APPLICATION NOTE

Removing Noise in Lithium-Ion Battery Cell Self-Discharge Data Sets

Manuel Moertelmaier
Keysight Laboratories

Efficient post-processing algorithms enable reliable classification of battery cells.

Introduction

Keysight's self-discharge measurement (SDM) solutions (figure 1) measure and characterize self-discharge performance of Li-Ion cells. Self-discharge currents are recorded across an array of cells, with a typical measurement completing in as little as a few minutes. This compares favorably to the 1-2 weeks required by the widely-used alternative method of open circuit voltage measurements.

In practice, SDM data shows significant noise, complicating interpretation. Here, we present a series of algorithms that effectively remove most of this noise. This strongly improves the reliability of cell classification.

We first discuss how SDM data are supposed to look in an idealized setting, and how they get distorted by noise. We then present alternative noise removal approaches and identify their strengths and weaknesses. We show how noise suppression improves classification and conclude with some recommendations for practical applications.

Self-discharge measurements can be used in a range of settings: In R&D, they can help characterizing cells; in production settings, they can be used to screen defective cells; in second-life applications, they can help estimate remaining cell lifetime.

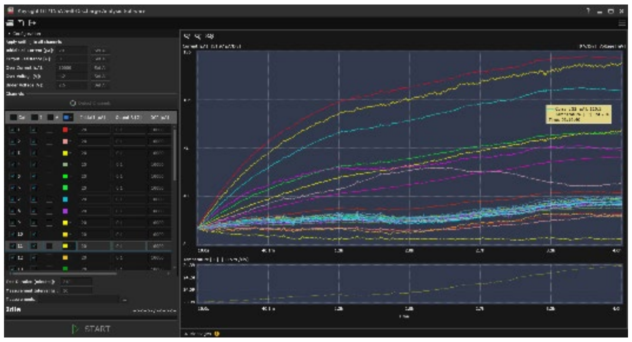


Figure 1 - Keysight solutions to collect and process SDM data:,BT2152B Self-Discharge Analyzer (left) and BT2155A Self-Discharge Analysis Software (right).

What is self-discharge current?

Most Li-Ion cells will gradually discharge even if they're not connected to anything. This loss of stored energy leads to lower-than-desired cell available capacity. And when cells are assembled into multiple-cell battery packs, differing rates of cell self-discharge leads to cell imbalances within the battery.

Typical battery management systems will discharge all the cells to the level of the lowest cell, decreasing effective battery life.

Self-discharge in Li-Ion cells can be modeled as shown in Figure 2.

- C_{eff} is the effective capacitance of the cell, storing the cell's charge.
- R_S is the cell internal or series resistance. R_S causes the cell voltage to drop as you pull more current from the cell, since $V_{cell} = V_{ocv} – (I * R_S)$
- R_{SD} is the parallel resistance through which the self-discharge current flows. When nothing is connected to the cell (open circuit), C_{eff} discharges through the high-value R_{SD}, generating tens or hundreds of μA of self-discharge current (I_d). Over weeks or months, this self-discharge path depletes the stored energy in C_{eff}, thus causing V_{cell} to drop

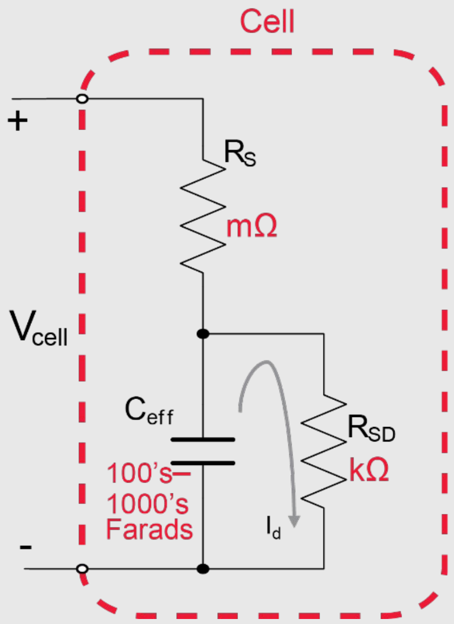


Figure 2 - Simplified model of self-discharge in Li-Ion cell

SDM Data in an Ideal Case

In an idealized setting, SDM curves converge to a stable value after an initial period of equilibration. The shape of the curves can be modeled as an exponential, as illustrated in figure 3. It is important to note that in practice, the time constant of the exponential curve may be longer than the available measurement time. In that case, the shape of the curve may look like a low-degree polynomial or like a straight line.

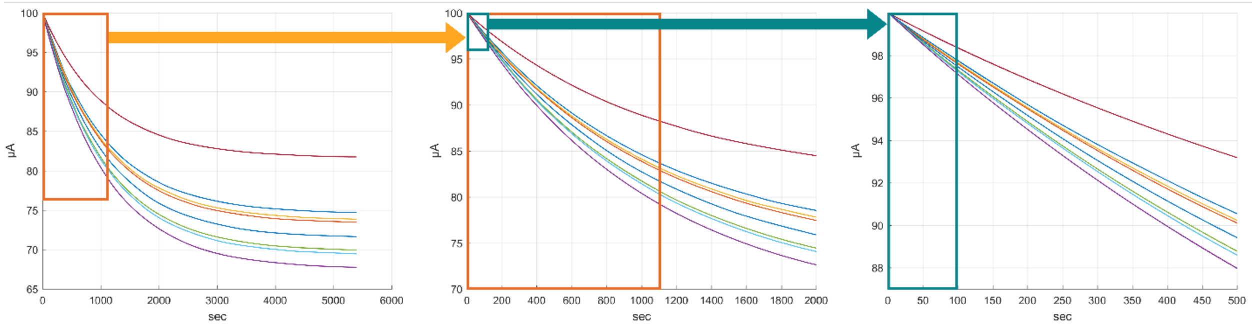


Figure 3 - Idealized shape of SDM curves. We expect, from physics, the shape of SDM curves to be exponential (left). However, in quick measurements we may observe only slight exponential curvature (center) or no curvature (right.) In all three cases, the curves are exponential, but the time scale of the measurement leads our eye to see the shape as linear.

Furthermore, SDM curves can be overlaid on longer-term, approximately linear, trends. These trends of induced current may be due to several factors, such as temperature changes, or charge redistribution, and may be different for each curve, as shown in figure 4.

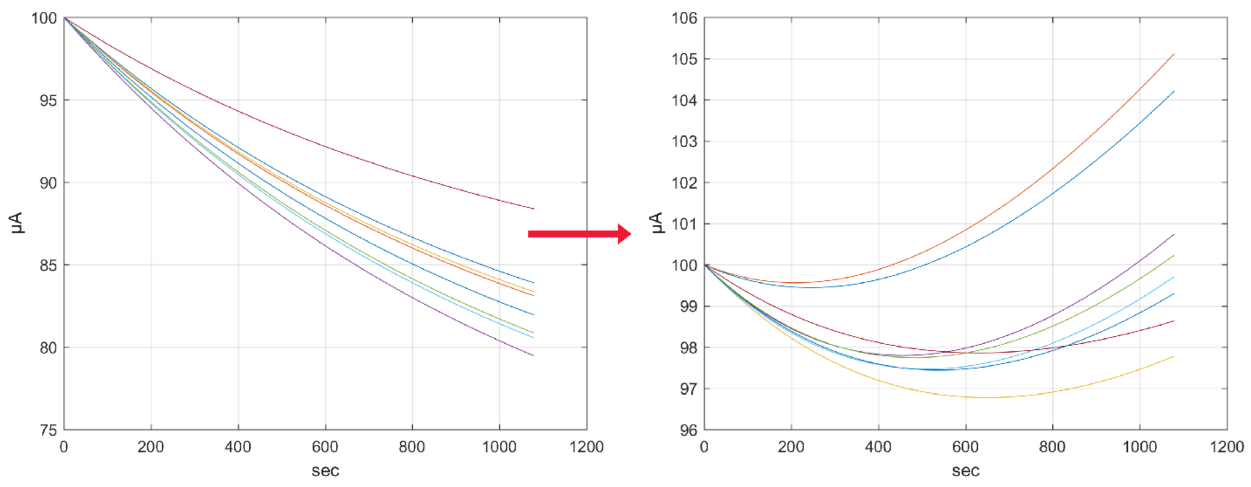


Figure 4 - SDM curves may display both exponential and strong linear components due to temperature change and charge redistribution

Distortion of SDM Data

In real-world settings, data are perturbed by factors within the environment, as well as within the measurement hardware. As mentioned above, the primary factor is temperature-induced cell open circuit voltage changes which translates into a noise component in the SDM current data. In the simplest case, there is one common noise source affecting all channels equally, as shown in figure 5, with electrical noise being added to the noise-free signals.

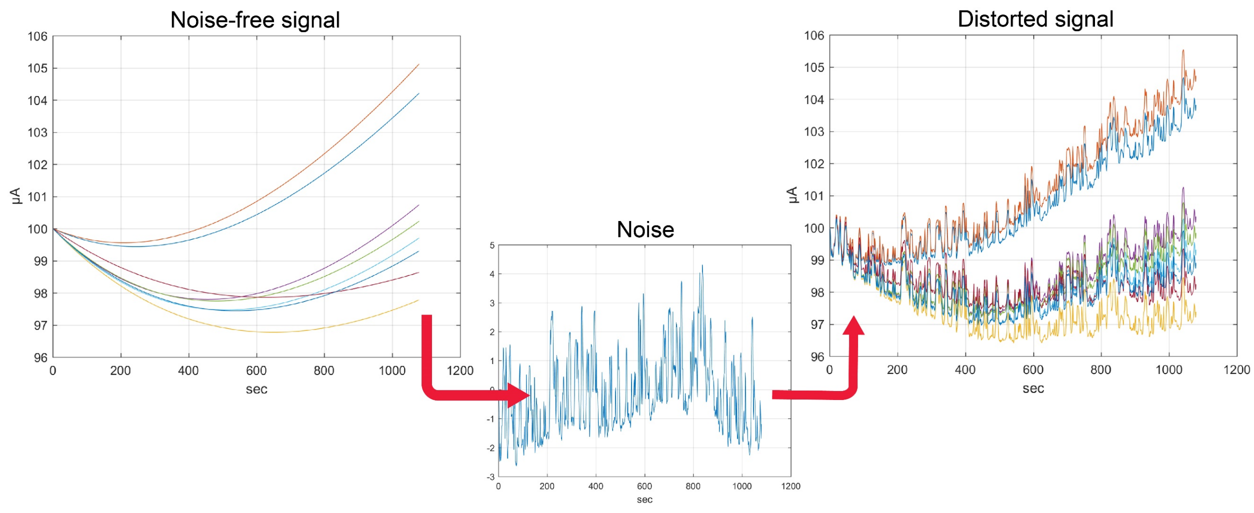


Figure 5 - Distortion of noise-free SDM curves (left) by a common noise source (center), resulting in noisy data (right)

In practice, however, the noise source may affect individual channels to a different degree. Furthermore, there may be more than one noise source present. This may result in a more complicated picture, as shown in figure 6.

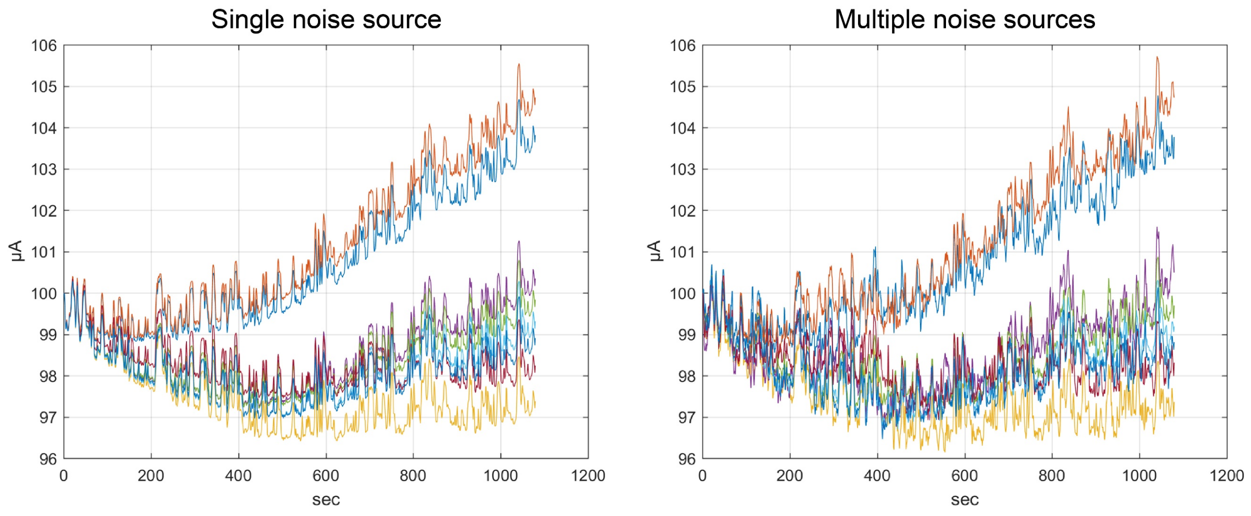


Figure 6 - SDM curves with a single common noise source (left) and with several different noise sources (right)

What Makes for a Good Noise Removal Algorithm?

There are several properties we wish to see in a noise removal algorithm. Not all of them, however, may be achievable at the same time.

- The algorithm should remove all or most of the noise.
- The algorithm should, however, preserve the original information in the data. That is, the algorithm should not introduce systematic distortions. The statistical distribution of the extracted parameters that we need for further decision-making should not display any biases.
- The denoised data should be interpretable. For example, the overall shape of denoised data curves should be visually comparable to that of raw data curves. This allows to use the same scoring criteria in both cases.
- The algorithm should be lightweight in terms of computing power, memory, and lines of code.
- Operation should be incremental, giving preliminary results on incomplete datasets. Results are continuously revised as more data become available. Convergence should be fast and stable.
- There should be no, or few, free parameters in the algorithm. If parameters need to be set, this should be achievable in a dedicated calibration session, using a representative data sample, with no further adjustment necessary afterwards.
- There should be as few constraints as possible regarding the number of data channels, the number of data items per channel, or the quality of the tested cells.

We now describe denoising algorithms based on calculating the median and on principal component analysis. We use the above criteria to discuss their strengths and weaknesses, with Table 1 providing a summary overview.

Noise Removal Based on Median Calculation

If there is a common noise source across channels, we should be able to extract its effects by comparing the individual channels. A simple approach would be to calculate the mean signal at any given time, in the hope that systematic variations of each curve cancel out, while the common noise is preserved. In practice, however, using the median instead of the mean tends to give more interpretable results. This is due to most cells in a given batch being of good quality, showing curves similar to one another.

The median will therefore also be similar to these curves and will be minimally influenced by whatever shape the small number of curves originating from bad cells will have. This allows robust comparison across experiments. However, the assumption that the majority of the cells in a measurement are good quality is critical for interpretability. As the fraction of bad cells approaches 50%, the median may start to fluctuate. In order for the median to be stable, it is also important to have a certain minimum number of parallel recordings available. The exact minimum number depends on noise levels and the fraction of bad cells in the data; 8 channels can be regarded as a general minimum.

The Method of Median Subtraction

The most straightforward use of the median is to subtract it from all signals. This eliminates the noise common to all channels and centers the signals of the majority of the curves, i.e. the “good” curves, around zero.

```
Algorithm MEDIAN_SUBTRACTION
Input: A set of curves of equal length, recorded at a set of time points common for all
        curves.
Output: A set of processed curves, with most curves centered around zero.
Assumptions: Most of the curves are of good quality.
             There is a single noise source affecting all curves roughly the same.
Pseudocode: FOR each time point in input data set
                Take, from each curve, the data item corresponding to that time point.
                Combine all items at that time point into a set.
                Calculate the median of that set.
                Subtract the median from each item in the set.
                Write the processed items in a new set of curves.
            END
                Output the processed curves
```

On the plus side, median subtraction is fast, simple, and robust. It yields results that are easy to interpret, albeit visually distinct from the raw data, which can be a drawback. Median subtraction assumes that noise is identical across channels, and that the majority of cells are of good quality. Otherwise, there are few constraints in terms of measurement settings, and data can be processed on-line, i.e. as it is acquired.

Median subtraction can be effective even if there are several noise sources present, as, generally, one source tends to dominate in influence. Figure 7 demonstrates this effect.

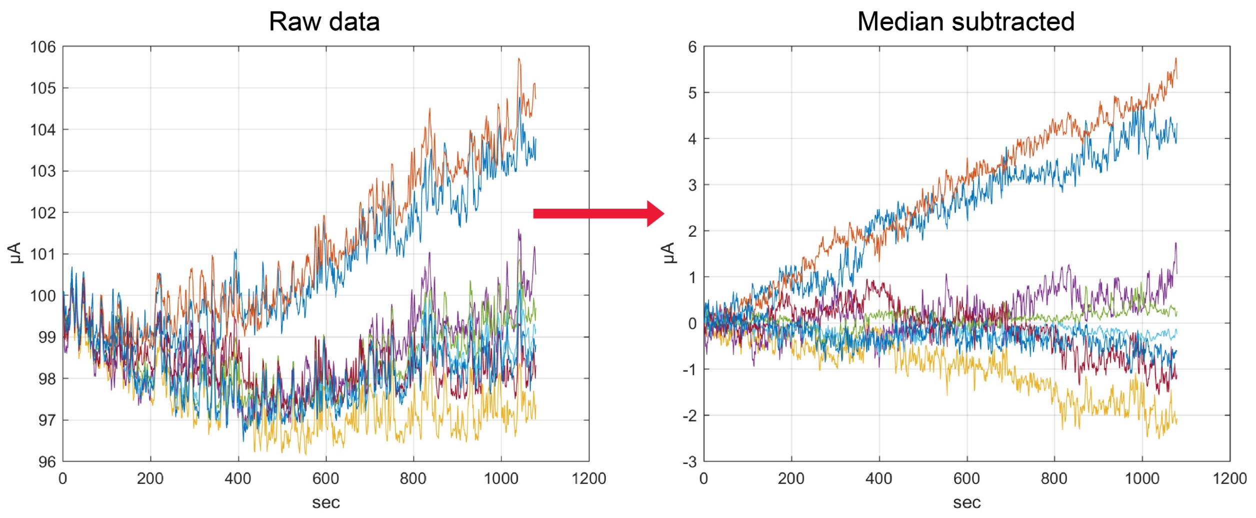


Figure 7 - SDM curves with multiple noise sources, as shown in figure 5, before and after median subtraction.

The Method of Median Fitting

If we want to preserve the overall shape of the curves, we can try to restore them by adding a fit of the median curve back to the data. This effectively discards any high-frequency variation in the median.

```
Algorithm MEDIAN_FITTING
Input: A set of curves of equal length, recorded at a set of time points common for all
        curves.
Output: A set of processed curves, resembling the original curves in overall shape.
Assumptions: Most of the curves are of good quality.
             There is a single noise source affecting all curves roughly the same.
Pseudocode: FOR each time point in input data set
                Take, from each curve, the data item corresponding to that time point.
                Combine all items into a set.
                Calculate the median of that set.
                Subtract the median from each item in the set.
                Write the processed items in a new set of curves.
                Write the median in a separate median curve.
             END
                Fit the median curve with an appropriate function.
                Add fitted median function to processed curves.
                Output resulting curves.
```

Curve fitting of the calculated median can be performed in several ways. If we expect the shape of the curves to be according to theoretical expectations, we can choose exponential, linear or polynomial functions to fit, or use combinations. A more generic solution is to use higher-order polynomial. Alternatively, the median may be processed with a low-pass filter, which is mathematically equivalent with fitting a limited series of sine waves.

The median fitting approach preserves most of the advantages of median subtraction, while making the result more visually similar to the raw data. Figure 8 shows an example. The most important difference to median subtraction is that for the curve-fitting step all of the median data must be available. This means that the method can only be executed once all measurements have been performed. As a compromise, the method can be applied repeatedly on partially complete data. In this case, however, the output may show some fluctuation, especially in early iterations. Another, smaller, drawback is the increased computational load from curve fitting. More important, median fitting can give inappropriate results in situations where the median is strongly deviating from any curve shape available to the fitting algorithm. This is less of a problem if a flexible class, like higher-order polynomials, is used.

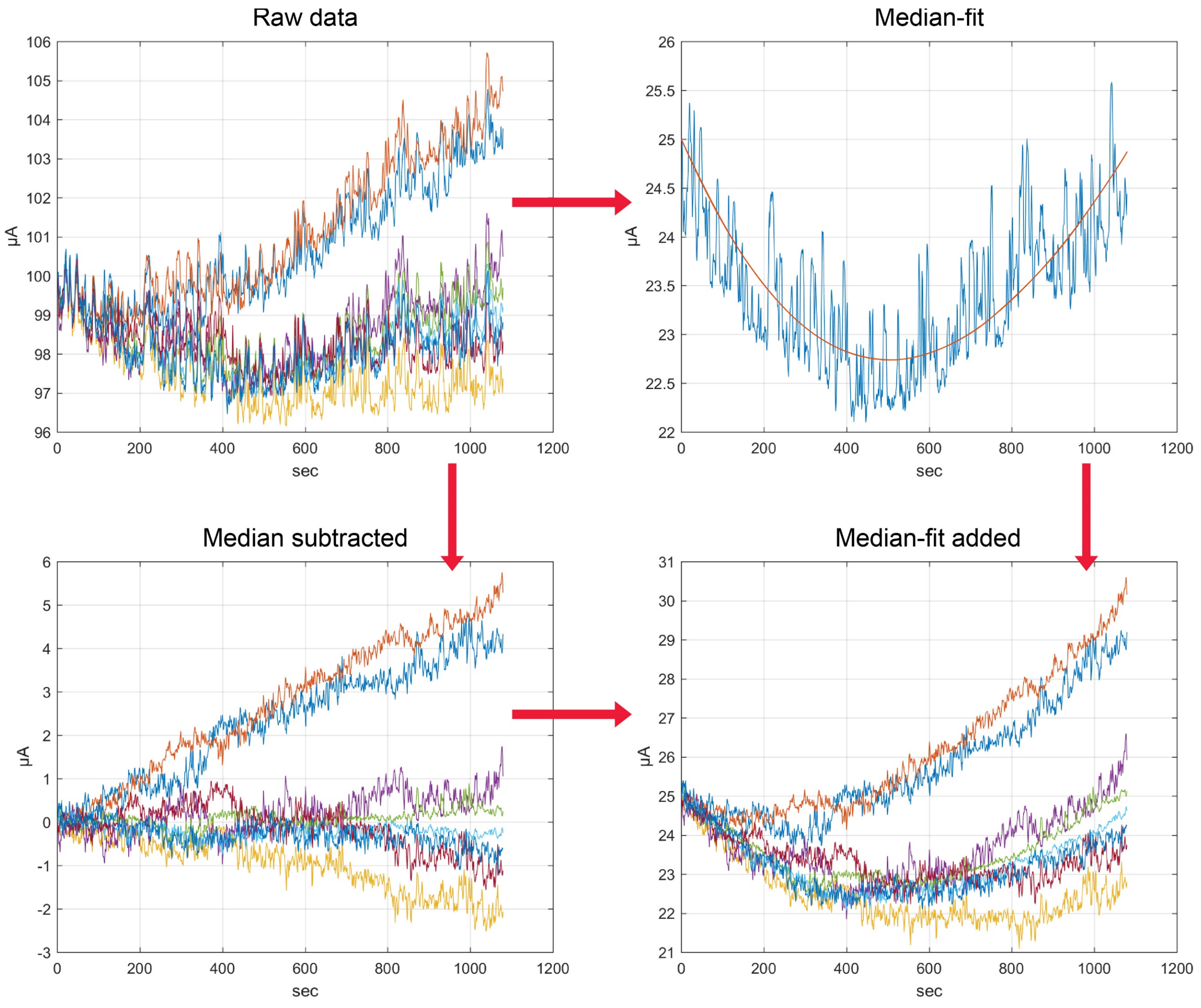


Figure 8- SDM curves, as in figure 6, processed with median fitting. The median, and the curve fitted to it, are shown in the upper right figure. The lower left figure shows the data with the median itself subtracted, and the lower right figure shows the data with the median itself subtracted, and the lower right figure shows the data with the fit of the median added.

Noise Removal Based on Principal Component Analysis

In practice, we may see data that shows distortions by multiple noise sources. In this case, median based methods may not be optimal. As an alternative, we can use methods based on the mathematical principle of principal component analysis (PCA). Here, we extract multiple common factors with variable impact across a set of curves. Those factors can then be selectively suppressed. This allows a greater degree of noise reduction than in median based methods, where only a single factor of variation is extracted.

It is important to point out that PCA requires a preprocessing step. For example, we expect, as noted above, SDM curves to be exponential in overall shape. If we fed PCA a set of such curves without any preprocessing, it would identify the exponential itself as the dominant factor of variation across all curves. In order to make PCA focus on the noise instead, we fit the raw data with appropriate curves – exponential, linear, and/or polynomial – and let PCA work only on the residual. The PCA-processed, noise-reduced residual can then be recombined with the fitted curves to get clean data, as shown in figure 9.

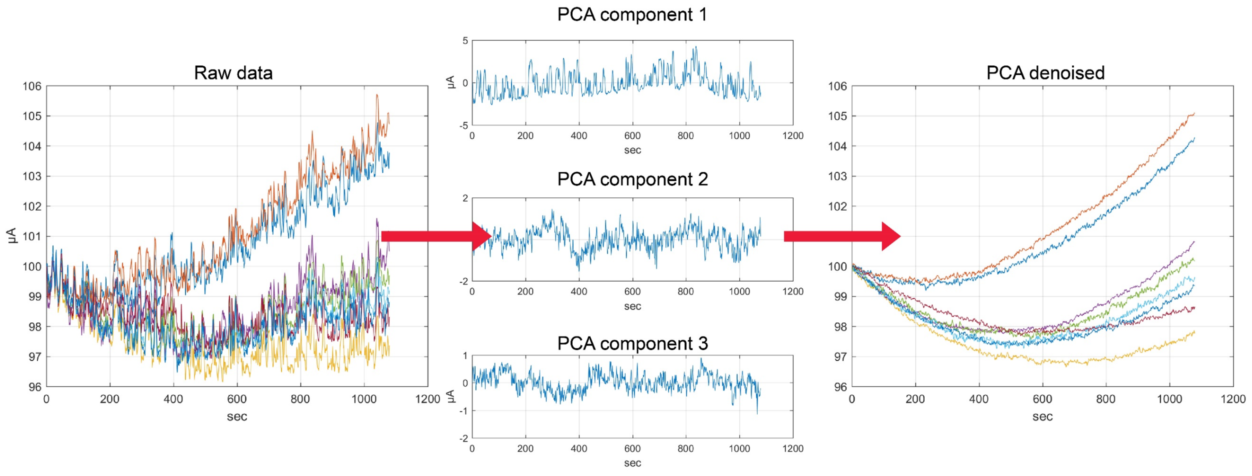


Figure 9 - SDM curves, as in figure 5, processed with PCA. The removed PCA components are shown in the center.

This, however, poses something of a circular problem: The initial curve fit of the unprocessed raw data does not benefit from the noise reduction of the PCA, and may be slightly flawed. An improved, but somewhat complicated solution is therefore to proceed iteratively. The denoised output of the first step is mixed, e.g. 10:90 with raw data, and the result is used for another cycle of curve-fit, PCA, and recombination. The admixture of denoised data is increased over 5-10 additional cycles. This allows the algorithm to find a self-consistent solution that optimally combines curve fitting and PCA.

This denoising approach is more complicated than median-based approaches, and the pseudocode below is best understood in reference to figure 10.

```
Algorithm PCA_DENOISE
Input: A set of curves of equal length, recorded at a set of time points common for all
  curves.
Output: A set of processed curves.
Assumptions: The curves have shapes which we can describe as mathematical functions:
             Exponential, linear, polynomial, etc.

             The number of data channels is significantly larger than the number of noise
             sources.
Pseudocode:
             INITIALIZE fit-curves as input curves.
             FOR several iterations
                 FOR each fit-curve
                     Fit the fit-curve with an appropriate function.
                     Subtract the fitted function from the fit-curve.
                     Store the residual.
                     Store the fitted function.
                 END
                 Perform PCA on the residuals.
                 Eliminate or attenuate 1-3 of the largest PCA components.
                 Transform back to generate denoised residuals.

                 Combine the stored fitted functions with their denoised residuals to
                          generate denoised curves.

                 Blend the denoised curves with the input curves to generate new fit-curves,
                          with more weight given to the denoised curves in each iteration.
             END
             Output the denoised curves
```

Special care must be taken in situations where we expect the noise to take the shape of linear trends superimposed on each curve, with the slope of the trend different for each curve.

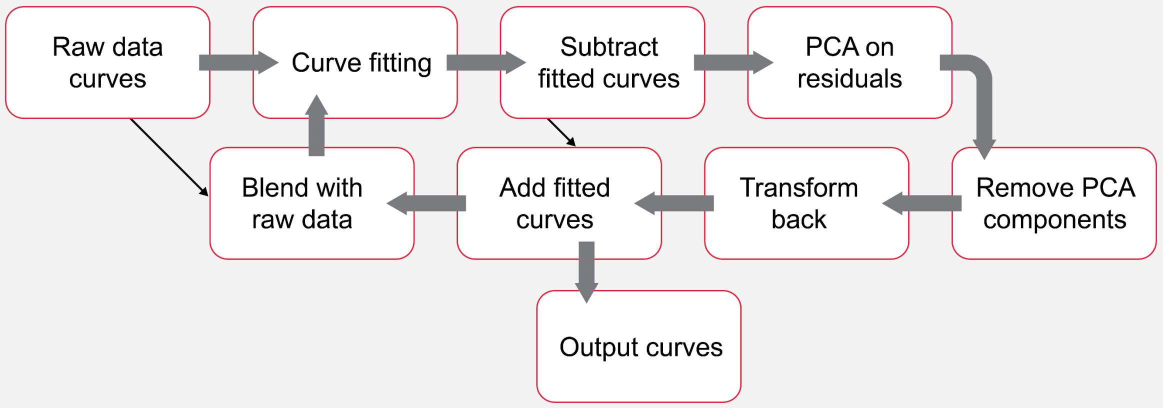


Figure 10 - Workflow of a denoising algorithm based on PCA

If we can reliably assume the underlying shape of the SDM curves to be exponential, we can use this as a fit, and rely on PCA to isolate the different linear trends impacting each curve as noise. When the SDM signals themselves have linear components, and we are interested in the true value of those components, it may become very difficult to disentangle them from linear trends due to noise. PCA cannot be relied on to identify which parts of the linear trend in each curve are due to signal and which are due to noise. Instead, it is recommended to extract the overall linear component in each curve with a fit, and allow PCA to work on the remaining noise. Once the linear trends are extracted, their noise and signal parts can potentially be separated with custom methods based on expert judgement, which, however, goes beyond the scope of this paper.

It is generally better to mitigate the actual causes of those sorts of trends rather than relying on PCA to remove them. For instance, if a trend in the data is present due to cell temperature changes causing changes in cell voltage (which can cause changes or noise in the measured self-discharge current), it is better to control the temperature of the environment in which the cells sit, especially for cells with a large temperature coefficient of voltage (TCV). In a similar manner, if a trend in the data is present due to charge redistribution within the cell, it is likely better to allow the cells to settle sufficiently to mitigate this effect.

PCA based approaches are relatively complicated, and computationally demanding, and are therefore best applied in situations where the noise level is high enough to test the limits of median based methods. PCA assumes that the number of channels is significantly larger than the number noise sources. It also assumes the number of measurement points per channel to be larger than the number of channels, a condition that is, in practice, typical. Finally, it requires the overall shape of the SDM curves to be realistically approximated by some known class, such as linear, exponential, polynomial, or combinations of those.

If all these conditions are met, PCA based methods can be highly effective, and can strongly outperform median based ones, especially in high-noise situations. However, PCA requires some care in its application. If curve-fitting is based on a wrong assumption about the underlying SDM curve shape, e.g. using linear fits when the data is really exponential, the method can introduce bias. PCA may falsely identify the difference between expected fit and reality as noise, and remove it, yielding a flawed, but apparently low-noise output. It is therefore important to check assumptions, perform tests of the method, repeatedly check the output, as well as the removed components, and exercise common sense in the judgement of results. Due to these risks, it is recommended to initially limit the use of PCA to R&D or process engineering settings, where human supervision is possible, while demands on noise removal are highest. Then, after experience with PCA and SDM is built up, and after this measurement process is characterized, it is possible to deploy PCA in an automated way in manufacturing.

Noise Reduction Improves Cell Classification

The usefulness of noise reduction is self-evident. In an R&D setting, denoised curves are a valuable input for further analysis. In a manufacturing environment, the primary motivation is accurate classification of good vs. bad cells. Effective noise suppression can indeed improve classification, as is shown in figure 11. In the noisy data, curves are hard to disentangle, or classify, whereas denoising leaves them separable, with the PCA-based method yielding a version of the curves that is more faithful to the original data than median-fitting. If we were to sort curves based on their last, i.e. rightmost data points, we would find that PCA completely restored the ordering of the original data, whereas noisy data is mostly unintelligible, with most of the order scrambled.

In practice, classification of cell quality will be based on multiple points of the curve. This can be achieved, for example, by fitting the curves, and using the fit parameters as input to a classifier. Denoising can improve the quality of such fits, and methods like PCA already create such fits as a natural by-product of their operation. Using such fits is a way to make use of all the data in the curve, yielding a more robust classification.

Comparison of our Methods

It is time to look back at the methods presented in the previous sections, and to summarize their strengths and weaknesses. Table 1 provides an overview of the "rating" of the algorithms with respect to the criteria listed on page 4. We find that no single algorithm outperforms all others in all categories. Instead, they all show different profiles, and should be used in the setting that best matches them.

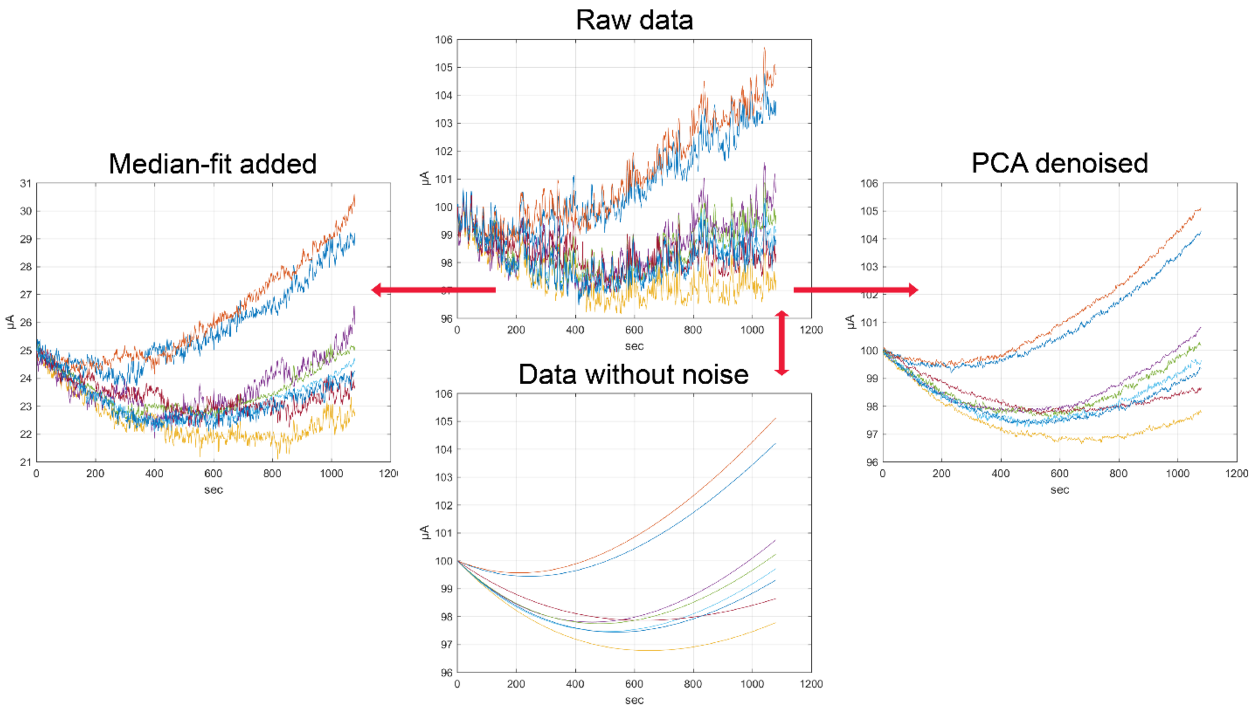


Figure 11 - We can use simulated raw data (center, top) derived from a known noise-free signal (center, bottom) to demonstrate the effect of median-based denoising (left) and PCA-based methods (right).

TABLE 1: PROFILE OF DIFFERENT NOISE REMOVAL METHODS

	Median subtraction	Median fit	PCA based
Effective noise removal	★★	★★	★★★
No bias, distortions	★★	★★	★★
Interpretable	★	★★★	★★★
Lightweight	★★★	★★	★
Incremental	★★★	★	★
Few free parameters	★★★	★★	★
Few constraints	★★	★★	★★

Summary and Recommendations

As we have shown in this paper, denoising algorithms can be helpful in the analysis of SDM data, especially in the presence of high noise levels. However, denoising methods have their limitations, and carry a cost in terms of resource demands and extra complexity introduced to the workflow. They should therefore never be a substitute for good measurement practices.

We therefore recommend a conservative approach to post-processing. That is, the primary goal should still be to obtain the best raw signal as possible. Then, different denoising methods should be tested. Among those that yield an acceptable final output signal, the most lightweight and simple should generally be picked.

In practice, there will be widely different constraints in terms of required signal quality, data availability, and computational resources, depending on whether the denoising should take place in a laboratory or on a manufacturing line.

Median subtraction, possibly with the fit of the median added back in, is a fast and robust method that often yields good results. It is the recommended method for fast classification and is especially useful if a single noise source is dominant.

PCA-based denoising can work with very high noise levels and can remove noise sources originating from multiple strong sources. It is, however, more complex, and requires care in the choice of an appropriate class of functions to fit to the data. It is well suited for a laboratory setting, where careful testing and human supervision are possible, at least when used on new data sets. After proper testing, it can also be deployed in automated or “hands-off” situations such as cell manufacturing.

Finally, in some cases, all denoising methods face intrinsic limits. Consider the scenario where our signal of interest is linear, and different in each channel. Due to slow temperature drift, and different TCVs for each cell, each channel also has a different linear drift signal overlaid. This situation is impossible to resolve mathematically, as we don’t know which fraction of each linear slope comes from signal and which from drift.

To repeat: Denoising algorithms, while being able to strongly improve signal quality, should always be applied in the context of good measurement “hygiene”, with maximum care, and under the right conditions.

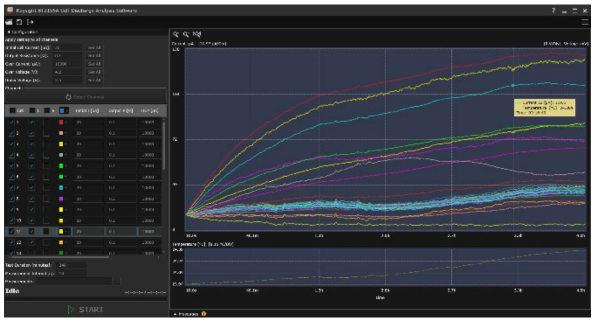
For more information on Li-Ion cell self-discharge and Keysight's Self-Discharge Measurement Solutions

Visit www.keysight.com/find/Self-Discharge

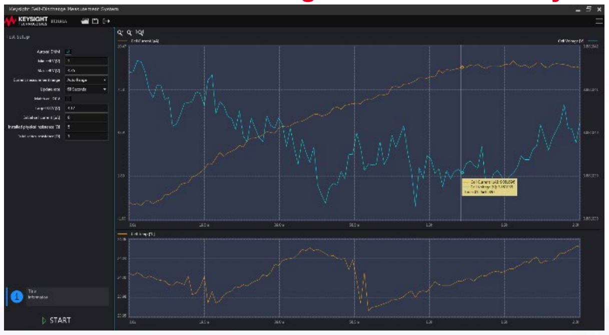
Application Note: Evaluate Lithium Ion Self-Discharge of Cells in a Fraction of the Time Traditionally Required (5992-2517EN), https://literature.cdn.keysight.com/litweb/pdf/5992-2517EN.pdf?id=2911018

BT2152B Self-Discharge Analyzer, www.keysight.com/find/BT2152

BT2155A Self-Discharge Analysis Software, www.keysight.com/find/BT2155



BT2191A Self-Discharge Measurement System, www.keysight.com/find/BT2191



Learn more at: www.keysight.com

For more information on Keysight Technologies' products, applications or services, please contact your local Keysight office. The complete list is available at: www.keysight.com/find/contactus