Eduardo Gomes Pereira

Modelagem e Controle para Gerenciamento Dinâmico de Energia em Processadores

Campina Grande, Brasil

7 de março de 2014

Eduardo Gomes Pereira

Modelagem e Controle para Gerenciamento Dinâmico de Energia em Processadores

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do título de Graduado em Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG

Unidade Acadêmica de Engenharia Elétrica

Orientador: Antonio Marcus Nogueira Lima

Campina Grande, Brasil

7 de março de 2014

Eduardo Gomes Pereira
Modelagem e Controle para Gerenciamento Dinâmico de Energia em Processadores/ Eduardo Gomes Pereira. – Campina Grande, Brasil, 7 de março de 2014-
25 p. : il. (algumas color.) ; 30 cm.

Orientador: Antonio Marcus Nogueira Lima

Trabalho de Conclusão de Curso – Universidade Federal de Campina Grande – UFCG
Unidade Acadêmica de Engenharia Elétrica , 7 de março de 2014.

filtro preditor, gerenciamento dinâmico de energia, predição de carga de trabalho, processo de Poisson.

1. Filtro Preditor. 2. Gerenciamento Dinâmico de Energia. 3. Predição de Carga de Trabalho. 4. Processo de Poisson. I. Antonio Marcus Nogueira Lima. II. Universidade Federal de Campina Grande. III. Unidade Acadêmica de Engenharia Elétrica. IV. Modelagem e Controle para Gerenciamento Dinâmico de Energia em Processadores

Eduardo Gomes Pereira

Modelagem e Controle para Gerenciamento Dinâmico de Energia em Processadores

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do título de Graduado em Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Brasil, 06 de fevereiro de 2014:

Antonio Marcus Nogueira Lima
Orientador

Saulo Oliveira Dornellas Luiz
Convidado

Campina Grande, Brasil
7 de março de 2014

Este trabalho é dedicado à minha família

Agradecimentos

Os agradecimentos deste trabalho são direcionados:

- A Deus, pela força e perserverança, pela capacidade e sabedoria, por este mundo incrível... por tudo;
- Ao professor Antonio Marcus, pela orientação e pelas “insistências insistentes”, pela oportunidade e pela confiança;
- Ao professor Marcos Morais, pelo convite para trabalhar, pelo auxílio e esclarecimento em diversas ocasiões, e pela paciência;
- Ao professor Saulo, pelo fornecimento de seu trabalho, e pela valiosa assistência durante a realização deste trabalho;
- Ao colega Reuben, pela ajuda e pelo tempo dedicado à instalação e configuração do Linux;
- À minha família, pelo apoio e compreensão, não apenas durante esta longa trajetória, mas durante toda a minha vida;
- A todos os amigos e colegas conquistados nestes cinco anos de curso, pelo compartilhamento de alegrias e tristezas, pelo sentimento de “não estou sozinho” nos momentos difíceis, e pelos bons e memoráveis momentos vividos no decorrer desta jornada;
- A todos os professores e a todas as pessoas anônimas que, de forma direta ou indireta, contribuíram para a realização deste trabalho.

"Prediction is very difficult, especially about the future."
(Niels Bohr, 1885 - 1962)

"To infinity... and beyond!"
(Buzz Lightyear, Toy Story)

Resumo

No presente trabalho, é apresentado o desenvolvimento e implementação de uma estratégia de controle para gerenciamento dinâmico de energia em processadores, considerando a carga de trabalho estocástica e variante no tempo. A carga de trabalha é modelada como um processo de Poisson, possibilitando uma abordagem analítica para o problema de gerenciamento de energia e provendo uma boa caracterização da carga de trabalho. Um filtro preditor adaptativo é utilizado para estimar o valor mais provável de requisições da carga de trabalho no instante de tempo futuro. Com base no modelo de carga de trabalho e no valor predito de requisições, uma estratégia ótima de controle é desenvolvida. Testes e resultados experimentais são exibidos para mostrar a aplicabilidade da implementação.

Palavras-chaves: filtro preditor, gerenciamento dinâmico de energia, predição de carga de trabalho, processo de Poisson.

Abstract

In the present work, we present the development and implementation of a control strategy for dynamic power management on processors, considering the workload stochastic and time-variant. The workload is modeled as a Poisson process, enabling an analytical approach to the power management problem and providing a good characterization for the workload. An adaptive predictor filter is used to estimate the most likely value requests from the workload for the future time instant. Based on the workload model and predicted value of requisitions, a optimal control strategy is developed. Experimental tests and results are shown to show the applicability of the implementation.

Key-words: predictor filter. dynamic power management. workload prediction, Poisson process.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DVFS	Dynamic Voltage and Frequency Scaling
GDE	Gerenciamento Dinâmico de Energia
LMS	Least Mean Square
RAM	Random Access Memory

Lista de símbolos

C_s	Capacitância de chaveamento
e	Erro de predição
f	Frequência de operação do processador
p_i	Potencia consumida no período ocioso no modelo baseado em utilização
P	Potencia
$\mathbf{r_{xy}}$	Matriz de correlação cruzada dos sinais de entrada e saída do filtro preditor
$\mathbf{R_x}$	Matriz de autocorrelação de sinais de entrada do filtro preditor
t	Instante de tempo
t_0	Parâmetro auxiliar, insante de tempo inicial
t_i	Intervalo de tempo no qual o procesador fica ocioso
t_p	Intervalo de tempo no qual o procesador atende a requisições de processamento
t_t	Intervalo de tempo no qual o procesador atende a requisições de tempo
T	Período de amostragem
V	Tensão de alimentação do processador
$\mathbf{w}$	Filtro preditor
x	Sinal de saída/entrada usado para predição
$\hat{x}$	Valor predito de x
$\mathbf{x}$	Vetor de sinais de entrada do filtro preditor
λ	Taxa média, parâmetro do processo de Poisson
λ_i	Parâmetro auxiliar
λ_i^*	Nível de osciosidade de referência
λ_p	Taxa média de requisições de processamento

λ_s	Taxa média de requisições de serviços
$\hat{\lambda_s}$	Estimativa de λ_s
λ_t	Taxa média de requisições de tempo
$\hat{\lambda_t}$	Estimativa de λ_t
γ	Fator de esquecimento
μ	Fator reajuste usado na atualização do filtro

Sumário

1 Introdução

1.1 Motivação

Nos últimos tempos, a eficiência energética em sistemas computacionais se tornou uma necessidade. A autonomia de sistemas computacionais alimentados à bateria depende da capacidade dela de armazenar carga e da energia requerida para o seu funcionamento. Em sistemas ligados à rede elétrica, também é necessário diminuir os custos devido ao consumo de energia, assim como o impacto ambiental. A redução do consumo de energia sem afetar o desempenho dos sistemas computacionais se tornou um desafio.

Tradicionalmente, o gerenciamento de energia em sistemas computacionais é focado em modos de operação (e.g. *sleep*, baixo consumo, etc) (1). No entanto, diferentes aplicações podem exigir diferentes usos de recursos, resultando em ociosidade ou subutilização dos recursos e um consumo desnecessário de energia. Muitos processadores de sistemas computacionais são dispositivos integrados que suportam estratégias agressivas de gerenciamento de energia, tais como: *clock gating* (2); escalonamento dinâmico de tensão e frequência ou DVFS (*dynamic voltage and frequency scaling*) (3); e *power gating*.

Se o processador tem suporte a DVFS, é possível reduzir a frequência e tensão do processador quando este está ocioso ou subutilizado e, consequentemente, reduzir o consumo de energia com pouco ou nenhum impacto no desempenho do sistema. O gerenciamento do consumo de energia em tempo de execução é chamado de Gerenciamento Dinâmico de Energia (GDE). O GDE consiste em monitorar os estados dos componentes do sistema e determinar uma política para minimizar o consumo de energia dada uma necessidade de desempenho.

Na literatura, os modelos e técnicas propostas para o GDE são geralmente voltados para cargas de trabalho bem definidas. Porém, na maioria dos casos, a carga de trabalho é aleatória, impossibilitando a aplicação de tais técnicas. Outras técnicas propostas na literatura são dirigidas a arquiteturas específicas e bem conhecidas. Todavia, estas técnicas são específicas para dispositivos particulares, não sendo estas replicáveis a outros dispositivos. Quando se trata de aplicações a arquiteturas e cargas de trabalho gerais, o GDE normalmente é baseado em um modelo de carga de trabalho caracterizado como um processo estocástico. No entanto, esta carga de trabalho é altamente dependente das necessidades de sistema e atividades dos usuários, o que leva o processo a ser variante no tempo. Surge então um problema para aplicação de controle devido às dificuldades no processo de modelagem do sistema. Esta dificuldade resulta da falta de conhecimento ou entendimento do processo a ser controlado.

1.2 Revisão da Literatura

Na literatura, podem ser encontradas soluções otimizadas para políticas de gerenciamento de energia quando a carga de trabalho é definida a priori. No entanto, na maioria das situações reais, a carga de trabalho não é conhecida a priori, o que impossibilita o uso de tais técnicas.

Muitos autores utilizam modelos baseados em cadeias de Markov para determinar a estratégia de gerenciamento de energia (1, 4, 5, 6, 7). Nestes modelos, a base da geração da política de gerenciamento é formada pelo conjunto de estados dos dispositivos que compõem o sistema. Para tais modelos, no entanto, o uso de políticas de energia é inviável quando o número de estados é grande. Estes modelos também não são bons para prever o comportamento da carga de trabalho.

Outros autores tratam o conjunto {processador; carga de trabalho} como um sistema dinâmico (8, 9, 10). Estes autores utilizam estratégias de controle bem estabelecidas, como controladores PID (Proporcional-Integral-Derivativo) e RST, para controlar a variável de saída (usualmente utilização do processador). Xia et al. (8) usaram um controlador PI para controlar o conjunto {processador; carga de trabalho} com carga de trabalho constante. Saulo et al.(10) mostraram que os parâmetros do controlador podem ser otimizados para um conjunto {processador; carga de trabalho} com carga de trabalho constante. Com base nisso, Saulo et al. (9, 11) realizaram a identificação do conjunto {processador; carga de trabalho} e a atualização do controlador para estender a otimização dos parâmetros do controlador a cargas de trabalho variantes no tempo. No entanto, as técnicas estabelecidas considerando o conjunto {processador; carga de trabalho} como um sistema dinâmico apresentam alguns problemas: i) a consideração do conjunto {processador; carga de trabalho} como um sistema dinâmico resulta em um nível de abstração muito grande da carga de trabalho; ii) o uso de tais estratégias impossibilida ao controlador aprender sobre padrões de comportamento da carga de trabalho; e iii) sendo as estratégias de controle predominantemente baseadas no erro, não é possível fazer um agendamento do sinal de controle caso a carga apresente um padrão de comportamento de múltipla periodicidade.

1.3 Objetivos

Dadas as dificuldades de modelagem e controle do consumo de energia em sistemas computacionais, o objetivo geral deste trabalho é a realização de modelagem e controle baseados em Processo de Poisson para cargas de trabalho não estacionárias de processadores visando minimizar o consumo de energia com pouco ou nenhum impacto no desempenho do sistema.

1.4 Organização do Texto

No Capítulo 2, são apresentadas a metodologia de construção do modelo, estimação e controle de carga de trabalho. A construção do modelo de carga trabalho é realizada com base em Processo de Poisson, sendo essa uma contribuição significativa deste trabalho. Baseado no conhecimento empírico sobre o comportamento do usuário e, consequentemente, da carga de trabalho, um filtro preditor adaptativo foi construído para estimar características da carga de trabalho e usar estratégias antecipativas de controle. A construção e atualização do filtro apresentam baixíssima complexidade computacional, característica importante para gerenciamento dinâmico de energia. Por fim, o modelo de carga de trabalho, juntamente com o comportamento previsto são utilizados para geração de uma estratégia ótima de controle em termos de consumo de energia e obedecendo a restrições de desempenho e margem de segurança.

No Capítulo 3, são apresentados os testes e resultados experimentais para comprovar a aplicabilidade da implementação proposta. Inicialmente, são apresentados testes de estimação de carga de trabalho, sendos estes úteis para evidenciar a eficiência do modelo e do filtro preditor. Na sequência, são exibidos os resultados da aplicação combinada de estimação de carga de trabalho e controle da frequência de operação do processador. Tais resultados comprovam a adequação da metodologia exposta como uma ferramenta válida para realização de gerenciamento dinâmico de energia em processadores.

No Capítulo 4, é apresentada uma breve síntese do trabalho, enfatizando os aspectos mais importantes do trabalho, bem como algumas limitações. Também são manifestadas possíveis atividades e ideias para continuação ou utilização deste trabalho.

2 Modelo e Estratégia de Controle

2.1 Introdução

Todo computador é formado por três elementos básicos principais: o processador, que controla o funcionamento do computador e executas as operações de processamento de dados; a memória, que armazena dados e programas utilizados pelo processador; e módulos de entrada e saída, que possibilitam a troca de dados entre o computador e o mundo externo.

O processador é a parte de *hardware* do computador que executa operações lógicas, aritiméticas e de entrada e saída seguindo sequências estabelecidas em programas. O processador pode ser pensado como um provedor de serviços ao qual chegam requisições ao longo do tempo. O provedor de serviços processa as requisições com uma taxa de serviço. Ao chegaram novas requisições, estando o processador ocupado ou desabilitado, é possível que surja uma fila de serviços que armazena as requisições que ainda não foram atendidas.

As requisições surgem aleatoriamente de acordo com necessidades do sistema e atividades do usuário. A taxa de serviços está relacionada à frequência de operação do processador, quanto maior a frequência de operação, maior a taxa de serviços. É permitido ao usuário controlar a frequência de operação do processador (obedecendo a certos limites, obviamente), no entanto, aumentar a frequência de operação resulta em aumento do consumo de energia do processador. Sendo assim, a escolha da frequência de operação do processador deve provocar uma taxa de serviço suficientemente aceitável para não gerar atrasos no atendimento de serviços e incitar o mínimo consumo de energia possível.

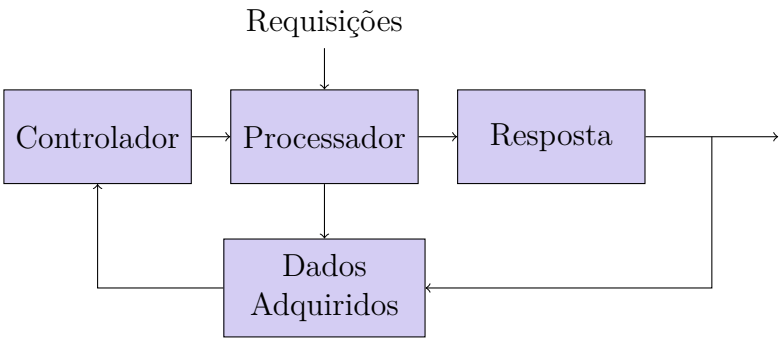

Figura 1 – Diagrama de blocos de um sistema de gerenciamento de energia.

Em GDE, para a escolha da frequência de operação do processador, geralmente são utilizadas informações do estado do processador, conforme ilustrado na Figura 1. O algorítmo de controle deve usar as informações disponíveis para determinar a frequência

de operação do processador. Este algorítmo deve apresentar baixa complexidade computacional, caso contrário, o algorítimo se tornará um grande requisitante de serviços.

Neste capítulo, serão apresentados os processos de modelagem e estimação da carga de trabalho, assim como determinada a estratégia ótima de controle em termos de consumo de energia para gerenciamento dinâmico de energia em processadores. A carga de trabalho é modelada como um conjunto de processos de Poisson, possibilitando um modelo simples e bom o suficiente para a aplicação pretendida. A carga de trabalho é estimada usando o algorítmo LSM normalizado, apresentando baixa complexidade computacional e rápida convergência. A estratégia ótima de controle é determinada partindo do modelo e considerando que a estimativa da carga de trabalho é suficientemente boa.

2.2 Modelos de Carga de Trabalho e Consumo de Energia

A maioria das estratégias de controle é gerada a partir de um modelo matemático do sistema a ser controlado. Em termos de controle, um bom modelo não precisa necessariamente prover o comportamento detalhado do sistema, um bom modelo é aquele que pode ser usado eficientemente na criação da lei de controle.

Nesta seção, é apresentado um modelo baseado em Processo Estocático de Poisson (12) para a carga de trabalho. Também são apresentados dois modelos para o consumo de energia de processadores: um baseado em frequência de operação; e outro baseado em ocupação. Os modelos apresentados são utilizados nas seções subsequêntes para elaboração da estratégia ótima de controle.

2.2.1 Modelo de Carga de trabalho

Em um intervalo de tempo, o processador passa parte deste intervalo atendendo a requisições de usuário ou de sistema, outra parte atendendo a (ou esperando) requisições de entrada e saída, e o intervalo de tempo restante ocioso. As requisições de usuário ou de sistema necessitam de processamento. Isto significa que seu tempo de atendimento é afetado pela frequência de operação do processador. Quanto mais rápida a frequência de operação do preocessador, mais rápido as requisições serão atendidas. As requisições de entrada e saída necessitam de tempo, isto é, não são afetados pela frequência de operação do processador, sua duração é a mesma independente da frequência de operação do processador. Neste trabalho, as requisições que necessitam de processamento são chamadas de requisições de processamento, e as requisições que necessitam de tempo são chamadas de requisições de tempo. Desta forma, um intervalo de tempo, T, pode ser expresso por

$$T = t_p + t_t + t_i \tag{2.1}$$

em que: t_p representa o tempo total do intervalo T que o processador ficou atendento às requisições de processamento; t_t representa o tempo total do intervalo T que o processador ficou atendento às requisições de tempo; e t_i representa o tempo total do intervalo T que o processador ficou inativo ou ocioso.

O processador é responsável por atender às requisições de processamento e de tempo. Em um sistema computacional operado por um usuário, as requisições são altamente dependentes das atividades do usuário, que são aleatórias. No entanto, é esperado que o comportamento das requisições não sofra mudanças com uma taxa média alta. Um usuário executando uma atividade tende a continuar executando esta mesma atividade. Se um usuário está assistindo a um vídeo ou editando um texto, é muito provável que, no instante de tempo seguinte, ele continue realizando esta ação. As chegadas de requisições de processamento e de tempo podem ser modeladas por um Processo Estocástico de Poisson com média de requisições sendo o parâmetro do processo.

Modelando as requições como um processo de Poisson e considerando que as requisições de tempo e processamento necessitam de um tempo para serem atendidas, dividindo a Equação 2.1 por T, é obtida

$$1 = \frac{t_p}{T} + \frac{t_t}{T} + \frac{t_i}{T} = \lambda_p^{'} + \lambda_t + \lambda_i \tag{2.2}$$

na qual: $\lambda_p^{'} = t_p/T$ corresponde a média de requisições de tempo de processamento; $\lambda_t = t_t/T$ corresponde à média de requisições de tempo; e $\lambda_i = t_i/T$ é um parâmetro auxiliar, visto que não existem requisições de tempo ocioso.

As requisições de processamento são antedidas mais rapidamente a uma frequência de operação do processador mais alta. Assim, $\lambda_p^{'}$ pode ser descrito por

$$\lambda_p^{'} = \frac{\lambda_p}{f} \tag{2.3}$$

na qual: λ_p representa as requisições de processamento; e f corresponde à frequência de operação do processador. Com isso, a Equação 2.2 pode ser reescrita como

$$1 = \frac{\lambda_p}{f} + \lambda_t + \lambda_i \tag{2.4}$$

que apresenta explicitamente a frequência de operação do processador.

2.2.2 Modelo de Consumo de Energia

Existem vários modelos para o consumo de energia do processador. Os modelos mais comuns são: o modelo baseado em frequência de operação, no qual, o consumo está diretamente relacionado à frequência de operação do processador, e sua utilização não é levada em consideração; e o modelo baseado em utilização, no qual, o consumo depende não apenas da frequência de operação do processador, mas também dos intervalos

de tempo utilizados para atender às requisições. Nesta subseção, são discutidos e apresentados ambos os modelos, mais adiante, estes modelos são utilizados na definição de estratégias ótimas de controle em termos de consumo de energia.

2.2.2.1 Modelo Baseado em Frequência de Operação

De acordo com o modelo baseado em frequência de operação, a taxa de consumo de energia dinâmica depende da frequência de operação do processador f, da capacitância de chaveamento C_s, e da tensão de alimentação V, podendo ser descrita pela seguinte equação (13)

$$P = C_s f V^2 \tag{2.5}$$

Assim, mantendo a capacitância de chaveamento e a tensão de alimentação constantes, reduzir a frequência de operação do processador resulta em menor consumo de energia.

É importante ressaltar que para frequências de operação mais elevadas existe uma tensão de alimentação mínima requerida para o funcionamento adequado do processador. Deste modo, a escolha de determinada frequência de operação implica em uma escolha de tensão de alimentação para o processador.

2.2.2.2 Modelo Baseado em Utilização

De acordo com o modelo baseado em utilização, a taxa de consumo de energia no intervalo ocioso independe da frequência de operação do processador. No entanto, o consumo durante os intervalos de tempo atendendo a requisições apresenta uma relação exponencial com a frequência de operação. Assim, a energia consumida em um intervalo de tempo $T = t_p + t_t + t_i$ é expressa por

$$E = C_s f^n \left(t_p + t_t\right) + p_i t_i \tag{2.6}$$

Dividindo pelo intervalo de tempo T, a taxa de consumo de energia é expressa por

$$P = C_s f^n \left(\lambda_p + \lambda_t\right) + p_i \lambda_i \tag{2.7}$$

2.3 Estimação de Carga de Trabalho

Conforme mencionado anteriormente, o processador é responsável por atender às requisições de processamento e de tempo, que são altamente dependentes das atividades do usuário. É esperado que o comportamento das requisições não sofra mudanças drásticas com uma taxa média alta, de forma que as requisições em um intervalo de tempo apresentam alta correlação com as requisições dos intervalos de tempo anteriores e/ou apresentem um comportamento de múltipla periodicidade[1].

[1] A maior parte dos processos opera em *loops*, fazendo requisições periódicas regularmente.

Com uma boa estimativa da chegada de requisições, é possível determinar um bom valor para a frequência de operação do processador, objetivando minimizar o consumo de energia. Para fazer esta estimativa, o clássico estimador de máxima verossimilhança para um processo de Poisson (14) pode ser utilizado. Considerando o período de amostragem unitário, o estimador de máxima verossimilhança é

$$\lambda(t) = \frac{1}{t - t_0} \sum_{\tau=t_0}^{t} X(\tau) \tag{2.8}$$

no qual: $X(\tau)$ corresponde à quantidade de requisições que chegaram no intervalo $[\tau - 1, \tau)$. Este estimador pode sofrer várias adaptações, como usar janelas de tempo pequenas ou utilizar recursividade para reduzir a complexidade computacional. No entanto, dependendo do comportamento, tal estimador necessita de uma janela de tempo grande para um bom funcionamento. Ainda, o estimador descrito na Equação 2.8 não funciona bem para comportamentos descritos por funções, a exemplo de comportamentos de múltipla periodicidade. Dado um comportamento que apresenta alta correlação com estados passados, a melhor forma de estimar valores futuros é usar um filtro preditor baseado nos estados passados.

2.3.1 Filtro de Wiener

Um bom filtro preditor é o Filtro de Wiener (12, 15, 16), utilizado para determinar uma estimativa de um processo estocático por meio de um filtro linear, considerando o processo estacionário. A saída do filtro preditor de ordem n é determinada por meio de

$$\hat{y}(t) = \sum_{i=1}^{n} w(i)\, x(t - i) \tag{2.9}$$

com erro de predição

$$e(t) = y(t) - \hat{y}(t) \tag{2.10}$$

onde: w representa um parâmetro do filtro e x representa dados de entrada do filtro. Quando utilizado para prever um comportamento estocástio, os dados de entrada do filtro são realizações passadas do processo, e a saída do filtro é a melhor predição para uma realização futura. Sendo assim, uma forma mais adequada para a Equação 2.10 é

$$\hat{x}(t) = \sum_{i=1}^{n} w(i)\, x(t - i) \tag{2.11}$$

com $\hat{x}(t)$ sendo uma estimativa de x com base em valores passados de x. Uma representação do filtro de Wiener é apresentada na Figura 2.

O filtro de Wiener de ordem n pode ser obtido usando a equação

$$\mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-\mathbf{1}} \mathbf{r}_{\mathbf{xy}} \tag{2.12}$$

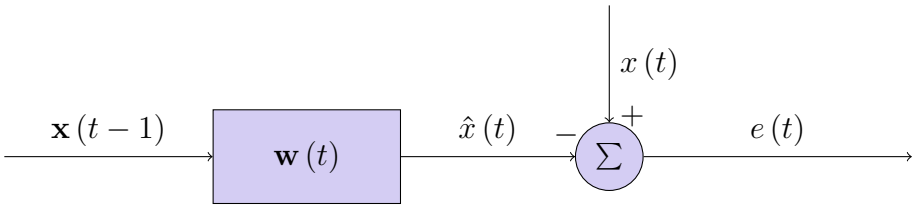

Figura 2 – Representação do Filtro de Wiener.

onde

$$\mathbf{R_x} = \begin{bmatrix} R_0 & R_1 & R_2 & \dots & R_{n-1} \\ R_1 & R_0 & R_1 & \dots & R_{n-2} \\ R_2 & R_1 & R_0 & \dots & R_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{n-1} & R_{n-2} & R_{n-3} & \dots & R_0 \end{bmatrix} \tag{2.13}$$

representa a matriz de autocorrelação dos sinais de entrada; e

$$\mathbf{r_{xy}} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} \tag{2.14}$$

representa o vetor de correlação cruzada entre os sinais de entrada e saída.

2.3.2 Filtro Adaptativo

Para determinar o filtro por meio da Equação 2.12, é necessário primeiro a obtenção de $\mathbf{R_x}$ e $\mathbf{r_{xy}}$. Esta determinação apresenta algumas desvantagens: é necessário um grande conjunto de dados; exige alta complexidade computacional; e o processo deve ser estacionário. O uso de filtros adaptativos possibilita contornar estas desvantagens. Nos filtros adaptativos, a aquisição dos dados e a determinação do filtro são realizadas ao mesmo tempo. Uma representação de um filtro adaptativo é exibida na Figura 3.

Para determinar os parâmetros do filtro dinâmicamente, alguns métodos podem ser utilizados, tais como o Método do Gradiente Determinístico (15); o Método de Newton (15); ou o método dos Mínimos Quadrádos Recursivo (15, 17). No entanto, tais métodos ainda apresentam uma complexidade computacional parcialmente elevada. O algorítimo LMS (Least Mean Square) apresenta baixa complexidade computacional, sendo ideal para a aplicação pretendida (15).

2.3.2.1 Algorítimo LMS Normalizado

O LMS (Least Mean Square) é um algorítmo de baixa complexidade computacional, cujo resultado converge na média para o filtro de Wiener (15). Uma variação do LMS

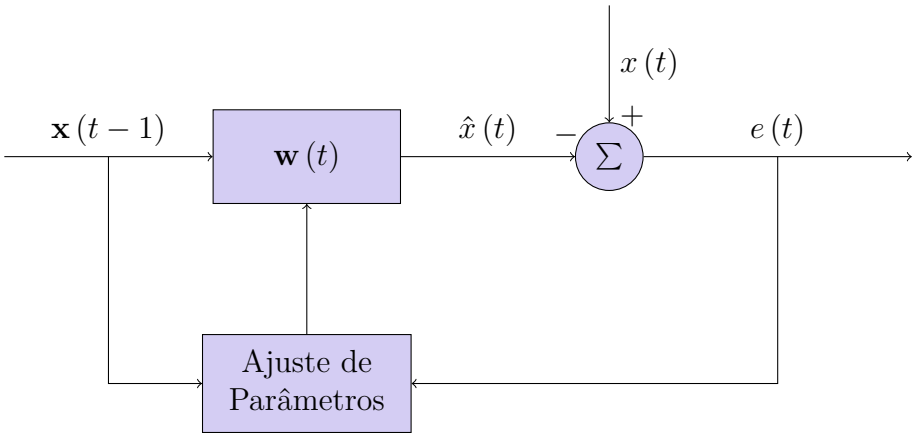

Figura 3 – Representação de um Filtro Preditor Adaptativo. O bloco "Ajuste de Parâmetros" é responsável por atualizar os parâmetros do filtro frequentemente.

é o LMS normalizado, que apresenta uma convergência mais lenta, porém um erro médio quadrático menor. A determinação do filtro por meio do Algorítimo LMS Normalizado é

$$\mathbf{w}(\mathbf{n}+\mathbf{1}) = \mathbf{w}(\mathbf{n}) + \frac{\mu}{(1-\gamma) + \mathbf{x}^T(n)\,\mathbf{x}(n)}\mathbf{x}(n)\,e(n) \tag{2.15}$$

onde:

$$e(t) = x(t) - \hat{x}(t) \tag{2.16}$$

é o erro de predição no instante de tempo t; γ é o fator de esquecimento; e μ é o fator de erro de reajuste, escolhido para minimização do erro médio quadrático como

$$\mu = \frac{1}{2\mathbf{x}^T(t)\,\mathbf{x}(t)} \tag{2.17}$$

2.4 Escolha da Frequência de Operação

No gerenciamento de energia de um processador, a variável de controle mais acessível ao usuário é a frequência de operação do processador. O valor de frequência escolhido deve ser tal que minimize o consumo de energia, mas também que não resulte em perda de desempenho nas atividades em execução pelo processador. Nesta seção, é apresentada a metodologia para a escolha da frequência ótima de operação do processador. Esta escolha é feita a partir dos modelos apresentados na seção 2.2 e fazendo uso dos valores de requisições estimados usando o filtro apresentado na seção 2.3.

2.4.1 Escolha de Frequência de Operação Baseada nas Atividades do Usuário

A escolha de frequência de operção do processador com base nas atividades de usuário usa o comportamento passado do usuário para escolher a frequência ótima em termos de consumo de energia. Neste caso, será considerado o modelo de consumo de

energia baseado em frequência de operação, apresentado na subseção 2.2.2.1, cujo consumo é determinado pela Equação 2.5, repetida aqui

$$P = C_s f V^2 \tag{2.18}$$

Pela Equação 2.18, é perceptível que o consumo é diretamento proporcional à frequência de operação do processador.

Para aplicações em tempo real, é amplamente sugerido na literatura que o processador não fique mais de 70% do tempo ocupado (18). Isso garante uma margem de segurança para atender a requisições inesperadas. Desta forma, a variável controlada escolhida é o período no qual o processador fica ocioso. Usando como base a Equação 2.4, a qual é repetida aqui

$$1 = \lambda_p^{'} + \lambda_t + \lambda_i = \frac{\lambda_p}{f} + \lambda_t + \lambda_i, \tag{2.19}$$

é possível determinar a frequência de operação do processador para atender às requisições de tempo e processamento e garantir o tempo ocioso para a margem de segurança. Isolando a frequência na

$$f = \frac{\lambda_p}{1 - (\lambda_t + \lambda_i)} \tag{2.20}$$

Para a escolha da frequência de operação do processador, são utilizados valores estimados, $\hat{\lambda_t}$, de λ_t, e $\hat{\lambda_p}$, de λ_p e o nível de osciosidade de segurança estabecedido λ_i^*, como variável de referência. Logo, a Equação 2.20 é reescrita

$$f = \frac{\hat{\lambda_p}}{1 - \left(\hat{\lambda_t} + \lambda_i^*\right)} \tag{2.21}$$

3 Resultados e Discussões

3.1 Metodologia de Testes

Nesta seção, a metodologia para aplicação de testes é relatada. Primeiramente, são descritas a implementação dos algorítmos de estimação de carga de trabalho e controle de frequência de operação. Na sequência, uma caracterização da plataforma experimental é apresentada. Por último, é apresentado um detalhamento da carga de trabalho utilizada para os testes experimentais.

3.1.1 Implementação

A estrégia de controle, bem como os processos de estimação de carga de trabalho e atualização do filtro preditor foram implementados em linguagem C/C++ (19). O fluxagrama da implementação desenvolvida pode ser visto na Figura 4.

Os parâmetros iniciais do filtro são escolhidos sempre 0. No bloco "realização de medidas", são determinados os valores de requisições de tempo e processamento que chegaram no decorrer do último período de amostragem. No bloco "Estimação de carga de trabalho", são realizadas estimações de requisições de tempo e processamento para cada processador por meio da Equação 2.11

$$\hat{x}(t) = \sum_{i=1}^{n} w(i)\, x(t-i) \tag{3.1}$$

Para evitar que o valor estimado seja negativo, o valor da estimativa é sempre limitado inferiormente em zero. No bloco "Aplicação da lei de controle", os valores estimados de requisições de tempo e processamento são utilizados para determinar a frequência de operação de cada processador por meio da Equação 3.2

$$f = \frac{\hat{\lambda_p}}{1 - \left(\hat{\lambda_t} + \lambda_i^*\right)} \tag{3.2}$$

O conjunto de frequências de operação dos processadores é discreto, sendo escolhido o valor existente no conjunto de frequência do processador imediatamente superior ao valor obtido usando a Equação 3.2. No bloco "determinação do erro de estimação", o erro de estimação (Equação 2.16) é determinado

$$e(t) = x(t) - \hat{x}(t) \tag{3.3}$$

No bloco "Atualização do filtro preditor", os valores das realizações do processo são utilizados para determinar o fator de reajuste (Equação 2.17)

$$\mu = \frac{1}{2\mathbf{x}^T(t)\,\mathbf{x}(t)} \tag{3.4}$$

e, juntamento com o erro de estimação, utilizados para atualizar o filtro preditor por meio da Equação 2.15

$$\mathbf{w}\left(\mathbf{n}+\mathbf{1}\right)=\mathbf{w}\left(\mathbf{n}\right)+\frac{\mu}{\left(1-\gamma\right)+\mathbf{x}^{T}\left(n\right)\mathbf{x}\left(n\right)}\mathbf{x}\left(n\right)e\left(n\right) \tag{3.5}$$

Este ciclo fica em repetição periodicamente para adequar a frequência de operação do processador às cargas de trabalho em execução naquele momento.

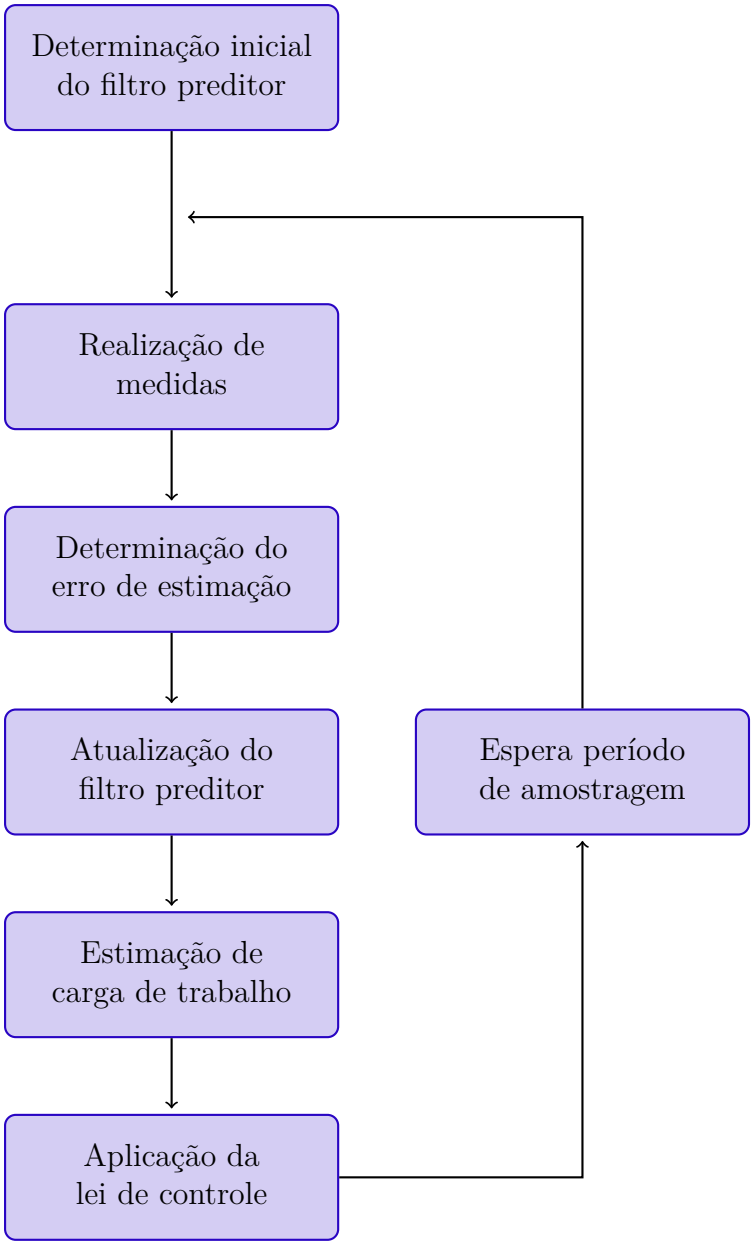

Figura 4 – Fluxograma da estratégia de controle.

3.1.2 Plataforma Experimental

Os testes experimentais foram executados em um *notebook* executado o sistema opecional Linux Mint, com processador Core i5-3337U, contendo 4 núcleos e conjunto de

frequências de operação {0,8000; 0,9000; 1,0000; 1,1000; 1,2000; 1,3000; 1,4000; 1,5000; 1,6000; 1,7000; 1,8000; 1,8100} GHz e 4GB de memória RAM (*Random Access Memory*).

3.1.3 Carga de Trabalho

Para a plataforma experimental descrita na subseção 3.1.2, à exceção de aplicações que requerem grande poder de processamento, a exemplo de jogos da última geração, é difícil executar cargas de trabalho rotineiras que necessitem de grande utilização do processador.

A carga de trabalho para testar a implementação da estratégia de controle consiste na execução de múltiplas aplicações, tais como navegador para Internet, processador de texto e aplicativos multimídia. Os aplicativos multimítia e de navegação para internet exigem operações de entrada e saída regularmente. Para gerar esta carga de trabalho, um *script* de 120 segundos foi gerado seguindo a seguinte sequência:

- segundo 0: começa execução do *script*.
- segundo 20: executa programa de aplicação multimídia: arquivo de vídeo em formato mp4, 67,3MB, programa *kaffeine*.
- segundo 40: executa programa de aplicação multimídia: arquivo de vídeo em formato mp4, 67,3MB, programa *totem*.
- segundo 60: abre programa de edição de texto: arquivo de texto em formato doc, 1,3MB, programa *libreoffice*.
- segundo 80: abre programa de navegação para Internet: 4 abas, vídeos do *youtube*, programa *google-chrome*.
- segundo 100: fecha programa de aplicação multimídia: programa *totem*.
- segundo 120: termina execução do *script*.

3.2 Estimação da Carga de Trabalho

Para validar a estimação da carga de trabalho, o *script* foi executado com todos os 4 núcleos na frequência de operação fixa de $0,8$GHz. Foram estimados os valores de carga de trabalho usando filtros preditores adaptativos de várias ordens.

Para estimação das requisições de processamento, o melhor filtro, em termos de minimização do erro médio quadrático de predição, foi o filtro de quinta ordem Filtros de ordem superior apresentaram erro médio quadrático semelhante ao do filtro de quinta

ordem. Para estimação das requisições de tempo, o melhor filtro, em termos de minimização do erro médio quadrático de predição, foi o filtro de terceira ordem. Filtros de ordem superior ou inferior apresentaram erro médio quadrático maior.

Na Figura 5, são apresentados os valores reais e os valores estimados de requisições de processamento usando fiiltro preditor adaptativo de quinta ordem. Na Figura 6, são apresentados os valores reais e os valores estimados de requisições de tempo usando fiiltro preditor adaptativo de terceira ordem. Para mostar que o filtro preditor é melhor que usar valores passados da carga de trabalho, na Figura 7, são apresentados os valores reais de carga de trabalho e os valores estimados de requisições de processamento considerando que o valor de requisições no instante de tempo seguinte é igual ao instante de tempo atual; na Figura 8, são apresentados os valores reais de carga de trabalho e os valores estimados de requisições de tempo considerando que o valor de requisições no instante de tempo seguinte é igual ao instante de tempo atual.

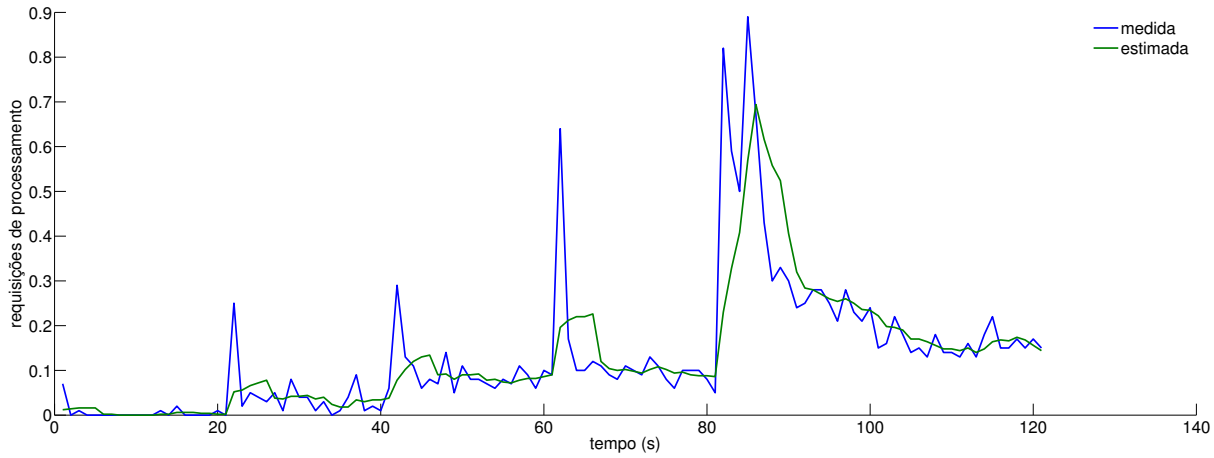

Figura 5 – Valores medidos e estimados de requisições de processamento usando filtro preditor adaptativo de quinta ordem.

Nas Figuras 9 e 10, são apresentados os erros de estimação para o filtro preditor e considerando a carga igual ao instante de tempo anterior. Na Figura 9, o erro exibido corresponde às requisições de processamento. Na Figura 10, o erro exibido corresponde às requisições de tempo. Os valores do erro médio quadrático para cada caso são exibidos na Tabela 1.

3.2.1 Discussão

Pelos resultados, é possível perceber que o filtro preditor é útil para prever requisições futuras, provendo erro pequeno de predição. Os errors de predição, são maiores quando acontecem pulsos de requisições, sendo estes imprevisíveis e instantâneos e desnecessária a mudança de frequência de operação do processador nestes casos. Os pulsos

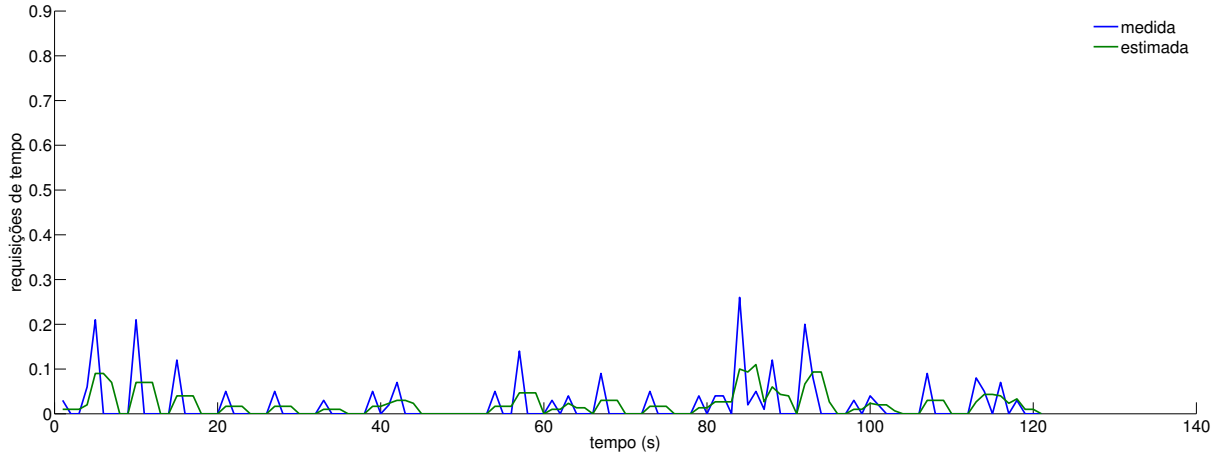

Figura 6 – Valores medidos e estimados de requisições de tempo usando filtro preditor adaptativo de terceira ordem.

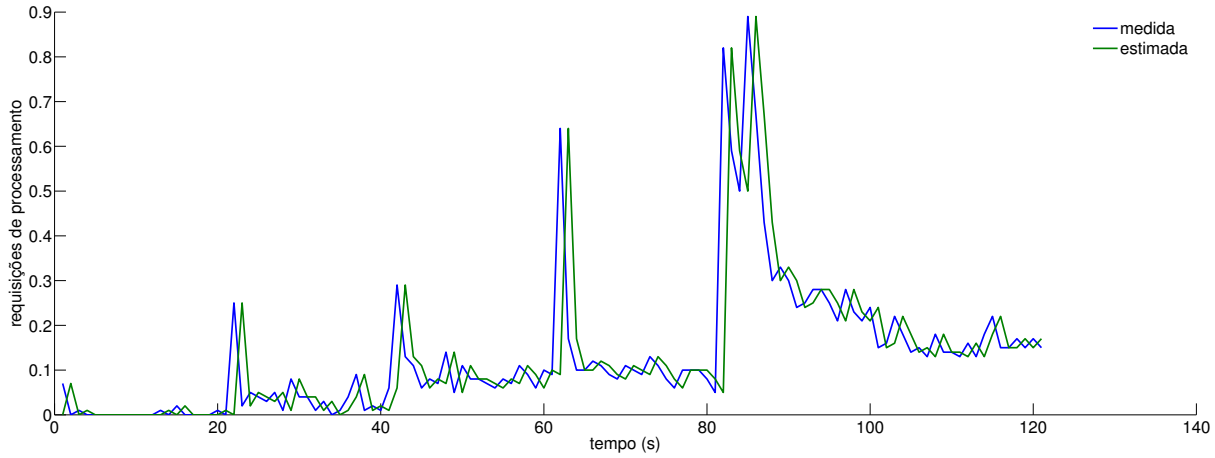

Figura 7 – Valores medidos e estimados de requisições de processamento usando instante de tempo anterior.

ocorreram nos instantes de abertura de novos programas, sendo estes acompanhados de degraus de requisições, necessários a cada nova aplicação aberta. O filtro preditor transforma o impulso em uma estimaativa exponencial de requisições, possibilitando atuar adequadamente na escolha da frequência de operação.

A maior parte das requisições de tempo surgiu em pulsos, o que pode ter sido determinante para que filtros de ordem maior apresentassem maior erro de predição. As requisições de processamento e de tempo apresentaram comportamentos diferentes, sendo este o motivo provável para que diferentes filtros sejam mais adequados para cada tipo de requisição.

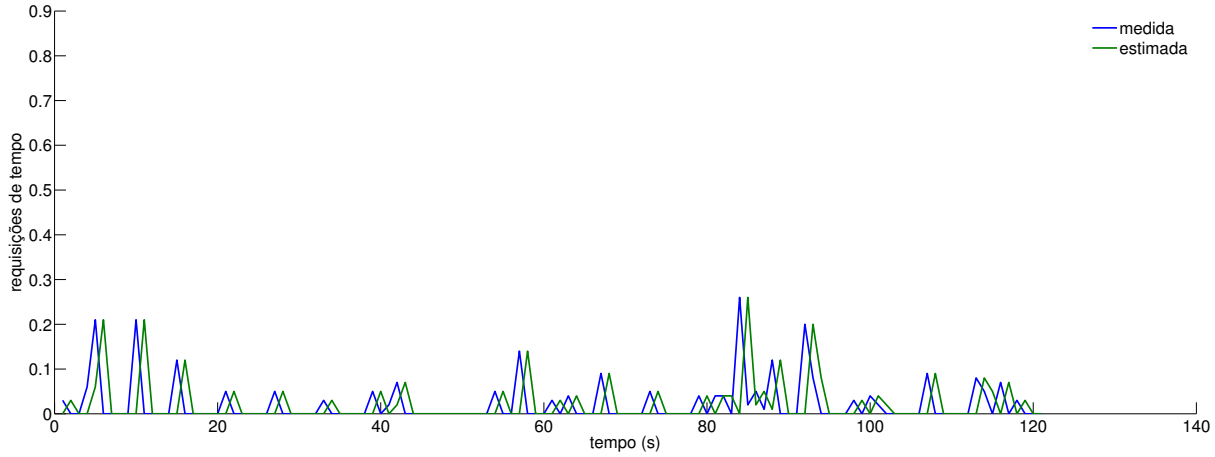

Figura 8 – Valores medidos e estimados de requisições de tempo usando instante de tempo anterior.

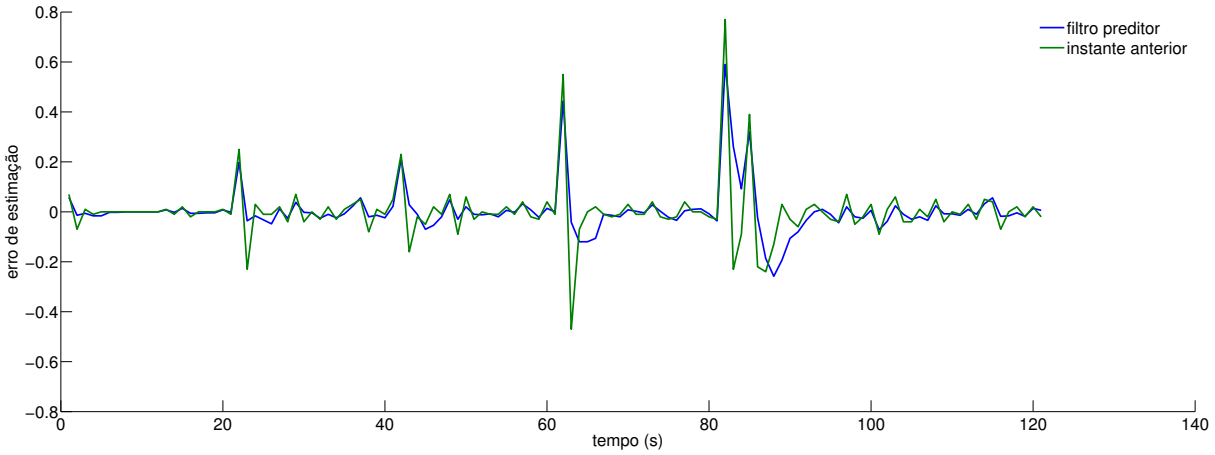

Figura 9 – Erros de estimação de requisições de processamento para filtro preditor adaptativo e requisições no instante de tempo anterior.

3.3 Aplicação Simultânea de Estimação de Carga de Trabalho e Controle de Frequênia de Operação

Para validar a estratégia de controle, o mesmo *script* apresentado na subseção 3.1.3 foi executado com os núcleos 1 a 3 a frequência fixa de $0,8$GHz e com ajuste frequência de operação do núcleo 0 de acordo com a Equação 3.2. Para estimação da carga de trabalho, um filtro preditor de ordem 5 foi utilizado para estimar requisições de processamento e um filtro preditor de ordem 3 foi utilizado para estimar requisições de tempo. Para os resultados apresentados nesta seção, a frequência de operção do processador 0 foi variada.

Na Figura 11, são apresentados os valores reais e os valores estimados das requisições de processamento usando o fiiltro preditor. Na Figura 12, são apresentados os valores

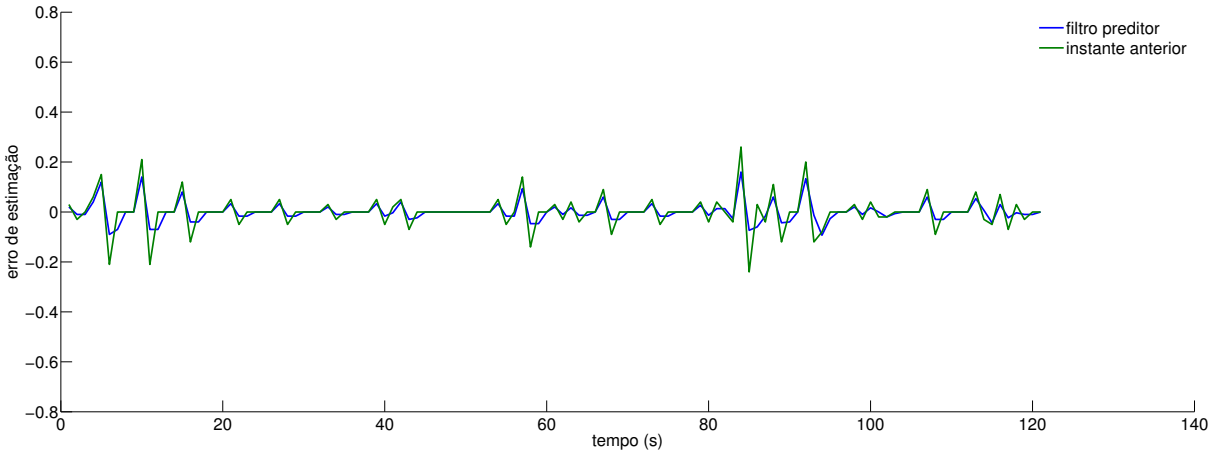

Figura 10 – Erros de estimação de requisições de tempo para filtro preditor adaptativo e requisições no instante de tempo anterior.

Tabela 1 – Erro médio quadrático de estimação para diversos casos.

		Erro médio quadrático
Requisições de Processamento	- Filtro Preditor	0,0089
Requisições de Processamento	- Instante de tempo anterior	0,0148
Requisições de Tempo	- Filtro Preditor	0,0015
Requisições de Tempo	- Instante de tempo anterior	0,0046

reais e os valores estimados das requisições de tempo usando o filtro preditor. Nas Figuras 13 e 14, são apresentados os erros de predição das requisições de tempo e de processamento. Na Figura 13, o erro exibido corresponde às requisições de processamento. Na Figura 14, o erro exibido corresponde às requisições de tempo.

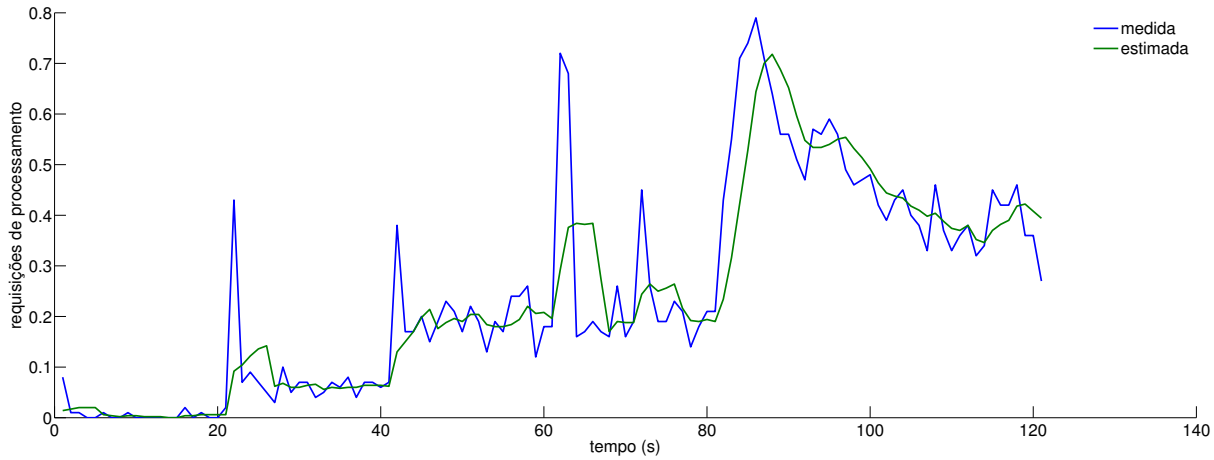

Figura 11 – Valores medidos e estimados de requisições de processamento usando filtro preditor adaptativo de quinta ordem.

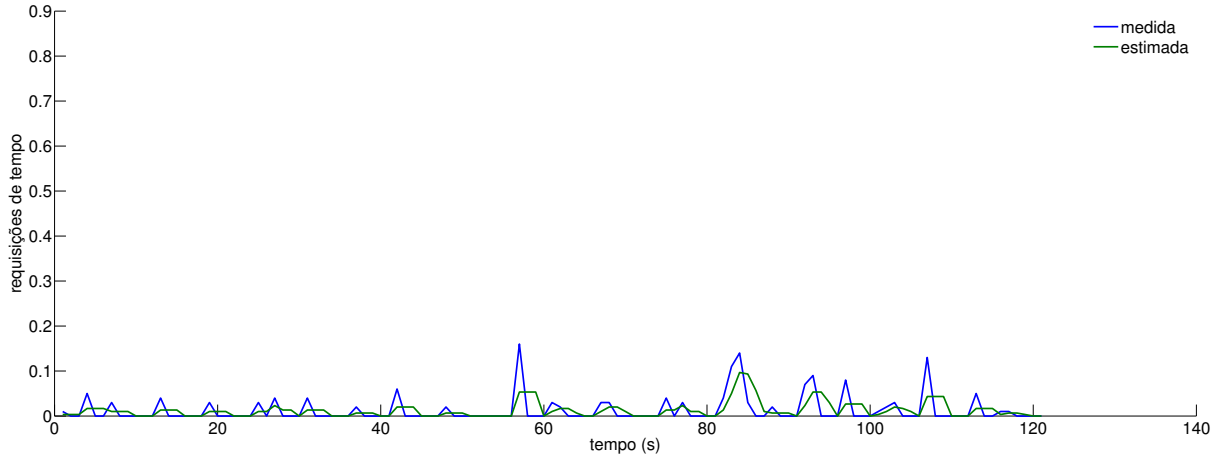

Figura 12 – Valores medidos e estimados de requisições de tempo usando filtro preditor adaptativo de terceira ordem.

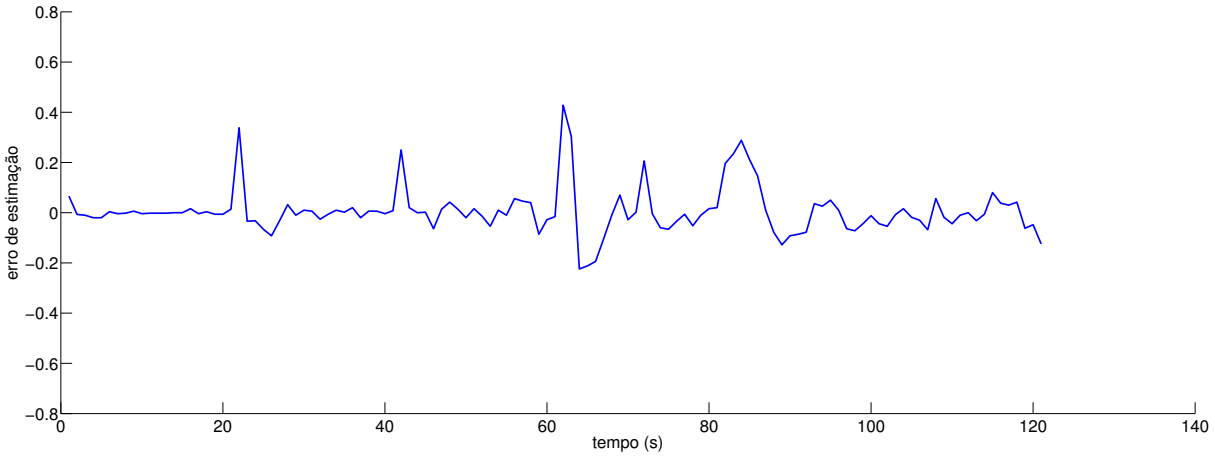

Figura 13 – Erros de estimação de requisições de processamento para filtro preditor.

A estratégia de controle desenvolvida visa controlar o tempo ocioso. Como a carga de trabalho gera pouca utilização na plataforma de testes utilizada, o período ocioso de referência escolhido foi de 60%.

Na Figura 15, são exibidos os valores normalizados da frequência de operação determinada pela Equação 3.2 ao longo do tempo. Também são exibidos os valores normalizados[1] de frequência de operação escolhidos para controlar o processador. Na Figura 16, são exibidos os valores reais da frequência de operação do processador ao longo do tempo. Na Figura 17, é exibido o nível ocioso do processador, que é a variável a ser controlada, ao longo do tempo.

[1] Neste caso, normalizado significa dividido pela menor frequência de operação do processador (0,800GHz).

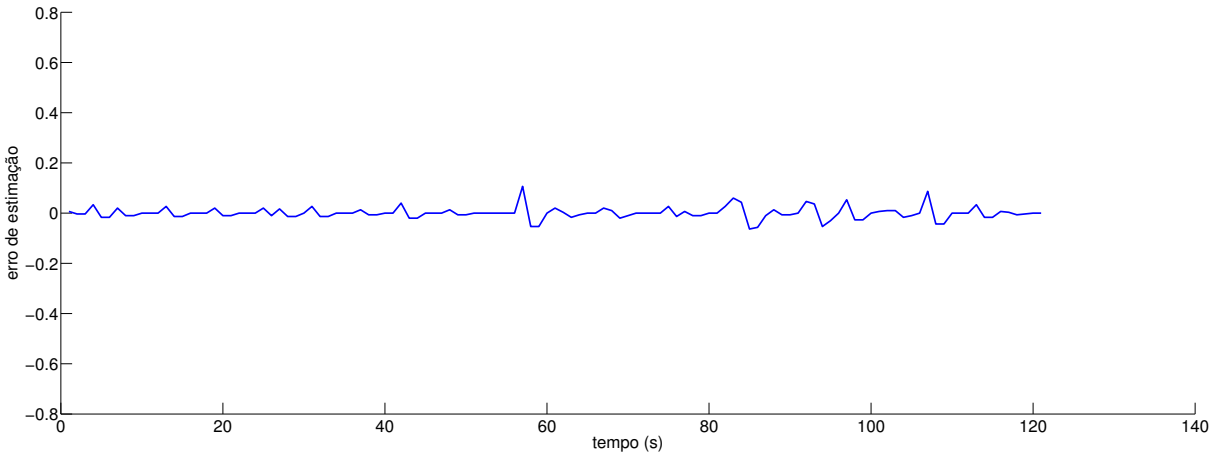

Figura 14 – Erros de estimação de requisições de tempo para filtro preditor.

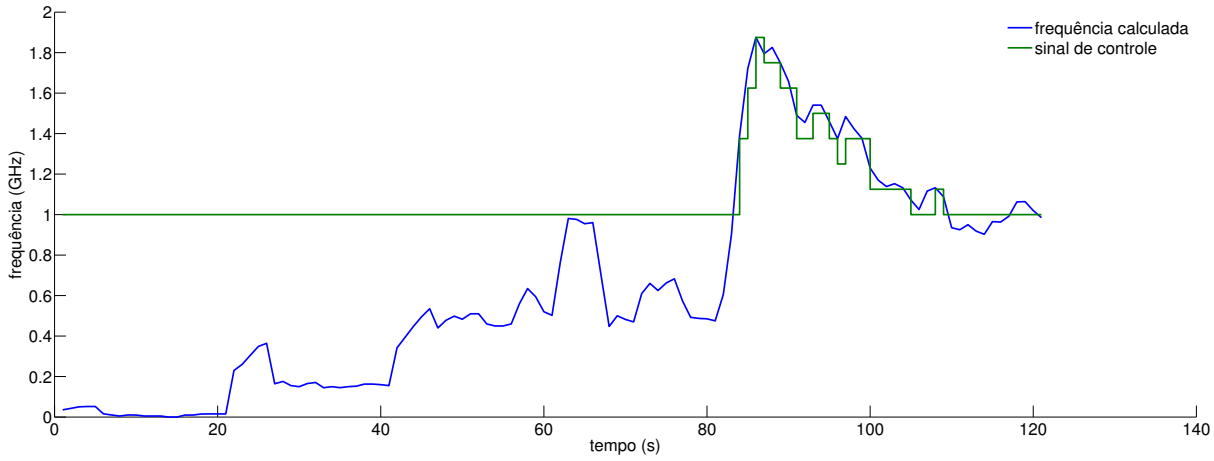

Figura 15 – Frequênca de operação normalizada para a variável controlada assumir o valor de referência.

3.3.1 Discussão

Pelos resultados, é possível perceber que a estratégia de controle desenvolvida, associada ao filtro preditor, pode ser efetiva no controle do período ocioso do processador.

O nível de ociosidade de referência do processador foi escolhido para ser 60%. Pela Figura 17, o nível de ociosidade é maior que o valor de referência no intervalo de tempo inicial, isso se deve à carga de trabalho ser pequena neste intervalo de tempo, sendo a frequência de operação escolhida a mínima possível. Uma carga de trabalho maior surge quando o navegador para internet é aberto com várias abas, neste caso, a frequência de operação do processador é variada, mantendo o nível de ociosidade do processador em torno do valor de referência com um pequeno desvio, o que é aceitável devido às grandes variações da requisições ao longo do tempo.

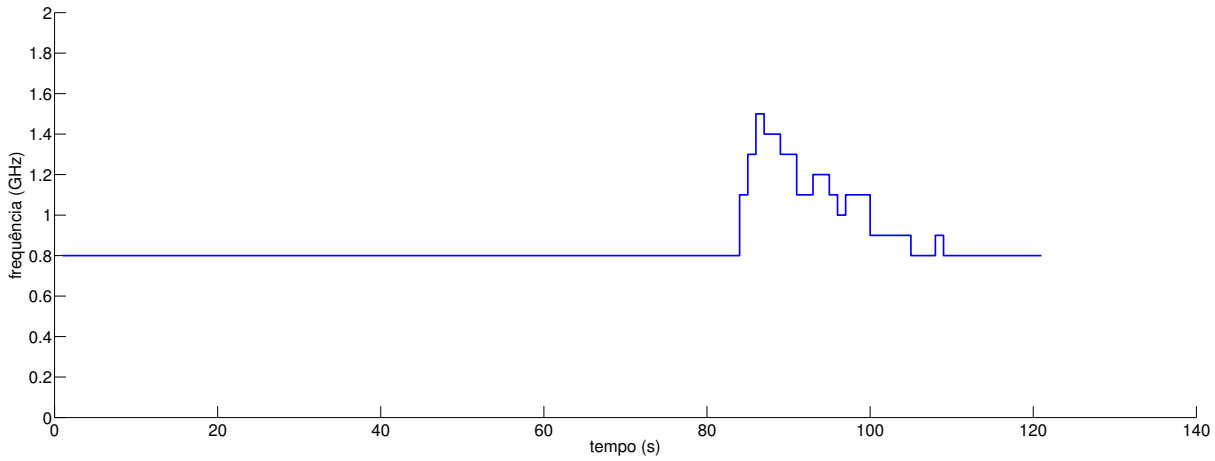

Figura 16 – Sinais de controle aplicados ao longo do tempo.

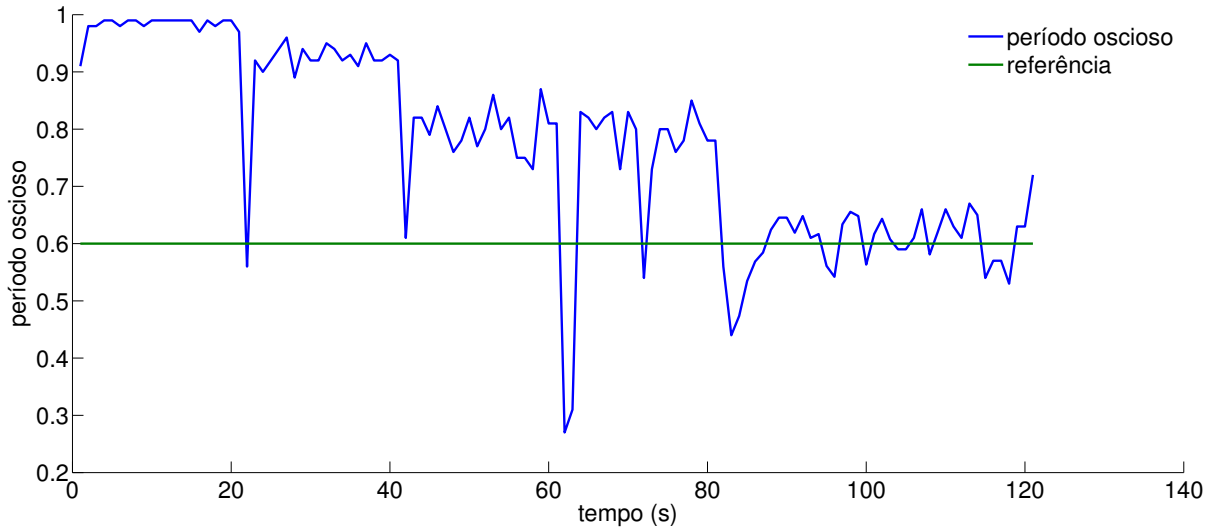

Figura 17 – Nível de ociosidade do processador ao longo do tempo.

4 Conclusão

Neste trabalho, um modelo para cargas de trabalho gerais e uma estratégia de controle para gerenciamento de energia em processadores foram apresentados. Nas seções seguintes: é apresentada uma breve síntese do trabalho desenvolvido; são levantadas algumas consideração importantes, tais como vantagens e comparações em relação a com outras abordagens e a implemenentação apresentada; e elencadas possíveis atividades futuras para continuação ou utilização do trabalho.

4.1 Síntese do Trabalho

Neste trabalho, um modelo para cargas de trabalho gerais foi apresentado. O modelo é baseado em um Processo Estocástico de Poisson e leva em consideração que as requisições de serviços ao processador apresentam comportamento altamente correlacionado com requisições passadas. A carga de trabalho é predita por meio de um filtro preditor baseado em correlação das requisições de serviço. O modelo, juntamente com a estimação da carga de trabalho, é então utilizado para o desenvolvimento de uma estratégia ótima de controle, visando não apenas minimizar o consumo de energia, mas também atender às requisições de serviço e oferecer uma margem de segurança para possíveis requisições urgentes e inesperadas.

Testes experimentais foram realizados para validar a abordagem proposta. Os testes experimentais foram executados em um notebook executando o sistema opecional Linux Mint, com processador Core i5-3337U, contendo 4 núcleos e conjunto de frequências de operação {0,8000; 0,9000; 1,0000; 1,1000; 1,2000; 1,3000; 1,4000; 1,5000; 1,6000; 1,7000; 1,8000; 1,8100} GHz e 4GB de memória RAM. A carga de trabalho consiste em abertura e execução de aplicações rotineiras, como aplicativos multimídia, editores de texto e navegadores para Internet. Pelos resultados obitidos, é possível comprovar a eficácia dos métodos de predição da carga de trabalho, assim como o desempenho do algorítmo de controle.

4.2 Considerações

A abordagem estocástica baseada em Poisson para o problema proposto apresenta algumas vantagens em relação às abordagens baseados em Cadeias de Markov ou Teoria de Controle.

Comparada com a abordagem baseada em Cadeias de Markov, a abordagem ba-

seada em Poisson possibilita evitar um grande número de estados (12×4 apenas para as frequências de operação no caso da plataforma de testes utilizadas neste trabalho). A abordagem também possibilita uma melhor tomada de decisão, já que usa dados passados ao invez de usar apenas o estado atual para determinar o que deve ser feito

Comparada com a abordagem baseada em Teoria de Controle, a abordagem baseada em Poisson apresenta um nível de abstração menor, contendo um melhor significado do processo em execução. Outra vantagem significativa é em relação a possibilidade de predição de comportamento da carga de trabalho e agendamento dos sinais de controle, o que, infelizmente, não é possível usando Teoria de Controle.

4.3 (Possíveis) Trabalhos Futuros

Uma lista de possíveis trabalhos futuros, dando continuidade ou utilizando o presente trabalho é apresentada a seguir:

- Testes de outros filtros preditores para a carga de trabalho;
- estimação da carga de trabalho considerando um comportamento de múltipla periodicidade (20);
- implementação da estratégia proposta em outros sistemas, tais como servidores, roteadores, etc.

Referências

1 BENINI, L.; BOGLIOLO, A.; MICHELI, G. D. A survey of design techniques for system-level dynamic power management. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, v. 8, n. 3, p. 299–316, 2000. ISSN 1063-8210. Citado 2 vezes nas páginas 1 e 2.

2 BENINI, L. et al. Symbolic synthesis of clock-gating logic for power optimization of control-oriented synchronous networks. In: *European Design and Test Conference, 1997. ED TC 97. Proceedings.* [S.l.: s.n.], 1997. p. 514–520. ISSN 1066-1409. Citado na página 1.

3 BROCK, B.; RAJAMANI, K. Dynamic power management for embedded systems [soc design]. In: *SOC Conference, 2003. Proceedings. IEEE International [Systems-on-Chip].* [S.l.: s.n.], 2003. p. 416–419. Citado na página 1.

4 SIMUNIC, T.; BENINI, L.; MICHELI, G. D. Event-driven power management of portable systems. In: *System Synthesis, 1999. Proceedings. 12th International Symposium on.* [S.l.: s.n.], 1999. p. 18–23. Citado na página 2.

5 LUIZ, S. O. D.; PERKUSICH, A.; LIMA, A. M. N. Stochastic learning-based weak estimation for dynamic power management. In: *XVIII Congresso Brasileiro de Automática, Anais do.* [S.l.: s.n.], 2010. p. 2113–2120. Citado na página 2.

6 PALEOLOGO, G. et al. Policy optimization for dynamic power management. In: *Design Automation Conference, 1998. Proceedings.* [S.l.: s.n.], 1998. p. 182–187. Citado na página 2.

7 QIU, Q.; PEDRAM, M. Dynamic power management based on continuous-time markov decision processes. In: *Design Automation Conference, 1999. Proceedings. 36th.* [S.l.: s.n.], 1999. p. 555–561. Citado na página 2.

8 XIA, F. et al. Control-theoretic dynamic voltage scaling for embedded controllers. *Computers Digital Techniques, IET*, v. 2, n. 5, p. 377–385, Setembro 2008. ISSN 1751-8601. Citado na página 2.

9 LUIZ, S.; PERKUSICH, A.; LIMA, A. Adaptive control for power management. In: *Control and Automation (ICCA), 2011 9th IEEE International Conference on.* [S.l.: s.n.], 2011. p. 1027–1032. ISSN 1948-3449. Citado na página 2.

10 LUIZ, S. et al. Identification and control for processor power management. In: *Consumer Electronics (ICCE), 2012 IEEE International Conference on.* [S.l.: s.n.], 2012. p. 410–411. ISSN 2158-3994. Citado na página 2.

11 LUIZ, S. et al. System identification and energy-aware processor utilization control. *Consumer Electronics, IEEE Transactions on*, v. 58, n. 1, p. 32–37, Fevereiro 2012. ISSN 0098-3063. Citado na página 2.

12 LEON-GARCIA, A. *Probability and Random Processes For EE's (3 ed)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007. ISBN 0131471228. Citado 2 vezes nas páginas 5 e 8.

13 SHEARER, F. *Power Management in Mobile Devices*. Newton, MA, USA: Newnes, 2007. ISBN 0750679581, 9780750679589. Citado na página 7.

14 SIMAR, L. Maximum likelihood estimation of a compound poisson process. *The Annals of Statistics*, The Institute of Mathematical Statistics, v. 4, n. 6, p. 1200–1209, 11 1976. Disponível em: <http://dx.doi.org/10.1214/aos/1176343651>. Citado na página 8.

15 HAYKIN, S. *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-322760-X. Citado 2 vezes nas páginas 8 e 9.

16 WIENER, N. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. [S.l.]: The MIT Press, 1964. ISBN 0262730057. Citado na página 8.

17 LJUNG, L. (Ed.). *System identification (2nd ed.): theory for the user*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999. ISBN 0-13-656695-2. Citado na página 9.

18 LAPLANTE, P. A. *Real-Time Systems Design and Analysis: An Engineer's Handbook*. Piscataway, NJ, USA: IEEE Press, 1992. ISBN 0780304020. Citado na página 11.

19 STROUSTRUP, B. *The C++ Programming Language*. 3. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN 0201700735. Citado na página 12.

20 KUHL, M.; WILSON, J.; JOHNSON, M. Estimating and simulating poisson processes having trends or multiple periodicities. *IIE Transactions*, Kluwer Academic Publishers, v. 29, n. 3, p. 201–211, 1997. ISSN 0740-817X. Disponível em: <http://dx.doi.org/10.1023/A%3A1018567028064>. Citado na página 23.