

A Heterogeneous System Architecture for Low-Power Wireless Sensor Nodes in Compute-intensive Distributed Applications

Andreas Engel, Andreas Koch
TU Darmstadt
Embedded Systems and Applications Group
Hochschulstraße 10, 64289 Darmstadt, Germany
engel,koch@esa.cs.tu-darmstadt.de

Thomas Siebel
Fraunhofer LBF
Research Division Smart Structures
Bartningstraße 47, 64289 Darmstadt, Germany
thomas.siebel@lbf.fraunhofer.de

Abstract—Wireless Sensor Networks (WSNs) are a key technology for future social and industrial developments. With increasingly complex applications, the compute demands for in-node / in-network processing have been growing steadily, while the capabilities of energy-harvesting or -storage systems have advanced only slowly. We present the Hardware-Accelerated Low Power Mote (HaLoMote), a *heterogeneous system architecture* for a wireless sensor node that achieves significantly better energy efficiency than traditional approaches, even for demanding applications requiring sensor sample rates of hundreds of Hertz. The paper discusses an evolution of the node hardware architecture and details the implementation of a signal processing chain required for Structural Health Monitoring (SHM) applications. The measurement accuracy of the WSN-based data acquisition system is compared against a wire-bound laboratory system showing that the dominant eigenfrequencies of a monitored structure can be detected with less than 1% error. Furthermore, the runtime and power requirements of the HaLoMote are compared against various software processors typically used for conventional WSN architectures. It can be shown that the HaLoMote is 2.3 times more energy-efficient than a state-of-the-art ARM Cortex-M3 based device.

I. INTRODUCTION

While heterogeneous computing architectures have been employed successfully in high-performance server, desktop, and even embedded systems such as mobile phones, they have seen only limited use in highly energy-critical settings, such as WSN nodes. But the increasing sophistication of WSN applications is accompanied by a corresponding demand for in-node computation, communication, security, and availability, which often conflicts with the still-limited energy supply. To meet the timing requirements, low-power microcontrollers (MCUs) (as used on the TelosB or Mica2 mote) are replaced by more powerful processors (e.g., 32 bit ARM on Imote2) at the cost of an increased power consumption in active mode (>100 mW) and longer wake-up times from deep sleep modes (>100 ms). Active power management on these nodes thus can only occur when measurements are completely suspended. However, improvements in energy storage and harvesting are made only slowly, and do not keep up with the growth in node capabilities. Dynamic Power Management (DPM) and in-

network data aggregation are well known strategies to achieve energy efficiency, but they become hard to realize at higher sampling frequencies.

As a more energy-efficient approach to homogeneous node architectures, we propose the use of a *heterogeneous system architecture* to handle even complex WSN scenarios. The architecture is able to handle compute-intensive tasks (e.g., data pre-processing) as well as long-term low-intensity operations (e.g., the RF communications protocol). To this end, our *HaLoMote* architecture employs a low-power Field Programmable Gate Array (FPGA), acting as Reconfigurable Compute Unit (RCU), for the first use, and a small 8 bit MCU with integrated RF functionality for the latter.

The architecture was designed for applications in which multiple sensors are sampled by each mote at several hundred hertz, thus producing large amounts of data that have to be preprocessed and aggregated. Such applications include e.g., the condition monitoring of vibrating machinery [11], vibration-based SHM [13], acoustic object localization [1] and video surveillance [20]. For such high-data rate applications, node-local processing can be used to reduce the necessary wireless communication bandwidth by techniques such as application-specific feature extraction (e.g., modal properties, object location), or generic data aggregation (e.g., lossless compression). But even low-data rate applications, such as environmental and industrial monitoring, may require more intense processing (e.g., for encryption protocols). The HaLoMote hardware accelerator is suitable for both use-cases.

While prior attempts at using reconfigurable computing in wireless sensor nodes have been made, their success has only been limited, as we will discuss in Section II. We present the evolution of the HaLoMote in Section III, while Section IV details a SHM application and its hardware-accelerated implementation. In Section VI, the accuracy of the proposed wireless data acquisition system is compared against a wire-bound laboratory system. Furthermore, the energy required per acquired sample is compared between the HaLoMote and various software processors.

II. RELATED WORK

WSNs have become popular for environmental monitoring in the last decades. Several research groups started to equip various bridge structures with wireless sensor networks ranging from small models with artificial excitation [3] over medium size pedestrian bridges [8] up to large size traffic bridges [5], [6], [17], [18], [23]. The number of involved sensor nodes ranges from 8 [8] to 70 [6]. The sampling rates and synchronization accuracy required for monitoring the distributed vibration of large structures are much harder to realize than monitoring slowly changing temperature or humidity. As the power supply of the sensor nodes is limited and the radio transceiver typically is the major power consumer of a sensor node, sensor data aggregation is required to reduce the overall communication and power requirements and thus extend the network lifetime or maintenance intervals. Very few of the related research projects actually support this feature, such as [6], [8]. Both execute complex in-sensor computations such as Frequency Domain Decomposition, or Stochastic Subspace Identification of a Filtered Hilbert-Huang Transformation on an XScale ARM processor. Compared to the traditional MCUs typically used in wireless sensor platforms, the XScale processor is very power-hungry, thus depleting even a relatively large 21 Ah battery in less than two months [6].

FPGA-based RCUs can perform complex computations more efficiently than MCUs and Digital Signal Processors (DSPs) [14]. In real-time applications, the use of RCUs often enables computations that cannot be performed by MCUs or DSPs at all under the given constraints. This made them attractive for use in sensor nodes performing compute-intensive applications (e.g., video and image compression) [4], [15], [27]. However, none of these systems could achieve truly low power operation: They all relied on FPGAs using Static RAM (SRAM) for configuration storage, which thus could not be powered down completely without losing the configuration data itself.

When energy actually becomes a first-class design goal, Flash-based FPGAs are far more suitable [21] for the RCU. Sensor nodes using a combination of a Flash-based Microsemi Igloo FPGA and a wireless transceiver have already been proposed [22], [25], [26]. However, despite the power advantages of Flash configuration storage, these architectures also turn out to be sub-optimal: All processing is performed on the RCU (even long-term low-intensity tasks), and when powered down, the radio transceiver is required to wake up the FPGA again. Thus, at least one of the two power-hungry devices has to be enabled all the time. Refinements which use very simple timekeeping on the RCU, employing inverter ring-based oscillators [16], to only power up the receiver periodically at pre-agreed times for data reception, are still sub-optimal: Due to the large timing inaccuracy (drift) of these oscillators, the power-down windows have to be shortened conservatively, leading to the system drawing higher power for longer intervals.

A better choice is a heterogeneous architecture *combining* an RCU and a low-power MCU. The Cookie WSN [24] and the PowWow Mote [2] have joined a small Microsemi Igloo FPGA

with a TI MSP430 MCU and an additional radio transceiver. However, both systems utilize the FPGA only for low-level handling of radio messages, instead of preprocessing the sensor data stream. Furthermore, the use of discrete MCU and RF components carries the burden of slower communication as well as more complex power management.

III. HALOMOTE HARDWARE ARCHITECTURE

To overcome the deficits of the WSN-architectures described in the last section, we proposed the HaLoMote architecture. In this section, we present its evolution driven by experiences gained in different applications.

The basic architecture of the original HaLoMote is shown in Figure 1. It heterogeneously combines software-programmable and reconfigurable compute units. The MCU handles less complex computations, such as the radio protocol and basic (yet precise) time-keeping. It is integrated with the RF components into a single System-on-Chip (SoC) and can directly access the human-machine interface (HMI) peripherals (e.g., LEDs) on the mainboard. The RCU is realized as a discrete FPGA based on non-volatile memory (NVM), allowing for deep sleep modes with fast shutdown and wakeup times as well as a very low static power draw. External sensors and additional memories are connected to the RCU to support the efficient preprocessing of the sampled data stream. Only the aggregated results are transferred to the MCU for RF transmission into the network.

For the first realization of the architecture, a Microsemi IGLOO AGL1000 FPGA and a TI CC2531 RF-SoC have been used and application-specific sensors and memories were flexibly attached using expansion headers. While it already exceeded the performance and power efficiency of homogeneous systems for real applications [13], the practical experiences gained identified a number of design weaknesses. This led to design improvements for a second implementation of the architecture, shown in Figure 2.

Detailed power profiling revealed that the 8051-based MCU-component of the CC2531 RF-SoC required significant energy even with the RF transceiver completely shut-down: With the control software on the MCU just initiating RCU operations (i.e., sensor sampling and data accumulation) at 128 Hz, the RF-SoC consumed between 34% and 48% of the overall system energy (depending on the actual load at the RCU) [13]. More

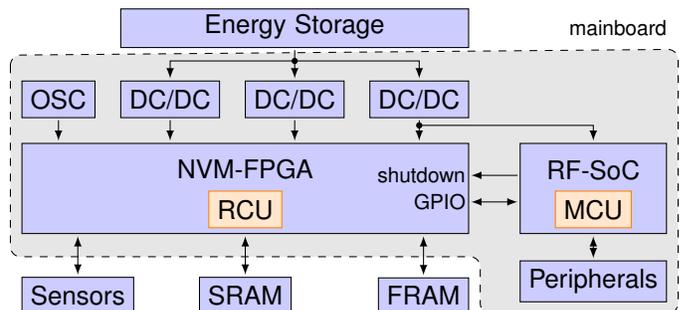


Fig. 1. Basic architecture of the Hardware-Accelerated Low Power Mote

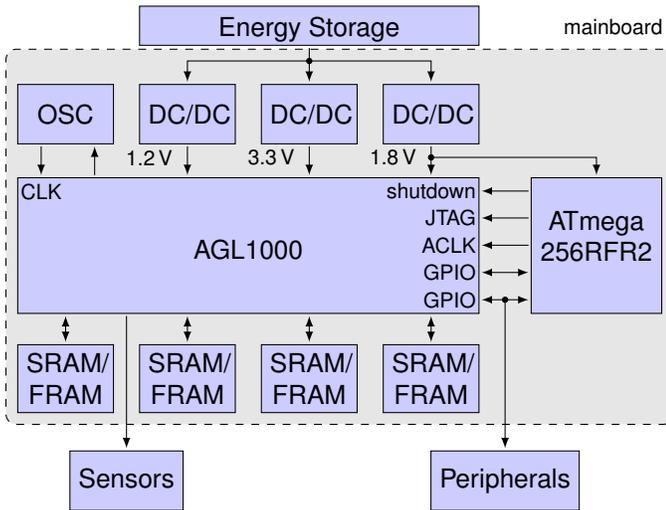


Fig. 2. Improved implementation of the HaLoMote

than half of the active time of the MCU ($31 \mu\text{s}$ out of $58 \mu\text{s}$) was spent waiting for the clock source of the RCU to become stable after waking it up. This led to a more refined clocking scheme which allows the MCU to provide an auxiliary clock for the RCU until the main oscillator started up.

In the second version, the TI RF-SoC itself was replaced by a more recent Atmel ATmega256RFR2 device, which not only has more RF throughput, but also more General Purpose Input/Output (GPIO) pins for communicating with the RCU. Furthermore, it can be operated with just a 1.8 V supply (instead of the 2.5 V used for the CC2531), thus allowing for more efficient switching regulators.

We excluded the power and area-consuming HMI-peripherals from the mainboard. While they can still be attached for debugging purposes, unused peripheral pins are now used to increase the communication bandwidth between MCU and RCU.

Most monitoring applications require a significant amount of external memory. By directly integrating four 1 Mbit serial SRAM devices on the mainboard, less demand was imposed on the expansion headers (now 40 pins, down from 148). This significantly shrunk the overall system size from $100 \text{ mm} \times 62 \text{ mm}$ down to $46 \text{ mm} \times 30 \text{ mm}$ (see Figure 3). We chose multiple serial instead of a single parallel memory to enable parallel independently addressed memory accesses by the RCU. Furthermore, we can now selectively replace one or more SRAMs with



Fig. 3. Current implementation of the HaLoMote architecture

pin-compatible non-volatile Ferroelectric RAMs (FRAMs) for even more aggressive power management. FRAM was chosen as persistent data storage as it clearly outperforms FLASH-based memory in terms of write performance (i.e., access time, granularity and power consumption) [19]. In particular, the MB85RS1MT modules can be written at 25 Mbit/s while drawing only 17 mW from the 1.8 V supply rail.

Finally, the new implementation allows over-the-air reconfiguration of the RCU by exposing the FPGA JTAG interface to the MCU. This comes, however, at the cost of an additional 3.3 V regulator responsible for providing the higher programming voltage.

IV. RANDOM DECREMENT TECHNIQUE FUNDAMENTALS

Civil infrastructures such as bridges are prone to fatigue and other load-induced damage. With increasing age, inspection intervals have to be scheduled more frequently to assure the secure operation of the infrastructure. Costly manual inspections thus have to be complemented by automated SHM. By periodically observing modal properties such as the eigenfrequencies, mode shapes, or the damping of the structures, damage or fatigue can be identified by significant deviations of these properties from reference measurements [9].

Modal parameters of an object are typically derived by observing its response to a well defined artificial excitation. While large structures can be excited by appropriate equipment, a significant amount of energy is required to drive such large shakers. Furthermore, the excited structures often have to be taken out of service to assure safety and measurement accuracy. For continuous automated SHM, the identification of the modal properties thus has to be based on the natural ambient vibration of the structure caused by wind or traffic. The major challenge of this Operational Modal Analysis (OMA) is the separation of the random components of the observed signals caused by the unknown excitation from the structure's actual response to this excitation. The Random Decrement Technique (RDT) was proposed for this purpose [7].

For the RDT, a set of sensors $S \subset \mathbb{N}$ is distributed all over the structure to acquire its vibrations in terms of acceleration or deflection as time series $(x_s : T \mapsto V)_{s \in S}$. For a finite sampling rate and measurement duration, the time domain is also finite and discrete, i.e., $T = \{0, \dots, n_t - 1\} \subset \mathbb{N}$. For simplicity, $V \subset \mathbb{R}$ can be assumed by abstracting from the finite measurement accuracy. As the OMA aims for the dynamic characteristics of the observed structure, the static components of the acquired signals (gravity or prestress) have to be eliminated by a high-pass filter, e.g., by applying a Finite Impulse Response (FIR) filter

$$\hat{x}_s(t) := \sum_{k=0}^{\min(n_f, t)} c_k \cdot x_s(t - k) \quad \forall (s, t) \in S \times T$$

of order n_f with $n_f + 1$ appropriate coefficients $c_k \in \mathbb{R}$.

To eliminate the random signal components, the RDT selects a subset of the sensors as references $R \subseteq S$ and a trigger level $l_r \in V$ for each reference $r \in R$. The points in time $t \in T$, at

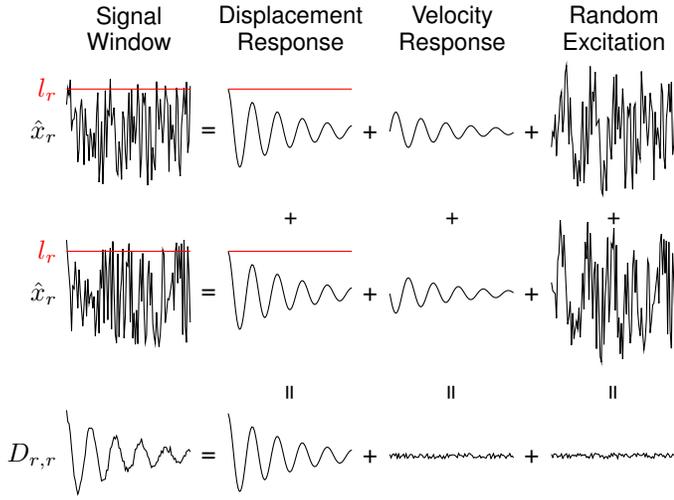


Fig. 4. Accumulation of triggered signal windows for the Random Decrement Technique

which a reference signal \hat{x}_r crosses l_r , are referred to as trigger events

$$E_r := \{t \in T : (\hat{x}_r(t) \geq l_r \wedge \hat{x}_r(t-1) < l_r) \vee (\hat{x}_r(t) < l_r \wedge \hat{x}_r(t-1) \geq l_r)\} \quad \forall r \in R.$$

A signal window $(\hat{x}_r(t+k))_{k=0}^{n_w-1}$ of fixed length $n_w \in \mathbb{N}$ starting at a trigger event is composed of the structure's response to its initial displacement $\hat{x}_r(t) = l_r$, its response to the initial velocity and the random ambient excitation, as shown in Figure 4. Assuming a zero-mean excitation, the random components are extinguished when accumulating a sufficient number of these triggered windows. The velocity response will also be eliminated, as each rising signal edge (with positive initial velocity) is followed by a falling signal edge (with negative initial velocity). Thus, the accumulated signal windows converge against the displacement response, which describes the structures free decay and can thus be used to derive its modal properties.

To estimate the mode shapes of the structure, spatial correlations between different sensor positions are required. Therefore, signal windows from all sensors are accumulated for each trigger event, resulting in $|S| \cdot |R|$ correlation functions

$$D_{s,r}(k) := \sum_{t \in E_r} \hat{x}_s(t+k) \quad \forall (s,r) \in S \times R, 0 \leq k < n_w.$$

Finally, the accumulated functions must be normalized by the number of detected trigger events, the trigger level and the standard deviation of the reference signals:

$$\hat{D}_{s,r}(k) := D_{s,r}(k) \cdot \frac{\sigma_r}{l_r \cdot |E_r|} \quad \forall (s,r) \in S \times R$$

$$\sigma_r := \sqrt{\frac{1}{|T|} \sum_{t \in T} \hat{x}_r(t)^2 - \left(\frac{1}{|T|} \sum_{t \in T} \hat{x}_r(t) \right)^2} \quad \forall r \in R$$

The normalized correlation functions $\hat{D}_{s,r}$ are the input of the subsequent modal analysis, which is performed on a central gateway and is thus not covered in this paper.

In addition to eliminating the random parts of the sampled signals, the RDT aggregates the $|S| \cdot |T|$ raw sensor samples down to $|S| \cdot |R| \cdot n_w$ correlation samples. The compression factor $\frac{|T|}{|R| \cdot n_w}$ increases with the measurement duration. n_w is typically chosen such that the correlation functions show the free decay of the structure, which may take several seconds for large bridges. For a measurement duration of several hours, which is required to collect a sufficient number of trigger events, the compression factor typically exceeds two or more decades. This is the major benefit of the RDT for the distributed WSN implementation of SHM applications.

However, the RDT increases the demand for in-sensor preprocessing. The computational complexity of the RDT is dominated by the FIR filter and the memory accesses required for the accumulation of the correlation functions. The latter becomes particularly complex if the correlation functions can not be stored in the few kilobytes of RAM provided by most WSN processing units thus requiring access to external memory. The RDT preprocessing linearly scales with the number of sensor channels to be processed by each sensor node. Each sensor typically provides three channels to capture the multidimensional movement of the structure and multiple nearby sensors may be connected to a single sensor node to simplify the deployment. Thus, assuming three to twelve sensor channels per mote is not unrealistic. As the sensor channels can be processed independently of each other, most of the RDT preprocessing can be parallelized. The next section details the RDT implementation on the HaLoMote RCU.

V. HARDWARE ARCHITECTURE FOR RDT

Figure 5 shows the sequential FIR implementation requiring one multiplier per sensor channel. The FIR taps are buffered in a Block RAM (BRAM) and the filtered value is passed to an additional First In, First Out (FIFO) buffer, which is integrated into the same BRAM as the FIR taps. The additional delay between the filtering of a sample and its further processing is required to ensure that trigger events captured at other sensor nodes can be flooded over the entire network [10].

Figure 6 shows the computational logic required for each sensor channel. A sensor specific control module requests the samples over the digital sensor interface. Although most of the control lines of the sensor interfaces (e.g., SPI or I²C) could be shared between multiple sensors, each sensor channel

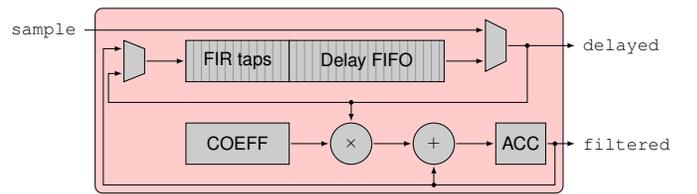


Fig. 5. FIR and FIFO in single BRAM

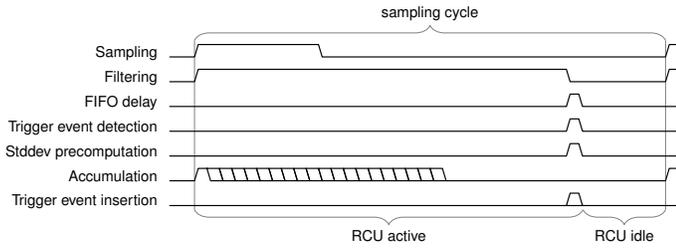


Fig. 11. Scheduling of parallel computations

reference channel share the same addressing signals, BRAM can be shared among those functions.

Figure 11 details the scheduling of the RDT operations. At the start of each sampling cycle, the MCU wakes up the RCU, which starts requesting the next sensor samples. The number of RCU cycles required for this operation depends on the number of bits to read, which typically do not exceed $3 \cdot 16$ bit. The high pass filter operates on the sample from the previous sampling cycle and can thus be executed in parallel to the sensor sampling. The filtering takes $n_f + 1$ RCU cycles. The trigger event detection, the computation for the standard deviation and the handling of the delay FIFO require one additional RCU cycle after the filtering operation. The accumulation of the delayed sample value from the last sampling cycle to the correlation functions is performed in parallel to the filtering. It requires one RCU cycle per registered trigger event. If available, new trigger events are inserted just before the RCU is shutdown again. As long as the number of registered trigger events does not exceed the order of the FIR filter, the overall execution time of the RDT kernel is fixed.

VI. EVALUATION

A. Measurement Accuracy

To evaluate the measurement accuracy of the proposed wireless data acquisition system, a laboratory scale testbed was set up as shown in Figure 10. A warren truss railroad bridge was modeled by connecting 54 metal rods with 24 metal joints (Figure 10a) resulting in 51 kg overall weight and a span width of 246 cm. The test structure can be excited by a G-scale railway model or an impact hammer.

Five HaLoMotes were attached to this structure (Figure 10b), each connected to four ADXL362 micro-electro-mechanical (MEMS) acceleration sensors (upper part of Figure 10c). Thus, the movement of all inner joints of the bridge can be observed in three dimensions with a resolution of 1 mg. However, only the acceleration orthogonal to the bridge deck is taken into account as this is the main direction of the ambient excitation caused by traffic. Due to the relatively large stiffness of the small bridge model, the relevant structural modes to be observed are located between 50 Hz and 100 Hz. Thus, to safely meet the Shannon-Nyquist lower limit, a sampling rate of 400 Hz was chosen. The wireless sensor nodes are synchronized with an accuracy of a few microseconds. The time synchronization protocol [12] is not detailed in this paper, but the achieved accuracy is sufficient for the required sampling rate.

The WSN-based data acquisition system does not capture the actual excitation of the structure, so an OMA is required as described in Section IV. A $n_f = 64$ tap high pass filter with a cut-off frequency of 20 Hz is applied to each sensor channel. This configuration was chosen as a trade-off between computational complexity and the filter quality. Static acceleration is damped by 60 dB, while all frequencies above 40 Hz are damped by less than 4×10^{-3} dB.

After the high-pass filtering, the RDT with two reference channels (nodes 3 and 13 as marked in Figure 10a) is applied. The trigger level of $l_3 = l_{13} = 200$ mg was determined experimentally and corresponds to the peak excitation injected by the train set. A window length of $n_w = 256$ is applied to capture the free decay of the structure within the first 640 ms after each trigger event.

The manual excitation of the structure for 10 s resulted in 60 trigger events registered at node 3 and 40 trigger events registered at node 13. The 40 resulting correlation functions $D_{1,3}, \dots, D_{20,3}, D_{1,13}, \dots, D_{20,13}$ were transmitted to a base station for the subsequent modal analysis. As shown in Figure 12 for $D_{3,13}$, these correlation functions characterize the free decay of the structure.

A second wire-bound data acquisition system was installed in parallel consisting of 12 PCB 356A16 integrated circuit piezoelectric (ICP) accelerometers (lower part of Figure 10c) controlled by the LMS Test.Lab 14A (LMS). Due to the limited number of input channels available at the SCADAS

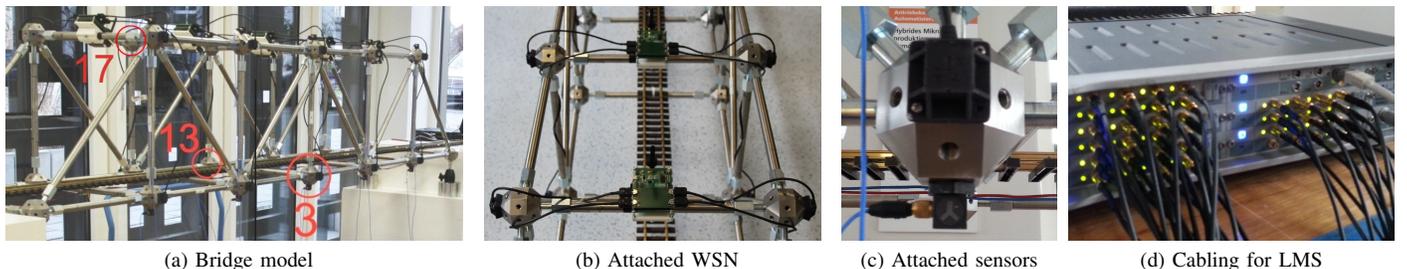


Fig. 10. Testbed consisting of a truss bridge model with attached wire-bound (LMS Test.Lab 14A with 12 ICP accelerometers) and wireless (HaLoMotes with 20 MEMS accelerometers) data acquisition systems

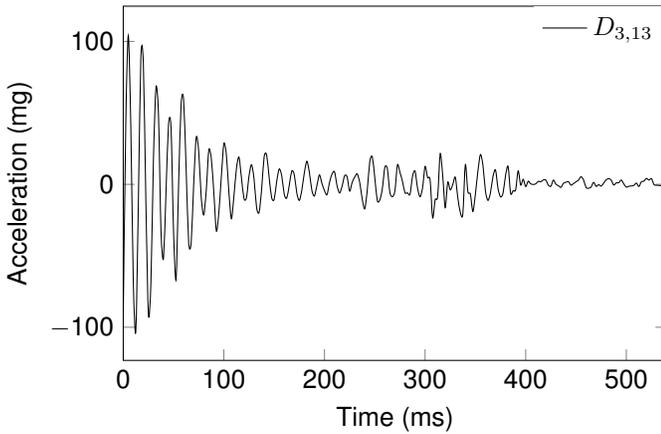


Fig. 12. Correlation function $D_{3,13}$ captured by the WSN

sensor front-end, only one half of the bridge can be completely observed by the reference system. In principle, both sides of the bridge could be analyzed independently, but moving the sensors from one side to another is not practical for long term observations as required by SHM. Instead, only two additional ICP sensors are installed in the other half of the bridge to assess the symmetry of the observed mode shapes. All sensors are sampled at 512 Hz with a resolution of 0.1 mg. The LMS system also captures the excitation provided by an impact hammer, so an Experimental Modal Analysis (EMA) can be performed thus providing more accurate results than the OMA-based WSN. Compared to the wireless data acquisition system, the cabling required for the LMS system becomes rather complex (Figure 10d) even though only 60 % of the structure is covered.

For the LMS measurement, five strokes with an impact hammer on node 17 (see Figure 10a) were injected in vertical direction at intervals of about 3 s to excite the vertical bending modes of the bridge model. The EMA results are averaged over those five individual measurements. The resulting frequency response functions at the lower central joint (marked as node 3 in Figure 10a) is shown in Figure 13. Below 40 Hz and above 110 Hz, the structure's characteristics can not be captured properly by the WSN-based system. However, the two dominating modes are located outside of the fuzzy frequency bands and can be captured with an accuracy of at least 1 % as summarized in Table I.

In addition to the eigenfrequencies, the actual mode shapes are of special interest for an SHM system as minor damage will be reflected in the deformation of the mode shapes before significant changes in the eigenfrequencies can be detected. Figure 14a shows the asymmetric vertical bending mode captured by both monitoring systems. Remember that the wire-bound LMS system has a limited view on the rear side of the structure due to input channel restrictions. However, the two nodes on the rear side are sufficient to detect the asymmetric character of the mode, i.e., the rear side is bending up while the front side is bending down. But only the WSN system provides a detailed view on both sides of the structure, which

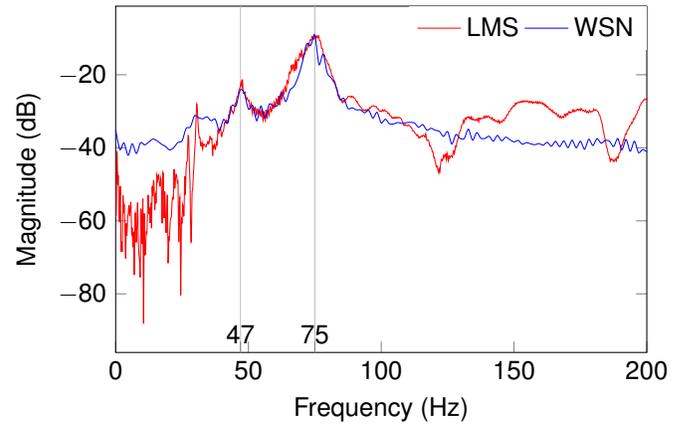


Fig. 13. Frequency Response Function at node 3 captured with LMS Test.Lab 14A and the Wireless Sensor Network

Eigenmode	by LMS	by WSN	relative deviation
Symmetric vertical bending	47.0 Hz	46.9 Hz	0.2 %
Asymmetric vertical bending	75.7 Hz	75.0 Hz	0.9 %

TABLE I
DETECTED DOMINANT EIGENFREQUENCIES

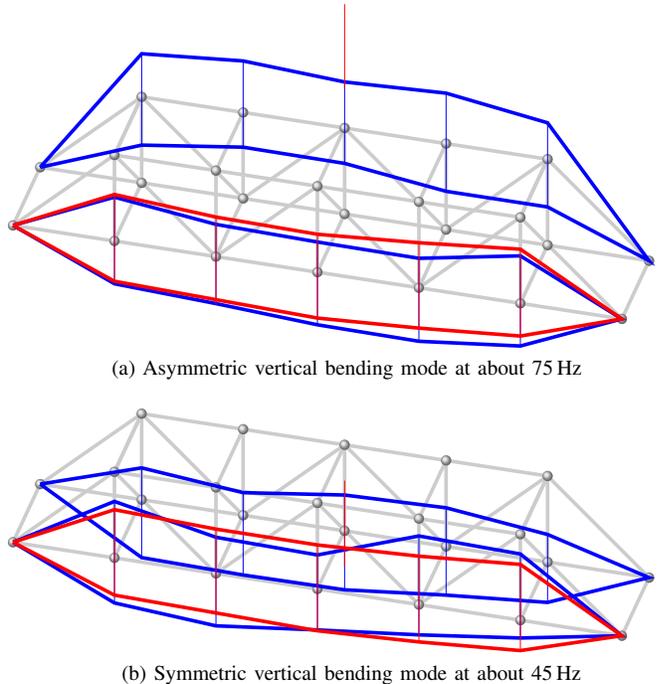


Fig. 14. Mode shapes captured with LMS Test.Lab 14A (red) and Wireless Sensor Network (blue)

is essential for the subsequent SHM analysis. This also holds true for the symmetric vertical bending mode shown in Figure 14b. Although the reduced accuracy of the WSN system is clearly visible in the mode shapes, the principal behavior of the structure can still be observed without the need for extensive cabling and well-known controlled excitation.

B. Resource Requirements

Finally, the runtime and energy required by the hardware accelerator of the HaLoMote architecture for the RDT computation is compared against software processors typically used in mobile and WSN applications. Namely, the TI CC2530 and the Atmel ATmega256RFR2 were chosen as representative 8 bit MCUs as these RF-SoCs have also been integrated into the HaLoMote. Furthermore, the TI CC430 RF-SoC was chosen as representative 16bit MCU, as the MSP430 architecture is widely used in many WSN motes. Most recent software processors for mobile applications are based on the ARM Cortex-M architecture, so the STM32F407 and the TI CC2650 were chosen as particularly powerful and energy-efficient state-of-the-art references.

For a fair comparison, all systems were configured with the same RDT settings as described in Section VI-A, i.e., 400 Hz sampling frequency, four sensor channels, two reference channels, 64-tap FIR filter and 256 samples per correlation function. As the actual work-load heavily depends on the amount of detected trigger events, all systems were fed with the same pre-recorded sensor samples stored in the processors code memories and the trigger levels were chosen such that the actual number of triggered events is 60 and 40, respectively, as observed in Section VI-A. The firmware for all systems was build with most recent compilers configured to optimize for execution speed. The average runtime per sampling cycle measured with peripheral timers was combined with data-sheet information about the power consumption and the wakeup time from sleep mode as shown in Table II. The deepest sleep mode with memory retention and enabled realtime-clock was chosen for each system respectively. To derive the overall energy spent per sampling cycle (E_{overall}), a power-consumption of P_{active} was assumed during wakeup, as the capacitors of the internal switching regulators have to be charged during the ramp-up. To better illustrate the consumed energy per sample by the different

processor architectures, the corresponding system live time (t_{alive}) achievable when supplied by a 1 Wh energy buffer (e.g., a typical NiMH cell) was derived. Note that this estimation takes only the processor into account, disregarding the sensors and the radio transceiver (which are assumed to be identical across the platforms examined).

As shown in Table II, the 8 bit MCUs do not achieve the required sampling period of 2500 μs . The CC430 requires about 50% of the sampling period for the RDT computations thus consuming 15.4 μJ per sampling cycle. The powerful Cortex-M4 device is nearly 30 times faster than the CC430, but its comparatively large power draw in idle mode still results in 11.4 μJ consumed per sampling cycle. The TI CC2650 proves to be the most energy-efficient software-processor under consideration as it requires only 1.6 μJ per sampling cycle. However, the HaLoMote FPGA requires only 28% of the energy of the most efficient software processor. Note that the HaLoMote MCU causes an additional overhead mainly caused by its wakeup (228 nJ) and idle (6 nJ) time. The combination of the hardware accelerator and the Atmel MCU, as used by the latest HaLoMote implementation, thus consumes only 44% of the energy of the most efficient software processor.

The energy efficiency of the HaLoMote easily exceeds that of the other platforms while actively performing computations, but it suffers when the node has to remain powered, but stays idle. In that case, its idle power consumption is 53 μW , which is nearly thirty times the power drawn by the CC2650 MCU in idle mode. As sensor motes spend most of their lifetime in idle mode, special care must be taken on the HaLoMote to address this issue. In the SHM application, this can be achieved by having a supervisory power manager suspend the sensor sampling when no traffic is present on the structure. For the specific use-case of railway bridges, these times may be more than 95% of the overall operating time. For these quiet periods, the non-volatile FRAM (see Section III) can be employed

Processing Unit	TI CC2530	ATmega256RFR2	TI CC430	STM32F407	TI CC2650	AGL1000
Architecture	8 bit 8051	8 bit AVR	16 bit MSP430	32 bit ARM Cortex-M4	32 bit ARM Cortex-M3	IGLOO FPGA
Compiler	SDCC 3.4	AVR GCC 4.3.1	CL430 4.4.3	ARM GCC 4.9.3	ARM GCC 4.9.3	SynplifyPro 2014.03
Main Clock [MHz]	32	16	20	128	48	8
V _{CC} [V]	2.0	1.8	2.4	1.8	1.8	1.2
I _{active} [mA]	6.5	3.7	4.6	40	2.9	25
P _{active} [mW]	13	6.7	11	72	5.2	30
sleep mode	LPM2	POWER-SAVE	LPM3	STOP	STANDBY	Flash-Freeze
I _{idle} [μA]	1	1.5	5.3	280	1	44
P _{idle} [μW]	2	2.7	12.7	156	1.8	53
t _{active} [cycles/sample]	257,795	43,387	24,985	5,463	7,150	68
t _{active} [$\mu\text{s}/\text{sample}$]	8,056	2,712	1,249	43	149	9
t _{wakeup} [$\mu\text{s}/\text{sample}$]	100	34	150	110	151	1
t _{idle} [$\mu\text{s}/\text{sample}$]	-	-	1,101	2,347	2,200	2,490
E _{active} [nJ/sample]	104,728	18,170	13,739	3,096	775	270
E _{wakeup} [nJ/sample]	1,300	228	1,650	7,920	785	30
E _{idle} [nJ/sample]	-	-	14	366	4	132
E _{overall} [nJ/sample]	-	-	15,403	11,382	1,564	432
t _{alive} [d]	-	-	7	9	67	241

TABLE II
RESOURCES REQUIRED FOR EXECUTING RDT ON VARIOUS PROCESSING UNITS

to store the internal state of the hardware kernels, which has to be preserved once the supervisory manager completely powers down the hardware accelerator. Note that the FPGA configuration data is not affected by such a shutdown, as it is held on-chip in non-volatile Flash memory.

For the concrete SHM configuration discussed in this section, about 48 kbit of runtime state has to be preserved across shutdowns (i.e., 41 kbit for the correlation functions, 3 kbit for the FIR taps, 3 kbit for the delay FIFO, and 1 kbit for the trigger events). Writing this state requires about 2 ms when using only one FRAM module. A state transfer between FRAM and FPGA requires a total of 188 μ J for both directions (state save and restore). This energy would be consumed by the FPGA in idle mode in about 3.5 s. Thus, state preservation and FPGA power-gating pays off for an idle duration of at least 4 s for the SHM application. Beyond the railway bridge use-case, such short idle-times occur even in many less frequently traveled automotive bridges. Thus, the capability of quickly and power-efficiently preserving the system state, while completely shutting down the accelerator, is attractive for a variety of applications.

VII. CONCLUSION

With the growing complexity of WSN applications, the use of heterogeneous computing architectures was shown to be profitable even for space- and power-constrained sensor nodes. The evolution of the HaLoMote was driven by practical experiences and resulted in a significantly more compact and efficient WSN mote. When used as a data acquisition system for distributed SHM applications, its accuracy can compete with a wire-bound laboratory system at least in the limited frequency range relevant for that applications while significantly reducing the deployment efforts. Finally, due to its hardware-accelerated signal processing chain, the energy-efficiency of the HaLoMote outperforms even most recent ARM-based processors designed for WSN applications by a factor of 2.3 thus improving the system lifetime when powered by a limited energy supply.

REFERENCES

- [1] S. Astapov, J. Ehala, and J.-S. Preden, "Collective acoustic localization in a network of dual channel low power devices," in *Proc. of the 21st Int. Conf. Mixed Design of Integrated Circuits Systems (MIXDES)*, June 2014, pp. 430–435.
- [2] O. Berder and O. Sentieys, "Powwow : Power optimized hardware/software framework for wireless motes," in *23rd Int. Conf. on Architecture of Computing Systems (ARCS)*, Februar 2010, pp. 1–5.
- [3] M. Bocca, J. Toivola, L. Eriksson, J. Hollmen, and H. Koivo, "Structural health monitoring in wireless sensor networks by the embedded goertzel algorithm," in *IEEE/ACM Int. Conf. on Cyber-Physical Systems (ICPS)*, April 2011, pp. 206–214.
- [4] A. Brokalakis, G.-G. Mplemenos, K. Papadopoulos, and I. Papaefstathiou, "Resense: An innovative, reconfigurable, powerful and energy efficient wsn node," in *IEEE Int. Conf. on Comm. (ICC)*, June 2011, pp. 1–5.
- [5] M. Chae, H. Yoo, J. Kim, and M. Cho, "Development of a wireless sensor network system for suspension bridge health monitoring," *Automation in Construction*, vol. 21, no. 0, pp. 237 – 252, 2012.
- [6] S. Cho, S. A. Jang, H. Jo, K. Mechitov, J. A. Rice, H.-J. Jung, C.-B. Yun, B. F. S. Jr, T. Nagayama, and J. Seo, "Structural health monitoring system of a cable-stayed bridge using a dense array of scalable smart sensor network," in *Proc. of the SPIE*, March 2010.
- [7] H. A. Cole, *On-line failure detection and damping measurement of aerospace structures by random decrement signatures*. Nielsen Engineering and Research Inc., 1973.

- [8] N. de Battista, J. M. W. Brownjohn, J. A. Rice, S.-H. Sim, and H. P. Tan, "Wireless structural monitoring of a multi-span footbridge with decentralised embedded data processing," in *Proc. of the 6th Int. Conf. on Structural Health Monitoring of Intelligent Infrastructure*, 2013.
- [9] A. Engel, A. Friedmann, M. Koch, J. Kohlfling, T. Siebel, D. Mayer, and A. Koch, "Hardware-accelerated wireless sensor network for distributed structural health monitoring," *Procedia Technology*, vol. 15, no. 0, pp. 738 – 747, 2014, 2nd Int. Conf. on System-Integrated Intelligence: Challenges for Product and Production Engineering.
- [10] A. Engel and A. Koch, "An energy-efficient wireless routing protocol for distributed structural health monitoring," in *Proc. of the 7th IFIP Wireless and Mobile Networking Conf.*, 2014.
- [11] —, "Hardware-accelerated data compression in low-power wireless sensor networks," in *Reconfigurable Computing: Architectures, Tools, and Applications*, ser. Lecture Notes in Computer Science, D. Goehringer, M. Santambrogio, J. M. Cardoso, and K. Bertels, Eds. Springer International Publishing, 2014, vol. 8405, pp. 167–178.
- [12] —, "Accelerated clock drift estimation for high-precision wireless time-synchronization," in *IEEE 40th Conf. on Local Computer Networks Workshops (LCN Workshops)*, 2015.
- [13] A. Engel, B. Liebig, and A. Koch, "Energy-efficient heterogeneous reconfigurable sensor node for distributed structural health monitoring," in *Conf. on Design and Architectures for Signal and Image Processing (DASIP)*, October 2012, pp. 1–8.
- [14] —, "Feasibility analysis of reconfigurable computing in low-power wireless sensor applications," in *Reconfigurable Computing: Architectures, Tools and Applications*, ser. Lecture Notes in Computer Science, A. Koch, R. Krishnamurthy, J. McAllister, R. Woods, and T. El-Ghazawi, Eds. Springer Berlin / Heidelberg, 2011, vol. 6578, pp. 261–268.
- [15] K. Goh, S. Ong, Y. Joe, P. Kusolpalin, W. Moh, and K. Ling, "Fpga based wireless sensor node for distributed process monitoring," in *7th IEEE Conf. on Industrial Electronics and Applications (ICIEA)*, 2012, pp. 1934–1939.
- [16] P. Grassi and D. Sciuto, "Energy-aware fpga-based architecture for wireless sensor networks," in *15th Euromicro Conf. on Digital System Design (DSD)*, Sept 2012, pp. 866–873.
- [17] X. Hu, B. Wang, and H. Ji, "A wireless sensor network-based structural health monitoring system for highway bridges," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 3, pp. 193–209, 2013.
- [18] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. of the 6th Int. Conf. on Inform. proc. in sensor networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 254–263.
- [19] M. Lueders, B. Eversmann, J. Gerber, K. Huber, R. Kuhn, M. Zwerg, D. Schmitt-Landsiedel, and R. Brederlow, "Architectural and circuit design techniques for power management of ultra-low-power mcu systems," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, pp. 2287–2296, November 2014.
- [20] A. Marcus and O. Marques, "An eye on visual sensor networks," *IEEE Potentials*, vol. 31, no. 2, pp. 38–43, March 2012.
- [21] T. Nyländén, J. Janhunen, J. Hannuksela, and O. Silven, "Fpga based application specific processing for sensor nodes," in *Int. Conf. on Embedded Computer Systems (SAMOS)*, July 2011, pp. 118 –123.
- [22] T. Nyländén, J. Boutellier, K. Nikunen, J. Hannuksela, and O. Silven, "Reconfigurable miniature sensor nodes for condition monitoring," in *Samos*, 2012.
- [23] S. N. Pakzad, G. L. Fenves, S. Kim, and D. E. Culler, "Design and implementation of scalable wireless sensor network for structural monitoring," *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 89–101, 2008.
- [24] V. Rosello, J. Portilla, and T. Riesgo, "Ultra low power fpga-based architecture for wake-up radio in wireless sensor networks," in *37th IEEE Annual Conf. on Industrial Electronics Society*, Nov 2011, pp. 3826–3831.
- [25] B. Stelte, "Toward development of high secure sensor network nodes using an fpga-based architecture," in *Proc. of the 6th Int. Wireless Communications and Mobile Computing Conf.*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 539–543.
- [26] L. Vera-Salas, S. Moreno-Tapia, R. Osornio-Rios, and R. de J Romero-Troncoso, "Reconfigurable node processing unit for a low-power wireless sensor network," in *Int. Conf. on Reconfigurable Computing and FPGAs (ReConFig)*, December 2010, pp. 173 –178.
- [27] C. H. Zhiyong, L. Y. Pan, Z. Zeng, and M. Q.-H. Meng, "A novel fpga-based wireless vision sensor node," in *Proc. IEEE Int. Conf. Automation and Logistics*, 2009, pp. 841–846.