yl
43 d1

```



```
44      d2
45      P1
46      P2
47      P3
48      P4
49      P5
50      O1
51      O2
52      O3
53      O4
54      O5
55      ins1
56      ins2
57      i1
58      i2
59      v2
60      dy1
61      v11
62      v12
63      io1
64      Zc
65      Xc
66      Pc
67      iLp
68      v2p
69      v1p
70      io1c
71      ZL
72      e2
73      e1
74      v1
75      m, dt, determnt, aux,
76          lMtrxA, cMtrxA, MtrxA[1..900]
77          lMtrxP, cMtrxP, MtrxP[1..900]
78          lMtrxN, cMtrxN, MtrxN[1..900]
79          lMtrxV, cMtrxV, MtrxV[1..900]
80          lMtrxO, cMtrxO, MtrxO[1..900]
81          lMtrxI, cMtrxI, MtrxI[1..900]
82          lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
83          lMtrxIns, cMtrxIns, MtrxIns[1..900]
84          lMtrxIs, cMtrxIs, MtrxIs[1..900]
85 OUTPUT v2
86 $INCLUDE ./RLib.m
87 INIT
88      HN1:=0
89      BN1:=0
90      HN2:=0
91      BN2:=0
92      HN3:=0
93      BN3:=0
94      HN[1..3]:=0
95      BN[1..3]:=0
96      ins1:=0
97      ins2:=0
98      i1:=0
99      i2:=0.01
100     v2:=0.01
101     e2:=0
102     e1:=0
103     O2:=0
```

```

104 P2:=0.01
105 P3:=0.01
106 io1:=0
107 ZL:=0
108 v1:=0
109 integral(v1):=0
110 integral(v2):=0
111 histdef(integral(v1)):=0
112 histdef(integral(v2)):=0
113 iLp:=0.1
114 v2p:=0.1
115 vlp:=0.1
116 io1c:=0
117 Zc:=0
118 ENDINIT
119 EXEC
120 H[1]:=11.937
121 B[1]:=0.4
122 H[2]:=20.292
123 B[2]:=0.9
124 H[3]:=24.987
125 B[3]:=1.2
126 H[4]:=37.242
127 B[4]:=1.45
128 H[5]:=58.887
129 B[5]:=1.58
130 H[6]:=119.366
131 B[6]:=1.68
132 H[7]:=226.796
133 B[7]:=1.74
134 H[8]:=612.747
135 B[8]:=1.81
136 H[9]:=1193.662
137 B[9]:=1.86
138 H[10]:=10000
139 B[10]:=1.97
140
141 -- Metodo de aproximacao linear por trechos
142 -- Esse loop calcula a indutancia de cada trecho
143 FOR i:=1 TO 9 DO
144     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
145 ENDFOR
146
147 -- Metodo de aproximacao linear por trechos parte 2
148 -- Segundo loop para calcular o l' para cada trecho
149 -- Utilizar Lp' para criar indutores equivalentes que facam
150 -- o mesmo efeito na indutancia total.
151
152 FOR i:=2 TO 9 DO
153     lp[i]:=1/((1/l[i])-(1/l[i-1]))
154 ENDFOR
155
156 d1:=arEnro*nuEsp1*(0.96)
157 d2:=arEnro*nuEsp2*(0.96)
158 P4:=Ld1/(nuEsp1*nuEsp1)
159 P5:=Ld2/(nuEsp2*nuEsp2)
160 v1:=inputV-rFonte*i1
161 v2:=(nuEsp2/nuEsp1)*v1
162
163 IF t=1e-5 THEN

```

```

164     v1p:=v1/cos(377*t)
165 ENDIF
166 IF t=1e-5 THEN
167     iLp:=iCarga/cos(377*t)
168 ENDIF
169 IF t=1e-5 THEN
170     v2p:=v2/cos(377*t)
171 ENDIF
172     ZL:=v2p/iLp
173
174     BN[1]:=integral(v1)
175     BN[1]:=BN[1]/d1
176     BN[2]:=integral(v2)
177     BN[2]:=BN[2]/d2
178
179     FOR i:=1 TO 2 DO
180         IF BN[i]>=0 THEN                                     -- Calculo de H para os trechos do sistema
181             HN[i]:=BN[i]/l[1]
182             FOR j:=2 TO 9 DO
183                 IF BN[i]>B[j] THEN
184                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
185                 ENDIF
186             ENDFOR
187         ELSE
188             HN[i]:=BN[i]/l[1]
189             FOR j:=2 TO 9 DO
190                 IF BN[i]<-B[j] THEN
191                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
192                 ENDIF
193             ENDFOR
194         ENDIF
195     ENDFOR
196
197     P1:=(BN[1]/HN[1])*arEnro/compEr
198     P2:=(BN[2]/HN[2])*arEnro/compEr
199     O1:=BN[1]*arEnro
200     O2:=BN[2]*arEnro
201     O4:=P4*(HN[1]*compEr+(O1/P1))
202     O5:=P5*(HN[2]*compEr+(O2/P2))
203     O3:=O1-O4
204     BN[3]:=O3/arJugo
205
206
207     IF BN[3]>=0 THEN                                     -- Calculo de H para os trechos do sistema
208         HN[3]:=BN[3]/l[1]
209         FOR i:=2 TO 9 DO
210             IF BN[3]>B[i] THEN
211                 HN[3]:=HN[3]+((BN[3]-B[i])/lp[i])
212             ENDIF
213         ENDFOR
214     ELSE
215         HN[3]:=BN[3]/l[1]
216         FOR i:=2 TO 9 DO
217             IF BN[3]<-B[i] THEN
218                 HN[3]:=HN[3]+((BN[3]+B[i])/lp[i])
219             ENDIF
220         ENDFOR
221     ENDIF
222
223     IF HN[3] < 0 OR HN[3] > 0 THEN

```

```

224     P3:=(BN[3]/HN[3])*arJugo/compJg
225
226     HN1:=HN[1]
227     BN1:=BN[1]
228     HN2:=HN[2]
229     BN2:=BN[2]
230     HN3:=HN[3]
231     BN3:=BN[3]
232
233     -- Essa parte corresponde ao calculo das matrizes RLib
234     m:=900
235     lMtrxA := 5
236     cMtrxA := 2
237     MtrxA[1..m]:=0
238     MtrxA[1..cMtrxA]                :=[1, -1]
239     MtrxA[cMtrxA+1..2*cMtrxA]      :=[0, 1]
240     MtrxA[2*cMtrxA+1..3*cMtrxA]    :=[-1, 0]
241     MtrxA[3*cMtrxA+1..4*cMtrxA]    :=[-1, 1]
242     MtrxA[4*cMtrxA+1..5*cMtrxA]    :=[0, -1]
243
244     lMtrxN := 2
245     cMtrxN := 2
246     MtrxN[1..m]:=0
247     MtrxN[0*cMtrxN+1] := nuEsp1
248     MtrxN[1*cMtrxN+2] := nuEsp2
249
250     lMtrxV := 2
251     cMtrxV := 1
252     MtrxV[1..m]:= 0
253     MtrxV[0*cMtrxV+1] := (e1)
254     MtrxV[1*cMtrxV+1] := (e2)
255
256     lMtrxO := 2
257     cMtrxO := 1
258     MtrxO[1..m]:=0
259     MtrxO[0*cMtrxO+1] := O1
260     MtrxO[1*cMtrxO+1] := O2
261
262     lMtrxInsIn := 2
263     cMtrxInsIn := 1
264     MtrxInsIn[1..m]:=0
265     MtrxInsIn[0*cMtrxInsIn+1] := Ins1
266     MtrxInsIn[1*cMtrxInsIn+1] := Ins2
267
268     dt := timestep
269
270     lMtrxP := 5,
271     cMtrxP := 5,
272     MtrxP[1..m]:=0
273     MtrxP[(1-1)*cMtrxP + 1] := P1
274     MtrxP[(2-1)*cMtrxP + 2] := P2
275     MtrxP[(3-1)*cMtrxP + 3] := P3
276     MtrxP[(4-1)*cMtrxP + 4] := P4
277     MtrxP[(5-1)*cMtrxP + 5] := P5
278
279     USE RNSMatrix as RNSM1
280     INPUT
281     deltaTime := dt,
282
283     lMatrixA := lMtrxA,

```

```

284      cMatrixA := cMtrxA,
285      MatrixA[1..m] := MtrxA[1..m]
286
287      lMP := lMtrxP,
288      cMP := cMtrxP,
289      MP[1..900] := MtrxP[1..m]
290
291      lMatrixN := lMtrxN,
292      cMatrixN := cMtrxN,
293      MatrixN[1..m] := MtrxN[1..m]
294
295      lMatrixV := lMtrxV,
296      cMatrixV := cMtrxV,
297      MatrixV[1..m] := MtrxV[1..m]
298
299      lMatrixO := lMtrxO,
300      cMatrixO := cMtrxO,
301      MatrixO[1..m] := MtrxO[1..m]
302
303      lMatrixInsIn := lMtrxInsIn,
304      cMatrixInsIn := cMtrxInsIn,
305      MatrixInsIn[1..m] := MtrxInsIn[1..m]
306      OUTPUT
307      MtrxIns[1..m] := MatrixIns[1..m],
308      MtrxIs[1..m] := MatrixIs[1..m]
309      ENDUSE
310      Ins1 := MtrxIns[1]
311      Ins2 := MtrxIns[2]
312      i1 := MtrxIs[1]
313      i2 := MtrxIs[2]
314
315      i1 := i1 + (iCarga*nuEsp2/nuEsp1)
316      i2 := i2 + iCarga
317
318      ENDIF
319      ENDEXEC
320      ENDMODEL
321      RECORD
322      RNS2.HN1 AS HN1
323      RNS2.HN2 AS HN2
324      RNS2.HN3 AS HN3
325      RNS2.BN1 AS BN1
326      RNS2.BN2 AS BN2
327      RNS2.BN3 AS BN3
328      RNS2.INS1 AS INS1
329      RNS2.INS2 AS INS2
330      RNS2.P1 AS P1
331      RNS2.P2 AS P2
332      RNS2.P3 AS P3
333      RNS2.P4 AS P4
334      RNS2.P5 AS P5
335      RNS2.O1 AS O1
336      RNS2.O2 AS O2
337      RNS2.O3 AS O3
338      RNS2.O4 AS O4
339      RNS2.O5 AS O5
340      RNS2.v1 AS v1
341      RNS2.v2 AS v2
342      RNS2.e2 AS e2
343      RNS2.e1 AS e1

```

```

344 RNS2.v11 AS v11
345 RNS2.v12 AS v12
346 RNS2.i1 AS i1
347 RNS2.i2 AS i2
348 RNS2.io1 AS io1
349 RNS2.io1c AS io1c
350 RNS2.iCarga AS iCarga
351 RNS2.ZL AS ZL
352 RNS2.Zc AS Zc
353 USE RNS2 AS RNS2
354 INPUT
355 inputV:= MM0001
356 iCarga:= MM0002
357 DATA
358 arEnro:= 0.454
359 arJugo:= 0.454
360 nuEsp1:= 65.
361 nuEsp2:= 450.
362 compEr:= 3.59
363 compJg:= 2.66
364 Ld1:= 2.472E-4
365 Ld2:= 0.011846
366 rFonte:= 0.008
367 OUTPUT
368 XX0002:=v2
369 ENDUSE
370 ENDMODELS
371 C 1 2 3 4 5 6 7 8
372 C 34567890123456789012345678901234567890123456789012345678901234567890
373 /BRANCH
374 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
375 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
376 XX0001 .01 0
377 XX0002XX0003 64.5 3
378 /SWITCH
379 C < n 1>< n 2>< Tclose ><Top/Tde >< Ie ><Vf/CLOP >< type >
380 XX0003 -1. 1.E3 1
381 /SOURCE
382 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
383 14XX0001 1.6E4 60. -1. 100.
384 60XX0002 0 1.E3
385 /OUTPUT
386 BLANK MODELS
387 BLANK BRANCH
388 BLANK SWITCH
389 BLANK SOURCE
390 BLANK OUTPUT
391 BLANK PLOT
392 BEGIN NEW DATA CASE
393 BLANK

```

Código Fonte A.6 – Código ATP para reator naturalmente saturado com 2 enrolamentos com plena carga e sem perdas

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW novembro, sabado 7, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hidalen at SEFAS/NTNU - NORWAY 1994-2009

```

```

6 C -----
7 POWER FREQUENCY          60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9   1.E-5   .05
10   500     1     1     1     1     0     0     1     0
11 /MODELS
12 MODELS
13 INPUT
14 MM0001 {v(XX0001)}
15 MM0002 {i(XX0003)}
16 OUTPUT
17   XX0002
18 MODEL RNS2
19 INPUT inputV              -- Tensao de Entrada
20   iCarga                  -- Corrente na Carga
21 DATA arEnro              -- Area Enrolamento
22   arJugo                  -- Area do Jugo
23   nuEsp1                  -- Numero de Espiras enrolamento 1
24   nuEsp2                  -- Numero de Espiras enrolamento 2
25   compEr                  -- Comprimento do Braco de Enrolamento
26   compJg                  -- Comprimento do Braco do Jugo
27   Ld1                     -- Indutancia de dispersao no enrolamento 1
28   Ld2                     -- Indutancia de dispersao no enrolamento 2
29   rFonte                  -- Resistencia da fonte
30 VAR B[1..10]              -- Campo magnetico dos nove trechos
31   H[1..10]                -- Densidade magnetica para os nove trechos
32   HN[1..3]
33   BN[1..3]
34   HN1
35   BN1
36   HN2
37   BN2
38   BN3
39   HN3
40   l[1..9]                 -- Indutancias para os nove trechos
41   lp[2..9]                -- Indutancias linhas (l') para os nove trechos
42   x, y, z, w, x1, y1
43   d1
44   d2
45   P1
46   P2
47   P3
48   P4
49   P5
50   O1
51   O2
52   O3
53   O4
54   O5
55   ins1
56   ins2
57   i1
58   i2
59   v2
60   dy1
61   v11
62   v12
63   io1
64   Zc
65   Xc

```

```
66     Pc
67     iLp
68     v2p
69     v1p
70     io1c
71     ZL
72     e2
73     e1
74     v1
75     m, dt, determnt, aux,
76     lMtrxA, cMtrxA, MtrxA[1..900]
77     lMtrxP, cMtrxP, MtrxP[1..900]
78     lMtrxN, cMtrxN, MtrxN[1..900]
79     lMtrxV, cMtrxV, MtrxV[1..900]
80     lMtrxO, cMtrxO, MtrxO[1..900]
81     lMtrxI, cMtrxI, MtrxI[1..900]
82     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
83     lMtrxIns, cMtrxIns, MtrxIns[1..900]
84     lMtrxIs, cMtrxIs, MtrxIs[1..900]
85 OUTPUT v2
86 $INCLUDE ./RLib.m
87 INIT
88     HN1:=0
89     BN1:=0
90     HN2:=0
91     BN2:=0
92     HN3:=0
93     BN3:=0
94     HN[1..3]:=0
95     BN[1..3]:=0
96     ins1:=0
97     ins2:=0
98     i1:=0
99     i2:=0.01
100    v2:=0.01
101    e2:=0
102    e1:=0
103    O2:=0
104    P2:=0.01
105    P3:=0.01
106    io1:=0
107    ZL:=0
108    v1:=0
109    integral(v1):=0
110    integral(v2):=0
111    histdef(integral(v1)):=0
112    histdef(integral(v2)):=0
113    iLp:=0.1
114    v2p:=0.1
115    v1p:=0.1
116    io1c:=0
117    Zc:=0
118 ENDINIT
119 EXEC
120     H[1]:=11.937
121     B[1]:=0.4
122     H[2]:=20.292
123     B[2]:=0.9
124     H[3]:=24.987
125     B[3]:=1.2
```



```

126 H[4]:=37.242
127 B[4]:=1.45
128 H[5]:=58.887
129 B[5]:=1.58
130 H[6]:=119.366
131 B[6]:=1.68
132 H[7]:=226.796
133 B[7]:=1.74
134 H[8]:=612.747
135 B[8]:=1.81
136 H[9]:=1193.662
137 B[9]:=1.86
138 H[10]:=10000
139 B[10]:=1.97
140
141 -- Metodo de aproximacao linear por trechos
142 -- Esse loop calcula a indutancia de cada trecho
143 FOR i:=1 TO 9 DO
144     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
145 ENDFOR
146
147 -- Metodo de aproximacao linear por trechos parte 2
148 -- Segundo loop para calcular o l' para cada trecho
149 -- Utilizar Lp' para criar indutores equivalentes que facam
150 -- o mesmo efeito na indutancia total.
151
152 FOR i:=2 TO 9 DO
153     lp[i]:=1/((1/l[i])-(1/l[i-1]))
154 ENDFOR
155
156 d1:=arEnro*nuEsp1*(0.96)
157 d2:=arEnro*nuEsp2*(0.96)
158 P4:=Ld1/(nuEsp1*nuEsp1)
159 P5:=Ld2/(nuEsp2*nuEsp2)
160 v1:=inputV-rFonte*i1
161 v2:=(nuEsp2/nuEsp1)*v1
162
163 IF t=1e-5 THEN
164     v1p:=v1/cos(376.99*t)
165 ENDFIF
166 IF t=1e-5 THEN
167     iLp:=iCarga/cos(376.99*t)
168 ENDFIF
169 IF t=1e-5 THEN
170     v2p:=v2/cos(376.99*t)
171 ENDFIF
172 ZL:=v2p/iLp
173
174 BN[1]:=integral(v1)
175 BN[1]:=BN[1]/d1
176 BN[2]:=integral(v2)
177 BN[2]:=BN[2]/d2
178
179     FOR i:=1 TO 2 DO
180         IF BN[i]>=0 THEN -- Calculo de H para os trechos do sistema
181             HN[i]:=BN[i]/l[1]
182             FOR j:=2 TO 9 DO
183                 IF BN[i]>B[j] THEN
184                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
185                 ENDFIF

```

```

186         ENDFOR
187     ELSE
188         HN[i]:=BN[i]/l[1]
189         FOR j:=2 TO 9 DO
190             IF BN[i]<-B[j] THEN
191                 HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
192             ENDIF
193         ENDFOR
194     ENDIF
195 ENDFOR
196
197
198 P1:=(BN[1]/HN[1])*arEnro/compEr
199 P2:=(BN[2]/HN[2])*arEnro/compEr
200 O1:=BN[1]*arEnro
201 O2:=BN[2]*arEnro
202 O4:=P4*(HN[1]*compEr+(O1/P1))
203 O5:=P5*(HN[2]*compEr+(O2/P2))
204 O3:=O1-O4
205 BN[3]:=O3/arJugo
206
207 IF BN[3]>=0 THEN -- Calculo de H para os trechos do sistema
208     HN[3]:=BN[3]/l[1]
209     FOR i:=2 TO 9 DO
210         IF BN[3]>B[i] THEN
211             HN[3]:=HN[3]+((BN[3]-B[i])/lp[i])
212         ENDIF
213     ENDFOR
214 ELSE
215     HN[3]:=BN[3]/l[1]
216     FOR i:=2 TO 9 DO
217         IF BN[3]<-B[i] THEN
218             HN[3]:=HN[3]+((BN[3]+B[i])/lp[i])
219         ENDIF
220     ENDFOR
221 ENDIF
222
223 IF HN[3] < 0 OR HN[3] > 0 THEN
224     P3:=(BN[3]/HN[3])*arJugo/compJg
225     HN1:=HN[1]
226     BN1:=BN[1]
227     HN2:=HN[2]
228     BN2:=BN[2]
229     HN3:=HN[3]
230     BN3:=BN[3]
231
232     -- Essa parte corresponde ao calculo das matrizes RLib
233     m:=900
234     lMtrxA := 5
235     cMtrxA := 2
236     MtrxA[1..m]:=0
237     MtrxA[1..cMtrxA] :=[1, -1]
238     MtrxA[cMtrxA+1..2*cMtrxA] :=[0, 1]
239     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[-1, 0]
240     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[-1, 1]
241     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[0, -1]
242
243     lMtrxN := 2
244     cMtrxN := 2
245     MtrxN[1..m]:=0

```

```

246     MtrxN[0*cMtrxN+1] := nuEsp1
247     MtrxN[1*cMtrxN+2] := nuEsp2
248
249     lMtrxV := 2
250     cMtrxV := 1
251     MtrxV[1..m] := 0
252     MtrxV[0*cMtrxV+1] := (v1)
253     MtrxV[1*cMtrxV+1] := (v2)
254
255     lMtrxO := 2
256     cMtrxO := 1
257     MtrxO[1..m] := 0
258     MtrxO[0*cMtrxO+1] := O1
259     MtrxO[1*cMtrxO+1] := O2
260
261     lMtrxInsIn := 2
262     cMtrxInsIn := 1
263     MtrxInsIn[1..m] := 0
264     MtrxInsIn[0*cMtrxInsIn+1] := Ins1
265     MtrxInsIn[1*cMtrxInsIn+1] := Ins2
266
267     dt := timestep
268
269     lMtrxP := 5,
270     cMtrxP := 5,
271     MtrxP[1..m] := 0
272     MtrxP[(1-1)*cMtrxP + 1] := P1
273     MtrxP[(2-1)*cMtrxP + 2] := P2
274     MtrxP[(3-1)*cMtrxP + 3] := P3
275     MtrxP[(4-1)*cMtrxP + 4] := P4
276     MtrxP[(5-1)*cMtrxP + 5] := P5
277
278     USE RNSMatrix as RNSM1
279     INPUT
280     deltaTime := dt,
281
282     lMatrixA := lMtrxA,
283     cMatrixA := cMtrxA,
284     MatrixA[1..m] := MtrxA[1..m]
285
286     lMP := lMtrxP,
287     cMP := cMtrxP,
288     MP[1..900] := MtrxP[1..m]
289
290     lMatrixN := lMtrxN,
291     cMatrixN := cMtrxN,
292     MatrixN[1..m] := MtrxN[1..m]
293
294     lMatrixV := lMtrxV,
295     cMatrixV := cMtrxV,
296     MatrixV[1..m] := MtrxV[1..m]
297
298     lMatrixO := lMtrxO,
299     cMatrixO := cMtrxO,
300     MatrixO[1..m] := MtrxO[1..m]
301
302     lMatrixInsIn := lMtrxInsIn,
303     cMatrixInsIn := cMtrxInsIn,
304     MatrixInsIn[1..m] := MtrxInsIn[1..m]
305     OUTPUT

```

```
306      MtrxIns[1..m] := MatrixIns[1..m],
307      MtrxIs[1..m] := MatrixIs[1..m]
308      ENDUSE
309      Ins1 := MtrxIns[1]
310      Ins2 := MtrxIns[2]
311      i1 := MtrxIs[1]
312      i2 := MtrxIs[2]
313
314      ENDIF
315      ENDEXEC
316      ENDMODEL
317      RECORD
318      RNS2.HN1 AS HN1
319      RNS2.HN2 AS HN2
320      RNS2.HN3 AS HN3
321      RNS2.BN1 AS BN1
322      RNS2.BN2 AS BN2
323      RNS2.BN3 AS BN3
324      RNS2.INS1 AS INS1
325      RNS2.INS2 AS INS2
326      RNS2.P1 AS P1
327      RNS2.P2 AS P2
328      RNS2.P3 AS P3
329      RNS2.P4 AS P4
330      RNS2.P5 AS P5
331      RNS2.O1 AS O1
332      RNS2.O2 AS O2
333      RNS2.O3 AS O3
334      RNS2.O4 AS O4
335      RNS2.O5 AS O5
336      RNS2.v1 AS v1
337      RNS2.v2 AS v2
338      RNS2.e2 AS e2
339      RNS2.e1 AS e1
340      RNS2.v11 AS v11
341      RNS2.v12 AS v12
342      RNS2.i1 AS i1
343      RNS2.i2 AS i2
344      RNS2.io1 AS io1
345      RNS2.io1c AS io1c
346      RNS2.iCarga AS iCarga
347      RNS2.ZL AS ZL
348      RNS2.Zc AS Zc
349      USE RNS2 AS RNS2
350      INPUT
351      inputV:= MM0001
352      iCarga:= MM0002
353      DATA
354      arEnro:= 0.454
355      arJugo:= 0.454
356      nuEsp1:= 65.
357      nuEsp2:= 450.
358      compEr:= 3.59
359      compJg:= 2.66
360      Ld1:= 2.472E-4
361      Ld2:= 0.011846
362      rFonte:= 0.107
363      OUTPUT
364      XX0002:=v2
365      ENDUSE
```

```

366 ENDMODELS
367 C      1      2      3      4      5      6      7      8
368 C 34567890123456789012345678901234567890123456789012345678901234567890
369 /BRANCH
370 C < n1 >> n2 ><ref1><ref2>> R >> L >> C >
371 C < n1 >> n2 ><ref1><ref2>> R >> A >> B ><Leng><><>0
372 XX0001 .01 0
373 XX0002XX0003 1.E6 3
374 /SWITCH
375 C < n 1>> n 2>> Tclose ><Top/Tde >> Ie ><Vf/CLOP >> type >
376 XX0003 -1. 1.E3 1
377 /SOURCE
378 C < n 1>><> Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
379 14XX0001 1.6E4 60. -1. 100.
380 60XX0002 0 1.E3
381 /OUTPUT
382 BLANK MODELS
383 BLANK BRANCH
384 BLANK SWITCH
385 BLANK SOURCE
386 BLANK OUTPUT
387 BLANK PLOT
388 BEGIN NEW DATA CASE
389 BLANK

```

Código Fonte A.7 – Código ATP para reator naturalmente saturado com 2 enrolamentos em vazio e sem perdas

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW dezembro, quinta-feira 3, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hoidalen at SEFAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY 60.
8 C dT >< Tmax >> Xopt >< Copt ><Epsiln>
9 5.E-6 .05 60. 1.E-180
10 500 1 1 1 1 0 0 1 0
11 /MODELS
12 MODELS
13 INPUT
14 M0001A {v(X0002A)}
15 M0001B {v(X0002B)}
16 M0001C {v(X0002C)}
17 OUTPUT
18 X0001A
19 X0001B
20 X0001C
21 MODEL RNS6Yd
22 INPUT v[1..3]
23 DATA Aw
24 Ay
25 Ar
26 N0
27 N3
28 Lw1
29 Lw2
30 Ly
31 Lr

```

```
32      Ld0
33      Ld3
34  VAR B[1..48]          -- Campo magnetico dos nove trechos
35      H[1..48]          -- Densidade magnetica para os nove trechos
36      HN[1..12]
37      BN[1..12]
38      HN1, HN2, HN3, HN4, HN5, HN6, HN7, HN8, HN9, HN10, HN11, HN12
39      BN1, BN2, BN3, BN4, BN5, BN6, BN7, BN8, BN9, BN10, BN11, BN12
40      l[1..47]          -- Indutancias para os nove trechos
41      lp[2..47]         -- Indutancias linhas (l') para os nove trechos
42      x
43      y
44      d1
45      d2
46      d3
47      d4
48      d5
49      d6
50      P1
51      P2
52      P3
53      P4
54      P5
55      P6
56      P7
57      P8
58      P9
59      P10
60      P11
61      P12
62      P13
63      P14
64      P15
65      P16
66      P17
67      P18
68      O1
69      O2
70      O3
71      O4
72      O5
73      O6
74      O7
75      O8
76      O9
77      O10
78      O11
79      O12
80      O13
81      O14
82      O15
83      O16
84      O17
85      O18
86      a1[1..6]
87      a2[1..6]
88      a3[1..6]
89      a4[1..6]
90      a5[1..6]
91      a6[1..6]
```

92	f1[1..6]
93	f2[1..6]
94	f3[1..6]
95	f4[1..6]
96	f5[1..6]
97	f6[1..6]
98	j1[1..7]
99	j2[1..7]
100	j3[1..7]
101	j4[1..7]
102	j5[1..7]
103	j6[1..7]
104	j7[1..7]
105	u1[1..6]
106	u2[1..6]
107	u3[1..6]
108	u4[1..6]
109	u5[1..6]
110	u6[1..6]
111	ra1[1..7]
112	ra2[1..7]
113	ra3[1..7]
114	ra4[1..7]
115	ra5[1..7]
116	ra6[1..7]
117	ra7[1..7]
118	g1[1..6]
119	g2[1..6]
120	g3[1..6]
121	g4[1..6]
122	g5[1..6]
123	g6[1..6]
124	m1[1..6]
125	m2[1..6]
126	m3[1..6]
127	m4[1..6]
128	m5[1..6]
129	m6[1..6]
130	q11
131	q22
132	q33
133	q44
134	q55
135	q66
136	ins1
137	ins2
138	ins3
139	ins4
140	ins5
141	ins6
142	i1A
143	i2A
144	i1B
145	i2B
146	i1C
147	i2C
148	i1[1..3]
149	i2[1..3]
150	v1A
151	v2A

152	v1B
153	v2B
154	v1C
155	v2C
156	v1[1..3]
157	v2[1..3]
158	v11p
159	v12p
160	v13p
161	v21p
162	v22p
163	v23p
164	v1p
165	v2p
166	v3p
167	iL1p
168	iL2p
169	iL3p
170	ZL1
171	ZL2
172	ZL3
173	io1A
174	io2A
175	io1B
176	io2B
177	io1C
178	io2C
179	e2[1..3]
180	Pc1
181	Pc2
182	Pc3
183	Xc1
184	Xc2
185	Xc3
186	Zc1
187	Zc2
188	Zc3
189	x1
190	x2
191	y1
192	y2
193	e11
194	e21
195	e22
196	e23
197	e1[1..3]
198	e12[1..3]
199	ip
200	e121
201	Zc
202	vn
203	in
204	j71
205	j72
206	j73
207	j74
208	j75
209	j76
210	j77
211	xla


```
212     xlb
213     xlc
214     z
215     m, dt, determnt, aux,
216     lMtrxA, cMtrxA, MtrxA[1..900]
217     lMtrxP, cMtrxP, MtrxP[1..900]
218     lMtrxN, cMtrxN, MtrxN[1..900]
219     lMtrxV, cMtrxV, MtrxV[1..900]
220     lMtrxO, cMtrxO, MtrxO[1..900]
221     lMtrxI, cMtrxI, MtrxI[1..900]
222     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
223     lMtrxIns, cMtrxIns, MtrxIns[1..900]
224     lMtrxIs, cMtrxIs, MtrxIs[1..900]
225 OUTPUT v2[1..3]
226 $INCLUDE ./RLib.m
227 INIT
228     HN1:=0
229     BN1:=0
230     BN3:=0
231     HN3:=0
232     BN5:=0
233     HN5:=0
234     BN7:=0
235     HN7:=0
236     BN8:=0
237     HN8:=0
238     HN[1..12]:=0
239     BN[1..12]:=0
240     ins1:=0.01
241     ins2:=0.01
242     ins3:=0.01
243     ins4:=0.01
244     ins5:=0.01
245     ins6:=0.01
246     i1[1..3]:=0.01
247     i2[1..3]:=0.01
248     v2[1..3]:=0.01
249     i1A:=0.01
250     i2A:=0.01
251     i1B:=0.01
252     i2B:=0.01
253     i1C:=0.01
254     i2C:=0.01
255     v1A:=0.01
256     v2A:=0.01
257     v1B:=0.01
258     v2B:=0.01
259     v1C:=0.01
260     v2C:=0.01
261     e21:=0.01
262     e22:=0.01
263     e23:=0.01
264     v1[1..3]:=0.01
265     P1:=0.01
266     P2:=0.01
267     P3:=0.01
268     P4:=0.01
269     P5:=0.01
270     P6:=0.01
271     P13:=0.01
```

```
272 P14:=0.01
273 P15:=0.01
274 P16:=0.01
275 P17:=0.01
276 P18:=0.01
277 m1[1..6]:=0
278 m2[1..6]:=0
279 m3[1..6]:=0
280 m4[1..6]:=0
281 m5[1..6]:=0
282 m6[1..6]:=0
283 j1[1..7]:=0.01
284 j2[1..7]:=0.01
285 j3[1..7]:=0.01
286 j4[1..7]:=0.01
287 j5[1..7]:=0.01
288 j6[1..7]:=0.01
289 j7[1..7]:=0.01
290 v1p:=0
291 v2p:=0
292 v3p:=0
293 v11p:=0
294 v12p:=0
295 v13p:=0
296 iL1p:=0
297 iL2p:=0
298 iL3p:=0
299 v21p:=0
300 v22p:=0
301 v23p:=0
302 io1A:=0
303 io2A:=0
304 io1B:=0
305 io2B:=0
306 io1C:=0
307 io2C:=0
308 q11:=0.01
309 q22:=0.01
310 q33:=0.01
311 q44:=0.01
312 q55:=0.01
313 q66:=0.01
314 in:=0
315 vn:=0
316 x1a:=0
317 x1b:=0
318 x1c:=0
319 ENDINIT
320 EXEC
321 H[1]:=11.937
322 B[1]:=0.4
323 H[2]:=13.608
324 B[2]:=0.499999
325 H[3]:=15.358
326 B[3]:=0.599998
327 H[4]:=17.109
328 B[4]:=0.699997
329 H[5]:=18.701
330 B[5]:=0.799995
331 H[6]:=20.292
```

```
332 B[6]:=0.899994
333 H[7]:=21.168
334 B[7]:=0.999992
335 H[8]:=21.884
336 B[8]:=1.049991
337 H[9]:=22.839
338 B[9]:=1.099991
339 H[10]:=23.635
340 B[10]:=1.14999
341 H[11]:=24.987
342 B[11]:=1.19999
343 H[12]:=26.42
344 B[12]:=1.24999
345 H[13]:=28.17
346 B[13]:=1.29999
347 H[14]:=30.637
348 B[14]:=1.349991
349 H[15]:=33.025
350 B[15]:=1.399992
351 H[16]:=37.242
352 B[16]:=1.449995
353 H[17]:=42.972
354 B[17]:=1.499998
355 H[18]:=45.757
356 B[18]:=1.52
357 H[19]:=48.94
358 B[19]:=1.540003
359 H[20]:=53.317
360 B[20]:=1.560006
361 H[21]:=58.887
362 B[21]:=1.580011
363 H[22]:=66.049
364 B[22]:=1.600016
365 H[23]:=76.394
366 B[23]:=1.620025
367 H[24]:=85.944
368 B[24]:=1.640033
369 H[25]:=103.451
370 B[25]:=1.660048
371 H[26]:=119.366
372 B[26]:=1.680062
373 H[27]:=147.218
374 B[27]:=1.700087
375 H[28]:=163.134
376 B[28]:=1.710101
377 H[29]:=179.049
378 B[29]:=1.720115
379 H[30]:=202.923
380 B[30]:=1.730136
381 H[31]:=226.796
382 B[31]:=1.740157
383 H[32]:=259.423
384 B[32]:=1.750186
385 H[33]:=282.5
386 B[33]:=1.760207
387 H[34]:=334.225
388 B[34]:=1.770253
389 H[35]:=377.993
390 B[35]:=1.780292
391 H[36]:=445.634
```

```
392 B[36]:=1.790352
393 H[37]:=501.338
394 B[37]:=1.800402
395 H[38]:=612.747
396 B[38]:=1.810502
397 H[39]:=700.282
398 B[39]:=1.82058
399 H[40]:=835.564
400 B[40]:=1.830701
401 H[41]:=954.93
402 B[41]:=1.840808
403 H[42]:=1114.085
404 B[42]:=1.850951
405 H[43]:=1193.662
406 B[43]:=1.861022
407 H[44]:=1392.606
408 B[44]:=1.871201
409 H[45]:=1591.549
410 B[45]:=1.881379
411 H[46]:=2500
412 B[46]:=1.922194
413 H[47]:=5000
414 B[47]:=1.964438
415 H[48]:=10000
416 B[48]:=1.978929
417
418 -- Metodo de aproximacao linear por trechos
419 -- Esse loop calcula a indutancia de cada trecho
420 FOR i:=1 TO 47 DO
421     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
422 ENDFOR
423
424 -- Metodo de aproximacao linear por trechos parte 2
425 -- Segundo loop para calcular o l' para cada trecho
426 -- Utilizar Lp' para criar indutores equivalentes que facam
427 -- o mesmo efeito na indutancia total.
428
429 FOR i:=2 TO 47 DO
430     lp[i]:=1/((1/l[i])-(1/l[i-1]))
431 ENDFOR
432
433 d1:=Aw*N0*0.96
434 d2:=Aw*N3*0.96
435 d3:=Aw*N0*0.96
436 d4:=Aw*N3*0.96
437 d5:=Aw*N0*0.96
438 d6:=Aw*N3*0.96
439 P7:=Ld0/(N0*N0)
440 P8:=Ld3/(N3*N3)
441 P9:=Ld0/(N0*N0)
442 P10:=Ld3/(N3*N3)
443 P11:=Ld0/(N0*N0)
444 P12:=Ld3/(N3*N3)
445 IF t=5e-5 THEN
446     v1p:=v[1]/cos(377*t)
447     v2p:=v[2]/cos(377*t-2.094)
448     v3p:=v[3]/cos(377*t+2.094)
449 ENDIF
450 v1[1]:=v[1]
451 v1[2]:=v[2]
```

```

452 v1[3]:=v[3]
453 IF t=5e-5 THEN
454   v11p:=v1[1]/cos(377*t)
455   v12p:=v1[2]/cos(377*t-2.094)
456   v13p:=v1[3]/cos(377*t+2.094)
457 ENDIF
458 v2[1]:=(v11p*N3/(N0))*cos(377*t+0.524)
459 v2[2]:=(v12p*N3/(N0))*cos(377*t-1.57)
460 v2[3]:=(v13p*N3/(N0))*cos(377*t+2.62)
461 v21p:=v11p*N3/(N0)
462 v22p:=v12p*N3/(N0)
463 v23p:=v13p*N3/(N0)
464
465 BN[1]:=((v11p/377)*sin(377*t))/d1
466 BN[2]:=BN[1]
467 BN[3]:=((v12p/377)*sin(377*t-2.094))/d3
468 BN[4]:=BN[3]
469 BN[5]:=((v13p/377)*sin(377*t+2.094))/d5
470 BN[6]:=BN[5]
471
472 FOR i:=1 TO 6 DO
473   IF BN[i]>=0 THEN                                     -- Calculo de H para os trechos do sistema
474     HN[i]:=BN[i]/l[1]
475     FOR j:=2 TO 47 DO
476       IF BN[i]>B[j] THEN
477         HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
478       ENDIF
479     ENDFOR
480   ELSE
481     HN[i]:=BN[i]/l[1]
482     FOR j:=2 TO 47 DO
483       IF BN[i]<-B[j] THEN
484         HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
485       ENDIF
486     ENDFOR
487   ENDIF
488 ENDFOR
489
490 P1:=(BN[1]/HN[1])*Aw/(Lw1)
491 P2:=(BN[2]/HN[1])*Aw/(Lw2)
492 P3:=(BN[3]/HN[3])*Aw/(Lw1)
493 P4:=(BN[4]/HN[3])*Aw/(Lw2)
494 P5:=(BN[5]/HN[5])*Aw/(Lw1)
495 P6:=(BN[6]/HN[5])*Aw/(Lw2)
496 O1:=BN[1]*Aw*0.96
497 O2:=BN[2]*Aw*0.96
498 O3:=BN[3]*Aw*0.96
499 O4:=BN[4]*Aw*0.96
500 O5:=BN[5]*Aw*0.96
501 O6:=BN[6]*Aw*0.96
502 O7:=P7*((HN[1]*Lw1)+(O1/P1))
503 O8:=P8*((HN[1]*Lw2)+(O2/P2))
504 O9:=P9*((HN[3]*Lw1)+(O3/P3))
505 O10:=P10*((HN[3]*Lw2)+(O4/P4))
506 O11:=P11*((HN[5]*Lw1)+(O5/P5))
507 O12:=P12*((HN[5]*Lw2)+(O6/P6))
508 O15:=(O1+O7)/2
509 O16:=(O1+O7)/2
510 O13:=O15
511 O17:=(O5+O11)/2

```

```

512     O18:=(O5+O11)/2
513     O14:=O18
514     BN[7]:=O13/(Ar*0.96)
515     BN[8]:=O14/(Ar*0.96)
516     BN[9]:=O15/(Ay*0.96)
517     BN[10]:=O16/(Ay*0.96)
518     BN[11]:=O17/(Ay*0.96)
519     BN[12]:=O18/(Ay*0.96)
520
521     FOR i:=7 TO 8 DO
522         IF BN[i]>=0 THEN -- Calculo de H para os trechos do sistema
523             HN[i]:=BN[i]/l[1]
524             FOR j:=2 TO 47 DO
525                 IF BN[i]>B[j] THEN
526                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
527                 ENDIF
528             ENDFOR
529         ELSE
530             HN[i]:=BN[i]/l[1]
531             FOR j:=2 TO 47 DO
532                 IF BN[i]<-B[j] THEN
533                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
534                 ENDIF
535             ENDFOR
536         ENDIF
537     ENDFOR
538
539     P13:=(BN[7]/HN[7])*Ar/Lr
540     P14:=(BN[8]/HN[8])*Ar/Lr
541     P15:=(BN[7]/HN[7])*Ay/Ly
542     P16:=(BN[7]/HN[7])*Ay/Ly
543     P17:=(BN[8]/HN[8])*Ay/Ly
544     P18:=(BN[8]/HN[8])*Ay/Ly
545     BN1:=BN[1]
546     BN2:=BN[2]
547     BN3:=BN[3]
548     BN4:=BN[4]
549     BN5:=BN[5]
550     BN6:=BN[6]
551     HN1:=HN[1]
552     HN2:=HN[2]
553     HN3:=HN[3]
554     HN4:=HN[4]
555     HN5:=HN[5]
556     HN6:=HN[6]
557
558     -- Essa parte corresponde ao calculo das matrizes RLib
559     m:=900
560     lMtrxA := 18
561     cMtrxA := 6
562     MtrxA[1..m]:=0
563     MtrxA[1..cMtrxA] :=[ 1,-1, 0, 0, 0, 0]
564     MtrxA[cMtrxA+1..2*cMtrxA] :=[ 0, 1, 0, 0, 0, 0]
565     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[ 0, 0, 1,-1, 0, 0]
566     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[ 0, 0, 0, 1, 0, 0]
567     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[ 0, 0, 0, 0, 1,-1]
568     MtrxA[5*cMtrxA+1..6*cMtrxA] :=[ 0, 0, 0, 0, 0, 1]
569     MtrxA[6*cMtrxA+1..7*cMtrxA] :=[-1, 1, 0, 0, 0, 0]
570     MtrxA[7*cMtrxA+1..8*cMtrxA] :=[ 0,-1, 0, 0, 0, 0]
571     MtrxA[8*cMtrxA+1..9*cMtrxA] :=[ 0, 0,-1, 1, 0, 0]

```

```

572 MtrxA[9*cMtrxA+1..10*cMtrxA] :=[ 0, 0, 0,-1, 0, 0]
573 MtrxA[10*cMtrxA+1..11*cMtrxA] :=[ 0, 0, 0, 0,-1, 1]
574 MtrxA[11*cMtrxA+1..12*cMtrxA] :=[ 0, 0, 0, 0, 0,-1]
575 MtrxA[12*cMtrxA+1..13*cMtrxA] :=[ 0, 0, 0, 0, 0, 0]
576 MtrxA[13*cMtrxA+1..14*cMtrxA] :=[ 0, 0, 0, 0, 0, 0]
577 MtrxA[14*cMtrxA+1..15*cMtrxA] :=[-1, 0, 0, 0, 0, 0]
578 MtrxA[15*cMtrxA+1..16*cMtrxA] :=[-1, 0, 1, 0, 0, 0]
579 MtrxA[16*cMtrxA+1..17*cMtrxA] :=[ 0, 0, 1, 0,-1, 0]
580 MtrxA[17*cMtrxA+1..18*cMtrxA] :=[ 0, 0, 0, 0,-1, 0]
581
582 lMtrxN := 6
583 cMtrxN := 6
584 MtrxN[1..m]:=0
585 MtrxN[0*cMtrxN+1] := N0
586 MtrxN[1*cMtrxN+2] := N3
587 MtrxN[2*cMtrxN+3] := N0
588 MtrxN[3*cMtrxN+4] := N3
589 MtrxN[4*cMtrxN+5] := N0
590 MtrxN[5*cMtrxN+6] := N3
591
592 lMtrxV := 6
593 cMtrxV := 1
594 MtrxV[1..m]:= 0
595 MtrxV[0*cMtrxV+1] := v1[1]
596 MtrxV[1*cMtrxV+1] := v1[2]
597 MtrxV[2*cMtrxV+1] := v1[3]
598 MtrxV[3*cMtrxV+1] := v2[1]
599 MtrxV[4*cMtrxV+1] := v2[2]
600 MtrxV[5*cMtrxV+1] := v2[3]
601
602 lMtrxO := 6
603 cMtrxO := 1
604 MtrxO[1..m]:=0
605 MtrxO[0*cMtrxO+1] := O1
606 MtrxO[1*cMtrxO+1] := O2
607 MtrxO[2*cMtrxO+1] := O3
608 MtrxO[3*cMtrxO+1] := O4
609 MtrxO[4*cMtrxO+1] := O5
610 MtrxO[5*cMtrxO+1] := O6
611
612
613 lMtrxInsIn := 6
614 cMtrxInsIn := 1
615 MtrxInsIn[1..m]:=0
616 MtrxInsIn[0*cMtrxInsIn+1] := ins1
617 MtrxInsIn[1*cMtrxInsIn+1] := ins2
618 MtrxInsIn[2*cMtrxInsIn+1] := ins3
619 MtrxInsIn[3*cMtrxInsIn+1] := ins4
620 MtrxInsIn[4*cMtrxInsIn+1] := ins5
621 MtrxInsIn[5*cMtrxInsIn+1] := ins6
622
623 dt := timestep
624
625 lMtrxP := 18,
626 cMtrxP := 18,
627 MtrxP[1..m]:=0
628 MtrxP[(1-1)*cMtrxP + 1] := P1
629 MtrxP[(2-1)*cMtrxP + 2] := P2
630 MtrxP[(3-1)*cMtrxP + 3] := P3
631 MtrxP[(4-1)*cMtrxP + 4] := P4

```

```
632 MtrxP[(5-1)*cMtrxP + 5] := P5
633 MtrxP[(6-1)*cMtrxP + 6] := P6
634 MtrxP[(7-1)*cMtrxP + 7] := P7
635 MtrxP[(8-1)*cMtrxP + 8] := P8
636 MtrxP[(9-1)*cMtrxP + 9] := P9
637 MtrxP[(10-1)*cMtrxP + 10] := P10
638 MtrxP[(11-1)*cMtrxP + 11] := P11
639 MtrxP[(12-1)*cMtrxP + 12] := P12
640 MtrxP[(13-1)*cMtrxP + 13] := P13
641 MtrxP[(14-1)*cMtrxP + 14] := P14
642 MtrxP[(15-1)*cMtrxP + 15] := P15
643 MtrxP[(16-1)*cMtrxP + 16] := P16
644 MtrxP[(17-1)*cMtrxP + 17] := P17
645 MtrxP[(18-1)*cMtrxP + 18] := P18
646
647 USE RNSMatrix as RNSM1
648 INPUT
649 deltaTime := dt,
650
651 lMatrixA := lMtrxA,
652 cMatrixA := cMtrxA,
653 MatrixA[1..m] := MtrxA[1..m]
654
655 lMP := lMtrxP,
656 cMP := cMtrxP,
657 MP[1..900] := MtrxP[1..m]
658
659 lMatrixN := lMtrxN,
660 cMatrixN := cMtrxN,
661 MatrixN[1..m] := MtrxN[1..m]
662
663 lMatrixV := lMtrxV,
664 cMatrixV := cMtrxV,
665 MatrixV[1..m] := MtrxV[1..m]
666
667 lMatrixO := lMtrxO,
668 cMatrixO := cMtrxO,
669 MatrixO[1..m] := MtrxO[1..m]
670
671 lMatrixInsIn := lMtrxInsIn,
672 cMatrixInsIn := cMtrxInsIn,
673 MatrixInsIn[1..m] := MtrxInsIn[1..m]
674 OUTPUT
675 MtrxIns[1..m] := MatrixIns[1..m],
676 MtrxIs[1..m] := MatrixIs[1..m]
677 ENDUSE
678 ins1 := MtrxIns[1]
679 ins2 := MtrxIns[2]
680 ins3 := MtrxIns[3]
681 ins4 := MtrxIns[4]
682 ins5 := MtrxIns[5]
683 ins6 := MtrxIns[6]
684
685 i1[1] := MtrxIs[1]
686 i2[1] := MtrxIs[2]
687 i1[2] := MtrxIs[3]
688 i2[2] := MtrxIs[4]
689 i1[3] := MtrxIs[5]
690 i2[3] := MtrxIs[6]
691
```



```
692     i1A:=i1[1] - (i1[2]/2) - (i1[3]/2)
693     i2A:=i2[1]
694     i1B:=i1[2] - (i1[1]/2) - (i1[3]/2)
695     i2B:=i2[2]
696     i1C:=i1[3] - (i1[2]/2) - (i1[1]/2)
697     i2C:=i2[3]
698     v1A:=v1[1]
699     v2A:=v2[1]
700     v1B:=v1[2]
701     v2B:=v2[2]
702     v1C:=v1[3]
703     v2C:=v2[3]
704
705     ENDEXEC
706     ENDMODEL
707     RECORD
708     RNS6Yd.HN1 AS HN1
709     RNS6Yd.BN1 AS BN1
710     RNS6Yd.HN3 AS HN3
711     RNS6Yd.BN3 AS BN3
712     RNS6Yd.HN5 AS HN5
713     RNS6Yd.BN5 AS BN5
714     RNS6Yd.HN7 AS HN7
715     RNS6Yd.BN7 AS BN7
716     RNS6Yd.HN8 AS HN8
717     RNS6Yd.BN8 AS BN8
718     RNS6Yd.BN9 AS BN9
719     RNS6Yd.HN9 AS HN9
720     RNS6Yd.BN10 AS BN10
721     RNS6Yd.HN10 AS HN10
722     RNS6Yd.BN11 AS BN11
723     RNS6Yd.HN11 AS HN11
724     RNS6Yd.BN12 AS BN12
725     RNS6Yd.HN12 AS HN12
726     RNS6Yd.i1A AS i1A
727     RNS6Yd.i2A AS i2A
728     RNS6Yd.i1B AS i1B
729     RNS6Yd.i2B AS i2B
730     RNS6Yd.i1C AS i1C
731     RNS6Yd.i2C AS i2C
732     RNS6Yd.ins1 AS ins1
733     RNS6Yd.ins2 AS ins2
734     RNS6Yd.ins3 AS ins3
735     RNS6Yd.ins4 AS ins4
736     RNS6Yd.ins5 AS ins5
737     RNS6Yd.ins6 AS ins6
738     RNS6Yd.v1A AS v1A
739     RNS6Yd.v2A AS v2A
740     RNS6Yd.v1B AS v1B
741     RNS6Yd.v2B AS v2B
742     RNS6Yd.v1C AS v1C
743     RNS6Yd.v2C AS v2C
744     RNS6Yd.io1A AS io1A
745     RNS6Yd.io2A AS io2A
746     RNS6Yd.io1B AS io1B
747     RNS6Yd.io2B AS io2B
748     RNS6Yd.io1C AS io1C
749     RNS6Yd.io2C AS io2C
750     RNS6Yd.v1p AS v1p
751     RNS6Yd.v2p AS v2p
```

```

752 RNS6Yd.v3p AS v3p
753 RNS6Yd.v11p AS v11p
754 RNS6Yd.v12p AS v12p
755 RNS6Yd.v13p AS v13p
756 RNS6Yd.v21p AS v21p
757 RNS6Yd.v22p AS v22p
758 RNS6Yd.v23p AS v23p
759 RNS6Yd.vn AS vn
760 RNS6Yd.in AS in
761 RNS6Yd.x1a AS x1a
762 RNS6Yd.x1b AS x1b
763 RNS6Yd.x1c AS x1c
764 RNS6Yd.O1 AS O1
765 RNS6Yd.O2 AS O2
766 RNS6Yd.O3 AS O3
767 RNS6Yd.O4 AS O4
768 RNS6Yd.O5 AS O5
769 RNS6Yd.O6 AS O6
770 RNS6Yd.O7 AS O7
771 RNS6Yd.O8 AS O8
772 RNS6Yd.O9 AS O9
773 RNS6Yd.O10 AS O10
774 RNS6Yd.O11 AS O11
775 RNS6Yd.O12 AS O12
776 RNS6Yd.O13 AS O13
777 RNS6Yd.O14 AS O14
778 RNS6Yd.O15 AS O15
779 RNS6Yd.O16 AS O16
780 RNS6Yd.O17 AS O17
781 RNS6Yd.O18 AS O18
782 USE RNS6Yd AS RNS6Yd
783 INPUT
784 v[1]:= M0001A
785 v[2]:= M0001B
786 v[3]:= M0001C
787 DATA
788 Aw:= 0.0158
789 Ay:= 0.0158
790 Ar:= 0.0158
791 N0:= 195.
792 N3:= 98.
793 Lw1:= 0.39
794 Lw2:= 0.27
795 Ly:= 0.3608
796 Lr:= 0.8
797 Ld0:= 3.88E-4
798 Ld3:= 0.00678
799 OUTPUT
800 X0001A:=v2[1]
801 X0001B:=v2[2]
802 X0001C:=v2[3]
803 ENDUSE
804 ENDMODELS
805 C 1 2 3 4 5 6 7 8
806 C 34567890123456789012345678901234567890123456789012345678901234567890
807 /BRANCH
808 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
809 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
810 X0002A .001 0
811 X0002B .001 0

```

```

812 X0002C .001 0
813 /SOURCE
814 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
815 14X0002A 0 2449.5 60. -1. 1.E3
816 14X0002B 0 2449.5 60. -120. -1. 1.E3
817 14X0002C 0 2449.5 60. 120. -1. 1.E3
818 /OUTPUT
819 BLANK MODELS
820 BLANK BRANCH
821 BLANK SWITCH
822 BLANK SOURCE
823 BLANK OUTPUT
824 BLANK PLOT
825 BEGIN NEW DATA CASE
826 BLANK

```

Código Fonte A.8 – Código ATP para reator naturalmente saturado com 6 enrolamentos

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW dezembro, quinta-feira 3, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hoidalen at SEFAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY 60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9 5.E-6 .05 60. 1.E-180
10 500 1 1 1 1 0 0 1 0
11 /MODELS
12 MODELS
13 INPUT
14 M0001A {v(X0002A)}
15 M0001B {v(X0002B)}
16 M0001C {v(X0002C)}
17 OUTPUT
18 X0001A
19 X0001B
20 X0001C
21 MODEL RNS6Yd
22 INPUT v[1..3]
23 DATA Aw
24 Ay
25 Ar
26 N0
27 N3
28 Lw1
29 Lw2
30 Ly
31 Lr
32 Ld0
33 Ld3
34 VAR B[1..48] -- Campo magnetico dos nove trechos
35 H[1..48] -- Densidade magnetica para os nove trechos
36 HN[1..12]
37 BN[1..12]
38 HN1,HN2,HN3,HN4,HN5,HN6,HN7,HN8,HN9,HN10,HN11,HN12
39 BN1,BN2,BN3,BN4,BN5,BN6,BN7,BN8,BN9,BN10,BN11,BN12
40 l[1..47] -- Indutancias para os nove trechos
41 lp[2..47] -- Indutancias linhas (l') para os nove trechos
42 x

```

43	y
44	d1
45	d2
46	d3
47	d4
48	d5
49	d6
50	P1
51	P2
52	P3
53	P4
54	P5
55	P6
56	P7
57	P8
58	P9
59	P10
60	P11
61	P12
62	P13
63	P14
64	P15
65	P16
66	P17
67	P18
68	O1
69	O2
70	O3
71	O4
72	O5
73	O6
74	O7
75	O8
76	O9
77	O10
78	O11
79	O12
80	O13
81	O14
82	O15
83	O16
84	O17
85	O18
86	a1[1..6]
87	a2[1..6]
88	a3[1..6]
89	a4[1..6]
90	a5[1..6]
91	a6[1..6]
92	f1[1..6]
93	f2[1..6]
94	f3[1..6]
95	f4[1..6]
96	f5[1..6]
97	f6[1..6]
98	j1[1..7]
99	j2[1..7]
100	j3[1..7]
101	j4[1..7]
102	j5[1..7]

103	j6[1..7]
104	j7[1..7]
105	u1[1..6]
106	u2[1..6]
107	u3[1..6]
108	u4[1..6]
109	u5[1..6]
110	u6[1..6]
111	ra1[1..7]
112	ra2[1..7]
113	ra3[1..7]
114	ra4[1..7]
115	ra5[1..7]
116	ra6[1..7]
117	ra7[1..7]
118	g1[1..6]
119	g2[1..6]
120	g3[1..6]
121	g4[1..6]
122	g5[1..6]
123	g6[1..6]
124	m1[1..6]
125	m2[1..6]
126	m3[1..6]
127	m4[1..6]
128	m5[1..6]
129	m6[1..6]
130	q11
131	q22
132	q33
133	q44
134	q55
135	q66
136	ins1
137	ins2
138	ins3
139	ins4
140	ins5
141	ins6
142	i1A
143	i2A
144	i1B
145	i2B
146	i1C
147	i2C
148	i1[1..3]
149	i2[1..3]
150	v1A
151	v2A
152	v1B
153	v2B
154	v1C
155	v2C
156	v1[1..3]
157	v2[1..3]
158	v11p
159	v12p
160	v13p
161	v21p
162	v22p

```
163 v23p
164 v1p
165 v2p
166 v3p
167 iL1p
168 iL2p
169 iL3p
170 ZL1
171 ZL2
172 ZL3
173 io1A
174 io2A
175 io1B
176 io2B
177 io1C
178 io2C
179 e2[1..3]
180 Pc1
181 Pc2
182 Pc3
183 Xc1
184 Xc2
185 Xc3
186 Zc1
187 Zc2
188 Zc3
189 x1
190 x2
191 y1
192 y2
193 e11
194 e21
195 e22
196 e23
197 e1[1..3]
198 e12[1..3]
199 ip
200 e121
201 Zc
202 vn
203 in
204 j71
205 j72
206 j73
207 j74
208 j75
209 j76
210 j77
211 xla
212 xlb
213 xlc
214 z
215 m, dt, determnt, aux,
216 lMtrxA, cMtrxA, MtrxA[1..900]
217 lMtrxP, cMtrxP, MtrxP[1..900]
218 lMtrxN, cMtrxN, MtrxN[1..900]
219 lMtrxV, cMtrxV, MtrxV[1..900]
220 lMtrxO, cMtrxO, MtrxO[1..900]
221 lMtrxI, cMtrxI, MtrxI[1..900]
222 lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
```

```
223         lMtrxIns, cMtrxIns, MtrxIns[1..900]
224         lMtrxIs, cMtrxIs, MtrxIs[1..900]
225         Hc -- for hysteresis
226 OUTPUT v2[1..3]
227 $INCLUDE ./RLib.m
228 INIT
229     HN1:=0
230     BN1:=0
231     BN3:=0
232     HN3:=0
233     BN5:=0
234     HN5:=0
235     BN7:=0
236     HN7:=0
237     BN8:=0
238     HN8:=0
239     HN[1..12]:=0
240     BN[1..12]:=0
241     ins1:=0.01
242     ins2:=0.01
243     ins3:=0.01
244     ins4:=0.01
245     ins5:=0.01
246     ins6:=0.01
247     i1[1..3]:=0.01
248     i2[1..3]:=0.01
249     v2[1..3]:=0.01
250     i1A:=0.01
251     i2A:=0.01
252     i1B:=0.01
253     i2B:=0.01
254     i1C:=0.01
255     i2C:=0.01
256     v1A:=0.01
257     v2A:=0.01
258     v1B:=0.01
259     v2B:=0.01
260     v1C:=0.01
261     v2C:=0.01
262     e21:=0.01
263     e22:=0.01
264     e23:=0.01
265     v1[1..3]:=0.01
266     P1:=0.01
267     P2:=0.01
268     P3:=0.01
269     P4:=0.01
270     P5:=0.01
271     P6:=0.01
272     P13:=0.01
273     P14:=0.01
274     P15:=0.01
275     P16:=0.01
276     P17:=0.01
277     P18:=0.01
278     m1[1..6]:=0
279     m2[1..6]:=0
280     m3[1..6]:=0
281     m4[1..6]:=0
282     m5[1..6]:=0
```

```
283 m6[1..6]:=0
284 j1[1..7]:=0.01
285 j2[1..7]:=0.01
286 j3[1..7]:=0.01
287 j4[1..7]:=0.01
288 j5[1..7]:=0.01
289 j6[1..7]:=0.01
290 j7[1..7]:=0.01
291 v1p:=0
292 v2p:=0
293 v3p:=0
294 v11p:=0
295 v12p:=0
296 v13p:=0
297 iL1p:=0
298 iL2p:=0
299 iL3p:=0
300 v21p:=0
301 v22p:=0
302 v23p:=0
303 io1A:=0
304 io2A:=0
305 io1B:=0
306 io2B:=0
307 io1C:=0
308 io2C:=0
309 q11:=0.01
310 q22:=0.01
311 q33:=0.01
312 q44:=0.01
313 q55:=0.01
314 q66:=0.01
315 in:=0
316 vn:=0
317 xla:=0
318 xlb:=0
319 xlc:=0
320 ENDINIT
321 EXEC
322 Hc:=53.133
323 H[1]:=11.937
324 B[1]:=0.4
325 H[2]:=13.608
326 B[2]:=0.499999
327 H[3]:=15.358
328 B[3]:=0.599998
329 H[4]:=17.109
330 B[4]:=0.699997
331 H[5]:=18.701
332 B[5]:=0.799995
333 H[6]:=20.292
334 B[6]:=0.899994
335 H[7]:=21.168
336 B[7]:=0.999992
337 H[8]:=21.884
338 B[8]:=1.049991
339 H[9]:=22.839
340 B[9]:=1.099991
341 H[10]:=23.635
342 B[10]:=1.14999
```



```
343 H[11]:=24.987
344 B[11]:=1.19999
345 H[12]:=26.42
346 B[12]:=1.24999
347 H[13]:=28.17
348 B[13]:=1.29999
349 H[14]:=30.637
350 B[14]:=1.349991
351 H[15]:=33.025
352 B[15]:=1.399992
353 H[16]:=37.242
354 B[16]:=1.449995
355 H[17]:=42.972
356 B[17]:=1.499998
357 H[18]:=45.757
358 B[18]:=1.52
359 H[19]:=48.94
360 B[19]:=1.540003
361 H[20]:=53.317
362 B[20]:=1.560006
363 H[21]:=58.887
364 B[21]:=1.580011
365 H[22]:=66.049
366 B[22]:=1.600016
367 H[23]:=76.394
368 B[23]:=1.620025
369 H[24]:=85.944
370 B[24]:=1.640033
371 H[25]:=103.451
372 B[25]:=1.660048
373 H[26]:=119.366
374 B[26]:=1.680062
375 H[27]:=147.218
376 B[27]:=1.700087
377 H[28]:=163.134
378 B[28]:=1.710101
379 H[29]:=179.049
380 B[29]:=1.720115
381 H[30]:=202.923
382 B[30]:=1.730136
383 H[31]:=226.796
384 B[31]:=1.740157
385 H[32]:=259.423
386 B[32]:=1.750186
387 H[33]:=282.5
388 B[33]:=1.760207
389 H[34]:=334.225
390 B[34]:=1.770253
391 H[35]:=377.993
392 B[35]:=1.780292
393 H[36]:=445.634
394 B[36]:=1.790352
395 H[37]:=501.338
396 B[37]:=1.800402
397 H[38]:=612.747
398 B[38]:=1.810502
399 H[39]:=700.282
400 B[39]:=1.82058
401 H[40]:=835.564
402 B[40]:=1.830701
```

```

403 H[41]:=954.93
404 B[41]:=1.840808
405 H[42]:=1114.085
406 B[42]:=1.850951
407 H[43]:=1193.662
408 B[43]:=1.861022
409 H[44]:=1392.606
410 B[44]:=1.871201
411 H[45]:=1591.549
412 B[45]:=1.881379
413 H[46]:=2500
414 B[46]:=1.922194
415 H[47]:=5000
416 B[47]:=1.964438
417 H[48]:=10000
418 B[48]:=1.978929
419
420 -- Metodo de aproximacao linear por trechos
421 -- Esse loop calcula a indutancia de cada trecho
422 FOR i:=1 TO 47 DO
423     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
424 ENDFOR
425
426 -- Metodo de aproximacao linear por trechos parte 2
427 -- Segundo loop para calcular o l' para cada trecho
428 -- Utilizar Lp' para criar indutores equivalentes que facam
429 -- o mesmo efeito na indutância total.
430
431 FOR i:=2 TO 47 DO
432     lp[i]:=1/((1/l[i])-(1/l[i-1]))
433 ENDFOR
434
435 d1:=Aw*N0*0.96
436 d2:=Aw*N3*0.96
437 d3:=Aw*N0*0.96
438 d4:=Aw*N3*0.96
439 d5:=Aw*N0*0.96
440 d6:=Aw*N3*0.96
441 P7:=Ld0/(N0*N0)
442 P8:=Ld3/(N3*N3)
443 P9:=Ld0/(N0*N0)
444 P10:=Ld3/(N3*N3)
445 P11:=Ld0/(N0*N0)
446 P12:=Ld3/(N3*N3)
447 IF t=5e-5 THEN
448     v1p:=v[1]/cos(377*t)
449     v2p:=v[2]/cos(377*t-2.094)
450     v3p:=v[3]/cos(377*t+2.094)
451 ENDFIF
452 v1[1]:=v[1]
453 v1[2]:=v[2]
454 v1[3]:=v[3]
455 IF t=5e-5 THEN
456     v11p:=v1[1]/cos(377*t)
457     v12p:=v1[2]/cos(377*t-2.094)
458     v13p:=v1[3]/cos(377*t+2.094)
459 ENDFIF
460 v2[1]:=(v11p*N3/(N0))*cos(377*t+0.524)
461 v2[2]:=(v12p*N3/(N0))*cos(377*t-1.57)
462 v2[3]:=(v13p*N3/(N0))*cos(377*t+2.62)

```

```

463 v21p:=v11p*N3/(N0)
464 v22p:=v12p*N3/(N0)
465 v23p:=v13p*N3/(N0)
466
467 BN[1]:=((v11p/377)*sin(377*t))/d1
468 BN[2]:=BN[1]
469 BN[3]:=((v12p/377)*sin(377*t-2.094))/d3
470 BN[4]:=BN[3]
471 BN[5]:=((v13p/377)*sin(377*t+2.094))/d5
472 BN[6]:=BN[5]
473
474     FOR i:=1 TO 6 DO
475         IF BN[i]>=0 THEN                                     -- Calculo de H para os trechos do sistema
476             HN[i]:=BN[i]/l[1]
477             FOR j:=2 TO 47 DO
478                 IF BN[i]>B[j] THEN
479                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
480                 ENDIF
481             ENDFOR
482         ELSE
483             HN[i]:=BN[i]/l[1]
484             FOR j:=2 TO 47 DO
485                 IF BN[i]<-B[j] THEN
486                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
487                 ENDIF
488             ENDFOR
489         ENDIF
490         IF v[1]<0 THEN
491             HN[i]:=HN[i]-Hc
492         ELSE
493             HN[i]:=HN[i]+Hc
494         ENDIF
495     ENDFOR
496
497     P1:=(BN[1]/HN[1])*Aw/(Lw1)
498     P2:=(BN[2]/HN[1])*Aw/(Lw2)
499     P3:=(BN[3]/HN[3])*Aw/(Lw1)
500     P4:=(BN[4]/HN[3])*Aw/(Lw2)
501     P5:=(BN[5]/HN[5])*Aw/(Lw1)
502     P6:=(BN[6]/HN[5])*Aw/(Lw2)
503     O1:=BN[1]*Aw*0.96
504     O2:=BN[2]*Aw*0.96
505     O3:=BN[3]*Aw*0.96
506     O4:=BN[4]*Aw*0.96
507     O5:=BN[5]*Aw*0.96
508     O6:=BN[6]*Aw*0.96
509     O7:=P7*((HN[1]*Lw1)+(O1/P1))
510     O8:=P8*((HN[1]*Lw2)+(O2/P2))
511     O9:=P9*((HN[3]*Lw1)+(O3/P3))
512     O10:=P10*((HN[3]*Lw2)+(O4/P4))
513     O11:=P11*((HN[5]*Lw1)+(O5/P5))
514     O12:=P12*((HN[5]*Lw2)+(O6/P6))
515     O15:=(O1+O7)/2
516     O16:=(O1+O7)/2
517     O13:=O15
518     O17:=(O5+O11)/2
519     O18:=(O5+O11)/2
520     O14:=O18
521     BN[7]:=O13/(Ar*0.96)
522     BN[8]:=O14/(Ar*0.96)

```

```

523     BN[9]:=015/(Ay*0.96)
524     BN[10]:=016/(Ay*0.96)
525     BN[11]:=017/(Ay*0.96)
526     BN[12]:=018/(Ay*0.96)
527
528     FOR i:=7 TO 8 DO
529         IF BN[i]>=0 THEN                                -- Calculo de H para os trechos do sistema
530             HN[i]:=BN[i]/l[1]
531             FOR j:=2 TO 47 DO
532                 IF BN[i]>B[j] THEN
533                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
534                 ENDIF
535             ENDFOR
536         ELSE
537             HN[i]:=BN[i]/l[1]
538             FOR j:=2 TO 47 DO
539                 IF BN[i]<-B[j] THEN
540                     HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
541                 ENDIF
542             ENDFOR
543         ENDIF
544     ENDFOR
545
546     P13:=(BN[7]/HN[7])*Ar/Lr
547     P14:=(BN[8]/HN[8])*Ar/Lr
548     P15:=(BN[7]/HN[7])*Ay/Ly
549     P16:=(BN[7]/HN[7])*Ay/Ly
550     P17:=(BN[8]/HN[8])*Ay/Ly
551     P18:=(BN[8]/HN[8])*Ay/Ly
552     BN1:=BN[1]
553     BN2:=BN[2]
554     BN3:=BN[3]
555     BN4:=BN[4]
556     BN5:=BN[5]
557     BN6:=BN[6]
558     HN1:=HN[1]
559     HN2:=HN[2]
560     HN3:=HN[3]
561     HN4:=HN[4]
562     HN5:=HN[5]
563     HN6:=HN[6]
564
565     -- Essa parte corresponde ao calculo das matrizes RLib
566     m:=900
567     lMtrxA := 18
568     cMtrxA := 6
569     MtrxA[1..m]:=0
570     MtrxA[1..cMtrxA] :=[ 1,-1, 0, 0, 0, 0]
571     MtrxA[cMtrxA+1..2*cMtrxA] :=[ 0, 1, 0, 0, 0, 0]
572     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[ 0, 0, 1,-1, 0, 0]
573     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[ 0, 0, 0, 1, 0, 0]
574     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[ 0, 0, 0, 0, 1,-1]
575     MtrxA[5*cMtrxA+1..6*cMtrxA] :=[ 0, 0, 0, 0, 0, 1]
576     MtrxA[6*cMtrxA+1..7*cMtrxA] :=[-1, 1, 0, 0, 0, 0]
577     MtrxA[7*cMtrxA+1..8*cMtrxA] :=[ 0,-1, 0, 0, 0, 0]
578     MtrxA[8*cMtrxA+1..9*cMtrxA] :=[ 0, 0,-1, 1, 0, 0]
579     MtrxA[9*cMtrxA+1..10*cMtrxA] :=[ 0, 0, 0,-1, 0, 0]
580     MtrxA[10*cMtrxA+1..11*cMtrxA] :=[ 0, 0, 0, 0,-1, 1]
581     MtrxA[11*cMtrxA+1..12*cMtrxA] :=[ 0, 0, 0, 0, 0,-1]
582     MtrxA[12*cMtrxA+1..13*cMtrxA] :=[ 0, 0, 0, 0, 0, 0]

```

```

583 MtrxA[13*cMtrxA+1..14*cMtrxA] :=[ 0, 0, 0, 0, 0, 0]
584 MtrxA[14*cMtrxA+1..15*cMtrxA] :=[-1, 0, 0, 0, 0, 0]
585 MtrxA[15*cMtrxA+1..16*cMtrxA] :=[-1, 0, 1, 0, 0, 0]
586 MtrxA[16*cMtrxA+1..17*cMtrxA] :=[ 0, 0, 1, 0,-1, 0]
587 MtrxA[17*cMtrxA+1..18*cMtrxA] :=[ 0, 0, 0, 0,-1, 0]
588
589 lMtrxN := 6
590 cMtrxN := 6
591 MtrxN[1..m]:=0
592 MtrxN[0*cMtrxN+1] := N0
593 MtrxN[1*cMtrxN+2] := N3
594 MtrxN[2*cMtrxN+3] := N0
595 MtrxN[3*cMtrxN+4] := N3
596 MtrxN[4*cMtrxN+5] := N0
597 MtrxN[5*cMtrxN+6] := N3
598
599 lMtrxV := 6
600 cMtrxV := 1
601 MtrxV[1..m] := 0
602 MtrxV[0*cMtrxV+1] := v1[1]
603 MtrxV[1*cMtrxV+1] := v1[2]
604 MtrxV[2*cMtrxV+1] := v1[3]
605 MtrxV[3*cMtrxV+1] := v2[1]
606 MtrxV[4*cMtrxV+1] := v2[2]
607 MtrxV[5*cMtrxV+1] := v2[3]
608
609 lMtrxO := 6
610 cMtrxO := 1
611 MtrxO[1..m]:=0
612 MtrxO[0*cMtrxO+1] := O1
613 MtrxO[1*cMtrxO+1] := O2
614 MtrxO[2*cMtrxO+1] := O3
615 MtrxO[3*cMtrxO+1] := O4
616 MtrxO[4*cMtrxO+1] := O5
617 MtrxO[5*cMtrxO+1] := O6
618
619
620 lMtrxInsIn := 6
621 cMtrxInsIn := 1
622 MtrxInsIn[1..m]:=0
623 MtrxInsIn[0*cMtrxInsIn+1] := ins1
624 MtrxInsIn[1*cMtrxInsIn+1] := ins2
625 MtrxInsIn[2*cMtrxInsIn+1] := ins3
626 MtrxInsIn[3*cMtrxInsIn+1] := ins4
627 MtrxInsIn[4*cMtrxInsIn+1] := ins5
628 MtrxInsIn[5*cMtrxInsIn+1] := ins6
629
630 dt := timestep
631
632 lMtrxP := 18,
633 cMtrxP := 18,
634 MtrxP[1..m]:=0
635 MtrxP[(1-1)*cMtrxP + 1] := P1
636 MtrxP[(2-1)*cMtrxP + 2] := P2
637 MtrxP[(3-1)*cMtrxP + 3] := P3
638 MtrxP[(4-1)*cMtrxP + 4] := P4
639 MtrxP[(5-1)*cMtrxP + 5] := P5
640 MtrxP[(6-1)*cMtrxP + 6] := P6
641 MtrxP[(7-1)*cMtrxP + 7] := P7
642 MtrxP[(8-1)*cMtrxP + 8] := P8

```

```

643     MtrxP[(9-1)*cMtrxP + 9] := P9
644     MtrxP[(10-1)*cMtrxP + 10] := P10
645     MtrxP[(11-1)*cMtrxP + 11] := P11
646     MtrxP[(12-1)*cMtrxP + 12] := P12
647     MtrxP[(13-1)*cMtrxP + 13] := P13
648     MtrxP[(14-1)*cMtrxP + 14] := P14
649     MtrxP[(15-1)*cMtrxP + 15] := P15
650     MtrxP[(16-1)*cMtrxP + 16] := P16
651     MtrxP[(17-1)*cMtrxP + 17] := P17
652     MtrxP[(18-1)*cMtrxP + 18] := P18
653
654     USE RNSMatrix as RNSM1
655     INPUT
656     deltaTime := dt,
657
658     lMatrixA := lMtrxA,
659     cMatrixA := cMtrxA,
660     MatrixA[1..m] := MtrxA[1..m]
661
662     lMP := lMtrxP,
663     cMP := cMtrxP,
664     MP[1..900] := MtrxP[1..m]
665
666     lMatrixN := lMtrxN,
667     cMatrixN := cMtrxN,
668     MatrixN[1..m] := MtrxN[1..m]
669
670     lMatrixV := lMtrxV,
671     cMatrixV := cMtrxV,
672     MatrixV[1..m] := MtrxV[1..m]
673
674     lMatrixO := lMtrxO,
675     cMatrixO := cMtrxO,
676     MatrixO[1..m] := MtrxO[1..m]
677
678     lMatrixInsIn := lMtrxInsIn,
679     cMatrixInsIn := cMtrxInsIn,
680     MatrixInsIn[1..m] := MtrxInsIn[1..m]
681     OUTPUT
682     MtrxIns[1..m] := MatrixIns[1..m],
683     MtrxIs[1..m] := MatrixIs[1..m]
684     ENDUSE
685     ins1 := MtrxIns[1]
686     ins2 := MtrxIns[2]
687     ins3 := MtrxIns[3]
688     ins4 := MtrxIns[4]
689     ins5 := MtrxIns[5]
690     ins6 := MtrxIns[6]
691
692     i1[1] := MtrxIs[1]
693     i2[1] := MtrxIs[2]
694     i1[2] := MtrxIs[3]
695     i2[2] := MtrxIs[4]
696     i1[3] := MtrxIs[5]
697     i2[3] := MtrxIs[6]
698
699     i1A:=i1[1] -(i1[2]/2)-(i1[3]/2)
700     i2A:=i2[1]
701     i1B:=i1[2] -(i1[1]/2)-(i1[3]/2)
702     i2B:=i2[2]

```

```
703      i1C:=i1[3] - (i1[2]/2) - (i1[1]/2)
704      i2C:=i2[3]
705      v1A:=v1[1]
706      v2A:=v2[1]
707      v1B:=v1[2]
708      v2B:=v2[2]
709      v1C:=v1[3]
710      v2C:=v2[3]
711
712      ENDEXEC
713      ENDMODEL
714      RECORD
715      RNS6Yd.HN1 AS HN1
716      RNS6Yd.BN1 AS BN1
717      RNS6Yd.HN3 AS HN3
718      RNS6Yd.BN3 AS BN3
719      RNS6Yd.HN5 AS HN5
720      RNS6Yd.BN5 AS BN5
721      RNS6Yd.HN7 AS HN7
722      RNS6Yd.BN7 AS BN7
723      RNS6Yd.HN8 AS HN8
724      RNS6Yd.BN8 AS BN8
725      RNS6Yd.BN9 AS BN9
726      RNS6Yd.HN9 AS HN9
727      RNS6Yd.BN10 AS BN10
728      RNS6Yd.HN10 AS HN10
729      RNS6Yd.BN11 AS BN11
730      RNS6Yd.HN11 AS HN11
731      RNS6Yd.BN12 AS BN12
732      RNS6Yd.HN12 AS HN12
733      RNS6Yd.i1A AS i1A
734      RNS6Yd.i2A AS i2A
735      RNS6Yd.i1B AS i1B
736      RNS6Yd.i2B AS i2B
737      RNS6Yd.i1C AS i1C
738      RNS6Yd.i2C AS i2C
739      RNS6Yd.ins1 AS ins1
740      RNS6Yd.ins2 AS ins2
741      RNS6Yd.ins3 AS ins3
742      RNS6Yd.ins4 AS ins4
743      RNS6Yd.ins5 AS ins5
744      RNS6Yd.ins6 AS ins6
745      RNS6Yd.v1A AS v1A
746      RNS6Yd.v2A AS v2A
747      RNS6Yd.v1B AS v1B
748      RNS6Yd.v2B AS v2B
749      RNS6Yd.v1C AS v1C
750      RNS6Yd.v2C AS v2C
751      RNS6Yd.io1A AS io1A
752      RNS6Yd.io2A AS io2A
753      RNS6Yd.io1B As io1B
754      RNS6Yd.io2B As io2B
755      RNS6Yd.io1C AS io1C
756      RNS6Yd.io2C AS io2C
757      RNS6Yd.v1p AS v1p
758      RNS6Yd.v2p AS v2p
759      RNS6Yd.v3p AS v3p
760      RNS6Yd.v11p AS v11p
761      RNS6Yd.v12p AS v12p
762      RNS6Yd.v13p AS v13p
```

```

763 RNS6Yd.v21p AS v21p
764 RNS6Yd.v22p AS v22p
765 RNS6Yd.v23p AS v23p
766 RNS6Yd.vn AS vn
767 RNS6Yd.in AS in
768 RNS6Yd.x1a AS x1a
769 RNS6Yd.x1b AS x1b
770 RNS6Yd.x1c AS x1c
771 RNS6Yd.O1 AS O1
772 RNS6Yd.O2 AS O2
773 RNS6Yd.O3 AS O3
774 RNS6Yd.O4 AS O4
775 RNS6Yd.O5 AS O5
776 RNS6Yd.O6 AS O6
777 RNS6Yd.O7 AS O7
778 RNS6Yd.O8 AS O8
779 RNS6Yd.O9 AS O9
780 RNS6Yd.O10 AS O10
781 RNS6Yd.O11 AS O11
782 RNS6Yd.O12 AS O12
783 RNS6Yd.O13 AS O13
784 RNS6Yd.O14 AS O14
785 RNS6Yd.O15 AS O15
786 RNS6Yd.O16 AS O16
787 RNS6Yd.O17 AS O17
788 RNS6Yd.O18 AS O18
789 USE RNS6Yd AS RNS6Yd
790 INPUT
791 v[1]:= M0001A
792 v[2]:= M0001B
793 v[3]:= M0001C
794 DATA
795 Aw:= 0.0158
796 Ay:= 0.0158
797 Ar:= 0.0158
798 N0:= 195.
799 N3:= 98.
800 Lw1:= 0.39
801 Lw2:= 0.27
802 Ly:= 0.3608
803 Lr:= 0.8
804 Ld0:= 3.88E-4
805 Ld3:= 0.00678
806 OUTPUT
807 X0001A:=v2[1]
808 X0001B:=v2[2]
809 X0001C:=v2[3]
810 ENDUSE
811 ENDMODELS
812 C 1 2 3 4 5 6 7 8
813 C 34567890123456789012345678901234567890123456789012345678901234567890
814 /BRANCH
815 C < n1 >> n2 ><ref1><ref2>< R >< L >< C >
816 C < n1 >> n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
817 X0002A .001 0
818 X0002B .001 0
819 X0002C .001 0
820 /SOURCE
821 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
822 14X0002A 0 2449.5 60. -1. 1.E3

```



```

823 14X0002B 0    2449.5    60.    -120.    -1.    1.E3
824 14X0002C 0    2449.5    60.    120.    -1.    1.E3
825 /OUTPUT
826 BLANK MODELS
827 BLANK BRANCH
828 BLANK SWITCH
829 BLANK SOURCE
830 BLANK OUTPUT
831 BLANK PLOT
832 BEGIN NEW DATA CASE
833 BLANK

```

Código Fonte A.9 – Código ATP para reator naturalmente saturado com 6 enrolamentos e considerando efeitos de histerese

```

1  BEGIN NEW DATA CASE
2  C -----
3  C Generated by ATPDRAW dezembro, quinta-feira 3, 2015
4  C A Bonneville Power Administration program
5  C by H. K. Hoidalén at SEFAS/NTNU - NORWAY 1994-2009
6  C -----
7  POWER FREQUENCY                60.
8  C dT >> Tmax >> Xopt >> Copt ><Epsiln>
9    5.E-6    .05    60.    1.E-180
10   500    1    1    1    1    0    0    1    0
11 /MODELS
12 MODELS
13 INPUT
14 M0001A {v(X0002A)}
15 M0001B {v(X0002B)}
16 M0001C {v(X0002C)}
17 OUTPUT
18   X0001A
19   X0001B
20   X0001C
21 MODEL RNS9
22 INPUT v[1..3]
23 DATA Aw
24   Ay
25   Ar
26   N1
27   N2
28   N3
29   Lw1
30   Lw2
31   Lw3
32   Ly
33   Lr
34   Ld1
35   Ld2
36   Ld3
37 VAR B[1..48]          -- Campo magnetico dos nove trechos
38   H[1..48]           -- Densidade magnetica para os nove trechos
39   HN[1..15]
40   BN[1..15]
41   HN1,HN2,HN3,HN4,HN5,HN6,HN7,HN8,HN9,HN10,HN11,HN12,HN13,HN14,HN15
42   BN1,BN2,BN3,BN4,BN5,BN6,BN7,BN8,BN9,BN10,BN11,BN12,BN13,BN14,BN15
43   l[1..47]           -- Indutancias para os nove trechos
44   lp[2..47]          -- Indutancias linhas (l') para os nove trechos

```

```

45     d1, d2, d3, d4, d5, d6, d7, d8, d9
46     P1, P2, P3, P4, P5, P6, P7, P8, P9
47     P10, P11, P12, P13, P14, P15, P16, P17, P18
48     P19, P20, P21, P22, P23, P24
49     O1, O2, O3, O4, O5, O6, O7, O8, O9
50     O10, O11, O12, O13, O14, O15, O16, O17, O18
51     O19, O20, O21, O22, O23, O24
52     ins1, ins2, ins3, ins4, ins5, ins6, ins7, ins8, ins9
53     i1A, i1B, i1C
54     i2A, i2B, i2C
55     i3A, i3B, i3C
56     i1[1..3], i2[1..3], i3[1..3]
57     v1A, v1B, v1C
58     v2A, v2B, v2C
59     v3A, v3B, v3C
60     v1[1..3], v2[1..3], v3[1..3]
61     v1p, v2p, v3p
62     v11p, v12p, v13p
63     v21p, v22p, v23p
64     io1A, io1B, io1C
65     io2A, io2B, io2C
66     io3A, io3B, io3C
67     --Library Variables
68     m, dt, determnt, aux,
69     lMtrxA, cMtrxA, MtrxA[1..900]
70     lMtrxP, cMtrxP, MtrxP[1..900]
71     lMtrxN, cMtrxN, MtrxN[1..900]
72     lMtrxV, cMtrxV, MtrxV[1..900]
73     lMtrxO, cMtrxO, MtrxO[1..900]
74     lMtrxI, cMtrxI, MtrxI[1..900]
75     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
76     lMtrxIns, cMtrxIns, MtrxIns[1..900]
77     lMtrxIs, cMtrxIs, MtrxIs[1..900]
78     OUTPUT v3[1..3]
79     $INCLUDE ./RLib.m
80     INIT
81     HN[1..15]:=0
82     BN[1..15]:=0
83     ins1:=0.01
84     ins2:=0.01
85     ins3:=0.01
86     ins4:=0.01
87     ins5:=0.01
88     ins6:=0.01
89     ins7:=0.01
90     ins8:=0.01
91     ins9:=0.01
92     i1[1..3]:=0.01
93     i2[1..3]:=0.01
94     i3[1..3]:=0.01
95     v1[1..3]:=0.01
96     v2[1..3]:=0.01
97     v3[1..3]:=0.01
98     i1A:=0.01
99     i2A:=0.01
100    i3A:=0.01
101    i1B:=0.01
102    i2B:=0.01
103    i3B:=0.01
104    i1C:=0.01

```

```
105 i2C:=0.01
106 i3C:=0.01
107 v1A:=0.01
108 v2A:=0.01
109 v3A:=0.01
110 v1B:=0.01
111 v2B:=0.01
112 v3B:=0.01
113 v1C:=0.01
114 v2C:=0.01
115 v3C:=0.01
116 P1:=0.01
117 P2:=0.01
118 P3:=0.01
119 P4:=0.01
120 P5:=0.01
121 P6:=0.01
122 P13:=0.01
123 P14:=0.01
124 P15:=0.01
125 P16:=0.01
126 P17:=0.01
127 P18:=0.01
128 P19:=0.01
129 P20:=0.01
130 P21:=0.01
131 P22:=0.01
132 P23:=0.01
133 P24:=0.01
134 v1p:=0
135 v2p:=0
136 v3p:=0
137 v11p:=0
138 v12p:=0
139 v13p:=0
140 v21p:=0
141 v22p:=0
142 v23p:=0
143 io1A:=0
144 io2A:=0
145 io3A:=0
146 io1B:=0
147 io2B:=0
148 io3B:=0
149 io1C:=0
150 io2C:=0
151 io3C:=0
152 ENDINIT
153 EXEC
154 H[1]:=11.937
155 B[1]:=0.4
156 H[2]:=13.608
157 B[2]:=0.499999
158 H[3]:=15.358
159 B[3]:=0.599998
160 H[4]:=17.109
161 B[4]:=0.699997
162 H[5]:=18.701
163 B[5]:=0.799995
164 H[6]:=20.292
```

```
165 B[6]:=0.899994
166 H[7]:=21.168
167 B[7]:=0.999992
168 H[8]:=21.884
169 B[8]:=1.049991
170 H[9]:=22.839
171 B[9]:=1.099991
172 H[10]:=23.635
173 B[10]:=1.14999
174 H[11]:=24.987
175 B[11]:=1.19999
176 H[12]:=26.42
177 B[12]:=1.24999
178 H[13]:=28.17
179 B[13]:=1.29999
180 H[14]:=30.637
181 B[14]:=1.349991
182 H[15]:=33.025
183 B[15]:=1.399992
184 H[16]:=37.242
185 B[16]:=1.449995
186 H[17]:=42.972
187 B[17]:=1.499998
188 H[18]:=45.757
189 B[18]:=1.52
190 H[19]:=48.94
191 B[19]:=1.540003
192 H[20]:=53.317
193 B[20]:=1.560006
194 H[21]:=58.887
195 B[21]:=1.580011
196 H[22]:=66.049
197 B[22]:=1.600016
198 H[23]:=76.394
199 B[23]:=1.620025
200 H[24]:=85.944
201 B[24]:=1.640033
202 H[25]:=103.451
203 B[25]:=1.660048
204 H[26]:=119.366
205 B[26]:=1.680062
206 H[27]:=147.218
207 B[27]:=1.700087
208 H[28]:=163.134
209 B[28]:=1.710101
210 H[29]:=179.049
211 B[29]:=1.720115
212 H[30]:=202.923
213 B[30]:=1.730136
214 H[31]:=226.796
215 B[31]:=1.740157
216 H[32]:=259.423
217 B[32]:=1.750186
218 H[33]:=282.5
219 B[33]:=1.760207
220 H[34]:=334.225
221 B[34]:=1.770253
222 H[35]:=377.993
223 B[35]:=1.780292
224 H[36]:=445.634
```

```

225 B[36]:=1.790352
226 H[37]:=501.338
227 B[37]:=1.800402
228 H[38]:=612.747
229 B[38]:=1.810502
230 H[39]:=700.282
231 B[39]:=1.82058
232 H[40]:=835.564
233 B[40]:=1.830701
234 H[41]:=954.93
235 B[41]:=1.840808
236 H[42]:=1114.085
237 B[42]:=1.850951
238 H[43]:=1193.662
239 B[43]:=1.861022
240 H[44]:=1392.606
241 B[44]:=1.871201
242 H[45]:=1591.549
243 B[45]:=1.881379
244 H[46]:=2500
245 B[46]:=1.922194
246 H[47]:=5000
247 B[47]:=1.964438
248 H[48]:=10000
249 B[48]:=1.978929
250
251 -- Metodo de aproximacao linear por trechos
252 -- Esse loop calcula a indutancia de cada trecho
253 FOR i:=1 TO 47 DO
254     l[i]:=(B[i+1]-B[i])/(H[i+1]-H[i])
255 ENDFOR
256
257 -- Metodo de aproximacao linear por trechos parte 2
258 -- Segundo loop para calcular o l' para cada trecho
259 -- Utilizar Lp' para criar indutores equivalentes que facam
260 -- o mesmo efeito na indutancia total.
261
262 FOR i:=2 TO 47 DO
263     lp[i]:=1/((1/l[i])-(1/l[i-1]))
264 ENDFOR
265
266 v1[1]:=v[1]
267 v1[2]:=v[2]
268 v1[3]:=v[3]
269
270 IF t=5e-5 THEN
271     v1p:=v[1]/cos(377*t)
272     v2p:=v[2]/cos(377*t-2.094)
273     v3p:=v[3]/cos(377*t+2.094)
274 ENDFIF
275 v1[1]:=((0.743*v1p))*cos(377*t)
276 v1[2]:=((0.743*v2p))*cos(377*t-2.094)
277 v1[3]:=((0.743*v3p))*cos(377*t+2.094)
278 IF t=5e-5 THEN
279     v11p:=v1[1]/cos(377*t)
280     v12p:=v1[2]/cos(377*t-2.094)
281     v13p:=v1[3]/cos(377*t+2.094)
282 ENDFIF
283 v2[1]:=((N2/N1)*v11p)*cos(377*t)
284 v2[2]:=((N2/N1)*v12p)*cos(377*t-2.094)

```

```

285 v2[3]:=((N2/N1)*v13p)*cos(377*t+2.094)
286 v21p:=((N2/N1)*v11p)
287 v22p:=((N2/N1)*v12p)
288 v23p:=((N2/N1)*v13p)
289 v3[1]:= (v11p*0.675)*cos(377*t-0.524)
290 v3[2]:= (v12p*0.675)*cos(377*t-2.094-0.524)
291 v3[3]:= (v13p*0.675)*cos(377*t+2.094-0.524)
292
293 --Enrolamentos da esquerda
294 d1:=Aw*(N1+N2)*0.96
295 BN[1]:=(((v11p+v21p)/(377))*sin(377*t))/d1
296 BN[2]:=BN[1]
297
298 --Enrolamentos do MEIO
299 d3:=Aw*(N1+N2)*0.96
300 BN[3]:=(((v12p+v22p)/(377))*sin(377*t-2.094))/d3
301 BN[4]:=BN[3]
302
303 --Enrolamentos da direita
304 d5:=Aw*(N1+N2)*0.96
305 BN[5]:=(((v13p+v23p)/(377))*sin(377*t+2.094))/d5
306 BN[6]:=BN[5]
307
308 FOR i:=1 TO 6 DO
309     IF BN[i]>=0 THEN                                     -- Calculo de H para os trechos do sistema
310         HN[i]:=BN[i]/l[1]
311         FOR j:=2 TO 47 DO
312             IF BN[i]>B[j] THEN
313                 HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
314             ENDIF
315         ENDFOR
316     ELSE
317         HN[i]:=BN[i]/l[1]
318         FOR j:=2 TO 47 DO
319             IF BN[i]<-B[j] THEN
320                 HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
321             ENDIF
322         ENDFOR
323     ENDIF
324 ENDFOR
325
326 -- PERMEANCIAS DOS ENROLAMENTOS
327 P1:=(BN[1]/HN[1])*Aw/(Lw1)
328 P2:=(BN[1]/HN[1])*Aw/(Lw2)
329 P3:=(BN[1]/HN[1])*Aw/(Lw3)
330
331 P4:=(BN[3]/HN[3])*Aw/(Lw1)
332 P5:=(BN[3]/HN[3])*Aw/(Lw2)
333 P6:=(BN[3]/HN[3])*Aw/(Lw3)
334
335 P7:=(BN[5]/HN[5])*Aw/(Lw1)
336 P8:=(BN[5]/HN[5])*Aw/(Lw2)
337 P9:=(BN[5]/HN[5])*Aw/(Lw3)
338
339 -- PERMEANCIAS DE DISPERSAO
340 P10:=Ld1/(N1*N1)
341 P11:=Ld2/(N2*N2)
342 P12:=Ld3/(N3*N3)
343 P13:=Ld1/(N1*N1)
344 P14:=Ld2/(N2*N2)

```

```

345 P15:=Ld3/(N3*N3)
346     P16:=Ld1/(N1*N1)
347 P17:=Ld2/(N2*N2)
348 P18:=Ld3/(N3*N3)
349
350     -- Fluxo principal
351     O1:=BN[1]*Aw*0.96
352     O2:=BN[1]*Aw*0.96
353     O3:=BN[1]*Aw*0.96
354     O4:=BN[3]*Aw*0.96
355     O5:=BN[3]*Aw*0.96
356     O6:=BN[3]*Aw*0.96
357     O7:=BN[5]*Aw*0.96
358     O8:=BN[5]*Aw*0.96
359     O9:=BN[5]*Aw*0.96
360
361     --Fluxo disperso
362     O10:=P10*((HN[1]*Lw1)+(O1/P1))
363     O11:=P11*((HN[1]*Lw2)+(O2/P2))
364     O12:=P12*((HN[1]*Lw3)+(O3/P3))
365     O13:=P13*((HN[3]*Lw1)+(O4/P4))
366     O14:=P14*((HN[3]*Lw2)+(O5/P5))
367     O15:=P15*((HN[3]*Lw3)+(O6/P6))
368     O16:=P16*((HN[5]*Lw1)+(O7/P7))
369     O17:=P17*((HN[5]*Lw2)+(O8/P8))
370     O18:=P18*((HN[5]*Lw3)+(O9/P9))
371
372     --Fluxo nos jugos e retornos
373     O21:=(O1-O10)/2
374     O22:=O21
375     O19:=O22
376     O23:=(O7-O16)/2
377     O24:=O23
378     O20:=O24
379
380     BN[7]:=O19/(Ar*0.96)
381     BN[8]:=O20/(Ar*0.96)
382     BN[9]:=O21/(Ay*0.96)
383     BN[10]:=O22/(Ay*0.96)
384     BN[11]:=O23/(Ay*0.96)
385     BN[12]:=O24/(Ay*0.96)
386
387     FOR i:=7 TO 12 DO
388         IF BN[i]>=0 THEN                                     -- Calculo de H para os trechos do sistema
389             HN[i]:=BN[i]/l[1]
390             FOR j:=2 TO 47 DO
391                 IF BN[i]>B[j] THEN
392                     HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
393             ENDIF
394         ENDFOR
395     ELSE
396         HN[i]:=BN[i]/l[1]
397         FOR j:=2 TO 47 DO
398             IF BN[i]<-B[j] THEN
399                 HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
400             ENDIF
401         ENDFOR
402     ENDIF
403 ENDFOR
404

```

```

405     P19:=(BN[7]/HN[7])*Ar/Lr
406     P20:=(BN[8]/HN[8])*Ar/Lr
407     P21:=(BN[9]/HN[9])*Ay/Ly
408     P22:=(BN[10]/HN[10])*Ay/Ly
409     P23:=(BN[11]/HN[11])*Ay/Ly
410     P24:=(BN[12]/HN[12])*Ay/Ly
411
412     BN1:=BN[1]
413     BN2:=BN[2]
414     BN3:=BN[3]
415     BN4:=BN[4]
416     BN5:=BN[5]
417     BN6:=BN[6]
418     BN7:=BN[7]
419     BN8:=BN[8]
420     BN9:=BN[9]
421     BN10:=BN[10]
422     BN11:=BN[11]
423     BN12:=BN[12]
424
425     HN1:=HN[1]
426     HN2:=HN[2]
427     HN3:=HN[3]
428     HN4:=HN[4]
429     HN5:=HN[5]
430     HN6:=HN[6]
431     HN7:=HN[7]
432     HN8:=HN[8]
433     HN9:=HN[9]
434     HN10:=HN[10]
435     HN11:=HN[11]
436     HN12:=HN[12]
437
438     -- Essa parte corresponde ao calculo das matrizes RLib
439     m:=900
440     lMtrxA := 24
441     cMtrxA := 9
442     MtrxA[1..m]:=0
443     MtrxA[1..cMtrxA] :=[ 1,-1, 0, 0, 0, 0, 0, 0, 0]
444     MtrxA[cMtrxA+1..2*cMtrxA] :=[ 0, 1,-1, 0, 0, 0, 0, 0, 0]
445     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[ 0, 0, 1, 0, 0, 0, 0, 0, 0]
446     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[ 0, 0, 0, 1,-1, 0, 0, 0, 0]
447     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[ 0, 0, 0, 0, 1,-1, 0, 0, 0]
448     MtrxA[5*cMtrxA+1..6*cMtrxA] :=[ 0, 0, 0, 0, 0, 1, 0, 0, 0]
449     MtrxA[6*cMtrxA+1..7*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 1,-1, 0]
450     MtrxA[7*cMtrxA+1..8*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 1,-1]
451     MtrxA[8*cMtrxA+1..9*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 1]
452     MtrxA[9*cMtrxA+1..10*cMtrxA] :=[ -1, 1, 0,0, 0, 0, 0, 0, 0]
453     MtrxA[10*cMtrxA+1..11*cMtrxA] :=[ 0,-1, 1, 0, 0, 0, 0, 0, 0]
454     MtrxA[11*cMtrxA+1..12*cMtrxA] :=[ 0, 0,-1, 0, 0, 0, 0, 0, 0]
455     MtrxA[12*cMtrxA+1..13*cMtrxA] :=[ 0, 0, 0,-1, 1, 0, 0, 0, 0]
456     MtrxA[13*cMtrxA+1..14*cMtrxA] :=[ 0, 0, 0, 0,-1, 1, 0, 0, 0]
457     MtrxA[14*cMtrxA+1..15*cMtrxA] :=[ 0, 0, 0, 0, 0,-1, 0, 0, 0]
458     MtrxA[15*cMtrxA+1..16*cMtrxA] :=[ 0, 0, 1, 0, 0, 0,-1, 1, 0]
459     MtrxA[16*cMtrxA+1..17*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0,-1, 1]
460     MtrxA[17*cMtrxA+1..18*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0,-1]
461     MtrxA[18*cMtrxA+1..19*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
462     MtrxA[19*cMtrxA+1..20*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
463     MtrxA[20*cMtrxA+1..21*cMtrxA] :=[ -1, 0, 0, 0,0, 0, 0, 0, 0]
464     MtrxA[21*cMtrxA+1..22*cMtrxA] :=[ -1, 0, 0, 1,0, 0, 0, 0, 0]

```



```

465     MtrxA[22*cMtrxA+1..23*cMtrxA] :=[ 0, 0, 0, 1,0, 0, -1 ,0 ,0]
466     MtrxA[23*cMtrxA+1..24*cMtrxA] :=[ 0, 0, 0, 0,0, 0, -1 ,0 ,0]
467
468
469
470     lMtrxN := 9
471     cMtrxN := 9
472     MtrxN[1..m]:=0
473     MtrxN[0*cMtrxN+1] := N1
474     MtrxN[1*cMtrxN+2] := N2
475     MtrxN[2*cMtrxN+3] := N3
476     MtrxN[3*cMtrxN+4] := N1
477     MtrxN[4*cMtrxN+5] := N2
478     MtrxN[5*cMtrxN+6] := N3
479     MtrxN[6*cMtrxN+7] := N1
480     MtrxN[7*cMtrxN+8] := N2
481     MtrxN[8*cMtrxN+9] := N3
482
483     lMtrxV := 9
484     cMtrxV := 1
485     MtrxV[1..m]:= 0
486     MtrxV[0*cMtrxV+1] := v1[1]
487     MtrxV[1*cMtrxV+1] := v2[1]
488     MtrxV[2*cMtrxV+1] := v3[1]
489     MtrxV[3*cMtrxV+1] := v1[2]
490     MtrxV[4*cMtrxV+1] := v2[2]
491     MtrxV[5*cMtrxV+1] := v3[2]
492     MtrxV[6*cMtrxV+1] := v1[3]
493     MtrxV[7*cMtrxV+1] := v2[3]
494     MtrxV[8*cMtrxV+1] := v3[3]
495
496     lMtrxO := 9
497     cMtrxO := 1
498     MtrxO[1..m]:=0
499     MtrxO[0*cMtrxO+1] := O1
500     MtrxO[1*cMtrxO+1] := O2
501     MtrxO[2*cMtrxO+1] := O3
502     MtrxO[3*cMtrxO+1] := O4
503     MtrxO[4*cMtrxO+1] := O5
504     MtrxO[5*cMtrxO+1] := O6
505     MtrxO[6*cMtrxO+1] := O7
506     MtrxO[7*cMtrxO+1] := O8
507     MtrxO[8*cMtrxO+1] := O9
508
509     lMtrxInsIn := 9
510     cMtrxInsIn := 1
511     MtrxInsIn[1..m]:=0
512     MtrxInsIn[0*cMtrxInsIn+1] := ins1
513     MtrxInsIn[1*cMtrxInsIn+1] := ins2
514     MtrxInsIn[2*cMtrxInsIn+1] := ins3
515     MtrxInsIn[3*cMtrxInsIn+1] := ins4
516     MtrxInsIn[4*cMtrxInsIn+1] := ins5
517     MtrxInsIn[5*cMtrxInsIn+1] := ins6
518     MtrxInsIn[6*cMtrxInsIn+1] := ins7
519     MtrxInsIn[7*cMtrxInsIn+1] := ins8
520     MtrxInsIn[8*cMtrxInsIn+1] := ins9
521
522     dt := timestep
523
524     lMtrxP := 24,

```

```
525     cMtrxP := 24,
526     MtrxP[1..m] := 0
527     MtrxP[(1-1)*cMtrxP + 1] := P1
528     MtrxP[(2-1)*cMtrxP + 2] := P2
529     MtrxP[(3-1)*cMtrxP + 3] := P3
530     MtrxP[(4-1)*cMtrxP + 4] := P4
531     MtrxP[(5-1)*cMtrxP + 5] := P5
532     MtrxP[(6-1)*cMtrxP + 6] := P6
533     MtrxP[(7-1)*cMtrxP + 7] := P7
534     MtrxP[(8-1)*cMtrxP + 8] := P8
535     MtrxP[(9-1)*cMtrxP + 9] := P9
536     MtrxP[(10-1)*cMtrxP + 10] := P10
537     MtrxP[(11-1)*cMtrxP + 11] := P11
538     MtrxP[(12-1)*cMtrxP + 12] := P12
539     MtrxP[(13-1)*cMtrxP + 13] := P13
540     MtrxP[(14-1)*cMtrxP + 14] := P14
541     MtrxP[(15-1)*cMtrxP + 15] := P15
542     MtrxP[(16-1)*cMtrxP + 16] := P16
543     MtrxP[(17-1)*cMtrxP + 17] := P17
544     MtrxP[(18-1)*cMtrxP + 18] := P18
545     MtrxP[(19-1)*cMtrxP + 19] := P19
546     MtrxP[(20-1)*cMtrxP + 20] := P20
547     MtrxP[(21-1)*cMtrxP + 21] := P21
548     MtrxP[(22-1)*cMtrxP + 22] := P22
549     MtrxP[(23-1)*cMtrxP + 23] := P23
550     MtrxP[(24-1)*cMtrxP + 24] := P24
551
552     USE RNSMatrix as RNSM1
553     INPUT
554     deltaTime := dt,
555
556     lMatrixA := lMtrxA,
557     cMatrixA := cMtrxA,
558     MatrixA[1..m] := MtrxA[1..m]
559
560     lMP := lMtrxP,
561     cMP := cMtrxP,
562     MP[1..900] := MtrxP[1..m]
563
564     lMatrixN := lMtrxN,
565     cMatrixN := cMtrxN,
566     MatrixN[1..m] := MtrxN[1..m]
567
568     lMatrixV := lMtrxV,
569     cMatrixV := cMtrxV,
570     MatrixV[1..m] := MtrxV[1..m]
571
572     lMatrixO := lMtrxO,
573     cMatrixO := cMtrxO,
574     MatrixO[1..m] := MtrxO[1..m]
575
576     lMatrixInsIn := lMtrxInsIn,
577     cMatrixInsIn := cMtrxInsIn,
578     MatrixInsIn[1..m] := MtrxInsIn[1..m]
579     OUTPUT
580     MtrxIns[1..m] := MatrixIns[1..m],
581     MtrxIs[1..m] := MatrixIs[1..m]
582     ENDUSE
583     ins1 := MtrxIns[1]
584     ins2 := MtrxIns[2]
```

```
585     ins3 := MtrxIns[3]
586     ins4 := MtrxIns[4]
587     ins5 := MtrxIns[5]
588     ins6 := MtrxIns[6]
589     ins7 := MtrxIns[7]
590     ins8 := MtrxIns[8]
591     ins9 := MtrxIns[9]
592
593     i1[1] := MtrxIs[1]
594     i2[1] := MtrxIs[2]
595     i3[1] := MtrxIs[3]
596     i1[2] := MtrxIs[4]
597     i2[2] := MtrxIs[5]
598     i3[2] := MtrxIs[6]
599     i1[3] := MtrxIs[7]
600     i2[3] := MtrxIs[8]
601     i3[3] := MtrxIs[9]
602
603     i1A:=i1[1]+(i2[2])-(i2[1]/2)-(i2[3]/2)-(i1[3]/2)-(i1[2]/2)
604     --      io1A+(io2B)-(io2A/2)-(io2C/2)-(io1C/2)-(io1B/2)
605
606     i1B:=i1[2]+(i2[3])-(i2[2]/2)-(i2[1]/2)-(i1[1]/2)-(i1[3]/2)
607     --      io1B+(io2C)-(io2B/2)-(io2A/2)-(io1A/2)-(io1C/2)
608
609     i1C:=i1[3]+(i2[1])-(i2[2]/2)-(i2[3]/2)-(i1[1]/2)-(i1[2]/2)
610     --      io1C+(io2A)-(io2B/2)-(io2C/2)-(io1A/2)-(io1B/2)
611
612     i2A:=i2[1]
613     i2B:=i2[2]
614     i2C:=i2[3]
615
616     i3A:=i3[1]
617     i3B:=i3[2]
618     i3C:=i3[3]
619
620     v1A:=v1[1]
621     v2A:=v2[1]
622     v3A:=v3[1]
623     v1B:=v1[2]
624     v2B:=v2[2]
625     v3B:=v3[2]
626     v1C:=v1[3]
627     v2C:=v2[3]
628     v3C:=v3[3]
629
630     ENDEXEC
631 ENDMODEL
632 RECORD
633     RNS9.HN1 AS HN1
634     RNS9.BN1 AS BN1
635     RNS9.HN3 AS HN3
636     RNS9.BN3 AS BN3
637     RNS9.HN5 AS HN5
638     RNS9.BN5 AS BN5
639     RNS9.HN7 AS HN7
640     RNS9.BN7 AS BN7
641     RNS9.HN8 AS HN8
642     RNS9.BN8 AS BN8
643     RNS9.BN9 AS BN9
644     RNS9.HN9 AS HN9
```

645 RNS9.BN10 **AS** BN10
646 RNS9.HN10 **AS** HN10
647 RNS9.BN11 **AS** BN11
648 RNS9.HN11 **AS** HN11
649 RNS9.BN12 **AS** BN12
650 RNS9.HN12 **AS** HN12
651 RNS9.i1A **AS** i1A
652 RNS9.i2A **AS** i2A
653 RNS9.i3A **AS** i3A
654 RNS9.i1B **AS** i1B
655 RNS9.i2B **AS** i2B
656 RNS9.i3B **AS** i3B
657 RNS9.i1C **AS** i1C
658 RNS9.i2C **AS** i2C
659 RNS9.i3C **AS** i3C
660 RNS9.ins1 **AS** ins1
661 RNS9.ins2 **AS** ins2
662 RNS9.ins3 **AS** ins3
663 RNS9.ins4 **AS** ins4
664 RNS9.ins5 **AS** ins5
665 RNS9.ins6 **AS** ins6
666 RNS9.ins7 **AS** ins7
667 RNS9.ins8 **AS** ins8
668 RNS9.ins9 **AS** ins9
669 RNS9.v1A **AS** v1A
670 RNS9.v2A **AS** v2A
671 RNS9.v3A **AS** v3A
672 RNS9.v1B **AS** v1B
673 RNS9.v2B **AS** v2B
674 RNS9.v3B **AS** v3B
675 RNS9.v1C **AS** v1C
676 RNS9.v2C **AS** v2C
677 RNS9.v3C **AS** v3C
678 RNS9.io1A **AS** io1A
679 RNS9.io2A **AS** io2A
680 RNS9.io1B **AS** io1B
681 RNS9.io2B **AS** io2B
682 RNS9.io1C **AS** io1C
683 RNS9.io2C **AS** io2C
684 RNS9.v11p **AS** v11p
685 RNS9.v12p **AS** v12p
686 RNS9.v13p **AS** v13p
687 RNS9.v21p **AS** v21p
688 RNS9.v22p **AS** v22p
689 RNS9.v23p **AS** v23p
690 RNS9.O1 **AS** O1
691 RNS9.O2 **AS** O2
692 RNS9.O3 **AS** O3
693 RNS9.O4 **AS** O4
694 RNS9.O5 **AS** O5
695 RNS9.O6 **AS** O6
696 RNS9.O7 **AS** O7
697 RNS9.O8 **AS** O8
698 RNS9.O9 **AS** O9
699 RNS9.O10 **AS** O10
700 RNS9.O11 **AS** O11
701 RNS9.O12 **AS** O12
702 RNS9.O13 **AS** O13
703 RNS9.O14 **AS** O14
704 RNS9.O15 **AS** O15

```

705 RNS9.O16 AS O16
706 RNS9.O17 AS O17
707 RNS9.O18 AS O18
708 RNS9.O19 AS O19
709 RNS9.O20 AS O20
710 RNS9.O21 AS O21
711 RNS9.O22 AS O22
712 RNS9.O23 AS O23
713 RNS9.O24 AS O24
714 USE RNS9 AS RNS9
715 INPUT
716 v[1]:= M0001A
717 v[2]:= M0001B
718 v[3]:= M0001C
719 DATA
720 Aw:= 0.0158
721 Ay:= 0.0158
722 Ar:= 0.0158
723 N1:= 145.
724 N2:= 77.
725 N3:= 98.
726 Lw1:= 0.29
727 Lw2:= 0.154
728 Lw3:= 0.27
729 Ly:= 0.3608
730 Lr:= 0.8
731 Ld1:= 5.119E-5
732 Ld2:= 1.279E-4
733 Ld3:= 0.0108
734 OUTPUT
735 X0001A:=v3[1]
736 X0001B:=v3[2]
737 X0001C:=v3[3]
738 ENDUSE
739 ENDMODELS
740 C 1 2 3 4 5 6 7 8
741 C 34567890123456789012345678901234567890123456789012345678901234567890
742 /BRANCH
743 C < n1 >< n2 ><ref1><ref2>< R >< L >< C >
744 C < n1 >< n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
745 X0002A 1. 0
746 X0002B 1. 0
747 X0002C 1. 0
748 /SOURCE
749 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
750 14X0002A 2449.5 60. -1. 100.
751 14X0002B 2449.5 60. -120. -1. 100.
752 14X0002C 2449.5 60. -240. -1. 100.
753 /OUTPUT
754 BLANK MODELS
755 BLANK BRANCH
756 BLANK SWITCH
757 BLANK SOURCE
758 BLANK OUTPUT
759 BLANK PLOT
760 BEGIN NEW DATA CASE
761 BLANK

```

```

1 BEGIN NEW DATA CASE
2 C -----
3 C Generated by ATPDRAW dezembro, quinta-feira 3, 2015
4 C A Bonneville Power Administration program
5 C by H. K. Hoidalen at SEFAS/NTNU - NORWAY 1994-2009
6 C -----
7 POWER FREQUENCY                60.
8 C dT >< Tmax >< Xopt >< Copt ><Epsiln>
9   5.E-6      .05      60.      1.E-180
10  500      1      1      1      1      0      0      1      0
11 /MODELS
12 MODELS
13 INPUT
14 M0001A {v(X0002A)}
15 M0001B {v(X0002B)}
16 M0001C {v(X0002C)}
17 OUTPUT
18   X0001A
19   X0001B
20   X0001C
21 MODEL RNS9
22 INPUT v[1..3]
23 DATA Aw
24   Ay
25   Ar
26   N1
27   N2
28     N3
29   Lw1
30   Lw2
31     Lw3
32   Ly
33   Lr
34   Ld1
35   Ld2
36     Ld3
37 VAR B[1..48]          -- Campo magnetico dos nove trechos
38   H[1..48]           -- Densidade magnetica para os nove trechos
39   HN[1..15]
40   BN[1..15]
41     HN1,HN2,HN3,HN4,HN5,HN6,HN7,HN8,HN9,HN10,HN11,HN12,HN13,HN14,HN15
42     BN1,BN2,BN3,BN4,BN5,BN6,BN7,BN8,BN9,BN10,BN11,BN12,BN13,BN14,BN15
43   l[1..47]          -- Indutancias para os nove trechos
44   lp[2..47]        -- Indutancias linhas (l') para os nove trechos
45   d1,d2,d3,d4,d5,d6,d7,d8,d9
46   P1,P2,P3,P4,P5,P6,P7,P8,P9
47   P10,P11,P12,P13,P14,P15,P16,P17,P18
48     P19, P20, P21, P22, P23, P24
49   O1,O2,O3,O4,O5,O6,O7,O8,O9
50   O10,O11,O12,O13,O14,O15,O16,O17,O18
51     O19, O20, O21, O22, O23, O24
52   ins1,ins2,ins3,ins4,ins5,ins6,ins7,ins8,ins9
53     i1A,i1B,i1C
54     i2A,i2B,i2C
55     i3A,i3B,i3C
56   i1[1..3],i2[1..3],i3[1..3]
57   v1A,v1B,v1C
58     v2A,v2B,v2C
59     v3A,v3B,v3C

```

```
60     v1[1..3],v2[1..3],v3[1..3]
61     v1p, v2p, v3p
62     v11p,v12p,v13p
63     v21p,v22p,v23p
64     io1A,io1B, io1C
65     io2A,io2B,io2C
66     io3A,io3B,io3C
67     --Library Variables
68     m, dt, determnt, aux,
69     lMtrxA, cMtrxA, MtrxA[1..900]
70     lMtrxP, cMtrxP, MtrxP[1..900]
71     lMtrxN, cMtrxN, MtrxN[1..900]
72     lMtrxV, cMtrxV, MtrxV[1..900]
73     lMtrxO, cMtrxO, MtrxO[1..900]
74     lMtrxI, cMtrxI, MtrxI[1..900]
75     lMtrxInsIn, cMtrxInsIn, MtrxInsIn[1..900]
76     lMtrxIns, cMtrxIns, MtrxIns[1..900]
77     lMtrxIs, cMtrxIs, MtrxIs[1..900]
78     Hc -- for hysteresis
79 OUTPUT v3[1..3]
80 $INCLUDE ./RLib.m
81 INIT
82     HN[1..15]:=0
83     BN[1..15]:=0
84     ins1:=0.01
85     ins2:=0.01
86     ins3:=0.01
87     ins4:=0.01
88     ins5:=0.01
89     ins6:=0.01
90     ins7:=0.01
91     ins8:=0.01
92     ins9:=0.01
93     i1[1..3]:=0.01
94     i2[1..3]:=0.01
95     i3[1..3]:=0.01
96     v1[1..3]:=0.01
97     v2[1..3]:=0.01
98     v3[1..3]:=0.01
99     i1A:=0.01
100    i2A:=0.01
101    i3A:=0.01
102    i1B:=0.01
103    i2B:=0.01
104    i3B:=0.01
105    i1C:=0.01
106    i2C:=0.01
107    i3C:=0.01
108    v1A:=0.01
109    v2A:=0.01
110    v3A:=0.01
111    v1B:=0.01
112    v2B:=0.01
113    v3B:=0.01
114    v1C:=0.01
115    v2C:=0.01
116    v3C:=0.01
117    P1:=0.01
118    P2:=0.01
119    P3:=0.01
```

```
120 P4:=0.01
121 P5:=0.01
122 P6:=0.01
123 P13:=0.01
124 P14:=0.01
125 P15:=0.01
126 P16:=0.01
127 P17:=0.01
128 P18:=0.01
129 P19:=0.01
130 P20:=0.01
131 P21:=0.01
132 P22:=0.01
133 P23:=0.01
134 P24:=0.01
135 v1p:=0
136 v2p:=0
137 v3p:=0
138 v11p:=0
139 v12p:=0
140 v13p:=0
141 v21p:=0
142 v22p:=0
143 v23p:=0
144 io1A:=0
145 io2A:=0
146 io3A:=0
147 io1B:=0
148 io2B:=0
149 io3B:=0
150 io1C:=0
151 io2C:=0
152 io3C:=0
153 ENDINIT
154 EXEC
155 Hc:=53.133
156 H[1]:=11.937
157 B[1]:=0.4
158 H[2]:=13.608
159 B[2]:=0.499999
160 H[3]:=15.358
161 B[3]:=0.599998
162 H[4]:=17.109
163 B[4]:=0.699997
164 H[5]:=18.701
165 B[5]:=0.799995
166 H[6]:=20.292
167 B[6]:=0.899994
168 H[7]:=21.168
169 B[7]:=0.999992
170 H[8]:=21.884
171 B[8]:=1.049991
172 H[9]:=22.839
173 B[9]:=1.099991
174 H[10]:=23.635
175 B[10]:=1.14999
176 H[11]:=24.987
177 B[11]:=1.19999
178 H[12]:=26.42
179 B[12]:=1.24999
```



```
180 H[13]:=28.17
181 B[13]:=1.29999
182 H[14]:=30.637
183 B[14]:=1.349991
184 H[15]:=33.025
185 B[15]:=1.399992
186 H[16]:=37.242
187 B[16]:=1.449995
188 H[17]:=42.972
189 B[17]:=1.499998
190 H[18]:=45.757
191 B[18]:=1.52
192 H[19]:=48.94
193 B[19]:=1.540003
194 H[20]:=53.317
195 B[20]:=1.560006
196 H[21]:=58.887
197 B[21]:=1.580011
198 H[22]:=66.049
199 B[22]:=1.600016
200 H[23]:=76.394
201 B[23]:=1.620025
202 H[24]:=85.944
203 B[24]:=1.640033
204 H[25]:=103.451
205 B[25]:=1.660048
206 H[26]:=119.366
207 B[26]:=1.680062
208 H[27]:=147.218
209 B[27]:=1.700087
210 H[28]:=163.134
211 B[28]:=1.710101
212 H[29]:=179.049
213 B[29]:=1.720115
214 H[30]:=202.923
215 B[30]:=1.730136
216 H[31]:=226.796
217 B[31]:=1.740157
218 H[32]:=259.423
219 B[32]:=1.750186
220 H[33]:=282.5
221 B[33]:=1.760207
222 H[34]:=334.225
223 B[34]:=1.770253
224 H[35]:=377.993
225 B[35]:=1.780292
226 H[36]:=445.634
227 B[36]:=1.790352
228 H[37]:=501.338
229 B[37]:=1.800402
230 H[38]:=612.747
231 B[38]:=1.810502
232 H[39]:=700.282
233 B[39]:=1.82058
234 H[40]:=835.564
235 B[40]:=1.830701
236 H[41]:=954.93
237 B[41]:=1.840808
238 H[42]:=1114.085
239 B[42]:=1.850951
```

```

240 H[43]:=1193.662
241 B[43]:=1.861022
242 H[44]:=1392.606
243 B[44]:=1.871201
244 H[45]:=1591.549
245 B[45]:=1.881379
246 H[46]:=2500
247 B[46]:=1.922194
248 H[47]:=5000
249 B[47]:=1.964438
250 H[48]:=10000
251 B[48]:=1.978929
252
253 -- Metodo de aproximação linear por trechos
254 -- Esse loop calcula a indutancia de cada trecho
255 FOR i:=1 TO 47 DO
256     l[i]:= (B[i+1]-B[i]) / (H[i+1]-H[i])
257 ENDFOR
258
259 -- Metodo de aproximação linear por trechos parte 2
260 -- Segundo loop para calcular o l' para cada trecho
261 -- Utilizar Lp' para criar indutores equivalentes que facam
262 -- o mesmo efeito na indutancia total.
263
264 FOR i:=2 TO 47 DO
265     lp[i]:= 1 / ((1/l[i]) - (1/l[i-1]))
266 ENDFOR
267
268 v1[1]:=v[1]
269 v1[2]:=v[2]
270 v1[3]:=v[3]
271
272 IF t=5e-5 THEN
273     v1p:=v[1]/cos(377*t)
274     v2p:=v[2]/cos(377*t-2.094)
275     v3p:=v[3]/cos(377*t+2.094)
276 ENDF
277 v1[1]:= ((0.743*v1p) *cos(377*t)
278 v1[2]:= ((0.743*v2p) *cos(377*t-2.094)
279 v1[3]:= ((0.743*v3p) *cos(377*t+2.094)
280 IF t=5e-5 THEN
281     v11p:=v1[1]/cos(377*t)
282     v12p:=v1[2]/cos(377*t-2.094)
283     v13p:=v1[3]/cos(377*t+2.094)
284 ENDF
285 v2[1]:= ((N2/N1) *v11p) *cos(377*t)
286 v2[2]:= ((N2/N1) *v12p) *cos(377*t-2.094)
287 v2[3]:= ((N2/N1) *v13p) *cos(377*t+2.094)
288 v21p:= ((N2/N1) *v11p)
289 v22p:= ((N2/N1) *v12p)
290 v23p:= ((N2/N1) *v13p)
291 v3[1]:= (v11p*0.675) *cos(377*t-0.524)
292 v3[2]:= (v12p*0.675) *cos(377*t-2.094-0.524)
293 v3[3]:= (v13p*0.675) *cos(377*t+2.094-0.524)
294
295 --Enrolamentos da esquerda
296 dl:=Aw*(N1+N2)*0.96
297 BN[1]:= ((v11p+v21p) / (377)) *sin(377*t) / dl
298 BN[2]:=BN[1]
299

```

```

300 --Enrolamentos do MEIO
301 d3:=Aw*(N1+N2)*0.96
302 BN[3]:=((v12p+v22p)/(377))*sin(377*t-2.094)/d3
303 BN[4]:=BN[3]
304
305 --Enrolamentos da direita
306 d5:=Aw*(N1+N2)*0.96
307 BN[5]:=((v13p+v23p)/(377))*sin(377*t+2.094)/d5
308 BN[6]:=BN[5]
309
310 FOR i:=1 TO 6 DO
311     IF BN[i]>=0 THEN -- Calculo de H para os trechos do sistema
312         HN[i]:=BN[i]/l[1]
313         FOR j:=2 TO 47 DO
314             IF BN[i]>B[j] THEN
315                 HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
316             ENDIF
317         ENDFOR
318     ELSE
319         HN[i]:=BN[i]/l[1]
320         FOR j:=2 TO 47 DO
321             IF BN[i]<-B[j] THEN
322                 HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
323             ENDIF
324         ENDFOR
325     ENDIF
326     IF v[1]<0 THEN
327         HN[i]:=HN[i]-Hc
328     ELSE
329         HN[i]:=HN[i]+Hc
330     ENDIF
331 ENDFOR
332
333 -- PERMEANCIAS DOS ENROLAMENTOS
334 P1:=(BN[1]/HN[1])*Aw/(Lw1)
335 P2:=(BN[1]/HN[1])*Aw/(Lw2)
336 P3:=(BN[1]/HN[1])*Aw/(Lw3)
337
338 P4:=(BN[3]/HN[3])*Aw/(Lw1)
339 P5:=(BN[3]/HN[3])*Aw/(Lw2)
340 P6:=(BN[3]/HN[3])*Aw/(Lw3)
341
342 P7:=(BN[5]/HN[5])*Aw/(Lw1)
343 P8:=(BN[5]/HN[5])*Aw/(Lw2)
344 P9:=(BN[5]/HN[5])*Aw/(Lw3)
345
346 -- PERMEANCIAS DE DISPERSAO
347 P10:=Ld1/(N1*N1)
348 P11:=Ld2/(N2*N2)
349 P12:=Ld3/(N3*N3)
350 P13:=Ld1/(N1*N1)
351 P14:=Ld2/(N2*N2)
352 P15:=Ld3/(N3*N3)
353 P16:=Ld1/(N1*N1)
354 P17:=Ld2/(N2*N2)
355 P18:=Ld3/(N3*N3)
356
357 -- Fluxo principal
358 O1:=BN[1]*Aw*0.96
359 O2:=BN[1]*Aw*0.96

```

```

360      O3:=BN[1]*Aw*0.96
361      O4:=BN[3]*Aw*0.96
362      O5:=BN[3]*Aw*0.96
363      O6:=BN[3]*Aw*0.96
364      O7:=BN[5]*Aw*0.96
365      O8:=BN[5]*Aw*0.96
366      O9:=BN[5]*Aw*0.96
367
368      --Fluxo disperso
369      O10:=P10*((HN[1]*Lw1)+(O1/P1))
370      O11:=P11*((HN[1]*Lw2)+(O2/P2))
371      O12:=P12*((HN[1]*Lw3)+(O3/P3))
372      O13:=P13*((HN[3]*Lw1)+(O4/P4))
373      O14:=P14*((HN[3]*Lw2)+(O5/P5))
374      O15:=P15*((HN[3]*Lw3)+(O6/P6))
375      O16:=P16*((HN[5]*Lw1)+(O7/P7))
376      O17:=P17*((HN[5]*Lw2)+(O8/P8))
377      O18:=P18*((HN[5]*Lw3)+(O9/P9))
378
379      --Fluxo nos jugos e retornos
380      O21:=(O1-O10)/2
381      O22:=O21
382      O19:=O22
383      O23:=(O7-O16)/2
384      O24:=O23
385      O20:=O24
386
387      BN[7]:=O19/(Ar*0.96)
388      BN[8]:=O20/(Ar*0.96)
389      BN[9]:=O21/(Ay*0.96)
390      BN[10]:=O22/(Ay*0.96)
391      BN[11]:=O23/(Ay*0.96)
392      BN[12]:=O24/(Ay*0.96)
393
394      FOR i:=7 TO 12 DO
395          IF BN[i]>=0 THEN                                -- Calculo de H para os trechos do sistema
396              HN[i]:=BN[i]/l[1]
397              FOR j:=2 TO 47 DO
398                  IF BN[i]>B[j] THEN
399                      HN[i]:=HN[i]+((BN[i]-B[j])/lp[j])
400                  ENDIF
401              ENDFOR
402          ELSE
403              HN[i]:=BN[i]/l[1]
404              FOR j:=2 TO 47 DO
405                  IF BN[i]<-B[j] THEN
406                      HN[i]:=HN[i]+((BN[i]+B[j])/lp[j])
407                  ENDIF
408              ENDFOR
409          ENDIF
410      ENDFOR
411
412      P19:=(BN[7]/HN[7])*Ar/Lr
413      P20:=(BN[8]/HN[8])*Ar/Lr
414      P21:=(BN[9]/HN[9])*Ay/Ly
415      P22:=(BN[10]/HN[10])*Ay/Ly
416      P23:=(BN[11]/HN[11])*Ay/Ly
417      P24:=(BN[12]/HN[12])*Ay/Ly
418
419      BN1:=BN[1]

```

```

420     BN2:=BN[2]
421     BN3:=BN[3]
422     BN4:=BN[4]
423     BN5:=BN[5]
424     BN6:=BN[6]
425     BN7:=BN[7]
426     BN8:=BN[8]
427     BN9:=BN[9]
428     BN10:=BN[10]
429     BN11:=BN[11]
430     BN12:=BN[12]
431
432     HN1:=HN[1]
433     HN2:=HN[2]
434     HN3:=HN[3]
435     HN4:=HN[4]
436     HN5:=HN[5]
437     HN6:=HN[6]
438     HN7:=HN[7]
439     HN8:=HN[8]
440     HN9:=HN[9]
441     HN10:=HN[10]
442     HN11:=HN[11]
443     HN12:=HN[12]
444
445     -- Essa parte corresponde ao calculo das matrizes RLib
446     m:=900
447     lMtrxA := 24
448     cMtrxA := 9
449     MtrxA[1..m]:=0
450     MtrxA[1..cMtrxA] :=[ 1,-1, 0, 0, 0, 0, 0, 0, 0, 0]
451     MtrxA[cMtrxA+1..2*cMtrxA] :=[ 0, 1,-1, 0, 0, 0, 0, 0, 0, 0]
452     MtrxA[2*cMtrxA+1..3*cMtrxA] :=[ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
453     MtrxA[3*cMtrxA+1..4*cMtrxA] :=[ 0, 0, 0, 1,-1, 0, 0, 0, 0, 0]
454     MtrxA[4*cMtrxA+1..5*cMtrxA] :=[ 0, 0, 0, 0, 1,-1, 0, 0, 0, 0]
455     MtrxA[5*cMtrxA+1..6*cMtrxA] :=[ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
456     MtrxA[6*cMtrxA+1..7*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 1,-1, 0, 0]
457     MtrxA[7*cMtrxA+1..8*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 1,-1]
458     MtrxA[8*cMtrxA+1..9*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 1]
459     MtrxA[9*cMtrxA+1..10*cMtrxA] :=[ -1, 1, 0,0, 0, 0, 0, 0, 0, 0]
460     MtrxA[10*cMtrxA+1..11*cMtrxA] :=[ 0,-1, 1, 0, 0, 0, 0, 0, 0, 0]
461     MtrxA[11*cMtrxA+1..12*cMtrxA] :=[ 0, 0,-1, 0, 0, 0, 0, 0, 0, 0]
462     MtrxA[12*cMtrxA+1..13*cMtrxA] :=[ 0, 0, 0,-1, 1, 0, 0, 0, 0, 0]
463     MtrxA[13*cMtrxA+1..14*cMtrxA] :=[ 0, 0, 0, 0,-1, 1, 0, 0, 0, 0]
464     MtrxA[14*cMtrxA+1..15*cMtrxA] :=[ 0, 0, 0, 0, 0,-1, 0, 0, 0, 0]
465     MtrxA[15*cMtrxA+1..16*cMtrxA] :=[ 0, 0, 1, 0, 0, 0,-1, 1, 0, 0]
466     MtrxA[16*cMtrxA+1..17*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0,-1, 1]
467     MtrxA[17*cMtrxA+1..18*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0,-1]
468     MtrxA[18*cMtrxA+1..19*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
469     MtrxA[19*cMtrxA+1..20*cMtrxA] :=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
470     MtrxA[20*cMtrxA+1..21*cMtrxA] :=[ -1, 0, 0, 0,0, 0, 0, 0, 0, 0]
471     MtrxA[21*cMtrxA+1..22*cMtrxA] :=[ -1, 0, 0, 1,0, 0, 0, 0, 0, 0]
472     MtrxA[22*cMtrxA+1..23*cMtrxA] :=[ 0, 0, 0, 1,0, 0, -1, 0, 0, 0]
473     MtrxA[23*cMtrxA+1..24*cMtrxA] :=[ 0, 0, 0, 0,0, 0, -1, 0, 0, 0]
474
475
476
477     lMtrxN := 9
478     cMtrxN := 9
479     MtrxN[1..m]:=0

```

```

480     MtrxN[0*cMtrxN+1] := N1
481     MtrxN[1*cMtrxN+2] := N2
482     MtrxN[2*cMtrxN+3] := N3
483     MtrxN[3*cMtrxN+4] := N1
484     MtrxN[4*cMtrxN+5] := N2
485     MtrxN[5*cMtrxN+6] := N3
486     MtrxN[6*cMtrxN+7] := N1
487     MtrxN[7*cMtrxN+8] := N2
488     MtrxN[8*cMtrxN+9] := N3
489
490     lMtrxV := 9
491     cMtrxV := 1
492     MtrxV[1..m] := 0
493     MtrxV[0*cMtrxV+1] := v1[1]
494     MtrxV[1*cMtrxV+1] := v2[1]
495     MtrxV[2*cMtrxV+1] := v3[1]
496     MtrxV[3*cMtrxV+1] := v1[2]
497     MtrxV[4*cMtrxV+1] := v2[2]
498     MtrxV[5*cMtrxV+1] := v3[2]
499     MtrxV[6*cMtrxV+1] := v1[3]
500     MtrxV[7*cMtrxV+1] := v2[3]
501     MtrxV[8*cMtrxV+1] := v3[3]
502
503     lMtrxO := 9
504     cMtrxO := 1
505     MtrxO[1..m] := 0
506     MtrxO[0*cMtrxO+1] := O1
507     MtrxO[1*cMtrxO+1] := O2
508     MtrxO[2*cMtrxO+1] := O3
509     MtrxO[3*cMtrxO+1] := O4
510     MtrxO[4*cMtrxO+1] := O5
511     MtrxO[5*cMtrxO+1] := O6
512     MtrxO[6*cMtrxO+1] := O7
513     MtrxO[7*cMtrxO+1] := O8
514     MtrxO[8*cMtrxO+1] := O9
515
516     lMtrxInsIn := 9
517     cMtrxInsIn := 1
518     MtrxInsIn[1..m] := 0
519     MtrxInsIn[0*cMtrxInsIn+1] := ins1
520     MtrxInsIn[1*cMtrxInsIn+1] := ins2
521     MtrxInsIn[2*cMtrxInsIn+1] := ins3
522     MtrxInsIn[3*cMtrxInsIn+1] := ins4
523     MtrxInsIn[4*cMtrxInsIn+1] := ins5
524     MtrxInsIn[5*cMtrxInsIn+1] := ins6
525     MtrxInsIn[6*cMtrxInsIn+1] := ins7
526     MtrxInsIn[7*cMtrxInsIn+1] := ins8
527     MtrxInsIn[8*cMtrxInsIn+1] := ins9
528
529     dt := timestep
530
531     lMtrxP := 24,
532     cMtrxP := 24,
533     MtrxP[1..m] := 0
534     MtrxP[(1-1)*cMtrxP + 1] := P1
535     MtrxP[(2-1)*cMtrxP + 2] := P2
536     MtrxP[(3-1)*cMtrxP + 3] := P3
537     MtrxP[(4-1)*cMtrxP + 4] := P4
538     MtrxP[(5-1)*cMtrxP + 5] := P5
539     MtrxP[(6-1)*cMtrxP + 6] := P6

```

```
540 MtrxP[(7-1)*cMtrxP + 7] := P7
541 MtrxP[(8-1)*cMtrxP + 8] := P8
542 MtrxP[(9-1)*cMtrxP + 9] := P9
543 MtrxP[(10-1)*cMtrxP + 10] := P10
544 MtrxP[(11-1)*cMtrxP + 11] := P11
545 MtrxP[(12-1)*cMtrxP + 12] := P12
546 MtrxP[(13-1)*cMtrxP + 13] := P13
547 MtrxP[(14-1)*cMtrxP + 14] := P14
548 MtrxP[(15-1)*cMtrxP + 15] := P15
549 MtrxP[(16-1)*cMtrxP + 16] := P16
550 MtrxP[(17-1)*cMtrxP + 17] := P17
551 MtrxP[(18-1)*cMtrxP + 18] := P18
552 MtrxP[(19-1)*cMtrxP + 19] := P19
553 MtrxP[(20-1)*cMtrxP + 20] := P20
554 MtrxP[(21-1)*cMtrxP + 21] := P21
555 MtrxP[(22-1)*cMtrxP + 22] := P22
556 MtrxP[(23-1)*cMtrxP + 23] := P23
557 MtrxP[(24-1)*cMtrxP + 24] := P24
558
559 USE RNSMatrix as RNSM1
560 INPUT
561 deltaTime := dt,
562
563 lMatrixA := lMtrxA,
564 cMatrixA := cMtrxA,
565 MatrixA[1..m] := MtrxA[1..m]
566
567 lMP := lMtrxP,
568 cMP := cMtrxP,
569 MP[1..900] := MtrxP[1..m]
570
571 lMatrixN := lMtrxN,
572 cMatrixN := cMtrxN,
573 MatrixN[1..m] := MtrxN[1..m]
574
575 lMatrixV := lMtrxV,
576 cMatrixV := cMtrxV,
577 MatrixV[1..m] := MtrxV[1..m]
578
579 lMatrixO := lMtrxO,
580 cMatrixO := cMtrxO,
581 MatrixO[1..m] := MtrxO[1..m]
582
583 lMatrixInsIn := lMtrxInsIn,
584 cMatrixInsIn := cMtrxInsIn,
585 MatrixInsIn[1..m] := MtrxInsIn[1..m]
586 OUTPUT
587 MtrxIns[1..m] := MatrixIns[1..m],
588 MtrxIs[1..m] := MatrixIs[1..m]
589 ENDUSE
590 ins1 := MtrxIns[1]
591 ins2 := MtrxIns[2]
592 ins3 := MtrxIns[3]
593 ins4 := MtrxIns[4]
594 ins5 := MtrxIns[5]
595 ins6 := MtrxIns[6]
596 ins7 := MtrxIns[7]
597 ins8 := MtrxIns[8]
598 ins9 := MtrxIns[9]
599
```

```
600     i1[1] := MtrxIs[1]
601     i2[1] := MtrxIs[2]
602     i3[1] := MtrxIs[3]
603     i1[2] := MtrxIs[4]
604     i2[2] := MtrxIs[5]
605     i3[2] := MtrxIs[6]
606     i1[3] := MtrxIs[7]
607     i2[3] := MtrxIs[8]
608     i3[3] := MtrxIs[9]
609
610     i1A:=i1[1]+(i2[2])-(i2[1]/2)-(i2[3]/2)-(i1[3]/2)-(i1[2]/2)
611     --      io1A+(io2B)-(io2A/2)-(io2C/2)-(io1C/2)-(io1B/2)
612
613     i1B:=i1[2]+(i2[3])-(i2[2]/2)-(i2[1]/2)-(i1[1]/2)-(i1[3]/2)
614     --      io1B+(io2C)-(io2B/2)-(io2A/2)-(io1A/2)-(io1C/2)
615
616     i1C:=i1[3]+(i2[1])-(i2[2]/2)-(i2[3]/2)-(i1[1]/2)-(i1[2]/2)
617     --      io1C+(io2A)-(io2B/2)-(io2C/2)-(io1A/2)-(io1B/2)
618
619     i2A:=i2[1]
620     i2B:=i2[2]
621     i2C:=i2[3]
622
623     i3A:=i3[1]
624     i3B:=i3[2]
625     i3C:=i3[3]
626
627     v1A:=v1[1]
628     v2A:=v2[1]
629     v3A:=v3[1]
630     v1B:=v1[2]
631     v2B:=v2[2]
632     v3B:=v3[2]
633     v1C:=v1[3]
634     v2C:=v2[3]
635     v3C:=v3[3]
636
637     ENDEXEC
638     ENDMODEL
639     RECORD
640     RNS9.HN1 AS HN1
641     RNS9.BN1 AS BN1
642     RNS9.HN3 AS HN3
643     RNS9.BN3 AS BN3
644     RNS9.HN5 AS HN5
645     RNS9.BN5 AS BN5
646     RNS9.HN7 AS HN7
647     RNS9.BN7 AS BN7
648     RNS9.HN8 AS HN8
649     RNS9.BN8 AS BN8
650     RNS9.BN9 AS BN9
651     RNS9.HN9 AS HN9
652     RNS9.BN10 AS BN10
653     RNS9.HN10 AS HN10
654     RNS9.BN11 AS BN11
655     RNS9.HN11 AS HN11
656     RNS9.BN12 AS BN12
657     RNS9.HN12 AS HN12
658     RNS9.i1A AS i1A
659     RNS9.i2A AS i2A
```


660 RNS9.i3A **AS** i3A
661 RNS9.i1B **AS** i1B
662 RNS9.i2B **AS** i2B
663 RNS9.i3B **AS** i3B
664 RNS9.i1C **AS** i1C
665 RNS9.i2C **AS** i2C
666 RNS9.i3C **AS** i3C
667 RNS9.ins1 **AS** ins1
668 RNS9.ins2 **AS** ins2
669 RNS9.ins3 **AS** ins3
670 RNS9.ins4 **AS** ins4
671 RNS9.ins5 **AS** ins5
672 RNS9.ins6 **AS** ins6
673 RNS9.ins7 **AS** ins7
674 RNS9.ins8 **AS** ins8
675 RNS9.ins9 **AS** ins9
676 RNS9.v1A **AS** v1A
677 RNS9.v2A **AS** v2A
678 RNS9.v3A **AS** v3A
679 RNS9.v1B **AS** v1B
680 RNS9.v2B **AS** v2B
681 RNS9.v3B **AS** v3B
682 RNS9.v1C **AS** v1C
683 RNS9.v2C **AS** v2C
684 RNS9.v3C **AS** v3C
685 RNS9.io1A **AS** io1A
686 RNS9.io2A **AS** io2A
687 RNS9.io1B **AS** io1B
688 RNS9.io2B **AS** io2B
689 RNS9.io1C **AS** io1C
690 RNS9.io2C **AS** io2C
691 RNS9.v11p **AS** v11p
692 RNS9.v12p **AS** v12p
693 RNS9.v13p **AS** v13p
694 RNS9.v21p **AS** v21p
695 RNS9.v22p **AS** v22p
696 RNS9.v23p **AS** v23p
697 RNS9.O1 **AS** O1
698 RNS9.O2 **AS** O2
699 RNS9.O3 **AS** O3
700 RNS9.O4 **AS** O4
701 RNS9.O5 **AS** O5
702 RNS9.O6 **AS** O6
703 RNS9.O7 **AS** O7
704 RNS9.O8 **AS** O8
705 RNS9.O9 **AS** O9
706 RNS9.O10 **AS** O10
707 RNS9.O11 **AS** O11
708 RNS9.O12 **AS** O12
709 RNS9.O13 **AS** O13
710 RNS9.O14 **AS** O14
711 RNS9.O15 **AS** O15
712 RNS9.O16 **AS** O16
713 RNS9.O17 **AS** O17
714 RNS9.O18 **AS** O18
715 RNS9.O19 **AS** O19
716 RNS9.O20 **AS** O20
717 RNS9.O21 **AS** O21
718 RNS9.O22 **AS** O22
719 RNS9.O23 **AS** O23

```

720   RNS9.O24 AS O24
721 USE RNS9 AS RNS9
722 INPUT
723   v[1]:= M0001A
724   v[2]:= M0001B
725   v[3]:= M0001C
726 DATA
727   Aw:= 0.0158
728   Ay:= 0.0158
729   Ar:= 0.0158
730   N1:= 145.
731   N2:= 77.
732   N3:= 98.
733   Lw1:= 0.29
734   Lw2:= 0.154
735   Lw3:= 0.27
736   Ly:= 0.3608
737   Lr:= 0.8
738   Ld1:= 5.119E-5
739   Ld2:= 1.279E-4
740   Ld3:= 0.0108
741 OUTPUT
742   X0001A:=v3[1]
743   X0001B:=v3[2]
744   X0001C:=v3[3]
745 ENDUSE
746 ENDMODELS
747 C      1      2      3      4      5      6      7      8
748 C 34567890123456789012345678901234567890123456789012345678901234567890
749 /BRANCH
750 C < n1 >> n2 ><ref1><ref2>< R >< L >< C >
751 C < n1 >> n2 ><ref1><ref2>< R >< A >< B ><Leng><><>0
752   X0002A      1.      0
753   X0002B      1.      0
754   X0002C      1.      0
755 /SOURCE
756 C < n 1><>< Ampl. >< Freq. ><Phase/T0>< A1 >< T1 >< TSTART >< TSTOP >
757 14X0002A      2449.5      60.      -1.      100.
758 14X0002B      2449.5      60.     -120.     -1.      100.
759 14X0002C      2449.5      60.     -240.     -1.      100.
760 /OUTPUT
761 BLANK MODELS
762 BLANK BRANCH
763 BLANK SWITCH
764 BLANK SOURCE
765 BLANK OUTPUT
766 BLANK PLOT
767 BEGIN NEW DATA CASE
768 BLANK

```

Código Fonte A.11 – Código ATP para reator naturalmente saturado com 9 enrolamentos e considerando efeitos de histerese