Power Hopping: An Automatic Power Optimization Method For Ultrasonic Motion Sensors

Abbass Hammoud, Grigorios G. Anagnostopoulos, Athanasios I. Kyritsis,
Michel Deriaz and Dimitri Konstantas

Information Science Institute, GSEM/CUI
University of Geneva
Geneva, Switzerland
Email: $\{abbass.hammoud, grigorios.anagnostopoulos, athanasios.kyritsis\}$
$\{michel.deriaz, dimitri.konstantas\}$@unige.ch

***Abstract*—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 (PIR), they require a higher power consumption in general. In this paper we propose *power hopping*, an automatic power optimization method that allows ultrasound motion sensors to optimize their transmitter power level. 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 can be as high as 78%. We also derive an upper bound limit of the method's convergence time.**

I. 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. Occupancy sensors have a wide range of applications, 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 [1].

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 the environment [2]. Different works have suggested algorithms to enhance the performance of PIR sensors and the processing of their output [3], [4], [5], [6]. 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. 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 works [7], [8], [9] use this technique to infer the occupancy at a specific location. 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 [10], [11], and [12].

Ultrasonic motion sensors are promising as they are more sensitive and accurate than PIR ones [1]. 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. Despite these advantages, ultrasound motion sensors are sill 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.

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.* [13] 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, we state that the required power for ultrasonic motion sensors is not fixed, but rather it 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

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 SmartHeat (aal-2014-153).

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, which can reach up to 78% in power reduction, depending on the sensor's environment.

The rest of this paper is organized as follows. First, Section II introduces some necessary details about the operation of ultrasound motion sensors. Section III explains the concept and algorithm of the suggested power hopping method. In Section IV we derive an upper limit for the convergence time, and in Section V we present the experimental evaluation of the method. Finally, Section VI concludes the paper and presents the future work.

II. 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:

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

To detect movements, the frequency spectrum of the current frame is compared against a reference still frame. The still frequency spectrum $\boldsymbol{Y_{still}}$ 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}:

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

where c is the speed of sound in air. The result of Equation 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 to be 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:

$$motion\ intensity = \frac{motion\ score}{threshold} \tag{5}$$

In a previous work, we show how the motion detection parameters ($\boldsymbol{Y_{still}}$, *threshold*) can be obtained automatically through self-calibration [14].

III. Power Hopping Method

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

$$P_{total} = P_{transmitter} + P_{receiver} + P_{processing} \tag{6}$$

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.

A. 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 $\boldsymbol{P_{max}}$ and $\boldsymbol{P_{min}}$ be respectively the maximum and minimum power levels of the system. $\boldsymbol{P_{max}}$ allows the system to work in all conditions. However, according to the sensor's environment, the system may still achieve the same 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. The value of $P_{optimal}$ should be between $\boldsymbol{P_{max}}$ and $\boldsymbol{P_{min}}$:

$$\boldsymbol{P_{min}} \leq P_{optimal} \leq \boldsymbol{P_{max}} \tag{7}$$

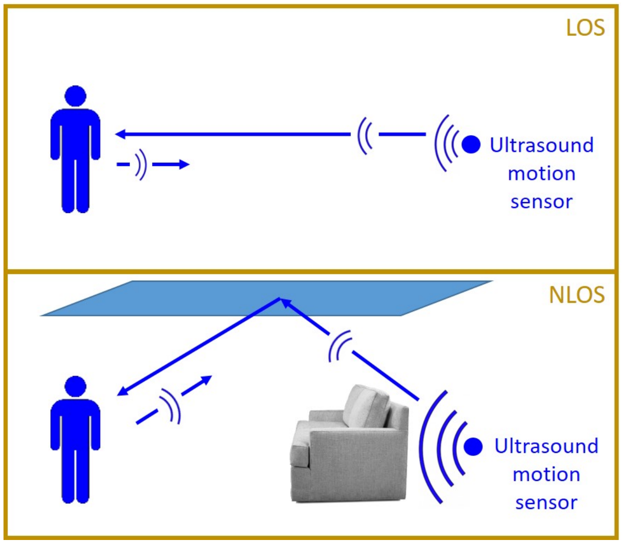


Figure 1. A different transmitter power is required in each case.

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.

B. 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 $\boldsymbol{Y_1}$ 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 $|\boldsymbol{Y_2}|$ 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 $|\boldsymbol{Y_{still}}|$ 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 $|\boldsymbol{Y_{still(new)}}|$ will be equal to $\{\alpha \times |\boldsymbol{Y_{still}}|\}$ (or $\{\sqrt{\beta} \times |\boldsymbol{Y_{still}}|\}$).

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

C. Power Hopping Algorithm

Initially, the transmitter power that is used by the system is $\boldsymbol{P_{max}}$. The parameters of the system ($|\boldsymbol{Y_{still}}|$, *threshold*) that are initially used correspond to $\boldsymbol{P_{max}}$. 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 $\boldsymbol{P_{max}}$, while $P_{invalid}$ is the transmitter power level which is too weak to detect motions and is initialized to $\boldsymbol{P_{min}}$.

When a motion is detected inside the room, the system hops between P_{valid} and $P_{candidate}$ back and forth several times. The number of times the system hops between two power levels is a design choice parameter, which we call it n_{hops}. For $P_{candidate}$ to be considered valid, it should detect the motions that P_{valid} can detect with the same intensity, otherwise it is considered invalid. When the system hops between P_{valid} and $P_{candidate}$, it calculates every time the motion score:

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

Note that in the previous equation, the new still frequency spectrum is calculated using the reasoning of corollary III-B (hence the square root in the equation).

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

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

The new threshold value is calculated as discussed in corollary III-B as well. The first condition in Equation 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 ($|\boldsymbol{Y_{still}}|$, *threshold*), otherwise it picks another candidate power level, as the middle value between P_{valid} and $P_{invalid}$. 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.

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 $(P_{valid} - P_{candidate} > \epsilon)$ 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

D. Power Hopping Example

In this section, we illustrate an example case showing how the power hopping method works. We consider the scenario shown in Figure 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 $P_1 = (P_{max}+P_{min})/2$.
- 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 $P_2 = (P_1+P_{min})/2$.
- 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 $P_2 - P_{candidate} < \epsilon$, 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.

IV. 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:

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

Solving for $n_{iterations}$ yields:

$$n_{iterations} < 1 + \log_2\left(\frac{P_{max} - P_{min}}{\epsilon}\right) \tag{12}$$

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

$$n_{iterations} = 1 + \lfloor \log_2\left(\frac{P_{max} - P_{min}}{\epsilon}\right) \rfloor \tag{13}$$

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:

$$\begin{aligned} t_{max} &= 2 \times n_{hops} \times t_{transmission} \\ &\times (1 + \lfloor \log_2\left(\frac{P_{max} - P_{min}}{\epsilon}\right) \rfloor) \end{aligned} \tag{16}$$

In our design, we choose n_{hops} to be 3, as a middle choice to make the switching decision robust while keeping the time required short enough. For a transmitted signal duration of $10ms$, and a desired resolution of $\epsilon = (P_{max}-P_{min})/128$, the maximum convergence time would be $t_{max} = 0.48sec$

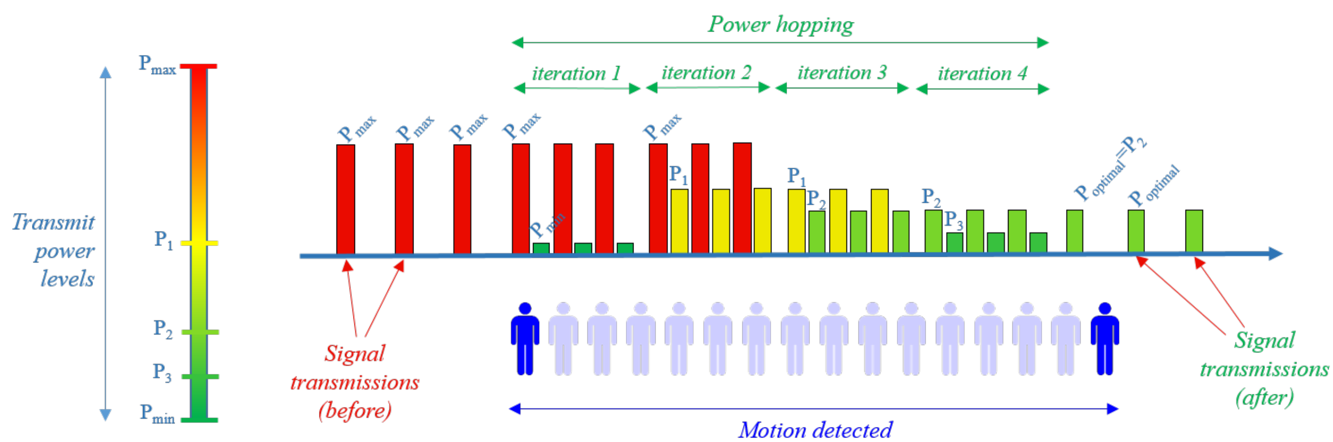


Figure 2. Power hopping example: adapting to the optimal transmitter power level.

Figure 3. Prototype of the ultrasound motion sensor used for testing.

Table I. Power saving as a result of the power hopping method

Case	Original Transmit Power	New Transmit Power	Power Saving
#1: large room, LOS	P_{max}	$0.31P_{max}$	69%
#2: large room, NLOS	P_{max}	$0.96P_{max}$	4%
#3: small room, LOS	P_{max}	$0.22P_{max}$	78%
#4: small room, NLOS	P_{max}	$0.76P_{max}$	24%

V. Experimental Evaluation

A. 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 Figure 3, is composed of a transmitter (a commodity speaker) and a receiver (a commodity microphone) both connected to a Raspberry Pi board [15], which acts as the control/processing unit, and implements the described power hopping method. The transmitted signal frequency used is 21kHz, and the frequency band considered in the processing of the received signal is 20kHz-22kHz.

B. Testing Methodology

The prototype of the sensor is fixed inside a certain area. For a chosen maximum transmitter power P_{max}, the detection parameters ($|\boldsymbol{Y_{still}}|$, *threshold*) are calibrated as described in [14]. 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 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 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 4 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}.

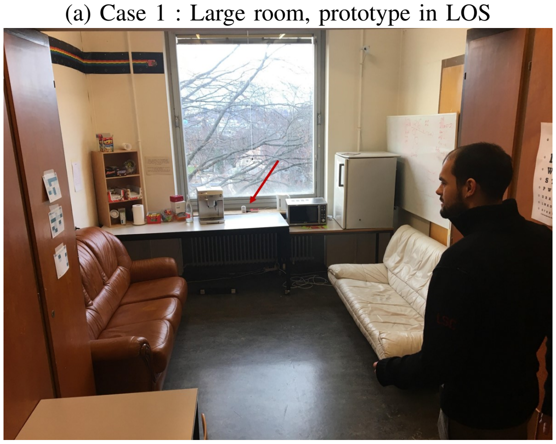
C. Results

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

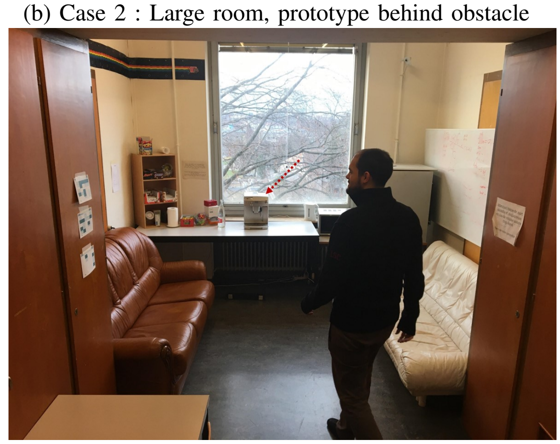
The power hopping method aims to find the optimal transmitter power, and to cut unnecessary transmitter power amount. In general, we see that an important power saving in the transmitter power can be achieved (up to 78% as in case 3). The obtained results show that the required transmitter power varies 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

(a) Case 1 : Large room, prototype in LOS

(b) Case 2 : Large room, prototype behind obstacle



(c) Case 3 : Small room, prototype in LOS



(d) Case 4 : Small room, prototype behind obstacle

Figure 4. Illustration of the different test cases

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.

VI. Conclusion and Future Work

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. A future plan is to make the system run the power hopping method occasionally, in order to validate the obtained optimal power level in time, and reflect any potential changes in the sensor's environment. In this regard, we will investigate techniques that would allow the system of detecting major changes in the environment in order to trigger the power hopping process automatically.

References

[1] T. Teixeira, G. Dublon, and A. Savvides, "A survey of human-sensing: Methods for detecting presence, count, location, track, and identity," *ACM Computing Surveys*, vol. 5, no. 1, 2010.

[2] X. Guo, T. D. K., G. P. Henze, and C. E. Waters, "The performance of occupancy-based lighting control systems: A review," *Lighting Research Technology*, vol. 42, no. 4, 2010.

[3] J. Yin, M. Fang, G. Mokhtari, and Q. Zhang, *Multi-resident Location Tracking in Smart Home through Non-wearable Unobtrusive Sensors*. Cham: Springer International Publishing, 2016, pp. 3–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-39601-9_1

[4] S. Narayana, R. V. Prasad, V. S. Rao, T. V. Prabhakar, S. S. Kowshik, and M. S. Iyer, "Pir sensors: Characterization and novel localization technique," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN '15. New York, NY, USA: ACM, 2015, pp. 142–153. [Online]. Available: http://doi.acm.org/10.1145/2737095.2742561

[5] X. Luo, T. Liu, B. Shen, Qinqun, L. Gao, and X. Luo, "Human indoor localization based on ceiling mounted pir sensor nodes," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 868–874.

[6] J. Kuutti, P. Saarikko, and R. E. Sepponen, "Real time building zone occupancy detection and activity visualization utilizing a visitor counting sensor network," in *Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference on*, Feb 2014, pp. 219–224.

[7] G. Mokhtari, Q. Zhang, G. Nourbakhsh, S. Ball, and M. Karunanithi, "Bluesound: A new resident identification sensor - using ultrasound array and ble technology for smart home platform," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[8] P. Jaramillo and J. P. Linnartz, "Hidden markov model for improved ultrasound-based presence detection," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, Oct 2015, pp. 68–74.

[9] L. I. L. Gonzalez, U. Großekathöfert, and O. Amft, "Novel stochastic model for presence detection using ultrasound ranging sensors," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, March 2014, pp. 55–60.

[10] D. Caicedo and A. Pandharipande, "Ultrasonic arrays for localized presence sensing," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 849–858, May 2012.

[11] B. Raj, K. Kalgaonkar, C. Harrison, and P. Dietz, "Ultrasonic doppler sensing in hci," *IEEE Pervasive Computing*, vol. 11, no. 2, pp. 24–29, Feb 2012.

[12] A. Mehmood, J. M. Sabatier, M. Bradley, and A. Ekimov, "Extraction of the velocity of walking human's body segments using ultrasonic doppler," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. EL316–EL322, 2010.

[13] P. Mishra, H. N. Shankar, P. D. G., and M. Mathur, "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*, Nov 2009, pp. 579–584.

[14] A. Hammoud, M. Deriaz, and D. Konstantas, "Ultrasense: A self-calibrating ultrasound-based room occupancy sensing system," *Procedia Computer Science*, vol. 109C, pp. 75–83, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT 2017, May 2017, Madeira, Portugal.

[15] "Raspberry Pi 3 Model B," https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, [Online].