BULETINUL ŞTIINŢIFIC al Universităţii "POLITEHNICA" din Timişoara, România,
Seria AUTOMATICĂ ŞI CALCULATOARE
SCIENTIFIC BULLETIN of The "POLITEHNICA" University of Timişoara, Romania,
Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol. 56 (70), No. 2, June 2011, ISSN 1224-600X

Improving the Sound Recording Quality of Wireless Sensors Using Automatic Gain Control Methods

Gábor Gosztolya* and László Tóth**

* Department of Informatics, University of Szeged, Árpád tér 1., 6720 Szeged, Hungary
Phone: (+36 62) 54-67-13, Fax: (+36 62) 546-737, E-Mail: ggabor@inf.u-szeged.hu, WWW: http://www.inf.u-szeged.hu/~ggabor

** Research Group on Artificial Intelligence of the Hungarian Academy of Sciences., Tisza Lajos krt. 103., 6720 Szeged, Hungary
Phone: (+36 62) 54-41-42, Fax: (+36 62) 546-737, E-Mail: tothl@inf-u-szeged.hu, WWW: http://www.inf.u-szeged.hu/~tothl

Abstract *– When performing speech recording it is desirable to have the speech signal in as a high quality as possible. In everyday recording conditions one of the most important aspects of sound quality is to have a uniform volume level, because it is very hard to understand (and to automatically recognize) an utterance with a volume level that varies considerably. Of course this uniform volume level should also be an average one, avoiding either too loud or too quiet recordings. To overcome this problem usually an approach called Automatic Gain Control is used, which is an adaptive method for controlling microphone sensitivity (gain). Wireless sensors are recent, low-powered devices, which are ideal for recording and transmitting observations such as speech, thus they are a good area for applying automatic gain control. Due to their low power consumption, however, only very simple solutions can be implemented. Here we will present a general gain control algorithm, then introduce two variations that we test in a situation which simulates the actual use. We perform evaluations by using two types of measurement: the first one compares local volume levels to recordings made under ideal conditions, while in the second we measure the understandability of the recordings made by applying standard speech recognition techniques. Our results in both cases confirm that it is indeed an area where automatic gain control can be applied, and that both our algorithms perform well in practice.*

Keywords: *wireless sensors, sound quality, automatic gain control, volume level, speech recognition.*

I. INTRODUCTION

Wireless sensors and wireless sensor networks have become increasingly popular recently. Their main application is the monitoring of their environment like movement detection and measuring temperature and light. They are also capable of recording audio data, which calls for porting a number of speech processing applications. Being a relatively new area, however, a number of quite basic issues for these sensors have to be addressed. Because of their limited processing capabilities, even the porting of the most basic algorithms might require some modifications. In this paper we will focus on a special problem of distant speech recording, namely the automatic adjustment of the recording gain.

The above problem arises from the fact that the positioning of wireless sensors cannot be known in advance. From the viewpoint of speech processing it means that a sensor having a microphone and transmitting what it records does not know how far it is from the speaker. It could also be the case that this target is continuously changing its position, i.e. moving away from or towards the sensor. Another situation might be that there are several speakers, each at a different distance from the sensor; but regardless these difficulties, the wireless sensor should always try to record the actual speech in as high a quality as possible.

Perhaps the most important factor affecting recording quality of a wireless sensor is that of setting the gain of the microphone. This way the sensitivity of the recording process can be affected. A good automatic gain control (AGC) algorithm [1] can eliminate or at least smooth the above-mentioned jumps in the volume level, which makes the effective processing of the signal more straightforward.

The structure of the paper is as follows. First we shall describe the problem of automatic gain control and its importance in our study. Then we will introduce our algorithms applied and explain how they work in detail. Next, we will describe the evaluation methodologies used and our testing environment. Finally, we will present the results obtained and draw our conclusions.

II. AUTOMATIC GAIN CONTROL IN WIRELESS SENSORS

The goal of automatic gain control is to dynamically adjust the sensitivity of the microphone, based on the actual acoustic observations. It is used in various situations where

signals having different strengths (usually from different sources) are present. A typical example is that of telephones or cell phones [1], but it is also used for controlling the amplitude of interfering signals using lasers [2] or in pacemakers [3]. Every case is special, however, thus our sensor-based environment also has its special requirements, which should be considered when developing applications for it. Perhaps the most important one is that, as these sensors were designed to have exceptionally low power consumption, they have very limited resources: they usually have a low-capacity processor, and an extremely small amount of RAM. They communicate via radio waves, which also have a limited bandwidth. In this scenario, applications designed to work on wireless sensors have to use as small a CPU time and RAM as possible.

The main reason for using gain control in wireless sensors is their limited resolution: for each sample we can represent it only using 10 bits of information, so we should attempt to use as much of it as possible. We could also follow the strategy of sending the information recorded as is, and amplify it later, in the device that receives the speech data. It would clearly have the advantage that we would not be handicapped by the computational limitations of sensors, so we could use CPU-demanding algorithms, or ones that require more memory. In this scenario, however, there would be a clear loss of information as each sample has this fixed 10-bit long representation. For a louder-speaking person large-amplitude values would get clipped, which could have been avoided if we had been using a lower gain value. In other cases, when the speech recorded is too quiet, we would not be using the whole 10 bits available, but only a part of it, which could be avoided by using a higher gain value. And it is clear that in both cases the quality of the recordings would suffer.

Fortunately in our wireless sensor environment the gain can be adjusted. It is represented on one byte only: the higher this value is, the more sensitive the microphone becomes, and vice versa: a lower value means less sensitivity. From experience we know that the microphone has a quasi-linear sensitivity as a function of the gain value, and its default value is 64.

III. THE ALGORITHMS APPLIED

Next, we will describe our algorithms introduced for automatic gain control. An arbitrary AGC method can be summed up in one sentence: *if the signal is too strong, lower the gain; otherwise, if the signal is too weak, increase the gain*. This simplicity, however, leads to difficulties at the point of its application: we could find no mention of a general AGC algorithm in the literature. It seems that all the details are heavily application-dependent, which is convenient for determining the threshold values for the concepts *too strong* and *too weak*, or finding out the ideal frequency of performing the checks and modifying the gain. On the other hand, there is no standard way even for the process of increasing or decreasing the gain (which seems to be an issue for which quite general solutions should exist), and further, we have to come up with a way of measuring the strongness or weakness of the signal (loudness in the case of sound signals). In this paper we sought to introduce actual algorithms for our particular problem, thus all these open issues have to be addressed. These issues were solved in a similar way as that for our algorithms introduced, and of course they suffer from the same hardware limitations, thus the resulting methods are indeed quite similar. Due to this, we begin with their common properties, defining a general gain control algorithm. In the following we will denote the actual gain value by *gain*, while *gain'* will be the new gain value; of course $0 \leq gain, gain' \leq 255$.

A. Working with Packets

One characteristic of wireless sensors is that they communicate via radio waves, sending a small chunk of data called a *packet* at a time. In our case there can be at most 114 bytes in a packet; assigning one two-byte integer to the number of the packet (to be able to recognize missing packets) 112 bytes remain. As the digital-analogue converter (DAC) of the microphone supplies observations of 10 bits per sample, we can send 88 10-bit samples in a packet, and the two remaining bytes could be used to send extra information. Of course the actual values may vary somewhat between different hardware, but the main concept (having small-sized packets which implies working with a small, fixed number of samples at a time) remains the same.

This arrangement makes it straightforward to handle all observed data in groups of 88 samples, i.e. $A = a_1 \ldots a_N$, and in our architecture $N=88$. Thus the input signal is organised into a (theoretically) endless flow of packet A_i-s, the last one being A_t; it makes it plausible to perform the same actions for each packet A_i. Of course some procedures may be performed after every nth packet, but the same lines of code can still be executed for each packet, and these do not refer to samples of another packet. (Although using some value representing another packet as a whole (like its energy level) is allowed.)

B. Relying on the Energy Level

Another common feature of our algorithms is that they both rely on the energy level of the speech signal observed. As the energy of a signal is closely connected with its volume level, controlling the energy level means controlling the volume. Moreover, the calculation of energy is computationally very cheap, which is a vital requirement in our case. In the actual solution we followed the packet-oriented strategy described above. First we calculated the mean of the values in the packet as the signals may contain a DC bias, i.e.

$$base_A = \frac{1}{N}\sum_{i=1}^{N} a_i . \quad (1)$$

Next we calculated the energy level of the packet, which is usually done by taking the squared sum of its values. Due to speed limitations we did not raise them to the second power; instead we just added up the absolute values of the difference of the sample and the above-calculated mean value, i.e.

$$energy_A = \sum_{i=1}^{N} |a_i - base_A|. \quad (2)$$

This value was then treated as the energy level of packet *A*, used both for voice activity detection (see below) and for measuring the loudness of the actual signal observed in order to control the gain. As our sensor boards produced sound signals with a 8861 samples per second sampling rate, one packet corresponds to roughly 10ms of the speech signal. We stored the energy levels of the last 50 packets, examining about half a second at a time; the sum of these values gave the full energy of this interval, i.e.

$$fullenergy_t = \sum_{i=t-49}^{t} energy_{A_i} . \quad (3)$$

Gain adjustments (including those of voice activity detection) were made every 10 packets, i.e. in roughly 100ms time intervals.

C. Voice Activity Detection

A typical gain control algorithm seeks to normalise the level of the observed signal to a pre-set value. This aim, however, becomes counterproductive if there is no voice activity at all; in this case only the basic noise of the microphone is present, which will be amplified to the highest level available by setting the gain very high. When the silent period ends, the first part of speech will be overamplified, leading to clipping and it will result in a loss of information. To overcome this problem we should detect these longer periods of silence, and set the gain level to an intermediate value there. This way we will not lose too much information at the beginning of the next speech portion (either if the speaker is too close or too far away), and then we can react to the current volume level quite quickly. In the field of speech recognition the problem of detecting longer silent parts is called Voice Activity Detection, and there exist several algorithms to solve it [4] [5]. The simplest of them are based on calculating some kind of loudness of the signal present, and treating it as silence if this loudness remains under a threshold for a long period.

We also followed this approach of voice activity detection; thus, to use sparingly the limited resources of wireless sensors, we based the Voice Activity Detection process on the same measurements as the actual gain control, reusing the values calculated above. That is, we used the full energy level of the last 50 packets (roughly 500ms), calculated in (3). For the sake of simplicity we assumed that the gain value was the same throughout this interval, which in practice worked quite well. After every ten packets (i.e. every 100ms) we checked to see whether the energy level for the last half a second divided by the gain value and a small constant stayed below a threshold T_{SIL}; if so, we considered it silence, and set the gain to an average value $gain_{SIL}$. So we used the condition

$$\frac{fullenergy_t}{gain + c_0} < T_{SIL} , \quad (4)$$

where c_0 was determined empirically. (It was necessary because using a *gain* value of 0 does not mean silence in our architecture.) If we interpreted the last part of signal as silence, it also meant that we did not adjust the gain any further this time.

D. The Common Parts in the Two Algorithms

Next we will introduce two algorithms which, based on their observations, dynamically adjust the gain level. Besides the similarities described above (working with packets, using packet-size energy levels and applying voice activity detection) there is another common aspect of their behaviour. They seek to keep the total energy of the last 500ms (i.e. $fullenergy_t$) at a T_{IDEAL} level; for this, if the full energy is lower than a threshold T_{LOW}, the gain is increased, whereas if it is higher than T_{HIGH}, it is decreased. T_{LOW} was set to 80% of T_{IDEAL}, while T_{HIGH} was 120% of it, which values were determined by simple preliminary tests. We used another threshold (T_{UHIGH}, being twice that of T_{HIGH}): if the full energy level exceeded this value, we supposed that the signal was clipped because its loudness was excessive, treating it as an overdrive was present, which called for an immediate and drastic reaction. At such times the gain was decreased by using the formula $gain' = ¾ \cdot gain$ based on preliminary tests again. For the pseudocode of this general algorithm, see Table 1.

TABLE 1. General Gain Control Algorithm.

Step	Instruction
1	*fullenergy* is the total energy of the last 50 packets
2	**if** *fullenergy* / (*gain* + c_0) < T_{SIL} **then**
3	*gain* is set to an intermediate value $gain_{SIL}$
4	**else** if *fullenergy* > T_{UHIGH} **then**
5	*gain* is reduced to its 3/4th
6	**else if** *fullenergy* > T_{HIGH} **then**
7	decrease *gain*
8	**else if** *fullenergy* < T_{LOW} **then**
9	increase *gain*
10	**end if**

E. Equal-Stepping Gain Control Algorithm

In this algorithm (*ES GCA*) we adjust the gain in small, equal steps: to increase the gain level we used the formula

$$gain' = gain + step,$$

and we applied

$$gain' = gain - step$$

to lower it. Naturally the best value of *step* has to be determined, which must be a positive integer. This algorithm is a simple and straightforward way of controlling gain.

F. Weighted-Average Gain Control Algorithm

The previous algorithm makes equal-sized steps up- and downwards regardless of the difference between the measured energy level and the ideal one. However, it might make sense to have big steps if this difference is big, and only small ones if it is small. The second algorithm (the *WA GCA*) follows this strategy.

Assuming that the energy level of a recording is linearly proportional to the gain used to obtain it (which is roughly the case for our particular sensor boards), we have that

$$\frac{fullenergy_{ideal}}{gain_{ideal}} = \frac{fullenergy_{t}}{gain}, \qquad (5)$$

where $fullenergy_{ideal}$ is the full energy level recorded under ideal conditions, determined by preliminary tests, $gain_{ideal}$ is the corresponding gain, $fullenergy_t$ is the total energy level recorded, and *gain* is the actual gain value. This formula can be rearranged to express the expected gain level; that is,

$$gain_{ideal} = \frac{fullenergy_{ideal} \cdot gain}{fullenergy_{t}}. \qquad (6)$$

Using this formula directly, however, would lead to frequent big jumps in the gain level used, which would clearly harm the recorded speech signal. To counter this effect we averaged the previously used gain level with this calculated one weighted via

$$gain' = w \cdot gain_{ideal} + (1 - w) \cdot gain, \qquad (7)$$

where $0 \le w \le 1$ is a weighting constant. After substituting $gain_{ideal}$ from (6) we get

$$gain' = \frac{gain\left(w \cdot fullenergy_{ideal} + (1 - w) fullenergy_{t}\right)}{fullenergy_{t}}. \qquad (8)$$

With the *w* weighting factor we can make the transition of gain much smoother by eliminating sudden jumps. Needless to say, the optimal value for *w* first has to be determined.

IV. THE TESTING PROCESS

Having defined the problem and presented the algorithms, next we will turn to a description of the testing process.

A. Hardware

In this study we used Crossbow Iris sensor nodes (*motes*) that have a 7.37 MHz processor with a RAM of *8K* bytes and a programmable flash memory of *128K* bytes. The microphone and other input peripherials are located on a piece of hardware that can be attached to the mote, on the so-called *sensor board*. We had Crossbow MTS300 sensor boards, which, besides the microphone, also contain light and temperature sensors.

B. The Recording Environment

To simulate real-life conditions we made recordings at four different distances: we positioned the sensor at 20, 50, 100 and 200 centimeters away from the speaker. The 50 centimeter-long distance served as an ideal recording position, 20 centimeter as the speaker being too close to the sensor, while the 100 and 200 centimeter values simulated speakers being quite far away. Testing was performed on Hungarian broadcast news: a five-minute-long signal was used. It was then modified to contain a wide range of volume level changes such as slowly growing quieter (simulating the speaker going away from the microphone), slowly growing louder (the speaker is approaching the microphone) and sudden jumps (simulating multiple speakers being present at different distances from the microphone). The same modified signal was played at different distances to the sensors with different gain control algorithms, but otherwise the situation (volume level, position) was exactly the same as before, so that their performances could be compared.

We made recordings using the two gain control algorithms at all four distances. To get a reference recording we recorded the original signal (i.e. the one with constant volume) at 50 centimeters with a fixed gain value. We then made two additional recordings at all four distances. First the original, constant volume-level signal was recorded without using gain control; these (the *basic recordings*) were treated as the ones in quasi-ideal circumstances: the speaker is in a fixed position and has a constant voice level, but it is not necessarily the optimal one: it could be too high or too low. The performance of the one at 50cm, compared to the reference recording, served as the *glass ceiling*: as these recordings are as near identical as possible, its score is the highest one available (at least in theory), and this value cannot be exceeded no matter what sophisticated gain control algorithms we develop. Next the signal with changes in volume level was recorded at each of the four distances, still without gain control. These were the *baseline recordings*, and their performance scores simulate the results we would get in real-life situations (multiple or moving speakers) without gain control. For a list of recordings made, see Table 2.

TABLE 2. The recordings made and the distances used.

Recording	Volume	Distance			
		20cm	50cm	100cm	200cm
Reference	constant		●		
Basic	constant	●	●	●	●
Baseline	varying	●	●	●	●
ES AGC	varying	●	●	●	●
WA AGC	varying	●	●	●	●

C. *Evaluation via the Energy Level*

One standard way of evaluating a gain control algorithm is to calculate the energy levels of the reference and the resulting recordings, and take their difference or ratio. Fortunately the energy level is very easy to calculate, and the volume of speech is directly related to its energy, so the more similar the energy levels of the original and the gain-controlled recordings are, the closer they are. The energy is calculated by taking the squared sum of the samples, i.e.

$$energy_S = \sum_{i=1}^{T} s(i)^2, \tag{9}$$

where s is the observed speech signal having a length of T. (Note that the evaluation of the resulting signals was not done on the wireless sensors, so we could apply even computationally demanding algorithms for it. This way we were not forced to calculate the energy using absolute values, but we could take the squared sum of the samples instead.) In this form it has one value for the whole signal, which could indeed be of interest; for this reason we will calculate

$$engratio_S = \frac{energy_S}{energy_{ref}}, \tag{10}$$

where $energy_{ref}$ is the total energy of the reference recording. This value, however, ignores the local variations within the signals: two recordings with quite different local values could have the same overall energy level. To overcome this weakness we introduced another measure. We calculated the energy levels in 500ms-long windows with a 80% overlap, moving the window in 100ms-long steps. To compare the energy of two signals (the reference one and one using a gain control algorithm) we calculated a squared error-like value by taking the squared sum of the difference between the energy levels of the corresponding windows:

$$diff_{A,B} = \sum_{i=1}^{K} \left(energy_{A(i)} - energy_{B(i)}\right)^2, \tag{11}$$

where $energy_{A(i)}$ is the energy level of signal A in the ith window, and K is the number of windows. One signal will always be the reference recording, thus we get one value for each recording made. These scores can be easily compared: the lower this number is, the closer the recording is to the reference one, thus the better it is. As these values are difficult to read, we calculated their *relative error reduction* (or *RER*) scores as well: the appropriate baseline recording had a score of 0%, whereas the reference recording having an energy difference of 0 had a score of 100%. The intermediate values were assigned linearly, e.g. for a baseline energy difference of 5000 and a score of 1500, the RER value will be 70%, as 5000 - 1500 is 70% of 5000, meaning this much of the error was eliminated.

D. *Evaluation via Sentence Recognition*

Energy levels can be calculated quickly, and they can be used very reliably to estimate the difference between the volume levels of two recordings. But this approach has a serious limitation: we adjust the gain to make the recording *more understandable*; if two signals have quite different energy levels, but both can be understood very well, this technique cannot detect it. Unfortunately, understandability is not a well-defined notion. Hence to assess it, we turned to standard techniques of speech recognition.

The aim of speech recognition is to transform spoken words to written text. In a typical speech recognition process, first features [6] are calculated from the input signal, usually on the basis of its spectral representation [7], which process is called *feature extraction*. In the next step, following the frame-based approach [8], small, equal-sized parts (the so-called *frames*) are classified independently and assigned to one of the possible phonemes, which is the phoneme classification subtask [9]. It is usually done by applying some statistical machine learning algorithm like Gaussian Mixture Models (GMMs) [10] or Artificial Neural Networks (ANNs) [11]. (The combined steps performed up to this point are usually called the *acoustic model*.) Next, based on the result of the classification and the probabilities of possible word-sequences (which are supplied by a *language model*) the most probable word sequence is chosen, which will be the transcript of the input speech signal. The accuracy value of this process can be determined by comparing this result to the real word sequence belonging to the speech signal. As one can see, it is a quite standard process, which makes it feasible for generating automatic measures.

One interesting aspect of this process is that, similar to the case of human hearing and comprehension, the accuracy value obtained decreases when the quality of the played recording becomes worse. It is so because in this case the input signal contains less and less information, which means that the signal processing and feature extraction parts calculate more noisy features. It produces a less and less reliable phoneme classification, finally resulting in word-level mistakes. It is known, however, that current speech recognition systems are much more sensitive to noise than human listeners.

One might find it surprising that this behaviour is actually beneficial for the application of sentence-level speech recognition for measuring the understandability of a recording, but this is due to two main reasons. Firstly, it follows the way human hearing works; and if we want to measure the amount of *human* understandability, any method that mirrors human hearing is of course helpful. And secondly, it is likely that, on the receiver side, we do not want to play the transmitted signal to a human, but we would like to process it by an automatic speech recognition system. In this case today's speech recognition technologies are quite capable of measuring the quality of the signal played.

In practice we followed the frame-based approach [8]: we divided the speech signal into small, equal-sized parts, which – after feature extraction – were classified as one of the possible phonemes. We could have measured the quality of signals based only on the result of phoneme classification [9]; performing sentence recognition, however, is a higher-level concept, which seems to be more meaningful.

We applied the standard 13 MFCC coefficients along with their derivatives and the second derivatives (MFCC + Δ + ΔΔ for short) [12] as features for phoneme classification, and applied Gaussian Mixture Models (GMMs) for it with 11 components [10].We performed the training of these GMMs on recordings of broadcast news, also recorded by using the wireless sensors, but this time using the fixed distance of 50 centimeters. The features were calculated by utilising the HTK toolkit [13].

For language modeling usually a solution called *N-gram modeling* [14] is used in speech recognition systems. In it the occurrence of all N possible successive words are counted, from which the probability of the Nth word can be determined; then the probability of a word sequence can be calculated by multiplying the probability of each word occurring. This approach unfortunately has the drawback that a quite big and relevant database (in our case written texts of broadcast news) is required, thus we opted for another, although simpler solution: we simply listed the possible words and allowed any combination of them with the same probability.

Measuring the performance of a continuous recognizer speech recognition application is somewhat more complicated than measuring it for an isolated word recognizer, where word-level accuracy scores can be determined readily as the number of exact word matches. In the case of sentence recognition the common method is to calculate the word-level *edit distance* of the two word sequences (the real and the resultant); that is, we construct the resulting sentence from the real transcript by using the following operations: inserting and deleting words, and replacing one word with another one. These operations have some cost (in our case the common values of 3, 3 and 4 were used, respectively), and then we pick an operation set having the lowest cost. Now we can calculate the following measures:

$$Correctnes \quad s = \frac{N - S - D}{N} \tag{12}$$

and

$$Accuracy \quad = \frac{N - S - D - I}{N}, \tag{13}$$

where N is the total number of words in all the original sentences, S is the number of substitutions, D is the number of deletions and I is the number of insertions. Correctness does not take the number of insertions into account, but the accuracy value can be negative, which is why usually both scores are used.

We again calculated the relative error reduction scores, where we had several options of choosing the maximum value for both the accuracy and correctness values. We could pick *100%* as the maximum, as is common in speech recognition. The drawback of this choice is that it totally ignores the recording conditions, and assumes that perfect recognition can be achieved. The reason for this is that in the field of speech recognition usually the best configuration is tuned, which in our case is the recording we chose for the role of a glass ceiling.

In the remaining two choices we do not consider *100%* accuracy as the maximum; instead we took the performance of a basic recording. Here we could choose either the one at the corresponding distance, or the one made at *50cm* (having the glass ceiling value). As both are valid options, we also calculated both ratios. The (original) error value is the difference between the accuracy scores of the basic and the baseline recordings; to express how much of it was eliminated (which is the RER score), we calculated the difference between the accuracy score of the actual and the baseline recordings, and divided it by the error value. These scores are referred to as "same distance" and "glass ceiling" RER values, whereas the first version, described above, was called the "absolute" RER score. (This process, of course, was repeated for the correctness values as well.)

V. RESULTS

First we had to set the parameter of both algorithms to the optimal value: for this we found the interval of values which worked well by preliminary tests, then explored it with a small step size. The *step* parameter of the Equal-Stepping Algorithm (ES) was tested between *1* and *6*, whereas for the Weighted-Averaging Algorithm (WA) w was tested between *1/32* and *10/32* with a step size of *1/32*.

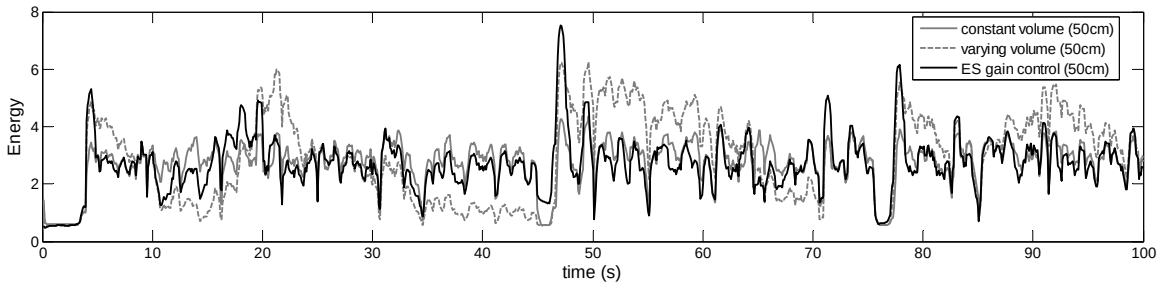

Fig. 1. Energy levels of the reference recording (grey continuous line), the baseline recording (grey rugged line) and the *ES* algorithm with *step = 3*.

A. Results Using the Energy Level

We found that the best parameter values were *step = 3* and *w = 5/32* when evaluating the recordings in terms of their energy level. First the full energy ratio was calculated for each recording (see Table 3). As can be seen, the distance clearly affects the energy ratios when we do not use a gain control: the recordings made at 20cm had a *50%* bigger total energy than the reference recording, whereas the recordings made at 100 and 200 centimeters had significantly less. Both gain control algorithms, however, could indeed compensate for the overall loudness (or quietness) at these distances, resulting in a total energy ratio very close to *1*. For the whole parameter intervals tested, this ratio varied between *0.77* and *1.08* for the ES algorithm and between *0.87* and *1.11* for the WA method.

TABLE 3. Full energy ratios of the two gain control algorithms (ES and WA).

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	1.53	1.00	0.75	0.51
Baseline	1.59	1.09	0.87	0.62
ES AGC, *step* = 3	1.05	0.96	0.93	0.86
WA AGC, *w* = 5/32	1.08	1.02	0.95	0.88

The energy level difference *diff* values, calculated according to (11), can be seen in Table 4, while the relative error reduction scores are shown in Table 5. It is not surprising that the *diff* values of the basic and baseline recordings increase when the distance changes from the optimal one. The only exception is the baseline recording at 100cm: it has a lower *diff* score than at 50cm, which is probably due to the high number of loud parts in the signal played. It may also be why a smaller score is obtained for the baseline signal than for the basic one at 200cm, leading to the negative RER score for the latter. The basic recording from 50cm has an exceptionally small difference value due to the indeterministic nature of recording (i.e. it was not *exactly* the same as the reference one). Both gain control methods, however, performed quite well. As the distance varied from the optimal one, the *diff* values increased slightly, but the RER scores reflect the fact that using gain control was an effective way of countering this effect. The *66.44-86.88%* and *62.12-83.39%* RER values (ES and WA algorithms, respectively) are quite good, and in almost every case these are higher scores than those of the basic recordings. The only exception is at 50cm, but it was practically impossible to beat this score (*99.93%*) there, and the values exceeding *70%* are also quite satisfactory.

TABLE 4. Energy differences of the two basic recording types and of the two gain control algorithms (ES and WA) relative to the reference recording.

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	7341	3	1647	6412
Baseline	13953	4439	3728	5835
ES AGC, *step* = 3	1580	1161	1251	1933
WA AGC, *w* = 5/32	2318	1300	1412	1761

TABLE 5. Energy difference relative error reductions scores of the two basic recording types and of the two gain control algorithms (ES and WA).

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	47.39%	99.93%	55.82%	-9.89%
Baseline	0.00%	0.00%	0.00%	0.00%
ES AGC, *step* = 3	88.68%	73.85%	66.44%	66.87%
WA AGC, *w* = 5/32	83.39%	70.71%	62.12%	69.82%

Visually inspecting the energy levels at 50cm using the ES algorithm with *step = 3* (see Figure 1), we may say that the algorithm seems to be quite effective. (The WA method produced a very similar curve with *w = 5/32*.) While the energy levels of the baseline recording greatly differ from the reference one, the gain control algorithm compensated for the jumps in volume: it usually differs from the reference recording by only a small amount. The only weakness of the method seems to be the periods after longer silences, where it resulted in much higher energy values than those of the reference.

Figure 2. shows the energy levels of the basic recordings (in the upper box), and of the ES algorithm with *step = 3* (in the lower box) at each distance tested. It can be clearly seen that the distance between the sensor and the sound source strongly affects the energy levels when there is no gain control: the four corresponding curves are quite different from each other. (Note that energy is displayed on a log-scale.) On the other hand, the energy levels of the recordings using a gain control algorithm fall fairly close to

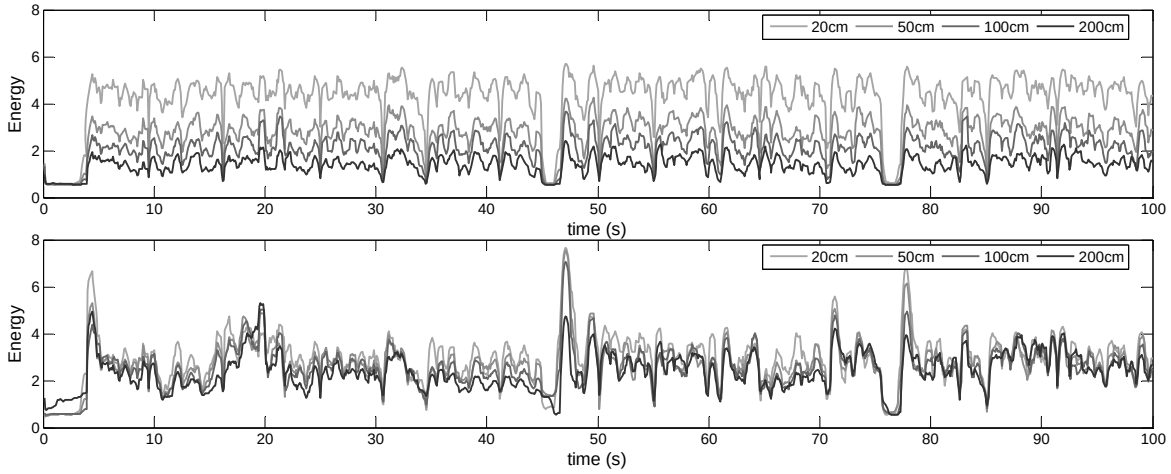

Fig. 2. Energy levels of the basic recording (up), and the signal with varying volume using the *ES* algorithm with *step = 3* (down) at different distances.

each other, indicating that the method was able to amplify sources having different volumes to roughly the same level, which agrees with our previous findings involving total energy ratios.

B. Results Using Sentence Recognition

Unlike evaluating the gain control methods in terms of the energy levels, we found no definite best parameter values when performing sentence recognition. Overall, we chose the parameters *step = 2* and *w = 7/32*, but we shall discuss this issue more throughly later.

The resulting correctness values can be seen in Table 6, while the corresponding accuracy scores are given in Table 7. We achieved *72.59%* and *70.17%* on the reference recording for correctness and accuracy, respectively, which are indeed very close to those of the basic recording made at 50cm (which were is as identical as was practically possible to the reference one). Considering the very noisy recordings due to the small microphone on the sensor board, and the simplicity of the language model (which usually significantly aids the speech recognition process [15]), we found this score surprisingly high.

TABLE 6. Correctness results of the two gain control algorithms (ES and WA). The reference recording produced a score of 72.59%.

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	67.07%	72.24%	52.41%	5.52%
Baseline	54.48%	59.31%	38.28%	7.07%
ES AGC, best values	74.31%	68.45%	43.45%	10.00%
ES AGC, *step* = 2	74.31%	65.86%	41.21%	10.00%
RER (same distance)	157.51%	50.66%	20.74%	–
RER (glass ceiling)	111.66%	50.66%	8.63%	4.50%
RER (absolute)	43.56%	16.10%	4.75%	3.15%
WA AGC, best values	75.17%	68.79%	45.34%	10.17%
WA AGC, *w* = 7/32	75.17%	64.48%	44.48%	10.17%
RER (same distance)	164.34%	39.98%	43.88%	–
RER (glass ceiling)	116.50%	39.98%	18.26%	4.76%
RER (absolute)	45.45%	12.71%	10.05%	3.34%

TABLE 7. Accuracy results of the two gain control algorithms (ES and WA). The reference recording produced a score of 70.17%.

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	63.28%	69.48%	50.17%	5.34%
Baseline	50.17%	56.72%	35.69%	6.55%
ES AGC, best values	70.69%	65.34%	40.86%	8.45%
ES AGC, *step* = 2	70.52%	63.10%	37.93%	8.45%
RER (same distance)	155.23%	50.00%	15.47%	–
RER (glass ceiling)	105.39%	50.00%	6.63%	3.02%
RER (absolute)	40.84%	14.74%	3.48%	2.03%
WA AGC, best values	72.41%	65.52%	43.10%	9.14%
WA AGC, *w* = 7/32	72.41%	64.48%	42.76%	9.14%
RER (same distance)	169.64%	60.82%	48.83%	–
RER (glass ceiling)	115.17%	60.82%	20.94%	4.12%
RER (absolute)	44.63%	17.93%	10.99%	2.77%

The performance of the basic and baseline recordings clearly show that, among the four tested cases, the distance of 50 cm could be considered as the ideal for both cases: the accuracy and correctness scores are the highest using this distance, while they fall when the sensor is closer or further away from the microphone. The extremely small scores of the recordings made at 200cm, however, are surprisingly low. Our previous tests involved phoneme recognition in the same circumstances as we had here [16], and they also showed a decrease in the phoneme classification scores at this distance, but by a much smaller amount (they fell from the 83.19% glass ceiling level to 53.75% and 53.05%, basic and baseline recordings, respectively). It could be, however, that this decrease in the phoneme identification performance made the acoustic model of speech recognition unreliable in practice, which brought the sentence-level recognition performance down to this level. This hypothesis is also corroborated by the small difference in the phonetic accuracy of the two recordings, suggesting that due to the low volume (caused by the very large distance), most parts of the utterances were just not distinguishable from background noise (i.e. silence). Low speaker volume could also be the reason for the higher speech recognition accuracy for the baseline

recording than that of the basic one: the baseline recording had a varying volume, so in a number of cases it was louder, aiding the understanding process in this situation, whereas the lower-volume parts did not degrade accuracy any further. (Note that this result mirrors our findings when we evaluated these two recordings via the use of energy levels: the baseline recording also performed better than the basic one measured by that evaluation metric.) It also means that calculating the relative error reduction using the basic score at the same distance makes no sense in this case.

Examining the performance of the gain control algorithms, perhaps the most interesting finding is that, unlike the evaluation done via energy levels, we could not find an optimal parameter value for either algorithm. Usually the settings which performed best at 50 and 100 centimeters produced worse sentence recognition scores at 20 and 200 centimeters than the other cases, which, in contrast, worked suboptimally when using the former distances. The reason for this is probably that at 50 and 100 centimeters the volume has a relatively small variation, which prefers methods with smaller, smoother changes. On the other hand, recordings made at 20 and 200 centimeters require more flexible gain control methods, which allow greater jumps in the gain. Of course, for a recording application it is unreasonable to expect constant switching between parameter values while recording, thus we chose one parameter setting for both algorithms that we considered best; but we also listed the best scores achieved for both algorithms and for all four distances in all the parameter intervals tested.

Looking at the scores, we may conclude that we could achieve significant improvements by using either of the gain control algorithms described here. The only exception was when we made recordings at 200 centimeters, where, despite the relatively high error reduction scores, the performance scores still remain at the unusable level. It is probably because training was performed on recordings made from a fixed distance (50 centimeters). In theory gain control could counter this effect by raising microphone sensitivity, but the Signal-to-Noise Ratio (SNR) [17] cannot be raised this way, because the background noise is also amplified.

In the other cases, however, the relative error reduction scores based on using the same distance are quite convincing (ranging from 20.74% to 164.34% and 15.47% to 169.64%, correctness and accuracy, respectively), and the other two RER values are also good. A noteworthy case is using gain control when recording at 20 centimeters, where even the glass ceiling value could be exceeded. These results imply that what we regarded as an ideal recording environment (using a constant volume-level recording from the best distance) was not the best one possible. The reason for this is probably that in human speech, even without artificially introducing volume level changes (as we did when constructing the baseline recordings and the ones recorded using gain control), there are also volume level changes present [18] that could also be handled by the use of gain control.

Overall, although there are small differences in the performance of the two methods, both achieved similarly good results, hence both can be recommended for practical use. The small advantage of the Weighted-Averaging Gain Control Algorithm over the Equal-Stepping one could be due to its smoother transition of gain level and its ability to make bigger jumps when required. Also, it was demonstrated that gain control could indeed be effective when using wireless sensors to record speech data, since the understandability of the sound signals transferred (measured in terms of speech recognition sentence-level error scores) improved significantly.

VI. CONCLUSIONS

Wireless sensors are recent, low-capacity devices used for monitoring their immediate environment, which includes recording and transmitting audio information. In this situation, however, there could be a big difference in the volume level of the observed signals due to the presence of multiple and/or moving speakers. Varying volume can indeed harm the understandability of signals, which can be compensated for by applying Automatic Gain Control methods. The two algorithms introduced in this work (which were designed to meet the particular requirements of our set-up, but could also be used elsewhere) proved successful when we measured their performance using the difference in volume levels and when applying sentence-level automatic speech recognition. Overall, the quality of a sound recording made could be improved significantly using some AGC technique like our solutions.

ACKNOWLEDGMENTS

This study was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 programme of the Hungarian National Development Agency.

REFERENCES

[1] S. Ramanathan and M. Steenstrup, “A survey of routing techniques for mobile communications networks”, Mobile Networks and Applications, vol. 1, no. 2, pp. 89 – 104, 1996.

[2] S. Kang, H. Choi, H. Yoon and K. Park, “Automatic gain control for the uniform amplitude of interferent signal in a Laser Doppler Vibrometer”, Proceedings of SICE-ICASE International Joint Conference SICE-ICCAS 2006, Busan, South Korea, pp. 1085 – 1090, 2006.

[3] C. S. Bae, “An automatic pacemaker sensing algorithm using automatic gain control”, Proceedings of the 1999 IEEE Region 10 Conference TENCON 99, Cheju Island, South Korea, pp. 375 – 378, 1999.

[4] Z. Tüske, P. Mihajlik, Z. Tobler and T. Fegyó, “Robust Voice Activity Detection based on the entropy of noise-suppressed spectrum”, 9th European Conference on Speech Communication and Technology Interspeech 2005 - Eurospeech, Lisboa, Portugal, pp. 245 – 248, 2005.

[5] J.M. Górriz, J. Ramírez, J.C. Segura and S. Hornillo, "Voice Activity Detection using higher order statistics", Proceedings of the 8th International Work-Conference on Artificial Neural Networks IWANN 2005, Barcelona, Spain, pp. 837 – 844, 2005.
[6] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, New Jersey, 1978.
[7] J. Benesty, M. M. Sondhi and Y. Huang, Springer Handbook of Speech Processing, Springer-Verlag, Berlin, 2008.
[8] L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, New Jersey, 1993.
[9] A. Kocsor, L. Tóth, A. Kuba Jr., K. Kovács, M. Jelasity, T. Gyimóthy and J. Csirik, "A comparative study of several feature space transformation and learning methods for phoneme classification", International Journal of Speech Technology, vol. 3, no. 3/4, pp. 263 – 276, 2000.
[10] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, Wiley & Sons, New York, 1973.
[11] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[12] X. Huang, A. Acero and H.-W. Hon, Spoken Language Processing, Prentice Hall, New Jersey, 2001.
[13] S. Young, "The HMM Toolkit (HTK) (software and manual)", http://htk.eng.cam.ac.uk/, 1995.
[14] F. J. Damerau, Markov Models and Linguistic Theory: An Experimental Study of a Model for English, Mouton De Gruyter, Berlin, 1971.
[15] F. Jelinek, Statistical Methods for Speech Recognition, The MIT Press, Cambridge, USA, 1997.
[16] G. Gosztolya, D. Paczolay and L. Tóth, "Automatic Gain Control algorithms for wireless sensors", Proceedings of the International Joint Conference on Computational Cybernetics and Technical Informatics ICCC-CONTI 2010, Timisoara, Romania, pp. 401 – 406, 2010.
[17] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition", Proceedings of the International Conference on Acoustic, Speech, and Signal Processing ICASSP 1995, Detroit, USA, pp. 153 – 156, 1995.
[18] M. Holmberg and D. Gelbart, "Automatic Speech Recognition with an adaptation model motivated by auditory processing", IEEE Transactions on Audio, Speech and Language Processing, vol. 14, no. 1, pp. 43 – 49, 2005.

Manuscript received September 8, 2010; revised December 4, 2010; accepted for publication April 7, 2011.