NDT.net - The e-Journal of Nondestructive Testing (December 2008)
For more papers of this publication click: www.ndt.net/search/docs.php3?MainSource=25

Event-based Acoustic Emission Technique for Structural Health Monitoring using Wireless Sensor Networks

SEBASTIAN A. BACHMAIER

Institute of Construction Materials, University of Stuttgart
Pfaffenwaldring 4, D-70550 Stuttgart, Germany
sebastian.bachmaier@iwb.uni-stuttgart.de

Abstract
Structural health monitoring of engineering buildings can be accomplished by analysis systems based on acoustic emission. Conventional systems use wired technology, i.e. the equipment has to be installed on-site using cumbersome cables for sensor connections and power supply. Wireless sensor networks (WSN) in contrast offer high flexibility due to easy deployment. Acoustic emission sensor nodes can be mounted to the structure at any lifecycle state of the building.
Acoustic emission hardware allows the recording of structure-borne sound with a high sampling rate. By signal processing routines structural information like the location of cracks can be determined. However, these analysis methods need high-performance analog-to-digital convertors. In WSN the single nodes do neither offer fast converters nor high processing performance due to the used low-power components. Therefore, the analysis or transmission of high-frequency measurements is not feasible if the system is designed for long-term operation.
Hence, a simple event-based acoustic emission analysis based on hit-rate recording is proposed for WSN. The hit-rate is an indicator of the ongoing damage process. An increase of damage results in a higher hit-rate. At a central control centre, an alarm can be generated as soon as the sensors report an elevated hit-rate. Thus, the presented system can be used as an early-warning system for structural instabilities.

In the paper, the acoustic emission specific challenges for WSN are presented and a solution is proposed. A detailed account is given on the following aspects: hit recognition; hit counting; threshold values; power consumption and constraints of the system.

1. Introduction

Many engineering buildings are inspected at regular intervals, to ensure a maximum degree of safety and furthermore to render possible a smooth operation of the structure. Buildings of interest are those from structural and underground engineering and bridge buildings. If the health condition of the structure is misjudged, this can have fatal consequences, as show the breakdowns of Bad Reichenhall, Germany in 2006 or in Minneapolis, USA in 2007.

Structural health monitoring by means of wireless sensor networks (WSN) represent a cost-effective possibility of monitoring the state of structures. The cost advantage results from the easy, wireless installation on site, as well as from the usage of state-of-the-art but nevertheless cheap components within the sensor nodes. To still make a reliable judgment of the (potential) damages or damage processes, the nodes can communicate their data with their neighbors. By this information aggregation an improved and more intelligent analysis can be achieved.

Acoustic emissions occur in structures and are caused by cracks in the material. Many evaluation methods for processing these emissions have been developed[1], like localization of events by array techniques, correlation techniques or moment tensor inversion. However, most of these analysis methods need computing power exceeding the possibilities of embedded systems. Therefore, simple mechanisms like hit counting are studied and assessed for their applicability on wireless sensor nodes.

This work consists of three parts, where each presents the present state of work regarding the following:

1. to examine the possibilities and restrictions of facile acoustic emission hit counting theoretically and
2. to build a system which applies in practice the principles found, and
3. to examine if practice is in line with the theory.

2. Overview: Wireless Sensor Networks Architecture

Key elements of WSNs are the actual sensor nodes which make up the network. The communication link between these nodes is wireless, i.e. a – usually – short range radio link. One of the nodes is configured as data sink or as gateway. The gateway node has an additional WAN communication link to allow remote control and data drain. This second link can be any wired or wireless wide area connection. The gateway relays the data by this second communication link via the internet to a server within the private network of the WSN operator, hosting a database and a web server. An example of a complete architecture is shown in Figure 1.

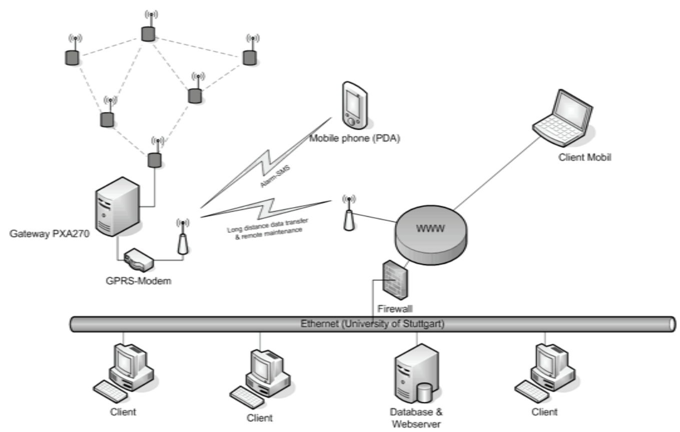

Figure 1: WSN architecture

The sensor nodes accommodate a low-power microcontroller (µC) and a low-power RF chip for the short range radio link. In the configuration the µC is used in, it has only limited capabilities of 4 MIPS, 10 KiB of RAM and 48 KiB of flash program memory. However, its integrated peripherals, such as ADC, DAC, timers, and serial interfaces, as well as advanced features like the DMA and interrupt capabilities and the flexible clock module, make it the best choice for low-power battery-driven applications.

3. Acoustic Emissions, Hits, Hit-rate and Events

Cracks in materials lead to waves in the material which cause measurable acoustic emissions. To measure the acoustic emissions (AEs), a sensor is needed. The piezoelectric effect as functional principle in many such sensors is found in different materials. In previous studies [2], a piezo-crystal sensor and polyvinylidene fluoride (PVDF) sensors were assessed and compared to micro-electromechanical systems (MEMS) on a microchip. For reasons of lacking sensitivity the use of PVDF is discouraged. It was also shown that MEMS show "poor characteristics in bandwidth, sensitivity, signal to noise ratio as well as power consumption"[2], so that the usage of piezo-crystals is advised for the given purpose.

The acoustic emissions caused by cracks can then be recorded with a suitable sensor which should have a frequency response from 20 kHz up to 30 kHz. If a larger bandwidth is used, the signal can be filtered to reduce undesired noise which is strong in the lower frequency components. If the signal is sampled (analog-to-digital-conversion) the resulting time series can be recorded and post-processed digitally. However, the amount of storage and processing power can be demanding, especially in miniaturized systems. Therefore, it is an interesting option not to sample the signal, but just to record threshold exceedings, called *hits*. This yields two advantages:

1. memory is saved, since there is no necessity to store samples, and
2. the system can stay in standby, since no power consuming sampling is performed.

Table 1 opposes the memory and power consumption of both the sampling variant and the hit counting variant.

Table 1: Resource needs for sampling and hit counting

	Memory Usage	**Power Consumption**[1]
ADC (@50 kHz sampling rate)	100 kiB/s	$P_{periphery}$ + 315 µW
Hit counting	depending on acoustical activity (typically several B/s)	$P_{periphery}$ + 24 µW

It shall not be left unmentioned, that though sampling a signal is more resource consuming, a more thorough examination and a more sophisticated evaluation of the signal can be achieved. A better noise suppression could be realized, as well as avoidance of misinterpreting of signals. In an array constellation, even fault localization is feasible. However, in this paper we like to demonstrate the possibilities of a simple regime.

In Figure 2 an example signal is displayed where at several times (depicted by the dotted lines) a hit is detected, since the threshold is exceeded. Each threshold crossing triggers an interrupt in the microcontroller (µC) which records the time of the crossing. Consecutive hits, i.e. hits within an interval of Δt, are counted as one acoustic *event*. In the example in Figure 2 six threshold crossings are registered, but they are counted as three events, only.

[1] estimations, based on specifications [8]

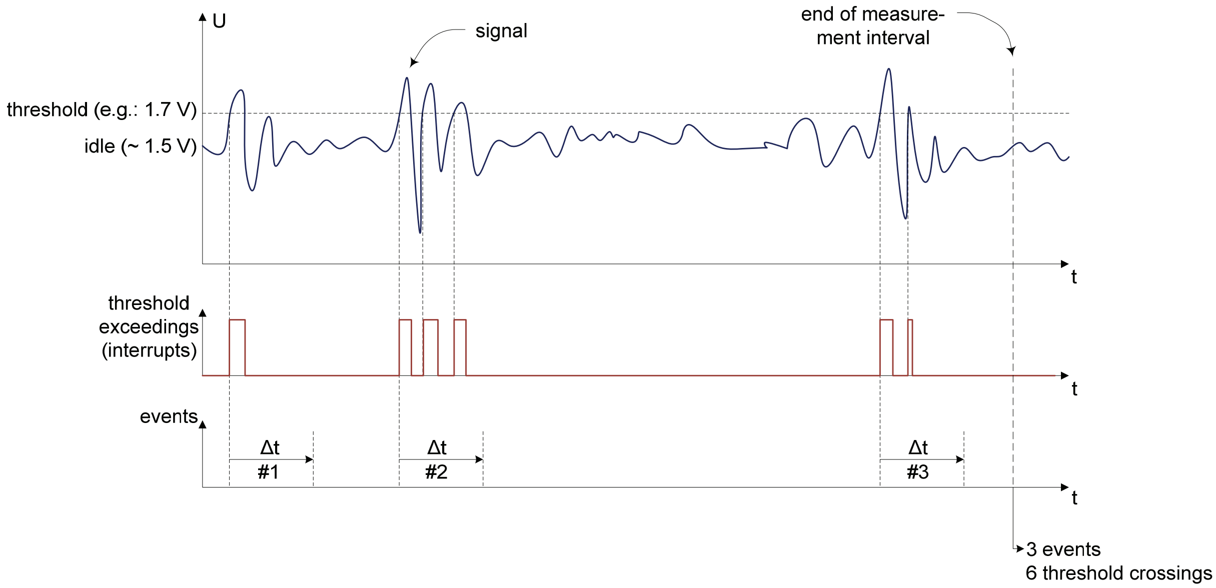

Figure 2: Sketch of a signal that exceeds the threshold several times, which each time sets off an interrupt in the µC where they are counted as events

Figure 3 shows the hits per time unit (called *hit-rate*) over time. Even if no cracks occur in the material, hits might be detected from time to time, resulting from ambient vibrations of the material (noise), which will be interpreted by the threshold detection mechanism as crack. This noise floor is shown in the first half of the curve of Figure 3. The second half shows the increase in hits once the material is stressed. The dotted line indicates an alarm threshold. As soon as the alarm threshold is exceeded, an alarm is issued to the operations centre and further measures, such as visual inspection or evacuation of a building, could be taken.

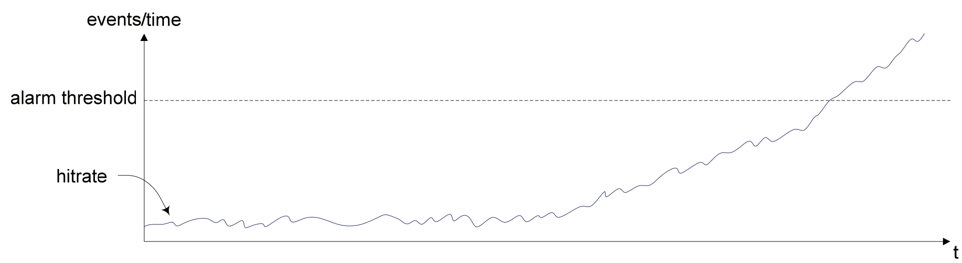

Figure 3: Hit-rate is the number of hits per time unit

Similar to the hit-rate, an event-rate can be defined and instead of transmitting the hit-rate, the event-rate can be transmitted to the operations centre. The events on the other hand can be classified on-mote to further reduce the transmitted amount of data, by only relaying events which belong with a high probability to cracks. This results in a three-options-scheme (cf. Figure 4), where the complexity of (the still simple) preprocessing increases but the transmission bandwidth decreases. A simple classification scheme used for option three is presented in the next chapter, which takes into account the time distribution of hits, and not only the time-grouping like for the event detection.

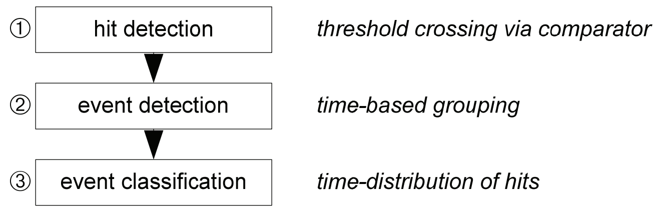

Figure 4: Stepwise preprocessing of hits leads to higher-value data

To determine optimal values for thresholds and boundaries, simulations with recorded data were conducted. The simulations and the findings are described in chapter 4.

4. Preprocessing Simulations

Basis of the simulations was data that was recorded during tests that were conducted in cooperation with the Technical University at Brunswick in December 2006 in Brunswick. The methodology by which was proceeded is described in the following: Firstly a graphical tool was developed, that allows the more or less swift manual classification of the 19,664 recorded data sets into three categories. (For details on the classification scheme refer to [3] and [4]). In a second step, a simulation tool was written to crosscheck if the proposed threshold-based method was able to discriminate between the classes of the classification scheme. The simulation is also used to find optimal parameters.

4.1 Reference Classification

For the manual classification a LabVIEW tool was written. LabVIEW was chosen for this task primarily for two reasons: firstly, VIs for data import exist in our workgroup for many file formats and secondly, displaying of data curves in a graphical way is easy in LabVIEW. Figure 5 shows a screenshot of this tool, which is named *ClassificaConcerto*. Users can easily proceed through any number of data sets, using either the mouse or the keyboard for swifter classification.

Output of the tool is a simple ASCII file with three columns, representing a sequence number, the dataset number and the classification, each separated by a semicolon and thus resulting in a Comma Separated Values (CSV) file. The CSV file can be read directly by various software packages, including Microsoft Excel, Mathworks Matlab and others.

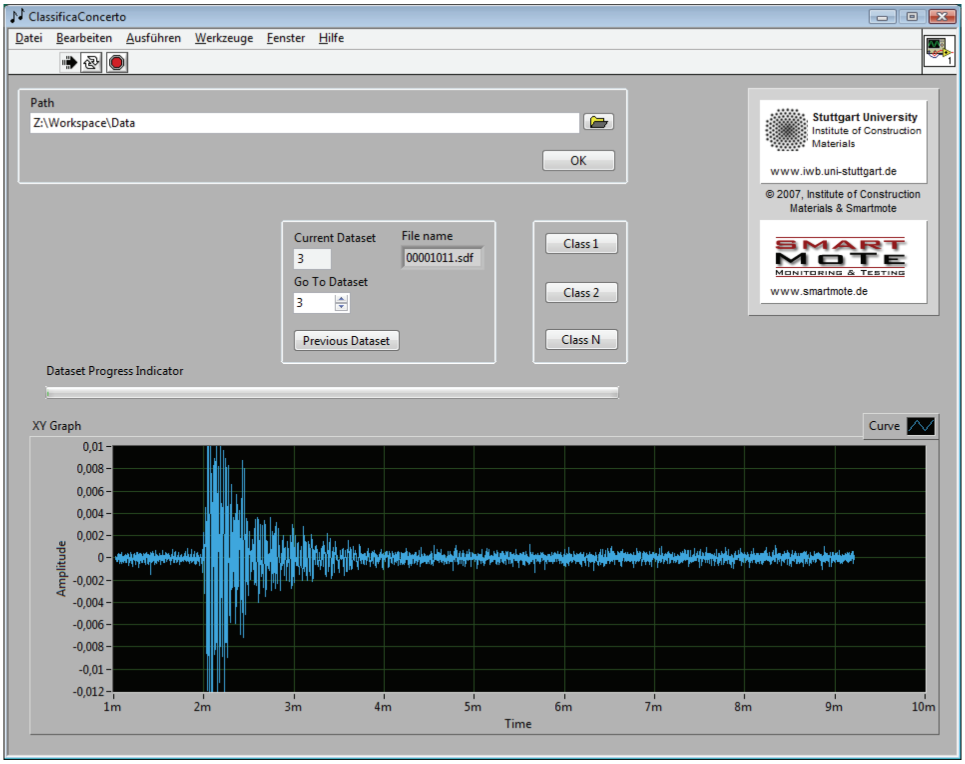

Figure 5 : Screenshot of the classification tool for manual classification into three classes; a curve of class 1 is depicted

The manual classification is time consuming and cumbersome. So, in a first run only 2,138 data sets were processed. In a next step, more data sets will be processed, to gain more confidence from a larger data basis. Results from the first classification batch showed, that 30.8 % of all data sets are class 1, 34.7 % belong to class 2 and 34.5 % to class N.

4.2 Computational Classification

All data sets were fed into a simple threshold detection algorithm, implemented in Matlab. In succession to the threshold detection, the cumulative sum was calculated. This totals curve was used for an automatic classification by evaluating the value of the totals curve at the end of the signal. This value has to lie within a lower and an upper bound for each of the three classes, which were derived from all the 2,138 manual classifications.

Figure 6 a) shows an example signal (black) and the corresponding threshold evaluation (red), as it was used in the first processing step. The threshold level for this example curve was at 5 % of the maximum signal strength. Figure 6 b) depicts the cumulative sum of the threshold detections for the acoustic signal of Figure 6 a).

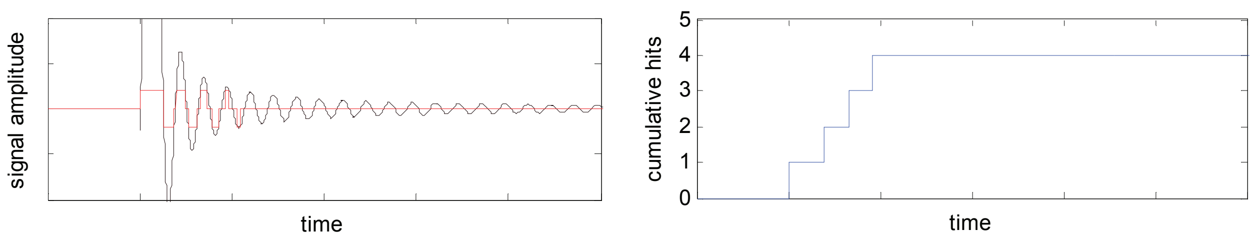

Figure 6: a) Sketch of an acoustic signal (black) with threshold detection (red)
b) Corresponding cumulative threshold detections/hits over time (blue)

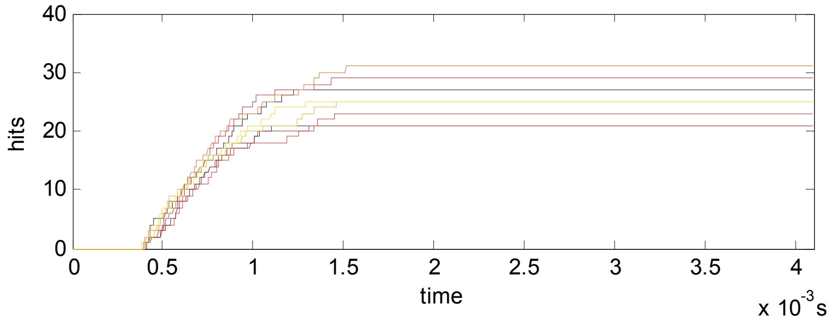

Figure 7 : Cumulated hits over time, with fanning out

Figure 7 depicts several cumulative sum curves, calculated on eight channels of one data set of real data for demonstration purposes. It can be seen that the curves look very much the same; however a *fanning* effect is noteworthy, i.e. the cumulative sum fans out even for very similar time domain curves. However, the fanning out is larger for signals of the different classes. This is used for automatic discrimination of signal classes.

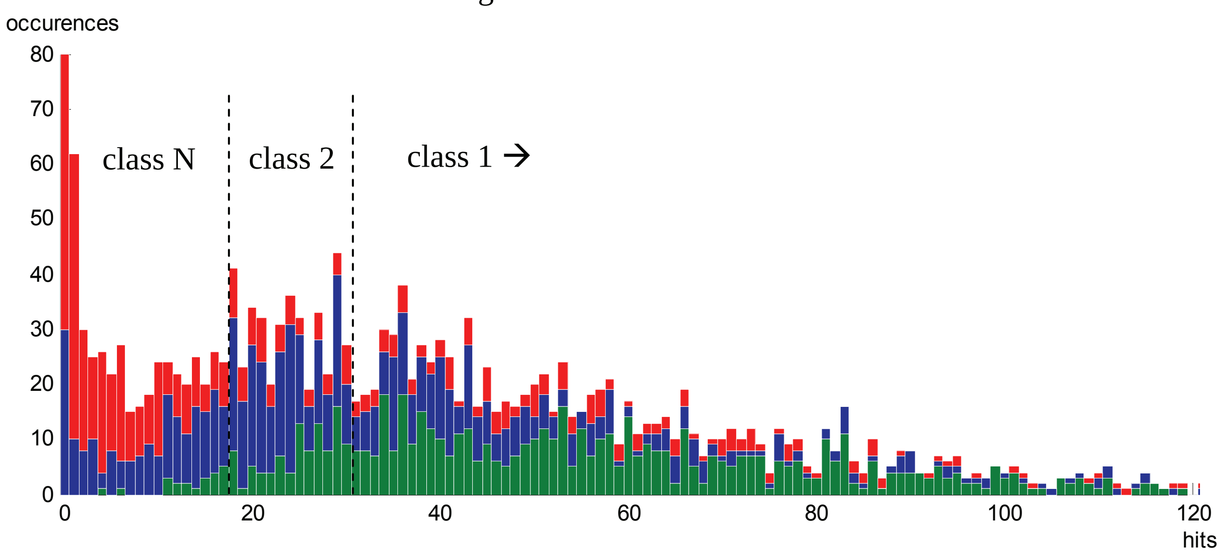

Figure 8: Histogram of the end value of the cumulative hit curve for the three signal classes: class 1 – green, class 2 – blue, class N – red

Figure 8 shows a histogram of the end values of the totals curves for hit detection threshold 5 %. It can be seen, that class 2 hit curves (blue) generally end up at higher values than those of class N (red) and similarly class 1 curves (green) end up at higher values than those of class 2. According to the lines in Figure 8 boundaries for automatic detection were selected. The boundaries were chosen manually for the time being. In a further step, the boundaries will be chosen computational maximizing the percentage of correct class mapping.

Threshold levels of 2.5 %, 5 % and 10 % have been used. Table 2 shows that the automatic classification obtained for a threshold level of 5 % of the maximum signal amplitude and the cumulative hit boundaries as depicted in Figure 8. It gives the percentage of classes that were contained in the respective automatically deduced class. For example, 52.0 % of the values that were found to be class 1 by the algorithm had been classified class 1 manually before.

Table 2: Probabilities of classes under the condition that a class was selected automatically

	Class 1	Class 2	Class N
Automatic Class 1	52.0 %	31.3 %	16.7 %
Automatic Class 2	19.8 %	40.1 %	40.1 %
Automatic Class N	3.1 %	27.1 %	69.7 %

5. Implementation in WSN

In a second step, the findings were implemented in an existing WSN system, which was built during the last years at the Institute of Construction Materials, University of Stuttgart. In the following, only the details relevant to hit-rate measurements are explained. For a general overview on the system, please refer to [5].

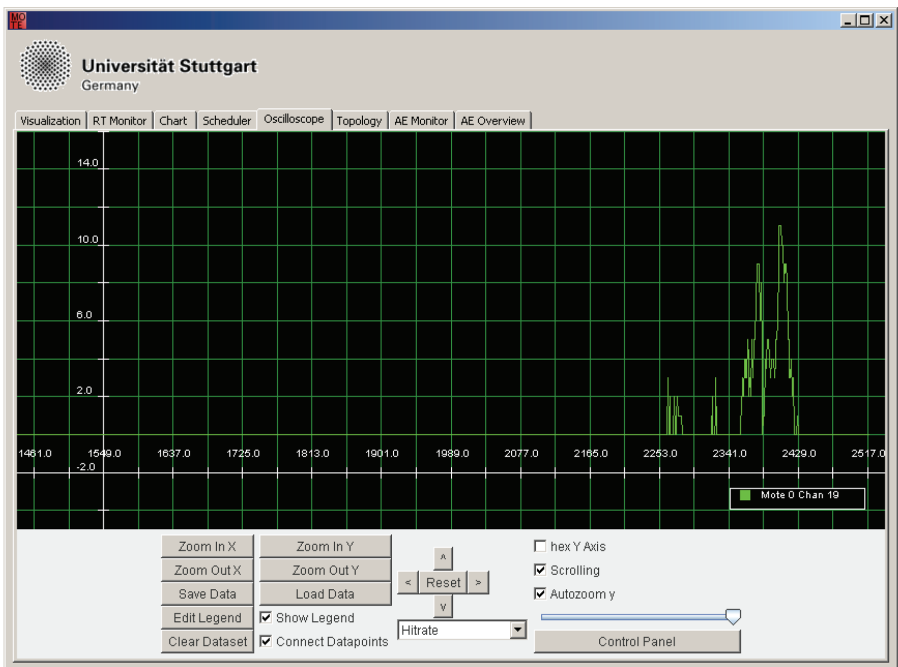

Figure 9: Hit-rate curve recorded and displayed by the graphical user interface

The hardware used was developed at the Institute of Construction Materials in the year 2006. This includes a signal conditioning board enabling the Tmote Sky^{TM} to process acoustic emissions. Basically, it comprises several amplifiers, filters and a signal detection circuit. The input signal is amplified by a factor of one-hundred before further (analog) processing. The signal detection circuit that was used to detect hits is shown in Figure 10. It can be seen, that the threshold level is adjustable to operate at an appropriate threshold level. According to the findings, it is a good option to set the level to 5 % of the maximum amplitude.

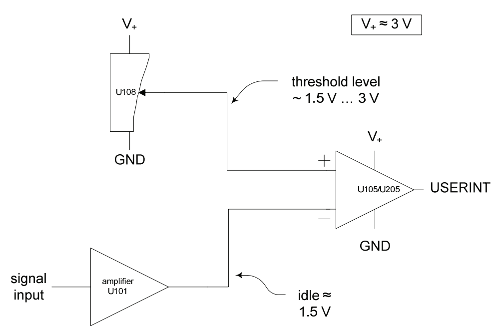

Figure 10: The trigger threshold can be set by adjusting potentiometer *U108*

The software comprises routines for the sensor node, written in NesC for TinyOS, as well as small adaptations to the graphical user interface (GUI). The oscilloscope view of the GUI is shown in Figure 9, where an example curve is depicted. The oscilloscope view plots data points on a time-line. Please note that the curve represents the bare hit-rate, as the event detection as well as the event classification have not been implemented on the sensor node, yet.

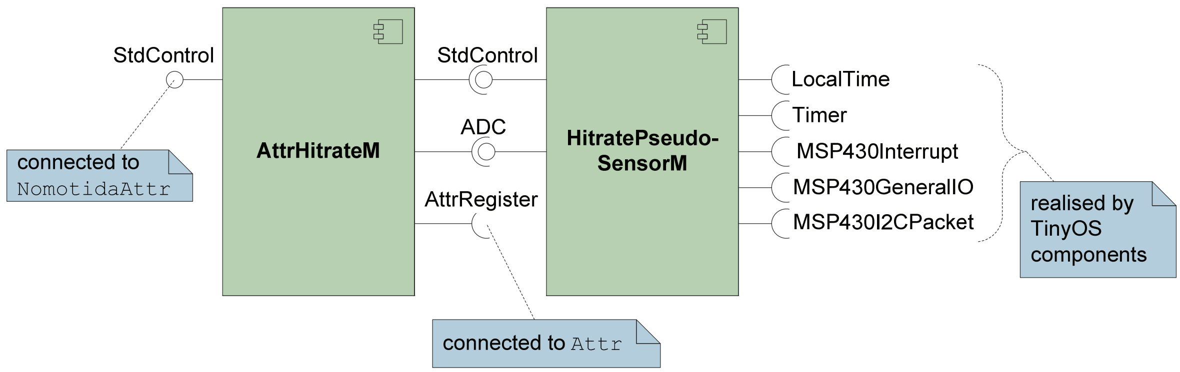

Figure 11: Component diagram of the hit-rate software components in UML 2 representation

Figure 11 shows the component diagram of the hit-rate software. The components are written in NesC for TinyOS. The existing software package running on the WSN nodes was extended by mounting a new hit-rate attribute. Due to the flexible attributes architecture, which is based on TinySchema, a part of TinyDB [6], adding new measurement attributes is easy. Besides the definition of the additional attribute a sensor component is necessary. It is the sensor component's responsibility to measure actual data and acting like an analog-to-digital-convertor component. One further software part is not shown in Figure 11: the interrupt service routine (ISR) that is called each time a threshold crossing is registered by the analog comparator. This ISR is similar in functionality to the threshold detection algorithm in the simulation.

Pre-processing as proposed and evaluated in simulation as described above, has to be implemented in the sensor component and the ISR.

6. Conclusions and further steps

The simple classification by threshold detection and sum curves works in theory, yielding an improvement over the simple hit counting. Further investigations will have to be done to determine if praxis is in line with theory. A first step is made and the hardware is available. However, on the software side, the cumulative curve calculation and the boundary checking still have to be implemented and should be considered as work in progress. Then, the system has to be applied to test specimens and to real structures to evaluate the practical relevance and see if further enhancements are necessary. Additionally, further simple processing methods, like band pass filtering, signal-to-noise calculation and others will be researched, to improve the quality of the detection of crack event. Also, more threshold levels and the automatic class boundary selection will have to be investigated systematically. In every step, special attention will be put on the implementation into a working system, extending the already functioning multi-hop sensor network nodes.

References

1. **Grosse, C., et al.** Acoustic Emission. [ed.] Kröplin, Wittel and Busse. *Damage in Composite Materials - Simulation and Non-Destructive Evaluation.* Norderstedt : ISD-Verlag, 2006.
2. **Sustainable Bridges.** *Technical Report on Wireless Sensor Networks using MEMS for Acoustic Emission Analysis including other Monitoring Tasks.* Stuttgart : European Union, 2007.
3. **Grosse, Christian, Krüger, Markus and Chatzichrisafis, Panagiotis.** Acoustic emission techniques using wireless sensor networks. *Proc. International Conference "Sustainable Bridges – Assessment for Future Traffic Demands and Longer Lives".* Okt. 2007.
4. **Chatzichrisafis, Panagiotis and Bachmaier, Sebastian.** *Data acquisition and processing for structural health monitoring using wireless sensor networks.* Stuttgart : Institute for Construction Materials, 2008. ISSN 0932-5921.
5. **Grosse, Christian and Krüger, Markus.** Structural health monitoring in civil engineering - applications using wireless sensor networks. *Proc. Conf. Emerging Technologies in Non-Destructive Testing (ETECH).* April 2007.
6. **Madden, Sam.** TinyDB: A Declarative Database for Sensor Networks. [Online] UCB. [Cited: March 28, 2008.] http://telegraph.cs.berkeley.edu/tinydb/index.htm.
7. **Grosse, Christian, Glaser, Steven D. and Krüger, Markus.** Wireless acoustic emission sensor networks for structural health monitoring in civil engineering. *Proc. European Conf. on Non-Destructive Testing (ECNDT).* Sept. 2006.
8. **Texas Instruments.** *MSP430x15x, MSP430x16x, MSP430x161x Mixed Signal Controller Datasheet.* [Datasheet] Rev. E, Dallas, Texas : s.n., 2006.