



@EVSymposium29
#EVS29

Electric Vehicle Symposium & Exhibition

June 19-22, 2016 | du 19 au 22 juin 2016

Montréal, Québec, Canada

Organized by:



Hosted by:



In Collaboration with:



Platinum Sponsors:



DRIVE ELECTRIC INNOVATION
PROPULSER L'INNOVATION EN MOBILITÉ ÉLECTRIQUE

EVS29.ORG



Real-Time Estimator Li-ion Cells Internal Resistance for Electric Vehicle Application

By **Alexandre O. Tessier**, Maxime R. Dubois and João P. Trovão

Electric – Transport, Energy Storage and Conversion Laboratory
Université de Sherbrooke, J1K 2R1, Québec, Canada



UNIVERSITÉ DE
SHERBROOKE

Voir au futur



e-TEESC Lab

electric - Transport,

Energy Storage and Conversion



Outline

- Context
- Short review on Li-ion internal resistance
- Short review on Battery management systems
- Data logging issues when applied on long periods
- New data logging method applicable for large scale BMS
- Recovering and filtering the stored data
- Results
- Future work

Context

- Electric Vehicles use thousands of li-ion cells
- Cells individuality may cause premature degradation if not monitored / managed
- Battery Management Systems are added to li-ion battery pack to perform those tasks
- Internal resistance is one of the most important parameters to acquire
- This study present a new way to manage memory in order to increase the complexity of the model without loosing its real-time feature

Short review on Li-ion internal resistance

- Internal resistance is defined as the equivalent **impedance** that would mimic the same voltage drop at the cell poles when submitted to a given current demand
- Internal resistance is known to change with :
 - Current amplitude & direction
 - Cell temperature (low temp brings high R_{in})
 - State-of-Charge (SOC)
 - Previous abuse (over discharges and over charges)
 - Age

Do we want to take all those in account with a very complex model ?

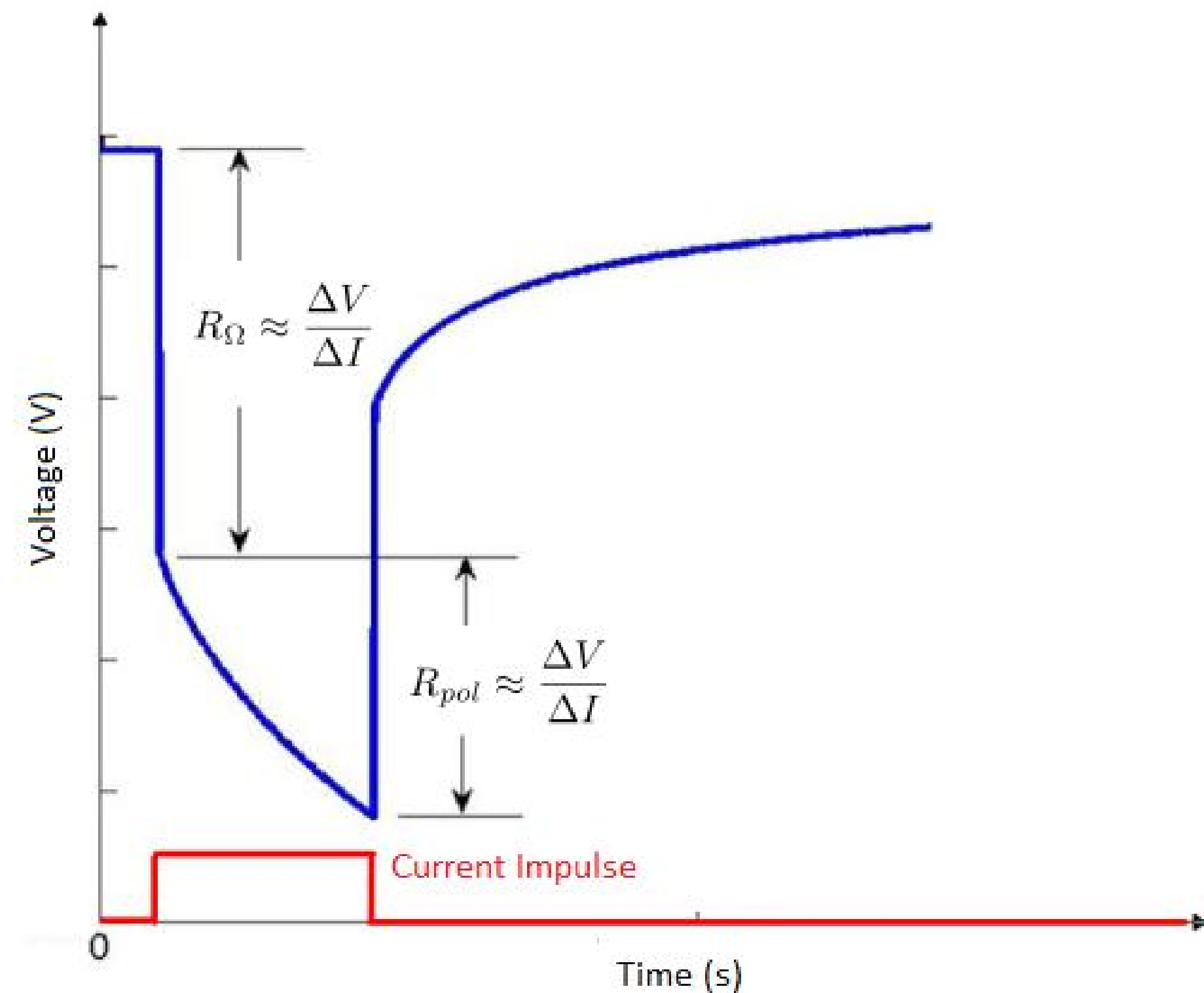
Answer : **NO !**

Why ? Because the CPU wont be able to compute it for every cells in a real-time system

Short review on Li-ion internal resistance

- Simple way to acquire the R_{in} :

$$R_{in} \approx \frac{\Delta V}{\Delta I}$$



Short review on battery management systems

Key roles :

- Safety
- Gather information for the user (like remaining energy)
- Take action when limits are reached
- Balance cell and optimize cell usage

Short review on battery management systems

BMS CPU Limitations :

- Optimize for (+ - x ÷) operations and binary manipulations
- Complex operation like power, square roots, exponential and logarithm are very time consuming and will shortly cause the system to lose its real-time capabilities
- Working with integers is faster and cheaper than working with floating point numbers
- Increasing CPU speed and arithmetic modules always come with a price

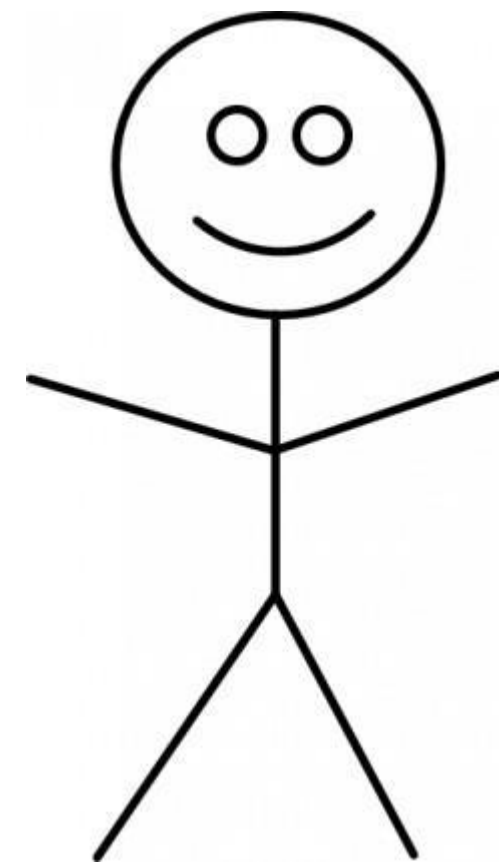
Data logging issues when applied on long periods

- In order to make trends of R_{in} variations depending of slow and fast changing parameters, the BMS must gather a large amount of data.
- RAM memory is not large enough to store every data in a BMS life
- Bigger but slower memory (generally FLASH) is used for long term recording
- **Chronological data storing system** is the typical way
- While size is not a major issue with today's technologies, **navigating** through a large quantity of data is one
- CPU speed can be increased easily (at a cost), but memory chip speed cannot be increased at will.

Data logging issues when applied on long periods

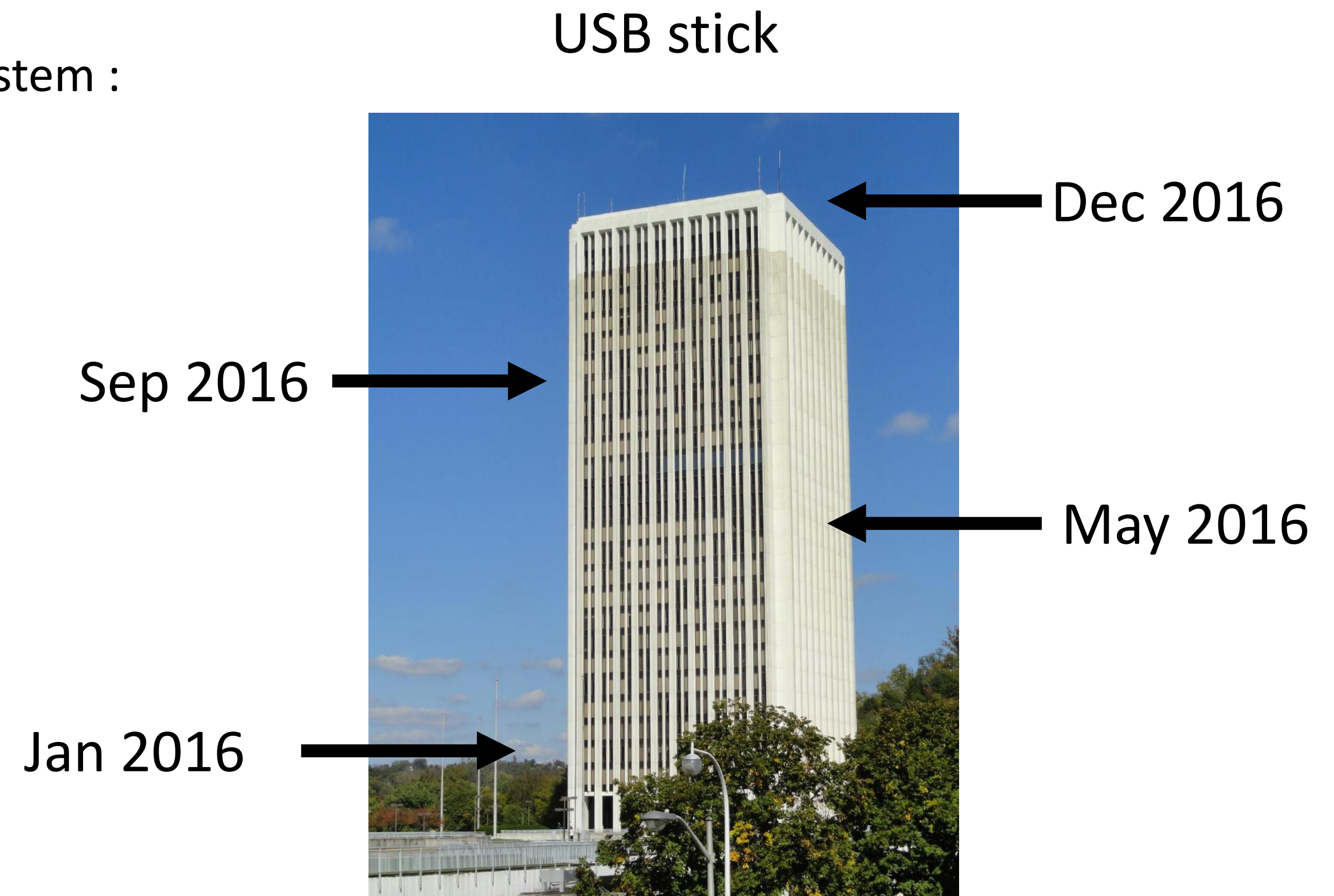
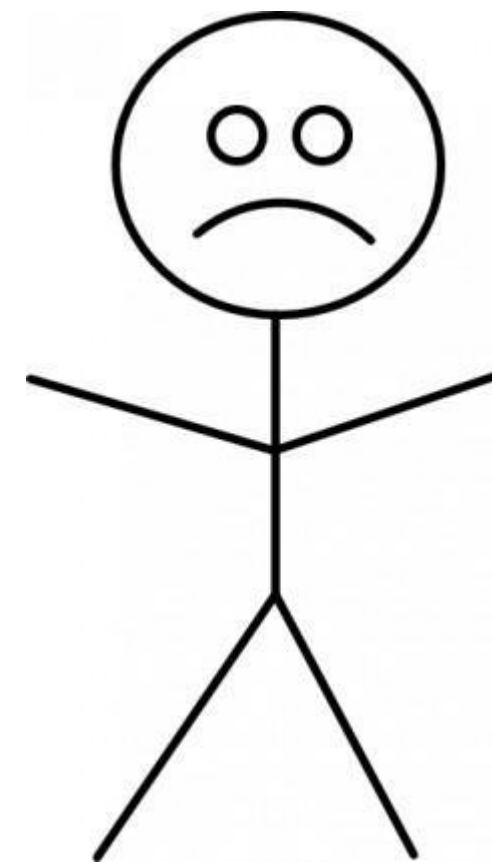
- Chronological data storing system :

USB stick



Data logging issues when applied on long periods

- Chronological data storing system :



Data logging issues when applied on long periods

- Chronological data storing system :

CPU



Hey, send me old data

I don't know, from the newest to the oldest

Ok, which one

Ok ...

USB stick



Data logging issues when applied on long periods

- Chronological data storing system :

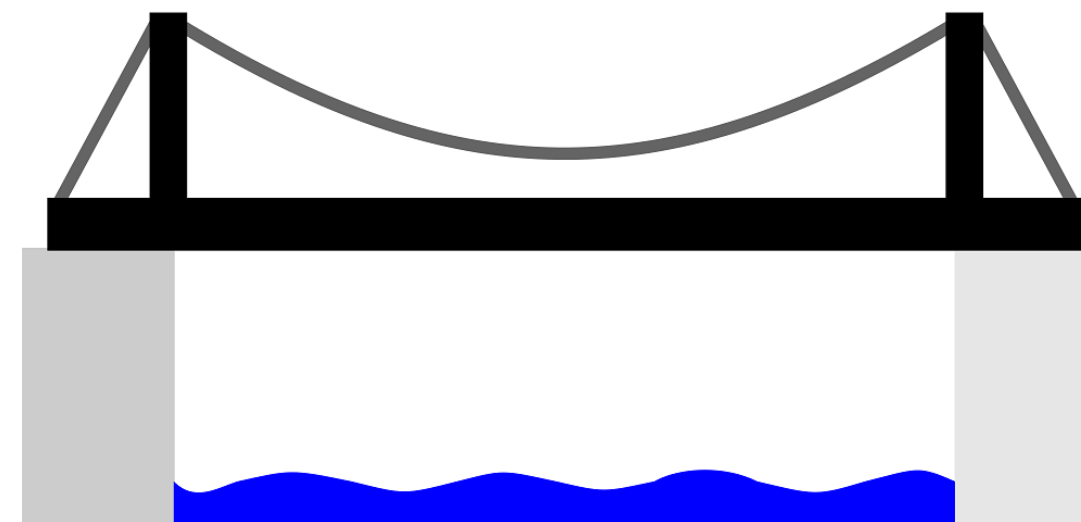
CPU



Problem here



Communication
protocol



USB stick



Data logging issues when applied on long periods

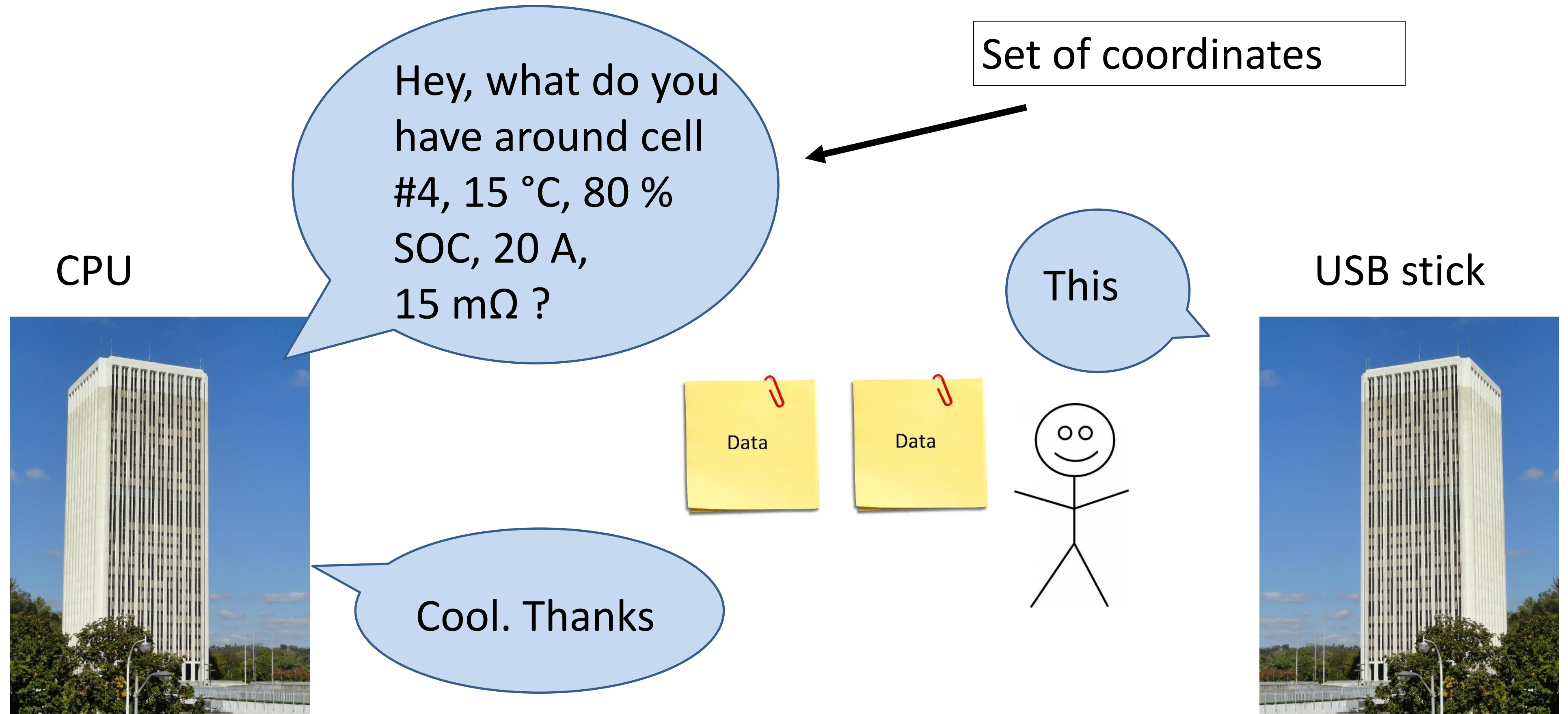
- When was the last time I had a strong acceleration like this ?
- When was the last time I had a hot day like today ?
- When was the last time I had a empty battery pack like this ?

All these answer will give interpretation to the measured R_{in} and cannot be answered with chronological data storing systems

New data logging method applicable for large scale BMS

- Instead, what if we would use the same building, but we place data in an other format:
 - Forget about the timestamp
 - Every floor is a cell
 - Every hall is a temperature
 - Every room is a state-of-charge
 - Every cabinet is a current value
 - Every drawer in the cabinet is a Rin
- }
- 5 dimension Matrix where coordinates represents measured conditions

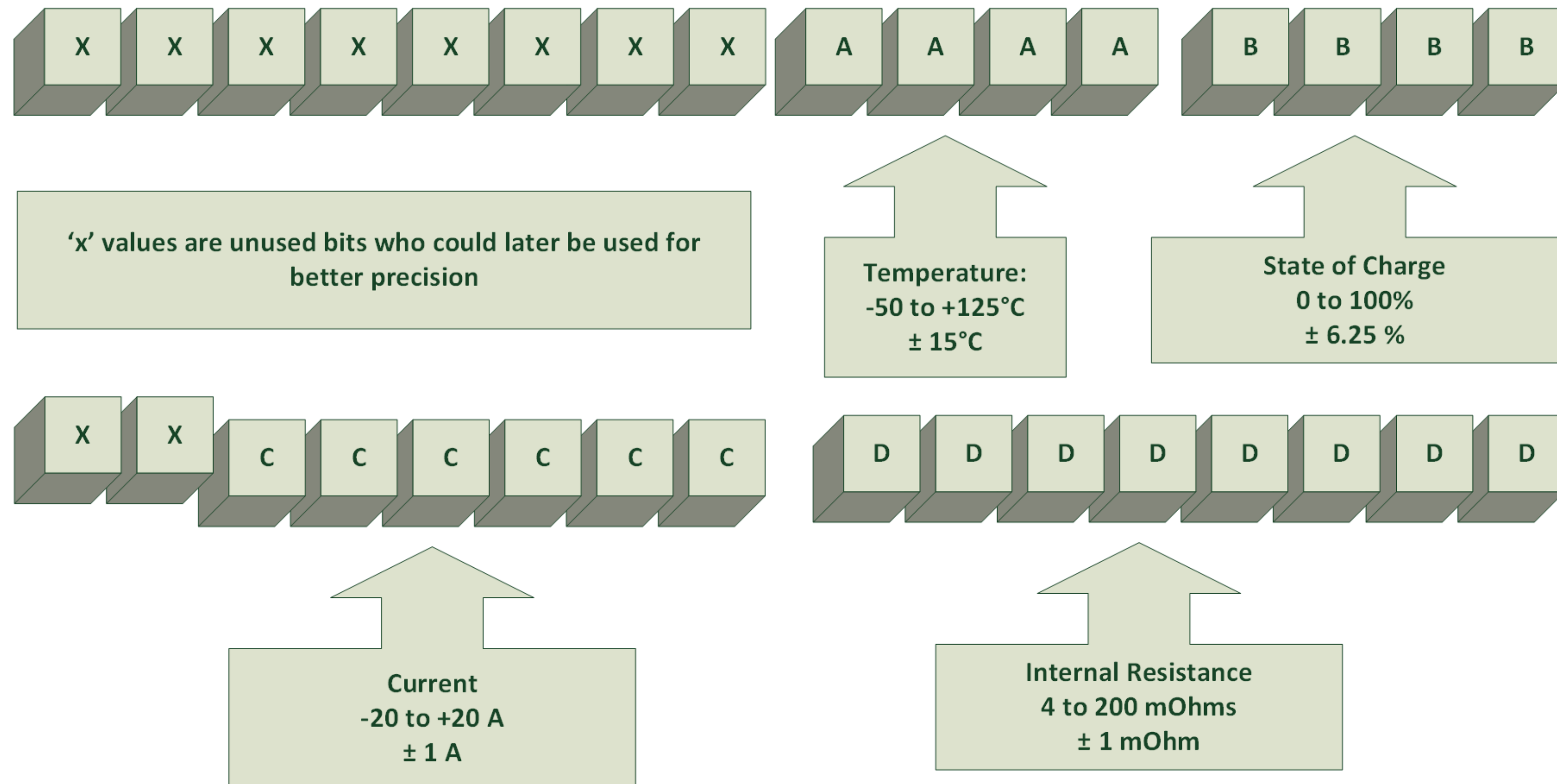
Data logging problems when applied on long periods



New data logging method applicable for large scale BMS

- The new method :
 - Convert the recorded values as a physical address in memory (32 bit)
 - like file cabinet drawer
 - For each new measure, you add 1 unit in the memory slot associated to these coordinates (like a **counter**)
 - you add a paper to the drawer
 - When you want to validate or correlate data, you look at surrounding memory slots around current point of measure (a **cluster**)
 - you look if any other drawer have more paper than yours (suggesting a misreading)
 - Delete old data by decaying every slot by 1 count every now and then
 - To prevent overflow and allow slow moving parameters to drift

New data logging method applicable for large scale BMS



Recovering and filtering the stored data

- Once surrounding clusters are recovered in RAM, you want to apply a low-pass filter to get the general trend
- Numeric low pass filter coefficients working on many axis can be generated, but it's very heavy on CPU time. Instead, we propose an average filter with only one axis : the distance with the current cluster's center.
- The closer the memory slot is from the center, the stronger the weight applied

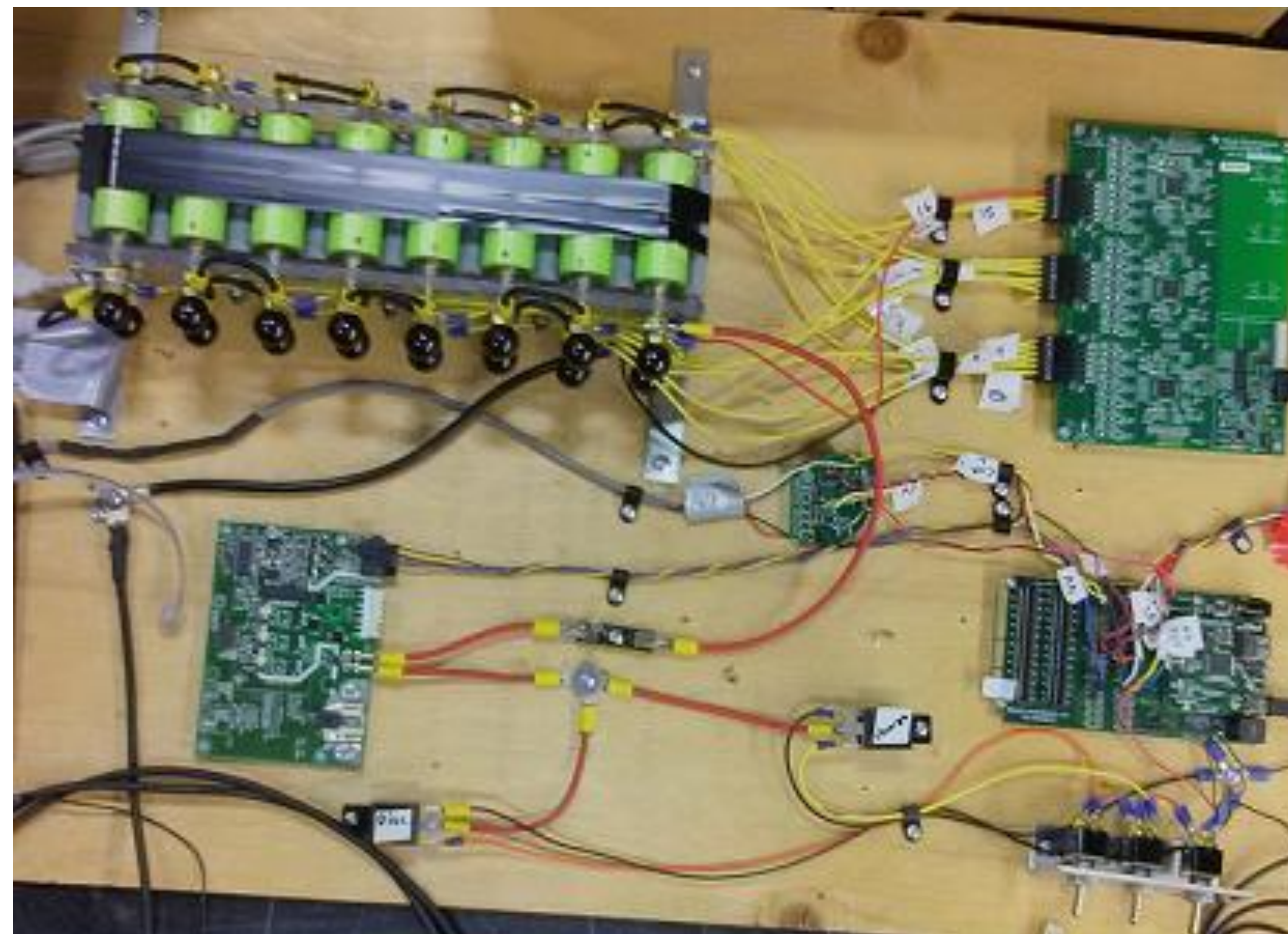
Recovering and filtering the stored data

$$R_{in\ filtered} = \frac{\sum_{\forall k \in \mathbb{E}} (R_{in(k)} * Count_{(k)} * Weight_{(k)})}{\sum_{\forall k \in \mathbb{E}} (Count_{(k)} * Weight_{(k)})}$$

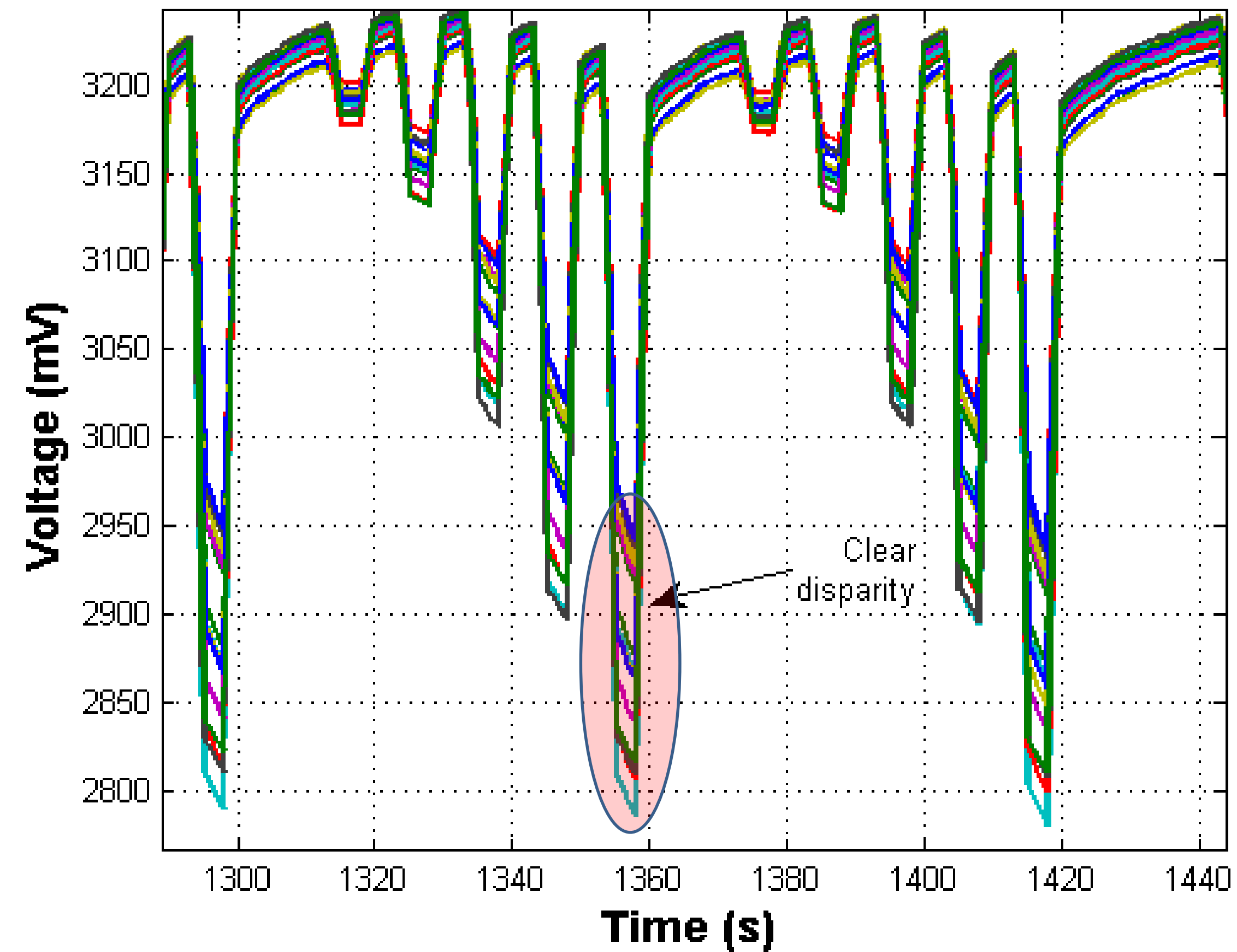
New data logging method applicable for large scale BMS

- Strength:
 - Relevant data's position is known
 - Recovering time / quantity of memory access is known and **constant**
 - Influence of each parameter on the Rin is now kept in memory
 - **CPU Friendly**
- Weakness:
 - Lose time reference
 - Some losses in precision

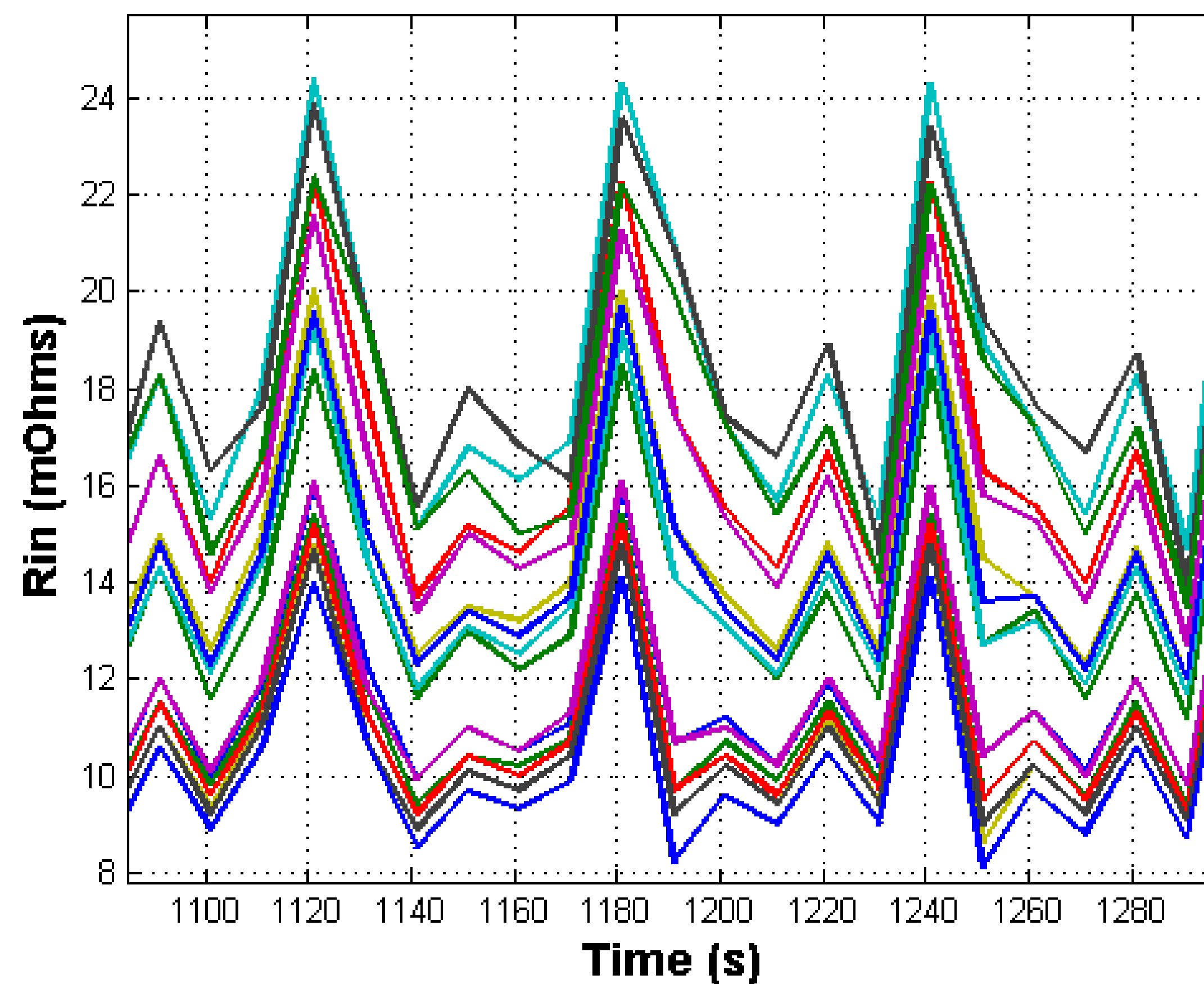
Test bench



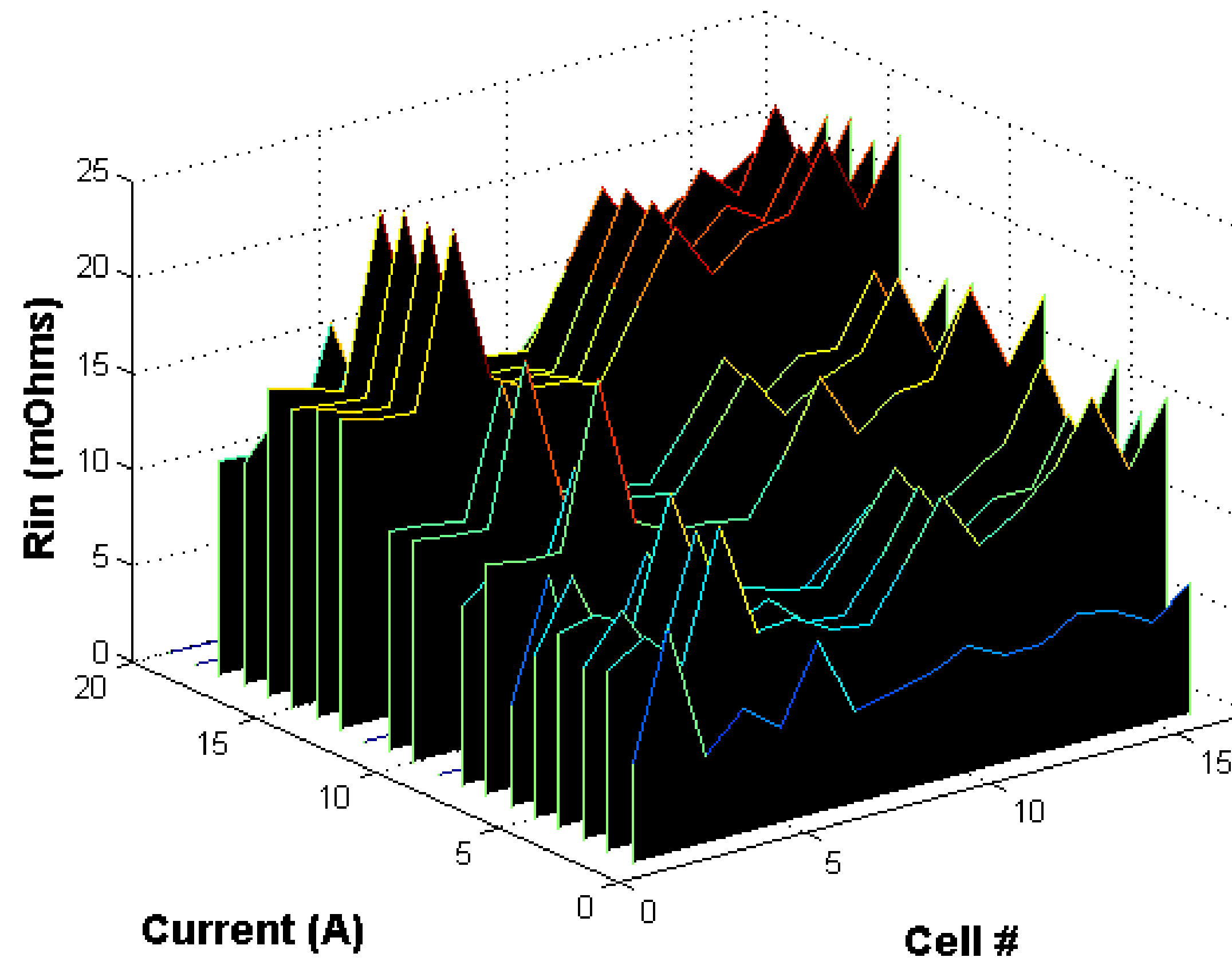
Results



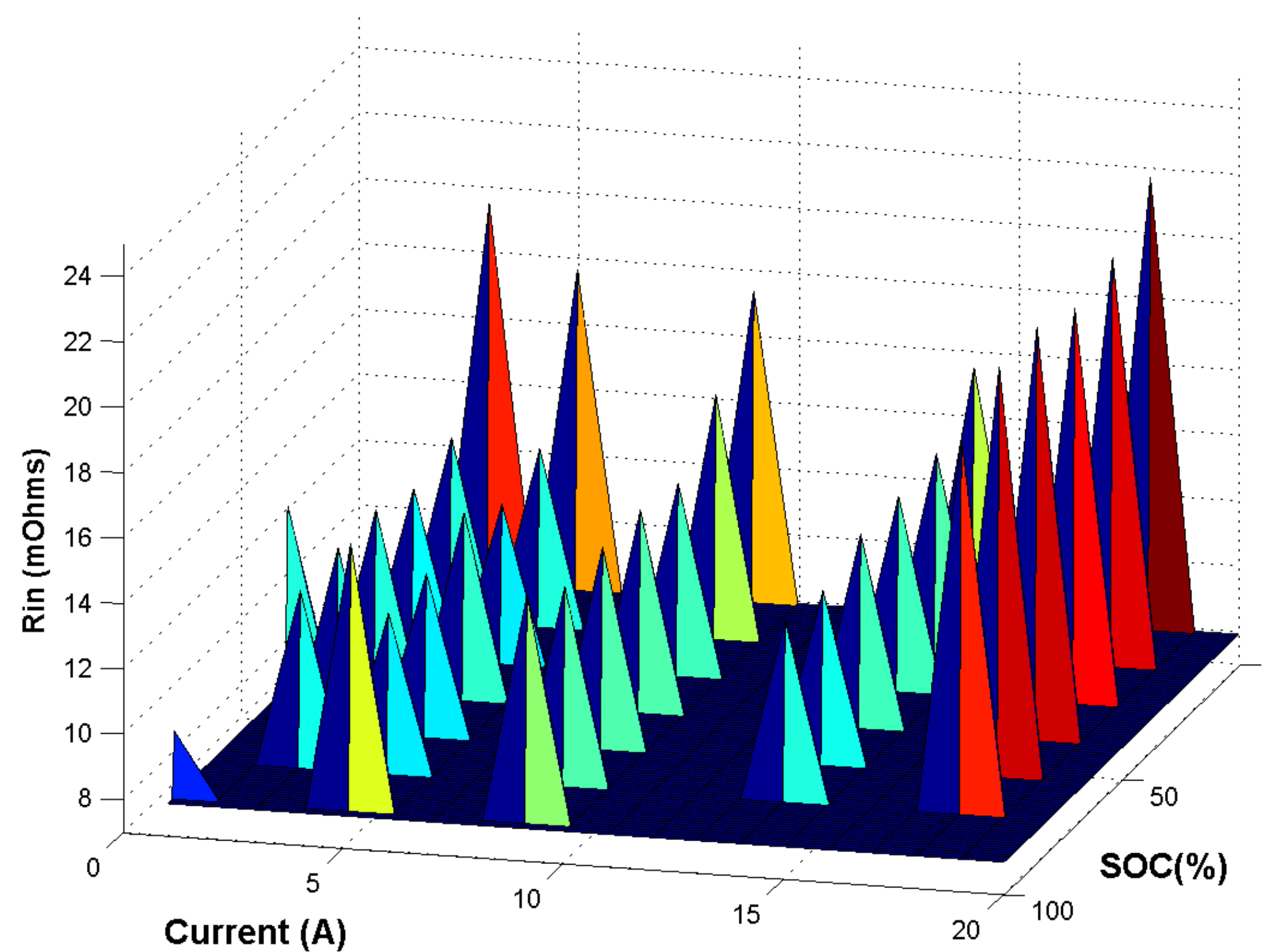
Results



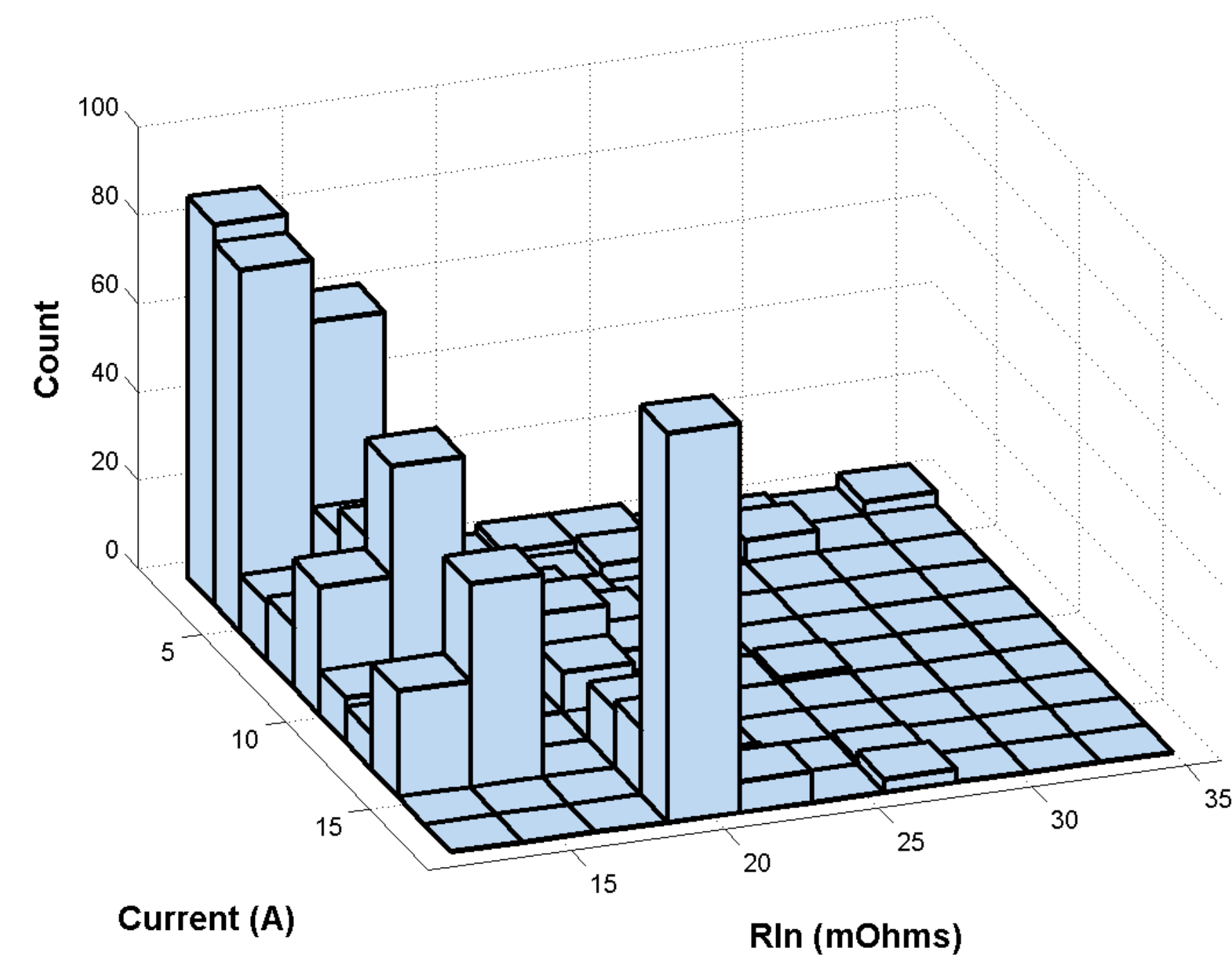
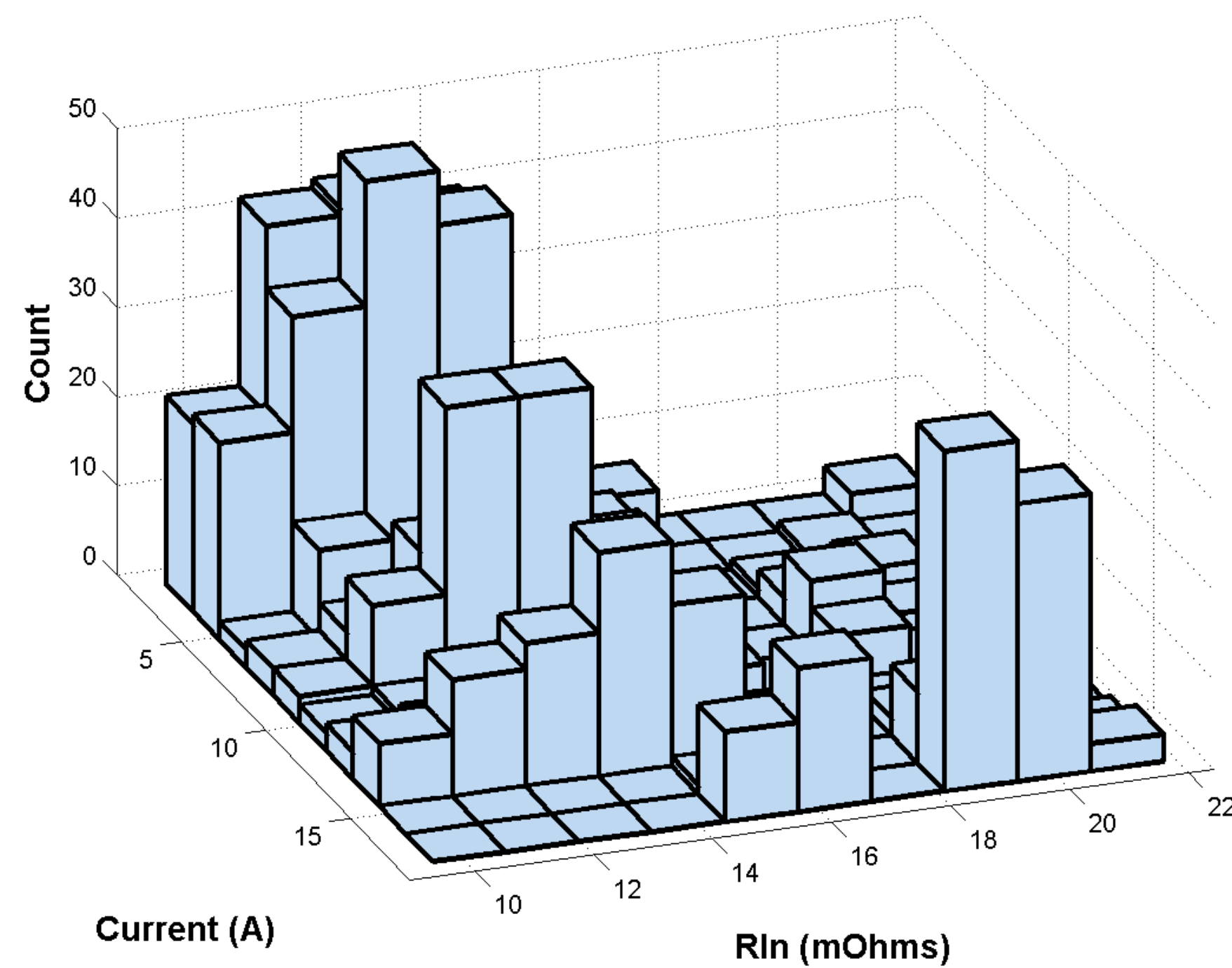
Results



Results



Results



Future work

- Complete the sorting and recovering algorithm explanations in future papers
- Play with every variable of the algorithm to measure their individual effects :
 - Window width of each memory slot in the cluster
 - Weight function against distance
 - Decay function
- Create a 16 bit memory space (with data_H and data_L) to allow count values to go up to 65 k (instead of current 255)
- Test the BMS in a real EV environment (with noise)

Thank you for your attention
Any questions ?

Please come see me at booth #1115
(Université de Sherbrooke) for other questions



UNIVERSITÉ DE
SHERBROOKE

