INTERACTIVE KNOWLEDGE DISCOVERY IN RECORDINGS FROM VEHICLE TESTS

[1]Theissler, Andreas* [1]Ulmer, Daniel [2]Dr Dear, Ian
[1]IT-Designers GmbH, Germany [2]Brunel University, United Kingdom

KEYWORDS – vehicle electronics, visual data mining, temporal data mining, time series, automotive trace analysis

ABSTRACT

Modern vehicles contain a highly complex network of hardware and software subsystems (1). To be able to locate faults or to evaluate the behaviour of subsystems, the communication on the vehicle's networks is being recorded by measurement systems (13) – so called data loggers. This is for example done during road trials or Hardware-in-the-loop (HiL)-tests. The amount of data resulting from each recording is huge – easily in the region of several million data points. Analyzing the recordings is very time-consuming. This paper discusses an approach for the analysis of mass data resulting from vehicle tests.

The recordings can be viewed as time series data (2). In order to uncover faults or evaluate the performance of algorithms, this research work proposes an approach to interactively explore the data recordings, which is referred to in this work as *interactive knowledge discovery from multivariate time series*.

The traditional way of presenting this type of data – plotted w.r.t. time or as individual scatter plots – is not sufficient due to the complexity of the data. In this work, a combination of various techniques from the fields of visual data exploration (7) (8) and temporal data mining (4) is applied. The approach combines and enhances the two existing visual data exploration techniques "parallel coordinates" (10) and "scatter plot matrix" (2) to cope with time series resulting from vehicle tests. Additionally a facility to query a time series by graphically formulating a search pattern is integrated. This enables the user to interactively analyse the recordings by formulating sophisticated filtering and querying operations. The approach was implemented and shall be named "Automotive Trace Miner".

INTRODUCTION

Current vehicles have up to 80 electronic control units (ECUs) communicating over bus systems. The ECUs read data measured by sensors, calculate values and control actuators. The sensors' and actuators' values are being transmitted over the vehicle's bus systems to other ECUs (1). This results in a highly complex network of software and hardware subsystems delivered by a variety of suppliers leading to a high potential for faults either caused by one individual subsystem or by the integration of those subsystems in the vehicle's network. To be able to locate faults or to evaluate the behaviour of subsystems the bus traffic is being recorded by measurement systems (13) – so called automotive data loggers. Analysis of the recordings allows for the reconstruction of the driving dynamics and situations the vehicle was in, e.g. abrupt steering manoeuvres or the vehicle's velocity can be determined from the data. These kind of recordings are conducted by the manufacturers during several testing phases, e.g. during road trials before start of production or during the development and prototyping of individual vehicle functions. Once vehicle production has started sporadic test drives are being conducted and recorded with chosen vehicles to ensure the vehicles' manufacturing quality.

The amount of data resulting from each recording is in the region of several million data points. Analyzing each recording in great detail is therefore not feasible. The traditional way of presenting this type of data – plotted w.r.t. time or as individual scatter plots – is not sufficient due to the high number of signals involved in vehicle functions. In this work, a combination of various techniques from temporal data mining (4) and visual data exploration (7) (8) is applied on the data. This allows for time-saving analysis of recordings from road-trials as well as HiL-tests (1). For example erroneous sensors or faults in the software of electronic control units can be identified by comparing the recordings to a data base of information generated from pre-defined driving manoeuvres thus revealing unwanted deviations. Another example is the application of the proposed approach to analysing field data to acquire knowledge of typical driving habits. This is particularly done in the field of commercial vehicles such as buses or trucks. The extracted knowledge is used in the construction of future vehicles or to influence or cope with the driving habits in order to optimize fuel consumption. Examples could be the driver's gear shifts and engine revolutions.

BACKGROUND

In this work, a recording refers to discrete time-stamped signal values recorded during tests. A recording typically contains multiple signals and is viewed as time series data. A time series is defined as a finite sequence of data points ordered by time (2) and denoted by x(t). The distance between the data points is finite and typically equidistant. A data point is one point of an univariate time series which corresponds to a signal's value at a given point in time.

Time series data can be univariate or multivariate. Measurements of one signal are referred to as an univariate time series x(t), while multivariate time series contain measurements of multiple signals, for example x(t), y(t) and z(t). In recordings from vehicle tests, multivariate time series can be observed. The individual time series are possibly correlated. So x(t) might be input data from a sensor, y(t) a calculated time series based on the sensor data and z(t) the resulting output data transmitted to a connected actuator.

The aim of the work is to discover knowledge in time series data resulting from vehicle tests. Following the definition of Fayyad (3), knowledge discovery is the "nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". This can be done in an automatic manner, by applying data mining algorithms, or in an interactive way by integrating the user into the process.

INTERACTIVE KNOWLEDGE DISCOVERY

Proper visualization techniques can support the user during interactive exploration of data. In (6) Shneiderman introduced the term information-seeking mantra which could be described as the steps to be taken to gain information from visualised data in general. Those steps are: *overview first*, *zoom* and *filter* and then *details on demand*. Further steps to be taken are *relate*, *history* and *extract*. Those steps – seven altogether – have shown to build the basis for the way data exploration should be done. *Overview first* refers to the need for the user to get an overview of the data, which – although seeming trivial – is a challenge when visualising mass data. Interesting subsets can be found and investigated further, which is then done by *zooming* in to analyse the found subsets. It is important to note, that zooming in this case is not just a way to resize the visualization under investigation. Moreover the capability to show more details at greater zoom levels is important. Due to the immense amount of data the user needs to be able to focus on an interesting subset. Reducing the data to a subset is done using *filtering* mechanisms. *Details on demand* refers to the possibility to request further details of individual objects by e.g. using a different visualization technique or simply a tool-tip. The

step *relate* refers to a way of showing relationships between items while *history* simply indicates the need for some form of undo and redo functionality. Finally the *extract* step allows the user to export interesting portions of the data.

While in (6) the steps are described in a very general context – the visualization of information in general – Keim related this principle to visual data exploration in (8). In visual data exploration, the exploration process takes place in an interactive way. Numerous different visualization techniques are known starting with standard 2D- or 3D-plots or charts formed from calculation tools to geometric techniques. In (8) Keim gives an overview of the topic. Although there is a variety of visualization techniques, meaningful visualization techniques for high-dimensional multivariate time series data with very many variables are rare. Most of the current visualization techniques were not especially developed to use time series data or use only a few time-dependent variables. The classical way to visualise time series data are line graphs, where the values are plotted as a function of time. This type of visualization is sufficient for a small number of time series. In many application domains, there is a variety of different time series that have to be related. In this case, visually relating the time series is not possible. For interactive knowledge discovery from multivariate time series, enhanced visualization techniques are required. This is particularly the case in analysing data from complex vehicle systems where the nature of the communication systems can result in unpredictable relationships between time series being produced when faults occur.

In this work an interactive approach is proposed that is designed to work on time series data. The approach combines the two existing visual data exploration techniques “parallel coordinates” (10) and “scatter plot matrix” (2). The two techniques are enhanced to cope with multivariate time series data and to enable the user to formulate sophisticated filtering and querying operations. This is combined with a facility to query a time series by graphically formulating a search pattern based on techniques of temporal data mining. Additionally, a variety of techniques to pre-process and transform the input time series are integrated. The approach shall be referred to as “Automotive Trace Miner” and consists of:

- enhanced scatter plot matrix
- enhanced parallel coordinates
- graphical time series query
- interaction between the three techniques
- pre-processing and transformation facility

Enhanced scatter plot matrix

Visually relating two variables can be done using a scatter plot or x/y-plot, where each variable is mapped to one Cartesian axis. For several variables a so called scatter plot matrix (2) can be used, which contains pairwise projections of the attributes. Information about time cannot be deduced from the diagram without additional tools. Visualizing several time series using scatter plot matrices will therefore enable the user to determine e.g. dependencies between different time series, but without being able to draw conclusions about the point in time. If the variables x, y and z are related, the type of relation becomes obvious and e.g. outliers can easily be determined. The information about when these outliers occurred is not present in the diagram. Depending on the number of data points and the size of the display ten to fifteen different variables can be realistically related that way. The relation though is strictly done pairwise, as can be seen in Figure 1.

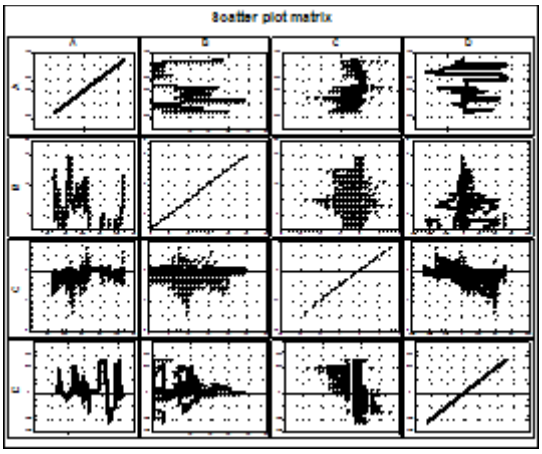

Figure 1: Scatter plot matrix

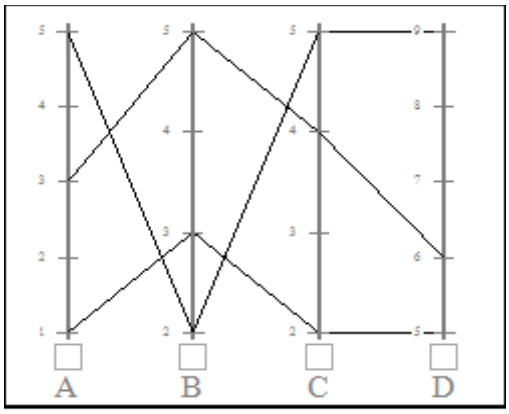

Figure 2: Parallel coordinates

The idea of a scatter plot matrix was enhanced by a number of features: data points in one of the x/y-plots can be highlighted by the user by drawing a rectangle in one of the x/y-plots. Each data point is identified by its timestamp. Based on the timestamp, the highlighting is propagated to the values of all time series in the remaining x/y-plots. This is referred to as "brushing". Sophisticated queries can be formulated incrementally by allowing subsequent brushing operations to be linked by the Boolean operators AND, OR and NAND which allows to interactively search for interesting parts in the data by refining or extending the currently highlighted data points. Unbrushed data points can be removed in order to focus on the interesting parts, corresponding to a filtering operation as demanded in (6). If a high number of data points are brushed, it is beneficial to highlight the data points using a colour gradient, in order to distinguish the data points.

Enhanced parallel coordinates

Using a scatter plot matrix, variables can be visually related in a pairwise manner. In technical processes one variable is often dependent on a combination of several variables. Relating *n* variables can be done using parallel coordinates, first introduced by Inselberg (10). Relating *n* time series of length *d* is done by drawing *n* parallel axes. Each variable's value is mapped to the position on its axis. The values on the axes are then connected by line segments which results in *d* lines. A parallel coordinates plot for four variables is shown in Figure 2. Without additional tools, the pure visualization is only beneficial for very small data sets. In (9) several enhancements of parallel coordinates are proposed with the focus being on the colouring of the lines. In this work, an enhancement of parallel coordinates is proposed to cope with time series from automotive recordings. It currently offers the following functionality:

Showing the distribution and general dependencies without prior user interaction can be achieved by drawing transparent lines, which is done using the α-channel of the colour. Ranges with a high density of data items get coloured in a more intense way while the remaining parts appear transparent. In addition colouring the data items of the parallel coordinates view using a colour gradient leads to a visualization that allows the overall-structure of the data to be recognised, as shown in Figure 7. The level of transparency as well as the axis the colour gradient is based on can be interactively influenced by the user.

A value range can be selected and thereby highlighted by the user (11), referred to as *brushing*. Individual brushing operations work on selected axes. In this work, the enhancement was made that subsequent brushing operations can be linked by Boolean operators. The Boolean operators are applied to the set of currently brushed data items and the

data items contained in the current brushing operation. This way, sophisticated queries can be formulated on a time series data base, e.g. find all parts in the data where:

- A(t) is in the range of [30, 50]
- OR B(t) is in the range [-100, 100]
- AND C(t) is NOT = 0
- AND D'(t) > 0, i.e. D(t) is increasing

The vector holding the timestamps is added as an additional axis, which allows deducing the timestamps when for example brushing data items. The highlighted data items can be coloured based on a colour gradient in order to distinguish the individual data items. Following the demanded *history* as described in (6), the history of sequential brushing operations can be visualised. Based on this history, redo and undo-functionality is offered. To focus on the relevant parts of the data, all brushed or unbrushed data items can be removed at any time. It is essential to arrange the axes in a meaningful way. The proposed approach allows for the re-arrangement of axes by the user. A more sophisticated approach could be to automatically place similar time series close to each other. As opposed to visual data mining tools for the analysis of unstructured data, working on time series allows to offer functions defined for time series: differentiation and integration as well as point-wise arithmetic like addition or subtraction of time series are integrated.

Graphically querying time series

A prototype was developed that allows the user to query a univariate input time series for a specified search pattern. The search pattern is formulated graphically and transformed into a time series, where the number of data points of the search pattern is typically much smaller than in the input time series. Therefore a sliding window approach is used. The distance of the search pattern to the input time series is calculated at every data point which results in a distance vector. The problem of comparing two time series is not about locating exact matches, it is about finding approximate matches. This is done using distance measures (5). In this work, the two distance measures Euclidean distance and dynamic time warping are used. The input time series can be normalized and transformed to an alternative time series representation – e.g. symbolic representation. The found matches are marked in the input time series as shown in Figure 3.

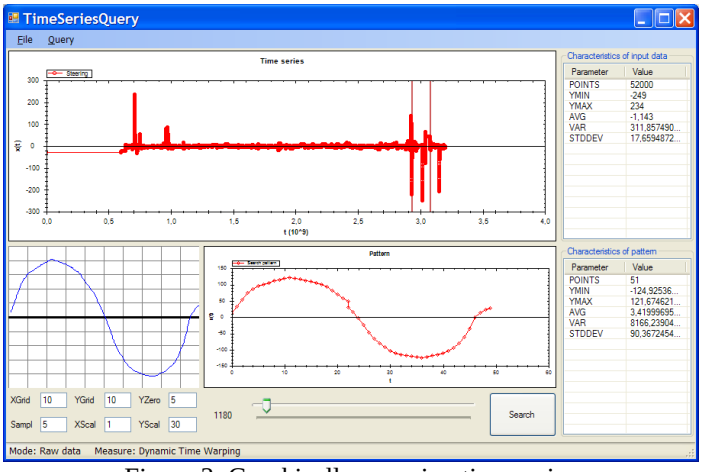

Figure 3: Graphically querying time series

Interaction between the techniques

The key to the interpretation of the recordings is linking the techniques and propagating selected data items between the visualizations. The integration of the proposed parallel

coordinates approach with a scatter plot matrix offers functionality like the identification of outliers inside a scatter plot. The outliers can be selected and highlighted in the corresponding parallel coordinates plot. Starting with the parallel coordinates, a scatter plot matrix can be visualized based for all included axes, for selected axes or only for the brushed data. This way, the benefits of those two visualization techniques can be used. The integration of the technique to graphically querying a time series together with the parallel coordinates and scatter plot matrix offers the additional benefit of being able to formulate very specific, shape-based search patterns on any of the time series shown in the enhanced parallel coordinates. The found matches are highlighted in the parallel coordinates or scatter plot matrix and can be further limited by e.g. applying the described Boolean brushing operations.

CASE STUDIES

The paper discusses the approach with the help of case studies based on data recorded during road trials with a series-production vehicle. The data has been manipulated for reasons of confidentiality. In the first case study, an anomaly was manually injected.

Case study 1: Timing behaviour

The first case study demonstrates how the proposed approach can be used to investigate timing behaviour concerning messages on the bus system. The ECUs inside the vehicle communicate by sending messages on the bus system. The majority of messages on the CAN bus is sent in a cyclic manner, following a pre-defined cycle time. Failure to meet the given timing requirements can result in failure of a vehicle function. On the individual ECUs, real-time operating systems are running. Scheduled tasks read signals, calculate a result and send the result on the bus. The CAN bus (15) is not deterministic, the message priority on the bus is based on the message id. If a message is required to be sent with a cycle time of, say, 10 ms the cycle time on the bus will not be exactly 10 ms, but is desired to be below an accepted deviation. A certain systematic jitter will be observable. In order for a vehicle function to work properly, the properties of the operating system task, the wall clock time of the algorithm running in the task and the message priority on the bus system need to be adjusted. An improper constellation of these parameters is one reason for erroneous deviations from a pre-defined cycle time. The deviation could depend on the operating point of an ECU, if the algorithms' computing times depend on the operating point, i.e. an algorithms wall clock time exceeds a pre-defined tolerance for certain input values. Additionally messages can be entirely lost. In order to detect lost messages, a message counter is transmitted.

The proposed approach was used to investigate the jitter and to detect deviations from the given timing constraints as follows: The analysis starts by displaying the data using the enhanced parallel coordinates visualization (Figure 4). The vector of timestamps is mapped to the first axis, labelled "Time". Two signals inside the message are mapped to the second and third axis ("Signal1", "Signal2"), the message counter to the fourth axis ("Counter"). The message counter is incremented for each message sent. In the given recording, the modulo operator was applied to the message counter by the ECU in order to limit the number of bits used on the bus, i.e. a correct sequence of values is 0,1, 2,…15, 0, 1.

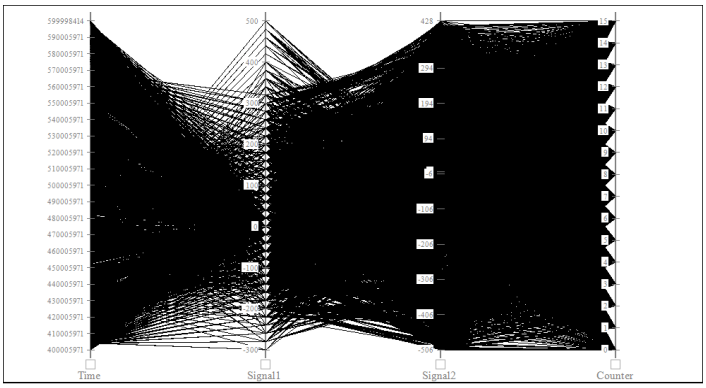

Figure 4: Parallel coordinates visualization

Without additional tools, no significant knowledge can be extracted from Figure 4. Following steps have been taken by the user resulting in Figure 5: The axis holding the message counter is duplicated and then differentiated. Lost messages could be detected at one glance by highlighting the portions of the data with values other than “1” and “-15” (“Counter_1’”). It can be seen from Figure 5 that there are no lost messages in the recording under investigation. The axis holding the timestamps is duplicated and then differentiated, which results in a vector holding the time differences between consecutive messages, the axis is labelled “Time_1’”. In a next step, the transparency value is increased in order to be able to distinguish value ranges with high and low frequencies. The dark regions show value ranges with high frequencies, while value ranges with low frequencies are displayed with light colouring. So, e.g. it can be seen, that the majority of messages have time differences of below 10250 µs.

Data items where the time difference between two consecutive messages is greater than 10500 µs, i.e. the deviation is greater than 500 µs are highlighted using a brushing operation on the second axis. The brushed data items are coloured following a linear colour gradient [green, yellow, red]. Two facts can be detected. First, the jitter seems to obey a certain period. This fact can be deduced from the five approximately equidistant green and yellow marked data items on the first axis, holding the vector of timestamps. Second, deviating from this detected period is an abnormal deviation from the cycle time by approximately 1 ms, which is highlighted with red colour. Following the data items highlighted with red colour shows that there is a possible dependency on the values of the two signals. Both signals have low values, when the deviation occurs.

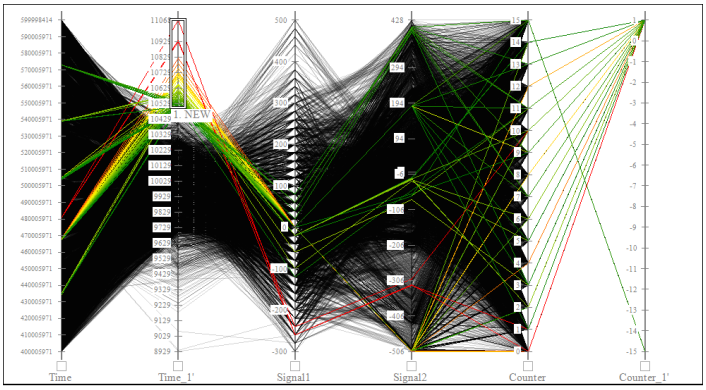

Figure 5: Enhanced parallel coordinates showing deviation from cycle time

Additionally, from the statistics integrated in the tool it can be seen that 0.46 % of the data are brushed, i.e. 0.46 % of all messages have a deviation from the cycle time of more than 500 µs.

The signals of interest were selected and displayed in the scatter plot matrix, where the highlighting of data is propagated.

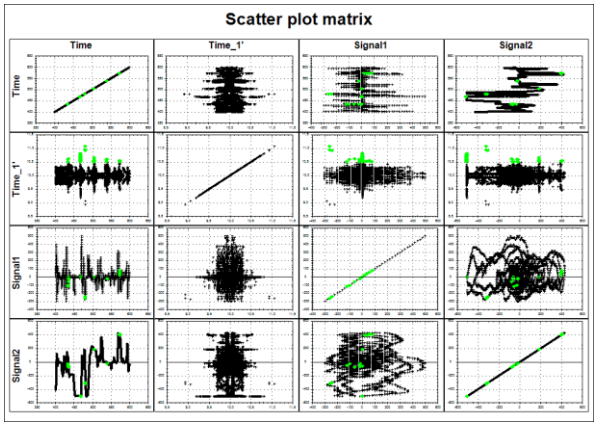

Figure 6: Scatter plot matrix

Case study: Relating different recordings

The previous case study worked on one individual recording. A challenge often encountered is the need to compare different recordings – e.g. from road trials. In this case study, the recordings of four road trials are compared. The four recordings are imported and an additional axis holding the number of the recording is generated. In order to get a quick overview, the data items are coloured following a colour gradient [blue, green, yellow, red], based on the axis holding the steering wheel angle.

The relation between steering wheel angle and the vehicle's yaw is usually anti-proportional, in other words the direction of the vehicle follows the steering wheel angle. When the vehicle is going through a left curve, the steering wheel angle is a negative integer, the yaw in turn is a positive integer. A deviation from this relation can be detected following some of the red lines connecting high values on the axes holding the steering wheel angle and the axis holding the yaw signal, as shown in Figure 7.

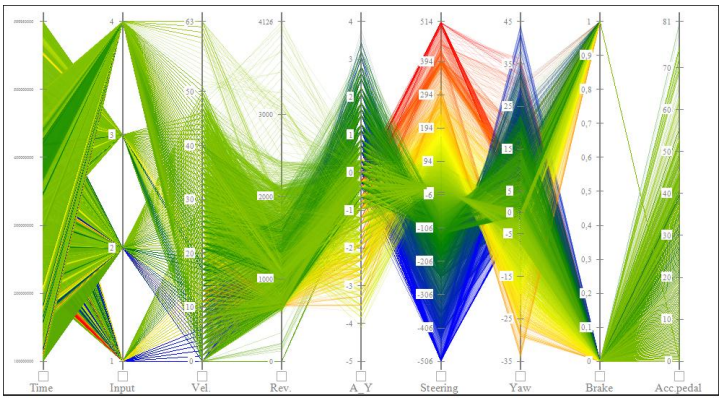

Figure 7: Colour gradient showing structure of the data

In order to focus on this anomaly, the data items are selected using the subsequent brushing operations:

- steering wheel angle right
- AND yaw rate of vehicle turning left
- AND NOT vehicle acceleration A_Y close to 0

From the resulting visualization shown in Figure 8 it can be seen in which of the road trials, the searched constellation is present, together with the point in time it occurred. Two occurrences were found. One of the road trials was driven on icy road conditions, so the

vehicle was sliding at that point in time. The second occurrence points to a situation where a steering maneouvre to the left was apruptly followed by a steering maneouvre to the right.

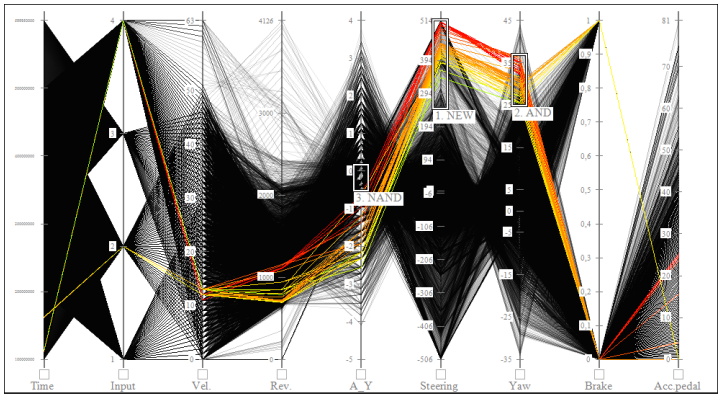

Figure 8: Query using Boolean operators

Case study: Query for driving manoeuvre

If specific patterns are known, e.g. profiles of one of the signals, the recordings can be queried as shown in this case study. In order to find all occurrences of right curves followed by left curves, the user graphically defines a search pattern based on the signal holding the steering wheel angle. The distance between the search pattern and the input time series “steering wheel angle” is calculated for each data point, which results in a distance vector. The user configures the allowable deviation from the search pattern by adjusting a threshold. This is done interactively using a slider or track bar in the user interface. All occurrences below the adjusted threshold are marked as matches and are highlighted in the two other visualizations as well (Figure 3).

CONCLUSION AND FUTURE WORK

The current approach already yields high effective and usable results. It integrates an enhanced version of parallel coordinates together with an enhanced scatter plot matrix and a time series query tool. Recordings from vehicle tests can be explored interactively as has been shown in the case studies. The approach allows for goal-oriented trouble-shooting as well as for exploration of recordings in order to find anomalies.

With the visualization techniques a number of fifteen to twenty signals can be related, depending on the number of data points and the size of the display. This is very helpful to investigate problems concerning subsystems or driver behaviour, where the signals are known by the user. Further visualization techniques are currently being evaluated to cope with a higher number of signals.

The existing approach will be applied on recordings resulting from different vehicle tests. An example is the analysis of measured data resulting from testing of modern driver assistance systems, where the input of multiple sensors is related using sensor fusion. This will be extended to include measured data of multiple vehicles with synchronized timestamps as proposed in (14). This is required when testing functions where interacting vehicles are involved, e.g. adaptive cruise control systems or in car-2-car environments.

The overall-goal of the research work is to develop a learning system that automatically detects anomalies (12) in recordings from vehicle tests. Therefore it will be investigated how the approach can be used to interactively discover rules to be stored in a knowledge base (16).

ACKNOWLEDGEMENTS

Many thanks for the support during implementation of the described solution go to Stephan Pressler, Daniel Hommel and Steffen Brauns.

REFERENCES

(1) Christoph Marscholik, Peter Subke. "Road vehicles – Diagnostic communication", Hüthig GmbH und Co. KG, 2008.

(2) Robert H. Shumway, David S. Stoffer. "Time Series Analysis and Its Applications: With R Examples", Springer Texts in Statistics. Springer Science+Business Media, 2nd edition, 2006.

(3) Usama Fayyad, Gregory Piatetsky-shapiro, Padhraic Smyth. "From data mining to knowledge discovery in databases". AI Magazine, 17:37–54, 1996.

(4) Cláudia M. Antunes, Arlindo L. Oliveira. "Temporal data mining: an overview". In KDD 2001 Workshop on Temporal Data Mining, pages 1 – 13, 2001.

(5) Eamonn Keogh, Selina Chu, David Hart , Michael Pazzani. "Segmenting Time Series: A Survey and Novel Approach". Data mining in Time Series Databases, 1993

(6) Ben Shneiderman. "The eyes have it: A task by data type taxonomy for information visualizations". In Proceedings of Visual Languages. IEEE Computer Science Press., pages 336–343, 1996.

(7) Maria C. Ferreira, Haim Levkowitz. "From visual data exploration to visual data mining: a survey", IEEE Transactions on Visualization and Computer Graphics, 9(3):378–394, 2003.

(8) Daniel A. Keim. "Visual exploration of large data sets", Communications of the ACM, 44:38–44, 2001.

(9) Edward J. Wegman. "Visual data mining". Center for Computational Statistics, George Mason University, 2003.

(10) Alfred Inselberg. "The Plane with Parallel Coordinates". Visual Computer Volume 1/ 1985, pages 69-91, 1985.

(11) Stephen Few. "Multivariate analysis using parallel coordinates". Perceptual Edge, 2006.

(12) Varun Chandola, Arindam Banerjee, Vipin Kumar. "Anomaly detection: A survey". ACM Computing Surveys, September 2009.

(13) Website "Tedradis by IT-Designers GmbH. The system for full-vehicle testing." www.tedradis.de, accessed: 23rd February 2010

(14) Daniel Ulmer, Andreas Theissler, Karsten Hünlich. "PC-Based Measuring and Test System for High-Precision Recording and In-The-Loop-Simulation of Driver Assistance Functions". Embedded World Conference, Germany 2010

(15) Konrad Etschberger. „Controller-Area-Network" 3. Auflage, Carl Hanser Verlag: 2004.

(16) Tom M. Mitchell. "Machine Learning". McGraw-Hill Education. 1997