

2020

## ZeChipC: Time Series Interpolation Method Based on Lebesgue Sampling

Luis Miralles-Pechuán  
*University College Dublin, Ireland*

Muhammad Atif Qureshi  
*Technological University Dublin, atif.qureshi@tudublin.ie*

Matthieu Bellucci  
*University College Dublin, Ireland*

*See next page for additional authors*

Follow this and additional works at: <https://arrow.tudublin.ie/buschmarbk>



Part of the [Business Intelligence Commons](#), and the [Computational Engineering Commons](#)

---

### Recommended Citation

Miralles-Pechuán L., Bellucci M., Qureshi M.A., Namee B.M. (2020) ZeChipC: Time Series Interpolation Method Based on Lebesgue Sampling. In: Martínez-Villaseñor L., Herrera-Alcántara O., Ponce H., Castro-Espinoza F.A. (eds) *Advances in Soft Computing. MICAI 2020. Lecture Notes in Computer Science*, vol 12468. Springer, Cham. [https://doi.org/10.1007/978-3-030-60884-2\\_14](https://doi.org/10.1007/978-3-030-60884-2_14)

This Conference Paper is brought to you for free and open access by the School of Marketing and Entrepreneurship at ARROW@TU Dublin. It has been accepted for inclusion in Books/Book Chapters by an authorized administrator of ARROW@TU Dublin. For more information, please contact [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [aisling.coyne@tudublin.ie](mailto:aisling.coyne@tudublin.ie), [vera.kilshaw@tudublin.ie](mailto:vera.kilshaw@tudublin.ie).



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 International License](#).  
Funder: Enterprise Ireland

---

**Authors**

Luis Miralles-Pechuán, Muhammad Atif Qureshi, Matthieu Bellucci, and Brian Mac Namee

# ZeChipC: Time Series Interpolation Method based on Lebesgue sampling

Luis Miralles-Pechuán<sup>1</sup>[0000–0002–7565–6894], Matthieu Bellucci<sup>1</sup>, M. Atif Qureshi <sup>\*2</sup>[0000–0003–4413–4476], and Brian Mac Namee<sup>1</sup>[0000–0003–2518–0274]

<sup>1</sup> Ireland’s Centre for Applied AI (CeADAR), University College Dublin, Dublin, Ireland [luis.miralles@ucd.ie](mailto:luis.miralles@ucd.ie), [matthieu.bellucci@gmail.com](mailto:matthieu.bellucci@gmail.com),

[brian.macnamee@ucd.ie](mailto:brian.macnamee@ucd.ie)

<sup>2</sup> Ireland’s Centre for Applied AI (CeADAR), Technological University Dublin, Dublin, Ireland.

[MuhammadAtif.Qureshi@tudublin.ie](mailto:MohammadAtif.Qureshi@tudublin.ie)

**Abstract.** In this paper, we present an interpolation method based on Lebesgue sampling that could help to develop systems based time series more efficiently. Our methods can transmit times series, frequently used in health monitoring, with the same level of accuracy but using much fewer data. Our method is based in Lebesgue sampling, which collects information depending on the values of the signal (e.g. the signal output is sampled when it crosses specific limits). Lebesgue sampling contains additional information about the shape of the signal in-between two sampled points. Using this information would allow generating an interpolated signal closer to the original one. In our contribution, we propose a novel time-series interpolation method designed explicitly for Lebesgue sampling called ZeChipC. ZeChipC is a combination of Zero-order hold and Piecewise Cubic Hermite Interpolating Polynomial(PCHIP) interpolation. ZeChipC includes new functionality to adapt the reconstructed signal to concave/convex regions. The proposed methods have been compared with state-of-the-art interpolation methods using Lebesgue sampling and have offered higher average performance.

**Keywords:** Lebesgue sampling interpolation method · Event-based interpolation · signal reconstruction using Lebesgue sampling · time series interpolation method

## 1 Introduction

Nowadays, a lot of time series data is produced, which represents the state of the environment over a period of time [5, 10]. These data points are generally captured by a piece of equipment called a sensor. The sensor can detect different events or changes in the environment and quantify the changes in the form of various measures such as temperature, pressure, noise, or light intensity. A limitation of collecting data points is the frequency at which the sensor records

---

\* Senior Author and provided level of mentorship.

the changes or events. The more frequently a sensor records a reading, the more expensive the running cost is. Likewise, the less frequent the sensor records the reading, the more difficult it is to capture and reconstruct the original behaviour of the event.

In practice, all signals have to be sampled because the number of points in a continuous environment is infinite. Capturing readings more often is economically more expensive due to the amount of data being stored, transmitted, and processed. The challenge while performing sampling is to preserve the vital information in the less amount of data points so that the objective of recording changes is met.

The periodic or Riemann sampling [5] is a conventional approach of sampling in the time series data. In this approach, the data is captured periodically, i.e. at an equidistant time intervals such as each second or each microsecond. Even though the approach is simple to implement, the shortcoming is that, when the sampled data fail to indicate changes that happen between the interval (also known as frequency aliasing), sampling needs to be readjusted at a higher frequency, resulting into more data collection. Due to this pitfall, many research findings advocated for the use of Lebesgue sampling, instead of Riemann sampling [7].

In the Lebesgue sampling, time-series data is sampled whenever a significant change takes place or when the measurement passes a certain limit [6]. The overall intuition of the Lebesgue sampling is to save unnecessary data from being stored, processed, or transmitted, which represents either no change or a trivial change compared to the previous data point.

In this contribution, it is proposed an interpolation method called ZeChipC to reconstruct the time series data sampled using Lebesgue sampling. ZeChipC uses **Z**ero-order hold and **P**Chip interpolation with a **C**oncavity/**C**onvexity functionality improvement<sup>3</sup>. To the best of our knowledge, this is the first interpolation method designed exclusively for Lebesgue sampling. The proposed methods have a higher performance because they exploit the particular properties of this kind of sampling.

The rest of the paper is organised as follows. In section 2, it has been elaborated a review of the state-of-the-art about the Lebesgue sampling technique. In section 3, it is presented the proposed interpolation method called ZeChipC, along with its simpler versions (ZeLi, ZeLiC and ZeChip). In section 4, two experiments using 67 different data sets are carried out. The objective of the first experiment is to compare the performance of the proposed methods against that of the state-of-the-art interpolation methods for Lebesgue sampling. The objective of the second experiment is used to compare the performance of Lebesgue and Riemann sampling with approximately the same number of samples and with the same methods. This is very useful in order to decide which is the best combination of sampling and interpolation methods when time series need to be

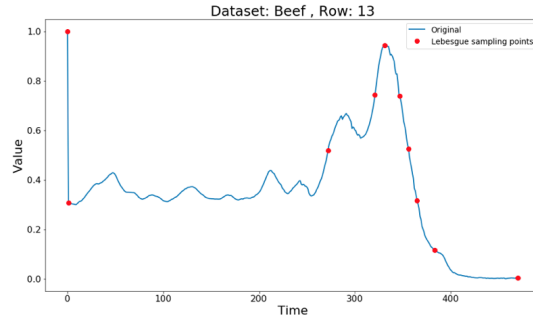
---

<sup>3</sup> The implementation code for the proposed algorithms can be found in: <https://github.com/shamrodia74/ZeLiC>

sampled. Finally, in section 5, the conclusions of the research are shown along with some possible future directions.

## 2 State of the art

Lebesgue sampling is an alternative to the traditional approach called Riemann, which samples time series at a constant frequency. Instead of periodically taking samples from a system like most health monitoring systems, the event-based method takes samples only when a predefined event happens, as shown in Figure 1. Some examples of typical events could be a sudden change in the signal, the signal reaching a preset limit, the arrival of a data package, or a change in the state of a system [1]. Even though Lebesgue sampling is more accurate than Riemann sampling, it is less extended because it is more difficult to implement [1]. Sampling using methods based on Lebesgue will allow developing systems with many advantages such as lower energy consumption and less bandwidth required. In systems based on Lebesgue less information needs to be processed, transmitted and stored.



**Fig. 1.** Lebesgue sampling takes points based on the output value of the signal. In this particular case, when the absolute difference is higher than 0.2.

In recent years, a great interest has aroused in applications implementing event-based sampling. For example, the “Send on Delta” algorithm takes advantage of Lebesgue sampling to reduce the information transmitted by wireless networks, frequently used in health monitoring systems, in order to increase the lifetime of the sensors’ batteries. Under this scheme, the sampling is performed only when there is a deviation in the signal higher than the delta value. Results show that by using this approach, it is possible to increase the lifetime of the sensors without any loss of quality in the resolution of the signals [8].

It is also interesting pointing out the convenience of using event-based sampling in the Fault Diagnosis and Prognosis (FDP) which are widely applied in the daily care of elderly (e.g. fall detection or human activity detection). In the

last years, it has been increasingly difficult to manage microcontrollers and embedded systems due to the volume of the information collected by sensors and to the complexity of the programs that they implement [13].

Lebesgue sampling can minimise energy consumption, storage space, computation time and the amount of data to be transmitted which turns out into enormous advantages for the development of systems that require to collect time-series related to health attributes [12]. It is therefore very interesting for companies who are in charge of developing health systems, as it significantly reduces the expense, without a negative impact on the precision of the measures.

In summary, the traditional approach for sampling and digital control has been working well in many applications in the health domain for many years. However, there are new domains where Riemann sampling has significant problems that can be easily solved by implementing Lebesgue sampling.

### 3 ZeChipC Lebesgue sampling interpolation method

In this section, first, we give some information that can be extracted from sampling based on events to develop methods that either need less sampled points to achieve a similar accuracy level than the state-of-the-art methods or that given the same amount of points, achieve higher accuracy. For the sake of the reading understanding, in the following lines, we will explain all the previously developed methods until we finally developed the ZeChipC algorithm. Therefore, we first describe in detail the development of ZeLi, our first and simplest interpolation method proposed for Lebesgue sampling. Following, we describe ZeLiC, which is an improved version of ZeLi with new functionality to adapt it to convex and concave regions. And lastly, we propose a method called ZeChipC, which is basically an adaptation of ZeLiC that uses PCHIP instead of Linear interpolation so that it can represent signals with curve regions.

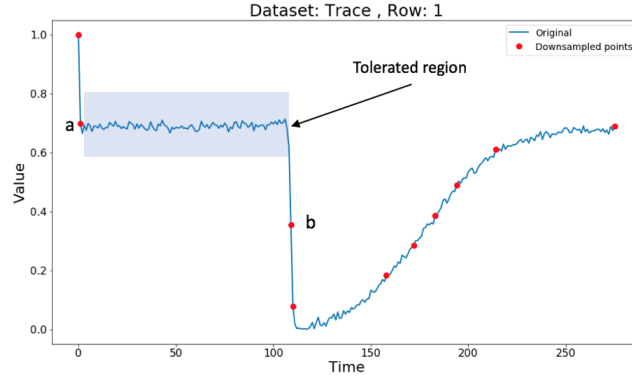
#### 3.1 Tolerated region

In order to understand how ZeLi (our first method) works, we have to introduce a new concept called tolerated region and why is it important for ZeLi. There are many possible implementations of Lebesgue sampling for time series such as sampling a point when it crosses a preset limit or when the percentage variation is higher than a given threshold. Our particular implementation of Lebesgue sampling is based on the variation of the signal. In other words, when the sensor detects that a point has a difference from the previous sampled point greater than a given threshold, then the point is captured. We can express this same idea in mathematical terms in the following way. Let  $y_i$  be the first sample of a signal and  $t$  the threshold, where  $t > 0$ . Then, the sensor captures the next point called  $y_{i+1}$  if and only if  $|y_{i+1} - y_i| \geq t$ .

Lebesgue sampling indirectly gives information on the behaviour of the signal between the two consecutive samples. It can be deduced that all the points between two pair of consecutive sampled points (not necessarily consecutive in

time intervals) are inside an interval delimited by the threshold, as shown in Figure 2. We know this because if the value of a point of the interval was out of the interval, this point would have been captured. Just remember that points are not sampled while the signal stays inside this allowed region. The *tolerated region* can be defined as  $[y_i - t, y_i + t]$ , where  $t$  (threshold) is the maximum allowed value of change.

As shown in Figure 2, if the difference between two consecutive sampled points  $a$  and  $b$  of a time series is very large in a given interval, we know that an abrupt change has happened (otherwise the point would have been collected earlier). On the other hand, if the difference between the sampled points is very small, it is quite probable that a smooth change has taken place. This simple principle is the basis on which our interpolation algorithms for Lebesgue sampling (ZeLi, ZeLiC, ZeChip, and ZeChipC) have been developed.



**Fig. 2.** The point  $b$  is very far away from the tolerated region of the point  $a$ , so we can deduce that an abrupt change has taken place.

In other words, any  $y$  in the interval  $I$  between the two points is inside the tolerated region. We can therefore significantly reduced the region of the possible values of  $y$  when performing the interpolation and therefore reducing the error when comparing the original signal with the reconstructed one.

### 3.2 ZeLi interpolation algorithm

From the information that can be extracted from the tolerance region, we will develop a set of methods to interpolate time series sampled with the Lebesgue approach. The simplest method and the first that is going to be explained is called ZeLi. The rest of the methods are improvements with respect to this first method.

**Combination of Zero-order hold and Linear interpolation** ZeLi interpolation combines **Z**ero-order hold interpolation and **L**inear interpolation, which explains the origin of its name to reconstruct the original signal from the sampled signal as shown in Figure 3. To decide whether to apply Zero-order hold (ZOH) or Linear interpolation, the *tolerance ratio* parameter is used. The *tolerance ratio* is a constant value that multiplies the interval of the tolerated region, that is:  $[(y_i - t) * tolerance\ ratio, (y_i + t) * tolerance\ ratio]$ , creating a new interval called increased tolerated region.

Therefore, the ZeLi algorithm contemplates two possible cases:

1. If the examined point is outside of the increased tolerance region, then ZOH interpolation is used.
2. Otherwise, Linear interpolation is used.

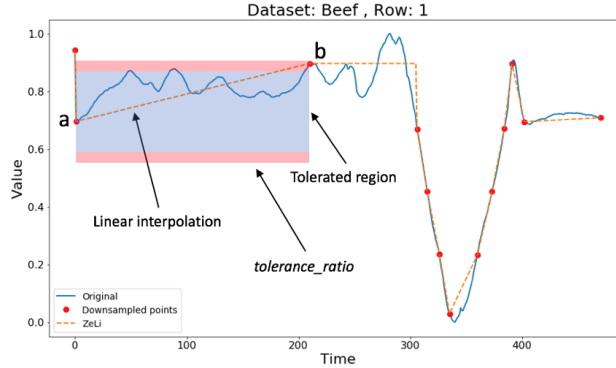
The justification of the algorithm is as follows. We know that all the points between two captured points should be in the tolerated region. If the difference between the values of the points is small, it is quite possible that between the two sampled points the signal follows a linear trend with small variations around it, therefore Linear interpolation is used. On the other hand, to minimise the error when the difference between the two sample points is very large (an abrupt change is presented), we interpolate all the points using ZOH. Although ZOH interpolation does not represent continuous signals in a smooth way, it is a very effective mechanism since it satisfies that all the values of the interval between a given point and the next point [a,b) are in the tolerated region. ZOH interpolation minimises the error because it keeps all the values in the middle of the tolerated region  $R_i$ , dividing by two the maximal possible error.

A visual intuition of this method can be seen in Figure 3. When the signal crosses the threshold in a continuous way, Linear interpolation would be used. This is because it can be assumed that the values of the previous points close to  $b$  will have similar values (obviously, the further those points are from  $b$ , the less probable this assumption would be). By contrast, if an abrupt change is presented, then the last but one point and all the previous points were somewhere in the limited region  $R_i$ . However, we cannot deduce a trend because the change is so abrupt. Due to this lack of information, we choose to minimise the maximal error by using ZOH interpolation. Since ZOH interpolation uses a horizontal line (same values for y-axis for all the points of the region) followed by a straight line to interpolate.

### 3.3 ZeLiC interpolation method

One of the disadvantages of using simple methods such as linear interpolation and ZOH is that they are not able to adapt smoothly to the signal when there is a change in the sign of the slope. To sort this out, we have developed new functionality to improve the performance of ZeLi when convex/concave regions are presented.





**Fig. 3.** ZeLi combines ZOH with Lineal interpolation. Between  $a$  and  $b$  ZeLi applies Line interpolation.

**Convex/Concave regions on time series** The shape of the signal when a slope change is presented is generally a convex or concave region. When Lebesgue sampling is used, a lack of information about the shape of the signal is presented when the signal changes the sign of the slope between two sampled points  $x_i$  and  $x_{i+1}$ , and that change takes place inside the tolerance region (i.e. before the signal hits the threshold). Additionally, neither Linear nor ZOH are able to adapt to convex or concave regions of a signal since those two methods connect the pairs of points individually.

In order to improve the adaptation of the proposed methods, we have studied the case of convexity and concavity, which are the inflexion points of a function. A function is convex if the line segment between any two points on the graph of the function lies above or on the graph. If we assume that the function  $f$  follows the trend of the sampled points, that is to say, if the values of  $f$  for a particular region are decreasing and, at some point, they start increasing, then, we can assume that  $f$  is convex on the interval  $I_i$ . At this point, it is important to remind that this is based on assumptions, we do not have enough information to support the assumptions, except the values and position of the sampled points.

### 3.4 ZeChip interpolation algorithm

ZeLi is based on a combination of ZOH and Linear interpolation, and in consequence, it shares some limitations of those type of interpolation. ZeLi is able to approximate time series with a high precision when they are composed of straight lines, that is to say, signals for which their first and second derivative present very constant values. However, when the signal function has curved lines, it is not possible to represent it using ZeLi.

We can apply the same idea of combining two interpolation methods as in ZeLi but replacing Linear interpolation with PCHIP interpolation. This new method is called ZeChip and is able to adapt much better to those signals that

present curves regions. In addition, the new method will include the advantages of PCHIP; a fast and powerful interpolation method that allows to represent regions using curved lines and obtaining a great precision. One of the shortcomings of ZeChip with respect to ZeLi is that, because it uses PCHIP interpolation instead of Linear interpolation, it has a higher complexity cost. Therefore, for the same given points, ZeChip will take more time to generate the interpolated signal than ZeLi.

## 4 Experiments

So far, we have discussed our proposed methods from a theoretical perspective. Still, to have certain evidence that our contribution can have a meaningful impact, we need to demonstrate that the proposed methods have a better general performance than that of the state-of-the-art. To this end, we have decided to perform the experiments using a large number of databases. We want to test the performance of our models against other interpolation methods under Lebesgue sampling. Besides that, we want to compare the performance of our method with Lebesgue sampling with the performance of other interpolating methods with Riemann sampling with a similar number of samples, so we can recommend that approach when time series need to be sampled.

### 4.1 Preparation of the experiments

To perform the experiments, we followed the methodology explained in Figure 4. First, we downsampled the original time series (using Lebesgue or Riemann sampling), then we reconstructed the original signals from the downsampled data (using Linear, PCHIP, ZOH... interpolation methods) and finally, we compared the original signal with the reconstructed one, using RMSE, to evaluate the performance of the different interpolation methods.



**Fig. 4.** The methodology to calculate the best interpolation method is based on the RMSE between the original signal and the reconstructed one.

There are many metrics in the state-of-the-art to calculate the difference between the original signal and the reconstructed signal. In this research, we applied a prevalent metric called root-mean-square-error (RMSE). The RMSE has been used in many research works to calculate the efficiency of the interpolation techniques [14, 9]. To this end, all the signals of all the datasets have been individually normalised between "0" and "1".

In order to conduct the experiments, we applied some of the most popular time series interpolation methods in state of the art such as ZOH, Linear interpolation, PCHIP, Shannon, Lasso, Natural Neighbour, Cubic, Multiquadric, Inverse Multiquadric, Gaussian, Quintic and Thin-Plate. These functions have been implemented using the Radial Basis Function (RBF) approximation/interpolation in python based on the books [4] and [11].

We have applied two different approaches strategies to downsample the signals:

- Lebesgue sampling: Our implementation of Lebesgue sampling is based on the absolute difference between the sampled values.
- Riemann sampling: The Riemann sampling is performed by using the same (or slightly higher) average number of points than in Lebesgue sampling but using a fix time interval over time.

To perform the experiments, all the signals of all the datasets have been interpolated between "0" and "1". The values of the parameters for the developed methods (ZeLi, ZeLiC, ZeChip, and ZeChipC) were: *tolerance ratio = 1.15*, *min distance=3*, and *previous distance=3*.

The objective of the first experiment is to evaluate the performance of our proposed method for interpolating time series from Lebesgue sampling. To this end, we compared our methods with those of the state-of-the-art. In this experiment, we applied Lebesgue sampling based on the difference between the values with a threshold of 0.05<sup>4</sup>.

The goal of the second experiment is a bit more ambitious than the one of the first experiment. We want to demonstrate that the best technique to interpolate and reconstruct any signal is to use Lebesgue sampling and our best-proposed method; ZeChipC. To this end, we will conduct a similar experiment to the first one, but this time with the same number of samples for both Lebesgue sampling and Riemann sampling. We will select 15% of the total samples of the signal for both Riemann and Lebesgue sampling. In the case of Lebesgue, we will tune the value of the threshold until selecting the same or slightly less (never more) number of samples than in Riemann sampling.

To carry out the experiments, we used Python 3.6.1 with Anaconda custom (x86\_64). We used a MacBook with macOS High Sierra with the following features. 2.3 GHz Intel Core i5, 8GB 2133 MHz, L2 Cache:256 KB, L3: 4MB.

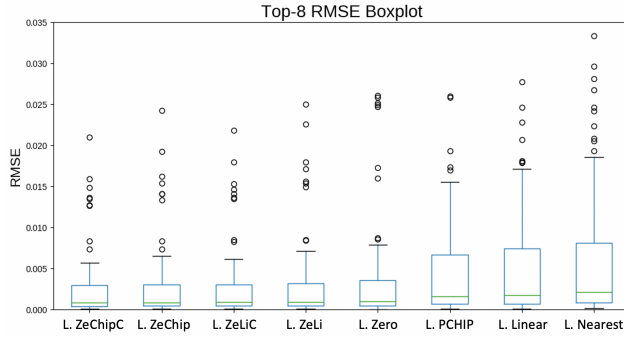
## 4.2 Data sets

The experiments have been conducted over the 67 databases of a repository called "The UEA and UCR time series classification repository" [2]. Some of the datasets have the training set and the testing set to perform classification techniques. Since we are not doing classification, we have simply joined the training set and testing set in a single dataset for each dataset of the repository.

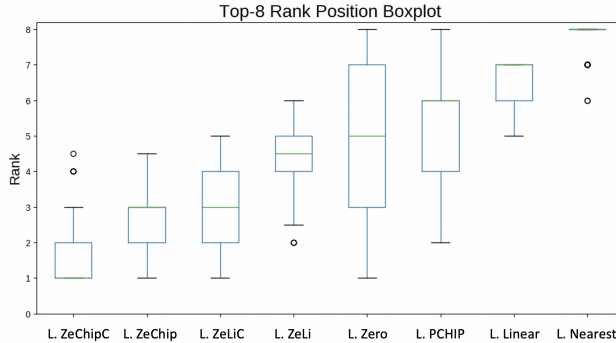
<sup>4</sup> Note that all the signals had been scaled between "0" and "1". From this perspective, 0.05 means a 5% difference in the maximum change between the possible values.

### 4.3 Experiment I

Figure 5 shows an illustrative summary of the performance of the best eight combinations of interpolation methods and sampling strategies. It could be possible that a method is strongly penalised in some dataset, and this will undermine its performance severely. To avoid this, we also calculated the average position, as displayed in Figure 6. We can see that the position order of the median RMSE is the same as the order in terms of the average position of the Ranking.



**Fig. 5.** Boxplot of the RMSE of the top-8 methods for all the 67 datasets, ordered by the median value.



**Fig. 6.** Boxplot of the rank position of the top-8 methods for all the 67 datasets.

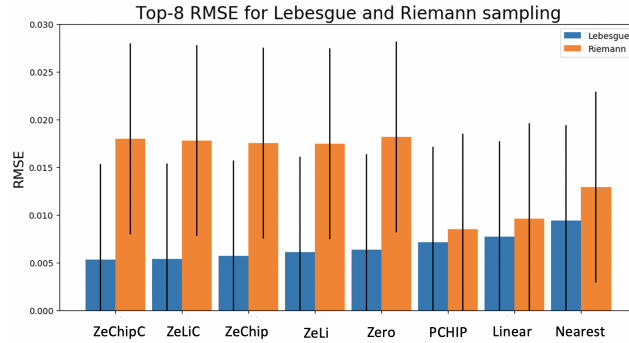
Figure 5 shows the box-plot of the RMSE score of the top-8 interpolation methods. In this plot, the interpolation methods are sorted by the 50th percentile, and from the Figure, it can be seen that ZeChipC performs best for the Lebesgue sampling and it produces least errors while reconstructing the signal.

Furthermore, it can be observed that interquartile range is smaller in magnitude compared to the rest of methods and as well whisker is at lower RMSE value, which establishes that ZeChipC performs very well in the overall spread of the reconstruction from the sampled single.

Likewise, Figure 6 shows a similar conclusion where ZeChipC is the winner in terms of the rank position of the reconstructed signal. The 50th percentile indicates that ZeChipC half of the time is a clear winner compared to the rest of the interpolation methods. Similarly, the interquartile range and the whisker establishes that overall ZeChipC outperforms the rest of the interpolation methods.

#### 4.4 Experiment II

In this experiment, results are shown in the same way as in the first one. First, in Figure 7, it is shown the performance based on the average RMSE of each combination (sampling and interpolation methods) using 15% of the samples for each of the 67 datasets.



**Fig. 7.** RMSE of the top-8 methods for all the 67 datasets.

As in the first experiment, in Figure 8, it is shown the average position of the best 12 methods of the dataset. We can see that the order of the positions of the Average Ranking is similar to that with the median RMSE value. For example, the order for the first eight combinations is the same.

We can also see that ZeChipC with Lebesgue sampling is the winner in terms of the rank position of the reconstructed signal. The 50th percentile shows that ZeChipC half of the time is the clear winner compared to the rest of the interpolation methods. Similarly, the interquartile range and the whisker establishes that overall ZeChipC outperforms the rest of the interpolation methods.

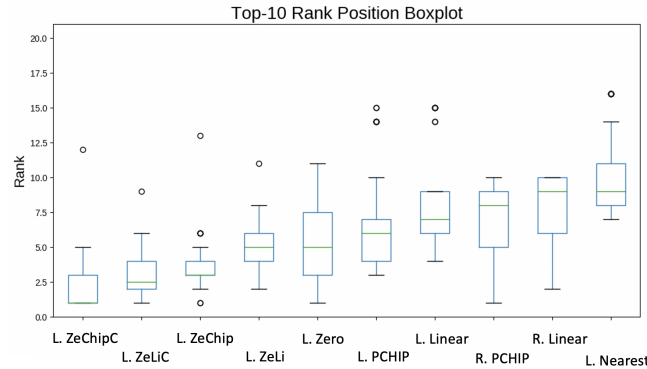


Fig. 8. Boxplot of the rank position of the top-10 methods for all the 67 datasets.

#### 4.5 Discussion of the experiments

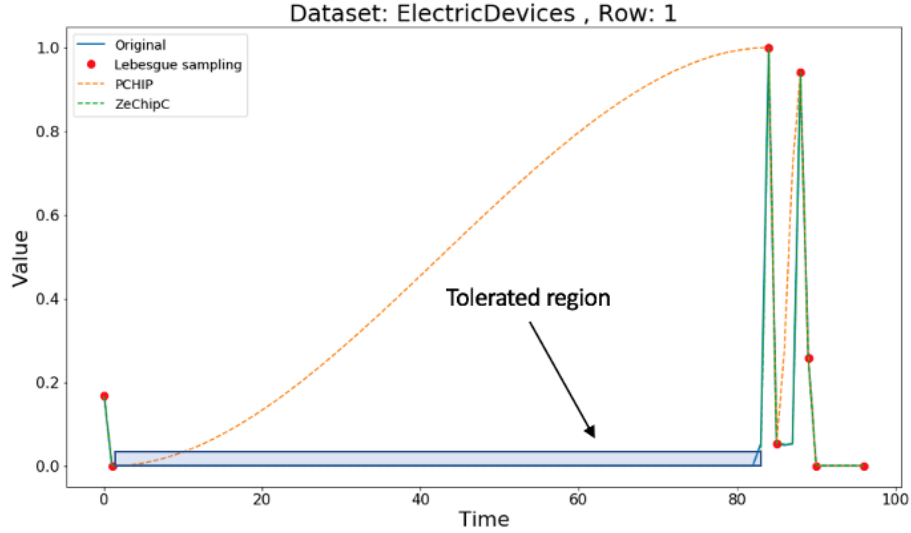
The interpolation method that offers the best performance in both experiments is ZeChipC. This method implements three ideas that have been presented throughout the paper.

First, it uses ZOH interpolation, which allows ZeChipC to adapt to abrupt changes. This improvement is shared by the other three developed methods (ZeLi, ZeLiC and ZeChip), and it can be clearly appreciated when we compare in experiment I the performance of ZeLi against Linear interpolation or ZeChip against PCHIP interpolation. ZOH is the only interpolation technique that guarantees that all the points are in the tolerated region which allows representing the shape of the signal more accurately. We can see a clear example of this in Figure 9 where PCHIP interpolation is out of the tolerated region while ZeChip is respecting it.

Second, ZeChipC includes new functionality to adapt to concave and convex regions. It is interesting to see that this improvement means an increment of 8.06% of ZeChipC with respect ZeChip (which does not implement the convexity/concavity functionality) in the first experiment and of 7.36% in the second. In the same way, there is an improvement of ZeLiC against ZeLi of 10.88% and 10.89% in the first and second experiments respectively.

The third and last idea consists of implementing PCHIP interpolation instead of Linear interpolation. The increase in the performance of this approach can be appreciated when ZeChip is compared against ZeLi, and when ZeChipC is compared against ZeLiC. In the first experiment, ZeChip has an improvement of 8.82% against ZeLi while in the second experiment, it has an improvement of 5.94%. In the same way, ZeChipC has an improvement against ZeLiC of 5.94% in the first and of 2.22% in the second.

In addition, it is worth stressing that the ZOH interpolation is better than Linear interpolation and even better than PCHIP interpolation. In fact, it is better than any other interpolation technique when Lebesgue sampling is used. PCHIP is the second best and Linear the third best. This is a confirmation



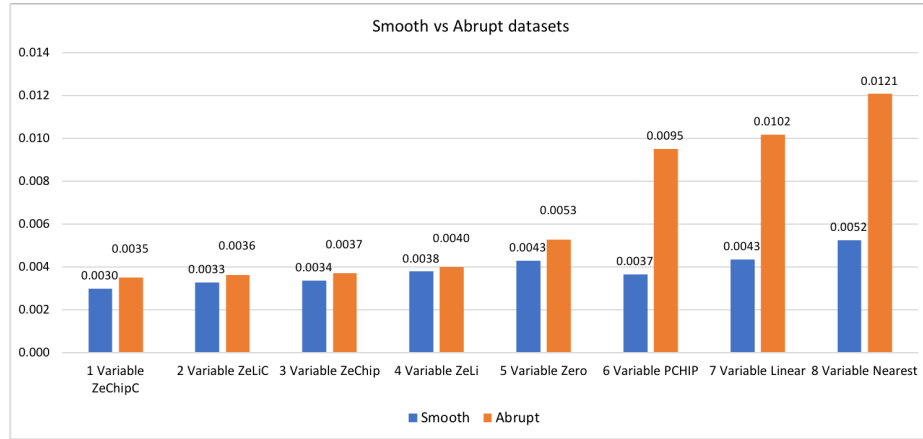
**Fig. 9.** PCHIP is interpolating outside the tolerated region and so, its performance is low.

that using ZOH for the algorithms ZeChipC and ZeLiC (as well as ZeChip and ZeLi) as a combination of these methods is a good idea. Regarding the rest of the methods, it seems that Lebesgue PCHIP, variable near, variable linear and variable ZOH always remain ahead of the rest of the methods. Our results strengthen the claims regarding that sampling based on Lebesgue sampling is more accurate than Riemann sampling (either in fixed or uniform with the same number of samples).

On the other hand, we could think that ZeChipC is not better because for some data sets simply because it has been in a better average position. To argue for the ZeChipC we have two arguments. In the position ranking, it has won 53 times out of 67 in the first experiment and 37 in the second. Furthermore, if we see the average position, it has been the first method in both experiments, 2.18 in the first experiment and 2.63 in the second.

Lastly, time-series smoothness is a concept that has been studied in detail in several investigations [3]. One of the most frequent ways of measuring it, and the one applied in our research is by calculating the standard deviation of the differences between the points (1st derivative). The lower the SD, the smoother the time series is.

The differences between the proposed methods and the rest of the methods are enlarged when the databases have a large number of changes. As it is shown in Figure 10, when the signals of a dataset have abrupt changes, the state-of-the-art methods do not "understand" that the change has occurred between the



**Fig. 10.** Average RMSE of the 15 smoother datasets against the 15 abrupter ones.

last tracked point and the previous instant and as a result, the signal is drawn out of the tolerance region as shown in Figure 9.

## 5 Conclusion

Throughout this investigation, we have developed an interpolation methodology that requires a smaller number of samples to achieve the same precision. This methodology is a solid basis on which systems related to ambient assisted living could be developed. Those systems could monitor time series related to health user attributes such as blood pressure, respiration, temperature, etc. but requiring fewer samples which makes them more efficient.

The main reason why the developed methods (ZeLi, ZeLiC, ZeChip and ZeChipC) have better results compared to the other interpolation methods is that the interpolation is performed by taking into account the Lebesgue sampling characteristics. That is to say, when there is an abrupt change ZOH interpolation is applied, otherwise (when there is a smooth change) Linear and PCHIP interpolation are applied. The proposed methods detect that there has been an abrupt change because the newly captured sample is far away from the tolerated region. Additionally, this decision can be optimised depending on the dataset using the tolerance ratio parameter.

On the other hand, the convexity/concavity functionality has performed very well. We can guess that when there is a change of sign in the slope of a signal, a concavity/convexity region has happened. Additionally, accurately calculating the exact point where the signal changes the slope and approximating its shape is a very complex and wide issue, although the applied implementation performed very well and boosted the performance of both methods: ZeLiC and ZeChipC.

The developed methods have been implemented based on the absolute difference with respect to the last sampled point. However, it is easily adaptable



to another kind of events in the health care domain that trigger sensors. For example, the sensor could be triggered when the output signal crosses a certain limit or when the percentage variation is higher than a preset limit. Using the same approach; Linear or PCHIP interpolation for smooth transitions and ZOH for abrupt changes will still be effective.

From the assumptions and contributions of this research, new and more effective interpolation methods could be designed. Lebesgue sampling is known in academia, but it is important to develop reliable and adapted tools to encourage the industry to make a transition to Lebesgue sampling.

**Acknowledgements.** This publication has emanated from research conducted with the support of Enterprise Ireland (EI), under Grant Number IP20160496 and TC20130013.

## References

1. Åström, K.J., Bernhardsson, B.: Comparison of periodic and event based sampling for first-order stochastic systems. *IFAC Proceedings Volumes* **32**(2), 5006–5011 (1999)
2. Bagnall, A., Lines, J., Vickers, W., Keogh, E.: The uea and ucr time series classification repository (2018), <http://timeseriesclassification.com>
3. Barnes, R.: Variogram tutorial. Golden, CO: Golden. Software. Available online at <http://www.goldensoftware.com/.variogramTutorial.pdf> (2003)
4. Fasshauer, G.E.: *Meshfree approximation methods with MATLAB*, vol. 6. World Scientific (2007)
5. Hamilton, J.D.: *Time series analysis*, vol. 2. Princeton university press Princeton, NJ (1994)
6. Heemels, W., Johansson, K.H., Tabuada, P.: An introduction to event-triggered and self-triggered control. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. pp. 3270–3285. IEEE (2012)
7. Meng, X., Chen, T.: Optimal sampling and performance comparison of periodic and event based impulse control. *IEEE Transactions on Automatic Control* **57**(12), 3252–3259 (2012)
8. Miskowicz, M.: Send-on-delta concept: an event-based data reporting strategy. *sensors* **6**(1), 49–63 (2006)
9. Mühlenstädt, T., Kuhnt, S.: Kernel interpolation. *Computational Statistics & Data Analysis* **55**(11), 2962–2974 (2011)
10. Sayakkara, A., Miralles-Pechuán, L., Le-Khac, N.A., Scanlon, M.: Cutting through the emissions: Feature selection from electromagnetic side-channel data for activity detection. *Forensic Science International: Digital Investigation* **32**, 300927 (2020)
11. Schimek, M.G.: *Smoothing and regression: approaches, computation, and application*. John Wiley & Sons (2013)
12. Yan, W., Zhang, B., Wang, X., Dou, W., Wang, J.: Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries. *IEEE Trans. Industrial Electronics* **63**(3), 1804–1812 (2016)
13. Zhang, B., Wang, X.: *Fault diagnosis and prognosis based on lebesgue sampling*. Tech. rep., University of South Carolina Columbia United States (2014)
14. Žukovič, M., Hristopulos, D.: Environmental time series interpolation based on spartan random processes. *Atmospheric Environment* **42**(33), 7669–7678 (2008)