expert

berlin

automotive engineering iav

International Conference on Calibration Methods and Automotive Data Analytics

Dr. Karsten Röpke, Prof. Dr. Clemens Gühmann, Matthias Schultalbers, Dr. Wolf Baumann, Dr. Mirko Knaak and 78 Co-Authors

International Conference
on Calibration Methods
and Automotive Data Analytics

International Conference on Calibration Methods and Automotive Data Analytics

Dr. Karsten Röpke, Prof. Clemens Gühmann,
Matthias Schultalbers, Dr. Wolf Baumann,
Dr. Mirko Knaak (eds.)
and 78 Co-Authors

expert›

berlin

automotive engineering iav

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Internet: www.expertverlag.de
eMail: info@verlag.expert

Printed in Germany

ISBN 978-3-8169-3463-9 (Print)
ISBN 978-3-8169-8463-4 (ePDF)

Preface

Discussions on electrification, air pollution control and driving bans in inner cities bring major challenges for powertrain development. Real Driving Emissions (RDE), Worldwide Harmonized Light-Duty Test Procedures (WLTP) and the next level of CO2 reduction enforce new development methods.

At the same time, new measurement technology and better IT infrastructure mean that ever larger amounts of data are available. Thereby, methods of digitization, e.g. Machine Learning, may be used in automotive development.

Another challenge arises from the ever-increasing number of vehicle variants. Many OEMs reduce the number of their engines to reduce costs. However, the basic engines are then installed with little hardware customization in numerous vehicle models. As a result, the application of derivatives and the systematic validation of an application play an important role.

In this book, the lectures of the International Conference on Calibration – Methods and Automotive Data Analytics held on May 21 and 22, 2019 in Berlin are contained.

We would like to thank all authors for their contributions to this conference.

Dr. Karsten Röpke, IAV GmbH

Prof. Dr. Clemens Gühmann, TU Berlin

Matthias Schultalbers, IAV GmbH

Dr. Wolf Baumann, IAV GmbH

Dr. Mirko Knaak, IAV GmbH

Contents

1 Data Analysis I

1.1 Segmentation of Multivariate Time Series with Convolutional Neural Networks

Yuncong Yu, Thomas Mayer, Eva-Maria Knoch, Michael Frey, Frank Gauterin

Abstract

This paper addresses an important problem of time series analysis at test benches. A new method is presented that allows automated segmentation of measurement time series using a Convolutional Neural Network (CNN). The CNN is trained for this purpose with specifically generated data. The results show a high quality and efficiency. The field of application of the algorithm is not limited to the automotive industry focused on here, but can be easily transferred to other areas that allow a visual segmentation of data.

Kurzfassung

Dieser Beitrag behandelt ein wichtiges Problem der Zeitreihenanalyse von Messreihen am Prüfstand. Es wird eine neue Methode vorgestellt, die mit Hilfe eines Convolutional Neural Networks (CNN) eine automatische Segmentierung von Messreihen ermöglicht. Das CNN wird hierzu mit gezielt generierten Daten trainiert. Das Ergebnis zeigt eine hohe Güte und Geschwindigkeit. Das Einsatzgebiet des Algorithmus ist nicht auf die hier fokussierte Anwendung in der Automobilindustrie beschränkt, sondern kann leicht auf andere Bereiche, die eine visuelle Segmentierung von Daten erlauben, übertragen werden.

1 Introduction

With rapidly accelerating product life cycles in the automotive industry, research and development based on modelling and simulation gain increasing significance. Nowadays models of systems in automobiles have grown extensively in complexity, which impedes comprehension and analysis of the system. Consequently, feasibility of manually evaluating the output data and optimizing the system suffers. A possible solution is a data analysis system that automates the evaluation process.

Examining of measurement and simulation data is one of the most common tasks for model analysis, evaluation and optimization. Many simulation and measurement results are time series which go through several phases. Therefore, segmentation is often the first fundamental step for further data mining.

2 Literature Review

2.1 Time Series Segmentation

Time series segmentation is partitioning a time series into several internally homogeneous [1, p. 466] and externally different but contiguous sub-time series [2, p. 1110]. It is often a preprocessing step in data mining that assists with extraction of interesting information in later steps [3, p. 50]. Time series segmentation algorithms can be classified into three categories: top-down, bottom-up and sliding windows [4, p. 2]. Top-down algorithms start from the coarsest segments and partition existing segments recursively [3, p. 38]. Whereas bottom-up algorithms begin with the finest possible segments and merge the most similar neighbors [3, pp. 39–40]. Sliding windows are like filters going through the time series and pick out the positions where the data characteristics change dramatically [3, p. 41]. Combinations of these algorithms are possible, for example, the combination of Sliding Windows and Bottom-Up (SWAB) [4].

These algorithms usually measure the internal homogeneity of a segment. This is where most of the innovations take place. One of the possible ways is to represent segments with simpler substitutions like lines by Piecewise Linear Representation (PLR) [4, p. 1] and polynomials by Piecewise Polynomial Representation (PPR) [5, p. 193]. A relatively advanced way to evaluate the homogeneity is Principle Component Analysis (PCA) [6].

In this study, a new approach based on a CNN is proposed, which falls into none of these three categories, as all segment boundaries in a time series are detected simultaneously. Hence, measurement of homogeneity is not necessary here. This approach reaches a decent accuracy, is universally applicable and highly efficient during application.

2.2 Convolutional Neural Networks

The Convolutional Neural Network (CNN) was first proposed by Yann LeCun and others in 1998 [7, p. 121]. The recent decade has witnessed a variety of image-related applications of CNNs. For instance in pulmonary nodule (a sign at the early stage of lung cancer) detection from Computed Tomography (CT) images [8] and facial recognition including even micro-expression recognition [9] [10]. The typical input data for CNNs are two-dimensional image data, however one-dimensional CNNs are proven to be a quick alternative to Recurrent Neural Networks (RNNs) for sequence data processing [11, p. 288]. A recent application case of one-dimensional CNNs is the detection of the QRS complex (a special signal pattern) in Electrocardiogram (ECG) for cardiovascular disease diagnosis [12].

The CNN used in this paper is responsible for detecting boundaries in time series. These boundaries partition the time series, which enables further segment-wise extraction of information.

3 Time Series Generator

Training of neural networks requires a quite large amount of training data. As there are not enough suitable data, usually labelled and proven, a time series generator was developed to provide sufficient data.

The main function of the time series generator is to mass produce random time series according to user configurations. User configurations include mainly the number of time series to generate, length of the time series, the number of channels in a time series, allowed numbers of segments in a channel, minimum segment length and intensity of different types of noise.

The time series generator can generate two types of data. The first type contains “single” time series, where each sample includes only one time series. An example of a single time series with three channels is shown in Figure 1. In each channel, the data curve contains several segments, like horizontal line segments, line segments with a certain slope unequal to 0 and segments corresponding to the step responses of first-order linear time-invariant (LTI) system (commonly known in German as “PT1”). There are random steps between segments. Segment boundaries in different channels are not completely random. The background behind this setting is that the time series can be regarded as the state of a system observed against time. Each channel represents a measured signal like temperature, pressure, voltage. A change in the observed system may result in sudden changes (boundaries in the curves) in several channels. These boundaries in different channels are usually close to each other because they indicate the same change in the observed system.

The second type contains pairs of time series, where each sample includes two similar time series, as shown in Figure 2. Pairs of time series are generated to represent pairs of measurement and simulation data to feed the CNN for an automatic time series comparing algorithm [13].

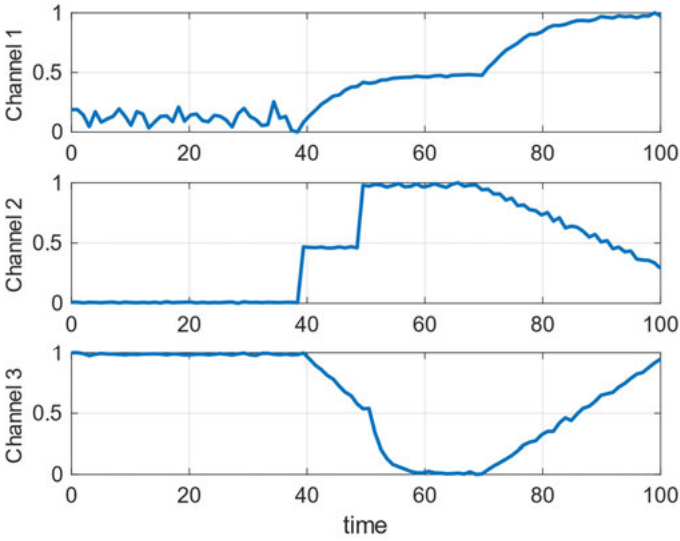

Figure 1: A single time series with three channels

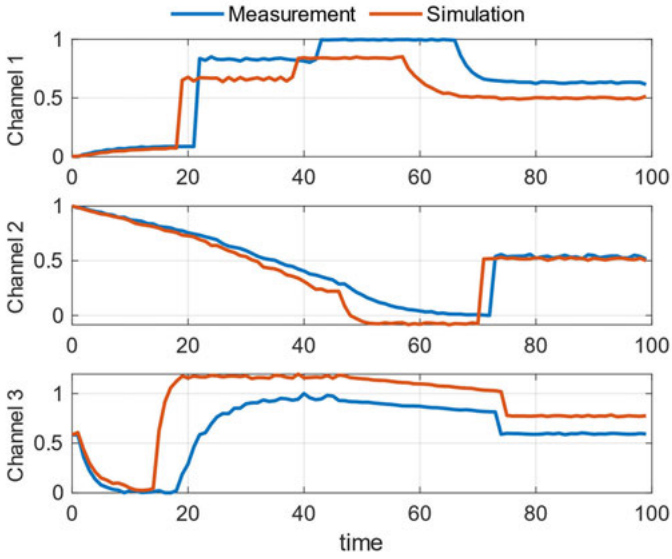

Figure 2: A pair of time series with three channels

4 Time Series Segmentation Algorithm

4.1 Requirements

In general, the time series segmentation algorithm splits a given time series into segments. These segments are internally homogeneous [1, p. 466]. Adjacent segments exhibit different characteristics and are contiguous in time. Contiguity means no overlap and no gap between two neighboring segments.

The time series used to develop, train and test the neural network for the time series segmentation algorithm are generated by the time series generator described in Chapter 3.

A question for segmentation is, whether to find the same boundaries for all the channels (“global” boundaries) or different boundaries for each channel (“local” boundaries). In this research, the time series segmentation algorithm segments a multivariate time series one channel after another separately, detecting “local” boundaries in each channel. Finding the “local” boundaries enables the calculation of time shifts between channels. If only global boundaries are needed, it is also completely feasible to train the CNN to read several channels simultaneously and segment them with the same set of boundaries. The detailed algorithm can be looked up in [14].

4.2 CNN for Segmentation

The structure of the developed CNN used for time series segmentation is shown in Figure 3. Closely related steps for processing input and output data are also presented.

The input data for the CNN is a single channel or visually a curve. The output data of the CNN also constitute a time series, which has the same length as the original input channel. Each value in the output time series indicates the probability of the time point being a boundary. Inside the CNN, the data go through three one-dimensional convolutional layers in the CNN. The deeper the layer, the more filters are applied and the longer these filters become, which is typical for a CNN. Untypical for this CNN is that there are no max-pooling layers because experiments without them show better results. An intuitive explanation could be that max-pooling leads to loss of location information due to its massive downsampling effect, during which the length of output data shortens. In a typical application of CNNs, an object in an image is detected regardless of its position. Whereas, boundary detection in a time series requires to know not only the existence of these boundaries, but also their time dependent positions. Out of the same reason, zero padding is used in each convolutional layer to ensure that the length of the interim output data always stays the same.

Detailed configurations of the applied CNN are shown in Table 1.

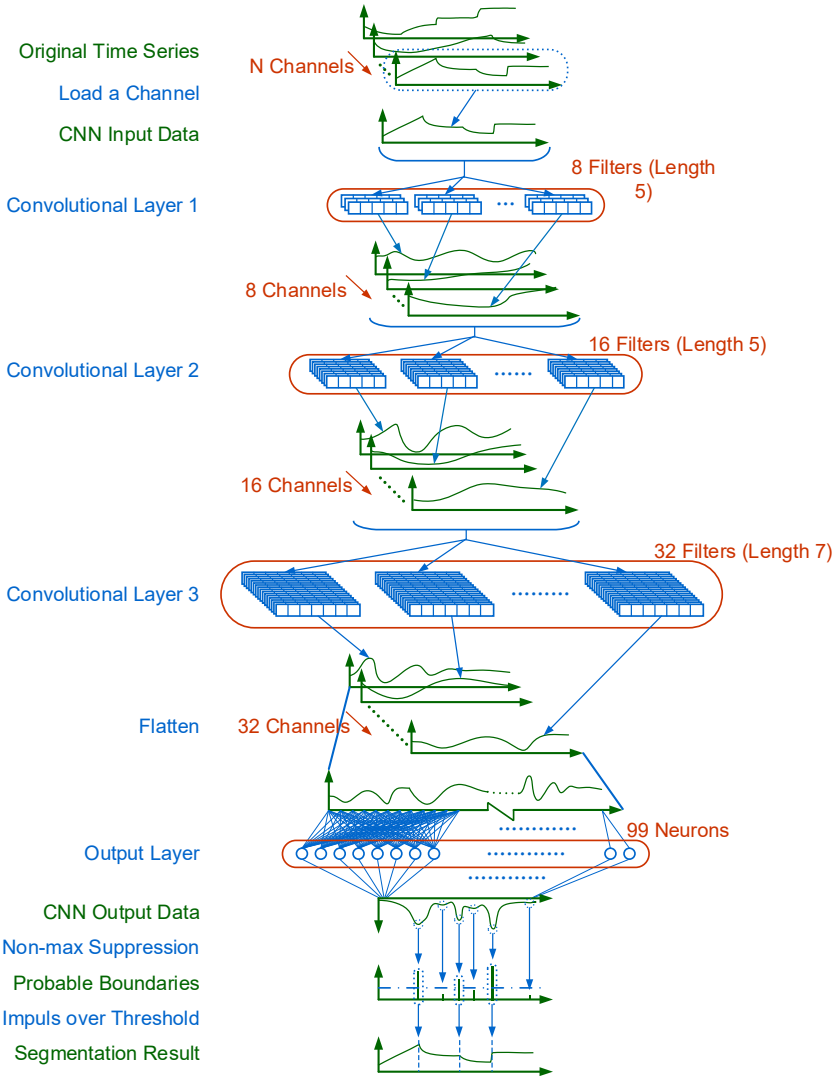

Figure 3: CNN for time series segmentation

Table 1: Configurations of the CNN for time series segmentation

Layer	Layer Type	Size	Activation Function
1 (input)	-	100	-
2	Convolutional	Number of filters: 8 Kernel Size: 5	ReLu
3	Convolutional	Number of filters: 16 Kernel Size: 5	ReLu
4	Convolutional	Number of filters: 32 Kernel Size: 7	ReLu
5	Flatten	-	-
6 (output)	Fully connected	100	sigmoid

This CNN is trained with 800'000 time series as training data and 200'000 time series as validation data. The optimiser Adam is used with the loss function binary cross-entropy, which is typical for multi-class, multi-label classification like this case. The training and validation data are generated with the time series generator described in Chapter 3.

The performance of the CNN is further evaluated and verified with 1'000 time series. They are generated separately with the time series generator and have no association with the training and validation data mentioned above. The results are shown in Table 2.

Table 2: Evaluation of the CNN for time series segmentation

	Number	rate
Time series samples	1'000	100%
Correctly segmented time series	808	80.8%
Boundaries	4'976	100%
Correctly detected boundaries (excluding multi-detected ones)	4'842	97.3%
Undetected boundaries	134	2.7%
Multi-detected boundaries	0	0%
Falsely detected boundaries	125	2.5%

A boundary is defined as correctly detected, when there is exactly one detected boundary within 2 seconds around it (the length of all the original time series is 100 time steps). Table 2 shows a satisfactory evaluation outcome. More than 97% of the boundaries are correctly detected; less than 3% of the boundaries are ignored or falsely detected.

5 Results

To illustrate the results of the segmentation algorithm, a sample from the automatically segmented time series is plotted in Figure 4. There are three channels in this time series, labeled from Channel 1 to Channel 3. The data curve in each channel is segmented with the time series segmentation algorithm. The dashed lines are detected boundaries. The plot shows satisfactory results intuitively.

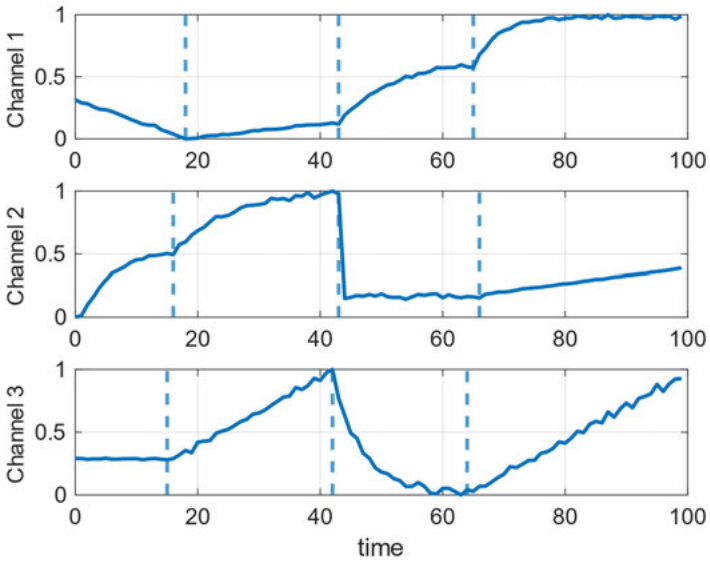

Figure 4: Examples of time series segmentation

6 Conclusion

In this study, a new method for segmentation of multivariate time series is presented. It is based on the techniques of convolutional neural networks (CNN). A time series generator is developed to support training of the CNN and testing of the method. Performance evaluation shows satisfactory results. 97% of the boundaries in the time series can be correctly detected.

The presented method favors two new approaches. On the one hand, it has proposed a way to use machine learning even when not enough training data is available. In this case, synthetic data adapted to the application can be generated to support training process. One the other hand, a new way to segment multivariate time series is put forward. Various segment types can be segmented, not only linear ones as with many conventional approaches. Besides, there are quite less parameters to tune during application and the algorithm runs very efficiently during application. Nevertheless, there is still much room for improvement of the presented method.

References

[1] Vasko, K. T. and Toivonen, H. T. T. 2002. Estimating the Number of Segments in Time Series Data Using Permutation Tests, *IEEE International Conference on Data Mining* , pp. 466–473.

[2] Graves, D. and Pedrycz, W. 2009. Multivariate Segmentation of Time Series with Differential Evolution, *IFSA/EUSFLAT Conference* , pp. 1108–1113.

[3] Lovrić, M., Milanović, M. and Stamenković, M. 2014. Algoritmic Methods for Segmentation of Time Series: An Overview. *Journal of Contemporary Economic and Business Issues*, Vol. 1, No. 1, pp. 31–53.

[4] Keogh, E., Chu, S., Hart, D. and Pazzani, M. 2004. Segmenting Time Series - A Survey and Novel Approach. M. Last, A. Kandel and H. Bunke (eds), *Data Mining in Time Series Databases* , pp. 1–21.

[5] Xu, Z., Zhang, R., Kotagiri, R. and Parampalli, U. 2012. An Adaptive Algorithm for Online Time Series Segmentation with Error Bound Guarantee, *Proceedings of the 15th International Conference on Extending Database Technology*. New York, NY, USA, ACM (EDBT '12), pp. 192–203.

[6] Abonyi, J., Feil, B., Nemeth, S. and Arva, P. 2005. Modified Gath-Geva Clustering for Fuzzy Segmentation of Multivariate Time-Series. *Fuzzy Sets and Systems*, Vol. 149, No. 1, pp. 39–56.

[7] Skansi, S. 2018. *Introduction to Deep Learning : From Logical Calculus to Artificial Intelligence.* Cham, Springer. (Undergraduate Topics in Computer Science SpringerLink : Bücher).

[8] Xie, H., Yang, D., Sun, N., Chen, Z. and Zhang, Y. 2019. Automated Pulmonary Nodule Detection in CT Images using Deep Convolutional Neural Networks. *Pattern Recognition*, Vol. 85, pp. 109–19.

[9] Singh, R. and Om, H. 2017. Newborn Face Recognition Using Deep Convolutional Neural Network. *Multimedia Tools and Applications*, Vol. 76, No. 18, pp. 19005–15.

[10] Peng, M., Wang, C., Chen, T., Liu, G. and Fu, X. 2017. Dual Temporal Scale Convolutional Neural Network for Micro-Expression Recognition. *Frontiers in Psychology*, Vol. 8, p. 1745.

[11] Chollet, F. 2018. *Deep Learning mit Python und Keras: das Praxis-Handbuch vom Entwickler der Keras-Bibliothek.* 1st ed. Frechen, mitp.

[12] Xiang, Y., Lin, Z. and Meng, J. 2018. Automatic QRS complex detection using two-level convolutional neural network. *BioMedical Engineering OnLine*, Vol. 17, No. 1, p. 13.

[13] Yu, Y., Mayer, T., Knoch, E.-M., Frey, M. and Gauterin, F. 2019. Time Series Comparison with Dynamic Time Warping, Convolutional Neural Network and Regression, *Proceedings of the International Conference on Calibration - Methods and Automotive Data Analytics*.

[14] Yu, Y. 2018. Analysis, Comparison and Interpretation of Multivariate Time Series. Master Thesis, Karlsruhe Institute of Technology.

1.2 Time Series Comparison with Dynamic Time Warping, Convolutional Neural Network and Regression

Yuncong Yu, Thomas Mayer, Eva-Maria Knoch, Michael Frey, Frank Gauterin

Abstract

This paper introduces a novel method for comparison of similar time series, especially measurement and simulation data to identify problems in the observed system. It employs the technique for time series segmentation proposed in [1] together with Dynamic Time Warping (DTW) to jointly segment pairs of measurement and simulation time series. Further, a Convolutional Neural Network (CNN) is used to identify the characteristics of segments. It is trained with synthetic data generated by the time series generator presented in [1]. Finally, the essential parameters are estimated with regression. Performance evaluation of each step is conducted and shows a high accuracy. The usage of this method is not restricted to evaluation of measurement and simulation time series, but can be extended to serve the general purpose of sequence data comparison.

Kurzfassung

In diesem Beitrag wird eine neuartige Methode zum Vergleich ähnlicher Zeitreihen, insbesondere Zeitreihen von Mess- und Simulationsdaten, vorgestellt. Ziel ist es, Unterschiede in den beiden Zeitreihen zu bestimmen, um eventuelle Probleme im beobachteten System zu erkennen. Die in [1] präsentierte Methode zur Segmentierung von Zeitreihen zusammen mit der Methode der dynamischen Zeitnormierung (DTW, engl. Dynamic Time Warping) wird hierbei angewendet, um eine synchronisierte Segmentierung von Mess- und Simulationsreihen zu ermöglichen. Außerdem wird ein Convolutional Neural Network (CNN) eingesetzt, um den funktionalen Zusammenhang der Zeitreihen innerhalb der Segmente zu klassifizieren. Das CNN wird hierzu mit synthetischen Daten trainiert, die von einem Zeitreihengenerator (siehe [1]) erzeugt werden. Die wesentlichen Parameter werden mit Hilfe von Regression geschätzt. Das vorgestellte Vorgehen zeigt eine hohe Prognosegüte. Der Einsatzbereich dieser Methode ist jedoch nicht nur auf die Auswertung von Mess- und Simulationszeitreihen beschränkt, sondern lässt sich auf den allgemeinen Vergleich von Datensequenzen erweitern.

1 Introduction

Model-based development plays an increasingly important role in automotive engineering. With rapidly growing complexity of models, their data analysis becomes much more difficult and often far too demanding for manual evaluation.

Many simulation results are in the form of time series showing the temporal response of a system under specific configuration. Comparison of simulation results with measurement is one of the essential ways to find error causes in the model. Therefore, techniques to compare time series play a key role in describing the dynamics behind the curves in automotive data analysis.

However, a direct comparison of two arbitrary curves is barely feasible. Because there are so many different characteristics that a curve can assume. Hence, a method is needed to handle the problem of time series comparison.

2 Literature Review

2.1 Time Series Comparison

Comparing two time series, especially a pair of measurement and simulation time series is one of the most common tasks in modelling and simulation. An intuitive way to compare two time series is to calculate the L1 norms (sum of absolute values) or L2 norm (Euclidean length) of their differences.

Two advanced techniques based on the methods mentioned above are Dynamic Time Warping (DTW) and Edit Distance with Real Penalty (ERP) [2, p. 569], which take into consideration the time shifts between two time series [3, p. 792]. ERP is first introduced and thoroughly described in [3]. DTW is used in this study and will be explained in the following section.

Other methods include Longest Common Subsequence (LCSS) and Edit Distance with Real Sequence (EDR), which are based on a notion called matching threshold. They are proven to be more robust against noise compared to the methods mentioned above [4, p. 673] [5, p. 491].

These traditional methods measure the distance or similarity of two time series. In this paper, a novel method for comparing two time series based on segmentation, CNN and regression is put forward. It is fully automated and can provide a detailed assessment digging deep into various aspects of the two compared time series.

2.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm to measure similarity of two time series [6, p. 38], as it can match points of two similar time series, for example, measurement and simulation data with time shifts between them. Consequently, a point at time t_1 in one time series may correspond to one point at time t_2 ($t_2 \neq t_1$) in the other times series. DTW can handle this problem of time distortion [6, p. 38]. However, there is the restriction that the start points and end points of two time series should match respectively. Figure 1 shows two time series denoted blue and orange respectively.

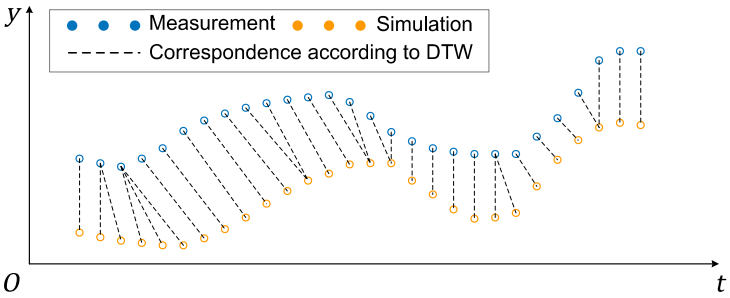

Figure 1: Matching of corresponding points on two time series with similar characteristics using Dynamic Time Warping (DTW)

Corresponding points are matched with black dashed lines as a result of DTW.

Details about implementation of DTW is not handled in this paper because the algorithm is extensive and can be looked up in [7, p. 193].

DTW is utilized in this paper to match measurement and simulation time series. Together with the time series segmentation algorithm described in [8], it enables simultaneous segmentation of both time series.

2.3 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a member of Artificial Neural Networks (ANNs), or Deep Neural Networks (DNNs) to be precise.

Generally, a CNN begins with several blocks, each of which is made up of one or more convolutional layers [9, p. 121] followed by a max-pooling layer. The output of the last block is then flattened out by a flatten layer and fed to a small conventional network with two or more fully connected layers. The last fully connected layer is also the output layer of the whole CNN [10, p. 439] [9, p. 126].

Configuration of CNNs means setting suitable hyperparameters. Parameters such as weights and bias are internal variables in a CNN. They are initialized and updated

automatically during training and not directly set by the user. On the contrary, hyperparameters are user configured. They influence the values of parameters, for example the number of layers, number and size of filters in the convolutional layers, activation functions and loss functions.

3 Comparison Method

3.1 Overview of the Algorithm

The proposed method compares segmented measurement and simulation data. It evaluates their similarity quantitatively from many aspects. In the first step, the measurement and simulation data are segmented jointly with the segmentation algorithm proposed in [1] together with the sequence matching algorithm DTW. Next, the segment characteristics of the measurement and simulation segments are identified with a CNN and compared. There are three segment types within the scope of this paper, namely “constant”, “linear” and “PT1”. “PT1” refers to the step response of a first-order linear time-invariant (LTI) system. Then the essential parameters (for instance the time constants for “PT1” segments) for the segment will be estimated according to the segment types using regression. Finally, error convergence and time shifts at both ends of each segment will be checked.

3.2 Joint Segmentation of Time Series Pairs with Dynamic Time Warping

With the method for time series segmentation proposed in [1], it is only possible to segment a single time series, not a pair of measurement and simulation data. If measurement and simulation data are segmented separately, they are likely to have different sets of boundaries, and it is intractable to find the relationship between the two sets of boundaries.

In the proposed algorithm, the CNN based segmentation method in [1] is applied to segment measurement and simulation time series separately in the first step. The results are probabilities of each time point being a boundary [1]. Then, DTW is used to match points on the simulation curve with those on the measurement curve. The mean value of the possibilities of corresponding points are calculated, and if this value for a pair of corresponding points is over 0.5, both points are recognized as boundaries, one for measurement curve, the other for simulation curve. Note that a pair of corresponding points may not be at the same time, which means that time shifts are taken into consideration during averaging.

Figure 2 shows an example of joint segmentation of measurement and simulation data. The blue curve refers to measurement data and the red one to simulation data. Both have three channels numbered from Channel 1 to Channel 3. They can be arbitrary variables, like temperature, pressure or current. The blue dots denote the real boundaries of the measurement data and the red ones the real boundaries of the simulation

data. The detected boundaries are delineated with dashed lines in corresponding colours.

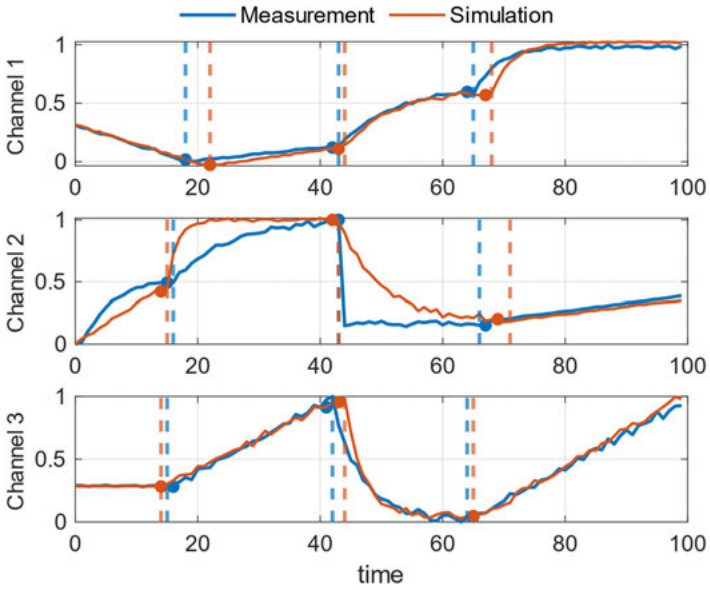

Figure 2: An example of joint segmentation of measurement and simulation data

The overall performance of the method based on the CNN proposed in [1] and DTW is evaluated with 1'000 pairs of measurement and simulation time series generated by the time series generator presented in [1]. Each of them comprises three channels. Each channel includes 1 to 4 segments, or rather 0 to 3 boundaries. The evaluation results with these test data are shown in Table 1.

Table 1: Evaluation of the joint segmentation of time series pairs

	Overall		Measurement		Simulation	
	Number	Rate	Number	Rate	Number	Rate
correctly segmented channels	2'055	69%	2'209	74%	2'076	69%
correctly detected boundaries	11'453	88%	5'842	90%	5'611	87%
undetected boundaries	1'417	11%	629	10%	788	12%
multi-detected boundaries	72	1%	0	0%	72	1%
falsely detected boundaries	876	7%	390	7%	486	8%

The results for segmentation of pairs of measurement and simulation data are slightly worse than the evaluation results for segmentation of only single time series in [1]. Correctly detected boundaries are slightly less than 90% and undetected a little more than 10% compared with 97.3% and 2.7% for single time series [1]. The accuracy decreases, because a much more complex problem is dealt with, where the measurement and simulation data are processed in conjunction. Nonetheless, the results are still well acceptable.

3.3 Identification of Segment Characteristics with a Convolutional Neural Network

Knowing the segment type is the prerequisite to apply regression and determine parameters of the segment. Comparing essential parameters is much easier as direct comparison of two curves and makes it possible to discover problems in simulation or measurement.

The presented approach is classification based on a convolutional neural network (CNN) to identify the segment characteristics. Traditional ways involve trying different models and comparing their errors. These methods are computationally expensive, especially when dealing with many possible segment types. On the contrary, CNNs usually work efficiently during application, regardless of the number of possible segment types. More efforts are needed only during training process. In this study, a trained CNN takes the segment as its input and gives one of the segment types, "constant", "linear" or "PT1" as output.

The configuration of this CNN is listed in Table 2.

The CNN is trained with around 32'000 segments as training data and around 8'000 segments as validation data using optimizer Adam with loss function categorical cross-entropy, which is typical for multi-class classification like this case. These data are generated by the time series generator in [1].

Table 2: Configurations of the CNN for segment type identification

Layer	Layer Type	Size	Activation Function
1 (input)	-	20	-
2	Convolutional	Number of filters: 8 Kernel size: 3	ReLU
3	Max-pooling	2	-
4	Convolutional	Number of filters: 16 Kernel size: 5	ReLU
5	Flatten	-	-
6	Fully connected	32	ReLU
7 (output)	Fully connected	3	softmax

3.4 Regression of Segment Parameters

After the segment types of both measurement data and simulation data are identified, parameters according to the segment types can be determined using regression. As mentioned, there are three types of segments dealt with in this study: “constant”, “linear” and “PT1”

The constant value of a horizontal line and the slope of a line segment can be found using linear regression. Calculating the time constant τ of a “PT1” segment, namely the step response of a first-order LTI system, is based on the differential formula [11, p. 194]

$$\tau \cdot \dot{y}(t) + y(t) = K \cdot u(t) \tag{1}$$

where y represents the values of the segment, t is time, K is the unknown constant gain of the “PT1” segment and $u(t)$ is a Heaviside step function defined as

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{2}$$

In order to carry out regression, the term $\dot{y}_t$ is substituted with its finite difference with sampling period Δt, in this case 1 time step:

$$y(t) \approx K \cdot u(t) - \tau \cdot \frac{y(t + \Delta t) - y(t - \Delta t)}{2\Delta t} \tag{3}$$

Now only the two constant coefficients τ and K are unknown and can be calculated using linear regression.

4 Results

By way of illustration, a time series sample with measurement and simulation data is processed using the presented method. In the first step, the pair of time series is segmented by the time series segmentation algorithm. The result is shown in Figure 3.

The blue dots denote the real boundaries of the measurement data and the orange ones the real boundaries of the simulation data. The detected boundaries are delineated with dashed lines in corresponding colours. The black dotted lines match boundaries of the measurement data to corresponding boundaries of the simulation data.

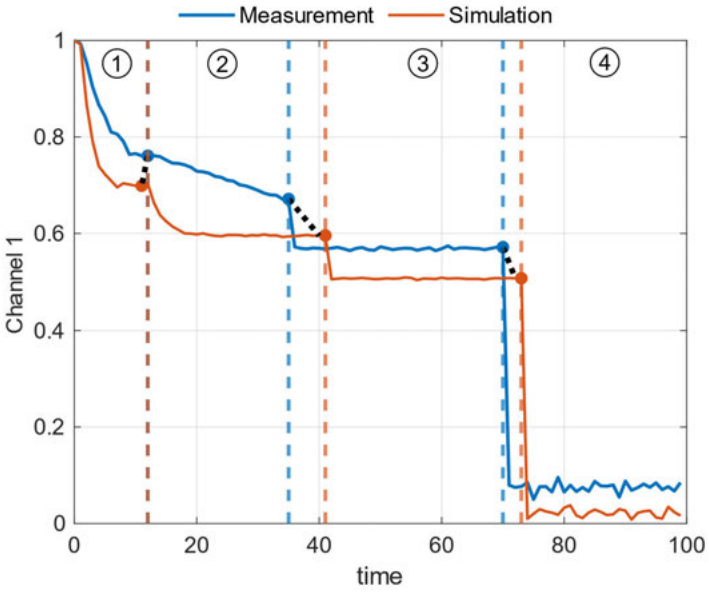

Figure 3: An example of a jointly segmented time series pair

Then, these two curves are evaluated one segment after another. For a better illustration, the chosen pair of time series contains only one channel (channel 1). For multivariate time series with more than one channels, the algorithm will go through each channel separately.

The comparison results are shown in Table 3. It shows detailed assessment of the measurement and simulation data that outstrips the conventional methods mentioned in Section 2.1 for comparing two time series.

Table 3:Comparison results of a sample time series

Segment Number		1	2	3	4
Segment Type	Measurement	PT1	Linear	Constant	Constant
	Simulation	PT1	PT1	Constant	Constant
	Comparison	Matched	Mismatched	Matched	Matched
Parameter	Parameter Name	Time constant	-	Constant value	Constant value
	Measurement	4.03	-	5.09×10^{-1}	2.49×10^{-2}
	Simulation	1.6	-	5.06×10^{-1}	-2.77×10^{-2}
Error Convergence	Left	-1.65×10^{-3}	-9.73×10^{-2}	-6.69×10^{-2}	-6.95×10^{2}
	Right	-5.62×10^{-2}	-7.55×10^{-2}	-6.43×10^{-2}	-4.40×10^{2}
Time Shift	Left	0	0	6	3
	Right	0	6	3	0

5 Conclusion

In this paper, a novel method for comparing pairs of measurement and simulation data is presented. Firstly, dynamic time warping combined with the time series segmentation algorithm proposed in [1] is used to segment the pair of time series. Then, a convolutional neural network is applied to identify the characteristics of the segments. Finally, regression is carried out to estimate the essential parameters of each segment according to the identified segment type.

With this method, it is possible to compare pairs of time series with the objective of detecting possible problems in the data set and better characterising their functional behaviour.

In a practical example, a reasonable accuracy is reached, with 87% boundaries in time series pairs detected, 98% segment characteristics identified and decent precision of parameter estimation even for complex segments like “PT1”.

Future work includes the extension of the method to more segment types, especially non-monotonic segments like polynomials and “PT2” (the step response of a second-order LTI system) and further validation of the method with real data.

References

[1] Yu, Y., Mayer, T., Knoch, E.-M., Frey, M. and Gauterin, F. 2019. Segmentation of Multivariate Time Series with Convolutional Neural Networks, *Proceedings of the International Conference on Calibration - Methods and Automotive Data Analytics*.

[2] Morse, M. D. and Patel, J. M. 2007. An Efficient and Accurate Method for Evaluating Time Series Similarity, *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA, ACM (SIGMOD ’07), pp. 569–580.

[3] Chen, L. and Ng, R. 2004. On the Marriage of Lp-norms and Edit Distance, *Proceedings of the 13th International Conference on Very Large Data Bases - Volume 30*, VLDB Endowment (VLDB ’04), pp. 792–803.

[4] M. Vlachos, G. Kollios and D. Gunopulos. 2002. Discovering Similar Multidimensional Trajectories, *Proceedings 18th International Conference on Data Engineering*, pp. 673–684.

[5] Chen, L., Özsu, M. T. and Oria, V. 2005. Robust and Fast Similarity Search for Moving Object Trajectories, *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA, ACM (SIGMOD ’05), pp. 491–502.

[6] Sio-Iong Ao. 2010. *Applied Time Series Analysis and Innovative Computing.* Dordrecht, Springer. (SpringerLink : Bücher, 59).