Sonification and its application to the study of cyclic variability in combustion in internal combustion engines

R Tily and **C J Brace***
Department of Mechanical Engineering, University of Bath, Bath, UK

The manuscript was received on 29 January 2010 and was accepted after revision for publication on 25 May 2010.

DOI: 10.1243/09544070JAUTO1504

Abstract: Cyclic variability (CV) in combustion has been studied over many years and is still an active area of research. Despite the application of many analytical tools, including, in more recent years, those connected with the study of non-linear dynamic systems, CV remains a challenging subject. In the past, most researchers into CV have presented their findings in the form of overall statistical measures. However, it is felt that perhaps more attention needs to be paid to sequences of combustion cycles on a detailed cycle-to-cycle basis, particularly where the aim is to try to formulate a control mechanism to reduce CV, and that sonification combined with self-similarity plots provides a useful tool to facilitate such an approach. The purpose of this paper is to introduce sonification to automotive engineers, who may be unfamiliar with it, and to give an example of its implementation and application in the study of CV in internal combustion engines.

Keywords: sonification, cyclic variability, combustion, data symbolization, MATLAB, self-similarity plots

1 INTRODUCTION

During a recent research project concerned with studying cyclic variability (CV) in spark ignition (SI) engines with various fuelling levels, sonification was applied to time series data consisting of net heat release per combustion cycle values. These had been calculated using a method given by Stone [**1**], from in-cylinder pressure data recorded during tests with a 4.2 l, V8 direct-injection SI engine. It is believed that this is the first time that sonification has been used in an automotive application. During the research, literature on sonification was reviewed and a method of sonification incorporating data symbolization was devised. This paper is primarily concerned with describing that implementation and not with the CV research and its findings. Those seeking a comprehensive review of CV should consult the excellent literature reviews by Young [**2**] and Ozdor *et al.* [**3**].

**Corresponding author: Department of Mechanical Engineering, University of Bath, Claverton Down, Bath BA2 7AY, UK.*
email: enscjb@bath.ac.uk

In the following section, sonification is defined and the forms that it can take are discussed. This is followed by a brief review of some of the applications of sonification in various research fields. Next, a description is given of the way in which time series data related to engine combustion performance can be sonified, using MATLAB, and used in the analysis of CV. The use of sonified time series data in conjunction with self-similarity plots, a particularly effective strategy, is then discussed. Finally, conclusions are presented outlining some thoughts on the usefulness of sonification and its role alongside more traditional methods of data analysis.

2 SONIFICATION

2.1 Definition of sonification

Sonification is considered a subtype of auditory display, which in turn is broadly defined as any display that uses sound to convey information [**4**]. More precisely, sonification is defined as the use of non-speech audio to convey information [**5**]. A simple everyday example is a sound that accompanies a warning, such as a fire alarm or a computer error message. The aim here is to supplement and enhance the warning. For instance, a visual warning message may not be noticed by itself, but with the addition of sound its presence is announced more forcefully. Another example is the Geiger counter. A Geiger counter is an instrument that has a display that is often in the form of dial with a needle that shows the intensity of any ionizing radiation that may be present. To supplement the radiation reading, a clicking sound is added whose rate varies with the intensity of the radiation.

Sonification is a vast field with many forms and applications, and the reader seeking a general overview should consult references [**4**] to [**6**]. This paper is concerned with a particular branch of sonification, which is the sonification of data, particularly data in the form of a time series. Although the visual display of information or data is by far the most dominant form of display, in recent years, researchers in science, engineering and other fields have studied ways of using sound either to supplement visual displays or to act as a substitute for them. The use of sonification brings a number of advantages to the task of data analysis. However, there are also a number of disadvantages associated with sonification. Both advantages and disadvantages are listed below.

2.1.1 Advantages of auditory displays

The advantages of auditory displays are as follows:

(a) eyes-free application (which frees the mind of visual distractions or overload);
(b) high temporal resolution (i.e. the method is inherently temporal);
(c) alerting (which enables a quick response to a particular situation or datum value);
(d) backgrounding of tasks (which allows concentration to be focused on a current foreground task);
(e) rapid detection (i.e. the ability to display large amounts of data quickly enables equally quick detection of datum or patterns of interest);
(f) parallel listening (because humans have the ability to handle several voices simultaneously, i.e. the 'party effect');
(g) high dimensionality (i.e. the ability to handle multi-variate data);
(h) effective response (which enables an appropriate response to be made promptly for a particular situation).

2.1.2 Disadvantages of auditory displays

The disadvantages of auditory displays are as follows:

(a) low resolution of variables (as mapping sound to data can be restrictive);
(b) lack of absolute values (because a sound itself does not directly convey a datum's actual value);
(c) annoyance (some sounds can become annoying);
(d) absence of persistence (as sound is inherently temporal and temporary);
(e) no printed copy (although a musical sonification can provide a score);
(f) user limitations (i.e. user training may be required before full benefit is derived).

2.2 Sonification categories

There are a number of categories, and even subcategories, of sonification. There does not appear to be a general consensus on categorization, and each researcher's classification of such categories may differ from others. Leaving aside such categories as earcons and auditory icons [**7**] and considering the sonification of data only, there are three broad categories. These are audification, parameter mapping, and model-based sonification.

2.2.1 Audification

Audification, also sometimes referred to as continuous sonification, is the most direct method of presenting data in a sonified form. It can be applied to a time series and involves generating an audio signal directly from the series, which can later be played back. The time series itself is treated as a waveform, although it may have to be pre-processed, e.g. by time compressing [**8**], or frequency shifted so that it is audible to humans. Audification requires a large number of samples even for a short audification. Data derived from seismic measurements or dynamic system analysis can often provide such data. Automotive engine data can also provide such data. It was found that in-cylinder pressure data, recorded on a 1° of crank angle resolution basis, could be treated as a sound wave and played back directly. The resultant sound was not unlike the sound heard from an engine's exhaust system tailpipe. Where there is only a limited amount of data (i.e. a short time series), the data can be expanded by interpolating values between datum pairs.

2.2.2 Parameter-mapping sonification

The concept of parameter-mapping sonification comes from the familiar data representation offered by scatter plots. Here, data attributes are represented by symbol size, colour, shape, etc. The corresponding auditory display may represent data attributes by using different pitches, timbres, amplitudes, and durations. Although the parameter-mapping approach is flexible and relatively easy to implement, it does present problems such as the possible unpleasantness of the sounds generated. For example, if the amplitude is matched to individual datum values, unexpectedly large datum values within the data can result in unpleasantly loud sounds. There are also the problems that there is no standard mapping procedure and that there is the need to train users to interpret the sonification.

2.2.3 Model-based sonification

Model-based sonification was conceived by Hermann and Ritter [**7**]. The approach is quite radical. Instead of approaching the data directly, the data are used to construct an auditory model. The user then interacts with the model rather than directly with the data. The concept is based on the idea that information can be extracted from objects and their physical attributes and features ascertained by interacting with them. For example, striking or scraping an object and listening to the sound obtained can give cues to an object's surface texture (either smooth or rough) or size. If this approach seems far fetched, think of a common situation where someone receiving a present in a box will often shake the box in an attempt to guess what is inside.

Hermann's work is interesting, but there is some doubt on how effective the model-based sonification method is in practice. Herman provides some examples of sonified data sets at his website [**9**], such as Fisher's [**10**] iris flowers data set. Differences can be heard between the provided sound files related to the iris flowers data. However, detailed knowledge of their relationship to the data and experience listening to many examples would be needed before it was possible to differentiate between different subsets of data and to ascertain the properties of a particular set of data from the sound files alone. At present, it is not felt that the method of model-based sonification is appropriate for the sonification of time series related to combustion cycles in an internal combustion (IC) engine.

3 APPLICATIONS OF SONIFICATION

Although data sonification is a relatively new method, it has already been applied to various fields such as the natural sciences, medicine, and engineering, as the examples below illustrate.

Sonification has been applied to deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) sequences [**11**] with the aim of assisting in pattern matching and analysis. Hermann *et al.* [**12**] applied it to Markov chain Monte Carlo (MCMC) simulations used for sampling from probability distributions. MCMC simulations can be used in cluster analysis which aims to classify objects into different groups. Bernhardt *et al.* [**13**] investigated the employment of sonification in reconnaissance tasks involving hyperspectral image data, with the aim of helping users to reject false alarms generated by automatic detection methods. Biomedical imaging is also another image-related area considered by researchers. Nattkemper *et al.* [**14**] considered the application of sonification to micrographic images used in cellular biology.

In the field of medical research, several researchers, including Bettermann *et al.* [**15**, **16**], Ballora *et al.* [**17**], and Hermann *et al.* [**18**], have applied sonification to heart rate monitoring and diagnosis. Electrocardiogram (ECG) waveforms can be stored and later automatically processed by computer programs that detect QRS complexes (QRS refers to the types of wave, classified as Q waves, R waves, and S waves). QRS complexes trigger the mechanical contraction of the heart's pumping chambers (ventricles). The intervals between QRS complexes are sometimes referred to as RR (the time duration between two consecutive R-type waves of the ECG) or NN (the time intervals between consecutive normal beats) intervals. The fluctuation in the length of these intervals can be monitored and used in the diagnosis of a patient's heart condition. A time series of RR intervals can be sonified and mapped as a sequence of pitches which can then be used as an additional method in the detection of abnormalities. Ballora *et al.* [**17**] carried out such a mapping using the SuperCollider software to assign pitches to individual intervals, with shorter intervals corresponding to higher pitches.

From the field of natural sciences, seismology plays an important role, such as in the detection and recording of earthquakes and volcanic eruptions. Dombois [**19**, **20**] pointed out that there have been only a relatively small number of papers published on auditory seismology. However, as early as the 1960s, sonification and auditory displays, using seismic data, were acknowledged as being useful tools in the detection of both natural and man-made earthquakes. Speeth [**21**] investigated the application of sonification to seismic signals as he felt it may offer a solution to the problems that were being experienced in distinguishing between natural and man-made earthquakes. The man-made earthquakes were due to underground explosions, such as those made during the testing of a nuclear bomb, which the Americans were particularly interested in detecting during the Cold War years of the 1950s and 1960s. Speeth carried out a series of experiments using audio tapes that had been produced from data obtained from seismometers, which had been digitalized, processed, and then converted, via a digital-to-analogue converter, into an audio signal which was then recorded onto tape. After some initial training, observers listening to the tapes were able to distinguish between natural earthquakes and those caused by man-made explosions in 90 per cent of the cases presented to them. However, similar work in 1965 by Frantti and Leverrault (as cited by Dombois [**20**]) reported only a 67 per cent success rate.

In more recent years, others have also investigated the use of sonified seismic data. Vicinanza [**22**] mapped data from seismograms obtained from Mount Etna in Sicily, to musical notes in Musical Instrument Digital Interface (MIDI) form, as an alternative to the usual waveform construction approach. Saue and Kjeld [**23**] also used the MIDI format when carrying out a pilot study on the use of sonification techniques for oil exploration. They felt that mapping seismic data to a sequence of MIDI notes was more appropriate than using waveforms, as typical seismic traces, obtained in oil exploration, last only a few seconds. Dombois [**20**] assessed various forms of audified seismic signals. The general conclusion drawn was that, although it was sometimes possible to distinguish between certain forms of signal, it did require practice to become accustomed to audified seismic events. It remains to be seen whether the sonification of seismic data makes much impact in the future of seismology, particularly in the prediction of earthquakes, which still remains a challenge. These findings suggest that it may also be a challenge to interpret an IC engine's sonified combustion cycle sequences in a meaningful way too.

4 IMPLEMENTATION

During the research, a number of user- and developer-based sound, music, and sonification software packages were reviewed. Although the sound and music development software packages reviewed could all be used to develop time series data sonification software, none appears to be ideally suited to the task, as they are primarily aimed at musical composition and sound synthesis. Surprisingly, in the literature, many researchers using sonification failed to report how they actually implemented it and which software tools they used. However, several refer to SuperCollider [**17**, **24**, **25**] which appears to be the most popular, compared with Csound, which was used by Ramaswamy *et al.* [**26**], and jMusic which was used by Vicinanza [**22**].

Because of previous experience in using MATLAB, and the fact that it is a very versatile data-processing and mathematical software package, it was decided to investigate the possibility of using MATLAB in the data-processing, sonification, and analysis aspects of the research. Apart from the useful toolboxes that either come with MATLAB or can be purchased separately, the freely available sound-, audio-, and music-based MATLAB toolboxes such as the MIDI Toolbox [**27**], MIR Toolbox 1.0 [**28**], and Auditory Toolbox 2.0 [**29**] contributed to the decision to use MATLAB as the main data-processing tool.

The availability of the MIDI Toolbox contributed to the decision to implement data sonification by mapping a time series' individual datum values to musical note values as represented by the MIDI system, which is an industry-standard protocol. Once a time series is sonified using this method, it can be played back and/or viewed as a musical score. It is felt that using a musical approach to sonification, together with familiar musical instrument sounds such as the piano, will result in an increase in a listener's ability to detect patterns within the time series. Matta *et al.* [**30**] suggested that listeners remember musical sounds and sequences better than more abstract sounds. Another advantage of using musical notes is that they can sound more pleasant than the alternative of mapping individual datum values to abstract sounds.

The MIDI standard defines 128 note values, giving a range of ten octaves and eight additional notes. This is satisfactory for the representation of Western music and is within the normal range of pitches that humans can hear. However, it does present a restriction with regard to the mapping of a times series of widely varying values to the alternative representation of MIDI note values. It also has to be borne in mind that those 128 notes include sharpened or flattened notes too. When those notes are excluded, there are only 75 natural notes left. Sharp and flat notes could be included in the sonification process, but this causes problems in that it is easier to detect differences between natural notes (two semitones apart, except between notes B and C and E and F) than between natural notes and sharpened or flattened notes (a difference of just one semitone).

Also, as a musical score produced by the sonification process also visually represents the structure and pattern of a time series under analysis, it is better to avoid using sharpened or flattened notes (accidentals) as they will occupy the same stave line as natural notes.

Although there are only 75 natural notes available for mapping, this restriction did not prove to be a problem in practice with the method used for sonification. Figure 1 shows a diagram of the sonification process used during the research.

The heat release time series data are first symbolized [**31**] before being sonified. There are, at least, a couple of approaches to the symbolization of a time series that can be taken with the aim of subsequently mapping to MIDI note values.

The first is to partition the data into multiple bands and then to map individual note values to each band so that the low notes correspond to low bands, and the higher notes to the higher bands. In net heat release value terms, this would cause the MIDI note pitches to reflect directly the magnitude of the individual heat release values. High pitches correspond to large heat release values, and low pitches to low heat release values. The partitioning itself can be achieved by simply dividing the time series values into a number of equally sized steps between the lowest and highest values. A more sophisticated method is to divide the time series values on a percentile basis. For example, if it is decided to partition the series into ten individual partitions (with each partition being assigned a musical note), the MATLAB's prctile function can be used according to

$$Y = \mathrm{prctile}(X, p) \tag{1}$$

where X is the time series and p is the percentage. For example, prctile(X, 50) returns the value at the middle (the median) of the range of the time series. It is then possible to issue a series of prctile commands, prctile(X, 0), prctile(X, 10), …, prctile(X, 90), prctile(X, 100), to define the start and the end of each partition.

The second approach, which was the approach used during the research, is to carry out a binary partition of the time series so that there are only two bands. The data's median value can be used to ensure that there is an equal probability that an individual datum value will be on either side of the partition. Each datum value is assigned a value of zero or one depending on whether it is below or above the median value. The data series will now consist of a binary string of ones and zeros. A binary partition by itself is obviously too coarse to reflect the variations between combustion cycles in any detail. Therefore the binary string is broken down into a series of binary words of length L. A moving window of length L is passed along the binary string, one binary digit at a time, so generating a series of binary words that can be converted into decimal values that can be mapped to MIDI note values. This type of binary partition approach is the same as used in symbol analysis, as described and used by Daw *et al.* [**32**, **33**]. Each binary word represents a sequence of combustion events and not an individual event.

Once the binary words, and hence their corresponding decimal values, have been generated from the symbolization of a net heat release per combustion cycle time series, the binary word values can be mapped to MIDI note values, and a sound file and musical score created. The actual mapping of word values to MIDI notes was performed using MATLAB,

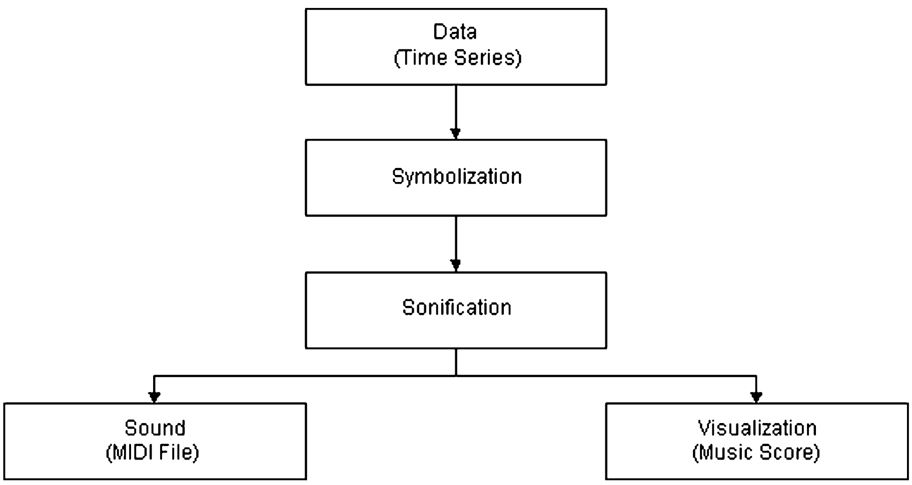

Fig. 1 The sonification process

as was the symbolization process itself. For example, the initial line in the MATLAB program SoundSymbolSequence.m, which is

$$\begin{aligned}\text{function } [\text{nm, hdata}] = \text{SoundSymbolSequence}\\ (\text{dataValues}, L, \text{useMedian},\\ \text{basePitch, maxNotes})\end{aligned}$$

helps to illustrate the approach taken. The function's arguments are as follows:

dataValues	a MATLAB vector or data file containing the actual time series' values
L	binary word length
useMedian	use the data's median value as the dividing value for the binary partition, else use (max(dataValues) – min(dataValues))/2
basePitch	the lowest MIDI note value to use (e.g. 60 which corresponds to middle C or C4)
maxNotes	the number of notes to generate (which corresponds to the number of combustion cycle values to process)

The function returns are as follows:

nm	a MIDI Toolbox-compatible note matrix
hdata	the actual binary or decimal word values themselves, which could be used to form a histogram or could be processed later

The note matrix is a MATLAB matrix that can be used in the MIDI Toolbox [**27**] software. An example of a note matrix is

$$\text{nm} = \begin{bmatrix} 0 & 0.9 & 1 & 60 & 80 & 0 & 0.55; \\ 1 & 0.9 & 1 & 62 & 80 & 0.61 & 0.55; \\ 2 & 0.9 & 1 & 64 & 80 & 1.22 & 0.55; \\ 3 & 0.9 & 1 & 65 & 80 & 1.83 & 0.55; \\ 4 & 0.9 & 1 & 67 & 80 & 2.44 & 0.55; \\ 5 & 0.9 & 1 & 69 & 80 & 3.05 & 0.55; \\ 6 & 0.9 & 1 & 71 & 80 & 3.66 & 0.55; \\ 7 & 0.9 & 1 & 72 & 80 & 4.27 & 0.55 \end{bmatrix}$$

This matrix defines eight notes, in the key of C major, in the fourth octave to the fifth octave: C4, D4, E4, F4, G4, A4, B4, and C5, as shown in Fig. 2

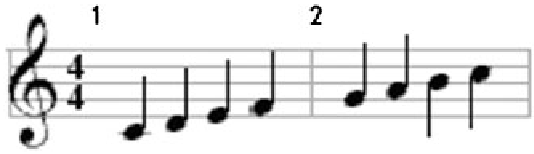

Fig. 2 One octave (C4 to C5) in C major. These notes correspond to those defined in the note matrix nm above

The columns in the note matrix nm are the following from left to right:

(a) onset (beat number);
(b) duration (in beats);
(c) MIDI channel number (1 to 16);
(d) MIDI pitch (i.e. note);
(e) velocity (how hard the note is 'hit', which corresponds to its volume);
(f) onset (in seconds);
(g) duration (in seconds).

The note matrix can be used as an argument to MIDI Toolbox functions such as playmidi and writemidi. writemidi was used to create MIDI files which could then be played or loaded into programs such as MagicScore Maestro 5.0 [**34**] which can display MIDI files in musical notation and play them in a variety of tempos.

During the research, there was some concern that the binary partition and binary word approach to the net heat release time series symbolization may not be appropriate and does not reflect the actual variation between combustion cycles. An investigation proved that this was not the case.

Figure 3(a) shows the first 100 net heat release values for cylinder 1 of the 4.2 l, V8 engine running at 2000 r/min with a 2 bar load and an air-to-fuel (A/F) ratio of 14.50. The bold solid line running across the plot is the median value, and a binary partition using this line results in the following binary string: 11100111100111110001101000011101011010000101 00111100001010000111000011010001101000110010 111110011100. Figure 3(b) shows the corresponding MIDI note numbers that have been obtained via the binary partition of the data and binary word length $L = 3$ (2^3 equals the maximum of eight possible note values). Note that there are only 96 notes, and not 100, because of the use of a sliding window of length $L = 3$. It can been seen from Fig. 3 that the MIDI notes (and hence a corresponding musical score) do, in fact, reflect the structure or pattern of the original heat release data. The sequence of musical notes rises and falls in time with the original heat release values. It is

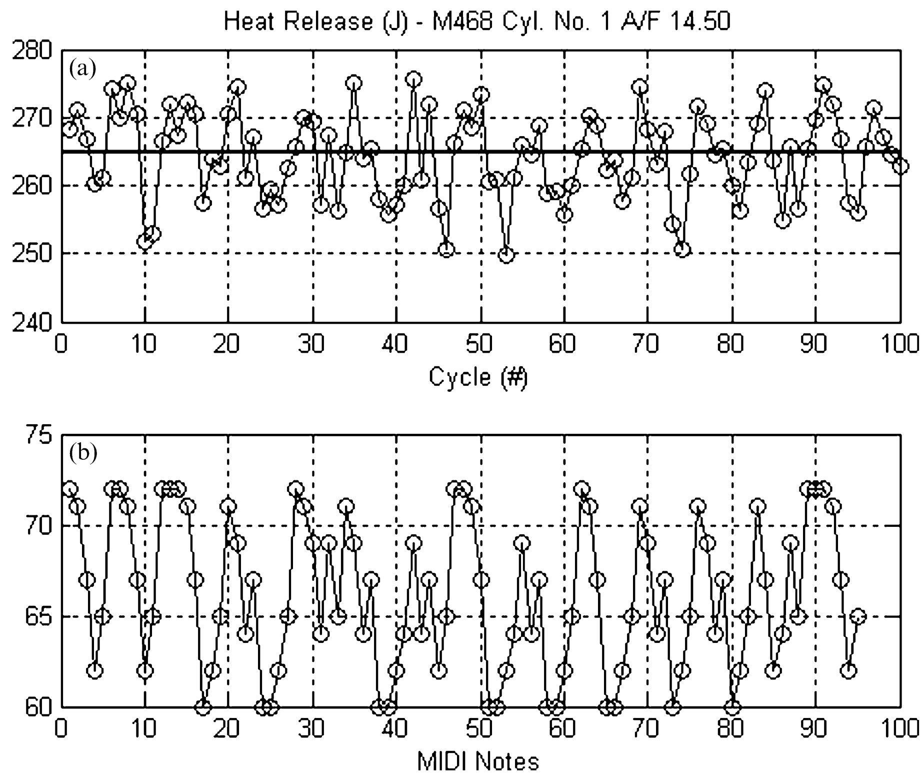

Fig. 3 (a) The first 100 net heat release values for cylinder 1 of the 4.2 l, V8 engine running at 2000 r/min, with a 2 bar load, at a stoichiometric A/F ratio of 14.50; (b) corresponding MIDI note values generated by the symbolization and MIDI note mapping program SoundSymbolSequence.m using a binary word length $L = 3$

also possible to see that some smoothing has taken place. This is a result of using a sliding window of length $L = 3$ and moving it forwards one binary digit at a time, so that the previous binary values play a part in the new value. The sliding window acts in a similar manner to a moving-average process. It should be noted that this sliding-window approach is the same as described in the review by Daw *et al.* [**31**] of symbolic analysis of experimental data, which clearly shows this method being used.

The first three points in Fig. 3(a) are all above the median, which corresponds to the binary 111; this, in turn, corresponds to 7, which when mapped to a MIDI note corresponds to note C5 (C, fifth octave). The next binary sequence is 110 (6) which is mapped to note B, and the third word is 100 (4) which is mapped to note G (below the other two notes). Therefore those first three notes are actually a smoothed version of the first three heat release values. Note that the small peak that the second value forms, in the original heat release data, has disappeared in the smoothed (coarse-grained) version, but that the basic pattern remains the same. Figure 4 is similar to Fig. 3, but here a word length $L = 4$ (2^4 equals the maximum of 16 possible notes) is being used. If anything, the graining is coarser, but the pattern of MIDI notes still follows that of the original heat release data. Rising note sequences are matched to corresponding rising heat-release value sequences, and falling sequences of notes are matched to falling sequences of individual combustion cycle heat-release values. Figure 5 shows the corresponding musical score.

Another concern with regard to time series symbolization and MIDI note mapping was the optimum choice of binary word length. Using a binary partition, experiments carried out revealed that the optimum word or sequence length is four. Although using a word length $L = 3$, which corresponds to eight musical notes, provides a clear visual and auditory pattern, it is not quite so clear as that provided by using a word length $L = 4$, which corresponds to 16 notes.

5 SONIFICATION AND SELF-SIMILARITY MATRICES AND PLOTS

Once the sonification process was defined and implemented, a series of tests was carried out to

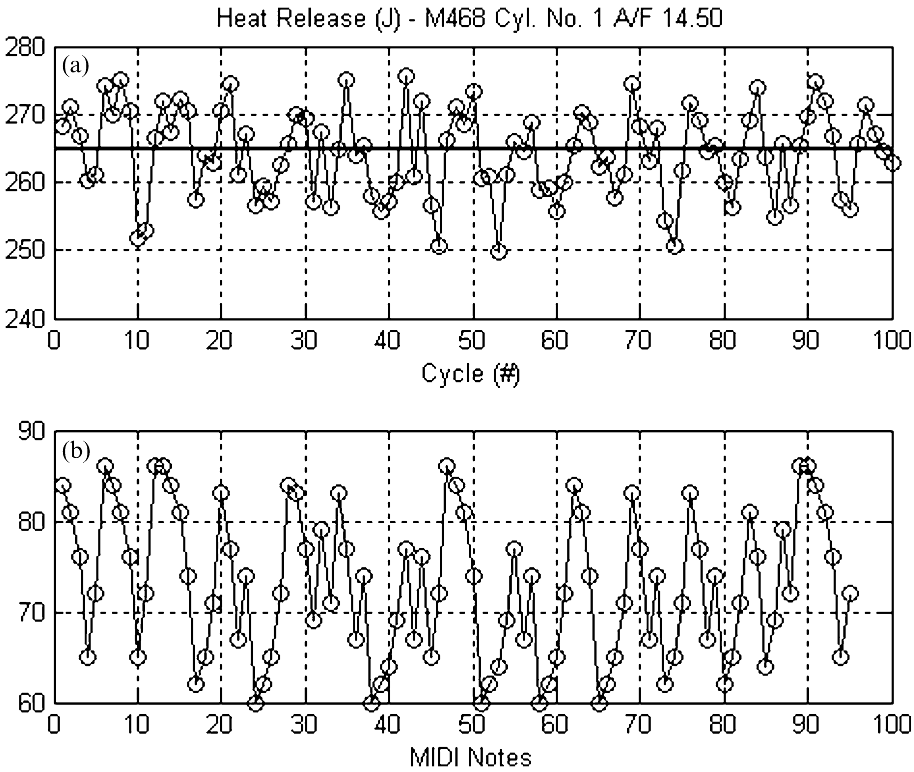

Fig. 4 (a) The first 100 net heat release values for cylinder 1 of the 4.2 l, V8 engine running at 2000 r/min, with a 2 bar load, at a stoichiometric A/F ratio of 14.50; (b) the corresponding MIDI note values generated by the symbolization and MIDI note mapping program SoundSymbolSequence.m using a binary word length $L = 4$

see how effective sonification was in differentiating between time series consisting of net heat release values for series of consecutive combustion events at various fuelling levels. It was found that it was not possible to differentiate between different sonified combustion sequences purely by listening to the sonified data. For instance, it was not possible to differentiate clearly between combustion cycle sequences for a cylinder for each of the four fuelling levels used in the tests. Sonification of combustion sequences, in the form of heat release values, does appear to be a worthwhile exercise, as it provides an additional dimension (sound) and helps the user to obtain a better feel for the data. However, its greatest strength is when it is used in conjunction with traditional visual representations of time series data. One such visual representation that was found to be particularly effective when used with a sonified time series was the self-similarity matrix and its corresponding plot. Self-similarity matrices have been

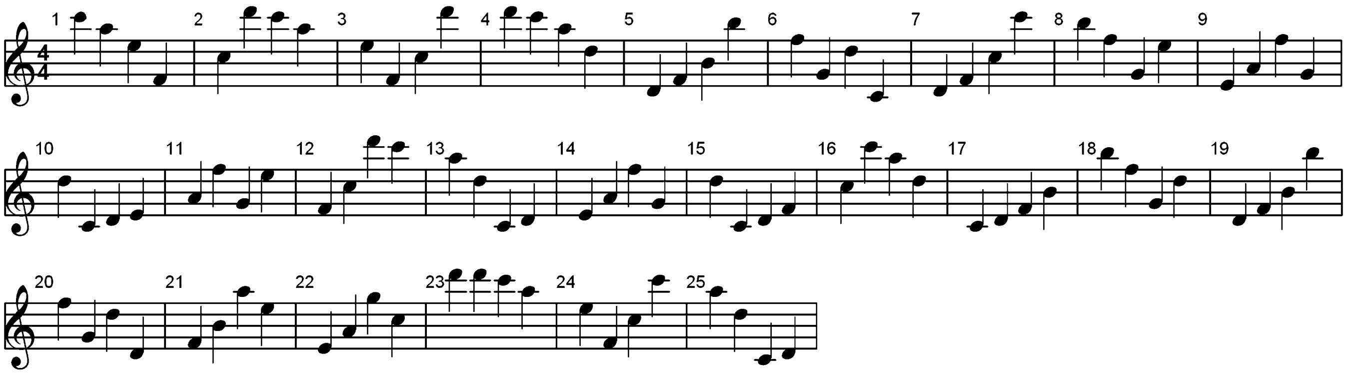

Fig. 5 The musical score that corresponds to the heat release time series and its corresponding MIDI note sequence shown in Fig. 4

used by researchers mostly with music in an analogue form [**35**, **36**]. However, Foote and Cooper [**37**] used self-similarity matrices with MIDI music directly. Self-similarity matrices are closely related to recurrence plots [**38**].

A self-similarity matrix is formed by comparing a particular measure at a point at the instants *i* and *j*. In the case of using a self-similarity matrix with music in MIDI form, a simple black-and-white matrix can be constructed, where matrix entries are white if note *i* and note *j* are of the same pitch or black if they are not equal [**37**]. Figure 6 shows part of such a matrix together with the corresponding piece of music.

The diagonal pattern shown in Fig. 6 results from the fact that there is the same number of steps between notes of the same pitch. For instance, there are seven steps (counting from note 1) between note 1 and note 7, which is of the same pitch. There are also seven steps between note 2 and note 8. There are seven steps between the other notes that share the same pitch. Translating this to CV, where the musical notes represent single or multiple combustion events (depending on the treatment of the data) represented by net heat release per cycle values, a pattern of repeating combustion events, of similar magnitudes, can be detected. Another common pattern found in self-similarity matrices is a chequerboard pattern. Figure 7 shows an example. The chequerboard pattern is the result of repeated alternating notes. The notes A4 (note A in the fourth octave) and F5 (note F in the fifth octave) alternate between beats 4 and 9. A third type of pattern found in self-similarity

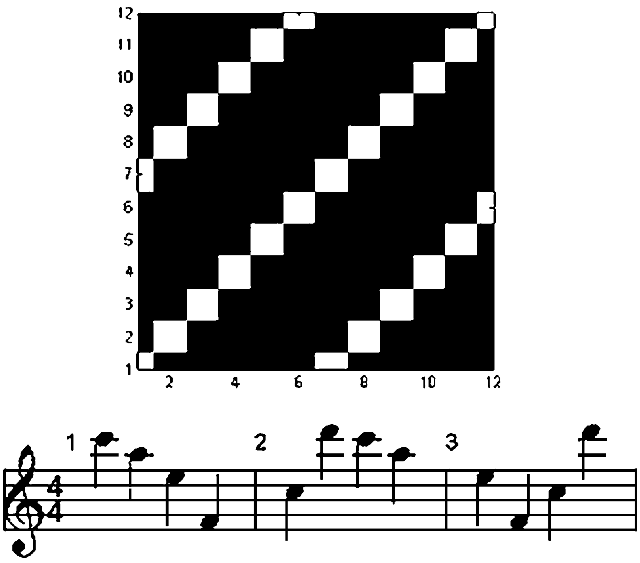

Fig. 6 A self-similarity matrix created from a MIDI file, with the corresponding music shown below it

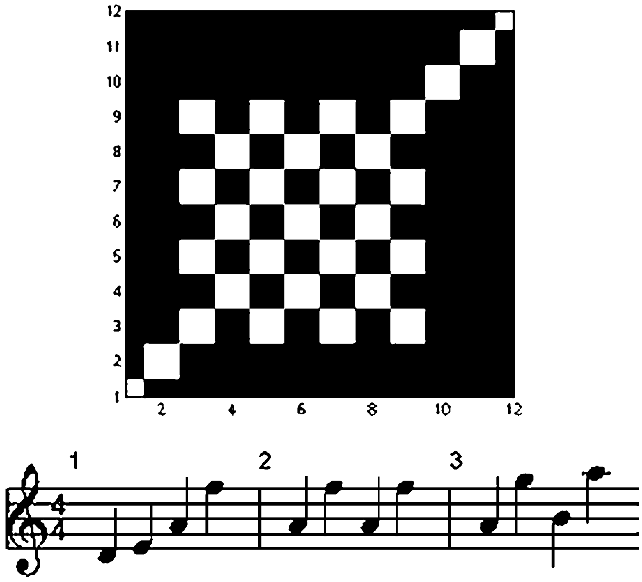

Fig. 7 An example of a chequerboard pattern in a self-similarity matrix, with the corresponding music shown below it

matrices is a solid block pattern. This is caused by repeated notes of the same value. Figure 8 shows an example. Here, middle C (C4, note C in the fourth octave) is repeated from beats 2 to 5. The final patterns are single vertical or horizontal lines. These will occur if a particular note value matches those of a sequence of repeating notes of the same value. An example of this can be seen in Fig. 9. Here the middle C note at the start of bar 5 matches the repeated sequence of the same note across bars 1 and 2 and corresponds to beats 2 to 5.

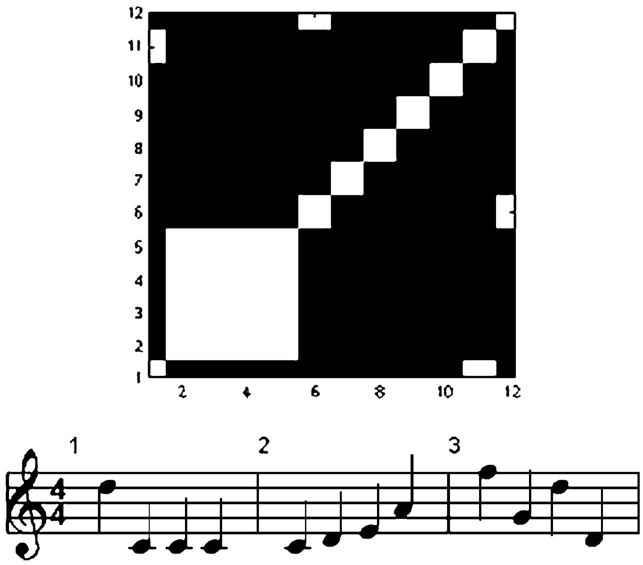

Fig. 8 An example of a block pattern in a self-similarity matrix, with the corresponding music shown below it

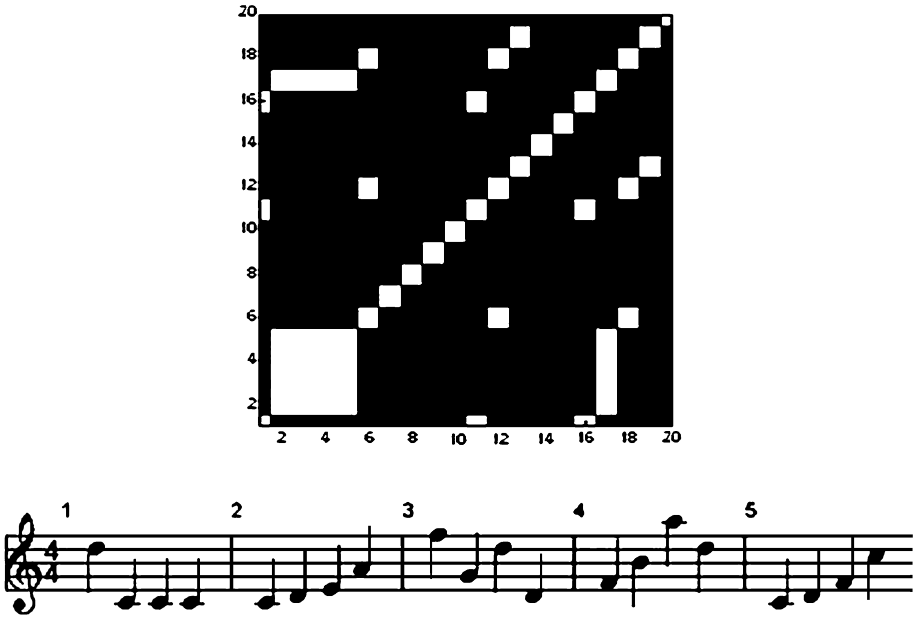

Fig. 9 An example of a block pattern and a horizontal and vertical line in a self-similarity matrix, with the corresponding music shown below it

As will be seen later, diagonal lines, chequerboards, solid rectangular blocks, and vertical or horizontal lines are easy to spot within self-similarity matrix plots and can be used to pinpoint certain types of combustion event behaviour within a time series using the axis markings of the plots.

As part of the analysis of CV, MATLAB programs were developed to calculate self-similarity matrices. The input data were MIDI files produced during the symbolization and sonification of heat release data for the 4.2 l, V8 engine, for all four cylinders and fuelling levels. Figure 10 shows the self-similarity matrices corresponding to MIDI files created using a binary word length $L=4$ in the symbolization process. The self-similarity matrices were created using the method, previously described above, where matrix entries are white if note i and note j are of the same pitch or black if they are not equal. Note that the self-similarity matrices shown in Fig. 10 are for only the first 100 MIDI notes. This has been done to illustrate the method in a clearer manner than would be possible using a much larger number of note values. All self-similarity matrices are symmetrical around the main central diagonal (running from the bottom left-hand corner to the top right-hand corner).

Figure 10 shows the self-similarity matrices corresponding to cylinder 1 of the right-hand bank of the Audi 4.2 l, V8 engine, running at 2000 r/min with a 2 bar load, at the usual fuelling levels used throughout the 4.2 l, V8 engine tests for A/F ratios of 14.50, 8.05, 19.84, and 20.47. The self-similarity matrices for A/F ratios 14.50 and 18.05 consist primarily of diagonal lines. There are a few very small rectangular blocks under the A/F ratio 18.05, indicating some short repetitions of the same note (or combustion sequences). It is also noticeable that, as the fuelling becomes leaner, the patterns in the self-similarity matrices become more sparse. This indicates that there are fewer matches between notes, and the corresponding combustion sequences, under leaner fuelling. Combustion event behaviour appears to become more random as repeatable behaviour lessens.

Figure 11 shows the corresponding musical scores related to the self-similarity matrices shown in Fig. 10. It can be seen that all the scores share the same basic pattern of a rising and falling, or vice versa, sequence of notes. However, it can also be observed that there is a gradual break-up of the overall rising and falling pattern as the fuelling becomes increasingly lean. This is particularly noticeable in the score for an A/F ratio of 20.47, where there are a number of bars that contain alternating high and low notes, which, in turn, correspond to high- and low-magnitude combustion events.

As the above examples have shown, the self-similarity matrix is a useful tool for examining any time series. Although they are equivalent to a recurrence plot with a threshold set so that only exact matches are recorded, they are generally easier

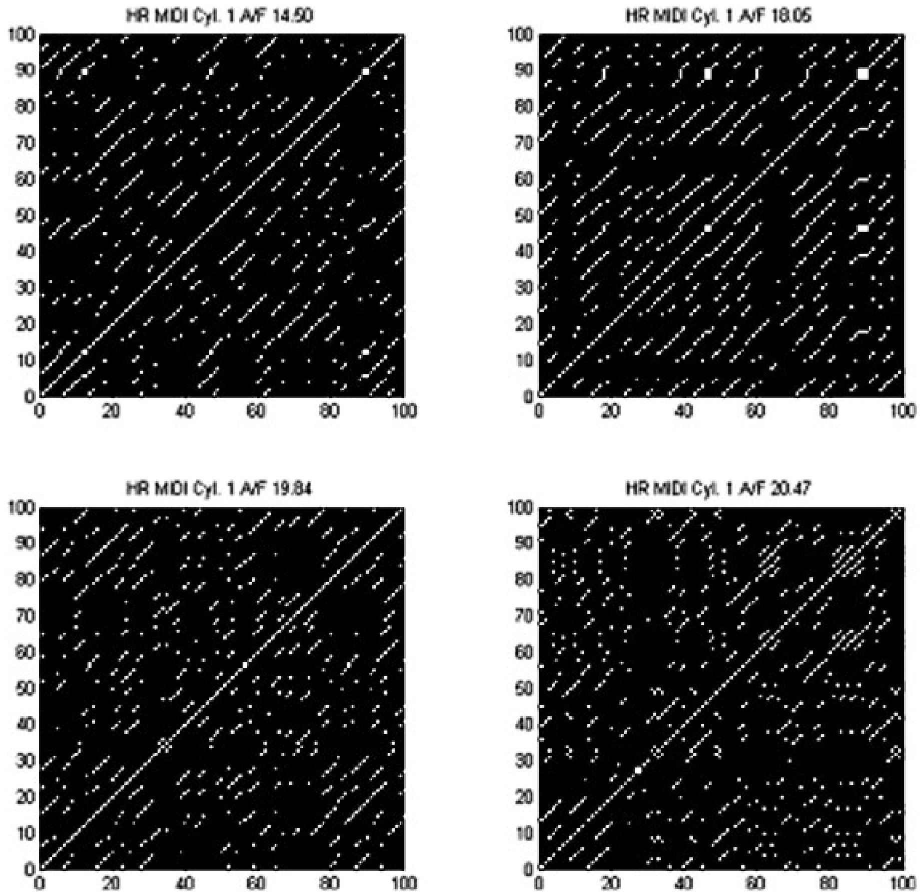

Fig. 10 Self-similarity matrices for cylinder 1 of the 4.2 l, V8 engine, running at 2000 r/min and 2 bar load, at A/F ratios of 14.50, 18.05, 19.84, and 20.47

to use than recurrence plots, particularly when compared with those that use multiple threshold values. Using a tool, such as MATLAB, which enables the user to zoom in and out and to scroll across a plot of a self-similarity matrix, it is quite easy to pick out features that represent various types of behaviour. A summary of those features is shown in Table 1.

6 CONCLUSIONS

It appears that sonification alone does not provide an easy and effective way to analyse sequences of combustion events, particularly when comparing sequences corresponding to different fuelling levels. This is due to the difficulty in aurally detecting differences in the characteristics of various combustion cycle sequences. The sonified data, in the form of a sequence of musical (MIDI) notes, is of a temporal nature, and the difficulty of remembering how a long series of notes sounded, particularly when trying to compare it with a sequence that is currently being listened to, makes the method ineffectual. It has to be borne in mind that a sonified sequence of combustion events, although musical in nature, does not have the characteristics of a typical piece of popular music. That is, it does not have strong memorable phrases, nor does it repeat certain sections exactly, such as in a chorus, which is common in popular music. However, it needs to be noted that the sonified combustion sequences do have a characteristic sonic pattern, generally consisting of rising and falling notes, which becomes more evident with repeated listening. Although this sonic pattern was found to be similar throughout the various combustion sequences, related to the different fuelling levels, this made the task of differentiating between different sequences difficult by aural means alone. Even though sonification alone does not prove to be very effective, it has been found to be a useful addition to the more usual visual approach to data analysis. The combination of the sounds of MIDI notes and a visual representation of a time series' values as a musical score has proved to be a very effective way of discerning patterns within a

Fig. 11 The musical scores related to the self-similarity plots shown in Fig. 10

Table 1 Features found within self-similarity matrices together with the corresponding behaviours. In the case of this study, the behaviour is also related to sequences of combustion events

Feature (pattern)	Behaviour
Diagonal lines	Repeating sequences of notes (equal intervals between notes)
Chequerboard	Repeating pattern of two alternating notes
Solid rectangular block	Repeating notes of the same value
Single vertical or horizontal line	A single note matches those of a sequence of consecutive notes of the same value elsewhere in the series
Regularly spaced broken vertical or horizontal lines	A single repeating, equally spaced value
Blank empty areas or blocks	No correlation between notes corresponding to the area

sequence of combustion events. It is also felt that the musical scores alone are a useful additional visual aid and help to observe the process of a variable over time and to pick out particular patterns. The musical score's horizontal stave lines help to define clearly the position and relative value of the variable under study. It has also been found that a musical score provides a relatively compact representation of a lengthy time series. However, either when looking at a traditional time series plot, or particularly with the equivalent musical score (which in itself is a by-product of the sonification process employed here), the act of simultaneously listening to a corresponding sound does help the listener to become more aware of the various patterns or anomalies present in a time series, including those examined during this study of CV in combustion. As an automotive analogy, take the example of how the addition of sound helps an automotive technician to detect a misfiring and vibrating engine. The sound reinforces the perception that the visual vibration is being observed. In a similar manner, data sonification acts as an additional sense and, because of the relative ease with which sonification can be implemented, it is felt worth considering.

Even though CV has received a great deal of attention by automotive researchers, it still remains an active and relevant area of research. One interesting fact that emerged during a literature review and this present research is that researchers have usually evaluated and presented their findings in terms of overall measures (e.g. statistical measures). They rarely present CV in combustion on a cycle-to-cycle basis. It is felt that perhaps more attention needs to be paid to sequences of combustion cycles on a cycle-to-cycle basis, and that sonification combined with self-similarity plots provides a useful tool to facilitate such an approach. These tools could prove useful to automotive researchers, including those investigating CV in new types of engine such as the homogeneous charge compression ignition engine.

REFERENCES

1 **Stone, R.** *Introduction to internal combustion engines,* 2nd edition, 1992 (Macmillan, London).

2 **Young, M. B.** Cyclic dispersion in the homogeneous-charge spark-ignition engine – a literature survey. SAE technical paper 810020, 1981.

3 **Ozdor, N., Dulger, M.,** and **Sher, E.** Cyclic variability in spark ignition engines: a literature survey. SAE technical paper 9409987, 1994.

4 **Walker, B. N.** and **Nees, M. A.** Theory of sonification. In *Handbook of sonification* (Eds T. Hermann, A. Hunt, and J. Neuhoff (Academic Press, New York) (in press).

5 **Kramer, G., Walker, B., Bonebright, T., Cook, P., Flowers, J., Miner, N., Neuhoff, J., Bargar, R., Barrass, S., Berger, J., Evreinov, G., Fitch, W., Gröhn, M., Handel, S., Kaper, H., Levkowitz, H., Lodha, S., Shinn-Cunningham, B., Simoni, M.,** and **Tipei, S.** The sonification report: status of the field and research agenda. Report prepared for the National Science Foundation, International Community for Auditory Display, Santa Fe, New Mexico, USA, 1999.

6 **Barrass, S.** and **Kramer, G.** Using sonification. *Multimedia Systems*, 1999, **7**(1), 23–32.

7 **Hermann, T.** and **Ritter, H.** Listen to your data: model-based sonification for data analysis. In *Advances in intelligent computing and multimedia systems* (Ed. M. R. Syed), 1999, pp. 189–194 (International Institute of Advanced Studies in System Research and Cybernetics, Tecumseh, Ontario).

8 **Barry, A.** Efficient listening with two ears: dichotic time compression and spatialization. In Proceedings of the Second International Conference on *Auditory Display* (*ICAD'94*), Santa Fe, New Mexico, USA, 7–9 November 1994, pp. 171–177 (International Community for Auditory Display, Santa Fe, New Mexico).

9 **Hermann, T.** Sonification.de: Thomas Hermann's research on sonification, data mining and ambient intelligence, 2006 available from http://www.sonification.de (accessed 12 November 2009).

10 **Fisher, R. A.** The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 1936, **7**, 179–188.

11 **Cullen, C.** and **Coyle, E.** Rhythmic parsing of sonified DNA and RNA sequences. In Proceedings

of the Irish Signals and Systems Conference (*ISSC 2003*), Limerick, Ireland, 1–2 July 2003, pp. 220–223.
12 **Hermann, T., Hansen, M. H.,** and **Ritter, H.** Sonification of Markov chain Monte Carlo simulations. In Proceedings of the Seventh International Conference on *Auditory display* (*ICAD 2001*), Espoo, Finland, 29 July–1 August 2001, pp. 208–216 (Laboratory of Acoustics and Audio Signal Processing and the Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, Espoo).
13 **Bernhardt, M., Cowell, C. E.,** and **Oxford, W. J.** Sonification of hyperspectral image data. In *Algorithms and technologies for multispectral, hyperspectral, and ultraspectral imagery XIII* (Eds S. S. Shen and P. E. Lewis), Proceedings of the SPIE, vol. 6565, 2007, p. 65650D (SPIE, Bellingham, Washington).
14 **Nattkemper, T. W., Hermann, T., Schubert, W.,** and **Ritter, H.** Look and listen: sonification and visualization of multiparameter micrographs. In Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Cancum, Mexico, 17–21 September 2003, pp. 1311–1314 (IEEE, New York).
15 **Bettermann, H., Amponsah, D., Cysarz, D.,** and **van Leeuwan, P.** Musical rhythms in heart period dynamics: a cross-cultural and interdisciplinary approach to cardiac rhythms. *Am. J. Phys., Heart Circulatory Physiol.*, 1999, **277**, 1762–1770.
16 **Bettermann, H., Amponsah, D., Cysarz, D.,** and **van Leeuwan, P.** Detecting cardiorespiratory coordination by respiratory pattern analysis of heart period dynamics – the musical rhythm approach. *Int. J. Bifurcation Chaos*, 2000, **10**(10), 2349–2360.
17 **Ballora, M., Pennycook, B., Plamen, C. I., Glass, L.,** and **Goldberger, A. L.** Heart rate sonification: a new approach to medical diagnosis. *Leonardo*, 2004, **37**(1), 41–46.
18 **Hermann, T., Baier, G., Stephani, U.,** and **Ritter, H.** Vocal sonification of pathologic EEG Features. In Proceedings of the 12th International Conference on *Auditory display* (*ICAD 2006*), London, UK, 20–23 June 2006, pp. 158–163 (Department of Computer Science, Queen Mary, University of London, London).
19 **Dombois, F.** Using audification in planetary seismology. In Proceedings of the Seventh International Conference on *Auditory display* (*ICAD 2001*), Espoo, Finland, 29 July–1 August 2001, pp. 227–230 (Laboratory of Acoustics and Audio Signal Processing and the Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, Espoo).
20 **Dombois, F.** Auditory seismology on free oscillations, focal mechanisms, explosions and synthetic seismograms. In Proceedings of the Eighth International Conference on *Auditory display* (*ICAD 2002*), Kyoto, Japan, 2–5 July 2002, pp. 27–30 (Advanced Telecommunications Research Institute, Kyoto).
21 **Speeth, S. D.** Seismometer sounds. *J. Acoust. Soc. Am.*, 1961, **33**(7), 909–916.
22 **Vicinanza, D.** A Java framework for data sonification and 3D graphic rendering. In Proceedings of *Sound and music computing* (*SMC '06*), Marseille, France, 18–20 May 2006 (Groupe de Musique Expérimentale de Marseille, Centre National de Création Musicale, Marseille).
23 **Saue, S.** and **Kjeld, O. K.** A platform for audiovisual seismic interpretation. In Proceedings of the Fourth International Conference on *Auditory display* (*ICAD '97*), Palo Alto, California, USA, 2–5 November 1997, pp. 47–56 (International Community for Auditory Display, Santa Fe, New Mexico).
24 **Bovermann, T., Hermann, T.,** and **Ritter, H.** The local heat exploration model for interactive sonification. In Proceedings of the 11th International Conference on *Auditory display (ICAD 2005*), Limerick, Ireland, 6–9 July 2005, pp. 85–91 (Interaction Design Centre, Department of Computer Science and Information Systems, University of Limerick, Limerick).
25 **Baier, G., Hermann, T., Sahle, S.,** and **Stephani, U.** Sonified epileptic rhythms. In Proceedings of the 12th International Conference on *Auditory display* (*ICAD 2006*), London, UK, 20–23 June 2006, pp. 148–151 (Department of Computer Science, Queen Mary, University of London, London).
26 **Ramaswamy, L., Hutchinson, T. C.,** and **Kuester, F.** Towards enhancing earthquake response interpretation using sonification. In Proceedings of the Tenth International Conference on *Auditory display* (*ICAD 2004*), Sydney, Australia, 6–9 July 2004 (International Community for Auditory Display, Santa Fe, New Mexico).
27 **Eerola, T.** and **Toiviainen, P.** (2004) MIR in MATLAB: the Midi Toolbox. In Proceedings of the Fifth International Conference on *Music information retrieval* (*ISMIR 2004)*, Barcelona, Spain, 2004, pp. 22–27 (Universitat Pompeu Fabra, Barcelona, Spain).
28 **Lartillot, O.** and **Toiviainen, P.** A MATLAB Toolbox for musical feature extraction from audio. In Proceedings of the Tenth International Conference on *Digital audio effects* (*DAFx-07*), Bordeaux, France, 10–15 September 2007, pp. 237–244 (Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, Bordeaux).
29 **Slaney, M.** Auditory toolbox version 2, Technical report no. 1998-010, Interval Research Corporation, 1998 available from http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/ (accessed 28 March 2009).
30 **Matta, S., Rudolph, H.,** and **Kumar, D. K.** Auditory eyes: representing visual information in sound and tactile cues. In Proceedings of the 13th European Signal Processing Conference 2005 (*EUSIPCO 2005*), Antalya, Turkey, 4–8 September 2005, pp. 710–717 (Curran Associates, Red Hook, New York).
31 **Daw, C. S., Finney, C. E. A.,** and **Tracy, E. R.** A review of symbolic analysis of experimental data. *Rev. Scient. Instrum.*, 2003, **74**(2), 915–930.
32 **Daw, C. S., Kennel, M. B., Finney, C. E. A.,** and **Connolly, F. T.** Cycle-by-cycle combustion varia-

tions in spark-ignited engines. In Proceedings of the Fourth Experimental Chaos Conference, Boca Raton, Florida, USA, 6–8 August 1997, pp. 329–340 (World Scientific, Singapore).

33 **Daw, C. S., Finney, C. E. A.,** and **Kennel, M. B.** Symbolic approach for measuring temporal 'irreversibility'. *Phys. Rev. E*, 2000, **62**(2), 1912–1921.

34 DG Software, MagicScore music notation software, 2008 available from http://www.musicaleditor.com (accessed 16 November 2009).

35 **Ellis, D. P. W.** Extracting information from music audio. *Commun. ACM*, 2006, **49**(8), 32–37.

36 **Paulus, J.** and **Klapuri, A.** Music structure analysis by finding repeated parts. In Proceedings of the First Audio and Music Computing for Multimedia Workshop (*AMCMM 2006*), Santa Barbara, California, USA, 27 October 2006, pp. 59–68 (ACM, New York).

37 **Foote, J.** and **Cooper, M.** Visualizing musical structure and rhythm via self-similarity. In Proceedings of the International Computer Music Conference, Havana, Cuba, 2001, pp. 419–422 (International Computer Music Association).

38 **Eckmann, J. P., Oliffson Kamphorst, S.,** and **Ruelle, D.** Recurrence plots of dynamical systems. *Europhys. Lett.*, 1987, **4**(9), 973–977.

APPENDIX

Notation

A/F	air-to-fuel
CV	cyclic variability
DNA	deoxyribonucleic acid
ECG	electrocardiogram
IC	internal combustion
MIDI	Musical Instrument Digital Interface
RNA	ribonucleic acid
SI	spark ignition