Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-018-0888-y

ORIGINAL RESEARCH

Adaptive power switching technique for ultrasonic motion sensors

Abbass Hammoud[1] · Michel Deriaz[1] · Dimitri Konstantas[1]

Received: 5 December 2017 / Accepted: 28 May 2018

Abstract

Smart sensing technologies play a key role in the core of smart systems, which form the rapidly evolving internet of things. In this context, buildings' occupancy information is an important input that allows smart systems to be seamlessly aware of and responsive to the inhabitants, thus ensuring their comfort. Ultrasonic motion sensors are used to obtain occupancy information of indoor spaces. Although they provide a high accuracy as compared to other sensors, like Passive InfraRed, they require a higher power consumption. In this work, we propose an adaptive power switching technique, which we call *power hopping*. This technique allows ultrasound motion sensors to optimize their transmitter power level, in order to best fit their surrounding environment. The objective is to reduce the overall energy consumption of these sensors. We have tested our method using a sensor prototype, and the results show that, depending on the sensor's environment, a possible saving in the transmitter power can be achieved, which reached up to 78% in our experiments. We also derive an upper bound limit of the method's convergence time, and we propose an automatic sensing method to detect potential changes in the sensor's environment.

Keywords Ultrasound · Motion sensors · Power switching · Environment sensing

1 Introduction

In the recent years, the interest in smart buildings is continuously increasing. Such buildings rely on a wide range of sensors that feed the smart systems with useful context information. In this regard, occupancy sensors represent an important input, allowing the systems to be seamlessly aware of and responsive to the occupants' needs, thus promoting their comfort, health and well-being. Indoor occupancy information is implied in a wide range of applications (Hammoud et al. 2016), from lighting control, Heating, Ventilation, Air Conditioning (HVAC), to assisted living and security systems. While several technologies have been developed for occupancy sensing, Passive InfraRed (PIR) and ultrasonic motion sensors remain the most prevalent in this respect (Teixeira et al. 2010).

PIR sensors are widely used to detect human presence, by responding to a change in the temperature pattern across the field of view of the sensor. A PIR sensor is considered passive as it does not emit any energy itself, but rather relies on the pattern of the received infrared radiation in its environment (Guo et al. 2010). Different works have suggested algorithms to enhance the performance of PIR sensors and the processing of their output (Yin et al. 2016; Narayana et al. 2015; Luo et al. 2016; Kuutti et al. 2014). PIR sensors are attractive because of their low power consumption. However, the main drawbacks of PIR sensors are their limited accuracy, and sensitivity to changes in the environment (sunlight, heating effect, etc.), as well as their limitation to work only in line-of-sight (LOS) conditions.

Ultrasonic sensors, which are based on non-audible acoustic signals, are another category of occupancy sensors. On one side, they can be used as ranging sensors to detect objects in the field of view, based on the time-of-flight (ToF) of the ultrasonic signal. Some

This work was co-funded by the State Secretariat for Education, Research and Innovation of the Swiss federal government and the European Union, in the frame of the EU AAL project EDLAH2 (aal-2015-022) and the AAL project ManyMe (aal-2016-063).

✉ Abbass Hammoud
abbass.hammoud@unige.ch

Michel Deriaz
michel.deriaz@unige.ch

Dimitri Konstantas
dimitri.konstantas@unige.ch

[1] Information Science Institute, GSEM/CUI, University of Geneva, Geneva, Switzerland

works (Mokhtari et al. 2017; Jaramillo and Linnartz 2015; Gonzalez et al. 2014) use this technique to infer the occupancy at a specific location. Ultrasound can also be used to detect still occupants.

On the other side, the ultrasonic motion sensors, which we address in our work, use active ultrasonic signals to sense human motions inside an area, based on the Doppler effect principle. These sensors are helpful to obtain fine information about the room occupancy, the direction of movements and speed of occupants. Example applications of these sensors are presented in Caicedo and Pandharipande (2012); Raj et al. (2012) and Mehmood et al. (2010).

Ultrasonic motion sensors are promising as they are more sensitive and accurate than PIR ones (Teixeira et al. 2010). Moreover, they are capable of sensing moving objects in non line-of-sight (NLOS), since the ultrasonic signals can propagate around objects, unlike infrared radiations (Hammoud et al. 2017a). Despite these advantages, ultrasound motion sensors are still not very popular, as it is the case with PIR ones. The fact that they are active, as compared to passive, makes their power consumption higher than PIR, and thus limits their potential applications.

Power consumption of sensors is a critical issue in the design of smart systems and sensor networks (Shnayder et al. 2004; Akyildiz et al. 2002; Pottie and Kaiser 2000). Therefore, it is important to reduce the individual energy consumption of sensors, especially when the energy source is limited (case of battery-based for example) (Lindsey and Raghavendra 2002). While many works have discussed the use of ultrasonic sensors in occupancy sensing, the issue of power consumption has not attracted sufficient interest. In their work, Mishra et al. (2009) try to reduce the *processing* power of ultrasound raging sensors used by robots to perceive the occupancy grid. They do so on the logic circuitry level, and show that the power consumption can be reduced by redesigning the processing logic circuit. However, and to the best of our knowledge, there is no research work in the literature that focuses on reducing the *transmitter* power consumption of ultrasonic sensors, as we suggest in our work.

In this paper, which extends our previous work (Hammoud et al. 2017b), we state that the required power for ultrasonic motion sensors is not fixed, but it rather varies as a function of the sensor's environment. We introduce the *power hopping* method as an automatic process to optimize the transmitter power level to best fit this environment. The method aims to reduce the power consumption of the sensor while preserving the performance. After testing the method using a sensor prototype, we validated that a possible saving in the transmitter power can be achieved depending on the sensor's environment, which reached up to 78% of power reduction in our experiments. Additionally, we derive an upper bound limit of the method's convergence time, and we present an automatic sensing method to detect potential changes in the environment.

The rest of this paper is organized as follows. First, Sect. 2 introduces some necessary details about the operation of ultrasound motion sensors. Sect. 3 explains the concept and algorithm of the suggested power hopping method. In Sect. 4 we derive an upper limit for the convergence time, and in Sect. 5 we present our technique to automatically detect changes in the sensor's environment. The experimental evaluation and results are shown in Sects. 6, and 7 presents the possible limitations of the proposed method. Finally, Sect. 8 concludes the paper.

2 Preliminaries

Ultrasound-based motion sensors use active ultrasonic signals to detect movements of people inside a certain area, based on the Doppler effect principle. They periodically transmit an ultrasonic signal and observe the corresponding reflected one. Frequency shifts in the received signal indicates the detection of movements, whereas the signal frequency remains intact when no movements occur.

Assuming that the transmitted signal is a sine pulse of frequency f_c and duration T, its sampled version can be represented by a discrete time sequence $x[n]$ of length $\lfloor T/Ts \rfloor$, where T_s is the sampling frequency. Let a *frame* represent one transmission/reception of the ultrasonic signal. The transmitted signal propagates through the environment, and reflects on obstacles and objects. Static objects (walls, furniture, etc) do not alter the signal frequency, while moving ones (people walking, etc) will cause some shifts in the signal frequency. The sensing unit records the received signal $y[n]$ for a certain frame, and computes the magnitude $\{|Y[k]|\}$ of its Discrete Fourier Transform (DFT), which represents its frequency spectrum:

$$|Y[k]| = |\mathcal{F}\{y[n]\}|_k = \left| \sum_{n=0}^{N-1} y[n] e^{-j\frac{2\pi}{N}nk} \right| \quad \text{for } k = 0, 1, \dots, N-1 \tag{1}$$

To detect movements, the frequency spectrum of the current frame is compared against a reference still frame. The still frequency spectrum Y_{still} corresponds to the case with no moving objects, and should be known to the system. The difference between the two spectra reflects the frequency shifts in the signal, and is computed by the system using the following quantity, which we call the *motion score*:

$$motion\ score = \sum_{k \in I} ||Y[k]| - |Y_{still}[k]|| \tag{2}$$

where I is the ultrasound frequency band to consider around the signal frequency f_c:

$$I = [f_c - \Delta_{f(max)}, f_c) \cup (f_c, f_c + \Delta_{f(max)}] \tag{3}$$

$\Delta_{f(max)}$ being the maximum Doppler shift, which is determined by the assumed maximum velocity of a person's motion v_{max}:

$$\Delta_{f(max)} = \frac{2v_{max}}{c} f_c \quad (4)$$

where c is the speed of sound in air. The result of Eq. 2 is compared with a *threshold* value. If it exceeds the threshold, it can be deduced that a movement is detected, otherwise if it is smaller than the threshold, the frequency differences can be considered due to noise and thus no movements are reported to be detected. We define the motion intensity as the ratio of the motion score to the threshold value:

$$\textit{motion intensity} = \frac{\textit{motion score}}{\textit{threshold}} \quad (5)$$

In a previous work, we showed how the motion detection parameters (Y_{still}, *threshold*) can be obtained automatically through self-calibration (Hammoud et al. 2017c).

3 Power hopping method

The total power consumption of an ultrasound motion sensor is mainly divided into signal transmission/reception and signal processing:

$$P_{total} = P_{transmitter} + P_{receiver} + P_{processing} \quad (6)$$

While the power required for the receiver and signal processing is independent from the sensor environment, the transmission power can be optimized to best fit a certain environment and cut unnecessary power consumption, thus reducing the overall power consumption of the sensing unit. The objective of the power hopping technique is to find the optimal level of transmitter power that the sensor can use, without jeopardizing the performance.

3.1 The best power for each setting

The required transmitter power varies from one environment to another, depending on variables like the room dimensions, presence of obstacles, and also hardware characteristics (receiver's sensitivity, etc.). Figure 1 shows how the installation environment would affect the required transmitter power. For example, if the LOS of the system is not blocked, the ultrasonic signal propagates easily and may need a low transmitter power. Whereas if the LOS is obstructed, as when the system is placed behind an obstacle or furniture, a stronger signal is needed to propagate around such obstacles.

Let P_{max} and P_{min} be respectively the maximum and minimum power levels of the system. P_{max} allows the system to work in all conditions. However, according to the sensor's environment, the system may still achieve the same

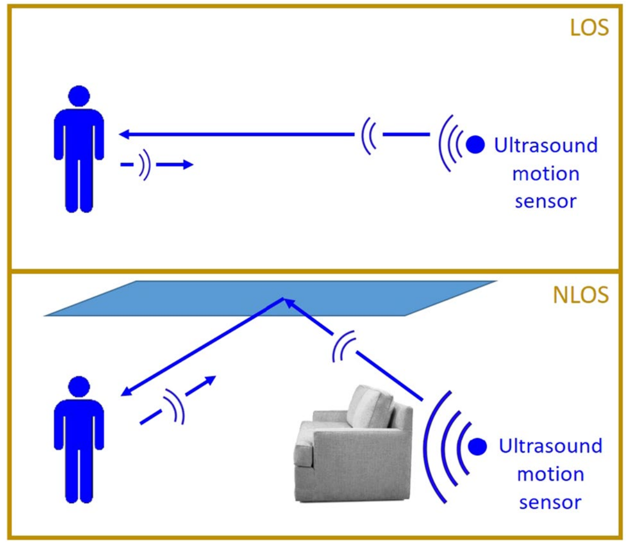


Fig. 1 A different transmitter power is required in each case

performance with a lower power level. Power hopping allows the system to adapt to the optimal transmitter power $P_{optimal}$, which is the lowest possible transmitter power that yields the same performance as P_{max}. The value of $P_{optimal}$ should lie between P_{max} and P_{min}:

$$P_{min} \leq P_{optimal} \leq P_{max} \quad (7)$$

The power hopping method is supposed to take place during the initialization phase, when the sensor is installed in a new environment. Once the optimal power $P_{optimal}$ is found, the system switches to this new transmitter power level. Subsequently, the system may run the power hopping process occasionally to reflect any possible changes in the environment.

3.2 Relation between transmitter power and frequency spectrum

Before introducing the algorithm, it is necessary to state the relation between the transmitter power and the frequency spectrum of the received signal.

Statement. *Let the vector* Y_1 *be the DFT of the received signal* $y_1[n]$ *that corresponds to a transmitted signal* $x_1[n]$, *and assuming that:*

- *The sensor's environment has a linear response*
- *The transmitter and receiver do not operate in their saturation region*
- *The effect of the noise on the received signal is negligible*

Then, if the amplitude of the transmitted signal is scaled by a constant α *such that* $x_2[n] = \alpha x_1[n]$, *the magnitude* $|Y_2|$ *of the DFT of the corresponding received signal* $y_2[n]$ *is such that*

$$|Y_2[k]| = \alpha |Y_1[k]| \qquad \forall k \in I$$

Proof Because the sensor's environment can be modeled by a linear system, when the transmitted signal is scaled by some constant α, the received signal will be also scaled by the same factor. The Fourier transform is also linear, so the scaling will also scale its result by the same factor, and therefore the magnitude of the Fourier transform, which represents the frequency spectrum of the received signal, will be scaled by α. $\square$

Corollary. *Assume that we have the still frequency spectrum $|Y_{still}|$ that corresponds to a certain transmitter power P. Since the power of a transmitted signal $x[n]$ of length N is*

$$P = \frac{1}{N}\sum_{n=0}^{N-1} |x[n]|^2, \tag{8}$$

if the amplitude of the transmitted signal $x[n]$ is scaled by α, then its power will be scaled by $\beta = \alpha^2$, and thus the new corresponding still frequency spectrum $|Y_{still(new)}|$ will be equal to $\{\alpha \times |Y_{still}|\}$ (or $\{\sqrt{\beta} \times |Y_{still}|\}$).

Following a similar reasoning, the new threshold value to be used for comparison needs also to be scaled by the same constant α.

3.3 Power hopping algorithm

Initially, the transmitter power that is used by the system is P_{max}. The parameters of the system ($|Y_{still}|$, *threshold*) that are initially used correspond to P_{max}. The system then tries to switch to a lower transmitter power $P_{candidate}$.

P_{valid} is the transmitter power level for which the sensor works well, and is initialized to P_{max}, while $P_{invalid}$ is the transmitter power level which is too weak to detect motions and is initialized to P_{min}.

When a motion is detected inside the room, the system hops between P_{valid} and $P_{candidate}$ back and forth several times. For $P_{candidate}$ to become valid, it should detect the motions that P_{valid} can detect with the same intensity every time, otherwise it is considered invalid. The number of times the system hops between the two power levels is a design choice parameter, which we call it n_{hops}. Setting the value of this parameter is a trade-off: on one hand, the higher n_{hops} is, the more robust the switching between the two power levels is, but also the convergence time of the power hopping method is longer. On the other hand if n_{hops} is low, the method converges faster, but the switching is less robust. In our design, we choose n_{hops} to be 3, which we empirically found to be a good middle choice to keep the switching robust while keeping the convergence time short enough.

When the system hops between P_{valid} and $P_{candidate}$, it calculates every time the motion score:

$$\begin{aligned} motion\ score_{(valid)} &= \sum_{k\in I} \big||Y_{valid}[k]| - |Y_{still}[k]|\big| \\ motion\ score_{(candidate)} &= \sum_{k\in I} \big||Y_{candidate}[k]| \\ &\quad - \sqrt{P_{candidate}/P_{valid}} \times |Y_{still}\ [k]|\big| \end{aligned} \tag{9}$$

Note that in Eq. 9, the new still frequency spectrum is calculated using the reasoning of Corollary 3.2 (hence the square root in the equation).

$P_{candidate}$ is then considered valid, if the following holds for each time:

$$\begin{cases} motion\ score_{(valid)} \quad > threshold \\ \quad and \\ motion\ score_{(candidate)} \quad > \sqrt{P_{candidate}/P_{valid}} \times threshold \\ \quad and \\ \frac{motion\ score_{(valid)}}{threshold} \quad \approx \frac{motion\ score_{(candidate)}}{\sqrt{P_{candidate}P_{valid}} \times threshold} \end{cases} \tag{10}$$

The new threshold value is calculated as discussed in Corollary 3.2 as well. The first condition in Eq. 10 indicates that a motion is being detected with P_{valid}, the second condition means that the motion can be also detected with $P_{candidate}$. The last condition requires that $P_{candidate}$ detects the motion with the same intensity compared to P_{valid}, ensuring that the switch of power levels is robust. The approximate equality, instead of full equality, is used to account for possible noise in the signals.

When $P_{candidate}$ is found to be valid, the system switches to this new power level and updates the parameters ($|Y_{still}|$, *threshold*), otherwise it picks another candidate power level, as the middle value between P_{valid} and $P_{invalid}$, similar to a binary search. We assume that during the short time that this iteration takes, it is valid to consider that a person's movement is continuous.

The system continues the power hopping method, until the valid power P_{valid} does not change more than a certain amount ε. At this time, the optimal power $P_{optimal}$ is assumed to be found, and the system switches to this new transmitter power level. Algorithm 1 presents the power hopping method in pseudo-code.

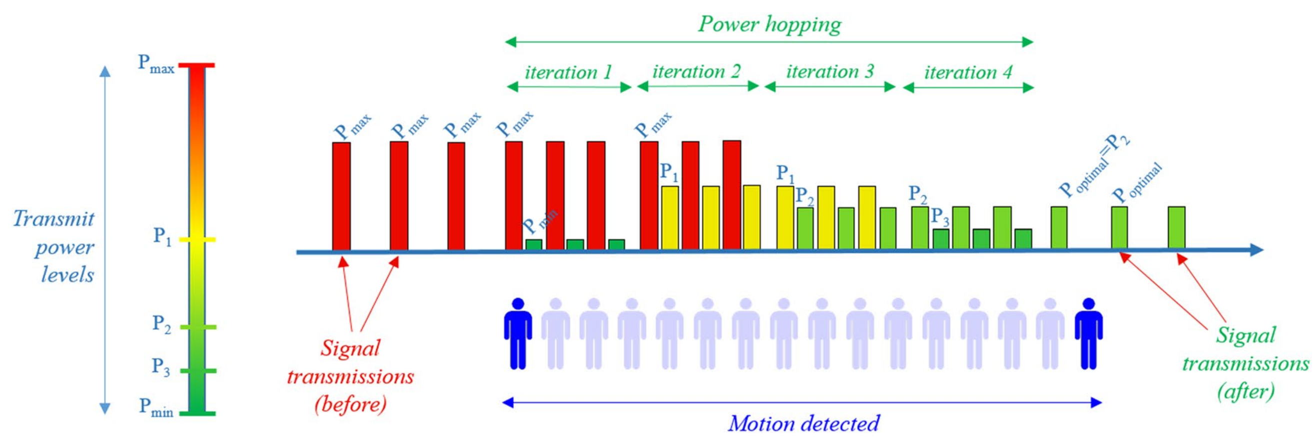


Fig. 2 Power hopping example: adapting to the optimal transmitter power level

Algorithm 1 Power hopping algorithm

procedure POWER HOPPING
initialization:
  $P_{valid} \leftarrow \boldsymbol{P_{max}}$
  $P_{invalid} \leftarrow \boldsymbol{P_{min}}$
  $P_{candidate} \leftarrow \boldsymbol{P_{min}}$
iteration:
  while $(P_{valid} - P_{candidate} > \varepsilon)$ and (*motion detected*) **do**
    hop between P_{valid} and $P_{candidate}$
    if $P_{candidate}$ *is valid* **then**
      $Y_{still}[k] \leftarrow \sqrt{P_{candidate}/P_{valid}} \times Y_{still}[k]\ \forall k \in I$
      $threshold \leftarrow \sqrt{P_{candidate}/P_{valid}} \times threshold$
      $P_{valid} \leftarrow P_{candidate}$
    else
      $P_{invalid} \leftarrow P_{candidate}$
    end if
    $P_{candidate} \leftarrow (P_{valid}+P_{invalid})/2$
  end while
result:
  $P_{optimal} \leftarrow P_{valid}$
end procedure

3.4 Power hopping example

In this section, we illustrate an example case showing how the power hopping method works. We consider the scenario shown in Fig. 2. First, the system is using the maximum power P_{max} to detect persons' movements. Once it detects a motion, it triggers the power hopping process. For the sake of this example, we assume that the power hopping converges in 4 iterations in this particular scenario:

- Iteration 1: P_{max} is a valid power level. P_{min} is the candidate power level. The system hops between P_{max} and P_{min}. P_{min} fails to detect motions, so the candidate power is updated to $P_1 = \left(P_{max} + P_{min}\right)/2$.
- Iteration 2: power hopping between P_{max} and P_1. P_1 succeeds to detect motions , so it becomes the valid power level. The new candidate power is now $P_2 = \left(P_1 + P_{min}\right)/2$.
- Iteration 3: power hopping between P_1 and P_2. P_2 succeeds to detect motions, so the updates take place similar to Iteration 2.
- Iteration 4: power hopping between P_2 and P_3. P_3 fails to detect motions. In this particular example, we assume that the new candidate power is such that $P_2 - P_{candidate} < \varepsilon$, and the power hopping process terminates at this point.

Power hopping converges to P_2, which is considered the optimal power level. The system switches to this power level, and from this moment on uses it to detect motions.

4 Convergence time

The time required for the power hopping method to converge, depends on several parameters. In this section, we derive an upper limit of this time.

Let $n_{iterations}$ be the number of iterations needed for the system to converge. The method runs as long as the following condition holds:

$$\frac{P_{max} - P_{min}}{2^{n_{iterations}-1}} > \varepsilon \tag{11}$$

Solving for $n_{iterations}$ yields:

$$n_{iterations} < 1 + \log[2]\frac{P_{max} - P_{min}}{\varepsilon} \tag{12}$$

which means that the maximum number of iterations for the method to converge is:

$$n_{iterations} = 1 + \lfloor \log[2]\frac{P_{max} - P_{min}}{\varepsilon} \rfloor \tag{13}$$

The total time of the process depends on the time required for each iteration. Assuming that the processing time after each transmission is negligible, the time it takes for each iteration $t_{iteration}$ depends on the number of hops n_{hops} from P_{valid} to $P_{candidate}$ and the time of each transmission $t_{transmission}$:

$$t_{iteration} = 2 \times n_{hops} \times t_{transmission} \tag{14}$$

The maximum required time for the power hopping process is:

$$t_{max} = t_{iteration} \times n_{iterations} \tag{15}$$

Yielding finally:

$$t_{max} = 2 \times n_{hops} \times t_{transmission} \times \left(1 + \left\lfloor \log[2] \frac{P_{max} - P_{min}}{\epsilon} \right\rfloor\right) \tag{16}$$

With $n_{hops} = 3$, for a transmitted signal duration of 10 ms, and a desired resolution of $\epsilon = (P_{max} - P_{min})/128$, the maximum convergence time would be $t_{max} = 0.48$ s.

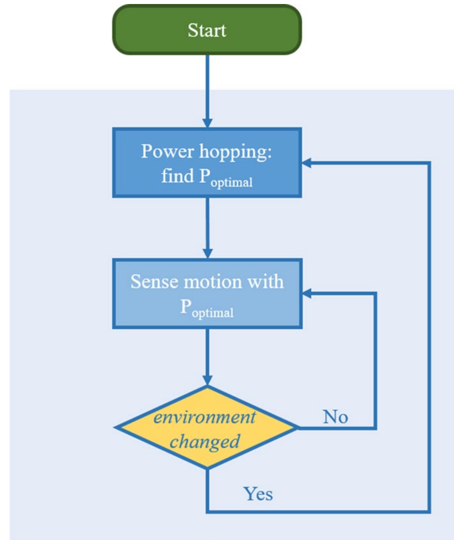


Fig. 3 Flow chart showing how the power hopping is triggered when the environment is changed

5 Automatic detection of environment changes

5.1 Objective

The proposed power hopping method finds the optimal transmit power for a given environment, as previously described. However, as indoor environments are likely to be changed with time (motion sensor smoved to a new place, furniture moved around, obstacle abundance changed, etc.), this optimal transmit power might become invalid, and needs to be recalculated for every new setting. Therefore, we have designed an automatic technique whose objective is to sense whether the surrounding environment has changed, and to re-trigger the power hopping process. As shown in the flow chart of Fig. 3, the motion sensor checks periodically for changes in the environment, and runs the power hopping process when some changes are detected.

5.2 Technique

Our developed technique consists of transmitting a specific ultrasonic signal and observing the corresponding reflected one. The sensor then processes the received signal to get the *reflection pattern* of the environment. We call a *reflection pattern*, the result of the environment's response to the emitted ultrasonic signal, which depends on several parameters, like the environment's dimensions, boundaries, the position of obstacles, furniture, etc. Therefore, any changes in this environment will be observed in the reflection pattern.

When the sensor switches to a new optimal transmit power level $P_{optimal}$, it records the corresponding reflection pattern of the environment. Then, it checks periodically if the reflection pattern has changed. Once the acquired reflection pattern of the environment does not match the recorded one, this indicates that the environment has changed and the power hopping technique is re-triggered, to compute the new optimal power level.

5.3 Obtaining the reflection pattern

The emitted ultrasound is a short-time signal in the non-audible frequency range. We have investigated different signal types, and our tests showed that a chirp signal is more immune to interference, as compared to a sinusoidal signal. Therefore, the emitted ultrasonic signal $x[n]$ is a chirp with 20 and 21 kHz as lower and upper frequency limits respectively. As the signal frequency range falls in the supported frequency range, the same hardware previously used can be leveraged for emitting the chirp signal and receiving the reflected one.

We denote the environment's impulse response by $h[n]$, which defines the multipath propagation of the emitted signal's reflections, caused by the obstacles and environment's boundaries. It can be written as:

$$h[n] = \sum_{m=0}^{M-1} a_m e^{j\phi_m} \delta(n - \tau_m) \tag{17}$$

where a_m, ϕ_m, and τ_m represent the signal attenuation, phase difference, and time delay of the mth multipath signal respectively.

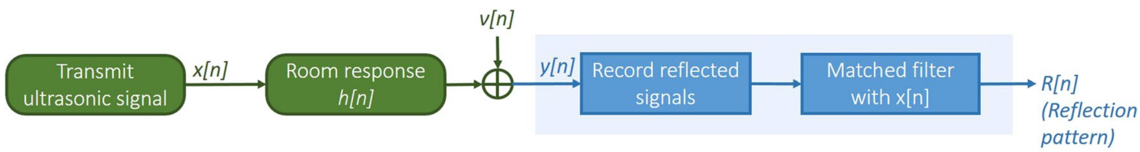


Fig. 4 Obtaining the reflection pattern

When the ultrasonic signal $x[n]$ is emitted in the environment, the received signal $y[n]$ is the convolution of the transmitted signal $x[n]$ with the discrete-time version of the room impulse response $h[n]$, plus an additive noise $v[n]$ assumed to be white Gaussian:

$$y[n] = x[n] * h[n] + v[n] \tag{18}$$

The assumed noise is used to model the random noise caused by uncontrolled sources (ambient noise in the environment, noise introduced by the receiver, etc.). The reflection pattern $R[n]$ is obtained by applying a matched filter to the received signal $y[n]$, as depicted in Fig. 4.

5.4 Comparing reflection patterns

Instead of calculating a numerical expression of the environment's response for each frame, we statistically process the reflection patterns in order to spot the variations in the environment response. We call $R_{ref}[n]$ the reference reflection pattern, which corresponds to the current optimal power level $P_{optimal}$. A new acquired reflection pattern $R[n]$ is compared to the reference $R_{ref}[n]$ in order to infer whether the indoor environment has changed. The comparison of two reflection patterns is achieved by cross-correlation, to determine the similarity between them. We denote by *similarity index*, the maximum value of the cross-correlation result in absolute value:

$$similarity\ index = max|cross\text{-}correlation(\boldsymbol{R_{ref}}, \boldsymbol{R})| \tag{19}$$

Figure 5 shows how two reflection patterns are compared. The similarity index is a value ranging between 0 and 1. A high index (close to 1) shows high similarity of the compared reflection patterns meaning that the environment did not change. On the other hand, a low similarity index indicates the compared reflection patterns are uncorrelated and therefore the environment response has changed. A threshold value is used to differentiate the similarity indices, which we empirically set to (0.9).

When the ultrasonic signal $x[n]$ is emitted in the indoor environment, it propagates in a semispherical pattern. The direct line-of-sight copy of the signal is the first one to be picked up by the receiver as it travels the shortest distance. Subsequent multipath copies of the signal scattered by different objects, obstacles, and enivronment's boundaries are received at later time instants. After a certain time duration T_{total}, the propagated signal vanishes (becomes too weak to be picked up by the receiver). In Fig. 6, we show an indicative example of the signal propagation in the case where the environment changes. This example aims to show only the concept of the difference in the environment's response, rather than the actual exact propagation obeying physics laws.

5.5 Algorithm

Algorithm 2 describes in pseudo-code the technique of detecting the variations in the sensor's environment.

Algorithm 2 Detecting environment changes

```
while (sensor is ON) do
    power hopping: find P_optimal
    compute R_ref[n]                          ▷ reference reflection pattern
    environment_changed ← false
    while (! environment_changed) do
        compute R[n]                          ▷ current reflection pattern
        calculate similarity_index (R_ref, R)
        if (similarity_index < 0.9) then
            environment_changed = true
        end if
    end while
end while
```

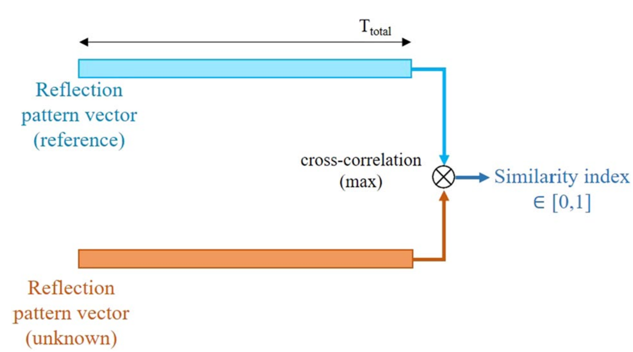
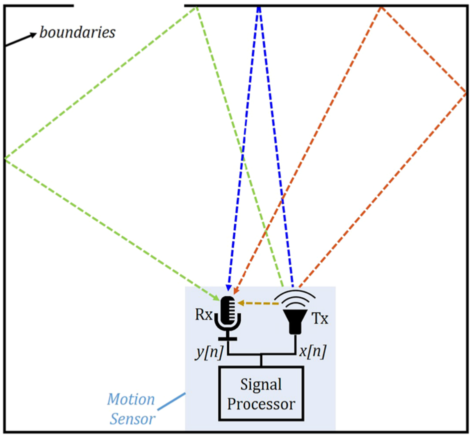
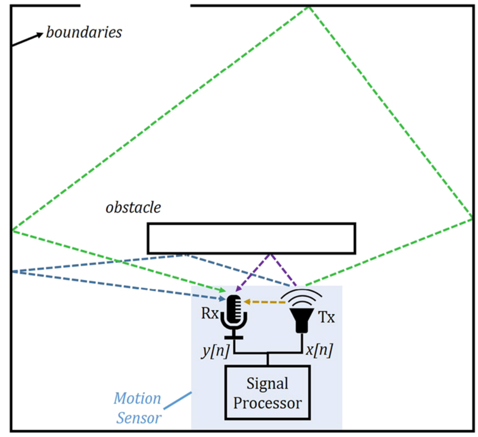


Fig. 5 Comparing two reflection patterns



(a) Motion sensor in LOS



(b) Motion sensor behind obstacle

Fig. 6 Indicative example showing the difference of the environment response, when there is a change in its layout

6 Experimental evaluation

6.1 Prototype

In order to test the performance of the suggested power hopping method, we implemented a prototype of an ultrasound motion sensor. The prototype, shown in Fig. 7, is composed of a transmitter (a commodity speaker) and a receiver (a commodity microphone) both connected to a Raspberry Pi board[1], which acts as the control/processing unit, and implements the described power hopping method. The transmitted signal frequency used is 21 kHz, and the frequency band considered in the processing of the received signal is 20–22 kHz.

6.2 Testing methodology

The prototype of the sensor is fixed inside a certain area. For a chosen maximum transmitter power P_{max}, the detection parameters ($|Y_{still}|$, *threshold*) are calibrated as described in Hammoud et al. (2017c).

The sensor waits to detect movements before triggering the power hopping process. A person walks to the designated area, moves for few seconds and then leaves the area. During this time, the sensor runs the power hopping method, and switches to the optimal power level.

Aiming to cover different environments, the previous testing process is repeated for 4 different cases, as follows:

- Case 1: The area is a large room (Room A) with dimensions $6 \times 7.8m$, the sensing unit is placed such that the LOS is not obstructed.
- Case 2: Same area of case 1, but the sensing unit is placed behind an obstacle blocking the LOS.
- Case 3: The area is a small room (Room B) with dimensions $6 \times 3.9m$, LOS not obstructed.
- Case 4: Same area of case 3, with the sensing unit placed behind an obstacle blocking the LOS.

Figure 8 illustrates the different test cases for which the power hopping method was tested. In each case, we note the obtained optimal power level $P_{optimal}$. Once the power hopping process is over, and as a double check, we verified that the new transmitter power is capable of detecting the motions in the room as the previous power P_{max}.

6.3 Power hopping results

The results summarized in Table 1 show the optimal level of transmitter power $P_{optimal}$ obtained through the power hopping process, and also the power saving in each case.

The power hopping method aims to find the optimal transmitter power, and to cut unnecessary transmitter power used. In general, we see that an important power saving in the transmitter power can be achieved (up to 78% as in case

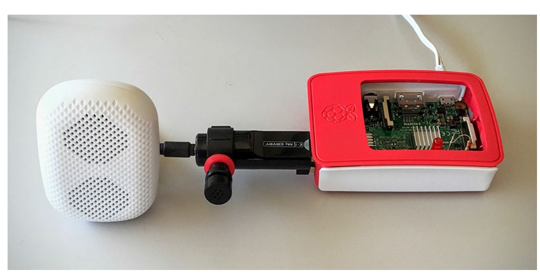


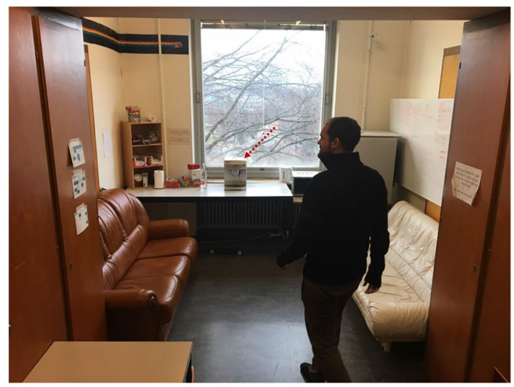
Fig. 7 Prototype of the ultrasound motion sensor used for testing

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

(a) Case 1 : Room A, prototype in LOS

(b) Case 2 : Room A, prototype behind obstacle

(c) Case 3 : Room B, prototype in LOS



(d) Case 4 : Room B, prototype behind obstacle

Fig. 8 Illustration of the different test cases

3). The obtained results show that the required transmitter power varies in function of the specific environment (room size, obstacles, etc.). A big room, compared to a small one, requires more signal power to cover the whole area and detect movements inside it. This explains why, under the same settings, the system placed in a small room (case 3) requires less power than the case of a big room (case 1), yielding more saving in the transmitter power (78 vs 69%). On the other hand, when the sensor is placed behind an obstacle, a higher transmitter power is needed to propagate around the obstacle and detect movements behind it, as compared to a case with a direct LOS. This is the reason why we notice more power savings in cases 1 (69%) and 3 (78%), compared to cases 2 (4%) and 4 (24%) respectively. Case 2 represents an extreme environment in terms of size and obstacles, this is why a very little saving in the transmitter power is achieved (4%). This shows that the full maximum power level P_{max} might still be needed in such environments.

It is worth noting that although the obtained results show that it is possible to achieve a saving in the transmitter power in some environments, the actual amount of power saving remain specific for the settings of each environment, and the values we obtained are only indicative in this regard.

6.4 Testing environment changes

In order to test our proposed technique for detecting the changes in the sensor's environment, we proceed as follows: For each of the scenarios tested above, when the power hopping process has converged and the $P_{optimal}$ is found, the corresponding reflection pattern is obtained by the sensor when the room is vacant. We compare all the collected reflection patterns corresponding to the different environments:

1. $R_1[n]$: reflection pattern of Room A, sensor in LOS.

Table 1 Power saving as a result of the power hopping method

Case	Original transmit power	New transmit power	Power saving
#1: Room A, LOS	P_{max}	$0.31\,P_{max}$	69%
#2: Room A, NLOS	P_{max}	$0.96\,P_{max}$	4%
#3: Room B, LOS	P_{max}	$0.22\,P_{max}$	78%
#4: Room B, NLOS	P_{max}	$0.76\,P_{max}$	24%

Table 2 Similarity indices of different reflection patterns

Reflection patterns	$R_1[n]$	$R_2[n]$	$R_3[n]$	$R_4[n]$
$R_1[n]$	1	0.41	0.26	0.35
$R_2[n]$		1	0.43	0.44
$R_3[n]$			1	0.62
$R_4[n]$				1

2. $R_2[n]$: reflection pattern of Room A, sensor in NLOS.
3. $R_3[n]$: reflection pattern of Room B, sensor in LOS.
4. $R_4[n]$: reflection pattern of Room B, sensor in NLOS.

The aim of having multiple cases is to check how well can the changes in the environment be detected using the reflection patterns. In the first case, we have a large room where the sensor is placed in the corner with a direct LOS. In the second case, the layout of the room is kept unchanged except that an obstacle is placed in front of the sensor, blocking the LOS. Since in this case, the sensor is supposed to detect that a change occurred and to recompute the optimal transmit power $P_{optimal}$, it is essential to check that this change is detectable by comparing the reflection patterns of the two Cases 1 and 2. The same reasoning is used when comparing Cases 3 and 4, but this time in a small room with the sensor placed in LOS and NLOS respectively.

On the other hand, comparing the reflection pattern of Case 1 or 2 with that of 3 or 4, shows whether placing the sensor in a completely new environment is detectable by our method. This can show that when the sensor is re-installed or moved to in a new place, it is able to detect this change and trigger the power hopping process accordingly, to recompute $P_{optimal}$.

6.4.1 Results

The similarity index between each two reflection patterns is calculated as described in Section 5, to test if the proposed technique is capable of detecting the changes in the sensor's environment. In Table 2, we present the values of the similarity indices between the different reflection patterns. The results show that the change in the environment, whether it is placing in a new environment (1 compared to 3 or 4 for instance), or variations within the same environment (1 compared to 2, or 3 compared to 4), is possible to detect using the value of the similarity index of reflection patterns. All the similarity indices fall below the threshold of 0.9 for reflection patterns of changed environment. This proves that the proposed technique is capable of sensing any changes in the sensor's environment, and triggering the power hopping when it is the case.

7 Limitations of the proposed method

We present in this section the possible limitations that could reduce the efficiency of our proposed method:

- The method detects major changes in the environment, and triggers the power hopping process when it is the case, in order to recalculate the new optimal power level. The new power level is then used for motion detection, until the environment changes again. However, if a given environment is continuously changing, then the optimal power level will be continuously recalculated and updated, thus reducing the general efficiency of the power hopping process in achieving power saving.
- As we saw previously in the obtained results, the cut in power consumption is higher with limited-size environments, and with low obstacle abundance. However, if the ultrasonic sensor is used to cover a very large area, or if the density of obstacles is relatively high, then the power reduction achieved by the power hopping method would not be significant.

8 Conclusion

In this work, we have presented the *power hopping* method, a power optimization technique for ultrasound motion sensors. The method aims to reduce the overall power consumption of these sensors, by cutting unnecessary transmitter power used. The results show that a possible saving in the transmitter power can be achieved, which can be significant or minor depending on the environment. The power hopping method can be very useful especially when the energy source is limited, like when the sensor is battery-powered, so that the battery life is extended. We have also derived an upper bound limit of the method's convergence time. Additionally, we have designed a technique to automatically detect potential changes in the sensor's environment. This technique complements the power hopping process by making sure the obtained optimal power level is valid for the unchanged environment, and automatically triggering the process when changes are detected.

References

Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40(8):102–114. https://doi.org/10.1109/MCOM.2002.1024422 (**ISSN 0163-6804**)

Caicedo D, Pandharipande A (2012) Ultrasonic arrays for localized presence sensing. IEEE Sens J 12(5):849–858. https://doi.org/10.1109/JSEN.2011.2161667 (**ISSN 1530-437X**)

Gonzalez LIL, Großekathöfert U, Amft O (2014) Novel stochastic model for presence detection using ultrasound ranging sensors. In:

2014 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops). pp. 55–60. https://doi.org/10.1109/PerComW.2014.6815164
Guo X, Tiller DK, Henze GP, Waters CE (2010) The performance of occupancy-based lighting control systems: a review. Light Res Technol 42(4):415–431
Hammoud A, Deriaz M, Konstantas D (2016) Robust ultrasound-based room-level localization system using cots components. In: 2016 Fourth international conference on ubiquitous positioning, indoor navigation and location based services (UPINLBS). pp 11–19. https://doi.org/10.1109/UPINLBS.2016.7809975
Hammoud A, Kyritsis AI, Deriaz M, Konstantas D (2017a) Enhanced still presence sensing with supervised learning over segmented ultrasonic reflections. In: 2017 International conference on indoor positioning and indoor navigation (IPIN), pp 1–8. https://doi.org/10.1109/IPIN.2017.8115920
Hammoud A, Anagnostopoulos G, Kyritsis A, Deriaz M, Konstantas D (2017b) Power hopping: an automatic power optimization method for ultrasonic motion sensors. In: The 14th IEEE international conference on ubiquitous intelligence and computing, '17, San Francisco, California
Hammoud A, Deriaz M, Konstantas D (2017c) Ultrasense: A self-calibrating ultrasound-based room occupancy sensing system. *Procedia Computer Science*, 109(Supplement C):75–83, 2017c. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2017.05.297. 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal
Jaramillo P, Linnartz JP (2015) Hidden Markov model for improved ultrasound-based presence detection. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (CIT/IUCC/DASC/PICOM). pp 68–74. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.14
Kuutti J, Saarikko P, Sepponen RE (2014) Real time building zone occupancy detection and activity visualization utilizing a visitor counting sensor network. In: 2014 11th international conference on remote engineering and virtual instrumentation (REV). pp 219–224. https://doi.org/10.1109/REV.2014.6784260
Lindsey S, Raghavendra CS (2002) PEGASIS: power-efficient gathering in sensor information systems. IEEE Aerosp Conf Proc 3:3-1125–3-1130. https://doi.org/10.1109/AERO.2002.103524
Luo X, Liu T, Shen B, Gao L, Luo X (2016) Human indoor localization based on ceiling mounted pir sensor nodes. In: 2016 13th IEEE Annual consumer communications networking conference (CCNC). pp. 868–874. https://doi.org/10.1109/CCNC.2016.7444903
Mehmood A, Sabatier JM, Bradley M, Ekimov A (2010) Extraction of the velocity of walking human's body segments using ultrasonic Doppler. J Acoust Soc Am 128(5):EL316–EL322. https://doi.org/10.1121/1.3501115
Mishra P, Shankar HN, Mathur M (2009) An ultra-low power real time embedded system for map generation using ultrasound sensors. In: 2009 Third UKSim European symposium on computer modeling and simulation. pp. 579–584. https://doi.org/10.1109/EMS.2009.87
Mokhtari G, Zhang Q, Nourbakhsh G, Ball S, Karunanithi M (2017) Bluesound: a new resident identification sensor–using ultrasound array and ble technology for smart home platform. IEEE Sens J 99:1–1. https://doi.org/10.1109/JSEN.2017.2647960 (**ISSN 1530-437X**)
Narayana S, Prasad RV, Rao VS, Prabhakar TV, Kowshik SS, Iyer MS (205) Pir sensors: characterization and novel localization technique. In *Proceedings of the 14th international conference on information processing in sensor networks*, IPSN '15, ACM, New York, NY, USA. pp.142–153, ISBN 978-1-4503-3475-4. https://doi.org/10.1145/2737095.2742561
Pottie GJ, Kaiser WJ (2000) Wireless integrated network sensors. Commun ACM 43(5):51–58. https://doi.org/10.1145/332833.332838 (**ISSN 0001-0782**)
Raj B, Kalgaonkar K, Harrison C, Dietz P (2012) Ultrasonic Doppler sensing in HCI. IEEE Pervasive Comput 11(2):24–29. https://doi.org/10.1109/MPRV.2012.17 (**ISSN 1536-1268**)
Shnayder V, Hempstead M, Chen BR, Allen GW, Welsh M (2004) Simulating the power consumption of large-scale sensor network applications. In Proceedings of the 2nd international conference on embedded networked sensor systems, SenSys '04, ACM, New York, NY, USA. pp 188–200. ISBN 1-58113-879-2. https://doi.org/10.1145/1031495.1031518
Teixeira T, Dublon G, Savvides A (2010) A survey of human-sensing: methods for detecting presence, count, location, track, and identity. ACM Comput Surv 5(1):59–69
Yin J, Fang M, Mokhtari G, Zhang Q (2016) Multi-resident location tracking in smart home through non-wearable unobtrusive sensors. Springer, Cham, pp 3–13. https://doi.org/10.1007/978-3-319-39601-9-1 (**ISBN 978-3-319-39601-9**)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.