

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

*Aplicação de técnicas de padrões de projeto orientados a
objeto na construção de framework para modelagem e
simulação dinâmica de geradores síncronos*

Ítalo Flexa Di Paolo

DM 13 / 2009

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2009

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Ítalo Flexa Di Paolo

*Aplicação de técnicas de padrões de projeto orientados a
objeto na construção de framework para modelagem e
simulação dinâmica de geradores síncronos*

DM 13 / 2009

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2009

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Ítalo Flexa Di Paolo

*Aplicação de técnicas de padrões de projeto orientados a
objeto na construção de framework para modelagem e
simulação dinâmica de geradores síncronos*

Dissertação submetida à Banca Examinadora do
Programa de Pós-Graduação em Engenharia Elétrica
(PPGEE) da Universidade Federal do Pará (UFPA) para
obtenção do Grau de Mestre em Engenharia Elétrica.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2009

CRÉDITOS DAS FOTOGRAFIAS

<i>Autor</i>	<i>Referência no Texto</i>
Fabrcio Gonzalez Nogueira	1, 2, 3 (2008), 6 (2009)
Ítalo Flexa Di Paolo	4, 5 (2009)
Thiago Damasceno Cordeiro	7 (2009)

D596a Di Paolo, Ítalo Flexa

Aplicação de técnicas de padrões de projeto orientados a objeto na construção de framework para modelagem e simulação dinâmica de geradores síncronos / Ítalo Flexa Di Paolo; orientador, Walter Barra Junior; 2009.

Dissertação (Mestrado) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2009.

1. Sistemas de energia elétrica – simulação. 2. Programação orientada a objetos. 3. Usinas hidrelétricas – automação. I. Título.

Ítalo Flexa Di Paolo

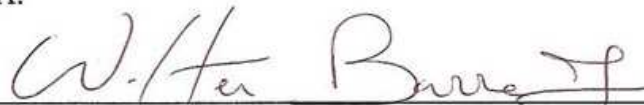
*Aplicação de técnicas de padrões de projeto orientados a
objeto na construção de framework para modelagem e
simulação dinâmica de geradores síncronos*

Dissertação de Mestrado submetida à avaliação da Banca Examinadora aprovada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Universidade Federal do Pará (UFPA) e julgada adequada para obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Sistemas de Energia Elétrica.

Aprovada em: 30 jun. 2009

BANCA EXAMINADORA:



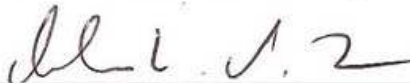
Prof. Dr. Walter Barra Junior (Orientador – PPGEE/UFPA)



Prof. Dr. José Augusto Lima Barreiros (Co-Orientador – PPGEE/UFPA)



Prof. Dr. Carlos Tavares da Costa Júnior (Membro – PPGEE/UFPA)



Prof. Dr. Orlando Fonseca Silva (Membro – FEE/UFPA)

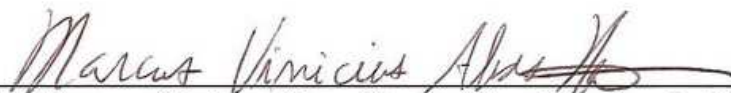


Prof. Dra. Carla Alessandra Lima Reis (Membro – PPGCC/UFPA)



Prof. Dr. André Maurício Damasceno Ferreira (Membro – DEPIN/UFPA)

VISTO:



Prof. Dr. Marcus Vinícius Alves Nunes (Coordenador do PPGEE/UFPA)

*Ao meu pai, Pasquale Di Paolo (in memorian)
que, certamente, estaria muito feliz
com mais esta vitória;*

*À minha mãe, Darcy e meus irmãos, Angela e Luciano,
pelo apoio familiar sempre imprescindível;*

*Ao meu primo, Ângelo Jorge Arthur Filho (in memorian),
que dedicou a maior parte de sua vida à UHE de Tucuruí
desde os tempos de sua construção,
partindo deste mundo em pleno exercício
de suas funções na Eletronorte.*

AGRADECIMENTOS

À minha mãe, Profa. MSc. Darcy Flexa Di Paolo, que sempre incentiva e investe em minha educação, oferecendo o que há de melhor na estrutura familiar: carinho;

Ao Prof. Dr. Walter Barra Junior, grande mestre e orientador deste trabalho, cujas dicas, empenho e disponibilidade foram essenciais para os resultados alcançados;

Ao Prof. Dr. José Augusto Lima Barreiros, pelo apoio incondicional em meus objetivos de entrar e concluir o Mestrado em Engenharia Elétrica, cujo exemplo de caráter e perseverança admiro desde os tempos de graduação;

Aos Profs. Drs. Orlando Fonseca Silva (Nick) e Carlos Tavares da Costa Júnior, pelo grande apoio e incentivo, prestados no decorrer desta caminhada;

Aos vários pesquisadores que passaram pelo PPGEE, deixando seus legados, e que contribuíram significativamente para o sucesso deste trabalho, entre eles, MSc. Adolfo Sena, Dr. André Ferreira, MSc. Barbara Medeiros, MSc. Jorge Risuenho, entre tantos outros que também integraram e integram o Grupo de Sistemas de Controle;

Ao consultor externo Eng. Venilton Oliveira, que com sua vasta experiência contribuiu em treinamentos e informações sobre o Sistema Interligado Nacional;

Aos colegas e companheiros de dias, noites e madrugadas Engs. Thiago Damasceno Cordeiro (de coração!), MSc. Fabrício Gonzalez Nogueira, MSc. Conceição Fonseca e Marcus Ciro Gomes, que com trabalho e amizade contribuíram significativamente na produção destes resultados;

Aos engenheiros e técnicos da Eletronorte Davi Moreira, Jânio De Lana e Ricardo Campos, simpáticos e sempre dispostos a contribuir neste projeto;

Aos colegas da Exodus Eder Rodrigues, Fábio Soares, Harry Cruz, Thiago Moutinho e Heverton Kikuchi, que participaram direta ou indiretamente desta caminhada, contribuindo com apoio e incentivo;

À minha namorada, Andréa Lia Amazonas, com quem muito tenho aprendido e que me ofereceu carinho e compreensão no decorrer desta caminhada;

Aos muitos amigos conquistados tanto nos laboratórios do Grupo de Sistemas de Controle quanto durante as disciplinas, grandes companheiros que muito contribuíram para que eu chegasse até aqui;

Aos professores e funcionários do PPGEE e de todo o Instituto de Tecnologia da UFPA, sempre presentes no incentivo aos alunos da pós-graduação;

Às Centrais Elétricas do Norte do Brasil (Eletronorte), pelo suporte recebido por esta pesquisa através do projeto de Pesquisa e Desenvolvimento (P&D) 63819, e

Aos funcionários da UHE de Tucuruí, que trabalham incansavelmente em problemas complexos que envolvem o fornecimento de energia elétrica para todo o Brasil.

A todos, meu muito obrigado!

Em seu encontro com a Natureza, a ciência invariavelmente provoca um sentimento de reverência e admiração. O próprio ato de compreender é uma celebração da união, da incorporação, ainda que numa escala muito modesta, à magnificência do Cosmos. E ao longo do tempo o desenvolvimento cumulativo do conhecimento em todo o mundo converte a ciência em algo que é quase uma metainteligência, capaz de ultrapassar as fronteiras das nações e das gerações.

Carl Sagan (1934-1997)

RESUMO

Este trabalho apresenta a concepção, desenvolvimento e testes de um *software* avançado, utilizando o paradigma da orientação a objetos. A ferramenta é destinada à execução de estudos dinâmicos avançados em sistemas elétricos de potência. A estratégia proposta baseia-se em padrões de projeto utilizando conceitos fundamentados em boas práticas de engenharia. Primeiramente, executa-se um mapeamento entre uma representação dinâmica no domínio da frequência (ou do tempo) (representação em diagramas de bloco) para um domínio orientado a objetos. Esta análise permite uma abordagem mais flexível para a descrição do sistema dinâmico, acelerando o processo de desenvolvimento do *software*. Em seguida, para explorar ainda mais a abordagem orientada a objetos, uma combinação dos padrões *Factory Method*, *Strategy*, *Builder* e *Facade* foi utilizada, de forma complementar, de modo a fazer face à complexidade observada em sistemas elétricos de potência, permitindo o desenvolvimento do *software*. Desta forma, possibilitando a simulação da dinâmica de sistemas elétricos de grande porte, incluindo modelos complexos de sistemas de controle adaptativo e digital, que são limitações de ferramentas similares atuais. Este *software*, denominado *DynaPower* foi testado para realizar estudos de simulação dinâmica relacionados com fenômenos de oscilações eletromecânicas na Usina Hidroelétrica de Tucuruí. Foram modeladas todas as 23 unidades geradoras presentes na usina, adotando-se uma representação pormenorizada das turbinas hidráulicas, bem como das máquinas síncronas e sistemas de controle associados, incluindo os reguladores automáticos de tensão, os reguladores de velocidade e os estabilizadores de sistemas de potência. Os resultados de simulações foram validados por comparação com dados medidos em testes de campo realizados na própria UHE de Tucuruí e através de resultados de outros simuladores comerciais. Os resultados mostraram bom desempenho do *software* simulador desenvolvido, o qual foi capaz de se reproduzir, em simulação dinâmica, muitos dos fenômenos observados em ensaios de campo na UHE de Tucuruí.

Palavras-chave: desenvolvimento orientado a objetos, padrões de projeto, dinâmica de sistemas elétricos de potência, usina hidroelétrica.

ABSTRACT

This work presents the design, development and tests of an advanced software tool, using the paradigm of object orientation. The tool is intended for advanced implementation of dynamic studies in electric power systems. The proposed strategy is based on design patterns, using sound software engineering concepts. Firstly, a mapping between time-frequency (block diagram) to an object oriented domain is performed. This analysis allows a more flexible system description and speeds up the overall software development process. In order, to exploit the object oriented approach, a mix of design pattern strategies as Factory Method, Strategy, Builder and Facade was used on a complementary basis, to cope with the complexity observed in electric power systems, allowing software development. This allows the simulation of large power system, including complex models for adaptive and digital controls, which are, nowadays, known limitations of existing commercial software tools for power engineering studies. This software, called *DynaPower*, was tested for dynamic simulation studies of phenomena related to the electromechanical oscillations of Tucuruí Hydroelectric Power Plant. Were modeled all the 23 power generating hydroelectric units present at hydropower plant, with a very detailed turbine representation as well as synchronous machines and related control systems representations, including automatic voltage regulators, speed governors and power system stabilizers. The simulation results were validated against data measured on field tests performed at Tucuruí Power Plant and using results from other commercial simulators. The results shown a very good performance of the developed simulator, which was able to reproduce in simulation many of dynamic phenomena observed on field tests of the HPP Tucuruí.

Key-Words: object-oriented development, design patterns, power system dynamics, hydropower plant.

LISTA DE DIAGRAMAS

Diagrama 1: Classificação de Estabilidade em Sistemas Elétricos de Potência.	27
Diagrama 2: Modelo π , usado para modelar Linhas de Transmissão e Transformadores.....	33
Diagrama 3: Regulador Automático de Tensão básico.	34
Diagrama 4: Estabilizador de Sistema de Potência básico.....	35
Diagrama 5: Regulador de Velocidade básico.	35
Diagrama 6: Turbina hidráulica linear.	36
Diagrama 7: Turbina hidráulica não-linear.....	36
Diagrama 8: Turbina hidráulica não-linear.....	37
Diagrama 9: Relacionamento entre os Diagramas da UML.....	42
Diagrama 10: Diagrama de Classes para implementação do Padrão de Projeto <i>Factory Method</i>	44
Diagrama 11: Arquitetura do Padrão de Projeto <i>Factory Method</i> para mapeamento entre diagrama de blocos e orientação a objetos.	46
Diagrama 12: Detalhes das classes principais que armazenam a estrutura do diagrama de blocos para executar as simulações.....	47
Diagrama 13: Detalhes das classes que mapeiam o diagrama de blocos para sistemas de potência.....	47
Diagrama 14: Diagrama de Classes para implementação do Padrão de Projeto <i>Strategy</i>	48
Diagrama 15: Arquitetura do Padrão de Projeto <i>Strategy</i> definindo a dinâmica específica de cada dispositivo do sistema	49
Diagrama 16: Diagrama de Classes para implementação do Padrão de Projeto <i>Builder</i>	50
Diagrama 17: Arquitetura do Projeto <i>Builder</i> na implementação dos métodos numéricos para simulação de sistemas de potência.....	52
Diagrama 18: Detalhes das classes que implementam os métodos numéricos para simulação.	52
Diagrama 19: Diagrama de objetos representativo para o sistema elétrico de potência.	53
Diagrama 20: Diagrama de Classes enfocando o problema da falta do Padrão <i>Facade</i>	54
Diagrama 21: Diagrama de Classes para implementação do Padrão de Projeto <i>Facade</i>	55
Diagrama 22: Diagrama de Atividades para Simulação de Sistemas Elétricos de Potência.....	56
Diagrama 23: Diagrama de Classes para implementação do padrão de projeto <i>Facade</i>	56
Diagrama 24: Diagrama de Blocos do RAT da primeira etapa.....	65
Diagrama 25: Malha principal do RAT da segunda etapa.....	66
Diagrama 26: Compensação de corrente reativa.	66

Diagrama 27: Limitador de corrente de armadura	66
Diagrama 28: Limitador de subexcitação.....	67
Diagrama 29: Limitador de sobreexcitação.	67
Diagrama 30: Limitador de corrente de excitação.	67
Diagrama 31: Limitador de enlace de fluxo (Volt/Hz).	67
Diagrama 32: Diagrama de classes dos RATs da UHE de Tucuruí.....	69
Diagrama 33: Diagrama de Blocos do ESP da primeira etapa.....	70
Diagrama 34: Diagrama de Blocos do ESP da segunda etapa.	71
Diagrama 35: Diagrama de classes dos ESPs da UHE de Tucuruí.....	73
Diagrama 36: Diagrama de classes das Turbinas da UHE de Tucuruí.	76
Diagrama 37: Diagrama de Blocos do RV da primeira etapa.	77
Diagrama 38: Diagrama de Blocos do RV da segunda etapa.	78
Diagrama 39: Diagrama de classes dos RVs da UHE de Tucuruí.	80
Diagrama 40: Diagrama de classes das UGHs da UHE de Tucuruí.....	81

LISTA DE GRÁFICOS

Gráfico 1: Curto-circuito trifásico aplicado na máquina 13.....	85
Gráfico 2: Degrau de +10% aplicado referência do RAT da máquina 13.....	85
Gráfico 3: Degrau de +10% aplicado referência do RV da máquina 13.....	86
Gráfico 4: Degrau de +10% aplicado na referência do RV de uma máquina da primeira etapa, obtido pelo ONS.....	87
Gráfico 5: Degrau de +10% aplicado na referência do RV de uma máquina da primeira etapa, utilizando <i>DynaPower</i> e Matlab.....	88
Gráfico 6: Degrau de -9,09% aplicado na referência do RV de uma máquina da primeira etapa, obtido pelo ONS.....	88
Gráfico 7: Degrau de -9,09% aplicado na referência do RV de uma máquina da primeira etapa, utilizando <i>DynaPower</i> e Matlab.....	89
Gráfico 8: Degrau de +2,9% aplicado na referência do RV da UGH 8, medido em campo e comparado com simulações utilizando <i>DynaPower</i> e Matlab.....	90
Gráfico 9: Degrau de -25% aplicado na referência do RV de uma máquina da segunda etapa, obtido pelo ONS.....	90
Gráfico 10: Degrau de -25% aplicado na referência do RV de uma máquina da segunda etapa, utilizando <i>DynaPower</i> e Matlab.....	91
Gráfico 11: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, obtido pelo ONS.....	91
Gráfico 12: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, utilizando o Matlab.....	92
Gráfico 13: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, utilizando o <i>DynaPower</i>	92
Gráfico 14: Degrau de +4% aplicado na referência do RAT da UGH 1.....	93
Gráfico 15: Degrau de +4% aplicado na referência do RAT da UGH 1, utilizando o Matlab.....	93
Gráfico 16: Degrau de +4% aplicado na referência do RAT da UGH 1, utilizando o <i>DynaPower</i>	94

LISTA DE ILUSTRAÇÕES

Fotografia 1: Visão interna da barragem da UHE de Tucuruí.	21
Fotografia 2: Máquinas da primeira etapa da construção da UHE de Tucuruí.....	62
Fotografia 3: Máquinas da segunda etapa da construção da UHE de Tucuruí.....	62
Fotografia 4: Rotor da UGH 8.	64
Fotografia 5: Ensaio sendo realizado no RAT da UGH 8.	65
Fotografia 6: Ensaio sendo realizado no ESP da UGH 8.	70
Desenho 1: Turbina Francis, similar às instaladas na UHE de Tucuruí.....	74
Fotografia 7: Ensaio sendo realizado no RV da UGH 8.....	77
Quadro 1: Síntese da quantidade de estados/saídas calculados para o cenário da UHE de Tucuruí	81
Quadro 2: Resultado do fluxo de carga para uma condição de operação no cenário construído para representar a UHE de Tucuruí.	84

LISTA DE ABREVIATURAS E SIGLAS

BCSSD	Biblioteca de Classes para Simulação de Sistemas Dinâmicos
CA	Corrente Alternada
CASE	Computer-Aided Software Engineering
CC	Corrente Contínua
Eletronorte	Centrais Elétricas do Norte do Brasil S/A
ESP	Estabilizador de Sistema de Potência
FEE	Faculdade de Engenharia Elétrica
GoF	Gang of Four
MFC	Microsoft Foundation Class Library
ONS	Operador Nacional do Sistema Elétrico
OO	Orientação a Objetos ou Orientado(a) a Objetos
P&D	Pesquisa e Desenvolvimento
p.u.	Por Unidade
PC	Personal Computer
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
PSP	Personal Software Process
RAT	Regulador Automático de Tensão
RUP	Rational Unified Process
RV	Regulador de Velocidade
SEP	Sistema Elétrico de Potência
SIN	Sistema Interligado Nacional
UFPA	Universidade Federal do Pará
UGH	Unidade Geradora Hidrelétrica
UHE	Usina Hidrelétrica
UML	Unified Modeling Language
XP	Extreme Programming

SUMÁRIO

1	INTRODUÇÃO	18
1.1	JUSTIFICATIVA.....	18
1.2	OBJETIVOS.....	22
1.3	METODOLOGIA	23
1.4	ORGANIZAÇÃO DO TEXTO	19
2	MODELAGEM DE SISTEMAS ELÉTRICOS DE POTÊNCIA	26
2.1	VISÃO GERAL.....	26
2.2	BARRAS.....	29
2.3	CARGAS.....	30
2.4	MÁQUINAS SÍNCRONAS.....	30
2.5	SISTEMAS DE TRANSMISSÃO.....	33
2.6	REGULADOR AUTOMÁTICO DE TENSÃO.....	33
2.7	ESTABILIZADOR DE SISTEMA DE POTÊNCIA.....	34
2.8	REGULADOR DE VELOCIDADE	35
2.9	TURBINAS HIDRÁULICAS.....	35
3	PADRÕES DE PROJETO ORIENTADOS A OBJETO	38
3.1	PROCESSO EM PROJETOS ORIENTADOS A OBJETO.....	38
3.2	CONCEITOS FUNDAMENTAIS DA ORIENTAÇÃO A OBJETOS	39
3.3	PADRÕES DE PROJETO.....	42
	3.3.1 Factory Method	44
	3.3.2 Strategy	48
	3.3.3 Builder	50
	3.3.4 Facade	54
4	DESENVOLVIMENTO DO FRAMEWORK PARA SIMULAÇÃO DINÂMICA COM MÚLTIPLAS UNIDADES GERADORAS NA UHE DE TUCURUÍ	58
4.1	CONSIDERAÇÕES SOBRE FRAMEWORKS PARA SIMULAÇÃO DINÂMICA DE SISTEMAS ELÉTRICOS DE POTÊNCIA.....	58
4.2	MODELAGEM DA USINA HIDRELÉTRICA DE TUCURUÍ.....	61
4.3	GERADORES.....	64
4.4	REGULADORES AUTOMÁTICOS DE TENSÃO.....	64
	4.4.1 Modelagem em Diagrama de Blocos	65

4.4.2 Modelagem em Espaços de Estados	67
4.4.3 Modelagem Orientada a Objetos	69
4.5 ESTABILIZADORES DE SISTEMAS DE POTÊNCIA.....	70
4.5.1 Modelagem em Diagrama de Blocos	70
4.5.2 Modelagem em Espaços de Estados	71
4.5.3 Modelagem Orientada a Objetos	73
4.6 TURBINAS.....	73
4.6.1 Modelagem em Diagrama de Blocos	74
4.6.2 Modelagem em Espaços de Estados	75
4.6.3 Modelagem Orientada a Objetos	76
4.7 REGULADORES DE VELOCIDADE	76
4.7.1 Modelagem em Diagrama de Blocos	77
4.7.2 Modelagem em Espaços de Estados	78
4.7.3 Modelagem Orientada a Objetos	79
4.8 CONSOLIDAÇÃO DA MODELAGEM ORIENTADA A OBJETOS	80
5 RESULTADOS E COMPARAÇÕES DE SIMULAÇÕES DINÂMICAS EM AMBIENTE COMPUTACIONAL E COM MEDIÇÕES EM CAMPO	82
5.1 FLUXO DE CARGA E FALTAS IMPLEMENTADAS.....	83
5.2 COMPARAÇÃO COM RESULTADOS DE TESTES EXPERIMENTAIS EM CAMPO E COM OS SIMULADORES MATLAB E ANATEM	87
6 CONCLUSÃO	95
REFERÊNCIAS	100
APÊNDICES	106
APÊNDICE A Diagrama unifilar das UGHs de Tucuruí interligadas ao Barramento Infinito.....	107
APÊNDICE B Desenvolvimento dos cálculos para obtenção das equações de estado para o RAT da segunda etapa.	108
APÊNDICE C Desenvolvimento dos cálculos para obtenção das equações de estado para o ESP da primeira etapa.	110
APÊNDICE D Desenvolvimento dos cálculos para obtenção das equações de estado para o ESP da segunda etapa.	112
APÊNDICE E Desenvolvimento dos cálculos para obtenção das equações de estado para o RV da primeira etapa.	115
ANEXOS	117

ANEXO A Parâmetros do modelo dos geradores da primeira etapa.....	118
ANEXO B Parâmetros do modelo dos geradores da segunda etapa.....	119
ANEXO C Parâmetros do modelo do RAT da primeira etapa	120
ANEXO D Parâmetros do modelo do RAT da segunda etapa	121
ANEXO E Parâmetros do modelo do ESP da primeira etapa	122
ANEXO F Parâmetros do modelo do ESP da segunda etapa.....	123
ANEXO G Parâmetros do modelo da Turbina da primeira etapa.....	124
ANEXO H Parâmetros do modelo da Turbina da segunda etapa	125
ANEXO I Parâmetros do modelo do RV da primeira etapa.....	126
ANEXO J Parâmetros do modelo do RV da segunda etapa	127

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

A atividade de simular Sistemas Elétricos de Potência (SEPs) é essencial para que o engenheiro possa avaliar e planejar atividades de expansão, de operação e o comportamento do sistema elétrico diante de eventos como perdas de linhas de transmissão, variações de carga, curtos-circuitos, dentre tantos outros.

Mais especificamente, os estudos de estabilidade têm o objetivo de garantir a manutenção do sincronismo entre as máquinas geradoras de todo o sistema, tanto em regime permanente quanto durante a ocorrência de distúrbios (ARRILLAGA; WATSON, 2001). Estes distúrbios podem ser provocados por diversos eventos, como as variações de carga que ocorrem a todo momento, desligamento de linhas de transmissão provocados por descargas atmosféricas, e curtos-circuitos de diversos tipos. Seja qual for o evento que provoque impacto e comprometa a estabilidade das unidades geradoras, o sistema necessita de controles para regular as variáveis principais e auxiliares na manutenção da estabilidade da planta, em tempo hábil para evitar colapsos e manter a qualidade da geração.

Isto porque os atuais sistemas de potência são predominantemente baseados em máquinas síncronas para a geração de potência elétrica, sendo uma condição necessária para sua operação satisfatória que todas as máquinas permaneçam em sincronismo, ou seja, operando com uma mesma frequência elétrica. Em Kundur et al. (2004) e Kundur (1994) apresenta-se que este tipo de estabilidade é regido predominantemente pela dinâmica dos ângulos dos rotores dos geradores síncronos, definida como estabilidade angular, embora um decaimento acentuado das magnitudes das tensões, ou colapso de tensão, em determinadas barras do sistema possa ser provocado pelo comportamento dinâmico das cargas do sistema.

A estabilidade de SEPs, de acordo com Kundur (1994) e Barcelos (2007), tem sido abordada segundo dois aspectos:

- Pequenas perturbações: pequenas variações de carga ou geração, que podem ocorrer continuamente e o sistema deve ser capaz de ajustar-se, cujos resultados são sentidos em menos de 1 segundo, e

- Grandes perturbações: faltas em linhas de transmissão, perda de unidades geradoras ou de cargas de grande capacidade, cujos resultados são sentidos em cerca de 10 segundos ou mais, chegando até em alguns minutos.

Em um ambiente virtual podem-se construir modelos físicos que representem um sistema real com a diversidade de detalhes demandados pelo estudo a ser realizado no sistema.

A primeira geração de ferramentas computacionais para sistemas de energia elétrica constituíam-se de programas aplicativos direcionados para um número limitado de funções (em sua grande maioria escritas em linguagem FORTRAN). Muito pouca ou nenhuma importância era dada à reutilização do código para futuros aplicativos que se fizessem necessários. Assim, cada vez que uma nova ferramenta era necessária um programa totalmente novo era implementado. Contrastando com esta realidade, o setor elétrico sempre exigiu ferramentas com características bastante dinâmicas, onde o surgimento de novos equipamentos (e por consequência novos modelos computacionais), o aperfeiçoamento dos modelos já existentes e o surgimento de novas metodologias de análise possam ser rapidamente assimiladas pelos programas aplicativos. Sendo assim, os softwares para a área de energia elétrica devem ser ágeis e extremamente flexíveis face a mudanças. Um projeto que não levar em consideração a possibilidade de mudanças ao longo de sua vida está sujeito ao risco de uma grande reformulação no futuro (MANZONI, 2005, p. 20).

Ainda segundo Manzoni (2005), atualmente tem surgido duas classes de ferramentas para agilizar a produção de *software* para a simulação de SEPs, que são os *frameworks* e os *toolkits*. Os *frameworks* são conjuntos de elementos que implementam mecanismos que são essenciais para um domínio específico de problemas (HORSTMANN, 2007, p. 312), principalmente direcionadas ao desenvolvimento de aplicativos, como o *.NET Framework* (MICROSOFT..., 2007). Já *toolkits* referem-se a aplicativos com interfaces gráficas ou especificamente definidas para usuários finais, como os *toolboxes* utilizados pelo ambiente Matlab e Simulink (MATLAB, 2008).

É neste contexto que surge o projeto de Pesquisa e Desenvolvimento (P&D) número 63819, financiado com recursos do fundo setorial para o setor elétrico brasileiro, intitulado: “Desenvolvimento de um Sistema Computacional para Análise dos Efeitos da Expansão e Modernização da UHE de Tucuruí”. Projeto este desenvolvido em parceria entre a Universidade Federal do Pará (UFPA), mais especificamente pelo grupo de sistemas de controle, ligado à Faculdade de Engenharia Elétrica (FEE) e ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), e as Centrais Elétricas do Norte do Brasil S/A (Eletronorte), tendo como objetivos:

- Realizar estudos de controle e estabilidade no âmbito da Usina Hidrelétrica (UHE) de Tucuruí, visando identificar modos de oscilação que necessitem de controle amortecedor;
- Modelar detalhadamente a UHE e sua rede elétrica periférica, considerando o restante do Sistema Interligado Nacional (SIN) através de equivalentes dinâmicos;
- Projetar os equivalentes dinâmicos e controles adaptativos para estudos diante de diversas contingências e avaliar seus impactos, visando maior eficiência na geração de energia;
- Desenvolver de um sistema computacional, com uso da engenharia de *software*, para simulações da dinâmica de sistemas de energia elétrica de grande porte para estudos de controle e estabilidade.

Para o alcance de tais objetivos, tendo presente as limitações das ferramentas existentes para simulação dinâmica em ambiente computacional, verificou-se a necessidade do desenvolvimento de um *software* específico que pudesse simular a dinâmica detalhada da UHE de Tucuruí, e ainda implementar equivalentes dinâmicos e diversas estratégias de controle para estudos de estabilidade, como controle digital ou adaptativo. Daí a iniciativa de desenvolver o sistema computacional *DynaPower*, no sentido de suprir as diversas limitações observadas.

A ferramenta utilizada como referência pelo Operador Nacional do Sistema Elétrico¹ (ONS) para estudos equivalentes, por exemplo, é o ANATEM... (2007) que, embora possua inúmeros recursos, não permite a implementação de técnicas de inteligência computacional ou controle digital; e o Matlab (2008), embora também possua muitos recursos e tenha a interface gráfica com o usuário, no ambiente Simulink, como grande diferencial, possui limitações quando utilizado com SEPs de grande porte, impossibilitando seu uso para esta finalidade.

A Fotografia 1 apresenta uma visão da barragem pelo interior da UHE de Tucuruí, em uma das visitas realizadas para ensaios experimentais em junho de 2008.

¹ “Entidade de direito privado, sem fins lucrativos [...] responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no [Brasil...]” (OPERADOR NACIONAL DO SISTEMA ELÉTRICO, 2008b).



Fotografia 1: Visão interna da barragem da UHE de Tucuruí.

Embora já exista um modelo do SIN² em uso e atualizado pelo ONS implementado no ANATEM... (2007), a área de cobertura da Eletronorte, neste modelo, é modelada como um agrupamento de geradores que representam diversas máquinas da usina, além de outras simplificações. Esta configuração é interessante para os estudos do ONS, mas apresenta limitações, por exemplo, ao desprezar uma série de fenômenos dinâmicos importantes, dentre os quais a dinâmica associada aos modos eletromecânicos intra-planta. Ou seja, não supre necessidades importantes da própria Eletronorte para estudos e planejamento a contento de sua operação, o que motivou a definição da modelagem mais detalhada e atualizada da usina, favorecendo estudos específicos da Eletronorte.

Outra limitação do ANATEM... (2007) é a dificuldade de implementação de modelos de controladores digitais e adaptativos, a não ser através de seus equivalentes contínuos oriundos de funções e recursos pré-definidos, que nem sempre representam, com fidelidade, os efeitos que se deseja considerar.

Portanto, justifica-se a iniciativa de desenvolvimento do *DynaPower* que foi desenvolvido utilizando-se do *framework* denominado Biblioteca de Classes para Simulação de Sistemas Dinâmicos (BCSSD), de forma a simular, em ambiente computacional, a dinâmica das Unidades Geradoras Hidrelétricas (UGHs) da UHE de

² A base de dados mais atual com o modelo do SIN utilizado pelo ONS está disponível em Operador Nacional do Sistema Elétrico (2008a).

Tucuruí, servindo de ferramenta essencial para os estudos posteriores de controle e estabilidade.

A opção pelo uso da BCSSD foi determinada por ser um produto de estudos do grupo de sistemas da UFPA, sendo apresentada e validada diversas vezes, como em Sena (1998, 2006) e Sena et al. (2005, 2006), fazendo parte do *know-how* conquistado pelo referido grupo.

1.2 OBJETIVOS

Os **objetivos centrais** deste trabalho foram: desenvolver um *software*, como simulador dinâmico, denominado *DynaPower*, utilizando padrões de projeto orientados a objeto e a BCSSD, e modelar detalhadamente as Unidades Geradoras Hidrelétricas (UGHs) da UHE de Tucuruí.

Os **objetivos específicos** foram:

- Definir os padrões de projetos orientados a objeto aplicados na modelagem das unidades geradoras;
- Adaptar a BCSSD, tanto para implementar os padrões de projeto definidos como para modernizar sua plataforma de desenvolvimento;
- Modelar os geradores da primeira e segunda etapa da UHE de Tucuruí, baseados em relatórios técnicos cedidos pela Eletronorte e em padrões de projeto definidos;
- Modelar os Reguladores Automáticos de Tensão (RATs), Estabilizadores de Sistemas de Potência (ESPs), Reguladores de Velocidade (RVs) e as turbinas, baseando-se em relatórios técnicos cedidos pela Eletronorte e nos modelos do SIN implementados no ANATEM... (2007);
- Implementar os modelos isoladamente e em conjunto utilizando o Matlab (2008) para fins de comparação e validação de resultados, e
- Comparar e validar os resultados das simulações dinâmicas geradas pelo *DynaPower* com as geradas com uso do ANATEM... (2007) e Matlab (2008), e com resultados de ensaios realizados em campo.

1.3 METODOLOGIA

Existe uma diversidade de paradigmas de desenvolvimento de aplicações computacionais. O paradigma utilizado, por exemplo, pela primeira geração de ferramentas de energia elétrica citada por Manzoni (2005) foi o procedimental, porém não é um fato determinante que a escolha do paradigma seja responsável pelas desvantagens como pouca reutilização de código, baixo acoplamento entre os módulos, ou difícil manutenção do código-fonte. Seja qual for o paradigma escolhido, técnicas específicas de engenharia de *software* devem ser utilizadas para garantir a qualidade do código-fonte produzido e do produto final a ser desenvolvido.

Portanto, as aplicações originadas deste período não apresentam as desvantagens citadas devido ao paradigma utilizado, mas sim pela falta de aplicação de técnicas específicas para garantir a qualidade e continuidade do *software*. Obviamente que, dependendo do paradigma utilizado, técnicas diferentes da engenharia de *software* devem ser aplicadas e alguns paradigmas se destacam em relação a outros devido a popularidade de suas ferramentas, linguagens de programação e material de apoio disponível. O paradigma da Orientação a Objetos (OO) é hoje o mais popular principalmente por dispor de diversas ferramentas e facilitar ao desenvolvedor a aplicação das técnicas de engenharia de *software* devido ao vasto material encontrado hoje na literatura e na Internet, como se pode observar em Battisti (2001), Paula Filho (2003) e Sommerville (2003).

Atualmente, o paradigma da orientação a objetos está entre os mais populares para o desenvolvimento de *software*, e já há uma diversidade de trabalhos com a sua aplicação na simulação de SEPs, tais como em Zhou (1996), Zhu (1999), Palma et al. (2003), Dzafic, Glavic e Tesnjak (2004), Machado Júnior (2005), Manzoni (2005), Sena (1998, 2006) e Sena et al. (2005, 2006).

Um *framework* orientado a objeto é, portanto, um conjunto de classes cooperativas que implementam mecanismos que são essenciais para um domínio específico de problemas. Um programador pode criar funcionalidades novas no domínio do problema que está trabalhando estendendo as classes do *framework*. Já um *framework* de aplicação é um conjunto de classes que implementa serviços comuns a certo tipo de aplicação (HORSTMANN, 2007, p. 312).

No caso do desenvolvimento do *DynaPower*, o *framework* utilizado foi a

BCSSD (SENA, 1998, 2006; SENA et al., 2005, 2006), e o *framework* aplicativo foi a implementação utilizando estratégias de padrões de projeto orientados a objeto, mais especificamente os padrões da *Gang of Four* (GoF), apresentadas em Freeman e Freeman, 2007 e Gamma et al., 2007. Esta estratégia é uma técnica da engenharia de *software* para garantir a reusabilidade do código-fonte e definir ao engenheiro responsável pela modelagem as diretrizes necessárias para modelar um determinado componente.

A BCSSD foi construída em C++ e, portanto, o *DynaPower*, tratando-se de uma continuidade deste legado iniciado em 1996 pelo grupo de sistemas de controle, ligado à FEE e ao PPGEE da UFPA, também foi desenvolvido nesta linguagem. Porém, a plataforma de desenvolvimento teve de ser atualizada devido à necessidade de melhorias em alguns requisitos, tais como o tempo de processamento da simulação e a facilidade em recursos de *debug* da aplicação, e também para suportar bibliotecas de execução de cálculos numéricos e matriciais. Sendo assim, a aplicação do *DynaPower* teve de ser migrada do Builder C++ 5 (BORLAND..., 2002) para o Visual Studio (MICROSOFT..., 2008). O banco de dados relacional utilizado foi PostgreSQL (2005) e mais recentemente o SQL Server (MICROSOFT..., 2005).

As fontes para a construção dos modelos das unidades geradoras foram os relatórios técnicos cedidos pela Eletronorte: General Electric ([19--]), Eletronorte (1988, 2001), Eletronorte e Alstom Power ([2000]), Operador Nacional do Sistema Elétrico (2004, 2007, 2003a, 2003b, 2006a, 2006b) e a base de dados atual do SIN, atualizada pelo ONS (OPERADOR NACIONAL DO SISTEMA ELÉTRICO, 2008a). Mesmo com toda esta documentação apresentada e estudada, em alguns momentos, durante o processo de desenvolvimento, foi necessário re-projetar alguns parâmetros devido inconsistências da dinâmica esperada com a observada. Assim, os modelos e parâmetros apresentados neste trabalho já são resultados de uma pesquisa detalhada na documentação e em estudos da dinâmica de cada dispositivo.

A validação de cada modelo implementado no *DynaPower*, foi feita com uso do Matlab e Simulink (MATLAB, 2008), utilizando técnicas apresentadas por Ferreira (2005) e adaptadas pelo grupo de sistemas de controle da UFPA; pelo ANATEM... (2007), utilizando a base de dados do SIN; comparando com resultados de ensaios também apresentados em relatórios técnicos cedidos pela Eletronorte, e também comparando com medições realizadas em campo.

Nota-se, portanto, uma ênfase metodológica interdisciplinar, abordando estudos de modelagem, controle e dinâmica de SEPs, controle e estabilidade de dispositivos dinâmicos e técnicas de engenharia de *software* na modelagem e implementação da simulação dinâmica em ambiente computacional.

1.4 ORGANIZAÇÃO DO TEXTO

O trabalho foi organizado em seis capítulos iniciando com a exposição dos motivos que impulsionaram o desenvolvimento do trabalho e sua devida base conceitual, passando pela definição das ferramentas trabalhadas e desenvolvidas e por fim, apresentando os principais resultados obtidos.

O primeiro capítulo apresenta os motivos que provocaram o desenvolvimento do trabalho, bem como os objetivos e metodologia utilizada.

O segundo capítulo apresenta o processo de modelagem de SEPs, enfocando as unidades geradoras, transformadores, linhas de transmissão e cargas.

O terceiro capítulo ressalta a técnica dos padrões de projeto orientados a objeto para uso geral que foram utilizadas para modelar as unidades geradoras da UHE de Tucuruí.

O quarto capítulo discorre sobre a modelagem final das unidades geradoras em diagrama de blocos e orientada a objeto, com suas devidas considerações.

O quinto capítulo apresenta os resultados de simulações realizadas pelo *DynaPower* e a comparação e validação com os resultados obtidos com uso de outras ferramentas já consagradas e também com medições realizadas na própria UHE.

E o sexto capítulo destaca as conclusões principais do trabalho, retomando os objetivos propostos e confrontando-os com os resultados alcançados, apontando ainda as devidas considerações e sugestões para trabalhos futuros.

2 MODELAGEM DE SISTEMAS ELÉTRICOS DE POTÊNCIA

2.1 VISÃO GERAL

Diante de quaisquer distúrbios que venham a ser provocados, é desejável que um SEP se mantenha estável, garantindo a qualidade do fornecimento de energia elétrica e também sua integridade.

Estabilidade é a condição de equilíbrio entre forças opostas. O mecanismo através do qual as máquinas síncronas interconectadas mantêm o sincronismo umas com as outras é através de forças restauradoras, as quais devem agir sempre que houver aceleração ou desaceleração de uma ou mais máquinas, em relação às demais máquinas do sistema. Em condições de regime permanente, há equilíbrio entre torque mecânico e torque elétrico para cada máquina, possibilitando manter a velocidade constante. Se o sistema é perturbado, este equilíbrio é afetado, resultando em aceleração ou desaceleração dos rotores das máquinas, de acordo com as leis de movimento de corpos rotativos. Se um gerador, temporariamente, estiver mais rápido que outro, a posição angular relativa da máquina será avançada. O resultado da diferença angular é o de transferir parte da carga da máquina lenta para a máquina rápida, dependendo do relacionamento entre os ângulos elétricos. Isto tende a reduzir a diferença de velocidade e daí a separação angular. O relacionamento entre os ângulos elétricos [...] é fortemente não-linear. Além de certo limite, o aumento da separação angular leva à instabilidade. [...]. (KUNDUR, 1994, p. 22, tradução nossa).

Os modelos utilizados em estudos dinâmicos de SEPs podem ser mais ou menos complexos de acordo com os fenômenos que se deseja estudar. Isto porque para o estudo de determinados fenômenos, simplificações podem ser feitas sem que afetem os resultados auferidos, devido a pouca influência de determinados fenômenos na faixa dinâmica de estudo. Esta simplificação está também diretamente relacionada ao esforço computacional na realização dos cálculos das simulações.

Neste sentido, Kundur (1994) classifica a estabilidade de sistemas de potência como mostra o Diagrama 1, o qual também é destacado o foco de estabilidade que este trabalho se propõe, envolvendo a estabilidade com grandes perturbações e instabilidade oscilatória.

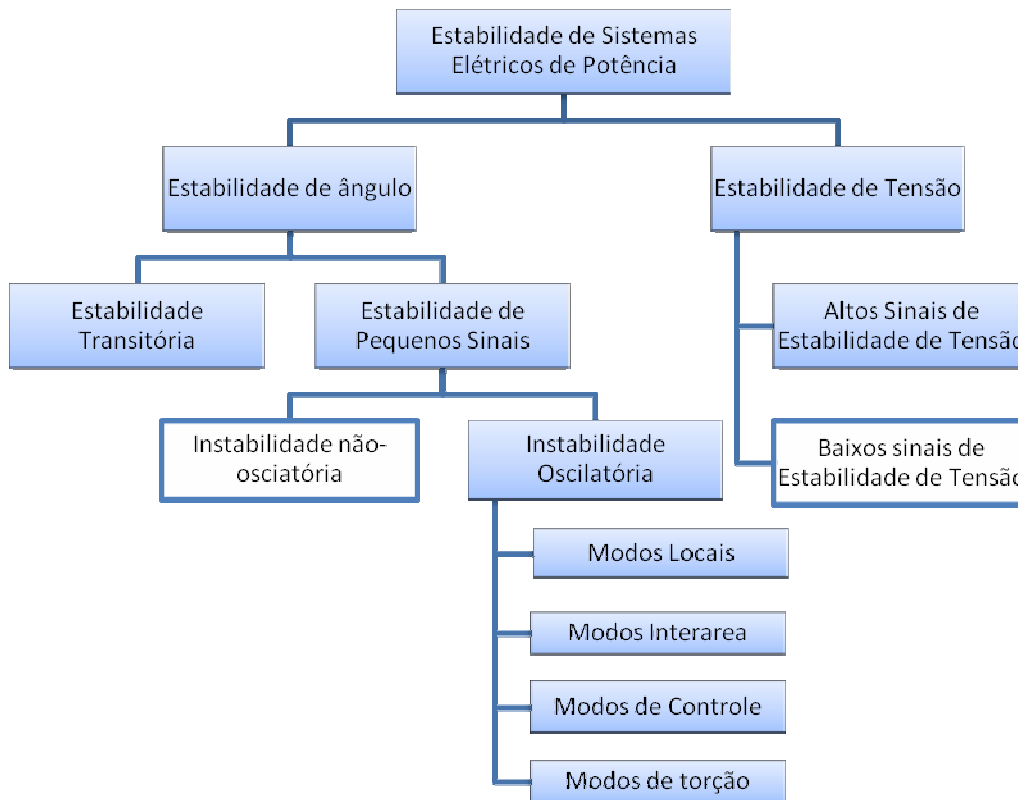


Diagrama 1: Classificação de Estabilidade em Sistemas Elétricos de Potência.

Fonte: Adaptado de Kundur (1994, p. 36).

Os ambientes computacionais modernos, com abundância de memória e alto poder de processamento, fazem de um *Personal Computer* (PC) um ambiente bastante adequado para a simulação de SEPs. Tradicionalmente, estas aplicações são desenvolvidas em linguagens de programação específicas como Fortran, C ou C++, banco de dados baseados em arquivos texto, e desenvolvimento ainda restrito a ambientes acadêmicos e científicos, o que explica, em parte, a baixa quantidade de interfaces gráficas disponíveis (ARRILLAGA; WATSON, 2001).

Atualmente, as plataformas de desenvolvimento de *software* já possibilitam a separação de estruturas da aplicação que podem ser desenvolvidas em linguagens e tecnologias diferentes. Pode-se separar a tecnologia utilizada para construir o núcleo que executa as instruções de simulação com a de outras estruturas direcionadas a fins específicos, como a interface gráfica ou os *frameworks* aplicativos, que são o foco deste estudo, e estendem funcionalidades de um *framework* base.

Os *software* comerciais atuais, específicos para o estudo da dinâmica de SEP, necessitam de muita habilidade e experiência para sua utilização, sendo que usuários novatos ainda necessitam de um elevado tempo de aprendizagem para se familiarizar

com as capacidades e limitações do aplicativo. Dessa forma, gasta-se um tempo considerável até que o usuário possa desenvolver seus próprios dispositivos e sistemas, ainda mais com a modernização dos dispositivos, exigindo também modelos cada vez mais complexos e exigindo cada vez mais das ferramentas de modelagem (ARRILLAGA; WATSON, 2001).

Os modelos são confeccionados através de linguagens de modelagem. Pode-se entender um modelo como algo que se representa em pequena escala e se deseja executar em grande escala. As linguagens fornecem vocabulários e regras para a combinação de palavras com a finalidade de comunicar algo. Portanto, uma linguagem de modelagem é a linguagem cujo vocabulário e regras têm seu foco voltado para a representação conceitual e física de um sistema, permitindo sua compreensão. Nenhum modelo é inteiramente suficiente, necessitando vários modelos, conectados entre si, para tornar possível entender determinados aspectos, ainda que o sistema a ser representado seja o mais trivial possível (BOOCH; RUMBAUGH; JACOBSON, 2000).

No decorrer deste trabalho, são utilizadas três linguagens de modelagem:

- Linguagem matemática: expressando modelos através de conjuntos de equações algébricas ou diferenciais;
- Diagrama de blocos: representações gráficas que permitem representar as relações entre cada elemento de um sistema dinâmico (OGATA, 2003), e
- *Unified Modeling Language* (UML): consagrada como linguagem padrão para modelar sistemas OO.

Como este trabalho trata da modelagem das unidades geradoras em usinas hidrelétricas, as técnicas aqui apresentadas tratam de geradores, seus controladores e atuadores, e sua vizinhança mais próxima, que incluem transformadores, cargas e o equivalente *Thévenin* do restante do sistema elétrico, também chamado de barramento infinito.

Para a análise de SEPs, é comum a normalização das variáveis trabalhadas utilizando a notação “por unidade”. As razões para esta normalização são muitas, dentre as quais: simplificar os cálculos computacionais condicionando melhor os números trabalhados, eliminar unidades de medida, representar variáveis físicas com proporções numericamente menores, entre outras. A equação (1) apresenta como é realizada a normalização (KUNDUR, 1994, p. 75).

$$\text{Quantidade por unidade} = \frac{\text{Quantidade atual}}{\text{Base da quantidade}} \quad (1)$$

Por estas razões, as representações dos modelos aqui apresentados utilizarão, por padrão, a representação de suas variáveis “por unidade” ou em p.u.

2.2 BARRAS

O cálculo do fluxo de carga foge ao escopo deste trabalho, porém, é importante destacar que é através dele que o ponto de operação do sistema é determinado, e só após sua conclusão é iniciada a etapa de simulação dinâmica.

Existem diversos métodos para o seu cálculo, como os métodos de *Gauss*, *Gauss-Seidel*, *Newton* e *Newton-Raphson* (ANDERSON; FOUAD, 1977; MONTICELLI, 1983; ARRILLAGA; WATSON, 2001; KUNDUR, 1994). Independentemente do método, as barras (ou nós de um SEP) são classificadas, de acordo com Kundur (1994), da seguinte forma:

- *Barras PV*, ou *barras de geração*: onde são ligados os geradores, e onde são especificados os valores de potência ativa e magnitude de tensão, e o fluxo de carga converge para encontrar os valores de potência reativa e ângulo do fasor tensão na barra.
- *Barras PQ*, ou *barras de carga*: onde são ligadas as cargas do sistema e onde são especificados os valores de potências ativa e reativa a ser consumido pela carga. O fluxo de carga converge para encontrar os valores de magnitude e ângulo do fasor tensão na barra.
- *Barra V θ* , ou *barra de referência*: é a referência angular do sistema, onde se especifica o ângulo (geralmente 0°, para facilitar a interpretação dos resultados) e a tensão (geralmente 1 p.u.), e o fluxo de carga converge para encontrar os valores de potências ativa e reativa. No estudo deste trabalho, esta barra representa um barramento infinito, que consome a maior parte da energia gerada pelos geradores da UHE, e que é uma aproximação ainda pouco detalhada para o SIN.

2.3 CARGAS

Cargas elétricas, neste estudo, simbolizam os consumidores da energia gerada, como cidades e indústrias. As cargas podem ser modeladas como admitâncias ou como potências negativas ligadas a uma barra (ARRILLAGA; WATSON, 2001). Neste estudo será utilizado o modelo de carga de impedância equivalente apresentado em (2), onde a impedância é calculada a partir do valor de magnitude da tensão na barra e das potências ativa e reativa consumidas, respectivamente P_L e Q_L , que somadas dão a potência aparente da carga.

$$S_L = P_L + jQ_L \quad (2)$$

2.4 MÁQUINAS SÍNCRONAS

As máquinas de geração na UHE de Tucuruí são classificadas como Máquinas Síncronas, ou seja:

Em condições de regime permanente, é uma máquina CA [Corrente Alternada] cuja velocidade é proporcional à frequência da corrente de sua armadura. O rotor, juntamente com o campo magnético criado pela corrente CC [Corrente Contínua] do campo do rotor, gira na mesma velocidade ou em sincronismo com o campo magnético girante, produzido pelas correntes de armadura [...]. (FITZGERALD, KINKSLEY JR., UMANS, 2006, p. 239).

Existem várias formas de modelar uma máquina síncrona, porém, a forma de interesse para estudos de estabilidade dinâmica é através da forma em espaços de estados representado por um conjunto de equações não-lineares diferenciais e algébricas. Utilizando como base Arrillaga e Watson (2001), divide-se o modelo em equações mecânicas e equações elétricas. As equações mecânicas são apresentadas em (3), sendo ω a velocidade angular do rotor da máquina, ω_0 a velocidade angular síncrona ($\omega_0 = 1$ p.u.), f_0 a frequência síncrona (que no Brasil é igual a 60Hz), H a constante de inércia, T_m o torque mecânico, T_e o torque elétrico (em p.u., o torque pode ser considerado numericamente igual à potência, considerando o SEP operando em sincronismo), D o coeficiente de amortecimento e δ o ângulo do rotor da máquina.

$$\begin{cases} \dot{\omega} = \frac{1}{2H} [T_m - T_e - D(\omega - \omega_0)] \\ \dot{\delta} = 2\pi f_0 (\omega - \omega_0) \end{cases} \quad (3)$$

E as equações elétricas são divididas em 5 modelos que variam conforme a complexidade:

- Modelo 1: considera apenas o estado estacionário, não fazendo uso de equações diferenciais elétricas. É um modelo dinâmico de 2ª ordem, representando apenas a dinâmica rotativa.
- Modelo 2: considera os efeitos transitórios do enrolamento de campo da máquina síncrona, fazendo uso de apenas uma equação diferencial elétrica. Modelo de 3ª ordem.
- Modelo 3: considera os efeitos transitórios no eixo “d” e “q”. Modelo de 4ª ordem.
- Modelo 4: considera os efeitos transitórios no eixo “q” e os efeitos subtransitórios nos eixos “d” e “q”. Modelo de 5ª ordem.
- Modelo 5: considera os efeitos transitórios e subtransitórios nos eixos “d” e “q”. Modelo de 6ª ordem.

Neste trabalho foi utilizado o Modelo 4, cujas equações elétricas são apresentadas em (4) (ARRILLAGA; WATSON, 2001; ANDERSON; FOUAD, 1977), considerando as respectivas constantes de fábrica do gerador.

$$\begin{cases} E'_q = [E_f + (X_d - X'_d)I_d - E'_q]/T'_{d0} \\ E''_q = [E'_q + (X'_d - X''_d)I_d - E''_q]/T''_{d0} \\ E''_d = [E'_d + (X'_q - X''_q)I_q - E''_d]/T''_{q0} \end{cases} \quad (4)$$

É importante ressaltar que para a implementação dinâmica em ambiente computacional, o algoritmo de implementação do simulador exige três etapas distintas de cálculo:

1. *Calcular as condições iniciais:* para iniciar as variáveis do bloco dinâmico, executado apenas no início da simulação.
2. *Calcular as derivadas:* utilizado para calcular as derivadas de cada variável de estado. Depende das equações diferenciais e do método de integração a ser utilizado.
3. *Calcular as saídas:* utilizado após os cálculos das derivadas das variáveis de estado para calcular as saídas de cada bloco dinâmico, e depende das variáveis de estado escolhidas em cada subsistema.

O conjunto de equações (3) e (4) apresentam as equações diferenciais do Modelo 4 do gerador. Para calcular as condições iniciais e as saídas, é preciso definir quais são as entradas e saídas que o modelo do gerador disponibilizará e quais serão as variáveis

de estado a serem calculadas.

As entradas utilizadas para o bloco dinâmico do gerador são a potência mecânica P_m , a tensão de campo E_f e as correntes direta I_d e em quadratura I_q .

Para o cálculo das condições iniciais, são utilizadas as variáveis potência ativa P_e , potência reativa Q_e , tensão V e ângulo θ na barra de geração, que são calculadas pelo fluxo de carga. Assim, as condições iniciais são definidas conforme segue em (5) (ARRILLAGA; WATSON, 2001; ANDERSON; FOUAD, 1977).

$$\left\{ \begin{array}{l} S = P + jQ, \quad V_t = V \angle \theta, \quad I = \text{conj} \left(\frac{S}{V_t} \right), \quad \varphi = \arg(S), \\ I_r = |I| \cos(\varphi), \quad I_x = -|I| \sin(\varphi), \\ \delta_{int} = \text{atan} \left(\frac{X_q I_r + R_a I_x}{V + R_a I_r - X_q I_x} \right), \quad \delta = \delta_{int} + \theta, \\ V_d = -V \sin(\delta_{int}), \quad V_q = V \cos(\delta_{int}), \\ I_d = -|I| \sin(\delta_{int} + \varphi), \quad I_q = |I| \cos(\delta_{int} + \varphi), \\ E''_d = V_d + X''_q I_q + R_a I_d, \\ E''_q = V_q + X''_d I_d + R_a I_q, \\ E'_q = E''_q - (X'_d - X''_d) I_d, \\ E_f = E'_q - (X_d - X'_d) I_d, \\ \omega = 1, \quad P_e = P + R_a |I|^2, \quad P_m = P_e, \\ P_t = V_d I_d + V_q I_q, \quad Q_t = -V_q I_d + V_d I_q \end{array} \right. \quad (5)$$

Para o cálculo das saídas, temos as equações apresentadas em (6).

$$\left\{ \begin{array}{l} V_q = E''_q - R_a I_q + X''_d I_d \\ V_d = E''_d - R_a I_d - X''_q I_q \\ V_t = \sqrt{V_d^2 + V_q^2} \\ P_t = V_d I_d + V_q I_q \\ Q_t = -V_q I_d + V_d I_q \end{array} \right. \quad (6)$$

Também é necessário medir a corrente de campo I_{fd} do gerador para algumas aplicações. Portanto, somam-se às equações das condições iniciais e do cálculo das saídas a equação (7), definida em Kundur (1994, p. 179-180).

$$I_{fd} = \frac{E'_q + (X_d - X'_d)(-I_d)}{X_d - X_L} \quad (7)$$

Modelada a máquina síncrona, apresenta-se, em seguida, a modelagem dos sistemas de transmissão e dos controles das máquinas. Estes últimos constituídos pelo Regulador Automático de Tensão (RAT), Estabilizador de Sistema de Potência (ESP) e Regulador de Velocidade (RV).

2.5 SISTEMAS DE TRANSMISSÃO

Os sistemas de transmissão convencionais são compostos por uma complexa rede de componentes passivos, principalmente linhas de transmissão e transformadores. Dentro dos objetivos dos estudos de estabilidade, é comum o uso de circuitos contendo os efeitos de indutância, capacitância e resistência para representar seus comportamentos (ARRILLAGA; WATSON, 2001).

Os SEPs são trifásicos, porém, nos estudos aqui realizados serão utilizadas representações unifilares. Neste contexto, as linhas de transmissão e os transformadores são modelados da mesma forma, com o chamado modelo π , ou π -equivalente (ARRILLAGA; WATSON, 2001; KUNDUR, 1994), como mostra o Diagrama 2, que contem os efeitos da impedância de curto-circuito (Z_e) e da admitância de magnetização ou de circuito aberto (Y_e), e as tensões em cada terminal identificadas por V_P e V_S .

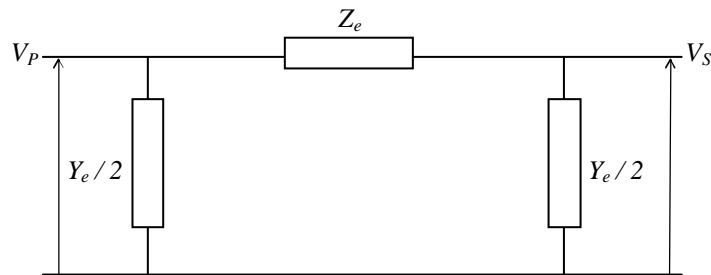


Diagrama 2: Modelo π , usado para modelar Linhas de Transmissão e Transformadores.

Fonte: Adaptado de Arrillaga e Watson (2001, p. 8) e Kundur (1994, p. 207, 236).

2.6 REGULADOR AUTOMÁTICO DE TENSÃO

O RAT é um controlador que monitora a tensão terminal do gerador V_t e atua no enrolamento de campo E_f . O sinal de V_t é comparado com o valor de referência (também chamado de *setpoint*) com o objetivo de manter a tensão terminal regulada (ARRILLAGA; WATSON, 2001).

Como existem diversas formas de se modelar um RAT, neste item será apresentado apenas um modelo básico apresentado em Arrillaga e Watson (2001, p. 237-239). De acordo com este modelo, o qual está representado no Diagrama 3, as variáveis de entrada mais comuns são a tensão de referência V_{ref} , a tensão terminal do gerador V_t e o sinal do ESP V_o . A tensão terminal passa por um filtro passa-alta e a diferença entre este sinal, a tensão de referência, o sinal do ESP e a realimentação passam por regulador amplificador, o qual é o bloco que guarda a principal dinâmica deste sistema, que tende a ter uma resposta simplificada como o de um sistema de primeira ordem. A excitatriz seria o bloco atuador e que tem como saída a tensão de campo E_f , que entra no gerador.

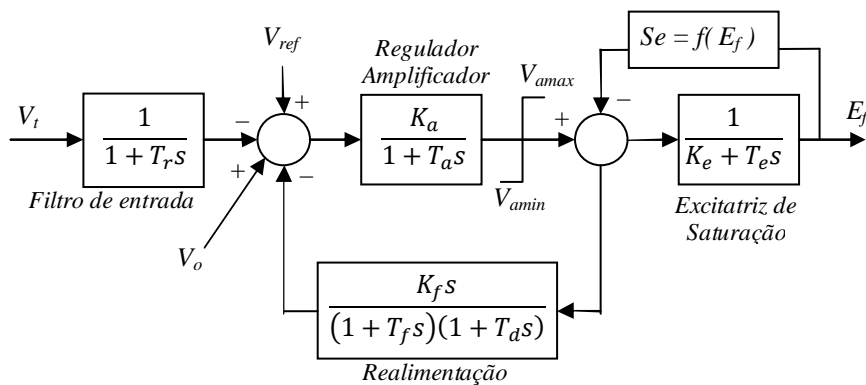


Diagrama 3: Regulador Automático de Tensão básico.

Fonte: Adaptado de IEEE Committee Report (1968 apud ARRILLAGA; WATSON, 2001, p. 238)

2.7 ESTABILIZADOR DE SISTEMA DE POTÊNCIA

Um controlador que pode atuar em conjunto com o RAT é o ESP. Sua função principal é a de amortecer pequenas oscilações eletromecânicas de baixa frequência, atuando diretamente no RAT, tendo como saída o sinal V_o , que é somado à referência do RAT (KUNDUR, 1994).

O Diagrama 4 apresenta um esquemático básico de um ESP, que tem como entrada a velocidade angular do gerador ω . Contêm um filtro passa-baixa para reduzir ruídos, um filtro *washout* ou passa-alta para eliminar o valor médio do sinal, dois compensador de fase ou blocos *lead-lags* e um ganho para determinar o sinal do amortecimento que ainda passa por um limitador. Porém, um ESP também pode ser projetado para ter como entrada outras variáveis como a potência ativa.

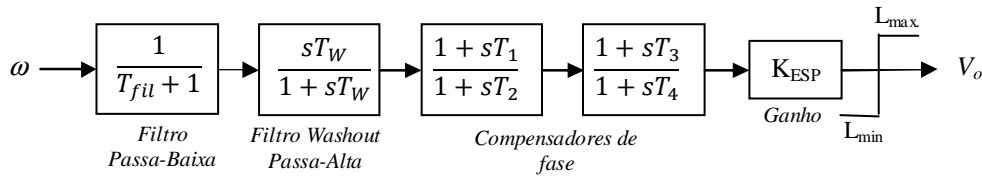


Diagrama 4: Estabilizador de Sistema de Potência básico.

Fonte: Baseado em Kundur (1994) e Kundur et al. (2003).

2.8 REGULADOR DE VELOCIDADE

No que se refere a geradores hidráulicos, o RV serve para regular a velocidade de rotação da turbina, controlando a abertura do distribuidor, com o objetivo de manter a frequência de rotação da máquina de acordo com a carga que está sendo demandada (KUNDUR, 1994).

Arrillaga e Watson (2001, p. 239-241) apresentam modelos gerais de RVs. O Diagrama 5 apresenta um destes modelos, que tem como entrada a velocidade elétrica angular do gerador ω , e sua referência $\omega_0 = 2\pi f_0$, a posição inicial de abertura da válvula G_s , e a potência da água P_b . O primeiro limitador após o somador de G_s e G_2 é um limitador da derivada desta soma, e o segundo é um limitador de amplitude desta derivada. P_{gv} é o sinal enviado à turbina para acelerar ou desacelerar a rotação da máquina, através do comando de abertura ou fechamento do distribuidor que controla a vazão da água, que é a fonte de energia primária.

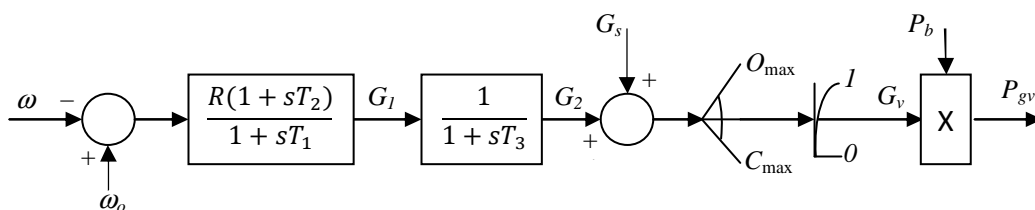


Diagrama 5: Regulador de Velocidade básico.

Fonte: Adaptado de IEEE Committee Report (1973 apud ARRILLAGA; WATSON, 2001, p. 240)

2.9 TURBINAS HIDRÁULICAS

As turbinas hidráulicas estão em contato direto com a fonte de energia primária que é a água do reservatório. Arrillaga e Watson (2001, p. 241-242) apresentam um modelo clássico e linear de turbina hidráulica, conforme apresenta o Diagrama 6. Como

entrada tem-se o sinal P_{gv} , que é a abertura do distribuidor, e como saída a potência mecânica P_m , que será uma das entradas do gerador.

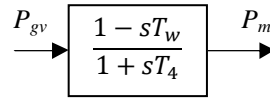


Diagrama 6: Turbina hidráulica linear.

Fonte: Adaptado de Arrillaga e Watson (2001, p. 241).

Modelos mais complexos, detalhados e não-lineares de turbinas hidráulicas também são apresentados nos Diagrama 7 e 8. No Diagrama 7, além de P_{gv} , também existem como entradas a constante H_0 que é o valor inicial da altura do reservatório, U_{nl} que é a velocidade da água do reservatório sem carga pendurada no gerador, e a variação de velocidade angular do gerador $\Delta\omega$.

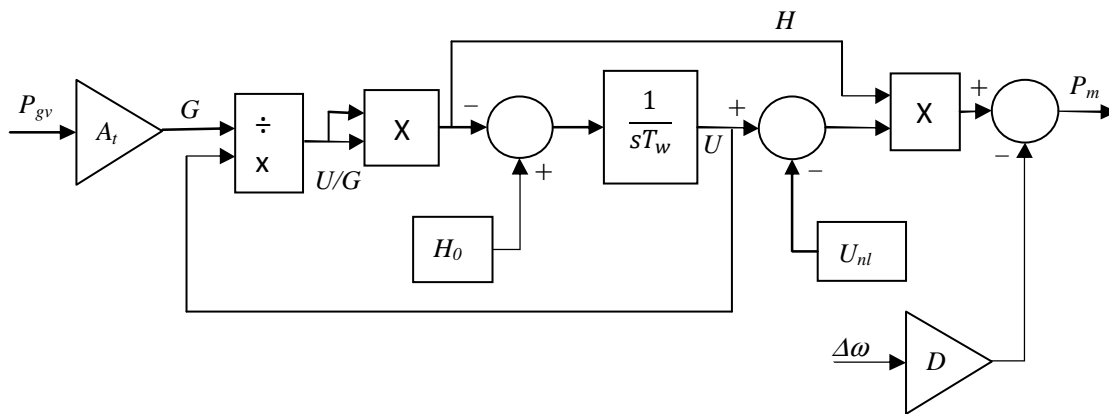


Diagrama 7: Turbina hidráulica não-linear.

Fonte: Adaptado de Kundur (1994, p. 391) e Operador Nacional do Sistema Elétrico (2003a, p. 5).

No Diagrama 8 nota-se que são consideradas as mesmas variáveis de entrada do Diagrama 7, porém, $\Delta\omega$ é multiplicado por P_{gv} e o produto passa pelo ganho D_t . A saída do somador final ainda é multiplicada pela divisão da potência base da turbina e a potência base da máquina, para aí sim produzir a potência mecânica P_m .

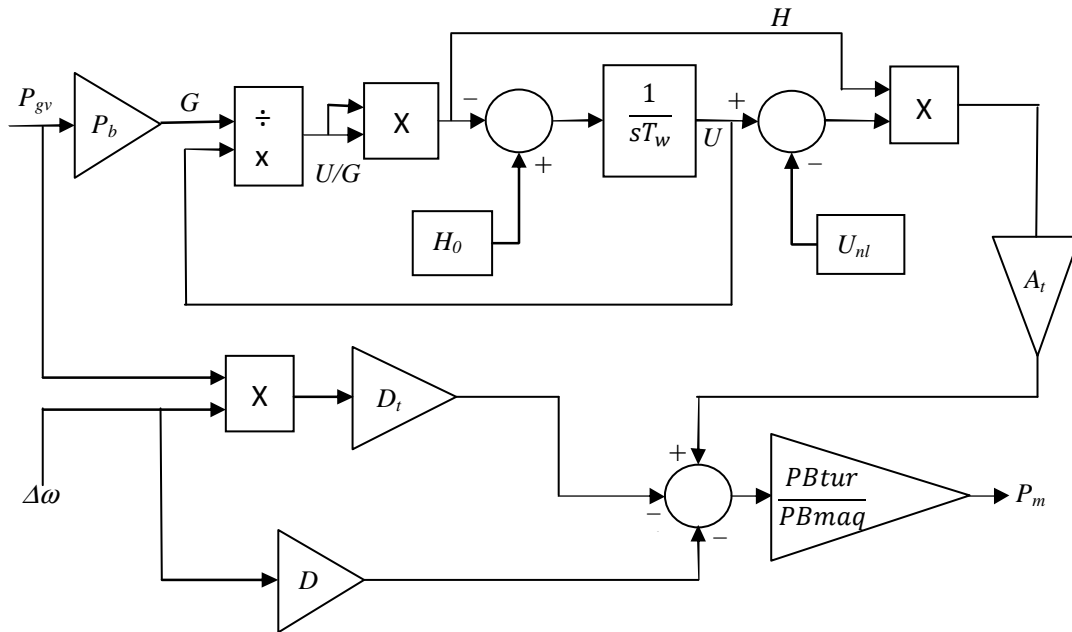


Diagrama 8: Turbina hidráulica não-linear.

Fonte: Adaptado de Operador Nacional do Sistema Elétrico (2008a) e Working... (1992, p. 168).

3 PADRÕES DE PROJETO ORIENTADOS A OBJETO

3.1 PROCESSO EM PROJETOS ORIENTADOS A OBJETO

Na engenharia de *software*, processos podem ser definidos como atividades de desenvolvimento, manutenção, aquisição e contratação de *software*, podendo, inclusive, definir sub-processos para cada um desses. Não é o foco deste trabalho discorrer sobre processos de *software* e suas implicações, porém, é importante destacar alguns conceitos adotados que nortearam o processo de desenvolvimento do Simulador. O processo de desenvolvimento tem como sub-processos a determinação de requisitos, análise, desenho, implementação e testes (PAULA FILHO, 2003).

Paula Filho (2003, p. 12) argumenta que o ponto de partida para a escolha de um processo de desenvolvimento de *software* inicia na escolha do ciclo de vida. A literatura apresenta diversos ciclos, dentre os quais (PAULA FILHO, 2003; SOMMERVILLE, 2003): codifica-remenda, cascata, cascata com realimentação, espiral, prototipagem evolutiva, entrega evolutiva, desenvolvimento formal de sistemas, desenvolvimento orientado a reuso, desenvolvimento incremental, dentre tantos outros.

Feita a escolha do ciclo mais adequado para o projeto, inicia-se a definição do processo de desenvolvimento. O processo de desenvolvimento deve ser bem definido e organizado, preocupando-se constantemente com sua evolução. São vários os processos existentes, tais como *Personal Software Process (PSP)*, *Rational Unified Process (RUP)*, *Extreme Programming (XP)*, Práxis, entre outros (PAULA FILHO, 2001; SOMMERVILLE, 2003).

Um processo de desenvolvimento OO inicia-se com a análise de requisitos e posterior elaboração e avaliação dos casos de uso, garantindo que o sistema se adequará à organização, e não o contrário, atestando o atendimento dos requisitos do usuário. Deve ser bastante flexível, permitindo reavaliações e readequações, dividindo-se genericamente em quatro fases (MATOS, 2002):

- a) **Concepção:** os requisitos são identificados e avaliados, fomentando discussões para as constatações dos requisitos do *software*, podendo-se utilizar ferramentas CASE (*Computer-Aided Software Engineering*) para UML ou protótipos;

- b) **Elaboração:** identificação e avaliação dos riscos, imprescindíveis para que um plano de projeto seja traçado e uma arquitetura de *software* seja vislumbrada;
- c) **Construção:** implementação do *software*, considerando a tecnologia a ser utilizada, e
- d) **Transição:** disponibilização do *software* para seus usuários para constatação dos objetivos, reavaliando documentações e modelos propostos.

Este é o contexto em que se aplicam os processos de desenvolvimento de *software* e, mais especificamente, o contexto adotado para o desenvolvimento do Simulador. Como dito anteriormente, não é o objetivo aqui abordar os detalhes do processo de desenvolvimento de *software* adotado, e sim apresentar este contexto em que foram aplicadas as técnicas de modelagem e implementação adotadas na implementação das unidades geradoras da UHE de Tucuruí em ambiente computacional.

3.2 CONCEITOS FUNDAMENTAIS DA ORIENTAÇÃO A OBJETOS

Para a OO, as classes são os blocos fundamentais de construção de qualquer *software*. Uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica, podendo implementar uma ou mais interfaces. Utilizam-se classes para capturar o vocabulário do *software* a ser desenvolvido, incluindo abstrações e implementações, podendo representar itens de *software*, *hardware* e até puramente conceituais (BOOCK; RUMBAUGH; JACOBSON, 2000, p. 47).

Uma forma prática de definir as classes é buscar por substantivos na especificação funcional de um *software* (HORSTMANN, 2007). Em SEPs, por exemplo, alguns dos substantivos são: geradores, RATs, ESPs, RVs e turbinas.

Objetos são instâncias de classes concretas (não abstratas) que armazenam informações resultantes de operações prévias, possuindo três características (HORSTMANN, 2007, p. 57).

- a) *Estado:* as informações podem determinar como o objeto se comportará

no futuro, e o conjunto de todas elas corresponde ao seu estado. O estado de um objeto pode se alterar ao longo do tempo, mas somente quando uma operação que implica em mudança de estado é realizada.

- b) *Comportamento*: é definido pelas operações, ou métodos, que suporta, além de eventos que deve responder.
- c) *Identidade*: é a forma de distinguir um objeto de outro, por mais que possuam estados e comportamentos idênticos. O nome do objeto é uma forma de identificá-lo durante a execução do programa pelo computador.

Assim como as classes correspondem aos substantivos da descrição funcional do *software*, os verbos correspondem às responsabilidades. Definida uma responsabilidade, deve-se encontrar a classe (e apenas uma classe) que assuma essa responsabilidade (HORSTMANN, 2007). Para SEPs, no item 2.2 deste trabalho, por exemplo, foram definidas as etapas de cálculo das equações de estado para implementação não só de uma máquina síncrona, como de qualquer bloco dinâmico, as quais são identificadas através de verbos: calcular as condições iniciais, calcular as derivadas e calcular as saídas.

Além da definição das classes e seus atributos, propriedades e comportamentos, também deve-se definir o relacionamento entre elas. A OO define uma diversidade de relacionamentos, tais como herança, dependência, agregação, composição, associação, herança múltipla, navegação associada, realização, dependência estereotipada, entre outros. Um estudo mais aprofundado de análise OO deve ser feito para estruturar o modelo. Boock, Rumbaugh e Jacobson (2000) apresentam uma diversidade desses relacionamentos e técnicas para identificá-los.

A linguagem de modelagem adotada atualmente em *software* OO é a UML. Com a UML é possível:

- a) Mostrar as fronteiras de um sistema e suas funções principais utilizando atores e casos de uso;
- b) Ilustrar a realização de casos de uso com diagramas de interação;
- c) Representar a estrutura estática de um sistema utilizando diagramas de classe;
- d) Modelar o comportamento de objetos com diagramas de transição de estado;
- e) Revelar a arquitetura de implementação física com diagramas de componente e implementação, e
- f) Estender sua funcionalidade através de estereótipos. (FURLAN, 1998, p. 32).

A UML passou a ser a ferramenta ideal para conceber, compreender, testar, validar, arquitetar (lógica e fisicamente) e ainda identificar todos os possíveis comportamentos do sistema, sendo bastante difundida no meio acadêmico e implementada por grandes empresas.

O modelo proposto pela UML é baseado em um conjunto de diagramas que abordam aspectos físicos e lógicos, e aspectos estáticos e dinâmicos do problema. Cada diagrama aborda aspectos distintos, sendo necessária uma abstração diferente para a elaboração de cada diagrama, estando todos, de alguma forma, ligados e coerentes com a análise do problema. Para Boock, Rumbaugh e Jacobson (2000), os diagramas da UML são:

- Casos de Uso;
- Interação:
 - Sequência, e
 - Colaboração.
- Gráficos de Estado;
- Atividades;
- Classes;
- Objetos;
- Componentes, e
- Implementação.

Os diagramas relacionados a aspectos físicos são os que modelam elementos tangíveis, como servidores ou elementos de *software* que interagem com o sistema, como bibliotecas (*frameworks*) ou arquivos executáveis. Já aspectos lógicos são os relacionados à regra de negócios do sistema.

Aspectos estáticos do sistema são representações de seu esqueleto, ou seja, modelos relativamente estáveis, relacionados à sua estrutura. Aspectos dinâmicos estão relacionados ao comportamento do sistema, representando partes que sofrem alteração, e como elas poderão sofrer.

Os diagramas se relacionam na UML como mostra o Diagrama 9:

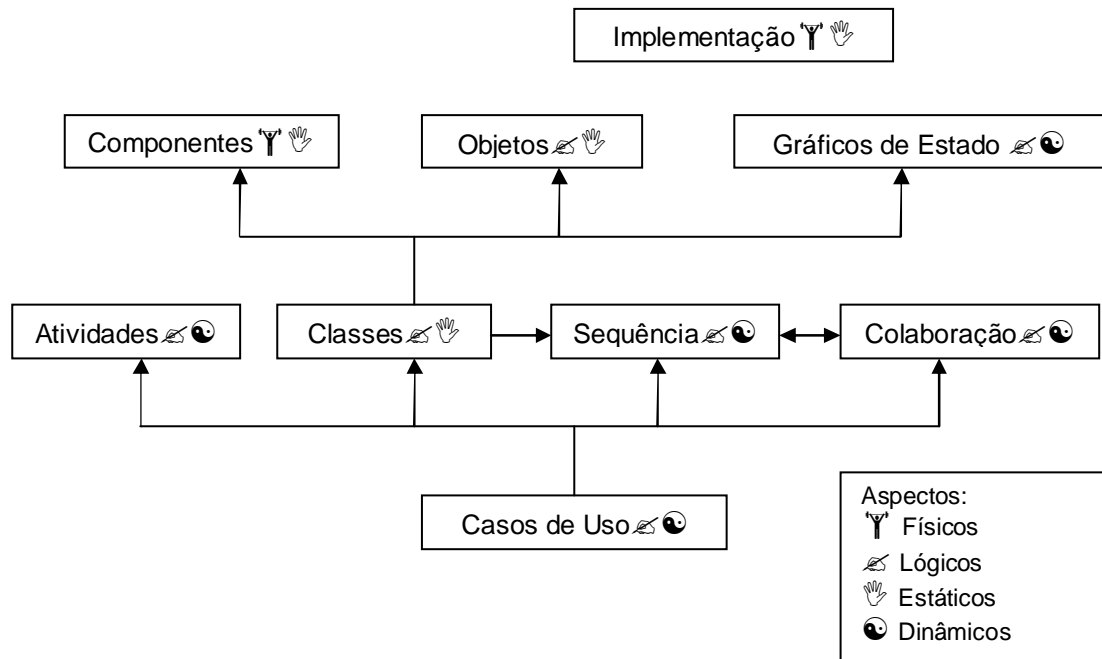


Diagrama 9: Relacionamento entre os Diagramas da UML.

Fonte: Adaptado de Silva (2001, p. 15).

No presente trabalho são abordados especificamente os aspectos estáticos e estruturais da modelagem OO para SEPs, principalmente com o uso de diagramas de classes que foram gerados com uso da ferramenta Enterprise Architect (ENTERPRISE..., 2005), que já implementa a notação UML 2.0, utilizando a técnica dos padrões de projeto. Outras abordagens e aspectos também foram utilizados durante o processo de desenvolvimento, os quais não são detalhados neste documento, mas que podem ser consultados nos relatórios produzidos no projeto de P&D (BARRA JUNIOR, 2008a, 2008b).

3.3 PADRÕES DE PROJETO

Antes de definir o que é um padrão de projeto, deve-se compreender o que é um *framework*. Um *framework* é um conjunto de classes cooperativas que implementam mecanismos que são essenciais para um domínio específico de problemas. Um programador pode criar funcionalidades novas no domínio do problema que está trabalhando estendendo as classes do *framework*. Já um *framework* de aplicação é um conjunto de classes que implementa serviços comuns a certo tipo de aplicação (HORSTMANN, 2007, p. 312).

Um padrão de projeto (ou *design pattern*) é uma regra geral para a modelagem e implementação de projetos OO (HORSTMANN, 2007). Padrões de projeto também podem ser definidos como itens de uma espécie de catálogo de problemas recorrentes que podem ser resolvidos de maneira sistematizada.

Pela grande quantidade existente de padrões de projeto, este catálogo pode ser organizado de diversas formas. As formas mais convencionais de organização são pelo seu escopo e pelos propósitos gerais (GAMMA et al., 2007).

Organização pelo escopo significa dizer se o padrão é aplicado primariamente a classes ou a objetos.

Os padrões de classes descrevem como as relações entre as classes são definidas através da hereditariedade [...] estabelecidas no momento da compilação.

Os padrões de objeto descrevem os relacionamentos entre os objetos e são definidos primariamente por composições[, ...] criados durante a execução e são muito mais dinâmicos e flexíveis (FREEMAN; FREEMAN, 2007, p. 469).

Pelos propósitos gerais, os padrões podem ser organizados como (FREEMAN; FREEMAN, 2007; GAMMA et al., 2007):

- *Criacionais*: envolvem a criação de instâncias de objetos, os quais fornecem alguma maneira de desconectar o cliente (responsável pelas requisições) dos objetos a partir dos quais serão geradas as instâncias.
- *Estruturais*: permitem a organização de classes ou objetos em estruturas maiores.
- *Comportamentais*: preocupam-se com a forma como as classes e objetos interagem e com a distribuição de responsabilidades.

Os avanços obtidos na pesquisa do *framework* BCSSD permitiram que agora ela pudesse ser utilizada na implementação da simulação dinâmica das unidades geradoras da UHE de Tucuruí, apresentada neste trabalho. Para a implementação deste *framework* de aplicação, foram utilizadas estratégias de padrões de projeto da *Gang of Four* (GoF), apresentadas em Freeman e Freeman (2007) e Gamma et al. (2007).

O estudo e desenvolvimento de aplicação de estratégias de padrões de projeto para SEPs não é recente, como será visto no Capítulo 4. Por hora, cabe apresentar os padrões de projeto utilizados no desenvolvimento e como foram úteis.

3.3.1 Factory Method

O padrão de projeto *Factory Method* possui escopo nas classes, não nos objetos, e é criacional. Ou seja, o cliente responsável por requisitar uma de suas instâncias não está interessado em detalhes da dinâmica interna das implementações, e sim em características gerais.

Como está centrado nas classes, o *Factory Method* define uma interface para criar um objeto, mas permite decidir qual classe instanciar, permitindo a uma classe deferir a instanciação para subclasses (FREEMAN; FREEMAN, 2007). É basicamente uma “fábrica de métodos”, cujos algoritmos específicos podem ser encapsulados em classes especializadas, enquanto uma classe maior agrega funcionalidades gerais.

Sua utilização neste trabalho é para o mapeamento entre os diagramas de blocos e a OO. A classe que agrega as funcionalidades gerais é a própria representação do bloco, enquanto que as classes especializadas são as que representam os elementos físicos do SEP.

O diagrama de classes que representa a estrutura para aplicação do *Factory Method* é apresentado no Diagrama 10. A classe abstrata *Produto* define a interface de objetos para a criação da fábrica de objetos. *ProdutoConcreto* implementa a interface ou classe abstrata *Produto*. *Criador* é a classe abstrata em que a fábrica de métodos é declarada com os métodos principais que serão usados pela *ProdutoConcreto*. E, *CriadorConcreto* se sobrepõe à fábrica de métodos *Criador* para retornar à *ProdutoConcreto*. *Criador* depende de suas subclasses para definir a fábrica de métodos e retornar uma instância adequada de um *ProdutoConcreto*.

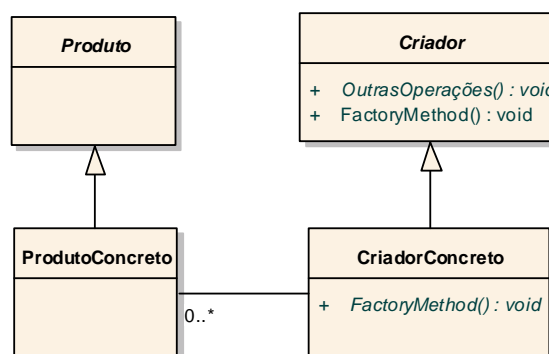


Diagrama 10: Diagrama de Classes para implementação do Padrão de Projeto *Factory Method*.

Fonte: Adaptado de Gamma et al. (2007, p. 108).

O mapeamento entre paradigmas de modelagem é muito comum em sistemas de engenharia. Entre os paradigmas de modelagem e programação, por exemplo, existe o procedimental, estrutural, orientado a objetos, orientado a aspectos e o relacional.

A própria matemática faz uso deste artifício para simplificar a complexidade de cálculos para se chegar a uma solução ou melhorar a análise dos resultados de um problema, como convertendo sistemas no domínio do tempo para a frequência (OGATA, 2003).

O fato é que esse mapeamento é sempre necessário porque, para um determinado contexto, um paradigma pode ser mais apropriado que outro. As razões são várias, como a popularidade de um paradigma, sua simplicidade de implementação, entre outros.

Em SEPs e em estudos de controle, é comum a representação de sistemas dinâmicos em diagramas de blocos, com a dinâmica representada tanto no domínio do tempo quanto no da frequência (OGATA, 2003; KUNDUR, 1994). Para a implementação em linguagens de programação modernas (C++, Java, C#, entre outras), o paradigma mais utilizado é o de OO (BOOCH, RUMBAUGH, JACOBSON, 2000; MATOS, 2002; SENA et al., 2005; SENA, 2006).

Diagramas de blocos são representações gráficas que permitem representar as relações entre cada elemento de um sistema (OGATA, 2003). Existe a facilidade de se construir o diagrama global do sistema conectando-se os diversos módulos conforme suas relações; e a possibilidade de ser utilizado pelos engenheiros para representar um número muito grande de sistemas.

Algumas ferramentas CASE já possuem um ambiente para programação baseado em diagramas de blocos, como o Simulink (MATLAB, 2008). Porém, quando há a necessidade de se programar em ambientes OO, esbarra-se sempre na escolha da metodologia a ser utilizada: como determinar os parâmetros dos blocos? Como aplicar a integração numérica? Como representar os sistemas dinâmicos?

Neste contexto, foi aplicado o padrão *Factory Method* para deixar o programador livre para modelar qualquer bloco que queira, simplesmente escrevendo nos códigos-fontes as equações dinâmicas que caracterizam o bloco, deixando encapsulados todos os demais “comportamentos” esperados para um bloco. Assim, pode-se representar através de uma linguagem de programação OO qualquer sistema que possa ser representado em diagrama de blocos.

O Diagrama 11 apresenta a estrutura de aplicação do padrão *Factory Method* na BCSSD, com as devidas associações apresentadas no Diagrama 10, para efetuar o mapeamento. Todo bloco possui uma estrutura básica codificada na classe *Bloco*, cujos elementos do sistema de potência são codificados, herdando estas características básicas e implementando suas funcionalidades específicas.

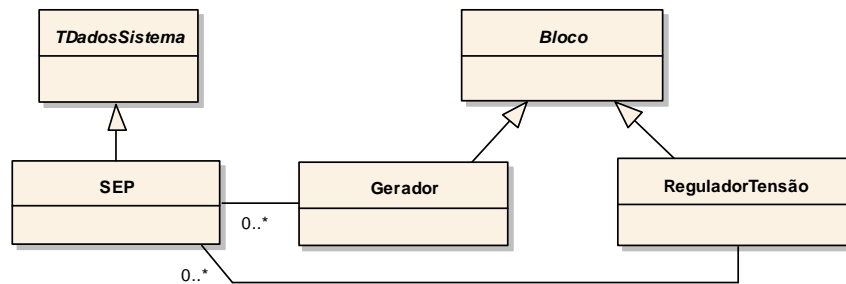


Diagrama 11: Arquitetura do Padrão de Projeto *Factory Method* para mapeamento entre diagrama de blocos e orientação a objetos.

O Diagrama 12 apresenta alguns detalhes das classes responsáveis por armazenar a estrutura do diagrama de blocos e executar a simulação. *TDadosSistema* é uma classe abstrata que contém os parâmetros do sistema. A BCSSD foi construída em C++, logo, suas coleções são representadas por ponteiros, como nos atributos *Geradores* e *ReguladoresTensão*. Todos os elementos do sistema são representados por atributos. Além deles, as demais estruturas utilizadas para execução do fluxo de carga e da própria simulação também são declaradas, como a matriz admitância de rede (Y) e a sua representação reduzida (Y_{red}). Os métodos que implementam as faltas no sistema também podem ser implementados aqui, como *SimularCurtoCircuito(...)*, usado para simular curtos-circuitos trifásicos, *SimularDegrauReferenciaRAT(...)* utilizado para simular um degrau na referência dos reguladores de tensão e *SimularDegrauReferenciaRV(...)* utilizado para simular um degrau na referência dos reguladores de velocidade. A classe *SEP* (Sistema Elétrico de Potência) é uma classe concreta que herda estas características, e possui efetivamente a representação do sistema orientado a objetos devidamente mapeado através dos diagramas de blocos, e executa as simulações de acordo com os parâmetros informados.

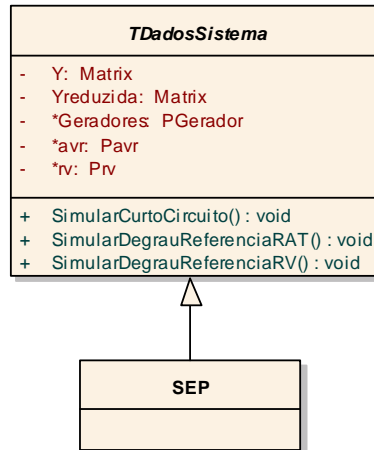


Diagrama 12: Detalhes das classes principais que armazenam a estrutura do diagrama de blocos para executar as simulações.

A “fábrica de métodos” é, então, apresentada com alguns detalhes no Diagrama 13. A classe *Bloco* possui os atributos como os pontos de entrada e saída dos dados, e os métodos abstratos para implementar suas funcionalidades, como obter os valores de entrada e de saída dos respectivos pontos, efetuar as conexões com outros blocos, conectando entradas de um bloco na saída de outro, por exemplo. Além disso, os métodos para o cálculo das derivadas ou variáveis de estado dos blocos, das saídas e das condições iniciais são declarados, mas implementados nas classes especializadas.

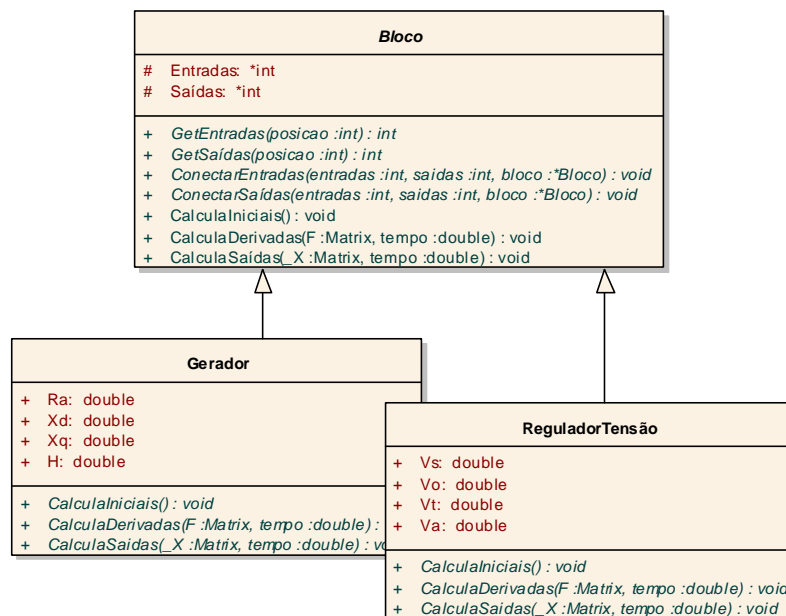


Diagrama 13: Detalhes das classes que mapeiam o diagrama de blocos para sistemas de potência.

As classes especializadas, que herdam da classe *Bloco*, são as que representam os elementos do sistema de potência. O Diagrama 13 mostra duas dessas classes com alguns detalhes: *Gerador* e *ReguladorTensão*. Nessas classes são declarados itens

específicos de cada bloco. Geradores, por exemplo, têm resistência de armadura (R_a), reatâncias de eixos (X_d e X_q), momento de inércia (H), entre outros. Uma maneira simples de representar Reguladores de Tensão é através de um bloco de primeira ordem, com um ganho (K_A) e uma constante de tempo (T_A). Posteriormente, no Capítulo 4 serão vistos com mais detalhes as modelagens dos elementos físicos do sistema.

São nessas classes especializadas que os métodos *CalculaDerivadas(...)*, utilizado para calcular as variáveis de estado em cada passo de integração, e *CalculaSaídas(...)*, utilizado para atualizar o vetor de saídas, são implementados, concluindo a estrutura do padrão de projeto e o mapeamento entre diagrama de blocos e a orientação a objetos.

3.3.2 Strategy

O padrão de projeto *Strategy* possui escopo nos objetos, não nas classes, e é comportamental. Ou seja, preocupa-se com a forma como as classes e objetos interagem e com a distribuição de responsabilidades.

Sua utilização neste trabalho serve como complementação ao *Factory Method*, definindo a dinâmica de cada bloco, para a implementação dos dispositivos de controle de regulação de tensão, ESP, regulador de velocidade dos geradores, e turbinas, já que cada dinâmica constitui uma responsabilidade diferente no paradigma OO. O Diagrama 14 apresenta a estrutura para aplicação do padrão *Strategy*, que contém a classe abstrata *Strategy* servindo de interface comum para todos os algoritmos implementados nas classes concretas *EstratégiaConcretaA*, *EstratégiaConcretaB* e *EstratégiaConcretaC*, e a classe *Contexto* é a que configura os objetos, mantendo as referências a cada um deles e definindo uma interface que permite a estratégia de acesso aos dados.

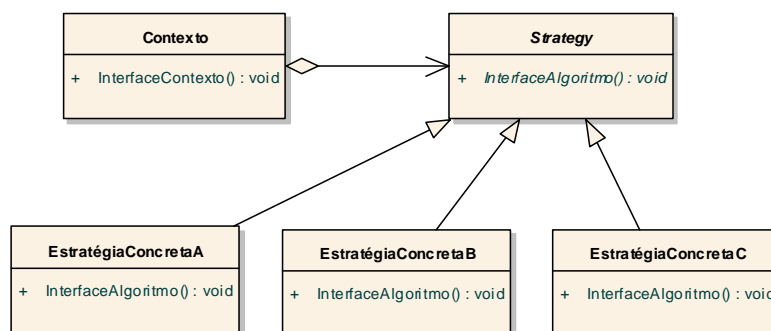


Diagrama 14: Diagrama de Classes para implementação do Padrão de Projeto *Strategy*.

Fonte: Adaptado de Gamma et al. (2007, p. 316).

O Diagrama 15 apresenta a configuração final da implementação do padrão *Strategy* para a implementar RATs e RVs. Os ESPs e as turbinas não estão apresentados, mas seguem a mesma lógica. Na classe *TDadosSistema*, que representará a instância do SEP, tem-se as coleções que armazenarão os RATs (**avr*), RVs (**rv*), ESPs (**esp*) e turbinas (**turb*). A classe abstrata *ReguladorVelocidade* fornece uma interface para a criação de diversos tipos de RVs, como reguladores hídricos, térmicos ou eólicos, com suas respectivas dinâmicas detalhadas. O mesmo para os reguladores de tensão. Nota-se um pequeno ajuste de modelo com relação à aplicação do *Factory Method*, pois com a aplicação deste padrão, a classe *TDadosSistema* passa a ser concreta e as classes *ReguladorTensão* e *ReguladorVelocidade* passam a ser abstratas, e continuam sendo especializações da classe *Bloco*. A classe *SEP* também deixa de existir, representando-se um sistema através de uma instância de *TDadosSistema*. Assim, a junção de diversos padrões começa a ajustar o modelo para uma implementação mais eficaz.

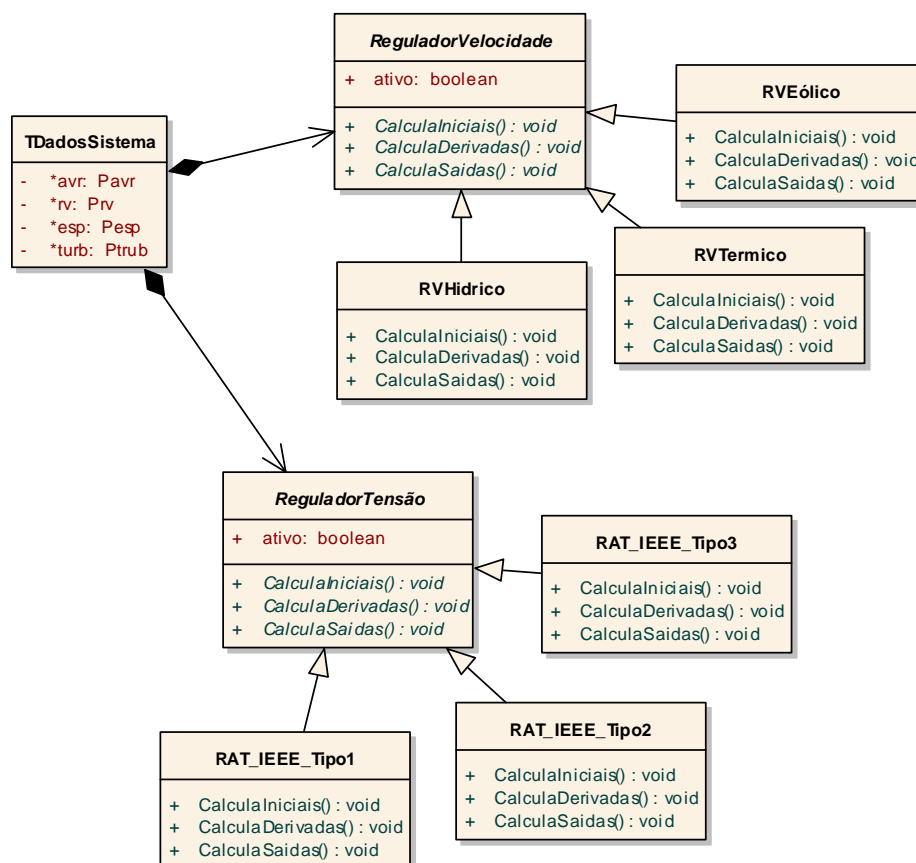


Diagrama 15: Arquitetura do Padrão de Projeto *Strategy* definindo a dinâmica específica de cada dispositivo do sistema.

3.3.3 Builder

O padrão de projeto *Builder* também é criacional e, como está centrado nas classes, separa a construção de um objeto complexo de sua representação, para que o mesmo processo de construção possa criar diferentes representações.

Este conceito é utilizado neste trabalho para abstrair do desenvolvedor das classes concretas, que representam os elementos de sistemas de potência, o método numérico utilizado na simulação. Assim, diferentes métodos numéricos podem ser implementados sem interferir na modelagem e implementação dos demais elementos do sistema.

O diagrama de classes que representa a estrutura para aplicação do *Builder* é apresentado no Diagrama 16. A classe abstrata *Builder* especifica a abstração que servirá de interface para a criação dos objetos, que são instâncias das classes *BuilderConcreto* e implementam a classe *Builder*. São estas as classes que definem e implementam as representações do produto final. A classe *Diretor* é o cliente que usa as implementações em *BuilderConcreto*, através das interfaces comuns implementadas em *Builder*. Neste trabalho, os produtos finais são as classes que encapsulam os métodos numéricos para a simulação dinâmica.

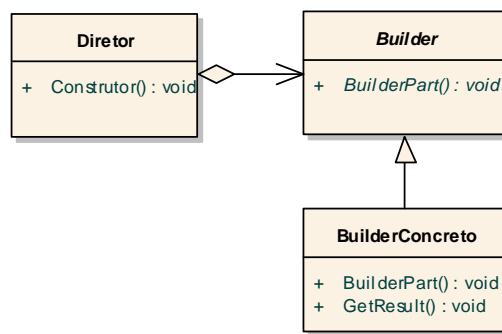


Diagrama 16: Diagrama de Classes para implementação do Padrão de Projeto *Builder*.

Fonte: Adaptado de Gamma et al. (2007, p. 97).

O padrão de projeto *Builder* possui os seguintes benefícios (FREEMAN; FREEMAN, 2007, p. 486):

- Encapsula a maneira como um objeto complexo é construído.
- Permite a construção de objetos em um processo variável, com múltiplas etapas.
- Oculta do cliente a representação interna dos produtos.
- As implementações de produtos podem ser incluídas e removidas porque o cliente só vê uma interface abstrata.

Muitos métodos numéricos são utilizados para simulações dinâmicas. Em sistemas de potência as equações diferenciais são ordinárias do tipo apresentado em (8), onde $x(t)$ é uma função variante no tempo que depende da condição inicial x_0 (CROW, 2003, p. 117).

$$\dot{x}(t) = f(x(t), t) \quad x(t_0) = x_0 \quad (8)$$

Os algoritmos básicos de integração apresentam a solução de t_n até t_{n+1} , com o tamanho do passo de integração h_{n+1} , baseado no cálculo que envolve os valores calculados previamente x_n, x_{n-1}, \dots , e funções $f(x_n, t_n), f(x_{n-1}, t_{n-1}), \dots$.

Para simulação de SEPs, podem-se utilizar métodos de passo único, que só utilizam informações do passo imediatamente anterior, ou passos múltiplos, que utilizam informações de mais de um passo anterior. Em sistemas de ordem muito elevada, os métodos de Runge Kutta, de passo simples e o trapezoidal, de passos múltiplos, são muito utilizados (ARRILLAGA; WATSON, 2001; CROW, 2003), e por isso foram implementados neste trabalho.

Em Sena et al. (2005) é apresentado o modelo de abstração e implementação de sistemas de potência. No Diagrama 17, a classe *TIntegrador* representa a implementação da classe concreta que comandará a integração numérica de todo o sistema, ou seja, será o cliente dos métodos de integração como a classe *Director* mostrada no Diagrama 16. As classes *Sistema* e *Bloco* representam as abstrações que irão implementar as dinâmicas de cada componente físico do sistema.

Além disso, o Diagrama 17 também apresenta a estrutura de aplicação do padrão *Builder* na BCSSD, com as devidas associações apresentadas no Diagrama 16. A classe abstrata *TIntegradorMétodo* faz o papel da classe *Builder*, e as classes *TRungeKutta4* e *Trapezoidal* são as classes concretas *ConcretBuilder* que implementam os métodos numéricos Runge Kutta de 4ª ordem e trapezoidal, respectivamente.

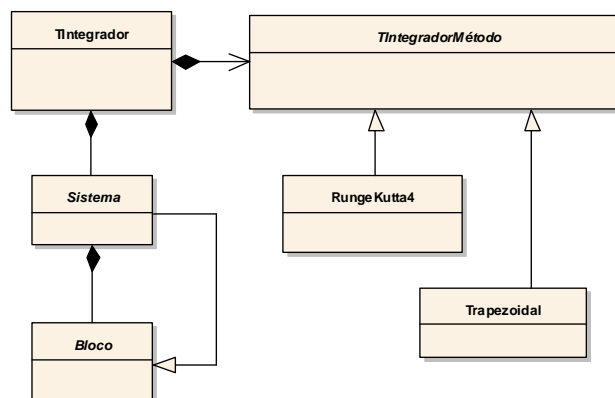


Diagrama 17: Arquitetura do Projeto *Builder* na implementação dos métodos numéricos para simulação de sistemas de potência.

As classes que implementam o padrão *Builder* são mais detalhadas no Diagrama 18, com destaque para o método abstrato *ToIntegrate(...)* da classe *TIntegradorMétodo*. Este método é efetivamente implementado nas classes concretas, utilizando uma interface padrão para qualquer método numérico.

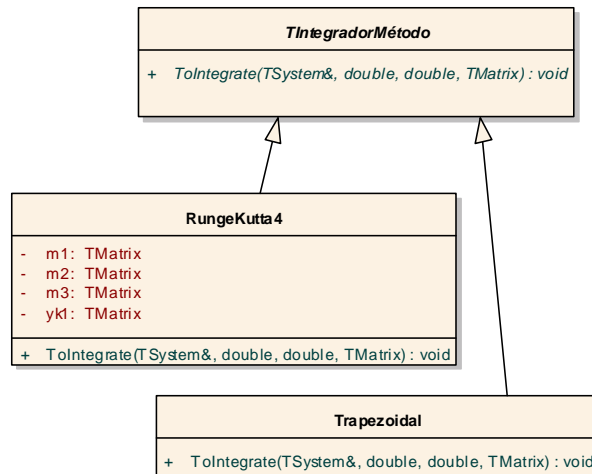


Diagrama 18: Detalhes das classes que implementam os métodos numéricos para simulação.

A classe *TIntegrador* é a real responsável por controlar a integração numérica de todo o sistema. Sua instância ordena a cada objeto, os quais representam a dinâmica de um elemento físico do sistema, o momento de efetuar os cálculos que estão nele encapsulados. Um gerador, por exemplo, seria implementado nesta proposta como uma instância da classe abstrata *Bloco*, e teria suas equações dinâmicas encapsuladas, independente do método de integração, e a instância da classe *TIntegrador* diz o instante no qual estes cálculos internos serão efetuados.

Os dados da dinâmica de todo o sistema são, assim, efetuados e a cada passo de integração os métodos implementados nas classes concretas dos métodos numéricos são invocados. É ela quem faz a interface entre as classes que representam a dinâmica do sistema de potência com o método numérico utilizado para a simulação dinâmica.

Os métodos de Runge-Kutta são os mais usados para os problemas de valor inicial devido sua simplicidade computacional, alta precisão e versatilidade nas aplicações (CUNHA, 2003, p. 185). Seu algoritmo básico de quarta ordem, apresentado no sistema de equações (9), é implementado no método *ToIntegrate(...)* da classe *RungeKutta4*, do framework desenvolvido neste trabalho.

$$\begin{cases} m_0 = h.f(x_k, u_k) \\ m_1 = h.f(x_k + \frac{h}{2}, u_k + \frac{m_0}{2}) \\ m_2 = h.f(x_k + \frac{h}{2}, u_k + \frac{m_1}{2}) \\ m_3 = h.f(x_{k+1}, u_k + m_2) \\ u_{k+1} = u_k + \frac{h}{6}(m_0 + 2m_1 + 2m_2 + m_3) \end{cases} \quad (9)$$

A função f é a função que se deseja integrar, x_k e y_k representam as variáveis de estado de entrada e saída, respectivamente, e h representa o passo de integração. Todos esses valores são passados como parâmetros para o método.

O algoritmo para a implementação do método Trapezoidal é apresentado em (10) (CROW, 2003, p. 123)

$$x_{k+1} = x_k + \frac{1}{2}h[f(x_k, t_k) + f(x_{k+1}, t_{k+1})] \quad (10)$$

Após a execução de um desses algoritmos, selecionados através de *TIntegrador*, são atualizadas as variáveis de estado e as saídas de cada bloco nesse passo de integração.

As saídas de cada bloco são calculadas através dos métodos implementados nas classes que representam a dinâmica de cada elemento físico do sistema de potência, cujos códigos ficam encapsulados de forma independente dos métodos numéricos. Assim, o algoritmo é repetido até que seja concluído o tempo total de simulação.

O Diagrama 19 apresenta um exemplo de diagrama de objetos, detalhando como é o relacionamento do objeto *Integrador*, que é instância de *TIntegrador*, e cada objeto instanciado de *Bloco*, como *Generator1* (Gerador 1, que é da primeira etapa da UHE de Tucuruí, por exemplo) e *Generator13* (Gerador 13, que é da segunda etapa, por exemplo), os quais são objetos que representam os Geradores. O Gerador 1 é instância de *Gerador* que, por sua vez, é uma realização da classe abstrata *Bloco*.

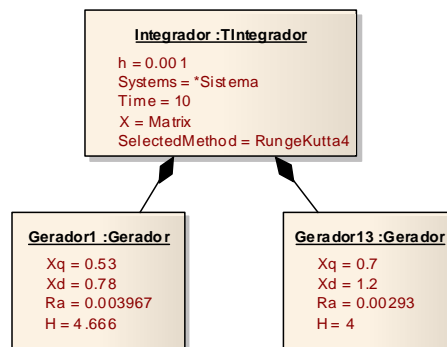


Diagrama 19: Diagrama de objetos representativo para o sistema elétrico de potência.

Como forma de selecionar o método de integração, deve-se especificar a propriedade *SelectedMethod* do objeto *Integrador*. No exemplo está selecionado o método Runge-Kutta de quarta ordem.

3.3.4 Facade

O padrão de projeto *Facade* possui escopo nos objetos, não nas classes, e é estrutural. Ou seja, permite a organização das classes ou objetos em estruturas maiores, facilitando o acesso de classes clientes a uma diversidade de funcionalidades em diferentes classes.

“O padrão *Facade* fornece uma interface unificada para um conjunto de interfaces em um subsistema. A *fachada* define uma interface de nível mais alto que facilita a utilização do subsistema” (FREEMAN; FREEMAN, 2007, p. 221, grifo nosso). O Diagrama 20 apresenta o problema da falta de uso do *Facade*, em que se tem classes clientes que utilizam serviços de diversas classes. Dependendo da quantidade de serviços e clientes disponíveis, pode-se tornar muito trabalhoso para o programador realizar manutenções ou até expandir sistemas baseados nesta arquitetura.

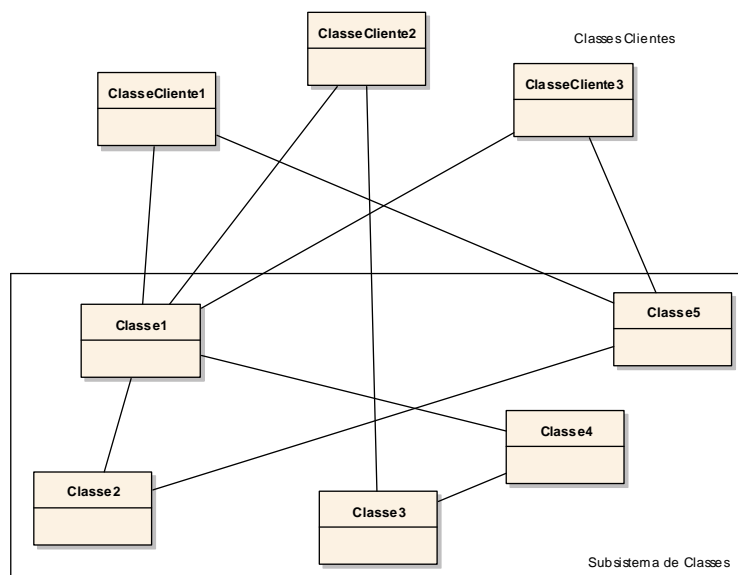


Diagrama 20: Diagrama de Classes enfocando o problema da falta do Padrão *Facade*.

Fonte: Adaptado de Gamma et al. (2007, p. 185).

O Diagrama 21 já apresenta o enfoque com a presença de uma classe de “fachada” para servir de interface entre os clientes e as diversas classes que fornecem os serviços, abstraindo das classes clientes os diversos formatos de implementação de suas

funcionalidades. A classe *Facade* sabe quais as responsabilidades de cada subsistema, e é responsável por enviar as requisições de cada cliente ao respectivo subsistema objetos. As classes do subsistema são as que implementam cada funcionalidade, não tendo conhecimento da fachada, não guardando nenhuma referência dela ou de algum cliente (GAMMA et al., 2007).

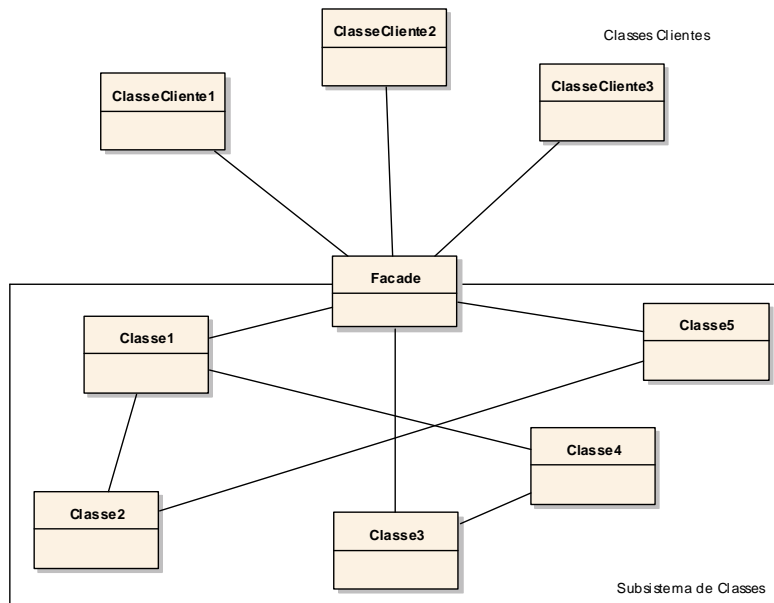


Diagrama 21: Diagrama de Classes para implementação do Padrão de Projeto *Facade*.

Fonte: Adaptado de Gamma et al. (2007, p. 185).

Para a realização da simulação dinâmica de um SEP, são realizadas as atividades apresentadas no Diagrama 22. A simulação é iniciada com o cálculo do fluxo de carga para definir a condição inicial de operação do sistema. Com a convergência dos cálculos, são executados os cálculos das condições iniciais de todos os blocos dinâmicos, ou no caso do *DynaPower*, de todas as classes concretas que implementam a dinâmica do sistema. A cada passo de integração, os métodos de cálculos de derivadas são executados e, em seguida, são calculadas as integrais de cada variável de estado com os métodos de integração numérica implementados. Por fim, são calculadas as saídas de cada bloco. Este processo se repete até que seja alcançado o tempo total da simulação.

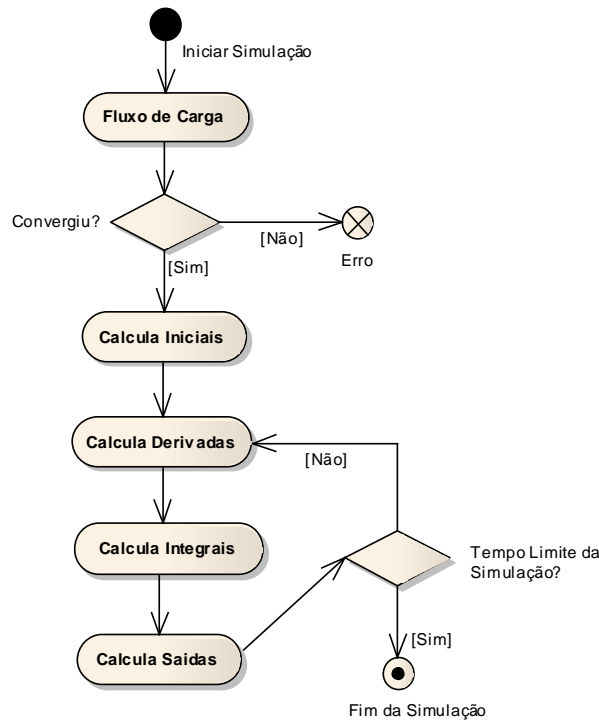


Diagrama 22: Diagrama de Atividades para Simulação de Sistemas Elétricos de Potência.

A classe que gerencia a integração numérica foi apresentada no padrão *Builder*, e a *fachada*, ou a implementação do padrão *Facade*, está justamente na abstração de diversos subsistemas de um SEP que pode ser modelado e implementado como um bloco único. Em Sena et al. (2006) é apresentada a estrutura do Diagrama 23, cuja classe *TIntegrador* é a classe cliente que é responsável por gerenciar a simulação e realizar os cálculos a cada passo de integração. É ela quem manda a ordem para a classe *Sistema*, que além de ser uma especialização da classe *Bloco*, serve de *fachada* para todos os blocos dinâmicos instanciados como objetos do SEP, e assim todos os cálculos são realizados, considerando cada dinâmica implementada.

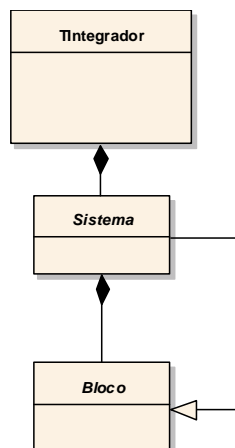


Diagrama 23: Diagrama de Classes para implementação do padrão de projeto *Facade*.

Fonte: Adaptado de Sena et al. (2006, p. 3).

Pode-se observar que diversas implementações da classe *Sistema* podem coexistir, contendo diversas implementações da classe *Bloco*. Um detalhe importante é que as implementações dos métodos de cálculo de derivadas e de saídas são as mesmas para todas as classes concretas que implementam blocos dinâmicos. Porém, o cálculo das condições iniciais, não, pois cada bloco dinâmico necessita de diferentes atributos do sistema para suas implementações, fazendo com que o uso do padrão *Facade* também facilite a manutenção e criação de novas classes concretas, ou novos blocos dinâmicos.

Outra vantagem do padrão *Facade* é a possibilidade de acoplar componentes de interfaces gráficas, visualizando em diversos níveis os sistemas e subsistemas de um determinado cenário, similar a como é feito no ambiente do Simulink (MATLAB, 2008).

4 DESENVOLVIMENTO DO FRAMEWORK PARA SIMULAÇÃO DINÂMICA COM MÚLTIPLAS UNIDADES GERADORAS NA UHE DE TUCURUÍ

4.1 CONSIDERAÇÕES SOBRE FRAMEWORKS PARA SIMULAÇÃO DINÂMICA DE SISTEMAS ELÉTRICOS DE POTÊNCIA

Estudos relacionados à simulação de SEPs com uso do paradigma OO não são recentes. Na década de 1990 houve um esforço da comunidade científica na exploração do paradigma OO e na avaliação de seu potencial como estratégia para atacar este tipo de problema.

Zhou (1996) explora a implementação de fluxos de carga em CC e em CA utilizando a linguagem C++, apresentando uma abordagem simples, porém funcional, da modelagem de um SEP, tendo classes concretas implementadas como sendo barras, e diversas interfaces (ou *templates* no C++) para representação de redes. Definindo, também, as implementações de cada método de cálculo e a estrutura de classes para manipulação de matrizes esparsas.

Zhu e Jossman (1999) apresentam já algumas definições de padrões de projeto para a modelagem de SEPs, definindo um arcabouço de classes para modelagem de sistemas pequenos, também utilizando classes concretas para representar cada elemento do sistema, como barras, transformadores, linhas de transmissão, cargas e etc, porém, fazendo uso de técnicas de redução e simplificação antes da implementação OO.

Frameworks não implementados, ou arcabouços de classes surgiram como propostas para a modelagem OO de SEP durante o final da década de 1990 e início da década de 2000. Palma et al. (2003) já apresenta um *framework* bastante complexo que modela não apenas a dinâmica de um SEP, como também outros elementos como fatores econômicos e ambientais tais como transporte de fontes primárias de energia (no caso de termelétricas), variação do volume do lago (no caso de hidrelétricas), entre outros. Faz uso de representação unifilar para modelagem do sistema, dividindo seus elementos em: de uma ligação (geradores, cargas, etc) e de dois pólos (transformadores, linhas de transmissão, etc). Embora apresente um objetivo bastante audacioso de abordar questões tão diferentes, ainda é apenas uma proposta *framework* que não foi implementada, mas que já apresenta um grande potencial que o uso da OO pode

alcançar.

Manzoni (2005) já apresenta um *framework* implementado para modelar sistemas físicos e com ferramentas computacionais para execução de cálculos de fluxos de carga e da dinâmica de sistemas com uso inclusive de vetores e matrizes esparsas. Faz uso das técnicas de modelagem e padrões de projeto OO simples, baseados apenas em classes concretas, especializações e agregações. Além disso, a arquitetura OO construída modela toda a estrutura física do sistema, deixando as demais aplicações para implementações externas, como as realizadas por Machado Júnior (2005), e em seu próprio trabalho que enfoca a dinâmica de SEP e a estrutura computacional utilizada para implementá-la.

O meio da década de 2000 já é marcado por implementações de *frameworks* complexos como este que serve para modelar e implementar a dinâmica de sistemas físicos, não necessariamente de sistemas de potência, e simular sua dinâmica com diferentes técnicas de integração numérica.

Selvan e Swarup (2006) também apresentam seu *framework* com suas próprias inovações para estudos de sistemas trifásicos desbalanceados, muito comuns em sistemas de distribuição de energia elétrica. Os equipamentos já não são modelados a partir de modelos unifilares, e sim de modelos trifásicos, envolvendo não apenas uma matemática mais detalhada, como também uma modelagem OO mais complexa.

O grupo de sistemas de controle, ligado à FEE e ao PPGEE da UFPA, tem considerável *know-how* no estudo de modelagem, controle e simulação da dinâmica de SEPs, acumulado após muitos anos pesquisas desenvolvidas pelos cientistas do grupo. Em Barreiros (1989) são tratados métodos de controle por posicionamento de pólos aplicados a ESPS, e em Barreiros (1995) os métodos de controle adaptativo aplicados a ESPs, utilizando técnicas de controle preditivo generalizado. Nesses trabalhos, foram desenvolvidos sistemas para simulação da dinâmica de SEPs, do tipo máquina-barras infinita, desenvolvido em linguagem FORTRAN.

Ferreira (1998) elabora um programa de simulação de sistema máquina-barras infinita, em ambiente Matlab/Simulink, utilizando blocos dinâmicos e o agrupamento de partes afins em subsistemas, com ESPs projetados usando controle *fuzzy* baseado em *Look-up Table*. Por outro lado, Costa Júnior (1999) desenvolve um *software* para simulação de sistema máquina-barras infinita, usando as técnicas de *s-functions* e de

máscaras para a inserção de dados de simulação, no projeto de ESPs usando redes de modelos e controladores locais com supervisor *fuzzy*. Barra Junior (2001) desenvolve um sistema multi-máquinas, baseado em *s-functions*, para simulação do modelo de três máquinas descrito em Anderson e Fouad (1977), no qual testou o projeto de controladores amortecedores usando técnicas de controle *neuro-fuzzy*.

Somando-se a estes *know-hows*, Ferreira (2005) desenvolve uma versão mais completa e estendida do sistema desenvolvido por Barra Junior (2001), generalizando para n máquinas síncronas, com a interface gráfica do Matlab/Simulink, usando *s-functions*. Sendo este o trabalho que serviu de base para a construção do aplicativo de referência para comparação e testes individuais e integrados dos modelos das UGHs da UHE de Tucuruí apresentados neste trabalho, cujos resultados de diversos ensaios também são apresentados em Barra Junior (2008b, 2009).

Sena (2006), também integrante do grupo de sistemas de controle da UFPA, apresenta uma proposta para um *framework* de SEP, desenvolvido em linguagem C++, agregando um complexo modelo OO, com uso de diversos padrões de projeto na construção da BCSSD. Faz uso de recursos matemáticos específicos para processadores baseados na arquitetura Intel com objetivo de aumentar a velocidade em que os cálculos são realizados, e também com o propósito de simular qualquer dinâmica, independente de ser ou não um SEP. O trabalho de Sena (2006), de modelagem OO, teve início em 1996 (SENA, 1998), orientado por professores do grupo de sistemas de controle da UFPA.

O presente trabalho apresenta-se, então, como parte integrante deste contexto de estado da técnica e *know-how* do grupo de sistemas de controle da UFPA que, dentro do projeto de P&D com a Eletronorte, aplicou técnicas do paradigma da orientação a objetos no desenvolvimento do *DynaPower*. Um simulador da dinâmica interna detalhada das UGHs da UHE de Tucuruí, originando, inclusive, outras produções, como em Di Paolo (2009a, 2009b) e Barra Junior (2008a, 2008b, 2009) que apresentam algumas das aplicações de padrões de projeto OO em SEPs multi-máquinas e diversos detalhes do processo de desenvolvimento de *software* e testes de aplicação, comparações de resultados, entre outras análises.

Diversas aplicações relacionadas a estes *frameworks* também vêm sendo exploradas, como uma aplicação para fluxo de potência ótimo, desenvolvido por Machado Júnior (2005), e *frameworks* para interfaces gráficas deste tipo de simulador,

apresentado por Borges (2006) e Dzafic, Glavic e Tesnjak (2004). Um detalhe importante nos estudos realizados sobre o estado da técnica das interfaces gráficas para estes tipos de aplicações revelou que ainda estão iniciando. Poucas são as propostas maduras de interfaces gráficas que possam explorar todos os recursos de um simulador para SEP de grande porte, fazendo com que os usuários deste tipo de *software* ainda necessitem de um *know-how* bastante elevado em diversas áreas do conhecimento, incluindo as técnicas e tecnologias de implementação do *software*.

Embora o Simulink (MATLAB, 2008) seja uma interface gráfica madura para modelagem e implementação de sistemas dinâmicos, sua aplicação em SEP de grande porte é feita mais em nível de aprendizado, pois para sistemas muito complexos a realização dos cálculos em sua plataforma consome muita memória e muito processamento que chegam a inviabilizar seu uso em estudos com complexidade computacional elevada.

Atualmente, o *software* utilizado pelo ONS para simulação dinâmica do SIN é o ANATEM... (2007), que ainda não possui interface gráfica e também não foi construído utilizando o paradigma da OO, e sim utilizando o paradigma estrutural, com a linguagem Fortran. Vale lembrar que o ANATEM... (2007) executa apenas a simulação dinâmica, sendo o fluxo de carga realizado pelo ANAREDE... (2007), também baseado no paradigma estrutural e em linguagem Fortran.

4.2 MODELAGEM DA USINA HIDRELÉTRICA DE TUCURUÍ

No Capítulo anterior foram apresentados os padrões de projeto OO utilizados na implementação do *DynaPower*. Neste Capítulo serão apresentados os modelos utilizados para representar UGHs da UHE de Tucuruí, bem como o cenário utilizado para simulação em todos os seus detalhes, considerando os padrões de projeto apresentados.

A UHE de Tucuruí é composta por 23 geradores. Em 1984 a UHE entrou em operação com 12 geradores de 350 MW, com máquinas de diferentes fabricantes, conhecidos como “geradores da primeira etapa” ou “geradores da primeira casa de força” (ver Fotografia 2).



Fotografia 2: Máquinas da primeira etapa da construção da UHE de Tucuruí.

E em 2002 começaram a operar mais 11 geradores de 390 MW, também de diferentes fabricantes, conhecidos como “geradores da segunda etapa” ou “geradores da segunda casa de força”, compondo uma carga potencial máxima de 8,49 GW de energia gerada para o SIN (ver Fotografia 3).



Fotografia 3: Máquinas da segunda etapa da construção da UHE de Tucuruí.

O modelo do SEP construído neste trabalho objetivou o estudo da dinâmica de todas as UGHs da UHE de Tucuruí, e seus respectivos controles. O Apêndice A apresenta o diagrama unifilar das UGHs interligadas ao barramento infinito, constituído por uma grande máquina que consome a maior parte da energia gerada. A interface com o barramento infinito é simbolizada por uma linha pontilhada passando pelo ramo 35, sendo a barra 36 a subestação da UHE. As máquinas $G1$ a $G12$ são as máquinas na primeira etapa, e as $G13$ a $G23$ são da segunda etapa.

A base comum utilizada é de 350 MVA de potência e $13,8\text{ kV}$ de tensão. Os ramos 24 a 34 são linhas de transmissão que possuem todas a mesma impedância série de $3,612.10^{-5} + j4,5682.10^{-4}\text{ p.u.}$, e susceptância *shunt* de $3,6.10^{-3}\text{ p.u.}$ Os ramos 1 a 23 são os transformadores das máquinas geradoras que possuem reatância série de $0,165\text{ p.u.}$ O ramo 35 é uma linha de transmissão configurada com os mesmos valores de impedância série e susceptância *shunt* das demais linhas. O transformador do ramo 36 foi configurado com uma reatância muito pequena de $1,65.10^{-7}\text{ p.u.}$, eletricamente muito perto da barra 24. A barra 37 é barramento infinito, que também é a referência deste sistema. Foram ainda penduradas pequenas cargas de 1 MW nas barras 25 a 35, e de 10 MW na barra 36, simbolizando o consumo interno da UHE. Na barra 37 foi pendurada uma carga de 2800 MW .

Nessas condições, a matriz admitância Y para as configurações apresentadas desta rede, a qual é gerada e utilizada pelo *DynaPower*, é quadrada, simétrica e de ordem 37 e com esparsidade igual a 99,92%.

No decorrer deste Capítulo serão apresentados todos os modelos e parâmetros utilizados para a representação das UGHs. As modelagens OO apresentadas serão baseadas nos padrões de projeto apresentados no Capítulo 3. Neste sentido, implementar uma nova dinâmica de um controle ou atuador no sistema resume-se a implementar uma nova classe concreta, cujas funcionalidades principais são herdadas de uma classe abstrata. Uma interface também poderia ser usada em vez de uma classe abstrata, porém, devido a implementação ser em C++, optou-se por esta estratégia.

Como as classes abstratas e a estrutura principal da implementação também já foram apresentadas no decorrer do Capítulo 3, os modelos OO da UHE de Tucuruí serão apresentados classe a classe, e serão consolidados no final deste Capítulo.

4.3 GERADORES

O modelo das máquinas síncronas utilizado já foi apresentado no Item 2.4, referente ao Modelo 4. A Fotografia 4 apresenta uma parte do rotor da Máquina 8, durante ensaios realizados na UGH em fevereiro de 2009.



Fotografia 4: Rotor da UGH 8.

Os parâmetros do modelo das máquinas da primeira etapa são apresentados no Anexo A (baseado em GENERAL ELECTRIC, [19--]; VILLARES EQUIPAMENTOS, [19--]), e os parâmetros das máquinas da segunda etapa são apresentados no Anexo B (ELETRONORTE, ALSTOM POWER, [2000]).

É importante ressaltar que em Operador Nacional do Sistema Elétrico (2008a), que é a base de dados utilizada pelo ONS como referência do SIN, a UHE de Tucuruí é representada como um agregado de 5 geradores equivalentes a toda a usina. Neste estudo, como já foi apresentado no Apêndice A, as 23 máquinas geradoras foram implementadas. Em seguida serão apresentados os controles e atuadores presentes nas UGHs.

4.4 REGULADORES AUTOMÁTICOS DE TENSÃO

Os RATs foram definidos no Item 2.6. Neste item serão abordados diretamente os modelos implementados no *DynaPower* e suas referências para implementação dentro dos padrões de projeto OO adotados. A Fotografia 5 apresenta o armário em que se localiza o RAT e seus controles e atuadores da UGH 8, em ensaios realizados em

confundido com a complexidade envolvida na construção dos equipamentos. O Diagrama 25 apresenta a malha principal do RAT da segunda etapa, e os Diagramas 26 a 31 todos os demais sinais detalhados. Os parâmetros encontram-se no Anexo D.

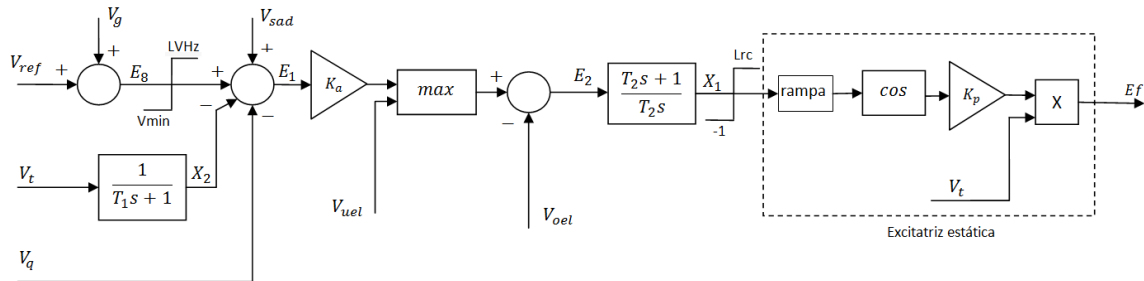


Diagrama 25: Malha principal do RAT da segunda etapa.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

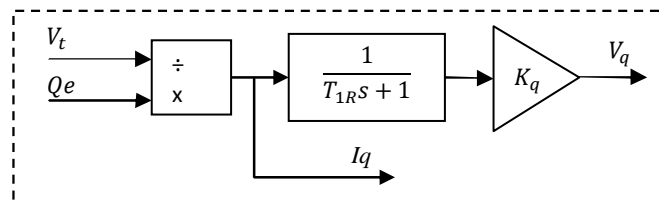


Diagrama 26: Compensação de corrente reativa.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

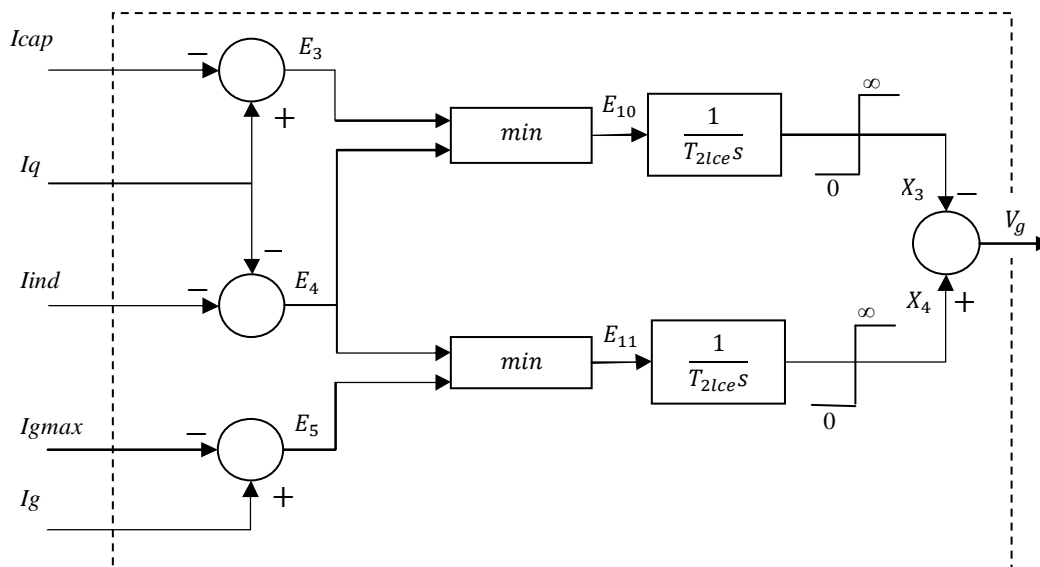


Diagrama 27: Limitador de corrente de armadura.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

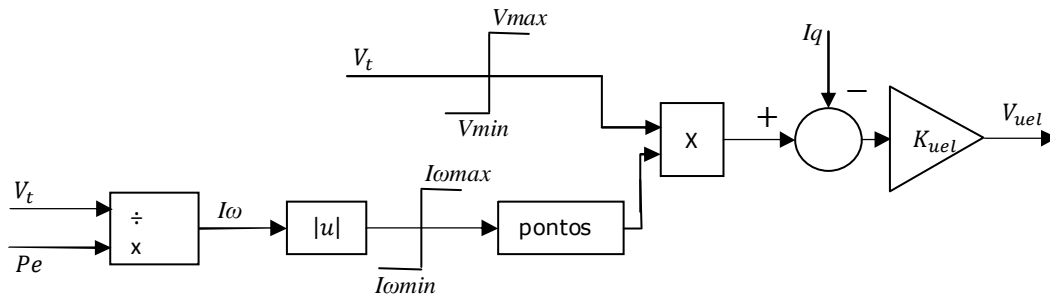


Diagrama 28: Limitador de subexcitação.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

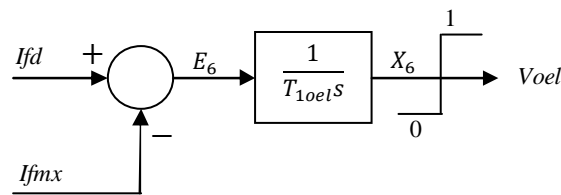


Diagrama 29: Limitador de sobreexcitação.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

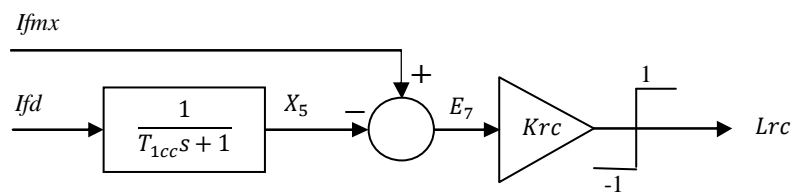


Diagrama 30: Limitador de corrente de excitação.

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

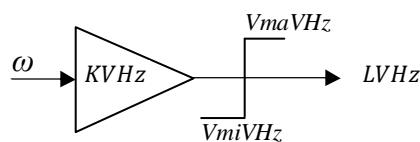


Diagrama 31: Limitador de enlace de fluxo (Volt/Hz).

Fonte: Baseado em Eletronorte (2001) e Operador Nacional do Sistema Elétrico (2004, 2008a).

4.4.2 Modelagem em Espaços de Estados

Um passo necessário para a implementação no *DynaPower* é converter os modelos dinâmicos em sua representação equivalente em espaços de estados. Para o RAT da primeira etapa, escolheram-se as variáveis de estado X_1 e X_2 , conforme apresenta o Diagrama 24, definindo-se as variáveis de estado que representam a dinâmica deste sistema em (11).

$$\begin{cases} \dot{x}_1 = \frac{1}{T_a}(Ka.Iq - x_1) \\ \dot{x}_2 = \frac{1}{V_{Amin}}(V_{Amax}.e_1 - x_2) \end{cases} \quad (11)$$

$$\text{Sendo } Iq = Q / V_T \text{ e } e_1 = V_{ref} + x_1 + V_{sad}.K_{sad} - V_T.$$

Como além das equações dinâmicas também são necessárias as condições iniciais e as de saídas, tem-se em (12) as condições iniciais, e em (13) a única equação de saída.

$$\begin{cases} x_1 = \frac{Q}{V_T}Ka \\ x_2 = 0 \\ V_{ref} = V_T - x_1 \end{cases} \quad (12)$$

$$E_f = e_2 = x_2K_i + E_{f0} + e_1T_{se}K_i \quad (13)$$

Para o RAT da segunda etapa, por ter sua estrutura dinâmica mais detalhada, o Apêndice B apresenta o desenvolvimento dos cálculos que culminaram nas variáveis de estado apresentadas em (14).

$$\begin{cases} \dot{x}_1 = \frac{e_2}{T_2} \\ \dot{x}_2 = \frac{1}{T_1}(V_T - x_2) \\ \dot{x}_3 = \frac{e_{10}}{T_1I_{ce}} \\ \dot{x}_4 = \frac{e_{11}}{T_2I_{ce}} \\ \dot{x}_5 = \frac{1}{T_{1cc}}(I_{fd} - x_5) \\ \dot{V}_{oel} = \frac{e_6}{T_{1oel}} \\ \dot{V}_q = \frac{1}{T_{1R}}(K_qI_q - V_q) \end{cases} \quad (14)$$

Em (15) são apresentadas as condições iniciais, e em (16) a equação de saída.

$$\begin{cases} L_{rc} = 1 \\ LVHZ = \omega K_{VHZ} \\ e_9 = 1 - \frac{2}{\pi} \arccos\left(\frac{E_f}{V_T K_p}\right) \\ x_1 = e_9 \\ x_2 = V_T \\ x_3 = -I_{cap} \\ x_4 = -I_{ind} \\ x_5 = I_{fd} \\ V_{oel} = 0 \\ V_q = \frac{Q_e K_q}{V_T} \\ V_{ref} = V_T \end{cases} \quad (15)$$

$$E_f = V_T K_p \cos(\text{rampa}(e_9)) \quad (16)$$

4.4.3 Modelagem Orientada a Objetos

O Diagrama 32 apresenta o diagrama de classes dos RATs da primeira (*TRATTucuruí1*) e da segunda etapa (*TRATTucuruí2*), de acordo com os padrões de projeto apresentados. A classe *ReguladorTensão* é a classe mãe, e é uma especialização da classe *Bloco*, tendo também um atributo *ativo*, que define se deve ou não considerar o RAT durante a simulação. No caso de não considerar, já está definido, também, que não se deve considerar o ESP que a instância do RAT estiver associada. Nota-se, também, na classe *TRATTucuruí2* a implementação de dois métodos que representam as funções *rampa* e *pontos*, com o mesmo nome do diagrama de blocos.

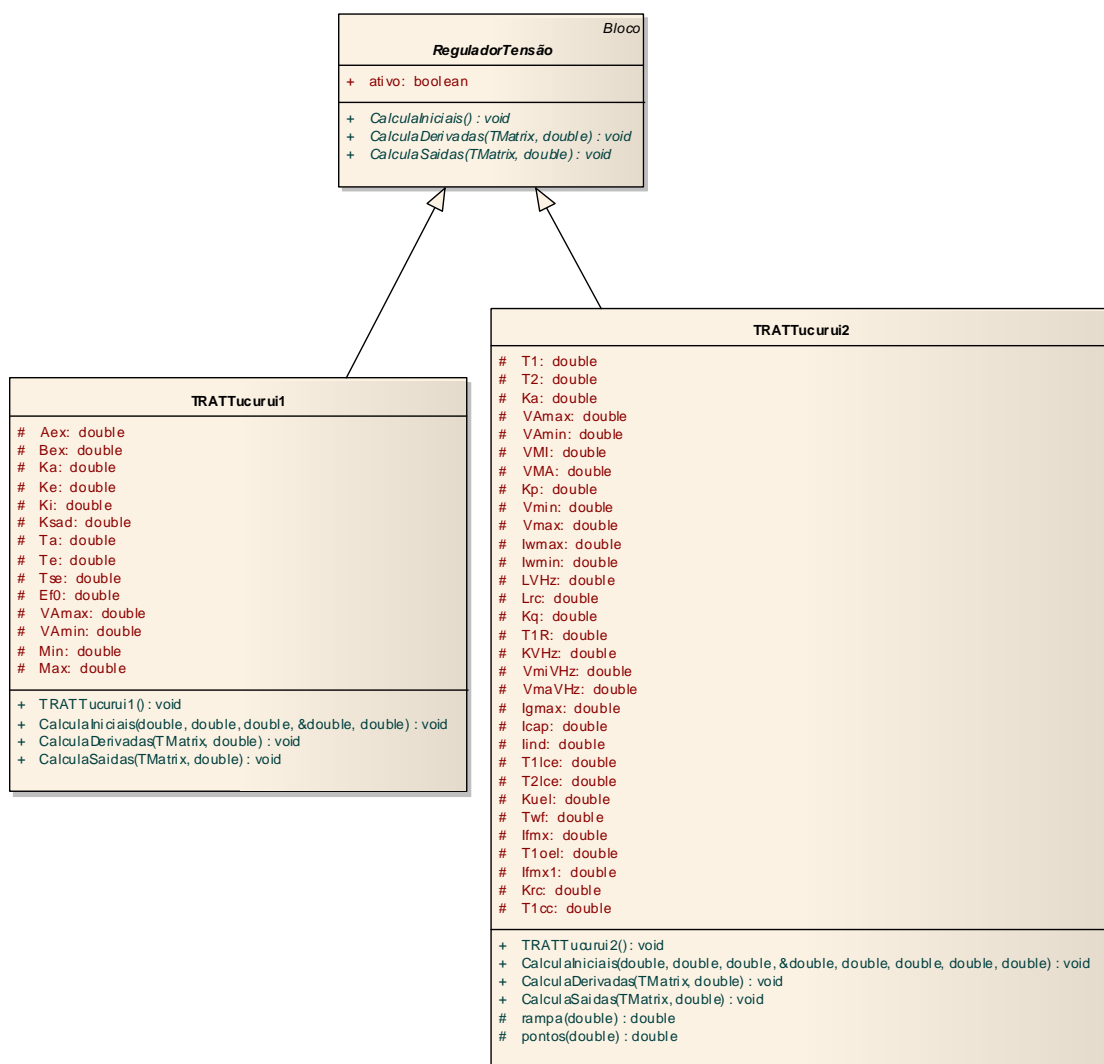


Diagrama 32: Diagrama de classes dos RATs da UHE de Tucuruí.

4.5 ESTABILIZADORES DE SISTEMAS DE POTÊNCIA

Os ESPs foram definidos no Item 2.7. Neste item serão abordados diretamente os modelos implementados no *DynaPower* e suas referências para implementação dentro dos padrões de projeto OO adotados. A Fotografia 6 apresenta a gaveta em que se localiza o controle do ESP na UGH 8, em ensaios realizados em fevereiro de 2009.



Fotografia 6: Ensaio sendo realizado no ESP da UGH 8.

4.5.1 Modelagem em Diagrama de Blocos

O modelo do ESP da primeira etapa foi baseado em Operador Nacional do Sistema Elétrico (2008a), cujo Diagrama 33 apresenta o seu diagrama de blocos. Os parâmetros deste modelo encontram-se no Anexo E.

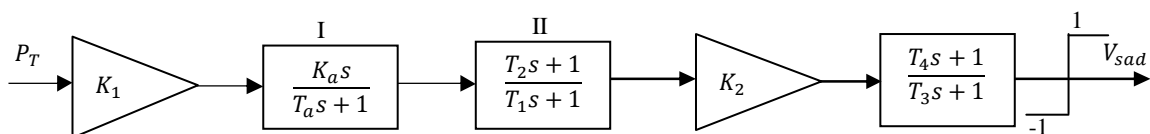


Diagrama 33: Diagrama de Blocos do ESP da primeira etapa.

Fonte: Baseado em Operador Nacional do Sistema Elétrico (2008a).

O modelo do ESP da segunda etapa foi baseado em Operador Nacional do Sistema Elétrico (2004, 2008a), e no Diagrama 34 é expresso sua representação em diagramas de blocos. Os parâmetros deste modelo encontram-se no Anexo F.

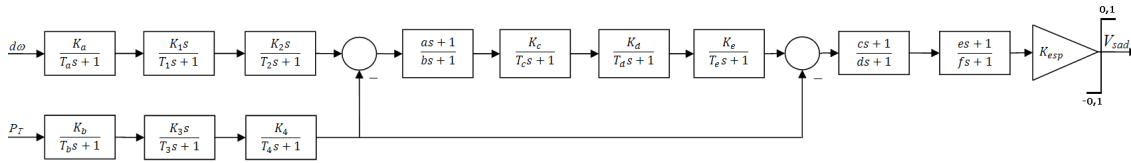


Diagrama 34: Diagrama de Blocos do ESP da segunda etapa.

Fonte: Baseado em Operador Nacional do Sistema Elétrico (2004, 2008a).

4.5.2 Modelagem em Espaços de Estados

As equações de estado com a dinâmica do ESP da primeira etapa estão apresentadas em (17), e foram calculadas conforme apresenta o Apêndice C, considerando a eliminação dos blocos com zeros³.

$$\begin{cases} \dot{x}_1 = \left(\frac{1}{T_a}\right) \left\{ K_1 P_T \left[\left(\frac{-K_a}{T_a}\right) - x_1 \right] \right\} \\ \dot{x}_2 = \left(\frac{1}{T_1}\right) [(1 - T_2/T_1)E_1 - x_2] \\ \dot{x}_3 = \left(\frac{1}{T_3}\right) [(1 - T_4/T_3)E_2 K_2 - x_3] \end{cases} \quad (17)$$

Em (18) são apresentadas as condições iniciais e em (19) o cálculo da saída V_{sad} do ESP da primeira etapa, que será a entrada para o RAT, considerando que V_{sad} tem sua amplitude limitada em -1 e $+1$.

$$\begin{cases} x_1 = K_1 P_T \left(\frac{-K_a}{T_a}\right) \\ x_2 = 0 \\ x_3 = 0 \end{cases} \quad (18)$$

$$\begin{cases} e_1 = K_1 P_T \left(\frac{K_a}{T_a}\right) + x_1 \\ e_2 = e_1 \left(\frac{T_2}{T_1}\right) + x_2 \\ V_{sad} = e_2 K_2 \left(\frac{T_4}{T_3}\right) + x_3 \end{cases} \quad (19)$$

Para o ESP da segunda etapa, o desenvolvimento dos cálculos para obtenção das equações de estado é apresentado no Apêndice D, cujo resultado é apresentado em (20).

³ A eliminação de blocos com funções de transferências com zeros precisa ser efetuada para que, no cálculo das equações dinâmicas, a variação da saída não dependa de variações do sinal de entrada neste bloco. Apenas desta forma é possível sua implementação computacional.

$$\left\{ \begin{array}{l} \dot{x}_1 = \left(\frac{1}{T_a}\right) [d\omega K_a - x_1] \\ \dot{x}_2 = \left(\frac{1}{T_1}\right) \left[x_1 \left(-\frac{K_1}{T_1}\right) - x_2\right] \\ \dot{x}_3 = \left(\frac{1}{T_2}\right) [e_1 (-K_2/T_2) - x_3] \\ \dot{x}_4 = \left(\frac{1}{T_b}\right) [P_T K_b - x_4] \\ \dot{x}_5 = \left(\frac{1}{T_3}\right) [x_4 (-K_3/T_3) - x_5] \\ \dot{x}_6 = \left(\frac{1}{T_4}\right) [e_3 K_4 - x_6] \\ \dot{x}_7 = \left(\frac{1}{b}\right) [e_2 (1 - a/b) - x_7] \\ \dot{x}_8 = \left(\frac{1}{T_c}\right) [e_4 K_c - x_8] \\ \dot{x}_9 = \left(\frac{1}{T_d}\right) [x_8 K_d - x_9] \\ \dot{x}_{10} = \left(\frac{1}{T_e}\right) [x_9 K_e - x_{10}] \\ \dot{x}_{11} = \left(\frac{1}{d}\right) \left[e_5 \left(1 - \frac{c}{d}\right) - x_{11}\right] \\ \dot{x}_{12} = \left(\frac{1}{f}\right) \left[e_6 \left(1 - \frac{e}{f}\right) - x_{12}\right] \end{array} \right. \quad (20)$$

As condições iniciais são apresentadas em (21), e a saída V_{sad} em (22), considerando que sua amplitude é limitada entre os valores $-0,1$ e $+0,1$.

$$\left\{ \begin{array}{l} \Delta\omega = \omega - \omega_0 \\ e_1 = x_1 \frac{K_1}{T_1} \\ e_3 = e_1 \frac{K_2}{T_2} + x_3 + x_6 \\ e_3 = x_4 \frac{K_3}{T_3} + x_5 \\ e_4 = x_2 \frac{a}{b} + x_7 \\ e_5 = x_{10} - x_6 \\ e_6 = e_5 \frac{c}{d} + x_{11} \\ x_1 = \Delta\omega K_a \\ x_2 = x_1 \frac{-K_3}{T_3} \\ x_3 = e_1 \frac{-K_2}{T_2} \\ x_4 = P_T K_b \\ x_5 = x_4 \frac{-K_3}{T_3} \\ x_6 = e_3 K_4 \\ x_7 = e_2 \left(1 - \frac{a}{b}\right) \\ x_8 = e_4 K_c \\ x_9 = x_8 K_d \\ x_{10} = x_9 K_e \\ x_{11} = e_5 \left(1 - \frac{c}{d}\right) \\ x_{12} = e_6 \left(1 - \frac{e}{f}\right) \end{array} \right. \quad (21)$$

$$\left\{ \begin{array}{l} e_7 = e_6 \frac{e}{f} + x_{12} \\ V_{sad} = e_7 K_{esp} \end{array} \right. \quad (22)$$

4.5.3 Modelagem Orientada a Objetos

O Diagrama 35 apresenta o diagrama de classes dos ESPs da primeira etapa (*TESPTucuruí1*) e da segunda etapa (*TESPTucuruí2*), de acordo com os padrões de projeto apresentados. A classe *ESP* é a classe mãe, e é uma especialização da classe *Bloco*, tendo também um atributo *ativo*, que define se deve ou não considerar o ESP durante a simulação.

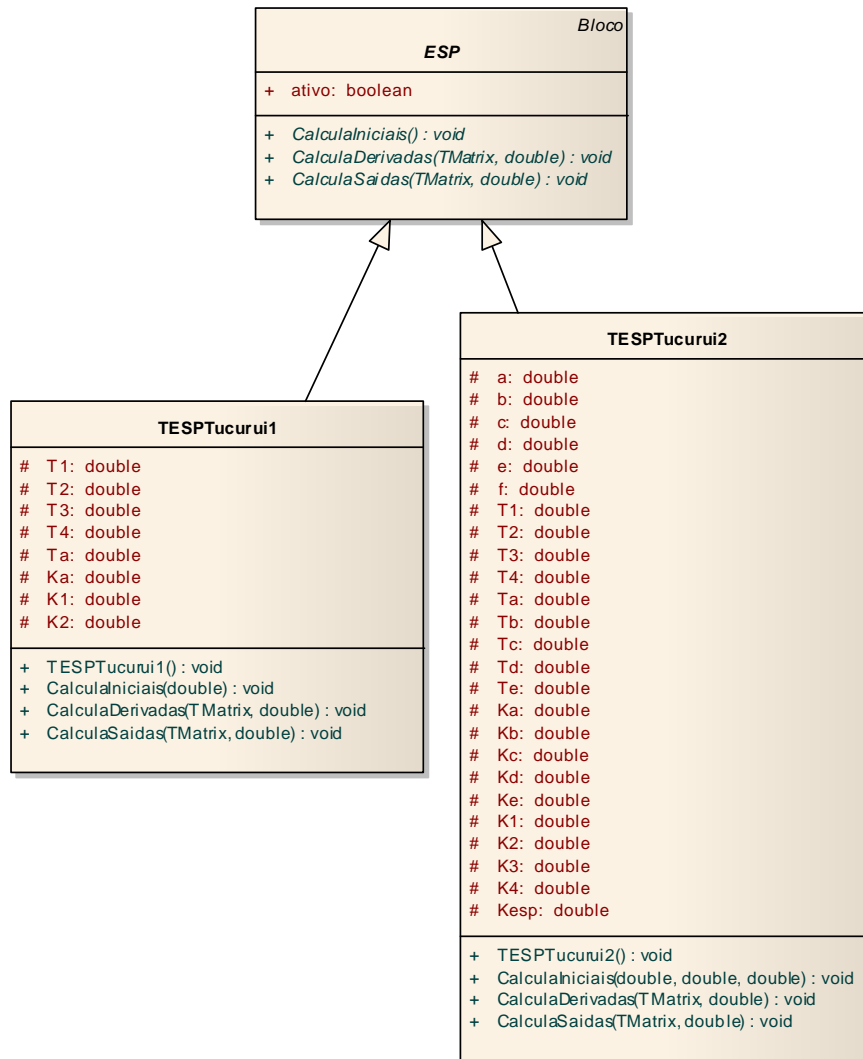
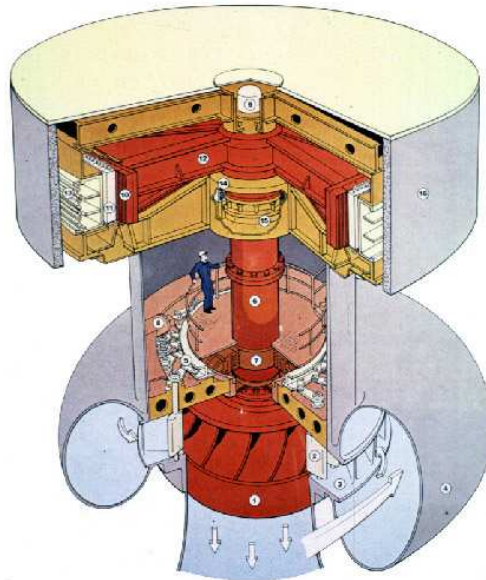


Diagrama 35: Diagrama de classes dos ESPs da UHE de Tucuruí.

4.6 TURBINAS

As turbinas foram definidas no Item 2.9. Neste item serão abordados diretamente os modelos implementados no *DynaPower* e suas referências para implementação dentro dos padrões de projeto OO adotados. O Desenho 1 apresenta uma

ilustração de turbina hidráulica Francis, do tipo das instaladas na UHE de Tucuruí, em que a água entra em um canal em espiral, fazendo movimentar as paletas e consequentemente gerando o torque no rotor. Nota-se no desenho que a posição da pessoa ilustrada é similar à apresentada na Fotografia 4.



Desenho 1: Turbina Francis, similar às instaladas na UHE de Tucuruí.

Fonte: Turbina..., 2009.

4.6.1 Modelagem em Diagrama de Blocos

Os modelos utilizados para implementação no *DynaPower* foram os mesmos modelos não-lineares apresentados no Item 2.9, nos Diagramas 7 e 8, os quais também servem de referência para os estudos do ONS. Ambos os modelos foram implementados para os estudos realizados neste trabalho, porém, os parâmetros apresentados em Operador Nacional do Sistema Elétrico (2003a) do modelo do Diagrama 7, não parecem estar ajustados para o comportamento esperado pelas máquinas da UHE de Tucuruí. Por isso, optou-se por manter o modelo do Diagrama 8 na implementação do cenário da UHE.

Os parâmetros dos modelos utilizados para as turbinas da primeira etapa são apresentados no Anexo G, baseados em Operador Nacional do Sistema Elétrico (2003a, 2008a), e para as turbinas da segunda etapa os parâmetros são apresentados no Anexo H, baseados em Operador Nacional do Sistema Elétrico (2006b, 2008a), considerando que os parâmetros do Diagrama 8 estão mais ajustados para o comportamento dinâmico esperado.

4.6.2 Modelagem em Espaços de Estados

Ambos os modelos não-lineares de turbina apresentados são sistemas de primeira ordem, portando, possuindo apenas uma equação de estado. Levando-se em consideração o modelo do Diagrama 7, a equação de estado é apresentada (23).

$$\begin{cases} G = g \cdot A_t \\ H = \left(\frac{U}{G}\right)^2 \\ \dot{U} = \frac{H_0 - H}{T_W} \end{cases} \quad (23)$$

Ainda para o Diagrama 7, em (24) são apresentadas as condições iniciais e em (25) o cálculo da saída.

$$\begin{cases} U = \frac{P_m}{H_0} + U_{nl} \\ G = \frac{U}{\sqrt{H_0}} \\ g = P_{gv} = \frac{G}{A_t} \end{cases} \quad (24)$$

$$\begin{cases} G = g \cdot A_t \\ \Delta\omega = \omega - \omega_0 \\ H = \left(\frac{U}{G}\right)^2 \\ P_m = H(U - U_{nl}) - D \cdot \Delta\omega \end{cases} \quad (25)$$

Para o modelo referente ao Diagrama 8, a equação de estado é apresentada em (26), e como se pode notar, tendo um comportamento dinâmico similar ao modelo do Diagrama 7.

$$\begin{cases} G = g \cdot P_b \\ H = \left(\frac{U}{G}\right)^2 \\ \dot{U} = \frac{H_0 - H}{T_W} \end{cases} \quad (26)$$

As condições iniciais são apresentadas em (27), e a saída em (28).

$$\begin{cases} U = \frac{P_m}{\frac{P_{Btur}}{A_t P_{Bmaq}} H_0} + U_{nl} \\ G = \frac{U}{\sqrt{H_0}} \\ g = P_{gv} = \frac{G}{P_b} \end{cases} \quad (27)$$

$$\begin{cases} G = g \cdot P_b \\ \Delta\omega = \omega - \omega_0 \\ H = \left(\frac{U}{G}\right)^2 \\ P_m = \frac{P_{Btur}}{P_{Bmaq}} A_t H(U - U_{nl}) - \Delta\omega(D + D_t g) \end{cases} \quad (28)$$

4.6.3 Modelagem Orientada a Objetos

O Diagrama 36 apresenta o diagrama de classes das turbinas. Como dito anteriormente, os dois modelos de turbina foram implementados, porém, durante as simulações, percebeu-se que o modelo e os parâmetros definidos no Diagrama 8 tiveram resultados melhores. O modelo do Diagrama 7 é definido pela classe *TTurbinaNaoLinearKundur*, e o modelo do Diagrama 8 é definido pela classe *TTurbinaNaoLinearAnatem*. A classe mãe *Turbina* não possui o atributo *ativo*, pois a dinâmica da turbina não deve ser desconsiderada em uma simulação.

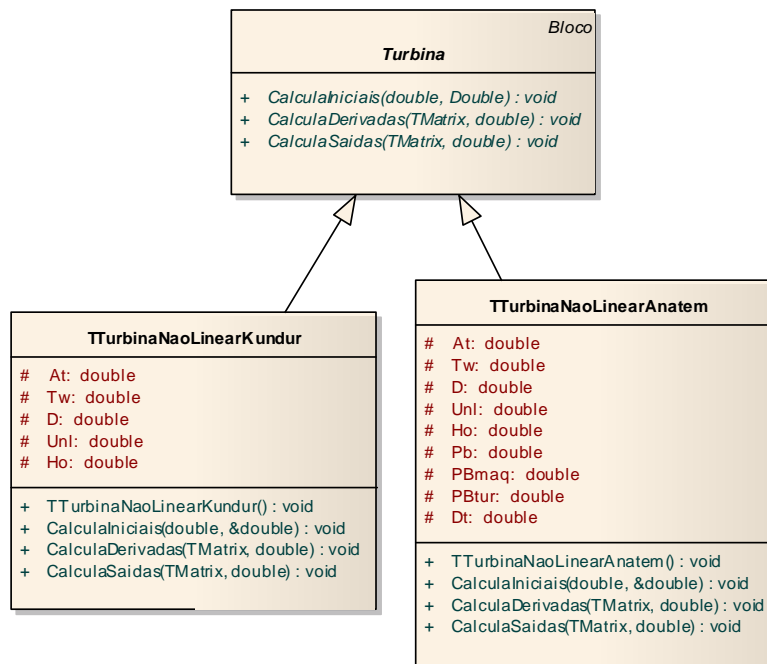


Diagrama 36: Diagrama de classes das Turbinas da UHE de Tucuruí.

4.7 REGULADORES DE VELOCIDADE

Os RVs foram definidos no Item 2.8. Neste item serão abordados diretamente os modelos implementados no *DynaPower* e suas referências para implementação dentro dos padrões de projeto OO adotados. A Fotografia 7 apresenta o armário em que se localiza o RV da UGH 8, em ensaios realizados em fevereiro de 2009.



Fotografia 7: Ensaio sendo realizado no RV da UGH 8.

4.7.1 Modelagem em Diagrama de Blocos

O modelo do RV da primeira etapa foi baseado em Operador Nacional do Sistema Elétrico (2003a, 2003b, 2008a) cujo Diagrama 37 apresenta o seu diagrama de blocos. Nota-se que foram modelados a válvula atuadora e servo-motor de comando, a válvula distribuidora e servo-motor principal, o servo-motor principal e os estatismos permanente e transitório. Os parâmetros deste modelo encontram-se no Anexo I.

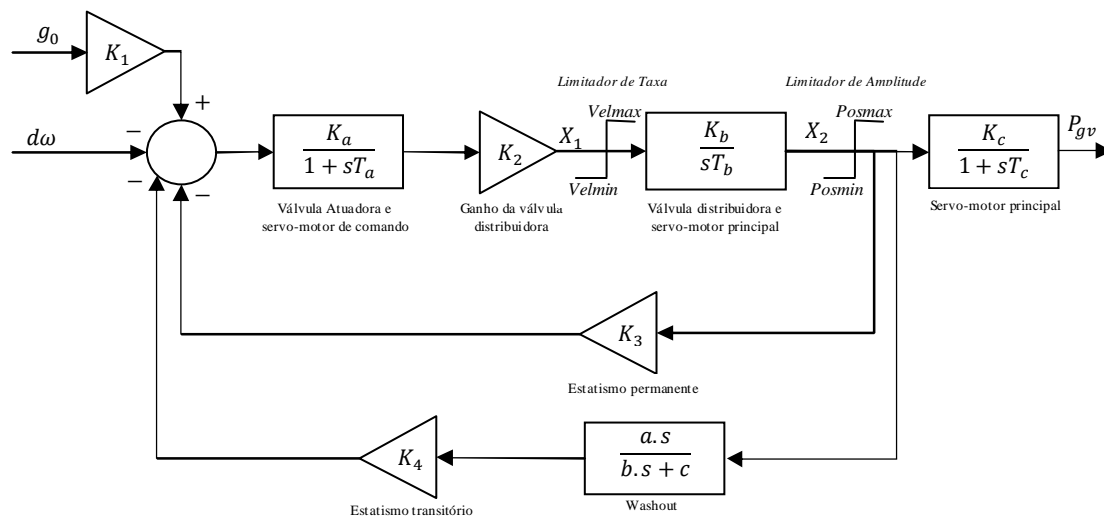


Diagrama 37: Diagrama de Blocos do RV da primeira etapa.

Fonte: Baseado em Operador Nacional do Sistema Elétrico (2003a, 2003b, 2008a).

O modelo do RV da segunda etapa foi baseado em Operador Nacional do Sistema Elétrico (2006a, 2006b, 2008a), e o Diagrama 38 apresenta sua representação em diagramas de blocos com destaque à modelagem do regulador e ao sistema de acionamento hidráulico. Os parâmetros deste modelo encontram-se no Anexo J.

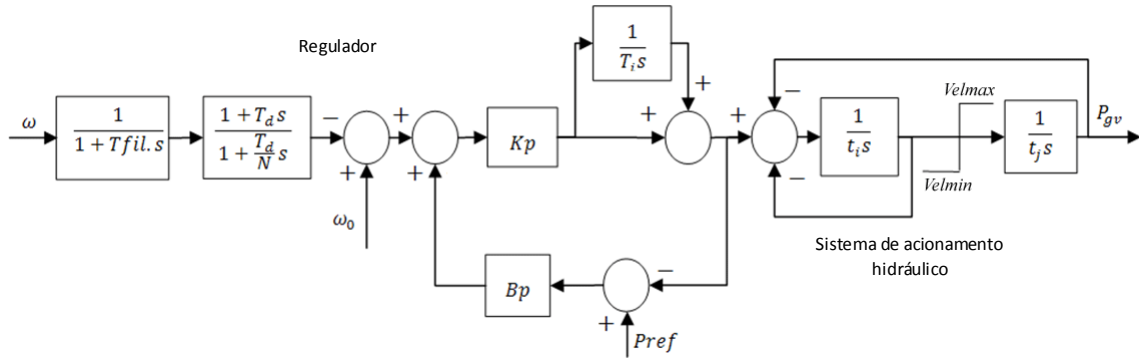


Diagrama 38: Diagrama de Blocos do RV da segunda etapa.

Fonte: Baseado em Operador Nacional do Sistema Elétrico (2006a, 2006b, 2008a).

4.7.2 Modelagem em Espaços de Estados

As equações de estado com a dinâmica do RV da primeira etapa estão apresentadas em (29), e foram calculadas conforme apresenta o Apêndice E, considerando a eliminação dos blocos com zeros.

$$\left\{ \begin{array}{l} e_1 = K_1 g_0 - \Delta\omega - x_2 K_3 - K_4 x_2 \frac{a}{b} + K_4 x_3 \\ \dot{P}_{gv} = \frac{1}{T_c} (x_2 K_c - P_{gv}) \\ \dot{x}_1 = \frac{K_2 K_a e_1 - x_1}{T_a}, \text{ considera ainda a taxa de variação } Velmin \text{ a } Velmax \\ \dot{x}_2 = \frac{K_b x_1}{T_b} \\ \dot{x}_3 = \frac{a.c.x_2}{b^2} - \frac{c.x_3}{b} \end{array} \right. \quad (29)$$

Em (30) são apresentadas as condições iniciais e em (31) o cálculo da saída, devendo-se considerar também que x_2 é limitado em amplitude entre $Posmin$ e $Posmax$.

$$\left\{ \begin{array}{l} x_1 = 0 \\ x_2 = \frac{P_{gv}}{K_c} = \frac{g}{K_c} \\ x_3 = \frac{a.x_2}{b} \end{array} \right. \quad (30)$$

$$P_{gv} = g \quad (31)$$

Para o RV da segunda etapa, e seguindo também o procedimento para retirada dos zeros, as equações de estado são apresentadas em (32).

$$\left\{ \begin{array}{l} e_1 = x_2 + Nx_1 \\ e_2 = \frac{\omega_0 - e_1 + B_p(g_0 - x_3)}{1 + K_p B_p} \\ g_1 = x_3 + e_2 K_p \\ e_3 = g_1 - P_{gv} - x_4 \\ \dot{P}_{gv} = \frac{x_4}{t_j} \\ \dot{x}_1 = \frac{\omega - x_1}{t_{fil}} \\ \dot{x}_2 = [x_1(1 - N) - x_2] \frac{N}{T_d} \\ \dot{x}_3 = \frac{e_2 K_p}{T_i} \\ \dot{x}_4 = \frac{e_3}{t_i}, \text{ considera ainda a taxa de variação } Velmin \text{ a } Velmax \end{array} \right. \quad (32)$$

As condições iniciais são apresentadas em (33), e a saída em (34), considerando que sua amplitude é limitada entre os valores $Lmax$ e $Lmin$.

$$\left\{ \begin{array}{l} x_1 = \omega \\ x_2 = x_1(1 - N) \\ x_3 = P_{gv} = g \\ x_4 = 0 \\ P_{ref} = P_{gv} \end{array} \right. \quad (33)$$

$$P_{gv} = g \quad (34)$$

4.7.3 Modelagem Orientada a Objetos

O Diagrama 39 apresenta o diagrama de classes dos RVs da primeira ($TRVTucuruí1$) e da segunda etapa ($TRVTucuruí2$), de acordo com os padrões de projeto apresentados. A classe *ReguladorVelocidade* é a classe mãe, e é uma especialização da classe *Bloco*, tendo também um atributo *ativo*, que define se deve ou não considerar o RV durante a simulação.

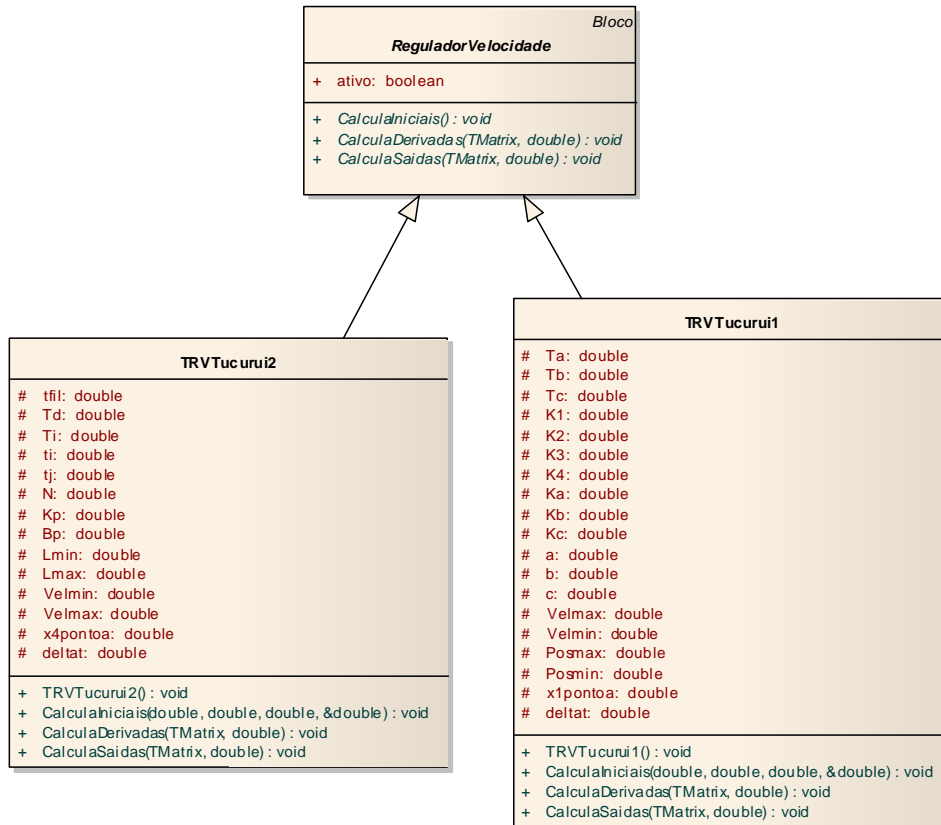


Diagrama 39: Diagrama de classes dos RVs da UHE de Tucuruí.

4.8 CONSOLIDAÇÃO DA MODELAGEM ORIENTADA A OBJETOS

O Diagrama 40 consolida o diagrama de classes modelado para as UGHs da UHE de Tucuruí, destacando as classes que implementam os controles e atuadores. As classes *Gerador* e a *TDadosSistema* pertencem à BCSSD, assim como a classe *Bloco* que foi suprimida mas suas especializações estão destacadas.

É importante ressaltar que cada instância destas classes, ou seja, cada objeto é que representará cada elemento físico do sistema individualmente, podendo-se atualizar parâmetros específicos em tempo de execução, ativar ou desativar blocos dinâmicos específicos, entre outras particularidades.

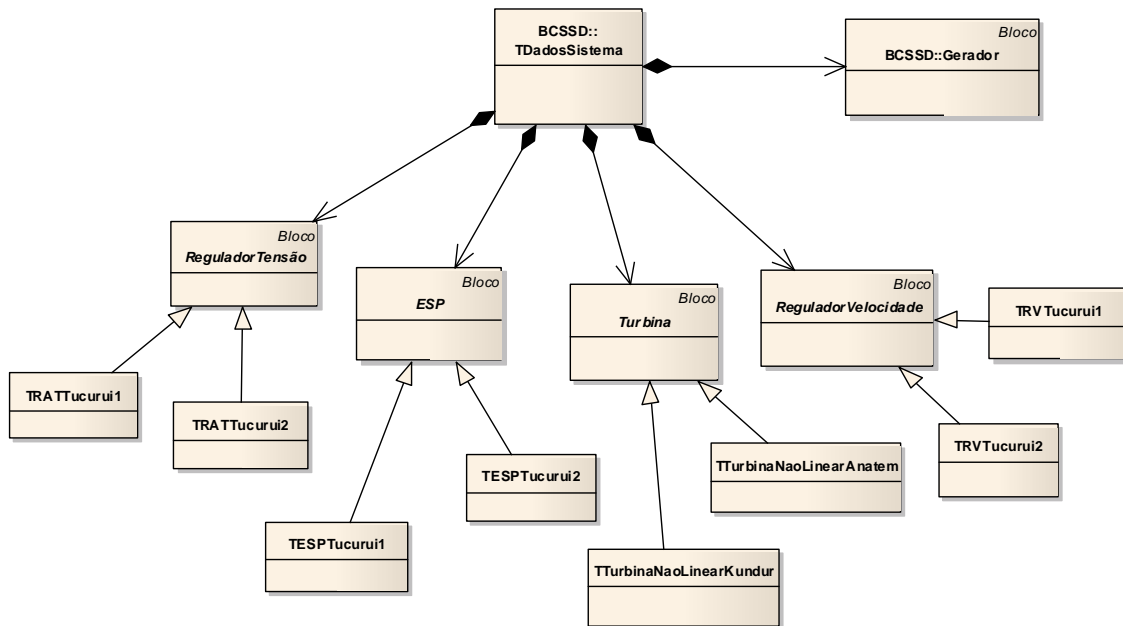


Diagrama 40: Diagrama de classes das UGHs da UHE de Tucuruí.

Com relação ao uso de memória, o Quadro 1 sintetiza a quantidade de estados/saídas⁴ existentes no cenário definido para a UHE de Tucuruí, sem considerar a máquina do barramento infinito, totalizando 697 estados/saídas. Com estes números, pode-se avaliar o grau de complexidade computacional exigido para a realização dos cálculos de uma simulação dinâmica.

Classe	Estados/Saídas	Quantidade na UHE	Total de Estados/Saídas
Gerador	11	23	253
TRATTucuruí1	3	12	36
TRATTucuruí2	8	11	88
TESPTucuruí1	4	12	48
TESPTucuruí2	13	11	143
TTurbinaNaoLinearAnatem	2	23	46
TRVTucuruí1	4	12	28
TRVTucuruí2	5	11	55
			697

Quadro 1: Síntese da quantidade de estados/saídas calculados para o cenário da UHE de Tucuruí

Cada estado/saída de cada bloco dinâmico é calculado em cada passo de integração, considerando ainda que, para os cálculos das derivadas, o método de RungeKutta de quarta ordem executa o método *CalculaDerivadas* quatro vezes.

⁴ Estados e saídas são armazenados em uma mesma estrutura de dados para simplificar a implementação. Quando uma saída não é um estado, seguramente dependerá deles e/ou das entradas.

5 RESULTADOS E COMPARAÇÕES DE SIMULAÇÕES DINÂMICAS EM AMBIENTE COMPUTACIONAL E COM MEDIÇÕES EM CAMPO

Os modelos apresentados para compor as UGHs da UHE de Tucuruí foram implementados e validados individualmente com o auxílio da ferramenta Matlab (2008), com uso do legado deixado por Ferreira (2005) que produziu uma ferramenta para simulação dinâmica de SEP. Esta ferramenta foi utilizada pelo grupo de sistema de controle da UFPA para modelar cada dispositivo das UGHs de Tucuruí, servindo como mais uma referência de comparação e validação dos resultados.

O uso do Matlab para implementação de pequenos SEPs é bastante funcional, porém, não é recomendado para a implementação de sistemas de grande porte como o SIN. Primeiro porque seu objetivo principal não é este, por tratar-se de uma ferramenta de uso muito geral. E depois, porque sua arquitetura para executar a simulação não comporta a quantidade de cálculos necessários para este tipo de aplicação. Daí tantos trabalhos de pesquisa envolvendo este tema.

A implementação do *DynaPower* envolveu a migração do código legado desenvolvido pelo grupo de sistemas de controle da UFPA da plataforma C++ em Builder C++ 5 (BORLAND..., 2002) para o Visual Studio 2008 (MICROSOFT..., 2008). Esta migração ocorreu por diversas razões, entre as quais:

- Uso de frameworks externos que não eram compatíveis com a versão do Builder C++ 5;
- Necessidade de modernizar o código e o ambiente de desenvolvimento;
- Mais recursos de debug existentes do Visual Studio 2008;
- Maior experiência da equipe na ferramenta, e
- Possibilidades de integração com outras plataformas de desenvolvimento e linguagens de programação oferecidas pelos recursos do *.NET Framework* (MICROSOFT..., 2007).

Em seguida, a adaptação do código necessitou de uma nova camada de acesso a dados para acessar o PostgreSQL (2005) e o SQL Server (MICROSOFT..., 2005), onde estão armazenados os dados dos cenários estudados, como os parâmetros das máquinas, controles, ramos, barras e etc, até a efetiva implementação de cada modelo apresentado neste trabalho.

Foi utilizado o servidor Subversion (2006) para controle de versões de arquivos e até o presente momento, as estatísticas fornecidas apresentam 27 semanas de trabalho no código-fonte, com 852 modificações em arquivos em 67 revisões, envolvendo 2 colaboradores. O processo de migração envolvendo tanto a conversão do projeto para a plataforma Visual Studio 2008 como o acesso a dados utilizando a tecnologia *Microsoft Foundation Class Library* (MFC) durou aproximadamente 2 meses e a agregação de *frameworks* específicos para execução de cálculos numéricos envolvendo esparsidade, mais 1 mês. Lembrando que estão sendo consideradas nestas 7 semanas as manipulações nos fontes do *DynaPower*, e que paralelo a essa atividade, foram feitos diversos estudos e treinamentos relacionados ao SIN e à utilização das ferramentas de referência do ONS produzidas e mantidas pelo CEPEL, aos estudos relacionados ao estado da técnica deste tipo de aplicação e aos legados deixados por produções científicas da própria UFPA, além do levantamento dos modelos aqui apresentados e seus ajustes, quando necessários.

Neste Capítulo serão, então, abordados os resultados obtidos com a implementação das UGHs da UHE de Tucuruí no ambiente de simulação do *DynaPower* e alguns dos testes e validações efetuados.

5.1 FLUXO DE CARGA E FALTAS IMPLEMENTADAS

O cálculo do fluxo de carga, como já dito no item 2.2, foge ao escopo deste estudo. Porém, é importante destacar que este cálculo é efetuado pelo próprio *DynaPower*, cujo legado da BCSSD já possui a implementação do método de *Newton-Raphson*, baseado em Monticelli (1983), sem considerar limites de reativos e de controle, e também considerando os limites de reativos e os compensadores estáticos de reativos. Não há uma aplicação separada para o cálculo do fluxo de carga, como o ANAREDE... (2007), para os usuários dos pacotes do CEPEL. Basta definir as constantes para cada tipo de barras que o *DynaPower* executa o fluxo de carga antes de iniciar a simulação dinâmica, parametrizando-se a tolerância de erro e a quantidade de iterações máximas para a convergência.

O Quadro 2 apresenta o resultado de um fluxo de carga calculado pelo *DynaPower* em uma dada condição de operação do cenário elaborado para a UHE de Tucuruí, já apresentado. Para este caso, foi utilizada uma tolerância de 10^{-6} , o fluxo de

carga convergiu em 4 iterações.

Barra	P _t (MW)	Q _t (MVar)	V _t (p.u.)	θ (Grau)	P _L (MW)	Q _L (MVar)
1	315	23,5298	1	9,1309	0	0
2	315	23,5298	1	9,1309	0	0
3	315	23,5298	1	9,1309	0	0
4	315	23,5298	1	9,1309	0	0
5	315	23,5298	1	9,1309	0	0
6	315	23,5298	1	9,1309	0	0
7	315	23,5298	1	9,1309	0	0
8	315	23,5298	1	9,1309	0	0
9	315	23,5298	1	9,1309	0	0
10	315	23,5298	1	9,1309	0	0
12	315	23,5298	1	9,1309	0	0
13	315	23,5298	1	9,1309	0	0
14	315	23,5298	1	9,1309	0	0
15	315	23,5298	1	9,1309	0	0
16	315	23,5298	1	9,1309	0	0
17	315	23,5298	1	9,1309	0	0
18	315	23,5298	1	9,1309	0	0
19	315	23,5298	1	9,1309	0	0
20	315	23,5298	1	9,1309	0	0
21	315	23,5298	1	9,1309	0	0
22	315	23,5298	1	9,1309	0	0
23	315	23,5298	1	9,1309	0	0
24	-4418,08	611,249	1	7,74944e-013	0	0
25	0	0	0,999995	0,590849	1	0
26	0	0	0,999995	0,590849	1	0
27	0	0	0,999995	0,590849	1	0
28	0	0	0,999995	0,590849	1	0
29	0	0	0,999995	0,590849	1	0
30	0	0	0,999995	0,590849	1	0
31	0	0	0,999995	0,590849	1	0
32	0	0	0,999995	0,590849	1	0
33	0	0	0,999995	0,590849	1	0
34	0	0	0,999995	0,590849	1	0
35	0	0	0,999996	0,614545	1	0
36	10	0	0,999992	0,543533	0	0
37	0	0	1	1,19336e-04	2800	0

Quadro 2: Resultado do fluxo de carga para uma condição de operação no cenário construído para representar a UHE de Tucuruí.

Após o cálculo do fluxo de carga, é iniciada a simulação dinâmica em um tempo determinado. As faltas implementadas atualmente no *DynaPower* são:

- Curto-circuito trifásico, informando-se a admitância complexa de curto-circuito;
- Degrau na referência do RAT, informando-se um número real que será multiplicado pela referência durante um tempo determinado, e
- Degrau na referência do RV, informando-se um número real que será multiplicado pela referência durante um tempo determinado.

O Gráfico 1 apresenta duas simulações de 4s de um curto-circuito trifásico executadas pelo *DynaPower*, na barra 13. Esta falta foi aplicada em 0,5s, com duração

de 10ms, e com uma admitância de curto-circuito de $10^5(1+j)$ p.u. Em azul a potência ativa com todos os controles ligados, porém, com o ESP da máquina desligado, e em vermelho a mesma simulação com o ESP da máquina ligado. Com o ESP ligado, percebe-se um maior amortecimento das oscilações, como era de se esperar.

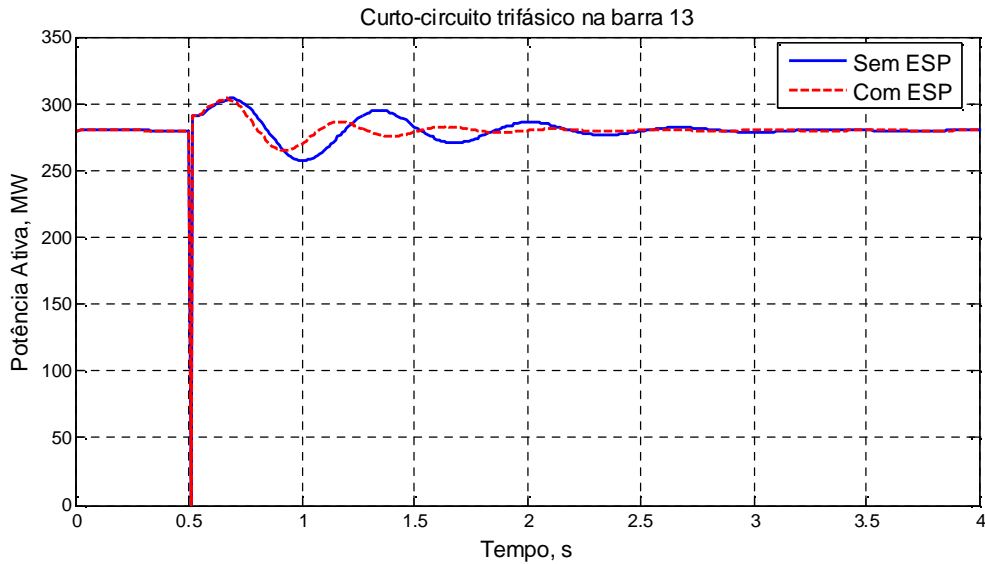


Gráfico 1: Curto-circuito trifásico aplicado na máquina 13.

O Gráfico 2 apresenta duas simulações de um degrau de +10% na referência do RAT da máquina 13, também com e sem a atuação do ESP. Nota-se um baixo amortecimento nesta situação. Situação que ocorre, realmente, na UHE de Tucuruí, pois os ESPs estão ajustados para amortecer os modos interarea, enquanto que os modos locais intra-planta são pouco amortecidos. Este baixo-amortecimento é tema de diversas pesquisas que envolveram e envolvem o grupo de sistemas de controle da UFPA, como em Risuenho (2005).

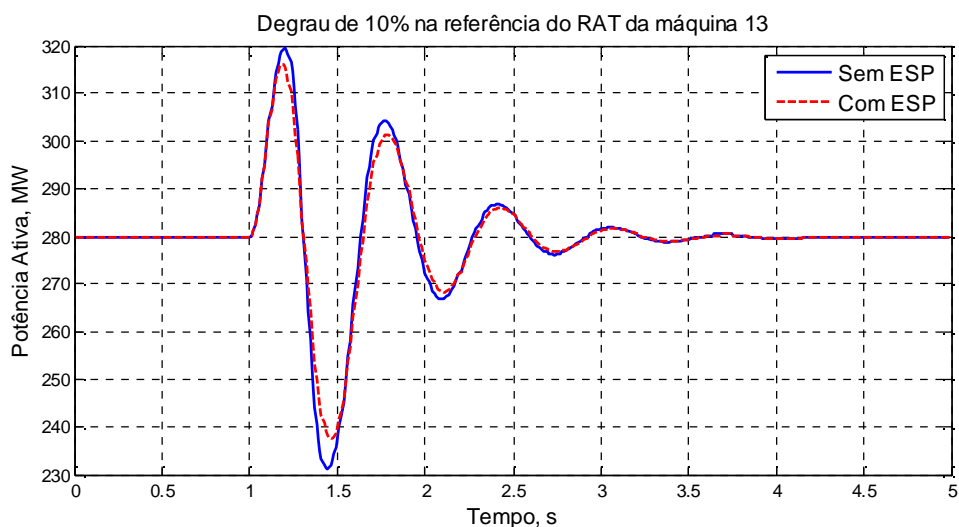


Gráfico 2: Degrau de +10% aplicado referência do RAT da máquina 13.

O Gráfico 3 apresenta a potência ativa de uma simulação de 40s, com a aplicação de um degrau de +10% na referência do RV da máquina 13. Observa-se o comportamento esperado de fase não-mínima com uma rápida queda da potência ativa no início da aplicação e uma subida lenta até que se atinja o regime.

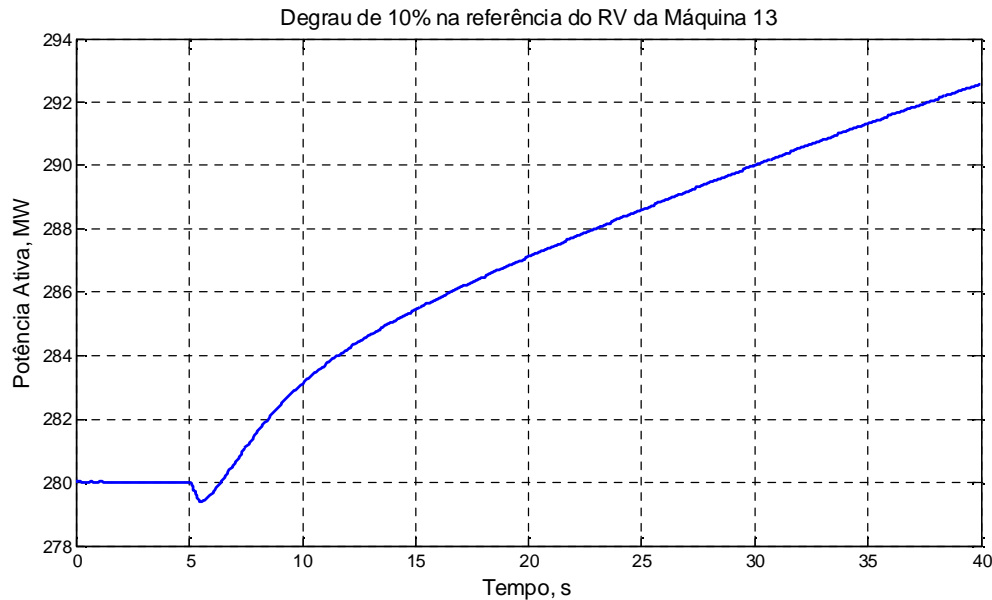


Gráfico 3: Degrau de +10% aplicado referência do RV da máquina 13.

Todos os modelos de máquinas e controles implementados no *DynaPower* foram antes testados individualmente no Matlab (2008) e os resultados comparados com os gráficos gerados pelo ANATEM... (2007) a partir da base de dados do SIN em Operador Nacional do Sistema Elétrico (2008a). Por isso, alguns dos parâmetros apresentados de cada modelo também foram frutos de ajustes e efetuados para obter comportamentos dinâmicos mais próximos dos resultados medidos.

Na BCSSD foi implementada a fórmula de *Kron* (ANDERSON; FOUAD, 1977; KUNDUR, 1994) para redução da ordem da matriz admitância da rede, o que inclusive é algo que pode ser revisto, no futuro, visando outras técnicas computacionalmente mais rápidas. Assim, todas as variáveis relacionadas às barras de geração estão disponíveis e, até o momento, não foram implementadas técnicas para visualização de dados de outras barras ou de determinados pontos do sistema, o que fica também em aberto para implementações futuras. Portanto, nas barras de geração, é possível visualizar diversas variáveis, tais como:

- Potência mecânica (P_m);
- Potência ativa (P_t);
- Potência reativa (Q_t);

- Tensão terminal (V_t);
- Tensão de campo (E_f);
- Corrente do eixo em quadratura (I_q);
- Corrente do eixo direto (I_d);
- Velocidade ou frequência (f), e
- Ângulo (Θ).

Em seguida serão apresentadas as comparações dos resultados obtidos com o *DynaPower* e com outros simuladores comerciais e com medições efetuadas em campo.

5.2 COMPARAÇÃO COM RESULTADOS DE TESTES EXPERIMENTAIS EM CAMPO E COM OS SIMULADORES MATLAB E ANATEM

O Gráfico 4 apresenta o resultado de medição e simulação realizada pelo ONS em Operador Nacional do Sistema Elétrico (2003a), com o modelo do SIN do ano de 2003. “Este teste foi realizado com a unidade sincronizada ao sistema elétrico com potência de 300MW e aplicado um degrau de +30MW[, ou +10% da potência atual], monitorando a potência do gerador [...]” (OPERADOR NACIONAL DO SISTEMA ELÉTRICO, 2003a). Em vermelho o resultado da simulação, feita no ANATEM, e em azul a medição em campo.

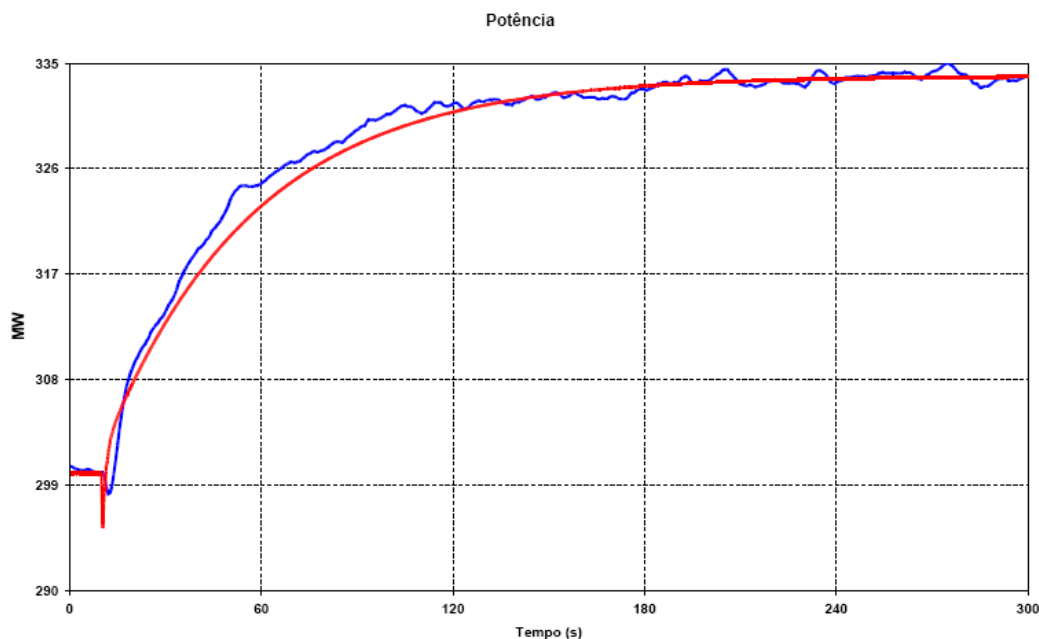


Gráfico 4: Degrau de +10% aplicado na referência do RV de uma máquina da primeira etapa, obtido pelo ONS.

Fonte: Operador Nacional do Sistema Elétrico (2003a, p. 14).

O Gráfico 5 apresenta os resultados deste mesmo cenário obtidos com uso do *DynaPower* e da ferramenta baseada em Ferreira (2005), implementada no Matlab (2008). Nota-se, perfeitamente, a similaridade do comportamento dinâmico.

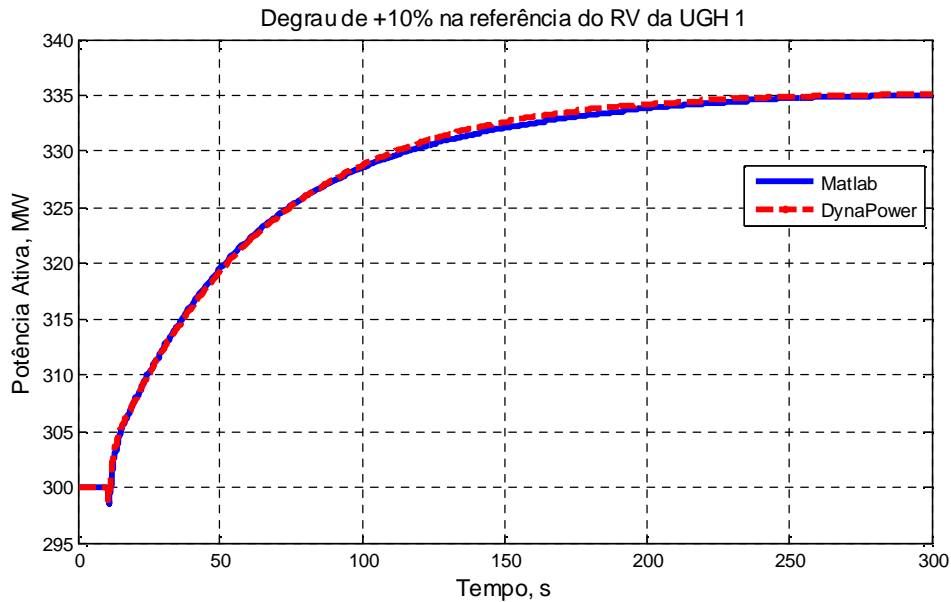


Gráfico 5: Degrau de +10% aplicado na referência do RV de uma máquina da primeira etapa, utilizando *DynaPower* e Matlab.

O Gráfico 6 apresenta o resultado de medição e simulação também retiradas de Operador Nacional do Sistema Elétrico (2003a). Neste caso, a máquina opera em 330MW e é aplicado um degraude -30MW (ou -9,09% da potência atual), monitorando a potência do gerador. Em vermelho o resultado da simulação, feita no ANATEM, e em azul a medição em campo.

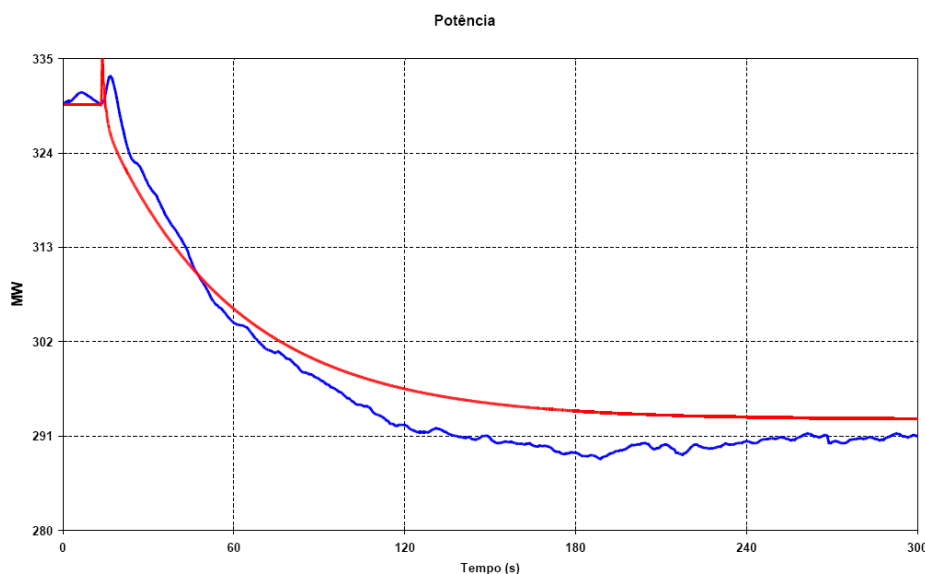


Gráfico 6: Degrau de -9,09% aplicado na referência do RV de uma máquina da primeira etapa, obtido pelo ONS.

Fonte: Operador Nacional do Sistema Elétrico (2003a, p. 15).

Esta mesma situação foi simulada com uso do *DynaPower* e do Matlab (2008), cujo resultado é apresentado no Gráfico 7. Pode-se constatar, também, a similaridade do comportamento dinâmico.

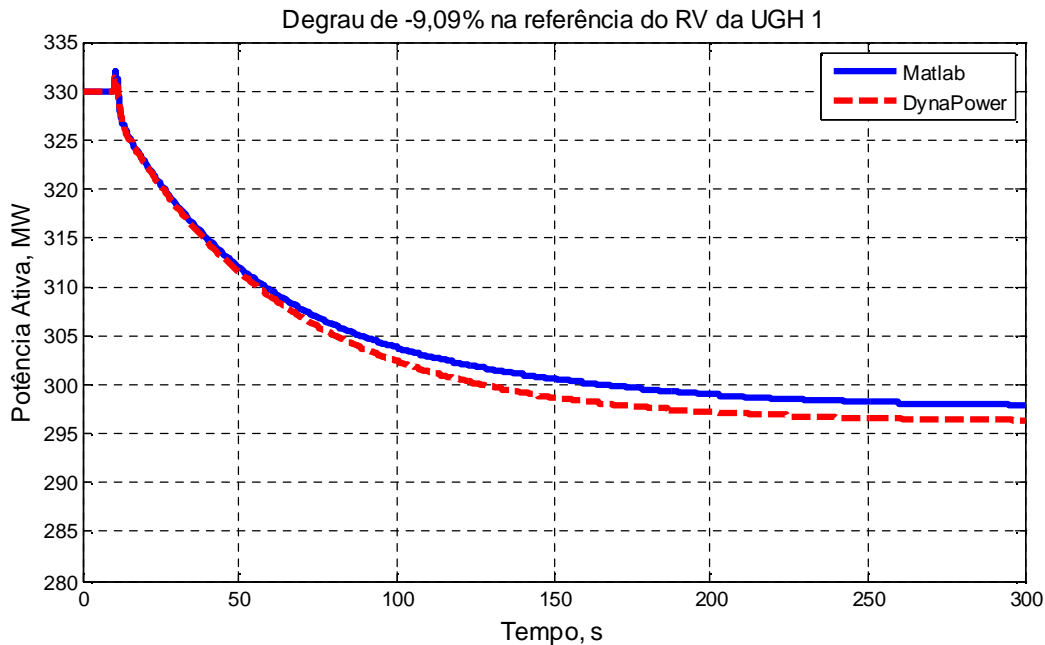


Gráfico 7: Degrau de -9,09% aplicado na referência do RV de uma máquina da primeira etapa, utilizando *DynaPower* e Matlab.

Em todos estes casos apresentados, a frequência ou velocidade das máquinas permaneceram constantes.

Em ensaios realizados pelo grupo de sistemas de controle da UFPA em abril de 2009 na UGH 8, a qual operava em 0,733 p.u., ou 256,5 MW, foi aplicado um degrau de +2,9% em 35 s na referência do RV, com duração de 200 s, e tempo total de 450 s. O Gráfico 8 apresenta a sobreposição das curvas de potência ativa medida neste ensaio, comparados aos resultados de simulação no Matlab (2008) e no *DynaPower*, mostrando como ambas as simulações acompanharam a dinâmica da máquina. Vale ressaltar que, nestas simulações, a constante de tempo do estatismo transitório b foi alterada para 4 s na simulação em Matlab (2008), e o ganho $K4$, também do estatismo transitório, foi alterado para 0,2 no *DynaPower*, e o efeito do limitador de taxa foi cancelado em ambas as simulações. Ajustes como estes são necessários para sintonizar a dinâmica das simulações com as medidas observadas em campo, gerando os resultados apresentados.

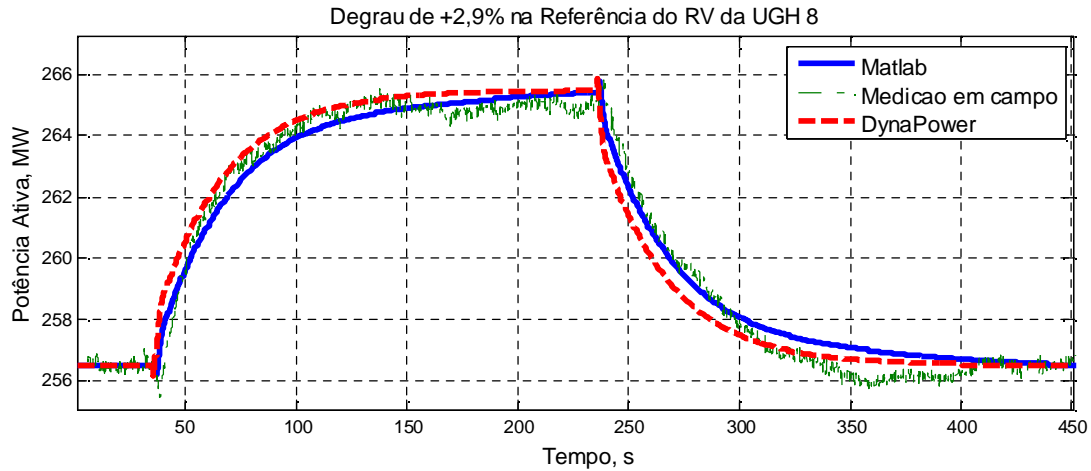


Gráfico 8: Degrau de +2,9% aplicado na referência do RV da UGH 8, medido em campo e comparado com simulações utilizando *DynaPower* e Matlab.

O Gráfico 9 apresenta resultados de medição e simulação em ANATEM, realizados pelo ONS, com uma máquina da segunda etapa, operando a 0,68 p.u. ou 265 MW. Foi aplicado um degraú de aproximadamente -25% em 90 s e retirado em 190 s. Em vermelho está o resultado da medição em campo e em azul o resultado obtido pela simulação em ANATEM, com um modelo do SIN de 2006.

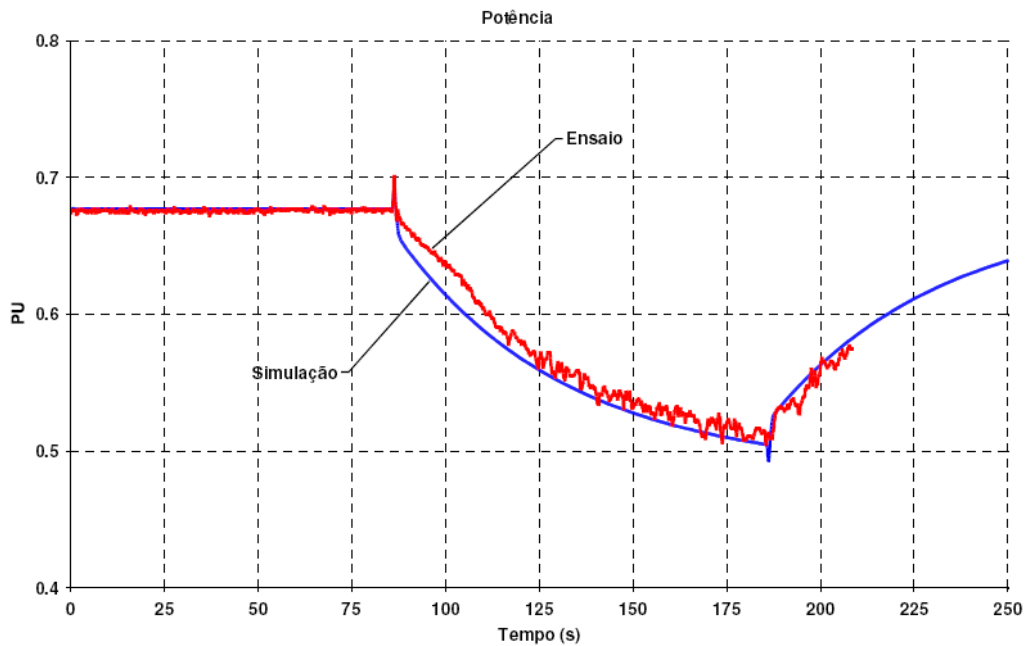


Gráfico 9: Degrau de -25% aplicado na referência do RV de uma máquina da segunda etapa, obtido pelo ONS.

Fonte: Operador Nacional do Sistema Elétrico (2006b, p. 13).

A mesma situação foi simulada no *DynaPower* e no Matlab (2008), cujos resultados estão no Gráfico 10. Nota-se perfeitamente a similaridade das curvas com os resultados obtidos pelo ONS.

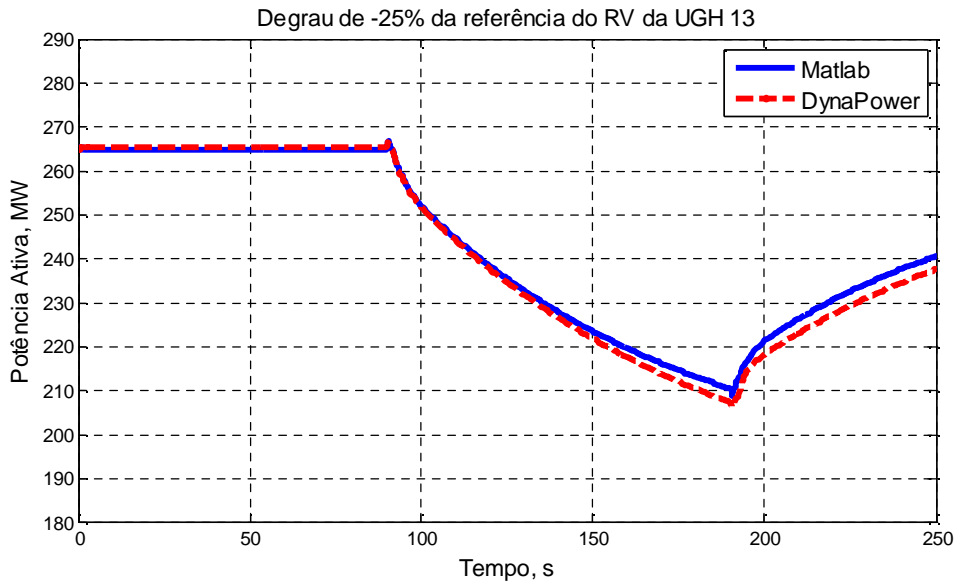


Gráfico 10: Degrau de -25% aplicado na referência do RV de uma máquina da segunda etapa, utilizando *DynaPower* e *Matlab*.

O Gráfico 11 apresenta resultados de medições obtidos pelo ONS, com um degrau de -2% na referência do RAT de uma máquina da segunda etapa, que operava a aproximadamente 0,91 p.u. ou 254,9 MW. Em vermelho o ensaio sem a presença do ESP, e em azul com a presença do ESP. Nota-se perfeitamente que, embora haja um bom amortecimento, há uma variação de frequência não desejável com a presença do ESP. É provável que os ajustes aplicados no ESP desta máquina no momento da realização deste ensaio não estivessem bem ajustados ou compatíveis com os parâmetros levantados neste trabalho. Além disso, também há um ligeiro atraso na curva azul, em relação à vermelha.

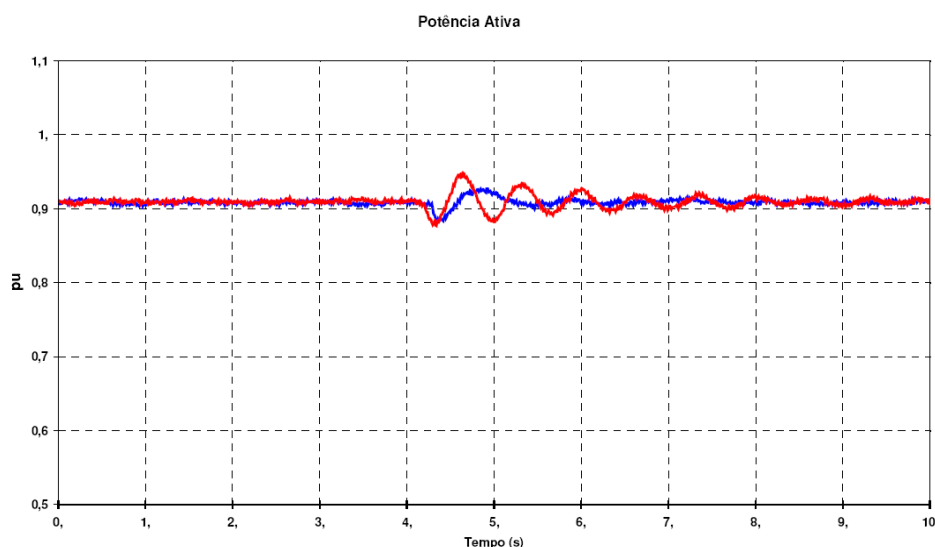


Gráfico 11: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, obtido pelo ONS.

Fonte: Operador Nacional do Sistema Elétrico (2004, p. 20).

O Gráfico 12 apresenta esta mesma situação simulada no Matlab (2008). Com os ajustes apresentados como parâmetros do ESP neste trabalho, não foi observada a variação de frequência, porém, para este ensaio apresentado, foi feito um ajuste no ESP para buscar uma curva parecida com a encontrada nos ensaios de campo e obteve-se este resultado. O que também valida o modelo.

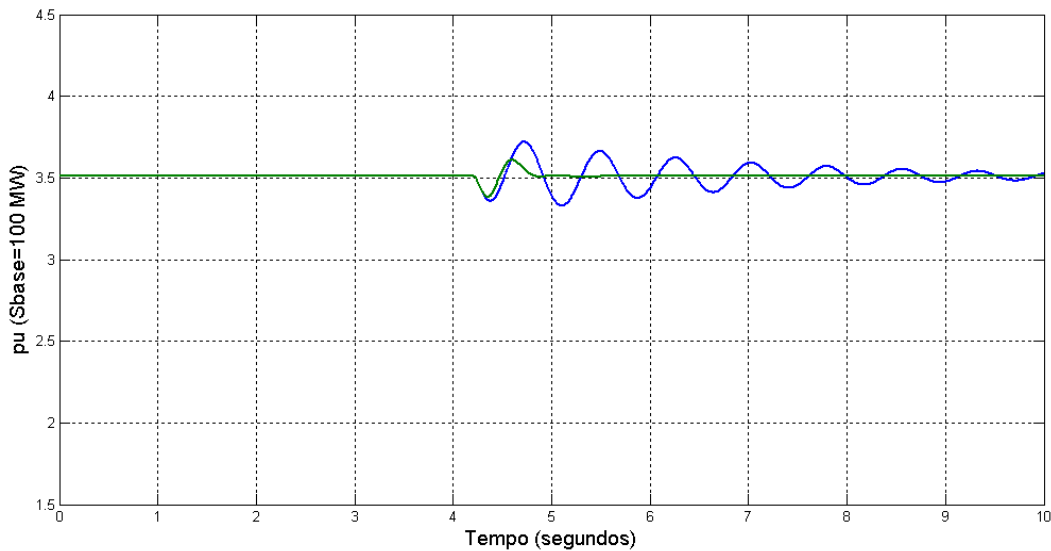


Gráfico 12: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, utilizando o Matlab.

O Gráfico 13 apresenta as curvas geradas pelo *DynaPower* com os parâmetros levantados e ajustados, sem a grande variação de frequência indesejável e com o amortecimento esperado e baixo.

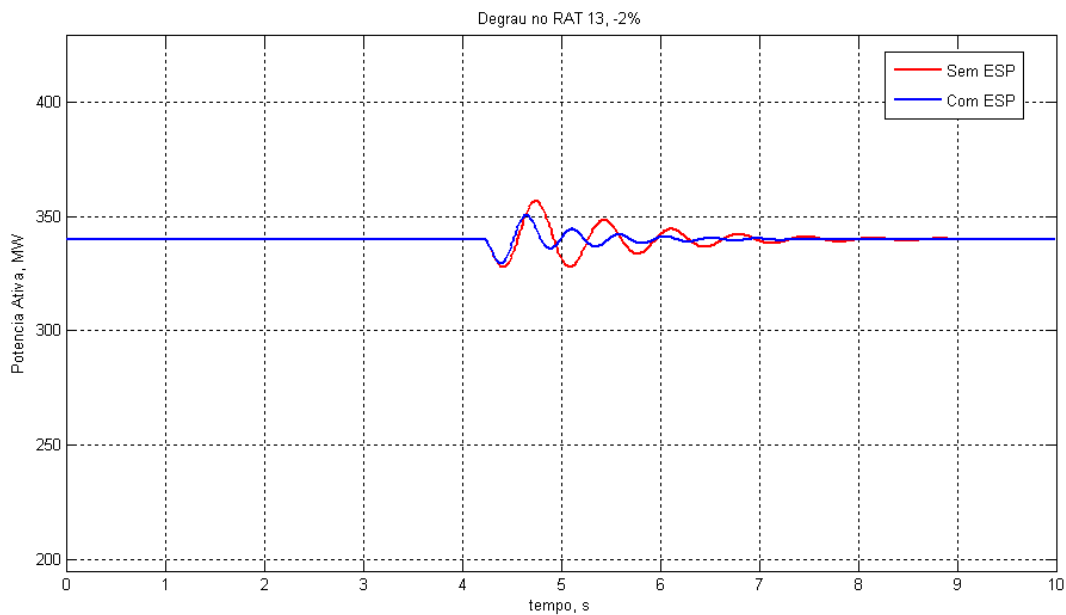


Gráfico 13: Degrau de -2% aplicado na referência do RAT de uma máquina da segunda etapa, utilizando o *DynaPower*.

O Gráfico 14 é resultado de ensaios em campo apresentados em Risuenho (2005) na UGH 1. Foi aplicado um degrau de +4%, com a máquina operando a 240 MW. Observa-se, novamente, o baixo amortecimento das oscilações com a presença do ESP.

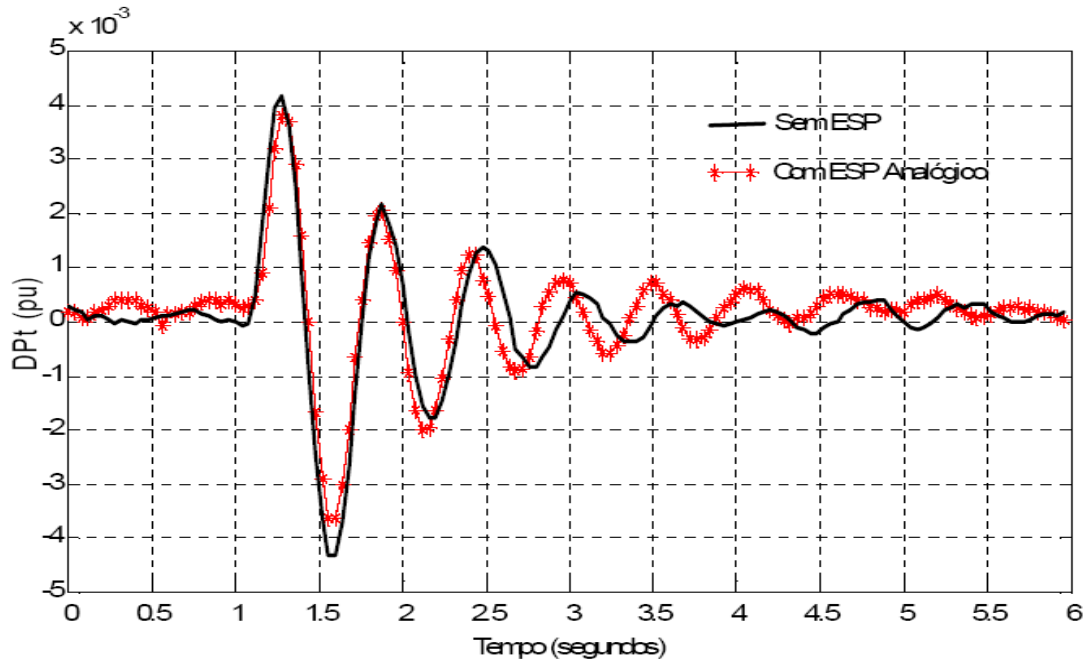


Gráfico 14: Degrau de +4% aplicado na referência do RAT da UGH 1.

Fonte: Risuenho, 2005, p. 115.

O Gráfico 15 apresenta a mesma situação simulada no Matlab (2008), notando-se perfeitamente um comportamento similar ao medido da medição em campo.

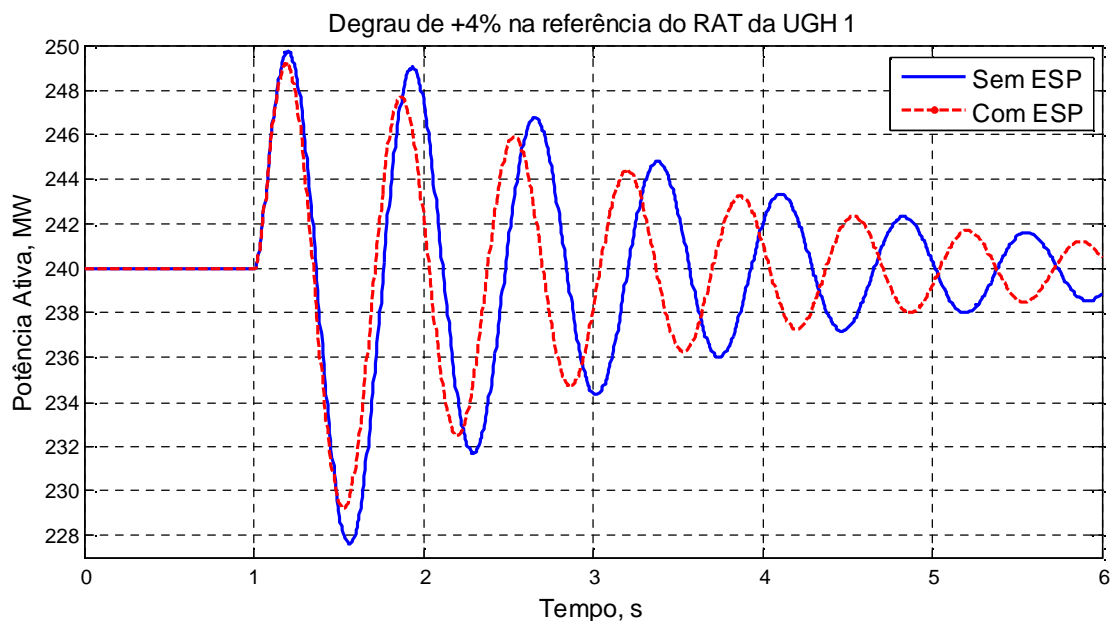


Gráfico 15: Degrau de +4% aplicado na referência do RAT da UGH 1, utilizando o Matlab.

O Gráfico 16 apresenta as curvas geradas pelo *DynaPower* desta mesma situação. Observando-se também compatibilidade entre os comportamentos dinâmicos apresentados em todas as situações.

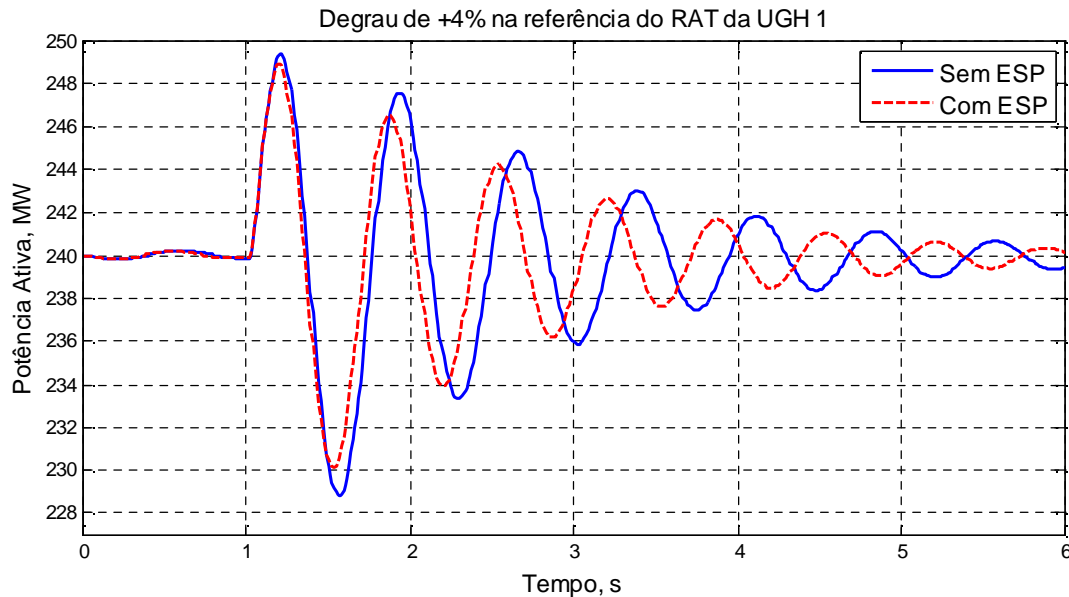


Gráfico 16: Degrau de +4% aplicado na referência do RAT da UGH 1, utilizando o *DynaPower*.

Diversos outros testes também foram executados. Para efetuar as comparações com as medições em campo, eram mais comuns dados disponíveis de potência ativa, daí o fato dos gráficos apresentados neste item apresentarem apenas esta grandeza. Outras diversas simulações e comparações com outros simuladores e com medições em campo podem ser consultadas em Barra Junior (2008b, 2009).

6 CONCLUSÃO

A modelagem para estudos de dinâmica das máquinas e de seus controles e atuadores são preocupações tanto da Eletronorte quanto de seus fornecedores e fabricantes. Estudos com os modelos devem ser feitos ainda em etapas de projeto, muito antes das instalações dos equipamentos na UHE. Alguns dos modelos já se consolidaram há muitos anos, como o das máquinas síncronas, enquanto dos equipamentos e dos controles eletrônicos, sejam analógicos ou digitais, são elaborados com técnicas que sempre estão se modernizando.

Os modelos são elaborados, inicialmente, por seus fabricantes e, ao efetuarem o comissionamento das instalações, entregam esses documentos com sugestões de ajustes dos parâmetros e informações qualitativas sobre a dinâmica desejada. Muitas destas documentações em poder da Eletronorte nos foram encaminhadas, porém, do período da instalação para cá, estudos do ONS e da própria Eletronorte resultaram tanto em modificações nos modelos quanto nos ajustes dos parâmetros, buscando maior eficiência das simulações, deixando os modelos mais próximos da realidade.

Os resultados apresentados neste trabalho refletem pesquisas nessas documentações, tanto de fabricantes, quanto da Eletronorte e do ONS, para determinar os modelos e os ajustes utilizados atualmente pela UHE de Tucuruí.

Com relação aos padrões de projeto OO, quatro deles foram definidos e implementados, de forma a edificar a arquitetura do *DynaPower*, adaptando a BCSSD e modernizando sua plataforma de desenvolvimento para o Visual Studio 2008 (MICROSOFT..., 2008).

Com o *DynaPower*, a BCSSD pôde ser utilizada de forma mais abrangente para modelar os geradores e controles da UHE de Tucuruí, podendo, inclusive, expandir gradativamente a aplicação para modelar todo o SIN e, devido aos padrões de projeto orientados a objeto utilizados, também torna-se possível a implementação de dispositivos mais complexos, como controles digitais e adaptativos.

O mapeamento entre paradigmas de modelagem é comum na área das ciências exatas. Especificamente entre os diagramas de blocos e a orientação a objetos é extremamente relevante devido às tecnologias como C++, Java e .NET serem orientadas

a objeto, com suas devidas peculiaridades, e na representação de SEP e sistemas dinâmicos ser mais utilizado o paradigma dos diagramas de blocos.

O padrão de projeto *Factory Method*, mostrou-se extremamente importante neste mapeamento, permitindo a representação de qualquer sistema dinâmico que possa ser representado através de diagrama de blocos.

Além disso, a estrutura do padrão também permite uma fácil implementação da simulação de faltas, como curto-circuito e o degrau na referência do RAT ou do RV, apresentados neste trabalho, como também a variação de carga, degrau na potência mecânica, saída de uma linha de transmissão, entre outras que podem e devem ser implementadas no futuro.

O *Factory Method* permite, ainda, encapsular a dinâmica do sistema em seus respectivos blocos e ter uma estrutura maior, responsável por guardar as informações das ligações entre eles e efetuar a simulação de forma eficaz e flexível para qualquer desenvolvedor, que pode representar qualquer bloco dinâmico utilizando uma linguagem orientada a objetos.

O padrão *Strategy* complementou o *Factory Method*, simplificando as especializações dos controles e atuadores, limitando a implementações de novos blocos dinâmicos a criação de simples classes concretas com suas dinâmicas definidas, independentemente da quantidade de estados necessários para sua representação. O que não se limita à implementação de equações de estado, podendo, inclusive, serem implementados blocos dinâmicos baseados em inteligência computacional, como controles adaptativos, ou digitais.

O padrão de projeto *Builder* permite encapsular os métodos numéricos de integração que efetuam as simulações de sistemas dinâmicos, independentemente da estrutura de classes que representa os elementos físicos do sistema. Tratando, assim, a integração numérica e a modelagem física dos elementos do sistema de forma totalmente desacoplada.

Os métodos Runge Kutta de quarta ordem e Trapezoidal foram implementados e apresentaram resultados numéricos muito próximos para efeitos práticos no sistema estudado, o que já era esperado devido à sua natureza. Além destes, outros métodos ainda podem ser implementados visando, inclusive, maior rapidez na execução dos cálculos, como métodos com passo variável. Embora as aplicações comerciais e em

pesquisas apresentadas em artigos deste tipo de aplicação já tenham consolidado o Runge Kutta de quarta ordem como mais usual.

O padrão de projeto *Facade* implementa uma forte qualidade que pode ser explorada principalmente por interfaces gráficas, tratando um conjunto complexo de blocos dinâmicos como um bloco único, dependendo do escopo que se deseja visualizar. Por exemplo, para apresentação do RAT da segunda etapa, muitos diagramas tiveram de ser apresentados neste trabalho. Para o usuário do *DynaPower*, dependendo do escopo que esteja visualizando, poderia ser desnecessário poluir a tela com todos aqueles detalhes. Com o *Facade*, todos aqueles blocos poderiam ser visualizados como um bloco único, com entradas e saídas, e seus detalhes vistos, se necessário.

A interface gráfica para o *DynaPower* atualmente está em construção, com protótipo a arquitetura definida. O estado da técnica deste tipo de aplicação também apresenta um déficit significativo neste tipo de implementação. Atenção especial a este estudo também pode ser dada no futuro, que também envolve uma complexidade razoável, envolvendo desde pesquisa com *frameworks* gráficos disponíveis, técnicas de interface homem-máquina, e os diversos relatórios e diagramas unifilares e em blocos necessários ao usuário, podendo até ser tratada em projeto específico.

A simplicidade em codificar blocos dinâmicos é uma das principais qualidades oferecidas no *DynaPower*, tornando possível a codificação de modelos complexos, baseados em diversas técnicas de modelagem baseadas em inteligência computacional, controle digital, dentre outros recursos que raramente são disponíveis nas atuais ferramentas comerciais em uso nos estudos dinâmicos do SIN. Além disso, a metodologia utilizada facilita a implementação de modelos de simulação de sistemas de grande porte, os quais podem ser simulados com uma camada, arquitetada de forma eficiente, abstraindo, dessa forma, diversas outras tão necessárias quanto complexas, como para o cálculo do fluxo de carga, integração numérica, cálculo da matriz admitância reduzida, coleta de dados para visualização de gráficos, entre outras.

Apenas a título de comparação, o Matlab (2008), mesmo tendo a usabilidade como maior forte, também tem limitações quando utilizado para simulações de sistemas de grande porte, impossibilitando sua utilização para esta finalidade específica. Em outras palavras, os resultados da aplicação das técnicas investigadas neste estudo permitem maior flexibilidade aos projetistas e acelera o desenvolvimento e a manutenção dos aplicativos, possibilitando dentre outras funcionalidades, o

acoplamento de componentes ou aplicações de interfaces gráficas que vierem a ser implementadas no futuro.

As UGHs da primeira e da segunda etapa da UHE de Tucuruí foram modeladas e as rotinas implementadas, tendo os resultados obtidos sido comparados com dois outros simuladores comerciais (ANATEM..., 2007; MATLAB, 2008) e com medições efetuadas na própria UHE. As medidas de campo utilizadas na validação deste estudo foram efetuadas tanto pelo ONS quanto pelo próprio grupo de sistemas de controle da UFPA, sendo que todos os resultados obtidos foram bem consistentes com as medidas de campo, validando, dessa forma, a implementação do *software DynaPower*.

Os modelos desenvolvidos para cada dispositivo foram testados primeiramente de forma isolada e depois em conjunto. Estes testes individualizados foram essenciais para a verificação da implementação correta da dinâmica de cada dispositivo, de acordo com a documentação técnica fornecida pela Eletronorte (dados de fabricantes) e pelo ONS (relatórios). Assim, cada dispositivo foi agregado ao modelo geral da UHE após ser devidamente testado de forma isolada, e o resultado do sistema completo pôde ser comprovado nos resultados atingidos aqui.

A partir deste trabalho, todo o SIN pode ser implementado, expandindo-se a rede externa da Eletronorte, e avançar até onde for necessário estudar seus detalhes, aplicando-se um barramento infinito ou equivalentes dinâmicos em que os detalhes não forem necessários.

Diversas técnicas de engenharia de *software* ainda podem ser aplicadas visando melhorias tanto na BCSSD quanto no *DynaPower*. O tempo de processamento⁵, por exemplo, para simular 300 s de um degrau na referência do RV, como apresentado neste trabalho, utilizando o método de integração Runge Kutta de quarta ordem, e passo de integração de 0,005 s, leva por volta de 9,3 min tanto no *DynaPower* como no Matlab (2008), enquanto no ANATEM... (2007) leva-se alguns segundos utilizando a base de dados de todo o SIN (OPERADOR NACIONAL DO SISTEMA ELÉTRICO, 2008a). Porém, deve-se também considerar que para escolher o caso que se deseja estudar, no caso do ANATEM, deve-se antes rodar o fluxo de carga no ANAREDE... (2007), e nem sempre é simples esta tarefa, pois o ANAREDE não tem uma interface muito amigável e muitos ajustes precisam ser feitos para a criação de um caso, enquanto que

⁵ A configuração do computador usado como referência foi um com processador Intel® Core 2 Quad, 2,4 GHz, com 1 GB de memória RAM, e durante a simulação, apenas um dos núcleos foi utilizado.

tanto na aplicação elaborada no Matlab por Ferreira (2005) quanto no *DynaPower*, o fluxo de carga é calculado no momento da execução da simulação, simplificando este processo.

De fato, o tempo de processamento é algo que ainda pode e deve ser trabalhado no futuro. Estudos de Dingle e Hildebrandt (1998) mostraram que o uso da OO, embora apresente muitas vantagens, acarretam, também, em mau uso do gerenciamento de memória na execução dos cálculos numéricos, e apresenta ainda técnicas para resolver este problema em aplicações em C++. Devido à alta complexidade computacional envolvida, aplicando estas técnicas, por exemplo, deve-se reduzir substancialmente o tempo de processamento, objetivando chegar perto e até reduzir o tempo praticado em simulações no ANATEM, que ainda tem o menor tempo de processamento dos aplicativos testados e disponíveis no mercado.

Outra possível abordagem para diminuir o tempo de processamento, e também complementar, é sugerida em Sena (2006), com a implementação eficiente de *frameworks* específicos para cálculo numérico, otimizado para processadores específicos, como o *Math Kernel Library* (INTEL..., [2008]).

Aplicações baseadas em inteligência computacional ou baseadas em equipamentos digitais não podem ser aplicadas no ANATEM, a não ser a partir de seus equivalentes contínuos, devido a arquitetura de *software* utilizada. Isto é uma limitação muito séria para aplicações em SEP de grande porte como o SIN. Com o *DynaPower* e a BCSSD, tais limitações deixam de existir, pois com os padrões de projeto adotados, qualquer dinâmica expressa em C++ pode ser implementada, seja ela qual for a técnica utilizada para modelar.

Conclui-se, então, o desafio desta pesquisa ao disponibilizar o *DynaPower* como uma ferramenta capaz de simular a dinâmica interna da UHE de Tucuruí, e a modelagem detalhada da usina, possibilitando estudos complexos da área de controle e estabilidade de sistemas dinâmicos, configurando-se assim em mais estes legados advindos dos estudos do grupo de sistemas de controle, ligado à FEE e ao PPGEE, da UFPA.

REFERÊNCIAS

ANAREDE – Análise de Redes Elétricas, Version 09.01.08. Rio de Janeiro: Centro de Pesquisas de Energia Elétrica – CEPEL, 2007.

ANATEM – Análise de Transitórios Eletromecânicos, Version 10.0.2. Rio de Janeiro: Centro de Pesquisas de Energia Elétrica – CEPEL, 2007.

ANDERSON, Paul M.; FOUAD, Abdel-Aziz A. *Power system control and stability*. 1. ed. Iowa, Estados Unidos: The Iowa State University Press, 1977.

ARRILLAGA, Jos; WATSON, Neville R. *Computer Modelling of Electrical Power Systems*. 2. ed. Estados Unidos: Wiley, 2001.

BARCELOS, Silvangela Lílian da Silva Lima. *Avaliação da operação conjunta dos TCSCS da interligação Norte/Sul do Sistema Elétrico Brasileiro*. Dissertação (Mestrado em Ciências e em Engenharia Elétrica) – COPPE/UFRJ, Rio de Janeiro, 2007.

BARRA JUNIOR, Walter. *Estratégias Neuro-Fuzzy Adaptativas Aplicadas ao Controle de Sistemas de Potência*. Tese (Doutorado em Engenharia Elétrica) – PPGEE/UFPA, Belém, 2001.

BARRA JUNIOR, Walter (Org.). *1º relatório quadrimestral do projeto “desenvolvimento de um sistema computacional para análise dos efeitos da expansão e modernização da UHE de Tucuruí”*. UFPA, maio 2008a.

_____. *2º relatório quadrimestral do projeto “desenvolvimento de um sistema computacional para análise dos efeitos da expansão e modernização da UHE de Tucuruí”*. UFPA, Belém, nov. 2008b.

_____. *3º relatório quadrimestral do projeto “desenvolvimento de um sistema computacional para análise dos efeitos da expansão e modernização da UHE de Tucuruí”*. UFPA, Belém, fev. 2009.

BARREIROS, José Augusto Lima. *A pole-shifting self-tuning power system stabilizer*. Dissertação (Mestrado em Engenharia Elétrica). University of Manchester Institute of Science and Technology (UMIST), Inglaterra, Manchester, 1989.

_____. *Métodos de controle adaptativo aplicados à síntese de estabilizadores de sistemas de potência*. Tese (Doutorado em Engenharia Elétrica). PPGEEL/UFSC, Santa Catarina, 1995.

BATTISTI, Júlio. *ASP.NET: uma nova revolução na construção de sites e aplicações web*. Rio de Janeiro: Axcel Books, 2001.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *UML: guia do usuário*. Rio de Janeiro: Campus, 2000.

BORGES, Thiago Trezza. *Desenvolvimento de um pacote gráfico para análise de redes*. Dissertação (Mestrado em Engenharia Elétrica) – UFJF, Juiz de Fora, 2006.

BORLAND C++ Builder Enterprise Suite, Version 5.0. Estados Unidos: Inprise Corporation, 2002.

COSTA JÚNIOR, Carlos Tavares da Costa. *Méthodes de Commande Adaptative par Supervision pour la Régulation d'un Générateur Entraîné par Turbine Hydraulique*. Tese (Doutorado em Automatique et Productique) – Institut National Polytechnique de Grenoble (INPG), França, Grenoble, 1999.

CROW, M. *Computational Methods for Electric Power Systems*. Estados Unidos: CRC Press, 2003.

CUNHA, Cristina. *Métodos Numéricos*. 2. ed. Campinas: Editora da Unicamp, 2003.

DI PAOLO, Ítalo Flexa et al. Creational object-oriented design pattern applied to the development of software tools for electric power systems dynamic simulations. *Artigo IEEE Latina.pdf* [Submetido à revista IEEE América Latina em mar. 2009 e aceito para publicação em ago. 2009]. Belém, 2009a.

_____. Padrão de projeto factory method aplicado à modelagem e estudos da dinâmica de sistemas elétricos de potência. In: III Congresso Brasileiro de Eficiência Energética (III CBEE). *Anais do III Congresso Brasileiro de Eficiência Energética*, Belém, 2009b.

DINGLE, Adair; HILDEBRANDT, Thomas H. Improving C++ Performance Using Temporaries. *IEEE Computer Society*, Estados Unidos, v. 31, p. 31-41, 1998. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=660188>. Acesso em: 12 jun. 2008.

DZAFIC, I; GLAVIC, M; TESNJAK, S. An Object-Oriented Graphical Package for Power System Simulation and Analysis. In: IEEE MELECON 2004. *Proceedings of the 12th IEEE Mediterranean*, Dubrovnik, Croácia, v. 3, 2004, p. 835-839. Disponível em: <<http://ieeexplore.ieee.org/ielx5/9339/29677/01348073.pdf?isnumber=29677>>. Acesso em: 09 jun. 2008.

ELETRONORTE. *Determinação da impedância equivalente do sistema Eletronorte na barra-500 KV da SE Presidente Dutra* – Geração Máxima. [S.l.], 1988, 12 p.

_____. *Sistema de excitação estática digital* – diagrama em blocos do regulador de tensão, [S.l.], 2001, 10 p.

ELETRONORTE; ALSTOM POWER. *Manual de Comissionamento Geradores* – UHE Tucuruí, identificação 1EHE452977, rev. 1, [S.l.], [2000]. p. 11-15.

ENTERPRISE Architect, Version 7.1.829. Creswick, Austrália: Sparx Systems, 2008.

FERREIRA, André Maurício Damasceno. *Projeto de Estabilizadores de Sistemas de Potência utilizando Técnicas de Controle Fuzzy*. Dissertação (Mestrado em Engenharia Elétrica) – PPGEE/UFPA, Belém, 1998.

_____. *Amortecimento de oscilações eletromecânicas em sistemas elétricos de potência utilizando controle robusto adaptativo em dispositivos FACTS TCSC*. Tese (Doutorado em Engenharia Elétrica) – PPGEE/UFGA, Belém, 2005.

FITZGERALD, A. E.; KINGSLEY JR., Charles; UMANS, Stephen D. *Máquinas Elétricas*. 6. ed. Porto Alegre: Bookman, 2006.

FREEMAN, Eric; FREEMAN, Elizabeth. *Use a Cabeça! Padrões de Projetos (Design Patterns)*. 2. ed. Rio de Janeiro: Alta Books, 2007.

FURLAN, José Davi. *Modelagem de objetos através da UML: the Unified Modeling Language*. São Paulo: Makron Books, 1998.

GAMMA, Erich. et al. *Design Patterns: elements of reusable object-oriented software*. 35. ed. Massachusetts, Estados Unidos: Addison Wesley, 2007.

GENERAL ELECTRIC. *Manual de Operação e Manutenção*. [S.l.], [19--], p. 1-8.

HORSTMANN, Cay. *Padrões e Projetos Orientados a Objetos*. 2. ed. Porto Alegre: Bookman, 2007.

INTEL Math Kernel Library, Version 10.1. Estados Unidos: Intel Corporation, [2008]. Disponível em: <<http://www.intel.com/cd/software/products/asm-na/eng/307757.htm>>. Acesso em: 12 jan. 2009.

KUNDUR, Prabha. *A Power System Stability and Control*, Estados Unidos: McGraw Hill, 1994.

KUNDUR, Prabha et al. Definition and Classification of Power System Stability. *IEEE Transactions on Power Systems*, Estados Unidos, v. 19, n. 2, p. 1387-1401, 2004. Disponível em: <<http://ieeexplore.ieee.org/iel5/59/29221/01318675.pdf?arnumber=1318675>>. Acesso em: 12 ago. 2008.

_____. Practical utility experience with and effective use of power system stabilizers. *IEEE Power Engineering Society General Meeting*, Estados Unidos, v. 3, p. 1777-1785, 2003. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1267426>. Acesso em: 12 ago. 2008.

MACHADO JÚNIOR, Zulmar Soares. *Desenvolvimento de um ambiente computacional flexível para aplicações de fluxo de potência ótimo utilizando modelagem orientada a objetos*. Tese (Doutorado em Ciências e em Engenharia Elétrica) – COPPE/UFRJ, Rio de Janeiro, 2005.

MANZONI, Alessandro. *Desenvolvimento de um sistema computacional orientado a objetos para sistemas elétricos de potência: aplicação a Simulação rápida e análise de estabilidade de tensão*. Tese (Doutorado em Ciências e em Engenharia Elétrica) – COPPE/UFRJ, Rio de Janeiro, 2005.

MATLAB, Version 7.6.0 (R2008a). Estados Unidos: The Math Works, 2008.

MATOS, Alexandre Veloso de Matos. *UML: Prático e Descomplicado*. São Paulo: Érica, 2002.

MICROSOFT .NET Framework, Version 3.5. Estados Unidos: Microsoft Corporation, 2007. Disponível em: <<http://www.microsoft.com/downloads/details.aspx?displaylang=pt-br&FamilyID=333325fd-ae52-4e35-b531-508d977d32a6>>. Acesso em: 17 fev. 2009.

MICROSOFT SQL Server, Version 2005 Express Edition. Estados Unidos: Microsoft Corporation, 2005.

MICROSOFT Visual Studio, Version 2008. Estados Unidos: Microsoft Corporation, 2008.

MONTICELLI, Alcir José. *Fluxo de carga em redes de energia*. São Paulo: Edgard Blücher Ltda., 1983.

OGATA, Katsuhico. *Engenharia de Controle Moderno*. Rio de Janeiro: Prentice Hall do Brasil, 2003.

OPERADOR NACIONAL DO SISTEMA ELÉTRICO. Casos de referência – estabilidade – transitórios eletromecânicos. *bdadosMAR2008.zip*: base de dados do sistema interligado nacional para uso no ANATEM e ANAT0. Rio de Janeiro, 2008a. Disponível em: <http://www.ons.org.br/avaliacao_condicao/casos_eletromecanicos.aspx>. Acesso em: 27 maio 2008.

_____. Diretoria de Planejamento Programação da Operação. *Avaliação da Regulação de Velocidade da UHE de Tucuruí na Ocorrência do Dia 20/12/2005 às 12h40min*. ONS RE 3/043/2006. Rio de Janeiro, 2006a.

_____. Diretoria de Planejamento Programação da Operação. *Novo Modelo do Regulador de Velocidade da UHE Tucuruí*. ONS RE 03/219/2003. Rio de Janeiro, 2003a.

_____. Diretoria de Planejamento Programação da Operação. *Otimização do Regulador de Velocidade da UHE Tucuruí*. ONS RE 03/220/2003. Rio de Janeiro, 2003b.

_____. Diretoria de Planejamento Programação da Operação. *Reajuste e Validação do Regulador de Tensão e PSS da UHE Tucuruí II*. ONS RE 03/161/2004. Rio de Janeiro, 2004.

_____. Diretoria de Planejamento Programação da Operação. *Validação e Otimização do Regulador de Velocidade da UHE Tucuruí - Segunda Etapa*. ONS RE 3/285/2006. Rio de Janeiro, 2006b.

_____. *O ONS*. Rio de Janeiro, 2008b. Disponível em: <http://www.ons.org.br/institucional/o_que_e_o_ons.aspx>. Acesso em: 17 fev. 2009.

PALMA, Rodrigo B. et al. Object Oriented Platform for an Integrated Analysis of Energy and Transportation Networks. *IEEE Transaction on Power Systems*, Estados Unidos, v. 18, n. 3, p. 1062-1069, 2003. Disponível em: <<http://ieeexplore.ieee.org/iel5/59/27352/01216147.pdf?arnumber=1216147>>. Acesso em: 15 ago. 2008.

PAULA FILHO, Wilson de Pádua. *Engenharia de Software: fundamentos, métodos e padrões*. 2. ed. Rio de Janeiro: LTC, 2003.

POSTGRESQL, Version 8.3.3. Estados Unidos: PostgreSQL Global Development Group, 2008. Disponível em: <<http://www.postgresql.org/download>>. Acesso em: 17 jun. 2008.

RISUENHO, Jorge Ricardo Ribeiro. *Desenvolvimento de um estabilizador digital de sistemas de potência para testes em unidades geradoras da UHE de Tucuruí*. Dissertação (Mestrado em Engenharia Elétrica) – PPGEE/UFPA, Belém, 2005.

SELVAN, M. P.; SWARUP, K. S. Modeling and analysis of unbalanced distribution system using object-oriented methodology. *Electric Power Systems Research*, Estados Unidos, n. 76, p. 968-979, 2006. Disponível em: <http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V30-4J616G3-2&_user=686466&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000037618&_version=1&_urlVersion=0&_userid=686466&md5=d538940190a33593a44577dc22821ca9>. Acesso em: 12 set. 2008.

SENA, José Adolfo da Silva. *Biblioteca de Simulação de Sistemas de Potência & SimuSinc 2.0*. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – UFPA, Belém, 1998.

_____. *Desenvolvimento de Framework para análise e simulação no domínio do tempo de sistemas elétricos de potência*. Trabalho de Qualificação (Qualificação ao Doutorado em Engenharia Elétrica) – PPGEE/UFPA, Belém, 2006.

SENA, José Adolfo da Silva et al. Estratégias de padrões de projeto do tipo bridge para o desenvolvimento de softwares para simulação de sistemas dinâmicos de grande porte. In: *Sixth Latin-American Congress on Electricity Generation and Transmission. Proceedings of Sixth Latin-American Congress on Electricity Generation and Transmission*, Mar del Plata, Argentina, 2005.

_____. *Framework para simulação de sistemas dinâmicos de grande Porte: aplicação em sistemas elétricos de potência*. In: *XVI Congresso Brasileiro de Automática (XVI CBA)*. *Anais do XVI Congresso Brasileiro de Automática*, Salvador, 2006.

SILVA, Douglas Marcos da. *Guia de Consulta Rápida UML*. São Paulo: Novatec, 2001.

SOMMERVILLE, Ian. *Engenharia de Software*. 6. ed. São Paulo: Addison Wesley, 2003.

SUBVERSION, Version 1.4.2 (r22196). Estados Unidos: CollabNet, 2006. Disponível em: <<http://subversion.tigris.org/downloads/subversion-1.4.2.tar.bz2.asc>>. Acesso em: 03 ago. 2008.

TURBINA hidráulica. In: WIKIPÉDIA – a enciclopédia livre. [S.l.], 2009. Disponível em: <http://pt.wikipedia.org/wiki/Turbina_hidráulica>. Acesso em: 05 mar. 2009.

VILLARES EQUIPAMENTOS. Divisão VIGESA. *Manual de Operação e Manutenção*. [S.l.], [19--], p. 1-11.

WORKING group on prime mover and energy supply models for system dynamic performance studies. Hydraulic turbine and turbine control models for system dynamic studies. *IEEE Transaction on Power Systems*, Estados Unidos, v. 7, n. 1, 1992. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=141700&isnumber=3807>>. Acesso em: 18 jun. 2008.

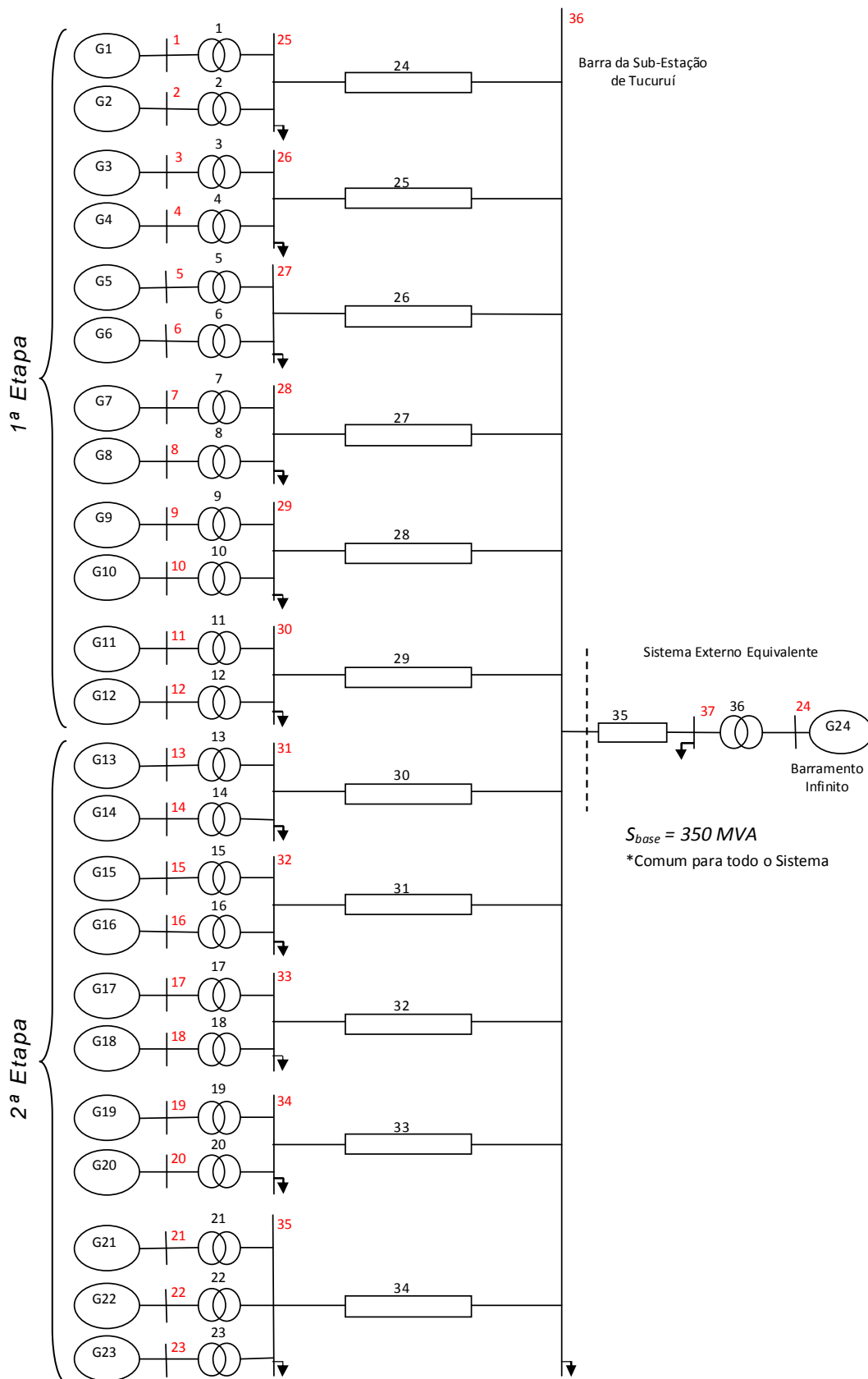
ZHOU, E. Z. Object-oriented Programming, C++ and Power System Simulation. *IEEE Transaction on Power Systems*, Estados Unidos, v. 11, n. 1, p. 206-215, 1996. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=486097&isnumber=10381>>. Acesso em: 26 set. 2008.

ZHU, Jun; JOSSMAN, Paul. Application of Design Patterns for Object-Oriented Modeling of Power Systems. *IEEE Transaction on Power Systems*, Estados Unidos, v. 14, n. 2, p. 532-537, 1999. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=761877&isnumber=16488>>. Acesso em: 25 ago. 2008.

APÊNDICES

APÊNDICE A

Diagrama unifilar das UGHs de Tucuruí interligadas ao Barramento Infinito



APÊNDICE B

Desenvolvimento dos cálculos para obtenção das equações de estado para o RAT da segunda etapa.

De acordo com os diagramas de bloco apresentados, as variáveis de estado escolhidas foram:

$$\dot{X}_1, \dot{X}_2, \dot{X}_3, \dot{X}_4, \dot{X}_5, V_{oel}, \dot{V}_Q$$

As entradas deste sistema são:

$$V_{ref}, V_{sad}, V_t, Q_e, P_e, I_{fd}, \omega, I_g;$$

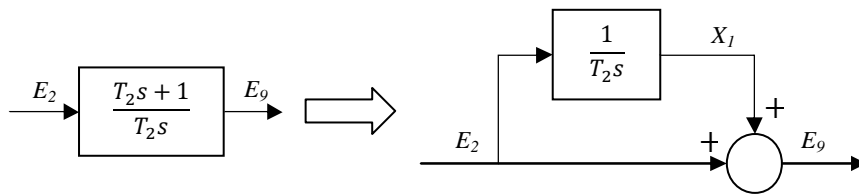
I_g é a corrente do gerador, a qual pode ser calculada internamente, sem que seja preciso uma conexão direta desta variável do gerador para recebê-la, calculando-se da seguinte forma, tendo V_T , P_e e Q_e que já são variáveis lidas:

$$I_g = \frac{V_T}{\sqrt{P_e^2 + Q_e^2}}$$

Assim, temos que:

$$E_1 = V_{sad} + V_{ref} - \dot{X}_2 - \dot{V}_q$$

Seguindo a estratégia de retirar zeros de blocos para construção das equações de estado:



Continuando o desenvolvimento:

$$E_2 \begin{cases} E_1 K_a - V_{oel}, \text{ se } E_1 K_a > V_{uel} \\ V_{uel} - V_{oel}, \text{ se } E_1 K_a < V_{uel} \end{cases}$$

$$X_1 = E_2 \frac{1}{T_2 s} \rightarrow \dot{x}_1 = \frac{1}{T_2} E_2$$

$$E_9 = X_1 + E_2$$

$$X_2 = V_T \frac{1}{T_1 s + 1} \rightarrow \dot{x}_2 = \frac{1}{T_1} [V_T - x_2]$$

$$E_{fd} = V_T [K_p \cos(\text{rampa}(E_9))]$$

$$I_q = \frac{Q_e}{V_T}$$

$$V_q = K_q I_q \frac{1}{T_{1R} s + 1} \rightarrow \dot{V}_q = \frac{1}{T_{1R}} [K_q I_q - V_q]$$

$$E_3 = I_q - I_{cap}$$

$$E_4 = -(I_q + I_{ind})$$

$$E_5 = I_g - I_{gmax}$$

$$E_{10} = \begin{cases} E_3, & \text{se } E_3 < E_4 \\ E_4, & \text{se } E_3 > E_4 \end{cases}$$

$$E_{11} = \begin{cases} E_4, & \text{se } E_4 < E_5 \\ E_5, & \text{se } E_4 > E_5 \end{cases}$$

$$X_3 = E_{10} \frac{1}{T_{1lce} \cdot s} \rightarrow \dot{x}_3 = \frac{1}{T_{1lce}} E_{10}$$

$$X_4 = E_{11} \frac{1}{T_{2lce} s} \rightarrow \dot{x}_4 = \frac{1}{T_{2lce}} E_{11}$$

$$V_g = X_4 - X_3$$

$$I_\omega = \frac{P_e}{V_T}$$

$$V_{uel} = K_{uel} [V_T (\text{pontos}(|I_\omega|)) - I_q]$$

$$E_6 = I_{fd} - I_{fmx}$$

$$X_5 = I_{fd} \frac{1}{T_{1cc} s + 1} \rightarrow \dot{x}_5 = \frac{1}{T_{1cc}} [I_{fd} - x_5]$$

$$E_7 = I_{fmx} - X_5$$

$$X_6 = E_6 \frac{1}{T_{1oel} s} \rightarrow \dot{x}_6 = \frac{1}{T_{1oel}} E_6$$

$$V_{oel} = X_6$$

$$L_{rc} = E_7 K_{rc}$$

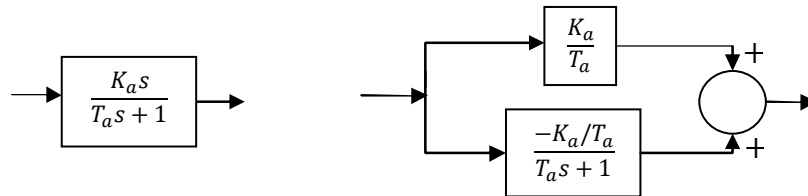
$$LVHz = \omega K_{VHz}$$

APÊNDICE C

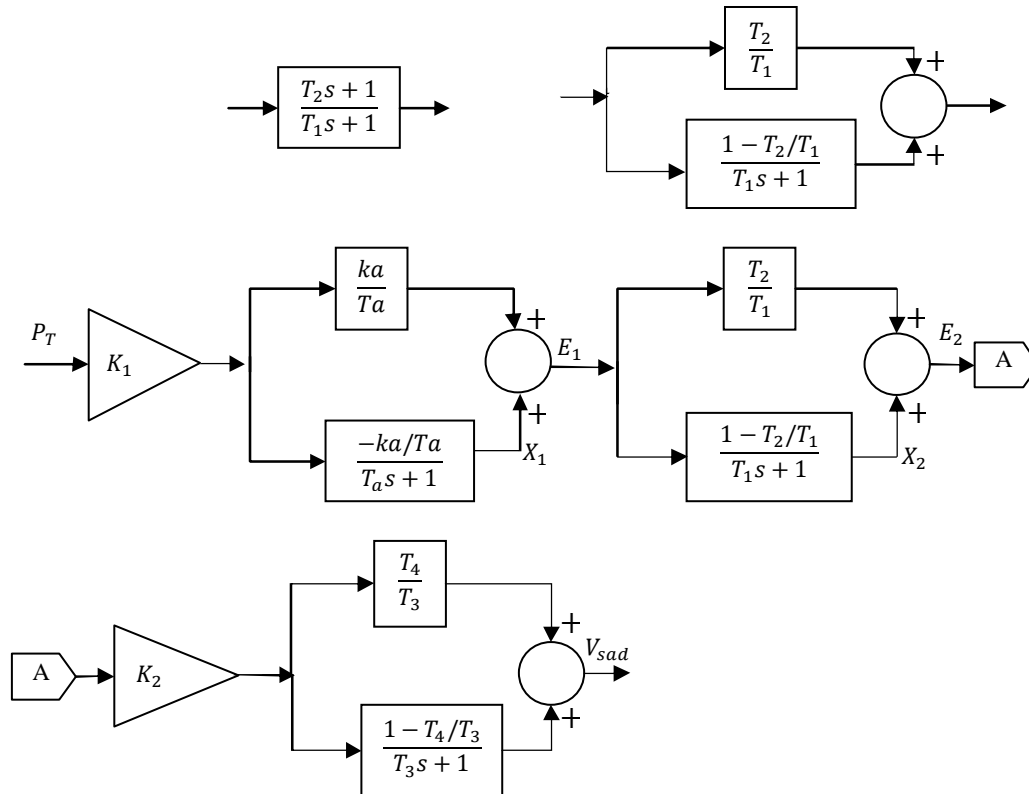
Desenvolvimento dos cálculos para obtenção das equações de estado para o ESP da primeira etapa.

Eliminação de blocos com zeros para implementação computacional das equações de estado:

Bloco I:



Bloco II:



Equações diferenciais:

$$X_1 = K_1 P_T \frac{-K_a/T_a}{T_a s + 1} \rightarrow T_a s X_1 = K_1 P_T \left[\left(\frac{-K_a}{T_a} \right) - X_1 \right]$$

$$\dot{x}_1 = \left(\frac{1}{T_a} \right) \left\{ K_1 P_T \left[\left(\frac{-K_a}{T_a} \right) - x_1 \right] \right\}$$

$$E_1 = K_1 P_T \left(\frac{K_a}{T_a} \right) + X_1$$

$$X_2 = E_1 \frac{1 - T_2/T_1}{T_1 s + 1} \rightarrow T_1 s X_2 = (1 - T_2/T_1)E_1 - X_2]$$

$$\dot{x}_2 = \left(\frac{1}{T_1}\right) [(1 - T_2/T_1)E_1 - x_2]$$

$$E_2 = \left(\frac{T_2}{T_1}\right) E_1 + X_2]$$

$$X_3 = E_2 K_2 \frac{1 - T_4/T_3}{T_3 s + 1} \rightarrow T_3 s X_3 = (1 - T_4/T_3)E_2 K_2 - X_3]$$

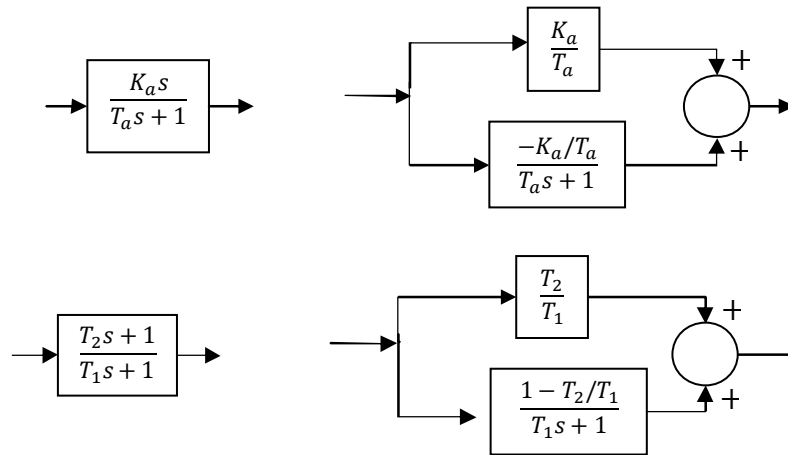
$$\dot{x}_3 = \left(\frac{1}{T_3}\right) [(1 - T_4/T_3)E_2 K_2 - x_3]$$

$$V_{\text{sad}} = E_2 K_2 \left(\frac{T_4}{T_3}\right) + X_3$$

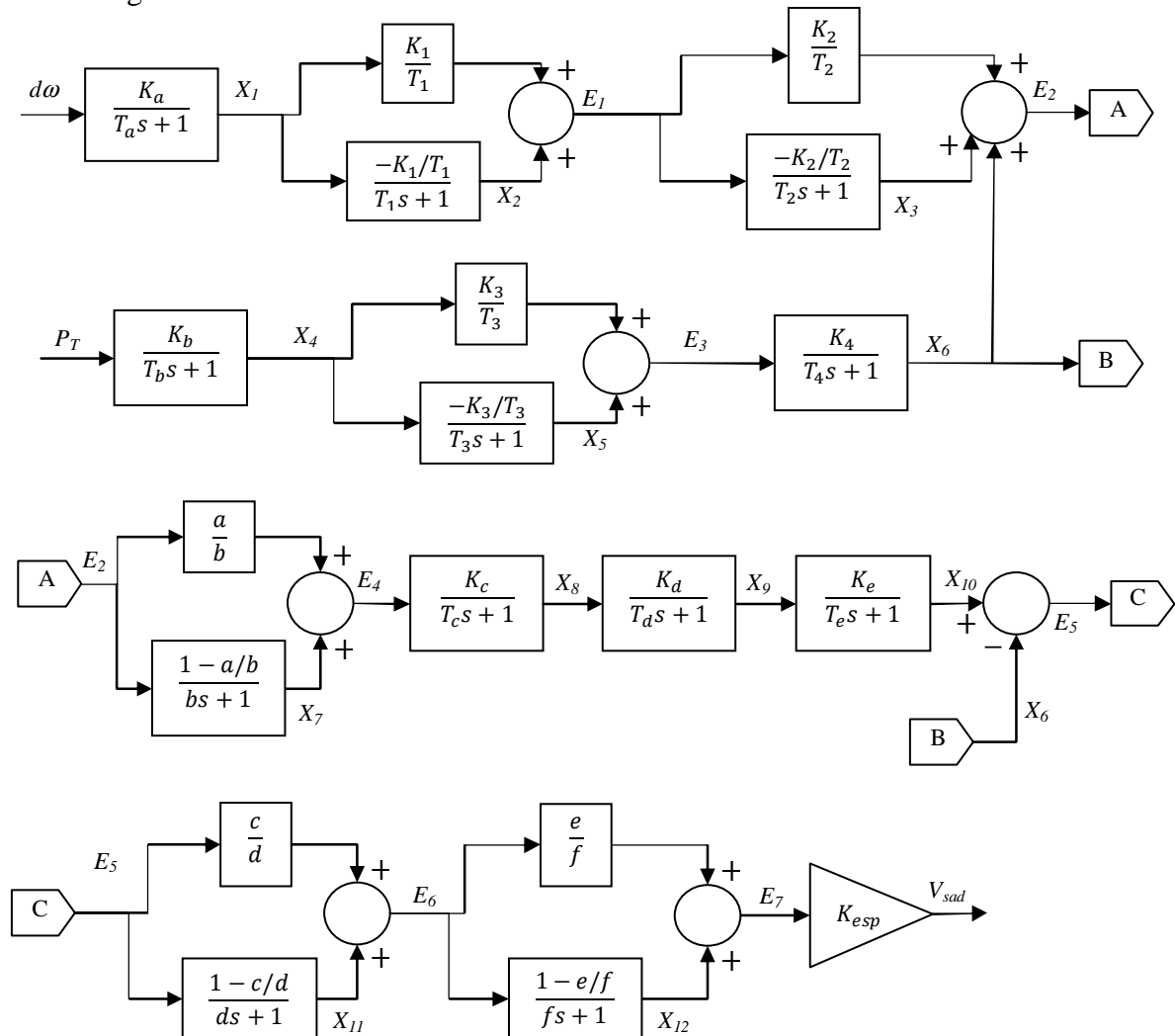
APÊNDICE D

Desenvolvimento dos cálculos para obtenção das equações de estado para o ESP da segunda etapa.

Eliminação de blocos com zeros para implementação computacional das equações de estado:



Novo diagrama de blocos:



Equações diferenciais:

$$X_1 = d\omega \frac{K_a}{T_a s + 1} \rightarrow T_a s X_1 = d\omega K_a - X_1$$

$$\dot{x}_1 = \left(\frac{1}{T_a}\right) [d\omega K_a - x_1]$$

$$X_2 = X_1 \frac{-K_1/T_1}{T_1 s + 1} \rightarrow T_1 s X_2 = X_1 \left(-\frac{K_1}{T_1}\right) - X_2$$

$$\dot{x}_2 = \left(\frac{1}{T_1}\right) \left[x_1 \left(-\frac{K_1}{T_1}\right) - x_2\right]$$

$$E_1 = x_1 \left(\frac{K_1}{T_1}\right) + x_2$$

$$X_3 = E_1 \frac{(-K_2/T_2)}{T_2 s + 1} \rightarrow T_2 s X_3 = E_1(-K_2/T_2) - X_3$$

$$\dot{x}_3 = \left(\frac{1}{T_2}\right) [e_1(-K_2/T_2) - x_3]$$

$$X_4 = P_T \frac{K_b}{T_b s + 1} \rightarrow T_b s X_4 = P_T K_b - X_4$$

$$\dot{x}_4 = \left(\frac{1}{T_b}\right) [P_T K_b - x_4]$$

$$X_5 = X_4 \frac{(-K_3/T_3)}{T_3 s + 1} \rightarrow T_3 s X_5 = X_4(-K_3/T_3) - X_5$$

$$\dot{x}_5 = \left(\frac{1}{T_3}\right) [x_4(-K_3/T_3) - x_5]$$

$$E_3 = X_4 \left(\frac{K_3}{T_3}\right) + X_5$$

$$X_6 = E_3 \frac{K_4}{T_4 s + 1} \rightarrow T_4 s X_6 = E_3 K_4 - X_6$$

$$\dot{x}_6 = \left(\frac{1}{T_4}\right) [e_3 K_4 - x_6]$$

$$E_2 = E_1 \left(\frac{K_2}{T_2}\right) + x_3 + x_6$$

$$X_7 = E_2 \frac{(1 - a/b)}{b s + 1} \rightarrow b s X_7 = E_2(1 - a/b) - X_7$$

$$\dot{x}_7 = \left(\frac{1}{b}\right) [e_2(1 - a/b) - x_7]$$

$$E_4 = E_2 \left(\frac{a}{b}\right) + x_7$$

$$X_8 = E_4 \frac{K_c}{T_c s + 1} \rightarrow T_c s X_8 = E_4 K_c - X_8$$

$$\dot{x}_8 = \left(\frac{1}{T_c} \right) [e_4 K_c - x_8]$$

$$X_9 = X_8 \frac{K_d}{T_d s + 1} \rightarrow T_d s X_9 = X_8 K_d - X_9$$

$$\dot{x}_9 = \left(\frac{1}{T_d} \right) [x_8 K_d - x_9]$$

$$X_{10} = X_9 \frac{K_e}{T_e s + 1} \rightarrow T_e s X_{10} = X_9 K_e - X_{10}$$

$$\dot{x}_{10} = \left(\frac{1}{T_e} \right) [x_9 K_e - x_{10}]$$

$$E_5 = x_{10} + x_6$$

$$X_{11} = E_5 \frac{(1 - c/d)}{d s + 1} \rightarrow d s X_{11} = E_5 \left(1 - \frac{c}{d} \right) - X_{11}$$

$$\dot{x}_{11} = \left(\frac{1}{d} \right) \left[e_5 \left(1 - \frac{c}{d} \right) - x_{11} \right]$$

$$E_6 = E_5 \left(\frac{c}{d} \right) + x_{11}$$

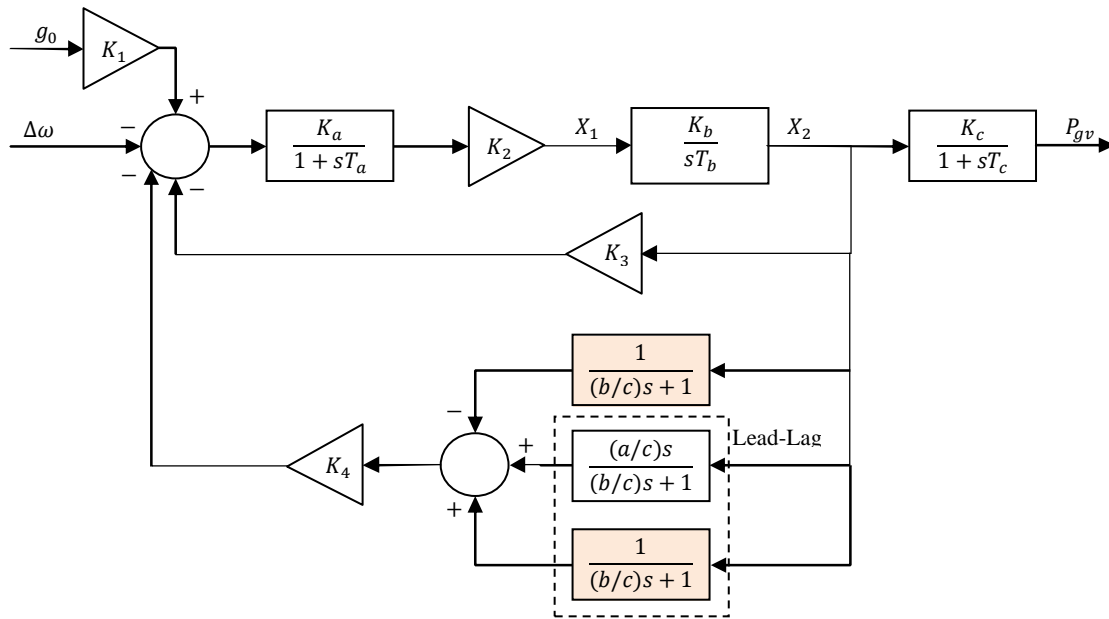
$$X_{12} = E_6 \frac{(1 - e/f)}{f s + 1} \rightarrow f s X_{12} = E_6 \left(1 - \frac{e}{f} \right) - X_{12}$$

$$\dot{x}_{12} = \left(\frac{1}{f} \right) \left[e_6 \left(1 - \frac{e}{f} \right) - x_{12} \right]$$

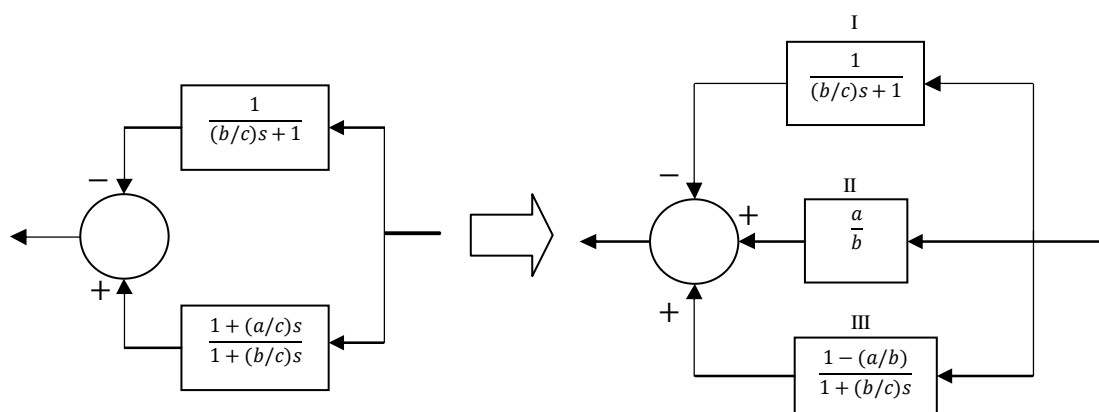
APÊNDICE E

Desenvolvimento dos cálculos para obtenção das equações de estado para o RV da primeira etapa.

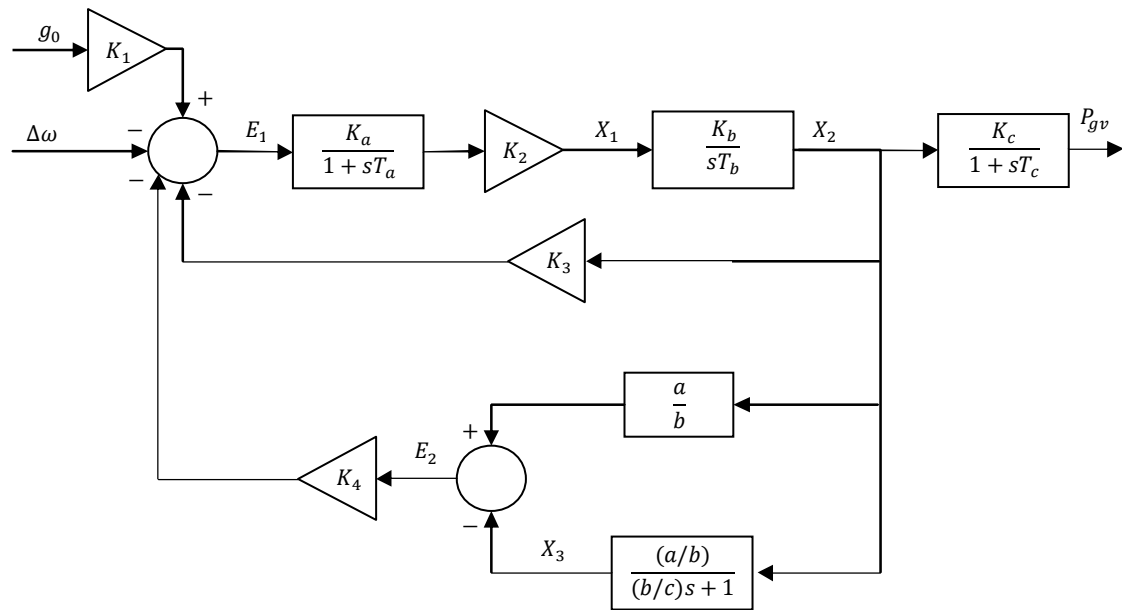
O bloco presente no ramo do estatismo transitório possui um zero na origem, devendo ser eliminado para que no cálculo das equações dinâmicas deste modelo, a variação da saída não dependa de variações do sinal de entrada neste bloco.



Dividem-se todos os membros do bloco que contem o zero pela constante c do denominador para que o termo independente seja unitário. Para forçar o aparecimento de um bloco *Lead-Lag*, soma-se à saída deste bloco, dois blocos iguais, com sinais diferentes e com o mesmo denominador deste.



Com a presença do bloco *Lead-Lag*, pode-se utilizar a uma estratégia de modelagem de modo que todos os blocos sejam de primeira ordem e sem a presença de zeros. Assim, somando os blocos I e III do diagrama anterior, tem-se:



Outra forma de implementar esta estratégia é retirando o zero do bloco $\frac{a \cdot s}{b \cdot s + c}$ através de divisão polinomial:

$$\frac{as}{bs + c} = \frac{a}{b} - \frac{ca/b}{bs + c} = \frac{(bs + c)a - (c \cdot a \cdot b)/b}{b(bs + c)} = \frac{bs \cdot a + ca - ca}{b(bs + c)} = \frac{as}{(bs + c)}$$

ANEXOS

ANEXO A⁶*Parâmetros do modelo dos geradores da primeira etapa*

Reatância síncrona, eixo quadratura	X_q	0,53	p.u.
Reatância síncrona, eixo direto	X_d	0,78	p.u.
Reatância transitória, eixo quadratura	X'_q	0,53	p.u.
Reatância transitória, eixo direto	X'_d	0,25	p.u.
Reatância subtransitória, eixo quadratura	X''_q	0,17	p.u.
Reatância subtransitória, eixo direto	X''_d	0,16	p.u.
Reatância entre rotor e estator	X_L	0,15	p.u.
Resistência de Armadura	R_a	0,003967	p.u.
Constante de tempo, eixo quadratura, transitório	T'_{q0}	0	s
Constante de tempo, eixo direto, transitório	T'_{d0}	5,38	s
Constante de tempo, eixo quadratura, subtransitório	T''_{q0}	0,0455	s
Constante de tempo, eixo direto, subtransitório	T''_{d0}	0,05	s
Constante de inércia	H	4,666	s
Coefficiente de amortecimento	D_a	0	W/rad.s

⁶ Baseado em General Electric ([19--]) e Villares Equipamentos ([19--]).

ANEXO B⁷*Parâmetros do modelo dos geradores da segunda etapa*

Reatância síncrona, eixo quadratura	X_q	0,7	p.u.
Reatância síncrona, eixo direto	X_d	1,2	p.u.
Reatância transitória, eixo quadratura	X'_q	0,7	p.u.
Reatância transitória, eixo direto	X'_d	0,3	p.u.
Reatância subtransitória, eixo quadratura	X''_q	0,22	p.u.
Reatância subtransitória, eixo direto	X''_d	0,2	p.u.
Reatância entre rotor e estator	X_L	0,12	p.u.
Resistência de Armadura	R_a	0,00293	p.u.
Constante de tempo, eixo quadratura, transitório	T'_{q0}	0	s
Constante de tempo, eixo direto, transitório	T'_{d0}	5	s
Constante de tempo, eixo quadratura, subtransitório	T''_{q0}	0,21	s
Constante de tempo, eixo direto, subtransitório	T''_{d0}	0,1	s
Constante de inércia	H	4	s
Coefficiente de amortecimento	D_a	0	W/rad.s

⁷ Baseado em Eletronorte e Alston... ([2000]).

ANEXO C⁸*Parâmetros do modelo do RAT da primeira etapa*

Aex	4,463
Bex	-2,988
Ka	0,0259
Ke	0,828
Ki	42,65
Ksad	-0,1
Ta	0,021
Te	-4,268
Tse	0,854
VAmáx	3,457
VAmín	4,53

⁸ Baseado em Operador Nacional do Sistema Elétrico (2008a).

ANEXO D⁹
Parâmetros do modelo do RAT da segunda etapa

Icap	0,1
Ifmx	1,8994
Ifmx1	2,4174
Igmax	1,12
Iind	0,1
Iwmax	1,3
Iwmin	0
Ka	7
Kp	8
Kq	0
Krc	10
Kuel	1
KVHz	1,1
T1	0,01
T1cc	0,02
T1lce	33
T1oel	5
T1R	0,012
T2	1,5
T2lce	33
Twf	0,4
VAmáx	1
VAmin	-1
VMA	1
VmaVHz	1,1
Vmax	1,1
VMI	-1
Vmin	0,9
VmiVHz	0,9

⁹ Baseado em Eletronorte (2001), Operador Nacional do Sistema Elétrico (2004, 2008a).

ANEXO E¹⁰
Parâmetros do modelo do ESP da primeira etapa

K1	5
K2	5,5
Ka	2,2
T1	0,8
T2	0,04
T3	0,4
T4	0,2
Ta	2,2

¹⁰ Baseado em Operador Nacional do Sistema Elétrico (2008a).

ANEXO F¹¹
Parâmetros do modelo do ESP da segunda etapa

a	0,4
b	0,1
c	0,09
d	0,03
e	0,09
f	0,03
K1	3
K2	3
K3	3
K4	0,375
Ka	1
Kb	1
Kc	1
Kd	1
Ke	1
Kesp	12
T1	3
T2	3
T3	3
T4	3
Ta	0,01
Tb	0,01
Tc	0,1
Td	0,1
Te	0,1

¹¹ Baseado em Operador Nacional do Sistema Elétrico (2004, 2008a).

ANEXO G¹²*Parâmetros do modelo da Turbina da primeira etapa*

Parâmetros para o modelo do Diagrama 7:

At	1
D	1
Ho	1
Tw	1,2
Unl	0,13

Parâmetros para o modelo do Diagrama 8:

At	1,2
D	1
Dt	0,5
Ho	1
Pb	1
PBmaq	350
PBtur	350
Tw	1,2
Unl	0,13

¹² Baseado em Operador Nacional do Sistema Elétrico (2003a, 2008a).

ANEXO H¹³*Parâmetros do modelo da Turbina da segunda etapa*

Parâmetros para o modelo do Diagrama 7:

At	1
D	1
Ho	1
Tw	1,54
Unl	0,18

Parâmetros para o modelo do Diagrama 8:

At	1,22
D	0
Dt	1,8
Ho	1
Pb	1
PBmaq	390
PBtur	382
Tw	1,54
Unl	0,18

¹³ Baseado em Operador Nacional do Sistema Elétrico (2006b, 2008a).

ANEXO I¹⁴*Parâmetros do modelo do RV da primeira etapa*

a	6
b	6
c	1
K1	0,05
K2	10
K3	0,05
K4	0,4
Ka	1
Kb	1
Kc	1
Posmax	1
Posmin	0
Ta	0,036
Tb	0,909
Tc	1
Velmax	0,0515
Velmin	-0.17

¹⁴ Baseado em Operador Nacional do Sistema Elétrico (2003a, 2003b, 2008a).

ANEXO J¹⁵*Parâmetros do modelo do RV da segunda etapa*

Bp	0,05
Kp	2,4
Lmax	1
Lmin	0
N	10
Td	0,67
Tfil	0,05
ti	0,016
Ti	10
tj	2,8
Velmax	0,05
Velmin	-0,05

¹⁵ Baseado em Operador Nacional do Sistema Elétrico (2006a, 2006b, 2008a).