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

Mian Dong **and Lin Zhong**

Rice University

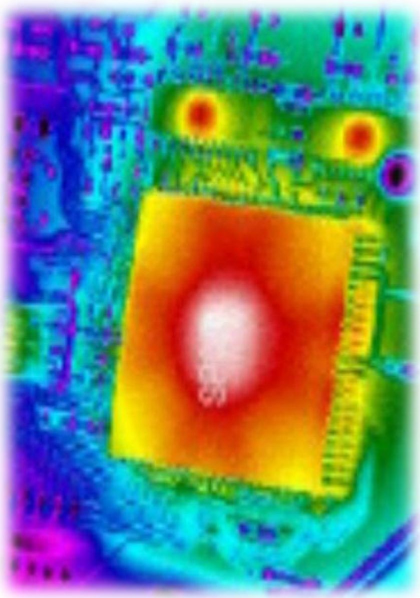
System Energy Model

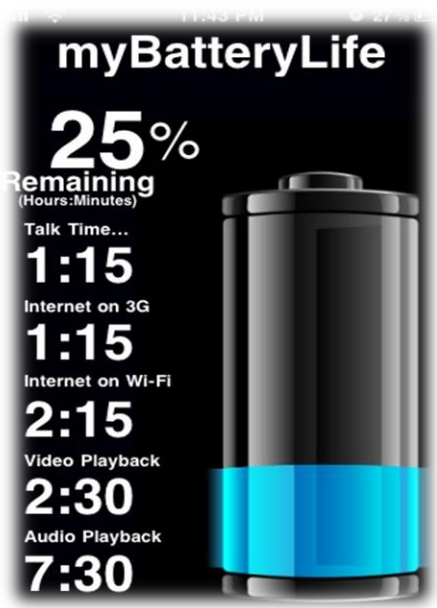
$$y(t) = f(x_1(t), x_2(t),\dots, x_p(t))$$

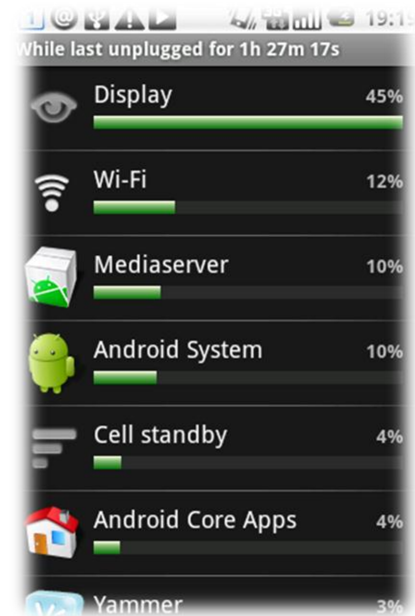
Response $y(t)$:
Energy consumed by the system in t

Predictors $x_i(t)$:
System status variables in t

Rate (1/*t*)



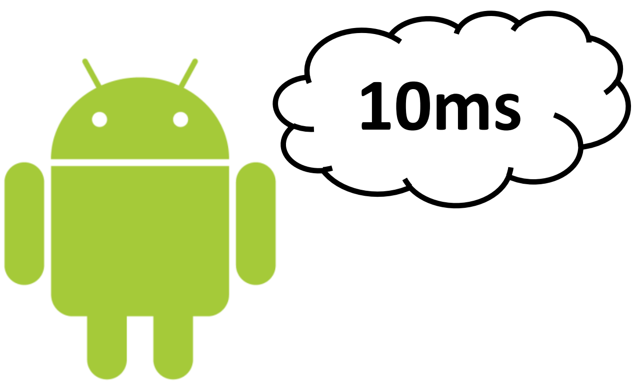




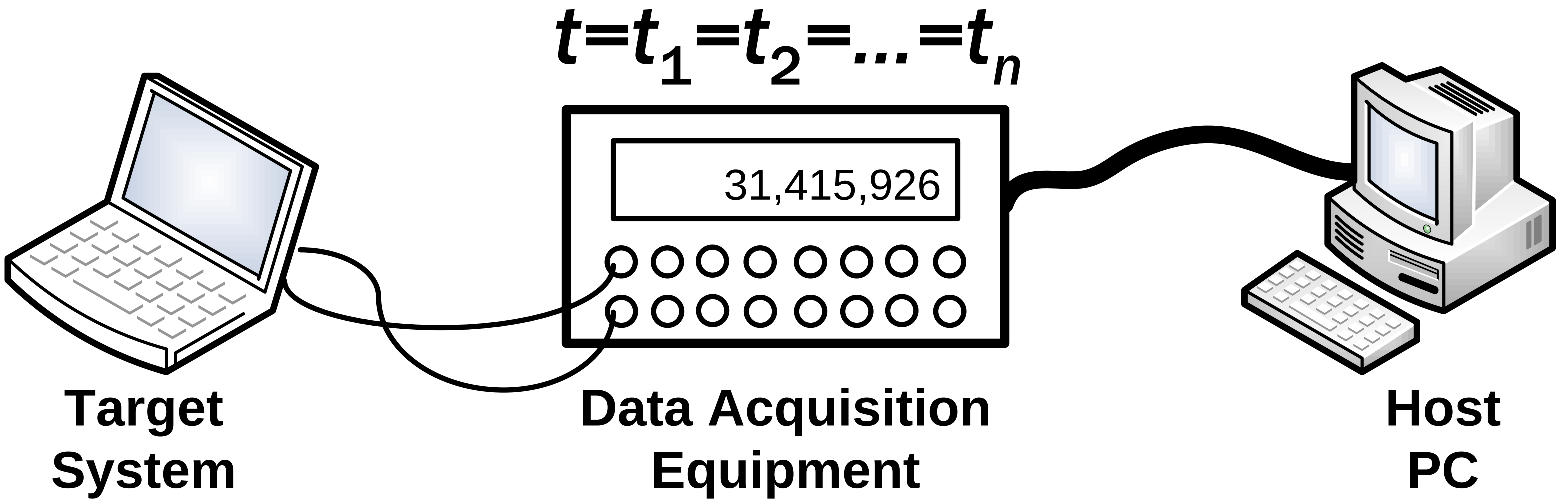
0.01Hz 1Hz 100Hz

A High-Rate Energy Model

is needed to provide an energy reading
at each OS scheduling interval

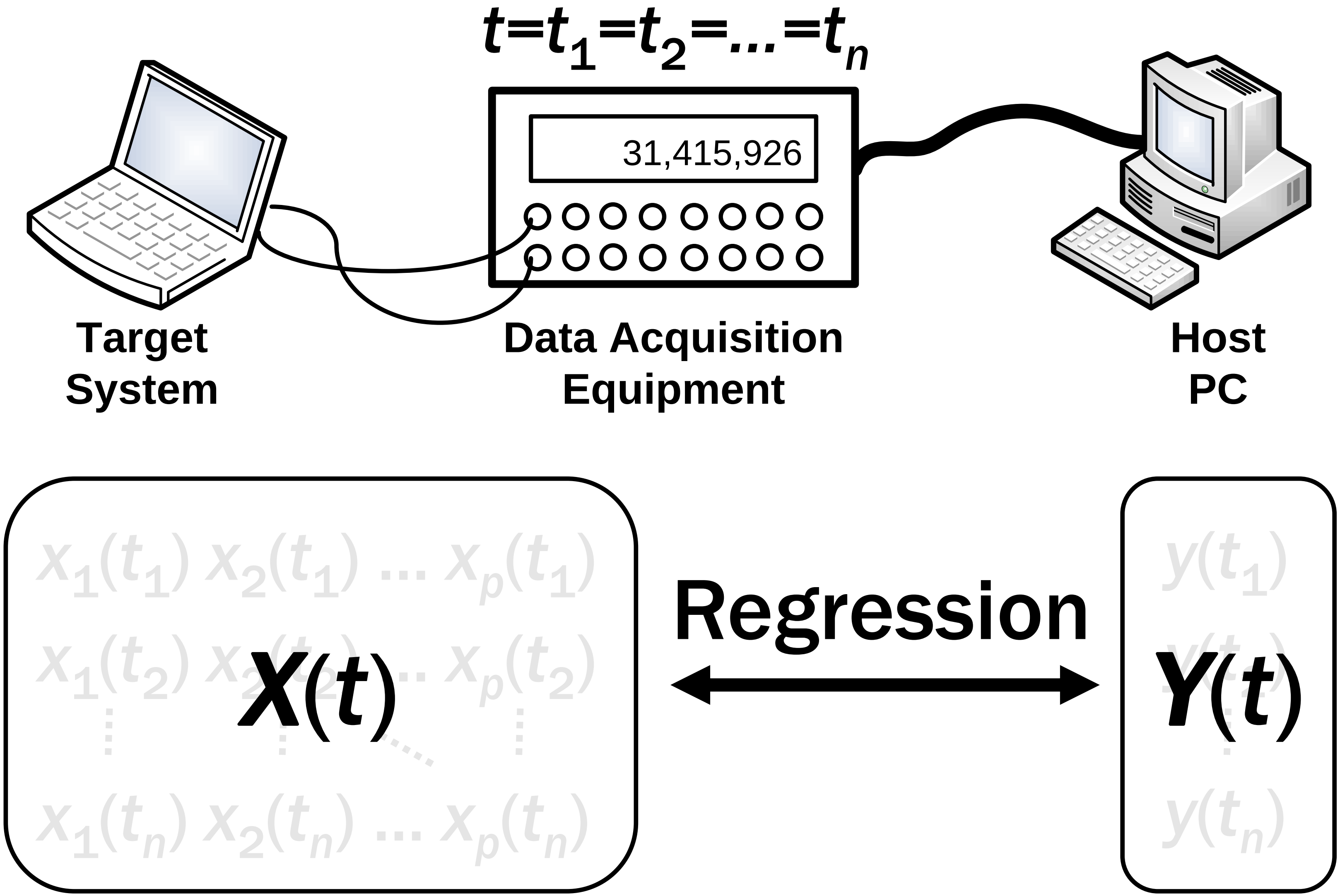


Model Construction

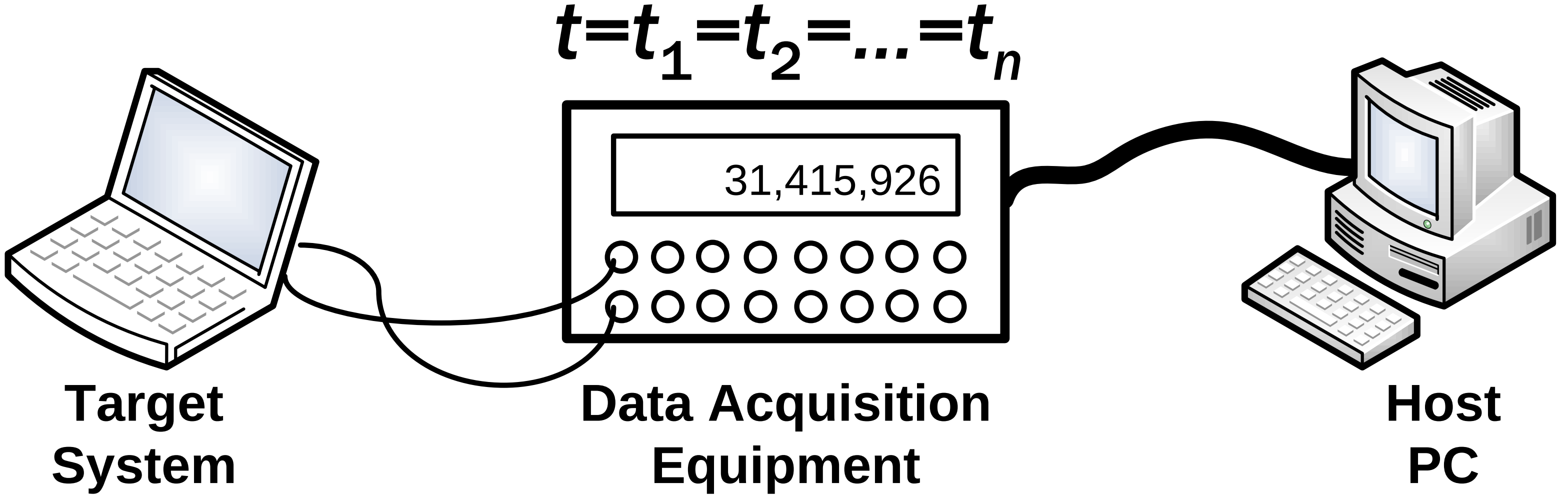


$$
\begin{matrix}
x_1(t_1) & x_2(t_1) & \dots & x_p(t_1) \\
x_1(t_2) & x_2(t_2) & \dots & x_p(t_2) \\
\vdots & \vdots & \ddots & \vdots \\
x_1(t_n) & x_2(t_n) & \dots & x_p(t_n)
\end{matrix}
\qquad
\begin{matrix}
y(t_1) \\
y(t_2) \\
\vdots \\
y(t_n)
\end{matrix}
$$

Model Construction



Model Construction

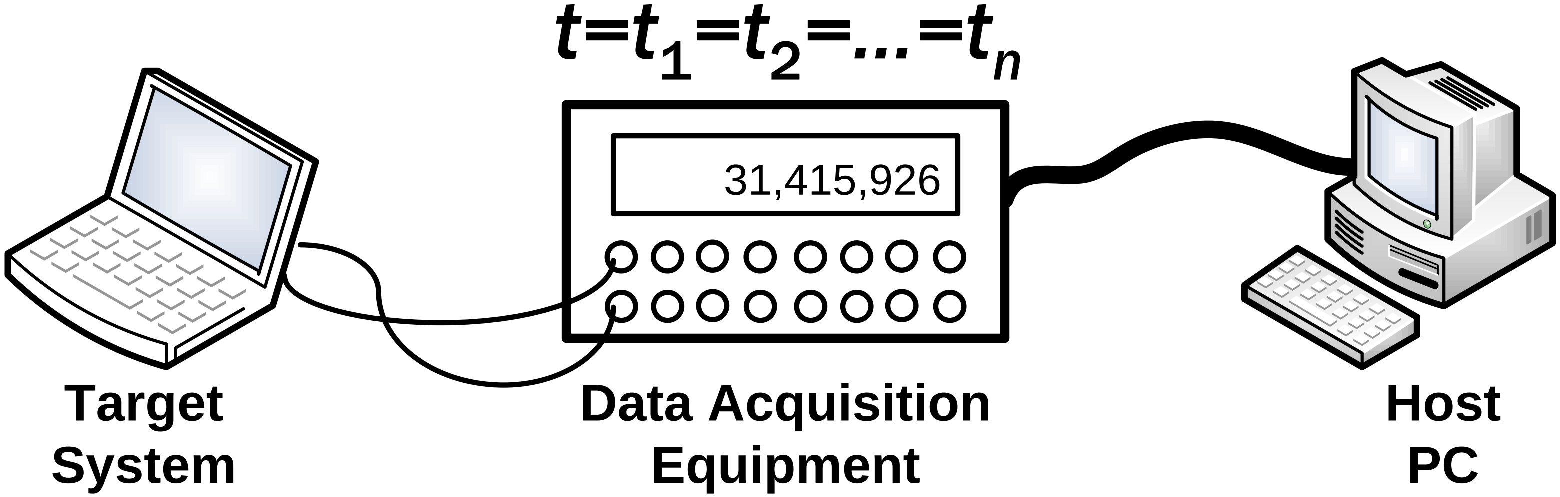


Linear Model:

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

$$\hat{\beta} = \text{argmin}_{\beta}(\|Y(t) - [1 \;\; X(t)]\beta\|_2)$$

Model Construction



Linear Model:

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

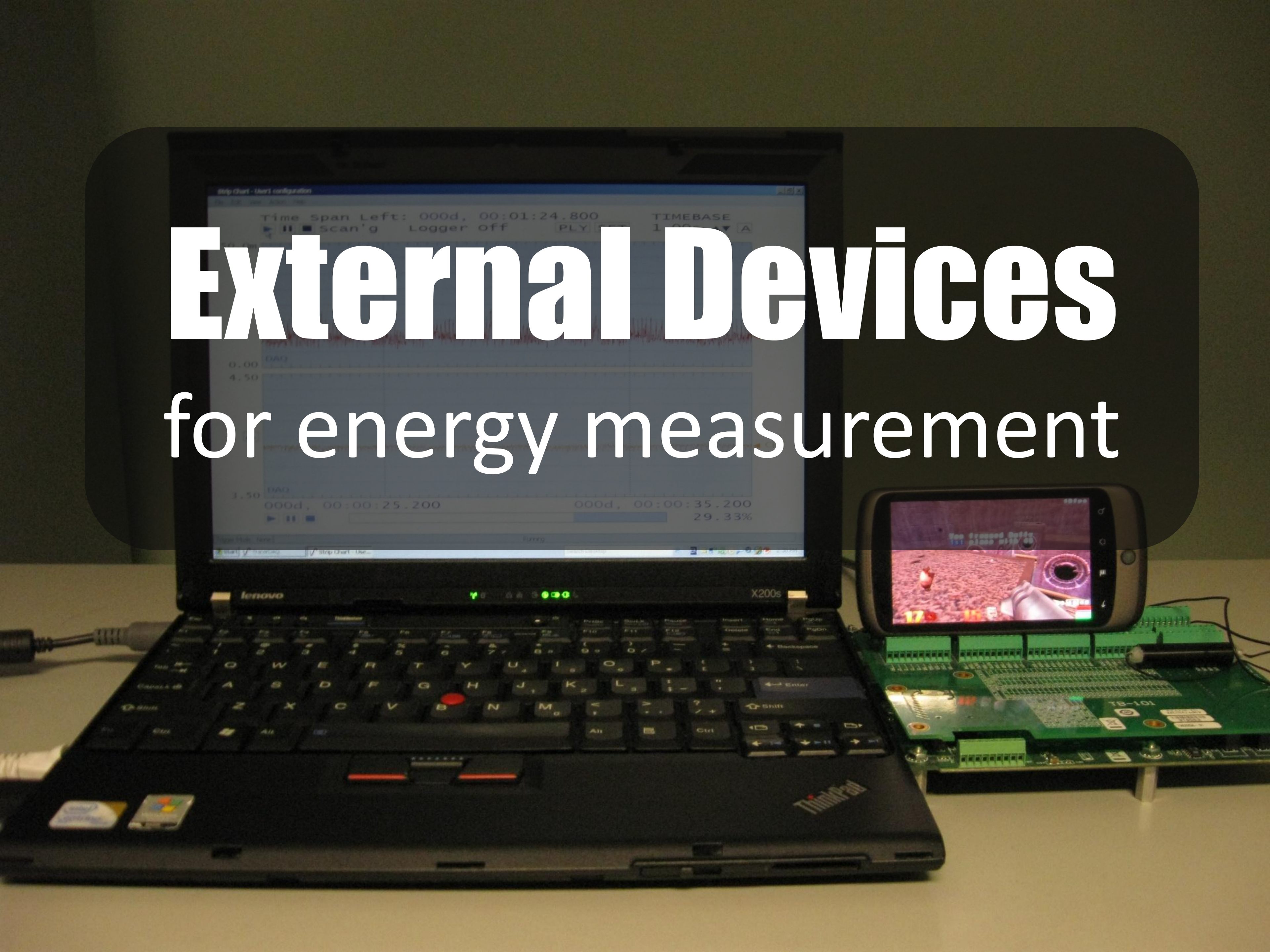
$$err(t_i) = \frac{\hat{y}(t_i) - y(t_i)}{y(t_i)}$$

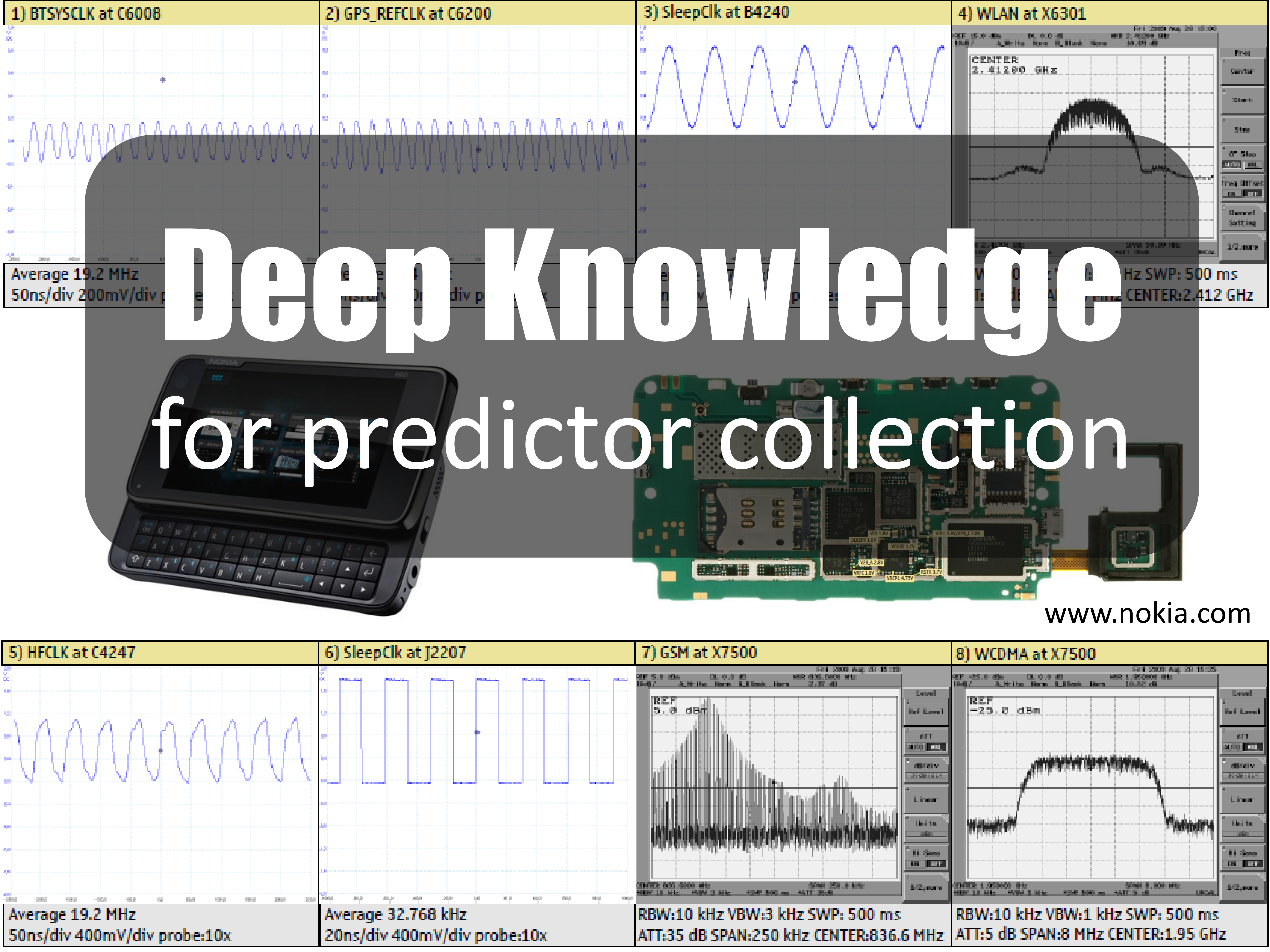
Mean Absolute
Root-Mean-Square

What are the limitations?

External Devices

for energy measurement




1) BTSYSCLK at C6008
2) GPS_REFCLK at C6200
3) SleepClk at B4240
4) WLAN at X6301
CENTER 2.41200 GHz
Average 19.2 MHz
Hz SWP: 500 ms
CENTER:2.412 GHz
Deep Knowledge
for predictor collection
www.nokia.com
5) HFCLK at C4247
6) SleepClk at J2207
7) GSM at X7500
8) WCDMA at X7500
REF 5.0 dBm
REF -25.0 dBm
Average 19.2 MHz
50ns/div 400mV/div probe:10x
Average 32.768 kHz
20ns/div 400mV/div probe:10x
RBW:10 kHz VBW:3 kHz SWP: 500 ms
ATT:35 dB SPAN:250 kHz CENTER:836.6 MHz
RBW:10 kHz VBW:1 kHz SWP: 500 ms
ATT:5 dB SPAN:8 MHz CENTER:1.95 GHz

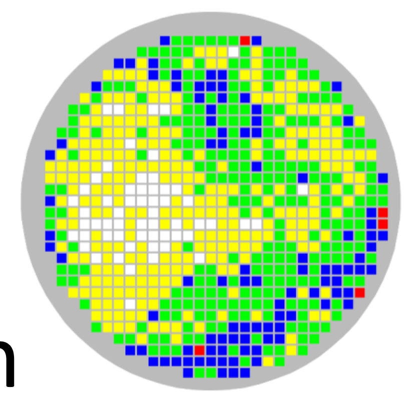

Exclusive Model
for a specific platform

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

External Devices

Deep Knowledge

Exclusive Model

Fixed Model

Battery Interface

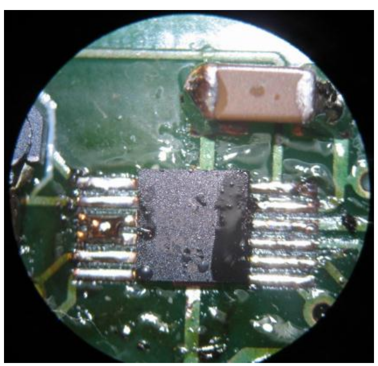
+

Statistical Learning

↓

Personalized Model

Battery Interface



State-of-the-art battery Interfaces are

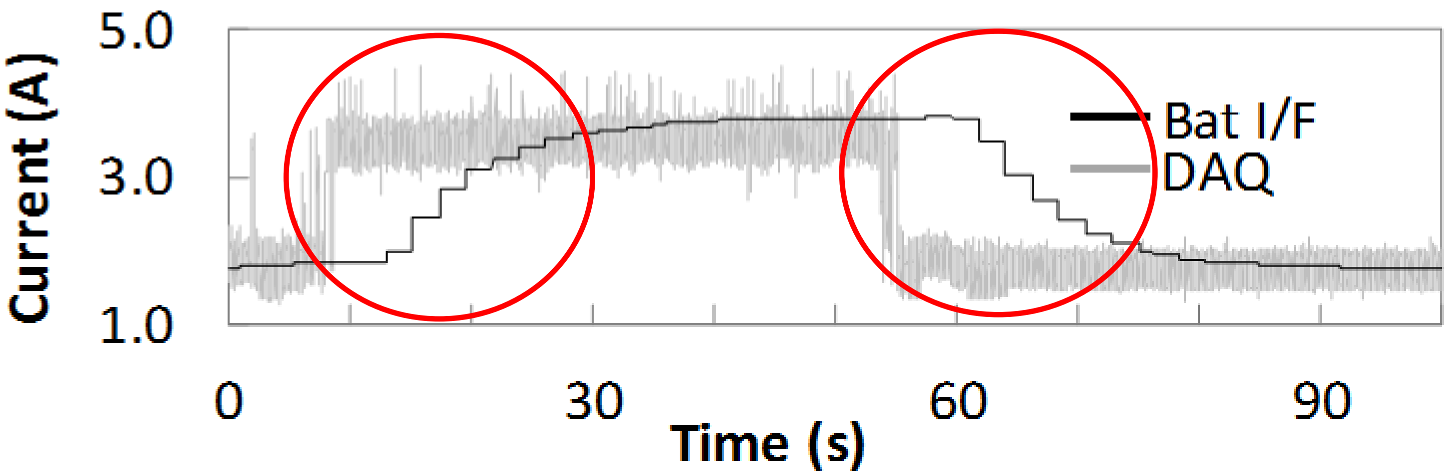
Low-rate/Inaccurate

	N85	T61	N900
Max Rate	4Hz	0.5Hz	0.1Hz
Accuracy	67%	82%	58%

Accuracy = 100% – Root_Mean_Square(Instant_Relative_Error)

Errors in battery interface readings

are **Non-Gaussian**



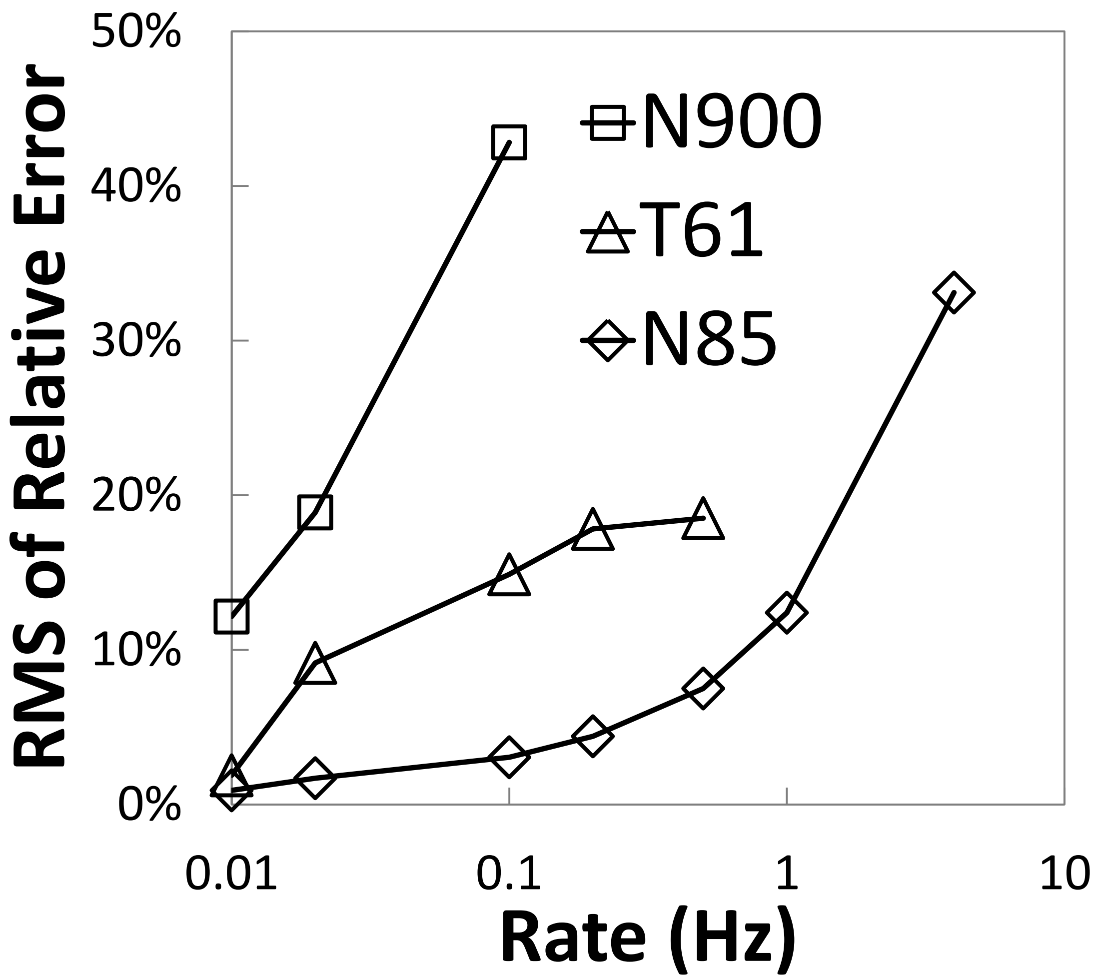
Low-Rate/Inaccurate
Battery Interface

⬇ **Statistical Learning**

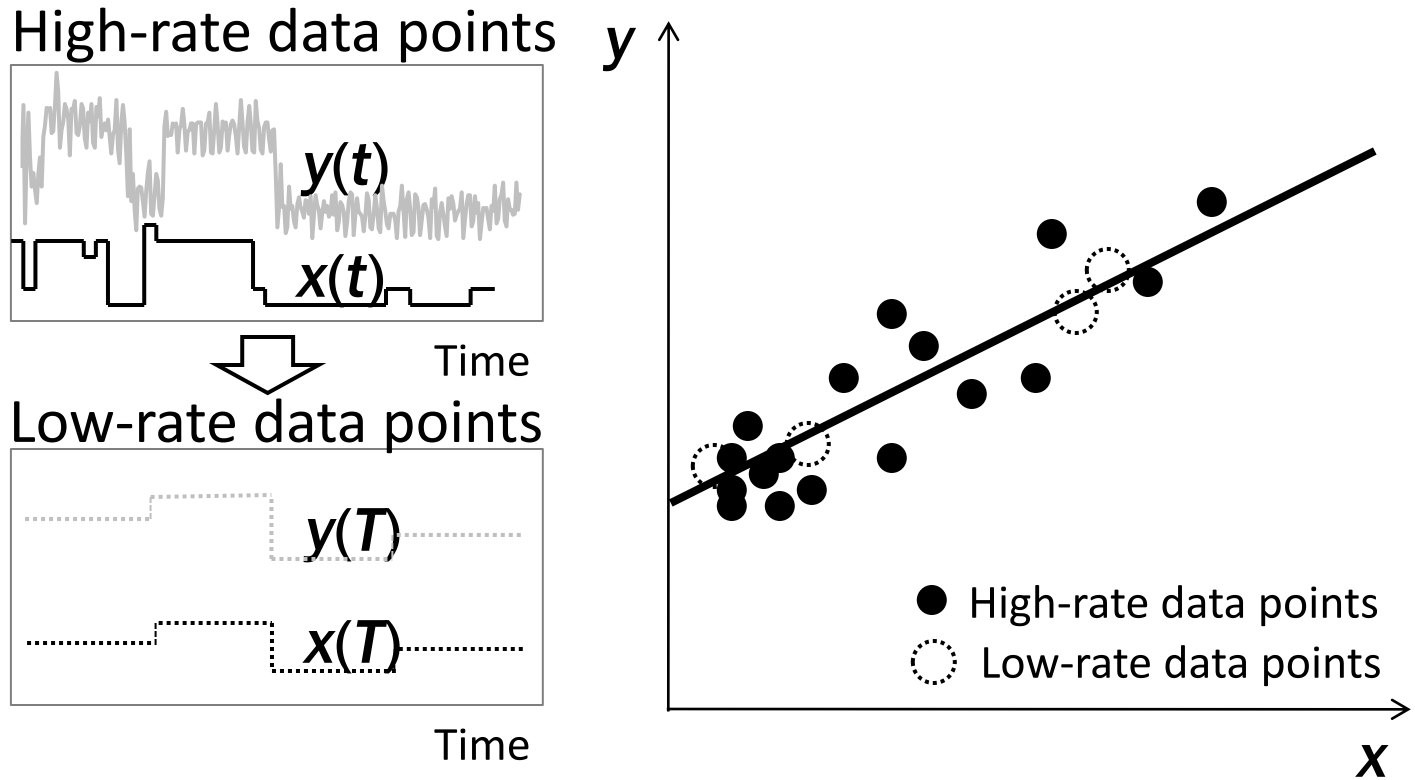
High-Rate/Accurate
System Energy Model

Averaged battery interface readings

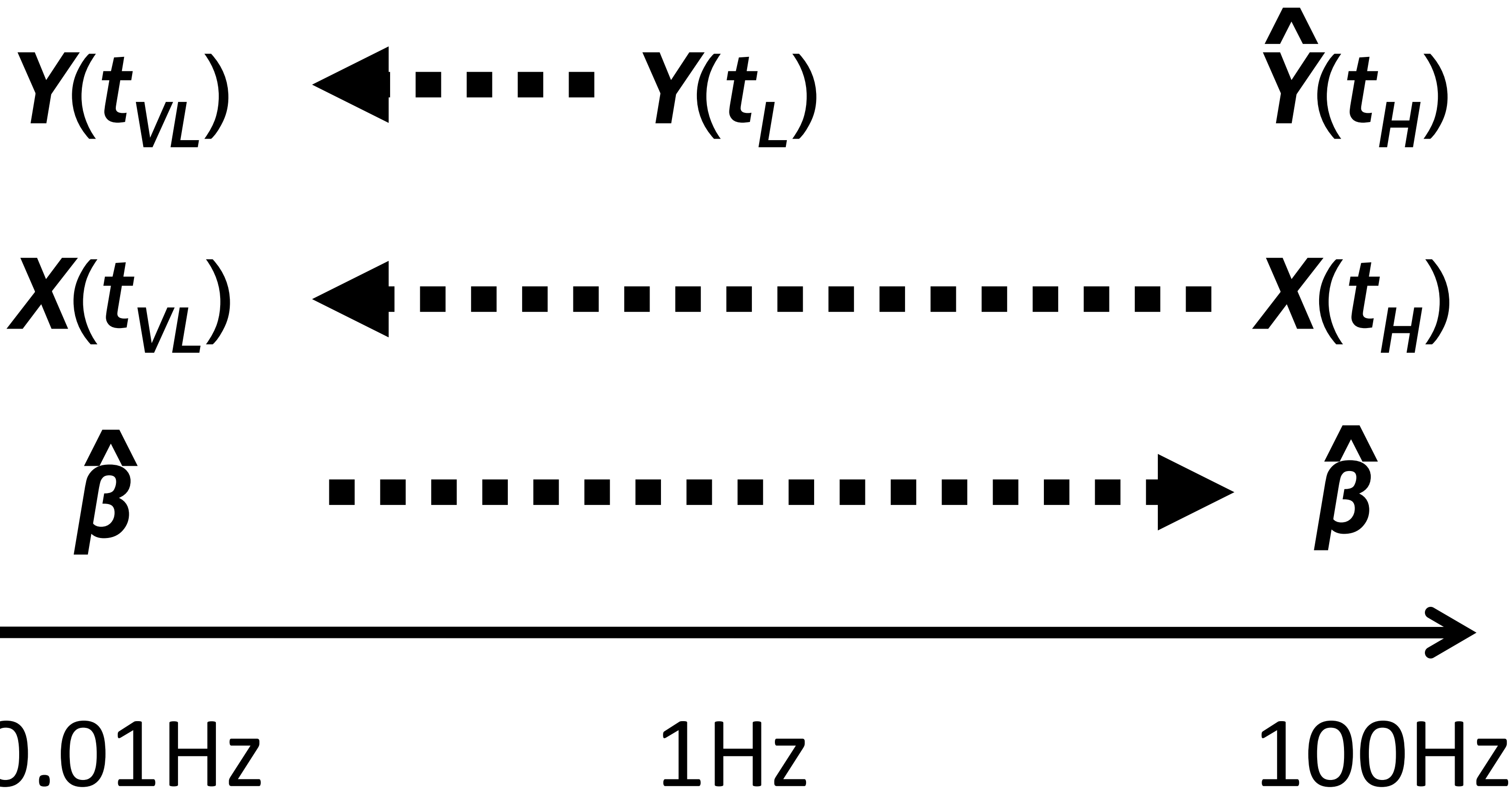
have
Higher Accuracy
but
Even Lower Rate

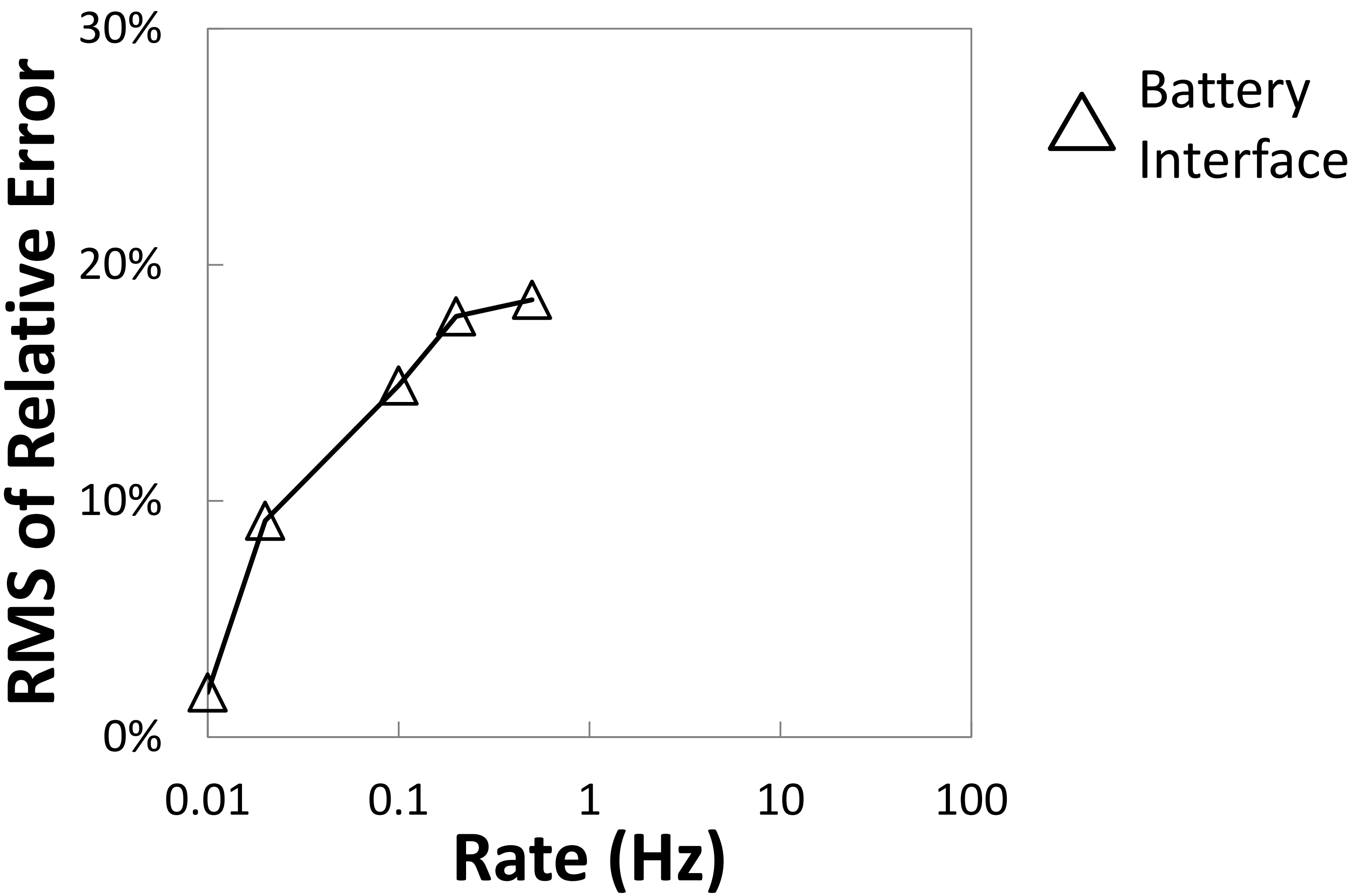


Linear models are **Independent on Time**

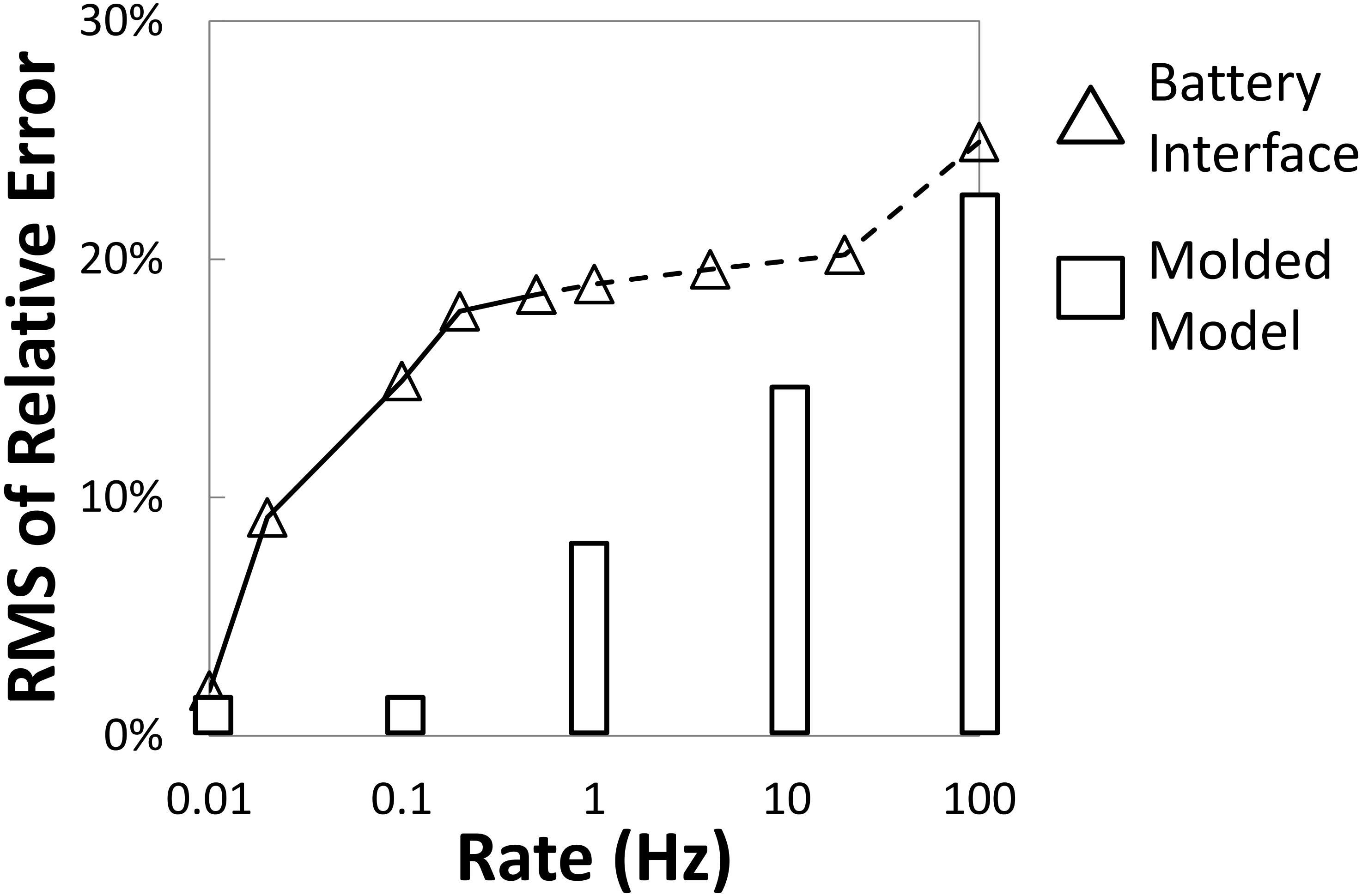


1. Model Molding




30%
20%
10%
0%
RMS of Relative Error
0.01
0.1
1
10
100
Rate (Hz)
Battery Interface

Model Molding improves rate



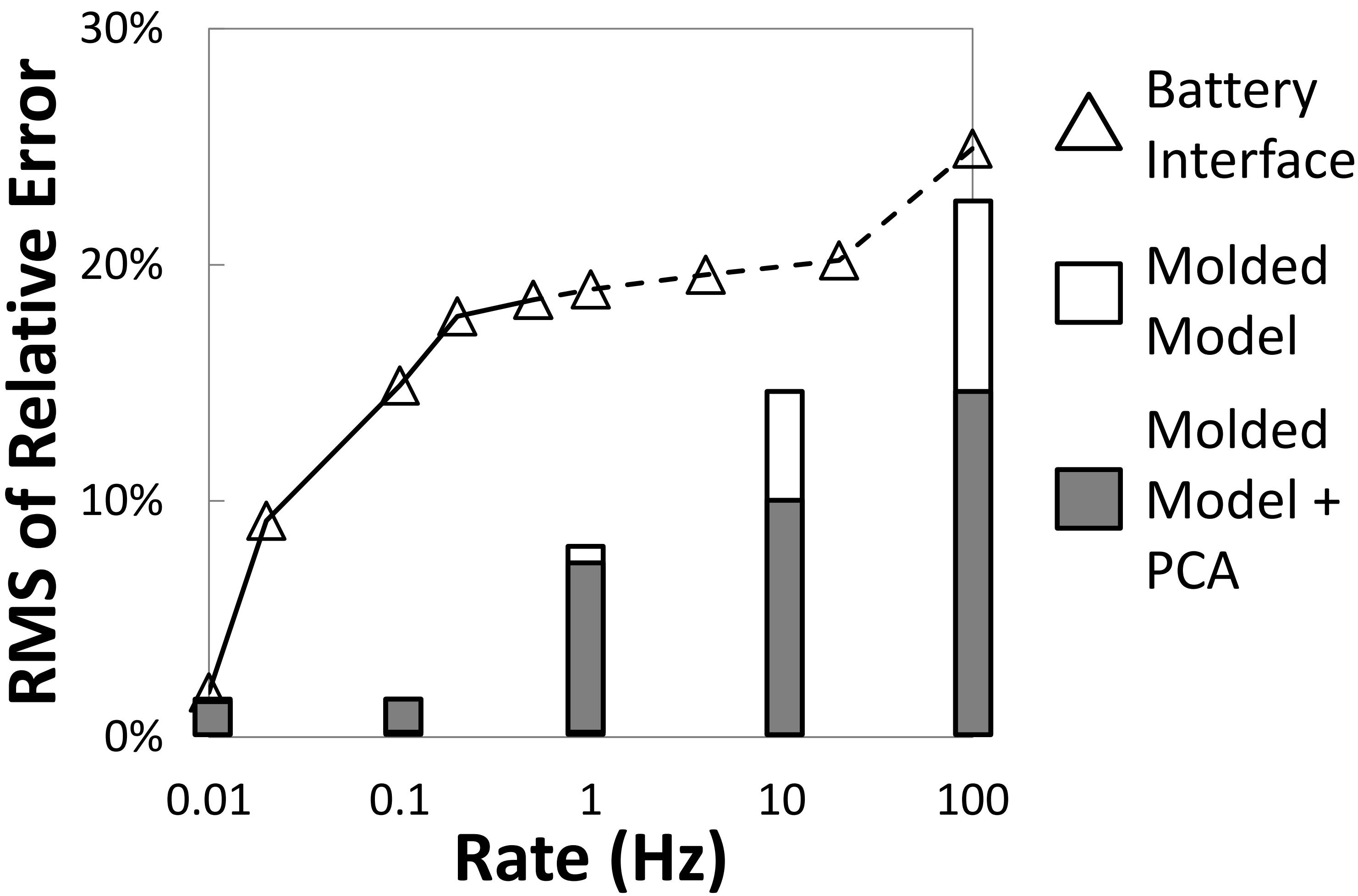
2. Predictor Transformation

$$x_1(t), x_2(t), \ldots, x_p(t)$$

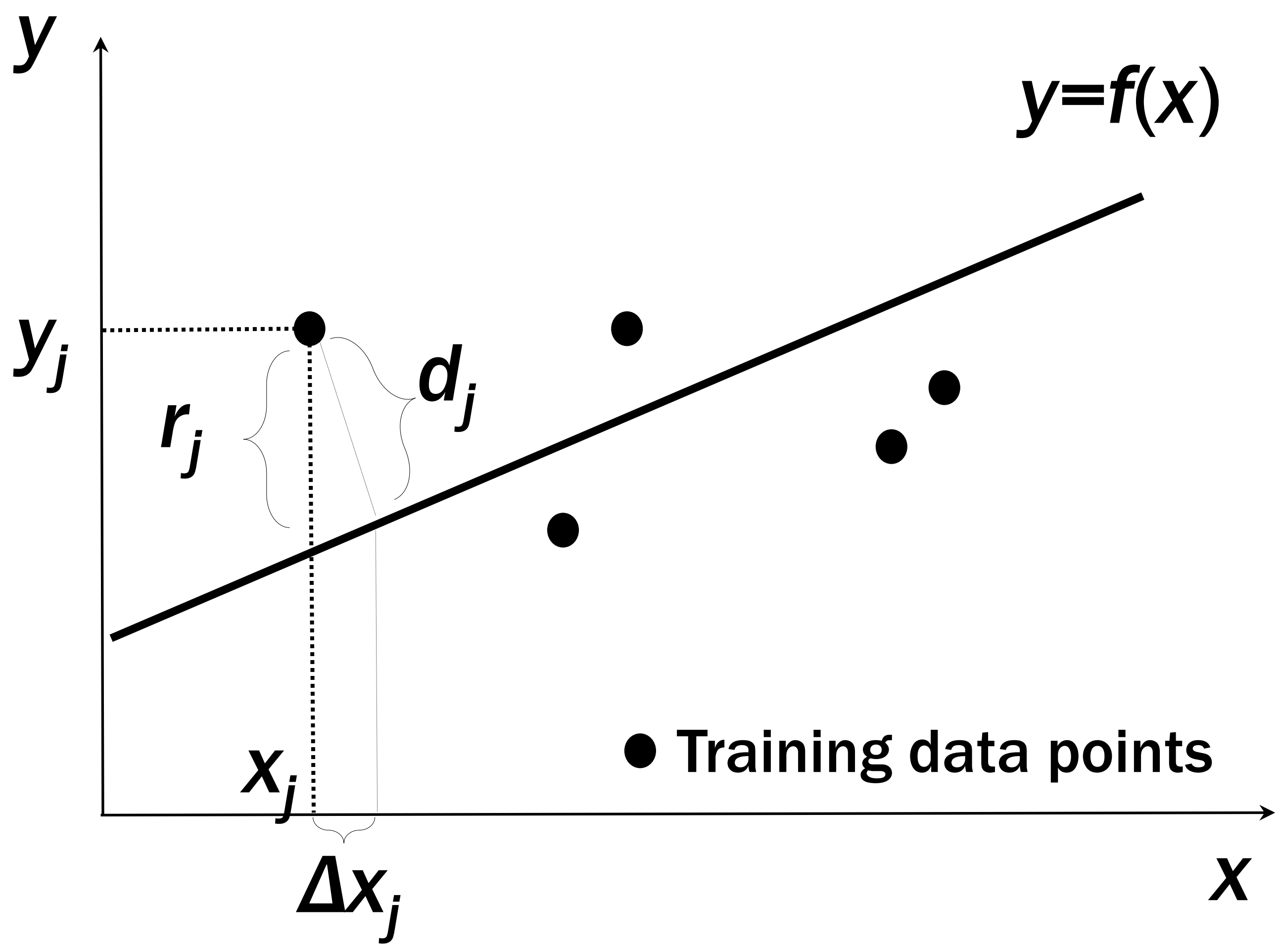
Principle Component Analysis

$$z_1(t), z_2(t), \ldots, z_L(t) \quad L \leq p$$

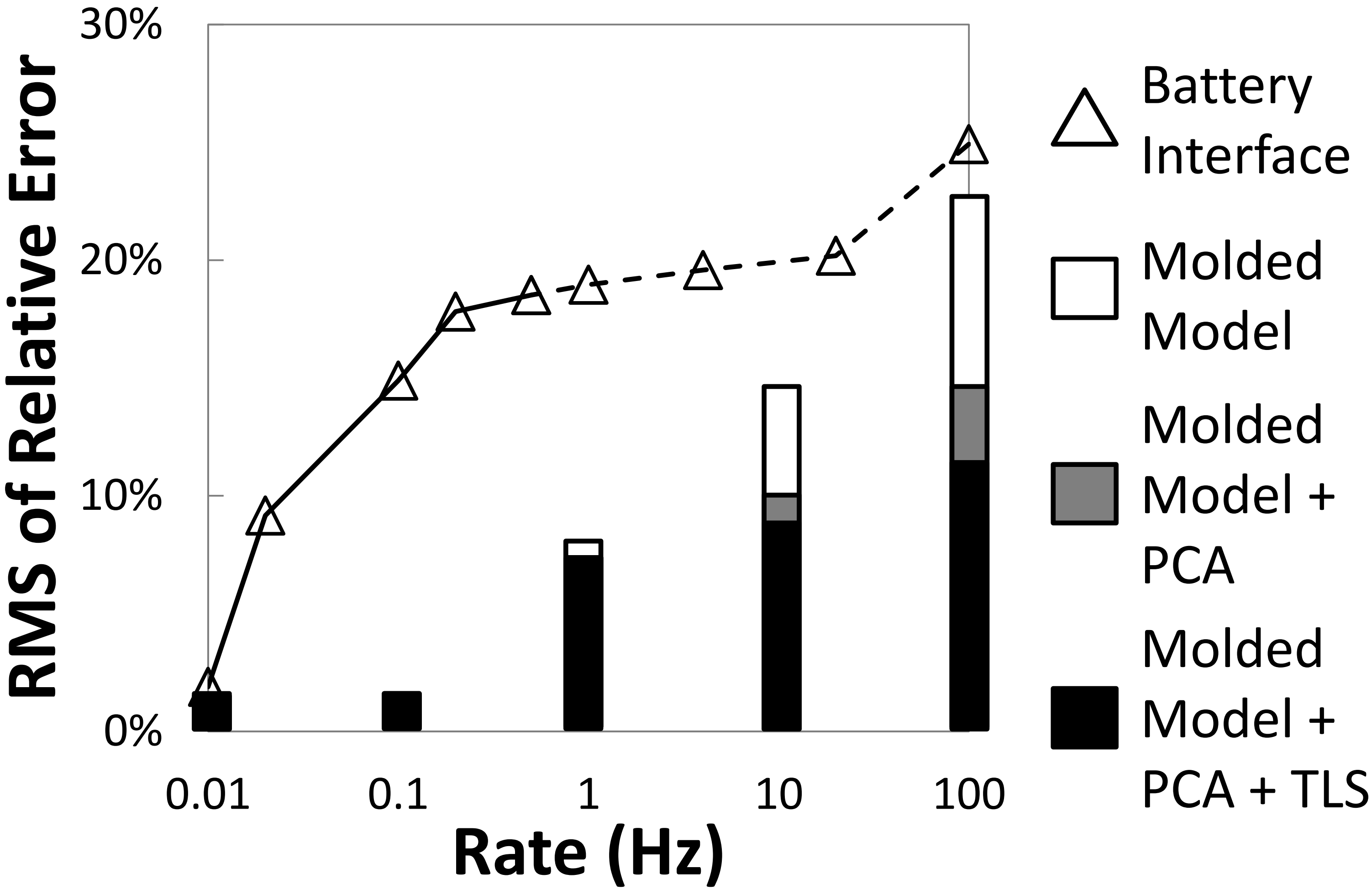
PCA improves accuracy

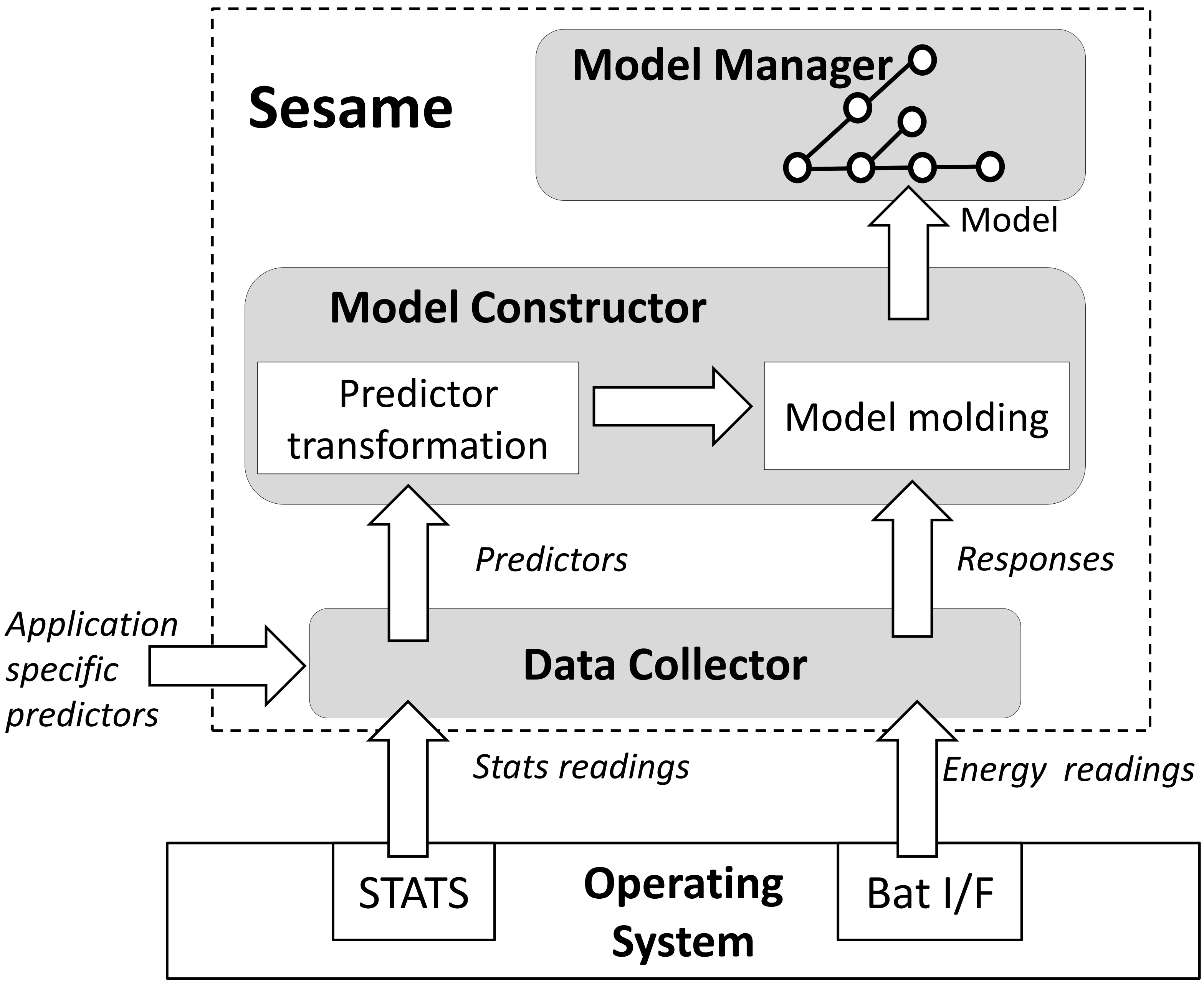


3. Total-Least-Square



TLS improves accuracy at high rate




Sesame
Model Manager
Model
Model Constructor
Predictor transformation
Model molding
Predictors
Responses
Application specific predictors
Data Collector
Stats readings
Energy readings
STATS
Operating System
Bat I/F

Implementation

N900

T61

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

	T61	N900
1Hz	95%	86%
100Hz	88%	82%

Accuracy = 100% – Root_Mean_Square(Instant_Relative_Error)

Field Study

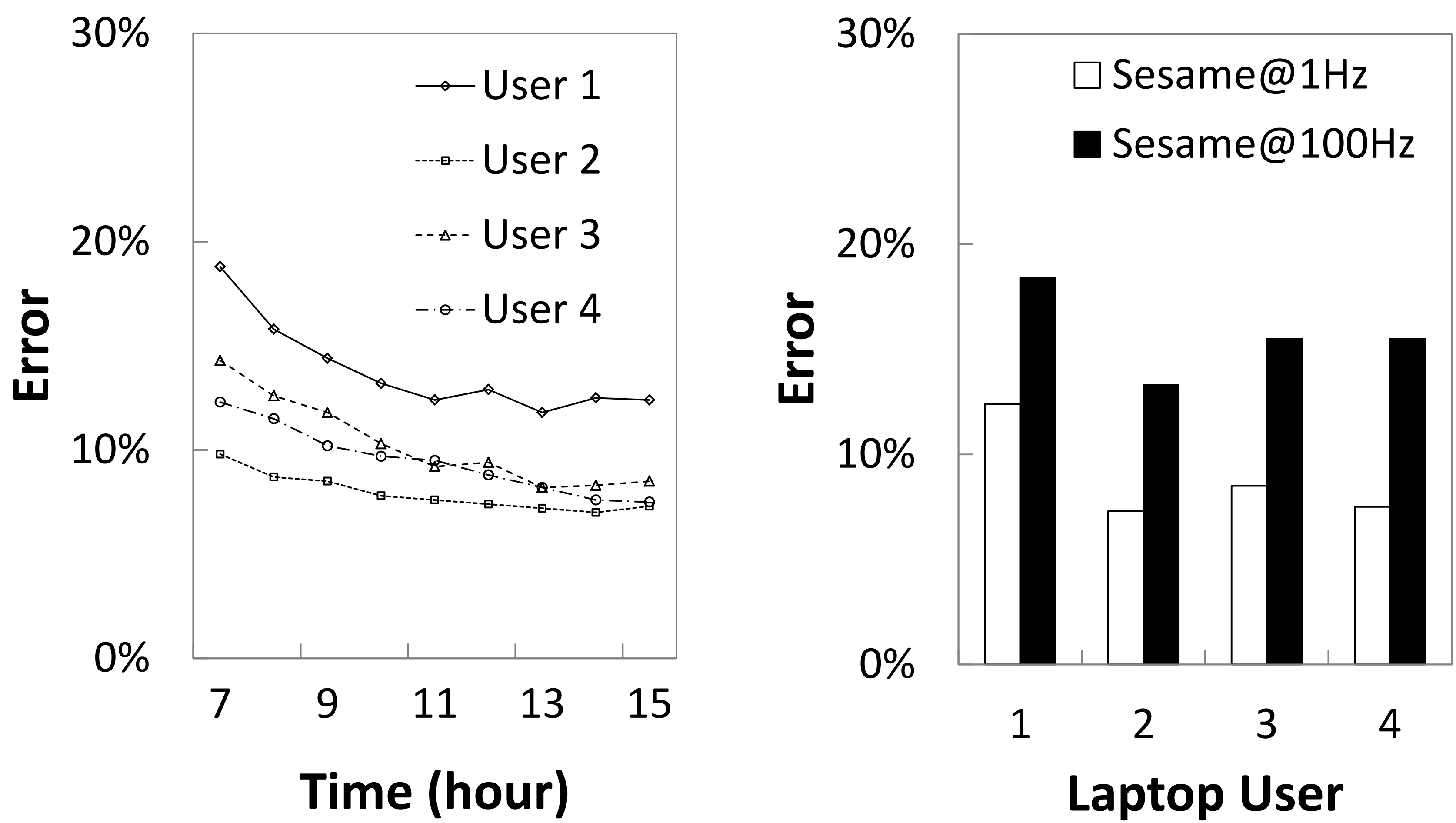
Day 1-5:

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

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

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

$$\vdots \qquad \vdots \qquad \vdots$$

$$x_p(t) = x_{p,1}(t) + \ldots + x_{p,n}(t)$$

Energy Contribution by Process j

$$y_j(t) = \beta_1 x_{1,j}(t) + \ldots + \beta_p x_{p,j}(t)$$

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