

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Análise de sistemas de controle para adaptação em tempo
real de modelo parcial de usina nuclear a dados
experimentais

Fernando Pontes Araújo

São Paulo

2008

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Análise de sistemas de controle para adaptação em tempo
real de modelo parcial de usina nuclear a dados
experimentais

Trabalho de formatura apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do título de Graduação em Engenharia

Fernando Pontes Araújo

Orientação: Marcelo Augusto Leal Alves (USP)

Alexander Kerner (IsaR/TUM)

Rafael Macián-Juan (TUM)

São Paulo

2008

FICHA CATALOGRÁFICA

Araújo, Fernando Pontes

Análise de sistemas de controle para adaptação em tempo real de modelo parcial de usina nuclear a dados experimentais / F.P. Araújo. -- São Paulo, 2008.

p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1.Controle (Teoria de sistemas e controle) 2.Energia nuclear 3.Simulação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II.t.

RESUMO

Este trabalho baseia-se no experimento PKL-III B3.1, realizado pelo consórcio KWU/Siemens a fim de obter dados sobre situações de acidente em usinas nucleares. De posse dos resultados, implementou-se um modelo computacional no programa de simulação GRS ATHLET com a finalidade de reproduzir os dados obtidos empiricamente, objetivo não completamente atingido devido a desvios significativos, principalmente nas vizinhanças de transientes. O escopo do trabalho é, utilizando os dados obtidos experimentalmente como referência, desenvolver e analisar sistemas de controle que atuem junto ao programa de simulação, aumentando sua fidelidade.

ABSTRACT

This report is based on the PKL-III B3.1 experiment, done by the KWU/Siemens consortium to obtain data on accident situations in nuclear power plants. With this data, a computational model was implemented on GRS ATHLET simulation software, aiming to reproduce the empirically obtained results, objective that was not entirely achieved due to significant error in simulation specially around transients. The scope of this report is to develop and analyze, using the experimental data as reference, control systems that act in conjunction with the simulation software, this way improving it's fidelity.

SUMÁRIO

LISTA DE ABREVIACÕES.....	7
LISTA DE FIGURAS.....	8
1 INTRODUÇÃO.....	9
2 O EXPERIMENTO PKL.....	10
2.1 O experimento PKL-III B3.1.....	12
3 AMBIENTE DE DESENVOLVIMENTO E SIMULAÇÃO.....	13
3.1 ATHLET.....	13
3.1.1 Descrição.....	13
3.1.2 Utilização.....	14
3.2 MATLAB.....	15
3.2.1 Descrição.....	15
3.2.2 Utilização.....	15
3.2.3 Acoplamento ATHLET - MATLAB.....	16
4 MODELO E UTILIZAÇÃO DO GERADOR DE VAPOR INTACTO.....	17
5 IMPLEMENTAÇÃO DOS ALGORITMOS DE CONTROLE.....	20
6 TIPOS DE MECANISMOS DE CONTROLE.....	22
6.1 White-box.....	22
6.2 Gray-box.....	23
6.3 Black-box.....	23
7 APRESENTAÇÃO DAS CURVAS SEM ADAPTAÇÃO.....	25
8 CONTROLADOR "CAIXA PRETA" (black-box).....	27
8.1 Teoria.....	27
8.2 Apresentação dos resultados.....	29
8.2.1 1 parâmetro.....	29
8.2.2 2 parâmetros.....	32
9 CONTROLADOR "CAIXA-CINZA" (gray-box).....	38
9.1 Curva de dados.....	38
9.2 Controle heurístico.....	39
10 DETECÇÃO DO INÍCIO DO TRANSIENTE.....	41
10.1 Utilização da detecção do início do transiente.....	43
11 CONCLUSÕES E DISCUSSÃO.....	45
ANEXO A – Códigos em MATLAB.....	47
ANEXO B – Excerto de arquivo de entrada, ATHLET.....	54
REFERÊNCIAS.....	56

LISTA DE ABREVIACOES

ATHLET – Analysis of Thermal-Hydraulics of Leaks and Transients

CSA – Cross Section Area (rea de seo transversal)

GRS – Gesellschaft fr Anlagen- und Reaktorsicherheit mbh

ISaR – Institute for Safety and Reliability

P_{DC} – Presso no tubo de descida

P_{DO} – Presso no domo

PKL – Primr-Kreis-Lauf (Funcionamento de Ciclo Primrio)

SPW – Speisewassermassenstrom (vazo mssica de gua de alimentao)

LISTA DE FIGURAS

Figura 1: Usina nuclear de Phillippsburg, Alemanha.....	10
Figura 2: Esquema da instalação PKL.....	11
Figura 3: PKL-III B3.1, esquema da instalação.....	12
Figura 4: ATHLET rodando em terminal, com arquivo de entrada ao fundo.....	13
Figura 5: MATLAB e ATHLET.....	15
Figura 6: Gerador de vapor intacto.....	17
Figura 7: Comparação entre simulação e experimento.....	18
Figura 10: Diferentes métodos de controle.....	22
Figura 8: Pressão no tubo de descida, (Pa) x tempo (s) (experimento, ATHLET).....	25
Figura 9: Pressão no domo (Pa) x tempo (s) (experimento, ATHLET).....	25
Figura 11: Diagrama de blocos, controlador PID.....	27
Figura 12: Pressão no domo (Pa) x tempo (s), CSA - domo, contr. P.....	29
Figura 13: Pressão no tubo de descida (Pa) x tempo (s), CSA – descida, contr. P.....	29
Figura 14: CSA (aberto/fechado) x tempo (s), CSA – domo, contr. P.....	30
Figura 15: CSA (aberto/fechado) x tempo (s), SPW – descida, contr. P.....	30
Figura 16: CSA (aberto/fechado) x tempo (s), SPW – descida, contr. P.....	31
Figura 17: SPW (kg/s) x tempo (s), SPW – descida, contr. P.....	31
Figura 18: Pressão no domo (Pa) x tempo (s), CSA – domo, contr. P, 2 param.....	32
Figura 19: Pressão no tubo de descida (Pa) x tempo (s), SPW – descida, contr. P, 2 param.....	33
Figura 20: CSA (aberto/fechado) x tempo (s), CSA – domo, contr. P, 2 param.....	33
Figura 21: SPW (kg/s) x tempo (s), SPW – descida, contr. P, 2 param.....	34
Figura 22: Pressão no domo (Pa) x tempo (s), SPW – domo, contr. P, 2 param.....	35
Figura 23: Pressão no tubo de descida (Pa) x tempo (s), CSA – descida, contr. P, 2 param.....	35
Figura 24: SPW (kg/s) x tempo (s), SPW – domo, contr. P, 2 param.....	36
Figura 25: CSA (aberto/fechado) x tempo (s), CSA – descida, contr. P, 2 param.....	36
Figura 26: CSA (aberto/fechado) x tempo (s), comportamento ideal.....	38
Figura 27: Pressão no domo (Pa) x tempo (s), CSA – domo, curva de CSA.....	39
Figura 28: Pressão no domo (Pa) x tempo (s), CSA – domo, controle heurístico.....	40
Figura 29: Esquema ilustrativo da detecção de transiente.....	41
Figura 30: Pressão no domo (Pa) x tempo (s), SPW – domo, contr. P, 2 param, detec. trans.....	43
Figura 31: Pressão no tubo de descida (Pa) x tempo (s), CSA – descida, contr. P, 2 param, detec. Trans.....	43

1 INTRODUÇÃO

Utilizando o programa GRS ATHLET, a equipe do Institute for Safety and Reliability (localizado em Garching, distrito adjacente a Munique, Alemanha) implementou um modelo da instalação experimental PKL-III, uma usina nuclear experimental, com o objetivo de reproduzir os resultados do experimento PKL-III B3.1 (a instalação e o experimento serão descritos mais adiante neste relatório).

Por se tratar de um programa em renovação (o código se encontra desatualizado), os resultados se desviavam continuamente do esperado, sobretudo para longos intervalos de tempo e na ocorrência de transientes. Foi proposta, então, a utilização de algoritmos adicionais de controle, externos ao programa, para correção das simulações, com base nos dados experimentais disponíveis.

No contexto de uma simulação de projeto maior e mais complicado, Kerner; Schaefer; Liu (2008) [1] mostraram que a aplicação de algoritmos de controle e otimização apresenta, para o modelo completo de uma usina nuclear, problemas inerentes difíceis de resolver. Assim sendo, o escopo deste trabalho é melhorar a simulação a partir de modelo parcial, para depois analisar e comparar os diferentes métodos utilizados.

O controle realizado a partir de exemplo pronto, através do qual uma comparação direta é possível, não é de aplicação prática imediata: claro, numa usina real em situação de emergência não se dispõe de tabelas que descrevem exatamente o está por acontecer, já que acidentes reais em grande escala jamais são experimentados por razões óbvias de segurança. O objetivo maior deste e de outros trabalhos semelhantes é aumentar a compreensão sobre o programa ATHLET e suas falhas, a fim de se efetuar correções no futuro.

2 O EXPERIMENTO PKL

PKL é a sigla, em alemão, para Funcionamento de Ciclo Primário (*Primär-Kreis-Lauf, tradução livre*). Este experimento se utilizou, na prática, de um modelo em escala reduzida do reator de água pressurizada presente na instalação Phillippsburg II, Alemanha, de 4 ciclos e 1300MW, no qual apenas o ciclo primário e partes essenciais do secundário (sem o condensador e a turbina) são modelados. O consórcio KWU/Siemens foi responsável pela construção, condução e documentação de todos os testes.



Figura 1: Usina nuclear de Phillippsburg, Alemanha

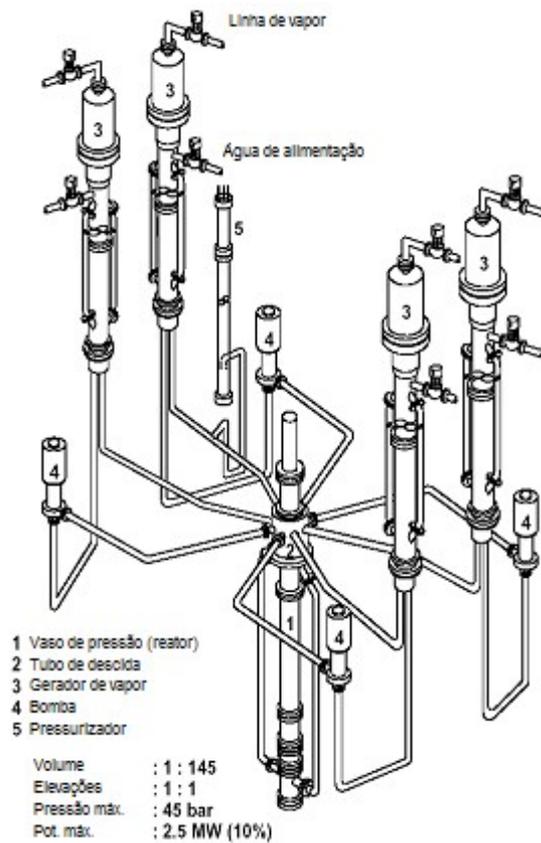


Figura 2: Esquema da instalação PKL

Os dois primeiros experimentos da série, PKL-I e PKL-II, tinham como objetivo estudar acidentes de perda de líquido refrigerante de pequenas ou grandes proporções (em inglês: *small/large break loss-of-coolant accident*, ou LOCA) e ocorreram no período de 1977 a 1986. Já o terceiro experimento da série, de 1986 a 1999, cujos dados foram utilizados neste trabalho, foi mais abrangente e teve como objetivo a análise termo-hidráulica dos ciclos primário e secundário, com ou sem perda de líquido refrigerante, e a conseqüente investigação de transientes.

Os transientes numa usina nuclear se caracterizam por mudanças repentinas e bruscas na potência gerada, resultante de mudanças de outras variáveis como por exemplo a pressão do vapor d'água nos geradores de vapor, abertura/fechamento de válvulas ou registros ou variação da vazão de água de alimentação (também no gerador de vapor).

2.1 O experimento PKL-III B3.1

O experimento PKL-III B3.1 consistiu especificamente na verificação da possibilidade de se desligar o reator apenas por meio do fluxo natural de líquido refrigerante (no caso, água) em apenas um dos geradores de vapor, após falha e posterior isolamento do outro.

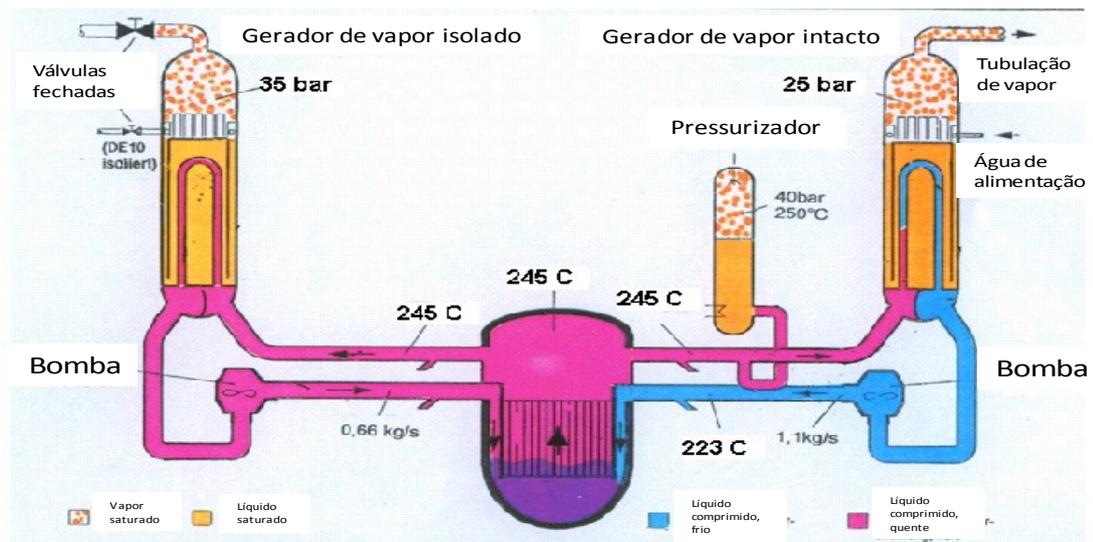


Figura 3: PKL-III B3.1, esquema da instalação

A instalação, representada na figura acima, é composta de um ciclo primário simplificado acoplado a dois geradores de vapor, um deles isolado, ou seja, com a saída de vapor e entrada de água de alimentação fechadas devido a algum tipo de acidente ou mal-funcionamento do mesmo ou dos circuitos de alimentação associados.

3 AMBIENTE DE DESENVOLVIMENTO E SIMULAÇÃO

3.1 ATHLET

3.1.1 Descrição

O ATHLET (*Analysis of THERmal-hydraulics of LEaks and Transients*, análise termo-hidráulica de vazamentos e transients, em tradução livre) é um software de simulação desenvolvido pela GRS, *Gesellschaft für Anlagen- und Reaktorsicherheit mbh*.

O software funciona por meio de módulos: dinâmica dos fluidos e termodinâmica (*thermo-fluiddynamics*, TFD); transferência e condução de calor (*heat transfer and heat conduction*, HECU); neutron-cinética (*neutron kinetics*, NEUKIN); código de simulação de controle geral (*general control simulation code*, GCSM); e o método de integração numérica (FEBE). Tais módulos possibilitam a descrição da geometria, condições iniciais, de contorno e limites físicos de uma instalação nuclear, assim como todas as ações de controle e segurança resultantes de sistemas presentes numa usina real, para posterior simulação e análise de parâmetros previamente escolhidos.

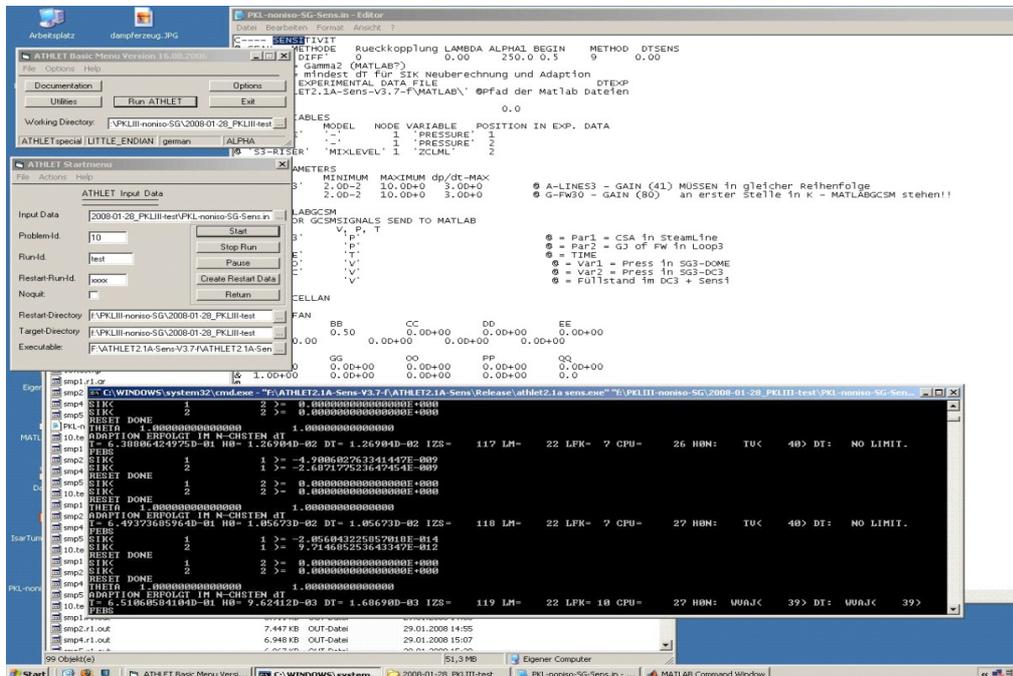


Figura 4: ATHLET rodando em terminal, com arquivo de entrada ao fundo

3.1.2 Utilização

A interface com o usuário se dá através de um arquivo de entrada, meramente um arquivo de texto com formatação específica. Com o modelo pronto, é possível gerar desenhos e esquemas da instalação para verificação e ilustração, e de posse de dados de simulação, o programa oferece ainda ferramentas para plotar cada parâmetro e variável, a fim de gerar documentação ao trabalho.

O usuário se utiliza dos módulos (transferência de calor, ações de controle, etc) para descrever o modelo no arquivo de entrada com o nível de detalhamento desejado. É possível adotar valores fixos, tabelas e formulações mais complexas para os coeficientes relacionados aos fenômenos físicos que ocorrem na instalação.

No arquivo de entrada devem também ser definidos os parâmetros que poderão ser modificados externamente e realimentados na simulação (com ajuda do MATLAB, por exemplo), assim como aqueles que serão apenas disponibilizados para leitura. Ambos os tipos de parâmetros encontram-se listados no Anexo B, um excerto do arquivo de entrada utilizado, para referência.

A confecção do arquivo para as simulações utilizadas não faz parte do escopo deste trabalho. Ao longo do desenvolvimento, ele nunca foi alterado.

Uma vez iniciada a simulação, não é possível interferir nos resultados por meio somente do ATHLET. Uma janela de terminal é aberta e algumas estatísticas numéricas dos passos são exibidas, e é possível pausar/abortar, como num processo comum.

Conforme especificado no arquivo de entrada, o intervalo de tempo entre pontos simulados (*steps*) é de 4 segundos, o que pode parecer um pouco espaçado demais, porém é adequado tendo em vista a ordem de grandeza do período simulado (mais de 20000 segundos).

3.2 MATLAB

3.2.1 Descrição

O MATLAB é um ambiente de programação voltado para a solução de problemas matemáticos e desenvolvimento de sistemas de controle, produzido pela norte americana Mathworks.

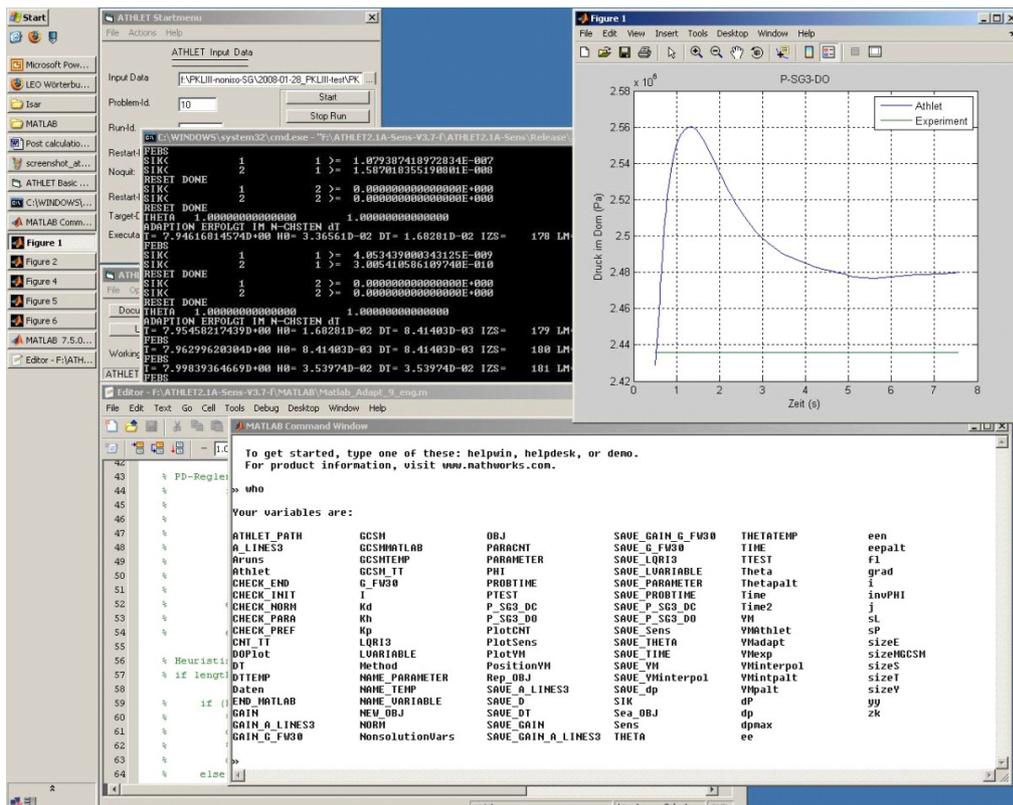


Figura 5: MATLAB e ATHLET

3.2.2 Utilização

O programa foi utilizado para integrar os dados experimentais à análise, implementar controladores e avaliar os resultados.

Por oferecer uma linguagem de programação interpretada, o MATLAB é bastante versátil. Pode-se escrever programas linha a linha e experimentá-las em tempo real, ou então utilizar programas já previamente escritos mas que são lidos da mesma maneira. Desta forma é possível verificar o valor de variáveis a qualquer instante, assim como criar novas variáveis e modificar os programas sem ter de recompilar toda a rotina. Encontrar e corrigir eventuais erros é significativamente mais rápido do que seria numa linguagem compilada, como o C.

3.2.3 Acoplamento ATHLET - MATLAB

O ATHLET e o MATLAB foram, num primeiro momento, acoplados através de um arquivo de texto. A cada passo de uma simulação, existe a opção de gerar um arquivo de texto com todos os parâmetros calculados para o instante de tempo em questão. Este mesmo arquivo é lido no início de cada passo, afim de obter as informações necessárias para o próximo cálculo. Um código em FORTRAN tomava o arquivo, o renomeava e tornava os parâmetros disponíveis ao usuário no MATLAB. Depois de feitas as modificações desejadas, o arquivo era então novamente renomeado e os parâmetros modificados, lidos pelo ATHLET e levados em conta para o cálculo do próximo passo.

Tal interface era totalmente funcional, porém muito lenta: o ato de escrever e acessar arquivos no disco rígido fazia com que mesmo códigos muito simples do MATLAB tornassem uma simulação demorada demais. Assim, o princípio foi mantido, porém ao invés de arquivos de texto no disco rígido, foi trabalhado um método de manipulação de memória RAM, definitivamente mais veloz.

4 MODELO E UTILIZAÇÃO DO GERADOR DE VAPOR INTACTO

Para desenvolvimento do sistema de controle foi tomado apenas o gerador de vapor intacto (ou seja, não isolado):

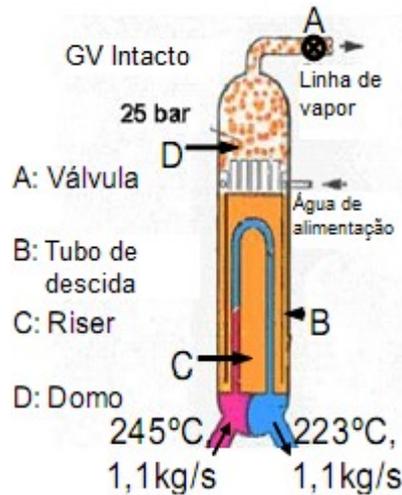


Figura 6: Gerador de vapor intacto

A água do primário, vinda do reator, entra por baixo, à esquerda (vermelho), troca calor com a água de alimentação presente no gerador de vapor e é encaminhada de volta ao reator, embaixo, à direita (azul). A água de alimentação entra no gerador de vapor pela parte superior do mesmo, seguindo por uma tubulação lateral (tubo de descida ou *downcomer*, B) até a base. O vapor produzido sobe pelo riser (tubo de subida) (C), passa pelo domo (D) e segue para a turbina depois de passar pelo registro de controle (A).

Observando-se apenas este gerador de vapor e as variações de pressão que nele ocorrem durante o processo de desligamento do reator, obtém-se uma representação fiel do atual estado do reator: a temperatura do núcleo influi diretamente a quantidade de vapor produzida, por conseguinte as pressões medidas no domo (D) e tubo de descida (B). Por isso, todos os algoritmos de controle foram desenvolvidos comparando valores reais (ou seja, experimentais) de pressão com valores simulados, sempre para o mesmo gerador de vapor isolado. Esclarecendo de outra forma: *o objetivo deste trabalho é fazer com que as curvas de pressão x tempo simuladas sejam tão próximas quanto possível das curvas experimentalmente determinadas quando da execução do teste PKL-III B3.1.*

Os parâmetros utilizados para composição dos sistemas de controle foram os seguintes:

- abertura do registro de controle de vazão do vapor (aberto/fechado, um número entre 0, totalmente fechado, e 1, totalmente aberta), CSA (*cross-section area*);
- vazão mássica de água de alimentação, SPW (do alemão *Speisewassermassenstrom*) (kg/s),

que são os dois parâmetros mais sensíveis para modular a pressão, de acordo com Kerner; Schaefer; Liu (2008) [1].

As variáveis a serem adaptadas:

- pressão da água no tubo de descida, P_{DC} (Pa);
- pressão do vapor no domo, P_{DO} (Pa).

Como já mencionado anteriormente, a simulação em intervalos de tempo longos e nas vizinhanças de transientes acumula erros e se distancia dos resultados obtidos experimentalmente, o que compromete sua validade.

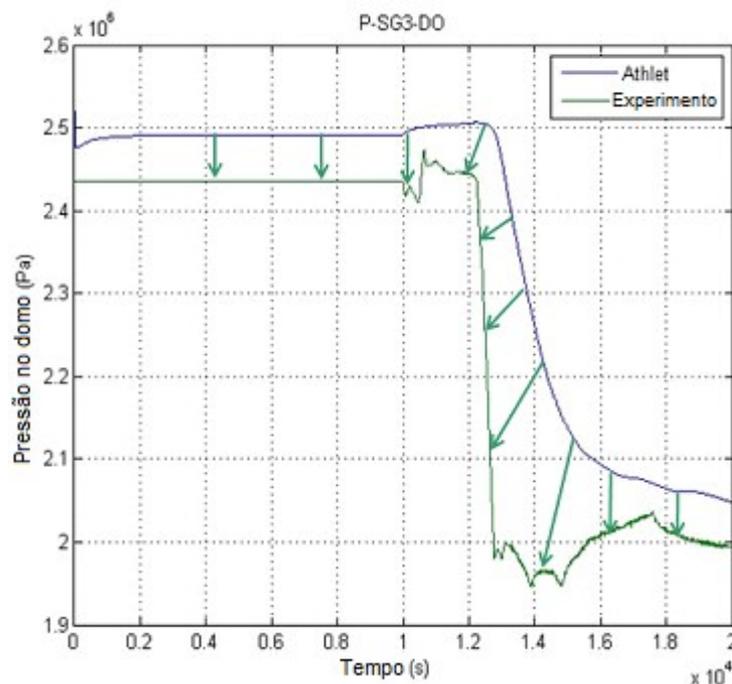


Figura 7: Comparação entre simulação e experimento

A figura acima mostra a discrepância entre simulação (linha azul, acima) e experimento (linha verde, abaixo), observando-se a pressão no domo do gerador de vapor intacto. O comportamento da simulação não é de todo ruim, já que com uma

simples translação os resultados seriam bastante condizentes com a realidade. Muitas oscilações, porém, não aparecem de maneira alguma na simulação, sobretudo: a acentuada inclinação da pressão no domo (também no tubo de descida), no início do transiente; e o valor mínimo, entre 12000 e 14000 segundos.

Os diferentes sistemas de controle implementados com ajuda do MATLAB têm o propósito de corrigir tais discrepâncias.

5 IMPLEMENTAÇÃO DOS ALGORITMOS DE CONTROLE

Pela própria arquitetura do ATHLET, variáveis de simulação não podem ser diretamente alteradas. Não é possível tomar, por exemplo, a vazão de água de alimentação do gerador de vapor e substituí-la diretamente por outro valor. Para realizar alguma alteração (mais especificamente, uma *ação de controle*) a única possibilidade é alterar o parâmetro THETA.

THETA é simplesmente um vetor de tamanho igual ao número de parâmetros modificáveis. Cada elemento do vetor tem relação direta com um parâmetro; ao início de cada passo da simulação, os valores correntes de cada parâmetro são multiplicados pelos THETAs correspondentes. Por exemplo, para registro de controle de vazão, CSA:

$$CSA_{t+1} = CSA_t * THETA(1) \quad (1)$$

Ou seja: THETA é o vetor modificador, porém apenas de parâmetros possíveis de se modificar em tempo real como a abertura do registro (mostrado em (1)) ou a vazão de água de alimentação (exemplo análogo).

Para o modelo utilizado neste trabalho, o vetor THETA é de dimensão 2, sendo que o primeiro elemento é o multiplicador de CSA, e o segundo, de SPW. Inicialmente (sem modificações), o vetor é o seguinte:

$$THETA = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (2)$$

Ou seja, sem a interferência de rotina de controle externa ao ATHLET, os valores iniciais de CSA e SPW, especificados no arquivo de entrada, serão mantidos até o final da simulação:

CSA = 0,116; SPW = 0,044 kg/s.

Como, por limitações físicas:

$$0 < CSA < 1 \quad (3)$$

$$0 < SPW < 0.1 \quad (4)$$

Utilizando (1):

$$0 < THETA(1) < 8,6207 \quad (5)$$

Analogamente:

$$0 < THETA(2) < 2,2609 \quad (6)$$

Assim, assegura-se que os limites físicos da vazão de água de alimentação, e de abertura/fechamento do registro de vapor nunca serão ultrapassados.

6 TIPOS DE MECANISMOS DE CONTROLE

Tipo \ Propriedades	white-box	gray-box	black-box
Conhecimento	Modelo completo	Parcial, ex: aprox. linear	Apenas entrada/saída
Benefícios esperados	↑	→	↓
Esforço de desenvolvimento	↑	Flexível (depende do objetivo)	↓
Exemplo	Observador, filtro Kalman, engenharia reversa	Controle heurístico	Controlador PID

Figura 10: Diferentes métodos de controle

Existem basicamente três divisões entre as regras de controle: *white-box* ("caixa branca"), *gray-box* ("caixa cinza") e *black-box* ("caixa preta"). Elas se diferenciam pelo grau de conhecimento necessário sobre o sistema a ser controlado e pela conseqüente complexidade do modelo matemático do controlador.

A seguir, cada categoria será brevemente descrita.

6.1 White-box

O nome é exatamente o oposto do *black-box* ("caixa preta"). Enquanto o outro é caracterizado pelo completo desconhecimento do que ocorre de fato no sistema, sendo lidas apenas entradas e saídas, o primeiro é exatamente o contrário: a partir de conhecendo profundo do processo a ser controlado, contrói-se um modelo matemático o mais detalhado possível a fim de tornar possível prever com razoável precisão o comportamento real do sistema.

Vantagens

Controle preciso, possibilidade de prever variáveis.

Desvantagens

Grande esforço de desenvolvimento; alto custo.

Exemplos

Observador de estados, filtro Kalman, engenharia reversa.

6.2 Gray-box

O nome ("caixa cinza"), diz respeito a um meio termo entre a "caixa preta" e a "caixa branca". Na prática, significa que não se possui (ou não se deseja incluir) conhecimento completo sobre o funcionamento do sistema; utiliza-se apenas algumas informações empíricas para produzir as ações de controle.

Vantagens

Esforço de desenvolvimento flexível; se bem projetado, produz resultados comparáveis ao white-box; custo médio.

Desvantagens

Dificuldade de incorporar conhecimento parcial a controladores "burros"; pode não ser suficiente para sistemas complexos.

Exemplo

Controle heurístico.

6.3 Black-box

A "caixa preta" ignora tudo o que acontece no processo/sistema que não sejam entradas ou saídas. As regras de controle são formuladas a partir apenas desses dados, a fim de regular o sistema em torno de um ponto de referência ou seguir uma trajetória.

Vantagens

Rápido desenvolvimento; projeto simples; baixo custo.

Desvantagens

Não apropriado para sistemas complexos; pode levar a ações de controle impossíveis de serem realizadas, devido aos limites físicos do processo/sistema.

Exemplo

Controlador PID.

Os algoritmos desenvolvidos neste trabalho incluem controladores dos tipos *gray-box* e *black-box*. O desenvolvimento de um observador de estados (*white-box*) foi tema de trabalho semelhante, também desenvolvido no *Institute for Safety and Reliability (ISaR)* por Ceuca (2008) [2].

7 APRESENTAÇÃO DAS CURVAS SEM ADAPTAÇÃO

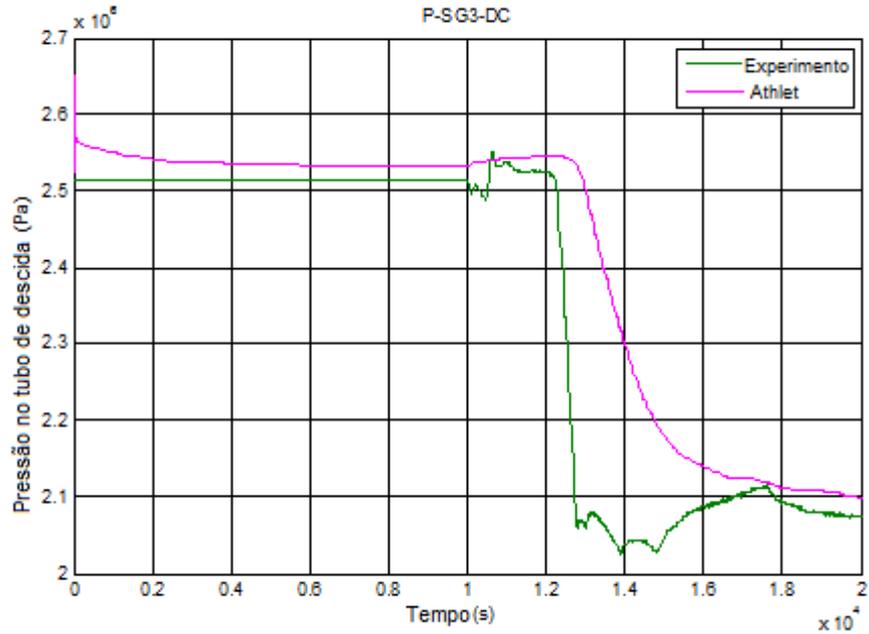


Figura 8: Pressão no tubo de descida, (Pa) x tempo (s) (experimento, ATHLET)

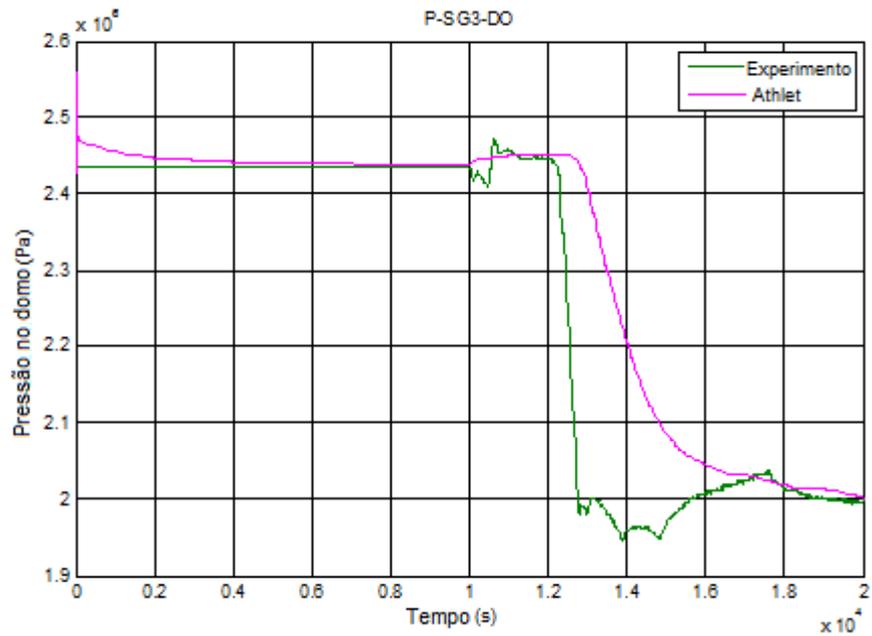


Figura 9: Pressão no domo (Pa) x tempo (s) (experimento, ATHLET)

Os resultados de simulações no ATHLET sem nenhuma correção por meio do MATLAB apresentam pouca fidelidade aos dados experimentais. A discrepância mais grave é o não acompanhamento da inclinação do gráfico da pressão, no início do transiente.

Importante observar que a pressão no tubo de descida e a no domo possuem praticamente a mesma curva, uma deslocada em relação à outra, o que prova que as perdas de carga sofridas dentro do próprio gerador de vapor são aproximadamente constantes.

As curvas aqui apresentadas, por serem o resultado "bruto" antes de qualquer trabalho realizado com o MATLAB, são o principal parâmetro de desempenho dos controladores desenvolvidos.

8 CONTROLADOR "CAIXA PRETA" (*black-box*)

8.1 Teoria

O desenvolvimento dos algoritmos partiu da análise da pressão experimental como entrada e da pressão resultante da simulação como saída, num instante de tempo.

O mecanismo de retroalimentação (*feedback*) é alimentado pelo erro entre essas duas grandezas. Assim, os dados da simulação são continuamente corrigidos para se aproximarem o máximo possível do experimento.

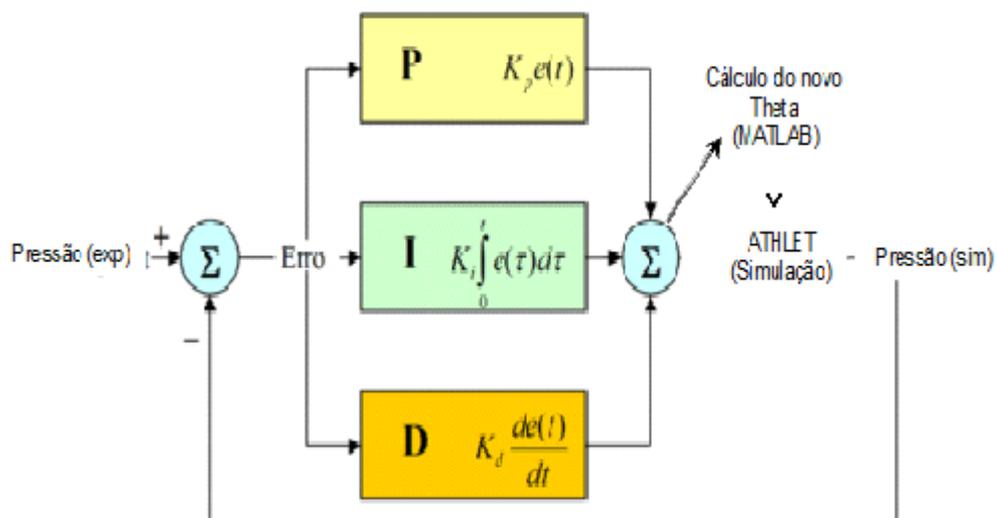


Figura 11: Diagrama de blocos, controlador PID

O valor experimental de pressão (P_{DO} ou P_{DC}) é comparado com o valor correspondente, simulado pelo ATHLET. O erro é calculado e utilizado para a determinação de um novo coeficiente THETA (em outras palavras: para modificar a abertura do registro CSA ou da vazão mássica SPW), valor que será lido pelo ATHLET no cálculo do próximo passo.

Embora o esquema mostre a composição de um controlador PID, na prática foi utilizado apenas um controlador P. Testes preliminares confirmaram a efetividade de se utilizar apenas correção proporcional: controladores mais elaborados com correção integral e diferencial adicionavam complexidade à sintonia e não produziam resposta substancialmente melhor ou mais rápida. Como o controle realizado pelas rotinas em MATLAB não ocorre de forma direta (é realimentado no ATHLET para novos cálculos), existe amortecimento e adequação física da pressão resultante das

alterações feitas em CSA e SPW, inerentes ao modelo matemático do ATHLET.

Foram realizadas primeiramente simulações adaptando apenas um parâmetro (CSA ou SPW), e depois adaptando ambos parâmetros ao mesmo tempo, cada um responsável por corrigir P_{DO} ou P_{DC} separadamente, porém. Caso contrário, as ações de controle seriam conflitantes.

Para referência, as rotinas correspondentes em MATLAB se encontram no anexo A.

8.2 Apresentação dos resultados

Os resultados serão apresentados e descritos de acordo com o número de parâmetros adaptados simultaneamente e método de controle utilizado.

8.2.1 1 parâmetro

- Parâmetro adaptado: CSA
- Variável de comparação: P_{DO}

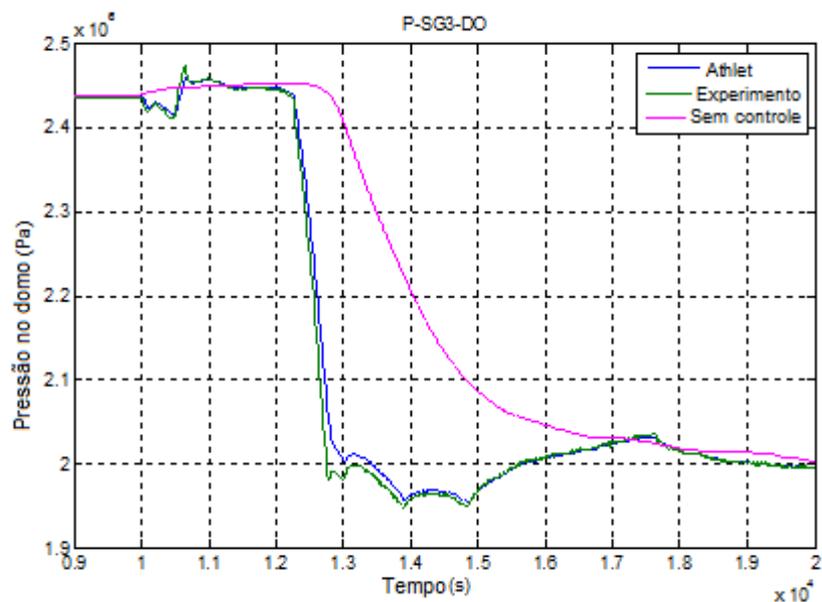


Figura 12: Pressão no domo (Pa) x tempo (s), CSA - domo, contr. P

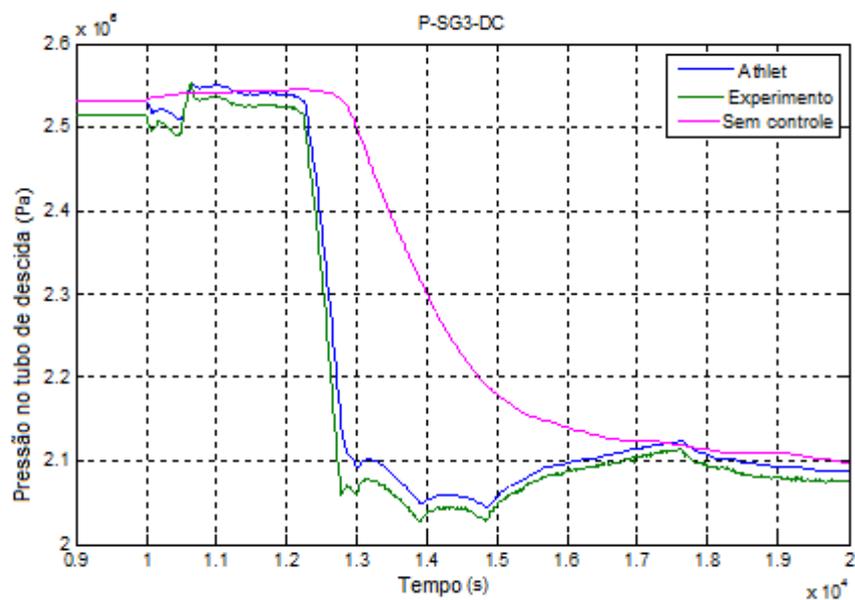


Figura 13: Pressão no tubo de descida (Pa) x tempo (s), CSA - descida, contr. P

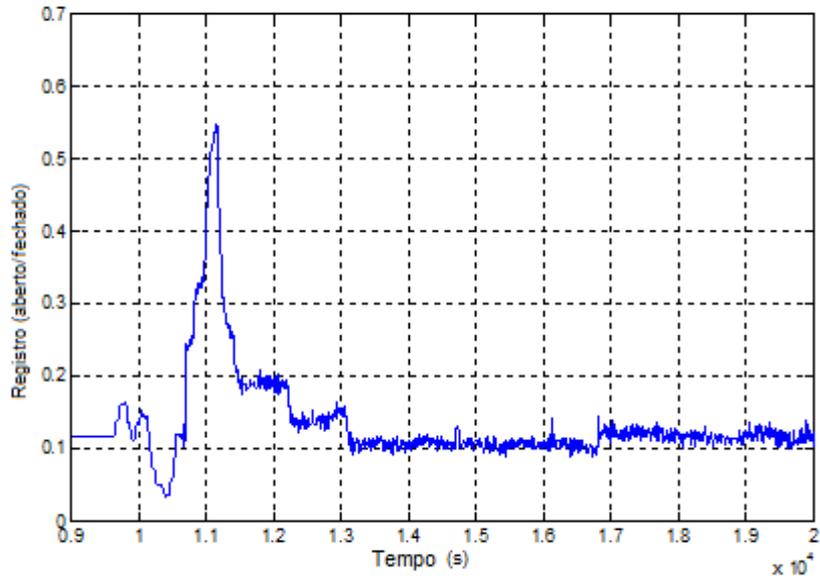


Figura 14: CSA (aberto/fechado) x tempo (s), CSA – domo, contr. P

O gráfico de SPW (kg/s) x tempo foi omitido, pois, no caso, $SPW = 0,044 \text{ kg/s}$ (constante).

- Parâmetro adaptado: SPW
- Variável de comparação: P_{DC}

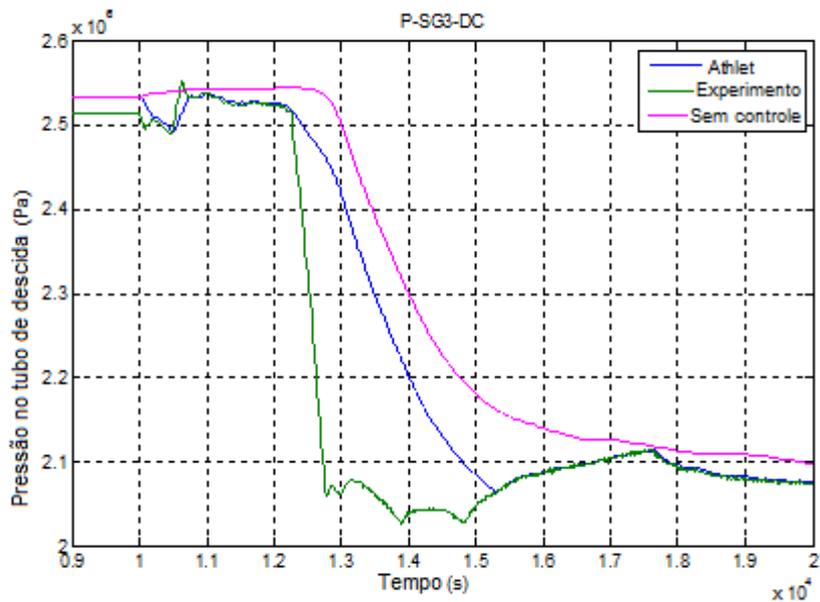


Figura 15: CSA (aberto/fechado) x tempo (s), SPW – descida, contr. P

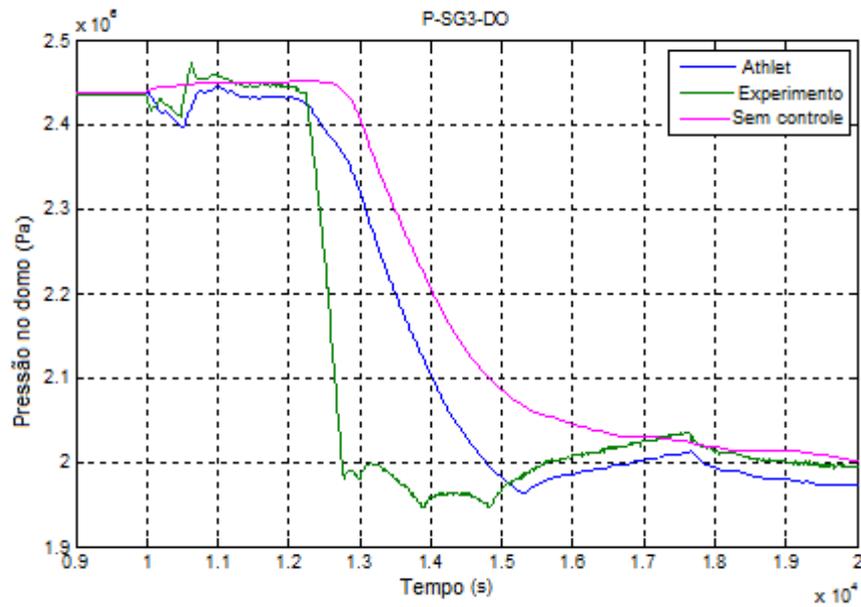


Figura 16: CSA (aberto/fechado) x tempo (s), SPW – descida, contr. P

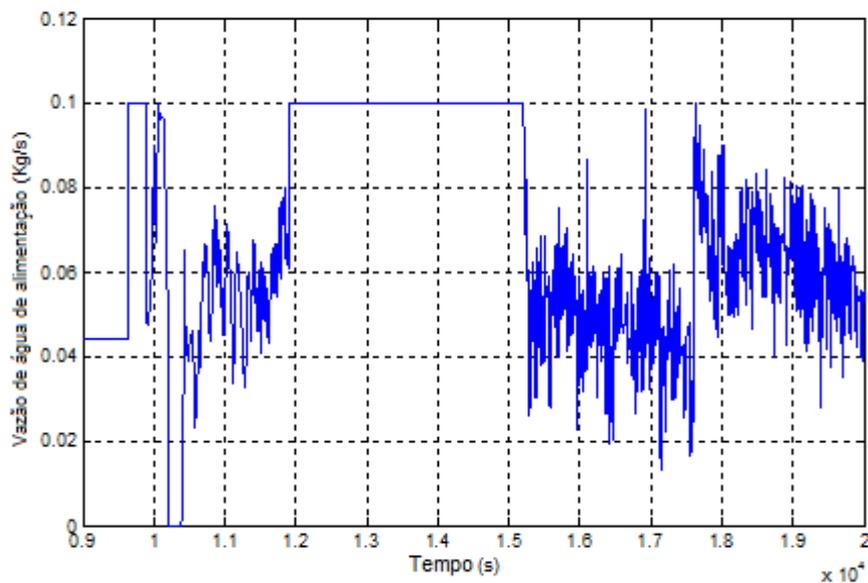


Figura 17: SPW (kg/s) x tempo (s), SPW – descida, contr. P

O gráfico de CSA (aberto/fechado) x tempo foi omitido, pois, no caso, $CSA = 0,116$ (constante).

Análise

Comprovou-se a pouca sensibilidade do par SPW - P_{DC} , e exatamente o contrário no par CSA - P_{DO} . Não só os gráficos de pressão x tempo provam esta constatação, como também os gráficos da variação dos parâmetros x tempo.

Enquanto o registro CSA foi capaz de produzir uma aproximação muito boa de P_{DO} aos valores experimentais, e nunca atingiu os limites físicos (estabelecidos na eq. (3)), a vazão mássica SPW se mostrou um parâmetro muito pouco sensível e também pouco relevante para a correção de P_{DC} . O limite superior de 0,1kg/s foi atingidos constantemente durante a simulação, e mesmo assim a vazão de água de alimentação não foi capaz de fazer com que P_{DC} se aproximasse dos valores experimentais no estágio mais crítico, ou seja, a partir do início do transiente.

8.2.2 2 parâmetros

Para controlar dois parâmetros ao mesmo tempo, fez-se simplesmente com que dois controladores P controlassem uma variável cada um, modificando um parâmetro correspondente (2 possibilidades):

- CSA modula P_{DO} , SPW modula P_{DC} (a);
- CSA modula P_{DC} , SPW modula P_{DO} (b).

a) CSA - P_{DO} ; SPW - P_{DC}

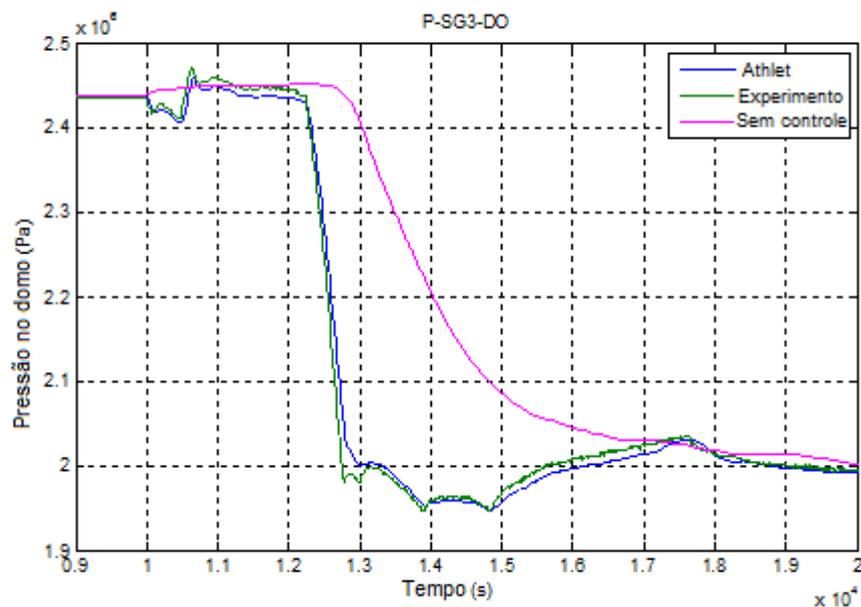


Figura 18: Pressão no domo (Pa) x tempo (s), CSA – domo, contr. P, 2 param.

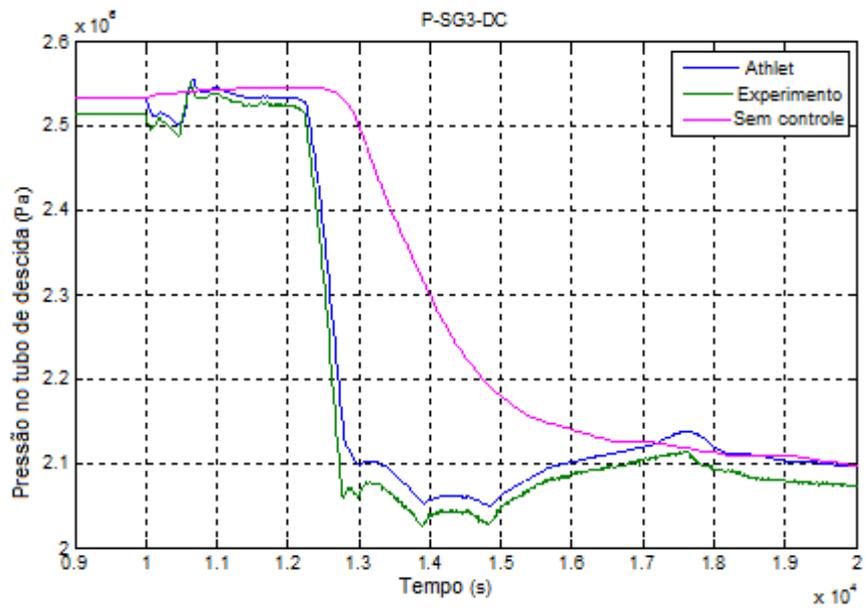


Figura 19: Pressão no tubo de descida (Pa) x tempo (s), SPW – descida, contr. P, 2 param.

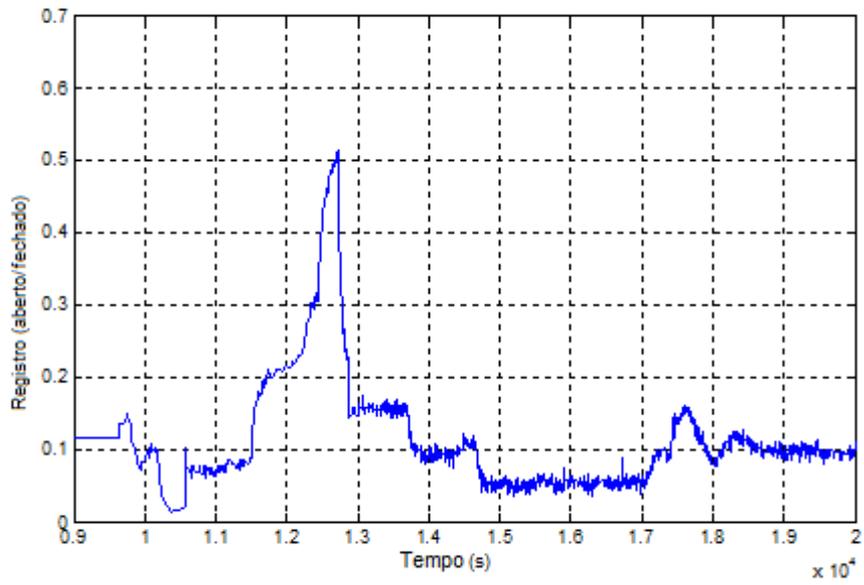


Figura 20: CSA (aberto/fechado) x tempo (s), CSA – domo, contr. P, 2 param.

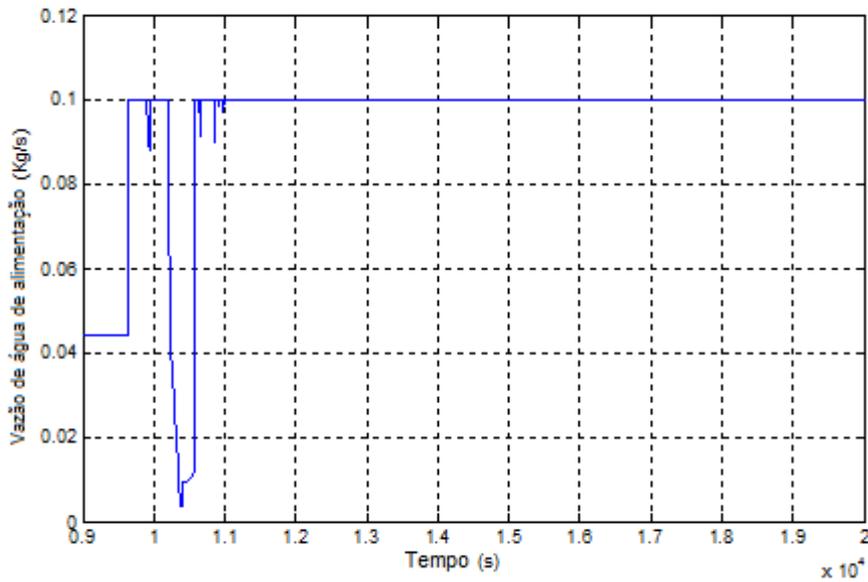


Figura 21: SPW (kg/s) x tempo (s), SPW – descida, contr. P, 2 param.

Análise

Como esperado, os resultados são praticamente iguais aos obtidos pela adaptação apenas de CSA e P_{DO} como variável de comparação. A variação de SPW para correção de P_{DC} confirmou-se igualmente ineficaz, dado que o valor ideal que permitiria ao controlador fazer as alterações necessárias está muito acima de 0,1 kg/s, o limite superior possível de ser bombeado para dentro do gerador de vapor. Testou-se a remoção do limite superior, o que resultou em falha na simulação: valores sem sentido físico causam divergência dos cálculos numéricos no ATHLET.

b) CSA - P_{DC} ; SPW - P_{DO}

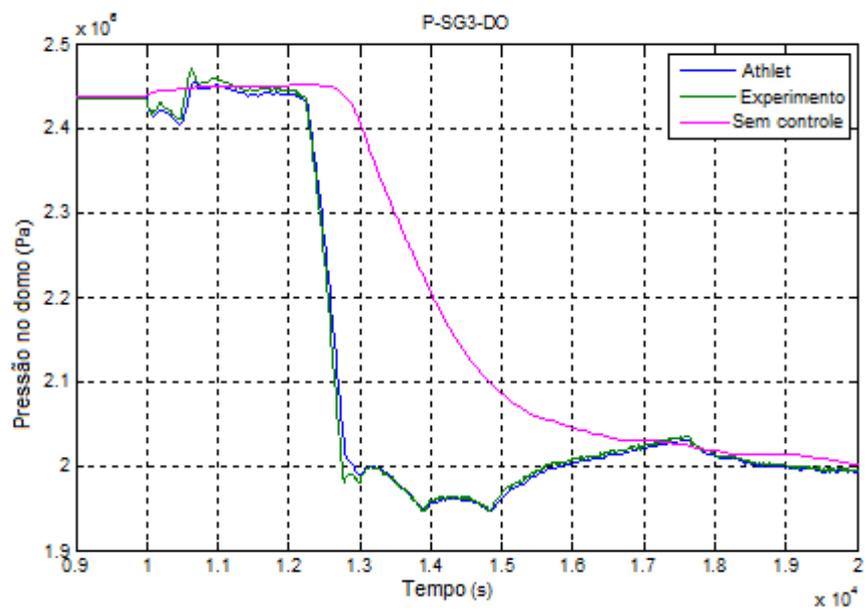


Figura 22: Pressão no domo (Pa) x tempo (s), SPW – domo, contr. P, 2 param.

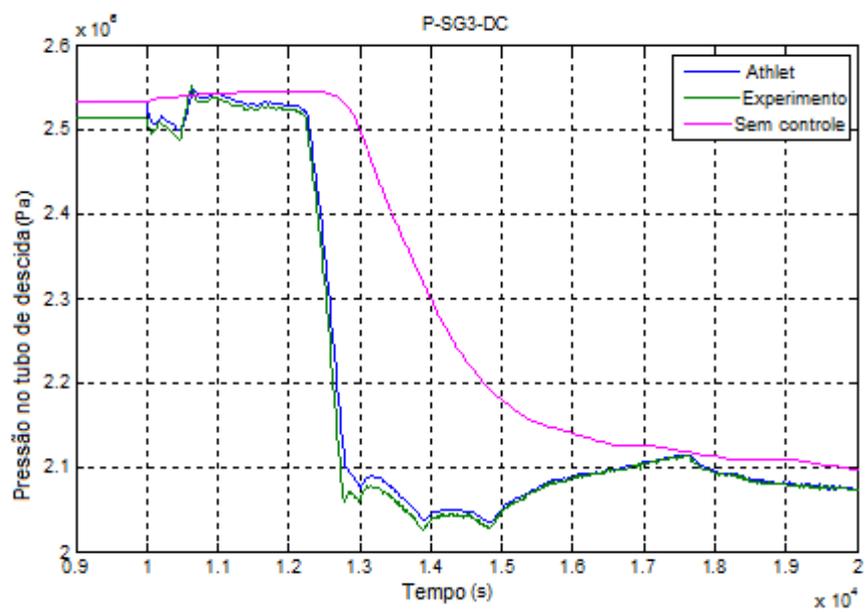


Figura 23: Pressão no tubo de descida (Pa) x tempo (s), CSA – descida, contr. P, 2 param.

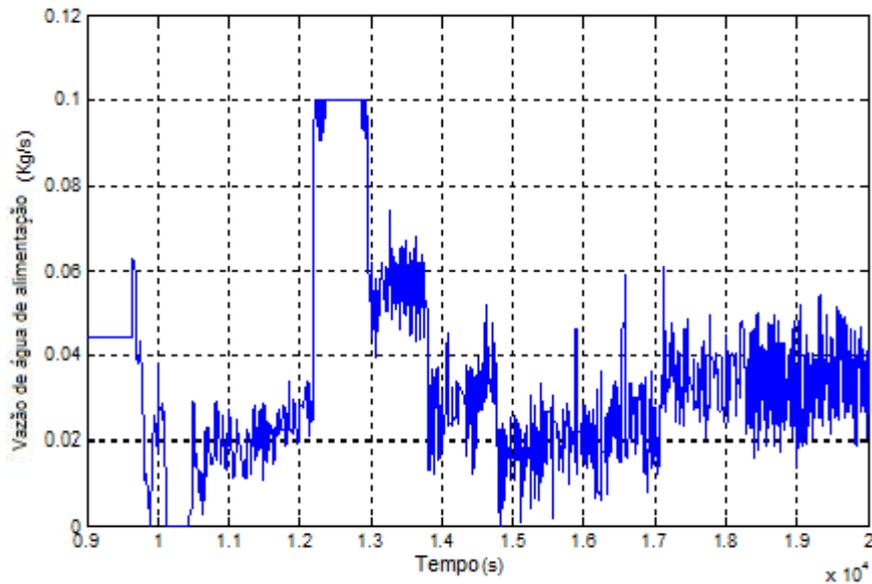


Figura 24: SPW (kg/s) x tempo (s), SPW – domo, contr. P, 2 param.

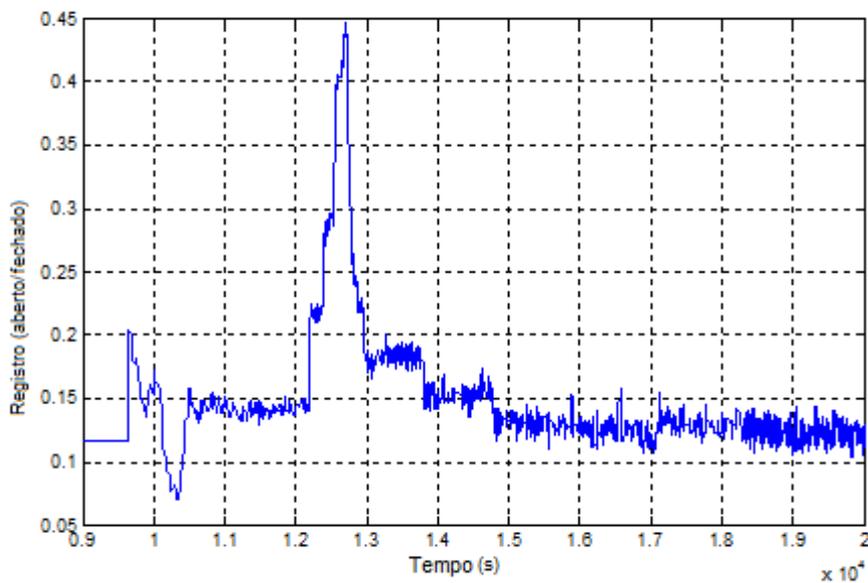


Figura 25: CSA (aberto/fechado) x tempo (s), CSA – descida, contr. P, 2 param.

Análise

Como uma idéia adicional, testou-se relacionar a abertura do registro, CSA, com a pressão no tubo de descida, P_{DC} , e a vazão mássica de água de alimentação, SPW, com a pressão no domo, P_{DO} , contrariando a lógica já que teoricamente o ideal é a relação já mostrada anteriormente, $CSA - P_{DO}$ e $SPW - P_{DC}$.

Surpreendentemente o comportamento encontrado após algumas horas foi bastante satisfatório. Nenhuma das duas pressões atingiu o nível de exatidão alcançado na

figura 12, mas em compensação ambas se aproximaram bastante do experimento, eliminando o deslocamento de P_{DC} em relação à referência, como vinha acontecendo até o momento. Outro fator que definiu esta configuração como ótima para o controlador P é o fato de desta vez a vazão SPW não estar "sobrecarregada" como nos outros testes: seus limites superior e inferior de atuação foram atingidos, porém por períodos relativamente pequenos, o que comprova o *aumento de sensibilidade* deste parâmetro nesta condição.

9 CONTROLADOR "CAIXA-CINZA" (*gray-box*)

9.1 Curva de dados

Conforme determinado por Kerner; Schaefer; Liu (2008) [1], o comportamento ideal observado para a abertura do registro CSA pode ser descrito pela figura 26:

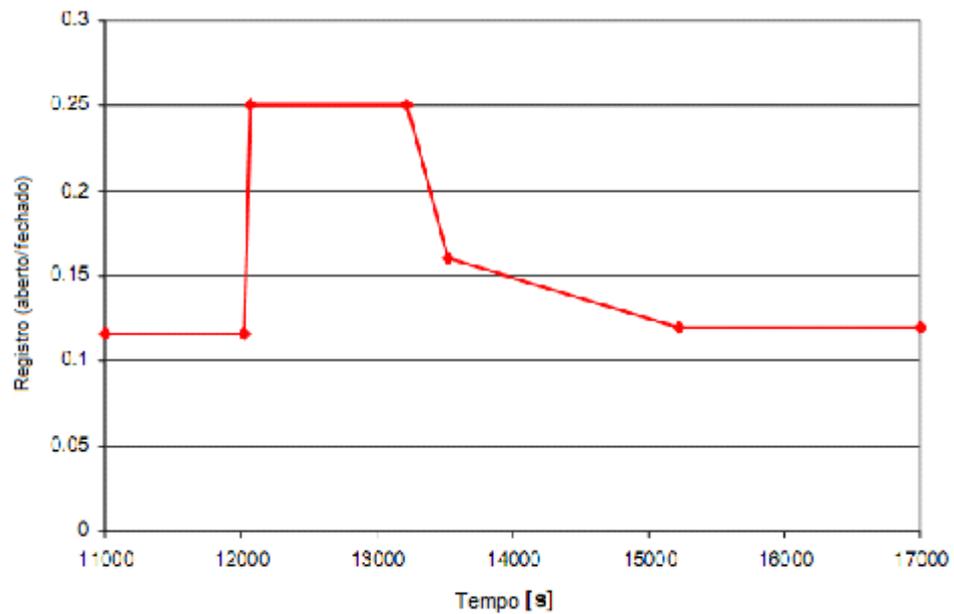


Figura 26: CSA (aberto/fechado) x tempo (s), comportamento ideal

Esta curva foi implementada nas simulações para simples verificação do comportamento do software em comparação com o experimento:

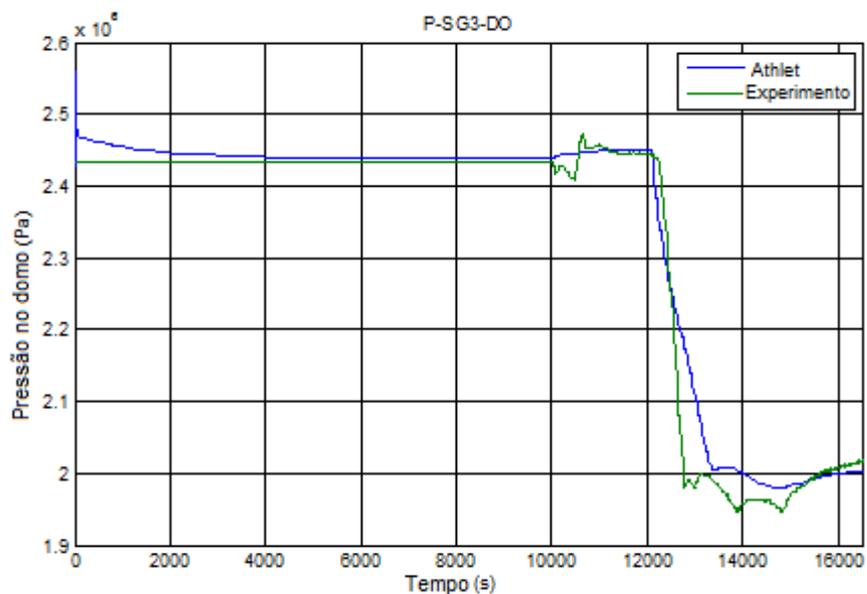


Figura 27: Pressão no domo (Pa) x tempo (s), CSA – domo, curva de CSA

Notou-se que a adaptação resultante da utilização da curva mostrada na figura 26 leva a resultados satisfatórios, porém inferiores aos conseguidos na seção anterior (controlador *black-box*).

A grande vantagem de se utilizar uma curva já "pronta" é a significativa redução do tempo computacional necessário para completar uma simulação completa, já que o código para implementar a curva consiste apenas numa tabela de valores a serem seguidos de acordo com o tempo.

9.2 Controle heurístico

Ao invés de utilizar diretamente a curva mostrada na figura 26, procurou-se também implementá-la indiretamente, por meio da seguinte formulação também determinada por Kerner; Schaefer; Liu (2008) [1],

$$\begin{aligned}\frac{d^2(\textit{press\~{a}o})}{dt^2} > 0 &\Rightarrow \textit{CSA} \downarrow \\ \frac{d^2(\textit{press\~{a}o})}{dt^2} < 0 &\Rightarrow \textit{CSA} \uparrow \\ \frac{d^2(\textit{press\~{a}o})}{dt^2} = 0 &\Rightarrow \textit{CSA} \rightarrow\end{aligned}\tag{7}$$

que é o comportamento intuitivo do registro CSA de acordo com a pressão no gerador de vapor: quando a pressão tende a aumentar cada vez mais com o tempo, a abertura do registro aumenta, a fim de conter tal variação; para evitar a perda exagerada de pressão por causas opostas, diminui-se a abertura; e, caso o gradiente de pressão seja constante, mantém-se a abertura.

A seguir, o melhor resultado obtido com este tipo de controlador:

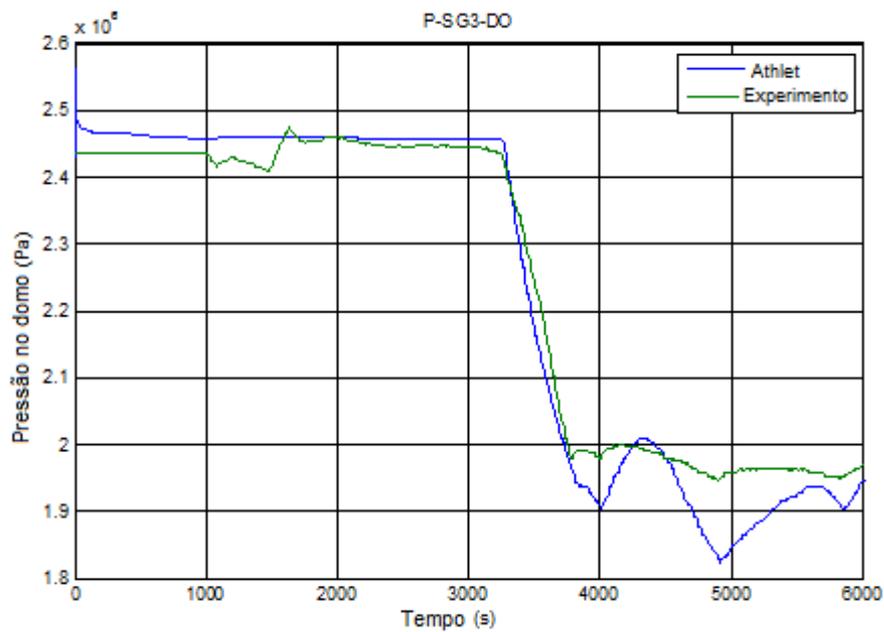


Figura 28: Pressão no domo (Pa) x tempo (s), CSA – domo, controle heurístico

A principal dificuldade inerente neste controlador é a manutenção da coerência após o início do transiente: a abertura de CSA continua sensível a variações da pressão, e a simples adição de um coeficiente de correção faz com que a reação ao início do transiente seja inadequada: os requisitos para cada momento (início do transiente, período subsequente) são conflitantes e tornaram a sintonia do controlador atividade extremamente dispendiosa (em questão de tempo, já que cada simulação leva 2h para terminar) de tentativa e erro.

10 DETECÇÃO DO INÍCIO DO TRANSIENTE

O fenômeno mais significativo no experimento PKLIII B3.1 é o aparecimento do transiente. A repentina e grande variação de pressão e potência da instalação não é detectada pelo software ATHLET com precisão, causando discrepância inaceitável nas vizinhanças do transiente.

Implementando códigos de adaptação (capítulos 8 e 9) para corrigir a resposta do software, o transiente é melhor simulado. O uso de uma rotina exclusiva para detectá-lo, porém, ainda foi de interesse devido aos seguintes fatores:

- não-necessidade de executar o código de controle em MATLAB até a ocorrência do transiente (economia de tempo de simulação);
- possibilidade de, após uma simulação, transladar a curva obtida em n intervalos de tempo (correspondentes ao atraso na detecção do transiente) a fim de obter uma aproximação ainda mais precisa do experimento;
- interesse do *Institute for Safety and Reliability* em novas possibilidades de detecção de transientes, para outras aplicações (o instante em que o fenômeno ocorre é parâmetro-chave de segurança).

Depois de repetidos esforços e tentativas, chegou-se a uma maneira simples, porém eficiente e razoavelmente rápida (levando em conta a escala de tempo) de detectar o instante de início do transiente.

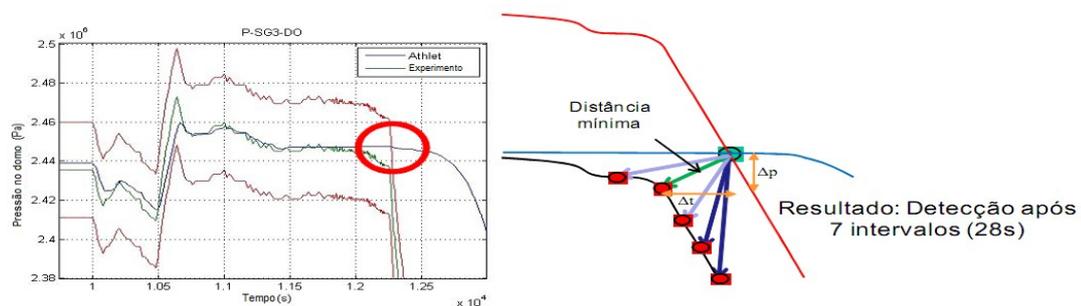


Figura 29: Esquema ilustrativo da detecção de transiente

Na figura acima, as curvas verde e azul são, respectivamente, pressão no domo experimental e simulada (como já mostrado anteriormente). As curvas vermelhas são simplesmente cópias translacionadas da curva verde, formando um limite para a diferença entre simulação e experimento.

Logo após as pequenas oscilações em torno de 10500s, adota-se um código de controle mais conservador, já que se o controlador atuar o tempo todo, acompanhando inclusive as pequenas oscilações, os limites não serão ultrapassados próximo à ocorrência do transiente, somente muito após o início do mesmo, comprometendo toda a detecção.

Quando a simulação ultrapassa os limites estipulados, calcula-se a distância do ponto fictício encontrado (tempo do transiente simulado, pressão de transiente simulado) a pontos experimentais, começando pelo correspondente (no mesmo instante de tempo) até o ponto ocorrido 10 intervalos de tempo antes. Não por acaso, o ponto de menor distância ao transiente fictício é uma estimativa excelente do transiente real, como observado na figura 29.

De acordo com Kerner; Schaefer; Liu (2008) [1] os limites de variação de pressão normais, ou seja, que não correspondem a um transiente, são de +/- 1%, exatamente o utilizado para traçar as linhas vermelhas da figura 29.

Os resultados obtidos para a detecção do início do transiente foram:

- Instante de ocorrência do transiente: 12246s
- P_{DO} correspondente: 2.419 MPa
- Atraso da detecção: 28s (7 intervalos de tempo)

Devido ao tamanho dos intervalos de tempo entre um cálculo e outro (4s), e a necessidade de ao menos alguns pontos até que os limites sejam atingidos no caso de um transiente, tem-se um intervalo de quase meio minuto até a detecção do transiente. A melhor alternativa para diminuir este tempo é a redução dos intervalos de tempo (até 0,5s, por exemplo), o que acarretaria porém em grande aumento de tempo computacional.

10.1 Utilização da detecção do início do transiente

Utilizando a melhor solução de adaptação encontrada (controlador P, 2 parâmetros: CSA -descida e SPW - domo) em conjunto com a detecção do transiente, foram obtidos os seguintes resultados:

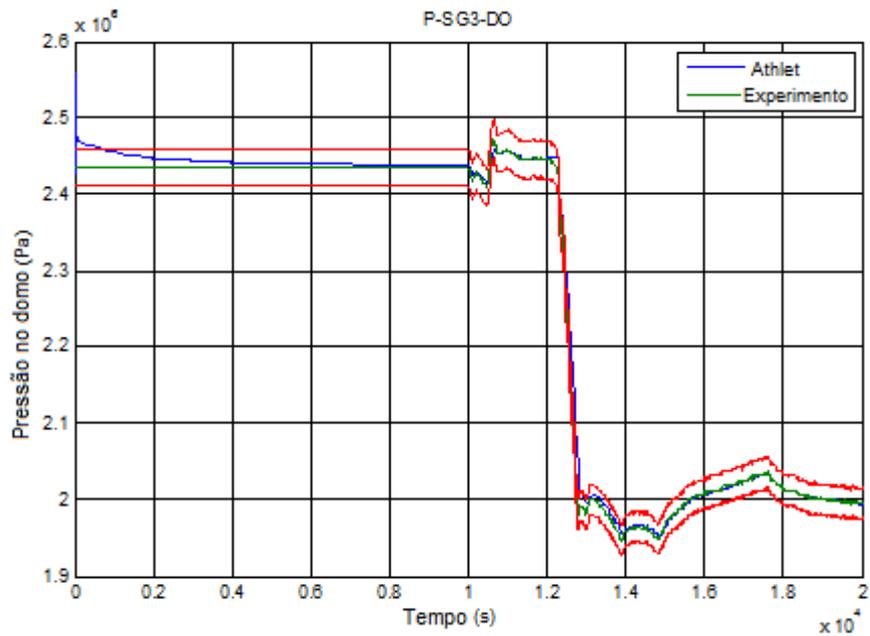


Figura 30: Pressão no domo (Pa) x tempo (s), SPW – domo, contr. P, 2 param, detec. trans.

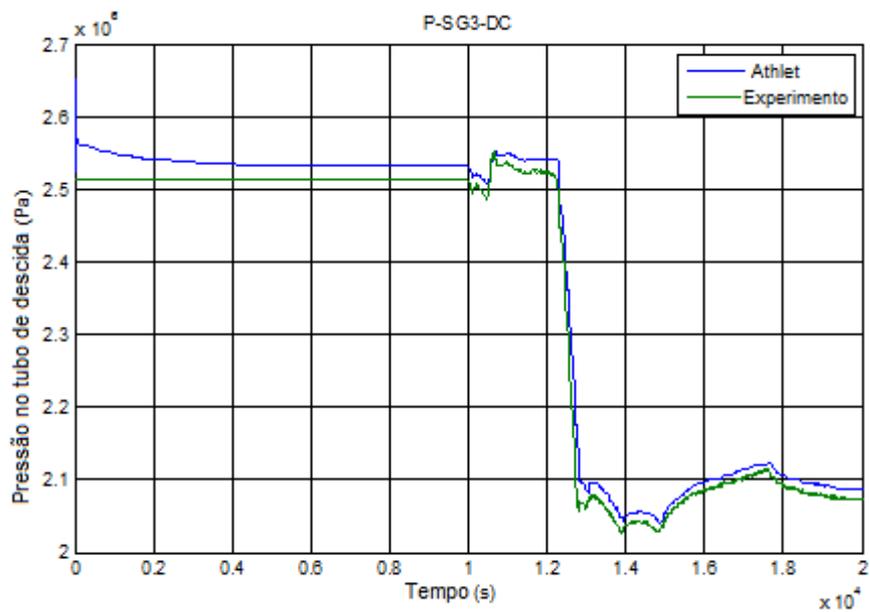


Figura 31: Pressão no tubo de descida (Pa) x tempo (s), CSA – descida, contr. P, 2 param, detec. Trans.

As curvas apresentadas são essencialmente as mesmas vistas na seção 5, exceto que agora, com a detecção do transiente, é conhecido o atraso da simulação em relação à curva real, e coeficientes do controlador específicos para atuação a partir do transiente podem ser utilizados.

A protuberância formada pela curva da simulação logo antes de o transiente ser detectado consiste num erro adicional, porém de ordem muito pequena (menor que 1%) e que portanto não compromete os resultados.

11 CONCLUSÕES E DISCUSSÃO

Embora de natureza bastante simples, os controladores do tipo *black-box* introduzidos no capítulo 8 foram de efetividade muito maior que a esperada. A aproximação obtida foi igual ou melhor que as mostradas por Kerner; Schaefer; Liu (2008) [1], obtidas por métodos avançados de análise de sensibilidade de parâmetros. Outra vantagem do sistema de controle proporcional mostrado neste trabalho é a baixo requisito computacional em comparação a outras soluções: quanto mais condicionais presentes numa rotina, mais tempo ela levará para ser calculada, o que pode ser decisivo quando não se dispões de clusters de alto desempenho dedicados exclusivamente à simulação, como era o caso.

A implementação de tais controladores, porém, não consistiu tarefa tão trivial: grande parte do período de trabalho foi despendido aprendendo e estudando o funcionamento do software ATHLET, bem como compreendendo fisicamente os fenômenos por ele simulados.

Se o controlador P foi um sucesso não esperado, pode-se dizer exatamente o contrário do controlador heurístico. Apesar de a regra de controle implementada ser bastante simples, a sintonia do controlador se mostrou muito delicada, não podendo ser atingida em tempo hábil para realização das outras tarefas previstas para o trabalho, o que foi decepcionante.

Muito tempo foi despendido, também, em busca de soluções mais satisfatórias para o controlador heurístico. Sua importância, porém, foi relativizada pelos orientadores do trabalho, já que esta parte consiste majoritariamente na verificação de resultados obtidos por Kerner; Schaefer; Liu (2008) [1].

Este trabalho, juntamente com o desenvolvido por Ceuca (2008) [2], é parte constituinte de uma frente de trabalho com o objetivo de somar melhorias ao software ATHLET, que, se utilizado sem correções, apresenta resultados inconclusivos para análises de segurança de usinas nucleares.

Em suma, foi experiência muito interessante desenvolver um trabalho em conjunto com um instituto alemão, e desafio muito grande o regime de trabalho independente exigido, aliado ao caráter específico do trabalho.

Para o futuro, pode-se refinar o controlador proporcional por meio da utilização de controladores proporcionais múltiplos, ou seja, utilizando-se uma matriz de correção

ao invés de apenas um coeficiente; controladores PD e PID também poderiam ajudar a reduzir o erro das simulações.

O método de detecção de transiente desenvolvido também não é ótimo, principalmente devido à necessidade de reduzir a ação de controle antes do transiente para facilitar e melhorar a precisão de detecção.

Todas estas melhorias encontram-se em desenvolvimento no ISaR, em Munique, pelo aluno Chen Kan Wu (Technische Universität München).

ANEXO A – Códigos em MATLAB

1) 1 parâmetro

- Parâmetro adaptado: CSA
- Variável de comparação: P_{DO}

% Betrachtung von 1 Parameter - CSA

```
if Daten(6, end) ~= Daten(6, end - 1) && not(exist('flag_regler', 'var'))
```

```
    flag_regler = 1;
```

```
end
```

```
if exist('flag_regler', 'var')
```

```
    % CSA
```

```
    % P-Regler - DO
```

```
    Kp(1) = 100;
```

```
    Theta(1) = 1 + Kp(1)*(Daten(4, end) - Daten(6, end))/max(Daten(4, end), Daten(6, end));
```

```
    Theta(1) = min(Theta(1), 8.6206896552);
```

```
    Theta(1) = max(Theta(1), 0);
```

```
end
```

- Parâmetro adaptado: SPW
- Variável de comparação: P_{DC}

% Betrachtung von 1 Parameter - Speisewassermassenstrom

```
if Daten(6, end) ~= Daten(6, end - 1) && not(exist('flag_regler', 'var'))
```

```
    flag_regler = 1;
```

```
end
```

```
if exist('flag_regler', 'var')
```

```
    % Speisewassermassenstrom
```

```
    % P-Regler - DC
```

```
    Kp(2) = 100;
```

```
    Theta(2) = 1 + Kp(2)*(Daten(5, end) - Daten(7, end))/max(Daten(5, end), Daten(7, end));
```

```
    Theta(2) = min(Theta(2), 2.2609088854);
```

```
    Theta(2) = max(Theta(2), 0);
```

```
end
```

2) 2 parâmetros

- CSA - P_{DO} ; SPW - P_{DC}

% Betrachtung von 2 Parameter

```
if Daten(6, end) ~= Daten(6, end - 1) && not(exist('flag_regler', 'var'))
```

```
    flag_regler = 1;
```

```
end
```

```
if exist('flag_regler', 'var')
```

```
    % CSA
```

```
    % P-Regler - DO
```

```
    Kp(1) = 50;
```

```
    Theta(1) = 1 + Kp(1)*(Daten(4, end) - Daten(6, end))/max(Daten(4, end), Daten(6, end));
```

```
    Theta(1) = min(Theta(1), 8.6206896552);
```

```
    Theta(1) = max(Theta(1), 0);
```

```
    % Speisewassermassenstrom
```

```
    % P-Regler - DC
```

```
    Kp(2) = 50;
```

```
    Theta(2) = 1 + Kp(2)*(Daten(5, end) - Daten(7, end))/max(Daten(5, end), Daten(7, end));
```

```
    Theta(2) = max(Theta(2), 0);
```

```
end
```

- CSA - P_{DC} ; SPW - P_{DO}

% Betrachtung von 2 Parameter UNGEKEHRT!

```
if Daten(6, end) ~= Daten(6, end - 1) && not(exist('flag_regler', 'var'))
```

```
    flag_regler = 1;
```

```
end
```

```
if exist('flag_regler', 'var')
```

```
    % CSA
```

```
    % P-Regler - DC
```

```
    Kp(1) = 100;
```

```
    Theta(1) = 1 + Kp(1)*(Daten(5, end) - Daten(7, end))/max(Daten(5, end), Daten(7, end));
```

```
    Theta(1) = min(Theta(1), 8.6206896552);
```

```
    Theta(1) = max(Theta(1), 0);
```

```
    % Speisewassermassenstrom
```

```
    % P-Regler - DO
```

```
    Kp(2) = 400;
```

```
    Theta(2) = 1 + Kp(2)*(Daten(4, end) - Daten(6, end))/max(Daten(4, end), Daten(6, end));
```

```
    Theta(2) = min(Theta(2), 2.2609088854);
```

```
    Theta(2) = max(Theta(2), 0); end
```

3) Controle Heurístico

- Implementação direta de valores tabelados

```
% Funktion - 10000
if TIME > 12062 && TIME < 12125
    Theta(1) = (0.0021*TIME - 25.2142)*8.6207;
elseif TIME > 12125 && TIME < 13250
    Theta(1) = 0.25*8.6207;
elseif TIME > 13250 && TIME < 13500
    Theta(1) = (-0.00036*TIME + 5.02)*8.6207;
elseif TIME > 13500 && TIME < 15250
    Theta(1) = (-2.5143*10^(-5)*TIME + 0.4994)*8.6207;
elseif TIME > 15250 && TIME < 17000
    Theta(1) = 1; % 0.116*8.6207
end
```

- Controlador heurístico

```
% Heuristische Betrachtung 14 - jetzt v6a
global abstand
tol = 250;
toldif = 10;
Kh = 70;
abstand = 2;

if Daten(1, end) > 12246 % Beginn des Transientes
    dif = Daten(6, end) - Daten(6, end - abstand);
    difa = Daten(4, end) - Daten(4, end - abstand);
    koeff = (Daten(6, end) - Daten(6, end - abstand))/max(Daten(6,
end - abstand), Daten(6, end));
    SAVE_KOEFF(length(Daten)) = koeff;
    SAVE_dif(length(Daten)) = dif;
    SAVE_difa(length(Daten)) = difa;

    % Betrachtung
    % Wenn nicht gleich oder nicht in der gleichen Richtung
    if (abs(dif) > tol) && ((abs(dif) - abs(difa)) > toldif) ||
((dif*difa) <= 0)
        if A_LINES3(1) > 0.7
            Theta(1) = SAVE_THETA(1, end - 1)*(1 - koeff);
        else
            Theta(1) = SAVE_THETA(1, end - 1)*(1 - Kh*koef);
        end
    end
    % Ende der Betrachtung
end
```

4) Detecção do transiente

```
% Adaption bis zum Transientenerkennung - vld
schritte = 20;
Kp(1) = 100;
totband = 0.01;

grenz_vor_trans = schritte/20000;

oberegrenze = Yminterpol(1)*(1 + totband);
unteregrenze = Yminterpol(1)*(1 - totband);
% SAVE_oberegrenze(length(Daten)) = oberegrenze;
% SAVE_unteregrenze(length(Daten)) = unteregrenze;

if ((P_SG3_DO(1) > oberegrenze || P_SG3_DO(1) < unteregrenze) &&
(not(exist('transzeit', 'var')) && exist('vortrans', 'var')))
    transzeit = TIME;
    transddom = P_SG3_DO(1);
    transcont = length(Daten);

    for a = 1:10
        d(a) = (transzeit - Daten(1, transcont - a))^2 + (transddom
- Daten(6, transcont - a))^2;
        if d(a) == min(d)
            verschiebung = a;
            transzeitexp = Daten(1, end - a);
        end
    end
else
    if (Daten(6, end) - Daten(6, end - 1)) ~= 0 &&
not(exist('start_flag', 'var'))
        start_flag = 1;
        SAVE_ctrl = [];
        tempo = [];
    end

    if exist('start_flag', 'var')

        if start_flag
            % P-Regler für CSA
            Theta(1) = 1 + Kp(1)*(Daten(4, end) - Daten(6,
end))/max(Daten(4, end), Daten(6, end));
            Theta(1) = min(Theta(1), 8.6206896552);
            Theta(1) = max(Theta(1), 0);

            if not(exist('cont', 'var'))
                cont = 1;
            else
                if cont == schritte
                    ctrl(1) = Daten(6, end - schritte);
                    ctrl(2) = Daten(6, end);
                    ctrl(3) = (max(ctrl(1), ctrl(2))/min(ctrl(1),
ctrl(2)) - 1);

                    SAVE_ctrl = [SAVE_ctrl ctrl'];
                    tempo = [tempo TIME];
                    if ctrl(3) < grenz_vor_trans && ctrl(3) >
0.00075
```

```

        if exist('schritt', 'var')
            if length(SAVE_ctrl) == (schritt + 1)
                start_flag = 0;
                vortrans = 1;
                ddom_ref = Daten(6, end);
            else
                schritt = length(SAVE_ctrl);
            end
        else
            schritt = length(SAVE_ctrl);
        end
    end
    cont = 2;
    return
else
    cont = cont + 1;
end
end %if not(exist('cont' ...

else
    if not(exist('transzeit', 'var'))
        % P-Regler für CSA - gerade Linie
        Theta(1) = 1 + Kp(1)*(Daten(4, end) -
ddom_ref)/max(Daten(4, end), ddom_ref);
        Theta(1) = min(Theta(1), 8.6206896552);
        Theta(1) = max(Theta(1), 0);
    end

    end % if start_flag ...

    end % if exist('start_flag' ...

end % if (P_SG3_DO(1) > ...

```

5) Controlador P + detecção do transiente

```
% Transientenerkennung und Adaption des Transientes - v1.3 (mit P-
Regler)

schritte = 20;
Kp(1) = 100;
totband = 0.01;

grenz_vor_trans = schritte/20000;

oberegrenze = Yminterpol(1)*(1 + totband);
unteregrenze = Yminterpol(1)*(1 - totband);
% SAVE_oberegrenze(length(Daten)) = oberegrenze;
% SAVE_unteregrenze(length(Daten)) = unteregrenze;

if ((P_SG3_DO(1) > oberegrenze || P_SG3_DO(1) < unteregrenze) &&
(not(exist('transzeit', 'var')) && exist('vortrans', 'var')))
    transzeit = TIME;
    transddom = P_SG3_DO(1);
    transcont = length(Daten);

    for a = 1:10
        d(a) = (transzeit - Daten(1, transcont - a))^2 + (transddom
- Daten(6, transcont - a))^2;
        if d(a) == min(d)
            verschiebung = a;
            transzeitexp = Daten(1, end - a);
        end
    end
else
    if (Daten(6, end) - Daten(6, end - 1)) ~= 0 &&
not(exist('start_flag', 'var'))
        start_flag = 1;
        SAVE_ctrl = [];
        tempo = [];
    end

    if exist('start_flag', 'var')

        if start_flag
            % P-Regler für CSA
            Theta(1) = 1 + Kp(1)*(Daten(4, end) - Daten(6,
end))/max(Daten(4, end), Daten(6, end));
            Theta(1) = min(Theta(1), 8.6206896552);
            Theta(1) = max(Theta(1), 0);

            if not(exist('cont', 'var'))
                cont = 1;
            else
                if cont == schritte
                    ctrl(1) = Daten(6, end - schritte);
                    ctrl(2) = Daten(6, end);
                    ctrl(3) = (max(ctrl(1), ctrl(2))/min(ctrl(1),
ctrl(2)) - 1);
                    SAVE_ctrl = [SAVE_ctrl ctrl'];
                    tempo = [tempo TIME];
                end
            end
        end
    end
end
```

```

0.00075
        if ctrl(3) < grenz_vor_trans && ctrl(3) >
            if exist('schritt', 'var')
                if length(SAVE_ctrl) == (schritt + 1)
                    start_flag = 0;
                    vortrans = 1;
                    ddom_ref = Daten(6, end);
                else
                    schritt = length(SAVE_ctrl);
                end
            else
                schritt = length(SAVE_ctrl);
            end
            end
            cont = 2;
            return
        else
            cont = cont + 1;
        end
    end %if not(exist('cont' ...

else
    if not(exist('transzeit', 'var'))
        % P-Regler für CSA - gerade Linie
        Theta(1) = 1 + Kp(1)*(Daten(4, end) -
ddom_ref)/max(Daten(4, end), ddom_ref);
        Theta(1) = min(Theta(1), 8.6206896552);
        Theta(1) = max(Theta(1), 0);
    else
        % CSA
        % P-Regler - DO
        Kp(1) = 200;
        Theta(1) = 1 + Kp(1)*(Daten(4, end) - Daten(6, end -
verschiebung))/max(Daten(4, end), Daten(6, end - transcont));
        Theta(1) = min(Theta(1), 8.6206896552);
        Theta(1) = max(Theta(1), 0);
    end

    end % if start_flag ...

    end % if exist('start_flag' ...

end % if (P_SG3_DO(1) > ...

```

ANEXO B – Excerto de arquivo de entrada, ATHLET

Módulo de controle, GCSM, incluindo as variáveis de saída disponibilizadas para o MATLAB.

```
@#####
C---- SENSITIVIT
@ SFAK      METHODE      Rueckkopplung LAMBDA ALPHA1 BEGIN      METHOD
DTSENS
  1.05D0 'DIFF'      1          0.00      250.0 0.5      9      0.00
@ Alpha1 = Gamma2 (MATLAB?)
@ DTSENS = mindest dT für SIK Neuberechnung und Adaption
@ NAME OF EXPERIMENTAL DATA FILE      DTEXP
  'C:\Users\fernando\Documents\MATLAB' @Pfad der Matlab Dateien
& 0.0
@
          0.0
K---- VARIABLES
@ OBJECT      MODEL      NODE VARIABLE      POSITION IN EXP. DATA
  'S3-DOME'    '-'          1 'PRESSURE'  1
  'S3-DC3'    '-'          1 'PRESSURE'  2
@ 'S3-RISER'  'MIXLEVEL'  1 'ZCLML'    2
@
K---- PARAMETERS
@ NAME      MINIMUM MAXIMUM dp/dt-MAX
  'A-LINES3'  2.0D-2  10.0D+0  3.0D+0      @ A-LINES3 - GAIN
(41) MÜSSEN in gleicher Reihenfolge
  'G-FW30'    2.0D-2  10.0D+0  3.0D+0      @ G-FW30 - GAIN (80)
@an erster Stelle in K - MATLABGCSM stehen!!
@
K---- MATLABGCSM
@ NAMES FOR GCSMSIGNALS SEND TO MATLAB
@ Name      V, P, T
  'A-LINES3'  'P'          @ = Par1 = CSA in
SteamLine
  'G-FW30'    'P'          @ = Par2 = GJ of FW
in Loop3
  'PROBTIME'  'T'          @ = TIME
  'P-SG3-DO'  'V'          @ = Var1 = Press
in SG3-DOME
  'P-SG3-DC'  'V'          @ = Var2 = Press
```

```

in SG3-DC3
'LQRI3'      'V'      @ = Füllstand im DC3 + Sensi
@
C---- MISCELLAN
@
K---- DRUFAN
@ AA          BB          CC          DD          EE
@ 0.00        0.50        0.0D+00    0.0D+00    0.0D+00
  3000. 0.00          0.0D+00    0.0D+00    0.0D+00
@
@ FF          GG          OO          PP          QQ
@ 1.0D+00    0.0D+00    0.0D+00    0.0D+00    0.0D+00
& 1.0D+00    0.0D+00    0.0D+00    0.0D+00    0.0
@
@#####

```

REFERÊNCIAS

- Kerner, Alexander; Schaefer, Anselm; Liu, Xiaojing. **Entwicklung eines Simulators mit automatischer Anpassung des Simulationsmodells an Messdatenreihen**. Relatório parcial. Institute for Safety and Reliability (ISaR)/ Technische Universität München, 2008.
- CEUCA, Sabin. **Untersuchung von Identifikationsmethoden zur Anpassung des Teilsimulationsmodells "Reaktorkern"**. Tese de mestrado. Institute for Safety and Reliability (ISaR)/ Technische Universität München, 2008.
- GRS, **ATHLET Mod. 2.1 Cycle A, Models and Methods**. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, 2006.
- SIEMENS-KWU, Especificação de testes, Teste PKL III B 3.1. **Abfahren der Anlage im Naturumlauf mit einem isolierten Dampferzeuger und einen Abfahrgradienten der noch einen Umlauf im isolierten Dampferzeuger ermöglicht**, 1989.
- OGATA, Katsuhiko. **Engenharia de Controle Moderno**. Prentice Hall, 2003.