

# Estimador Embarcado para Análise de Redes em Cidades Inteligentes

Renata Helena Panseri  
PUC-Campinas  
renata.panseri@hotmail.com

Lia Toledo Moreira Mota Mota  
PUC-Campinas  
lia.mota@puc-campinas.edu.br

Argemiro Bevilacqua  
PUC-Campinas  
bevilacqua.argemiro@gmail.com

Lorenzo Campos Coiado  
UNICAMP  
lorenzo.coiado@gmail.com

Marcus Fabius Henriques de Carvalho  
PUC-Campinas  
marcius@puc-campinas.edu.br

Luiza Higinio Santos  
PUC-Campinas  
luizahiginoss@gmail.com

**Resumo** — Este trabalho propõe o desenvolvimento de um algoritmo de estimação para análise de dados coletados em redes de diferentes infraestruturas urbanas integradas à Rede de Sensores Sem Fio (RSSF). Será utilizada a plataforma Proxy-IP para interligar os sensores e a Internet, transmitindo os dados por meio da rede Ethernet ou da rede WLAN. Para a análise dos dados coletados pela RSSF, será utilizado o Método dos Mínimos Quadrados (MMQ).

**Palavras Chaves** — Estimadores, Rede de Sensores Sem Fio, Proxy-IP, Método dos Mínimos Quadrados, Sistemas de Infraestrutura Urbana.

## I. INTRODUÇÃO

A tecnologia da “Internet das Coisas” (*Internet of Things – IoT*) é uma revolução tecnológica que permite a conexão de objetos à Internet, através de sensores integrados, que coletam informações e agem sobre esses dados por meio da rede. Diversas são as aplicações do *IoT*, podendo ser usado na área de saúde, de segurança, na indústria e também nas “Cidades Inteligentes” (*Smart Cities*)[1]. É um recurso que pode viabilizar as “*Smart Cities*”, permitindo o sensoriamento e o monitoramento das cidades.

Os dados que são coletados pelo *IoT* são analisados a fim de se identificar e examinar padrões e de se prever eventos futuros. Dessa maneira, pode-se transformar dados básicos em informação. [1]

A tomada de decisão é um fator chave para que a infraestrutura de uma cidade funcione regularmente. A rede de sensores conectados as cidades inteligentes procura solucionar problemas através da análise de dados por meio de algoritmos, que possibilitam a tomada de decisões [2]. As “*Smart Cities*” são monitoradas a partir de sistemas de gerenciamento integrado com aplicações diversas, o que permite aprimorar o desempenho das atividades realizadas no

cenário urbano e, conseqüentemente, melhorar o conforto da população [3]. Sendo assim, é importante o estudo do monitoramento urbano utilizando estimadores embarcados.

Uma arquitetura de Rede de Sensores Sem Fio (RSSF) se faz presente nas *Smart Cities* e visa garantir a eficiência e a precisão operacional do sistema, o qual pode ser representado por aplicação, *software* embarcado e infraestrutura. Com base em procedimentos matemáticos e computacionais, pode-se obter informações úteis de dados crus coletados por sensores, por meio do tratamento desses dados.

Este trabalho propõem o desenvolvimento de um algoritmo de estimação direcionado à análise de redes que pode ser aplicado a diferentes sistemas de infraestrutura urbana. Este algoritmo baseia-se no Método dos Mínimos Quadrados (MMQ). Neste estudo foi desenvolvido um estimador embarcado utilizando o MMQ em um Proxy-IP (plataforma aberta e de baixo custo), o qual permite a associação de uma RSSF à Internet.

## II. SENSORES E REDES DE SENSORES SEM FIO

Os sensores são responsáveis por transformar uma grandeza em grandeza elétrica, tendo a capacidade de ler uma condição ou característica específica do ambiente [4].

A estrutura de uma RSSF constitui-se de um conjunto de nós sensores, um nó sorvedouro e um computador central. Os nós sensores fazem a leitura dos dados, medindo o fenômeno a ser analisado, processando e formatando as informações, que serão transmitidas para o nó sorvedouro, responsável por enviar os dados para o computador central. Os nós sensores se interconectam através de uma interface sem fio e são geralmente alimentados por pilhas ou baterias [5]. A Figura 1 ilustra a arquitetura de uma RSSF.



Figura 1 - Arquitetura de uma RSSF. Fonte: Adaptado de [21]

### III. MÉTODO DOS MÍNIMOS QUADRADOS: ORIGEM, CONCEITO E FORMULAÇÃO

O Método dos Mínimos Quadrados foi elaborado por Karl Friedrich Gauss no final do século XVIII, a fim de solucionar o problema da estimação de parâmetros que caracterizam o deslocamento de cometas e de planetas a partir de medidas realizadas previamente. [6]

Na formulação desenvolvida por Gauss, tem-se uma quantidade mínima de medidas para que seja possível a determinação das grandezas desconhecidas, sendo preciso haver mais medidas do que a quantidade mínima definida, devido aos erros apresentados por essas medições. Para que se obtenha dados coerentes, por conseguinte, é necessário haver precisão nas equações que representam o comportamento, ou dinâmica, do sistema.

Os parâmetros estimados têm de ser adequados ao tipo das medidas, de modo que os ruídos – diferença entre as medidas reais e as medidas calculadas a partir dos parâmetros – sejam os menores possíveis. O escopo desse método é, portanto, processar os valores das variáveis independentes e dependentes e, então, definir um vetor final de parâmetros estimados [6].

Para o desenvolvimento do método, Friedrich Gauss estabeleceu um sistema de equações em forma de matriz, como se vê na Figura 2 seguinte.

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & h_{m3} & \cdots & h_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_m \end{bmatrix}$$

Figura 2 - Sistema de equações em forma de matriz [2]

Portanto, tem-se que:

$$Hx = z \tag{1}$$

Em que “H” é a matriz de parâmetros do modelo, o vetor “x” é o vetor coluna de valores de entrada do sistema e o

vetor “z” representa o vetor coluna de valores de saída do sistema.

Por conseguinte, deve-se considerar o problema de solução para um valor de  $\hat{x}$  em que o erro de medição estimado  $H\hat{x} - z$  seja minimizado. Pode-se determinar o erro de estimação pelo seu quadrado:

$$\varepsilon(\hat{x})^2 = |H\hat{x} - z|^2 = \sum_{i=1}^m [\sum_{j=1}^n h_{ij}\hat{x}_j - z_i]^2 \tag{2}$$

A função chegará ao seu mínimo valor quando todas as suas derivadas em relação à  $\hat{x}_k$  forem igual à zero:

$$0 = (d\varepsilon^2)/(d\hat{x}_k) = 2 \sum_{i=1}^m h_{ik} [\sum_{j=1}^n h_{ij}\hat{x}_j - z_i] \tag{3}$$

Sendo que,

$$\sum_{j=1}^n h_{ij}\hat{x}_j - z_i = (H\hat{x} - z)_i \tag{4}$$

A i-ésima fileira de  $H\hat{x} - z$  e a soma mais externa é igual ao ponto do produto da k-gésima coluna de H com  $H\hat{x} - z$ . Dessa forma, a equação (3) pode ser descrita do seguinte modo:

$$0 = 2H^T[H\hat{x} - z] = 2H^TH\hat{x} - 2H^Tz \tag{5}$$

Ou,

$$H^TH\hat{x} = H^Tz \tag{6}$$

Visto que a matriz transposta  $H^T$  é definida como na Figura 3.

$$H^T = \begin{bmatrix} h_{11} & h_{21} & h_{31} & \cdots & h_{m1} \\ h_{12} & h_{22} & h_{32} & \cdots & h_{m2} \\ h_{13} & h_{23} & h_{33} & \cdots & h_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{1n} & h_{2n} & h_{3n} & \cdots & h_{mn} \end{bmatrix}$$

Figura 3 – Ilustração da definição da matriz transposta de H [12]

Portanto,

$$\hat{x} = (H^TH)^{-1}(H^Tz) \tag{7}$$

A equação (7) apresentada é denominada equação normal da problemática linear dos mínimos quadrados.

Este projeto aborda quatro modelos matemáticos do MMQ:

$$1) Y = K * X \tag{8}$$

$$2) Y = K * X + M \tag{9}$$

$$3) Y = K * X_1 + L * X_2 \tag{10}$$

$$4) Y = K * X_1 + L * X_2 + M \tag{11}$$

Logo, para as equações (8) e (9), “X” representa os valores de entrada e, para as equações (10) e (11), “X<sub>1</sub>” e “X<sub>2</sub>” são os representantes dos valores de entrada. “Y” representa os valores de saída, em que “K”, “M” e “L” são os parâmetros a serem estimados.

Para determinar os parâmetros, utiliza-se a equação (7). Por uma simples questão de conveniência, trocar-se-á “H” por “X”, “z” por “Y” e “x” por “b”, obtendo-se:

$$b = (X^T * X)^{-1} * X^T * Y \quad (12)$$

No modelo 1, “b” resulta em um número escalar, sendo, portanto, o próprio parâmetro “K”, ou seja,  $b=K$ , como se pode verificar na equação (13):

$$K = (X^T * X)^{-1} * X^T * Y \quad (13)$$

No modelo 2, porém, “b” tem como resultado um vetor  $2 \times 1$ , isto é, um vetor de duas linhas e uma coluna. Sendo assim, o valor da primeira linha é correspondente à “K” e o da segunda, à “M”. Já no modelo 3, “b” novamente resulta num vetor  $2 \times 1$ , sendo que o valor da primeira linha corresponde à “K” e o da segunda, à “L”. Por último, no modelo 4, “b” tem como resultado um vetor  $3 \times 1$ , sendo o valor da primeira linha referente à “K”, o da segunda referente à “L” e o da terceira, à “M”.

Para que se possa obter o valor de saída estimado “ $Y_{estimado}$ ”, multiplica-se o valor de entrada “X” pelo “b” previamente calculado. Deste modo:

$$Y_{estimado} = X * b \quad (14)$$

“ $Y_{estimado}$ ” tem como resultado um vetor. Ulteriormente, para se obter o valor do Resíduo Simples (RS), é realizado o seguinte cálculo:

$$RS = Y_{estimado} - Y \quad (15)$$

Para encontrar o modelo do MMQ mais adequado em uma determinada aplicação, deve-se avaliar qual o modelo com o menor resíduo. Porém, como o RS é um vetor, torna-se difícil a comparação entre os valores de resíduo obtidos em cada modelo. Assim, é conveniente transformar o vetor em um número escalar e, para isso, calcula-se dois outros tipos de resíduo: o Resíduo Quadrático Médio (RQM) e o Resíduo Absoluto Médio (RAM).

Para encontrar o RQM, eleva-se ao quadrado cada número do vetor do RS anteriormente calculado, de modo a ter-se apenas números positivos, resultando em um novo vetor, o qual pode ser denominado  $RS^2$ . Após, é calculada a média aritmética do vetor  $RS^2$ , isto é, soma-se os valores de todos os seus elementos e o resultado da soma é dividido pelo número de elementos do vetor. Desse modo, consegue-se encontrar um valor médio do vetor RS.

Já para calcular o valor do RAM, é preciso colocar em módulo todos os elementos de RS, para que se tenha apenas números positivos, resultando em um novo vetor, que pode ser denominado  $|RS|$ . Após, é calculada a média aritmética de  $|RS|$ . Tanto o RQM quanto o RAM são formas consistentes de se obter o valor médio do vetor RS, de modo que seja possível comparar os valores dos resíduos de cada modelo do MMQ e verificar qual o modelo mais

adequado.

#### IV. SENSORES E REDES DE SENSORES SEM FIO

O programa *Scilab* é um *software* livre, de código aberto e de linguagem de programação de alto nível que permite o desenvolvimento de algoritmos de alta complexidade com poucas linhas de código. [7]

Esta plataforma possibilita diversas aplicações industriais e educacionais, como em processamento de sinais, análise estatística e otimização numérica. O programa possui o editor gráfico Xcos que possibilita o desenvolvimento de sistemas dinâmicos. [7] Pela facilidade de uso, o Scilab foi utilizado para a montagem e a resolução de matrizes a fim de estimar os parâmetros do Método dos Mínimos Quadrados (MMQ).

#### V. PROXY-IP

O Proxy-IP faz a interconexão entre sensores e a rede TCP/IP. Ele utiliza o protocolo SNMP e as seguintes interfaces de comunicação: rede Ethernet, rede Wi-Fi e transmissão Serial. [8]

A interface de gerenciamento do Proxy IP é a Web LuCI, escrita em linguagem Lua, e seu painel de monitoramento é o *Dash Board*, desenvolvido em linguagem PHP, que permite a visualização das informações de forma gráfica [8].

Os sistemas Proxy-IP possuem Sistema Operacional (SO) Linux embarcado *OpenWrt* [9], o qual é gratuito e *Open Source* [10]. O *OpenWrt* apresenta já embutida a linguagem Lua,

A partir dos dados coletados pelo Proxy-IP, é possível obter decisões e estimações por meio do Método dos Mínimos Quadrados e, transmiti-los de modo inteligente para o sistema supervisorio na Internet.

Neste trabalho, utilizou-se apenas um AP (*Acess Point*) e um *laptop*, pois o foco é a análise de dados. Abaixo estão ilustrados os equipamentos utilizados na Figura 5 a seguir.



Figura 5 - Foto dos equipamentos utilizados no trabalho: *laptop* à esquerda e AP à direita

O *laptop*, visto à esquerda na Figura 5, é um MAC Os X e o AP, visto à direita, é um TP-Link modelo TL-WR740N. Os principais recursos do AP apresentado, de acordo com o site da TP-Link, são:

- Taxa de transferência *wireless* de 150Mbps;

- É compatível com os produtos da IEEE 802.11n/b/g;
- *Firewall* SPI/NAT;
- Controle de largura de banda IP;
- Fácil configuração de criptografia WPA;
- Sistema *wireless* WDS. [11]

## VI. LINGUAGEM LUA

A linguagem Lua é leve, pequena, extensível e de *script* multiparadigma, sendo utilizada no desenvolvimento de editores de imagem, jogos, controle de robôs, entre outros. [12] Como é um software livre e código aberto, é possível o compartilhamento do código-fonte com o mundo [13.] Além disso, existem manuais de referência e vários livros acerca da linguagem, facilitando o estudo desse *software*.

Lua é capaz de rodar em sistemas Windows e Unix, em dispositivos móveis (IOs, Android), entre outros. O *software* utiliza compilador C padrão e possibilita estender programas desenvolvidos em outras linguagens, como Java e C [14].

## VII. LINUX EMBARCADO

Existem tamanhos de sistemas embarcados controlados por Linux: “*extra large*”, grande, médio e reduzido (ou pequeno). As características referentes aos sistemas embarcados reduzidos são: utilização de microprocessador de propósito geral, não possuem disco rígido, consumo baixo de energia, interface homem-máquina simples, pouca memória e aplicações já embutidas.

Em geral, estes sistemas são autônomos e projetados para um objetivo específico. São apresentados prontos para uso, com *hardware* e *software* previamente instalados [15].

A plataforma Linux possui amplos recursos de programação e não requer licença para uso. O *OpenWRT* é um Linux embarcado e possibilita personalizar o sistema embarcado para diversas finalidades por meio do gerenciamento de pacotes,—Este sistema é livre, de código aberto e fácil configuração [15].

O *OpenWRT* foi escolhido para este trabalho por possuir “suporte nativo para linguagens de programação Bourne shell, Lua e C; suporte nativo para controle de tráfego incluindo gerência de banda; e disponibilidade do comando “iw”, poderoso utilitário de linha de comando capaz de coletar dados sobre conexões sem fio; e é configurável, possui pacotes de *software* opcionais” [9]

## VIII. RESULTADOS E ANÁLISE

### A. O código em Linguagem Lua

O código em Lua tem como característica a menor quantidade de comandos, o que traz agilidade na utilização do *software*. O programa pode ser dividido em três grandes partes: início, processamento e apresentação de valores. No início, há uma exposição sobre o Método dos mínimos Quadrados (MMQ) e sobre o formato do arquivo de entrada, sendo solicitado que o usuário digite o nome do arquivo. Em

seguida os dados são processados e os cálculos realizados. Por fim, são apresentados os valores dos resíduos e dos parâmetros estimados.

Os quatro modelos estimados pelo MMQ abordados neste trabalho são:

$$1) Y = K \cdot X \quad (8)$$

$$2) Y = K \cdot X + M \quad (9)$$

$$3) Y = K \cdot X_1 + L \cdot X_2 \quad (10)$$

$$4) Y = K \cdot X_1 + L \cdot X_2 + M \quad (11)$$

Ao inserir o nome do arquivo de entrada, é feita a conferência do formato do arquivo (“.txt”) e a separação dos valores deve ser realizada por meio de um espaço em branco. As duas situações possíveis são:

- Para os modelos 1 e 2, equações (8) e (9), a primeira e segunda linha do arquivo devem conter respectivamente os valores referentes a “X” e “Y”.
- Já para os modelos 3 e 4, equações (10) e (11), as três linhas do arquivo devem conter respectivamente os valores referentes a “X<sub>1</sub>”, “X<sub>2</sub>” e “Y”.

Se o usuário inserir um arquivo com o formato correto, o programa verificará a quantidade de linhas e considerará os modelos correspondentes, duas linhas – modelos 1 ou 2, três linhas – modelos 3 ou 4, evitando cálculos desnecessários. Em seguida os parâmetros “Y<sub>estimado</sub>” e o Resíduo Simples são calculados. O cálculo do resíduo é o determinante do modelo aplicado: dentre os dois modelos selecionados pela quantidade de linhas, o que apresentar menor resíduo será o melhor aplicado àquela situação. Assim, são calculados o Resíduo Quadrático Médio (RQM) e o Resíduo Absoluto Médio (RAM). O fluxograma do algoritmo correspondente ao processo realizado pelo *software* é apresentado na Figura 6 a seguir para melhor compreensão.

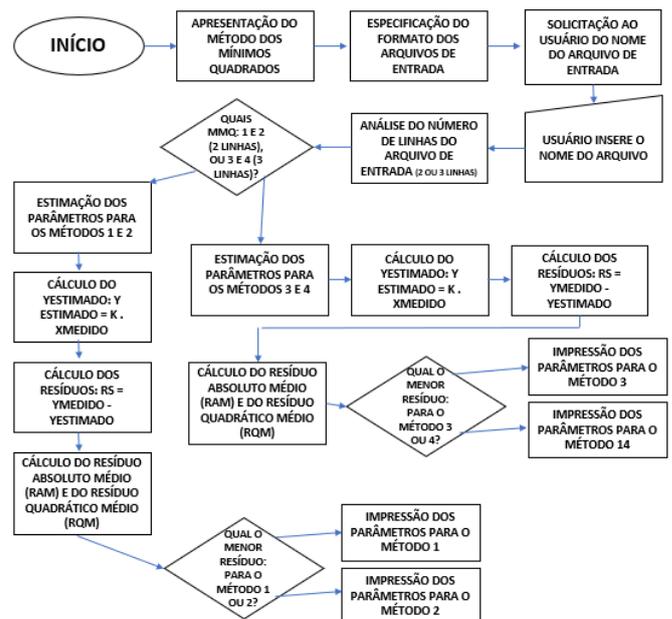


Figura 6 - Fluxograma referente ao procedimento realizado pelo *software*

A mensagem explicativa apresentada pelo programa

acerca dos cálculos do MMQ pode ser visualizada na Figura 7.

```
IDENTIFICACAO DE MODELOS PELO METODO DOS MINIMOS QUADRADOS

Modelo 1: Y=X*K, onde o objetivo e calcular K. K=(inv(X'*X))*X'*Y
0 residuo e: RS=Yestimado-Y, Yestimado=K*X

Modelo 2: Y=X*K+M, onde o objetivo e calcular K e M. K e a primeira
linha do vetor b e M e a segunda.
X=[X 1] adiciona uma coluna de uns ao vetor X.
b=(inv(X'*X))*X'*Y 0 residuo e: RS=Yestimado-Y, Yestimado=b*X

Modelo 3: Y=X1*K+X2*L, onde o objetivo e estimar K e L.
K e a primeira linha do vetor b, L a segunda.
X=[X1 X2] junta dois vetores X1 e X2
b=(inv(X'*X))*X'*Y. 0 residuo e: RS=Yestimado-Y, Yestimado=b*X

Modelo 4: Y=X1*K+X2*L+M, onde o objetivo e estimar K, L e M.
K e a primeira linha do vetor b, L a segunda e M a terceira.
Apos juntar X1 com X2, adiciona uma coluna de uns.
X=[X1 X2] X=[X 1]
b=(inv(X'*X))*X'*Y 0 residuo e: RS=Yestimado-Y, Yestimado=b*X

Teclre <ENTER> para continuar:
```

Figura 7 - Mensagem do programa apresentando os modelos do MMQ (foto tirada da tela do computador).

A Figura 8 exemplifica a segunda mensagem apresentada pelo programa, que diz respeito ao formato do arquivo que deve ser inserido pelo usuário, já que uma configuração específica do arquivo é requerida para realização dos cálculos.

```
FORMATO DOS ARQUIVOS DE ENTRADA

Para os modelos 1 e 2, os arquivos de entrada terao o seguinte formato:
- A primeira linha contera os dados de entrada X
- A segunda linha contera os dados de saida Y

Para os modelos 3 e 4, os arquivos de entrada terao o seguinte formato:
- A primeira linha contera os dados de entrada X1
- A segunda linha contera os dados de entrada X2
- A terceira linha contera os dados de saida Y

O formato do arquivo que for passado como entrada agora,
determinara quais os modelos serao testados: 1 e 2 ou 3 e 4

Digite o nome do arquivo de entrada: rssi.txt
```

Figura 8 - Mensagem do programa apresentando o formato dos arquivos de entrada (foto tirada da tela do computador).

### B. Aplicação do programa em Lua

Foi realizada a análise da relação entre: quantidade de pessoas em um ambiente fechado e o RSSI (“Received Signal Strength Indicator”), em português “Indicador de Intensidade do Sinal Recebido”) – o RSSI é referente à intensidade do sinal de rádio frequência que atinge um detector. A pesquisa sobre esta relação foi realizada previamente no Grupo de Pesquisa e, por conseguinte, foram utilizados os dados anteriormente coletados para a análise por meio do MMQ. Os valores medidos podem ser visualizados na Tabela 1.

Tabela 1 - Dados referentes à quantidade de pessoas, média e variância normalizados [16]

Grupo	Quantidade de Pessoas	Média	Variância
1	2	0.9545695	0
2	7	0.1754486	0.0331607
3	11	0.1720357	0.1062682
4	12	0.1861132	0.0155931
5	18	0.0730919	0.1717080
6	22	0.0509317	0.4419327
7	27	0.0	0.8563973
8	28	0.0835066	0.1106246
9	30	0.0359905	0.5867438
10	32	0.0	1.0

Neste caso, a saída Y é o número de pessoas, a entrada X<sub>1</sub> é a média do RSSI e a entrada X<sub>2</sub> é a variância do RSSI. Obteve-se os seguintes resultados da Figura 9.

```
IMPRESSAO DOS RESULTADOS

Modelo 3: Y = X1*K+X2*L, b = (inv(X'*X))*X'*Y, K = b[1], L = b[2]
K = 7.5544346327842
L = 37.362453830901
Residuo Absoluto Medio (RAM) = 6.6321816702406
Residuo Quadratico Medio (RQM) = 49.145757090806

Modelo 4: Y = X1*K+X2*L+M, b = (inv(X'*X))*X'*Y,
K = b[1], L = b[2], M = b[3]
K = -11.881586590248
L = 20.249994674412 M = 12.838221845645
Residuo Absoluto Medio (RAM) = 2.3643761120955
Residuo Quadratico Medio (RQM) = 8.4228426128567

Resposta Final: O Modelo 4 é o mais adequado neste caso, pois apresentou menor residuo.
```

Figura 9 - Resultados obtidos pelo programa para o Caso 1 (foto tirada da tela do computador).

Como é possível ver, o programa mostrou que o Modelo 4 do MMQ é o mais apropriado a aplicação em questão, pois é o que possui o menor resíduo. Assim, quando os valores de entrada e de saída forem estimados, ao se utilizar o Modelo 4 eles irão se aproximar mais da realidade do que se forem utilizados os Modelos 1, 2 ou 3. A comparação entre os valores de RQM e RAM mostra que o Modelo 3 apresentou um valor de RAM 2,8 vezes maior que o RAM do Modelo 4, e um valor de RQM 5,84 vezes maior em relação ao Modelo 4. Os parâmetros obtidos foram: K = -11,88; L = 20,25 e M = 12,84.

## IX. CONCLUSÃO E COMENTÁRIOS

A utilização da linguagem Lua possibilitou que o programa de aplicação do MMQ funcionasse em equipamentos de *hardware* de tamanho reduzido com sistema operacional Linux embarcado, de baixo custo e portáteis. Como o equipamento de *hardware* é o mesmo utilizado para coleta de dados dos sensores, é possível a realização de um processamento prévio de alto nível para a tomada de decisões fundamentais nos mesmos instante e local onde os dados estão sendo coletados. A partir deste mecanismo, apenas informações relevantes são enviadas para o sistema supervisor e não um grande volume de dados sem significado.

A confiabilidade do código desenvolvido pôde ser verificada a partir da comparação entre os resultados obtidos

pelo *software* em Lua e pelo *Scilab*. O programa elaborado foi capaz de estimar os parâmetros a partir do Método dos Mínimos Quadrados e de identificar o melhor modelo para cada caso. As facilidades de uso do programa foram satisfatórias e sua utilização em campo é possível justamente pelo tamanho reduzido do *hardware*.

Sendo assim é possível utilizar um Linux embarcado para a realização dos cálculos do MMQ, com finalidade de estimar o comportamento de uma determinada grandeza com base em dados amostrais. Este trabalho pode contribuir a tecnologia da “Internet das Coisas”, amplamente utilizada em “Cidades Inteligentes”, com aplicações diversas relacionadas à análise de redes em infraestruturas urbanas.

## AGRADECIMENTOS

Os autores agradecem à PUC-Campinas e ao CNPq pelo apoio ao desenvolvimento deste trabalho.

## REFERÊNCIAS

- [1] SAS. **Internet of Things: what it is and why it matters**. Disponível em <[https://www.sas.com/pt\\_br/insights/big-data/internet-das-coisas.html](https://www.sas.com/pt_br/insights/big-data/internet-das-coisas.html)>. Acesso em 06/07/2017.
- [2] FAPESP. **Chamada Cidades Inteligentes tem resultado da etapa de enquadramento**. Disponível em <<http://www.fapesp.br/10362>>. Acesso em 17/02/2017.
- [3] WEISS, Marco Cesar et al. **Cidades Inteligentes como nova prática para o gerenciamento dos serviços e infraestruturas urbanas: a experiência da cidade de Porto Alegre**. Disponível em <<http://www.scielo.br/pdf/urbe/v7n3/2175-3369-urbe-2175-3369007003AO01.pdf>>. Acesso em 17/02/2017.
- [4] CLASSROOM. **Tipos de sensores**. Disponível em <<http://classroom.orange.com/pt/tipos-de-sensores.html>>. Acesso em 8/11/2016.
- [5] LOUREIRO, Antônio A. F. et al. **Redes de Sensores Sem Fio**. Disponível em <<http://homepages.dcc.ufmg.br/~loureiro/cm/docs/sbrc03.pdf>>. Acesso em 8/11/2016.
- [6] MOREIRA, Lia Toledo. **Estimação Dinâmica de Parâmetros da Carga em Sistemas de Energia Elétrica**. Campinas, SP, 2001.
- [7] BRSCILAB. **Mas o que é o Scilab**. Disponível em <<http://brscilab.blogspot.com.br/2012/10/mas-o-que-e-o-scilab.html>>. Acesso em 17/02/2017.
- [8] MACHADO, L. F. **Proxy-IP de baixo custo e múltiplos sensores para cidades inteligentes**. 2015. Dissertação (MESTRADO PROFIS GESTÃO REDES TELECOMUNICAÇÕES) – Pontifícia Universidade Católica de Campinas.
- [9] OPENWRT. **OpenWrt**. Disponível em <<https://openwrt.org/>>. Acesso em 06/07/2017.
- [10] WIKI OPENWRT. **Table of Hardware [OpenWrt Wiki]**. Disponível em <<http://wiki.openwrt.org/toh/start>>. Acesso em 06/07/2017.
- [11] TP-LINK. **Roteador Wireless N 150Mbps TL-WR740N**. Disponível em <[http://www.tp-link.com.br/products/details/cat-9\\_TL-WR740N.html](http://www.tp-link.com.br/products/details/cat-9_TL-WR740N.html)>. Acesso em 19/07/2017.
- [12] HARDWARE. **Lua – conhecendo um pouco sobre a linguagem de programação 100% brasileira**. Disponível em <<http://www.hardware.com.br/comunidade/lua-linguagem/1304947/>>. Acesso em 22/03/2017.
- [13] OPENSUSE. **Software Livre e de Código Aberto**. Disponível em <[https://pt.opensuse.org/Software\\_Livre\\_e\\_de\\_Código\\_Aberto](https://pt.opensuse.org/Software_Livre_e_de_Código_Aberto)>. Acesso em 05/07/2017.
- [14] LUA. **A Linguagem de Programação Lua**. Disponível em <<https://www.lua.org/portugues.html>>. Acesso em 22/03/2017.
- [15] BEVILACQUA, Argemiro. **Deteção e mitigação de anomalia na camada MAC em redes IEEE 802.11**. 2015. 101 p. Dissertação (Mestrado em Engenharia Elétrica) - Pontifícia Universidade Católica de Campinas, Campinas, 2015.
- [16] OLIVEIRA, E. D. et al. **Monitoramento de temperatura do ar e de superfície em edificações usando redes sem fio padrão IEEE 802.15.4**. In: XIII Encontro Nacional e IX Encontro Latino-americano de Conforto no Ambiente Construído, 2015, Campinas (SP). Anais, 2015.