Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

Intelligent Signal Processing in an Automated Measurement Data Analysis System

Henrik Isernhagen*, Helmut Neemann†, Steffen Kühn* and Clemens Gühmann*
*Technische Universität Berlin, Department of Electronic Measurement and Diagnostic Technology
Sekr. EN 13, Einsteinufer 17, 10587 Berlin, Germany
email: {henrik.isernhagen, steffen.kuehn, clemens.guehmann}@tu-berlin.de
phone: +49 30 314-22280, fax: +49 30 314-22120
†IAV GmbH, Rockwellstraße 16, 38518 Gifhorn, Germany
email: helmut.neemann@iav.de

***Abstract*— In the automotive sector a huge amount of measurement data is recorded for validation and safeguarding of vehicle components. These data has to be automatically evaluated for an effective data analysis. Therefore, we need a sophisticated approach, which offers a flexible and powerful parametrisation and different signal processing algorithms for multiple applications. In this paper software and signal evaluation modules for an automated analysis of vehicle measurement data are presented. The data can be evaluated signal or message based with a parametrisation with reusable XML templates. Exemplary, we describe three evaluation modules integrating different signal processing approaches: signal analysis using an analytical signal description in combination with fuzzy logic, an efficient sliding frequency detection and the detection of predefined patterns using a modified dynamic time warping algorithm. Furthermore, an approach for a connected evaluation in consideration of time correlation is presented. Concluding, we discuss a practical application.**

I. Introduction

In the development of new vehicle components for the automotive sector the complexity and integration of electronic units has a considerable increase. In contrast to this the time of development and test has been abbreviated in consequence of shorter vehicle generation cycles. According to these conditions an effective testing and safeguarding is necessary. One approach to obtain the requirements is an automated measurement data analysis.

In terms of the decreasing costs of data memory often the complete bus communication of a vehicle under test is recorded today. Taking this into account we get up to 2500 parallel signals and about 80MB of data per hour measuring run. In consideration of non equidistant timestamps, especially at the common used CAN-Bus (Controller Area Network), we need a sophisticated evaluation implementation. The recorded measurement data can be evaluated automatically to different system and failure states. On the one hand, the data can be processed message based e. g. for protocol specific analysis like the network management. On the other hand, the design offers the possibility to evaluate the measurement data signal based. Miscellaneous evaluation modules are available for different signal processing tasks. The signal processing can be parametrised by XML templates. It also offers different functions for data preprocessing. In XML the software offers a multifunctional tool in form of an analytical descriptive language with fuzzy connection for an elementary signal evaluation and combination. Furthermore, there exist different evaluation modules for particular data analysis, like a modified dynamic time warping algorithm for the detection of predefined signal patterns or an efficient sliding single tone detection.

In this paper an approach for an automated measurement data analysis is presented. Section II shows the concept and the evaluation structure of the automated analysis. Section III presents different intelligent signal processing modules and explains the functionality and possibilities of the available signal evaluation. The practical application will be described in Section IV.

II. Automated Measurement Data Analysis

In consequence of the huge amount of measurement data stored in various data files an automated data analysis requires an effective implementation. The data has to be evaluated to different requirements and applications and the user needs the possibility for an individual and reusable analysing configuration. The automated evaluation system presented below fulfils these requests with a flexible and expandable process.

A. Concept

To obtain an universal data evaluation the framework supports different data logger systems and file formats, respectively. For signal analysis the approach provides different evaluation modules. There are modules for signal preprocessing, message based analysis modules in context of CAN and extended signal processing modules. The user has the possibility of defining an individual signal analysing process by a descriptive language. Furthermore, we provide a flexible connection of evaluation results and a powerful protocol creation. A further important point is the ability to manage configurations, evaluation results and measurement data in a data base file system. According to this the parametrisation and result protocols are XML files.

B. Structure

The structure of the automated measurement data analysis is shown in Fig. 1. The evaluation engine sets up the central

1-4244-0707-9/07/$25.00 ©2007 IEEE

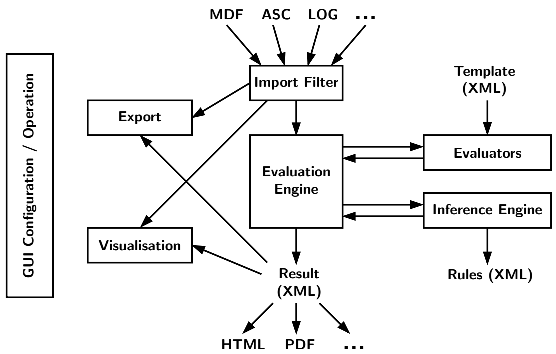

Fig. 1. Software structure of the automated analysis

module, which controls the evaluation process. First, the engine reads the configuration and parametrise the evaluation process accordingly. Subsequently, it reads the measurement data sequentially via the import filter and delivers them to the so called evaluators. Huge data traces can be handled unproblematically with the sequential data processing point by point. The import filter supports the common automotive measurement data file formats, including CAN.

The evaluators execute the ultimate data analysis. They work either message or signal based and return a measure for the appearance of the respective analysis event to the evaluation engine. In Section III the functionality and the flexibility of the evaluators are presented in detail. The evaluation engine analyses the result measures and generates an entry in the evaluation protocol in accordance with the specification.

The automated measurement data analysis is parametrised by template files in the XML format. A template file contains the name of the used evaluator, the required signals, preprocessing tasks, if necessary, and the evaluator specific parameters. For a parallel analysis and reutilisation the template files will be selected by a configuration file. The result protocol is also a XML file. So it can be easily transferred to a HTML or PDF file by the XSLT transformation.

Furthermore, we have modules for result visualisation and data export. After the evaluation, the analysed signals can be visualised with markers at the signals where detected events occur. The information of time, duration and value of the event is received automatically from the result protocol. In addition, it is possible to do an automated data export in form of a stream cut with a pre and post trigger around the event for further inspection or result documentation.

Operation of the evaluation process, choice of configuration, setting of directory paths, etc., can be done with a GUI or with batch files. In the future, the connection to an inference engine for information extraction is planned. By that the sources of faults can be derived from a knowledge base e. g. for maintenance.

C. Implementation

The automated data evaluation system is implemented in Java for platform independence. Java offers many efficient libraries for various applications. By using a modular concept the system is expandable and can easily be connected to other applications.

III. Signal Evaluation

The presented approach gives us extensive possibilities for different signal evaluation applications. The automated analysis can process the data to several events in parallel through a modular structure, a sequential data processing and the power of different evaluation modules in form of the evaluators. We provide a programming language as a free customisable signal processing and further evaluators in the form of specialised evaluation functions. Moreover, the approach offers the ability to combine evaluators to describe and to detect complex events.

A. Evaluators

In principle, an evaluator is a function of a signal or a message. Normally, the output is a real number, which is a measure for the appearance of a defined event. In the broader sense the measure can be defined as a probability. A “zero” stands for “event occurs definitely not” and a “one” represents “event appears definitely”. The input data comes from the evaluation engine and the evaluator passes back the measure. The output measure can also be an input for other evaluators, so they can be used in any combination.

We present below three evaluators for different applications as examples. The first evaluator can be individually parametrised for various signal processing task. The second evaluator is a specific frequency detector working with an efficient calculation. The evaluator presented as third includes an algorithm for the detection of predefined signal patterns in a continuous data stream. The automated analysis system includes further evaluators for various other tasks like message based network monitoring for CAN or signal state classification.

1) Analytical Signal Description: This evaluator is the general basis of the signal processing system and is labelled as the *FuzzyEvaluator*. The *FuzzyEvaluator* unifies different requirements in one signal based evaluator. It combines an analytical signal description with methods of the fuzzy logic. In addition, the evaluator allows time displacement and operation on intervals. For these requests we have defined a syntax in form of an analytical descriptive language, which is oriented on a verbal description of the analysis process.

The verbal oriented syntax makes the usage of the *FuzzyEvaluator* very comfortable. The user needs no knowledge in a programming language like java. He can describe the analysis process in an intuitive verbal order. We provide different functions like interval operations, relational and conjunction operations, mathematical functions as well simple filter operations. Furthermore, variables and internal signals can be defined.

A kind of fuzziness can be included through the connection of the analysis process with functions of the fuzzy logic. The result of the *FuzzyEvalutor* is a measure for the appearance of the described event and lies in the interval $[0, 1]$. The internal interim results are connected by the rules of the fuzzy logic. In case of an AND operation that is a minimum calculation. Furthermore, a fuzziness can be specified in connection with the relational operations. As a result, a threshold comparison (e.g. is greater than) is not longer a dual decision, but a floating transition from "zero" (occurs definitely not) to "one" (completely fulfilled).

An additional feature is the integration of time shifts and the access to intervals during the analysing process. During the evaluation it will be processed by a delay of signals and interim results. The time delay refers to the actual evaluation time point. It is specified by a negative value associated to the past. The sequential data processing allows only a relation to past values. By this the *FuzzyEvaluator* is suited for an evaluation in a temporal combination: on the one hand, for the connection of the analytical signal description, and on the other hand, for the result combination of external evaluators as presented in Section III-B. Furthermore, the evaluator provides different interval operations like a moving average or the calculation of a mean gradient.

2) Sliding Frequency Detection: The standard method for spectrum analysis is the discrete Fourier transform (DFT) which is typically implemented in form of the fast Fourier transform (FFT). In order to get a continuous data analysis the FFT normally works on a sliding window to get a time resolution of the spectrum analysis using the short-time Fourier transform (STFT). Typically, the time resolution is considered to be slow, because all Fourier coefficients have to be calculated new for each new window with a complexity of $\mathcal{O}(N \log N)$.

For signal analysis there is often only a subset of the centre frequencies of interest. Regarding to this, the FFT is very inefficient, because most of the calculations will be discarded. The detection of a single bin can be calculated with a single bin DFT or with the Goertzel algorithm to decrease the number of complex operations [1] [2] [3]. The Goertzel algorithm works as a second-order infinitive impulse response (IIR) filter, with two real feedback coefficients and a single complex forward coefficient. It needs $N + 2$ real multiplications and $2N + 2$ real additions. The advantages in respect to the FFT are less amount of filter coefficients storage, no storing of block input data is needed and N does not need to be a power of 2. But the single tone detection generates a result after N input samples and has to be initialised again, afterwards.

A solution for a sample by sample evaluation is the sliding DFT and the sliding Goertzel algorithm described in [1] [2] [3], which gives the actual spectral bin with every sample. For frequency evaluation we implement the sliding Goertzel algorithm with a time-domain windowing in the frequency-domain. The transfer function $H(z)$ of the sliding Goertzel algorithm is defined by

$$H(z) = \frac{1 - \mathrm{e}^{-j2\pi m/N}(1 - z^{-1})}{1 - \cos(2\pi m/N)z^{-1} + z^{-2}} \tag{1}$$

with $m = N \cdot f_i/f_s$. Where f_i is the frequency of interest in Hz. f_s is the sampling rate in Hz. N is the window length.

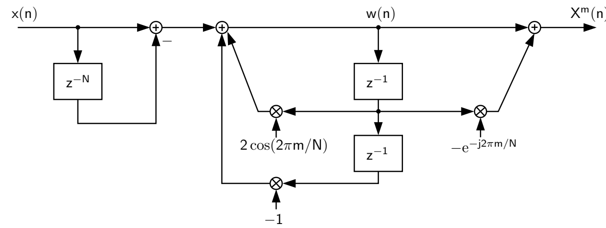

Fig. 2. Structure of the sliding Goertzel filter

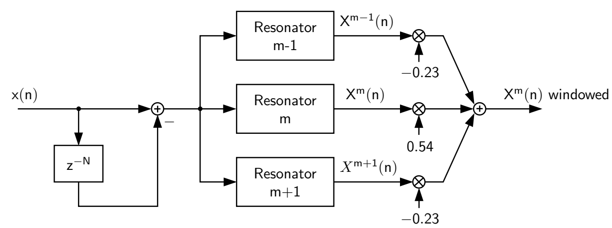

Fig. 3. Hamming frequency-domain windowing with sliding resonators

m is an integer. The filter structure is shown in Fig. 2. Each new DFT is efficiently computed directly from the results of the previous DFT with only three real multiplications and four real additions. The initial DFT after N input samples needs $N + 2$ real multiplications and $3N + 1$ real additions.

To reduce the spectral leakage we use the concept of windowing in the time-domain. However, the sample by sample computation does not work with time-domain windowing. We can solve this problem with the convolution theorem properties of discrete systems. So we implement the windowing in the frequency-domain by a convolution of the relevant spectral bins with the DFT of the window function. For example, the DFT of a Hamming window has only three non-zero values. The Hamming frequency-domain windowing for the m-th bin is defined as the following three-term convolution:

$$X^m_{HW}(n) = -0.23X^{m-1}(n) + 0.54X^m(n) - 0.23X^{m+1}(n) \tag{2}$$

According to this, the m-th Hamming-windowed spectral bin $X^m_{HW}(n)$ can be computed using three resonators, which calculate the adjacent DFT bins X^{m-1}, X^m, X^{m+1} (as shown in Fig. 3). The three resonators are implemented in form of the sliding Goertzel algorithm.

3) Detection of Predefined Patterns using Dynamic Time Warping: Occasionally, patterns are characterised by a characteristic shape over the time by one or more channels. The measurements will vary in their form for each individual case. Distortions can occur, and deviations in the signal amplitude, too. For this reason a fuzzy search method is needed, which compares the signal with the pattern by tolerating a slight deviation.

Such a method can be realised by *dynamic time warping* (DTW) [4] [5]. The basic method serves to determine a warping function $w : i \rightarrow j$ with $i \in [1, n]$ and $j \in [1, m]$

so that the distance

$$D(n,m) = \frac{1}{n}\sum_{i=1}^{n} d(\boldsymbol{x}(i), \boldsymbol{y}(w(i))) \tag{3}$$

between two given sequences $S_1 = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(n)\}$ and $S_2 = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(m)\}$ becomes minimal. As local distance criterion d different vector norms are suitable. A concrete example for d is the Euclidean norm.

The special advantage of DTW is the fact that $D(n,m)$ can be computed particularly efficiently with *dynamic programming* [6]. The *local constraint* [5] we use is defined by

$$\begin{aligned} D(\tau,t) = &\min\{D(\tau-1,t-1), D(\tau,t-1)\} \\ &+ d(\boldsymbol{x}(\tau), \boldsymbol{y}(t)). \end{aligned} \tag{4}$$

The index $\tau \in [1,n]$ goes over all samples of the model $\boldsymbol{x}$ (boundary conditions: $D(\tau=0,t=0)=0$, $D(\tau=0,t>0)=\infty$, $D(\tau>0,t=0)=\infty$). The index t marks the current sample of the examined signal $\boldsymbol{y}$.

The local decision $\min\{.\}$ has the following meaning: maintain either the distance $D(\tau,t-1)$ or take the predecessor distance $D(\tau-1,t-1)$, if it is more favourable. However, the rule of Eq. (4) causes that no model samples can be jumped over. In principle, no signal compression is possible, but only a signal stretching. That means $w(i)$ is always greater than or equal to i. This is unproblematic, because the model can be selected in such a way that it corresponds to the maximal possible compression. On the other hand, Eq. (4) has a significant advantage, because it makes a particularly efficient "spotting" possible, i.e. infinite long signals $\boldsymbol{y}(t)$ can be scanned for the occurrence of a short model $\boldsymbol{x}(t)$. The basic procedure always presupposes $w(1)=1$ and $w(n)=m$, so that the starting point and the endpoint are fixed from the beginning. Here a "spotting" is not possible, because $w(n)$ would be ∞ and the pattern would be stretched infinitely. The special structure of Eq. (4) makes it possible to define the boundary conditions $D(\tau=0,t\geq 0)=0$. Therefore, every sample of $\boldsymbol{y}(t)$ can be a starting point. The value $D(n,t)$ computed by the modified DTW is then a measure for the agreement between model $\boldsymbol{x}$ and signal $\boldsymbol{y}$ at the time t.

The entire algorithm is extremely efficient and has a time complexity of $\mathcal{O}(n)$ per time step. The memory requirement is limited to the values $\boldsymbol{D}(t-1) = \{D(1,t-1), D(2,t-1), \ldots, D(n,t-1)\}$. At each time t this vector is updated by Eq. (4) to $\boldsymbol{D}(t)$. All in all, the method structure is very similar to a finite impulse response (FIR) filtering.

B. Connected Evaluation in Consideration of Time Correlation

The detection of a defined system state often depends on different events in multiple signals. In terms of an analytical signal description this can be evaluated by simple and-or conjunctions with the *FuzzyEvaluator*.

In addition, we have the possibility to combine different evaluators to detect complex states. The output measure of a signal based evaluator can be the input of another signal based evaluator. The evaluator output measure for the appearance of the event represents internally a new signal. By that we can combine several evaluators to an unitised configuration. This means that a complex state can be divided into partial states which are detected separately, following a result summarisation to one measure.

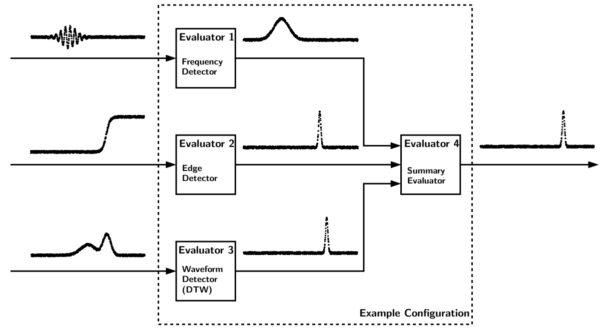

Fig. 4. Example configuration of a connected evaluation

An example for a unitised configuration is shown in Fig. 4. The three evaluators in the input layer analyse the signal stream to defined partial events of the complex state. This is a preprocessing in form of the detection of characteristical signal shapes. At this point we have an information conversion. The relevant signal information will be transferred into an easier accessible information for further evaluations. In Fig. 4 this is the detection of a sinus burst, of a rising edge and of a predefined waveform with the modified dynamic time warping algorithm. The summary evaluator analyses the results of the input layer and calculates a summary measure for the complex state for each input sample.

The summarisation process has not only to consider a correlation between several signals. There can also be a correlation in time between the different partial events [7]. The evaluation process has to take into account these possible time shifts in the result summary. Normally, the time shifts can not be exactly parametrised. Mostly, they will be described as an interval in which the particular event occurs. This means that the summary evaluation needs a connected evaluation considering relative time correlations on different events. With the containment of the time correlation by indicating an allowed time interval for each partial event this is a connected maximum calculation over the different intervals.

The *FuzzyEvaluator* can also be used for these requirements. Starting from the temporal last event the evaluation intervals of the adjacent events have to be parametrised. The time shifts of the interval boundary of the adjacent events have to be described relatively to the actual evaluation time point as a negative time value following the sequential data processing. With the interval boundaries for each partial event we get a fixed evaluation raster, which slides over the respective signals during the sequential evaluation.

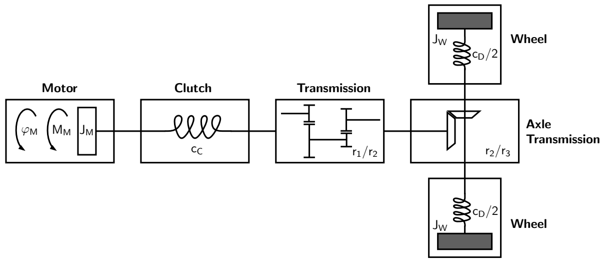

Fig. 5. Simplified oscillation model for jerking

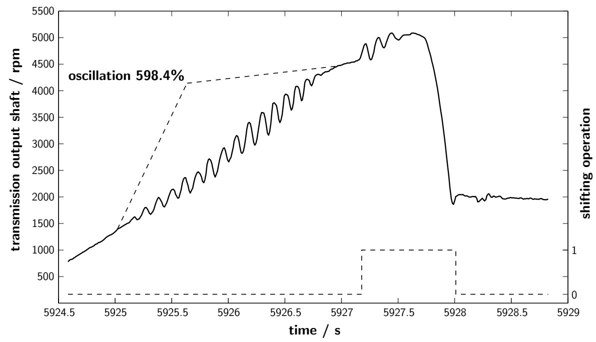

Fig. 6. Oscillation on the transmission output shaft

IV. Application in Practice

The following is a description of a practical application of the automated measurement data analysis approach. In endurance tests a new automatic transmission system is analysed for safeguarding. The test cars are provided with measurement data loggers, and the collected data is evaluated off-line. The evaluation is parametrised by XML templates.

A main focus is the detection of jerking oscillations on the powertrain. They result in a vehicle longitudinal oscillation. The reasons for jerking oscillations can be load alternation clunks, bucking engine operations, and deceleration jerking ("bonanza effect"). The frequency range of the jerking oscillation is between 8 and 20 Hz. Passengers sense these frequencies as particularly disturbing.

A simplified oscillation model of the jerking oscillation on the powertrain is shown in Fig 5. The engine torque M_M accelerates the inertia torque J_M. It also affects the transmission input shaft with the clutch torsion-spring rate c_C, with the transmission ratio r_1/r_2 on the transmission output shaft, and from there via the axle transmission with the ratio r_2/r_3 on the two elastic driving shafts with the torsion-spring rate c_D. Furthermore, the wheels with the inertia moment J_W distort against the axle transmission output shaft and against the tires [8]. The overall distortion can be up to a half turn.

We use the sliding frequency algorithm (as described in Section III-A.2) for the detection of jerking oscillations on the powertrain. The centre frequency of interest is set to 10 Hz, and its amplitude is set to 40 rpm. The analysed signal is the rotational speed of the transmission output shaft. An evaluation result is shown in Fig. 6. The marker specifies an event recognition of 598.4%. This means that the maximal amplitude of the centre frequency bin of 10 Hz is 5.984 times the nominal amplitude of 40 rpm.

Besides, the recorded measurement data of the endurance test is evaluated in parallel with other templates to different tasks. First, the intervention of a driver assistance system like the anti-lock braking system (ABS), or the electronic stability program (ESP) is observed, because they also lead to a jerking on the powertrain. Then, the fault memories entries of the control units, the moments of the fault memory entries, and the trigger times are read out.

V. Conclusion

In this paper, we presented an approach for an automated measurement data analysis with intelligent signal processing modules. The automated measurement data analysis allows a signal or message based data evaluation of parallel events. The evaluation process can be parametrised efficiently by XML templates. Moreover, the system supports the common automotive measurement file formats and it can handle huge data traces with a sequential data processing.

The signal processing is executed by different evaluators. They analyse the signals to defined events and give a measure for the appearance. We described three different evaluators for multiple applications: an analytical descriptive language with fuzzy logic operations for an individually programmable signal processing evaluator, an efficient sliding frequency detector, and an evaluator for the detection of predefined patterns in a continuous data stream. Furthermore, we discussed a combined evaluation to detect complex events in multiple signals considering the time correlation.

Concluding, the evaluation of an endurance test of an automatic transmission system is presented as a practical application. This example shows the flexibility and the manifold possibilities of the automated measurement data analysis.

References

[1] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Mag.*, vol. 3, pp. 74–80, March 2003.

[2] ——, "An update to the sliding DFT," *IEEE Signal Processing Mag.*, vol. 1, pp. 110–111, January 2004.

[3] R. G. Lyons, *Understanding Digital Signal Processing*. Prentice Hall, 2004.

[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[5] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, March 2005.

[6] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[7] H. Isernhagen, S. Kühn, C. Gühmann, H. Neemann, and M. Papendieck, "Automated measuring data analysis in the vehicle development (in German)," in *Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG 2006*, VDI. VDI-Verlag, March 2006, pp. 389–398.

[8] M. Mitschke and H. Wallentowitz, *Dynamik der Kraftfahrzeuge*, 4th ed. Springer, 2004.