



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

LAÉCIO TRAJANO DE SALES

TRABALHO DE CONCLUSÃO DE CURSO
IDENTIFICAÇÃO DO MODELO DE POTÊNCIA DE UM SISTEMA
COMPUTACIONAL

Campina Grande, Paraíba
Março de 2013

LAÉCIO TRAJANO DE SALES

IDENTIFICAÇÃO DO MODELO DE POTÊNCIA DE UM SISTEMA
COMPUTACIONAL

*Trabalho de Conclusão de Curso submetido à
Coordenação do curso de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Engenheiro Eletricista.*

Área de Concentração: Controle e Automação

Orientador:

Professor Saulo Oliveira Dornellas Luiz, D. Sc.

Campina Grande, Paraíba
Março de 2013

LAÉCIO TRAJANO DE SALES

IDENTIFICAÇÃO DO MODELO DE POTÊNCIA DE UM SISTEMA COMPUTACIONAL

Trabalho de Conclusão de Curso submetido à
Coordenação do Curso de Engenharia Elétrica da
Universidade Federal de Campina Grande como parte dos
requisitos necessários para a obtenção do grau de
Engenheiro Eletricista.

Área de Concentração: Controle e Automação

Aprovado em 08/03/2013

Professor Jaidilson Jó da Silva, D. Sc
Universidade Federal de Campina Grande
Avaliador

Professor Saulo Oliveira Dornellas Luiz, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

AGRADECIMENTOS

Primeiramente, agradeço a Deus pela oportunidade de estar nesse mundo e por todas as graças concedidas ao longo de minha caminhada.

À minha família, em especial aos meus pais Severino dos Ramos e Maria da Guia, e meus irmãos, Wagner Trajano e Ricardo Trajano, pelo amor, incentivo e apoio incondicional necessário para que eu concluísse o curso de Engenharia Elétrica.

Às amizades iniciadas durante o curso que contribuíram para a conclusão do mesmo e que permeará por toda vida, em especial aos amigos do Embedded.

A Bruna por todo o apoio prestado durante a conclusão desse trabalho.

A todos aqueles que compõem o Departamento de Engenharia Elétrica, em especial os professores, a quem devo grande respeito e admiração pelos ensinamentos que me proporcionaram tornar um engenheiro eletricista.

Ao Professor Saulo Oliveira Dornellas Luiz pela paciência e pelas valiosas orientações e contribuições ao longo da minha formação acadêmica.

A todos aqueles que não foram citados, mas que de forma direta ou indiretamente fizeram parte de minha vida acadêmica.

“Não se constrói uma reputação com o que ainda não foi feito.”

Henry Ford.

LISTA DE ILUSTRAÇÕES

Figura 1: Bloco que relaciona as entradas e a saída de um modelo	4
Figura 2: Diagrama de blocos de um sistema com gerenciamento de energia.....	7
Figura 3: Imagem do <i>notebook</i> Positivo Select 8010.....	10
Figura 4: Diagrama de blocos para medição de tensão e corrente.	10
Figura 5: Diagrama elétrico para medição de tensão e corrente do Positivo Select através do módulo de aquisição de dados NI USB 6210.....	11
Figura 6: Imagem do Módulo de Aquisição de Dados NI USB 6210.....	12
Figura 7: Diagrama de blocos do programa LabVIEW.....	12
Figura 8: Representação do Painel frontal no programa LabVIEW.....	13
Figura 9: Representação do Painel frontal do <i>software Components Logger</i>	15
Figura 10: Contribuição individual dos dispositivos para obtenção do modelo	21
Figura 11: Contribuição individual dos dispositivos para obtenção do modelo	21
Figura 12: Contribuição individual dos dispositivos para obtenção do modelo	22
Figura 13: Contribuição individual dos dispositivos para obtenção do modelo	22
Figura 14: Gráfico do modelo de potência do <i>Notebook Positivo Select</i>	23
Figura 15: Consumo de potência medida e estimada – teste 1.....	24
Figura 16: Comparativo entre as potências médias – teste 1.....	25
Figura 17: Consumo de potência medida e estimada – teste 2.....	26
Figura 18: Comparativo entre as potências médias – teste 2.....	26
Figura 19: Representação do gerador de carga CPU Stress Utility.....	32

LISTA DE TABELAS

Tabela 1: Métodos presentes na implementação do Components Logger a serem utilizados para a obtenção dos estados dos dispositivos .	14
Tabela 2: Descrição das entradas do sistema.	16
Tabela 3: Potência média consumida e Valores médios dos parâmetros do dispositivo.	18
Tabela 4: Parâmetros obtidos a partir da estimação por mínimos quadrados com restrições	19
Tabela 5: Relação das entradas com variáveis especificadas na Equação 17.	20
Tabela 6: Erro percentual para cada intervalo de medição – teste 1.	25
Tabela 7: Erro percentual obtido em cada intervalo de medição – teste 2.	26

SUMÁRIO

1	Introdução	1
1.1	Objetivos	2
1.2	Metodologia	3
1.3	Estrutura do Trabalho	3
2	Fundamentação Teórica	4
2.1	Método dos Mínimos quadrados	4
2.2	Algoritmo de identificação pelo método dos mínimos quadrados com restrições	6
3	Plataforma Experimental	9
3.1	Notebook Positivo Select 8010	9
3.2	Sistema para medição de potência	10
3.2.1	Circuito adaptador	10
3.2.2	Módulo de Aquisição de dados NI USB – 6210	11
3.2.3	LabVIEW	12
3.2.4	MATLAB	13
3.2.5	<i>Components Logger</i>	13
4	Procedimentos Experimentais	16
4.1	Cargas de Trabalho	16
4.1.1	Testes variando o brilho do LCD	17
5	Resultados Obtidos	19
5.1	Identificação do modelo de potência	19
5.2	Validação do modelo	24
6	Conclusão	28
	Anexo A – descrição dos experimentos	29
	Anexo A1 - Testes variando o brilho de tela	29
	Anexo A2 - Testes com a Unidade de DVD_RW	30
	Anexo A3 - Testes com a <i>Webcam</i>	31
	Anexo A4 - Testes com alocação de memória	31
	Anexo A5 - Testes com carga de CPU	32
	Anexo A.6 - Testes com as interfaces de redes, <i>Ethernet</i> e <i>Wi-fi</i>	33
	Anexo A.6.1 – Testes com <i>download</i> pela <i>Wi-fi</i>	33
	Anexo A.6.2 – Testes com <i>download</i> pela <i>Ethernet</i>	34

Anexo A.6.3 – Testes de <i>upload</i> pela <i>Wi-fi</i>	34
Anexo A.6.4 – Testes de <i>upload</i> pela <i>Ethernet</i>	34
Anexo A.6.5 – Testes com ativação/desativação da <i>Wi-fi</i>	35
Anexo A.6.6 – Testes com ativação/desativação da <i>Ethernet</i>	35
Anexo B - Testes para validação do modelo de potência.....	36
Anexo B.1 - Carga Variável 1	36
Anexo B.2 - Carga Variável 2	37
Anexo C - Scripts do MatLab.....	38
Anexo C.1 - Script para obtenção dos valores de médios de potencia e utilização dos dispositivos.....	38
Anexo C.2 - Script para obtenção do modelo de potência	39
Anexo C.3 - Script para validação do modelo	40
Bibliografia.....	43

1 INTRODUÇÃO

A eficiência energética sempre será considerada um assunto de interesse dentro de todas as áreas da Engenharia. Mas, atualmente, com o aumento do poder de computação, a grande conectividade e o grande aumento recente da variedade de serviços e aplicativos disponíveis nos dispositivos móveis como *notebooks*, *netbooks*, *smartphones* e *tablets*, este assunto se tornou prioridade em muitos centros de pesquisas e empresas [1].

Os usuários de dispositivos móveis desejam obter a maior autonomia possível de seus aparelhos. Ter que parar para recarregá-lo em um curto intervalo de tempo pode ser uma experiência decepcionante e pode desencorajar muitas pessoas a utilizá-los [1].

Com este panorama em vista, estão sendo pesquisadas diversas melhorias para a autonomia desses aparelhos. Uma possibilidade é a tentativa de desenvolver novos tipos de baterias [2]. Todavia, o aumento no tamanho e na capacidade das baterias não vem acompanhando os avanços das indústrias de semicondutores. Por consequência, o melhor caminho seria a redução do consumo energético destes dispositivos [1]. Surgiu nesse contexto, o Gerenciamento Dinâmico de Energia (GDE) com o objetivo de desenvolver políticas de gerenciamento, a fim de reduzir o consumo de energia dos dispositivos, através de técnicas como o escalonamento dinâmico de tensão e frequência (DVFS) e estados de baixo consumo de energia dos dispositivos.

Contudo, o gerenciamento de energia eficaz em sistemas computacionais não é algo simples de se desenvolver, pois se devem levar em conta diversos itens, por exemplo, implicações no desempenho do sistema. Com isso, torna-se de grande relevância o conhecimento da dinâmica do sistema, sendo necessária a implementação de métodos que estimem com precisão o comportamento da carga de trabalho e a potência consumida pelo sistema. Abrindo campo, nesse contexto, para estudo e desenvolvimento de modelos de potência.

Modelos de potência tornaram-se uma ferramenta bastante útil para a otimização e gerenciamento de energia em dispositivos móveis. Alguns modelos de potência são desenvolvidos em laboratório, com a ajuda de um segundo computador dedicado a aquisição de dados, ocasionando um extensivo trabalho, devido à grande quantidade de

experimentos a serem realizados. É importante ressaltar que estes métodos não se adequam aos diversos tipos de *hardwares* utilizados em dispositivos móveis, já que são projetados para modelarem uma plataforma em particular. Outros métodos avaliam a estimativa de potência para intervalos de tempo maiores que um segundo, o que não é razoável para um modelo de potência mais acurado. Outro paradigma é a auto-modelagem, em que o próprio sistema móvel gera automaticamente o seu modelo de potência sem qualquer auxílio externo. Neste caso, as medições de potencia são realizadas através de uma interface de bateria inteligente [3].

Enfim, decidir qual metodologia utilizar na modelagem de um sistema é uma etapa fundamental no desenvolvimento do modelo.

Neste trabalho, para obtenção das medições de potência do sistema, será utilizado o método auxiliado por um sistema externo de aquisição de dados e para identificação do modelo matemático, será utilizado o método dos mínimos quadrados com restrições. Por se tratar de um modelo de potência, os parâmetros a serem estimados estão associados a componentes de potencia, com isso, ficam estabelecidas as restrições de que todos os parâmetros estimados devem ser não negativos, evidenciando o termo “restrições” no método utilizado.

1.1 OBJETIVOS

O objetivo deste trabalho é a estimação de um modelo de potência, para um sistema computacional, pelo método dos mínimos quadrados com restrições, a partir dos estados de energia dos dispositivos, tais como LCD (*liquid crystal display*), HD (disco rígido), Unidade de leitura ótica, *Webcam*, processador e interfaces de rede.

O sistema computacional estudado é um *notebook*, com processador Intel® Core i7-620M. O Sistema Operacional (SO) instalado no *notebook* é o *Windows 7 Home Basic*.

Além disso, para a estimação do modelo, são necessárias medições de potência, as quais são realizadas por meio de um sistema de aquisição de dados, que faz a leitura das medições de tensão e corrente que a fonte/bateria fornece ao sistema.

1.2 METODOLOGIA

Para a realização deste trabalho de conclusão de curso, foram realizadas as seguintes atividades:

- Consulta às referências bibliográficas sobre gerenciamento dinâmico de energia e identificação de modelos de potência disponíveis;
- Implementação de um sistema para medição de potência consumida pela plataforma alvo e aquisição dos estados dos dispositivos.
- Implementação de um algoritmo de Mínimos Quadrados com Restrições, para estimação do modelo de potência em função dos estados dos dispositivos da plataforma alvo;
- Realização de experimentos para validação do modelo de potência do sistema.

1.3 ESTRUTURA DO TRABALHO

Na Seção 2 é apresentado o método dos mínimos quadrados com restrições, que é uma ferramenta matemática para a estimação de modelos de sistemas. Em seguida, na Seção 3, é apresentada a plataforma experimental, o sistema para medição de potência, a plataforma para Aquisição de Dados (DAQ), os *softwares* para coleta de dados LabVIEW e *Components* Logger, o software para tratamento de dados *MatLab*. Na Seção 4 são apresentados os experimentos realizados para a modelagem, estimação e modelo de potência. Na Seção 5 são apresentados os resultados obtidos no processo de estimação e validação do modelo de potência. E por fim, na Seção 6, são apresentadas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordadas as bases teóricas que permitirão desenvolver a metodologia deste trabalho, o qual será focado no desenvolvimento de um modelo de potência através do algoritmo dos mínimos quadrados, baseado na observação dos estados de energia dos dispositivos do sistema.

2.1 MÉTODO DOS MÍNIMOS QUADRADOS

O algoritmo dos mínimos quadrados (MQ) é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados de entrada, tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados. Tais diferenças são chamadas resíduos [4].

Este algoritmo processa todas as observações da entrada e da saída de um sistema simultaneamente e produz uma estimativa única do vetor de parâmetros. Este é chamado de algoritmo em batelada e é usado quando as estimativas são necessárias somente uma vez e processadas *off-line* [5].

O processo a ser modelado é o consumo de potência fornecida ao *notebook*, tendo como entradas os estados dos dispositivos. O bloco que representa o modelo de potência pode ser visualizado na Fig.1.

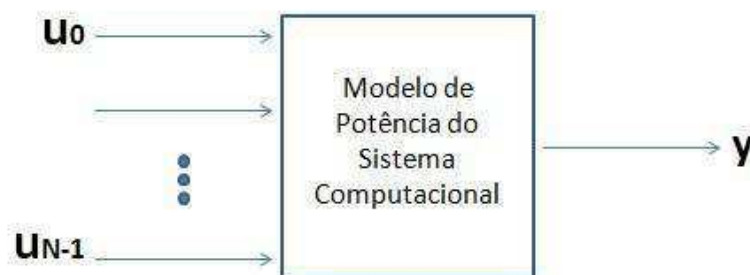


Figura 1: Bloco que relaciona as entradas e a saída de um modelo

Uma relação básica entre as entradas $u(t)$ com a saída $y(t)$ de um sistema, semelhante ao apresentado na Fig. 1, é a equação de diferenças linear, apresentada em (1):

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-1) + \dots + b_m u(t-m) \quad (1)$$

Os valores de entrada e saída que são utilizados para o modelo de potência do sistema computacional, são obtidos por amostragem, ou seja, tem-se um modelo discreto no tempo. Para melhor manipulação, a equação em (1) é colocada na forma em (2). De forma que é possível observar que o valor da saída atual do sistema depende dos valores passados de entradas e saída observadas [6].

$$y(t) = -a_1 y(t-1) - \dots - a_n y(t-n) + b_1 u(t-1) + \dots + b_m u(t-m) \quad (2)$$

Para uma notação mais compacta, são introduzidos os vetores:

$$\theta = [a_1 \dots a_n b_1 \dots b_m]^T \quad (3)$$

$$\varphi(t) = [-y(t-1) \dots -y(t-n) \ u(t-1) \dots u(t-m)]^T \quad (4)$$

Onde θ é o vetor de parâmetros que se deseja estimar, e $\varphi(t)$ é o vetor de entradas e saídas passadas do sistema.

E assim, a relação em (4) pode ser reescrita da forma:

$$y(t) = \varphi^T(t)\theta \quad (5)$$

Para enfatizar que a estimativa de $y(t)$ depende das entradas passadas, saídas passadas e também de θ , ela é denotada como:

$$\hat{y}(t|\theta) = \varphi^T(t)\theta \quad (6)$$

Agora, suponha que para um determinado sistema o qual não sabemos os valores dos parâmetros em θ , mas temos gravado as entradas e saídas em um intervalo de tempo $1 \leq t \leq N$:

$$Z^N = [u(1), y(1), \dots, u(N), y(N)] \quad (7)$$

Uma aproximação intuitiva, para selecionar θ em (1) através de (6), de modo a ajustar os valores calculados $\hat{y}(t|\theta)$ como uma estimativa das saídas medidas pelo método dos mínimos quadrados:

$$\min_{\theta} V_N(\theta, Z^N) \quad (8)$$

Onde,

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t|\theta))^2 = \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi^T(t)\theta)^2 \quad (9)$$

Iremos designar o valor de θ que minimiza (8) por $\hat{\theta}_N$:

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta, Z^N) \quad (10)$$

“ $\arg \min_{\theta}$ ” significa o argumento de minimização, ou seja, o valor de θ que minimiza V_N .

Dado que V_N é quadrático em θ , podemos encontrar o valor mínimo, facilmente definindo a derivada igual a zero:

$$0 = \frac{d}{d\theta} V_N(\theta, Z^N) = \frac{2}{N} \sum_{t=1}^N \varphi(t)(y(t) - \varphi^T(t)\theta)$$

que resulta em

$$\sum_{t=1}^N \varphi(t)y(t) = \sum_{t=1}^N \varphi(t)\varphi^T(t)\theta \quad (11)$$

Ou

$$\hat{\theta}_N = \left[\sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t)y(t) \quad (12)$$

Uma vez que os vetores de $\varphi(t)$ são definidos, a solução pode ser encontrada por um *software* de métodos numéricos como o *MatLab*.

2.2 ALGORITMO DE IDENTIFICAÇÃO PELO MÉTODO DOS MÍNIMOS QUADRADOS COM RESTRIÇÕES

Um sistema com gerenciamento de energia é um sistema de computador cujos dispositivos têm estados múltiplos de potência, cada um com uma potência e nível de desempenho específico. Como mostrado na Fig. 2, um sistema com gerenciamento de energia é composto por um conjunto de dispositivos $P = \{p_i, i = 1, 2, \dots, P\}$. Cada dispositivo é fornecido com um conjunto de estados de energia $S_i = \{s_j | j = 1, 2, \dots, S_i\}$, por exemplo, o brilho da tela. Para cada dispositivo, é possível atribuir uma métrica de utilização μ_i (carga do dispositivo). A potencia consumida y_i pelo dispositivo p_i é uma

função $c_i(s_i, \mu_i)$ do estado de energia s_i do dispositivo e da sua utilização μ_i . A estimação do consumo de potência \hat{y} do sistema de potência controlável é mostrado em (13), onde Y_0 é a potencia estática (operação dos circuitos básicos e manutenção dos estados de baixa energia dos dispositivos) e a componente $\sum_{i=1}^P y_i$ representa a potência dinâmica, que é uma função dos estados de energia e utilização dos dispositivos.

$$\hat{y} = \sum_{i=1}^P y_i + Y_0 = \sum_{i=1}^P c_i(s_i, \mu_i) + Y_0 \quad (13)$$

Se é possível representar a componente de potência $c_i(s_i, \mu_i)$ como $b_i \cdot f_i(s_i, \mu_i)$, onde b_i é uma constante (chamada parâmetro) e f_i é uma função, para todo $i = 1, 2, \dots, P$, em seguida, substituindo $f_i(s_i, \mu_i)$ por u_i (chamado entrada), a expressão em (13) é reescrita como (14).

$$\hat{y} = \sum_{i=1}^P b_i u_i + Y_0 \quad (14)$$

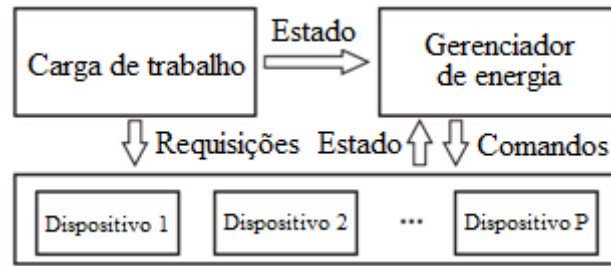


Figura 2: Diagrama de blocos de um sistema com gerenciamento de energia

O problema da estimação do modelo de potência de um sistema computacional é, portanto, a avaliação dos parâmetros b_i , $i = 1, 2, \dots, P$, e Y_0 . A estimação do consumo de potência \hat{y} , a potência estática Y_0 e as componentes b_i e u_i , para todo $i = 1, 2, \dots, P$ são não-negativos. Portanto, se as entradas $u_i \geq 0$, então, é possível considerar que $b_i \geq 0$, para todos os $i = 1, 2, \dots, P$. Assim, a estimativa dos parâmetros b_i , $i = 1, 2, \dots, P$, e Y_0 pode ser formulada como um problema de mínimos quadrados com restrições [7], utilizando-se a restrição $b_i \geq 0$, para todos os $i = 1, 2, \dots, P$ e $Y_0 \geq 0$.

Para avaliar os parâmetros, é possível realizar $m \in \mathbb{N}$ experimentos para adquirir os dados de identificação, o qual é composto de um vetor de medições de potência $Y = [y_1 \ \dots \ y_m]^T$ e uma matriz de entradas $U = [u_{ji}], j = 1, \dots, m, i = 1, \dots, P$. Usando (2) e da matriz de entradas, a relação entre um vetor de estimativas $\hat{Y} = [\hat{y}_1 \ \dots \ \hat{y}_m]^T$ e o vetor de parâmetros $\theta = [b_1 \ \dots \ b_P \ Y_0]^T$ é mostrado em (15).

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} u_{11} & \dots & u_{1P} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ u_{m1} & \dots & u_{mP} & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_P \\ Y_0 \end{bmatrix} \quad (15)$$

Seja $C = [U \ 1]_{m \times 1}$, então, (15) pode ser escrito como $\hat{Y} = C\theta$. O objetivo da estimação do modelo de potência do sistema computacional é encontrar os parâmetros do vetor θ que minimiza o quadrado da norma-2 da diferença entre o vetor de estimação e o vetor de medições de potência $\|\hat{Y} - Y\|_2^2$ de tal forma que cada elemento do vetor de parâmetros seja não-negativo, tal como mostrado em (16).

$$\min_{\theta} \|C\theta - Y\|_2^2 \quad \text{tal que } \theta \geq 0 \quad (16)$$

O algoritmo dos mínimos quadrados apresentado nesta seção, será a ferramenta matemática utilizada para fazer a identificação do modelo de potência da plataforma alvo especificada na seção seguinte.

3 PLATAFORMA EXPERIMENTAL

A seguir são apresentadas as características da plataforma alvo, e o sistema utilizado para medição de potência.

3.1 NOTEBOOK POSITIVO SELECT 8010

As características do *Notebook Positivo Select 8010*, ilustrado na Fig. 3 são as seguintes [8]:

- Número de série: 1A244Q525
- Processador Intel® *Core i7 620M*, máxima frequência - 2.66 GHz;
- *Windows 7 Home Basic*
- Monitor 14" WXGA (1366x768) TFT LCD, compatível com *Microsoft DirectX 10*
- Processamento de vídeo integrado, com suporte a gráficos 3D/2D
- Memória RAM 8 GB DDR3
- HD 640 GB
- *Webcam* de 1,3 megapixel e microfone integrados
- Unidade de DVD±RW *Dual Layer*
- Rede sem fio *Azurewave AW-NE 104H*
- Rede 10/100/1000 Mbps, *Gigabit Ethernet*
- Teclado Português Brasil, 88 teclas, 3 teclas de atalho
- *Mouse* tipo *Touchpad*, Toque múltiplo, *Scroll*, 2 botões
- 3 conectores USB, 1 conector VGA, 1 conector RJ-45, 1 conector HDMI, 1 soquete para leitor de cartão
- Bateria Li-ion, 4400mAh
- Peso 2,2 kg (com LCD, unidade ótica e bateria de 6 células)



Figura 3: Imagem do *notebook* Positivo Select 8010.

3.2 SISTEMA PARA MEDIÇÃO DE POTÊNCIA

A seguir são apresentados todos os componentes utilizados nas medições de tensão e corrente da plataforma alvo, para se fazer o estudo sobre o consumo de potência dos dispositivos.

3.2.1 CIRCUITO ADAPTADOR

Para medir a tensão entre os terminais da bateria/fonte e a corrente fornecida ao Notebook, pode ser utilizado o módulo de aquisição de dados NI USB-6210 conectado a um *computador pessoal* (PC) via *Universal Serial Bus* (USB), como ilustrado no diagrama de blocos da Fig 4. No PC foi utilizado o programa LabVIEW no sistema operacional Windows para aquisição dos dados fornecidos pelo NI USB-6210.

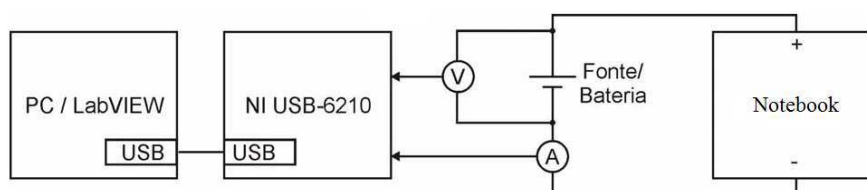


Figura 4: Diagrama de blocos para medição de tensão e corrente.

Para medição do valor v da tensão entre os terminais positivo (+) e negativo (-) do Notebook, foi utilizado um divisor resistivo com duas resistências $R_1 = 10055,3\Omega$ e $R_2 = 9945,0\Omega$, pois a tensão de saída da fonte do Notebook é de 19,4V, superior à máxima tensão de entrada suportada pelos canais analógicos do NI USB-6210. O canal de entrada analógico A/2 do módulo de aquisição de dados NI USB-6210 foi conectado ao nó entre os dois resistores R_1 e R_2 e o canal A/GND do NI USB-6210 foi conectado

ao terminal negativo (-) da bateria. Assim a tensão v_1 lida no canal A/2 é igual a $vR_1/(R_1 + R_2)$. No programa LabVIEW, este valor v_1 é multiplicado por $(R_1 + R_2)/R_1 = 2,011$, obtendo assim diretamente o valor v da tensão entre os terminais positivo (+) e negativo (-) do Notebook.

Para medir o valor i da corrente da bateria que flui do terminal negativo (-) do Notebook para o terminal negativo (-) da bateria/fonte, foi colocada uma resistência $R = 0,232\Omega$ entre esses dois terminais. O canal A/1 do NI USB-6210 foi conectado ao terminal negativo (-) do Notebook. Assim o canal A/1 do NI USB-6210 mede a tensão Ri sobre a resistência R , que é proporcional à corrente i . No programa LabVIEW, o valor da resistência R é especificado, e os valores adquiridos no canal A/1 são automaticamente divididos por R , já fornecendo diretamente os valores de corrente i , conforme representado no diagrama de blocos da Fig. 5.

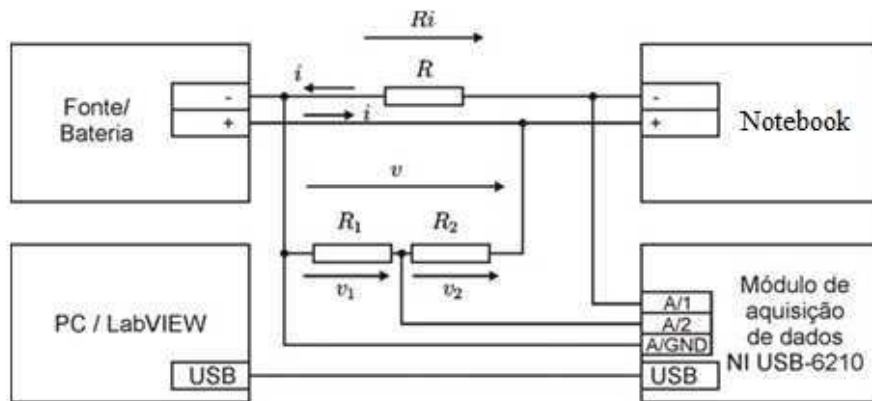


Figura 5: Diagrama elétrico para medição de tensão e corrente do Positivo Select através do módulo de aquisição de dados NI USB 6210.

3.2.2 MÓDULO DE AQUISIÇÃO DE DADOS NI USB – 6210

A aquisição de dados é o processo de medição de um fenômeno elétrico ou físico, como tensão, corrente, temperatura, pressão ou som, com o auxílio de um computador [9].

Para efetuar a medição de tensão e corrente fornecidas pela fonte ao Notebook, utilizou-se o sistema de aquisição de dados da *National Instruments*, o NI-USB-6210 ilustrado na Fig. 6, o qual se conecta a um computador via USB (*Universal Serial Bus*).



Figura 6: Imagem do Módulo de Aquisição de Dados NI USB 6210

3.2.3 LABVIEW

LabVIEW é uma linguagem de programação gráfica que utiliza ícones, em vez de linhas de texto, para criar aplicações. No LabVIEW, pode-se construir uma interface de usuário, utilizando um conjunto de ferramentas e objetos, o qual se destinará a aquisição e processamento de dados em tempo real.

No computador pessoal foi utilizado o programa LabVIEW no sistema operacional Windows para aquisição dos dados fornecidos pelo NI USB-6210.

O diagrama de blocos para aquisição de dados no programa LabVIEW é apresentado na Fig. 7.

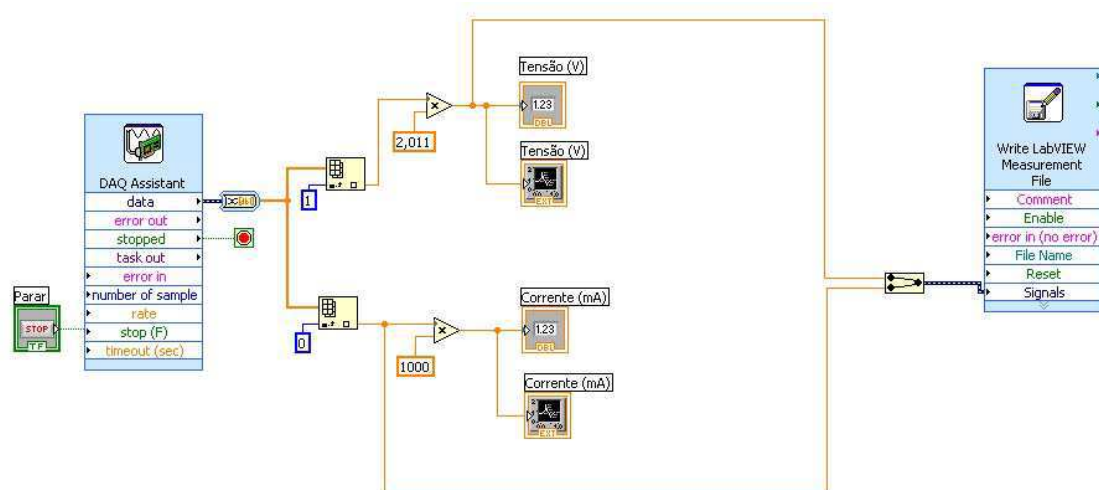


Figura 7: Diagrama de blocos do programa LabVIEW.

O bloco *DAQ Assistant* é o driver do LabVIEW com o NI USB-6210 e sua saída são os sinais de tensão e corrente, que são exibidos no painel frontal do LabVIEW em valor instantâneo e como forma de onda, conforme a Fig. 8. O bloco *Write LabVIEW Measurement File* salva em um arquivo de dados os valores de tensão e corrente obtidos pelo NI USB-6210. Em um script no Matlab, este arquivo é aberto e os dados são

tratados para obtenção do sinal de potência $p = vi$ ao longo do tempo, como também seu valor médio.

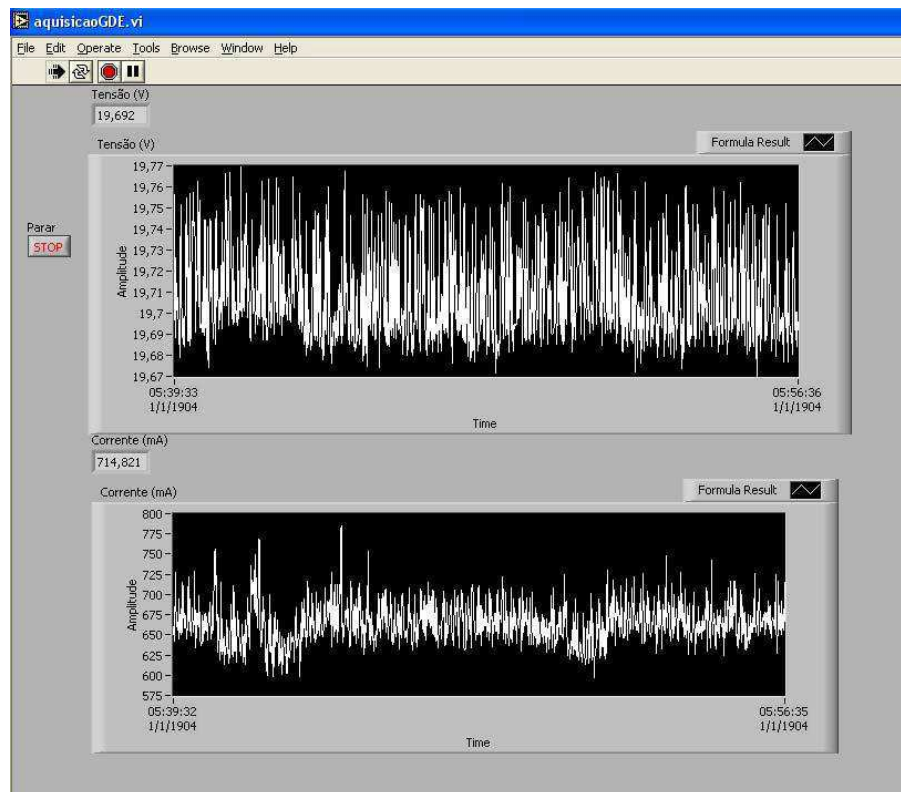


Figura 8: Representação do Painel frontal no programa LabVIEW

3.2.4 MATLAB

MatLab é um software interativo de alto desempenho voltado para o desenvolvimento de algoritmos, análise de dados, cálculo numérico, processamento de sinais além de várias outras funcionalidades voltadas para as áreas da engenharia.

O MatLab foi utilizado para fazer o tratamento estatístico dos dados obtidos a partir do módulo de aquisição de dados e dessa forma obter-se os valores de tensão, corrente e potência média, que foi o parâmetro utilizado na elaboração do modelo, como para a implementação do algoritmo dos mínimos quadrados, para estimação e validação do modelo de potência.

3.2.5 COMPONENTS LOGGER

Components Logger é um software desenvolvido em Linguagem de programação C#, e tem como função básica capturar os estados dos dispositivos do sistema, tais como brilho atual do LCD, frequência do processador, carga da CPU,

utilização de memória, utilização de HD, utilização de *wi-fi/ethernet* e taxa de *Download/Upload*. Na implementação do *Components Logger*, cada dispositivo possui uma classe que implementa a classe abstrata *ComponentDataReader*. O método *UpdateData* é implementado da forma mais adequada para cada dispositivo. Na Tabela 1, estão as principais funções e classes utilizadas por cada classe de dispositivos para leitura do seu respectivo estado, implementadas no método *UpdateData*.

Tabela 1: Métodos presentes na implementação do Components Logger a serem utilizados para a obtenção dos estados dos dispositivos [10].

Dispositivo	Função/Classe	Descrição
Processador	Win32_Processor.CurrentClockSpeed	Lê a frequência do processador.
Processador	Win32_Processor.LoadPercentage	Lê a carga média atual dos processadores (medida num intervalo de 1 segundo)
LCD	PowerReadACValueIndex, presente na biblioteca powrprof.dll. (para alimentação AC)	Lê o nível do brilho do LCD quando o computador estiver com alimentação AC.
LCD	PowerReadDCValueIndex, presente na biblioteca powrprof.dll. (para alimentação DC)	Lê o nível do brilho do LCD quando o computador estiver com alimentação DC.
HD	É usado um PerformanceCounter, com atributo <i>CategoryName</i> = "PhysicalDisk", com o atributo <i>CounterName</i> = "Disk Bytes/sec" e com o atributo <i>InstanceName</i> igual ao da unidade de disco equivalente ao HD (selecionado pelo usuário no próprio <i>ComponentsLogger</i>).	Lê a taxa de IO do HD em bytes/segundo.
Memória RAM	É usado um PerformanceCounter, com atributo <i>CategoryName</i> = "Memory" e com o atributo <i>CounterName</i> = "Available MBytes"	Lê a quantidade disponível de memória RAM. Sendo a quantidade ocupada

		igual à diferença entre o total de memória e o valor lido.
Taxa de <i>Upload</i> (usado para <i>Ethernet</i> e <i>Wi-Fi</i>)	É usado um PerformanceCounter, com atributo CategoryName = "Network Interface", com o atributo CounterName = "Bytes Sent/sec" e com o atributo InstanceName igual ao nome da interface de rede selecionada pelo usuário, no próprio ComponentsLogger.	Lê a taxa de <i>Upload</i> de uma interface de rede em bytes por segundo.
Taxa de <i>Download</i> (usado para <i>Ethernet</i> e <i>Wi-Fi</i>)	É usado um PerformanceCounter, com atributo CategoryName = "Network Interface", com o atributo CounterName = "Bytes Received/sec" e com o atributo InstanceName igual ao nome da interface de rede selecionada pelo usuário, no próprio ComponentsLogger.	Lê a taxa de <i>Download</i> de uma interface de rede em bytes por segundo.

O software *Components Logger* foi utilizado para realizar a obtenção dos estados dos dispositivos e posterior armazenamento em arquivo .txt. O painel frontal do *Components Logger* é apresentado na Fig. 9.

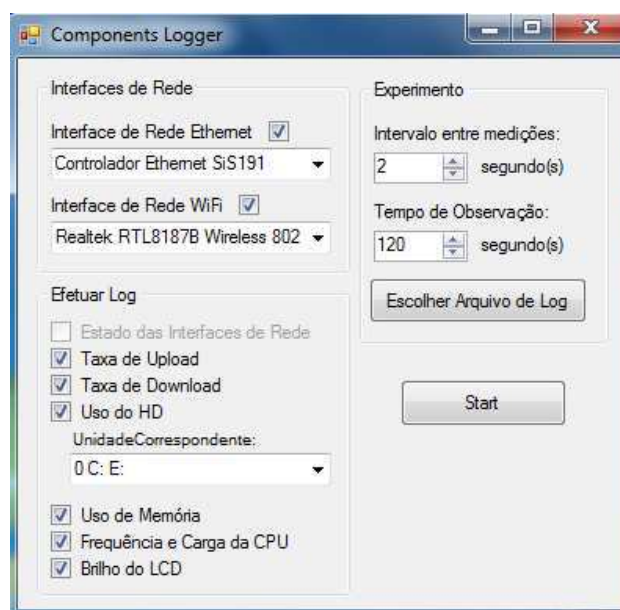


Figura 9: Representação do Painel frontal do *software Components Logger*.

4 PROCEDIMENTOS EXPERIMENTAIS

Nesta seção, são apresentados os recursos empregados para promover as alterações dos estados de energia dos dispositivos da plataforma alvo. A forma encontrada para medir a contribuição individual de cada dispositivo no consumo total de potência do sistema, foi realizar um conjunto de testes utilizando os vários dispositivos (entradas do sistema) individualmente, e em cada teste específico, mantendo os demais dispositivos desligados ou em seu estado de mais baixo consumo.

4.1 CARGAS DE TRABALHO

Considerando-se os dispositivos da plataforma experimental, a estimação do modelo de potência descrito na seção anterior, deve considerar as entradas u_i , $i = 0, 1, 2, \dots, P$, definidos na Tabela 2. Existem $P = 11$ entradas diferentes. Todos os experimentos foram realizados utilizando-se o sistema de medição e aquisição de dados descritos na seção 3.2

Tabela 2: Descrição das entradas do sistema.

Entradas	Descrição	Valores possíveis	Unidade/Estado
u_0	Brilho do LCD	$u_0 \in \{0; 0,14; 0,28; 0,42; 0,56; 0,7; 0,85; 1,0\}$	%
u_1	<i>Ethernet</i> ligada	$u_1 \in \{0, 1\}$	0 – Desligado 1 – Ligado
u_2	<i>Wi-fi</i> ligada	$u_2 \in \{0, 1\}$	0 – Desligado 1 – Ligado
u_3	<i>Webcam</i> ligada	$u_3 \in \{0, 1\}$	0 – Desligado 1 – Ligado
u_4	Unidade de DVD_RW ligada	$u_4 \in \{0, 1\}$	0 – Desligado 1 – Ligado
u_5	Taxa de <i>Download</i> da <i>Ethernet</i>	$u_5 \geq 0$	B/s
u_6	Taxa de <i>Upload</i> da <i>Ethernet</i>	$u_6 \geq 0$	B/s

u_7	Taxa de <i>Download</i> da <i>Wi-fi</i>	$u_7 \geq 0$	B/s
u_8	Taxa de <i>Upload</i> da <i>Wi-fi</i>	$u_8 \geq 0$	B/s
u_9	Taxa de transferência de dados do HD	$u_9 \geq 0$	B/s
u_{10}	Alocação de memória	$u_{10} \geq 0$	MB
u_{11}	Entrada proporcional a componente de potência dinâmica do processador ¹ $0,01 \cdot cpu_{utilização}(\%)$ $\cdot cpu_{frequencia}^3$	$u_{11} \geq 0$	(GHz) ³
1 - A potência dinâmica do processador é proporcional à atividade de chaveamento dos transistores, a uma carga capacitiva C , ao quadrado da tensão v , e a frequência f . Desse modo atividade de chaveamento pode ser obtida por meio da utilização da CPU, e a tensão pode ser aproximada pela frequência, já que em geral a frequência é proporcional à tensão.			

Para estimar os parâmetros do modelo b_i , $i = 0, 1, 2, \dots, P$, e Y_0 , os experimentos foram realizados variando o brilho do LCD (de 0% a 100%) de acordo com os intervalos especificados na Tabela 2, variando a utilização da CPU (de 15% a 95%), realizando operações de *download* e *upload* com as interfaces de rede *Ethernet* e *Wi-Fi*, realizando alocação de memória, utilizando a *Webcam* e a Unidade de leitura ótica (DVD_RW) da plataforma alvo. Os experimentos estão detalhados no Anexo A.

O *software Components Logger*, foi utilizado para medir e armazenar as entradas u_i , $i = 0, 1, \dots, P$, tal como definido na Tabela 2. O período de amostragem para a medição foi de 2 segundos. Um período de amostragem, menor do que 2 segundos aumentaria a utilização do processador pelo *software Components Logger*, afetando assim as medições do processador.

Para exemplificar a metodologia utilizada na identificação do modelo de potência, a seção seguinte ilustra detalhadamente os testes feitos com o LCD.

4.1.1 TESTES VARIANDO O BRILHO DO LCD

Os testes com LCD foram realizados desligando-se todos os demais dispositivos do sistema, tais como os *drivers* de *Ethernet*, *Wi-fi*, *Webcam*, Unidade de leitura ótica, sem nenhuma carga de trabalho adicional a não ser o próprio funcionamento do sistema, e em seguida variando-se o brilho de tela nos intervalos (0%, 14%, 28%, 42%, 56%,

70%, 84% e 100%) em cada um dos testes de 2 minutos. Realizados os testes, os dados foram tratados a partir de um *script* criado no ambiente MatLab conforme Anexo C.1. Os valores obtidos para esse conjunto de testes são os valores médios dos parâmetros conforme especificado na Tabela 3.

Tabela 3: Potência média consumida e Valores médios dos parâmetros dos dispositivos.

Valores Médios Brilho LCD	0%	14%	28%	42%	56%	70%	84%	100%
Potência (W)	14,764	14,809	14,962	15,042	15,340	16,065	17,099	17,111
HD (Bytes/s)	5558,1	4250,6	40605,5	1213,4	1789,1	2319,2	2518,0	3147,8
Memória (MB)	1159,5	1157,9	1155,5	1154,9	1158,2	1158,6	1156,9	1160,2
CPU_carga (%)	0,0857	0,0913	0,1083	0,1018	0,092	0,1037	0,0935	0,0975
Freq_cpu (MHz)	1,2197	1,2397	1,2668	1,2107	1,2152	1,2646	1,2108	1,2419
Down_Eth (B/s)	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Up_Eth (B/s)	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Down_wifi (B/s)	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Up_wifi (B/s)	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Webcam	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Unid_DVD_RW	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

A mesma metodologia de testes foi aplicada para os demais dispositivos do sistema. Todos os testes realizados estão discriminados no Anexo A.

Uma vez realizado o conjunto de testes para cada dispositivo, aplica-se o algoritmo de identificação pelo método dos mínimos quadrados com restrições. Os resultados obtidos são apresentados na próxima seção.

5 RESULTADOS OBTIDOS

Neste capítulo são apresentados os resultados experimentais obtidos mediante aplicação do algoritmo de mínimos quadrados com restrições, empregado na identificação do modelo de potência de um sistema computacional

5.1 IDENTIFICAÇÃO DO MODELO DE POTÊNCIA

Para aplicar o algoritmo de identificação pelo método dos mínimos quadrados com restrições, foi utilizado um *script* no ambiente MatLab utilizando a função *lsqlin()*, detalhada no Anexo C.2, com o intuito de obter os parâmetros b_i , $i = 1, 2, \dots, P$, e Y_0 do modelo de potência do sistema, resultando nos valores mostrados na Tabela 4.

Tabela 4: Parâmetros obtidos a partir da estimação por mínimos quadrados com restrições

Parâmetros	Valores	Componente do sistema
Y_0	14,6926	Operações básicas do sistema
b_0	0,2979	Wi-fi_on
b_1	0,3471	Ether_on
b_2	0,7743	WebCam_on
b_3	2,0858	DVD_RW_on
b_4	1,7581	CPU_utilização
b_5	$6,3237 \cdot 10^{-7}$	Hd
b_6	$3,3134 \cdot 10^{-14}$	Memória
b_7	1,6831	LCD
b_8	$6,4120 \cdot 10^{-14}$	Down_ether
b_9	$3,7562 \cdot 10^{-7}$	Up_ether
b_{10}	$8,5077 \cdot 10^{-7}$	Down_wi-fi
b_{11}	$1,7054 \cdot 10^{-6}$	Up_wi-fi

A partir dos parâmetros obtidos pelo método dos mínimos quadrados com restrições, propõem-se o seguinte modelo para obtenção da estimativa de potência do sistema, conforme Equação (17):

*Potência*_{estimada}

$$\begin{aligned}
 &= Y_0 + b_0 \cdot Wifi_{ligada} + b_1 \cdot Ethernet_{ligada} + b_2 \\
 &\cdot Webcam_{ligada} + b_3 \cdot DVD_{RW_{ligado}} \\
 &+ b_4 \left(0,01 \cdot cpu_{utilização(\%)} \cdot Freq_{CPU}^3 \right) + b_5 \cdot HD_{taxa} + b_6 \quad (17) \\
 &\cdot Mem_{alocada} + b_7 \cdot LCD_{(\%)brilho} + b_8 \cdot Eth_{download} \\
 &+ b_9 \cdot Eth_{upload} + b_{10} \cdot Wifi_{download} + b_{11} \cdot Wifi_{upload}
 \end{aligned}$$

A Tabela 5 relaciona as entradas dos sistema com as variáveis especificadas na Equação (17).

Tabela 5: Relação das entradas com variáveis especificadas na Equação 17.

Entradas	Descrição	Variável relacionada na equação (17)
u_0	Brilho do LCD	$LCD_{(\%)brilho}$
u_1	<i>Ethernet</i> ligada	$Ethernet_{ligada}$
u_2	<i>Wi-fi</i> ligada	$Wifi_{ligada}$
u_3	<i>Webcam</i> ligada	$Webcam_{ligada}$
u_4	Unidade de DVD_RW ligada	$DVD_{RW_{ligado}}$
u_5	Taxa de <i>Download</i> da <i>Ethernet</i>	$Eth_{download}$
u_6	Taxa de <i>Upload</i> da <i>Ethernet</i>	Eth_{upload}
u_7	Taxa de <i>Download</i> da <i>Wi-fi</i>	$Wifi_{download}$
u_8	Taxa de <i>Upload</i> da <i>Wi-fi</i>	$Wifi_{upload}$
u_9	Taxa de transferência de dados do HD	HD_{taxa}
u_{10}	Alocação de memória	$Mem_{alocada}$
u_{11}	Entrada proporcional a componente de potência dinâmica do processador ¹ $0,01 \cdot cpu_{utilização(\%)} \cdot cpu_{frequencia}^3$	$0,01 \cdot cpu_{utilização(\%)} \cdot Freq_{CPU}^3$

E com isso obteve-se o modelo de potência do sistema, conforme Fig. 10 a 13, no qual os gráficos evidenciam o ajuste do modelo aos parâmetros medidos para cada dispositivo individualmente.

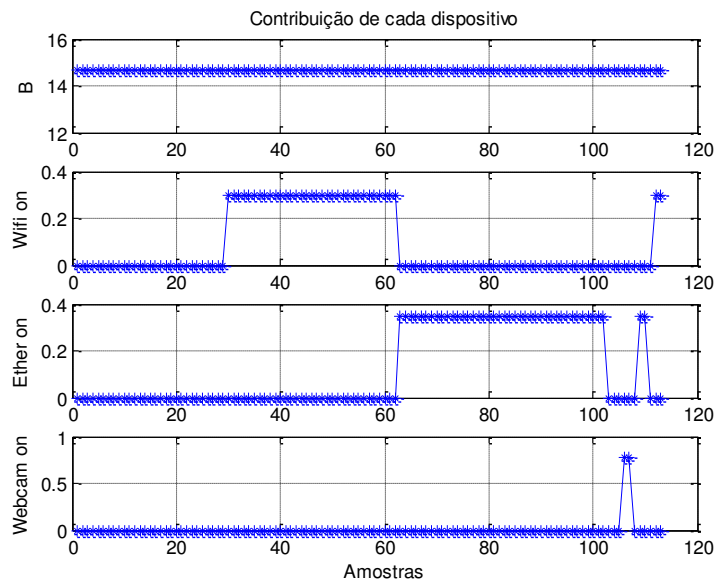


Figura 10: Contribuição individual dos dispositivos para obtenção do modelo

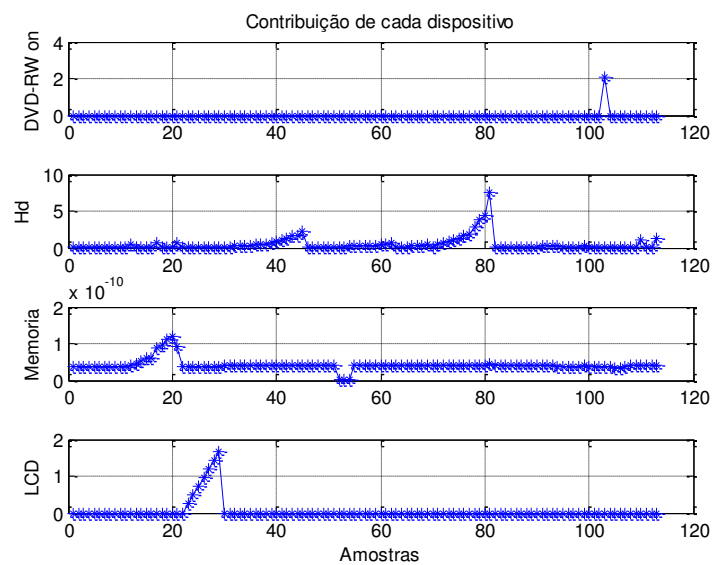


Figura 11: Contribuição individual dos dispositivos para obtenção do modelo

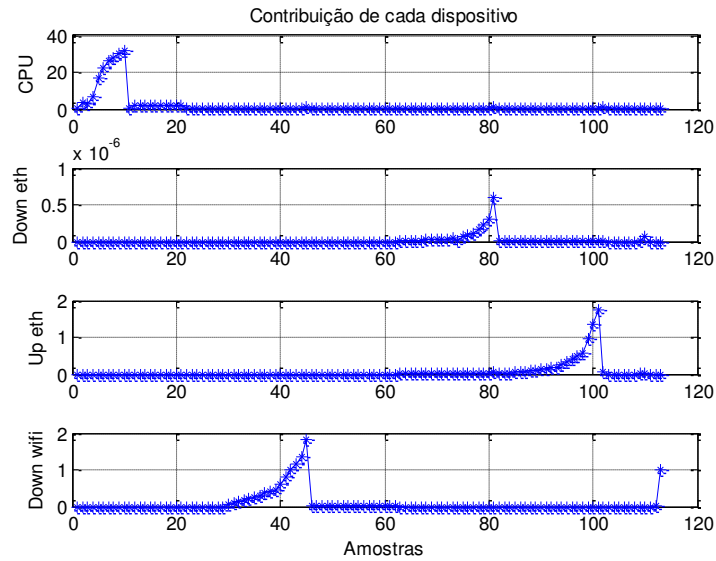


Figura 12: Contribuição individual dos dispositivos para obtenção do modelo

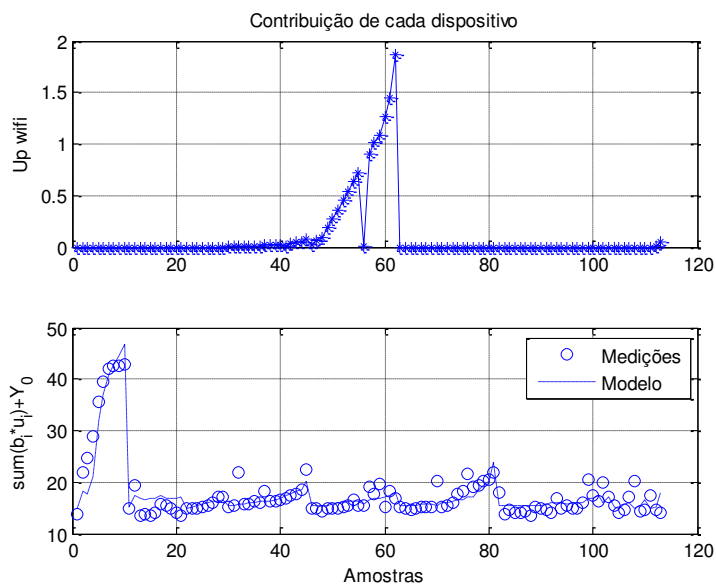


Figura 13: Contribuição individual dos dispositivos para obtenção do modelo

Observando as Fig. 10 a 13 podemos concluir que a contribuição de cada dispositivo para construção do gráfico representado na Fig. 14, pode-se ser evidenciada a partir das amostras relacionadas a cada experimento descritas a seguir:

- Amostras de 0 a 11: teste com CPU;
 - Amostras de 12 a 22: teste com memória;
 - Amostras de 23 a 30: teste com LCD;
 - Amostras de 31 a 63: teste com *Wi-fi* ligada
- $\left\{ \begin{array}{l} 31 \text{ a } 46: \textit{download} ; \\ 47 \text{ a } 63: \textit{upload}; \end{array} \right.$

Amostras de 64 a 102: teste com *Ethernet* ligada

{ 64 a 82: *download*;
 { 83 a 102: *upload*;

Amostras de 102 a 104: teste com *DVD_RW*;

Amostras de 105 a 108: teste com *WebCam*;

Amostras de 109 a 111: teste de *download* com *Ethernet*;

Amostras de 111 a 113: teste de *download* com *Wi-fi*,

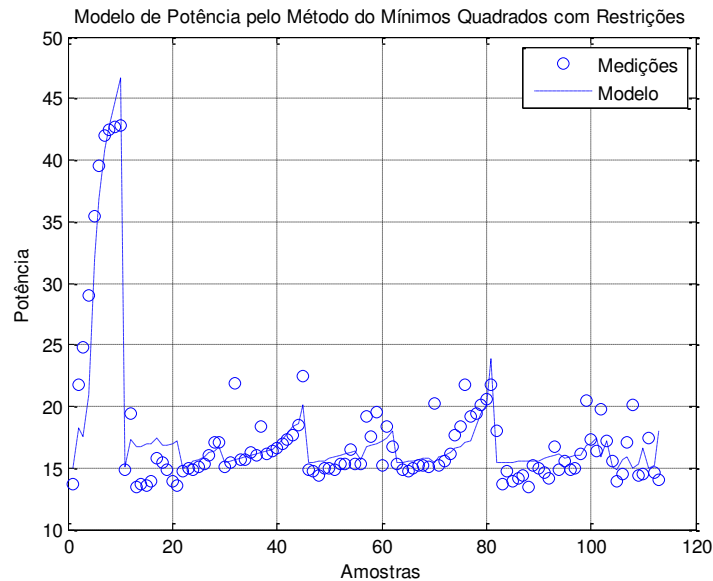


Figura 14: Gráfico do modelo de potência do *Notebook Positivo Select*.

Referenciando os testes descritos nos anexos tem-se:

Amostras de 0 a 11: Anexo A5;

Amostras de 12 a 22: Anexo A4;

Amostras de 23 a 30: Anexo A1;

Amostras de 31 a 63: teste com *Wi-fi* ligada

{ 31 a 46: Anexo A.6.1;

{ 47 a 63: Anexo A.6.3;

Amostras de 64 a 102: teste com *Ethernet* ligada

{ 64 a 82: Anexo A.6.2;

{ 83 a 102: Anexo A.6.4

Amostras de 102 a 104: Anexo A2;

Amostras de 105 a 108: Anexo A3;

Amostras de 109 a 111: Anexo A.6.6

Amostras de 111 a 113: Anexo A.6.5

5.2 VALIDAÇÃO DO MODELO

Para validação do modelo foram realizados dois testes com intuito de coletar os dados de entrada, ou seja, todos os parâmetros contidos na Tabela 2, bem os dados de potência consumida pelo sistema. De tal forma que fosse possível aplicar o modelo obtido aos dados de entrada do sistema e fazer a comparação da potência estimada pelo modelo com a potência medida na realização do experimento, para tanto, foi elaborado um script no MatLab, detalhado no Anexo C.3.

O objetivo dos experimentos foi variar a carga interna do sistema, tal como ativação/desativação de dispositivos, variação no brilho de tela, aumento de uso de CPU, entre outros, com o intuito de observar o comportamento do modelo e definir se o método utilizado é realmente eficiente na modelagem de sistemas como o descrito neste trabalho.

A descrição dos experimentos para validação do modelo está contida no Anexo B, a priori os testes foram realizados de modo que a carga de trabalho do sistema fosse modificada a cada minuto de teste, durante o intervalo total de 10 minutos do experimento.

A partir dos experimentos realizados obtiveram-se os seguintes resultados, apresentados nas Fig. 15, 16 e Tabela 6, para o teste 1, e Fig. 17, 18 e Tabela 7 para o teste 2.

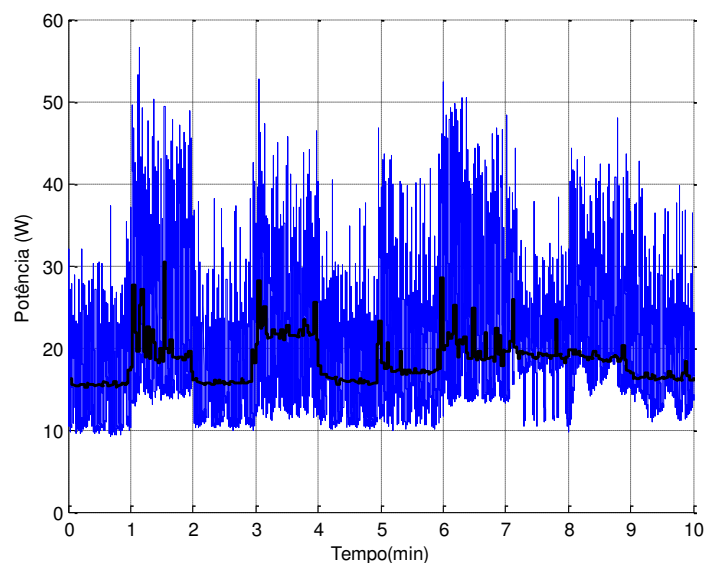


Figura 15: Consumo de potência medida e estimada – teste 1.

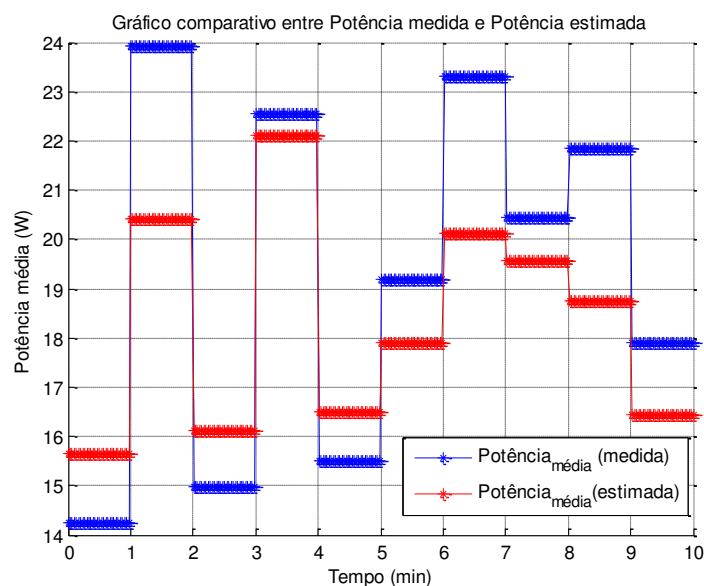


Figura 16: Comparativo entre as potências médias – teste 1.

Tabela 6: Erro percentual para cada intervalo de medição – teste 1.

Tempo	Potência medida (W)	Potência estimada (W)	$erro \% = \frac{ Pot_{medida} - Pot_{estimada} }{Pot_{medida}} \cdot 100$
0 – 1min	14,2376	15,6506	9,92
1 – 2min	23,913	20,4262	14,58
2 – 3min	14,992	16,1340	7,61
3 – 4min	22,541	22,1073	1,92
4 – 5min	15,527	16,5073	6,31
5 – 6min	19,210	17,8987	6,68
6 – 7min	23,318	20,1377	13,63
7 – 8min	20,449	19,5562	4,36
8 – 9min	21,831	18,7592	14,07
9 – 10min	17,919	16,4534	8,17
Erro percentual médio			8,72

No experimento 1, pôde-se observar que a potência média estimada pelo modelo apresentou erro percentual máximo de 14,58% no momento em que o sistema passou de uma carga de trabalho mínima para uma carga de trabalho alta, conforme apresentado na Fig. 16 e Tabela 6. O erro percentual médio foi de 8,72%, portanto, não apresentou uma boa estimativa do consumo de potência para o experimento em questão.

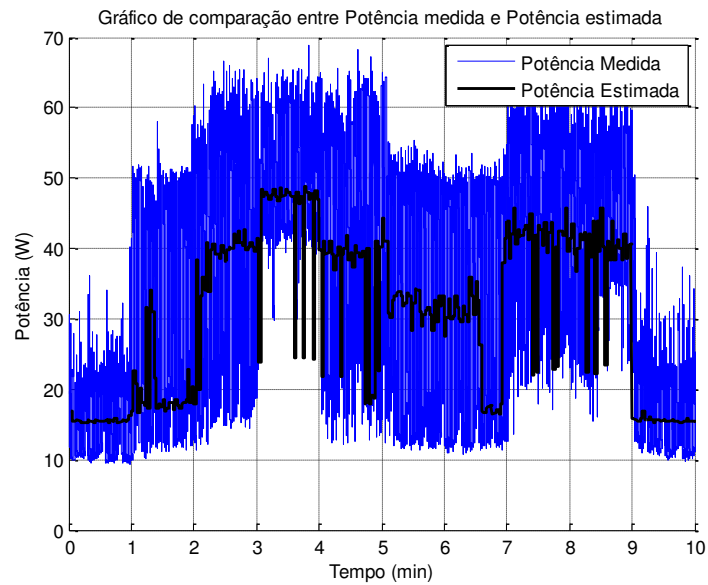


Figura 17: Consumo de potência medida e estimada – teste 2.

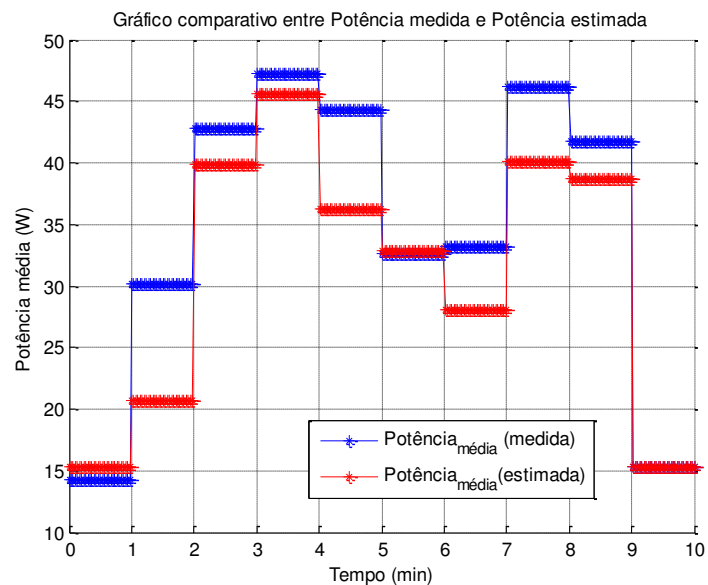


Figura 18: Comparativo entre as potências médias – teste 2.

Tabela 7: Erro percentual obtido em cada intervalo de medição – teste 2.

Tempo	Potência medida	Potência estimada	$erro \% = \frac{ Pot_{medida} - Pot_{estimada} }{Pot_{medida}} \cdot 100$
0 – 1min	14,2269	15,3540	7,92
1 – 2min	30,1796	20,6610	31,53
2 – 3min	42,7774	39,9169	6,68
3 – 4min	47,2176	45,5811	3,46
4 – 5min	44,2735	36,1996	18,23
5 – 6min	32,6062	32,9103	0,932

6 – 7min	33,1681	28,0450	15,44
7 – 8min	46,2111	40,1172	13,18
8 – 9min	41,7977	38,7451	7,30
9 – 10min	15,3447	15,3631	0,119
Erro percentual médio			10,47

Observando a Fig. 18 e a Tabela 7, fica evidente que a potência média estimada pelo modelo no experimento 2, apresentou um erro percentual bastante significativo na maior parte das comparações entre a potência estimada e a potência medida nos experimentos, chegando a um erro percentual máximo de 31,53% e médio de 10,47%.

De modo geral o modelo de potência identificado pelo método de mínimos quadrados com restrições não apresentou uma boa estimativa da potência consumida pelo sistema.

6 CONCLUSÃO

Neste trabalho, foi apresentado um algoritmo para estimação do modelo de potência de um sistema computacional, conhecido como mínimos quadrados com restrições. Este último termo refere-se ao fato de que os parâmetros do modelo estão todas associados a componentes de potência, portanto existe a restrição de serem todos não-negativos.

A partir de um conjunto de experimentos, com o intuito de coletar os dados relativos aos estados dos dispositivos do sistema computacional, tais como brilho do LCD, HD (Disco Rígido), Unidade de leitura ótica, *Webcam*, processador e interfaces de rede, foi possível aplicar o algoritmo dos mínimos quadrados com restrições e obter um modelo que estimasse o comportamento do sistema relativo à potência consumida. Por fim, testes para validação do modelo foram realizados e os resultados obtidos apresentaram erros percentuais em torno de 8,72% e 10,47% em cada um dos experimentos realizados.

No decorrer deste trabalho, disciplinas cursadas durante a graduação como Introdução a Programação, Cálculo Numérico, Circuitos Elétricos, Eletrônica, Controle Digital, Sistemas de Aquisição de Dados e Interface exerceram um papel bastante importante para a resolução do problema proposto.

Sabe-se, contudo que os avanços não param e torna-se cada vez mais necessário, experimentar novos métodos para resolução dos problemas de engenharia.

Com isso, coloca-se como proposta para trabalhos futuros:

- Investigar o motivo pelo qual o modelo não apresentou resultados satisfatórios;
- Aplicar outros métodos de identificação para o modelo do sistema proposto no trabalho, de modo a escolher um método mais eficiente para este tipo de aplicação.

ANEXO A – DESCRIÇÃO DOS EXPERIMENTOS

Regras gerais para todos os testes a seguir:

- Selecionar o perfil Equilibrado do *Windows 7*;
- Desconectar a bateria do *notebook*, deixando este apenas ligado à fonte, conforme especificado no diagrama elétrico da Fig. 4;
- Quando for mencionado “dispositivos”, leia-se todos os dispositivos contidos na Tabela 2;
- Ocioso, indica que todos os dispositivos estão desligados ou quando não for possível, está em seu estado de mais baixo consumo.
- Para coletar os dados referentes aos estados dos dispositivos do sistema, executar o *Components Logger*, marcar todos os *check boxes* da tela, escolher intervalos entre medições como 2s e tempo de observação como 120s. Em seguida escolher arquivo de *log* e salva-lo com extensão .txt. Os dados serão armazenados neste arquivo.
- Para medição de tensão e corrente a partir do módulo de aquisição de dados, executar o LabVIEW, abrir o Instrumento Virtual (VI) aquisiçãoGDE.vi.
- Em todos os testes, o *Components Logger* e o aquisiçãoGDE.vi devem ser iniciados simultaneamente.

ANEXO A1 - TESTES VARIANDO O BRILHO DE TELA

Regras gerais para todos os testes com este dispositivo:

- Deixar o background como a área de trabalho padrão do positivo Select 8010.
- Desligar todos os demais dispositivos do sistema, a partir do gerenciador de dispositivos;

- Realizar testes de 2 minutos variando apenas o brilho de tela, sem acrescentar carga adicional nenhuma ao sistema.
- Os intervalos percentuais foram obtidos variando-se o brilho de tela a partir dos botões físicos $F_n + F8$ (diminuir o brilho) ou $F_n + F9$ (aumentar o brilho), em seguida verificava-se o percentual definido em: (Opções de energia → Configurações avançadas → Brilho do vídeo).

Teste 1: Brilho de tela em 0%.

Teste 2: Brilho de tela em 14%.

Teste 3: Brilho de tela em 28%.

Teste 4: Brilho de tela em 42%.

Teste 5: Brilho de tela em 56%.

Teste 6: Brilho de tela em 70%.

Teste 7: Brilho de tela em 84%.

Teste 8: Brilho de tela em 100%.

ANEXO A2 - TESTES COM A UNIDADE DE DVD_RW

Regras gerais para todos os testes com este dispositivo:

- Desligar todos os demais dispositivos do sistema, a partir do gerenciador de dispositivos;
- Auto-falantes no menor nível, sem áudio;
- Realizar teste de 2 minutos inserindo uma mídia de áudio (CD) na Unidade DVD_RW, e reproduzir uma música com o *Windows Media Player*;

Teste 1: Ocioso, todos os dispositivos desligados, inclusive a Unidade de DVD_RW;

Teste 2: Ativar a Unidade de DVD_RW, e reproduzir uma música do cd de áudio;

Teste 3: Ocioso, a Unidade de DVD_RW permanece ativa, mas não utilizada.

ANEXO A3 - TESTES COM A *WEBCAM*

Regras gerais para todos os testes com este dispositivo:

- Desligar todos os demais dispositivos do sistema, a partir do gerenciador de dispositivos;
- Realizar teste de 2 minutos executando o software Mycam da Positivo.

Teste 1: Ocioso, todos os dispositivos desligados;

Teste 2: *Webcam* ligada, executando o *software* Mycam;

Teste 3: *Webcam* ligada, mas não utilizada

ANEXO A4 - TESTES COM ALOCAÇÃO DE MEMÓRIA

Regras gerais para todos os testes com este dispositivo:

- Desligar todos os dispositivos do sistema, a partir do gerenciador de dispositivos;
- A alocação de memória deve ser realizada com o software MemAlloc, o qual consegue apenas complementar o espaço de memória reservada no sistema para o valor a qual se deseja reservar.
- Realizar teste de 2 minutos executando o MemAlloc e variando a memória alocada do sistema em 50MB para cada teste, de 450MB até 900MB.

Teste 1: Deixar a memória do sistema em 450MB e realizar o teste de 2 minutos;

Teste 2: Deixar a memória do sistema em 500MB e realizar o teste de 2 minutos;

•
•
•

Teste 10: Deixar a memória do sistema em 900MB e realizar o teste de 2 minutos;

Teste 11: Deixar a memória do sistema no máximo possível e realizar o teste de 2 minutos;

ANEXO A5 - TESTES COM CARGA DE CPU

Regras gerais para todos os testes com este dispositivo:

- Desligar todos os dispositivos do sistema, a partir do gerenciador de dispositivos;
- A variação da carga de trabalho deve ser realizada com o *software* CPU Stress, ilustrado na Fig. 19, o qual consegue manter a carga da CPU estável ao longo de todo o teste;
- O número de *threads* a ser selecionado no *software* deve corresponder ao número de núcleos utilizados pelo *notebook*;
- Realizar teste de 2 minutos executando o CPU Stress variando a carga do sistema em 10% para cada teste, de 15% a 95%;

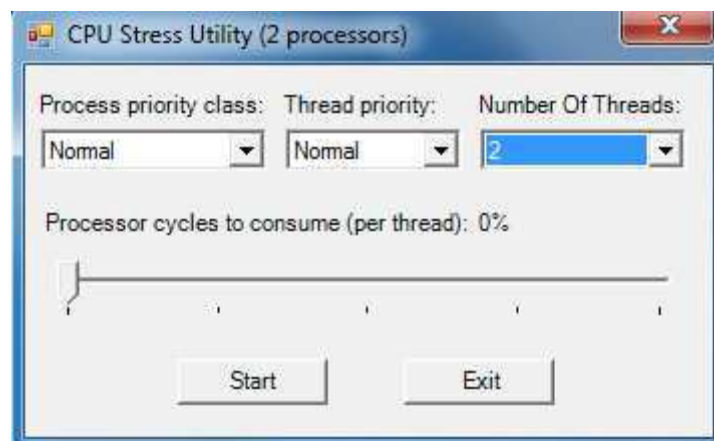


Figura 19: Representação do gerador de carga CPU Stress Utility.

Teste 1: Ocioso, sem carga de trabalho;

Teste 2: Executar o CPU Stress com carga de 15%, realizar o teste de 2 minutos;

•
•
•

Teste 10: Executar o CPU Stress com carga de 95%, realizar o teste de 2 minutos;

Teste 11: Reproduzir a música *Just a Dream* com o *Windows Media Player* e realizar o teste de 2 minutos, sem nenhuma outra carga adicional;

Teste 12: Reproduzir o vídeo WildLife com o *Windows Media Player* e realizar o teste de 2 minutos, sem nenhuma outra carga adicional;

ANEXO A.6 - TESTES COM AS INTERFACES DE REDES, *ETHERNET E WI-FI*

Regras gerais para todos os testes com estes dispositivos:

- Configurar uma rede interna com servidor dedicado, em um segundo computador, para isto pode ser utilizado o Apache Tomcat que funciona como um servidor *web* Java.
- Conectar este servidor a um roteador com capacidade de realizar conexões via rede *Ethernet* e *Wi-fi*, possibilitando a realização dos testes por estes dois tipos de conexões;
- Para realização destes testes, o arquivo utilizado para *download* deve levar em consideração o seu tamanho, uma vez que em testes onde a taxa de *download* for muito alta, o *download* poderá ser concluído antes do tempo previsto para realização do teste, que seria de 2 minutos;
- Nestes experimentos foram utilizados o arquivo *visual_studio.iso* o qual possuía um tamanho de 3 GB, sendo suficiente para a realização dos testes, sem que o *download* fosse concluído antes do término do experimento;
- Realizar teste de 2 minutos fazendo o *download* de um arquivo armazenado no servidor, através do *software Free Download Manager*, *software* que tem como função limitar a taxa de *download*;
- Desligar todos os dispositivos do sistema, a partir do gerenciador de dispositivos;
- Para cada teste variar a taxa de *download* como descrito nos testes, deixando-a em um patamar constante em cada um deles.

ANEXO A.6.1 – TESTES COM *DOWNLOAD* PELA *WI-FI*

Testes 1 a 10: Iniciar o *download* a partir do *Free Download* com taxa de *download* de 50 kB/s. Variar a taxa de 50kB/s de uma teste para o outro, terminando com 500 kB/s no teste 10.

Testes 11 a 15: Mesmo procedimento anterior com variação de taxa de 200kb/s de um teste para o outro, iniciando com 700 kB/s no teste 11 e terminando com 1500 kb/s no teste 15.

Teste 16: Iniciar o *download* a partir do *Free Download* com taxa de *download* ilimitada.

ANEXO A.6.2 – TESTES COM *DOWNLOAD* PELA *ETHERNET*

Testes 1 a 10: Iniciar o *download* a partir do *Free Download* com taxa de *download* de 50 kB/s. Variar a taxa de 50kB/s de um teste para o outro, terminando com 500 kB/s no teste 10.

Testes 11 a 15: Mesmo procedimento anterior com variação da taxa de 200kB/s de uma teste para o outro, iniciando com 700 kB/s no teste 11 e terminando com 1500 kB/s no teste 15.

Testes 16 à 19: Mesmo procedimento anterior com variação da taxa de 1000 kB/s de uma teste para o outro, iniciando com 2500 kB/s o teste 16 e terminando com 5500 kB/s no teste 19.

Teste 20: Iniciar o *download* a partir do *Free Download* com taxa de *download* ilimitada.

ANEXO A.6.3 – TESTES DE *UPLOAD* PELA *WI-FI*

Mesmo procedimento dos testes anteriores, só que neste caso o servidor Apache Tomcat foi instalado no próprio *notebook* Positivo, e o *download* será feito a partir de uma segunda máquina da rede interna, a qual limitará a taxa de *download* através do Free Download Manager, ou seja, será simulado *upload* com taxa limitada a partir do *notebook* Positivo pela interface de rede *Wi-fi*.

ANEXO A.6.4 – TESTES DE *UPLOAD* PELA *ETHERNET*

Mesmo procedimento dos testes anteriores, só que neste caso o servidor Apache Tomcat foi instalado no próprio *notebook* Positivo, e o *download* será feito a partir de uma segunda máquina da rede interna, a qual limitará a taxa de *download* através do Free Download Manager, ou seja, será simulado *upload* com taxa limitada a partir do *notebook* Positivo pela interface de rede *Ethernet*.

ANEXO A.6.5 – TESTES COM ATIVAÇÃO/DESATIVAÇÃO DA *WI-FI*

Teste 1: Ocioso, todos os dispositivos desligados;

Teste 2: *Wi-fi* ligada, mas não utilizada;

Teste 13: *Wi-fi* ligada, fazendo *download* a partir do *Free Download* com taxa de *download* ilimitada.

ANEXO A.6.6 – TESTES COM ATIVAÇÃO/DESATIVAÇÃO DA *ETHERNET*

Teste 1: Ocioso, todos os dispositivos desligados;

Teste 2: *Ethernet* ligada, mas não utilizada;

Teste 13: *Ethernet* ligada, fazendo *download* a partir do *Free Download* com taxa de *download* ilimitada.

ANEXO B - TESTES PARA VALIDAÇÃO DO MODELO DE POTÊNCIA

Regras gerais para todos os testes a seguir: mesmo procedimento utilizado no Anexo A.

ANEXO B.1 - CARGA VARIÁVEL 1

O teste foi realizado num intervalo de tempo de 10 minutos, variando a carga de trabalho do sistema a cada 1 minuto de teste, conforme descrito a seguir:

- 0 min – LCD 0%, todos os dispositivos desligados;
- 1 min – LCD 100%, *Wi-fi* ativa (sem utilização), reprodução do vídeo *WildLife* com *Windows Media Player*;
- 2 min – LCD 0%, *Wi-fi* ativa (sem utilização), sem vídeo.
- 3 min – LCD 70%, *Wi-fi* (*download* de arquivo da rede interna), reprodução da música *Just a dream* com *Windows Media Player*;
- 4 min – LCD 0%, *Wi-fi* apenas ativa (sem utilização), sem música;
- 5 min – LCD 84%, *Wi-fi* (desativada), reprodução da música *Just a dream* com *Windows Media Player*.
- 6 min – LCD 100%, *Ethernet* ativa (sem utilização), reprodução do vídeo *WildLife* com *Windows Media Player*
- 7 min – LCD 0%, *Ethernet* ativa (*download* do arquivo da rede interna), webcam ativa (mas sem utilização)
- 8 min – LCD 100%, *Ethernet* ativa (sem utilização), *Webcam* sendo utilizada por meio do *software* *Mycam*.
- 9 min – LCD 0%, ocioso;
- 10 min – Fim.

ANEXO B.2 - CARGA VARIÁVEL 2

O teste foi realizado num intervalo de tempo de 10 minutos, variando a carga de trabalho do sistema a cada 1 minuto de teste, conforme descrito a seguir:

- 0 min –, LCD 0%, ocioso;
- 1 min – LCD 70%, *Wi-fi* ativa (sem utilização), reprodução da música *Just a dream* com *Windows Media Player*, Execução do *CPU Stress* com 30% de carga;
- 2 min – LCD 100%, *Wi-fi* ativa (vídeo no youtube [11]), Execução do *CPU Stress* com 60% de carga;
- 3 min – LCD 100%, *Wi-fi* ativa (vídeo no youtube), Execução do *CPU Stress* com 90% de carga;
- 4 min – LCD 84%, *Wi-fi* (vídeo no youtube), Execução do *CPU Stress* com 60% de carga;
- 5 min – LCD 28%, *Wi-fi* ativa (sem utilização), Execução do *CPU Stress* com 30% de carga;
- 6 min – LCD 28%, *Wi-fi* ativa (sem utilização), Execução do *CPU Stress* com 30% de carga;
- 7 min – LCD 28%, *Wi-fi* ativa (vídeo no youtube), Execução do *CPU Stress* com 60% de carga;
- 8 min – LCD 28%, *Wi-fi* ativa (sem utilização), reprodução do vídeo *WildLife* com *Windows Media Player*, Execução do *CPU Stress* com 60% de carga;
- 9 min – LCD 0%, ocioso;
- 10 – Fim.

ANEXO C - SCRIPTS DO MATLAB

ANEXO C.1 - SCRIPT PARA OBTENÇÃO DOS VALORES MÉDIOS DE POTENCIA E UTILIZAÇÃO DOS DISPOSITIVOS

```

% vetor de valores de potencia
dados = load('aquisicaoGDE_8.lvm');

fa = 1000; %frequencia de amostragem
tempo = 1/fa*(1:size(dados(:,1),1));

tensao = dados(:,1);
corrente = dados(:,2);
potencia = tensao.*corrente;

dadosLog = load('log.txt');
HD = dadosLog(:,3);
Memoria = dadosLog(:,4);
Carga = dadosLog(:,1);
Freq = dadosLog(:,2);
Down_ethernet = dadosLog(:,6);
Up_ethernet = dadosLog(:,7);
Down_wifi = dadosLog(:,8);
Up_wifi = dadosLog(:,9);

fprintf('\n%s\n', ['Potência média: ' num2str(mean(potencia), '%11.4f') '
W'])
fprintf('\n%s\n', ['Tensão média: ' num2str(mean(tensao), '%11.4f') '
V'])
fprintf('\n%s\n', ['Corrente média: ' num2str(mean(corrente), '%11.4f') '
A'])
fprintf('\n%s\n', ['HD médio: ' num2str(mean(HD), '%11.4f') '
Bytes/s'])
fprintf('\n%s\n', ['Memória média: ' num2str(mean(Memoria), '%11.4f') '
MB'])
fprintf('\n%s\n', ['Carga média: ' num2str(mean(Carga), '%11.4f') '
%'])
fprintf('\n%s\n', ['Frequência média: ' num2str(mean(Freq), '%11.4f') '
Hz'])
fprintf('\n%s\n', ['Download ethernet: '
num2str(mean(Down_ethernet), '%11.4f') ' Bytes/s'])
fprintf('\n%s\n', ['Upload ethernet: '
num2str(mean(Up_ethernet), '%11.4f') ' Bytes/s'])
fprintf('\n%s\n', ['Download wifi: ' num2str(mean(Down_wifi), '%11.4f')
' Bytes/s'])
fprintf('\n%s\n', ['Upload wifi: ' num2str(mean(Up_wifi), '%11.4f') '
Bytes/s'])

```

ANEXO C.2 - SCRIPT PARA OBTENÇÃO DO MODELO DE POTÊNCIA

```

%Modelo de Potência para o Notebook Positivo

Clear all;
Clc;
Z = load('ComTodos1.txt');

%wifi_on, Eth_on, DVD_on, Webcam_on, Potência, Hd(bytes/seg), Memória (MB), Br
ilho_LCD, Carga_Cpu, Freq_CPU, Download_ether, Upload_ether, Download_wifi,
Upload_wifi

C= [ones(size(Z,1),1) Z(:,1:4) Z(:,6:8) Z(:,9).*(Z(:,10).^3)
Z(:,11:14) ];
d = Z(:,5);

A = [];
b = [];
Aeq = [];
beq = [];
lb= 0*ones(size(C,2),1); %limite inferior
ub= Inf(size(C,2),1); %limite superior

ub(1) = inf;

options =
optimset('LargeScale','on','Diagnostics','on','MaxIter',400,'TolFun',
eps)

%lsqlin()-> função para aplicação do algoritmo dos mínimos quadrados
%com restrições
[x,resnorm,options,exitflag]=lsqlin(C,d,A,b,Aeq,beq,lb,ub,[], options)

tol=(resnorm^0.5)/size(C,1)

%wifi_on, Eth_on, DVD_on, Webcam_on, Potência, Hd(bytes/seg), Memória (MB), Br
ilho_LCD, Carga_Cpu, Freq_CPU, Download_ether, Upload_ether, Download_wifi,
Upload_wifi

B=x(1,1)
Wifi_on=x(2,1)
Ether_on=x(3,1)
WebCam_on=x(4,1)
DVD_on=x(5,1)
Hd=x(6,1)
Memoria=x(7,1)
LCD=x(8,1)
CPU=x(9,1)
Down_ether=x(10,1)
Up_ether=x(11,1)
Down_wifi=x(12,1)
Up_wifi=x(13,1)

```



```

fprintf('\n%s\n', ['R²: ' num2str((resnorm), '%11.11f')])

%Potencia = LCD*Z(1:103,1)+
Ether_on*Z(1:103,2)+Wifi_on*Z(1:103,3)+Hd*Z(1:103,4) +
Memoria*Z(1:103,5) + CPU*Z(1:103,6).*(Z(1:103,7).^3)
+B+Down_ether*Z(1:103,8)+Up_ether*Z(1:103,9)+Down_wifi*Z(1:103,10)+Up_
wifi*Z(1:103,11)
Potencia = C*x;
figure(1)
plot(d, 'o')
hold on
plot(Potencia, '--');%axis([0,120,0,20])
hold off
legend('Medições', 'Modelo')

dispositivos = {'B' 'Wifi on' 'Ether on' 'Hd' 'Memoria' 'LCD' 'CPU'
'Down eth' 'Up eth' 'Down wifi' 'Up wifi'}
figure(2)
for j = 1:6
    componentePotencia = C(:,j)*x(j,1);
    subplot(6,1,j)
    plot(componentePotencia, '--');
    ylabel(dispositivos(j))
end
figure(3)
for j = 7:11
    componentePotencia = C(:,j)*x(j,1);
    subplot(6,1,j-6)
    plot(componentePotencia, '--');
    ylabel(dispositivos(j))
end
subplot(6,1,6)
plot(d, 'o')
hold on
plot(Potencia, '--')
hold off
legend('Medições', 'Modelo')

```

ANEXO C.3 - SCRIPT PARA VALIDAÇÃO DO MODELO DE

POTÊNCIA

```

close all;
clear all;
clc;

% vetor de valores de potencia
dados = load('aquisicaoGDE_2.lvm');
fa = 1000; %frequencia de amostragem
tempo = 1/fa*(1:size(dados(:,1),1));

tensao = dados(:,1);
corrente = dados(:,2);

```

```

potencia = tensao.*corrente;

fprintf('\n%s\n', ['Potência média: ' num2str(mean(potencia), '%11.4f')
'W'])
fprintf('\n%s\n', ['Tensão média: ' num2str(mean(tensao), '%11.4f')
'V'])
fprintf('\n%s\n', ['Corrente média: ' num2str(mean(corrente), '%11.4f')
'A'])
dadosLog1 = load('ParaTeste2.txt');
dadosLog = load('teste_2.txt');
tempoLog = (1:size(dadosLog(:,1),1));

Estados_disp= [ones(size(dadosLog,1),1) dadosLog1(:,1:4)
dadosLog(:,1).*(dadosLog(:,2).^3) dadosLog(:,3:9) ];

Const=[ 14.6926 %B
0.2979 %Wifi_on
0.3471 %Ether_on
0.7743 %WebCam_on
2.0858 %DVD_on
1.7581 %CPU
6.3237e-007 %Hd
3.3134e-014 %Memoria
1.6831 %LCD
6.4120e-014 %Down_ether
3.7562e-007 %Up_ether
8.5077e-007 %Down_wifi
1.7054e-006]; %Up_wifi

fprintf('\n%s\n', ['Constante B = ' num2str(Const(1,1))])

Pot = Estados_disp*Const;

%*****Potencia média estimada em cada intervalo*****%
fprintf('\n%s\n', ['Potência esitmada 1min: '
num2str(mean(Pot(1:30,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 2min: '
num2str(mean(Pot(31:60,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 3min: '
num2str(mean(Pot(61:90,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 4min: '
num2str(mean(Pot(91:120,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 5min: '
num2str(mean(Pot(121:150,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 6min: '
num2str(mean(Pot(151:180,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 7min: '
num2str(mean(Pot(181:210,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 8min: '
num2str(mean(Pot(211:240,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 9min: '
num2str(mean(Pot(241:270,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência esitmada 10min: '
num2str(mean(Pot(271:300,1)), '%11.4f') 'W'])

%*****Potencia média medida em cada intervalo*****%
fprintf('\n%s\n', ['Potência media 1min: '
num2str(mean(potencia(1:6071,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 2min: '
num2str(mean(potencia(6072:12143,1)), '%11.4f') 'W'])

```

```

fprintf('\n%s\n', ['Potência media 3min: '
num2str(mean(potencia(12144:18215,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 4min: '
num2str(mean(potencia(18216:24287,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 5min: '
num2str(mean(potencia(24288:30359,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 6min: '
num2str(mean(potencia(30340:36411,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 7min: '
num2str(mean(potencia(36412:42483,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 8min: '
num2str(mean(potencia(42484:48555,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 9min: '
num2str(mean(potencia(48556:54627,1)), '%11.4f') 'W'])
fprintf('\n%s\n', ['Potência media 10min: '
num2str(mean(potencia(54628:60710,1)), '%11.4f') 'W'])

figure(1)
hold on
stairs((1/60)*tempo*10, potencia), xlabel('Tempo (min)'), ylabel('Potência
(W)')
stairs((1/60)*tempoLog*2, Pot, 'k', 'LineWidth', 2
), axis([0, 10, 0, 60]), grid
hold off

t = linspace(0, 10, 300);
y1 = mean(potencia(1:6067, 1)) * ones(30, 1);
y2 = mean(potencia(6068:12135, 1)) * ones(30, 1);
y3 = mean(potencia(12136:18203, 1)) * ones(30, 1);
y4 = mean(potencia(18204:24271, 1)) * ones(30, 1);
y5 = mean(potencia(24272:30339, 1)) * ones(30, 1);
y6 = mean(potencia(30340:36407, 1)) * ones(30, 1);
y7 = mean(potencia(36408:42475, 1)) * ones(30, 1);
y8 = mean(potencia(42475:48542, 1)) * ones(30, 1);
y9 = mean(potencia(48543:54610, 1)) * ones(30, 1);
y10 = mean(potencia(54611:60677, 1)) * ones(30, 1);
y_total = [y1; y2; y3; y4; y5; y6; y7; y8; y9; y10];

y1^ = mean(Pot(1:30, 1)) * ones(30, 1);
y2^ = mean(Pot(31:60, 1)) * ones(30, 1);
y3^ = mean(Pot(61:90, 1)) * ones(30, 1);
y4^ = mean(Pot(91:120, 1)) * ones(30, 1);
y5^ = mean(Pot(121:150, 1)) * ones(30, 1);
y6^ = mean(Pot(151:180, 1)) * ones(30, 1);
y7^ = mean(Pot(181:210, 1)) * ones(30, 1);
y8^ = mean(Pot(211:240, 1)) * ones(30, 1);
y9^ = mean(Pot(241:270, 1)) * ones(30, 1);
y10^ = mean(Pot(271:300, 1)) * ones(30, 1);
y^_total = [y1^; y2^; y3^; y4^; y5^; y6^; y7^; y8^; y9^; y10^];

figure(2)
hold on
plot(t, y_total, 'b-*'), title('Gráfico comparativo entre Potência
medida e Potência estimada')
plot(t, y^_total, 'r-*'), xlabel('Tempo (min)'), ylabel('Potência média
(W)')
hold off, grid, legend('Potência_m_é_d_i_a
(medida)', 'Potência_m_é_d_i_a(estimada)')

```

BIBLIOGRAFIA

- [1] Miettinen, A. P.; Nurminen, J. K. *Energy efficiency of mobile clients in cloud computing*, Nokia Research Center, 2011.
- [2] Pinto, T. T. S. Evolução das memórias, baterias e processadores para dispositivos móveis, 2012. Disponível em: http://grenoble.ime.usp.br/~gold/cursos/2012/movel/mono-1st/2006-2_TassoTirapani.pdf. Acesso em 03 de Março de 2013.
- [3] MIAN, D.; ZHONG, L. *Self-Constructive High-Rate System Energy Modeling for Battery-Powered Mobile Systems*. **Proc. MobiSys**, 2011. 335-348.
- [4] Inácio, J. F. S. Implementação de algoritmo de mínimos quadrados para estimar eficiência de motores de indução. 2010.
- [5] Oliveira, P. J. R. de. Estimção de parâmetros de um motor de indução utilizando um modelo contínuo no tempo. 1998.
- [6] LJUNG, L. *System Identification: Theory for the User*. 2. ed. Englewoods Cliffs, New Jersey 07632: Prentice Hall PTR, 1998. Hardcover. ISBN 0138816409.
- [7] T. Coleman, Y. Li, “A reflective neston method for minimizing a quadratic function subject to bounds on some of the variables,” *SIAM Journal on Optimization*, vol 6, no. 4, pp 1040-1058, 1996
- [8] Positivo *Premium Select 8010*. Disponível em: <http://www.positivoinformatica.com.br/www/pessoal/imprensa/positivo-informatica-lanca-linha-de-notebooks-personalizaveis>. Acesso em 01 de Março de 2013.
- [9] *National Instruments*. Disponível em: <http://www.ni.com/data-acquisition/pt/>. Acesso em 03 de Março de 2013.
- [10] Atributos da classe Win32_Processor. Disponível em: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373(v=vs.85).aspx). Acesso em 04 de Março de 2013.
- [11] The Four Horsemen Discussion – Dawkins, Dennett, Harris, Hitchens (1 of 2). Disponível em: <https://www.youtube.com/watch?v=TfNgCaqLIR0>. Acesso em 02 de Março de 2013.