

Gerenciamento Dinâmico de Energia em Processadores com Cargas de Trabalho Variantes no Tempo

Saulo Oliveira Dornellas Luiz

Tese de Doutorado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich

Orientador

Antonio Marcus Nogueira Lima

Orientador

Campina Grande, Paraíba, Brasil

©Saulo Oliveira Dornellas Luiz, Abril de 2012

Gerenciamento Dinâmico de Energia em Processadores com Cargas de Trabalho Variantes no Tempo

Saulo Oliveira Dornellas Luiz

Tese de Doutorado apresentada em Abril de 2012

Angelo Perkusich

Orientador

Antonio Marcus Nogueira Lima

Orientador

Cursino Brandão Jacobina

Componente da Banca

Péricles Rezende de Barros

Componente da Banca

Carlos Eduardo Trabuco Dórea

Componente da Banca

José Roberto Castilho Piqueira

Componente da Banca

Campina Grande, Paraíba, Brasil, Abril de 2012

L953g Luiz, Saulo Oliveira Dornellas
Gerenciamento dinâmico de energia em processadores com
cargas de trabalho variantes no tempo / Saulo Oliveira
Dornellas Luiz. - Campina Grande, 2012.
130 f. : il.

Tese (Doutorado em Engenharia Elétrica) - Universidade
Federal de Campina Grande, Centro de Engenharia Elétrica e
Informática.

1. Gerenciamento Dinâmico de Energia 2. Sistemas de
Computação 3. Cadeias de Markov 4. Modelo da Carga de
Trabalho 5. Controle Adaptativo 6. Estimacão 7. Tese I.
Perkusich, Angelo, Dr. II. Lima, Antonio Marcus Nogueira,
Dr. III. Universidade Federal de Campina Grande - Campina
Grande(PB) IV. Título

CDU 621.316:004(043)


**GERENCIAMENTO DINÂMICO DE ENERGIA EM PROCESSADORES COM
CARGAS DE TRABALHO VARIANTES NO TEMPO**


SAULO OLIVEIRA DORNELLAS LUIZ

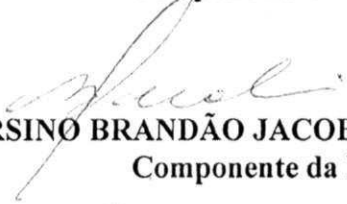
Tese Aprovada em 27.04.2012


ANGELO PERKUSICH, D.Sc., UFCG
Orientador (Ausência Justificada)


ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFPE
Orientador


JOSÉ ROBERTO CASTILHO PIQUEIRA, Dr., USP
Componente da Banca


CARLOS EDUARDO TRABUCO DÓREA, Dr., UFRN
Componente da Banca


CURSINO BRANDÃO JACOBINA, Dr.Ing., UFCG
Componente da Banca


PÉRICLES REZENDE BARROS, Ph.D., UFCG
Componente da Banca

CAMPINA GRANDE – PB
ABRIL - 2012

À minha família.

Agradecimentos

A Deus, que me fortalece e me conduz.

À minha família: minha noiva Otávia, pelo amor, carinho, estímulo e companheirismo; meus pais Alba e José Luiz, minha irmã Simone, meus avós maternos Francisca e Eneuton, e avós paternos Rita e José Luiz, pelo seu amor, seus esforços para me conduzirem a esse caminho, e seus exemplos de vida que me emocionam e me dão forças para ir cada vez mais longe; a todos os meus tios e tias; meus padrinhos Adjamir e Francinete; minhas tias avós Ir. Adélia e *In memoriam* Ir. Maria; minhas primas e primos; D. Eugênia, Paulo e Montini.

Aos professores Angelo Perkusich e Antonio Marcus Nogueira Lima, pela orientação, confiança, atenção, tempo dedicado e todo o apoio para a realização deste trabalho.

A todos os meus amigos, pelo companheirismo e apoio.

A todos os meus professores, do Ensino Fundamental à Pós-Graduação, pela sua contribuição na minha formação.

Aos integrantes do DEE/CEEI/UFCG e os funcionários da COPELE, pelo apoio.

A todos os alunos, professores e funcionários do Laboratório de Sistemas Embarcados e Computação Pervasiva, e do Laboratório de Instrumentação Eletrônica e Controle, que contribuíram direta ou indiretamente para a realização deste trabalho.

A todos os colegas da UNIVASF, em especial os coordenadores Edmundo, Valentim, Ricardo e Fabrício, pelo apoio para que eu pudesse realizar as atividades do meu doutorado.

Ao CNPq, pela bolsa de pesquisa concedida no início do doutorado, antes de eu me tornar professor do CECOMP/UNIVASF, e posteriormente professor do DEE/CEEI/UFCG.

Resumo

A necessidade de economizar energia está presente em quase todos os sistemas de computação. Para lidar com essa demanda crescente por gerenciamento de energia, os fabricantes de semicondutores têm desenvolvido várias técnicas de projeto de baixo consumo, tais como: estados de baixo consumo, escalonamento dinâmico de tensão e frequência e *clock gating*. Por exemplo, no caso do gerenciamento de energia de um processador ao nível do sistema operacional, o objetivo é escolher o par de tensão e frequência mais adequado para minimizar o consumo de energia e garantir o nível de desempenho desejado pelo usuário. Um grande desafio para o caso de sistemas computacionais de propósito geral é que a carga de trabalho do sistema é inicialmente desconhecida e variante no tempo. Assim é necessário um *gerenciador de energia*, em que os estados de energia do sistema são escolhidos através de um procedimento denominado *política de gerenciamento de energia*. Na literatura, há uma grande quantidade de políticas consideradas ótimas para sistemas alvo específicos. Além disso, em grande parte essas políticas podem ser consideradas de *malha aberta*, pois, em tempo de execução, o desempenho do sistema não é comparado a um valor de referência de desempenho especificado. Com o objetivo de garantir que o desempenho do sistema seja igual a um valor de referência de desempenho especificado, técnicas de controle em *malha fechada* tem sido aplicadas ao problema de gerenciamento dinâmico de energia em diferentes cenários, tais como, o desempenho de aplicações, o nível de utilização de um microprocessador embarcado executando um conjunto conhecido de tarefas de tempo-real de controle, a ocupação de um *buffer* num decodificador multimídia, etc.. Neste trabalho, foi desenvolvida uma técnica de controle adaptativo para gerenciamento dinâmico de energia, com o objetivo de garantir um nível de desempenho especificado e minimizar o consumo de energia de um processador provido de escalonamento dinâmico de tensão e frequência, considerando que a carga de trabalho do sistema é inicialmente desconhecida e variante no tempo. Essa técnica foi concebida o mais geral possível, para que possa ser utilizada em diferentes tipos de sistemas computacionais. Nos resultados experimentais, a técnica de controle adaptativo desenvolvida superou, tanto em potência quanto em desempenho do sistema, outras políticas de gerenciamento de energia do estado da arte.

Abstract

The need to save power is present in almost every computer system. To handle this increasing demand for power management, the semiconductor manufacturers have been developing several low power design techniques, such as: low power states, dynamic voltage and frequency scaling, and clock gating. For example, in the case of processor power management at the operating system level, the goal is to choose the most suitable pair of voltage and frequency to minimize the power consumption and guarantee the performance level desired by the user. A great challenge for the dynamic power management of general purpose computer systems is that the system workload is initially unknown and time-variant. Therefore it is necessary a *power manager*, where the system power states are chosen by means of a procedure named *power management policy*. In the literature, there are many power management policies which are claimed to be optimum for specific targeted systems. Furthermore, most of those policies may be considered as *open loop* policies, because, during execution time, the system performance is not compared to a specified reference performance value. To guarantee that the system performance is equal to a specified reference performance value, *closed loop* control techniques have been applied to the power management problem at several cases, such as, the applications performance, the utilization of an embedded microprocessor running a known set of real-time control tasks, the occupancy of a buffer in a multimedia decoder, etc.. In this Work, an adaptive control technique for dynamic power management was developed with the goal of guaranteeing a specified reference performance value and minimizing the power consumption of a processor with dynamic voltage and frequency scaling, considering that the system workload is initially unknown and time-variant. This technique was conceived to be as general as possible, so that it may be used for different types of computer systems. In the experimental results, the developed adaptive control technique outperformed other state-of-the art policies both in power and system performance.

Sumário

1	Introdução	1
1.1	Delimitação	2
1.2	Relevância	7
1.3	Problema abordado	7
1.4	Hipótese	8
1.5	Objetivo geral	8
1.6	Objetivos específicos	8
1.7	Organização do Texto	8
2	Abordagem estocástica para gerenciamento de energia	10
2.1	Introdução	10
2.2	Políticas para cargas de trabalho estacionárias	12
2.3	Políticas para cargas de trabalho não-estacionárias	27
2.3.1	Método de janelas deslizantes de múltiplos comprimentos	28
2.3.1.1	Janelas de requisições	28
2.3.1.2	Erro de amostragem	34
2.3.1.3	Tempo de preenchimento da janela	35
2.3.1.4	Janelas de estimativas	36
2.3.1.5	Seletor de janela	37
2.3.2	Identificação de cargas de trabalho por meio de aprendizado estocástico	41
2.3.2.1	Propriedades estatísticas das estimativas	43

2.3.2.2	Atraso de identificação	44
2.3.2.3	Erro de amostragem	46
2.3.3	Obtenção da política de energia	47
2.3.4	Comparação de várias técnicas de estimação de cargas de trabalho . . .	50
2.4	Resumo	57
3	Controle em malha fechada para gerenciamento de energia	60
3.1	Introdução	60
3.2	O sistema com gerenciamento de energia	62
3.3	O bloco <i>Quantizador</i>	63
3.4	O bloco <i>Estimar perda de requisições</i>	63
3.5	O modelo discreto do sistema	65
3.6	O projeto do controlador	66
3.7	Resultados e Discussão	69
3.8	Resumo	80
4	Controle Adaptativo para gerenciamento de energia	81
4.1	Introdução	81
4.2	Controle adaptativo para gerenciamento de energia	82
4.2.1	O modelo discreto do sistema	85
4.2.2	O regulador auto-sintonizável	86
4.2.2.1	A estimação dos parâmetros do modelo do processo	86
4.2.2.2	O projeto do controlador	88
4.3	Resultados experimentais	91
4.3.1	Plataforma experimental	91
4.3.2	O controle de utilização do processador	93
4.3.2.1	Bloco sensor	95
4.3.2.2	Atuador	97

4.3.2.3	Seleção do período de amostragem	101
4.3.2.4	A estimação de parâmetros do processo	101
4.3.2.5	O projeto do controlador	102
4.3.2.6	O controlador	103
4.3.2.7	Resultados	103
4.4	Resumo	114
5	Conclusões e trabalhos futuros	117
5.1	Trabalhos futuros	119
A	Projeto do controlador linear geral	121
B	Escolha do fator de esquecimento	123
	Referências Bibliográficas	125

Glossário

A-D	Analógico-digital
ACPI	<i>Advanced Configuration and Power Interface</i>
BRA	<i>Buffer da Requisição Anterior</i>
DVS	Escalonamento Dinâmico de Tensão (<i>Dynamic Voltage Scaling</i>)
EA	Estado Atual
EFAE	Estimação Fraca baseada em Aprendizado Estocástico
EMQ	Erro Médio Quadrático
EMV	Estimativas de Máxima-verossimilhança
FS	Fila de Serviços
GDE	Gerenciamento Dinâmico de Energia
GE	Gerenciador de energia
HMM	Modelo Escondido de Markov (<i>Hidden Markov Model</i>)
JMC-EFAE	Técnica de Janelas deslizantes de Múltiplos Comprimentos combinada com Estimação Fraca baseada em Aprendizado Estocástico
JMC-EMV	Técnica de Janelas deslizantes de Múltiplos Comprimentos combinada com Estimativas de Máxima-verossimilhança
MDPP	Alocação de pólos de grau mínimo (<i>Minimum-degree pole placement</i>)
MQR	Algoritmo de mínimos quadrados recursivo
NoC	<i>Network-on-chip</i>
OP	Otimização de Política
PAO	Política Adaptativa Ótima
PS	Provedor de Serviços
PWM	Modulação por Largura de Pulso (<i>Pulse-Width Modulation</i>)

QoS	Qualidade do Serviço (<i>Quality of Service</i>)
RS	Requisitante de Serviços
SISO	Entrada única e saída única (<i>Single Input, Single Output</i>)
SO	Sistema operacional
USB	<i>Universal Serial Bus</i>

Lista de Tabelas

2.1	Erro médio quadrático da identificação do RS para a abordagem com método de multi-janela	32
2.2	Comparação dos métodos de identificação do RS.	41
3.1	Estados de energia	69
3.2	Estados de energia normalizados	70
4.1	Utilização do processador pelo próprio gerenciador de energia	102

Lista de Figuras

2.1	Componentes do modelo de um sistema com gerenciamento de energia.	10
2.2	Probabilidade π_1 de que um requisitante de serviços com conjunto de estados $\{r_0, r_1\}$ está no estado r_1	13
2.3	O PS não serviu nenhuma requisição e o número de requisições que chegaram mais as que já estavam na fila não excede o tamanho da fila.	15
2.4	O PS serviu uma requisição e a soma do número de requisições que estavam na fila mais as que chegaram menos a que foi servida não excede o tamanho da fila.	16
2.5	A fila estava vazia e assim permanece, pois não apareceu nenhuma nova requisição.	17
2.6	A fila se torna cheia e a soma do número de requisições que estavam na fila mais as que chegaram excede o tamanho da fila.	18
2.7	Determinação do comando do gerenciador de energia.	21
2.8	Cadeia de Markov representando o provedor de serviços (PS).	22
2.9	Cadeia de Markov representando o requisitante de serviços (RS).	23
2.10	Cadeia de Markov representando a fila de serviços (FS).	23
2.11	Exemplos de cargas de trabalho do sistema.	25
2.12	Consumo de energia e perda de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (RS).	26
2.13	Políticas que (não) satisfazem a restrição de desempenho.	26
2.14	Classificação das cargas de trabalho do sistema.	27
2.15	Exemplo de um RS não-estacionário.	27

2.16	Valor de R_i ao longo do tempo para uma carga não-estacionária.	28
2.17	Componentes do modelo do sistema baseado em processos de Markov no tempo discreto para ambiente não-estacionário.	29
2.18	Exemplo de uso de multi-janela para RS de dois estados.	30
2.19	Variância de X	30
2.20	Variância da estimativa de $p_{r_i, r_j}^{RS}(t) = 0,5$	31
2.21	Identificação de um RS por meio do método de multi-janela com comprimento 100.	33
2.22	Identificação de um RS por meio do método de multi-janela com comprimento 200.	33
2.23	Identificação de um RS por meio do método de multi-janela com comprimento 400.	34
2.24	O erro de amostragem (linha de maior espessura) após a transição de uma carga de trabalho estacionária u_r para outra u_s	35
2.25	O atraso de identificação (linha de maior espessura) durante a transição de uma carga de trabalho estacionária u_r para outra u_s	36
2.26	Exemplo do bloco <i>janelas de estimativas</i> como mostrado na Fig. 2.17 para um requisitante de serviços com dois estados.	37
2.27	Método de janelas deslizantes de múltiplos comprimentos: (a) Estimativas de $p_{r_0, r_0}^{RS}(t)$; (b) comprimentos de janela utilizados. Estimativas de $p_{r_0, r_0}^{RS}(t)$ por meio do método de multi-janela [1] com janelas de comprimento: (c) 50; (d) 250; (e) 450.	39
2.28	Método de janelas deslizantes de múltiplos comprimentos: (a) Estimativas de $p_{r_1, r_1}^{RS}(t)$; (b) comprimentos de janela escolhidos. Estimativas de $p_{r_1, r_1}^{RS}(t)$ por meio do método de multi-janela [1] com janelas de comprimento: (c) 50; (d) 250; (e) 450.	40
2.29	(a) Comprimentos de janela escolhidos. Variâncias das estimativas com janelas de comprimento: (b) 50; (c) 250; (d) 450.	40
2.30	Estimação fraca baseada em aprendizado estocástico para ambientes não-estacionários.	42
2.31	Bloco <i>estimação fraca baseada em aprendizado estocástico</i> para um requisitante de serviços de dois estados.	43

2.32	Transição do requisitante de serviços do estado $s_r(t - 1) = r_i$ para o estado $s_r(t) = r_j$	43
2.33	O atraso de identificação (linha de maior espessura) durante a transição de uma carga de trabalho estacionária u_r para outra u_s	44
2.34	Intervalos de amostragem quando o valor da estimativa $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ é atualizado.	45
2.35	O erro de amostragem (linha de maior espessura) após a transição de uma carga de trabalho estacionária u_r para outra u_s	47
2.36	Exemplo de tabela de políticas com $NS_0 = NS_1 = 5$	48
2.37	Tabela de decisão e exemplo de seleção de linhas da tabela de políticas.	49
2.38	Exemplo de tabela de políticas com $NS_0 = NS_1 = 5$	49
2.39	Probabilidades de transição do modelo do requisitante de serviços não-estacionário.	52
2.40	A abordagem de multi-janela (<i>EMV</i>), a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima verossimilhança (<i>JMC-EMV</i>), e a política adaptativa ótima (<i>PAO</i>) (ideal). (a) Erro médio quadrático (<i>EMQ₀₀</i>) das estimativas da probabilidade de transição $p_{r_0, r_0}^{RS}(t)$; (b) Erro médio quadrático (<i>EMQ₁₁</i>) das estimativas da probabilidade de transição $p_{r_1, r_1}^{RS}(t)$; (c) consumo de corrente médio; (d) perda de requisições média.	54
2.41	A técnica de estimação fraca baseada em aprendizado estocástico (<i>EFAE</i>), a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimação fraca baseada em aprendizado estocástico (<i>JMC-EFAE</i>), e a política adaptativa ótima (ideal) (<i>PAO</i>). (a) Erro médio quadrático (<i>EMQ₀₀</i>) das estimativas da probabilidade de transição $p_{r_0, r_0}^{RS}(t)$; (b) erro médio quadrático (<i>EMQ₁₁</i>) das estimativas da probabilidade de transição $p_{r_1, r_1}^{RS}(t)$; (c) consumo de corrente médio; (d) perda de requisições média.	56
2.42	Tempo de simulação do gerenciador de energia.	57
3.1	Gerenciamento de energia baseado no princípio de controle em malha fechada.	60
3.2	Diagrama de blocos do controle em malha fechada para gerenciamento de energia.	61
3.3	O quantizador de comandos.	63
3.4	A perda de requisições esperada.	64
3.5	Diagrama de blocos do controle em malha fechada para gerenciamento de energia, destacando-se controlador e processo.	65

3.6	Diagrama de blocos do modelo discreto do controlador e do sistema com gerenciamento de energia.	67
3.7	Diagrama de blocos do controlador e processo considerando o bloco <i>Quantizador</i> como um atuador não-linear.	68
3.8	Experimento para coletar dados de entrada $u(t)$ e saída $\hat{l}(t)$	72
3.9	Abordagem estocástica: potência média e perdas de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (<i>RS</i>).	74
3.10	Abordagem estocástica: verificação da satisfação da restrição de penalidade de desempenho para as cargas de trabalho leves.	75
3.11	Controle em malha fechada: potência média e perda de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (<i>RS</i>).	76
3.12	Controle em malha fechada: verificação da satisfação da restrição de penalidade de desempenho para todas as cargas de trabalho.	77
3.13	Exemplo de simulação para a carga de trabalho $p_{r_0,r_0}^{RS} = 0,5$ e $p_{r_1,r_1}^{RS} = 0,5$	78
3.14	Diferença de potência média. Valores negativos representam cargas de trabalho para as quais o controle em malha fechada oferece menor potência média do que a abordagem estocástica.	79
3.15	Diferença de potência média para as políticas estocásticas aceitáveis. Valores negativos representam cargas de trabalho para as quais o controle em malha fechada oferece menor potência média do que a abordagem estocástica.	79
4.1	Os componentes do sistema com gerenciamento de energia.	83
4.2	Diagrama de blocos do controle em malha fechada para o gerenciamento de energia do processador.	83
4.3	Diagrama de blocos do regulador auto-sintonizável.	86
4.4	Diagrama de blocos do modelo discreto do controlador e do sistema com gerenciamento de energia.	88
4.5	A estimação dos parâmetros do processo e o projeto do controlador.	89
4.6	Diagrama de blocos do controlador e o processo considerando o atuador não-linear.	89

4.7	Diagrama de blocos do controlador proporcional-integral considerando o atuador não-linear.	90
4.8	Esquema para implementação da técnica de controle adaptativo para gerenciamento de energia.	92
4.9	Fluxograma da técnica de controle adaptativo para gerenciamento de energia. . .	93
4.10	Efeito da redução da frequência no tempo de execução de uma tarefa.	94
4.11	Exemplo da resposta transitória para uma entrada aleatória.	96
4.12	Representação para o sinal de modulação por largura de pulso.	99
4.13	Representação para o sinal de modulação de primeira ordem delta-sigma. . . .	100
4.14	Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a reprodução, por meio do tocador <i>totem</i> [2], da música de domínio público on000001.mp3 [3].	106
4.15	Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a reprodução, por meio do tocador <i>totem</i> [2], do vídeo de domínio público citizenengineermov [4].	107
4.16	Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a execução do <i>benchmark ebizzy</i> [5].	108
4.17	Exemplo da resposta transitória para a política Userspace@0,800GHz, com a carga de trabalho variável ao longo do tempo sendo: (i) a reprodução, por meio do tocador <i>totem</i> [2], do vídeo de domínio público citizenengineermov [4], (ii) a reprodução, por meio do tocador <i>totem</i> [2], da música de domínio público on000001.mp3 [3], e (iii) a execução do <i>benchmark ebizzy</i> [5].	109
4.18	Exemplo da resposta transitória para a política Userspace@1,867GHz, com a carga de trabalho variável ao longo do tempo sendo: (i) a reprodução, por meio do tocador <i>totem</i> [2], do vídeo de domínio público citizenengineermov [4], (ii) a reprodução, por meio do tocador <i>totem</i> [2], da música de domínio público on000001.mp3 [3], e (iii) a execução do <i>benchmark ebizzy</i> [5].	110
4.19	Exemplo da resposta transitória para a técnica de controle adaptativo.	112
4.20	Consumo de potência, penalidade de desempenho e a utilização do processador para as políticas de gerenciamento de energia consideradas no estudo de caso. . .	114

B.1 O limitante inferior da excitação persistente e o erro de controle médio quadrático medidos para $\lambda = 0,900, 0,905, 0,910, \dots, 0,995$ 124

Capítulo 1

Introdução

A necessidade de economizar energia está presente em quase todos os sistemas de computação. Para dispositivos alimentados a bateria, tais como *notebooks*, telefones celulares, tocadores de música, *tablets*, etc., existe uma diferença crescente entre o desempenho requerido pelo sistema eletrônico (processadores mais poderosos, maior memória, unidades de armazenamento, interfaces de rede, etc.) e a densidade de energia da bateria [6]. Assim, a indústria de dispositivos móveis tem realizado esforços para estender a autonomia da bateria usando eficientemente a energia disponível e atendendo as demandas do usuário. No caso de dispositivos alimentados pela rede elétrica, tais como computadores pessoais de mesa, servidores, etc., os custos devido ao consumo de eletricidade e resfriamento representam gastos que precisam ser minimizados [7]. Para lidar com essa demanda crescente por gerenciamento de energia, os fabricantes de semicondutores têm desenvolvido várias técnicas de projeto de baixo consumo, tais como: estados de baixo consumo, escalonamento dinâmico de tensão e frequência e *clock gating* [6].

As técnicas de redução de consumo de energia são classificadas em estáticas e dinâmicas [8]. As técnicas estáticas são usadas na fase de projeto; são exemplos a síntese e a compilação para baixo consumo. Já as técnicas dinâmicas, denominadas *Gerenciamento Dinâmico de Energia* (GDE), possibilitam a redução do consumo de energia nesses sistemas em tempo de execução (durante a utilização do sistema), por meio do desligamento ou redução da frequência/tensão de operação de componentes de sistemas que estão ociosos ou subutilizados. Segundo Lu, Benini e Micheli [9], gerenciamento dinâmico de energia (GDE) é a escolha dinâmica de estados de energia para um dado sistema com base na sua carga de trabalho. Num dispositivo, um *gerenciador de energia* (GE) monitora os estados dos componentes e do sistema e controla seus estados de energia por meio de um procedimento denominado *política de gerenciamento de energia* [1], que decide quais estados de operação vão assumir os componentes do sistema [10].

O gerenciamento de energia dirigido pelo sistema operacional tem duas categorias: (i) es-

calonamento dinâmico de tensão e frequência do processador; (ii) o gerenciamento dos estados de energia dos dispositivos (disco rígido, monitor, interfaces de rede, etc.). Quanto ao gerenciamento de energia de dispositivos, é possível destacar dois tipos: (i) o gerenciamento de energia no nível do dispositivo (GEND), uma abordagem centrada em *hardware*, que é baseada na simples observação das requisições feitas ao dispositivo, as quais são consideradas indistinguíveis, por não serem observadas as suas fontes; (ii) gerenciamento de energia baseado em tarefas (GEBT), uma abordagem centrada em *software*, que utiliza informações provenientes do sistema operacional para classificar as requisições para um dispositivo de acordo com as tarefas às quais elas são associadas, isto é, os processos do sistema operacional.

1.1 Delimitação

O desenvolvimento de técnicas para o uso eficiente de energia se tornou um desafio chave para o projeto de sistemas computacionais. O objetivo é a redução do consumo de energia de sistemas em tempo de execução, por meio do desligamento ou redução da frequência/tensão de operação de componentes de sistemas que estão ociosos ou sub-utilizados. Por exemplo, no caso do gerenciamento de energia do processador realizado no nível do sistema operacional, o objetivo é escolher o par de tensão e frequência adequado para minimizar o consumo de energia e garantir o nível de desempenho desejado pelo usuário. Para sistemas computacionais de propósito geral, a carga de trabalho do sistema é inicialmente desconhecida e variante no tempo [1, 11]. Por exemplo, num computador pessoal, na sua inicialização são desconhecidas as aplicações que serão executadas pelo usuário, que são fruto de suas necessidades pessoais. Também, são desconhecidos os instantes em que a carga de trabalho muda, a sequência de aplicações, ou os intervalos de tempo em que o sistema computacional permanece ocioso. Para cada aplicação em execução, ou um conjunto destas, a demanda de processamento é diferente, o que implica em oportunidades diferentes para redução do consumo de energia.

Nos atuais circuitos CMOS, o consumo de potência é composto basicamente pela potência estática e a potência dinâmica (chaveamento e de curto-circuito) [6, 12, 13]:

- A potência estática ou de fuga (*leakage*) ocorre devido à corrente através do substrato. Infelizmente, a potência estática está presente tanto em estados de energia ativos quanto em *standby*. A potência estática é proporcional à tensão v e a corrente de fuga I_{fuga} , ou seja, $P_{fuga} = vI_{fuga}$;
- A potência de chaveamento capacitivo ocorre devido ao carregamento e descarregamento da carga capacitiva na saída de cada porta. Essa componente da potência é proporcional

à atividade de chaveamento A (em que A é adimensional e $0 \leq A \leq 1$), à carga capacitiva C , ao quadrado da tensão v , e a frequência f , ou seja, $P_{\text{chaveamento}} = ACv^2f$;

- A potência de curto-circuito ocorre porque um caminho direto de corrente é estabelecido entre a fonte e a terra por um pequeno período durante tempos não-nulos de subida e descida em transistores. Essa componente da potência é proporcional à tensão v , e uma corrente de curto-circuito I_{curto} , ou seja, $P_{\text{curto}} = vI_{\text{curto}}$.

Se num processador há o suporte a escalonamento dinâmico de tensão e frequência, então é possível escolher, em tempo de execução, a tensão v e a frequência f , e, como consequência, mudar as componentes de consumo de energia: estática, chaveamento capacitivo e potência de curto-circuito. O objetivo principal é minimizar o consumo total de energia, e também satisfazer o desempenho desejado.

Para o caso de apenas um provedor de serviços, como um processador, a literatura de gerenciamento de energia é rica em técnicas e formalismos com soluções otimizadas [1, 10, 14–19]. A aplicação de tais técnicas é possível se forem obtidos modelos acurados de consumo de energia e desempenho de um processador para o projeto dessas políticas de gerenciamento de energia. De fato, o objetivo principal da minimização do consumo de energia em processadores é muito semelhante ao problema de otimização formulado por Benini et al. [10] no contexto de sistemas com gerenciamento de energia modelados como processos de decisão de Markov. O problema de otimização consiste na escolha de uma política de energia que minimiza o consumo de energia esperado, sob uma restrição de penalidade de desempenho. Esses autores definiram um modelo de processo de decisão de Markov de um servidor *Web* com dois processadores assimétricos, ou seja, com mais de um provedor de serviços. O modelo foi obtido compondo-se os modelos de processo de decisão de Markov de cada processador.

É importante considerar situações em que as políticas de gerenciamento de energia não são definidas a partir de modelos acurados de consumo de energia e desempenho de um processador. Por exemplo, os gerenciadores de energia *Ondemand* [20] do Sistema Operacional Linux e o *Processor Power Manager* (PPM) [21] do Sistema Operacional Windows são implementados para uma grande variedade de processadores, não sendo possível incorporar ou considerar, no projeto destes gerenciadores de energia, informações sobre modelos de consumo de energia de cada processador disponível no mercado. Assim, a principal informação considerada pelo gerenciador de energia nestes exemplos é a *utilização do processador*, isto é, a fração de tempo durante o qual o processador está ativo, executando instruções. Se a utilização do processador aumenta, é necessário escolher uma frequência de operação maior, para evitar que a utilização do processador se torne igual a 100%, causando uma penalidade de desempenho (o travamento na execução de um vídeo, a lentidão na execução de uma tarefa de uso intensivo de CPU etc.).

Se a utilização do processador diminui, é possível utilizar frequências menores, para assim economizar energia. Contudo esse tipo de política de gerenciamento de energia é implementado de maneira heurística. Por exemplo, no caso do *Ondemand* [20]: quando a utilização do processador é maior que ou igual a um parâmetro *up_threshold*, a máxima frequência disponível é escolhida; se a utilização do processador diminui, a frequência é gradualmente reduzida. Para uma política assim concebida, não é utilizado um modelo da relação entre a utilização do processador e a frequência de operação, o que pode resultar em penalidades de desempenho ou aumento do consumo de energia. Por exemplo, quando a utilização do processador diminui, em vez de reduzir a frequência gradualmente, esta poderia ser reduzida a níveis menores mais rapidamente sem que fosse comprometido o desempenho.

Uma política de gerenciamento de energia desenvolvida a partir do resultado do problema de otimização formulado por Benini et al. [10] pode ser considerada de *malha aberta*, pois, em tempo de execução os resultados obtidos não são comparados às restrições utilizadas na formulação do problema de otimização [19]. E assim, não se garante que tais restrições sejam respeitadas. Com o objetivo de garantir que o desempenho do sistema seja igual a um valor de referência de desempenho especificado, técnicas de controle em malha fechada têm sido aplicadas ao problema de gerenciamento dinâmico de energia em diferentes cenários, tais como: servidores [7, 22], sistemas embarcados de tempo real [23, 24], *networks-on-chip* [25], microprocessadores embarcados [26], reprodução de vídeo [27], processadores com escalonamento dinâmico de tensão e frequência [19], etc..

Wang et al. [7] desenvolveram algoritmos com realimentação para o gerenciamento de energia de um grupo de servidores usando escalonamento de frequência. Os algoritmos são orientados tanto para eficiência quanto para limitação do uso de potência, para prover uma boa solução de compromisso entre limitação de potência, eficiência, e desempenho de aplicações. Um algoritmo para desempenho foi projetado usando um controlador integral, onde a entrada é o erro observado na utilização de um servidor e a saída é a frequência do processador. Um algoritmo de limitação de potência foi também projetado usando um controlador integral, onde a entrada é o erro entre o orçamento de potência do servidor e o consumo atual, e a saída é a frequência do processador. No algoritmo para desempenho, não foi obtido um modelo da relação entre a utilização do processador e a frequência de operação, e não foram apresentados métodos para escolher os parâmetros do controlador integral.

Sridharan, Gupta e Mahapatra [23] introduziram algoritmos para realizar gerenciamento de energia de sistemas embarcados de tempo-real e para preservar a confiança geral do sistema. Foi usado um controlador proporcional para calcular o número adicional de cópias de um trabalho para assegurar as restrições (*constraints*) de confiança em tempo de execução. Ogras, Marculescu e Marculescu [25] dividiram uma rede-em-chip em múltiplas ilhas de tensão e frequência,

que são interconectadas por filas. Um modelo de espaço de estados da ocupação das filas foi derivado a partir da utilização das filas entre os domínios pelas ilhas de tensão e frequência. Controladores em malha fechada foram projetados para manter a ocupação das filas em níveis de referência por meio da mudança da velocidade de processamento em cada ilha de tensão e frequência. Verifica-se que as abordagens de Sridharan, Gupta e Mahapatra [23] e Ogras, Marculescu e Marculescu [25] são especificamente associadas às plataformas alvo e suas cargas de trabalho, não sendo possível aplicá-las ao caso geral de processadores com escalonamento dinâmico de tensão e frequência e com cargas de trabalho inicialmente desconhecidas e variantes no tempo.

Wang et al. [24] realizaram o controle da utilização de processadores em sistemas de tempo real distribuídos executando em ambientes imprevisíveis. Os objetivos de controle foram o atendimento a prazos fim-a-fim (*end-to-end deadlines*) e a minimização de potência. Esses autores consideraram apenas uma relação inversamente proporcional entre a frequência do processador e o tempo de execução das tarefas, enquanto, em geral o tempo de execução das tarefas apresenta componentes que não dependem da frequência do processador.

Tian et al. [26] desenvolveram uma metodologia de projeto de controle para gerenciamento de energia por meio de escalonamento dinâmico de tensão (*DVS*). Sua metodologia foi direcionada para microprocessadores embarcados que executam um conjunto conhecido de tarefas de tempo-real de controle. Um modelo analítico foi derivado para o sistema considerando o pior caso do tempo de execução das tarefas de tempo-real de controle, e o controlador foi projetado usando-se teoria de controle em malha fechada. O gerenciador de energia é composto por um controlador proporcional integral, onde a entrada é o erro no nível de utilização do processador, e a saída é a frequência do processador. Infelizmente, nesse trabalho a frequência do processador é considerada um valor contínuo, enquanto processadores reais têm apenas um número limitado de níveis de tensão e frequência.

Lu, Lach e Stan [27] abordaram o problema de gerenciamento de energia de um sistema de reprodução de vídeo. Um *buffer* foi colocado entre o decodificador e a memória do *display*, e um controlador proporcional integral é responsável pelo ajuste da velocidade do decodificador por meio de escalonamento dinâmico de tensão (*DVS*). A entrada do controlador é a ocupação do buffer, e a saída é a decisão de tensão do processador. Se o número de quadros no buffer é menor do que um limiar inferior ou maior que um limiar superior, então o controlador é usado para ajustar o número de quadros para o intervalo desejado. Verifica-se que essa abordagem está especificamente associada à carga de trabalho do sistema.

Minerick, Freeh e Kogge [28] implementaram, para o sistema operacional Linux, um mecanismo de realimentação denominado conservação de energia (*energy conservation*), que controla a frequência e tensão do processador de um sistema alimentado por bateria para assegurar

que a energia da bateria seja maior ou igual a um valor de referência de energia num instante de tempo específico no futuro. A principal desvantagem dessa técnica é que o desempenho pode ser reduzido abaixo de níveis aceitáveis porque a conservação de energia da bateria tem maior prioridade.

Devadas e Aydin [29] abordaram o problema de gerenciamento de energia de um sistema embarcado de tempo real constituído de um processador com escalonamento dinâmico de tensão e frequência (DVFS) e múltiplos dispositivos com gerenciamento dinâmico de energia, ou seja, estados do tipo ativo ou de baixo consumo de energia. Considerando um modelo de energia do sistema em função da frequência do processador e dos estados de energia dos dispositivos, esses autores propuseram um algoritmo para determinar a frequência ótima do processador e as transições de estados dos dispositivos para minimizar a energia do sistema e também satisfazer o prazo da tarefa executada no processador. A aplicação desse algoritmo requer, no entanto, o conhecimento *a priori* do modelo de energia do sistema, assim como a relação entre o tempo de execução da tarefa e a frequência do processador.

Luiz, Lima e Perkusich [19] aplicaram controle em malha fechada em um problema de gerenciamento de energia de um processador para que a penalidade de desempenho do sistema seguisse a restrição de penalidade de desempenho. A identificação do modelo do sistema foi realizada durante o projeto do gerenciador de energia, sendo utilizado para isso o pior caso para a carga de trabalho. Luiz et al. [30, 31] consideraram que, se a carga de trabalho é conhecida *a priori*, então o projeto do controlador pode ser otimizado. Então, um algoritmo de estimação de mínimos quadrados foi aplicado para obter-se um modelo da dinâmica do sistema com gerenciamento de energia quando a carga de trabalho é conhecida.

Contudo, um processador é tipicamente sujeito a cargas de trabalho inicialmente desconhecidas e variantes no tempo. Verifica-se portanto que técnicas de gerenciamento de energia baseadas em controle em malha fechada podem se beneficiar da adaptação contínua dos parâmetros do controlador. Então, Luiz, Perkusich e Lima [32] aplicaram controle adaptativo para o problema de gerenciamento de energia de um processador. O modelo do sistema com gerenciamento de energia foi estimado por meio de um algoritmo de estimação de mínimos quadrados recursivos. Assim, um controlador linear geral pôde ser reprojeto em tempo de execução, de acordo com as flutuações da carga de trabalho. Contudo, não foi avaliado o desempenho do algoritmo de estimação de mínimos quadrados recursivos para transições rápidas de cargas de trabalho, tais como períodos inativos de tempo seguidos de cargas de uso intensivo de CPU.

1.2 Relevância

Em situações em que não se dispõe de modelos acurados de potência e desempenho para o projeto de uma política de gerenciamento de energia, e a carga de trabalho do sistema é inicialmente *desconhecida* e *variante no tempo*, verifica-se portanto a necessidade de analisar o problema de gerenciamento de energia utilizando ferramentas de modelagem, estimação e controle para melhor explorar as características do comportamento dinâmico dos sinais de interesse para gerenciamento de energia de processadores: utilização do processador, penalidade de desempenho, frequência, tensão, potência, temperatura, etc..

Gerenciadores de energia do processador dirigido pelo sistema operacional, tais como o *Ondemand* [20] do Sistema Operacional Linux e o *Processor Power Manager* (PPM) [21] do Sistema Operacional Windows, são ainda formulados heurísticamente. Neste trabalho há o interesse na identificação de um modelo da relação entre a frequência de operação do processador e algum sinal de interesse para gerenciamento de energia, e.g. utilização do processador, para que as decisões do gerenciador de energia sejam mais eficientes em potência e desempenho. Por exemplo, no caso do *Ondemand* [20]: quando a utilização do processador se torna menor que um parâmetro *up_threshold* (após um período de carga de trabalho intensa), a frequência do processador é gradualmente reduzida. Se um modelo da utilização do processador fosse estimado em tempo de execução, poderia ser escolhida uma frequência menor que a que o gerenciador *Ondemand* escolheria, ainda satisfazendo o desempenho desejado. Considerando que a carga de trabalho do sistema é inicialmente *desconhecida* e *variante no tempo*, e que até mesmo a política de escalonamento de processos do sistema operacional pode influenciar a distribuição de carga de trabalho ao longo do tempo, esse *modelo do sinal de interesse para gerenciamento de energia do processador em função da frequência* deve ser estimado recursivamente em tempo de execução.

1.3 Problema abordado

O problema de interesse neste trabalho é: “*Como garantir um dado nível de desempenho e minimizar o consumo de energia de um processador provido de escalonamento dinâmico de tensão e frequência, por meio de um gerenciador de energia executado no sistema operacional, considerando que a carga de trabalho do sistema é inicialmente desconhecida e variante no tempo, a política de escalonamento de processos do sistema operacional é inicialmente desconhecida, e o consumo de energia de cada estado do processador é desconhecido?*”

1.4 Hipótese

Para realizar o projeto de um gerenciador de energia considerando o problema abordado neste trabalho, é necessária a aplicação de técnicas para as quais não há necessidade do conhecimento *a-priori* da carga de trabalho do sistema e o consumo de energia de cada estado do processador, e que garantam um nível de desempenho desejado e minimizem o consumo de energia. Então, para o problema abordado neste trabalho é considerada a hipótese de que: “*Técnicas de estimação e controle em malha fechada permitem garantir um determinado nível de desempenho e minimizar o consumo de energia.*”

1.5 Objetivo geral

O presente trabalho visa desenvolver técnicas de modelagem, estimação e controle para gerenciamento dinâmico de energia de processadores para garantir o atendimento aos níveis de desempenho desejados e minimizar o consumo de energia.

1.6 Objetivos específicos

- Analisar abordagens básicas de gerenciamento dinâmico de energia;
- aplicar controle em malha fechada a um indicador de desempenho de um sistema com gerenciamento de energia;
- aplicar técnicas de modelagem, estimação e controle a um sistema com gerenciamento de energia.

1.7 Organização do Texto

No Capítulo 2 é apresentada a base conceitual da área de gerenciamento de energia, com conceitos gerais que são aplicados ao longo de todo o texto. É realizada inicialmente uma revisão da abordagem básica de gerenciamento dinâmico de energia introduzida por Benini et al. [10] e Chung et al. [1], que é eficiente para sistemas computacionais que podem ser modelados por meio de processos de decisão de Markov. Tomando como motivação a necessidade de estimar cargas de trabalho inicialmente desconhecidas e não-estacionárias, é apresentada uma importante contribuição deste trabalho à abordagem básica com processos de decisão de Markov: uma

técnica de estimação da carga de trabalho baseada em janelas deslizantes de múltiplos comprimentos (*JMC*) [18]. Enquanto a abordagem proposta por Chung et al. [1], baseada em janelas de comprimento fixo, está sujeita aos efeitos intensos de erro de amostragem e atraso de identificação, a contribuição deste trabalho, baseada em janelas deslizantes de múltiplos comprimentos, se beneficia tanto do pequeno erro de amostragem de comprimentos de janela grandes e o pequeno atraso de identificação de comprimentos de janela pequenos. Foi demonstrado como essa técnica pode ser combinada com qualquer método para calcular as estimativas, por exemplo, estimativas de máxima verossimilhança [1], ou estimação fraca baseada em aprendizado estocástico [17].

Como a técnica introduzida por Benini et al. [10] oferece garantias apenas para os valores médios de desempenho do sistema, foi discutida no Capítulo 3 a necessidade da aplicação de *controle em malha fechada* para garantir o atendimento aos níveis de desempenho desejados. Essa discussão é ainda realizada no contexto de sistemas computacionais modelados como processos de decisão de Markov. Uma contribuição deste trabalho foi a obtenção de um modelo dinâmico de tempo discreto a partir do modelo em processos de decisão de Markov do sistema com gerenciamento de energia, e a aplicação de técnicas de identificação de sistemas [19]. A identificação do modelo do sistema é realizada em tempo de projeto, sendo utilizado o pior caso para a carga de trabalho. Com base nesse modelo, foi realizado o controle em malha fechada de um indicador de desempenho do sistema com gerenciamento de energia, resultando em políticas de energia que superaram as abordagens estocásticas apresentadas no Capítulo 2.

Contudo, na técnica apresentada no Capítulo 3, a identificação do modelo do sistema e o projeto do controlador ainda foram realizados utilizando-se o modelo de processos de decisão de Markov do sistema, que precisa ser conhecido *a priori*. Nesta técnica, é utilizado apenas um modelo do sistema obtido em tempo de projeto, o qual pode ser consideravelmente diferente se o sistema é submetido a cargas de trabalho variantes no tempo. Então, no Capítulo 4, foi aplicada a abordagem de controle adaptativo para realizar a estimação do modelo do sistema em tempo de execução. Para essa abordagem, não foi necessário um modelo *a priori* do sistema com gerenciamento de energia. Finalmente, no Capítulo 5 são apresentadas as conclusões e os trabalhos futuros.

Capítulo 2

Abordagem estocástica para gerenciamento de energia

2.1 Introdução

Um sistema com gerenciamento de energia é modelado por Benini et al. [10] como um *provedor de serviços* (PS), um *requisitante de serviços* (RS), uma *fila de serviços* (FS), e um *gerenciador de energia* (GE), como apresentado na Fig. 2.1. O provedor de serviços processa as requisições de um único *requisitante de serviços* (RS), e possui vários estados de operação, cada um com uma taxa de serviço. É possível que haja uma *fila de serviços* (FS) que armazena requisições que ainda não puderam ser atendidas pelo provedor de serviços, ou porque ele está ocupado ou está com taxa de serviço nula.

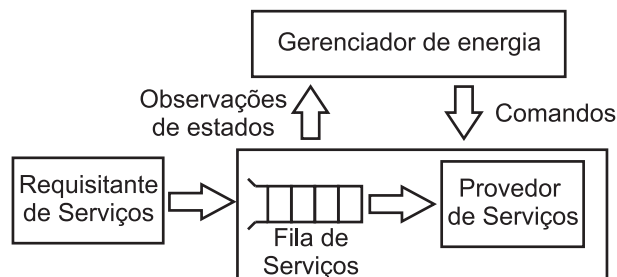


Figura 2.1: Componentes do modelo de um sistema com gerenciamento de energia.

Theocharous et al. [33] indicam que melhorias significativas no desempenho de técnicas de gerenciamento de energia podem ser obtidas por meio de modelos em processos estocásticos. Cheung e Xu [34], Qiu, Tan e Wu [35], e Tan e Qiu [36] consideram que a carga de trabalho possui estados parcialmente observáveis e/ou número desconhecido de estados. Por exemplo, o estado de um usuário de um computador de propósito geral não pode ser estimado somente

pelo uso do teclado ou *mouse*, pois o usuário pode apenas estar lendo um texto. Nessa situação, caso a política de gerenciamento de energia decida desligar o monitor para economizar energia, o usuário pode ficar irritado. O problema de gerenciamento de energia precisa então tratar a incerteza sobre a carga de trabalho, que é em alguns casos modelada como um modelo escondido de Markov (HMM).

Para o caso em que não há incertezas nas observações dos estados da carga de trabalho, como por exemplo as operações de leitura e escrita em um disco rígido, Benini et al. [10] apresentaram uma técnica que consiste de um modelo estocástico baseado em processos de decisão de Markov para a formulação da política de energia. É obtida uma solução exata do problema de otimização da política de energia em tempo polinomial resolvendo-se um problema de otimização linear. O problema de otimização é apenas resolvido para o caso estacionário, o que em grande quantidade dos problemas reais de gerenciamento dinâmico de energia pode ser uma desvantagem, dado que é difícil ou impossível caracterizar previamente a carga de trabalho do sistema, devido à não-estacionaridade.

Ren, Krogh e Marculescu [37] particularizam o problema de identificação da carga de trabalho, propondo um modelo em que o gerenciador de energia solicita do sistema operacional a informação sobre qual o processo que gera as requisições, cuja carga de trabalho é conhecida. Cada tipo de carga de trabalho é denominado de “modo” do sistema. Esta técnica é, no entanto, restrita a sistemas em que o gerenciador de energia pode solicitar do sistema o “modo” da carga de trabalho. Chung et al. [1] apontaram desvantagens na formulação de gerenciamento dinâmico de energia (GDE) como um problema estocástico ótimo de controle quando é apenas considerado o caso estacionário, como realizado por Benini et al. [10]. Assim Chung et al. [1] apresentaram um esquema de GDE adaptativo utilizando janelas de comprimento fixo para estimar cargas de trabalho inicialmente desconhecidas, não-estacionárias, e também cujo “modo” é desconhecido. Os efeitos de erro de amostragem e atraso de identificação de uma carga de trabalho não-estacionária de um dispositivo alimentado por bateria foram analisados por Luiz, Perkusich e Lima [16]. Simulações usando um modelo em cadeia de Markov de um sistema embarcado e um modelo analítico da bateria sugeriram que o comprimento de janela ótimo é aquele que resulta num erro médio quadrático mínimo durante a estimação dos parâmetros da carga de trabalho, assim aumentando a autonomia da bateria com uma penalidade de desempenho aceitável.

Neste capítulo, é apresentada inicialmente a abordagem de gerenciamento dinâmico de energia (GDE) introduzida por Benini et al. [10]. Essa abordagem é baseada em modelos estocásticos e é adequada apenas para cargas de trabalho estacionárias. Em seguida, são apresentadas as abordagens que lidam com cargas de trabalho desconhecidas e não-estacionárias utilizando janelas deslizantes de múltiplos comprimentos [18] para superar as limitações das

estimativas de máxima verossimilhança com número fixo de amostras (janela de comprimento fixo). Uma implementação dessa técnica de janelas deslizantes de múltiplos comprimentos é descrita, e esta técnica é comparada a outras abordagens que utilizam janelas de comprimento fixo ou parâmetros fixos.

2.2 Políticas para cargas de trabalho estacionárias

Nesta seção, considera-se que os componentes do sistema com gerenciamento de energia, como mostrado na Fig. 2.1, são modelados como cadeias de Markov estacionárias e discretas no tempo. Assume-se que as requisições são indistinguíveis, não havendo prioridade de serviço, e a fila de serviço possui tamanho finito. O gerenciador de energia toma decisões sobre o estado do provedor de serviços com base na história do sistema. Doravante, neste texto, o tempo é considerado discreto: $t = 0, 1, 2, \dots$. E o intervalo h entre dois valores consecutivos de tempo é denominado *período de amostragem*.

Definição 1 *Um requisitante de serviços (RS) é um par $(\mathcal{M}_{RS}, z(r))$ em que:*

- \mathcal{M}_{RS} é uma cadeia de Markov estacionária com conjunto de estados $\mathcal{R} = \{r_i | i = 0, 1, \dots, (R - 1)\}$ e matriz de transição \mathbf{P}^{RS} ;
- $z(r)$ é uma função $z : \mathcal{R} \rightarrow \mathbb{N}$ que representa o número de requisições lançadas por período pelo requisitante de serviços quando no estado r .

Seja $s_r(t)$ o estado do requisitante de serviços no instante t . A probabilidade de transição $P(s_r(t) = r_j | s_r(t-1) = r_i)$ de um estado $r_i \in \mathcal{R}$ para um estado $r_j \in \mathcal{R}$ é denotada por p_{r_i, r_j}^{RS} . A probabilidade de que o requisitante de serviços esteja no estado r_j é

$$\pi_j \triangleq \lim_{t \rightarrow \infty} P[s_r(t) = r_j]$$

Para o caso de um requisitante de serviços estacionário com dois estados, as expressões para π_0 e π_1 são mostradas em (2.1) e (2.2) respectivamente. Como mostrado na Fig. 2.2, a probabilidade π_1 de que o requisitante de serviços está no estado r_1 é uma função de ambas as probabilidades de transição p_{r_0, r_0}^{RS} e p_{r_1, r_1}^{RS} . Quanto maior a probabilidade π_1 , mais intensa a carga de trabalho é, porque a probabilidade de que uma requisição é passada num dado intervalo de amostragem aumenta.

$$\pi_0 = \frac{1 - p_{r_1, r_1}^{RS}}{2 - (p_{r_0, r_0}^{RS} + p_{r_1, r_1}^{RS})} \quad (2.1)$$

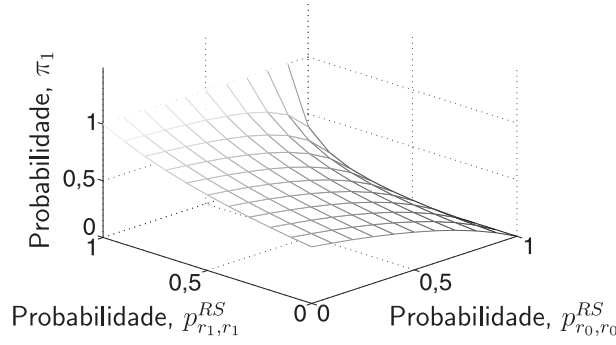


Figura 2.2: Probabilidade π_1 de que um requisitante de serviços com conjunto de estados $\{r_0, r_1\}$ está no estado r_1 .

$$\pi_1 = \frac{1 - p_{r_0, r_0}^{RS}}{2 - (p_{r_0, r_0}^{RS} + p_{r_1, r_1}^{RS})} \quad (2.2)$$

Definição 2 Um provedor de serviços (PS) é uma tripla $(\mathcal{M}_{PS}(a), b(s,a), c(s,a))$ em que:

- $\mathcal{M}_{PS}(a)$ é uma cadeia de Markov estacionária controlável com conjunto de estados $\mathcal{S} = \{s_i | i = 1, 2, \dots, S\}$, conjunto de comandos $\mathcal{A} = \{a_i | i = 1, 2, \dots, A\}$ e matriz de probabilidades de transição $\mathbf{P}^{PS}(a)$;
- $b(s,a)$ é uma função $b : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, que representa uma taxa de serviço associada a cada estado $s \in \mathcal{S}$ e comando $a \in \mathcal{A}$. Tal taxa de serviço é a probabilidade de o PS completar o atendimento de uma requisição de serviço num período, dado que o PS está no estado s e comando a foi passado no início do período;
- $c(s,a)$ é uma função $c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, que representa um indicador de consumo de potência, corrente ou energia associada a cada estado $s \in \mathcal{S}$ e comando $a \in \mathcal{A}$, i.e. o consumo (em unidades arbitrárias: W, A ou J) do PS num período, dado que o comando a foi tomado e o PS está no estado s .

Definição 3 Uma fila de serviço (FS) de tamanho $Q - 1$ é uma cadeia de Markov estacionária controlável $\mathcal{M}_{FS}(a, s, r)$ com conjunto de estados $\mathcal{Q} = \{q_i | i = 0, 1, \dots, (Q - 1)\}$, variável de controle $\theta = (a, s, r)$, conjunto de comandos $\Theta = \mathcal{A} \times \mathcal{S} \times \mathcal{R}$ e matriz de transição $\mathbf{P}^{FS}(a, s, r)$. As probabilidades de transição são definidas em (2.3).

$$p_{q_i, q_j}^{FS}(a, s, r) = \begin{cases} 1 - b(s,a), & \text{se } j = i + z(r) \text{ e } 0 < i + z(r) < Q \\ b(s,a), & \text{se } j = i + z(r) - 1 \text{ e } 0 \leq i + z(r) - 1 < Q \\ 1, & \text{se } j = i = 0 \text{ e } z(r) = 0 \\ 1, & \text{se } j = Q - 1 \text{ e } i + z(r) \geq Q \\ 0, & \text{caso contrário} \end{cases} \quad (2.3)$$

Os casos da expressão (2.3) podem ser melhor entendidos da seguinte forma:

- Caso $j = i + z(r)$ e $0 < i + z(r) < Q$: a probabilidade de transição é $1 - b(a,s)$ pois o PS não serviu nenhuma requisição ($j = i + z(r)$), e o número de requisições que chegaram ($z(r)$) mais as que já estavam na fila (i) não excede o tamanho da fila ($Q - 1$). Na Fig. 2.3 é mostrado um exemplo onde um RS gera uma requisição que é armazenada numa FS de comprimento 2 que já possuía uma requisição pendente, pois o PS ainda não atendeu nenhuma delas;
- Caso $j = i + z(r) - 1$ e $0 \leq i + z(r) - 1 < Q$: a probabilidade de transição é $b(a,s)$ pois o PS serviu uma requisição ($j = i + z(r) - 1$), e a soma do número de requisições que estavam na fila (i) mais as que chegaram ($z(r)$) menos a que foi servida não excede o tamanho da fila ($Q - 1$). Na Fig. 2.4 é mostrado um exemplo onde um RS gera uma requisição que é armazenada numa FS de comprimento 2 que já possuía uma requisição pendente a qual é atendida pelo PS, e a cadeia de Markov da FS permanece no mesmo estado;
- Caso $j = i = 0$ e $z(r) = 0$: a probabilidade de transição é 1 pois este é o caso limite em que a fila estava vazia e assim permanece ($j = i = 0$), pois não apareceu nenhuma nova requisição ($z(r) = 0$). Na Fig. 2.5 é mostrado um exemplo onde um RS não gera nenhuma requisição a ser armazenada numa FS de comprimento 2 que estava vazia e assim permanece;
- Caso $j = Q - 1$ e $i + z(r) \geq Q$: a probabilidade de transição é 1 pois este é o caso limite em que a fila se torna cheia ($j = Q - 1$) e a soma do número de requisições que estavam na fila (i) mais as que chegaram ($z(r)$) excede o tamanho da fila. Uma perda de requisição ocorre sempre que uma nova requisição do RS não pode ser colocada na fila porque esta já está cheia. Na Fig. 2.6 é mostrado um exemplo com um RS que gera duas requisições e uma FS de comprimento 2 que já possuía uma requisição pendente; a fila fica cheia e a requisição que excedeu o comprimento da fila ou é atendida pelo PS ou é perdida;

O estado do sistema $x = (s, r, q)$ é a composição dos estados do provedor de serviços, requisitante de serviços e fila de serviços respectivamente. Então, o sistema tem o conjunto de estados $\mathcal{X} = \mathcal{S} \times \mathcal{R} \times \mathcal{Q}$ com $X = S \cdot R \cdot Q$ estados. O comando a a ser submetido pelo gerenciador de energia (GE) é uma variável aleatória com distribuição de probabilidade discreta $\delta^{(t)} = (p_{a_0}, p_{a_1}, \dots, p_{a_{A-1}})$, denominada a *decisão* do gerenciador de energia, que associa um valor de probabilidade p_a para cada comando $a \in \mathcal{A}$ no tempo t . A política π do gerenciador de energia é um conjunto de decisões $[\delta^{(0)}, \delta^{(1)}, \dots]$ no tempo $t = 0, 1, \dots$.

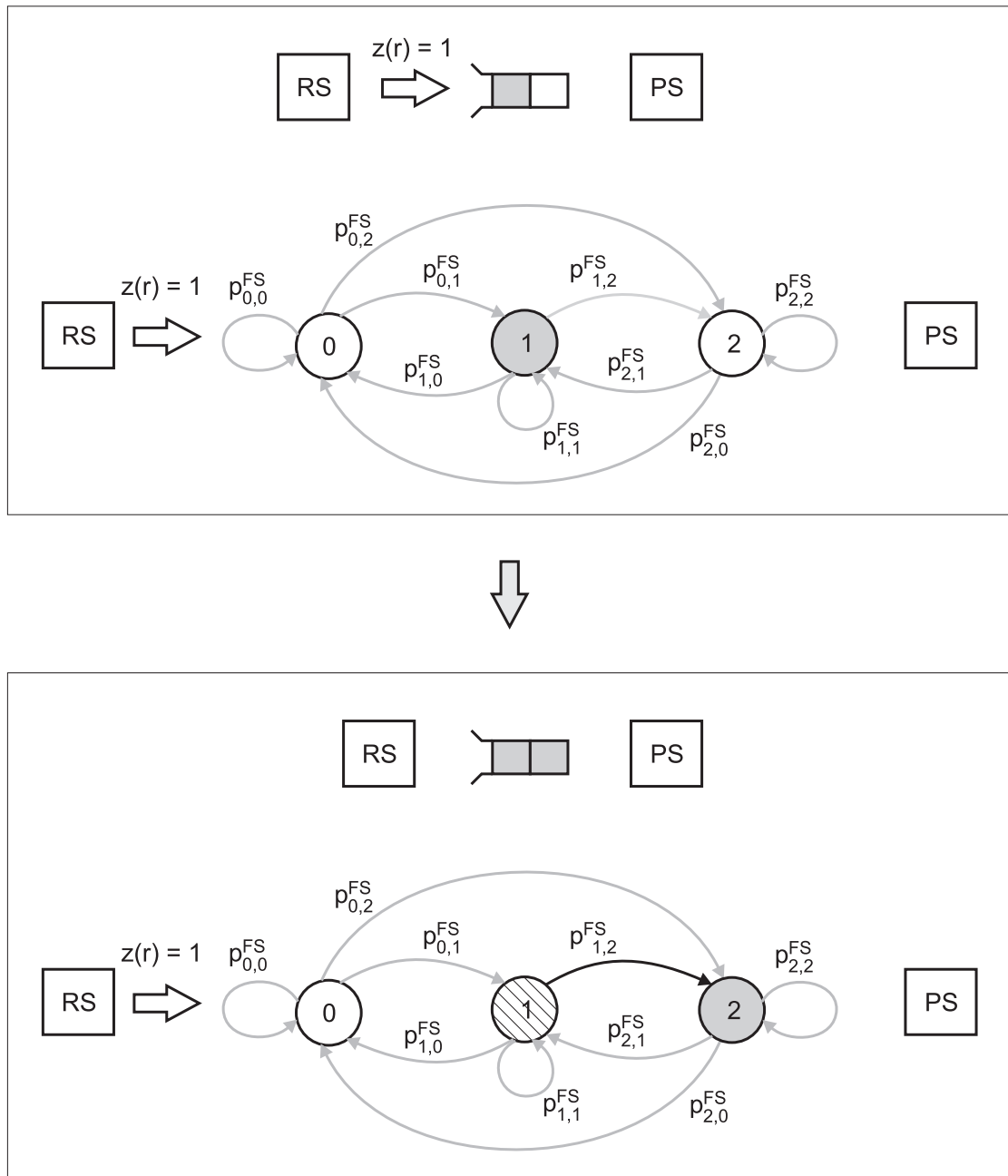


Figura 2.3: O PS não serviu nenhuma requisição e o número de requisições que chegaram mais as que já estavam na fila não excede o tamanho da fila.

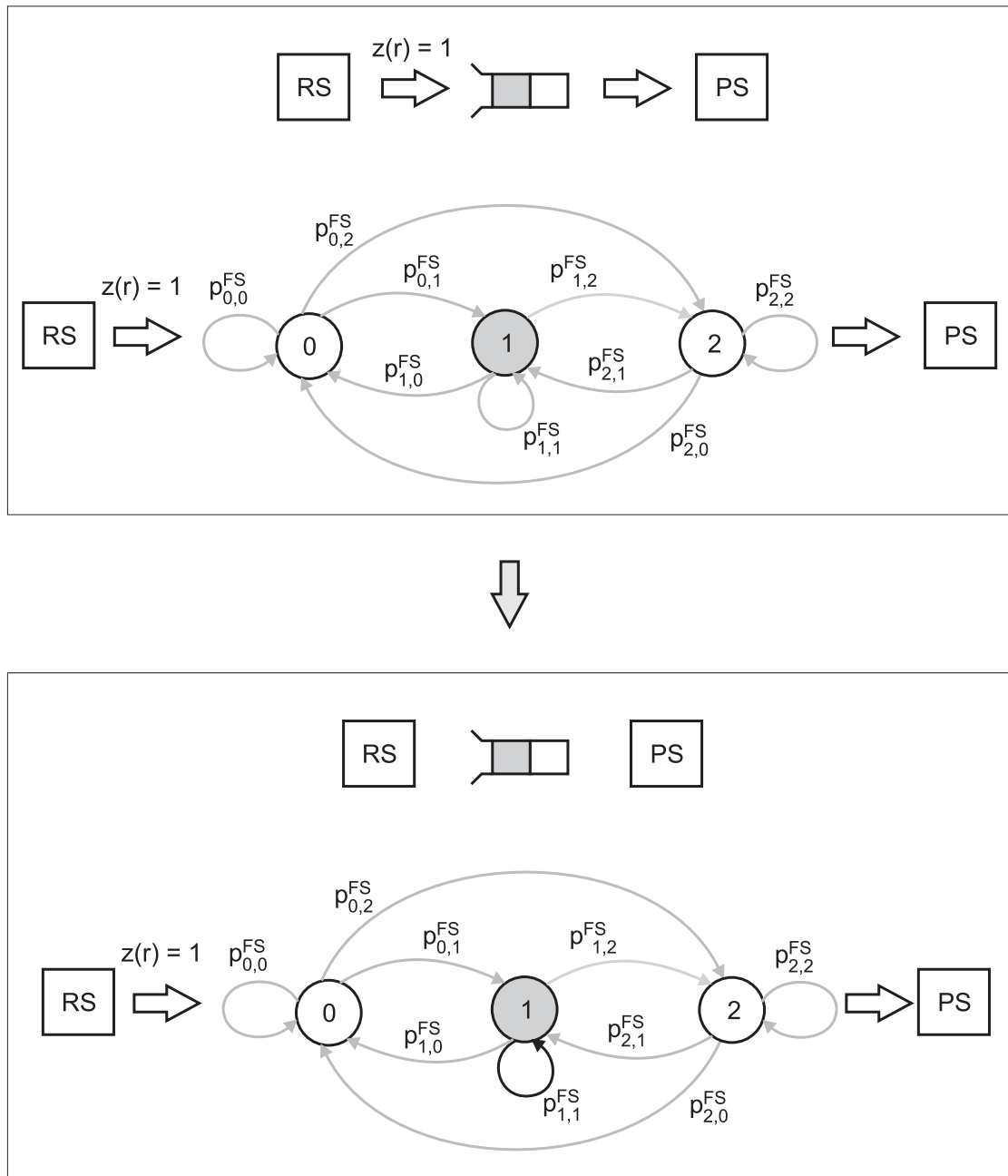


Figura 2.4: O PS serviu uma requisição e a soma do número de requisições que estavam na fila mais as que chegaram menos a que foi servida não excede o tamanho da fila.

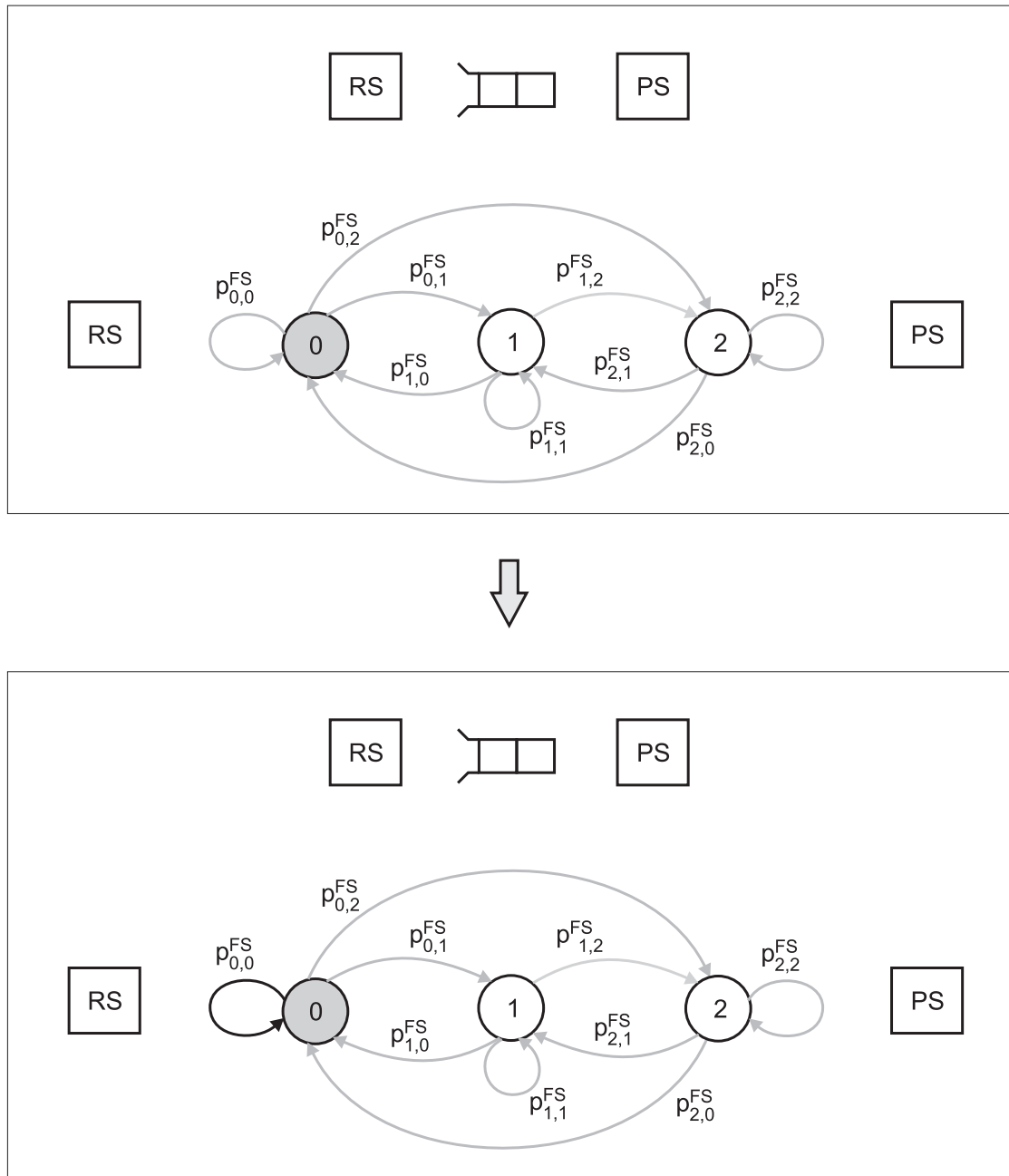


Figura 2.5: A fila estava vazia e assim permanece, pois não apareceu nenhuma nova requisição.

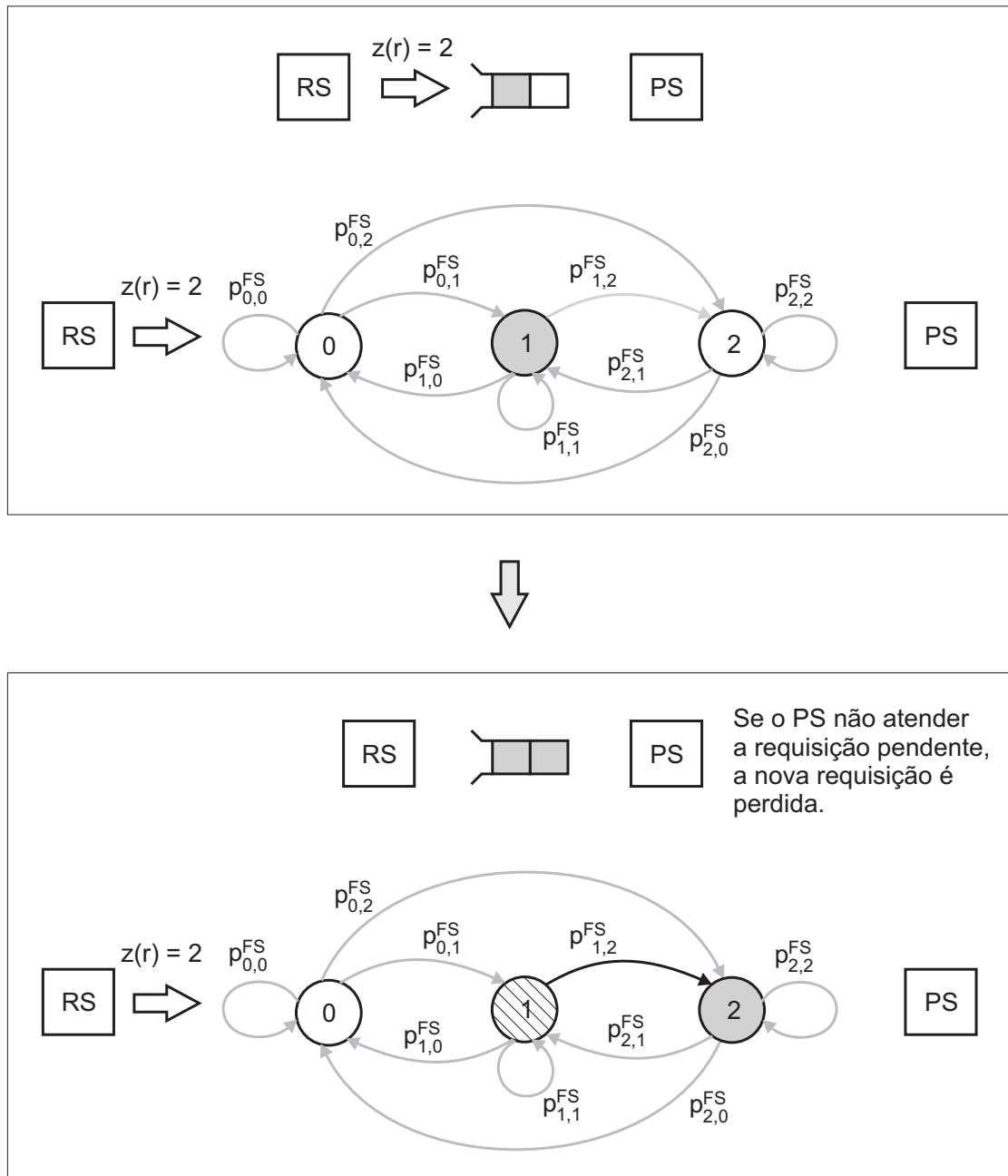


Figura 2.6: A fila se torna cheia e a soma do número de requisições que estavam na fila mais as que chegaram excede o tamanho da fila.

A matriz de transição $P(a)$ do sistema quando o comando $a \in \mathcal{A}$ é escolhido é uma matriz $X \times X$ com cada elemento $p_{x_i, x_j}(a)$ como definido em (2.4). A matriz de transição $P_{\delta^{(t)}}$ do sistema quando a decisão $\delta^{(t)}$ é tomada no tempo t é definida em (2.5). A matriz de transição $P_\pi(t)$ do sistema do tempo 0 a t é definida em (2.6).

$$\begin{aligned} p_{x_i, x_j}(a) &= \text{Prob} \left[x_j = (s', r', q') \mid x_i = (s, r, q), a \right] = \\ &= p_{s, s'}^{PS}(a) p_{r, r'}^{RS} p_{q, q'}^{FS}(a, s, r) \end{aligned} \quad (2.4)$$

$$P_{\delta^{(t)}} = \sum_{p_a \in \delta^{(t)}} p_a P(a) \quad (2.5)$$

$$P_\pi(t) = \prod_{\tau=0}^{t-1} P_{\delta^{(\tau)}} \quad (2.6)$$

Supõe-se que o sistema será usado até certo *horizonte de tempo* T_f , que é finito e aleatório. Por exemplo, o instante de tempo em que a bateria que alimenta o sistema descarrega. Então, cada probabilidade de transição do sistema deve ser multiplicada por um *fator de desconto* $0 < \xi < 1$, e o horizonte de tempo T_f é modelado como uma variável aleatória geometricamente distribuída com média $E[T_f] = (1 - \xi)^{-1}$. Uma vez que $E[T_f]$ é estimado, o fator de desconto $\xi = 1 - 1/E[T_f]$. Usando o fator de desconto ξ , a matriz de transição $P_\pi(t)$ do sistema de 0 a t é $P_\pi(t) = \prod_{\tau=0}^{t-1} (\xi P_{\delta^{(\tau)}}) = \xi^{t+1} \prod_{\tau=0}^{t-1} P_{\delta^{(\tau)}}$. Seja $p(0)$ a distribuição inicial de probabilidades de estados do sistema no instante de tempo 0. Então a distribuição de probabilidades de estados do sistema no tempo t é

$$p(t) = p(0) P_\pi(t). \quad (2.7)$$

Uma *política estacionária* $\pi = [\delta, \delta, \dots]$ usa a mesma decisão δ a todo instante de tempo t . A decisão δ de uma política estacionária depende do estado do sistema $x \in \mathcal{X}$, então ela é denotada por δ_x . Se o sistema está no estado $x \in \mathcal{X}$, a decisão δ_x é tomada, e o consumo esperado é $c(x, \delta_x) = \sum_{p_a \in \delta_x} p_a c(s, a)$. A penalidade de desempenho do sistema no estado $x \in \mathcal{X}$ é $d(x)$ e é definida pelo usuário, e.g., $d(x) = q$ é o número de requisições pendentes na fila. Os vetores de consumo e penalidade de desempenho da decisão δ são mostrados em (2.10). O consumo esperado e penalidade de desempenho esperada do provedor de serviços (PS) no instante t sob a política π são apresentadas em (2.8) e (2.9) respectivamente.

$$E_\pi [\mathbf{c}_\delta(t)] = p(t) c_\delta \quad (2.8)$$

$$E_\pi [\mathbf{d}_\delta(t)] = p(t) d_\delta \quad (2.9)$$

$$c_\delta = \begin{pmatrix} c(x_0, \delta_{x_0}) \\ \vdots \\ c(x_{X-1}, \delta_{x_{X-1}}) \end{pmatrix} \quad d_\delta = \begin{pmatrix} d(x_0) \\ \vdots \\ d(x_{X-1}) \end{pmatrix} \quad (2.10)$$

O *problema de otimização* de política (*PO*), expresso em (2.11), consiste na escolha da política estacionária π que minimiza o consumo esperado para uma dada restrição de penalidade de desempenho, onde D é a penalidade de desempenho máxima aceitável.

$$\begin{aligned} \mathbf{PO} : \quad & \min_{\pi} \sum_{t=0}^{\infty} E_{\pi} [\mathbf{c}_{\delta}(t)] \\ & \text{tal que } \sum_{t=0}^{\infty} E_{\pi} [\mathbf{d}_{\delta}(t)] \leq D \end{aligned} \quad (2.11)$$

Como descrito por Benini et al. [10], o problema de otimização em (2.11) pode ser convertido no problema de programação linear em (2.12) e o resultado é uma política estacionária, expressa como uma matriz, denominada *tabela de decisão*, que tem X linhas, i.e. o número de estados do sistema, e A colunas, i.e. o número de comandos que podem ser submetidos pelo gerenciador de energia para o provedor de serviços. Cada linha da tabela de decisão é uma decisão δ_x associada ao estado do sistema $x \in \mathcal{X}$. Então, cada elemento da tabela de decisão é a probabilidade de o gerenciador de energia passar um comando $a \in A$ quando o sistema está no estado $x = (s,r,q)$. Em (2.12) as variáveis $f_{x,a}$ são denominadas *frequências de estado-ação* e cada uma representa o número esperado de vezes que, estando o sistema no estado x , o comando a é escolhido pelo gerenciador de energia.

$$\begin{aligned} \mathbf{PL1} : \quad & \min \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} f_{x,a} c(x,a) \\ & \text{tal que } \sum_{a \in \mathcal{A}} f_{x,a} - \xi \sum_{y \in \mathcal{X}} \sum_{a \in \mathcal{A}} p_{y,x}(a) f_{y,a} = p_x^{(1)}, \forall x \in \mathcal{X} \\ & \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} f_{x,a} d(x) \leq D \\ & f_{x,a} \geq 0, \forall x \in \mathcal{X}, a \in \mathcal{A} \end{aligned} \quad (2.12)$$

Para resolver o programa linear em (2.12), é possível utilizar a ferramenta *linprog* do *Toolbox* de Otimização do Matlab [38], cuja implementação para otimização em larga escala é baseada também num algoritmo de ponto interior. A probabilidade $m_{x,a}$ de o gerenciador de energia escolher o comando a quando o sistema está no estado x é estimada em (2.13).

$$m_{x,a} = \frac{f_{x,a}}{\sum_{a' \in \mathcal{A}} f_{x,a'}} \quad (2.13)$$

As probabilidades $m_{x,a}$ são organizadas na tabela de decisão M_{π} em (2.14), em que as linhas variam com os estados $x_0, x_1 \dots x_{X-1}$ do sistema e as colunas com os comandos $a_0, a_1, \dots a_{A-1}$. Em tempo de execução, a política de energia toma a linha de M_{π} correspondente ao estado atual x do sistema: $(m_{x,a_0} \quad m_{x,a_1} \quad \dots \quad m_{x,a_{A-1}})$, denominada *decisão*. Essa linha representa a

distribuição de probabilidades de o GE passar o comando a com probabilidade $m_{x,a}$ dado que o estado do sistema é x .

$$M_\pi = \begin{bmatrix} m_{x_0,a_0} & m_{x_0,a_1} & \dots & m_{x_0,a_{A-1}} \\ m_{x_1,a_0} & m_{x_1,a_1} & \dots & m_{x_1,a_{A-1}} \\ \vdots & \vdots & \vdots & \vdots \\ m_{x_{X-1},a_0} & m_{x_{X-1},a_1} & \dots & m_{x_{X-1},a_{A-1}} \end{bmatrix} \quad (2.14)$$

Deseja-se escolher o comando a que o GE vai passar ao PS. Para isso é realizado o sorteio de um número aleatório R uniformemente distribuído no intervalo $[0, 1]$ e escolhido o comando a_i tal que $\sum_{k=0}^{i-1} m_{x,a_k} < R \leq \sum_{k=0}^i m_{x,a_k}$ caso $m_{x,a_0} < R \leq 1$, ou a_0 caso $0 \leq R \leq m_{x,a_0}$, como mostrado na Fig. 2.7.

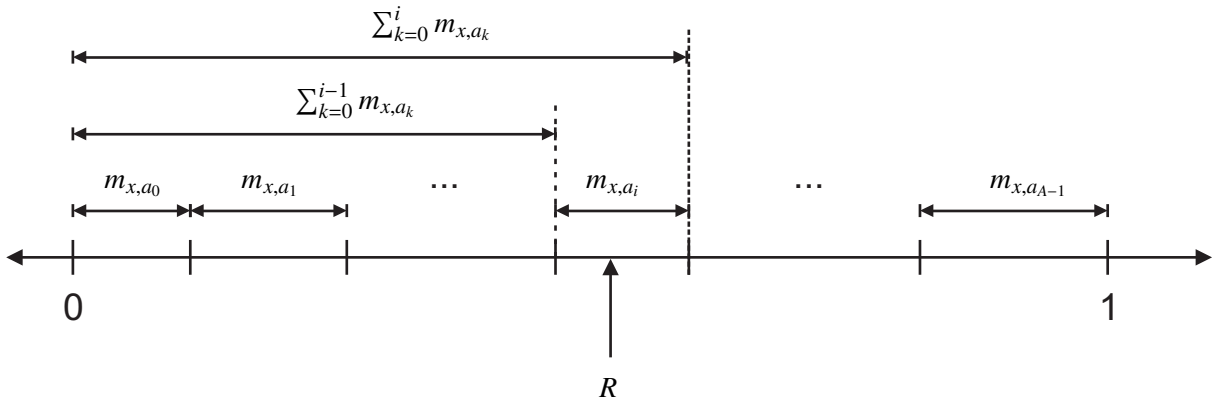


Figura 2.7: Determinação do comando do gerenciador de energia.

Exemplo 1 Considere um provedor de serviços (PS) $(M_{PS}(a), b(s,a), c(s,a))$ onde: $M_{PS}(a)$ é uma cadeia de Markov estacionária e controlável com conjunto de estados $\mathcal{S} = \{s_0, s_1\}$; conjunto de comandos $\mathcal{A} = \{a_0, a_1\}$, significando mudar para s_0 e mudar para s_1 , respectivamente; matriz de transição $\mathbf{P}^{PS}(a)$ em (2.15) e (2.16); a matriz $c(s,a)$ em (2.17) especifica o consumo de corrente (em mA) do provedor de serviços quando no estado s e quando o comando a é passado pelo gerenciador de energia; e a matriz $b(s,a)$ em (2.18) especifica a taxa de serviço do provedor de serviços quando no estado s e quando o comando a é passado pelo gerenciador de energia, i.e., a probabilidade de completar o atendimento de uma requisição num intervalo de amostragem. O estado s_0 tem maior consumo e melhor desempenho do que o estado s_1 . As probabilidades de transição do provedor de serviços tem os valores apresentados em (2.15) e (2.16), porque, no sistema modelado neste exemplo, os tempos de transição entre os estados do provedor de serviços são desprezíveis com respeito ao intervalo de amostragem. Pela mesma razão, o consumo de corrente $c(s,a)$ em (2.17) e a taxa de serviço $b(s,a)$ em (2.18) somente dependem do estado atual s do provedor de serviços. O modelo em cadeia de Markov do provedor

de serviços é mostrado na Fig. 2.8.

$$P^{PS}(a_0) = \begin{matrix} & s_0 & s_1 \\ s_0 & \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \\ s_1 & \end{matrix} \quad (2.15)$$

$$P^{PS}(a_1) = \begin{matrix} & s_0 & s_1 \\ s_0 & \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \\ s_1 & \end{matrix} \quad (2.16)$$

$$c(s,a) = \begin{matrix} & a_0 & a_1 \\ s_0 & \begin{bmatrix} 318,7 & 318,7 \\ 294,9 & 294,9 \end{bmatrix} \\ s_1 & \end{matrix} \quad (2.17)$$

$$b(s,a) = \begin{matrix} & a_0 & a_1 \\ s_0 & \begin{bmatrix} 1 & 1 \\ 0,3125 & 0,3125 \end{bmatrix} \\ s_1 & \end{matrix} \quad (2.18)$$

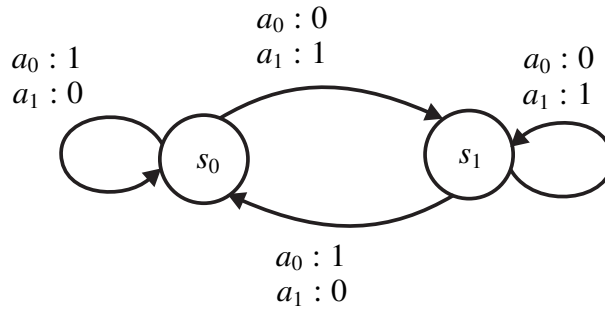


Figura 2.8: Cadeia de Markov representando o provedor de serviços (PS).

Agora considere um requisitante de serviços (RS) $(\mathcal{M}_{RS}, z(r))$ com conjunto de estados $\mathcal{R} = \{r_0, r_1\}$ com função $z(r)$ identificando o número de requisições que o requisitante de serviços lança por período de amostragem quando no estado r . Para o requisitante de serviços desse exemplo, utiliza-se uma correspondência um-a-um entre as imagens de z e os estados do requisitante de serviços: $z(r_0) = 0$, $z(r_1) = 1$. A matriz de transição \mathbf{P}^{RS} é mostrada em (2.19) e o modelo em cadeia de Markov do requisitante de serviços, na Fig. 2.9.

$$P^{RS} = \begin{matrix} & r_0 & r_1 \\ r_0 & \begin{bmatrix} 0,6 & 0,4 \\ 0,5 & 0,5 \end{bmatrix} \\ r_1 & \end{matrix} \quad (2.19)$$

A fila de serviços (FS) $\mathcal{M}_{FS}(a,s,r)$ de comprimento $Q = 1$ tem o conjunto de estados $\mathcal{Q} = \{q_0, q_1\}$, i.e., em q_0 a fila de serviços está vazia, e em q_1 a fila de serviços tem uma requisição

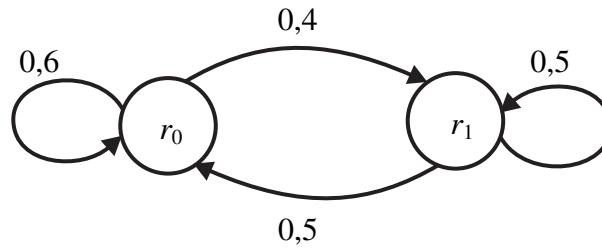


Figura 2.9: Cadeia de Markov representando o requisitante de serviços (RS).

pendente. A matriz de transição \mathbf{P}^{FS} é mostrada em (2.20) e (2.21). O modelo em cadeia de Markov da fila de serviços é mostrado na Fig. 2.10.

$$P^{FS}(a, s, r = 0) = \begin{matrix} & q_0 & q_1 \\ q_0 & \begin{bmatrix} 1 & 0 \\ b(s, a) & 1 - b(s, a) \end{bmatrix} \\ q_1 & \end{matrix} \quad (2.20)$$

$$P^{FS}(a, s, r = 1) = \begin{matrix} & q_0 & q_1 \\ q_0 & \begin{bmatrix} b(s, a) & 1 - b(s, a) \\ 0 & 1 \end{bmatrix} \\ q_1 & \end{matrix} \quad (2.21)$$

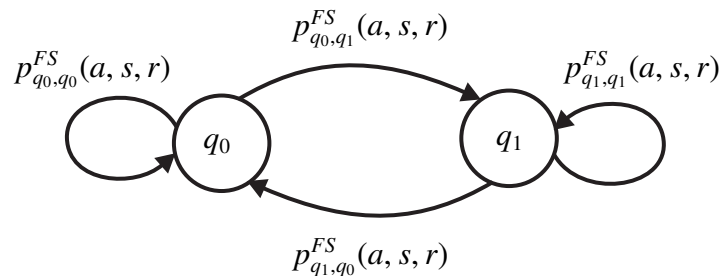


Figura 2.10: Cadeia de Markov representando a fila de serviços (FS).

Deseja-se simular o modelo do sistema durante um horizonte de tempo de $T_f = 10000$. Então, o fator de desconto $\xi = 1 - 1/10000 = 0,9999$. A restrição de penalidade de desempenho definida para esse sistema é a perda de requisição média L . A perda de requisição acontece se uma nova requisição é gerada pelo requisitante de serviços (RS), mas a fila de serviços (FS) já está cheia. A máxima perda de requisições média é 10%, então a restrição de penalidade de desempenho é $L = 0,10T_f = 1000$. Seja E uma variável aleatória com espaço amostral $\{0, 1\}$ e distribuição de probabilidades: $p(E = 0) = 1 - b(s, a)$ i.e. $E = 0$ quando o provedor de serviços não serve a requisição; e $p(E = 1) = b(s, a)$ i.e. $E = 1$ quando o provedor de serviços serve uma requisição. Seja $e(t)$ uma realização de E no instante t . Para cada estado do sistema $x = (s, r, q)$, o número de requisições pendentes na fila $d(x)$ é igual a q , i.e. o estado da fila. Considerando a perda de requisições $l(t)$ em (2.22), existe uma perda de requisições se o número de requisições vindouras ($z(r(t))$) mais a ocupação da fila ($q(t)$) menos “uma

possível requisição servida” ($e(t)$) excede a capacidade da fila $Q - 1$. Na prática, uma perda de requisição não representa uma falta de serviço real, mas a condição indesejável de muitas requisições esperando para serem servidas [10].

$$l(t) = \begin{cases} 1, & \text{se } z(r(t)) + q(t) - e(t) > Q - 1 \\ 0, & \text{caso contrário} \end{cases} \quad (2.22)$$

O problema de otimização em (2.11) é convertido em um problema de programação linear em [10] e a solução é a tabela de decisão mostrada em (2.23).

$$M_\pi = \begin{matrix} & a_0 & a_1 \\ \begin{matrix} (s_0, r_0, q_0) \\ (s_0, r_0, q_1) \\ (s_0, r_1, q_0) \\ (s_0, r_1, q_1) \\ (s_1, r_0, q_0) \\ (s_1, r_0, q_1) \\ (s_1, r_1, q_0) \\ (s_1, r_1, q_1) \end{matrix} & \begin{pmatrix} 0,5433 & 0,4567 \\ 0,3811 & 0,6189 \\ 1,0000 & 0,0000 \\ 0,3253 & 0,6747 \\ 0,2748 & 0,7252 \\ 1,0000 & 0,0000 \\ 1,0000 & 0,0000 \\ 0,3988 & 0,6012 \end{pmatrix} \end{matrix} \quad (2.23)$$

O consumo de corrente médio para o sistema é 310,09mA, i.e. 2,7% menos do que se o sistema está sempre no estado s_0 . A perda de requisição média é 2,22% e é menor que a restrição de penalidade de desempenho. \triangle

Exemplo 2 Considere o mesmo provedor de serviços, fila de serviços e requisitante de serviços do Exemplo 1. O consumo de corrente e perda de requisição média serão analisadas para diferentes combinações das probabilidades de transição do requisitante de serviços, i.e., para diferentes cargas de trabalho. Como mostrado na Fig. 2.2, quanto maior a probabilidade π_1 de que o requisitante de serviços está no estado r_1 , mais intensa a carga de trabalho é. Dois exemplos de traços de estados do requisitante de serviços (RS) são mostrados na Fig. 2.11: (i) para as probabilidades de transição $p_{r_0, r_0}^{RS} = 0,1$ e $p_{r_1, r_1}^{RS} = 0,9$, tem-se que $\pi_1 = 0,9$; (ii) para as probabilidades de transição $p_{r_0, r_0}^{RS} = 0,9$ e $p_{r_1, r_1}^{RS} = 0,1$, tem-se que $\pi_1 = 0,1$. Para manter a penalidade de desempenho sob a restrição, i.e., manter o desempenho do sistema em níveis aceitáveis, o gerenciador de energia passa comandos para mudar o estado de energia do provedor de serviços, tanto aumentando a taxa de serviços para cargas de trabalho intensas ou diminuindo a taxa de serviço para cargas de trabalho menos intensas.

O modelo do sistema desse exemplo é simulado para diferentes requisitantes de serviço estacionários, com diferentes combinações de $p_{r_0, r_0}^{RS} = 0, 0,1, 0,2, \dots, 1,0$ e $p_{r_1, r_1}^{RS} = 0, 0,1, 0,2, \dots,$

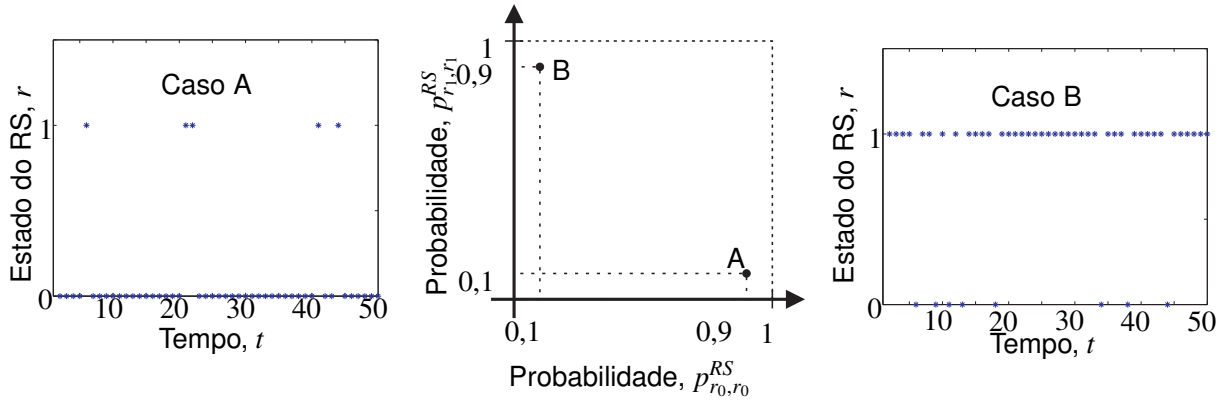


Figura 2.11: Exemplos de cargas de trabalho do sistema.

1,0, e os resultados de consumo de corrente (corrente média em mA) e penalidade de desempenho (perda de requisições média) são mostrados na Fig. 2.12. Para a carga de trabalho mais intensa, quando $p_{r_0, r_0}^{RS} = 0$ e $p_{r_1, r_1}^{RS} = 1$, a corrente média é 318,7mA porque o provedor de serviços está sempre no estado s_0 , que tem o maior consumo de corrente (318,7mA), mas tem a melhor taxa de serviço $b(s_0, a) = 1$ para $a = a_0, a_1$, i.e., quando o provedor de serviços está no estado s_0 , ele serve uma requisição no intervalo de amostragem atual com probabilidade 1. Então, a perda de requisições média é 0. Para a carga de trabalho menos intensa, quando $p_{r_0, r_0}^{RS} = 1$ e $p_{r_1, r_1}^{RS} = 0$, a corrente média é 294,9mA porque o provedor de serviços está sempre no estado s_1 que tem o menor consumo de corrente (294,9mA), mas tem a pior taxa de serviço $b(s_1, a) = 0,3125$ para $a = a_0, a_1$, i.e., quando o provedor de serviços está no estado s_1 , ele atende uma requisição no intervalo de amostragem atual com probabilidade 0,3125. Apesar disso, a perda de requisições média é 0 porque nenhuma requisição é lançada, dado que $p_{r_0, r_0}^{RS} = 1$ e $p_{r_1, r_1}^{RS} = 0$. Para algumas cargas de trabalho, a perda de requisições é maior que a restrição de desempenho, e as políticas otimizadas para essas cargas de trabalho não são aceitáveis, como mostrado na Fig. 2.13. Além disso, para algumas cargas de trabalho, a perda de requisições é menor ou igual à restrição de desempenho, i.e. as políticas satisfazem a restrição de penalidade de desempenho, mas não há garantias que a explorem totalmente para economizar mais energia.

Em geral, o plano $p_{r_0, r_0}^{RS} p_{r_1, r_1}^{RS}$ pode ser dividido em duas regiões: (i) uma para a qual $\pi_1 > 0,5 \Rightarrow (1 - p_{r_0, r_0}^{RS}) / [2 - (p_{r_0, r_0}^{RS} + p_{r_1, r_1}^{RS})] > 0,5 \Rightarrow p_{r_1, r_1}^{RS} > p_{r_0, r_0}^{RS}$ e relacionado a cargas de trabalho intensas; (ii) e uma para a qual $\pi_1 < 0,5 \Rightarrow (1 - p_{r_0, r_0}^{RS}) / [2 - (p_{r_0, r_0}^{RS} + p_{r_1, r_1}^{RS})] > 0,5 \Rightarrow p_{r_1, r_1}^{RS} < p_{r_0, r_0}^{RS}$ e relacionado a cargas de trabalho leves, como ilustrado na Fig. 2.14. \triangle

Benini et al. [10] destacaram que a qualidade da solução do problema de otimização de política depende da satisfação de várias hipóteses sobre o modelo do sistema, por exemplo: (i) a chegada de requisições de serviços podem ser modeladas como uma cadeia de Markov; (ii) os atrasos nas transições de estados do provedor de serviços podem ser modelados como

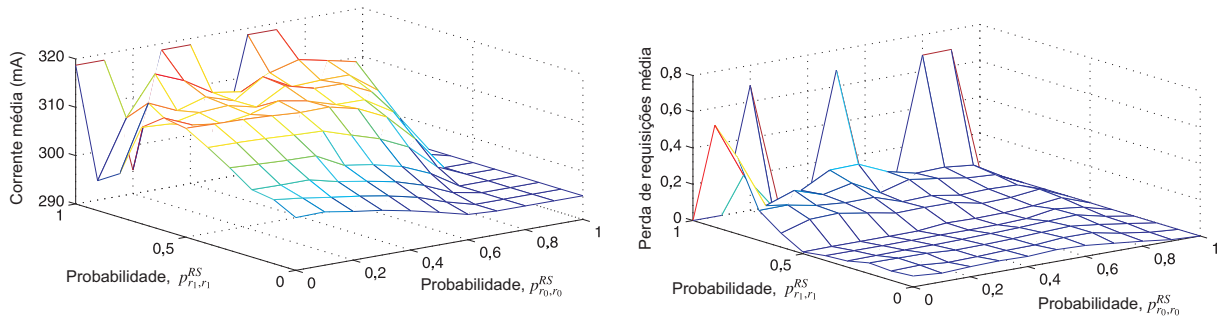


Figura 2.12: Consumo de energia e perda de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (RS).

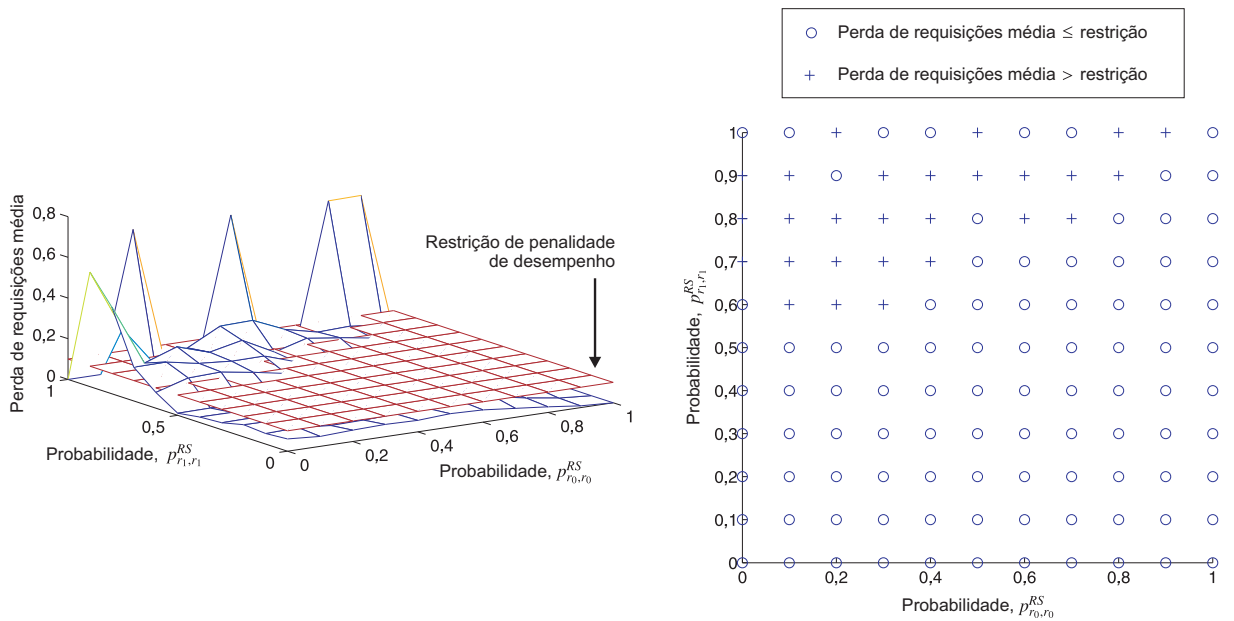


Figura 2.13: Políticas que (não) satisfazem a restrição de desempenho.

variáveis aleatórias com distribuição geométrica; e (iii) os parâmetros do modelo e as funções de custo são disponíveis antes da resolução do problema de otimização. No projeto da política de gerenciamento de energia, é necessário tolerar a inexatidão do modelo para que seja possível obter a solução do problema de otimização, tendo como consequência a redução da qualidade dessa solução. No entanto, é possível aplicar técnicas para adequar as aproximações do modelo ao nível desejado de exatidão para o problema de otimização de política. Por exemplo, quando as probabilidades de transição dos estados do requisitante de serviços são variantes no tempo, ou seja, quando a carga de trabalho é não-estacionária, devem ser aplicadas técnicas de estimação de cargas de trabalho não-estacionárias, que são apresentadas a seguir.

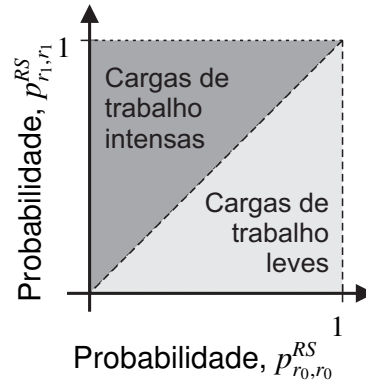


Figura 2.14: Classificação das cargas de trabalho do sistema.

2.3 Políticas para cargas de trabalho não-estacionárias

Num ambiente não-estacionário, admite-se que o RS tem suas probabilidades de transição de estados dependentes do tempo, tal como ilustrado na Fig. 2.15 para um RS de dois estados. Denota-se uma carga de trabalho não-estacionária por U^l e ela é modelada como uma série de N_u cargas de trabalho estacionárias u_s , $s = 0, 1, \dots, N_u - 1$ que possuem diferentes probabilidades de requisição de usuário. Uma carga de trabalho não-estacionária é então representada por $U^{N_u} = (u_0, u_1, \dots, u_{N_u-1})$. Sendo R_i a probabilidade de um RS permanecer no estado r_i , i.e. $R_i = p_{r_i, r_i}^{RS}$, percebe-se que R_i para o caso não-estacionário é uma função da seqüência de cargas estacionárias dada. É ilustrado na Fig. 2.16 um exemplo de uma carga não-estacionária ao longo do tempo. Denomina-se *política adaptativa ótima* a técnica que utiliza uma tabela de decisão otimizada para o R_i de cada carga de trabalho u_s [1] (a obtenção da tabela de decisão é ilustrada no Exemplo 1). Essa é a técnica ótima para o caso não-estacionário, mas ela é ideal e não pode ser implementada na prática, pois requer o conhecimento perfeito da carga de trabalho não-estacionária. Há portanto a necessidade da estimação das probabilidades de transição dos estados do RS em cargas de trabalho não-estacionárias.

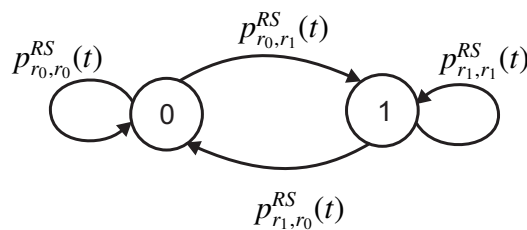


Figura 2.15: Exemplo de um RS não-estacionário.

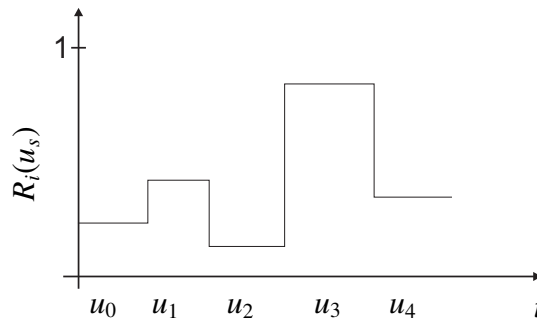


Figura 2.16: Valor de R_i ao longo do tempo para uma carga não-estacionária.

2.3.1 Método de janelas deslizantes de múltiplos comprimentos

É apresentada nesta seção uma técnica de estimação de cargas de trabalho não-estacionárias baseada em janelas deslizantes de múltiplos comprimentos [18]. A apresentação inicial é realizada por meio de estimativas de máxima verossimilhança, mas depois é generalizada para qualquer técnica de estimação, sendo apresentado um exemplo em que são comparados os resultados de consumo de energia e desempenho ao utilizar-se a técnica proposta com estimativas de máxima verossimilhança ou estimação fraca baseada em aprendizado estocástico [17].

Para se obterem resultados próximos da política adaptativa ótima, é apresentado nessa seção o sistema com gerenciamento de energia para o ambiente não-estacionário ilustrado na Fig. 2.17. O gerenciador de energia precisa estimar a carga de trabalho do sistema e escolher uma política ótima para a estimativa obtida. Para isso são utilizadas múltiplas *janelas de requisições* do RS de comprimentos diferentes para gerar estimativas da carga de trabalho, as quais são armazenadas em *janelas de estimativas* para cálculo de sua variância. Assim as estimativas da janela que apresentar maior comprimento e variância abaixo de um determinado limite aceitável são obtidas por um *seletor de janela*. Uma *tabela de políticas* contém tabelas de decisão pré-calculadas para valores selecionados de R_i . Assim num *interpolador de políticas* são observados os estados do RS, FS e PS e utilizadas as tabelas de decisão otimizadas para os valores R_i mais próximos das estimativas selecionadas para obter assim uma decisão, com base na qual será realizada a *escolha do comando* a ser passado para o PS.

2.3.1.1 Janelas de requisições

Nessa seção é mostrado como o bloco *janelas de requisições* da Fig. 2.17 armazena as requisições do RS e obtém as estimativas de probabilidade de transição do modelo do RS. A abordagem de identificação das probabilidades de transição do RS com multi-janela permite manter a história prévia de cada estado separadamente. Um exemplo é mostrado na Fig. 2.18 para um RS de dois estados. O número de janelas é igual ao número de estados do RS. Cada

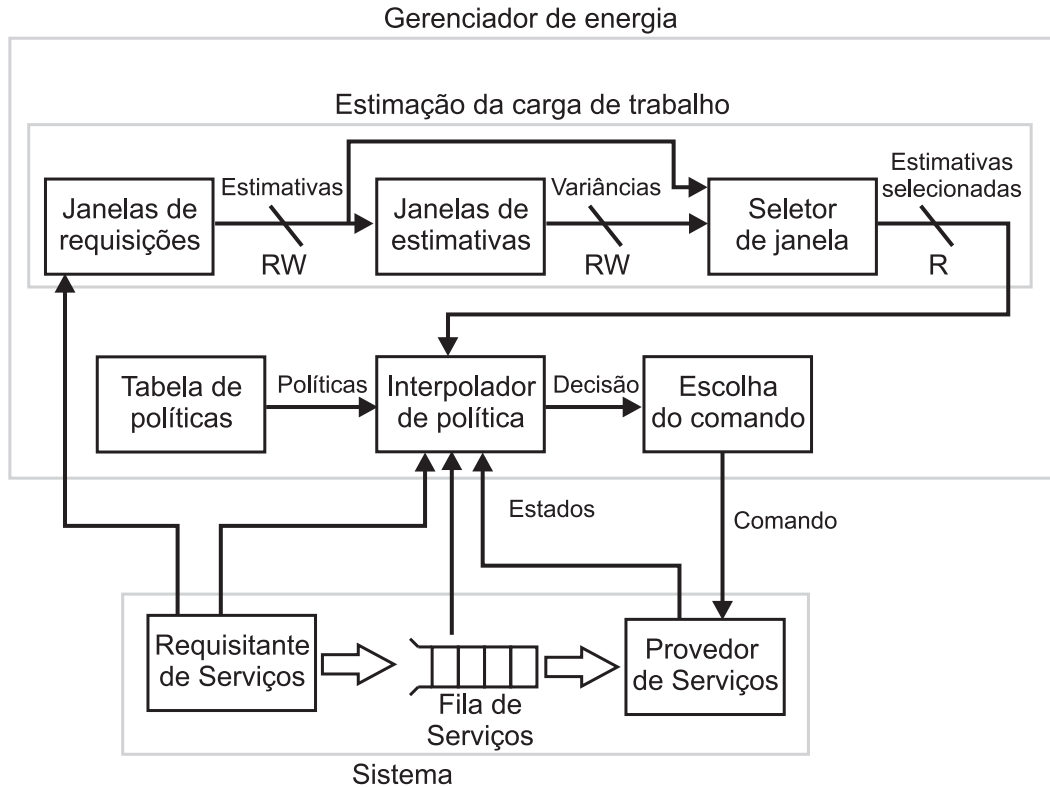


Figura 2.17: Componentes do modelo do sistema baseado em processos de Markov no tempo discreto para ambiente não-estacionário.

janela W_i corresponde a um estado $s_r = r_i$ e todas elas possuem comprimento l_w . As transições anteriores a partir do estado i são armazenadas na janela W_i . A notação $W_i^k(t)$ significa o conteúdo da posição k da janela W_i no instante de tempo t . O *Buffer da Requisição Anterior* (BRA) armazena $s_r(t - 1)$ e controla o seletor de janela, que seleciona a janela W_i tal que $s_r(t - 1) = r_i$. Apenas essa janela selecionada desloca o seu conteúdo de uma posição para a direita, ou seja, $W_i^{k+1}(t) = W_i^k(t), \forall k = 0, \dots, l_w - 2$, e armazena $s_r(t)$ na posição $W_i^0(t)$.

Seja “==” a operação de equivalência, i.e., o resultado é “1” quando os argumentos são iguais, e “0” caso contrário. Seja X a variável aleatória igual ao resultado da operação $(W_i^k(t) == r_j)$, na qual é verificado se o estado armazenado em $W_i^k(t)$ é r_j , indicando a ocorrência de uma transição do estado r_i para o estado r_j . Assim, X é uma variável aleatória discreta de espaço amostral $\{0, 1\}$ com probabilidades $P(X = 1) = p_{r_i, r_j}^{RS}(t)$ e $P(X = 0) = 1 - p_{r_i, r_j}^{RS}(t)$. As expressões da esperança $E(X)$ e a variância $V(X)$ são apresentadas em (2.24) e (2.25) respectivamente, e o gráfico da variância em função da probabilidade $p_{r_i, r_j}^{RS}(t)$ é mostrado na Fig. 2.19.

$$E(X) = \sum_{i=1}^2 x_i P(X = x_i) = 0 \cdot P(X = 0) + 1 \cdot P(X = 1) = p_{r_i, r_j}^{RS}(t) \quad (2.24)$$

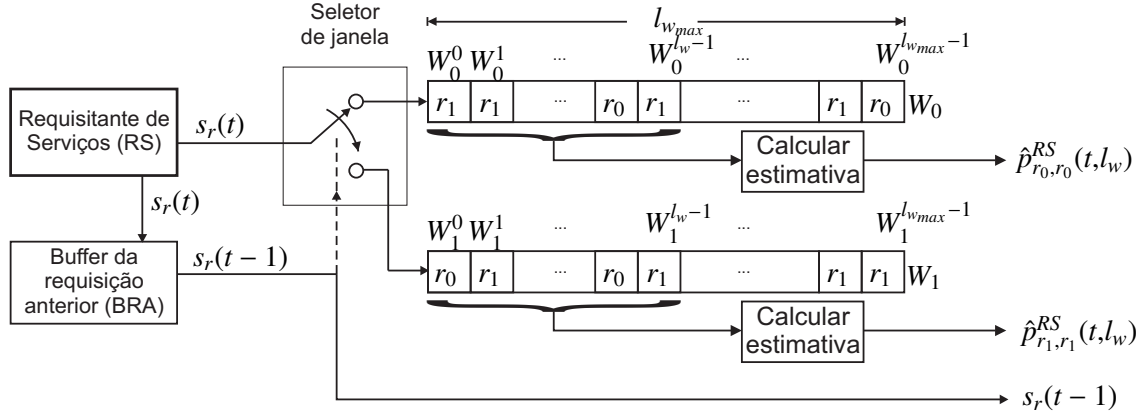
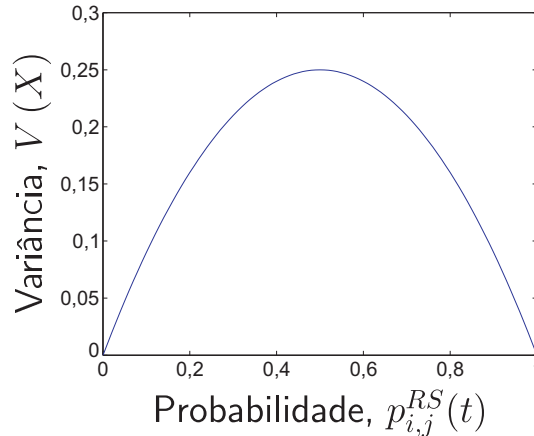


Figura 2.18: Exemplo de uso de multi-janela para RS de dois estados.

$$\begin{aligned}
 V(X) &= E(X - E(X))^2 = E(X - p_{r_i, r_j}^{RS}(t))^2 = \\
 &= (1 - p_{r_i, r_j}^{RS}(t))^2 \cdot p_{r_i, r_j}^{RS}(t) + (0 - p_{r_i, r_j}^{RS}(t))^2 \cdot (1 - p_{r_i, r_j}^{RS}(t)) = p_{r_i, r_j}^{RS}(t) - [p_{r_i, r_j}^{RS}(t)]^2
 \end{aligned} \quad (2.25)$$

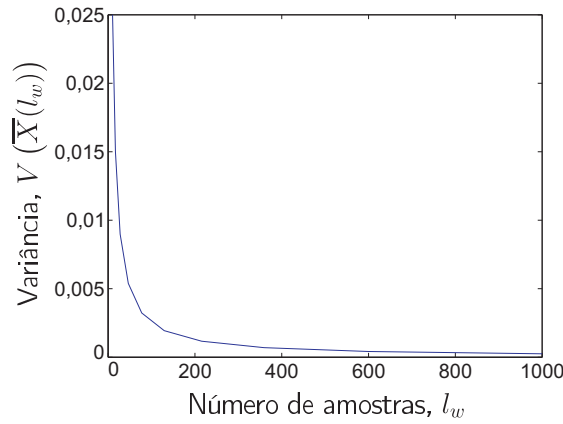

 Figura 2.19: Variância de X .

É possível tomar l_w amostras de X , sendo cada uma de valor $X_k = (W_i^k(t) == r_j)$. A variável aleatória $Y(l_w) = \sum_{k=0}^{l_w-1} X_k$ tem distribuição binomial. A partir da amostra (X_0, \dots, X_{l_w-1}) é calculada a *média amostral* $\bar{X}(l_w)$ em (2.26), a qual também é uma variável aleatória. A esperança $E(\bar{X}(l_w))$ e a variância $V(\bar{X}(l_w))$ são calculadas em (2.27) e (2.28) respectivamente. Fixando $p_{r_i, r_j}^{RS}(t) = 0,5$, é mostrada na Fig. 2.20 a variância $V(\bar{X}(l_w))$ em função do número l_w de amostras. Como $E(\bar{X}(l_w))$ é igual ao parâmetro verdadeiro a ser estimado $p_{r_i, r_j}^{RS}(t)$, então sua estimativa $\hat{p}_{r_i, r_j}^{RS}(t, l_w) = \bar{X}(l_w)$, $\forall i, j$ é não-tendenciosa. Como a variância $V(\bar{X}(l_w))$ tende a zero com o aumento de l_w , como mostrado em (2.29), a estimativa $\hat{p}_{r_i, r_j}^{RS}(t, l_w)$ é coerente [39]. A expressão do cálculo da estimativa por meio da média amostral de X em (2.26) pode ser reescrita substituindo-se X_k por $(W_i^k(t) == r_j)$, o que é apresentado em (2.30).

$$\bar{X}(l_w) = \frac{1}{l_w} \sum_{k=0}^{l_w-1} X_k = \frac{Y(l_w)}{l_w} \quad (2.26)$$

$$E(\bar{X}(l_w)) = E\left(\frac{Y(l_w)}{l_w}\right) = \frac{1}{l_w} l_w p_{r_i, r_j}^{RS}(t) = p_{r_i, r_j}^{RS}(t) \quad (2.27)$$

$$V(\bar{X}(l_w)) = V\left(\frac{Y(l_w)}{l_w}\right) = \frac{1}{l_w^2} l_w p_{r_i, r_j}^{RS}(t) (1 - p_{r_i, r_j}^{RS}(t)) = \frac{p_{r_i, r_j}^{RS}(t) (1 - p_{r_i, r_j}^{RS}(t))}{l_w} \quad (2.28)$$


 Figura 2.20: Variância da estimativa de $p_{r_i, r_j}^{RS}(t) = 0,5$.

$$\lim_{l_w \rightarrow \infty} V(\bar{X}(l_w)) = \lim_{l_w \rightarrow \infty} \frac{p_{r_i, r_j}^{RS}(t) (1 - p_{r_i, r_j}^{RS}(t))}{l_w} = 0 \quad (2.29)$$

$$\hat{p}_{r_i, r_j}^{RS}(t, l_w) = \frac{\sum_{k=0}^{l_w-1} (W_i^k(t) == r_j)}{l_w} \text{ para todo } i, j. \quad (2.30)$$

Para comparar-se a qualidade de identificação por um dado método é útil a utilização de um indicador quantitativo, como por exemplo, o erro médio quadrático (EMQ), cuja definição é oferecida a seguir.

Definição 4 Seja $e_{ij}(t) = \hat{p}_{r_i, r_j}^{RS}(t) - p_{r_i, r_j}^{RS}(t)$ o erro de estimação da probabilidade $p_{r_i, r_j}^{RS}(t)$. Dado que um RS é identificado ao longo de uma simulação ou execução durante os passos de $k = 1 \dots m$, o erro médio quadrático EMQ_{ij} da estimativa $\hat{p}_{r_i, r_j}^{RS}(t)$ em relação a $p_{r_i, r_j}^{RS}(t)$ é calculado conforme (2.31), e essa é uma grandeza é adimensional.

$$EMQ_{ij} = \frac{1}{m} \sum_{k=1}^m (e_{ij}(t_k))^2 = \frac{1}{m} \sum_{k=1}^m [\hat{p}_{r_i, r_j}^{RS}(t_k) - p_{r_i, r_j}^{RS}(t_k)]^2 \quad (2.31)$$

No exemplo a seguir, o desempenho da estimação do RS é comparado para diferentes comprimentos de janela. São verificadas a qualidade da estimativa e o atraso para a estimativa se tornar razoavelmente próxima do valor real.

Exemplo 3 *Seja o RS com matriz de transição explicitada em (2.32). Foi utilizado o método de identificação do RS com a abordagem de multi-janela por meio de três comprimentos de janela: 100, 200 e 400. Os resultados são mostrados nas Figuras 2.21, 2.22 e 2.23, respectivamente. Foram utilizados o período de amostragem de 1s e o tempo de simulação de 4000s. Em cada figura, a linha tracejada representa a probabilidade real, e a linha sólida é a identificação da probabilidade.*

$$P^{RS} = \begin{matrix} & 0 & 1 \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} 0,1 & 0,9 \\ 0,6 & 0,4 \end{bmatrix} \end{matrix} \quad (2.32)$$

Um comparativo quantitativo para o desempenho da estimação do RS para os diferentes tamanhos de janela pode ser realizado por meio do cálculo do EMQ para as probabilidades $p_{0,0}^{RS}(t)$ e $p_{1,1}^{RS}(t)$ entre os passos de tempo 1000 e 4000, conforme apresentado na Tabela 2.1. Dentre as estimativas desse exemplo, 1000 passos é o pior caso para o tempo necessário para a estimativa se tornar razoavelmente próxima do seu valor teórico, o que ocorre na identificação de $p_{0,0}^{RS}$ na Fig. 2.23. Percebe-se que, quanto maior o comprimento da janela, menores os erros médios quadráticos.

Tabela 2.1: Erro médio quadrático da identificação do RS para a abordagem com método de multi-janela

Tamanho da janela	EMQ_{00}	EMQ_{11}
100	$7,6650 \cdot 10^{-4}$	0,0022
200	$2,7289 \cdot 10^{-4}$	0,0011
400	$6,8629 \cdot 10^{-5}$	$2,3352 \cdot 10^{-4}$

Percebe-se pelo exemplo 3 que, aumentado-se o número de amostras l_w para identificação de $p_{r_i,r_j}^{RS}(t)$:

- melhora-se a qualidade das amostras, pois se reduz a variância em (2.29);
- mas o tempo inicial para obter-se as amostras aumenta, acarretando num atraso inicial tanto maior quanto maior o número de amostras.

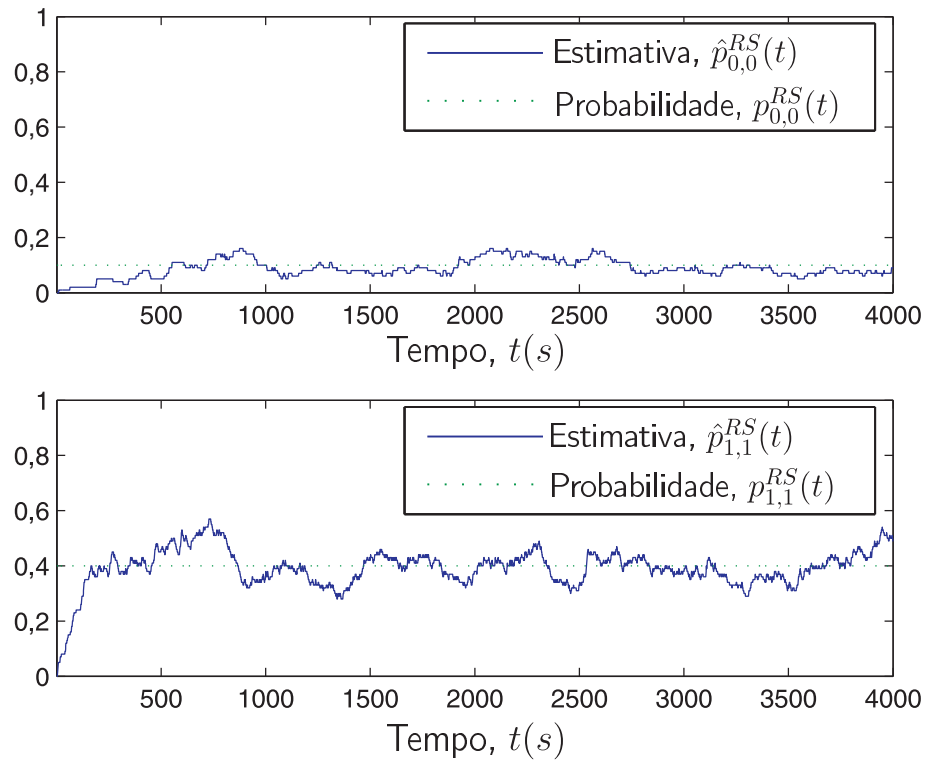


Figura 2.21: Identificação de um RS por meio do método de multi-janela com comprimento 100.

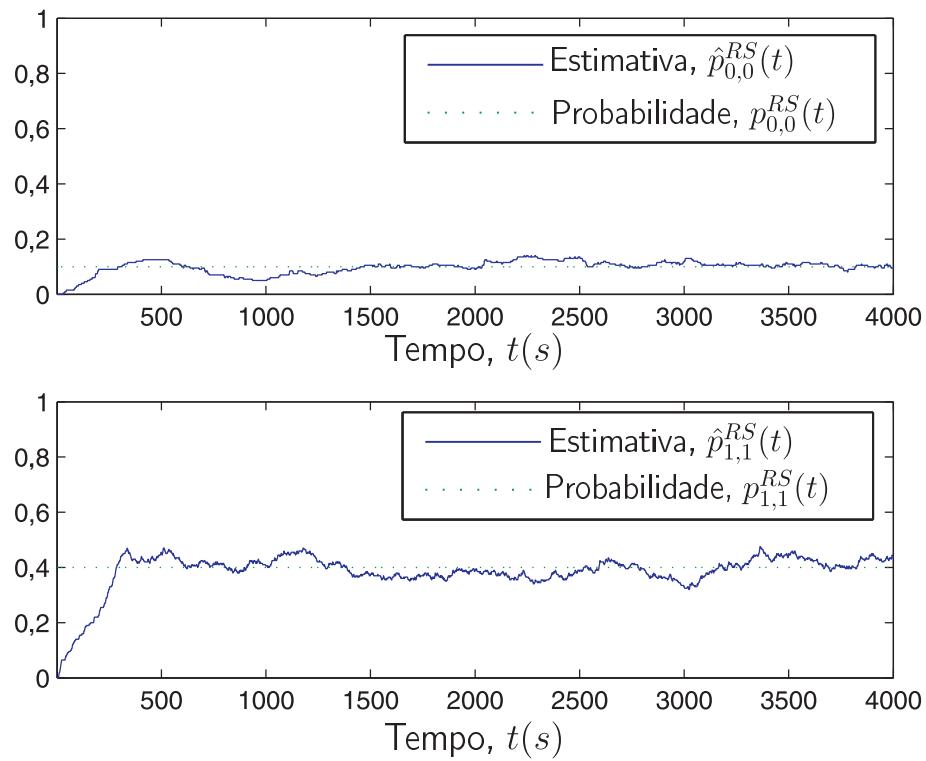


Figura 2.22: Identificação de um RS por meio do método de multi-janela com comprimento 200.

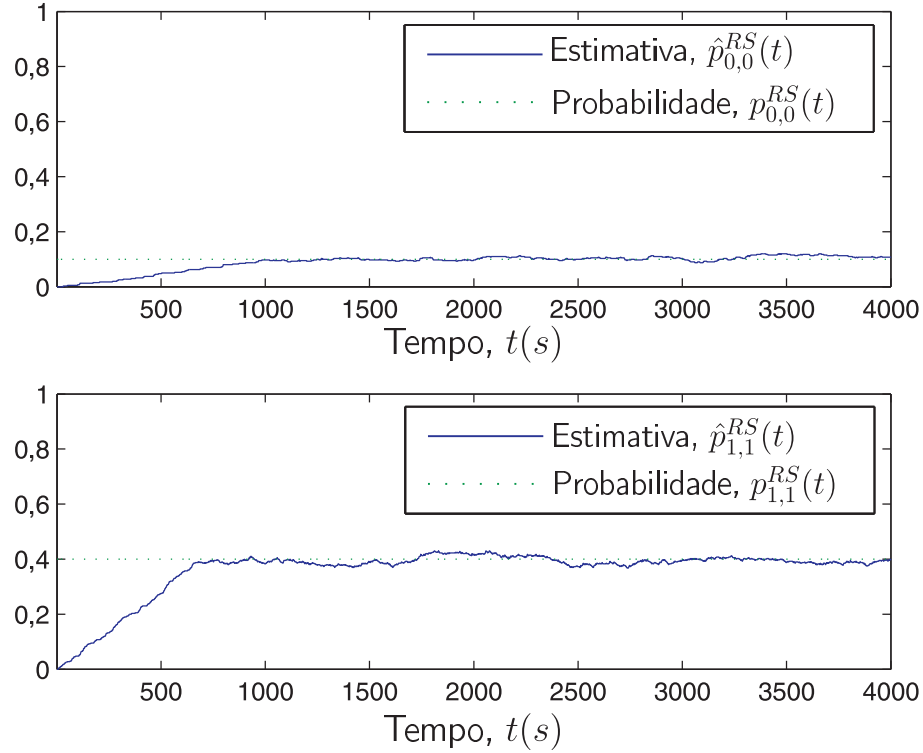


Figura 2.23: Identificação de um RS por meio do método de multi-janela com comprimento 400.

Nas próximas duas seções esses dois aspectos são caracterizados formalmente, para então na Seção 2.3.1.5 ser apresentada uma técnica para obter-se proveito tanto do menor tempo de preenchimento de janelas menores, quanto do baixo erro de amostragem de janelas maiores.

2.3.1.2 Erro de amostragem

Após o atraso de identificação de uma carga estacionária u_r para outra u_s , como mostrado na Fig. 2.24, as estimativas oscilam em torno dos valores de probabilidades de transição verdadeiros $p_{r_i,r_j}^{RS}(t)$. Esse efeito é denominado o *erro de amostragem* (nomeado *resolution error* em [1]). Diminuir o comprimento da janela é benéfico para reduzir o atraso de identificação, mas isso aumenta o erro de amostragem.

Seja ε um número positivo, e n o número de amostras. Pela *lei dos grandes números* (formulação de Bernoulli) [39],

$$P \left[|\hat{p}_{r_i,r_j}^{RS}(t) - p_{r_i,r_j}^{RS}(t)| < \varepsilon \right] \geq 1 - \frac{p_{r_i,r_j}^{RS}(t) (1 - p_{r_i,r_j}^{RS}(t))}{n\varepsilon^2}. \quad (2.33)$$

O valor ε será denominado *erro de amostragem*. Assim, para obter

$$P \left[|\hat{p}_{r_i,r_j}^{RS}(t) - p_{r_i,r_j}^{RS}(t)| < \varepsilon \right] \geq 1 - \delta,$$

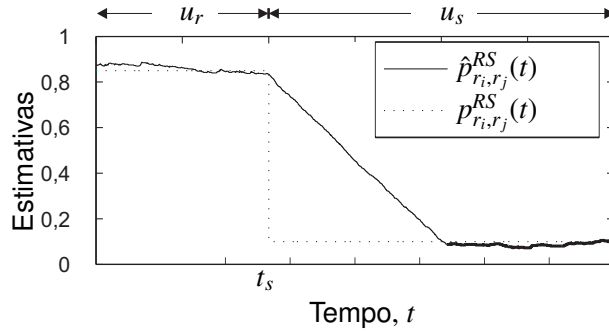


Figura 2.24: O erro de amostragem (linha de maior espessura) após a transição de uma carga de trabalho estacionária u_r para outra u_s .

em que $0 < \delta < 1$, é necessário utilizar um número de amostras n que satisfaça

$$n \geq \frac{p_{r_i, r_j}^{RS}(t)(1 - p_{r_i, r_j}^{RS}(t))}{\delta \varepsilon^2}. \quad (2.34)$$

Como o parâmetro $p_{r_i, r_j}^{RS}(t)$ é desconhecido, é preciso aplicar

$$\max_{p_{r_i, r_j}^{RS}(t)} p_{r_i, r_j}^{RS}(t)(1 - p_{r_i, r_j}^{RS}(t)) = 1/4,$$

para $p_{r_i, r_j}^{RS}(t) = 1/2$. Logo, para obter $P[|\hat{p}_{r_i, r_j}^{RS}(t) - p_{r_i, r_j}^{RS}(t)| < \varepsilon] \geq 1 - \delta$, é necessário utilizar

$$n \geq \frac{1}{4\delta\varepsilon^2}. \quad (2.35)$$

Assim é possível diminuir o erro de amostragem tanto quanto se deseje aumentando o número de amostras.

2.3.1.3 Tempo de preenchimento da janela

Como mostrado na Fig. 2.25, quando uma transição de uma carga de trabalho estacionária u_r para outra u_s ocorre em $t = t_s$, existe um período de tempo T_{p_i} necessário para que as estimativas $\hat{p}_{r_i, r_j}^{RS}(t)$ convirjam para a nova carga de trabalho u_s . Esse período de tempo é denominado o *atraso de identificação* (nomeado *adaptation time* em [1]). Para o estado r_i do requisitante de serviços, o atraso de identificação é denotado T_{p_i} . O valor máximo dos atrasos de identificação para todos os estados do requisitante de serviços r_i , $i = 0, \dots, R - 1$, é denotado T_p .

Considerando a utilização do método de multi-janela para estimar as probabilidades de transição de um RS que está transitando de um regime estacionário u_r para outro regime u_s , qual o número de passos médio T_{p_j} para preenchimento da janela j de comprimento l_w com

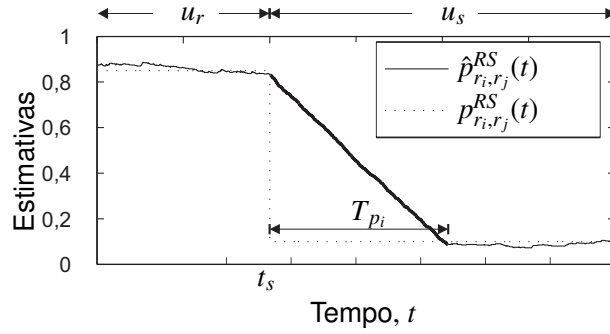


Figura 2.25: O atraso de identificação (linha de maior espessura) durante a transição de uma carga de trabalho estacionária u_r para outra u_s .

transições apenas do novo regime u_s ? Um estimador de máxima verossimilhança para o limite $\pi_j(t) \triangleq \lim_{n \rightarrow \infty} P[s_r(t) = r_j]$ é

$$\hat{\pi}_j = \frac{N_j}{N}$$

em que N é o número total de amostras e N_j é o número de amostras em que o estado da cadeia de Markov é r_j . Para o caso do método de multi-janela, N_j corresponde ao comprimento da janela l_w pois esta só recebe um novo elemento quando, no passo imediatamente anterior, a cadeia de Markov estava no estado r_j . E N corresponde ao número de passos médio T_{p_j} para preenchimento da janela j . Assim uma estimativa para T_{p_j} é

$$\hat{\pi}_j = \frac{l_w}{\hat{T}_{p_j}} \Rightarrow \hat{T}_{p_j} = \frac{l_w}{\hat{\pi}_j}.$$

É de interesse o valor de T_{p_j} correspondente ao valor real π_j . Assim define-se:

$$T_{p_j} \triangleq \frac{l_w}{\pi_j}. \quad (2.36)$$

E o tempo de preenchimento de todas as janelas T_p é o máximo dos tempos de preenchimento T_{p_j} de cada janela:

$$T_p \triangleq \max_j \{T_{p_j}\}. \quad (2.37)$$

2.3.1.4 Janelas de estimativas

Nessa seção é mostrado como o bloco *janelas de estimativas* da Fig. 2.26 permite estimar as variâncias das estimativas obtidas por meio das janelas de diferentes comprimentos. A variância

$V(\hat{p}_{r_i, r_j}^{RS}(t))$ é um valor esperado, contudo ela precisa ser estimada das amostras de uma janela de comprimento l_w . As estimativas advindas do bloco *janelas de requisições* são armazenadas em *janelas de estimativas*. É calculada então a média das estimativas por meio de (2.38). E essa média é utilizada para o cálculo da variância por meio de (2.39).

$$\bar{p}_{r_i, r_j}^{RS}(t, l_w) = \frac{1}{l_w} \sum_{k=0}^{l_w-1} (\hat{p}_{r_i, r_j}^{RS}(t - t_k, l_w)) \quad (2.38)$$

$$S(r_i, r_j, t, l_w) = \frac{1}{l_w - 1} \sum_{k=0}^{l_w-1} (\hat{p}_{r_i, r_j}^{RS}(t - t_k, l_w) - \bar{p}_{r_i, r_j}^{RS}(t, l_w))^2 \quad (2.39)$$

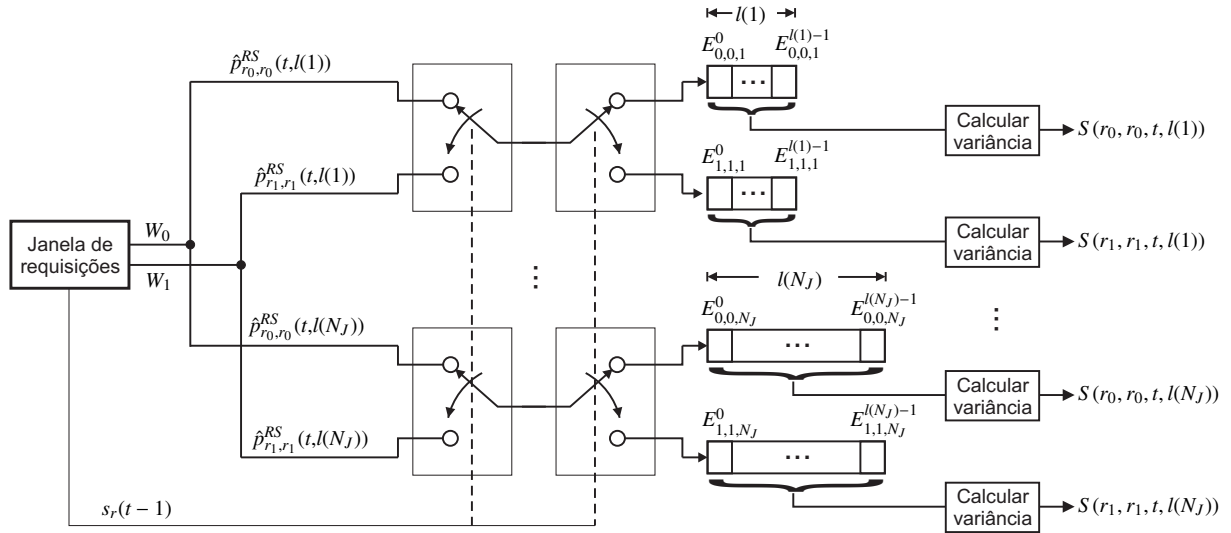


Figura 2.26: Exemplo do bloco *janelas de estimativas* como mostrado na Fig. 2.17 para um requisitante de serviços com dois estados.

2.3.1.5 Seletor de janela

Nessa seção é mostrado como o bloco *seletor de janela* da Fig. 2.17 permite, dentre as estimativas das janelas de diferentes comprimentos, escolher as que melhor se aproximam das probabilidades de transição do modelo do RS segundo um determinado critério. Na estimação das probabilidades de transição do RS, deseja-se um erro de estimação o menor possível, o que pode ser obtido com um número de amostras suficientemente grande, como mostrado na Seção 2.3.1.2. No entanto no início do processo de estimação, não há tantas amostras quanto se deseja. Como mostrado na Seção 2.3.1.3, quanto maior o comprimento da janela, maior é o seu tempo de preenchimento. E assim corre-se o risco de que, no início do processo de gerenciamento de energia ou no momento de transição de uma carga de trabalho estacionária para

outra, sejam utilizadas estimativas muito distantes das probabilidades de transição reais do RS. Assim, é necessário que, em tais situações, utilizem-se comprimentos de janela pequenos, capazes de identificar rapidamente as probabilidades de transição do RS, apesar de possuírem um erro de amostragem maior. E posteriormente é possível utilizar gradualmente comprimentos de janela maiores para diminuir o erro de amostragem. Nessa seção é introduzida uma estratégia de escolha do comprimento da janela baseada na variância das estimativas.

A variância $V(X)$ (expressão (2.25)) assume valor máximo $1/4$ para $p_{r_i, r_j}^{RS}(t) = 1/2$. Assim a variância da estimativa $V(\hat{p}_{r_i, r_j}^{RS}(t))$ assume valor máximo $\frac{1}{4l_w}$ também para $p_{r_i, r_j}^{RS}(t) = 1/2$. A probabilidade $p_{r_i, r_j}^{RS}(t)$ a ser estimada é desconhecida. Então, utilizar uma janela l_w para obter uma estimativa $\hat{p}_{r_i, r_j}^{RS}(t)$ com variância $V(\hat{p}_{r_i, r_j}^{RS}(t))$ menor ou igual a $\frac{1}{4l_w}$ pode ser insatisfatório para os casos em que a probabilidade $p_{r_i, r_j}^{RS}(t)$ é diferente de $1/2$ pois a variância $V(\hat{p}_{r_i, r_j}^{RS}(t))$ é menor para esses casos, como mostrado na Fig. 2.19. Para $p_{r_i, r_j}^{RS}(t) = 0$ ou 1 por exemplo, a variância $V(\hat{p}_{r_i, r_j}^{RS}(t))$ é nula. Então, é possível escolher

$$Vac(\hat{p}_{r_i, r_j}^{RS}(t), l_w) = \frac{\hat{p}_{r_i, r_j}^{RS}(t)(1 - \hat{p}_{r_i, r_j}^{RS}(t))}{l_w}$$

para a variância aceitável $Vac(\hat{p}_{r_i, r_j}^{RS}(t), l_w)$ em função do comprimento l_w da janela.

É possível utilizar ao mesmo tempo janelas de comprimentos diferentes e escolher a estimativa da maior janela que apresenta variâncias menores que $Vac(\hat{p}_{r_i, r_j}^{RS}(t), l_w)$. Sejam J , D e N_J números inteiros constantes, e $l = (J, J + D, J + 2D, \dots, J + (N_J - 1)D)$ uma sequência de comprimentos de janela. É utilizado então o Algoritmo 1 para a escolha do comprimento da janela no método de janelas deslizantes de múltiplos comprimentos.

Algoritmo 1: Algoritmo do seletor de janela

Entrada: $l = (J, J + D, J + 2D, \dots, J + (N_J - 1)D)$, N_J e $S(r_i, r_i, t, l(k))$, $i = 1 \dots R$

Saída: Janela escolhida j_e

$j_e = l(1)$;

$k = 2$;

repita

se $\bigcap_{i=1}^R \{S(r_i, r_i, t, l(k)) \leq Vac(\hat{p}_{r_i, r_i}^{RS}(t), l(k))\}$ **então**
 $j_e = l(k)$;

fim

$k = k + 1$;

até $k \leq N_J$;

No início da simulação/execução do método de janelas deslizantes de múltiplos comprimentos, é necessário que os valores de variância $S(r_i, r_i, t, l(k))$ sejam calculados apenas para $t \geq l(k)$ para que haja valores de estimativas suficientes. E os valores de $S(r_i, r_i, t, l(k))$ para

$t < l(k)$ devem ser inicializados para um valor alto, e.g., 1. Caso contrário, os valores de $S(r_i, r_i, t, l(k))$ para $t < l(k)$ serão bem menores que $Vac(\hat{p}_{r_i, r_i}^{RS}(t), l(k))$ e assim seria selecionada uma janela de comprimento grande a partir do início, quando se deve utilizar a janela de menor comprimento para que sejam obtidas estimativas próximas dos valores reais o mais rápido possível.

Exemplo 4 Seja o RS não-estacionário especificado em (2.40) e (2.41). Para identificar o modelo do RS, foi utilizado o método de janelas deslizantes de múltiplos comprimentos (JMC) e o de multi-janela (EMV) [1] com janelas de comprimento 50, 250 e 450. As estimativas do RS por meio dos dois métodos são mostradas nas Figuras 2.27 e 2.28. A escolha dos comprimentos de janela no método de janelas deslizantes de múltiplos comprimentos foi baseada no cálculo da variância das estimativas para cada comprimento de janela, como mostrado na Fig. 2.29.

$$p_{r_0, r_0}^{RS}(t) = \begin{cases} 0,2, & \text{se } 0 \leq t \leq 5000s \\ 0,8, & \text{se } 5000s < t \leq 10000s \end{cases} \quad (2.40)$$

$$p_{r_1, r_1}^{RS}(t) = \begin{cases} 0,1, & \text{se } 0 \leq t \leq 5000s \\ 0,4, & \text{se } 5000s < t \leq 10000s \end{cases} \quad (2.41)$$

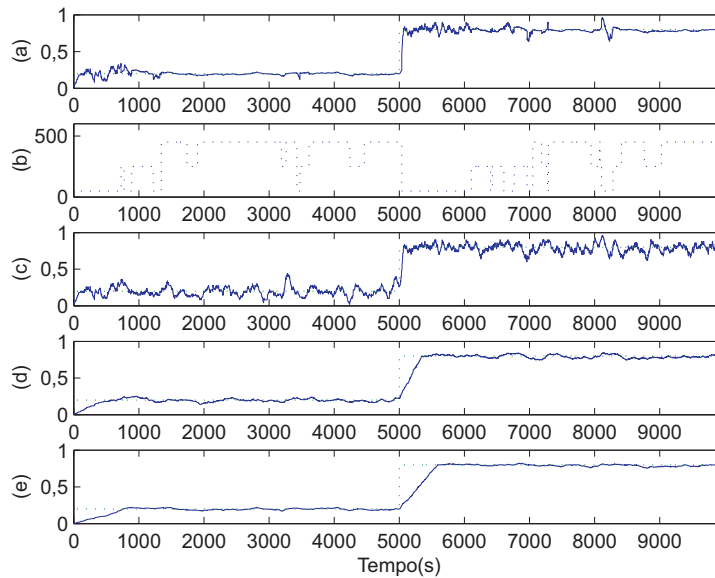


Figura 2.27: Método de janelas deslizantes de múltiplos comprimentos: (a) Estimativas de $p_{r_0, r_0}^{RS}(t)$; (b) comprimentos de janela utilizados. Estimativas de $p_{r_0, r_0}^{RS}(t)$ por meio do método de multi-janela [1] com janelas de comprimento: (c) 50; (d) 250; (e) 450.

São apresentados na Tabela 2.2 os erros médios quadráticos (EMQ) ao longo de toda a simulação para o método de janelas deslizantes de múltiplos comprimentos e o de multi-janela [1]. Verifica-se que, quanto maior o comprimento da janela, maior o erro médio quadrático para o método de multi-janela [1] pois, apesar do menor erro de amostragem quando

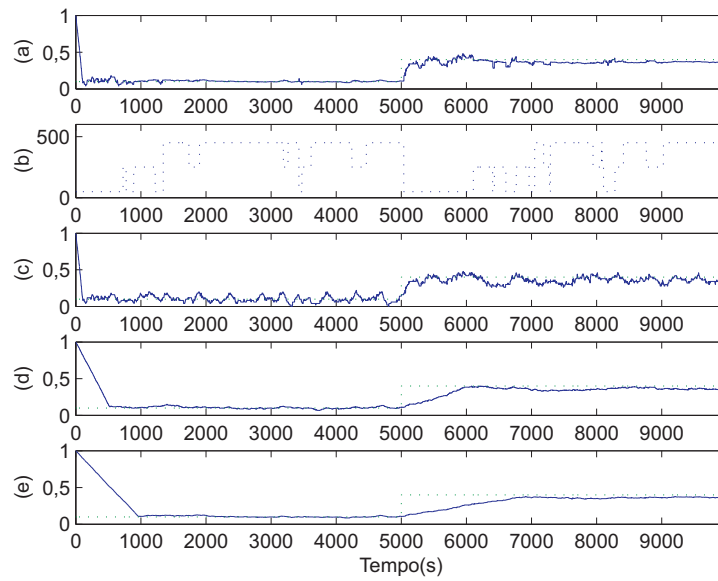


Figura 2.28: Método de janelas deslizantes de múltiplos comprimentos: (a) Estimativas de $p_{r_1,r_1}^{RS}(t)$; (b) comprimentos de janela escolhidos. Estimativas de $p_{r_1,r_1}^{RS}(t)$ por meio do método de multi-janela [1] com janelas de comprimento: (c) 50; (d) 250; (e) 450.

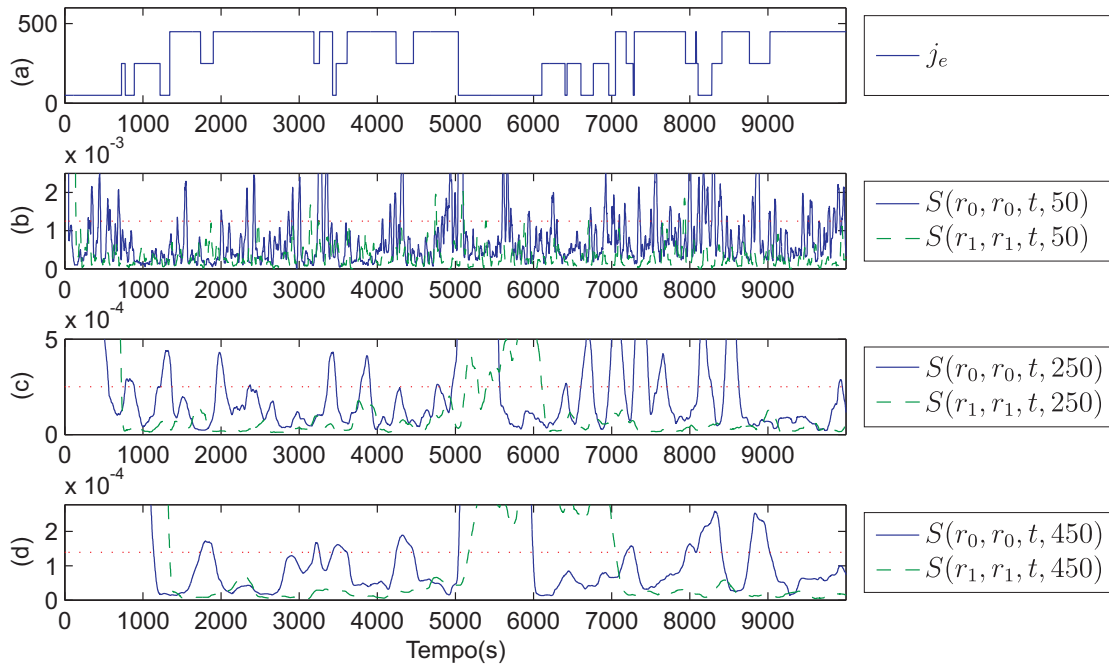


Figura 2.29: (a) Comprimentos de janela escolhidos. Variâncias das estimativas com janelas de comprimento: (b) 50; (c) 250; (d) 450.

as janelas já estão preenchidas, durante o tempo de preenchimento as estimativas ficam muito distantes da probabilidade verdadeira. Como observado nas Figuras 2.27 e 2.28, o método de janelas deslizantes de múltiplos comprimentos utiliza comprimentos de janela menores no início da simulação ou quando há a transição de uma carga estacionária para outra. Com a diminuição da variância das janelas de comprimento maior, atingindo os valores aceitáveis para cada janela, tais janelas são adotadas, beneficiando-se do menor erro de amostragem. Assim os erros médios quadráticos para o método de janelas deslizantes de múltiplos comprimentos são menores que qualquer comprimento de janela pelo método de multi-janela [1].

Tabela 2.2: Comparação dos métodos de identificação do RS.

Método	$EMQ_{00}(\cdot 10^{-3})$	$EMQ_{11}(\cdot 10^{-3})$
JMC	2,5	4,471
EMV [1] ($l_w = 50$)	4,297	6,472
EMV [1] ($l_w = 250$)	5,409	18,631
EMV [1] ($l_w = 450$)	8,833	32,551

Na próxima seção, será apresentado outro método de identificação: *estimação fraca baseada em aprendizado estocástico* [17, 19]. Este método pode ser aplicado no lugar do método de multi-janela [1] no bloco *Calcular estimativa* mostrado na Fig. 2.18 e também ser combinado com o método de janelas deslizantes de múltiplos comprimentos.

2.3.2 Identificação de cargas de trabalho por meio de aprendizado estocástico

É apresentada nesta seção a aplicação da *estimação fraca baseada em aprendizado estocástico* para identificar cargas de trabalho não-estacionárias [17, 19]. Os efeitos de atraso de identificação e erro de amostragem encontrados em estimação fraca baseada em aprendizado estocástico foram formalmente descritos e expressões analíticas foram derivadas. Em comparação a abordagens anteriores de estimação, essa técnica permite melhores resultados em acurácia de estimação, desempenho e consumo de energia. Além disso, a estimação fraca baseada em aprendizado estocástico requer menos esforço computacional do que o método de multi-janela [1].

A técnica de gerenciamento dinâmico de energia (*GDE*) usando estimação fraca baseada em aprendizado estocástico pode ser descrita com a ajuda do diagrama de blocos mostrado na Fig. 2.30. A técnica proposta foi concebida para ambientes não-estacionários como descrito a seguir. No gerenciador de energia (*GE*), é realizada a estimação da carga de trabalho do sistema e a escolha do comando a ser passado ao provedor de serviço. Para isso, é utilizada estimação

fraca baseada em aprendizado estocástico, 1, para obter *estimativas* das probabilidades de transição do requisitante de serviços. Uma *tabela de políticas*, 2, contém tabelas de decisão geradas em tempo de projeto como resultado do problema de otimização de política (*OP*) . Então, o *interpolador de políticas*, 3, introduzido em [1], tem como entradas os estados do requisitante de serviços, 5, fila de serviço, 6, provedor de serviços, 7, e as *estimativas* para a interpolação das tabelas de decisão. O resultado é uma *decisão* que é usada para a *escolha do comando*, 4, a ser passado para o provedor de serviços. No texto seguinte, a estimação fraca baseada em aprendizado estocástico para identificação da carga de trabalho é analisada.

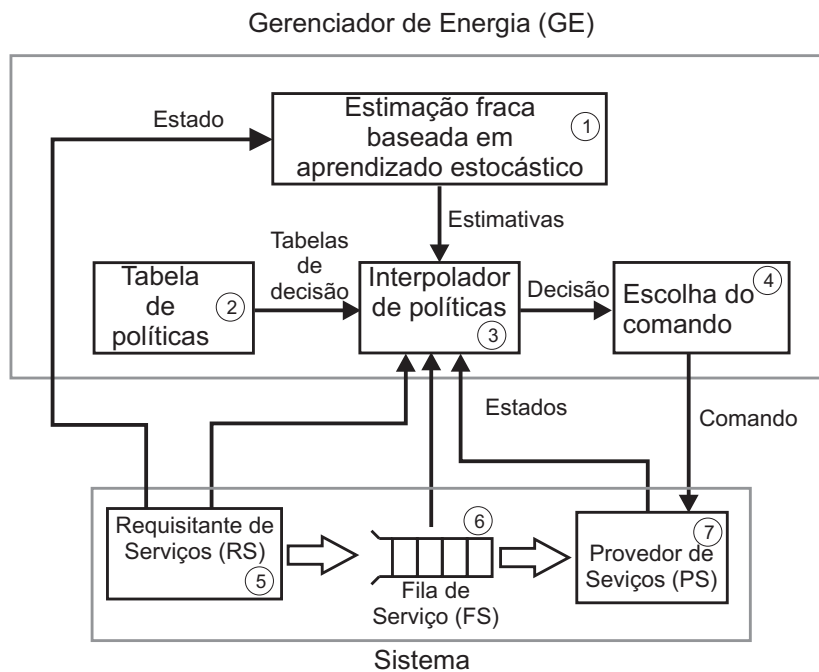


Figura 2.30: Estimação fraca baseada em aprendizado estocástico para ambientes não-estacionários.

No bloco *estimação fraca baseada em aprendizado estocástico* mostrado na Fig. 2.30, as estimativas das probabilidades de transição do requisitante de serviços são calculadas. Seja $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ a estimativa da probabilidade de transição do estado r_i para o estado r_j com o parâmetro $0 < \lambda < 1$. De acordo com Oommen e Rueda [40], tipicamente deve-se escolher valores para o parâmetro λ no intervalo $[0,9, 0,99]$.

Um exemplo é mostrado na Fig. 2.31 para um requisitante de serviços de dois estados, em que apenas $\hat{p}_{r_0, r_0}^{RS}(t, \lambda)$ e $\hat{p}_{r_1, r_1}^{RS}(t, \lambda)$ são obtidas porque as demais podem ser calculadas sempre que necessário: $\hat{p}_{r_0, r_1}^{RS}(t, \lambda) = 1 - \hat{p}_{r_0, r_0}^{RS}(t, \lambda)$ e $\hat{p}_{r_1, r_0}^{RS}(t, \lambda) = 1 - \hat{p}_{r_1, r_1}^{RS}(t, \lambda)$. O valor da estimativa $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ só é atualizado quando a entrada $s_r(t - 1)$ para o bloco *Calcular estimativa* é igual a r_i , i.e., quando o estado do requisitante de serviços $s_r(t - 1) = r_i$, como mostrado na Fig. 2.32. Caso contrário, o bloco *Calcular estimativa* como mostrado na Fig. 2.31 oferece como saída a

estimativa previamente calculada $\hat{p}_{r_i, r_j}^{RS}(t-1, \lambda)$.

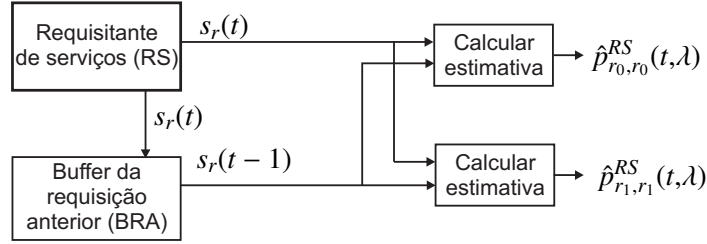


Figura 2.31: Bloco *estimaco fraca baseada em aprendizado estocstico* para um requisitante de servios de dois estados.

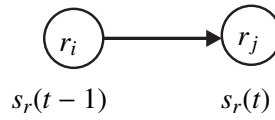


Figura 2.32: Transio do requisitante de servios do estado $s_r(t-1) = r_i$ para o estado $s_r(t) = r_j$.

2.3.2.1 Propriedades estatsticas das estimativas

As estimativas das probabilidades de transio do modelo em cadeia de Markov do requisitante de servios (*RS*) so obtidas no bloco *Calcular estimativa* na Fig. 2.31 por meio de estimaco fraca baseada em aprendizado estocstico [40]. As propriedades estatsticas dessa abordagem de estimaco so analisadas nesta seo. Seja “==” a operao de equivalncia, i.e., o resultado  “1” se os argumentos so iguais, e “0” caso contrrio. E seja \wedge o ‘E’ lgico. Seja $X(t) = (s_r(t-1) == r_i) \wedge (s_r(t) == r_j)$ uma varivel aleatria identificando se o estado do requisitante de servios (*RS*)  r_i no tempo $t-1$, e r_j no tempo t . Quando $X(t) = 1$, houve uma transio do estado r_i para o estado r_j . Ento, X  uma varivel aleatria discreta com espao amostral $\{0,1\}$ com probabilidades $P(X = 1) = p_{r_i, r_j}^{RS}(t)$ e $P(X = 0) = 1 - p_{r_i, r_j}^{RS}(t)$. O valor esperado $E(X)$ e a varincia $V(X)$ so apresentados em (2.42) e (2.43) respectivamente.

$$E(X) = \sum_{i=1}^2 x_i P(X = x_i) = p_{r_i, r_j}^{RS}(t) \quad (2.42)$$

$$V(X) = E(X - E(X))^2 = p_{r_i, r_j}^{RS}(t) - [p_{r_i, r_j}^{RS}(t)]^2 \quad (2.43)$$

A operao realizada no bloco *Calcular estimativa* (Fig. 2.31)  mostrada no Algoritmo 2, cuja verso para distribuies binomiais foi primeiramente introduzida por Oommen e Rueda [40]. Nesse algoritmo, $\hat{P}(X = 1)$ e $\hat{P}(X = 0)$ so variveis internas. Em $t = 0$, elas so inicializadas com valores definidos pelo usurio, e.g. $\hat{P}(X = 1) = 0,5$ e $\hat{P}(X = 0) = 0,5$, e so atualizadas sempre que $(s_r(t-1) == r_i)$.

Algoritmo 2: Calcular estimativa

Dados: $\hat{P}(X = 1)$, $\hat{P}(X = 0)$ e λ

Entrada: $s_r(t)$ e $s_r(t - 1)$

Saída: $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$

se $(s_r(t - 1) == r_i)$ **então**

se $(s_r(t) == r_j)$ então	$\hat{P}(X = 1) = 1 - \lambda \hat{P}(X = 0)$;
senão	
$\hat{P}(X = 1) = \lambda \hat{P}(X = 1)$;	
$\hat{P}(X = 0) = 1 - \hat{P}(X = 1)$;	

$\hat{p}_{r_i, r_j}^{RS}(t, \lambda) = \hat{P}(X = 1)$;

2.3.2.2 Atraso de identificação

Como mostrado na Fig. 2.33, quando uma transição de uma carga de trabalho estacionária u_r para outra u_s ocorre em $t = t_s$, existe um período de tempo T_{p_i} necessário para que as estimativas $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ convirjam para a nova carga de trabalho u_s . Esse período de tempo é denominado o *atraso de identificação* (nomeado *adaptation time* em [1]). Para o estado r_i do requisitante de serviços, o atraso de identificação é denotado T_{p_i} . O valor máximo dos atrasos de identificação para todos os estados do requisitante de serviços r_i , $i = 0, \dots, R - 1$, é denotado T_p .

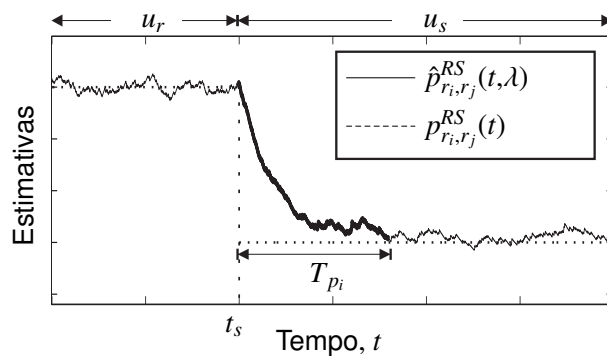


Figura 2.33: O atraso de identificação (linha de maior espessura) durante a transição de uma carga de trabalho estacionária u_r para outra u_s .

Teorema 1 *Se uma transição de uma carga de trabalho u_r para outra u_s ocorre em $t = t_s$, e a carga de trabalho u_s permanece por pelo menos T_{p_i} intervalos de amostragem como mostrado na Fig. 2.33, então, para um dado $0 < \epsilon < 1$, existe um T_{p_i} tal que a inequação em (2.44) é verdadeira, onde p_{r_i, r_j}^{RS} é o valor constante de $p_{r_i, r_j}^{RS}(t)$ durante a carga de trabalho u_s . O período de tempo T_{p_i} é nomeado o atraso de identificação para o estado r_i do requisitante de serviços*

e satisfaz (2.45), onde $\pi_i \triangleq \lim_{t \rightarrow \infty} P[s_r(t) = r_i | u_s]$ é a probabilidade de que o requisitante de serviços (RS) está no estado r_i . E o atraso de identificação T_p para todos os estados do requisitante de serviços r_i , $i = 0, \dots, R - 1$ é mostrado em (2.46).

$$|E[\hat{p}_{r_i, r_j}^{RS}(t_s + T_{p_i}, \lambda)] - p_{r_i, r_j}^{RS}| \leq \epsilon \quad (2.44)$$

$$T_{p_i} \geq \frac{\ln(\epsilon)}{\pi_i \ln(\lambda)} \quad (2.45)$$

$$T_p \triangleq \max_i \{T_{p_i}\} \geq \frac{\ln(\epsilon)}{\ln(\lambda)} \max_i \left\{ \frac{1}{\pi_i} \right\} \quad (2.46)$$

Prova. Dado que o valor da estimativa $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ só é atualizado quando o estado do requisitante de serviços $s_r(t - 1)$ é igual a r_i , consideremos que o estado do requisitante de serviços $s_r(t - 1)$ é observado por T_{p_i} intervalos de amostragem e há n ocorrências do estado r_i como mostrado na Fig 2.34. Então, para um T_{p_i} suficientemente grande, temos $n/T_{p_i} \cong \pi_i \Rightarrow n \cong T_{p_i} \pi_i$.

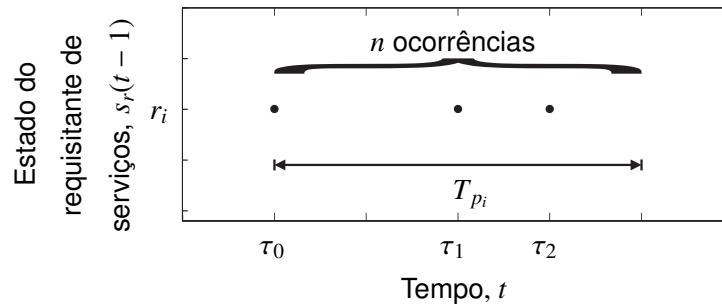


Figura 2.34: Intervalos de amostragem quando o valor da estimativa $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ é atualizado.

Agora, consideremos a sequência $(\tau_0, \tau_1, \tau_2, \dots)$ de intervalos $\tau_n = t_s + t$ tal que $s_r(t_s + t - 1) = r_i$ por $n = 0, 1, 2, \dots$ como mostrado na Fig. 2.34, i.e., a sequência de intervalos quando o valor da estimativa $\hat{p}_{r_i, r_j}^{RS}(t_s + t, \lambda)$ é atualizada. Calcularemos agora o número necessário n de instantes τ_n tal que as estimativas $\hat{p}_{r_i, r_j}^{RS}(\tau_n, \lambda)$ converjam para a nova carga de trabalho u_s . Do Algoritmo 2, o valor esperado da estimativa $\hat{p}_{r_i, r_j}^{RS}(\tau_n, \lambda)$ dado $\hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda) = \hat{P}(X = 1)$ é mostrado em (2.47). Tomando o valor esperado em ambos os lados da equação uma segunda vez, a expressão em (2.47) é reescrita em (2.48) e estendida para $E[\hat{p}_{r_i, r_j}^{RS}(\tau_0, \lambda)]$ em (2.49).

$$\begin{aligned} E[\hat{p}_{r_i, r_j}^{RS}(\tau_n, \lambda) | \hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda)] &= \\ &= p_{r_i, r_j}^{RS} \left(1 - \lambda \left(1 - \hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda) \right) \right) + \left(1 - p_{r_i, r_j}^{RS} \right) \lambda \hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda) = \\ &= (1 - \lambda) p_{r_i, r_j}^{RS} + \lambda \hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda) \end{aligned} \quad (2.47)$$

$$E \left[\hat{p}_{r_i, r_j}^{RS}(\tau_n, \lambda) \right] = (1 - \lambda) p_{r_i, r_j}^{RS} + \lambda E \left[\hat{p}_{r_i, r_j}^{RS}(\tau_{n-1}, \lambda) \right] = \quad (2.48)$$

$$= (1 - \lambda^n) p_{r_i, r_j}^{RS} + \lambda^n E \left[\hat{p}_{r_i, r_j}^{RS}(\tau_0, \lambda) \right] \quad (2.49)$$

O número necessário n de instantes τ_n para que as estimativas $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$ convirjam para a nova carga de trabalho u_s deve satisfazer a inequação em (2.44). Usando a expressão para $E \left[\hat{p}_{r_i, r_j}^{RS}(\tau_n, \lambda) \right]$ in (2.49), essa inequação é reescrita em (2.50). Considerando que $|E \left[\hat{p}_{r_i, r_j}^{RS}(\tau_0, \lambda) \right] - p_{r_i, r_j}^{RS}(u_s)| \leq 1$, assim como $0 < \epsilon < 1$ e $0 < \lambda < 1$ a inequação em (2.51) é verdadeira.

$$\lambda^n |E \left[\hat{p}_{r_i, r_j}^{SR}(\tau_0, \lambda) \right] - p_{r_i, r_j}^{SR}| \leq \epsilon \Rightarrow \quad (2.50)$$

$$\Rightarrow \lambda^n \leq \frac{\epsilon}{|E \left[\hat{p}_{r_i, r_j}^{SR}(\tau_0, \lambda) \right] - p_{r_i, r_j}^{SR}|} \Rightarrow$$

$$\Rightarrow \lambda^n \leq \epsilon \Rightarrow n \geq \frac{\ln(\epsilon)}{\ln(\lambda)} \quad (2.51)$$

Repondo $n \cong T_p \pi_i$ em (2.51), então a inequação (2.45) é verdadeira. E o atraso de identificação para todos os estados do requisitante de serviços T_p pode ser definido como o máximo de todos os atrasos de identificação T_{p_i} , $i = 0, \dots, R - 1$, como expresso em (2.46). ■

Quando o parâmetro $0 < \lambda < 1$ aumenta, o atraso de identificação T_p também aumenta. Assim, para combater o atraso de identificação, o parâmetro λ deve ser mantido tão pequeno quanto possível, por exemplo, igual a 0,9.

2.3.2.3 Erro de amostragem

Após o atraso de identificação de uma carga estacionária u_r para outra u_s , como mostrado na Fig. 2.35, as estimativas oscilam em torno dos valores de probabilidades de transição verdadeiros $p_{r_i, r_j}^{RS}(t)$. Esse efeito é denominado o *erro de amostragem* (nomeado *resolution error* em [1]). Diminuir o parâmetro λ é benéfico para reduzir o atraso de identificação, mas isso aumenta o erro de amostragem.

A variância da estimativa $\hat{p}_{r_i, r_j}^{RS}(t, \lambda)$, mostrada em (2.52), é um indicador do erro de amostragem (a prova para essa expressão da variância foi apresentada por Oommen e Rueda [40]). A variância $V \left(\hat{p}_{r_i, r_j}^{RS}(t, \lambda) \right)$ é mínima para $\lambda = 1$ e máxima para $\lambda = 0$. Assim, quando λ aumenta, o erro de amostragem diminui. Mas, como mostrado na Seção 2.3.2.2, quando λ aumenta, o atraso de identificação T_p também aumenta. Então, deve existir uma solução de compromisso entre erro de amostragem e atraso de identificação.

$$V \left(\hat{p}_{r_i, r_j}^{RS}(t, \lambda) \right) = \frac{(1 - \lambda) p_{r_i, r_j}^{RS}(t) \left(1 - p_{r_i, r_j}^{RS}(t) \right)}{(1 + \lambda)} \quad (2.52)$$

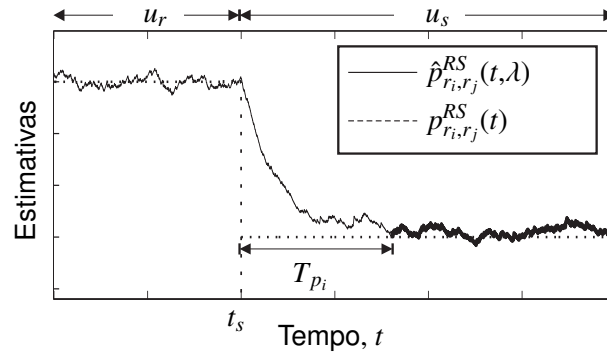


Figura 2.35: O erro de amostragem (linha de maior espessura) após a transição de uma carga de trabalho estacionária u_r para outra u_s .

2.3.3 Obtenção da política de energia

Uma vez obtidas as estimativas das probabilidades de transição do requisitante de serviços por meio de estimativas de máxima verossimilhança, estimação fraca baseada em aprendizado estocástico, ou outra técnica de identificação da carga de trabalho, é necessário obter uma política de energia apropriada para essas estimativas. A computação das técnicas de otimização introduzidas por Benini et al. [10] em tempo de execução é inviável, pois devido ao alto custo computacional de se recalcular um política ótima a cada período de amostragem, as hipóteses de que o gerenciador de energia toma decisões rápidas com gasto de energia desprezível podem ser violadas. Como alternativa para esse problema, Chung et al. [1] propõem um método de tabela de consulta (*look-up table*) com interpolação linear para a obtenção da política de energia.

Considerando um requisitante de serviços com dois estados, sua matriz de transição pode ser caracterizada pelas probabilidades de requisição de usuário $R_i \in [0, 1]$, $i = 0, 1$. Cada dimensão dessa matriz é amostrada com um número finito de amostras, cada uma denotada por R_{ij} , $j = 0, 1, \dots, NS_i - 1$, onde NS_i é o número de amostras para a dimensão i . Tal como ilustrado na Fig. 2.36, é construída uma tabela de consulta denominada *tabela de políticas*.

A cada elemento da tabela de políticas corresponde um requisitante de serviços com probabilidades de requisições de usuário (R_{0j}, R_{1k}) , ($j = 0, 1, \dots, NS_0 - 1$ e $k = 0, 1, \dots, NS_1 - 1$). Por sua vez, um elemento da tabela de políticas é uma tabela de decisão (ver Exemplo 1), que é uma tabela com número de linhas igual ao número de estados do sistema e número de colunas igual ao número de elementos do conjunto de comandos que podem ser passados pelo gerenciador de energia para o provedor de serviços. Para cada par (R_{0j}, R_{1k}) , é realizada uma otimização de política de energia ainda na fase de projeto do gerenciador de energia.

Em tempo de execução, num dado período de amostragem, uma estimativa (\hat{R}_0, \hat{R}_1) é obtida e são escolhidos dois índices consecutivos para cada dimensão: $R_{0j} \leq \hat{R}_0 \leq R_{0(j+1)}$ e $R_{1k} \leq \hat{R}_1 \leq$

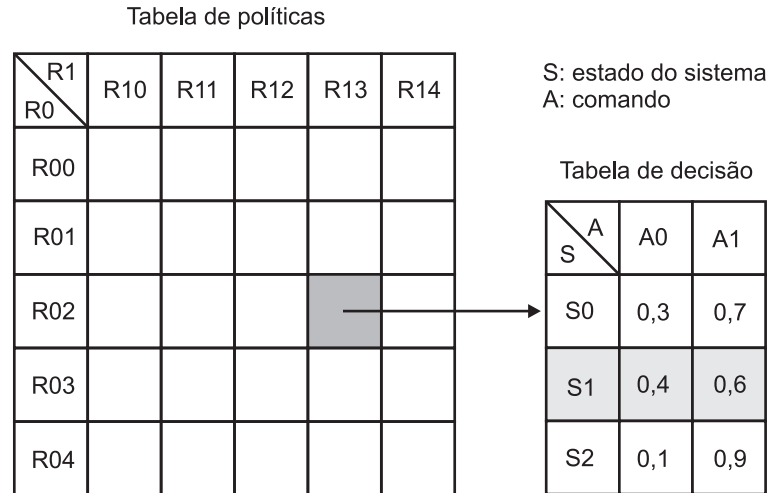


Figura 2.36: Exemplo de tabela de políticas com $NS_0 = NS_1 = 5$.

$R_{1(k+1)}$. O cálculo da tabela de decisão para o estado do sistema e o par (\hat{R}_0, \hat{R}_1) utiliza quatro tabelas de decisão correspondentes a esses índices escolhidos, havendo, em cada uma dessas, uma linha associada ao estado atual do sistema, denotada por EA . Uma técnica de interpolação bi-dimensional é aplicada a essas quatro linhas para calcular-se a linha de decisão final. Seja $f : \mathbb{R} \rightarrow \mathbb{R}$ uma função unidimensional, então é possível interpolar linearmente a imagem $f(x)$ de qualquer ponto x entre dois outros, x_1 e x_2 , por meio da expressão (2.53). Para realizar a técnica de interpolação bi-dimensional, a de interpolação unidimensional apresentada em (2.53) é aplicada recursivamente. Nos casos em que $\hat{R}_i > R_{i(NS_i-1)}$ ou $\hat{R}_i < R_{i0}$, deve ser utilizada extrapolação.

$$f(x) = \frac{(f(x_2) - f(x_1))x + x_2f(x_1) - x_1f(x_2)}{x_2 - x_1} \tag{2.53}$$

Exemplo 5 Seja $R_{0j} \in \{0, 0,25, 0,5, 0,75, 1\}$ e $R_{1k} \in \{0, 0,25, 0,5, 0,75, 1\}$. As estimativas de probabilidade de requisição são $\hat{R}_0 = 1/3$ e $\hat{R}_1 = 0,625$. Assim $R_{01} = 0,25$, $R_{02} = 0,5$, $R_{12} = 0,5$, $R_{13} = 0,75$, pois $R_{01} < \hat{R}_0 < R_{02}$ e $R_{12} < \hat{R}_1 < R_{13}$, tal como ilustrado na Fig. 2.37. Um exemplo do processo iterativo de interpolação é mostrado na Fig. 2.38. Como exemplo, o cálculo da probabilidade referente ao comando A0 para (R_{01}, \hat{R}_1) é mostrado em (2.54), e o cálculo da probabilidade referente ao comando A0 para (\hat{R}_0, \hat{R}_1) é mostrado em (2.55).

$$\frac{(0,5 - 0,7) \cdot 0,625 + 0,75 \cdot 0,7 - 0,5 \cdot 0,5}{0,75 - 0,5} = 0,6 \tag{2.54}$$

$$\frac{(0,3 - 0,6) \cdot \frac{1}{3} + 0,5 \cdot 0,6 - 0,25 \cdot 0,3}{0,5 - 0,25} = 0,5 \tag{2.55}$$

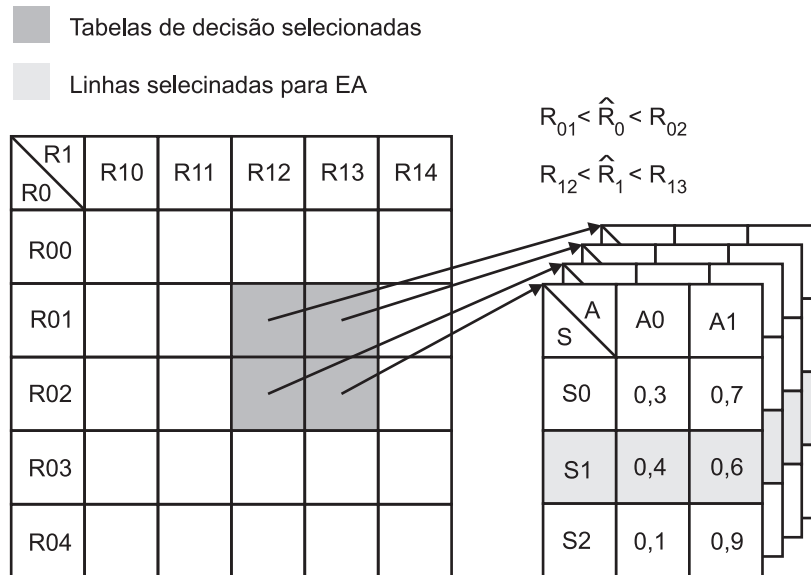


Figura 2.37: Tabela de decisão e exemplo de seleção de linhas da tabela de políticas.

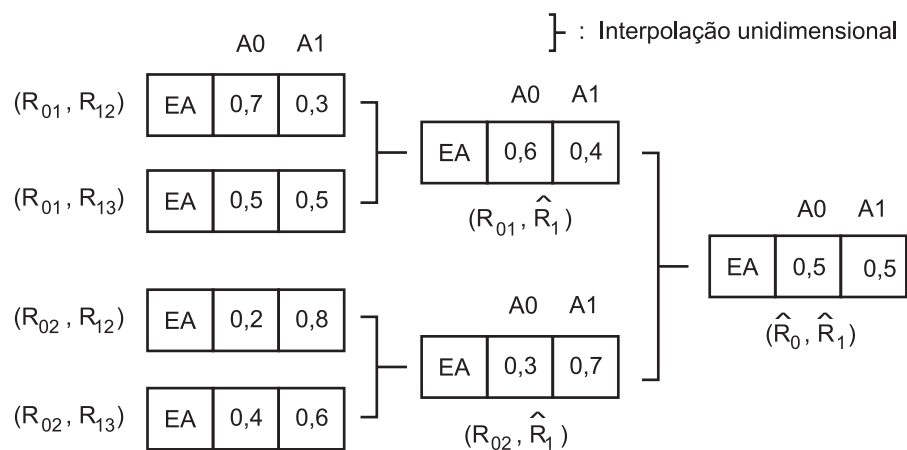


Figura 2.38: Exemplo de tabela de políticas com $NS_0 = NS_1 = 5$.

2.3.4 Comparação de várias técnicas de estimação de cargas de trabalho

Como Chung et al. [1] e Ren, Krogh e Marculescu [37], nesta seção também é simulada a técnica de gerenciamento de energia (*GDE*) apresentada neste capítulo com base no modelo de sistema mostrado na Fig. 2.1. Várias simulações foram usadas para tornar evidente a eficiência da técnica introduzida neste capítulo com relação à obtenção do erro de amostragem pequeno para comprimentos de janela grandes, e pequenos atrasos de identificação para comprimentos de janela pequenos. Verifica-se também a possibilidade de adaptação quando a carga de trabalho de um sistema real é estacionária ou altamente não-estacionária.

Neste capítulo são comparadas quatro diferentes técnicas usadas para estimar as probabilidades de transição do requisitante de serviços executando para a mesma carga de trabalho:

1. a técnica de multi-janela [1] usando estimativas de máxima-verossimilhança (*EMV*) com comprimentos de janela fixos $l_w = 10, 20, 30, \dots, 500$;
2. a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima-verossimilhança (*JMC-EMV*) com os comprimentos de janela $l = (50, 100, 150, \dots, 500)$ no bloco *janelas de estimativas* da Fig. 2.17;
3. a técnica de estimação fraca baseada em aprendizado estocástico (*EFAE*) [40] com parâmetros fixos $\lambda = 0,9, 0,91, 0,92, \dots, 0,99$;
4. a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*) com os parâmetros $\lambda = 0,97, 0,98, 0,99$ no bloco *janelas de estimativas* da Fig. 2.17.

A corrente consumida pelo provedor de serviços e a perda de requisições média obtidas com essas técnicas são comparadas com a política adaptativa ótima (ideal) (*PAO*), que não requer qualquer técnica de estimação de carga de trabalho pois considera-se que as cargas de trabalho são conhecidas *a-priori*, o que não acontece em geral para cargas de trabalho reais.

Considere um provedor de serviços (*PS*) ($\mathcal{M}_{PS}(a), b(s,a), c(s,a)$) onde: $\mathcal{M}_{PS}(a)$ é uma cadeia de Markov estacionária controlável com conjunto de estados $\mathcal{S} = \{s_0, s_1\}$; conjunto de controle $\mathcal{A} = \{a_0, a_1\}$, significando *trocar para s_0* e *trocar para s_1* , respectivamente; matriz de transição $\mathbf{P}^{PS}(a)$ em (2.56); a matriz $c(s,a)$ em (2.57) especifica o consumo de corrente (em *mA*) do provedor de serviços quando no estado s e quando o comando a é passado pelo gerenciador de energia; e a matriz $b(s,a)$ em (2.58) especifica a taxa de serviço do provedor de serviços quando no estado s e quando o comando a é passado pelo gerenciador de energia, i.e., a probabilidade de servir uma requisição em um período de amostragem. O estado s_0 tem maior consumo e melhor desempenho do que o estado s_1 . De (2.56), as probabilidades de transição

do provedor de serviços são modeladas como determinísticas, porque, no sistema modelado, os tempos de transição entre os estados do provedor de serviços são desprezíveis com respeito ao intervalo de amostragem. Pela mesma razão, o consumo de corrente $c(s,a)$ em (2.57) e a taxa de serviço $b(s,a)$ em (2.58) somente dependem do estado atual s do provedor de serviços.

$$\begin{aligned}
 P^{PS}(a_0) &= \begin{matrix} & s_0 & s_1 \\ s_0 & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ s_1 & \begin{bmatrix} 1 & 0 \end{bmatrix} \end{matrix} \\
 P^{PS}(a_1) &= \begin{matrix} & s_0 & s_1 \\ s_0 & \begin{bmatrix} 0 & 1 \end{bmatrix} \\ s_1 & \begin{bmatrix} 0 & 1 \end{bmatrix} \end{matrix} \tag{2.56}
 \end{aligned}$$

$$c(s,a) = \begin{matrix} & a_0 & a_1 \\ s_0 & \begin{bmatrix} 300 & 300 \end{bmatrix} \\ s_1 & \begin{bmatrix} 200 & 200 \end{bmatrix} \end{matrix} \tag{2.57}$$

$$b(s,a) = \begin{matrix} & a_0 & a_1 \\ s_0 & \begin{bmatrix} 1 & 1 \end{bmatrix} \\ s_1 & \begin{bmatrix} 0,8 & 0,8 \end{bmatrix} \end{matrix} \tag{2.58}$$

A fila de serviços (FS) $\mathcal{M}_{FS}(a, s, r)$ de comprimento $Q = 1$ tem conjunto de estados $\mathcal{Q} = \{q_0, q_1\}$, i.e., em q_0 a fila de serviços está vazia, e em q_1 a fila de serviços possui uma requisição pendente. A matriz de transição \mathbf{P}^{FS} , cujos elementos são obtidos por meio de (2.3), é mostrada em (2.59) e (2.60).

$$P^{FS}(a, s, r = 0) = \begin{matrix} & q_0 & q_1 \\ q_0 & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ q_1 & \begin{bmatrix} b(s,a) & 1 - b(s,a) \end{bmatrix} \end{matrix} \tag{2.59}$$

$$P^{FS}(a, s, r = 1) = \begin{matrix} & q_0 & q_1 \\ q_0 & \begin{bmatrix} b(s,a) & 1 - b(s,a) \end{bmatrix} \\ q_1 & \begin{bmatrix} 0 & 1 \end{bmatrix} \end{matrix} \tag{2.60}$$

Considere o requisitante de serviços não-estacionário (RS) especificado em (2.61). Essa carga de trabalho não-estacionária é composta da série de cargas de trabalho estacionárias: u_0 durante $0 \leq t \leq 1500$ períodos de amostragem, u_1 durante $1500 < t \leq 3000$ períodos de amostragem, e u_2 durante $3000 < t \leq 4500$ períodos de amostragem. Como mostrado na

Fig. 2.39, essa carga de trabalho não-estacionária foi assim escolhida por apresentar uma carga intensa u_0 , uma carga média u_1 e uma carga leve u_2 . Espera-se que esse sistema seja usado durante um horizonte de tempo $T_f = 4500$ períodos de amostragem. Por exemplo, num sistema computacional alimentado a bateria, o horizonte de tempo é a duração do intervalo de tempo desde o início da utilização do sistema, quando a bateria está com carga total, até o instante em que a bateria é considerada descarregada e o sistema não mais funciona. Para cada técnica de identificação de carga de trabalho e cada janela de tamanho fixo ou parâmetro, a simulação foi repetida 30 vezes.

$$\begin{aligned}
 p_{r_0, r_0}^{RS}(t) &= \begin{cases} 0,2, & \text{se } 0 \leq t \leq 1500 \\ 0,5, & \text{se } 1500 < t \leq 3000 \\ 0,8, & \text{se } 3000 < t \leq 4500 \end{cases} \\
 p_{r_1, r_1}^{RS}(t) &= \begin{cases} 0,8, & \text{se } 0 \leq t \leq 1500 \\ 0,5, & \text{se } 1500 < t \leq 3000 \\ 0,2, & \text{se } 3000 < t \leq 4500 \end{cases}
 \end{aligned} \tag{2.61}$$

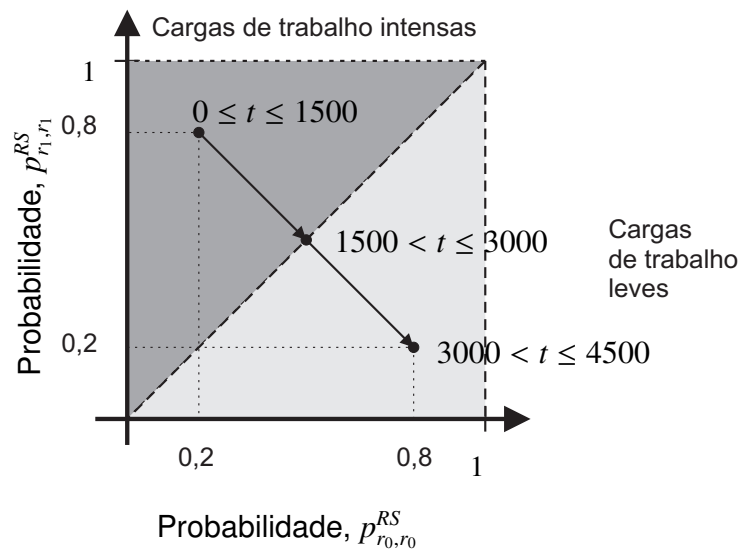


Figura 2.39: Probabilidades de transição do modelo do requisitante de serviços não-estacionário.

As restrições de penalidade de desempenho são: máxima ocupação média da fila de 25%, e máxima perda de requisições média de $L = 10\%$. Quatro indicadores foram obtidos para que fosse avaliada a qualidade de cada abordagem de estimação de carga de trabalho:

- I : consumo médio de corrente (mA)
- \hat{L} : perda de requisições média

- EMQ_{00} : erro médio quadrático da estimativa $\hat{p}_{r_0, r_0}^{RS}(t)$ com respeito a $p_{r_0, r_0}^{RS}(t)$
- EMQ_{11} : erro médio quadrático da estimativa $\hat{p}_{r_1, r_1}^{RS}(t)$ com respeito a $p_{r_1, r_1}^{RS}(t)$

Caso 1: A técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima-verossimilhança (*JMC-EMV*) é comparada aqui com a técnica de multi-janela [1] e a política adaptativa ótima (ideal) (*PAO*). Os erros médios quadráticos EMQ_{00} e EMQ_{11} obtidos com a técnica de multi-janela (*EMV*) [1] são respectivamente mostrados nas Figuras 2.40(a) e 2.40(b): (i) ambos aumentam para comprimentos de janela menores que 70 devido ao erro de amostragem, o principal efeito para comprimentos de janela pequenos; (ii) ambos aumentam para comprimentos de janela maiores que 70 devido ao atraso de identificação, o principal efeito para comprimentos de janela grandes. Por outro lado, os erros médios quadráticos EMQ_{00} e EMQ_{11} obtidos com a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima-verossimilhança (*JMC-EMV*) foram menores que todos os erros médios quadráticos obtidos com a técnica de multi-janela (*EMV*) [1] porque a primeira evita o erro de amostragem associado a comprimentos de janela pequenos, assim como o atraso de identificação para janelas de comprimento grandes.

O consumo de corrente e a perda de requisições média são mostrados nas Figuras 2.40(c) e 2.40(d) junto com barras verticais que representam o respectivo desvio padrão ($\pm\sigma$). Como mostrado no lado direito da Fig. 2.40(d), as barras de erro para a perda de requisições média, \hat{L} , para a técnica de multi-janela (*EMV*) [1] com comprimentos de janela $l_w = 10, 20, 30, 40$ possuem porções acima da restrição de perda de requisições $L = 0,1$. Assim, apesar de oferecerem os menores valores de consumo de corrente, esses comprimentos de janela não são aceitáveis com respeito a desempenho. Para a carga de trabalho desse exemplo, o consumo de corrente aumenta com o comprimento da janela e a perda de requisições média diminui. Como mostrado no lado esquerdo da Fig. 2.40(d), a barra de erro para a perda de requisições média, \hat{L} , na técnica introduzida de janelas deslizantes de múltiplos comprimentos (*JMC-EMV*), encontra-se quase inteiramente abaixo da restrição de perda de requisições $L = 0,1$. Assim, a corrente média para a técnica de janelas deslizantes de múltiplos comprimentos (*JMC-EMV*) é tão próxima quanto possível da corrente média da política adaptativa ótima (*PAO*), mas ainda oferecendo perda de requisições média aceitável.

Caso 2: A técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*) é comparada aqui com a técnica de estimação fraca baseada em aprendizado estocástico (*EFAE*) e a política adaptativa ótima (*PAO*) (ideal). Os erros médios quadráticos EMQ_{00} e EMQ_{11} obtidos com a técnica de estimação fraca baseada em aprendizado estocástico (*EFAE*) [40] são respectivamente mostrados nas Figuras 2.41(a) e 2.41(b): (i) ambos diminuem para o parâmetro λ menor que 0,97 devido

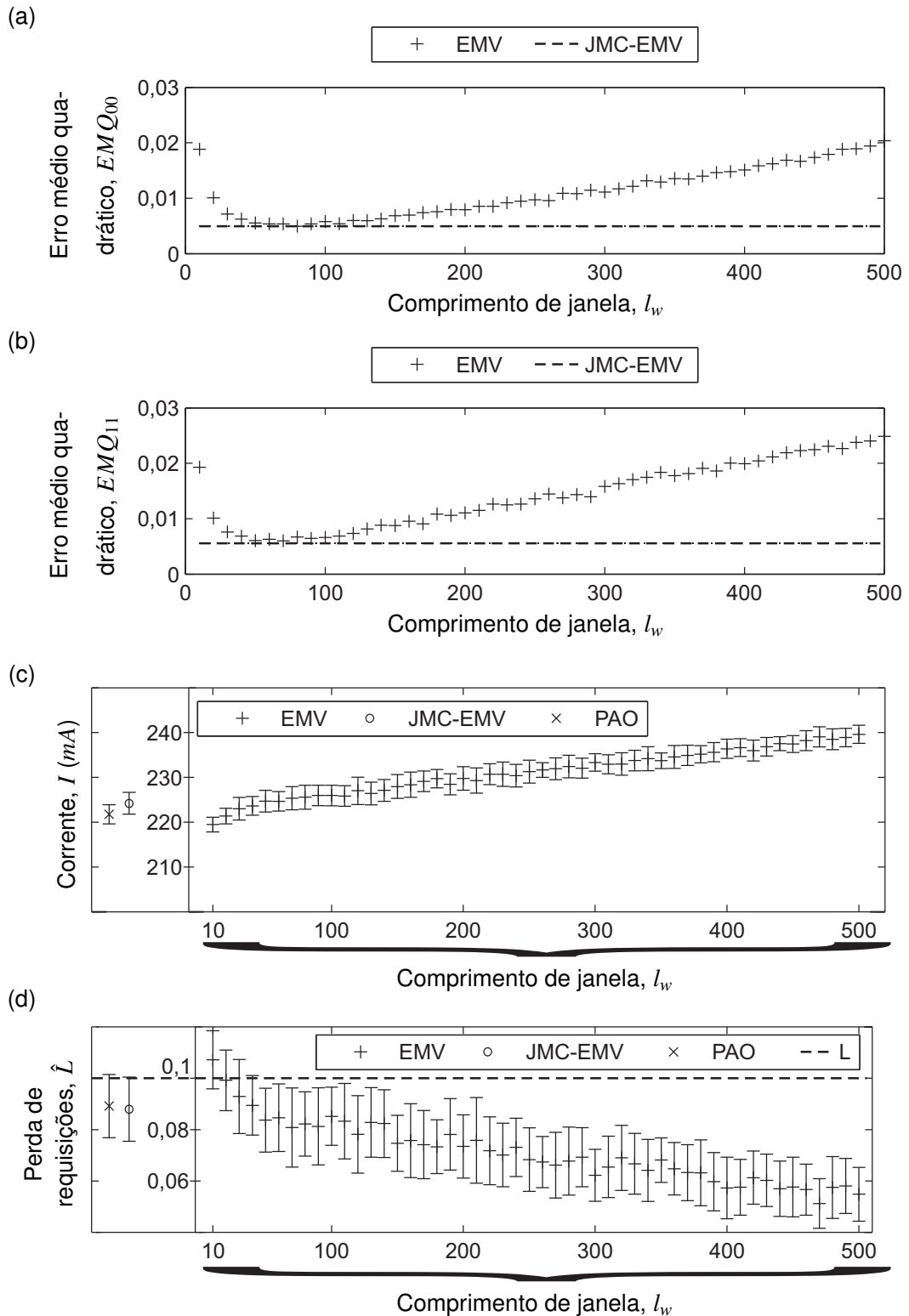


Figura 2.40: A abordagem de multi-janela (*EMV*), a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima verossimilhança (*JMC-EMV*), e a política adaptativa ótima (*PAO*) (ideal). (a) Erro médio quadrático (EMQ_{00}) das estimativas da probabilidade de transição $p_{r_0,r_0}^{RS}(t)$; (b) Erro médio quadrático (EMQ_{11}) das estimativas da probabilidade de transição $p_{r_1,r_1}^{RS}(t)$; (c) consumo de corrente médio; (d) perda de requisições média.

ao erro de amostragem, o principal efeito para pequenos valores de λ ; (ii) ambos aumentam para o parâmetro λ maior que 0,97 devido ao atraso de identificação, o principal efeito para valores grandes de λ . Os erros médios quadráticos EMQ_{00} e EMQ_{11} obtidos com a técnica introduzida de janelas deslizantes de múltiplos comprimentos (*JMC-EFAE*) foram próximos dos menores valores de erro médio quadrático obtidos com a estimação fraca baseada em aprendizado estocástico (*EFAE*) [40] dado que a primeira evita o erro de amostragem dos pequenos valores de λ e o atraso de identificação de valores grandes de λ .

O consumo de corrente e a perda de requisições média são mostrados nas Figuras 2.41(c) e 2.41(d) juntamente com barras de erro verticais que representam o respectivo desvio padrão ($\pm\sigma$). Como mostrado no lado direito da Fig. 2.41(d), todas as barras de erro para a perda de requisições média, \hat{L} , para a estimação fraca baseada em aprendizado estocástico (*EFAE*) [40] possuem porções acima da restrição especificada de perda de requisições $L = 0,1$. Assim, eles não são aceitáveis com respeito a desempenho. Para a carga de trabalho considerada nesse exemplo, o consumo de corrente aumenta com o parâmetro λ e a perda de requisições média diminui. Como mostrado no lado esquerdo da Fig. 2.41(d), a barra de erro para a perda de requisições média, \hat{L} , na técnica introduzida de janelas deslizantes de múltiplos comprimentos (*JMC-EFAE*) encontra-se inteiramente abaixo da restrição especificada de perda de requisições $L = 0,1$. A corrente média para a técnica introduzida de janelas deslizantes de múltiplos comprimentos (*JMC-EFAE*) não está próxima da corrente média da política adaptativa ótima (*PAO*) mas ainda oferece perda de requisição média aceitável.

A técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima verossimilhança (*JMC-EMV*) ou estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*) introduz algum *overhead* em comparação às técnicas de multi-janela (*EMV*), a estimação fraca baseada em aprendizado estocástico (*EFAE*) e a política adaptativa ótima (*PAO*) devido aos blocos *janelas de estimativas* e *seletor de janela* como mostrado na Fig. 2.17. Para avaliar esse *overhead*, o gerenciador de energia usando cada uma dessas técnicas de estimação foi perfilado usando o MatLab em um Pentium Core 2 Duo, com frequência de 2.13 GHz e 1Gb de memória. Como mostrado na Fig. 2.42, a política adaptativa ótima (*PAO*) apresenta menor tempo que as demais técnicas porque não há estimação da carga de trabalho. O *overhead* da técnica de multi-janela (*EMV*) aumenta linearmente com o comprimento da janela. O *overhead* da estimação fraca baseada em aprendizado estocástico (*EFAE*) é aproximadamente igual para todos os valores do parâmetro, porque um valor diferente não muda a quantidade de operações para realizar a estimação. O *overhead* da técnica de janelas deslizantes de múltiplos comprimentos combinada com estimativas de máxima verossimilhança (*JMC-EMV*) foi aproximadamente 13 vezes o *overhead* do de multi-janela (*EMV*) com maior comprimento de janela $l_w = 500$. E o *overhead* da técnica de janelas deslizantes de múltiplos comprimentos combinada

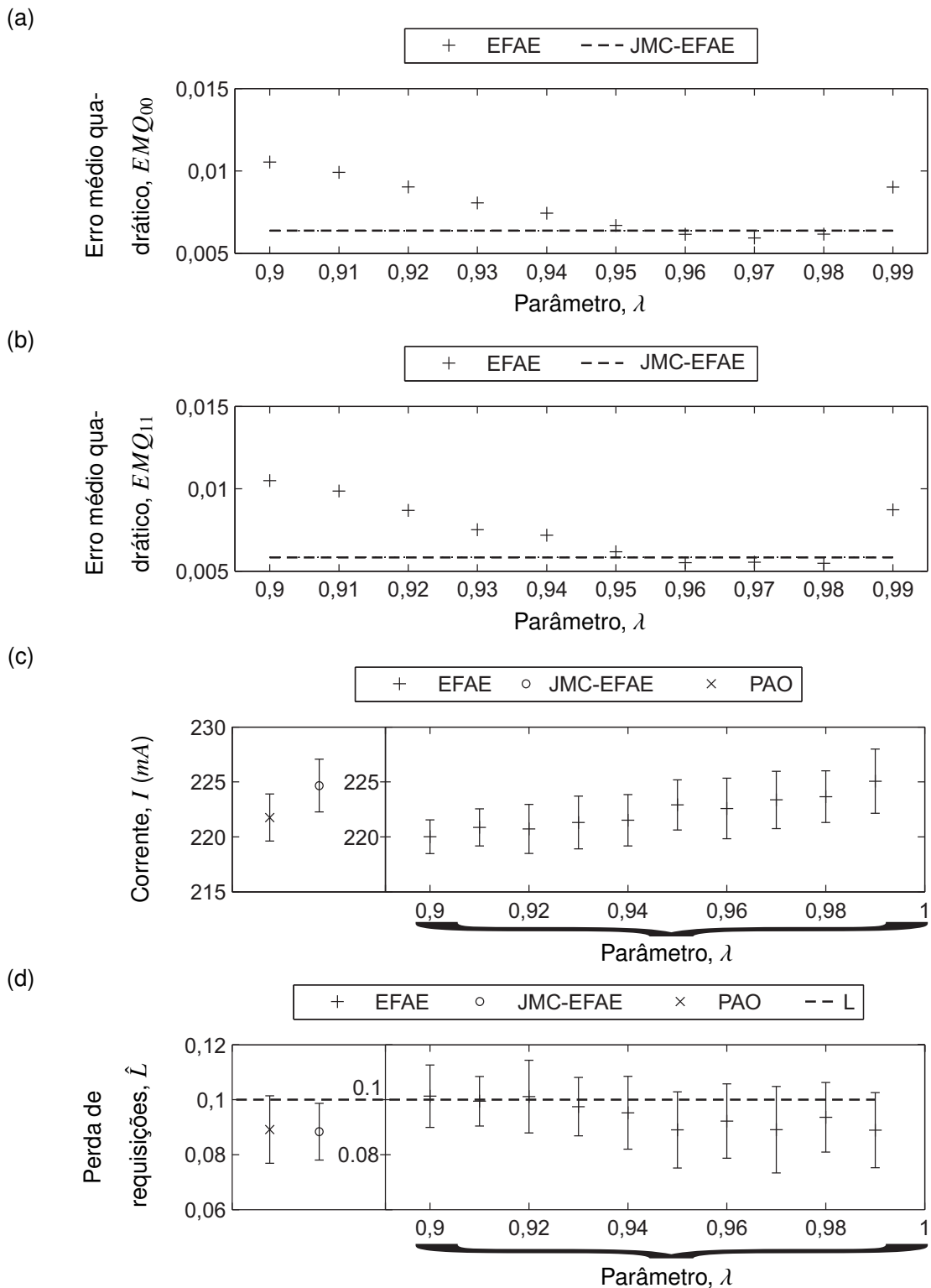


Figura 2.41: A técnica de estimação fraca baseada em aprendizado estocástico (*EFAE*), a técnica introduzida de janelas deslizantes de múltiplos comprimentos combinada com estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*), e a política adaptativa ótima (ideal) (*PAO*). (a) Erro médio quadrático (EMQ_{00}) das estimativas da probabilidade de transição $p_{r_0,r_0}^{RS}(t)$; (b) erro médio quadrático (EMQ_{11}) das estimativas da probabilidade de transição $p_{r_1,r_1}^{RS}(t)$; (c) consumo de corrente médio; (d) perda de requisições média.

com estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*) foi aproximadamente 3 vezes o *overhead* médio da técnica de estimação fraca baseada em aprendizado estocástico (*EFAE*) com parâmetros $\lambda = 0,9, 0,91, 0,92, \dots, 0,99$. Então, o *overhead* introduzido pela técnica de janelas deslizantes de múltiplos comprimentos combinada tanto com estimativas de máxima verossimilhança (*JMC-EMV*) ou estimação fraca baseada em aprendizado estocástico (*JMC-EFAE*) depende da combinação de comprimentos de janela (l_w) ou de parâmetros (λ) usados em cada técnica.

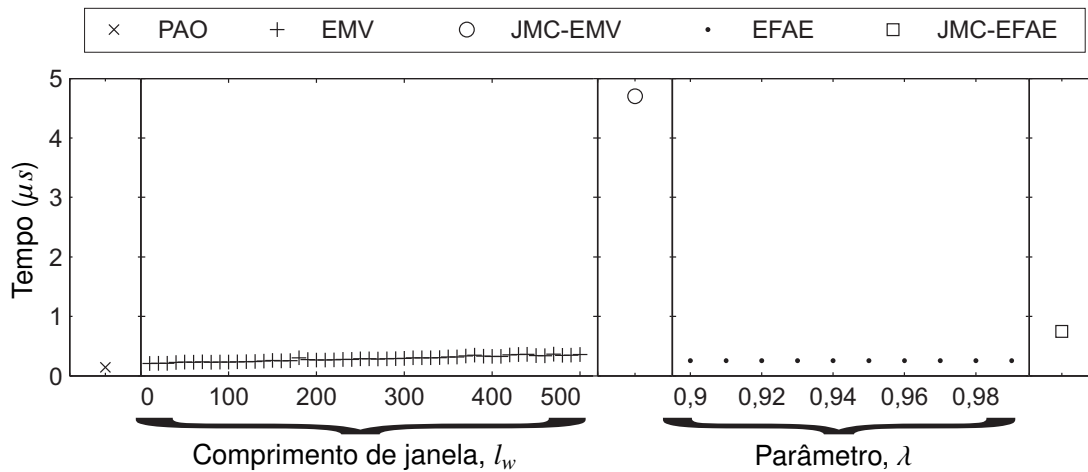


Figura 2.42: Tempo de simulação do gerenciador de energia.

2.4 Resumo

A identificação do modelo da carga de trabalho não-estacionária é essencial para técnicas de gerenciamento de energia (*GDE*) que dependem de modelos da carga de trabalho, tais como *time-out* [8], desligamento preditivo do sistema [41], árvore de aprendizado adaptativo [42] e abordagens de aprendizado da carga de trabalho baseadas em janelas deslizantes de comprimento fixo [1]. No último caso, o projetista do sistema é limitado a políticas sub-ótimas. Neste capítulo, o problema de gerenciamento de energia sob condições de carga de trabalho não-estacionária foi rigorosamente analisado e resolvido usando-se uma técnica de janelas deslizantes de múltiplos comprimentos (*JMC*) que resulta em políticas de gerenciamento de energia próximas do caso ótimo.

Os efeitos de atraso de identificação e erro de amostragem foram formalmente apresentados e expressões analíticas foram derivadas para demonstrar sua relação com os comprimentos de janela para estimativas de máxima verossimilhança, ou o parâmetro λ para estimação fraca baseada em aprendizado estocástico. Para a técnica introduzida de janelas deslizantes de múltiplos comprimentos (*JMC*), as estimativas foram calculadas considerando-se tanto múltiplas janelas

com múltiplos comprimentos ou múltiplos parâmetros. As variâncias das estimativas foram calculadas e usadas para prover o gerenciador de energia de relativamente boas estimativas da carga de trabalho verdadeira do sistema. Durante as transições entre cargas de trabalho estacionárias, comprimentos de janela pequenos foram usados para acompanhar as mudanças rápidas nas estatísticas da carga de trabalho, apesar do erro de amostragem maior. Após esse período de transição, comprimentos de janela maiores foram usados para reduzir o erro de amostragem. Assim, ao contrário das abordagens passadas usando janelas deslizantes de comprimento fixo, a técnica introduzida nesse capítulo de fato se beneficia tanto do pequeno erro de amostragem de comprimentos de janela grandes e o pequeno atraso de identificação de comprimentos de janela pequenos. A técnica introduzida é capaz de se adaptar caso a carga de trabalho de um sistema real for estacionária ou altamente não-estacionária. Além disso, a técnica de janelas deslizantes de múltiplos comprimentos pode ser facilmente combinada com qualquer técnica usada para calcular as estimativas. Nesse capítulo, ela foi combinada tanto com estimativas de máxima verossimilhança quanto com estimação fraca baseada em aprendizado estocástico. A aplicabilidade da técnica introduzida foi testada em um sistema modelado por cadeias de Markov discretas. A técnica de janelas deslizantes de múltiplos comprimentos superou as abordagens anteriores tanto em desempenho quanto em consumo de energia. Assim, o projeto do gerenciador de energia com estimação da carga de trabalho usando-se janelas deslizantes de múltiplos comprimentos é flexível e confiável.

Como verificado na Seção 2.3.4, não se garante que a solução do problema de otimização enunciado por Benini et al. [10] respeita a restrição de penalidade de desempenho para toda carga de trabalho, e assim, pode não oferecer uma economia de energia ótima. Há portanto duas possíveis consequências: (i) se a penalidade de desempenho final é maior que a restrição de penalidade de desempenho, então o desempenho do sistema é insatisfatório, ou seja o gerenciador de energia deteriorou o desempenho do sistema; (ii) se a penalidade de desempenho final é menor que a restrição de penalidade de desempenho, então ainda havia uma margem para reduzir o desempenho do sistema e assim economizar energia. Logo, uma política de gerenciamento de energia que usa modelos estocásticos pode ser considerada um sistema em malha aberta, porque ela não ajusta a penalidade de desempenho atual à restrição de desempenho de referência. A solução ótima depende fortemente da qualidade do modelo, i.e. os parâmetros do modelo estocástico e o conhecimento da carga de trabalho. Adicionalmente, se a restrição de penalidade de desempenho muda durante a operação do sistema com gerenciamento de energia, então existem duas opções possíveis: (i) o problema de otimização deve ser resolvido em tempo de execução, o que não é prático, porque isso demanda tempo e energia consideráveis; ou (ii) as políticas ótimas devem ser pré-computadas em tempo de projeto e armazenadas na memória, o que demanda espaço em memória considerável. Essas limitações da solução estocástica [10] justificam a procura por outras soluções para gerenciamento de energia. No Capítulo 3, será

apresentada a abordagem de Controle em malha fechada para gerenciamento de energia, que foi formulada com o objetivo de garantir que será atingido o valor exato da restrição de penalidade de desempenho em cada instante de tempo.

Capítulo 3

Controle em malha fechada para gerenciamento de energia

3.1 Introdução

Nas técnicas de controle em malha fechada para gerenciamento de energia em geral é utilizado um esquema em que a variável controlada corresponde ao desempenho do sistema, como mostrado na Fig. 3.1. O desempenho é medido e comparado ao valor de referência de desempenho, resultando no erro de desempenho, que é a entrada para um controlador, cuja saída é um comando para mudar o estado de energia do sistema com gerenciamento de energia. Então, deseja-se garantir que a penalidade de desempenho do sistema atingirá a restrição de penalidade de desempenho desejada. O controle em malha fechada tem sido empregado em muitos cenários diferentes de gerenciamento de energia, tais como: servidores [7], sistemas embarcados de tempo-real [23], rede-em-chip (NoC, *network-on-chip*) [25], microprocessadores embarcados [26] e reprodução de vídeo [27].

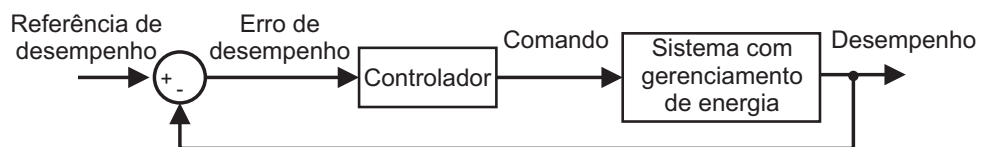


Figura 3.1: Gerenciamento de energia baseado no princípio de controle em malha fechada.

Neste capítulo, é apresentada uma abordagem de gerenciamento dinâmico de energia (*GDE*) baseada no modelo de um processador com escalonamento dinâmico de tensão e frequência (*DVFS*) como um processo de decisão de Markov usando a metodologia de Benini et al. [10], e a utilização desse modelo juntamente com controle em malha fechada para que a penalidade de desempenho do sistema seja igual à restrição de penalidade de desempenho [19]. A técnica de

gerenciamento dinâmico de energia (*GDE*) introduzida neste capítulo pode ser descrita com a ajuda do diagrama de blocos mostrado na Fig. 3.2. A ‘variável controlada’ é escolhida como a perda média de requisições $l(t)$ do sistema, que é um indicador de desempenho. A ‘variável manipulada’ é o estado do provedor de serviços s , que é alterado por meio do comando $a(t)$ passado pelo gerenciador de energia. Dado que o usuário do sistema aceita que a perda de requisições média $l(t)$ do sistema seja degradada até uma restrição de penalidade de desempenho L definida pelo usuário, então o gerenciador de energia pode passar um comando $a(t)$ para mudar o estado s do provedor de serviços e então reduzir a taxa de serviço $b(s, a)$. O ‘valor de referência’ $l_C(t)$ é a penalidade de desempenho máxima (tal como D usada para o *problema de otimização* de política em (2.11)). De acordo com os estados do requisitante de serviços r , fila de serviços q , e provedor de serviços s , pode ocorrer uma perda de requisição $l(t)$, como mostrado em (2.22), que é usada para estimar o desempenho do sistema por meio do bloco *Estimar perda de requisições média*, cuja saída $\hat{l}(t)$ é a estimativa de perda de requisições. Um controlador toma o valor de referência $l_C(t)$ e a estimativa de perda de requisições $\hat{l}(t)$ para produzir o sinal de controle $u(t)$. Mas, de acordo com a Definição 2 (Capítulo 2, pág. 13), o conjunto de controle \mathcal{A} do provedor de serviços é discreto e finito. Então, o bloco *Quantizador* é usado para mapear o sinal de controle contínuo $u(t)$ num comando discreto $a(t)$. No que se segue, um modelo estocástico para o controle de malha fechada para sistemas com gerenciamento de energia modelados por cadeias de Markov é analisado.

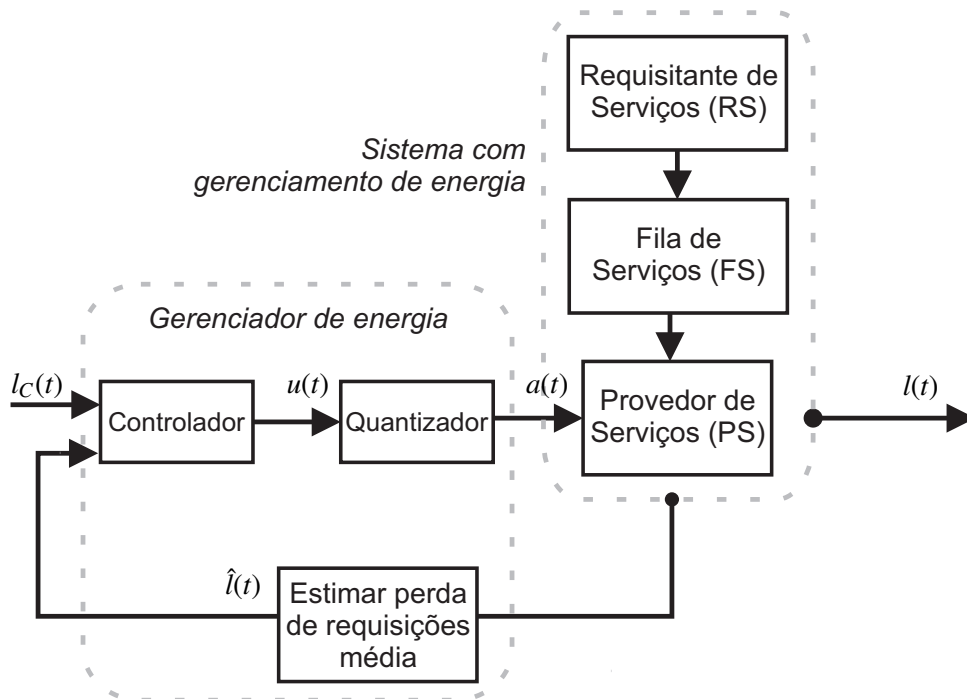


Figura 3.2: Diagrama de blocos do controle em malha fechada para gerenciamento de energia.

3.2 O sistema com gerenciamento de energia

Para sistemas tais como processadores com escalonamento dinâmico de tensão e frequência, o tempo de chaveamento entre os estados de tensão e frequência pode ser da ordem de nanosegundos [43], e assim pode ser considerado desprezível em relação ao período T do gerenciador de energia (nas simulações apresentadas neste capítulo, adotado como $T = 1s$), que é igual ao intervalo de amostragem do sistema modelado por cadeias de Markov. Então os custos de transição entre os estados de tensão e frequência (energia e tempo de transição) podem ser desprezados. Assim, a concepção do modelo em cadeia de Markov do provedor de serviços de tais sistemas é baseada em duas hipóteses: (i) o consumo $c(s,a)$ e a taxa de serviço $c(b,a)$ apenas dependem do estado atual do provedor de serviços s porque os custos de energia e tempo para as transições são desprezíveis; (ii) as transições entre os estados de energia são consideradas determinísticas, i.e., cada elemento $p_{s_i,s_j}^{PS}(a_l)$ da matriz de transição $\mathbf{P}^{PS}(a)$ é calculado em (3.1), em que o comando a_l significa “mudar para s_l ”.

$$p_{s_i,s_j}^{PS}(a_l) = \begin{cases} 1, & \text{se } s_j = s_l \\ 0, & \text{caso contrário} \end{cases} \quad (3.1)$$

Vamos ordenar os estados do provedor de serviços em ordem decrescente de taxas de serviço $b(s_0) > b(s_1) > \dots > b(s_{S-1})$, o que corresponde a uma ordem crescente de penalidades de desempenho $1 - b(s_0) < 1 - b(s_1) < \dots < 1 - b(s_{S-1})$. A razão para isso é o fato de que, como mostrado na Fig. 3.2, a estimativa da perda de requisições $\hat{l}(t)$ é comparada com o valor de referência $l_C(t)$. Considere o erro $e(t) = l_C(t) - \hat{l}(t)$. Se $\hat{l}(t) > l_C(t) \Rightarrow e(t) < 0$, i.e. a estimativa atual de perda de requisições é maior do que a restrição de penalidade de desempenho, então o gerenciador de energia deve passar um comando a para que o provedor de serviços entre num estado com menor penalidade de desempenho $1 - b(s)$. Da mesma forma, se $\hat{l}(t) < l_C(t) \Rightarrow e(t) > 0$, i.e., a estimativa da perda de requisições atual é menor do que a restrição de penalidade de desempenho, então o gerenciador de energia deve passar um comando a para que o provedor de serviços entre num estado com maior penalidade de desempenho $1 - b(s)$.

As expressões (2.8) e (2.9) derivadas no Capítulo 2 respectivamente para o consumo esperado e a penalidade de desempenho esperada do provedor de serviços (PS) no tempo t foram baseadas no fato de que, usando-se uma política estocástica π , o comando a a ser passado pelo gerenciador de energia (GE) é a realização de uma *variável aleatória* com distribuição discreta de probabilidades $\delta_x = (p_{a_0}, p_{a_1}, \dots, p_{a_{A-1}})$ que depende do estado atual x do sistema. No caso da técnica de controle em malha fechada mostrada na Fig. 3.2, o comando $a(t)$ é *determinístico*. Para escolher um comando discreto $a(t)$ a partir do sinal de controle contínuo $u(t)$, é utilizado o bloco *Quantizador*.

3.3 O bloco *Quantizador*

No bloco *Quantizador*, como mostrado na Fig. 3.2, o valor do sinal de controle $u(t)$ é mapeado ao comando $a(t)$ de acordo com a função $f_Q(u(t))$ em (3.2) e como mostrado na Fig. 3.3. Então, $f_Q(1 - b(s_i)) = a_i$, para $i = 0, \dots, X - 1$. Se $u(t) = 1 - b(s_i)$ é mantido constante por tempo suficiente, então $\lim_{t \rightarrow \infty} s(t) = s_i$ e $\lim_{t \rightarrow \infty} l(t) = 1 - b(s_i)$ como mostrado na Fig. 3.4. Então o quantizador é projetado para que o ganho estático seja unitário. Além disso, se $1 - b(s_i) < u(t) < 1 - b(s_{i+1})$ a menor perda de requisição esperada é $1 - b(s_i)$.

$$a(t) = f_Q(u(t)) = \begin{cases} a_0, & \text{se } u(t) < 1 - b(s_1) \\ a_1, & \text{se } 1 - b(s_1) \leq u(t) < 1 - b(s_2) \\ \vdots & \\ a_{X-1}, & \text{se } u(t) \geq 1 - b(s_{X-1}) \end{cases} \quad (3.2)$$

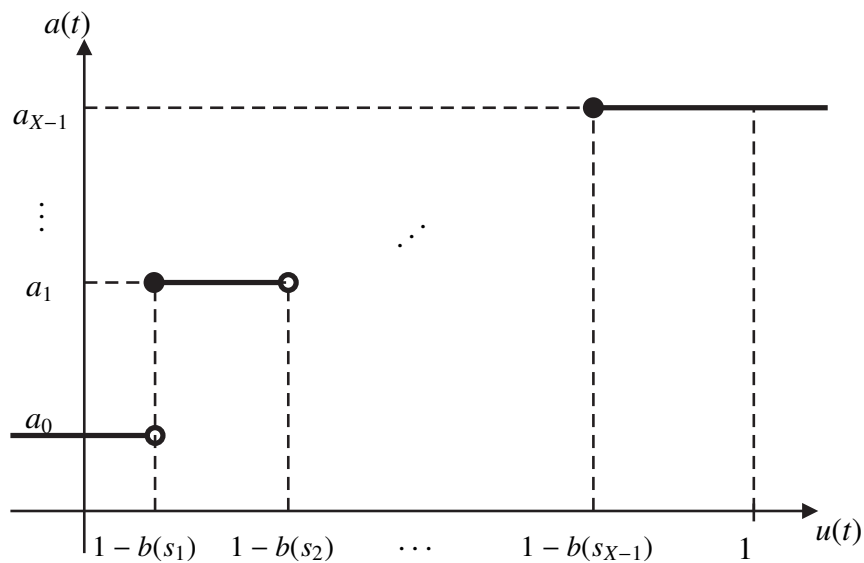


Figura 3.3: O quantizador de comandos.

3.4 O bloco *Estimar perda de requisições*

Como mostrado na Fig. 3.2, a saída do sistema com gerenciamento de energia é a perda de requisições $l(t)$, mas o objetivo do gerenciamento de energia é que a perda de requisições esperada $E[l(t)]$ deva satisfazer a restrição de penalidade de desempenho, a qual, na formulação do problema de controle, é representada pelo sinal de referência $l_C(t)$. Assim, o bloco *Estimar perda de requisições* tem como entrada a perda de requisições atual $l(t)$, e como saída uma estimativa $\hat{l}(t)$ a ser comparada com a restrição de penalidade de desempenho $l_C(t)$. A perda de requisições $l(t)$ pode assumir os valores 0 e 1, como apresentado em (2.22) (Capítulo 2, pág. 24). Contudo,

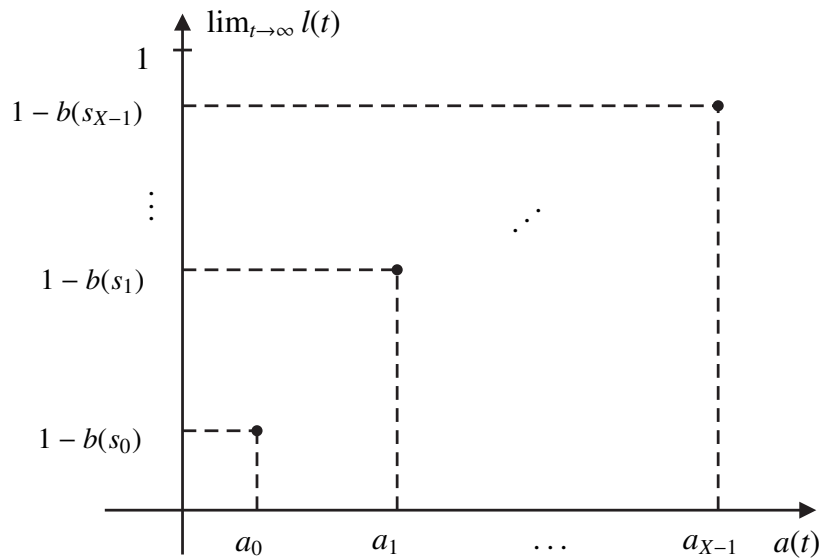


Figura 3.4: A perda de requisições esperada.

espera-se que a variável controlada possa assumir valores contínuos. Uma vez que estamos interessados no valor esperado de $l(t)$, uma abordagem possível é utilizar a Estimção fraca baseada em aprendizado estocástico [40] para obter o valor médio de $l(t)$.

A operação realizada no bloco *Estimar perda de requisições* (Fig. 3.2) é descrita no Algoritmo 3, cuja versão para distribuições binomiais foi primeiramente introduzida por Oommen e Rueda [40]. Nesse algoritmo, o parâmetro $0 < \lambda < 1$, $\hat{P}(X = 1)$ e $\hat{P}(X = 0)$ são variáveis internas. Em $t = 0$, elas são inicializadas para valores definidos pelo usuário, e.g. $\hat{P}(X = 1) = l_C(0)$ e $\hat{P}(X = 0) = 1 - l_C(0)$, e são atualizadas a cada intervalo de amostragem t .

Algoritmo 3: Estimar perda de requisições.

Dados: $\hat{P}(X = 1)$, $\hat{P}(X = 0)$ e λ

Entrada: $l(t)$

Saída: $\hat{l}(t)$

se $l(t) \neq 0$ **então**

$\hat{P}(X = 0) = \lambda \hat{P}(X = 0)$;

senão

$\hat{P}(X = 0) = 1 - \lambda \hat{P}(X = 1)$;

$\hat{P}(X = 1) = 1 - \hat{P}(X = 0)$;

$\hat{l}(t) = \hat{P}(X = 1)$;

A diminuição do parâmetro λ é benéfico para reduzir o atraso de identificação, mas também aumenta o erro de amostragem. De acordo com Oommen e Rueda [40], um intervalo interessante para esse parâmetro é $0,9 \leq \lambda < 1$. Neste trabalho, é utilizado o valor $\lambda = 0,999$.

3.5 O modelo discreto do sistema

Como mostrado na Fig. 3.2, as interfaces do controlador com o provedor de serviços são o valor de referência $l_C(t)$, a estimativa de perda de requisições $\hat{l}(t)$ e o sinal de controle $u(t)$. O controlador não possui conhecimento nem da estrutura interna do sistema com gerenciamento de energia nem de seu modelo em cadeia de Markov. Assim, os blocos *requisitante de serviços*, *fila de serviços*, *provedor de serviços*, *Quantizador* e *Estimar perda de requisições* podem ser representados como uma entidade única, denominada de *processo*, como mostrado na Fig. 3.5 (não confundir com o termo 'processo' no contexto de sistemas operacionais que define uma execução de um programa de computador).

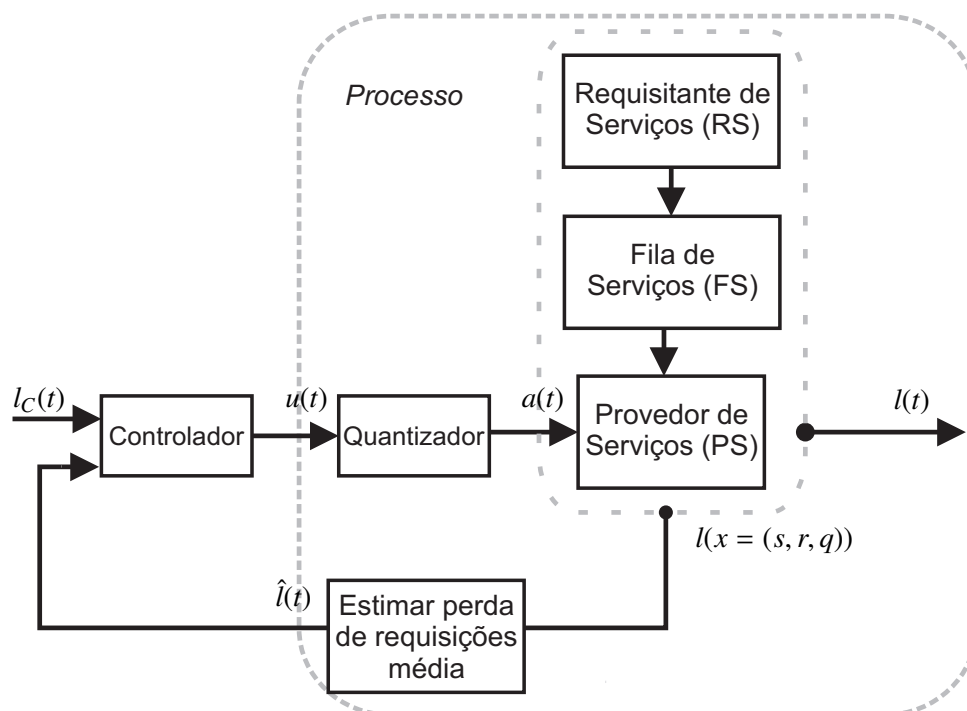


Figura 3.5: Diagrama de blocos do controle em malha fechada para gerenciamento de energia, destacando-se controlador e processo.

Para projetar o controlador, é necessário identificar a função de transferência do processo $G_P(z)$. Infelizmente, o modelo estocástico em (2.4) a (2.7) e (2.9) não é útil para essa tarefa por várias razões: (i) ele não é nem na forma de equações diferenciais nem na forma de espaço de estados; (ii) a entrada a é um parâmetro da matriz de transição do sistema, mas ele não é aditivo para a distribuição de probabilidades dos estados do sistema $p(t)$; (iii) o valor instantâneo da perda de requisições esperada não pode ser diretamente medido porque se trata de um valor esperado. Mas a função de transferência do processo $G_P(z)$ pode ser estimada utilizando-se alguma estrutura de modelo [44] tal como ARX em (3.3) ou ARMAX em (3.4), onde $A(q)$, $B(q)$ e $C(q)$ são os polinômios definidos em (3.5), (3.6) e (3.7) respectivamente, e $e_C(t)$ é o erro

do modelo.

$$A(q)\hat{l}(t) = B(q)u(t) + e_C(t) \quad (3.3)$$

$$A(q)\hat{l}(t) = B(q)u(t) + C(q)e_C(t) \quad (3.4)$$

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (3.5)$$

$$B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (3.6)$$

$$C(q) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c} \quad (3.7)$$

O processo de identificação da função de transferência do processo $G_P(z)$ deve seguir os passos [44]: (i) projetar um experimento para coletar dados da entrada $u(t)$ e saída $\hat{l}(t)$; (ii) escolher um conjunto de modelos (ARX, ARMAX, ...); (iii) escolher o melhor modelo no conjunto de acordo com um dado critério.

3.6 O projeto do controlador

Vamos supor que o processo pode ser descrito pelo sistema de entrada única e saída única (SISO) em (3.8). Seja $\deg F$ o grau do polinômio F . O parâmetro $d_0 = \deg A - \deg B$ é denominado o *excesso de pólos*. Vamos aplicar a alocação de pólos de grau mínimo (MDPP) [45] para projetar o controlador linear geral em (3.9), onde R , S e T são polinômios. O controlador e o processo são mostrados na Fig. 3.6.

$$A\hat{l}(t) = B[u(t) + e_C(t)] \quad (3.8)$$

$$Ru(t) = Tl_C(t) - S\hat{l}(t) \quad (3.9)$$

Isolando $u(t)$ em (3.8) e o repondo em (3.9), é obtida a expressão para o sistema em malha fechada em (3.10). Então, o polinômio característico de malha fechada é A_c , apresentado em em (3.11). A_c é um parâmetro de projeto e é escolhido de tal forma que ofereça as propriedades desejadas do sistema em malha fechada. Então, R e S são resolvidos de (3.11), que é denominada a *equação de Diofanto*. Outro parâmetro de projeto é a resposta desejada ao sinal

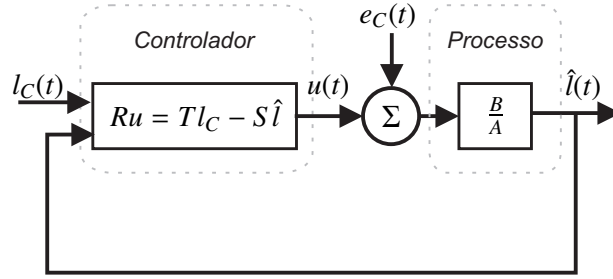


Figura 3.6: Diagrama de blocos do modelo discreto do controlador e do sistema com gerenciamento de energia.

de controle como representado pela dinâmica em (3.12). Então a condição em (3.13) deve ser válida. O seguidor de modelo em (3.13) é usado para determinar o polinômio T .

$$\hat{l}(t) = \frac{BT}{RA + BS} l_C(t) + \frac{RB}{RA + BS} e_C(t) \quad (3.10)$$

$$RA + BS = A_c \quad (3.11)$$

$$A_m \hat{l}_m(t) = B_m l_C(t) \quad (3.12)$$

$$\frac{BT}{RA + BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} \quad (3.13)$$

O bloco *Quantizador* é de fato um atuador não-linear com a característica de saturação em (3.14). Então é necessário projetar o controlador considerando os efeitos do bloco *Quantizador*. O controlador com compensação *antiwindup* [46], como representado na Fig. 3.7, é expresso em (3.15), onde $A_{aw}(q)$ é o polinômio característico de um observador *antiwindup*. E o caso particular como observador *deadbeat*, fazendo-se $A_{aw}(q) = 1$, é mostrado em (3.16).

$$\text{sat}(u) = \begin{cases} 0 & \text{se } u \leq 0 \\ u & \text{se } 0 < u < 1 \\ 1 & \text{se } u \geq 1 \end{cases} \quad (3.14)$$

$$\begin{aligned} A_{aw} v &= T l_C - S \hat{l} + (A_{aw} - R) u \\ u &= \text{sat}(v) \end{aligned} \quad (3.15)$$

$$u(t) = \text{sat}(T l_C(t) - S \hat{l}(t) + (1 - R) u(t)) \quad (3.16)$$

Exemplo 6 Do procedimento de identificação descrito na seção anterior, um processo pode ser descrito pelo sistema de entrada única e saída única (SISO) em (3.17). Então $B(q) = b_1$ e $A(q) = q + a_1$.

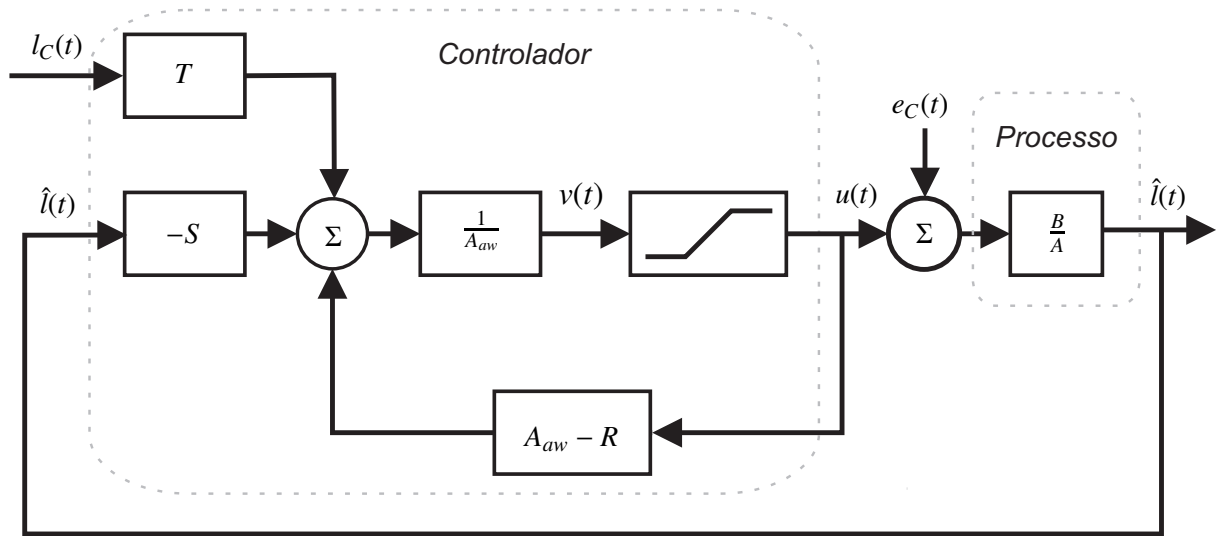


Figura 3.7: Diagrama de blocos do controlador e processo considerando o bloco *Quantizador* como um atuador não-linear.

$$(q + a_1)\hat{l}(t) = b_1u(t) \quad (3.17)$$

É possível usar e.g. o projeto deadbeat [46], onde os pólos desejados são todos alocados na origem, resultando no polinômio característico do sistema de malha fechada como $P(z) = z^n$, onde n é a ordem do polinômio característico do sistema em malha fechada, e, no caso atual, é 2. Então, requer-se o seguidor de modelo em (3.18).

$$\frac{BT}{RA + BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} = \frac{1}{q^2} \quad (3.18)$$

Utilizando o procedimento de alocação de pólos de grau mínimo (MDPP) [45], é obtida a lei de controle em (3.19). E o controlador com compensação antiwindup usando um observador deadbeat é mostrado em (3.20).

$$Ru = Tl_c - S\hat{l} \Rightarrow u(t) = a_1u(t-1) + \frac{1}{b_1}l_c(t-1) - \frac{a_1^2}{b_1}\hat{l}(t-1) \quad (3.19)$$

$$u(t) = \text{sat} \left(a_1u(t-1) + \frac{1}{b_1}l_c(t-1) - \frac{a_1^2}{b_1}\hat{l}(t-1) \right) \quad (3.20)$$

3.7 Resultados e Discussão

Nessa seção, são comparadas duas técnicas diferentes de gerenciamento de energia:

1. a abordagem estocástica [10] usando processos de decisão de Markov
2. a técnica proposta de controle em malha fechada para sistemas com gerenciamento de energia e modelados por processos de decisão de Markov

Considere um processador com escalonamento dinâmico de tensão e frequência cujos estados de energia são mostrados na Tabela 3.1. No que se segue, as frequências f e tensões v serão normalizadas. Por exemplo, a frequência normalizada f_{n_i} do estado s_1 é o quociente entre as frequências 168 Mhz e 192 Mhz. De acordo com [6], a potência dinâmica (de chaveamento) P de circuitos CMOS é proporcional a fv^2 . Então as potências dinâmicas dos estados de energia da Tabela 3.1 também podem ser normalizadas, como demonstrado em (3.21) para a potência normalizada P_{n_i} de um estado i . Na Tabela 3.2 são apresentadas as potências normalizadas de todos os estados.

$$P_{n_i} = \frac{P_i}{P_0} = \frac{f_i v_i^2}{f_0 v_0^2} = \frac{f_i}{f_0} \left(\frac{v_i}{v_0} \right)^2 = f_{n_i} v_{n_i}^2. \quad (3.21)$$

Tabela 3.1: Estados de energia

Nome	Frequência, f (MHz)	Tensão, v (V)
s_0	192	1,5
s_1	168	1,5
s_2	84	1,5
s_3	84	1,1
s_4	60	1,5
s_5	60	1,1

O gerenciador de energia é acionado a cada $T = 1s$ (intervalo de amostragem) para controlar o estado de energia do processador. Dado que o tempo de transição entre os estados de energia (usando-se escalonamento dinâmico de tensão e frequência) é muito menor que o intervalo T de amostragem do gerenciador de energia, então os custos de chaveamento (tempo e energia) são desprezíveis.

O modelo do processador é um provedor de serviços (PS) ($\mathcal{M}_{PS}(a), b(s,a), c(s,a)$) onde: $\mathcal{M}_{PS}(a)$ é uma cadeia de Markov controlável e estacionária com conjunto de estados $\mathcal{S} = \{s_0, s_1, \dots, s_5\}$; conjunto de controle $\mathcal{A} = \{a_0, a_1, \dots, a_5\}$, onde a_i significa *mudar para* s_i , $i = 0, 1, \dots, 5$. Dado que os custos de transição (tempo e energia) são desprezíveis, as transições

Tabela 3.2: Estados de energia normalizados

Nome	Frequência normalizada, f_n	Tensão normalizada, v_n	Potência normalizada, $f_n v_n^2$
s_0	1	1	1
s_1	0,875	1	0,875
s_2	0,4375	1	0,4375
s_3	0,4375	0,733	0,235
s_4	0,3125	1	0,3125
s_5	0,3125	0,733	0,168

entre os estados de energia são consideradas determinísticas. Assim cada elemento $p_{s_i, s_j}^{PS}(a_l)$ da matriz de transição $\mathbf{P}^{PS}(a)$ é calculado em (3.22); o consumo de energia $c(s, a)$ é a potência normalizada $f_n v_n^2$ do estado s , e a taxa de serviço $b(s, a)$ é a frequência normalizada do estado s , como mostrado na Tabela 3.2.

$$p_{s_i, s_j}^{PS}(a_l) = \begin{cases} 1, & \text{se } s_j = s_l \\ 0, & \text{caso contrário} \end{cases} \quad (3.22)$$

A fila de serviços (FS) $\mathcal{M}_{FS}(a, s, r)$ de comprimento $Q = 1$ tem o conjunto de estados $\mathcal{Q} = \{q_0, q_1\}$, i.e., em q_0 a fila de serviços está vazia, e em q_1 a fila de serviços tem uma requisição pendente.

Considere um requisitante de serviços (RS) ($\mathcal{M}_{RS}, z(r)$) com o conjunto de estados $\mathcal{R} = \{r_0, r_1\}$ com a função $z(r)$ identificando o número de requisições que o requisitante de serviços submete por intervalo de amostragem quando no estado r : $z(r_0) = 0$, $z(r_1) = 1$. O estado r_0 representa um estado inativo do requisitante de serviços, e.g. quando nem o núcleo do sistema operacional está sendo executado e nenhum processo ativo requer o processador. No escalonador do sistema operacional, e.g., essa situação ocorre quando o processo *idle* é escalonado [47].

É importante perceber que os estados de energia s_2 e s_4 possuem as mesmas frequências que s_3 e s_5 respectivamente. Mas, apesar de possuírem as mesmas frequências, eles possuem tensões diferentes. Então, quando usando s_2 ou s_4 , é interessante reduzir a tensão tão cedo quanto possível. O bloco *Quantizador* como mostrado na Fig. 3.2 tem a função $f_Q(u(t))$ em (3.23), onde $1 - b(s_2)$ e $1 - b(s_4)$ são feitos um pouco menores que $1 - b(s_3)$ e $1 - b(s_5)$ respectivamente, para que os estados com menor tensão sejam utilizados logo depois de seus equivalentes de maior tensão. No bloco *Estimar perda de requisições* (Fig. 3.2) é aplicado o Algoritmo 3 com parâmetro $\lambda = 0,999$.

$$a(t) = f_Q(u(t)) = \begin{cases} a_0, & \text{se } u(t) < 0,1250 \\ a_1, & \text{se } 0,1250 \leq u(t) < 0,5620 \\ a_2, & \text{se } 0,5620 \leq u(t) < 0,5625 \\ a_3, & \text{se } 0,5625 \leq u(t) < 0,6870 \\ a_4, & \text{se } 0,6870 \leq u(t) < 0,6875 \\ a_5, & \text{se } u(t) \geq 0,6875 \end{cases} \quad (3.23)$$

Como todos os elementos do *processo* estão especificados, é necessário estimar a função de transferência do processo $G_P(z)$. O experimento realizado para coletar dados foi concebido para o pior caso de carga de trabalho, como mostrado em (3.24), i.e., existe uma nova requisição a cada instante (discreto) de tempo t . A entrada $u(t)$ foi concebida para ser representativa de cada comando a em (3.25) como mostrado na Fig. 3.8.

$$P^{RS} = \begin{matrix} & r_0 & r_1 \\ r_0 & \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \\ r_1 & \end{matrix} \quad (3.24)$$

$$u(t) = \begin{cases} 0,6875, & \text{se } t \leq 10000 \\ 0,6870, & \text{se } 10000 < t \leq 20000 \\ 0,5625, & \text{se } 20000 < t \leq 30000 \\ 0,5620, & \text{se } 30000 < t \leq 40000 \\ 0,1250, & \text{se } 40000 < t \leq 50000 \\ 0, & \text{se } 50000 < t \leq 60000 \\ 0,6875, & \text{se } 60000 < t \leq 70000 \\ 0,6870, & \text{se } 70000 < t \leq 80000 \\ 0,5625, & \text{se } 80000 < t \leq 90000 \\ 0,5620, & \text{se } 90000 < t \leq 100000 \\ 0,1250, & \text{se } 100000 < t \leq 110000 \\ 0, & \text{se } 110000 < t \leq 120000 \end{cases} \quad (3.25)$$

As estruturas de modelo ARX e ARMAX foram usadas para o procedimento de estimação com as ordens dos polinômios em (3.5), (3.6) e (3.7) variando de 1 a 11. O melhor ajuste foi obtido para a estrutura de modelo ARX $A(q)y(t) = B(q)u(t) + e_c(t)$ com $A(q)$ em (3.26) e $B(q)$ em (3.27). A função de transferência do processo $G_P(z)$ é mostrada em (3.28). Usando o projeto de controlador *deadbeat* do Exemplo 6, os parâmetros do controlador são $R(q) = q - 0,999$, $S(q) = 1012,7$, e $T(q) = 1014,7$.

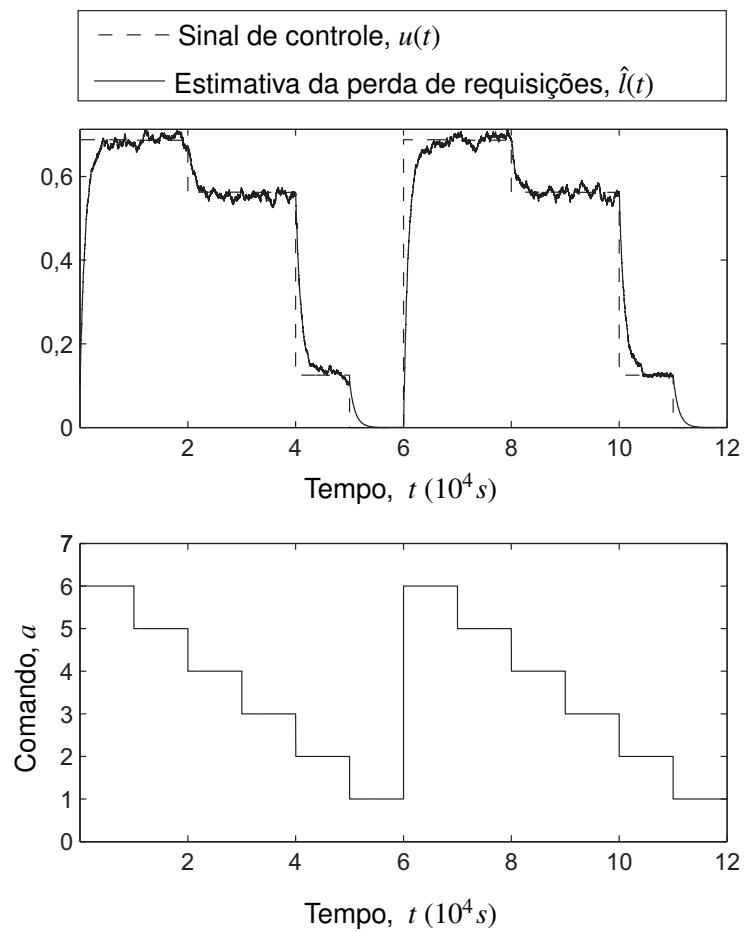


Figura 3.8: Experimento para coletar dados de entrada $u(t)$ e saída $\hat{l}(t)$.

$$A(q) = 1 - 0,999q^{-1} \quad (3.26)$$

$$B(q) = 0,0009855q^{-1} \quad (3.27)$$

$$G_P(z) = \frac{B(z)}{A(z)} = \frac{0,0009855}{z - 0,999} \quad (3.28)$$

$$u(t) = \text{sat}(0,999u(t-1) + 1014,7l_c(t-1) - 1012,7\hat{l}(t-1)) \quad (3.29)$$

A restrição de penalidade de desempenho é: a máxima perda de requisições média de $L = 10\%$. Então o ‘valor de referência’ $l_c(t) = 0,1$. O modelo do sistema desse exemplo foi simulado para diferentes requisitantes de serviços estacionários, com $p_{r_0, r_0}^{RS} = 0, 0,1, 0,2, \dots, 1,0$ e $p_{r_1, r_1}^{RS} = 0, 0,1, 0,2, \dots, 1,0$. Dois indicadores foram calculados para avaliar o desempenho das abordagens de gerenciamento de energia:

- P : potência média
- \hat{L} : perda de requisições média

Usando a abordagem estocástica [10], a potência média e a perda de requisições média são mostrados na Fig. 3.9. Para algumas cargas de trabalho, a perda de requisições é maior que a restrição de penalidade de desempenho, e as políticas otimizadas para essas cargas de trabalho não são aceitáveis, como mostrado na Fig. 3.10.

Usando a técnica proposta de controle em malha fechada para sistemas com gerenciamento de energia e modelados por cadeias de Markov, a potência média e a perda de requisições média são mostrados na Fig. 3.11. Para todas as cargas de trabalho, a perda de requisições é menor do que ou igual à restrição de penalidade de desempenho, como mostrado na Fig. 3.12. Para algumas cargas de trabalho leves, a perda de requisições é bem menor que a restrição de penalidade de desempenho, mas isso não representa uma limitação do controlador. Na realidade, isto ocorre porque para essas cargas de trabalho, a taxa de requisições por intervalo de amostragem não é suficiente para causar uma perda de requisições média igual à restrição de penalidade de desempenho, mesmo estando o provedor de serviço no estado de mais baixa taxa de serviço.

Um exemplo de simulação da técnica proposta de controle em malha fechada para a carga de trabalho $p_{r_0, r_0}^{RS} = 0,5$ e $p_{r_1, r_1}^{RS} = 0,5$ é mostrado na Fig. 3.13, em que são ilustrados os estados do requisitante de serviços, r , a fila de serviços, q , o provedor de serviços, s , a perda de requisições, $l(t)$, a estimativa de perda de requisições, $\hat{l}(t)$, o sinal de controle, $u(t)$ e o comando $a(t)$. Quando $\hat{l}(t) < l_c(t)$, e.g. durante $1 \leq t \leq 17$ na Fig. 3.13, o controlador passa comandos para reduzir a taxa de serviço do provedor de serviços e, assim, economizar energia. Quando ocorre uma perda de requisições, e.g. em $t = 18$ na Fig. 3.13, a estimativa da perda de requisições média

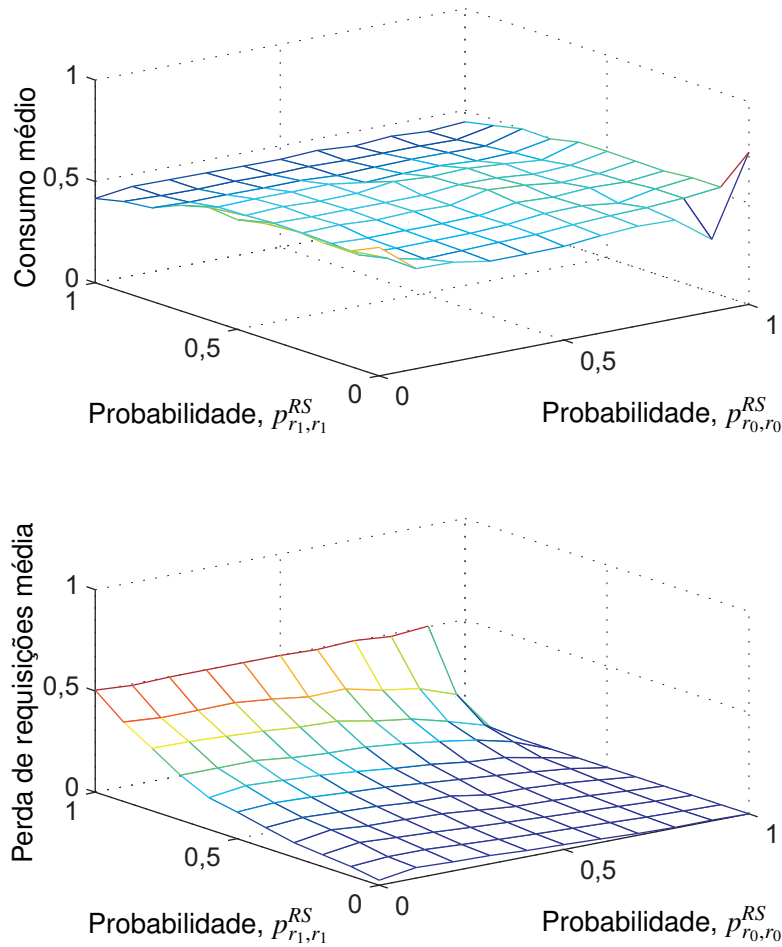


Figura 3.9: Abordagem estocástica: potência média e perdas de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (RS).

$\hat{l}(t)$ aumenta, e o controlador passa comandos para aumentar a taxa de serviço do provedor de serviços.

A diferença entre a potência média da abordagem estocástica [10] e o controle em malha fechada é mostrada na Fig. 3.14. Se considerarmos apenas as políticas estocásticas que satisfazem a restrição de penalidade de desempenho como mostrado na Fig. 3.10, então alguns pontos são removidos do gráfico de diferença da Fig. 3.14, resultando no gráfico da Fig. 3.15. Para quase todas as políticas estocásticas aceitáveis, o controle em malha fechada oferece menor consumo que a abordagem estocástica, exceto para a carga de trabalho $p_{r_0, r_0}^{RS} = 0,5$ e $p_{r_1, r_1}^{RS} = 0,5$, onde o consumo do controle em malha fechada é um pouco maior que a abordagem estocástica. Além disso, como mostrado na Fig. 3.10, por meio da abordagem estocástica não é possível satisfazer a restrição de penalidade de desempenho para cargas de trabalho intensivas e explorar

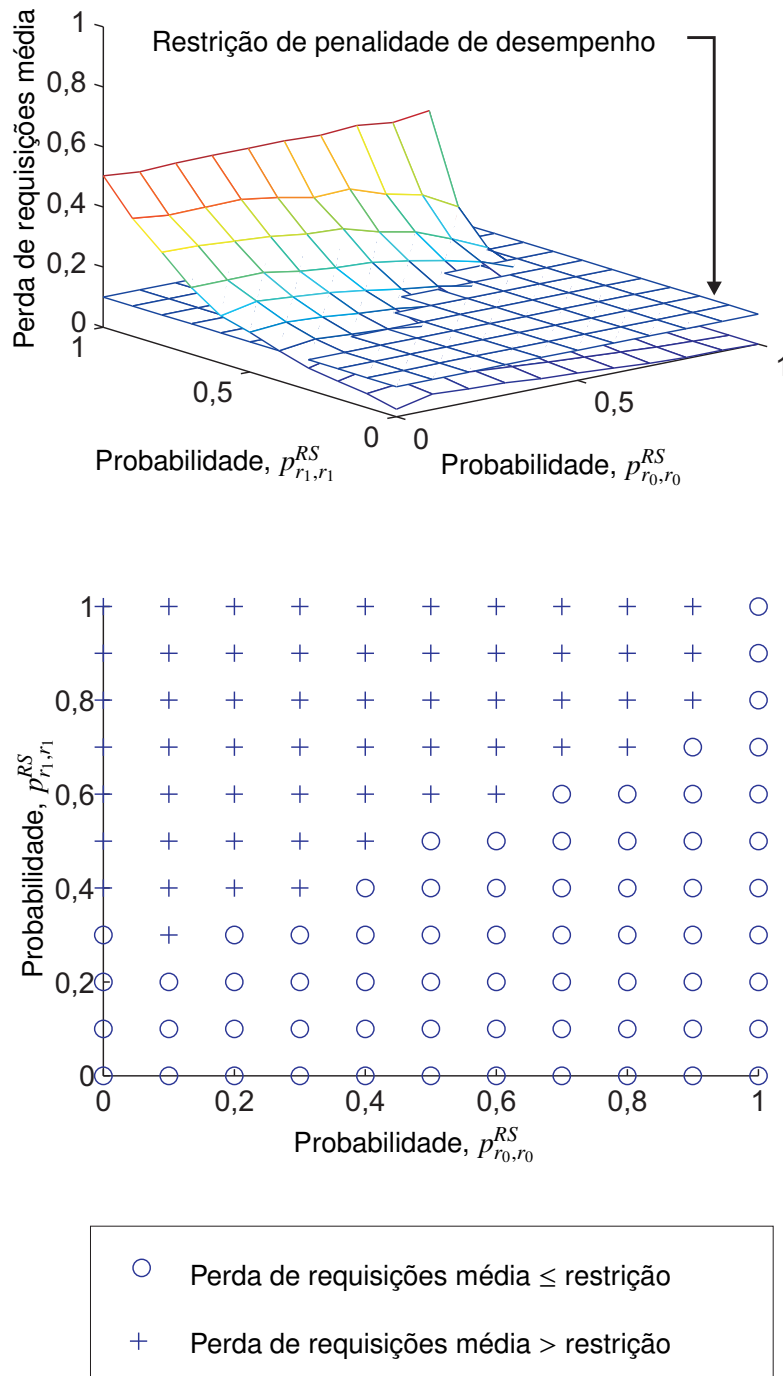


Figura 3.10: Abordagem estocástica: verificação da satisfação da restrição de penalidade de desempenho para as cargas de trabalho leves.

completamente as oportunidades de economia de energia para cargas de trabalho leves. Assim o controle em malha fechada garante que a perda de requisições média do sistema segue a restrição de penalidade de desempenho e explora plenamente a economia de energia.

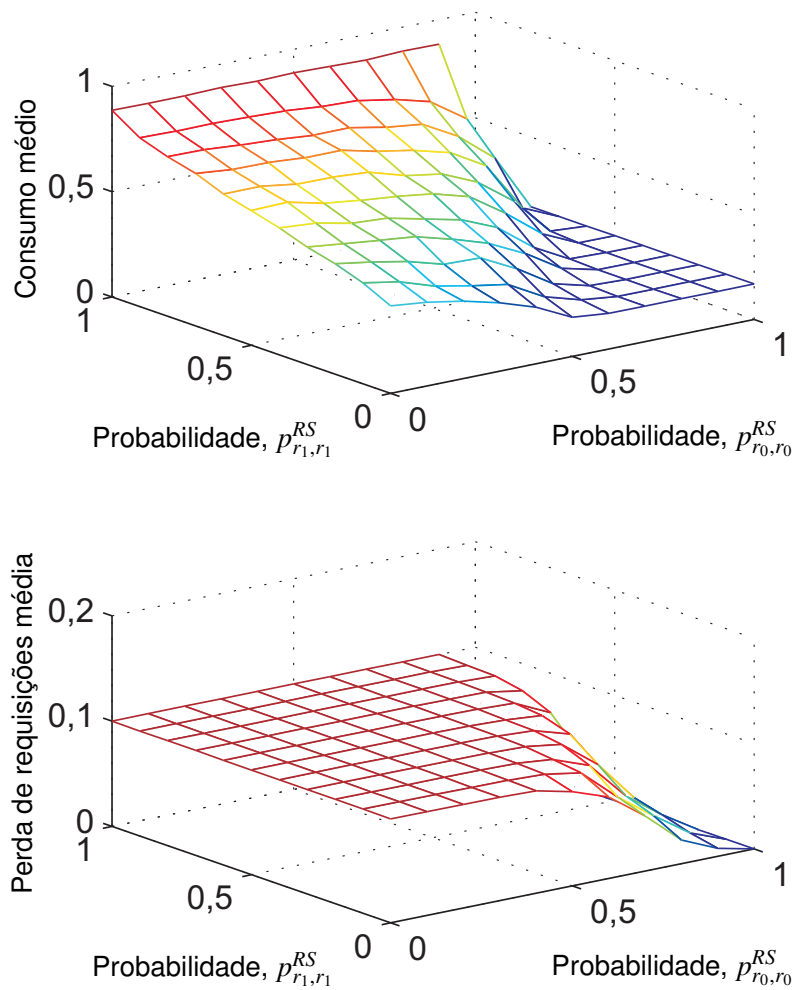


Figura 3.11: Controle em malha fechada: potência média e perda de requisições de um sistema com gerenciamento de energia para diferentes probabilidades de transição do requisitante de serviços (RS).

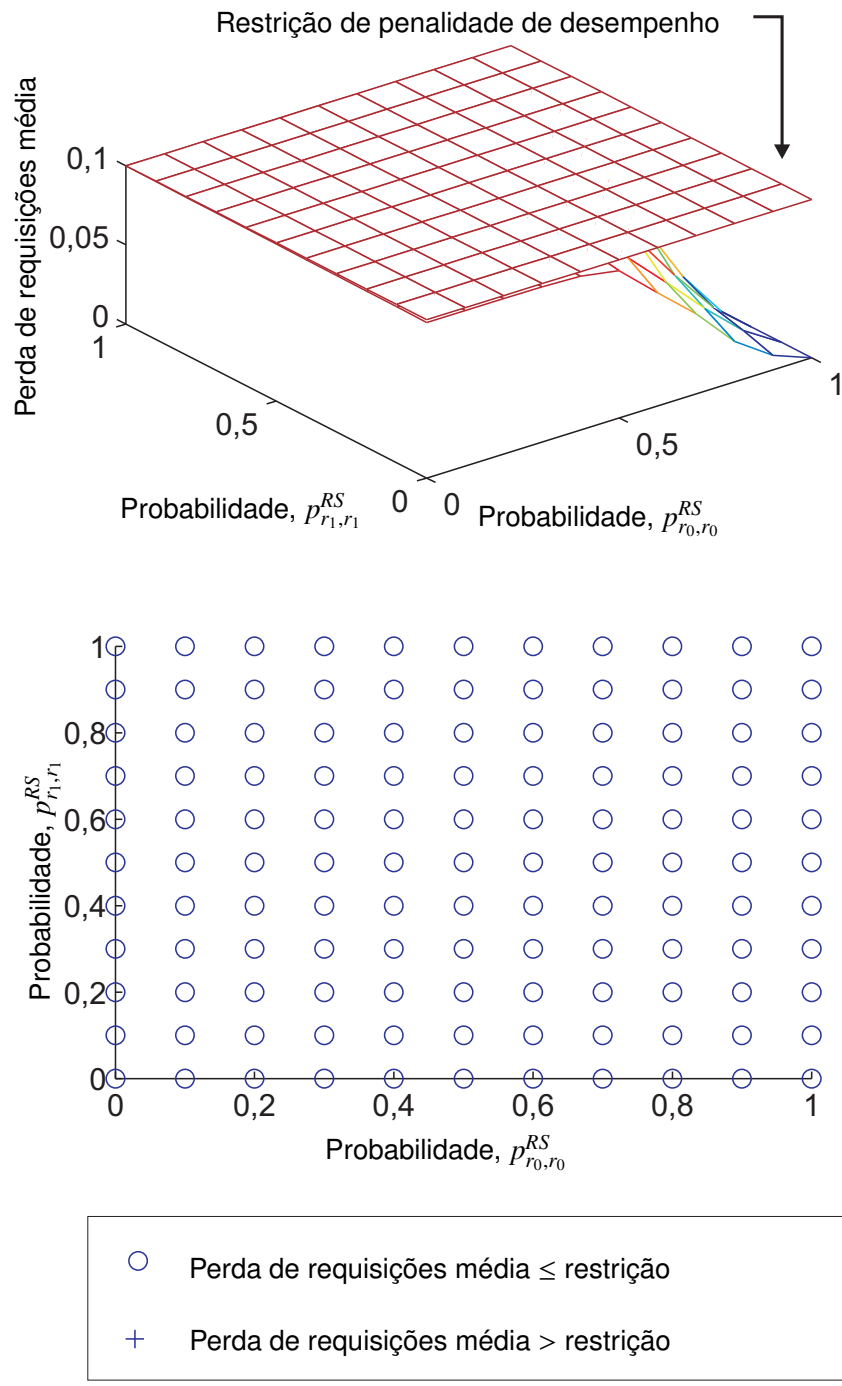


Figura 3.12: Controle em malha fechada: verificação da satisfação da restrição de penalidade de desempenho para todas as cargas de trabalho.

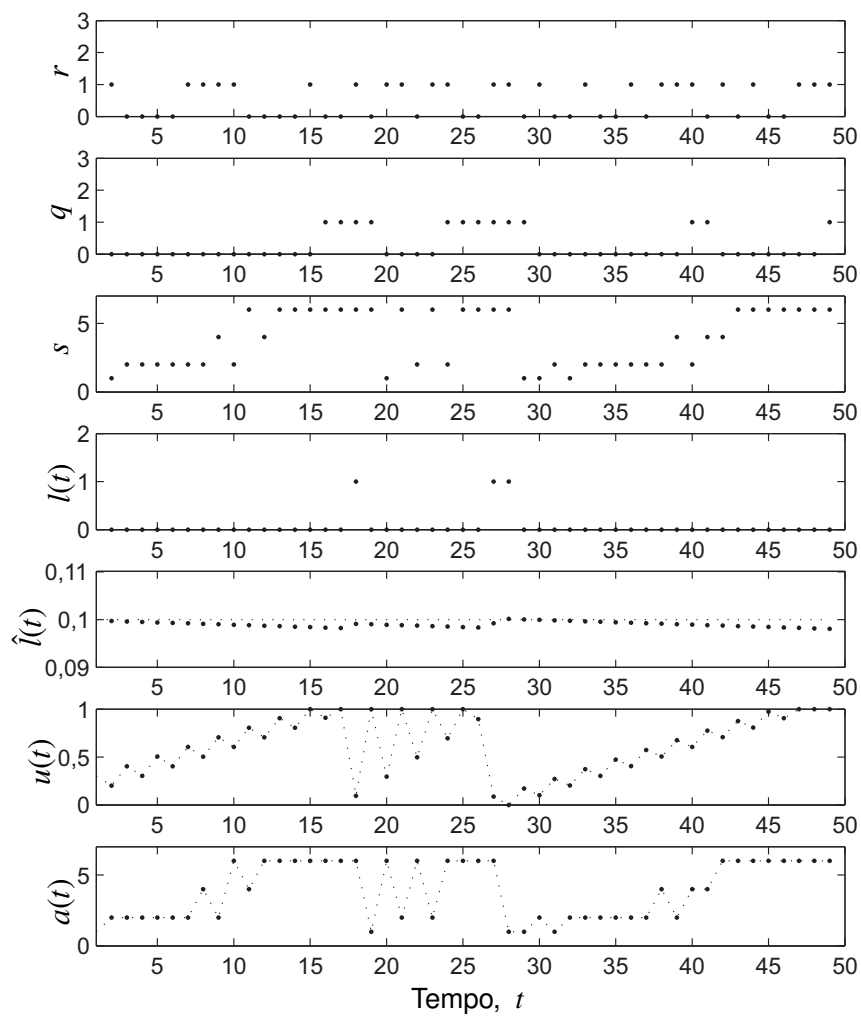


Figura 3.13: Exemplo de simulação para a carga de trabalho $p_{r_0, r_0}^{RS} = 0,5$ e $p_{r_1, r_1}^{RS} = 0,5$.

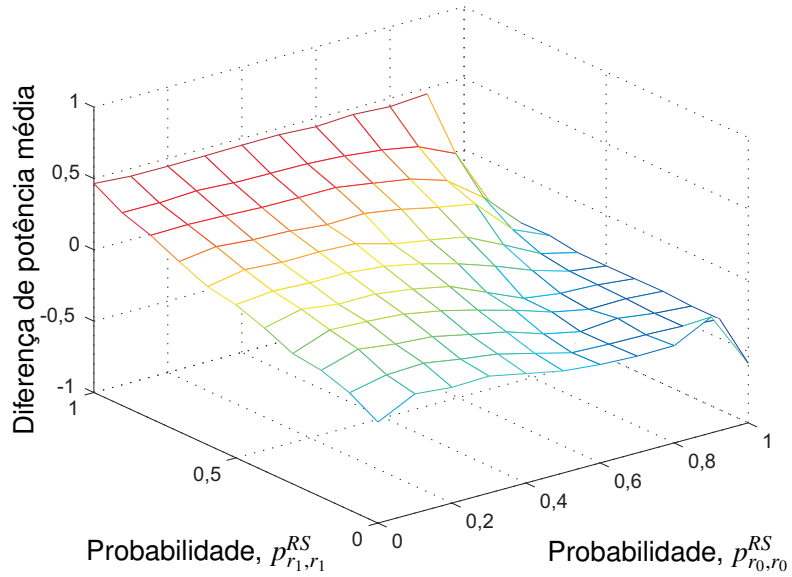


Figura 3.14: Diferença de potência média. Valores negativos representam cargas de trabalho para as quais o controle em malha fechada oferece menor potência média do que a abordagem estocástica.

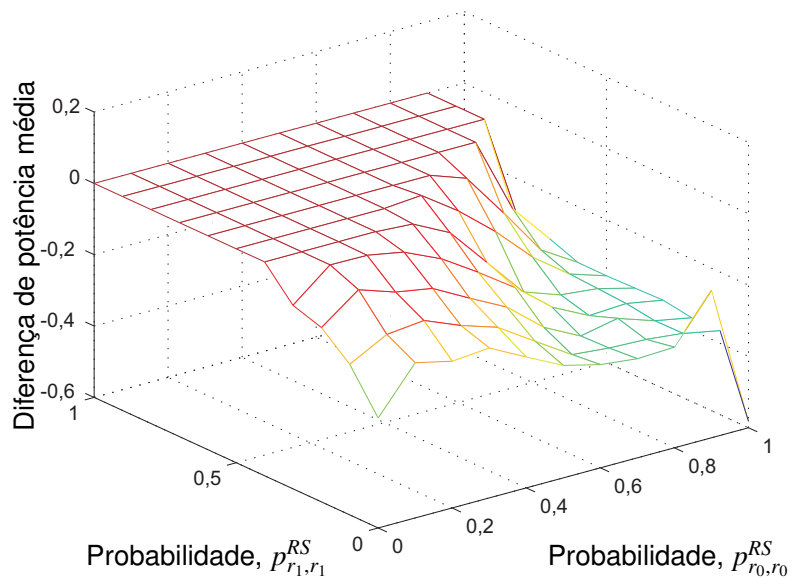


Figura 3.15: Diferença de potência média para as políticas estocásticas aceitáveis. Valores negativos representam cargas de trabalho para as quais o controle em malha fechada oferece menor potência média do que a abordagem estocástica.

3.8 Resumo

O comportamento estocástico de sistemas com gerenciamento de energia pode ser eficientemente modelado por meio de cadeias de Markov controláveis. Usando-se esse modelo, políticas podem ser derivadas a partir da solução de um problema de otimização. Mas a eficiência da abordagem estocástica para gerenciamento de energia depende da identificação acurada da carga de trabalho do sistema. Os resultados do problema de otimização em termos de desempenho e consumo de energia só são garantidos na média. Além disso, a política de gerenciamento de energia utilizando-se o resultado do problema de otimização pode ser considerada de *malha aberta*, ou seja, em tempo de execução os resultados obtidos não são comparados às restrições utilizadas na formulação do problema de otimização. E assim, não se garante que tais restrições serão respeitadas.

Neste capítulo foi derivado um modelo dinâmico de tempo discreto a partir do modelo estocástico do sistema com gerenciamento de energia. A combinação de controle em malha fechada e o modelo em cadeia de Markov do sistema com gerenciamento de energia foi efetivo, resultando em políticas de energia que superaram as abordagens estocásticas anteriores. A integração do modelo dinâmico de tempo discreto do controlador com o sistema foi descrita, e uma metodologia para projetar o controlador foi apresentada.

Por meio do controle em malha fechada para sistemas com gerenciamento de energia, o projeto do gerenciador de energia foi mais simples que as abordagens estocásticas anteriores, e, em tempo de execução, a política de energia obtida toma consideravelmente menos tempo para ser executada, diminuindo assim os custos de energia e tempo para a computação das decisões do gerenciador de energia. Além disso, se uma nova restrição de penalidade de desempenho é desejada em tempo de execução, o gerenciador de energia pode ser ajustado automaticamente. No caso das abordagens estocásticas anteriores, um novo conjunto de políticas ótimas deveria ser computado.

Na técnica apresentada no Capítulo 3, a identificação do modelo do sistema é realizada em tempo de projeto, sendo utilizado o pior caso para a carga de trabalho. Contudo a carga de trabalho de sistemas computacionais é não-estacionária, como apresentado no Capítulo 2, sendo necessária a identificação do sistema em tempo de execução para que os parâmetros do controlador estejam ajustados à carga de trabalho atual. A partir destas considerações, no Capítulo 4 serão apresentados os princípios para a aplicação de controle em malha fechada para o gerenciamento de energia de processadores, oferecendo ênfase: (i) na identificação do modelo do sistema em tempo de execução; e (ii) na aplicação de controle adaptativo para ajustar os parâmetros do controlador.

Capítulo 4

Controle Adaptativo para gerenciamento de energia

4.1 Introdução

No Capítulo 3 foi mostrado que, com a solução para o problema de otimização de política formulado por Benini et al. [10], a restrição de penalidade de desempenho pode ser satisfeita, mas não há garantias de que tal restrição seja totalmente explorada. Foi apresentada então uma política de gerenciamento de energia com controle em malha fechada [19] em que a variável controlada é a restrição de penalidade de desempenho. Um processador com escalonamento dinâmico de tensão e frequência (*DVFS*) foi modelado como um processo de decisão de Markov utilizando a metodologia de Benini et al. [10]. O controlador foi projetado com base num modelo estimado a partir do pior caso da carga de trabalho. No entanto, um processador está tipicamente sujeito a cargas de trabalho inicialmente desconhecidas ou não-estacionárias. Uma conclusão importante é que técnicas de controle em malha fechada para gerenciamento de energia teriam melhor desempenho caso houvesse a adaptação contínua dos parâmetros do controlador.

Assim, neste capítulo é superada a falta de adaptação contínua dos parâmetros do controlador apresentado no Capítulo 3 por meio de uma técnica de controle adaptativo para gerenciamento dinâmico de energia. A técnica apresentada é baseada em modelagem, estimação e controle, que permitem o controle efetivo do desempenho e a minimização do consumo de energia. A principal contribuição apresentada neste capítulo é a aplicação de controle adaptativo para sistemas com gerenciamento de energia e cargas de trabalho variantes no tempo e inicialmente desconhecidas.

Com base nos resultados da identificação, apresentados na Seção 4.2.2.1, e a especificação desejada para o comportamento dinâmico do processo, é possível projetar o controlador, apre-

sentado na Seção 4.2.2.2. Na Seção 4.3, um estudo de caso é apresentado: na Seção 4.3.1, uma plataforma experimental é descrita, e na Seção 4.3.2 a técnica de controle adaptativo é aplicada ao controle da utilização do processador.

4.2 Controle adaptativo para gerenciamento de energia

Em técnicas de controle em malha fechada para gerenciamento de energia, em geral, é aplicado um esquema em que o valor de referência é um valor desejado para o desempenho, como apresentado na Fig. 3.1. O desempenho do sistema é medido e comparado com o valor de referência de desempenho, resultando no erro de desempenho, que é a entrada para um controlador, cuja saída é um comando para mudar o estado de energia do sistema com gerenciamento de energia. O objetivo é garantir que o desempenho do sistema seja igual ao valor de referência de desempenho desejado.

Para o propósito de gerenciamento dinâmico de energia, um processador com um núcleo pode ser representado como o diagrama de blocos mostrado na Fig. 4.1. No restante deste texto, essa estrutura é denominada sistema com gerenciamento de energia e ela está de acordo com o modelo básico para gerenciamento de energia apresentado por Benini et al. [10], sendo composto de:

- um *requisitante de serviços*, que é a fonte de todas as requisições para o processador, i.e., processos do usuário e do sistema operacional, interrupções de *hardware* e de *software*, etc;
- uma *fila de serviços* em que as requisições de serviço são enfileiradas para serem servidas pelo processador imediatamente ou no futuro, como, por exemplo, o escalonador num sistema operacional, onde os processos são enfileirados;
- um *provedor de serviços*, representando o processador.

O escalonador pode remover um processo que está sendo executado no processador e colocá-lo na fila de serviços para ser executado futuramente (veja a seta de menor espessura que sai do processador para a fila na Fig. 4.1). Neste texto considera-se que, no processador, há períodos inativos de tempo, ou seja, no processador há capacidade de processamento suficiente para executar a carga de trabalho, permanecendo a utilização do processador menor que 100%. Então é possível reduzir a frequência de operação de um processador sub-utilizado e assim economizar energia.

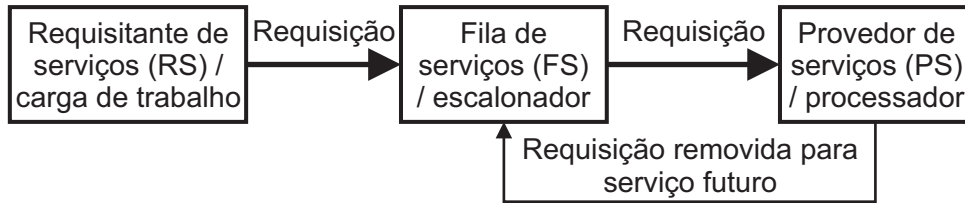


Figura 4.1: Os componentes do sistema com gerenciamento de energia.

Para o gerenciamento de energia de um processador, o controle em malha fechada pode ser aplicado como mostrado na Fig. 4.2. A ‘variável controlada’ $y(t)$ é uma grandeza mensurável do sistema com gerenciamento de energia no instante de tempo t , e.g., temperatura do circuito integrado, consumo de potência, utilização do processador, perda de requisições (e.g. a perda do prazo de uma tarefa de tempo-real), etc. A ‘variável manipulada’ é o estado de energia s do provedor de serviços. O gerenciador de energia aplica um comando $c(t)$ para mudar o estado do provedor de serviços.

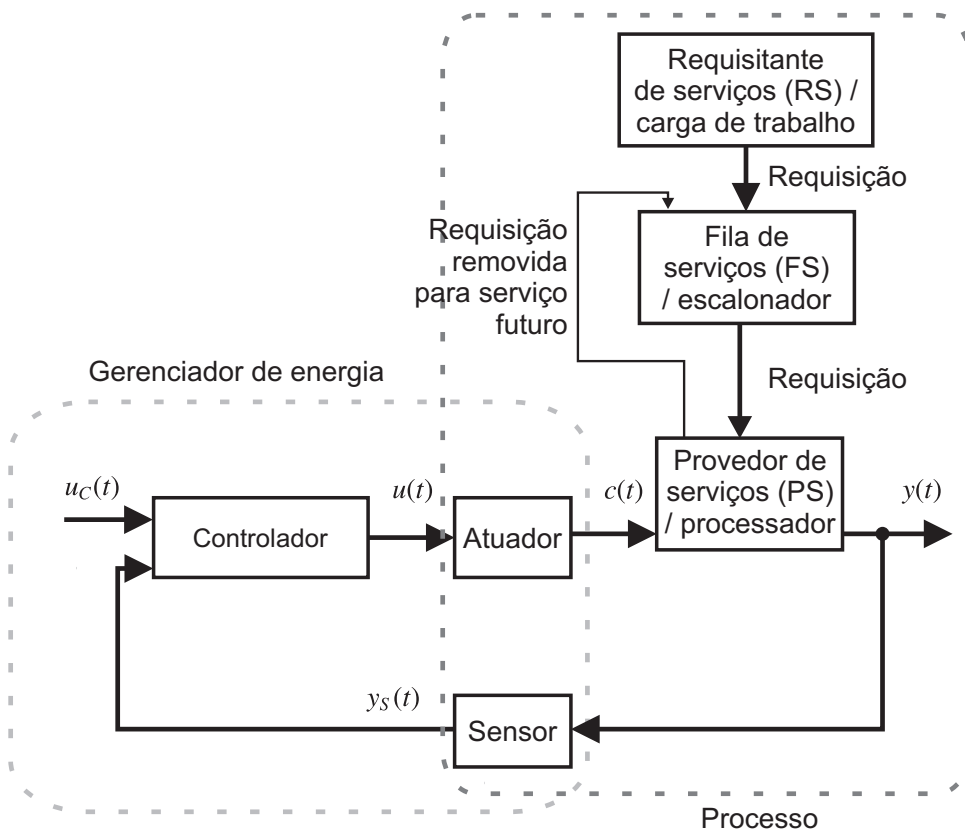


Figura 4.2: Diagrama de blocos do controle em malha fechada para o gerenciamento de energia do processador.

Quando a ‘variável controlada’ $y(t)$ representa uma grandeza física, tal como temperatura, então a aquisição desse sinal requer um sensor, um conversor analógico-digital (A-D), e um *device driver* para tornar os dados disponíveis ao sistema operacional. Quando a ‘variável contro-

lada' $y(t)$ é um indicador do próprio sistema operacional, tal como a utilização do processador, perda de requisições, ou *QoS* (Qualidade do Serviço), então algum *software* é necessário para fornecer os dados apropriados ao sistema operacional. Em ambos os casos, os elementos de *hardware* e *software* necessários para a aquisição da estimativa de $y(t)$ (i.e., $y_S(t)$) são doravante denominados de *Sensor*. A saída $y_S(t)$ do bloco *Sensor* é um sinal quantizado, e assim é necessário avaliar se a resolução do conversor analógico-digital (A-D) ou do elemento de *software* é suficiente para a implementação do gerenciador de energia.

A 'entrada de referência' $u_C(t)$ representa o valor desejado para a 'variável controlada' $y(t)$. O *Controlador* tem como entradas o valor de referência $u_C(t)$ e a estimativa $y_S(t)$, e tem como saída o sinal de controle $u(t)$. Entretanto, o conjunto de comandos C do provedor de serviços (um processador) é discreto e finito, pois este corresponde ao conjunto de estados de energia possíveis por meio do escalonamento dinâmico de tensão e frequência. Então, o bloco *Atuador* é empregado para mapear o sinal de controle contínuo $u(t)$ para um comando discreto $c(t)$, ou modular uma sequência de comandos aplicados em sub-intervalos do período de amostragem, tal como um sinal de modulação por largura de pulso. O comando aplicado corresponde ao estado de energia desejado para o processador. Deve ser escolhido o estado ou a sequência de estados que resultará no menor consumo de energia, ainda garantindo o nível de desempenho desejado. Portanto, o bloco *Atuador* tem um papel fundamental na minimização do consumo de energia. Nesse bloco, há a interface apropriada para o sistema operacional para aplicar um comando $c(t)$ para o processador, e.g., um dos *estados-P*¹ da especificação ACPI (*Advanced Configuration and Power Interface*)² [48]. O bloco *Atuador* é então especificamente projetado para o sistema operacional e o processador para os quais o gerenciador de energia é usado. Por exemplo, no sistema operacional (SO) Linux, o estado-P de um processador Intel é alterado por meio da interface *cpufreq* [49]. Para esse SO, o atuador deve escrever o estado-P desejado num arquivo relacionado à interface *cpufreq* no sistema de arquivos. Como mostrado na Fig. 4.2, as interfaces entre o controlador e o sistema com gerenciamento de energia são a entrada de referência $u_C(t)$, a estimativa da saída $y_S(t)$ e o sinal de controle $u(t)$. Assim, os blocos *requisitante de serviços*, *fila de serviços*, *provedor de serviços*, *atuador* e *sensor* podem ser representados como uma entidade única, denominada de *processo*, como mostrado na Fig. 4.2.

O objetivo do gerenciamento de energia é minimizar o consumo de energia do sistema tal que as requisições de serviço sejam atendidas com um nível desejado de desempenho, i.e., é desejável realizar mais computação com menos energia [50]. O controle em malha fechada tem

¹Em arquiteturas Intel, o *estado-P* é caracterizado por um par de tensão e frequência de operação do processador.

²A ACPI é uma especificação industrial para a realização, por meio do sistema operacional, de gerenciamento de energia e gerenciamento térmico de dispositivos móveis, *desktops* e servidores.

sido aplicado para resolver esse problema em processadores [7, 19, 23, 25–27]. As diversas abordagens de controle em malha fechada para gerenciamento de energia diferem com respeito ao sistema com gerenciamento de energia, as escolhas das variáveis ‘controlada’ e ‘manipulada’, e a ‘entrada de referência’. Para cada uma dessas escolhas, o projeto é realizado de acordo com o sistema com gerenciamento de energia. Por exemplo, para um sistema com fortes restrições de consumo de energia [7], a variável controlada é a potência, a variável manipulada é o estado de energia do sistema, e a entrada de referência é o ‘orçamento de potência’, isto é, o consumo de energia máximo desejável do sistema com gerenciamento de energia. Para sistemas com fortes restrições de desempenho, a variável controlada é algum indicador de desempenho, e.g., o tempo de execução de uma tarefa num sistema em tempo-real [26], ou o número de quadros de vídeo esperando em uma fila para serem decodificados por um reprodutor de vídeo [27]. A variável manipulada também é o estado de energia do sistema, e a entrada de referência é o valor desejado para o indicador de desempenho medido.

4.2.1 O modelo discreto do sistema

Para projetar o controlador, é necessário algum conhecimento sobre o modelo do sistema com gerenciamento de energia. Contudo, devido às características do sistema com gerenciamento de energia, talvez não seja possível derivar um modelo dinâmico simples para representar o sistema. Com base nos resultados do Capítulo 3, são aplicadas no presente capítulo técnicas de identificação de sistemas [44] para determinar um modelo aproximado e representar a dinâmica do sistema com gerenciamento de energia. O modelo do sistema de entrada única e saída única (*SISO*) em (4.1) é adequado para projetar o controlador do sistema com gerenciamento de energia representado na Fig. 4.2. Os polinômios $A(q)$ e $B(q)$ são representados em (4.2) e (4.3), e $v(t)$ denota um sinal de perturbação.

$$A(q)y_S(t) = B(q)[u(t) + v(t)] \quad (4.1)$$

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (4.2)$$

$$B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (4.3)$$

Um processador está tipicamente sujeito a cargas de trabalho inicialmente desconhecidas ou não-estacionárias. Então o modelo do sistema com entrada única e saída única (*SISO*) em (4.1) também apresenta essas características. No Capítulo 3, foi considerado que as técnicas baseadas em controle em malha fechada para gerenciamento de energia poderiam ser beneficiadas com

a adaptação contínua dos parâmetros do controlador. Então, o bloco *Controlador* mostrado na Fig. 4.2 será implementado como um regulador auto-sintonizável, com base na teoria de Controle Adaptativo, como apresentado na Seção 4.2.2.

4.2.2 O regulador auto-sintonizável

O controlador mostrado na Fig. 4.2, doravante denominado *regulador auto-sintonizável*, é composto dos blocos representados na Fig. 4.3: o bloco *Estimação* é um estimador recursivo cujas saídas são estimativas dos parâmetros do modelo do processo, e.g. os polinômios $A(q)$ e $B(q)$ do modelo do sistema de entrada única e saída única (SISO) em (4.1); esses *parâmetros do processo* e a *especificação* de malha fechada para o comportamento do sistema são as entradas do bloco *projeto do controlador*, em que o projeto é realizado; o *controlador* é uma função dos *parâmetros do controlador*. O diagrama de blocos mostrado na Fig. 4.3 é bem conhecido e utilizado em vários problemas de controle [45]. Então, a principal contribuição do atual capítulo é a aplicação da técnica de controle adaptativo para sistemas com gerenciamento de energia com cargas de trabalho inicialmente desconhecidas e não-estacionárias. No restante desta seção, os blocos *Estimação*, *Projeto do controlador* e *Controlador* são descritos.

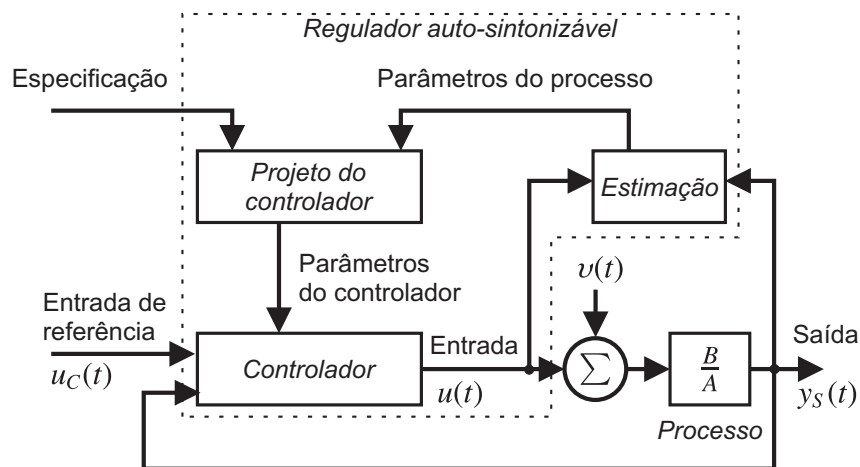


Figura 4.3: Diagrama de blocos do regulador auto-sintonizável.

4.2.2.1 A estimação dos parâmetros do modelo do processo

Nesta seção, é descrito o bloco *Estimação* representado na Fig. 4.3. Se os parâmetros do processo mudam ao longo do tempo devido à carga de trabalho não-estacionária, então, é necessário realizar a estimação recursiva dos parâmetros do modelo do processo.

A relação mais básica entre a entrada $u(t)$ e a saída $y_S(t)$ é a *equação de diferenças linear* em (4.4).

$$A(q)y_S(t) = B(q)u(t) \quad (4.4)$$

A partir de (4.4), o próximo valor da saída é:

$$y_S(t) = (1 - A(q))y_S(t) + B(q)u(t) \quad (4.5)$$

De acordo com (4.2), (4.3) e (4.5), seja θ e $\varphi(t)$ definidos em (4.6) e (4.7) respectivamente.

$$\theta = \left[a_1 \quad \dots \quad a_{n_a} \quad b_1 \quad \dots \quad b_{n_b} \right]^T \quad (4.6)$$

$$\varphi(t) = \left[-y_S(t-1) \quad \dots \quad -y_S(t-n_a) \quad u(t-1) \quad \dots \quad u(t-n_b) \right]^T \quad (4.7)$$

Para enfatizar que a estimativa da saída $y_S(t)$ depende das entradas passadas, saídas passadas e também de θ , ela é denotada como $\hat{y}_S(t|\theta)$, e então (4.5) é escrita como:

$$\hat{y}_S(t|\theta) = \varphi^T(t)\theta \quad (4.8)$$

Para o problema de gerenciamento de energia, é importante considerar que o modelo em (4.4) do sistema com gerenciamento de energia é variante no tempo. Então, é necessário identificar esse modelo recursivamente em tempo de execução, após cada novo período de amostragem. Uma possível técnica para realizar essa tarefa é o algoritmo de *mínimos quadrados recursivo* (MQR) [44], para o qual são aplicadas as expressões (4.9), onde I é a matriz identidade e o parâmetro $\lambda(t)$ é associado a uma sequência de pesos $\beta(t,k)$ em (4.10). É necessário calcular as matrizes $L(t)$ e $P(t)$ antes de calcular a estimativa dos parâmetros $\hat{\theta}(t)$.

$$\begin{aligned} L(t) &= \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \\ P(t) &= \frac{1}{\lambda(t)} \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \right] \\ &= \frac{1}{\lambda(t)} \left[I - L(t)\varphi^T(t) \right] P(t-1) \\ \varepsilon(t) &= y_S(t) - \varphi^T(t)\hat{\theta}(t-1) \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + L(t)\varepsilon(t) \end{aligned} \quad (4.9)$$

$$\beta(t,k) = \lambda(t)\beta(t-1,k), \quad 0 \leq k \leq t-1 \quad (4.10)$$

$$\beta(t,t) = 1$$

4.2.2.2 O projeto do controlador

Como o processo pode ser descrito pelo modelo de sistema de entrada única e saída única (*SISO*) em (4.1), é possível aplicar a alocação de pólos de grau mínimo (*MDPP*) [45] para projetar o controlador linear geral em (4.11), onde $R(q)$, $S(q)$ e $T(q)$ são polinômios. O controlador e o processo são mostrados na Fig. 4.4.

$$R(q)u(t) = T(q)u_C(t) - S(q)y_S(t) \quad (4.11)$$

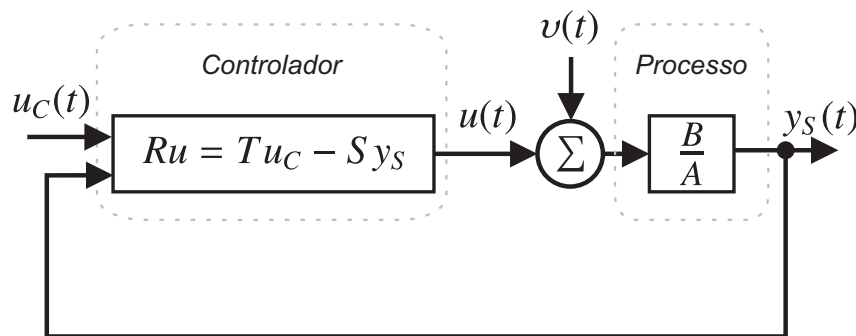


Figura 4.4: Diagrama de blocos do modelo discreto do controlador e do sistema com gerenciamento de energia.

Isolando $u(t)$ em (4.1) e o repondo em (4.11), é obtida a expressão para o sistema em malha fechada em (4.12), onde o operador q foi omitido nos polinômios $A(q)$, $B(q)$, $R(q)$, $S(q)$ e $T(q)$. Então, o polinômio característico de malha fechada é (4.13). A_c é um parâmetro de projeto e é escolhido de tal forma que ofereça as propriedades desejadas do sistema em malha fechada. Então, os polinômios R e S são resolvidos de (4.13), que é denominada a *equação de Diofanto*. Outro parâmetro de projeto é a resposta desejada ao sinal de controle como representado pela dinâmica em (4.14). Então a condição em (4.15) deve ser válida. O seguidor de modelo é usado para determinar o polinômio T . A estimação dos parâmetros do processo e o projeto do controlador são mostrados na Fig. 4.5.

$$y_S(t) = \frac{BT}{RA + BS}u_C(t) + \frac{RB}{RA + BS}v(t) \quad (4.12)$$

$$RA + BS = A_c \quad (4.13)$$

$$A_m y_{S,m}(t) = B_m u_C(t) \quad (4.14)$$

$$\frac{BT}{RA + BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} \quad (4.15)$$

O bloco *Atuador* mostrado na Fig. 4.2 é não-linear porque o conjunto de estados de energia possíveis devido ao escalonamento dinâmico de tensão e frequência é discreto e finito, tendo um

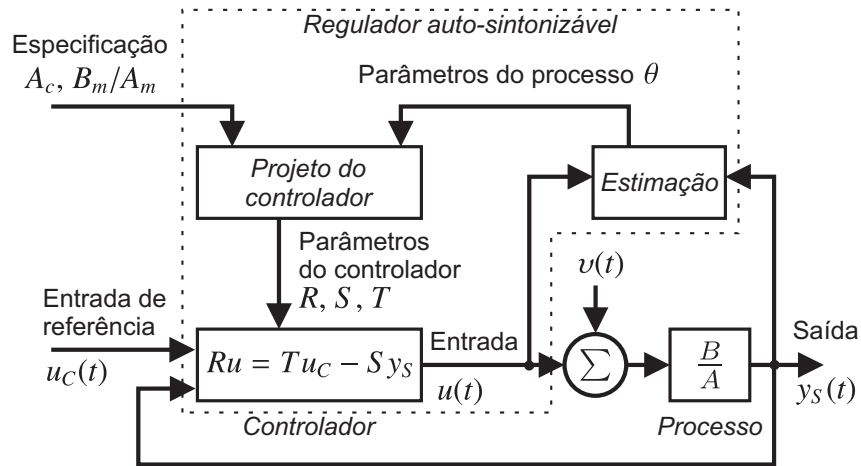


Figura 4.5: A estimação dos parâmetros do processo e o projeto do controlador.

estado de energia mínimo e um máximo. Essa característica de saturação é expressa em (4.16), onde u_{min} e u_{max} correspondem aos limites do atuador. Então, para o projeto do controlador, os efeitos do bloco *Atuador* devem ser considerados por meio da compensação *antiwindup* [46], como mostrado em (4.17) e na Fig. 4.6. Para um observador *deadbeat*, A_{aw} é igual a 1, e (4.17) é reescrito como (4.18).

$$sat(u) = \begin{cases} u_{min} & \text{se } u \leq u_{min} \\ u & \text{se } u_{min} < u < u_{max} \\ u_{max} & \text{se } u \geq u_{max} \end{cases} \quad (4.16)$$

$$A_{aw}\psi = Tuc - Sys + (A_{aw} - R)u$$

$$u = sat(\psi) \quad (4.17)$$

$$u(t) = sat(Tuc(t) - Sys(t) + (1 - R)u(t)) \quad (4.18)$$

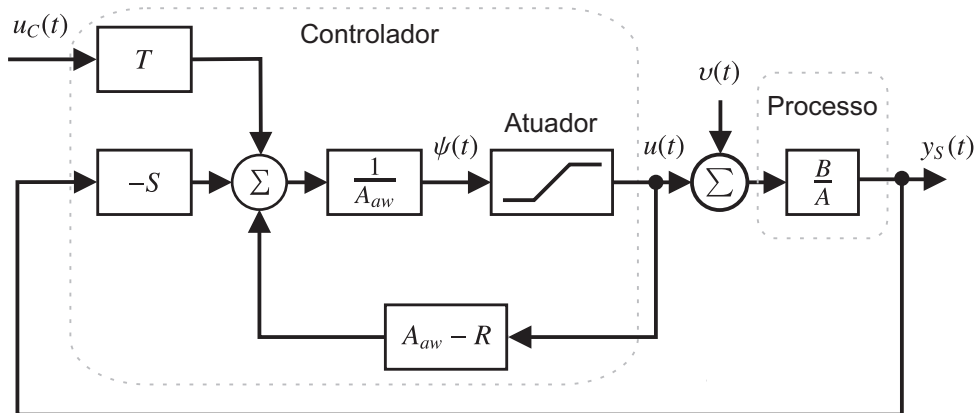


Figura 4.6: Diagrama de blocos do controlador e o processo considerando o atuador não-linear.

Um controlador proporcional-integral [46] também pode ser aplicado ao regulador auto-sintonizável. A entrada para o controlador é o erro $e(t) = uc(t) - ys(t)$. A parte proporcional é

$P(t) = Ke(t)$, em que K é o ganho proporcional. A parte integral é $I(t+1) = I(t) + \frac{Kh}{T_i}e(t)$, onde T_i é o tempo de integração, e h é o período de amostragem. A ação de controle proporcional e integral é $u(t) = P(t) + I(t)$. Considerando que as transformadas Z das partes proporcional e integral são $P(z) = KE(z)$ e $I(z) = \frac{Kh}{T_i} \frac{E(z)}{z-1}$, a transformada Z da ação de controle proporcional-integral é derivada em (4.19). Então a função de transferência $G_C(z)$ do controlador proporcional-integral é apresentada em (4.20). Considerando a função de transferência $G_P(z)$ do processo, a função de transferência do sistema em malha fechada é apresentada em (4.21). Os parâmetros do controlador proporcional-integral podem ser escolhidos com base no princípio de alocação de pólos [46].

$$U(z) = P(z) + I(z) = KE(z) + \frac{Kh}{T_i} \frac{E(z)}{z-1} \quad (4.19)$$

$$G_C(z) = \frac{U(z)}{E(z)} = K \frac{T_i z + (h - T_i)}{T_i(z-1)} \quad (4.20)$$

$$G(z) = \frac{Y_S(z)}{U_C(z)} = \frac{G_C(z)G_P(z)}{1 + G_C(z)G_P(z)} \quad (4.21)$$

Para o projeto do controlador proporcional-integral, os efeitos do bloco *Atuador* devem ser considerados por meio da compensação *antiwindup* [46], como apresentado na Fig. 4.7 e no Algoritmo 4. Uma malha extra de realimentação é incluída para medir o sinal de erro $e_S(t)$, igual à diferença entre o sinal saturado $u(t)$ (calculado por meio da característica de saturação em (4.16)) e o sinal auxiliar $\psi(t)$. O erro $e_S(t)$ é multiplicado pelo ganho $1/T_i$ e adicionado à entrada do integrador. Duas situações podem ocorrer: (i) se o atuador não está saturado, o erro $e_S(t)$ é zero; (ii) se o atuador está saturado, o erro $e_S(t)$ não é zero, e a malha extra de realimentação reinicializa o integrador com a constante de tempo T_i , resultando no sinal de controle igual ao limite de saturação.

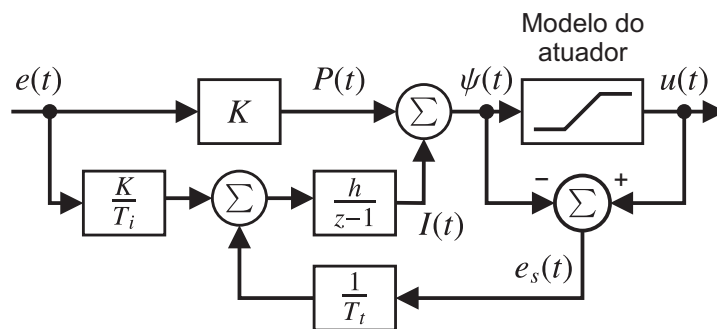


Figura 4.7: Diagrama de blocos do controlador proporcional-integral considerando o atuador não-linear.

Na Seção 4.3 um estudo de caso será considerado. Os blocos sensor, atuador e controlador representados na Fig. 4.2 serão especializados para a variável controlada escolhida. Então, a técnica de controle adaptativo para gerenciamento de energia de processadores será comparada a outras técnicas.

Algoritmo 4: Controlador proporcional-integral considerando o atuador não-linear.

Dados: $P(t), I(t), \psi(t), e_S(t)$

Entrada: $e(t)$

Saída: $u(t)$

$$P(t) = Ke(t) ;$$

$$\psi(t) = P(t) + I(t) ;$$

$$u(t) = \text{sat}(\psi(t)) ;$$

$$e_S(t) = u(t) - \psi(t) ;$$

$$I(t+1) = I(t) + \frac{Kh}{T_i}e(t) + \frac{h}{T_i}e_S(t) ;$$

4.3 Resultados experimentais

Nesta seção, são descritos inicialmente os recursos de *hardware* e *software* para gerenciamento de energia de uma plataforma experimental. Um estudo de caso é apresentado com a ‘utilização do processador’ como a variável controlada. Posteriormente, são apresentados resultados experimentais para comparar a técnica de controle adaptativo a outras técnicas.

4.3.1 Plataforma experimental

O estudo de caso apresentado na Seção 4.3.2 foi implementado na linguagem de programação C e executado num *notebook* com um processador Intel Pentium M 750 com o conjunto de estados de energia $S = \{s_0, s_1, s_2, s_3, s_4\} = \{1,867GHz, 1,600GHz, 1,333GHz, 1,067GHz, 0,800GHz\}$. É associado ao conjunto S um conjunto de comandos $C = \{c_0, c_1, c_2, c_3, c_4\} = \{1867000, 1600000, 1333000, 1067000, 800000\}$, onde c_j significa o comando “realize a transição para o estado s_j ”. O sistema operacional foi a distribuição Ubuntu 10.10 com a versão 2.6.35-22-generic (i686) do núcleo do Linux. Essa plataforma experimental tem escalonamento dinâmico de tensão e frequência, e estados de sono profundo (*deep sleep*), implementados de acordo com o padrão ACPI. Então, o conjunto de estados de energia é o conjunto de *estados-P* (do ACPI [51]) do processador Intel Pentium M 750. O sistema operacional Linux tem um *driver* ACPI por meio do qual é possível trocar o estado de energia do processador quando um comando é aplicado a partir de um gerenciador de energia executando no núcleo do sistema operacional ou uma aplicação em espaço de usuário. Para a plataforma experimental usada neste estudo de caso, o tempo de transição entre *estados-P* é igual a $110000ns$, e o período de amostragem h do gerenciador de energia deve ser escolhido de tal forma que o tempo de transição entre *estados-P* possa ser desprezado quando comparado ao período de amostragem h do gerenciador de energia. Se essa condição é satisfeita, então os custos de transição (energia e tempo para a transição entre os *estados-P*) podem ser desprezados.

Alguns dados importantes para o gerenciamento de energia são disponíveis em sistemas de arquivo que são atualizados em tempo de execução pelo núcleo do Linux:

- *estatísticas do processador* [24] (o tempo durante o qual o processador executa diversos tipos de processos, serve interrupções, etc.): `/proc/stat/`
- *interface cpufreq* [49] (ler e escrever a frequência do processador): `/sys/devices/system/cpu/cpu0/cpufreq`. Um dos comandos do conjunto C é escrito no arquivo `scaling_setspeed` para mudar o estado de energia do processador.

A política de energia apresentada na Seção 4.2 foi implementada como uma aplicação em espaço de usuário, i.e., o *gerenciador de energia* em que o arquivo relacionado às *estatísticas do processador* é lido, a política de gerenciamento de energia é executada, e as políticas escolhidas para o processador são escritas na interface *cpufreq*, como mostrado no diagrama de blocos³ da Fig. 4.8, e no fluxograma da Fig. 4.9.

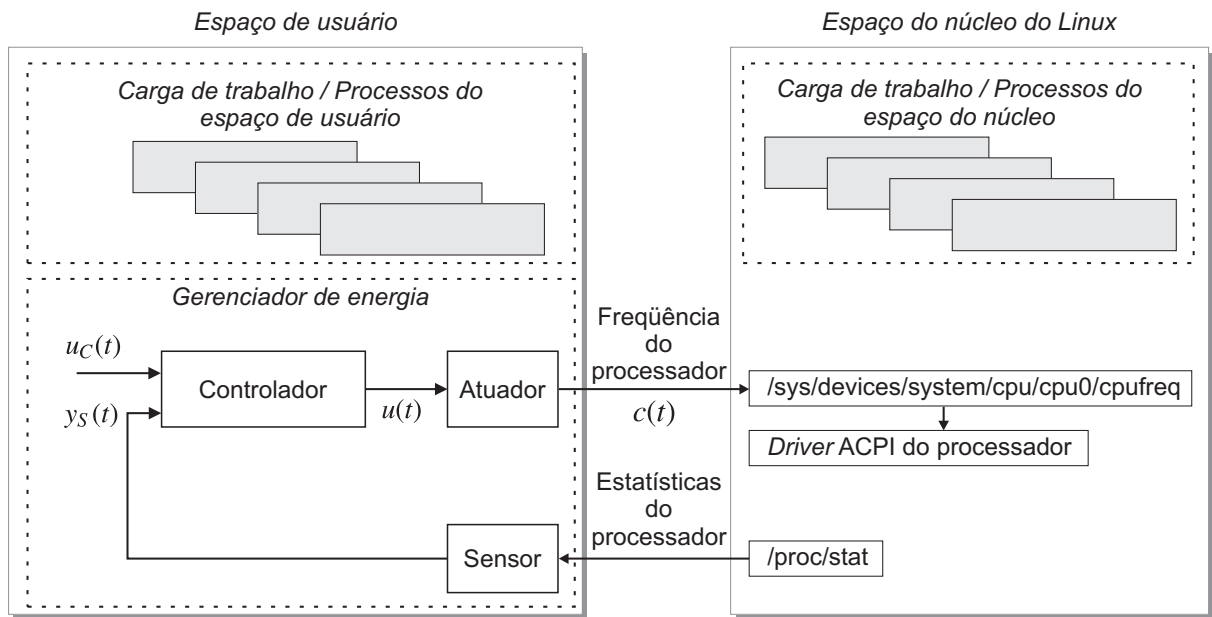


Figura 4.8: Esquema para implementação da técnica de controle adaptativo para gerenciamento de energia.

Um estudo de caso é apresentado na Seção 4.3.2, focando no controle de utilização do processador. Os blocos sensor, atuador e controlador na Fig. 4.2 são detalhados, considerando as características da plataforma experimental. Várias políticas de energia são então comparadas.

³O arquivo `/proc/stat` contém dados relacionados à utilização do processador. Esse arquivo é lido pelo gerenciador de energia, e a frequência escolhida para o processador é escrita na interface `cpufreq`.

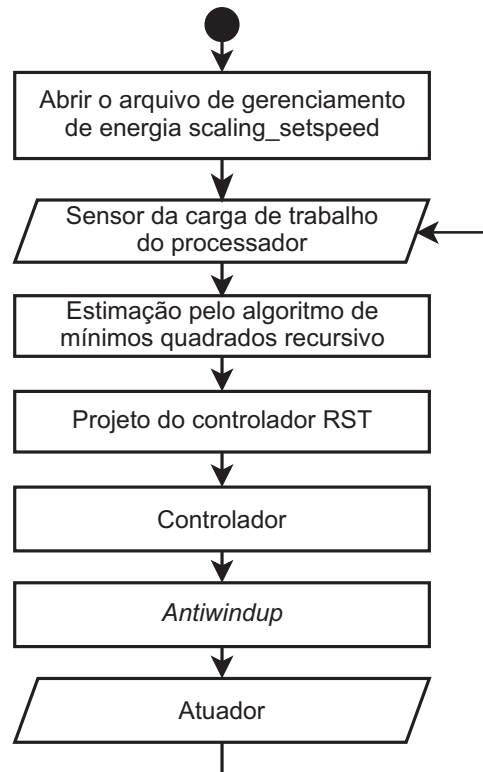


Figura 4.9: Fluxograma da técnica de controle adaptativo para gerenciamento de energia.

4.3.2 O controle de utilização do processador

Nesse estudo de caso, a abordagem de controle adaptativo é aplicada à plataforma experimental apresentada na Seção 4.3.1, para a qual a variável controlada é a utilização do processador, isto é, a fração de tempo durante o qual o processador está ativo, executando instruções. O controle de utilização do processador é importante em alguns sistemas de computação. Por exemplo, administradores de um sistema com três servidores HTTP Apache desejam que a utilização do processador de cada servidor não seja maior que 66%, para que, se algum servidor falhar, os outros dois possam servir toda a carga de trabalho [52]. Num sistema de computação de propósito geral, deseja-se que a utilização do processador seja menor que 100%, pois, se surgem novas requisições da carga de trabalho quando a utilização do processador é igual a 100%, essas requisições não serão servidas imediatamente.

Inicialmente é necessário analisar a relação entre a ‘variável controlada’ $y(t)$ (a utilização do processador) e o sinal de controle $u(t)$, que afeta a frequência de operação do processador. O tempo de execução δ de uma tarefa pode ser subdividido em dois subintervalos [53]: (i) *on-chip*, δ^{on} , quando as instruções são executadas na CPU, e (ii) *off-chip*, δ^{off} , quando a CPU aguarda o final da execução de transações com a memória externa. O subintervalo δ^{off} não depende da frequência do processador. Mas o subintervalo δ^{on} é inversamente proporcional à frequência. Na Fig. 4.10, consideremos que, sendo a frequência do processador a máxima frequência s_0

no conjunto \mathcal{S} de frequências do processador, o subintervalo *on-chip* é $\delta^{on}(s_0)$ para uma tarefa que é executada durante o período de amostragem h ; então, se a frequência fosse $s < s_0$, o subintervalo *on-chip* seria $\delta^{on}(s) = \frac{s_0}{s}\delta^{on}(s_0)$. Logo, o tempo de execução δ de uma tarefa é $\frac{s_0}{s}\delta^{on}(s_0) + \delta^{off}$.

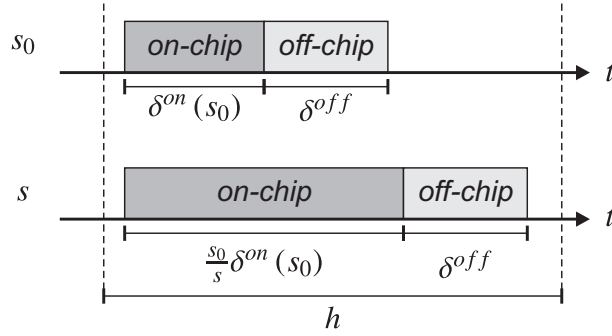


Figura 4.10: Efeito da redução da frequência no tempo de execução de uma tarefa.

Se N múltiplas tarefas $i = 0, 1, \dots, N-1$, são executadas durante o período de amostragem h , cada uma durante um intervalo δ_i , então o período de tempo Δ durante o qual o processador está ativo é apresentado em (4.22). Seja $\Delta^{on}(s_0) = \sum_{i=0}^{N-1} \delta_i^{on}(s_0)$ e $\Delta^{off} = \sum_{i=0}^{N-1} \delta_i^{off}$. Então o tempo de execução Δ das múltiplas tarefas durante o período de amostragem h é $\frac{s_0}{s}\Delta^{on}(s_0) + \Delta^{off}$. A utilização do processador y é igual a $\frac{\Delta}{h}$. Seja $y_{on} = \frac{\Delta^{on}(s_0)}{h}$ e $y_{off} = \frac{\Delta^{off}}{h}$. Então a utilização do processador y é dada em (4.23). Seja o sinal de controle u igual a $\frac{s_0}{s}$. Então, o sinal de controle u afetaria a variável controlada y (a utilização do processador) de acordo com a relação em (4.24).

$$\Delta = \sum_{i=0}^{N-1} \delta_i = \frac{s_0}{s} \sum_{i=0}^{N-1} \delta_i^{on}(s_0) + \sum_{i=0}^{N-1} \delta_i^{off} \quad (4.22)$$

$$y = \frac{\Delta}{h} = \frac{s_0}{s} y^{on} + y^{off} \quad (4.23)$$

$$y = u y^{on} + y^{off} \quad (4.24)$$

Contudo, para cargas de trabalho variantes no tempo, o mesmo conjunto de tarefas pode não se repetir ao longo dos períodos de amostragem, i.e., os parâmetros y^{on} e y^{off} são variantes no tempo. Então, no presente estudo de caso, o sistema de primeira ordem de entrada única e saída única (*SISO*) em (4.25) foi escolhido como o modelo do processo, para incluir tanto a relação em (4.23) quanto a dinâmica do sistema com gerenciamento de energia, como apresentado na Fig. 4.2. Por exemplo, quando uma nova frequência é escolhida devido ao sinal de controle $u(t)$, os tempos de execução das tarefas mudam, mas a utilização do processador medida $y_S(t)$ pode não mudar instantaneamente, porque os novos tempos de execução das tarefas podem influenciar a ordem em que as tarefas são executadas [54]. Então os parâmetros a_1 e b_1 serão estimados por meio do algoritmo de mínimos quadrados recursivos. A função de transferência

pulsada $G_P(z)$ [46] do processo em (4.25) é representada em (4.26).

$$(1 + a_1 q^{-1}) y_S(t) = b_1 q^{-1} u(t) \quad (4.25)$$

$$G_P(z) = \frac{Y_S(z)}{U(z)} = \frac{b_1}{z + a_1} \quad (4.26)$$

Os objetivos de gerenciamento de energia são controlar a utilização do processador $y_S(t)$ e minimizar o consumo de energia. A frequência s é escolhida tal que $\frac{s_0}{s}$ é igual ao sinal de controle $u(t)$ porque: (i) se $y_S(t)$ é menor que $u_C(t)$, há oportunidades para economizar energia, então $u(t)$ deve ser aumentado, e conseqüentemente um menor valor de frequência s é escolhido; (ii) se $y_S(t)$ é maior que $u_C(t)$, é necessário aumentar o desempenho, logo $u(t)$ deve ser reduzido, e conseqüentemente um maior valor de frequência s é escolhido.

O modelo em (4.25) é determinístico, enquanto em grande parte dos sistemas computacionais está presente uma intensa componente estocástica no sinal medido $y_S(t)$, que é a utilização do processador. Por exemplo, na Fig. 4.11(a) é apresentada a utilização do processador $y_S(t)$, e na Fig. 4.11(b) a entrada $u(t)$ num experimento em que o período de amostragem h é igual a $110ms$, a carga de trabalho é a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público *citizenengineermov*.mov [4], e a entrada $u(t)$ é uma sequência aleatória. Verifica-se que, quando a entrada $u(t)$ muda, a saída $y_S(t)$ não assume um novo valor médio instantaneamente, mas apenas após alguns períodos de amostragem. Além disso, há uma intensa componente estocástica na saída $y_S(t)$, pois o processamento exigido para a reprodução do vídeo varia a cada período de amostragem.

Dado que múltiplas tarefas podem ser executadas durante o período de amostragem, o modelo em (4.25) representa um comportamento médio do processo. Além disso, como a carga de trabalho é variante no tempo, é difícil prever o comportamento de cargas de trabalho futuras, até mesmo utilizando um preditor acurado e complexo [55]. Hellerstein et al. [52] também reconhecem o modelo em (4.25) como uma aproximação do comportamento do sistema real, mas enfatizaram que esse modelo é, apesar de simples, útil para a análise e projeto de controle em malha fechada para sistemas computacionais. Por exemplo, o modelo em (4.25) é utilizado como o modelo da relação entre a utilização do processador e o número máximo de conexões permitidas para um servidor HTTP Apache [52].

4.3.2.1 Bloco sensor

A utilização do processador é medida por meio do bloco sensor, representado na Fig. 4.2. Então a utilização do processador é a ‘variável controlada’, denotada por $y(t)$. Para a plataforma experimental, o papel do bloco sensor é ler o arquivo */proc/stat*, e calcular a utilização do processador, de acordo como procedimento apresentado por Wang et al. [24]. O conteúdo do arquivo

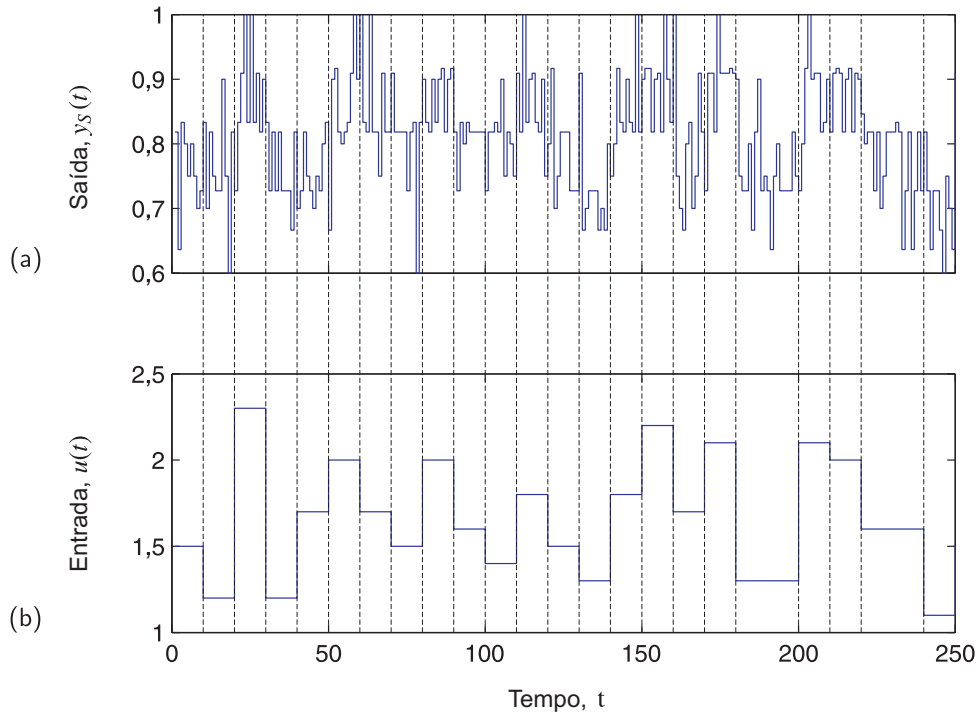


Figura 4.11: Exemplo da resposta transitória para uma entrada aleatória.

`/proc/stat` é um conjunto de contadores que são incrementados a cada período de amostragem T_K do núcleo do Linux. Então a resolução do sensor é T_K/h (onde h é o período de amostragem do gerenciador de energia). Um exemplo das primeiras linhas desse arquivo é apresentado abaixo:

```
cpu 3892 148 1875 65748 2847 0 26 0 0 0
cpu0 3892 148 1875 65748 2847 0 26 0 0 0
```

A linha `cpu` contém contadores associados a todo o processador, enquanto as linhas seguintes contêm contadores específicos de cada núcleo do processador. Cada coluna corresponde a um contador que é incrementado quando é realizada uma determinada atividade no processador:

- `user`: processos executados em espaço de usuário;
- `nice`: processos executados em espaço de usuário, e com baixa prioridade;
- `system`: processos executados em espaço de núcleo;
- `idle`: contador incrementado quando o processador está inativo;
- `iowait`: espera por uma operação de entrada/saída ser completada;
- `irq`: serviço de interrupções;
- `softirq`: serviço de interrupções de *software*;

- **steal**: “tempo roubado,” ou seja, o tempo utilizado na execução de outros sistemas operacionais num ambiente virtualizado;
- **guest**: tempo utilizado executando uma CPU virtual para outro sistema operacional sob controle do núcleo do sistema operacional Linux.

Para medir a razão entre o tempo inativo do processador e o período de amostragem, é necessário considerar que os contadores no arquivo `/proc/stat` são sempre incrementados. Portanto, é necessário medir as diferenças. Assim, se $S(t)$ é a soma dos contadores na linha `cpu` no instante de tempo t , e $I(t)$ é o valor da coluna `idle`, então as diferenças $S(t) - S(t - 1)$ e $I(t) - I(t - 1)$ correspondem respectivamente aos incrementos de $S(t)$ e $I(t)$ durante o período de amostragem. Então a razão entre o tempo inativo do processador e o período de amostragem equivale à razão em (4.27). E a saída $y_S(t)$ do bloco sensor, ou seja, a utilização do processador é o complemento de $i(t)$, o que é representado em (4.28).

$$i(t) = \frac{I(t) - I(t - 1)}{S(t) - S(t - 1)} \quad (4.27)$$

$$y_S(t) = 1 - i(t) \quad (4.28)$$

O problema de quantização está presente no bloco sensor porque $i(t)$ em (4.27) é igual a um quociente de números naturais. E assim a quantidade de níveis de quantização é igual a $S(t) - S(t - 1)$. Quanto maior o período de amostragem, maior será $S(t) - S(t - 1)$, e a saída $y_S(t)$ do bloco sensor terá melhor resolução. Por exemplo, se $S(t) - S(t - 1) = 10$, os possíveis valores de $y_S(t)$ são $0/10, 1/10, \dots, 10/10$. Se $S(t) - S(t - 1) = 100$, os possíveis valores de $y_S(t)$ são $0/100, 1/100, \dots, 100/100$. Portanto, não é interessante adotar um período de amostragem muito pequeno, pois a resolução da saída será pequena. E também não é interessante adotar um período de amostragem muito grande, pois, a utilização alta devido a uma carga em rajada durante um curto intervalo de tempo durante o período de amostragem pode ser filtrado, e não haverá reação no controlador a essa mudança na carga de trabalho.

4.3.2.2 Atuador

No bloco *Atuador*, como apresentado na Fig. 4.2, o sinal de controle $u(t)$ é mapeado para o comando $c(t)$. O sinal de controle $u(t)$ é uma variável contínua, enquanto a saída $c(t)$ do bloco atuador representado na Fig. 4.2 pertence a um conjunto de comandos discreto e finito $C = \{c_0, c_1, c_2, \dots, c_{|C|-1}\}$, onde $|C|$ é a cardinalidade de C . Assim a saída do controlador $u(t)$ ou deve ser quantizada para um comando discreto $c(t)$, que é constante durante o período de amostragem, ou deve ser modulada usando uma sequência de comandos aplicados em sub-intervalos do período de amostragem, tal como um sinal de modulação por largura de pulso. Neste trabalho foram modificadas e implementadas três técnicas para escolher o comando discreto $c(t)$:

(i) a função de quantização f_Q apresentada no Capítulo 3 em (3.2); (ii) a modulação por largura de pulso (*Pulse-Width Modulation (PWM)*); e (iii) o modulador de primeira ordem delta-sigma apresentado por Lefurgy, Wang e Ware [56]. Essas técnicas são apresentadas no restante desta seção.

A função de quantização f_Q apresentada no Capítulo 3 é uma função de \mathbb{R} para o conjunto de comandos $C = \{c_0, c_1, c_2, \dots, c_{|C|-1}\}$. Ordenando o conjunto C como uma sequência decrescente $(c_0, c_1, c_2, \dots, c_{|C|-1})$, caso deseja-se mudar a parte *on-chip* da utilização do processador $y(t)$ por um fator $u(t)$ tal que $\frac{c_0}{c_{j-1}} < u(t) \leq \frac{c_0}{c_j}$, então a função f_Q em (4.29) mapeia a entrada $u(t)$ para o comando c_j . Quando é aplicado o comando c_j pelo bloco atuador, o processador passa a operar no estado de energia s_j . Então a parte de *on-chip* da utilização do processador é multiplicada pelo fator s_0/s_j . A função de quantização f_Q pode ser implementada como o Algoritmo 5.

$$c(t) = f_Q(u(t)) = \begin{cases} c_0, & \text{se } u(t) \leq 1 \\ c_1, & \text{se } 1 < u(t) \leq \frac{c_0}{c_1} \\ c_2, & \text{se } \frac{c_0}{c_1} < u(t) \leq \frac{c_0}{c_2} \\ \vdots & \\ c_j, & \text{se } \frac{c_0}{c_{j-1}} < u(t) \leq \frac{c_0}{c_j} \\ \vdots & \\ c_{|C|-1}, & \text{caso contrário} \end{cases} \quad (4.29)$$

Algoritmo 5: Função de quantização f_Q

Dados: $j, (c_0, c_1, c_2, \dots, c_{|C|-1})$

Entrada: $u(t)$

Saída: $c(t)$

$j \leftarrow 0$;

se $u(t) \leq 1$ **então**

$c(t) \leftarrow c_0$;

senão se $u(t) \leq \frac{c_0}{c_{|C|-2}}$ **então**

para $j \leftarrow 1$ **até** $|C| - 2$ **faça**

se $\frac{c_0}{c_{j-1}} < u(t) \leq \frac{c_0}{c_j}$ **então**

$c(t) \leftarrow c_j$;

senão

$c(t) \leftarrow c_{|C|-1}$;

Como só é possível mapear a saída do controlador $u(t)$ para um dos comandos no conjunto finito $C = \{c_0, c_1, c_2, \dots, c_{|C|-1}\}$, a característica de saturação em (4.16) pode ser reescrita como (4.30). Pelo restante desta seção, a saída do controlador $u(t)$ é considerada como a sua

versão saturada $sat(u(t))$.

$$sat(u(t)) = \begin{cases} 1 & \text{se } u(t) \leq 1 \\ u(t) & \text{se } 1 < u(t) < \frac{c_0}{c_{|C|-1}} \\ \frac{c_0}{c_{|C|-1}} & \text{se } u(t) \geq \frac{c_0}{c_{|C|-1}} \end{cases} \quad (4.30)$$

Se o número de elementos do conjunto de comandos C é pequeno, então o fator $\frac{c_0}{c_j}$ da saída c_j da função de quantização f_Q em (4.29) pode desviar consideravelmente do nível desejado, i.e. o valor da entrada $u(t)$. Ordenando o conjunto C como uma sequência decrescente $(c_0, c_1, c_2, \dots, c_{|C|-1})$, então, durante um período de amostragem de duração h , um comando $c_{sup} = c_{j-1}$ pode ser aplicado num subintervalo com duração Dh , e outro comando $c_{inf} = c_j$ pode ser aplicado num subintervalo com duração $(1 - D)h$, onde $0 \leq D \leq 1$. O resultado é um sinal de modulação por largura de pulso $v(\tau)$ (em que a variável de tempo τ é contínua e t é discreta) como representado na Fig. 4.12. Com base no princípio da função de quantização f_Q em (4.29), os comandos c_{sup} e c_{inf} são escolhidos utilizando-se o Algoritmo 6. E o ciclo de trabalho D é tal que o valor médio do sinal $v(\tau)$ durante o intervalo de amostragem $\bar{v} = (1/h) \int_{th}^{(t+1)h} v(\tau) d\tau$ é igual a $u(t)$. Assim

$$\frac{1}{h} \int_{th}^{(t+1)h} v(\tau) d\tau = u(t) \Rightarrow \frac{1}{h} \left(\frac{c_0}{c_{sup}} Dh + \frac{c_0}{c_{inf}} (1 - D)h \right) = u(t) \Rightarrow \quad (4.31)$$

$$\Rightarrow D = \frac{u(t) - \frac{c_0}{c_{inf}}}{\frac{c_0}{c_{sup}} - \frac{c_0}{c_{inf}}} \quad (4.32)$$

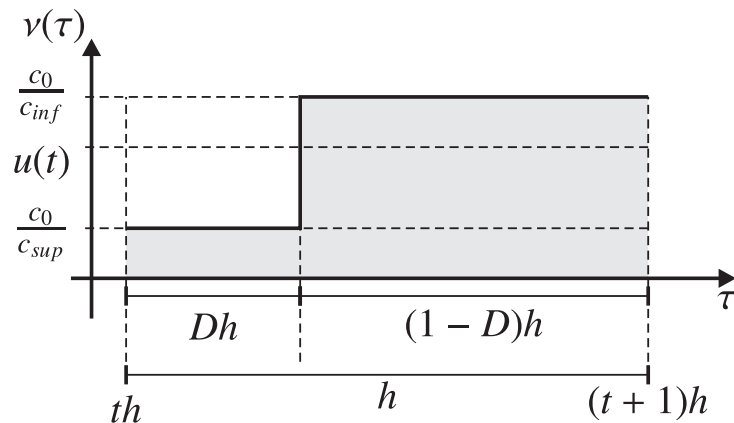


Figura 4.12: Representação para o sinal de modulação por largura de pulso.

Algoritmo 6: Calcular c_{sup} e c_{inf}

Dados: j
Entrada: $(c_0, c_1, c_2, \dots, c_{|C|-1}), u(t) \in [1, c_0/c_{|C|-1}]$
Saída: c_{sup}, c_{inf}
 $j \leftarrow 0$;

se $u(t) \leq 1$ **então**
 $\quad c_{sup} \leftarrow c_0$;

 $\quad c_{inf} \leftarrow c_0$;

senão
 \quad **para** $j \leftarrow 1$ **até** $|C| - 1$ **faça**
 \quad **se** $c_0/c_{j-1} < u(t) \leq c_0/c_j$ **então**
 \quad $\quad c_{sup} \leftarrow c_{j-1}$;

 \quad $\quad c_{inf} \leftarrow c_j$;

Além dos métodos descritos anteriormente, cada período de amostragem pode ser subdividido em $N_{ds} > 2$ subintervalos e uma sequência de comandos apropriada pode ser aplicada tal que o valor médio seja próximo o suficiente do valor desejado. Essa sequência é $(c(\kappa = 0), c(\kappa = 1), \dots, c(\kappa = N_{ds} - 1))$, onde κ é o índice do subintervalo, e $c(\kappa)$ é ou $c_{sup} = c_{j-1}$ ou $c_{inf} = c_j$, como mostrado na Fig. 4.13 para $N_{ds} = 10$. Os comandos c_{sup} e c_{inf} são calculados a cada período de amostragem t , como apresentado no Algoritmo 6. Em cada subintervalo κ do período de amostragem t , a sequência pode ser obtida a partir do modulador de primeira ordem delta-sigma apresentado por Lefurgy, Wang e Ware [56], que é aplicado no Algoritmo 7.

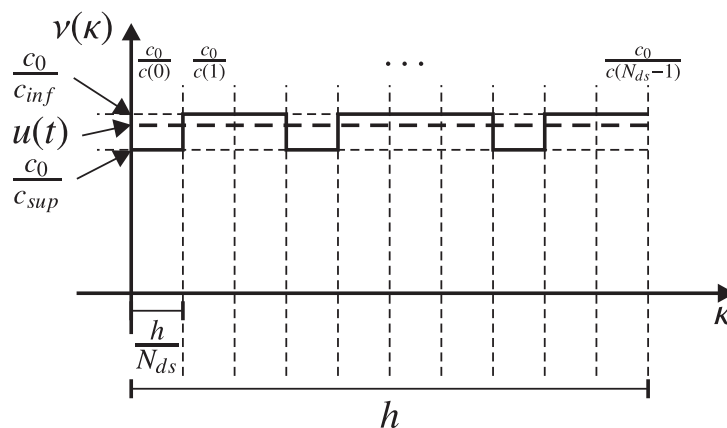


Figura 4.13: Representação para o sinal de modulação de primeira ordem delta-sigma.

Algoritmo 7: Modulador de primeira ordem delta-sigma**Dados:** diferença, soma**Entrada:** $u(t)$, c_{sup} , c_{inf} **Saída:** $c(\kappa)$ diferença $\leftarrow u(t) - \frac{c_0}{c(\kappa-1)}$;soma \leftarrow diferença + soma ;**se** $soma > 0$ **então**└ $c(\kappa) \leftarrow c_{inf}$;**senão**└ $c(\kappa) \leftarrow c_{sup}$;**4.3.2.3 Seleção do período de amostragem**

A utilização do processador deve ser amostrada a cada período de amostragem para que o gerenciador de energia aplique o comando de acordo com os níveis desejados de potência e desempenho. O gerenciador de energia é um programa executado pelo processador, tal como o sistema operacional e a carga de trabalho do sistema. Então, se o período de amostragem é muito pequeno, o gerenciador de energia tem uma alta utilização do processador (devido a ele próprio), afetando a execução da carga de trabalho. Se o período de amostragem é muito grande, o gerenciador de energia tem baixa utilização do processador e consumo de energia (devido a ele próprio), mas pode haver perda de oportunidades de adaptar os estados de energia do processador para cargas de trabalho em rajada.

Considerando que a latência de transição entre os estados de energia da plataforma experimental é igual a $110000ns$, o período de amostragem padrão da política de energia *Ondemand* do Linux [49] é 1000 vezes a latência de transição, ou seja, $110ms$. Neste trabalho, também foi escolhido o período de amostragem h igual a $110ms$, porque: (i) o gerenciador de energia poderá adaptar os estados de energia para cargas de trabalho em rajada; (ii) como apresentado na Tabela 4.1, o custo do gerenciador de energia, i.e. a utilização do processador pelo próprio gerenciador de energia, é negligenciável para todas as implementações do bloco atuador apresentado na Seção 4.3.2.2.

4.3.2.4 A estimação de parâmetros do processo

Para aplicar o algoritmo de mínimos quadrados recursivos em (4.9) para um sistema variante no tempo, é necessário escolher adequadamente o perfil de esquecimento $\beta(t,k)$ em (4.10). Um método comum [44] é selecionar $\lambda(t)$ como um valor constante λ , o *fator de esquecimento*.

Tabela 4.1: Utilização do processador pelo próprio gerenciador de energia

Implementação do bloco Atuador	Utilização de CPU pelo gerenciador de energia
Função de quantização f_Q	0,3%
Modulação por largura de pulso (PWM)	0,4%
Modulador delta-sigma com $N_{ds} = 10$	1,0%

Assim $\beta(t,k)$ é igual a λ^{t-k} . Para escolher λ , os indicadores considerados neste trabalho foram um limitante inferior da excitação persistente e o erro de controle médio quadrático, que são descritos no restante desta seção.

A entrada $u(t)$ é persistentemente excitante de ordem n_a (a ordem do sistema) se, para todo t , há um inteiro m tal que

$$\rho_1 I < \sum_{k=t}^{t+m} \varphi(k) \varphi^T(k) < \rho_2 I \quad (4.33)$$

onde $\rho_1, \rho_2 > 0$ e I é a matriz identidade [45]. O valor ρ_1 que satisfaz a primeira inequação em (4.33) é um limitante inferior da excitação persistente. O erro de controle médio quadrático é

$$V = \frac{1}{t_f - t_i + 1} \sum_{t=t_i}^{t_f} [u_C(t) - y_S(t)]^2 \text{ para } t \in [t_i, t_f]. \quad (4.34)$$

Experimentos foram realizados na plataforma experimental para avaliar os indicadores $V(\lambda)$ e $\rho_1(\lambda)$ como funções do parâmetro λ para os valores constantes 0,900, 0,905, ..., 0,995. Esses experimentos são descritos no Apêndice B. O fator de esquecimento $\lambda = 0,935$ foi escolhido por representar uma solução de compromisso entre o limitante inferior da excitação persistente e o erro de controle médio quadrático.

4.3.2.5 O projeto do controlador

A metodologia de alocação de pólos de grau mínimo (*MDPP*) [45] pode ser aplicada ao controlador linear geral apresentado na Seção 4.2.2.2. Para o processo em (4.25) e a especificação para a malha fechada como $\frac{BT}{RA+BS} = \frac{\beta_1 q + \beta_0}{q^2 + \alpha_1 q + \alpha_0}$, os polinômios do controlador linear geral são $R(q) = q + (\alpha_1 - a_1)$, $S(q) = (\alpha_0 - \alpha_1 a_1 + a_1^2)/b_1$, e $T(q) = \frac{\beta_1}{b_1} q + \frac{\beta_0}{b_1}$, como apresentado no Apêndice A.

Para o controlador proporcional-integral apresentado na Seção 4.2.2.2, considerando o sistema de entrada única e saída única (*SISO*) em (4.26) e a função de transferência do controlador proporcional-integral em (4.20), a função de transferência do sistema em malha fechada, calcu-

lada a partir de (4.21), é apresentada em (4.35).

$$G(z) = \frac{Kb_1z + Kb_1\frac{h-T_i}{T_i}}{z^2 + (a_1 - 1 + Kb_1)z + (-a_1 + Kb_1\frac{h-T_i}{T_i})} \quad (4.35)$$

Para o polinômio característico do sistema em malha fechada da forma $z^2 + \alpha_1z + \alpha_0$, então o ganho proporcional e o tempo de integração são apresentados em (4.36) e (4.37) respectivamente.

$$K = \frac{\alpha_1 - a_1 + 1}{b_1} \quad (4.36)$$

$$T_i = \frac{h(\alpha_1 - a_1 + 1)}{\alpha_1 + \alpha_0 + 1} \quad (4.37)$$

A especificação para a malha fechada $\frac{0,1839q+0,1308}{q^2-1,049q+0,3635}$ corresponde a um fator de amortecimento igual a 0,7 e a uma frequência natural igual a 6,571. Esses parâmetros permitem um sobre-sinal de 0,05 e um tempo de acomodação de 1s, considerando que o período de amostragem é $h = 110ms$. Então os valores $\beta_1 = 0,1839$, $\beta_0 = 0,1308$, $\alpha_1 = -1,049$ e $\alpha_0 = 0,3635$ foram utilizados no bloco *Projeto do controlador* da Fig. 4.3 para calcular os parâmetros do controlador.

4.3.2.6 O controlador

A lei de controle implementada no bloco *Controlador* do regulador auto-sintonizável considerando o controlador linear geral com compensação *antiwindup* é apresentada em (4.18). O controlador proporcional-integral foi implementado com o Algoritmo 4. Para ambos os controladores, os parâmetros foram calculados como descrito na Seção 4.3.2.5.

Como apresentado na Seção 4.3.2.1, a resolução do bloco sensor é igual a T_K/h . Assim, quando a saída $y_S(t)$ é aproximadamente igual à entrada de referência $u_C(t)$, é possível que a saída varia entre os valores $Y_{S_L} = iT_K/h$ e $Y_{S_H} = (i+1)T_K/h$, onde i é o menor inteiro não-negativo tal que $iT_K/h \leq u_C(t) \leq (i+1)T_K/h$. Para evitar frequentes chaveamentos do estado de energia do processador devido à resolução do bloco sensor, o sinal de controle $u(t)$ somente deve ser atualizado para o bloco atuador se $y_S(t) < Y_{S_L}$ ou $y_S(t) > Y_{S_H}$.

4.3.2.7 Resultados

Nesta seção, são comparadas as políticas de gerenciamento de energia seguintes:

1. *Ondemand* [49]: quando a utilização do processador é maior que ou igual a um parâmetro *up_threshold*, a máxima frequência disponível é escolhida. Se a utilização do processador diminui, a frequência é gradualmente reduzida.

2. *Conservative* [49]: essa política é similar ao *Ondemand*, com a diferença de que, a frequência é aumentada ou diminuída gradativamente (num passo de $freq_step$) de acordo com a utilização do processador. A frequência é reduzida se a utilização do processador permanece menor que um parâmetro $down_threshold$.
3. *Userspace* [49]: a frequência $s \in \mathcal{S}$ é constante ao longo do tempo.
4. A *técnica de controle adaptativo* apresentada neste capítulo, com o controlador linear geral (*RST*) implementado como a lei de controle em (4.18) e três implementações alternativas do bloco atuador: (i) *RST+Quantizador*: a função de quantização f_Q apresentada no Algoritmo 5; (ii) *RST+PWM*: a modulação por largura de pulso (PWM) com os comandos c_{sup} e c_{inf} escolhidos utilizando-se o Algoritmo 6 e o ciclo de trabalho avaliado em (4.32); e (iii) *RST+DS*: o modulador delta-sigma de primeira ordem apresentado no Algoritmo 7 com $N_{ds} = 10$ sub-intervalos.
5. *PI+PWM*: a *técnica de controle adaptativo* apresentada neste capítulo, com o controlador proporcional-integral implementado com base no Algoritmo 4, e o bloco atuador implementado com a modulação por largura de pulso (PWM).

Os indicadores utilizados para comparar as políticas são: (i) a potência média da plataforma experimental; (ii) a penalidade de desempenho média; e (iii) a utilização média do processador. Uma penalidade de desempenho é considerada como o evento “a utilização do processador $y(t)$ é igual a 1, i.e. 100%”. Esse indicador é denominado *penalidade de desempenho* porque, se surgem novas requisições da carga de trabalho quando a utilização do processador é igual a 100%, essas requisições não serão servidas imediatamente. Essa situação é semelhante ao conceito de *perda de requisições* [10]. Então, de acordo com a carga de trabalho do sistema, e o estado de energia do processador, a penalidade de desempenho $l(t)$ é definida em (4.38).

$$l(t) = \begin{cases} 1, & \text{se } y_S(t) = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (4.38)$$

Para avaliar o consumo de potência $p(t)$, o módulo de aquisição de dados NI USB-6210⁴ foi utilizado para medir a tensão e corrente de alimentação do sistema com gerenciamento de energia. A corrente foi obtida por meio da medição da tensão em um resistor *shunt* igual a $9m\Omega$. O módulo NI USB-6210 foi conectado a um computador pessoal por meio de uma conexão USB (*Universal Serial Bus*). No computador pessoal, que foi dedicado à aquisição de dados, o programa LabVIEW foi utilizado para a aquisição dos dados fornecidos pelo NI USB-6210.

⁴O módulo de aquisição de dados NI USB-6210 é um dispositivo em que estão presentes 16 entradas analógicas, 4 entradas digitais e 4 saídas digitais. Os conversores analógico-digitais conectados às entradas analógicas permitem a aquisição de dados com resolução de 16 *bits*.

No restante deste texto, o valor médio \bar{x} de um sinal $x(t)$, a partir de um instante inicial t_i a um instante final t_f , é calculado como mostrado em (4.39).

$$\bar{x} = \frac{1}{t_f - t_i + 1} \sum_{t=t_i}^{t=t_f} x(t) \quad (4.39)$$

Serão descritas a seguir alguns tipos de carga de trabalho utilizadas neste estudo de caso, usando a nomenclatura adotada nos Capítulos 2 e 3:

- *Carga de trabalho leve*: considerando a carga de trabalho do sistema como a reprodução, por meio do tocador *totem* [2], da música de domínio público on000001.mp3 [3], são apresentados na Fig. 4.14 o consumo médio de potência do sistema com gerenciamento de energia \bar{p} , a penalidade de desempenho média \bar{l} , e a utilização média do processador \bar{y}_S para a política Userspace. Verifica-se que a utilização média do processador tem valor máximo para Userspace@0,800GHz. Assim, não é possível uma utilização média maior que esse valor, pois ela é obtida para o menor valor de frequência do processador. Em consequência, ao se utilizar a técnica de gerenciamento de energia apresentada no Capítulo 4 com um valor de referência maior que a utilização média do processador para Userspace@0,800GHz, haveria um erro em regime permanente. Para cargas de trabalho leves, não é possível então que a utilização do processador como saída do sistema em malha fechada atinja um valor de referência maior que a utilização média do processador quando este está em sua mínima frequência. Verifica-se a partir da Fig. 4.14, que a potência média aumenta com a frequência, enquanto a penalidade de desempenho e a utilização do processador diminuem. Se a penalidade de desempenho média de aproximadamente 0,4% para a frequência mínima for aceitável, então esta deve ser escolhida, pois permite a menor potência média. Se a penalidade de desempenho média para a frequência mínima não for aceitável, então deve ser escolhida a menor frequência para a qual haja penalidade de desempenho média nula, e.g. Userspace@1,333GHz.
- *Carga de trabalho intensa*: considerando a carga de trabalho do sistema como a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público citizenengine-erm.mov [4], são apresentados na Fig. 4.15 o consumo médio de potência do sistema com gerenciamento de energia \bar{p} , a penalidade de desempenho média \bar{l} , e a utilização média do processador \bar{y}_S para a política Userspace. Verifica-se que a utilização média do processador tem valor máximo para Userspace@0,800GHz e mínimo para Userspace@1,867GHz. Assim, é possível variar a utilização média entre esses dois valores. Para cargas de trabalho intensas, é possível então que a utilização do processador como saída do sistema em malha fechada atinja um valor de referência maior ou igual à utiliza-

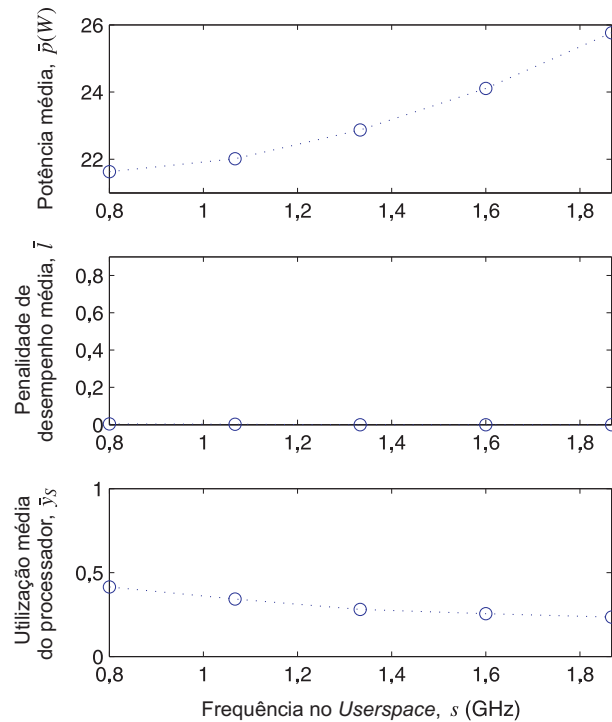


Figura 4.14: Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a reprodução, por meio do tocador *totem* [2], da música de domínio público *on000001.mp3* [3].

ção média do processador quando este está em sua máxima frequência, e menor ou igual à utilização média do processador quando este está em sua mínima frequência.

- *Pior caso da carga de trabalho*: considerando a carga de trabalho do sistema como a execução do *benchmark* *ebizzy* [5], são apresentadas na Fig. 4.16 o consumo médio de potência do sistema com gerenciamento de energia \bar{p} , a penalidade de desempenho média \bar{l} , e a utilização média do processador \bar{y}_s para a política Userspace. Verifica-se que a utilização média do processador tem valor constante e igual a 100% para todas as frequências. Isso acontece porque essa carga de trabalho é de uso intensivo de CPU, i.e., independentemente da frequência do processador, sua utilização é sempre máxima. Assim, não é possível uma utilização menor que 100%. E, em consequência, ao se utilizar a técnica de gerenciamento de energia apresentada no Capítulo 4 com um valor de referência menor que 1, haveria um erro em regime permanente. Para cargas de trabalho de uso intensivo de CPU, não é possível então que a utilização do processador como saída do sistema em malha fechada atinja um valor de referência menor que 1.
- Para uma *carga de trabalho variante ao longo do tempo*, o processador pode estar sujeito a cargas de trabalho leves, cargas de trabalho intensas e o pior caso da carga de trabalho em instantes diferentes de tempo. A carga de trabalho considerada neste estudo de

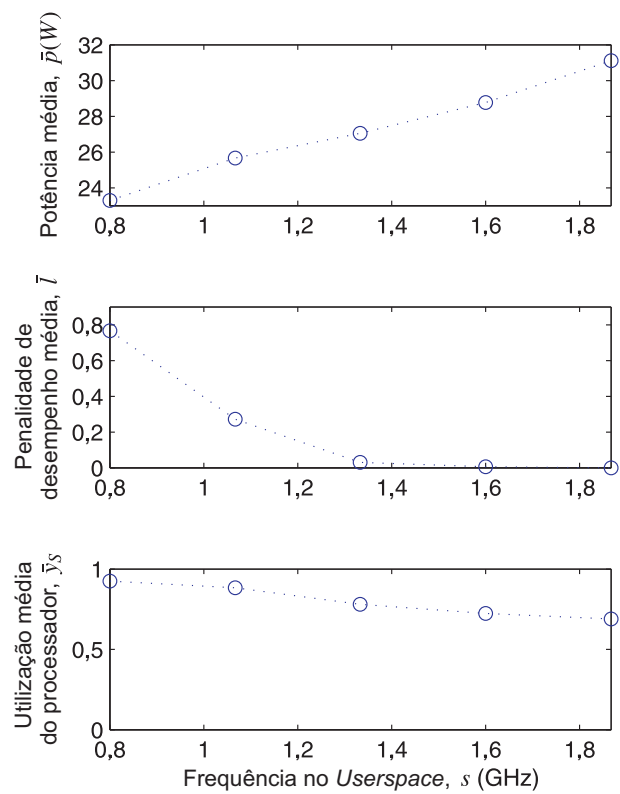


Figura 4.15: Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público *citizenengineermv.mov* [4].

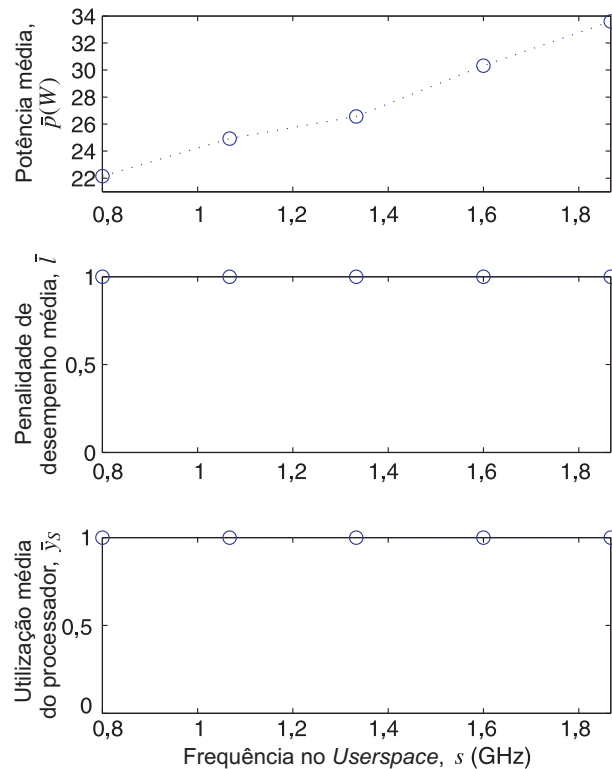


Figura 4.16: Consumo de potência, penalidade de desempenho e a utilização do processador para a política Userspace, com a carga de trabalho sendo a execução do *benchmark* ebizzy [5].

caso foi a sequência: (i) durante o intervalo $[0, 5s]$, um período inativo; (ii) $[5s, 185s]$, a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público *citizenengineeringvm.mov* [4], cuja codificação, resolução e taxa de quadros são *avc1*, 1280×720 e 30 quadros/s respectivamente; (iii) $[185s, 195s]$, período inativo; (iv) $[195s, 255s]$, a reprodução, por meio do tocador *totem* [2], do áudio de domínio público *on000001.mp3* [3]; (v) $[255s, 265s]$, período inativo; (vi) $[265s, 275s]$, a execução do *benchmark* ebizzy [5]; (vii) $[275s, 286s]$, período inativo. Nos experimentos, os valores $t_i = 1$ e $t_f = 2600$ períodos de amostragem foram usados em (4.39), porque a duração da carga de trabalho variante foi $286s = 2600 \cdot 110ms$. O *benchmark* ebizzy é executado com uma duração fixa de 10 segundos, o que é obtido por meio de uma configuração na inicialização do próprio executável do *benchmark*.

Por exemplo, nas Figuras 4.17 e 4.18 são apresentadas as respostas transitórias para a política Userspace respectivamente a 0,800GHz e 1,867GHz. Comparando a reprodução da música nas Figuras 4.17 e 4.18, verifica-se que a utilização do processador para Userspace@0,800GHz é maior que para Userspace@1,867GHz, mas não satura em 1, ou seja, a música é reproduzida normalmente. Comparando a reprodução do vídeo nas Figuras 4.17 e 4.18, verifica-se que a utilização do processador para Userspace@0,800GHz além de ser maior que para Userspace@1,867GHz, satura em 1 durante a maior parte

do tempo, ou seja, o vídeo não é reproduzido normalmente. Para a carga ebizzy, não é observada uma diferença significativa entre as Figuras 4.17 e 4.18 quanto à utilização do processador, que é saturada em 1. A diferença principal é a vazão (*throttling*) de CPU, que é maior para Userspace@1,867GHz.

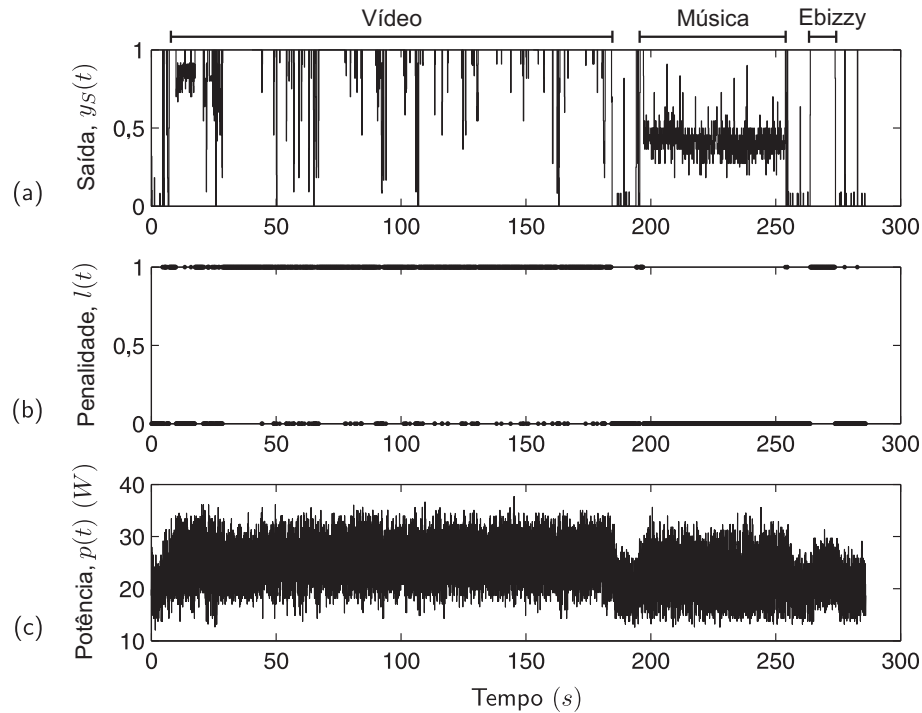


Figura 4.17: Exemplo da resposta transitória para a política Userspace@0,800GHz, com a carga de trabalho variável ao longo do tempo sendo: (i) a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público citizenengineermov.mov [4], (ii) a reprodução, por meio do tocador *totem* [2], da música de domínio público on000001.mp3 [3], e (iii) a execução do *benchmark ebizzy* [5].

A partir das considerações sobre os diferentes tipos de carga de trabalho, espera-se que a técnica de gerenciamento de energia apresentada neste capítulo tenha comportamentos adequados para cada uma delas. Para uma carga de trabalho leve, em que há pequena probabilidade da utilização do processador saturar em 1, é possível utilizar a mínima frequência do processador para minimizar o consumo de energia. Para uma carga de trabalho intensa, para a qual a utilização do processador saturaria em 1 se fosse utilizada a mínima frequência do processador, é necessário algum sistema para manter a utilização do processador menor que 1, e ao mesmo tempo minimizar o consumo de energia. Para o pior caso da carga de trabalho, em que a utilização do processador satura em 1 independentemente da frequência do processador, é necessário utilizar a máxima frequência para garantir o desempenho necessário e atender à carga no menor período de tempo possível, após o qual há inatividade e a oportunidade de reduzir a frequência

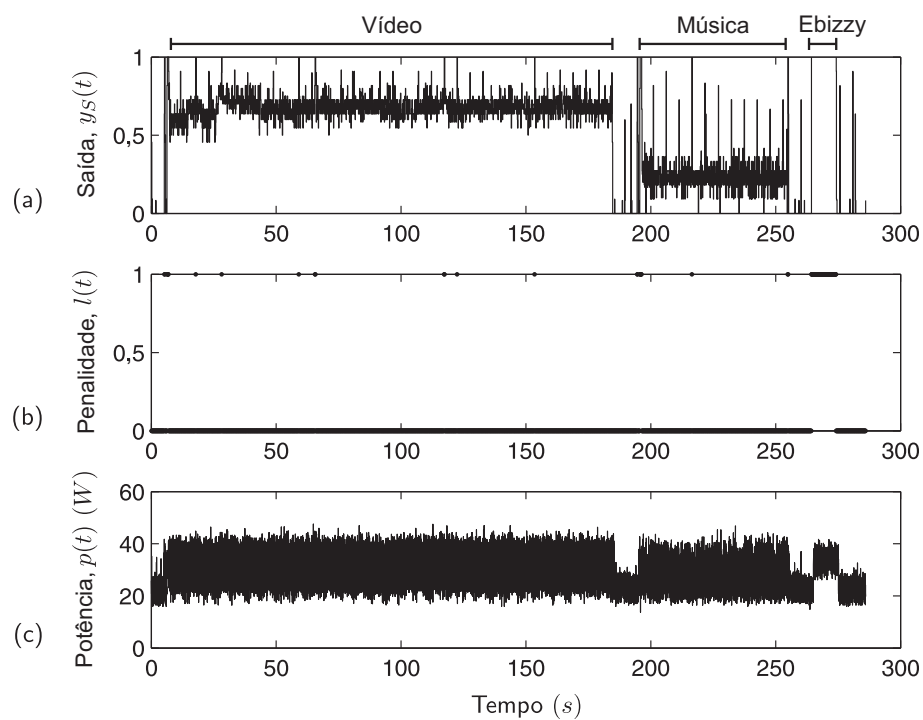


Figura 4.18: Exemplo da resposta transitória para a política Userspace@1,867GHz, com a carga de trabalho variável ao longo do tempo sendo: (i) a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público citizenengineermv.mov [4], (ii) a reprodução, por meio do tocador *totem* [2], da música de domínio público on000001.mp3 [3], e (iii) a execução do benchmark ebizzy [5].

do processador. Para a técnica de gerenciamento de energia apresentada no Capítulo 4, esse comportamento desejado é implementado:

- na lei de controle em (4.18): se $y_S(t)$ aumenta, então $u(t)$ diminui, e maiores frequências são aplicadas. Se $y_S(t)$ diminui, então $u(t)$ aumenta, e menores frequências são aplicadas.
- no Algoritmo 4: se $e(t) < 0$, então $u(t)$ diminui, e maiores frequências são aplicadas. Se $e(t) > 0$, então $u(t)$ aumenta, e menores frequências são aplicadas.
- na escolha das menores frequências a serem aplicadas pelo bloco atuador representado na Fig. 4.2, de acordo com saída do controlador $u(t)$, com o objetivo de *minimização do consumo de energia*. Essa ação depende da implementação do bloco *atuador*: a função de quantização f_Q no Algoritmo 5, ou a escolha de comandos no Algoritmo 6 para o sinal PWM ou o modulador de primeira ordem delta-sigma [56].

Para a técnica de controle adaptativo introduzida para gerenciamento de energia com a entrada de referência $u_C(t) = 0,80, \forall t$, a modulação por largura de pulso (PWM) e o período de amostragem h igual a $110ms$, a resposta transitória durante um experimento é apresentada na Fig. 4.19. Durante o intervalo $[0, 5s]$, o processador está inativo, e $y_S(t) \approx 0$, como apresentado na Fig. 4.19(b). O sinal de controle $u(t)$ (veja Fig. 4.19(a)) é então saturado ao valor máximo $s_0/s_4 \approx 2,33$ (correspondendo à frequência $0,800GHz$) porque a saída $y_S(t)$ é menor do que a entrada de referência $u_C(t)$. Apesar do fato de que $y_S(t)$ não pode ser aumentado para alcançar $u_C(t)$ quando o processador está inativo, a decisão do controlador é correta, porque a mínima frequência é aplicada e há economia de energia. No início do intervalo $[5s, 185s]$, o tocador totem é inicializado e a saída $y_S(t)$ é subitamente aumentada. O sinal de controle $u(t)$ é reduzido porque $y_S(t)$ é maior do que $u_C(t)$. Após aproximadamente $7,8s$, um regime estacionário é alcançado, em que a saída $y_S(t)$ varia em torno da entrada de referência $u_C(t)$ devido ao ruído de medição e variações ao longo do tempo na demanda de processamento da reprodução do vídeo. Durante os intervalos $[185s, 195s]$, $[255s, 265s]$ e $[275s, 286s]$, o processador está inativo, e a decisão do controlador é a mesma que para $[0, 5s]$. A saída $y_S(t)$ é menor do que a entrada de referência $u_C(t)$ durante a maior parte do intervalo $[195s, 255s]$, porque a reprodução do arquivo de áudio implicou em baixa utilização do processador mesmo com a menor frequência do processador. O sinal de controle $u(t)$ é saturado no máximo valor $s_0/s_4 \approx 2,33$ (correspondendo à frequência $0,800GHz$) e há economia de energia. Durante o período $[265s, 275s]$, a saída $y_S(t)$ é maior do que a entrada de referência $u_C(t)$ porque a carga de trabalho é de uso intensivo de CPU, e o sinal de controle $u(t)$ é saturado no menor valor $s_0/s_0 = 1,0$ (correspondendo à frequência $1,867GHz$). Apesar do fato de não ser possível reduzir a saída $y_S(t)$ para o valor de $u_C(t)$, a decisão do controlador é correta, porque a máxima frequência é aplicada para satisfazer a demanda de desempenho. O modelo do processo é o modelo de primeira ordem, entrada-única,

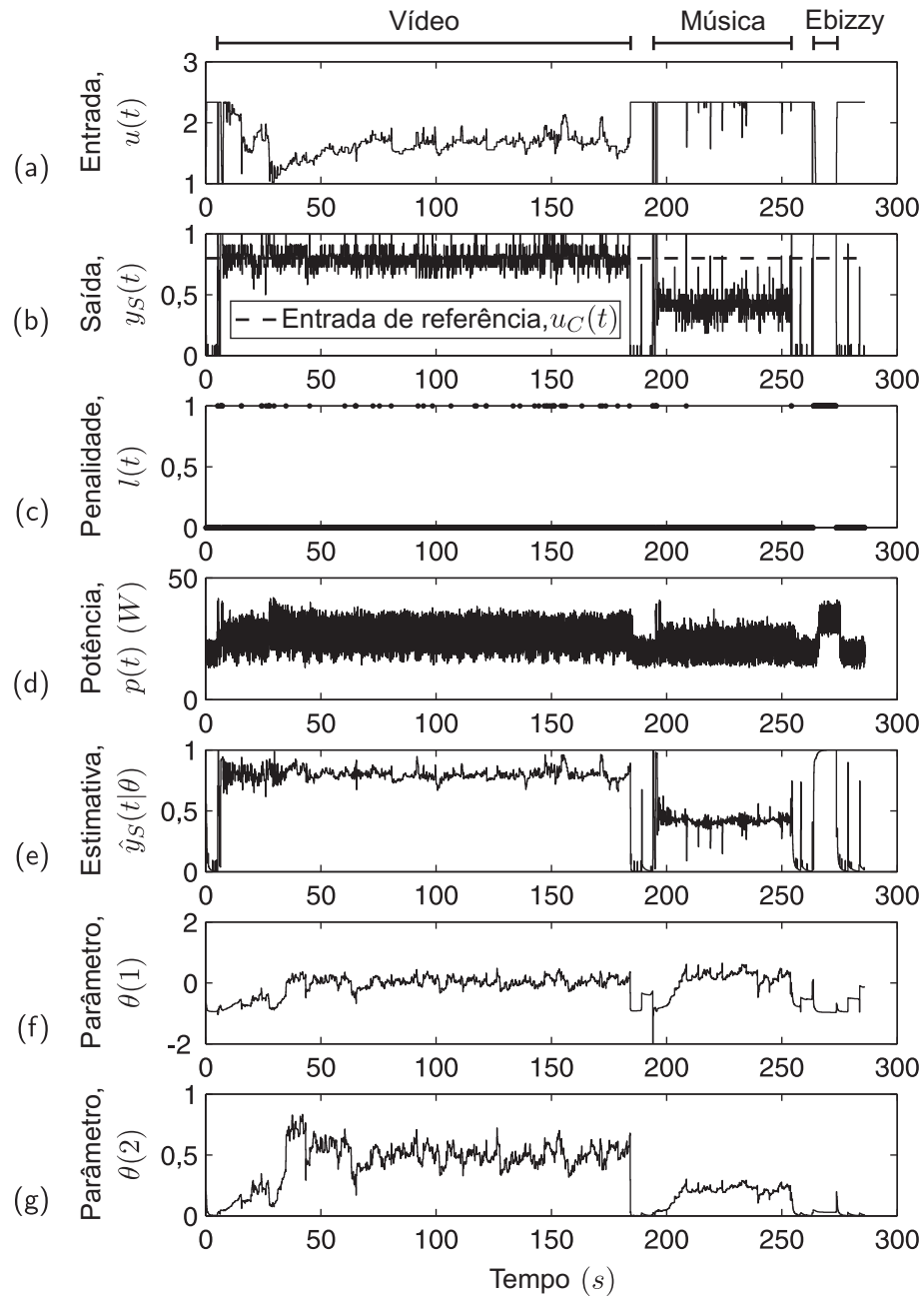


Figura 4.19: Exemplo da resposta transitória para a técnica de controle adaptativo.

saída-única (SISO) em (4.25). Então a saída do bloco de *Estimação* na Fig. 4.3 é o vetor de parâmetros θ em (4.6), que é igual a $\begin{bmatrix} a_1 & b_1 \end{bmatrix}^T$. Nas Figuras 4.19(f) e 4.19(g), as componentes de θ serão representadas respectivamente como $\theta(1) = a_1$ e $\theta(2) = b_1$. A estimativa $\hat{y}_S(t|\theta)$ foi obtida a partir dos parâmetros em θ por meio de (4.8) e ela tem a mesma dinâmica do sinal $y_S(t)$ (veja a Fig. 4.19(e)). As penalidades de desempenho são apresentadas na Fig. 4.19(c).

Para todas as políticas consideradas neste estudo de caso, a potência média do sistema com gerenciamento de energia \bar{p} , a penalidade de desempenho média \bar{l} , e a utilização média do processador \bar{y} são apresentadas na Fig. 4.20. Os resultados experimentais para as frequências constantes na política Userspace são apresentados no lado esquerdo da Fig. 4.20. A potência média aumenta com a frequência, enquanto a penalidade de desempenho e a utilização do processador diminuem. Os resultados para as políticas Ondemand e Conservative são apresentados como linhas tracejadas ao longo dos gráficos apresentados na Fig. 4.20, para que os resultados de Ondemand e Conservative possam ser comparados com os resultados de Userspace e a técnica de controle adaptativo. Neste experimento, a potência para o Ondemand foi menor que para Conservative, mas o primeiro teve maiores penalidade de desempenho e utilização do processador que o último. A potência para o Ondemand é menor que para Userspace@1,333GHz (leia-se Userspace com frequência constante igual a 1,333GHz), e a penalidade de desempenho é menor do que Userspace@1,067GHz.

Os resultados para a técnica de controle adaptativo são apresentados no lado direito da Fig. 4.20, com diversos valores para a entrada de referência u_C . Em geral, para cada valor da entrada de referência u_C , a potência foi menor para RST + Quantizador. A penalidade de desempenho e a utilização do processador para cada entrada de referência u_C apresentaram valores próximos para as implementações da técnica de controle adaptativo, exceto para $u_C = 0,90$ e $0,95$, em que a penalidade de desempenho foi menor para PI + PWM. Então para a maior parte dos valores da entrada de referência u_C , as implementações apresentadas da técnica de controle adaptativo foram equivalentes em relação a desempenho.

Para todas as implementações da técnica de controle adaptativo apresentada, quando a entrada de referência aumenta, a penalidade de desempenho também aumenta, porque, como apresentado na Fig. 4.19(b), a variável controlada $y_S(t)$ varia em torno da entrada de referência $u_C(t)$, e, quando $u_C(t)$ está próximo de 1, aumenta a probabilidade de que uma penalidade de desempenho aconteça. No entanto, para $u_C(t) \leq 0,80$, a penalidade de desempenho para todas as implementações da técnica de controle adaptativo foi menor que a penalidade de desempenho para Ondemand. Além disso, quando a entrada de referência aumenta, a potência diminui, porque menores frequências são necessárias para aumentar a utilização do processador, como apresentado na utilização média do processador \bar{y}_S para a política Userspace na Fig. 4.20. Por exemplo, para $u_C \geq 0,85$, as potências médias para todas as implementações da técnica de

controle adaptativo foram menores que a potência média para Ondemand; e para $u_C = 0,95$, a potência média foi menor que a potência para Userspace@0,800GHz. Adicionalmente, para $u_C = 0,70, 0,75, \dots, 0,95$, as potências médias para todas as implementações da técnica de controle adaptativo foram menores que as potências para Conservative e Userspace@1,867GHz.

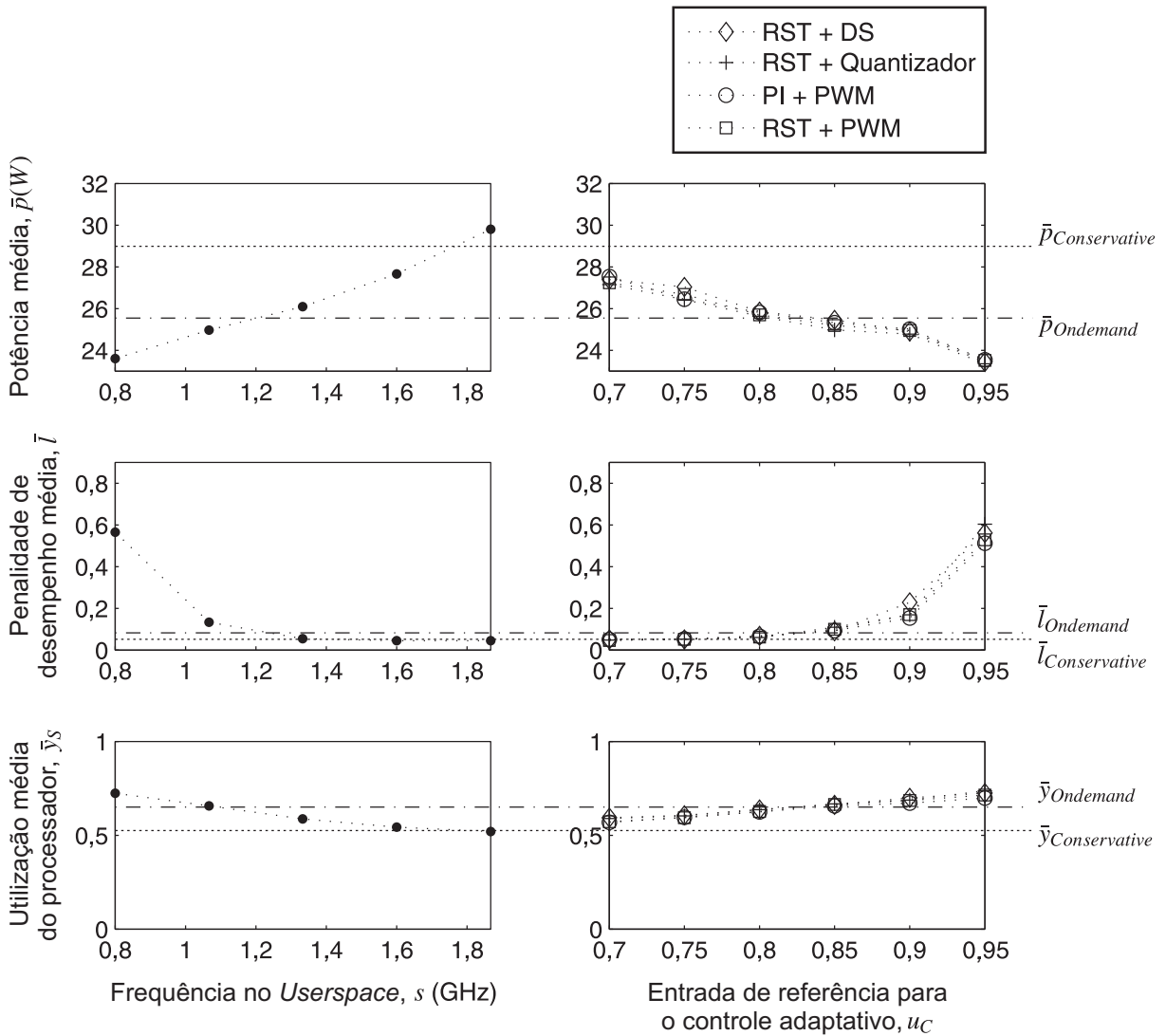


Figura 4.20: Consumo de potência, penalidade de desempenho e a utilização do processador para as políticas de gerenciamento de energia consideradas no estudo de caso.

4.4 Resumo

Na literatura de controle em malha fechada para gerenciamento de energia, os modelos e técnicas apresentadas são geralmente especializadas para a arquitetura considerada. Além disso, as soluções anteriores em malha fechada são somente adequadas quando um modelo acurado de potência e desempenho do sistema é disponível, e a maioria dessas técnicas é otimizada para

cargas de trabalho específicas. Contudo, cargas de trabalho reais de computadores de propósito geral são inicialmente desconhecidas e variantes no tempo. Então, para superar as dificuldades impostas pelas incertezas no modelo do sistema com gerenciamento de energia e a presença de cargas de trabalho variantes no tempo, uma técnica baseada em estimação e controle em malha fechada foi aplicada ao problema de gerenciamento dinâmico de energia do processador neste capítulo: controle adaptativo. Ao contrário de trabalhos anteriores, um modelo acurado de potência e desempenho do sistema não é necessário, e o controlador aplicado ao problema de gerenciamento de energia pode ser reprojetoado recursivamente em tempo de execução, de acordo com as flutuações da carga de trabalho.

Todos os blocos necessários para realizar controle adaptativo para gerenciamento de energia foram extensivamente analisados: estimação, controlador, atuador, sensor, e projeto do controlador. Dado que cargas de trabalho de computadores de propósito geral são variantes no tempo, é necessário realizar a estimação recursiva dos parâmetros do modelo do sistema. O bloco de *estimação* foi implementado com base no algoritmo de mínimos quadrados recursivos [44], que foi otimizado para o problema de gerenciamento de energia, considerando a *excitação persistente* [45] e o erro de controle médio quadrático.

Dois controladores foram projetados para o problema de gerenciamento de energia: (i) um controlador linear geral [45]; e (ii) um controlador proporcional-integral [46]. Considerando que o conjunto de possíveis estados de energia por meio de escalonamento dinâmico de tensão e frequência é discreto e finito, tendo tanto um estado de energia mínimo quanto um máximo, ambos o controlador linear geral e o controlador proporcional-integral foram estendidos com compensação *antiwindup* [46].

O sinal de controle é contínuo, mas o conjunto de possíveis estados de energia por meio de escalonamento dinâmico de tensão e frequência é discreto e finito. Então, uma função de quantização ou um método de modulação devem ser utilizados no bloco *atuador* para mapear o sinal de controle contínuo num comando discreto, que é o estado de energia desejado para o sistema. O *atuador* tem a interface adequada com o sistema operacional para aplicar um comando ao processador. Três implementações para o bloco *atuador* foram apresentadas: (i) uma função de quantização; (ii) a modulação por largura de pulso; (iii) e a modulação delta-sigma.

A execução do próprio gerenciador de energia deve causar custos de tempo de execução e consumo de potência desprezíveis para o sistema com gerenciamento de energia. Por exemplo, o gerenciador de energia não deve causar maior consumo de energia do que a quantidade que ele economiza, nem prejudicar a execução normal das demais aplicações do sistema devido a uso excessivo de CPU. A utilização do processador pela técnica introduzida foi desprezível, isto é, o gerenciador de energia é executado no sistema com gerenciamento de energia com pequeno

custo para o processador.

Uma plataforma experimental foi utilizada para avaliar a técnica apresentada. Os recursos de gerenciamento de energia dessa plataforma foram apresentados e várias políticas de energia foram comparadas. Para cada uma dessas políticas, os indicadores foram: a potência média do sistema com gerenciamento de energia, a penalidade de desempenho média, e a utilização média do processador. Foi mostrado que a técnica introduzida pode ser otimizada tanto para consumo de energia reduzido quanto para desempenho, dependendo da escolha da entrada de referência.

Capítulo 5

Conclusões e trabalhos futuros

A minimização do consumo de energia é essencial para diversos tipos de sistemas computacionais. Por exemplo, para sistemas alimentados a bateria, busca-se aumentar a autonomia da bateria. Para sistemas alimentados pela rede elétrica, o objetivo é reduzir os custos com consumo de energia e resfriamento do sistema computacional. Neste contexto, o objetivo do gerenciamento de energia de sistemas computacionais é minimizar o consumo de energia, tal que a demanda de desempenho do sistema seja satisfeita.

Na literatura de gerenciamento dinâmico de energia, os modelos e técnicas propostas são geralmente especializados para a arquitetura considerada. Além disso, as soluções de gerenciamento de energia baseadas em malha aberta são apenas adequadas quando há disponibilidade de um modelo acurado de consumo de energia e desempenho do sistema. E a maioria dessas soluções é otimizada para cargas de trabalho específicas. Contudo, cargas de trabalho de computadores de propósito geral são inicialmente desconhecidas e variantes no tempo.

Assim, nesta Tese, foi abordado o problema de “como garantir um dado nível de desempenho e minimizar o consumo de energia de um processador provido de escalonamento dinâmico de tensão e frequência, através de um gerenciador de energia executado no sistema operacional, considerando que a carga de trabalho do sistema é inicialmente desconhecida e variante no tempo, a política de escalonamento de processos do sistema operacional é inicialmente desconhecida, e o consumo de energia de cada estado do processador é desconhecido”. Considerando esse problema, foram apresentadas técnicas de estimação e controle em malha fechada para garantir um determinado nível de desempenho e minimizar o consumo de energia em diferentes contextos.

No Capítulo 2, foi apresentada uma importante contribuição ao gerenciamento de energia de sistemas modelados como processos de decisão de Markov: uma técnica de estimação da carga de trabalho baseada em janelas deslizantes de múltiplos comprimentos (*JMC*). A partir

do conhecimento *a-priori* do desempenho e consumo de energia do sistema considerado, e de uma carga de trabalho estacionária, foi apresentada uma metodologia para a obtenção de um modelo utilizando processos de decisão de Markov. Para cada tipo de carga de trabalho estacionária, deve ser resolvido um problema de otimização de política, resultando numa política estacionária. Se a carga de trabalho do sistema é inicialmente desconhecida e não-estacionária, então é necessário utilizar uma técnica de estimação da carga de trabalho do sistema para que seja possível escolher a política estacionária otimizada para a carga de trabalho atual. Para realizar a estimação da carga de trabalho, foi proposta a técnica baseada em janelas deslizantes de múltiplos comprimentos. Verificou-se que essa técnica superou uma abordagem anterior baseada em janelas de comprimento fixo. No entanto a abordagem estocástica para gerenciamento de energia, quanto à resolução do problema de otimização de política, requer o conhecimento *a-priori* do desempenho e consumo de energia do sistema para derivar o modelo em processos de decisão de Markov. Verificou-se também que não se garante que a solução do problema de otimização respeita a restrição de penalidade de desempenho para toda carga de trabalho, e assim, pode não oferecer uma economia de energia ótima.

A partir da necessidade de garantir o atendimento aos níveis de desempenho desejados, foi apresentada no Capítulo 3 outra contribuição deste trabalho: a aplicação de controle em malha fechada para gerenciamento de energia. Discutiu-se a obtenção de um modelo dinâmico de tempo discreto a partir do modelo em processos de decisão de Markov do sistema com gerenciamento de energia, por meio de técnicas de identificação de sistemas. A identificação do modelo do sistema foi realizada a partir da coleta de dados de entrada e saída em um experimento utilizando o pior caso para a carga de trabalho, e a aplicação de um algoritmo de mínimos quadrados. Com base nesse modelo, foi realizado o controle em malha fechada de um indicador de penalidade de desempenho do sistema com gerenciamento de energia, resultando em políticas de energia que superaram as abordagens estocásticas apresentadas no Capítulo 2.

Contudo, na técnica apresentada no Capítulo 3, a identificação do modelo do sistema e o projeto do controlador ainda foram realizados utilizando-se o modelo de processos de decisão de Markov do sistema, que precisa ser conhecido *a priori*. Nesta técnica, é utilizado apenas um modelo do sistema obtido a partir do pior caso carga de trabalho, o qual pode ser consideravelmente diferente se o sistema é submetido a cargas de trabalho variantes no tempo. Uma conclusão imediata foi que os resultados de desempenho e consumo de energia de técnicas de controle em malha fechada para gerenciamento de energia seriam aperfeiçoados por meio da adaptação contínua dos parâmetros do controlador de acordo com a carga de trabalho do sistema. Então, no Capítulo 4, foi aplicada a abordagem de Controle Adaptativo para realizar a identificação do modelo do sistema em tempo de execução por meio de um algoritmo de mínimos quadrados recursivos. Para esta abordagem, não foi necessário um modelo *a priori* do

sistema com gerenciamento de energia. Assim, foi possível reprojeter o controlador, tal como um controlador linear geral ou um proporcional-integral em tempo de execução, de acordo com as flutuações da carga de trabalho. As principais contribuições foram: (i) ao contrário de trabalhos anteriores, um modelo acurado de consumo de energia e desempenho do processador não é necessário, porque a política de gerenciamento de energia apresentada foi baseada em técnicas de modelagem, estimação e controle em malha fechada, que permitiram o controle efetivo do desempenho e a minimização do consumo de energia; (ii) um controlador linear geral e um controlador proporcional-integral foram projetados de acordo com as especificações de gerenciamento dinâmico de energia; (iii) a técnica apresentada de controle adaptativo foi aplicada a um sistema com cargas de trabalho inicialmente desconhecidas e variantes no tempo.

5.1 Trabalhos futuros

A partir desta Tese, podem ser derivadas várias linhas de trabalhos futuros:

- A implementação da técnica de estimação da carga de trabalho baseada em janelas deslizantes de múltiplos comprimentos para um sistema específico, tal como servidores ou discos rígidos, deve permitir a definição de um conjunto ótimo de comprimentos de janela.
- A técnica proposta de controle adaptativo foi implementada como uma aplicação em espaço de usuário, mas ela foi comparada a políticas de energia implementadas no núcleo do sistema operacional, em que há menor custo em desempenho e potência para medir a utilização do processador, passar comandos para mudar a frequência de operação do processador, e menos chaveamento de contexto. Assim, os resultados experimentais apresentados nesse Trabalho podem ser aprimorados por meio da implementação da técnica proposta no núcleo do sistema operacional.
- Na plataforma experimental utilizada, a variável controlada é a utilização do processador, porque este é o indicador usada pelas políticas de energia implementadas no núcleo do Linux. Contudo é também possível aplicar a técnica introduzida de controle adaptativo a sistemas com outros tipos de variáveis controladas, tais como orçamento de potência (*power budget*) ou vazão de processamento.
- A técnica introduzida de controle adaptativo foi implementada numa plataforma experimental composta de um único núcleo. Essa técnica pode ser diretamente aplicada a sistemas com múltiplos núcleos em que as malhas de controle associadas aos núcleos são

desacopladas. Caso contrário, a estimação e o projeto do controlador devem ser implementados considerando o caso de múltiplas entradas e múltiplas saídas (*MIMO*).

- A entrada de referência pode ser selecionada com base no nível de penalidade de desempenho aceitável para a carga de trabalho atual. Assim, um esquema para selecionar a entrada de referência com base na aplicação beneficiaria de: (i) reduzido consumo de potência para cargas de trabalho com baixas demandas de desempenho; (ii) e desempenho superior para cargas de trabalho com altas demandas de desempenho.
- Neste trabalho, considerando que a estimação dos parâmetros do processo é realizada em malha fechada, indicadores tais como a excitação persistente e o erro de controle médio quadrático devem ser considerados para assegurar o desempenho tanto da estimação dos parâmetros quanto do controlador. Assim, uma direção interessante de pesquisa é a seleção dinâmica do fator de esquecimento aplicado ao algoritmo de mínimos quadrados recursivos em tempo de execução considerando o caso de cargas de trabalho variantes no tempo.
- O projeto do controlador considerando a minimização de ambos o erro de controle e a frequência do processador é desafiador, devido ao fato de que o modelo do sistema com gerenciamento de energia é variante no tempo. Assim, o projeto por alocação de pólos pode ser aprimorado considerando esses indicadores.
- Finalmente, considerando o uso de tecnologias como *clock gating* e *power gating* em conjunto com escalonamento dinâmico de tensão e frequência, é necessário que o gerenciador de energia responsável pelas decisões de tensão e frequência não seja executado quando o processador está ocioso, e assim possam ser utilizados *clock gating* e *power gating* durante períodos de tempo mais longos. No entanto, essa estratégia sugere que as decisões de tensão e frequência sejam: (i) dirigidas a tempo (*time-driven*), enquanto o processador está com carga de trabalho (a execução periódica do gerenciador de energia para estimação da carga de trabalho, execução da política de gerenciamento de energia, e alteração do estado de energia do processador); e (ii) dirigidas a eventos (*event-driven*), enquanto o processador está ocioso (o gerenciador de energia deve permanecer ocioso, sendo somente executado se houver uma interrupção a partir do sistema operacional).

Apêndice A

Projeto do controlador linear geral

Neste apêndice é apresentado o projeto do controlador linear geral em (3.9) por meio da alocação de pólos de grau mínimo (MDPP) [45]. A expressão para o sistema em malha fechada é apresentada em (3.10). Vamos fatorizar o polinômio B como $B = B^+B^-$, onde B^+ é um polinômio mônico cujos zeros são estáveis e tão bem amortecidos que o controlador pode cancelá-los, e B^- tem zeros instáveis e pobremente amortecidos que não podem ser cancelados. Considerando essas condições, B^- deve ser um fator de B_m . Então nós fatorizamos $B_m = B^-B'_m$. De (3.13), A_m deve ser um fator de A_c . Além disso B^+ é cancelado. Então $A_c = A_oA_mB^+$.

Devido ao fato de B^+ ser um fator de ambos B e A_c , pode ser visto de (3.11) que B^+ é também um fator de R . Então $R = R'B^+$. A equação de Diofanto pode ser reescrita como (A.1). E usando $B = B^+B^-$, $B_m = B^-B'_m$, e $A_c = A_oA_mB^+$ em (3.13), conseguimos $T = A_oB'_m$.

$$R'A + B^-S = A_oA_m = A'_c \quad (\text{A.1})$$

A solução para a equação de Diofanto deve satisfazer as condições $\deg S \leq \deg R$ e $\deg T \leq \deg R$ para que o controlador seja causal. Sempre existe uma solução tal que $\deg S < \deg A$, que é denominada a *solução de grau mínimo* para a equação de Diofanto [45]. Esse procedimento de projeto é exemplificado a seguir.

Exemplo 7 *Seja o sistema de entrada única e saída única (SISO) em (3.17) e a especificação para a malha fechada em (3.18). Dado que $B(q) = b_1$, vamos escolher $B^+(q) = 1$ e $B^-(q) = b_1$, então*

$$\begin{aligned} B_m = B^-B'_m &\Rightarrow 1 = b_1B'_m \Rightarrow B'_m = \frac{1}{b_1} \\ A_c = A_oA_mB^+ &\Rightarrow q^2 = A_oq^2 \cdot 1 \Rightarrow A_o = 1 \end{aligned}$$

$$T = A_o B'_m = \frac{1}{b_1}$$

Dado que $\deg R = \deg A_c - \deg A = 2 - 1 = 1$, então $R(q) = r_0 q + r_1$ e

$$R = R' B^+ \Rightarrow R'(q) = r_0 q + r_1$$

Dado que $\deg S < \deg A \Rightarrow \deg S = 0$ e $S(q) = s_1$. Repondo $A(q) = q + a_1$, $B^-(q) = b_1$, $R'(q) = r_0 q + r_1$ e $S(q) = s_1$ em (A.1), obtemos (A.2) e então (A.3).

$$(r_0 q + r_1)(q + a_1) + b_1 s_1 = q^2 \Rightarrow \quad (\text{A.2})$$

$$r_0 q^2 + (r_1 + a_1 r_0) q + (a_1 r_1 + b_1 s_1) = q^2 \quad (\text{A.3})$$

Logo $r_0 = 1$, $r_1 = -a_1$, e $s_1 = a_1^2/b_1$. Finalmente $R(q) = R' B^+ = r_0 q + r_1 = q - a_1$, $S(q) = a_1^2/b_1$, e $T(q) = 1/b_1$.

Exemplo 8 Seja o sistema de entrada única e saída única (SISO) em (3.17) e a especificação para a malha fechada em (A.4).

$$\frac{BT}{RA + BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} = \frac{\beta_1 q + \beta_0}{q^2 + \alpha_1 q + \alpha_0} \quad (\text{A.4})$$

Dado que $B(q) = b_1$, vamos escolher $B^+(q) = 1$ e $B^-(q) = b_1$, então

$$B_m = B^- B'_m \Rightarrow \beta_1 q + \beta_0 = b_1 B'_m \Rightarrow B'_m \Rightarrow \frac{\beta_1}{b_1} q + \frac{\beta_0}{b_1}$$

$$A_c = A_o A_m B^+ \Rightarrow q^2 + \alpha_1 q + \alpha_0 = A_o (q^2 + \alpha_1 q + \alpha_0) \cdot 1 \Rightarrow A_o = 1$$

$$T = A_o B'_m = \frac{\beta_1}{b_1} q + \frac{\beta_0}{b_1}$$

Dado que $\deg R = \deg A_c - \deg A = 2 - 1 = 1$, então $R(q) = r_0 q + r_1$ e

$$R = R' B^+ \Rightarrow R'(q) = r_0 q + r_1$$

Dado que $\deg S < \deg A \Rightarrow \deg S = 0$ e $S(q) = s_1$. Repondo $A(q) = q + a_1$, $B^-(q) = b_1$, $R'(q) = r_0 q + r_1$ e $S(q) = s_1$ em (A.1), obtemos (A.5) e então (A.6).

$$(r_0 q + r_1)(q + a_1) + b_1 s_1 = (q^2 + \alpha_1 q + \alpha_0) \Rightarrow \quad (\text{A.5})$$

$$r_0 q^2 + (r_1 + a_1 r_0) q + (a_1 r_1 + b_1 s_1) = (q^2 + \alpha_1 q + \alpha_0) \quad (\text{A.6})$$

Logo $r_0 = 1$, $r_1 = \alpha_1 - a_1$, e $s_1 = (\alpha_0 - \alpha_1 a_1 + a_1^2)/b_1$. Finalmente $R(q) = q + (\alpha_1 - a_1)$, $S(q) = (\alpha_0 - \alpha_1 a_1 + a_1^2)/b_1$, e $T(q) = \frac{\beta_1}{b_1} q + \frac{\beta_0}{b_1}$.

Apêndice B

Escolha do fator de esquecimento

Neste apêndice, é apresentado o procedimento para escolha do fator de esquecimento utilizado para a estimação de parâmetros do processo, apresentada na Seção 4.3.2.4. Um conjunto de experimentos foi realizado para avaliar os indicadores $V(\lambda)$ em (4.34) e $\rho_1(\lambda)$ em (4.33) (calculado por meio da função *feasp* do *software* MatLab, utilizada para resolução de inequação matriciais lineares) como funções do fator de esquecimento λ usando: (i) a carga de trabalho como a reprodução, por meio do tocador *totem* [2], do vídeo de domínio público *citizenengineermov*.mov [4]; (ii) o gerenciador de energia sendo a técnica proposta de controle adaptativo com a entrada de referência $u_C(t) = 0,75, \forall t$, e a modulação por largura de pulso (PWM) implementada no bloco *atuador*; (iii) os indicadores medidos do instante inicial $t_i = 70$ períodos de amostragem (após a inicialização do tocador *totem*) ao instante final $t_f = 1670$ (após aproximadamente 176 segundos de reprodução do vídeo); (iv) o valor m em (4.33) igual a 100; (v) os valores constantes $\lambda = 0,900, 0,905, 0,910, \dots, 0,995$; (vi) para cada valor constante λ , o experimento foi repetido 10 vezes. O limitante inferior da excitação persistente e o erro de controle médio quadrático são apresentados na Fig. B.1 juntamente com barras de erro verticais representando os respectivos desvios padrões ($\pm\sigma$). Deseja-se escolher o parâmetro λ tal que a excitação persistente seja maximizada, e o erro de controle médio quadrático seja minimizado. Quando λ aumenta, o limitante inferior da excitação persistente aumenta, mas o erro de controle médio quadrático também aumenta. Assim, verifica-se que esses indicadores são conflitantes para a escolha do fator de esquecimento. O fator de esquecimento $\lambda = 0,935$ foi escolhido por representar uma solução de compromisso entre o limitante inferior da excitação persistente e o erro de controle médio quadrático.

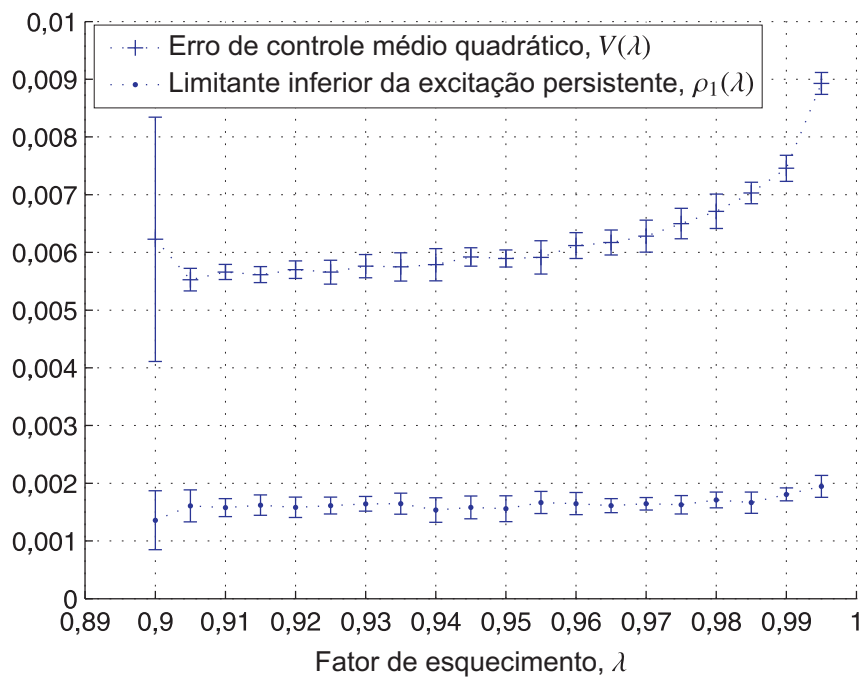


Figura B.1: O limitante inferior da excitação persistente e o erro de controle médio quadrático medidos para $\lambda = 0,900, 0,905, 0,910, \dots, 0,995$.

Referências Bibliográficas

- 1 CHUNG, E.-Y.; BENINI, L.; BOGLIOLO, A.; LU, Y.-H.; MICHELI, G. D. Dynamic power management for nonstationary service requests. *IEEE Trans. Comput.*, IEEE Computer Society, Washington, DC, USA, v. 51, n. 11, p. 1345–1361, 2002. ISSN 0018-9340.
- 2 COOPER, P.; DRAKE, L.; THIESSEN, R.; BULTJE, R.; MÜLLER, T.-P. *Totem*. ago. 2011. Disponível em: <<http://projects.gnome.org/totem/>>.
- 3 LEVY, A. *Werther (Aurora Luminosa: música brasileira no alvorecer do séc. XX)*. out. 2011. Disponível em: <<http://www.dominiopublico.gov.br>>.
- 4 FRIED, L.; TORRONE, P. *Citizen Engineer: Volume 01*. ago. 2010. Disponível em: <<http://www.clearbits.net/torrents/180-citizen-engineer>>.
- 5 HENSON, V. *ebizzy*. Agosto 2010. Acessado em 22 de Agosto de 2010. Disponível em: <<http://sourceforge.net/projects/ebizzy/>>.
- 6 SHEARER, F. *Power Management in Mobile Devices*. 1. ed. Burlington, USA: Newnes, 2008. ISBN 978-0-7506-7958-9.
- 7 WANG, Z.; ZHU, X.; MCCARTHY, C.; RANGANATHAN, P.; TALWAR, V. Feedback control algorithms for power management of servers. In: *Conf. Rec. FeBid'08*. Annapolis, MD: ACM, 2008.
- 8 LU, Y.-H.; MICHELI, G. D. Comparing system-level power management policies. *IEEE Design & Test of Computers*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 18, n. 2, p. 10–19, 2001. ISSN 0740-7475.
- 9 LU, Y.-H.; BENINI, L.; MICHELI, G. D. Operating-system directed power reduction. In: *Conf. Rec. ISLPED '00*. New York, NY, USA: ACM, 2000. p. 37–42. ISBN 1-58113-190-9.
- 10 BENINI, L.; BOGLIOLO, A.; PALEOLOGO, G.; MICHELI, G. D. Policy optimization for dynamic power management. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, v. 18, n. 6, p. 813–833, jun. 1999. ISSN 0278-0070.

- 11 PETTIS, N. E. *Automatic configuration and selection of power management policies*. Tese (Doutorado), West Lafayette, IN, USA, 2008. Orientador-Lu, Yung-Hsiang.
- 12 Power aware computing. In: GRAYBILL, R.; MELHEM, R. (Ed.). New York, NY, USA: Kluwer Academic Publishers, 2002. p. 319. ISBN 978-0-306-46786-8.
- 13 PANDA, P.; SHRIVASTAVA, A.; SILPA, B.; GUMMIDIPUDI, K. *Power-efficient System Design*. New York, NY, USA: Springer, 2010. 12 p. ISBN 9781441963871.
- 14 RONG, P.; PEDRAM, M. Battery-aware power management based on markovian decision processes. In: *Conf. Rec. ICCAD'02*. New York, NY, USA: ACM Press, 2002. p. 707–713. ISBN 0-7803-7607-2.
- 15 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N.; GORGÔNIO, K. Técnica de gerenciamento dinâmico de energia orientada à autonomia da bateria para sistemas embarcados. In: *Conf. Rec. CBA'08*. Juiz de Fora, MG, Brasil: Anais do CBA 2008, 2008.
- 16 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N. Workload estimation for power management. In: *Conf. Rec. ICCE'09*. Las Vegas: IEEE Consumer Electronics Society, 2009. p. 7.2–4.
- 17 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N. Stochastic learning-based weak estimation for dynamic power management. In: *Conf. Rec. CBA'10*. Bonito, MS, Brasil: Anais do CBA 2010, 2010.
- 18 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N. Multisize sliding window in workload estimation for dynamic power management. *IEEE Transactions on Computers*, v. 59, n. 12, p. 1625–1639, 2010. ISSN 0018-9340. Disponível em: <<http://dx.doi.org/10.1109/TC.2010.90>>.
- 19 LUIZ, S. O. D.; LIMA, A. M. N.; PERKUSICH, A. Modeling, estimation and control for power management in embedded systems. In: *Conf. Rec. DINCON'09*. Bauru, SP, Brasil: Anais do DINCON'09, 2009.
- 20 BRODOWSKI, D. *CPUFreq Governors*. 2010. Acessado em 22 de Agosto de 2010. Disponível em: <<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.33.1.tar.bz2>>.
- 21 MICROSOFT. *Processor Power Management in Windows 7 and Windows Server 2008 R2*. Redmond, WA, USA, 2010. Disponível em: <<http://www.microsoft.com>>.
- 22 QIN, W.; WANG, Q. Modeling and control design for performance management of web servers via an lpv approach. *IEEE Trans. Control Syst. Technol.*, v. 15, n. 2, p. 259–275, mar. 2007. ISSN 1063-6536.

- 23 SRIDHARAN, R.; GUPTA, N.; MAHAPATRA, R. Feedback-controlled reliability-aware power management for real-time embedded systems. In: *Conf. Rec. DAC'08*. New York, NY, USA: ACM, 2008. p. 185–190. ISBN 978-1-60558-115-6.
- 24 WANG, X.; FU, X.; LIU, X.; GU, Z. Power-aware cpu utilization control for distributed real-time systems. In: *Conf. Rec. RTAS'09*. Washington, DC, USA: IEEE Computer Society, 2009. p. 233–242. ISBN 978-0-7695-3636-1.
- 25 OGRAS, U. Y.; MARCULESCU, R.; MARCULESCU, D. Variation-adaptive feedback control for networks-on-chip with multiple clock domains. In: *Conf. Rec. DAC'08*. New York, NY, USA: ACM, 2008. p. 614–619. ISBN 978-1-60558-115-6.
- 26 TIAN, Y.-C.; XIA, F.; SUN, Y.; DONG, J. Control-theoretic dynamic voltage scaling for embedded controllers. *Computers & Digital Techniques, IET*, v. 2, n. 5, p. 377–385, set. 2008. ISSN 1751-8601.
- 27 LU, Z.; LACH, J.; STAN, M.; SKADRON, K. Reducing multimedia decode power using feedback control. *Conf. Rec. ICCD'03*, p. 489–496, out. 2003. ISSN 1063-6404.
- 28 MINERICK, R. J.; FREEH, V. W.; KOGGE, P. M. Dynamic power management using feedback. In: *Conf. Rec. COLP'02*. Charlottesville, Va: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2002. p. 6–1 – 6–10.
- 29 DEVADAS, V.; AYDIN, H. On the interplay of voltage/frequency scaling and device power management for frame-based real-time embedded applications. *IEEE Transactions on Computers*, v. 61, n. 1, p. 31–44, 2012. ISSN 0018-9340.
- 30 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N.; SILVA, J. J.; ALMEIDA, H.; SILVA, G. R. B. Identification and control for processor power management. In: *Proc. Conf. ICCE'12*. Piscataway: IEEE Consumer Electronics Society, 2012. Disponível em: <<http://dx.doi.org/10.1109/ICCE.2012.6161924>>.
- 31 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N.; SILVA, J. J.; ALMEIDA, H. System identification and energy-aware processor utilization control. *IEEE Transactions on Consumer Electronics*, v. 58, n. 1, p. 32–37, fev. 2012. ISSN 0098-3063. Disponível em: <<http://dx.doi.org/10.1109/TCE.2012.6170052>>.
- 32 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N. Adaptive control for power management. In: *Proc. Conf. ICCA'11*. Santiago, Chile: IEEE Control Systems Society, 2011. Disponível em: <<http://dx.doi.org/10.1109/ICCA.2011.6138009>>.

- 33 THEOCHAROUS, G.; MANNOR, S.; SHAH, N.; GANDHI, P.; KVETON, B.; SIDDIQI, S.; YU, C.-H. Machine learning for adaptive power management. *Intel Technology Journal*, v. 10, n. 4, p. 299–312, 2006. ISSN 1535-864X. Disponível em: <<http://download.intel.com/technology/itj/2006/v10i4/v10-i4-art05.pdf>>.
- 34 CHEUNG, Y.-M.; XU, L. An rpcl-based approach for markov model identification with unknown state number. *IEEE Signal Processing Letters*, v. 7, n. 10, p. 284–287, out. 2000. ISSN 1558-2361.
- 35 QIU, Q.; TAN, Y.; WU, Q. Stochastic modeling and optimization for robust power management in a partially observable system. In: *Conf. Rec. DATE'07*. Nice Acropolis, France: EDAA, 2007. p. 1–6.
- 36 TAN, Y.; QIU, Q. A framework of stochastic power management using hidden markov model. In: *Conf. Rec. DATE'08*. Munich, Germany: EDAA, 2008. p. 92–97.
- 37 REN, Z.; KROGH, B.; MARCULESCU, R. Hierarchical adaptive dynamic power management. *IEEE Transactions on Computers*, v. 54, n. 4, p. 409–420, abr. 2005. ISSN 0018-9340.
- 38 MATHWORKS, I. T. *linprog*. 2008. Disponível em: <<http://www.mathworks.com>>.
- 39 MEYER, P. L. *Probabilidade: aplicações à estatística*. 2. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 1983. ISBN 85-216-0294-4.
- 40 OOMMEN, B. J.; RUEDA, L. Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments. *Pattern Recognition*, v. 39, n. 3, p. 328–341, 2006. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com>>.
- 41 HWANG, C.-H.; WU, A. C.-H. A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.*, ACM, New York, NY, USA, v. 5, n. 2, p. 226–241, 2000. ISSN 1084-4309.
- 42 CHUNG, E.-Y.; BENINI, L.; MICHELI, G. D. Dynamic power management using adaptive learning tree. In: *Conf. Rec. ICCAD'99*. Piscataway, NJ, USA: IEEE Press, 1999. p. 274–279. ISBN 0-7803-5832-5.
- 43 WANG, Y.; MA, K.; WANG, X. Temperature-constrained power control for chip multiprocessors with online model estimation. *SIGARCH Comput. Archit. News*, ACM, New York, NY, USA, v. 37, n. 3, p. 314–324, 2009. ISSN 0163-5964.

- 44 LJUNG, L. *System Identification: Theory for the User*. 2. ed. Englewoods Cliffs, New Jersey 07632: Prentice Hall PTR, 1998. Hardcover. ISBN 0138816409.
- 45 ÅSTRÖM, K. J.; WITTENMARK, B. *Adaptive Control*. 2. ed. New Jersey: Prentice Hall, 1994. ISBN 978-0201558661.
- 46 ÅSTRÖM, K. J.; WITTENMARK, B. *Computer-Controlled Systems: Theory and Design*. 3. ed. New Jersey: Prentice Hall, 1997. ISBN 0-13-314899-8.
- 47 BENINI, L.; BOGLIOLO, A.; CAVALLUCCI, S.; RICCÓ, B. Monitoring system activity for os-directed dynamic power management. In: *Conf. Rec. ISLPED'98*. New York, NY, USA: ACM Press, 1998. p. 185–190. ISBN 1-58113-059-7.
- 48 NAVEH, A.; ROTEM, E.; MENDELSON, A.; GOCHMAN, S.; CHABUKSWAR, R.; KRISHNAN, K.; KUMAR, A. Power and thermal management in the intel core duo processor. *Intel Technology Journal*, v. 10, n. 2, p. 109–122, maio 2006. ISSN 1535-766X. Disponível em: <<http://developer.intel.com>>.
- 49 PALLIPADI, V.; STARIKOVSKIY, A. The ondemand governor: past, present and future. In: *Conf. Rec. Linux Symposium*. Ottawa, Ontario, Canada: Linux Symposium, 2006. v. 2, p. 223–238.
- 50 SRINIVASAN, V.; SHENOY, G. R.; VADDAGIRI, S.; SARMA, D.; PALLIPADI, V. Energy-aware task and interrupt management in linux. In: *Conf. Rec. 2008 Linux Symposium*. Ottawa, CN: Proceedings of the Linux Symposium, 2008. v. 2.
- 51 HEWLETT-PACKARD; INTEL; MICROSOFT; PHOENIX; TOSHIBA. *Advanced Configuration and Power Interface Specification*. Palo Alto, CA, USA, 2010. Disponível em: <<http://www.acpi.info/spec.htm>>.
- 52 HELLERSTEIN, J. L.; DIAO, Y.; PAREKH, S.; TILBURY, D. M. *Feedback Control of Computing Systems*. Hoboken, NJ, USA: John Wiley & Sons, 2004. ISBN 047126637X.
- 53 CHOI, K.; LEE, W.; SOMA, R.; PEDRAM, M. Dynamic voltage and frequency scaling under a precise energy model considering variable and fixed components of the system power dissipation. *Proc. Conf. ICCAD'04*, IEEE Computer Society, Los Alamitos, CA, USA, p. 29–34, 2004.
- 54 VENKATACHALAM, V.; FRANZ, M. Power reduction techniques for microprocessor systems. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 37, p. 195–237, set. 2005. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1108956.1108957>>.

- 55 Energy-aware system design: Algorithms and architectures. In: KYUNG, C.-M.; YOO, S. (Ed.). New York, NY, USA: Springer, 2011. p. 58. ISBN 9789400716780.
- 56 LEFURGY, C.; WANG, X.; WARE, M. Server-level power control. *Proc. Conf. ICAC'07*, IEEE Computer Society, Los Alamitos, CA, USA, p. 4, 2007.