A multi-layered energy consumption model for smart wireless acoustic sensor networks

Gert Dekkers*[a,b], Fernando Rosas[c,d], Steven Lauwereins[b], Sreeraj Rajendran[b], Sofie Pollin[b], Bart Vanrumste[b], Toon van Waterschoot[b], Marian Verhelst[b], and Peter Karsmakers[a]

[a] *Department of Computer Science, KU Leuven, Belgium*
[b] *Department of Electrical Engineering, KU Leuven, Belgium*
[c] *Department of Mathematics, Imperial College London, UK*
[d] *Department of Electrical and Electronic Engineering, Imperial College London, UK*

February 5, 2019

Abstract

Smart sensing is expected to become a pervasive technology in smart cities and environments of the near future. These services are improving their capabilities due to integrated devices shrinking in size while maintaining their computational power, which can run diverse Machine Learning algorithms and achieve high performance in various data-processing tasks. One attractive sensor modality to be used for smart sensing are acoustic sensors, which can convey highly informative data while keeping a moderate energy consumption. Unfortunately, the energy budget of current wireless sensor networks is usually not enough to support the requirements of standard microphones. Therefore, energy efficiency needs to be increased at all layers — sensing, signal processing and communication — in order to bring wireless smart acoustic sensors into the market.

To help to attain this goal, this paper introduces *WASN-EM*: an energy consumption model for wireless acoustic sensors networks (WASN), whose aim is to aid in the development of novel techniques to increase the energy-efficient of smart wireless acoustic sensors. This model provides a first step of exploration prior to custom design of a smart wireless acoustic sensor, and also can be used to compare the energy consumption of different protocols.

Keywords: wireless sensor networks, smart acoustic sensing, energy consumption model.

1 Introduction

Recent advances in hardware miniaturization is enabling integrated devices containing wireless radios, processing and sensing capabilities to shrink their sizes, while their computational power is maintained or even increased [1]. This, along with a recent surge of powerful Machine Learning algorithms that can accomplish various data-driven tasks, has caused a rising interest in smart cities and environments. Such a smart environment typically uses a network of wireless sensors for acquiring information to offer smart functionality [2]. Scenarios where this have been taking place include security, smart city, health monitoring and entertainment [2–6].

An attractive type of sensor to use in these applications are acoustic sensors (i.e. microphones). Compared to other sensors, acoustic sensors can convey highly informative data, including sounds with semantic contect (e.g. speech), noises that represent a warning (e.g. screams), sounds with intrinsic

*For questions related to the model: gert.dekkers@kuleuven.be

meaning in a particular environment (e.g. a water faucet running within a kitchen), etc. Detecting these informative acoustic events can be benefitial for numerous tasks, including speech recognition, surveillance and monitoring, and many others [7].

In order to allow an wireless acoustic sensor network (WASN) to be easily installed, wireless battery-powered architectures are preferred to avoid extensive use of wiring [8]. Unfortunately, this brings additional challenges as the lifetime of the devices can be compromised by the energy consumption of acoustic sensors and wireless transmission, which usually go beyond what common current sensor network architectures can provide [9].

Increasing the energy efficiency of sensors can be tackled from the different layers of the processing chain, including the sensing, signal processing and communication modules [10]. In effect, the total amount of consumed energy depends strongly on particular parameters related to hardware dependency. Ideally, one would optimize these parameters for each particular hardware design, but in practice such approach would require tedious measurement campaings. From a signal processing point of view, energy is often reduced by means of limiting the amount of arithmetic operations. Such an approach might not always be valid due to costly memory accesses. Additionally, if the goal is to design a smart wireless acoustic sensor, it is important to know the energy contribution of each layer to motivate an optimization.

The literature provides a number of modeling efforts about various aspects of wireless sensors networks (see [11] and references therein). Substantial efforts has been done in increasing the energy efficiency of the wireless communication module, ranging from the physical layer [12], multihop and routing [13, 14] and network layer protocols [15, 16]. Regarding the processing of audio information to retrieve relevant information, deep learning has recently become popular [17–19]. Yang et al have introduced an energy estimation model to estimate the energy consumption of Deep Neural Networks (DNN) [20]. The model is based on power numbers from their Eyeriss DNN accelerator chip and provides an estimate on the energy consumption given an optimized dataflow [21]. The disadvantage of the model is that it is not flexible as it is limited to the bounds of that particular chip. To the best of our knowledge, no open-source model is available that cover all layers of a smart acoustic sensor.

In order to aid the design of novel configurations for allowing the increase of energy efficiency of a range of sensing devices, in this report we introduce the *WASN-EM*: an Energy Model for Wireless Acoustic Sensor Networks. The goal of the model is threefold:

(a) to bridge the gap between the machine learning and hardware community regarding (energy-efficient) design of smart wireless acoustic sensors,
(b) to be flexible to adjust to various hardware configurations, and
(c) to provide a simple and open-source software [22] such that the community can contribute.

The model can act as a first step prior to custom design of a smart wireless acoustic sensor and provide a common ground for researchers to compare energy consumption, computational complexity and memory storage.

In the sequel, an overview of the proposed model is provided in section 2 as it is composed out of three separate models: sensing, processing and communication. Section 3 elaborates on the modelling of the sensing layer which includes the microphone, power amplifier and analog-to-digital converter. Section 4 covers the processing layer where an hardware architecture model is introduced that provides an energy consumption estimate of the arithmic operations and the accompanying memory accesses. Additionally, some common algorithms for processing audio information are introduced of which an energy consumption estimate can be obtained using the proposed hardware architecture model. Section 5 elaborates on the model of the communication layer. The model covers the power amplifier and other electronic components based on a hypothetical hardware architecture along with the effects of re-transmission. The final section provides some guidelines on interesting parameters to experiment with.

2 System model

Let us consider an scenario where the goal is to monitor a particular environment to acquire information about the activities that are taking place. This could correspond to an apartment, where by determining the activities that are taking place an automated system could optimize a range of services including lighting, heating, etc. One way to harvest information about an environment is to deploy a WASN consisting of multiple acoustic sensors nodes with wireless communication capabilities, and a central connection/processing device that can gather and process the sensed data (see Figure 1). Each node consists out of a acoustic sensing, processing and communication module, being capable of:

(1) capturing and digitizing acoustic information,
(2) processing the resulting acoustic data to provide a meaningfull output and/or to reduce the amount of bits to communicate,
(3) transmitting the processed information with a central connection/processing point, and
(4) receiving data from a central connection/processing point.

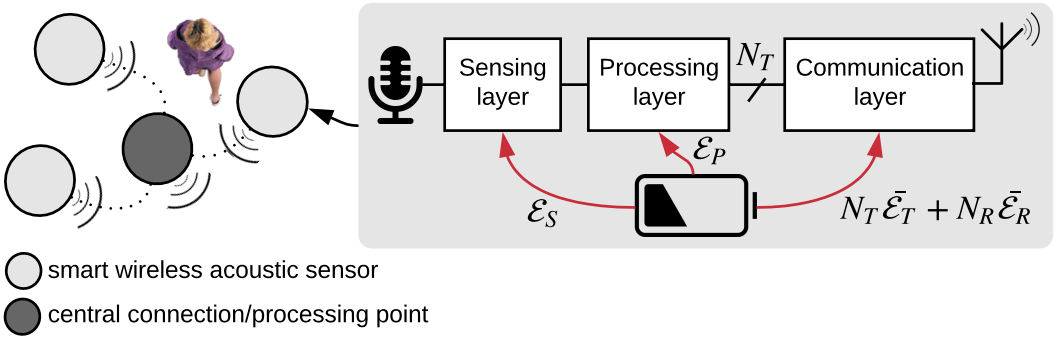


Figure 1: Scenario description

Given the aforementioned scenario, let us consider a single duty cycle where a single node measures the environment during Δ seconds and subsequently does some processing on the data. Consequently, this generates N_T bits of information, spending $\mathcal{E}_{\mathrm{S}}$ and $\mathcal{E}_{\mathrm{P}}$ joules in the sensing and data processing step respectively. The processed information is divided in $N_T/(r_{\mathrm{u}}L_{\mathrm{u}})$ forward frames, where L_{u} is the number of payload bits per frame in the uplink direction and r_{u} is its code rate, which are transmitted directly to a central connection point (sink) using designated time-slots. After each frame transmission trial, the sink send back a feedback frame which acknowledges correct reception or requests a re-transmission. Similarly, the communication module can receive $N_R/(r_{\mathrm{d}}L_{\mathrm{d}})$ frames where N_R is total amount of received informative bits and L_u and r_u are the number of payload bits and the code rate in the downlink direction. Hence, the total energy consumption of the audio sensor node can be modeled as follows:

$$\bar{\mathcal{E}}_{\mathrm{node}} = \mathcal{E}_{\mathrm{S}} + \mathcal{E}_{\mathrm{P}} + N_T\bar{\mathcal{E}}_{\mathrm{T}} + N_R\bar{\mathcal{E}}_{\mathrm{R}} \ . \tag{1}$$

Above, $\bar{\mathcal{E}}_{\mathrm{T}}$ and $\bar{\mathcal{E}}_{\mathrm{R}}$ is the average total energy consumption per information bit that is correctly transmitted and respectively received.

Let us assume that the node has to be sensing the environment δ percent of the time (i.e. its duty cycle). Let us also assume that the node carries n_{b} batteries with a charge of $\mathcal{B}$ Joules each. Then, by neglecting the energy consumption of the node when it is in sleep mode, the lifetime of the node can be estimated to be equal to

$$\mathcal{L} = \delta^{-1}\frac{n_{\mathrm{b}}\mathcal{B}}{\bar{\mathcal{E}}_{\mathrm{node}}}\Delta \ . \tag{2}$$

In the following, we will develop expressions for $\mathcal{E}_{\mathrm{S}}$, $\mathcal{E}_{\mathrm{P}}$ and $\mathcal{E}_{\mathrm{T}}$.

3 Energy consumption of sensing

The energy consumption expended in acoustic sensing can be expressed as follows:

$$\mathcal{E}_{\mathrm{S}} = \mathcal{E}_{\mathrm{mic}} + \mathcal{E}_{\mathrm{LNA,mic}} + \mathcal{E}_{\mathrm{ADC,mic}} \ . \tag{3}$$

Above, the energy consumed by the analog front-end $\mathcal{E}_{\mathrm{S}}$ consists out of the energy consumed by the microphone $\mathcal{E}_{\mathrm{mic}}$, the consumption of the low-noise amplifier (LNA) $\mathcal{E}_{\mathrm{LNA,mic}}$ and the consumption of the analog-to-digital convertor (ADC) in digitalising the signal $\mathcal{E}_{\mathrm{ADC}}$.

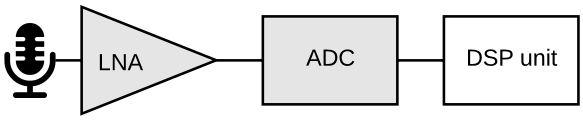


Figure 2: Microphone analog front-end (marked in grey) along with the processing layer

The power consumption of the microphone can be expressed as follows:

$$P_{\mathrm{mic}} = \begin{cases} 0 & \text{if passive mic or switched off,} \\ P_{\mathrm{mic,act}} & \text{if active and powered on,} \end{cases} \tag{4}$$

so, the energy consumption will be given by

$$\mathcal{E}_{\mathrm{mic}} = \begin{cases} 0 & \text{if passive mic, or} \\ \Delta P_{\mathrm{mic,act}} & \text{if active.} \end{cases} \tag{5}$$

The energy consumption of the LNA can be calculated as

$$\mathcal{E}_{\mathrm{LNA,mic}} = I_{\mathrm{LNA,mic}} V_{\mathrm{dd}}^{\mathrm{LNA,mic}} \Delta \ . \tag{6}$$

Above, $V_{\mathrm{dd}}^{\mathrm{LNA,mic}}$ is the voltage supply level and $I_{\mathrm{LNA,mic}}$ is the average current drawn by the LNA, which can be calculated as

$$I_{\mathrm{LNA,mic}} = \frac{\pi u_{\mathrm{T}} 4kTW_{\mathrm{ADC}}}{2} \left(\frac{\mathrm{NEF}}{v_{\mathrm{n,in}}^{\mathrm{rms}}} \right)^2 \ , \tag{7}$$

where k is the Boltzmann constant, T is the temperature in Kelvin, $u_{\mathrm{T}} = kT/q_{\mathrm{e}}$ with q_{e} equal to the charge of the electron, W_{ADC} is the ADC bandwidth, $v_{\mathrm{n,in}}^{\mathrm{rms}}$ is the RMS voltage of the noise at the input of the LNA, and NEF is the *noise efficiency factor*, which was proposed in [23] and which value in average designs is between 5-10. Typical values for $v_{\mathrm{n,in}}^{\mathrm{rms}}$ are

$$v_{\mathrm{n,in}}^{\mathrm{rms}} = \begin{cases} 10\mu W & \text{for passive microphones} \ , \\ 100\mu W & \text{for active microphones} \ . \end{cases} \tag{8}$$

The energy consumption of the ADC can be computed as

$$\mathcal{E}_{\mathrm{ADC,mic}} = P_{\mathrm{ADC,mic}} \Delta \ . \tag{9}$$

Above, $P_{\mathrm{ADC,mic}}$ is the power consumption of the ADC, which can be calculated as

$$P_{\mathrm{ADC,mic}} = 2^{n_{\mathrm{mic}}} f_{\mathrm{s,mic}} \mathrm{FOM} \ , \tag{10}$$

where n_{mic} is the resolution of the ADC, $f_{\mathrm{s,mic}}$ is the sampling frequency and FOM is the *figure of merit* of the ADC.

An overview of other relevant hardware related parameters with the used values is given in Appendix A Table 2.

4 Energy consumption of processing

The goal of the (optional) local processing is to translate the raw audio information to a lower dimension to reduce the amount of communicated bits. The processed information could already be the final required output (e.g. a classification output) or the output of a feature extraction stage. From a signal processing point of view, energy consumption is often reduced by means of limiting the amount of arithmetic operations. Such an approach might not always be valid due to costly memory accesses. Here the energy consumption $\mathcal{E}_{\mathrm{P}}$ due processing of the acquired information is defined as

$$\mathcal{E}_{\mathrm{P}} = \overbrace{\mathcal{E}_{\mathrm{cc}}\Big(\sum_{j=1}^{J_{\mathrm{ALU}}} c_j n_j^{\mathrm{DSP}} + \sum_{k=1}^{J_{\mathrm{MEM}}} \tau_{m,k}(M_{\mathrm{a},k} + M_{\mathrm{s},k})\Big)}^{\mathcal{E}_{\mathrm{op}}} + \overbrace{\Delta \sum_{k=1}^{J_{\mathrm{MEM}}} (\mathcal{E}_{\mathrm{ma,k}} M_{\mathrm{a},k} + \mathcal{E}_{\mathrm{ms,k}} M_{\mathrm{s},k})}^{\mathcal{E}_{\mathrm{m}}} \ , \tag{11}$$

which consists of the consumption due to arithmetic operations $\mathcal{E}_{\mathrm{op}}$ and due to memory $\mathcal{E}_{\mathrm{m}}$.

Regarding the energy consumed due to the amount of arithmetic operations $\mathcal{E}_{\mathrm{op}}$, this is composed out of the amount of clock cycles spend in a) performing a set of arithmetic operations and b) waiting for a particular memory access where $\mathcal{E}_{\mathrm{cc}}$ denotes the energy consumption per clock cycle. In case of arithmetic operations, c_j is the number of clock cycle required by the j-th arithmetic operation which is performed n_j^{DSP} times during the digital signal processing and J_{ALU} is the number of different arithmetic operations the microprocessor performs. In case of memory latency, $\tau_{m,k}$ refers to the memory access time, $M_{\mathrm{a},k}$ the amount of bits accessed or stored $M_{\mathrm{s},k}$ in a particular memory k. In the model we distinguish the following arithmetic operations: 1) multiply-accumulate (MAC), 2) addition and subtraction, 3) multiplication, 4) division, 5) comparison (including maximum and minimum), 6) natural exponentiation and 7) logarithm. Depending on the hypothetical hardware architecture (e.g. CPU, ASIC, ...), each of these operations take a different amount of clock cycles and energy cost per clock cycle. This model focusses on a microcontroller-based wireless acoustic sensor without any hardware acceleration. A model for a custom chip is not provided but in general these could provide energy gains of a factor 500 till 1000 for the processing layer [24].

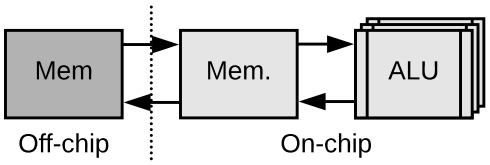


Figure 3: The hardware architecture model

The energy consumed by the memory $\mathcal{E}_{\mathrm{m}}$, is decomposed into the energy required for accessing and storing which depends on the amount of bits accessed $M_{\mathrm{a},k}$ or stored $M_{\mathrm{s},k}$ in a particular memory k. The energy consumed by accessing and storing one bit is defined by $\mathcal{E}_{\mathrm{ma,k}}$ and storage $\mathcal{E}_{\mathrm{ms,k}}$ respectively for $k = [0, 1, \ldots, J_{\mathrm{MEM}}]$ with J_{MEM} the amount of available memories. Typical hardware architectures have multiple memory types available each having a different energy consumption for storage and access. The least consuming memory is typically close to the processor unit but limited in size. When the needed memory is not sufficient, data movement is needed to and from more consuming memories. It is therefore important to maximize data reuse on the least consuming memories to limit data movement [25]. In this model we assume: a) an architecture with on- and off-chip memory as shown by Figure 3, b) equal energy cost for each operation per clock cycle b) equal cost for memory read and writes and c) in-place computation such that memory accesses can easily be derived from a particular arithmetic operation. Additionally, it is explicitly defined where the information should be stored/accessed for each building block in the processing chain. An overview of the parameters of the hardware architecture model are

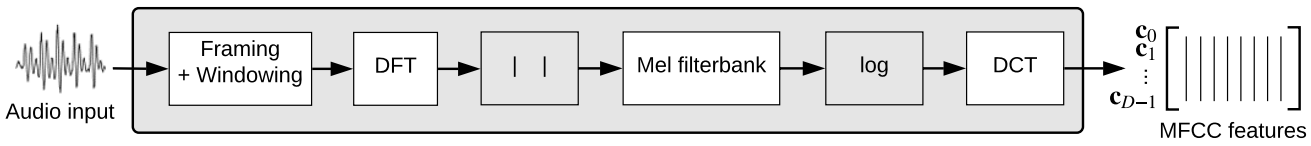


Figure 4: Mel-Frequency Cepstral Coefficients feature extraction process. The raw acoustic data is transformed to the feature domain by applying a (1) framing and windowing, (2) Discrete Fourier Transform, (3) Mel-filterbank, (4) logarithm and (5) Discrete Cosine Transform.

given in Appendix A Table 3. The amount of clock cycles is obtained by testing on a Cortex M4 device and verified by checking the instruction set.

In the following subsections the explanation and an energy model for some typical algorithms used in the field of automatic sound recognition are provided. The depicted problem is that of classifying an input audio stream in one out of all classes. A typical system to solve such a problem consists of Mel-Frequency (Cepstral) Coefficients as feature extraction followed by a (Deep) Neural Network based architecture as classifier [17–19]. Both consist of several modular building-blocks which will be described in the two following subsections.

4.1 Feature extraction: Mel-Frequency Cepstral Coefficients

The Mel-Frequency Cepstral Coefficients (MFCC) feature extraction algorithm originates from the domain of automatic speech recognition and is based on the perception of sound by the human auditory system [26]. Despite the fact that MFCC was developed for that task, it is shown that it is also usable for automatic sound recognition due to their ability to represent the amplitude spectrum in a compact form [17–19]. Figure 4 is an overview of the MFCC feature extraction process and involves the following main components: (1) framing and windowing, (2) Discrete Fourier Transform (DFT), (3) Mel-frequency filterbank, (4) logarithmic operation and (5) Discrete Cosine Transform (DCT). In recent years, related to the popularity of deep learning, researchers tend to use an intermediate output of the MFCC algorithm (Mel-frequency filterbank) or even the raw audio waveform due to the neural network being able to learn a feature representation from the provided data. Up to date, the building blocks of MFCC are still one of the dominant signal processing algorithms used for speech and audio classification tasks. In the following subsections these building blocks are explained.

4.1.1 Framing and windowing

The framing and windowing operation of the feature extraction process transfers the raw acoustic waveform into short overlapping segments. These segments, further called frames, are typically 30 ms long with an overlap of 10 ms. Each frame $\mathbf{f}$ is then typically windowed with an Hamming-window $\mathbf{h}$ to reduce spectral leakage. The windowing operation is defined as $\mathbf{s}_n = \mathbf{h}_n\mathbf{f}_n$ for $n = [0, 1, \ldots, N_t - 1]$ with N_t the amount of samples in one frame. The amount of operations for this stage consists of N_t multiplies for one frame. As it can be computed in-place, the total needed storage is $\cdot N_t \cdot S$ bits for the output and $\cdot N_t \cdot S$ bits for the parameters, where S represents the word size. As a multiply needs 3 memory accesses, this results in a total of $6 \cdot N_t$ memory accesses.

4.1.2 Discrete Fourier transform

The framing and windowing operation are followed by the DFT that transforms the frame $\mathbf{s}$ in the time-domain into a frame $\mathbf{z}$ in the frequency domain. Typically, the Fast Fourier Transform (FFT) can be used which is a computational efficient variant for computing the DFT [27]. Here, a frame is zero-padded to the next radix-2 number. In case of a radix-2 FFT implementation the amount of operations are

$N_f/2 \cdot log_2(N_f)$ complex multiplies and $N_f \cdot log_2(N_f)$ complex additions with N_f the length of the (zero-padded) input frame. A complex multiplication is assumed to consist of 4 multiplications and 2 additions. By assuming an in-place algorithm, the total needed storage is $N_f \cdot S$ bits along with $15 \cdot N_f \cdot log_2(N_f)$ memory accesses.

4.1.3 log Mel-frequency transform

As a first step only the power spectrum $|\mathbf{z}|^2$ up to $N_f/2$ samples is retained since studies have shown that the amplitude of the spectrum is of more importance than the phase information. Then, the Mel-frequency filterbank smooths the high-dimensional magnitude spectrum such that it reflects the sensitivity of the human auditory system to frequency shifts where the lower frequencies are perceptually of more importance than the higher ones. The filterbank is defined by overlapping triangular frequency response bandpass filters with a constant spacing and bandwidth in the Mel-frequency domain.

Typically, the number of bands N_m is set in the range between 20 and 60 [17,18]. The log Mel features can be computed by using:

$$\mathbf{m}_b = \log\Big(\sum_{k=0}^{N_f/2-1} \mathbf{W}_{bk}|\mathbf{z}_k|^2\Big) \ , \tag{12}$$

with $b = [0, 1, \ldots, N_m - 1]$ and $\mathbf{W}_{bk}$ the weight of the Mel filter bank in band b at frequency k. In a final step a logarithm operation is performed on the Mel features, which is also motivated by human perception of sound as humans hear loudness on a logarithmic scale. The amount of operations summarises to $N_f/2 \cdot N_m$ multiply-accumulates and N_m logarithms. In total $(N_f/2 \cdot N_m + N_m) \cdot S$ bits need to be stored along with $2 \cdot N_f \cdot N_m + 2 \cdot N_m$ memory accesses.

4.1.4 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) expresses the Mel features in terms of a sum of cosine functions. These cosine functions describe the amplitude envelope of the Mel features. A limited set of coefficients (typically 14) are retained as they contain the elementary aspects of the shape. The DCT on the log Mel features is defined as:

$$\mathbf{c}_d = \sum_{b=0}^{N_m-1} \mathbf{m}_b \cos\Big[d\Big(b + \frac{1}{2}\Big)\frac{\pi}{N_m}\Big] \ , \tag{13}$$

with $d = [0, 1, \ldots, D-1]$ and $[\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{N_c-1}]$ as the MFCC feature vector with length N_c. The amount of operations consist of $N_m \cdot N_c$ multiply-accumulates. In total $(N_m + 1) \cdot N_c \cdot S$ bits need to be stored along with $4 \cdot N_m \cdot N_c$ memory accesses.

4.2 Classifier: Artificial Neural Network

Artificial Neural Networks, inspired by biological neural networks, automatically learn tasks based on provided data (and desired output) [25]. A (Deep) Neural Network architecture is highly customizable and constructed using multiple layers. Following subsections roughly introduce several common layers used in (Deep) Neural Network architectures along with the amount of operations.

4.2.1 Fully-Connected layer

The main building-block of the Fully Connected (FC) layer is an artificial neuron which is modelled using a modified version of a perceptron. A perceptron is a linear classifier able to discriminate two classes [28]. The formal definition of the modified perceptron is $f(x) = \sigma(\mathbf{w}^T\mathbf{x})$, with $\mathbf{x}$ the input vector, $\mathbf{w}$ the learned weight vector and σ an activation function to (non-linearly) transform the output value. The input vector $\mathbf{x}$ is augmented with a scalar of value 1 to allow for a shift in the discriminating hyperplane. Different from

the original perceptron is that the activation function is not restricted to a threshold. A perceptron can be extended to multi-class classification by stacking multiple perceptrons (one for each class) and is referred as a multi-class perceptron. When used in Neural Networks, this is denoted as a FC layer. To allow for non-linear classification multiple FC layers can be concatenated where each output of the previous layer is connected to all inputs of that particular layer. In between those layers, a non-linear activation function should be used to create the non-linearity. As output layer the activation function typically consists of a Softmax operation to provide a probabilistic output. These activations are defined in section 4.2.2. The amount of operations for the FC layer are $L_n(L_i+1)$ multiply-accumulates, with L_n the amount of neurons and L_i the size of the input vector. In total $L_n \cdot (L_i+2) \cdot S$ bits need to be stored along with $4 \cdot L_n \cdot (L_i+1)$ memory accesses.

4.2.2 Activation function

Activation functions introduce non-linear properties to the Neural Network. It does a non-linear mapping on the output of a artificial neuron to either provide input to a following layer or a probabilistic output on the final layer. Numerous activations have been proposed over past decades. Currently, the most commonly used are the Rectified Linear Unit (ReLU), Logistic and Tanh [19]. ReLU is described as $\sigma(\mathbf{z}_k) = \max(0, \mathbf{z}_k)$ with $\mathbf{z}$ the output of the previous layer and $k = [0, 1, \ldots, L_n - 1]$. As it does not provide a probabilistic output, it is only used in-between layers. Logistic and Tanh are defined as $\sigma(\mathbf{z}_k) = 1/(1 + e^{-\mathbf{z}_k})$ and $\sigma(\mathbf{z}_k) = 2/(1 + e^{-2\mathbf{z}_k}) - 1$ respectively. As output layer, a Softmax function is typically used to compute the probabilities for each class. The Softmax is defined as $\sigma(\mathbf{z}_k) = e^{\mathbf{z}_k} / \sum_{k=0}^{K-1} e^{\mathbf{z}_k}$. For ReLU, the total amount of operations devoted to the activation function is L_n comparisons along with $3 \cdot L_n$ memory accesses. The Tanh activation function consists of $2 \cdot L_n$ additions, and L_n divisions and exponentials which results in $9 \cdot L_n$ memory accesses. In case of Softmax and Logistic the total amount of operations summarises to L_n additions, divisions and exponentials which results in $9 \cdot L_n$ memory accesses.

4.2.3 Convolutional layer

In case of more-than-one-dimensional input data, connecting all inputs to a FC layer might lead to an unreasonable amount of weights. A convolutional layer is similar to a FC layer as they are also made up of artificial neurons that have learnable weights. A convolution layer however, convolves the input data with multiple so-called templates. It is assumed that a particular template, smaller than the input data, can be reused at multiple positions in the input data. At each convolution index these templates are locally-connected to the input data and output one activation. Due to the weight sharing the amount of weights is reduced compared to directly using a FC layer.

The hyperparameters that define a convolutional layer are the number of templates T_n, template dimensions $T_{d,k}$, convolution strides $T_{s,k}$ and amount of zero padding $T_{p,k}$ on the input data for a particular dimension index k. The output size of a convolution layer for a particular template T_n is defined as $L_{o,k} = (L_{i,k} - T_{d,k} + 2T_{p,k})/T_{s,k} + 1$, with $L_{i,k}$ and $L_{o,k}$ the length of the input- and output data respectively at dimension k. This leads to total amount of operations of $T_n \cdot \prod_{k=0}^{L_d}((L_{i,k} - T_{d,k} + 2T_{p,k})/T_{s,k} + 1) \cdot (\prod_{k=0}^{L_d} T_{d,k} + 1)$ multiply-accumulates. The amount of memory accesses are the amount of operations multiplied by 4. The total needed storage consists of $T_n \cdot (\prod_{k=0}^{L_d} T_{d,k} + 1) \cdot S$ bits for the weights and $(\prod_{k=0}^{L_d} L_{o,k}) \cdot S$ bits as output.

4.2.4 Pooling layer

It is a common practice to introduce a pooling layer in-between succesive convolutional layers. Such a layer performs undersampling on the previous output to reduce overfitting and the amount of parameters and computation in the network. Similar to a convolutional layer, the pooling layer iterates over the entire input data. Different from a convolutional layer is that, instead of a matrix product of the locally-connected data and a template, it calculates a summary of the current locally-connected part of the

data. This summary could either be average or a max. operation. The pooling layer therefor consists of $\prod_{k=0}^{L_d}((L_{i,k} - T_{d,k} + 2T_{p,k})/T_{s,k} + 1) \cdot (\prod_{k=0}^{L_d} T_{d,k} - 1)$ operations. The amount of memory accesses are the amount of operations multiplied by 3. The needed output storage summarises to $(\prod_{k=0}^{L_d} L_{o,k}) \cdot S$ bits.

4.2.5 Batch Normalization

Batch Normalization (BN) was introduced to compensate for the so-called *internal covariate shift* that slows down the training of the network [29]. BN performs a standard normalization, during training on each mini-batch, on the output of the activations of each layer seperately. During the test phase this adds an additional shift and scale to each activation output. For a particular layer the amount of operations are L_n additions and multiplications, which results in $6 \cdot L_n$ memory accesses. The total stored information is $2 \cdot L_n \cdot S$ bits for the shift and scale.

5 Energy consumption of communications

This section focuses in describing and modeling the energy consumption of the communication module of the sensor node. We assume that the node is equipped with N_t transmitter and N_r receiveing antennas and corresponding transceiver branches, respectively [30]. If a node has only one antenna then $N_t = N_r = 1$ is used.

By default, the node is assumed to be in a low power consumption (sleep) mode [10]. At its designated time the node wakes up, and engages a transmission and reception of frames with the central connection point respectively. In case of transmission, if x attempts are decoded with errors, the transmitter will declare an outage and go to sleep mode for one coherence time of the channel. Let us denote as τ_{out} the number of outages and τ_x the number of transmission trials required to achieve a decoding without errors ($\tau_x \in \{1, \ldots, x\}$). These are random variables with mean values given by $\bar{\tau}_{\text{out}}$ and $\bar{\tau}_x$, whose values depends on the modulation, coding scheme and fading statistics [31].

The energy consumption per correctly transmitted information bit, $\bar{\mathcal{E}}_{\text{T}}$, can be modeled as [32]

$$\bar{\mathcal{E}}_{\text{T}} = (1 + \bar{\tau}_{\text{out}}) \frac{\mathcal{E}_{\text{st}}}{N_T} + \mathcal{E}_{\text{enc}} + (\mathcal{E}_{\text{etx,b}} + \mathcal{E}_{\text{PA,b}} + \mathcal{E}_{\text{erx,fb}}) (x\bar{\tau}_{\text{out}} + \bar{\tau}_x) \ . \tag{14}$$

Above, $\mathcal{E}_{\text{st}}$ is the startup energy required to awake the node from the low power mode, $\mathcal{E}_{\text{enc}}$ is the energy required to encode the forward message, $\mathcal{E}_{\text{etx,b}}$ and $\mathcal{E}_{\text{erx,fb}}$ are the energy consumption of the baseband and radio-frequency electronic components that perform the forward transmission and the reception of the feedback frame respectively and $\mathcal{E}_{\text{PA,b}}$ is the energy consumption of the power amplifier (which is responsible of the electromagnetic irradiation) for sending an information bit.

By analogy, the total energy used per correctly received bit, which involves demodulating forward frames and transmitting the feedback frames, can be modeled as [32]

$$\bar{\mathcal{E}}_{\text{R}} = (1 + \bar{\tau}_{\text{out}}) \frac{\mathcal{E}_{\text{st}}}{N_R} + (\mathcal{E}_{\text{dec}} + \mathcal{E}_{\text{erx,b}} + \mathcal{E}_{\text{etx,fb}} + \mathcal{E}_{\text{PA,fb}}) (x\bar{\tau}_{\text{out}} + \bar{\tau}_x) \ . \tag{15}$$

Above, $\mathcal{E}_{\text{erx,b}}$ and $\mathcal{E}_{\text{etx,fb}}$ are the energy consumption of the baseband and radio-frequency electronic components that perform the forward reception and the transmission of the feedback frame respectively and $\mathcal{E}_{\text{PA,fb}}$ is the energy consumption of the power amplifier for transmitting feedback frames.

5.1 Modeling the energy consumption of the PA

Let us express $\mathcal{E}_{\text{PA}}$, the total consumption due to the irradiated power, as

$$\mathcal{E}_{\text{PA}} = \mathcal{E}_{\text{PA,b}} + \mathcal{E}_{\text{PA,fb}} = \sum_{j=1}^{N_{\text{t}}} P_{\text{PA}}^{(j)} T_{\text{b}} + \sum_{j=1}^{N_{\text{t}}} P_{\text{PA}}^{(j)} T_{\text{fb}} \ , \tag{16}$$

where $P_{\mathrm{PA}}^{(j)}$ is the power consumption of the PA of the j-th transceiver branch. The total time per information bit in the forward direction, T_{b}, is calculated as [30]

$$T_{\mathrm{b}} = \frac{1}{rR_{\mathrm{s}}}\left(\frac{1}{\omega b} + \frac{H}{\omega L} + \frac{N_{\mathrm{t}}O_{\mathrm{a}} + O_{\mathrm{b}}}{L}\right) \ , \tag{17}$$

where R_{s} is the physical layer symbol-rate, r is the code rate of the coding scheme (percentage of data per payload bits), ω is the multiplexing gain of the MIMO modulation, $b = \log_2(M)$ is the number of bits per complex symbol, H and L are the number of bits in the header and payload of the frame, O_{a} is the acquisition overhead per transceiver branch and O_{b} is the remaining overhead, which is approximately independent of the antenna array size (both O_{a} and O_{b} are measured in bits [33]). Similarly, the total time per feedback frame, T_{fb}, is given by

$$T_{\mathrm{fb}} = \frac{F}{r\omega R_{\mathrm{s}} L} \ , \tag{18}$$

where F is the number of bits of the feedback frame.

Let us relate the power consumption of the PAs with the signal-to-noise ratio (SNR). The j-th transmit antenna radiates $P_{\mathrm{tx}}^{(j)}$ watts, which are provided by the corresponding power amplifier (PA). The PA's power consumption is modeled by [33]

$$P_{\mathrm{PA}}^{(j)} = \frac{1}{\eta} P_{\mathrm{tx}}^{(j)} \ , \tag{19}$$

where η the average efficiency of the PA. In general, the average PA efficiency can be more precisely modeled using the distribution of the output power of the underlying signal. If we limit the analysis to linear PAs, such as Class A and Class B PAs (as many mobile and wireless communication devices require linear PAs), then we can approximate η with

$$\eta = \left(\frac{\bar{P}_{\mathrm{tx}}}{P_{\max}}\right)^{\beta} \eta_{\max} \ , \tag{20}$$

where $\bar{P}_{\mathrm{tx}}$ is the average radiated power (which we assume is the same for all transmitter antennas), $P_{\max}$ is the maximal PA output and

$$\eta_{\max,\ \text{class A}} = 0.5 \quad \text{and} \quad \beta_{\text{class A}} = 1 \tag{21}$$

$$\eta_{\max,\ \text{class B}} = 0.785 \quad \text{and} \quad \beta_{\text{class B}} = 0.5 \ . \tag{22}$$

In these equations, $P_{\text{back-off}} = P_{\max}/\bar{P}_{\mathrm{tx}}$ is the back-off of the PA. Highest efficiency is achieved by constant-envelope signals for which $P_{\text{back-off}} = 1$. In general, one can calculate the back-off coefficient as $P_{\text{back-off}} = \xi/S$, where ξ is the peak-to-average power ratio of the modulation (which is usually calculated as $\xi = 3(\sqrt{M}-1)/(\sqrt{m}+1)$) and S accounts any additional back-off that may be taken when the wireless link has excess link budget and transmit power can be decreased further. Finally, the relationship between the PA consumption P_{PA} and the radiated power P_{tx} is calculated as:

$$P_{\mathrm{tx}}^{(j)} = \left(\frac{S}{\xi}\right)^{\beta} \eta_{\max} P_{\mathrm{PA}}^{(j)} \ . \tag{23}$$

The transmission power attenuates over the air with path loss and arrives at the receiver with a mean power given by

$$P_{\mathrm{rx}}^{(j)} = \frac{P_{\mathrm{tx}}^{(j)}}{A_0 d^{\alpha}} \ , \tag{24}$$

where d is the distance between transmitter and receiver and α is the path loss exponent. Above, A_0 is a parameter that is defined by the free-space friss equation

$$A_0 = \frac{1}{G_{\mathrm{t}} G_{\mathrm{r}}}\left(\frac{4\pi}{\lambda}\right)^2 \ , \tag{25}$$

where G_{t} and G_{r} are the transmitter and receiver antenna gains and λ is the carrier wavelength. Finally, if σ_{s}^2 is the average received power per symbol at the input point of the decision stage of the receiver (which is located after the MIMO decoder), the total received signal power is given by

$$\sum_{j=1}^{N_{\mathrm{t}}} \bar{P}_{\mathrm{rx}}^{(j)} = \omega \sigma_{\mathrm{s}}^2 = \omega \sigma_{\mathrm{n}}^2 \bar{\gamma} \ , \tag{26}$$

where σ_{n}^2 is the thermal noise power and $\bar{\gamma}$ is the average SNR. In general, $\sigma_{\mathrm{n}}^2 = N_0 W N_{\mathrm{f}} M_{\mathrm{L}}$, where N_0 is the power spectral density of the baseband-equivalent additive white Gaussian noise (AWGN), W is the transmission bandwidth, N_{f} is the noise figure of the receiver's front end and M_{L} is a link margin term which represents any other additive noise or interference [34].

With all this, one finds that

$$\sum_{j=1}^{N_{\mathrm{t}}} P_{\mathrm{PA}}^{(j)} = \left(\frac{\xi}{S}\right)^{\beta} \frac{1}{\eta_{\max}} \sum_{j=1}^{N_{\mathrm{t}}} P_{\mathrm{tx}}^{(j)} \tag{27}$$

$$= \left(\frac{\xi}{S}\right)^{\beta} \frac{A_0 d^{\alpha}}{\eta_{\max}} \sum_{j=1}^{N_{\mathrm{t}}} P_{\mathrm{rx}}^{(j)} \tag{28}$$

$$= \left(\frac{\xi}{S}\right)^{\beta} \frac{N_0 W N_{\mathrm{f}} M_{\mathrm{L}} A_0}{\eta_{\max}} \omega d^{\alpha} \bar{\gamma} \tag{29}$$

$$= A \omega d^{\alpha} \bar{\gamma} \tag{30}$$

with A a constant.

5.2 Modeling the energy consumption of the other electronic components

Let us assume that the device is equipped with N_{t} antennas, using an architecture as shown in Figure 5. Then, one can express the electronic consumption of the transmitter per information bit as [30]

$$\mathcal{E}_{\mathrm{etx,b}} = N_{\mathrm{t}} P_{\mathrm{etx}} T_{\mathrm{b}} = N_{\mathrm{t}} (P_{\mathrm{DAC}} + 2P_{\mathrm{filter}} + P_{\mathrm{LO}} + P_{\mathrm{mixer}}) T_{\mathrm{b}} \ , \tag{31}$$

where P_{etx} is the power consumption of the electronic components (filters, mixer, DAC and local oscillator) that perform the transmission per transceiver branch. A similar equation for $\mathcal{E}_{\mathrm{etx,fb}}$ can be obtained by replacing T_{b} with T_{fb} which are defined in (17) and (18) respectively.

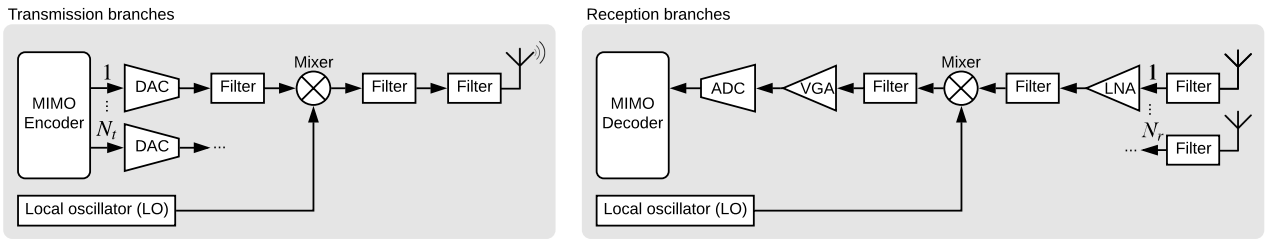


Figure 5: MIMO architecture considered in this work

Analogously, one can show that

$$\mathcal{E}_{\mathrm{erx,b}} = N_{\mathrm{r}} P_{\mathrm{erx}} T_{\mathrm{b}} = N_{\mathrm{r}} (3P_{\mathrm{filter}} + P_{\mathrm{LNA}} + P_{\mathrm{LO}} + P_{\mathrm{mixer}} + P_{\mathrm{VGA}} + P_{\mathrm{ADC}}) T_{\mathrm{b}} \ , \tag{32}$$

where P_{erx} is the power consumption of the electronic components (filters, mixer, ADC, VGA and local oscillator) that perform the forward and feedback frame reception per branch. A similar equation can be obtained for $\mathcal{E}_{\text{erx,fb}}$.

Following [33], we will model the energy consumption of DACs as $P_{\text{DAC}} = \beta(P_{\text{DAC}}^{\text{static}} + P_{\text{DAC}}^{\text{dyn}})$, where $P_{\text{DAC}}^{\text{static}}$ (resp. $P_{\text{DAC}}^{\text{static}}$) is the static (resp. dynamic) power consumption and β is a correcting factor to incorporate some second order effects. If a binary-weighted current-steering DAC is considered [35], then

$$P_{\text{DAC}}^{\text{static}} = V_{\text{dd}} I_{\text{unit}} \mathbb{E}\left\{\sum_{i=0}^{n_1} 2^i b_i\right\} = \frac{1}{2} V_{\text{dd}} I_{\text{unit}} (2^{n_1} - 1) \ , \tag{33}$$

where n_1 is the resolution, b_i are independent Bernoulli random variables with parameter $1/2$, V_{dd} is the power supply voltage and I_{unit} is the unit current source corresponding to the least significant bit. The dynamic consumption can be approximated as $P_{\text{DAC}}^{\text{dyn}} = (1/2) n_1 C_{\text{p}} f_{\text{s}}^{\text{DAC}} V_{\text{dd}}^2$, where C_{p} is the parasitic capacitance of each switch, the $1/2$ is the switching probability and $f_{\text{s}}^{\text{DAC}}$ is the sampling frequency. Hence, the total consumption of the DAC is expressed as

$$P_{\text{DAC}} = \frac{\beta}{2}\left[V_{\text{dd}} I_{\text{unit}} (2^{n_1} - 1) + n_1 C_{\text{p}} f_{\text{s}}^{\text{DAC}} V_{\text{dd}}^2\right] \ . \tag{34}$$

In turn, the ADC consumption can be computed using (10).

5.3 Modeling the energy consumption of encoding and decoding forward frames

The computations requred for encoding and decoding data frames can be demanding, depending on the choice of coding scheme [32]. Therefore, it makes sense to include the energy costs of these operations into the energy budget. However, for simplicity we neglect the coding and decoding costs of headers and feedback frames, which usually are either uncoded or use lightweigth codes whose processing can safely be neglected.

By considering that the encoding has to be done for each frame, its cost is shared among the L_u payload bits. Similarly to (11), the energy consumption for encoding one frame — normalized per payload bit — is given by

$$\mathcal{E}_{\text{enc}} = \frac{1}{N_T} \mathcal{E}_{\text{cc}} \sum_{j=1}^{J_{\text{ALU}}} c_j n_j^{\text{enc}} \ , \tag{35}$$

with n_j^{enc} the amount of times the j-th arithmetic operation is performed. Note that it is straightforward to write an equation for the decoding cost equivalent to (35). More information on the computational complexity of the used error correction code (BCH code) can be retrieved in [32].

Finally, following [32] our modeling does not includes the cost of memory storage and access, which is left for future work.

5.4 Re-transmission Statistics

For computing the statistic of retransmissions due to decoding errors, let us derive an expression for $\bar{\tau}_{\text{out}}$ and $\bar{\tau}_x$ following [32]. As each outage declaration are independent events, τ_{out} will be a geometric random variable with p.d.f. $\mathbb{P}\{\tau_{\text{out}} = j\} = (1 - q_x) q_x^j$, where $q_x = 1 - \mathbb{P}\{\tau \leq x\}$ is the outage probability. Then, a direct calculation shows that its mean value is given by

$$\bar{\tau}_{\text{out}} = \frac{q_x}{1 - q_x} \ . \tag{36}$$

The p.d.f. of τ_x is given by

$$\mathbb{P}\{\tau_x = t\} = \mathbb{P}\{\tau = t | \tau \leq x\} = \frac{\mathbb{P}\{\tau = t\}}{1 - q_x} \tag{37}$$

and, hence, it mean value is calculated as

$$\bar{\tau}_x = \sum_{t=1}^{x} t \cdot \mathbb{P}\{\tau_x = t\} = \frac{1}{1-q_x} \sum_{t=1}^{x} t \cdot \mathbb{P}\{\tau = t\} \ . \tag{38}$$

Finally, one can find that

$$x\bar{\tau}_{\text{out}} + \bar{\tau}_x = \frac{1}{1-q_x} \left(x q_x + \sum_{t=1}^{x} t\mathbb{P}\{\tau = t\} \right) \ . \tag{39}$$

The values of $\bar{\tau}_{\text{out}}$ and $\bar{\tau}_x$ depend strongly on the correlations of the wireless channels. Let us consider two extreme examples: fast fading and static fading channels. In fast fading channels the frame error rate of each transmission trial are i.i.d. random variables, while in static channels there are the same random variable. Then, one have that $q_x^{\text{ff}} = \bar{P}_{\text{f}}^x$ and $q_x^{\text{bf}} = \mathbb{E}\left\{P_{\text{f}}^x\right\}$, and a direct application of the Jensen inequality shows that $q_{\text{x}}^{\text{ff}} \leq q_{\text{x}}^{\text{bf}}$. Also, by defining $\Phi(x)$ as

$$\Phi(x) = \frac{1}{1 - \mathbb{E}\left\{P_{\text{f}}^x\right\}} \mathbb{E}\left\{ \frac{1 - P_{\text{f}}^x}{1 - P_{\text{f}}} \right\} \ , \tag{40}$$

it can be shown that $x\bar{\tau}_x^{\text{bf}} + \bar{\tau}_{\text{out}}^{\text{bf}} = \Phi(x)$ and $x\bar{\tau}_x^{\text{ff}} + \bar{\tau}_{\text{out}}^{\text{ff}} = \Phi(1)$. It can also be shown that $\Phi(x)$ is a increasing function, so that in fast fading scenarios one usually requires more transmissions trails.

Particular functional forms can be given for P_{f} depending on the error correcting code scheme in use. For the sake of concreteness, in the sequel we present a derivation valid for BCH codes, following the derivation shown in [36]. Let us denote as n the legth of each codeword, and assuming that $n < L$ let's define $n_{\text{c}} = L/n$ ($n_{\text{c}} \in \mathbb{N}$) as the number of codewords per payload. In order to decode a frame correctly, one needs to correctly obtain H correct header symbols and n_{c} codewords with at least $(n-t) = \lambda$ correct symbols, where t is the maximum number of bits that the FEC block code is able to correct in each codeword. Therefore, by taking into consideration the various possible permutations, $\bar{P}_{\text{f}}$ can be expressed in terms of the bit error rate of the M-ary modulation $P_{\text{b}}(\gamma)$ and the binary modulation symbol error rate $P_{\text{bin}}(\gamma)$ as

$$\bar{P}_{\text{f}}(\bar{\gamma}) = 1 - \left[1 - \bar{P}_{\text{bin}}(\bar{\gamma})\right]^H \left[\sum_{j=0}^{t} \binom{n}{j} \left[1 - \bar{P}_{\text{b}}(\bar{\gamma})\right]^{n-j} \bar{P}_{\text{b}}(\bar{\gamma})^j \right]^{n_c} \ . \tag{41}$$

Above, γ corresponds to the signal-to-noise ratio and $\bar{\gamma} = \mathbb{E}\left\{\gamma\right\}$. Also, please note that we are using the shorthand notation $\bar{P}_{\text{bin}}(\bar{\gamma}) = \mathbb{E}\{P_{\text{bin}}(\gamma)\}$ and $\bar{P}_{\text{b}}(\bar{\gamma}) = \mathbb{E}\{P_{\text{b}}(\gamma)\}$, and that (41) is only valid for scenarios that experience fast-fading conditions.

Finally, let us remark that simple methods to approximate the error rates of MIMO channels are available in [37].

5.5 Final model

Using the material presented in the above subsections, one an finally express (14) as

$$\bar{\mathcal{E}}_{\text{T}} = \frac{1}{N_T} \left(\frac{\mathcal{E}_{\text{st}}}{1-q_x} + \mathcal{E}_{\text{cc}} \sum_{j=1}^{J_{\text{ALU}}} c_j n_j^{\text{enc}} \right) + \left[\left(N_{\text{t}} P_{\text{etx}} + A d^{\alpha} \omega \bar{\gamma} \right) T_{\text{b}} + N_{\text{t}} P_{\text{erx}} T_{\text{fb}} \right] \Phi(x) \ . \tag{42}$$

This equation allows to express the average energy consumption per successfully transferred bit in terms of a number of design parameters. Similarly, the average energy consumption per succesfully received bit, given by (43), is expanded to

$$\bar{\mathcal{E}}_{\mathrm{R}} = \frac{\mathcal{E}_{\mathrm{st}}}{(1-q_x)N_R} + \left(\frac{1}{N_R}\mathcal{E}_{\mathrm{cc}} \sum_{j=1}^{J_{\mathrm{ALU}}} c_j n_j^{\mathrm{dec}} + N_{\mathrm{r}} P_{\mathrm{erx}} T_{\mathrm{b}} + (N_{\mathrm{r}} P_{\mathrm{etx}} + A d^{\alpha} \omega \bar{\gamma}) T_{\mathrm{fb}} \right) \Phi(x) \ . \tag{43}$$

An overview of the relevant hardware related parameters along with the used values is given in Appendix A Table 4.

6 Model analysis

The proposed model contains separate blocks for each of the four considered layers (sensing, processing and communication), and each of them are populated by a considerable amount of parameters. However, some of these parameters are more interesting to be explored than others. Although we don't develop an exhaustive exploration of all these parameters, the remaining of this section provides some guidelines on aspects of interest that can be explored.

Table 1 provides a summary of some parameters from the sensing and communication layers that have a strong influence on the energy efficiency of the sensor node.* In case of the sensing layer, $f_{s,mic}$ and n_{mic} are interesting parameters as they have a big impact on the consumed energy because it regulate the amount of information to be processed and communicated. Naturally, more collected information equals a higher energy consumption.

In case of the communication layer, one parameter to explore is the amount of communicated bits N_T that depends on both the processing and sensing layer. An interesting trade-off is the energy spend in processing versus the one spent in communication. Another parameter of interest is the number of bits t the FEC can correct. A higher number leads to more communication frame overhead but less re-transmission due to errors, as shown in Refs. [32, 36]. Another parameter of interest is the transmission bandwidth W, which is determined usually be the communication standard that the network uses (Zigbee, Bluetooth, etc). In general a higher bandwidth is beneficial as transmission times – and hence the baseline electronic consumption of the transceiver – are reduced. Please note that the effect of interference is not included in this model, being left for future work. Finally, one can explore the impact of various the communication channel via the path-loss coefficient α and the SNR e. In busy scenarios with many obstables the path-loss is higher and the received signal strength is smaller, which usually causes more re-transmission and, in turn, a lower energy efficiency.

Parameter	Description	Value
$f_{s,mic}$	Sampling frequency of the microphone	16 kHz
n_{mic}	ADC resolution	12 bit
N_T	Amount of bits to communicate	-
t	Number of bits the FEC can correct	4
W	Transmission bandwidth	1 MHz
α	Path-loss coefficient	3.2
e	SNR of communication	25 dB

Table 1: Experimental parameters along with the default value in the source code [22]

*The processing layer is not covered in the table; one can compare various feature extraction and classifier architectures, or no processing at all.

A Default parameters used in the MATLAB implementation

Parameter	Description	Value
T	Room temperature	290 K
$P_{\text{mic,act}}$	Active microphone power consumption	10 mW
$V_{\text{dd}}^{\text{LNA,mic}}$	Voltage supply of the LNA-mic	1.5 V
NEF	Noise efficiency factor	6
FOM_{ADC}	Figure of merit of the ADC	500 fJ/conv. [38]
$f_{s,mic}$	Sampling frequency of the microphone	16 kHz
n_{mic}	ADC resolution	12 bit

Table 2: Overview of the default parameters used in the MATLAB implementation [22] for the sensing layer

Parameter	Description	Value
$\mathcal{E}_{\text{op}}$	Energy per operation (GP proc.)	500 pJ [24,39]
	Energy per operation (GP DSP)	100 pJ [24]
$\mathcal{E}_{\text{ma}}$	Energy per memory access (on-chip SRAM)	100 fJ/bit [40]
	Energy per memory access (off-chip SRAM)	100 pJ/bit [41]
	Energy per memory access (off-chip DRAM)	100 pJ/bit [24]
$\mathcal{E}_{\text{ms}}$	Energy memory leakage (on-chip SRAM)	50 pW/bit [40]
	Energy memory leakage (off-chip SRAM)	10 pW/bit [41]
	Energy memory leakage (off-chip DRAM)	75 pW/bit [24]
τ_m	Memory access time (SRAM)	10 ns
	Memory access time (DRAM)	100 ns
S	Word size	32 bit
c_{mac}	Multiply-accumulate cost	12 ops. [39]
c_{add}	Addition cost	7 op. [39]
c_{mul}	Multiplication cost	7 op. [39]
c_{div}	Division cost	20 op. [39]
c_{cmp}	Comparator cost	15 op. [39]
c_{exp}	Natural exponential cost	3700 op. [39]
c_{log}	Logarithmic cost	4000 ops. [39]

Table 3: Overview of the default parameters used in the MATLAB implementation [22] for the processing layer

Parameter	Description	Value
$\mathcal{E}_{\text{st}}$	Start-up energy	94 μJ [42]
P_{filter}	Filter power consumption	1 mW [43]
P_{mixer}	Mixer power consumption	1 mW [43]
$P_{\text{LNA,rx}}$	LNA power consumption	3 mW [43]
P_{VGA}	VGA power consumption	5 mW [43]
P_{LO}	Local oscilator consumption	22.5 mW [43]
n_1	Resolution of the Tx DAC	10 levels [33]
$f_{\text{s}}^{\text{DAC}}$	DAC sampling frequency	4 MHz
$V_{\text{dd}}^{\text{DAC}}$	Voltage supply of the DAC	3 V [33]
I_{unit}	DAC unit current source	10 μA [33]
C_{p}	DAC parasitic capacitance	1 pF [33]
n_2	Resolution of Rx ADC	10 levels [33]
$f_{\text{s}}^{\text{ADC}}$	DAC sampling frequency	4 MHz
η_{max}	PA efficiency (Class B)	0.785 %
β	Exponent for Class B PA	0.5
S	Additional back-off coefficient	0 dB
G_{t}	Transmitter antenna gain	1.8
G_{r}	Receiver antenna gain	1.8
f_{c}	Carrier frequency	2.4 GHz [44]
W	Bandwidth	1 MHz [43]
R_{s}	Symbol rate	0.125 MBaud [43]
M	M-ary number	2 (BPSK)
N_{f}	Receiver noise figure	16 dB [43]
M_{l}	Link margin	20 dB
α	Path-loss coefficient	3.2
t	Number of bits the FEC can correct	4
H	Frame Header	2 bytes [44]
L	Payload	127 bytes [44]
O_{a}	Acquisition overhead	4 bytes [44]
O_{b}	Estimation and synchronization overhead	1 bytes [44]
F	Feedback frame length	5 bytes

Table 4: Overview of the default parameters used in the MATLAB implementation [22] for the communication layer

References

[1] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, no. 5, pp. 67–77, May 2011.

[2] P. Rawat, K. D. Singh, and J. M. Chaouchi, Hakimaand Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, Apr 2014.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* Acm, 2002, pp. 88–97.

[4] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 266–282, First 2014.

[5] F. Erden, S. Velipasalar, A. Z. Alkar, and A. E. Cetin, "Sensors in assisted living: A survey of signal and image processing methods," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 36–44, March 2016.

[6] I. A. T. Hashem, V. Chang, N. B. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, and H. Chiroma, "The role of big data in smart city," *International Journal of Information Management*, vol. 36, no. 5, pp. 748 – 758, 2016.

[7] M. Vacher, F. Portet, A. Fleury, and N. Noury, "Development of audio sensing technology for ambient assisted living: Applications and challenges," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 2, no. 1, pp. 35–54, January 2011.

[8] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *2011 18th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov 2011, pp. 1–6.

[9] S. Lauwereins, "Cross-layer self-adaptivity for ultra-low power responsive IoT devices," 2018.

[10] H. Karl and A. Willig, *Protocols and architectures for wireless sensor networks.* John Wiley & Sons, 2007.

[11] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[12] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking.* ACM, 2001, pp. 272–287.

[13] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[14] F. Rosas, R. D. Souza, M. Verhelst, and S. Pollin, "Energy-efficient mimo multihop communications using the antenna selection scheme," in *Wireless Communication Systems (ISWCS), 2015 International Symposium on.* IEEE, 2015, pp. 686–690.

[15] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1567–1576.

[16] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 171–180.

[17] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct 2015.

[18] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[19] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 Challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Munich, Germany, November 2017.

[20] T. Yang, Y. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 1916–1920.

[21] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 367–379.

[22] G. Dekkers and F. Rosas, "WASN EM: a multi-layered Energy Model for Wireless Acoustic Sensor Networks," 2018. [Online]. Available: https://github.com/gertdekkers/WASN_EM/

[23] M. Steyaert and W. Sansen, "A micropower low-noise monolithic instrumentation amplifier for medical purposes," *Solid-State Circuits, IEEE Journal of*, vol. 22, no. 6, pp. 1163–1168, Dec 1987.

[24] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2014, pp. 10–14.

[25] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec 2017.

[26] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pp. 357–366, 1980.

[27] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[28] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, pp. 65–386, 1958.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, pp. 448–456.

[30] F. Rosas and C. Oberli, "Impact of the channel state information on the energy-efficiency of mimo communications," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4156–4169, 2015.

[31] ——, "Modulation and snr optimization for achieving energy-efficient communications over short-range fading channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4286–4295, 2012.

[32] F. Rosas, R. D. Souza, M. E. Pellenz, C. Oberli, G. Brante, M. Verhelst, and S. Pollin, "Optimizing the code rate of energy-constrained wireless communications with harq," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 191–205, 2016.

[33] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, Sept. 2005.

[34] ——, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1089–1098, Aug. 2004.

[35] J. J. W. M. Gustavsson and N. N. Tan, *CMOS Data Converters for Communications.* Boston, MA: Kluwer, 2000.

[36] F. Rosas, G. Brante, R. D. Souza, and C. Oberli, "Optimizing the code rate for achieving energy-efficient wireless communications," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE.* IEEE, 2014, pp. 775–780.

[37] F. Rosas and C. Oberli, "Nakagami-m approximations for multiple-input multiple-output singular value decomposition transmissions," *IET Communications*, vol. 7, no. 6, pp. 554–561, 2013.

[38] B. Murmann. Adc performance survey 1997-2014. [Online]. Available: {http://www.stanford.edu/~murmann/adcsurvey.html}

[39] *Cortex-M4: Technical Reference Manual*, ARM Limited, March 2010.

[40] T. Haine, Q. Nguyen, F. Stas, L. Moreau, D. Flandre, and D. Bol, "An 80-mhz 0.4v ulv sram macro in 28nm fdsoi achieving 28-fj/bit access energy with a ulp bitcell and on-chip adaptive back bias generation," in *ESSCIRC 2017 - 43rd IEEE European Solid State Circuits Conference*, Sept 2017, pp. 312–315.

[41] *CY62126EV30 MoBL: 1-Mbit (64K x 16) Static RAM*, Cypress Semiconductor Corporation, 2017, rev. *P.

[42] M. Siekkinen, M. Hiienkari, J. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, 2012, pp. 232–237.

[43] A. Balankutty, S.-A. Yu, Y. Feng, and P. Kinget, "A 0.6-V Zero-IF/Low-IF Receiver With Integrated Fractional-N Synthesizer for 2.4-GHz ISM-Band Applications," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 3, pp. 538–553, March 2010.

[44] *Specifications for Local and Metropolitan Area Networks- Specific Requirements Part 15.4*, IEEE Std. 802.15.4, 2006.