When Scavengers Meet Industrial Wireless

Borja Martinez, Xavier Vilajosana, *Member, IEEE*, Fabien Chraim,
Ignasi Vilajosana, and Kristofer S. J. Pister

***Abstract* —Recent standardization efforts on industrial low-power wireless communication technologies clearly bet for the Time Slotted Channel Hopping (TSCH) medium access control (MAC) layer as it proved to achieve 99.999% reliability while ensuring deterministic behavior. Standards such as WirelessHART, ISA100.11a, and IEEE802.15.4e rooted at the TSCH MAC layer are used to connect millions of industrial devices today, enabling the emergence of the Industrial Internet paradigm. At that point and due to the ultralow energy profile of TSCH networks, scavengers come into play, enabling autonomously powered control and monitoring systems on industries. However, putting these systems together requires a clear understanding of their behavior. Therefore, this paper presents a methodology and a model to reliably dimension scavenger properties to network requirements and application needs, allowing industries to optimize the adoption of that technologies while keeping technical risks low.**

***Index Terms*—Energy scavenging, industrial wireless, low-power modeling, self-powered wireless sensor networks.**

I. Introduction

INDUSTRIAL wireless mesh networks are being consolidated by standardization efforts under the Time Slotted Channel Hopping (TSCH) scheme. This technique has been adopted by major industrial low-power wireless standards such as WirelessHART [1] and ISA100.11a [2] and, more recently, as a part of the IEEE802.15.4e standard [3]. As of today, several commercial low-power wireless networking providers are offering 99.999% reliable medium access control (MAC) layers, e.g., [4], that provide radio duty cycles well below 1%, thereby reducing the mote power consumption and increasing the network lifetime. This is facilitating the introduction of new monitoring and actuating devices that aim to improve the security, process automation, efficiency, and productivity of the industries and devises a clear roadmap to the Industrial Internet paradigm.

Even with the consolidation of industrial wireless communications, nowadays considered a mature technology, plant operators are reluctant to introduce mesh networks into their process, despite their very low energy profiles. One of the major obstacles of course is the power source of various wireless sensors and actuators. In many situations, batteries are too limited and difficult to replace. On top of that, certain devices cannot be powered by wires due to mechanical constraints, or because the cost of installing wires cannot be justified [5]. This is one area where scavenging technologies can be taken advantage of, particularly where the industrial setting becomes an interesting harvesting source (momentum, temperature gradients, etc. [6]).

Vibrational energy scavenging [7] is well studied when combined with motors and rotational objects [8], [9], which are omnipresent in heavy industries. Peltier effect devices [10] have also proven to obtain interesting amounts of energy when temperature gradients are nonnegligible, e.g., in cold and hot pipes also common in many industries. Recently, RF scavenging has been emerging as a very promising source of energy. However, augmenting wireless sensing and actuation devices with energy scavenging requires a clear understanding of the power dynamics of each application since harvested energy constitutes a new source of variability to the device.

In this paper, we present a methodology for precisely quantifying the power consumption on a per-application basis, which allows the designers to appropriately size their scavenging and storage devices. The presented work goes one step further than previous research by closing the loop and modeling the energy inputs and outputs in a per-subsystem basis. The model enables the adjustment of the application parameters and balancing the cost of communication, processing, and data acquisition subsystems according to the amount of harvested energy.

This paper is organized as follows. Section II reviews the state of the art and related work. In Section III, we present our model, broken down for data acquisition, processing, and communication. Then, in Section IV, we apply our model to the case of motor vibration sensing, as a means of an example. In Section V, we discuss the implications of our model for energy scavenging, before concluding in Section VI.

II. Related Work

Recent approaches to model the use of scavenging techniques in industrial wireless application can be found in the literature. Wang *et al.* [11] presented a model for the power consumption of a sensor network by analyzing the consumption of individual components that are involved in the communication. The provided analysis is suitable to dimension the energy spent during the communication process, but neither scavenging techniques nor the cost of the application itself is considered.

B. Martinez is with the Universidad Autònoma de Barcelona, 08193 Barcelona, Spain, and also with Worldsensing, 08013 Barcelona, Spain (e-mail: borja.martinez@uab.cat).
X. Vilajosana is with the Berkeley Sensor and Actuator Center, University of California at Berkeley, Berkeley, CA 94720 USA, with Worldsensing, 08013 Barcelona, Spain, and also with the Universitat Oberta de Catalunya, 08007 Barcelona, Spain.
F. Chraim was with the Berkeley Sensor and Actuator Center, University of California at Berkeley, Berkeley, CA 94720 USA. He is now with Solfice Research, Inc., Berkeley, CA 94710 USA.
I. Vilajosana is with Worldsensing, 08013 Barcelona, Spain.
K. S. J. Pister is with the Berkeley Sensor and Actuator Center,University of California at Berkeley, Berkeley, CA 94720 USA.

Torah *et al.* [8] assumed dependence between the harvesting pattern and the applications needs. In their approach, they build an application that wakes the microcontroller up and transmits a packet only when enough energy has been harvested. The approach draws a best effort system that cannot provide any sort of determinism, which is often required for industrial applications, method clearly does not allow for the use of TSCH-type layers. On top, this work does not provide a clear modeling of the application energetic requirements nor a detailed network energy consumption analysis. Recently, Liu *et al.* [12] presented a completely autonomous system using energy scavenging, proving the suitability of self-powered networked applications. This first demonstrator, however, is still far from industrial applications, and yet, its energy consumption profile is not well defined. Waterbury and Wright [9] presented a self-powered industrial application based on vibrational harvesting. Their work is focused on characterizing the amount of energy that can be obtained from an electric motor, complementing our work, but they do not outline how this energy is distributed among the application subsystems and therefore how this energy spending can be optimized. Lu and Gungor [13] already pointed out the need for energy-harvesting techniques in industrial wireless sensor network and presented an initial study on how different energy-harvesting techniques influence the device design in general terms. Nasiri *et al.* [14] presented a deep study on how photovoltaic cells in indoor scenarios can be dimensioned to power low-power electronic devices. Nevertheless, a fixed application setup is used, and the problem of adapting the application parameters to the available energy is not addressed. Tan and Panda [15] presented an analogous approach focusing the scavenging techniques on capturing the energy from thermal variations and photovoltaic cells. Recently, Magno *et al.* [16] presented a wireless structural health monitoring system, which incorporates a multisource energy scavenging subsystem; the article analyzes the energy input obtained by the harvesters, but does not take into account in their model the specific low-power wireless communications protocols and bandwidth requirements to achieve deterministic industrial communications. Other energy sources such as RF scavenging have been also recently studied by Musseta *et al.* [17], but the studies in question lack system-wide modeling and do not point out the impact of various application components to dimension the RF scavengers. Piezoelectric vibrational scavenging modeling has been addressed by D'hulst *et al.* [18]. Their work focuses on the analysis and modeling of the vibrational scavengers that can be used to harvest energy from vibrating sources such as motors, but the paper lacks a system-wide view and a much-needed discussion of whether the obtained energy is enough to power a device.

In general, a systematic study of how energy is distributed in the whole system has not been addressed in the literature. This paper improves current state of the art by defining a general methodology introducing a new perspective that can be applied to different industrial applications. A model is presented as a tool that aims to facilitate the application dimensioning and scavenger selection at predeployment phase. The model takes into account the three components that play a fundamental role in a realistic industrial application: standard networking technologies, sensing, and processing. This approach has not been tackled by the current literature, and results demonstrate that the contribution to the energy consumption of the different components is of the same order of magnitude, and consequently, all of them must be taken into account in order to explore the feasible options and fine-tune the application parameters.

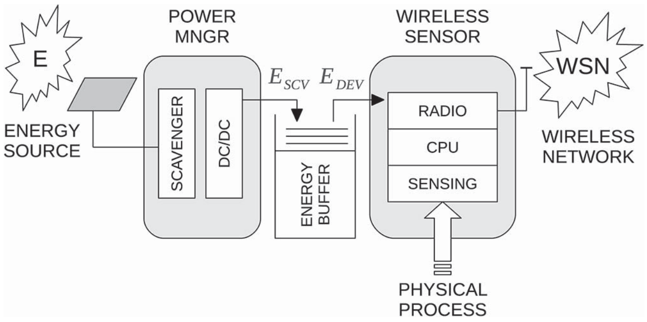

Fig. 1. Self-powered wireless sensor device diagram.

III. Model for Early Power Estimation

A *self-powered* wireless sensor device is characterized by a sustainable provision of energy. In a general model, as depicted in Fig. 1, the energy scavenged from the medium E_{SCV} must be greater than the energy required by the device E_{DEV} along its operational lifetime. Formally,

$$E_{\text{SCV}} = \int_0^\infty P_{\text{SCV}}(t)dt \geq \int_0^\infty P_{\text{DEV}}(t)dt = E_{\text{DEV}}. \qquad (1)$$

Despite this being a necessary condition during the operation of a device, the instantaneous power supplied from the harvester $P_{\text{SCV}}(t)$ (conditioned by the environment) is in general independent of the energy consumption rate $P_{\text{DEV}}(t)$. The latter is governed by the specific application. Then, a properly sized energy buffer is required to balance the generation and demand asymmetries, as shown by Jiang *et al.* [19]. With that assumption in mind, dimensioning the energy generation and the energy consumption can be addressed independently. Thus, (1) can be reduced to a simpler form $\bar{P}_{\text{SCV}} \geq \bar{P}_{\text{DEV}}$ in terms of average behavior on the characteristic cycle of operation of the application. The problem of sustainability of a wireless sensor network is then described by a clear understanding of the energy generating pattern, the characterization of the operational energy demand, and a dimensioning of the energy transfer buffer.

As shown in (2), the power required to operate a wireless sensor device can be broken down into three main blocks: energy required for data sensing (acquisition), for application (data processing), and for communication (networking), i.e.,

$$P_{\text{DEV}} = P_{\text{NET}} + P_{\text{ACQ}} + P_{\text{PRC}}. \qquad (2)$$

Our model therefore is based on an atomic breakdown of each building block, where the instantaneous power consumption is integrated over the duration of the corresponding task and averaged out its characteristic temporal scale (period of repetition). The next subsections go into a detailed analysis of each energy building block.

A. Model for TSCH Networks Energy

TSCH networks show an ultralow power consumption profile due to the low-power nature of IEEE802.15.4-compliant radios and due to the fact that nodes are synchronized and actions occur at specific moments in time [20], enabling nodes to optimize the usage of their resources. Since the actions that occur at each time slot are well known, the energy consumption can be modeled on a slot-per-slot basis. With precise modeling, the task of precisely calculating the energy consumption of a network according to its traffic requirements becomes straightforward. In this paper, we apply the model derived in [21] as a tool to estimate the energy consumption of the addressed scenarios. Note that our aim in this case is to introduce the use of the previously defined model into the presented methodology.

When a device joins a TSCH network, it obtains information about the duration of each time slot, as well as the number of slots in a slotframe. At each slot, the node can either transmit, receive, or keep its radio off. A scheduling entity is responsible for building the schedule that will satisfy the bandwidth and latency needs of the different flows in the network. The schedule allows for a fine-grained tradeoff between latency, bandwidth, redundancy, and power consumption. The overall energy consumption of a mote is the sum of the energy consumed in each slot, given that each type of slot has an energy consumption profile related to the hardware and the activity it is performing (e.g., transmit, receive, and sleep).

In a TSCH network, slots are grouped in slotframes that repeat over time (see Fig. 2). The used model is based on profiling the energy consumption on each of those slots and computing the overall energy consumption for each slotframe. The average power can be obtained by dividing the total energy E_{SF} by the slotframe period T_{SF}, as indicated by (3). Since slotframes repeat cyclically, this period represents the characteristic temporal scale of the network. Then, applying the model is a matter of counting the number of slots of each type and calculating the energy consumed by every one of the slots. Note that the following formula is a summary of how the energy consumption of the network needs to be calculated. It is important to recall that each type of slot (active, idle, off, etc.) has a different energy consumption profile, and therefore, $E^{(i)}_{\text{SLOT}}$ is dependent to what is happening on that slot (see [21]). Thus,

$$P_{\text{NET}} = \frac{E_{\text{SF}}}{T_{\text{SF}}} = \frac{1}{T_{\text{SF}}} \sum_{i=1}^{N_{\text{SLOTS}}} E^{(i)}_{\text{SLOT}}. \tag{3}$$

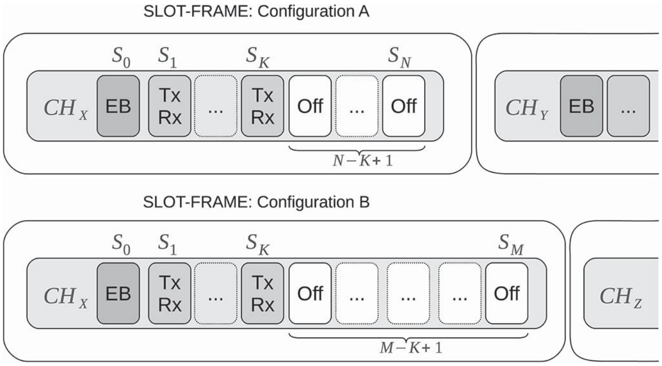

Fig. 2. Example of two TSCH slotframe configurations with different numbers of slots N and M and assuming $M > N$. The first slot is used for network discovery by means of *enhaced beacons*. Then, K *data* slots for transmission and reception are common in both configurations. Configuration A has $N - K$ *sleep* slots (unused), whereas Configuration B has $M - K$ sleep slots, which means that a node running in this configuration will be idle for longer periods. This *slotframe* repeats over time.

During a slot, the actions of different modules (microcontroller, radio, etc.) can be profiled according to the MAC layer timing defined by the standard (e.g., IEEE 802.15.4e) and summarized in [21]. Then, the hardware datasheets can be used to estimate the energy consumed by each action according to the energy state of the components currently being used and the time required by the action. The referenced study shows that manufacturer claims provide a fairly accurate estimate of the actual (measured) power consumption. The total energy consumed during a slot is then computed as the sum of the individual contributions, mainly defined by the microcontroller and the radio state, as shown in

$$E_{\text{SLOT}} = \sum_{k=1}^{N_{\text{TASKS}}} E^{(k)}\ (\mu C,\ \text{Radio}). \tag{4}$$

For the purpose of the presented methodology, we describe the parameter dependence and the impact of the network configuration on the energy consumption of the application. In a TSCH network, the slotframe length N_{SLOT} determines how often actions repeat, which usually depends on application requirements. The amount of scheduled slots (i.e., *transmit* or *receive*) depends on the traffic requirements of the application and, as the transceivers often consume higher amounts of energy, the communication cost dominates as the traffic on the network increases. Latency, robustness, and energy consumption are compromised.

Energy consumption can be reduced by increasing the length of the slotframe, i.e., by inserting more *sleep* slots or by disabling some *active* slots so they become sleep slots. A node only is active in certain of the timeslots in the slotframe, which are used to send or receive information. In the rest of the nonactive slots, the node remains switched off. If the number of active slots remains constant and the slotframe size increases, the ratio of sleep slots increases. This means that the average energy spent by the node is smaller as it is less active. The same effect is obtained changing active per sleep slots. However, the reduction of activity comes at the cost of less bandwidth and increased latency. Reliability is also compromised, as less redundant links to neighbors are expected.

B. Modeling Data Acquisition Energy

It would be impossible to come up with a model that captures all of the sensing techniques. However, it is reasonable to say that most applications fall under the following two categories: regular sensing (with a fixed interval) and event-driven sensing. In regular sensing, a sensor is woken up at regular intervals to collect one or more samples and then sent back to sleep. Of course, when energy is freely available, the sensor can be left on permanently, but that is not the case of most battery-operated devices, where duty cycling the sensor becomes necessary. In event-driven sensing, a random event triggers a series of samples from the sensor. This event can be internal to the sensor (e.g., crossing a preset threshold) or external (e.g., a request of sensor data coming from the plant operator).

To quantify the energy consumption of the sensing component in the wireless device, we must first look at the energy required to capture one sample. Then, the model is generalized to account for more than one regular sensing intervals (with different periods and sampling requirements) and various event types. Equation (5) models the energy consumption of the acquisition component, i.e.,

$$E_{\mathrm{ACQ}} = E_{\mathrm{SMP}} \cdot \left[\sum_{k=1}^{K} N_{S,\mathrm{REG}}^{(k)} + \sum_{l=1}^{L} P_{S,\mathrm{EVT}}^{(l)} \cdot N_{S,\mathrm{EVT}}^{(l)} \right]. \quad (5)$$

In (5), K is the number of sensor sampling on a regular basis, whereas L represents the number of sensors triggered by events. The following variables are used.

- E_{SMP} is the energy of one sample, as presented in Fig. 5.
- $N_{S,\mathrm{REG}}$ is the number of samples taken during one regular sensing interval.
- P_{EVT} is the probability of an event occurring in one sensing interval.
- $N_{S,\mathrm{EVT}}$ is the number of samples taken following the occurrence of an event.

C. Modeling Local Data Processing Energy

The sensing and networking components are common to all applications running on the same platform, and thus, the same profiling can be reused for further modeling. In contrast, application developers need to be able to estimate the energy consumption of specific hardware at design time, without having to provide the actual hardware implementation. This enables the exploration of different alternatives while minimizing the risk.

To estimate the energy drained from the battery by an application task, we use a method proposed and originally validated in [22]. Starting from a high-level description of the algorithm (e.g., MATLAB/Octave), the number of operations to process the original sensed signal is recorded, accounting basically for the number of arithmetic operations, namely, additions, multiplications, divisions, and comparisons, i.e., the main actors in signal processing loops. Thus, depending on selected hardware architecture, we map these counters into the corresponding number of microcontroller clock cycles, and subsequently, the latter is mapped into the corresponding energy expenditure.

This method offers accurate results as long as the CPU tasks rely mainly on arithmetic instructions (as digital signal processing algorithms do). However, it is no longer applicable when the microcontroller is involved in nonarithmetic-based tasks, such as dealing with a protocol stack. In that case, an alternative method is presented through the use of *instruction set simulators*.

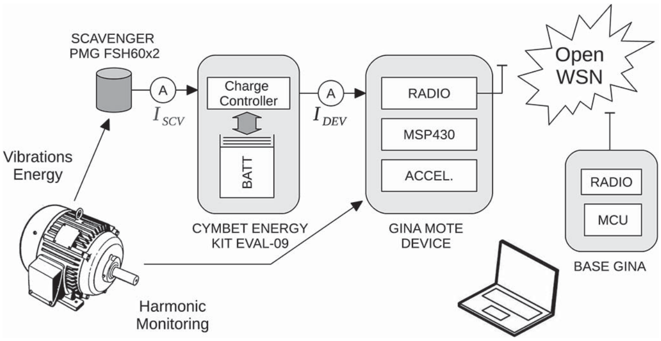

Fig. 3. Experimental setup diagram.

IV. Application-Oriented Model Validation

Rotary machines of all sizes are the propelling force behind many industries (refining, chemical, energy production, and so on). In the constant quest for better efficiency and lower energy consumption, vibration monitoring of rotating machines becomes necessary. In short, plant operators aim at increasing the load on the machines and their running time, while reducing their downtime and energy consumption. This is accomplished by installing vibration sensors (accelerometers, for example) and performing a harmonic analysis of those vibrations to monitor the health of the machinery [23]. For cost and logistical purposes, it is often desired that the sensing equipment be self-powered and wireless, thereby forcing the system designer to carefully plan the energy consumption [13]. Here, we illustrate the model presented in Section III by taking the vibration monitoring application as our case study.

As described in Section III and for each subsystem, the atomic energy contributors are identified. As a general rule, the current consumption is obtained from the device's datasheet. This consumption is then multiplied by the duration of each task (estimated according to some selected set of network/application parameters), to get the total charge drained at each stage.[1] The estimation of each contributor is compared with the real measurement to show the correctness of the proposed methodology.

The experimental setup we used in our application is shown in Fig. 3. The GINA platform [24] was used for sensing, processing, and networking. It holds the Texas Instruments MSP430f2618 16-bit microcontroller, the Atmel AT86RF231 IEEE802.15.4 radio, and the STMicroelectronics LIS344ALHTR three-axis analog accelerometer. For energy storage, a lithium polymer battery was used with a battery charger controller. The selected vibrational scavenger is the Perpetuum PMG FSH60x2, which is attached to the case of the motor in a position that maximizes both the vibrations transferred to the sensor and those being scavenged. All experiments are measured with the NI9203 16-bit analog current acquisition module, with a resolution of 0.6 μA per least significant bit.

[1] In our case, all components are sourced to the same voltage level. The model is presented with currents instead of power and charge instead of energy (i.e., normalized by the voltage). As the reference values in the datasheets of the components, batteries, and scavengers are given in intensity units, we avoid continuous conversions and facilitate the experimental measurement as only current proves are required.

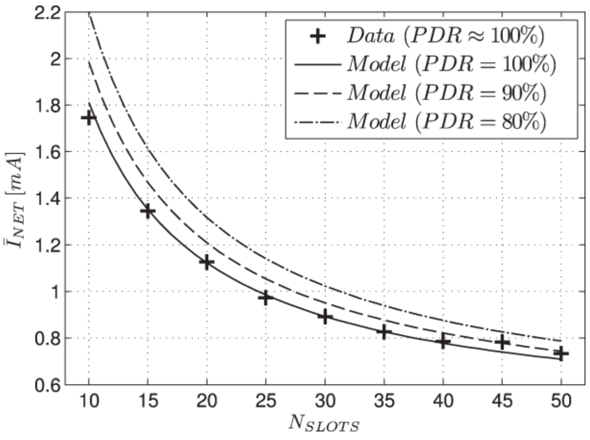

Fig. 4. Network model for different values of PDR compared with experimental data. N_{ACT} is fixed to three active slots, and T_{SLOT} = 0.015 s.

A. Network Energy Profiling

To profile the energy consumed by TSCH networks, we apply the model defined in [21] to determine the charge used in each slotframe with different network setups. This configuration has been also validated in an experimental setting with the GINA mote running the OpenWSN protocol stack [25]. In short, the average current required to maintain the network can be approximated by

$$I_{\mathrm{NET}} \approx \frac{\bar{Q}_{\mathrm{MSG}} \cdot N_{\mathrm{ACT}}}{T_{\mathrm{SLOT}} \cdot N_{\mathrm{SLOTS}}} + I_C. \tag{6}$$

In (6), $\bar{I}_C$ accounts for the activity of the μC to manage the network (system) and can be regarded as constant. T_{SLOT} is a fixed network parameter. $\bar{Q}_{\mathrm{MSG}}$, representing the average charge required per active packet, depends on the radio technology and the network state. A lower packet delivery ratio (PDR) implies a higher average current per packet arrived with success. This means that, once the number of active packets N_{ACT} in the slotframe is provisioned, the number of slots N_{SLOTS} in the frame becomes the control parameter for the network energy, giving characteristic functional dependence on $I_{\mathrm{NET}} \propto 1/N_{\mathrm{SLOTS}}$, which is shown in Fig. 4.

In this experiment, slotframes from 10 to 50 slots have been used, all of them being configured to deal with the deterministic traffic of three packets per slotframe. The PDR measured in our setup was close to 100%. Fig. 4 shows the results compared with the model adopted, as well as the predicted current consumption in environments with lower PDR.

B. Data Acquisition Profiling

In general, to estimate the energy required in collecting one sample, we have to look at a breakdown of the microprocessor/sensor activity during that sample. We have represented in Fig. 5 the following tasks.

- Sensor startup time (T_{SETT}).
- Processor and sensor energy required to initiate sampling (T_{WUP}). Could correspond to toggling a pin or sending a command on a communication bus such as I2C and SPI.

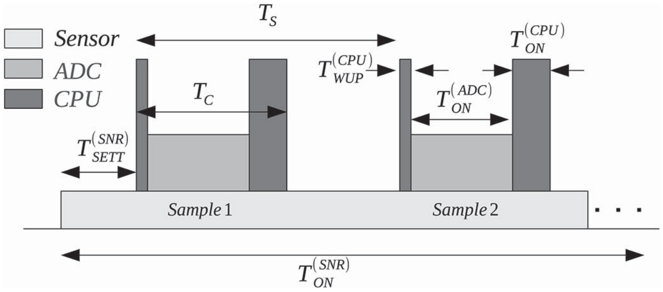

Fig. 5. Energy consumption breakdown for one sample.

TABLE I
Time Required by the Acquisition of One Single Sample, Broken Down in Different Tasks. Estimated Conversion Time $\hat{T}_C$ Is Compared With Measured Time T_C

Task[2]	Cycles(MHz)	$\hat{T}[us]$	$T[\mu s]$	$\delta T[\%]$
Sample and Hold (ADC)	5×13 (5)	13.0		
A-D conversion (ADC)	5×16 (5)	16.0		
Samples Storage (CPU)	164(16)	10.2		
System wake-up		1.1		
Total Conversion T_C		40.3	39.4	2.2

- Analog-to-digital conversion (T_{ADC}). This happens on the sensor if it is a digital one, on the microprocessor if the sensor is analog.
- Transfer of data from the ADC output to the processor memory (T_{CPU}). This task could be an internal memory operation or an external serial/parallel communication.

In our application example, we are dealing with an analog sensor that is turned on from the microprocessor and that provides a reading through the processor's ADC. The sensor startup time ($T_{\mathrm{SETT}} \approx 5$ ms) is too long to turn it on and off between consecutive samples ($T_S = 4$ ms at $F_S = 250$ Hz). Therefore, the sensor is required to remain in its *on* state while collecting a record of samples. Consequently, sensor and processor must be considered independently.

The charge drained by the sensor in the data acquisition stage is computed using the total time the sensor must remain active while capturing N_S samples with a sampling period T_S. Then, the total charge is given by $Q_{\mathrm{SNR}} = I_{\mathrm{SNR}} \cdot N_S \cdot T_S$. On the other hand, as the ramp-up time of the CPU is fast enough, it can operate on a sample-by-sample basis. For each sample, the energy contribution can be obtained by looking at the number of cycles required by the internal ADC and the cycles required to move the data to memory. The time required to sample five ADC channels, convert them, and commit them to memory is represented in Table I, along with the total time T_C. Since each conversion is independent, the total charge required is proportional to the active time at each capture: $Q_{\mathrm{ADC}} = I_{\mathrm{CPU}} \cdot T_C \cdot N_S$.

When combining both contributions, simple linear dependence on the number of samples is obtained, i.e., $I_{\mathrm{ACQ}} \propto N_S$. Finally, the average current of the acquisition block is computed by dividing the total charge by the time lapsed between consecutive records, as shown in the following:

$$\bar{I}_{\mathrm{ACQ}} \cong \frac{N_S \cdot (T_S \cdot I_{\mathrm{SNR}} + T_C \cdot I_{\mathrm{CPU}})}{T_{\mathrm{RCD}}}. \tag{7}$$

TABLE II
TIME AND CHARGE CONSUMED IN FFT PROCESSING. SIMULATED RESULTS $\hat{T}_{\mathrm{MAT}}$ AND $\hat{T}_{\mathrm{ISS}}$ ARE COMPARED WITH MEASURED T_{PRC}

N_{FFT}	$T_{PRC}[ms]$	$\hat{T}_{MAT}[ms]$	$\delta T_{MAT}[\%]$	$\hat{T}_{ISS}[ms]$	$\delta T_{ISS}[\%]$
64	14.6	14.3	2.1	14.7	0.4
128	33.5	33.2	0.8	33.5	0.0
256	75.3	75.8	0.6	75.4	0.1
512	167.8	170.2	1.4	168.0	0.1
1024	369.8	377.8	2.1	370.3	0.1

In this block, two control parameters appear, the number of samples per record N_S and the time between consecutive records T_{RCD}.

C. Data Processing Profiling

Here, we study the contribution to the energy cost of the data processing. The vibration monitoring application is mainly concerned with the frequency content of the acceleration signals. As such, a *fast Fourier transform* (FFT) is required to find the relevant dominant frequency and harmonics in the signal. Two different approaches were addressed to obtain a proper estimation of the computation time of the FFT. First, a MATLAB implementation of the algorithm was used to estimate the number of cycles required, following the method used in [22]. In the second approach, the code was moved to a device-specific implementation and was simulated with the IAR Workbench instruction set simulator.

Table II compares both approaches, showing that the time estimated through the number of MATLAB operations T_{MAT} did not differ significantly from that derived from the simulator T_{ISS}, and the latter is in perfect agreement with the time measured with the algorithm running on the MSP430 processor T_{PRC}. With this example, we show that it is possible to estimate the number of cycles required by the data processing algorithms at design stage, even without the final source code implementation. For further information, please refer to [22]. This processing time is used to obtain a fairly accurate energy consumption estimation, which is critical when designing the final solution over the platform.

For modeling purposes, it is important to identify functional dependence on a controlled parameter. The Radix-2 implementation of the FFT has a well-known $N\log(N)$ complexity. The associated energy cost in (8) is proportional to this relation, being $\bar{Q}_{\mathrm{OP}}$ a magnitude representing the average cost per operation. Fig. 6 shows the fitting of (8) to the experimental data. Thus,

$$\bar{I}_{\mathrm{PRC}} \simeq \frac{\bar{Q}_{\mathrm{OP}} \cdot N \cdot \log(N)}{T_{\mathrm{RCD}}}. \quad (8)$$

D. Joint Model Validation

Once individual energy consumption contributors have been profiled, a joint model can be built as a tool to understand the main contributors on energy consumption for a specific application setting, as represented in Fig. 7.

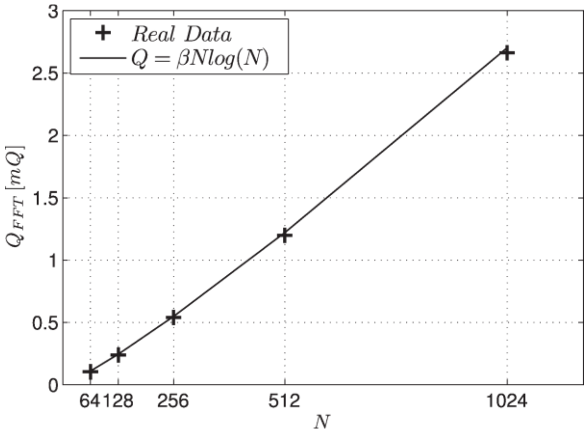

Fig. 6. Charge drained to compute an N-point FFT. A numerical value for β can be found in Table III.

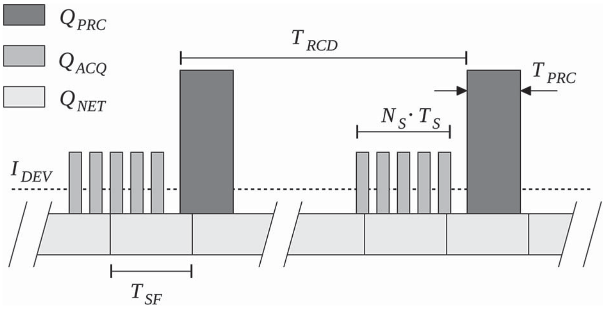

Fig. 7. Application charge drained by different components.

TABLE III
TECHNOLOGICAL PARAMETERS OBTAINED BY MODELING COMPARED WITH EXPERIMENTAL DATA FITTING

Contribution	Parameter	Model	Fitting	Units
Acquisition	α	3.50	3.49	$[\mu C]$
Processing	β	0.269	0.263	$[\mu C]$
Network (Radio)	γ	68.9	63.9	$[\mu C]$
Network (System)	δ	0.433	0.475	$[mA]$

As asserted before, the main parameter involved in network consumption is N_{SLOTS}, which is related to the number of active and sleep slots. Assuming a fix number of active slots, by incrementing N_{SLOTS}, we are introducing sleep slots to the schedule and therefore reducing the average consumption.

In terms of sensing and processing, two remarks should be made. First, a record is defined as the process of waking up, taking N_S samples and computing an FFT to analyze them. We assume that the number of points computed by the FFT and the number of samples taken by the ADC are the same, i.e., $N_S \doteq N$. This means that the number of points in a record is a parameter that affects simultaneously the energy expenditure of both sensing and processing procedures. Second, once the number of points to be sampled and analyzed is fixed, the duty-cycled behavior of the application makes the average power depend directly on the time between records T_{RCD}. As the time between records is increased, less power is consumed. Therefore, the time interval between consecutive records gives us the timescale for power averaging.

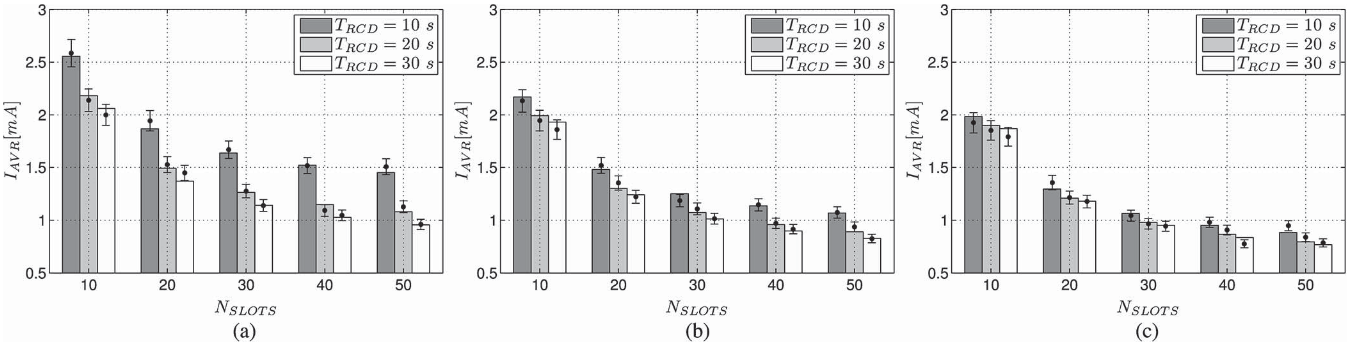

Fig. 8. Comparison of model predictions and experimental results for different parameter configurations. (a) N_{FFT} = 1024. (b) N_{FFT} = 512. (c) N_{FFT} = 256.

Equation (9) combines the three contributions (6)–(8), differentiating between technological and applications parameters, i.e.,

$$\bar{I}_{\mathrm{DEV}} = \frac{\alpha N}{T_{\mathrm{RCD}}} + \frac{\beta N \log(N)}{T_{\mathrm{RCD}}} + \frac{\gamma N_{\mathrm{ACT}}}{T_{\mathrm{SLOT}} N_{\mathrm{SLOTS}}} + \delta. \quad (9)$$

Constants α, β, γ, and δ only depend on the particular choice of sensor, microcontroller unit, and radio technologies, respectively. From the individual contribution, α represents the charge per sample $\bar{Q}_S$, β is interpreted as a cost per operation $\bar{Q}_{\mathrm{OP}}$, whereas γ is an estimator of the average charge per message $\bar{Q}_{\mathrm{MSG}}$. These constants can be easily modified to evaluate alternative technologies. In turn, N_{SLOTS}, N, and T_{RCD} are application parameters that can be tuned in order to meet the specifications, once the specific technology is established.

Table III compares the value of α, β, γ, and δ obtained by modeling with the experimental data fitting for a GINA mote running the OpenWSN protocol stack. With this technology fixed, Fig. 8 shows the experimental results compared with those predicted by (9) for different application configurations. The dots in the figures represent the measured current values, whereas the bars around them represent an allowable deviation of 5% due to variations in the PDR or other environmental conditions. The vertical bars show the estimated values by the model, plotted for different numbers of slots in a slotframe N_{SLOTS}, different recording intervals T_{RCD}, and different numbers of samples collected and processed N. These results demonstrate the correctness of the model.

V. Model in Action

By using this model, the applications engineer can make better informed technology-related decisions (both hardware and software) at the design stage. This methodical approach to wireless sensing allows for a reduction in the prototyping stage, as well as a quicker route to successful deployments.

This section discusses how the presented model can be used for the following: 1) understanding the energy balance of an application; 2) assessing on the energy harvester selection; and 3) coping with scavenging dynamics by adapting the application behavior.

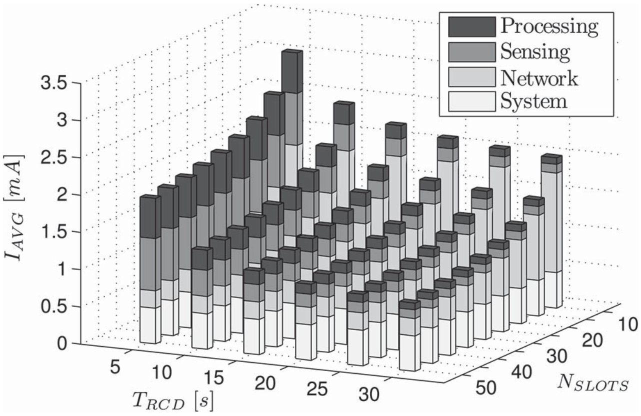

Fig. 9. Contribution of recording interval and network configuration using 1024 points for the FFT (processing time is fixed).

A. Application Energy Balance

Experimental measurements on a system only provide the energy consumed by the entire system without giving any knowledge about the distribution of the consumption among the different subsystems. This makes it difficult to identify the main contributors to the energy consumption and the parameters on which depends the application, thereby limiting the scope of energy optimizations. The presented model, however, enables us to determine the amount of energy consumed by each of the subsystems in the application and understand the energy balance between components.

Fig. 9 presents a simulation obtained by applying (9) to different network and recording period configurations, considering 1024 samples per record. Bars present the contribution to the energy consumption of the network, sampling and processing components, according to the number of slots per slotframe and the recording interval.

An asymptotic behavior can be seen in both axes in Fig. 9. Holding the interval time fixed, the overall energy consumption is reduced when increasing the number of slots in a slotframe. However, the asymptotic decrease limits the amount of energy that can be saved. At a certain point, increasing the number of slots in the network does not significantly reduce the energy consumed. Analogously, as the recording interval increases, the energy savings decrease.

This graphical representation can be used as a tool to determine which of the parameters yields the highest energy savings once optimized.

B. Scavenging

Scavenger selection is guided by the condition defined in (1), and a desirable situation comes when the $\bar{P}_{SCV} \geq \bar{P}_{DEV}$ condition arises. Dimensioning of the scavenger depends on the configuration and application requirements of the wireless sensor devices. On one hand, a fixed network and application configuration can easily be used to draw the upper bound of $\bar{P}_{DEV}$ and therefore select the right size for the scavenger. On the other hand, given a particular scavenger, the network and application parameters can be tuned to meet the available average current condition. To draw an example of the second approach, a GINA mote was connected, as shown in Fig. 3. The vibration of an industrial pump has been measured for 24 h, in order to characterize the variability in the amplitude of vibrations and so the energy produced.

The fundamental harmonic is defined by the specific frequency that provides the main contribution to the harvested energy. That should be the nominal frequency of the selected harvester. Fig. 10(a) shows the spectrogram obtained for a chunk of 1 h. In this particular application, the maxim peak of the spectrum has been found at 60 Hz. At this frequency, the measured amplitude of acceleration is almost constant in time: $\bar{a} \pm \sigma_a = 0.0985 \pm 0.0016$ [g]. In our example, we selected the PMG FSH60x1 model, with a resonator adjusted at 60 Hz. This device provides a current of $\bar{I}_{SCV} \approx 1.6$ mA for a 0.98 [g] [see Fig. 10(b)]. Based on this response, Fig. 10(c) shows the suitable subset of parameters to make the application self-sustainable, defined by the region above the white curve. The dashed lines show the expected current inside 3σ limits.

C. Dynamic Scavenging

In many applications, the incoming energy to the system is only available intermittently. In other cases, the scavenger cannot provide the expected energy because some environmental variables have changed. In both cases, the application should be able to dynamically adapt to the new conditions to remain energetically sustainable. This is accomplished by selecting a suitable parameter and mapping the consumption to the expected energy input. In our application, the number of slots is fixed once the network has been established (the length of the slotframe cannot be changed without involving a cascade of changes in other nodes). Active slots can be deactivated and converted to sleep slots, therefore reducing the throughput of the node. However, this comes with a considerable energetic cost to the network as it might trigger certain rescheduling in other nodes. Then, a more suitable control parameter is the time lapse between records T_{RCD}. Alternatively, the number of points to compute the FFT can be also reduced, thus compromising the quality of the harmonic analysis.

To dynamically adjust energy consumption to variations of the energy scavenged, the application needs to keep track of the amplitude of the fundamental vibrational harmonic, analyze it, and estimate the expected input current for the next cycle accordingly.

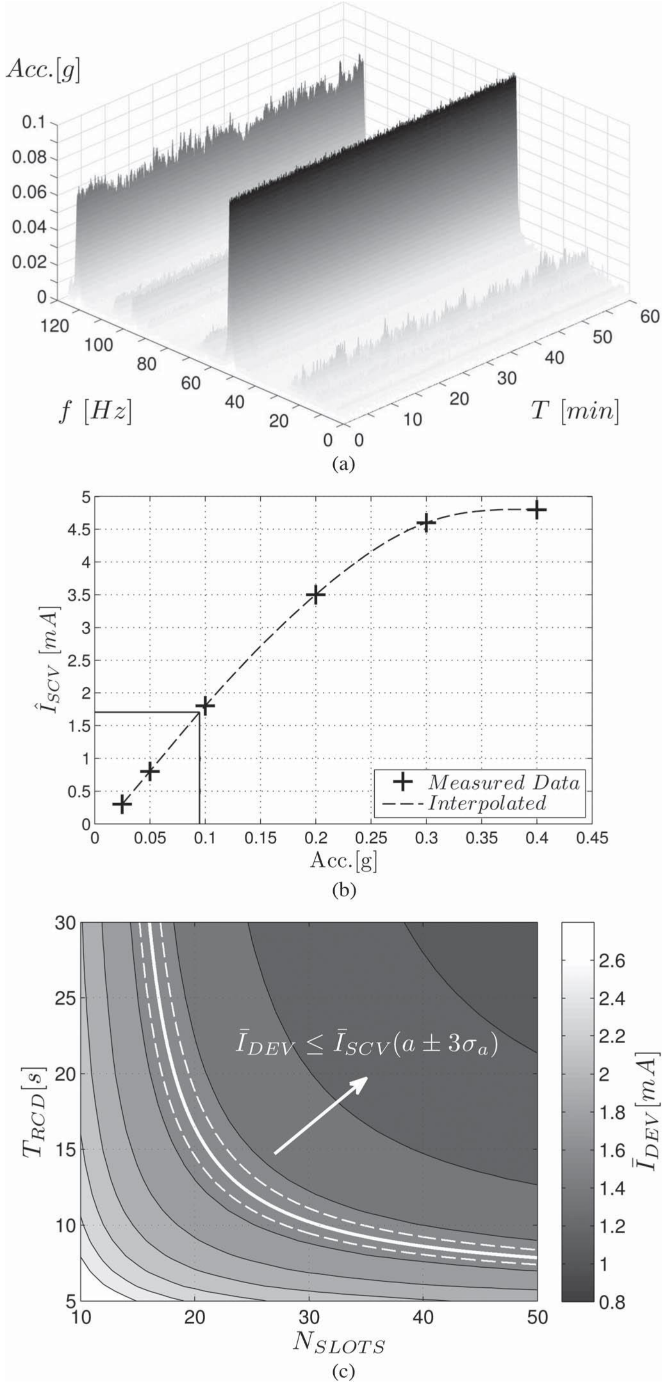

Fig. 10. Configuration of application settings according to available current. (a) Spectrogram of the vibration magnitude during 1 h of monitoring. (b) Vibration amplitude mapped to the current supplied by the harvester. (c) Grayscale of the side bar represents the device average current. Given the estimated production of the specific harvester (1.5 mA in this example), the white line limits the self-powered region, that is, an acceptable combination of parameters that guarantee an autonomous operation.

With this information, the time for the next record can be scheduled as follows. Fig. 11 shows the operational regions for different network schedules. To select the interleaving time, the vibrational amplitude is analyzed, and the expected normalized

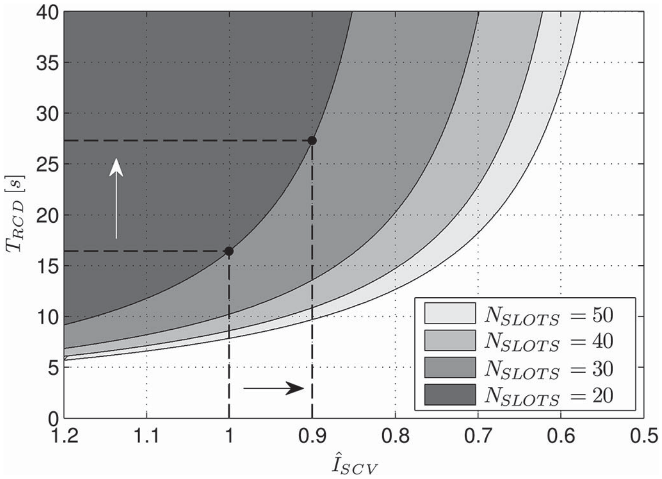

Fig. 11. Dependence of the time between records and the normalized harvester current for different network configurations. Gray areas represent the feasible zones. The system must react to a reduction in the (black arrow) harvested current with an increase in the (white arrow) time lapsed between records.

current is computed. This value is used to determine the timeout for the next wake-up by finding the T_{RCD} in the border of the region.

VI. Conclusion

This paper has addressed the convergence of energy scavengers with industrial wireless sensing and actuating applications. A methodology based on a parameterizable model has been presented in order to understand energy spending and facilitate scavenger selection. The methodology aims to reduce technology adoption/integration risks as energy consumption can be precisely estimated without the need of prototyping or building actual devices. We show that, by decomposing the sources of energy consumption, optimization can be done in a more accurate manner and system designer efforts can be put in the right direction, consequently minimizing costs and risk. A precise estimation also enables a clear dimensioning of the components of the devices, particularly with respect to the energy buffer and harvester. It has been shown that TSCH networks facilitate the energy profiling due to their determinism and slotted structure.

Throughout this paper, we make use of an industrial application to illustrate how an analytical model emerges as a tool to facilitate the application configuration and scavenger selection at predeployment stages. Given a TSCH-based wireless application and knowing its bandwidth, sampling, and processing requirements, an accurate estimation of energy demands is used to determine what scavenger is required to make it self-sustainable. In addition, the parametrized model can be used to enable different modes of operation in case of varying requirements. The presented model has been experimentally validated using a wireless sensor network platform running the OpenWSN protocol stack.

The presented methodology can be easily applied to a wide range of applications and sources of energy and can be summarized in five simple steps.

1) Find a source of energy in your environment.
2) Measure the magnitude of the available energy and its duty cycle.
3) Run a simulation within the feasible limits of application parameters.
4) Select a suitable harvester sized accordingly to the energy available and the first estimation of the application consumption.
5) Fine-tune the application parameters accordingly with the selected harvester.

References

[1] *WirelessHART Specification 75: TDMA Data-Link Layer*, HCF_SPEC-75, 2008.

[2] *ISA-100.11a-2011: Wireless Systems for Industrial Automation: Process Control and Related Applications*, ISA, Madrid, Spain, May 2011.

[3] *IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, 802.15.4e-2012, Apr. 16, 2012.

[4] L. Doherty, W. Lindsay, and J. Simon, "Channel-specific wireless sensor network path data," in *Proc. ICCCN*, 2007, pp. 89–94.

[5] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[6] D. Steingart and J. Polastre, "Energy harvesting white paper," White Paper, 2008.

[7] A. Khaligh and P. Zeng, "Kinetic energy harvesting using piezoelectric and electromagnetic technologies: State of the art," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 850–860, Mar. 2010.

[8] R. Torah *et al.*, "Self-powered autonomous wireless sensor node using vibration energy harvesting," *Meas. Sci. Technol.*, vol. 19, no. 12, p. 125202, Dec. 2008.

[9] A. Waterbury and P. K. Wright, "Vibration energy harvesting to power condition monitoring sensors for industrial and manufacturing equipment," *Proc. Inst. Mech. Eng., J. Mech. Eng. Sci.*, vol. 227, no. 6, pp. 1187–1202, Jun. 2013.

[10] J. Carmo, L. Goncalves, and J. Correia, "Thermoelectric microconverter for energy harvesting systems," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 861–867, Mar. 2010.

[11] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Proc. 3rd Annu. IEEE Commun. Soc. SECON*, 2006, vol. 1, pp. 286–295.

[12] V. Liu *et al.*, "Ambient backscatter: Wireless communication out of thin air," in *Proc. ACM SIGCOMM Conf.*, New York, NY, USA, 2013, pp. 39–50.

[13] B. Lu and V. Gungor, "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4651–4659, Nov. 2009.

[14] A. Nasiri, S. Zabalawi, and G. Mandic, "Indoor power harvesting using photovoltaic cells for low-power applications," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4502–4509, Nov. 2009.

[15] Y. Tan and S. Panda, "Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4424–4435, Sep. 2011.

[16] M. Magno *et al.*, "Extended wireless monitoring through intelligent hybrid energy supply," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 1871–1881, Apr. 2014.

[17] M. Mussetta, H. Shadmehr, F. Grimaccia, A. Gandelli, and R. Zich, "Optimization of a radio frequency energy harvesting device," in *Proc. IEEE CEC*, 2012, pp. 1–5.

[18] R. D'hulst, T. Sterken, R. Puers, G. Deconinck, and J. Driesen, "Power processing circuits for piezoelectric vibration-based energy harvesters," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4170–4177, Dec. 2010.

[19] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proc. 4th Int. Symp. IPSN*, Piscataway, NJ, USA, 2005, IEEE Press.

[20] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. Pister, "Adaptive synchronization in IEEE802.15.4e networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 795–802, Feb. 2014.

[21] X. Vilajosana *et al.*, “A realistic energy consumption model for TSCH networks,” *IEEE Sens. J.*, vol. 14, no. 2, pp. 482–489, Feb. 2014.

[22] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, “On the performance of lossy compression schemes for energy constrained sensor networking,” *ACM Trans. Sens. Netw.*, vol. 11, no. 1, pp. 15:1–15:34, Aug. 2014.

[23] A. Muszynska, “Vibrational diagnostics of rotating machinery malfunctions,” *Int. J. Rotating Mach.*, vol. 1, no. 3/4, pp. 237–266, 1995.

[24] A. Mehta and K. Pister, “WARPWING: A complete open-source control platform for miniature robots,” in *Proc. IEEE Int. Conf. IROS*, 2010, pp. 5169–5174.

[25] T. Watteyne *et al.*, “OpenWSN: A standards-based low-power wireless development environment,” *Trans. Emerging Telecommun. Technol.*, vol. 23, no. 5, pp. 480–493, Aug. 2012.