Self-Constructive High-Rate System Energy Modeling for Battery-Powered Mobile Systems

*Mian Dong* and *Lin Zhong*

*Rice University*
System Energy Model

\[ y(t) = f(x_1(t), x_2(t), \ldots, x_p(t)) \]

Response \( y(t) \):
Energy consumed by the system in \( t \)

Predictors \( x_i(t) \):
System status variables in \( t \)
Rate \( \frac{1}{t} \)

- 0.01Hz
- 1Hz
- 100Hz
A High-Rate Energy Model is needed to provide an energy reading at each OS scheduling interval 10ms.
Model Construction

\[ t = t_1 = t_2 = \ldots = t_n \]

\[
\begin{align*}
x_1(t_1) & \times x_2(t_1) \times \ldots \times x_p(t_1) & y(t_1) \\
x_1(t_2) & \times x_2(t_2) \times \ldots \times x_p(t_2) & y(t_2) \\
\vdots & \vdots & \vdots \\
x_1(t_n) & \times x_2(t_n) \times \ldots \times x_p(t_n) & y(t_n)
\end{align*}
\]
Model Construction

$t = t_1 = t_2 = \ldots = t_n$

Regression Model Construction

\[ X(t) = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \ldots & x_p(t_1) \\ x_1(t_2) & x_2(t_2) & \ldots & x_p(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_n) & x_2(t_n) & \ldots & x_p(t_n) \end{bmatrix} \]

\[ Y(t) = \begin{bmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_n) \end{bmatrix} \]
Model Construction

\[ t = t_1 = t_2 = \ldots = t_n \]

Linear Model:

\[ y(t) = \beta_0 + \beta_1 x_1(t) + \ldots + \beta_p x_p(t) \]

\[ \hat{\beta} = \arg\min_{\beta} (\| Y(t) - [1 \ X(t)]\beta \|_2) \]
Model Construction

\[ t = t_1 = t_2 = \ldots = t_n \]

Linear Model:

\[ \hat{y}(t) = \hat{\beta}_0 + \hat{\beta}_1 x_1(t) + \ldots + \hat{\beta}_p x_p(t) \]

\[ err(t_i) = \frac{\hat{y}(t_i) - y(t_i)}{y(t_i)} \]

Mean Absolute
Root-Mean-Square
What are the limitations?
External Devices for energy measurement
Deep Knowledge for predictor collection
Exclusive Model for a specific platform
Fixed Model for all instances of the same platform
Dependencies of system energy models on **Hardware & Usage** suggest “personalized” models be constructed for a mobile system.
Self-Constructive System Energy Modeling
Battery Interface + Statistical Learning \rightarrow Personalized Model
Battery Interface
State-of-the-art battery Interfaces are **Low-rate/Inaccurate**

<table>
<thead>
<tr>
<th></th>
<th>N85</th>
<th>T61</th>
<th>N900</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max Rate</strong></td>
<td>4Hz</td>
<td>0.5Hz</td>
<td>0.1Hz</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>67%</td>
<td>82%</td>
<td>58%</td>
</tr>
</tbody>
</table>

Accuracy = 100% – Root_Mean_Square(Instant_Relative_Error)
Errors in battery interface readings are Non-Gaussian.
Low-Rate/Inaccurate Battery Interface

Statistical Learning

High-Rate/Accurate System Energy Model
Averaged battery interface readings have Higher Accuracy but Even Lower Rate.
Linear models are Independent on Time

High-rate data points

Low-rate data points

\[ y(t) \]

\[ x(t) \]

\[ y(T) \]

\[ x(T) \]

\[ y \]

\[ x \]
1. Model Molding

\[ Y(t_{VL}) \leftarrow \cdots \rightarrow Y(t_L) \rightarrow \hat{Y}(t_H) \]

\[ X(t_{VL}) \leftarrow \cdots \rightarrow X(t_H) \]

\[ \hat{\beta} \rightarrow \hat{\beta} \]

0.01Hz \quad 1Hz \quad 100Hz
Model Molding improves rate

![Graph showing RMS of Relative Error vs Rate (Hz)]

- Triangle: Battery Interface
- Square: Molded Model

Rate (Hz)

0.01 0.1 1 10 100
2. Predictor Transformation

\[ x_1(t), x_2(t), \ldots, x_p(t) \]

\[ \downarrow \]

Principle Component Analysis

\[ z_1(t), z_2(t), \ldots, z_L(t) \quad L \leq p \]
PCA improves accuracy
3. Total-Least-Square

Training data points

\( y = f(x) \)
TLS improves accuracy at high rate
Implementation

N900

T61
Sesame is able to generate energy models with a rate up to **100Hz**

<table>
<thead>
<tr>
<th></th>
<th>T61</th>
<th>N900</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Hz</td>
<td>95%</td>
<td>86%</td>
</tr>
<tr>
<td>100Hz</td>
<td>88%</td>
<td>82%</td>
</tr>
</tbody>
</table>

Accuracy = 100% – Root_Mean_Square(Instant_Relative_Error)
Field Study

Day 1-5: Model Construction

Day 6: Model Evaluation
Models were generated within 15 hours
Sesame is able to construct models of high accuracy because of:

1. Sophisticated Statistical Methods
2. Capability to Adapt Models
Sesame is a high-rate/accurate Virtual Power Meter
and creates new opportunities in

Energy Optimization & Management
Software Optimization

\[ y(t) = \beta_0 + \beta_1 x_1(t) + \ldots + \beta_p x_p(t) \]

“Knob” provided by target software
Energy Accounting

\[ y(t) = \beta_0 + \beta_1 x_1(t) + \ldots + \beta_p x_p(t) \]

n Processes
Energy Accounting

\[ y(t) = \beta_0 + \beta_1 x_1(t) + \ldots + \beta_p x_p(t) \]

\[ x_1(t) = x_{1,1}(t) + \ldots + x_{1,n}(t) \]

\[ \vdots \quad \vdots \quad \vdots \]

\[ x_p(t) = x_{p,1}(t) + \ldots + x_{p,n}(t) \]
Energy Contribution by Process $j$

$$y_j(t) = \beta_1 x_{1,j}(t) + ... + \beta_p x_{p,j}(t)$$
Sesame can be also used for

Servers and Workstations
Conclusions

• Self-Modeling is necessary to adapt to the changes in hardware and usage

• Statistical methods help to construct high-rate/accurate models from low-rate/inaccurate battery interfaces

• Sesame creates new opportunities in system energy optimization and management