

Automatic Gain Control Algorithms for Wireless Sensors

Gábor Gosztolya
University of Szeged, Hungary
Department of Informatics
Email: ggabor@inf.u-szeged.hu

Dénes Paczolay
University of Szeged, Hungary
Department of Informatics
Email: pdenes@inf.u-szeged.hu

László Tóth
Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences
Szeged, Hungary
Email: tothl@inf.u-szeged.hu

Abstract—Wireless sensors are small-capacity devices with low consumption. Their capabilities already exceed the limit required for telephone-quality audio recording and processing, which calls for porting a number of speech processing applications. To do it, however, dynamically adjusting the sensitivity of their microphone is vital, which is called Automatic Gain Control (AGC). In this study we employed two simple algorithms for this task and will show that they can indeed be very effective in distant recording conditions, especially as it requires techniques which have low computational requirements.

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 standard 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 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.

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

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 observations. It is used in various situations where signals of different volumes (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 them. Perhaps the most important one is that, as these sensors were designed to have 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 only 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*; of course, our algorithms also followed this approach. In this paper we sought to introduce actual algorithms for our particular problem, so they were designed with the same hardware limitations in mind, thus they are quite similar. Due to this, we begin with their common properties. In the following we will denote the actual gain value by *gain*, while *gain'* will be the new gain value.

A. Working in Packets

One characteristic of wireless sensors is that they communicate via radio waves, sending a small chunk of data called *packet* at a time. In our case there can be at most 114 bytes in a packet; after assigning some bytes to auxiliary bits of information, we can send 88 10-bit precise samples. Of course the actual values may vary between different hardware, but the main concept (having small-sized packets which implies 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. $A = a_1 \dots a_N$, 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 n th packet, but the same lines of code can still be executed for each packet, because we do not refer to samples of another packet. (Although using some value representing another packet as a whole (like its energy level) is common.)

B. Relying on the Energy Level

Further, both algorithms 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 the A packet, used both for voice activity detection and by the gain control

Algorithm 1 General Gain Control Algorithm

```

totalenergy is the total energy of the last 50 packets
if totalenergy/(gain + c0) < TSIL then
    gain is set to an intermediate value gainSIL
else if totalenergy > THIGH then
    gain is reduced to its 3/4th
else if totalenergy > THIGH then
    decrease gain
else if totalenergy < TLOW then
    increase gain
end if

```

algorithms. As 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 total energy of this interval, i.e.

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

Gain adjustments (including those of voice activity detection) were made every 10 packets, i.e. in 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 resulting 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 close or far away), and then we can react to the current volume level quite quickly. In the actual solution we divided the total energy of the last 50 packets by the gain value and a small constant; if the result was below some T_{SIL} , we set the gain to an intermediate value.

D. The Common 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 in 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 at a T_{IDEAL} level; for this, if the total 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 80% of T_{IDEAL} , while T_{HIGH} was 120% of it. We used another threshold (T_{UHIGH} , being twice that of T_{HIGH}); if the total energy level exceeded this value, we supposed that the signal was clipped because it was excessive, which called for an immediate and drastic reaction. At such

times the gain was decreased by using the formula $gain' = \frac{3}{4} \cdot gain$ based on preliminary tests (see Algorithm 1).

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 first 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

$$\frac{totalenergy_{ideal}}{gain_{ideal}} = \frac{totalenergy_t}{gain}, \quad (4)$$

where $totalenergy_{ideal}$ is the total energy level recorded under ideal conditions, determined by preliminary tests, $gain_{ideal}$ is the corresponding gain, $totalenergy_t$ is the total energy level recorded, and $gain$ is the actual gain value. This formula can be rearranged to express the expected gain level as

$$gain_{ideal} = \frac{totalenergy_{ideal} \cdot gain}{totalenergy_t}. \quad (5)$$

Using this formula, 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

$$gain' = w \cdot gain_{ideal} + (1 - w) \cdot gain, \quad (6)$$

where $0 \leq w \leq 1$ is a weighting factor. After substituting $gain_{ideal}$ from Eq. (5) we get

$$gain' = \frac{gain(w \cdot totalenergy_{ideal} + (1 - w) \cdot totalenergy_t)}{totalenergy_t}. \quad (7)$$

With the w weighting factor we can make the transition of gain much smoother, eliminating the sudden jumps; of course the optimal value for w 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.

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

TABLE I
THE RECORDINGS MADE AND THE DISTANCES USED.

A. Hardware

In this study we used Crossbow Iris sensor nodes (*motest*) that have a 2.4 GHz processor with a RAM of 8K bytes and a programmable flash memory of 128K bytes. The microphone and other input peripherals are located on a piece of hardware that can be attached to it, 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 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.

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, and this value cannot be exceeded no matter what sophisticated gain control algorithms we apply. 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 I.

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 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.

$$energy_S = \sum_{i=1}^T s(i)^2, \quad (8)$$

where s is the observed speech signal having a length of T . In this form it has one value for the whole signal, which could indeed be of interest; for this reason we will calculate

$$energyratio_S = \frac{energy_S}{energy_{ref}}, \quad (9)$$

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 problem 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 we calculated a squared error-like value by taking the squared sum of the difference between the energy levels of the corresponding windows:

$$diff_{A,B} = \sum_{i=1}^K (energy_{A(i)} - energy_{B(i)})^2, \quad (10)$$

where $energy_{A(i)}$ is the energy level of signal A in the i th 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 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 this much of the error was eliminated.

D. Evaluation via Phoneme Classification

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 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 distinguish between the two cases.

To measure this “understandability” we turned to standard techniques of speech recognition, following the frame-based approach [4]: we divided the speech signal into small, equal-sized parts, which – after feature extraction – were classified

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

TABLE II
TOTAL ENERGY RATIOS OF THE TWO GAIN CONTROL ALGORITHMS (ES AND WA).

as one of the possible phonemes. Usually we check whether the result of this phoneme-identification matches the correct label of the corresponding frame, performed for each frame of the utterance. To do it, however, the correct labeling has to be known in advance, which requires great manual effort. As we wanted to know how the result of a gain control algorithm differs from the one *made under ideal conditions*, we used the reference recording: the result of phoneme classification on this signal was treated as the correct labeling of frames.

We applied the standard 13 MFCC coefficients along with their derivatives and the second derivatives ($MFCC + \Delta + \Delta\Delta$ for short) [5] as features for phoneme classification, and applied Artificial Neural Networks (ANN) for it [6]. As usually several hours of hand-labeled and hand-segmented recordings are required for training, we used another database for this task [7]. This database was recorded over the phone, and not via the wireless sensors, but the recording conditions were very similar: it had a sampling frequency very close to 8861 Hz (8000 Hz) and about the same level of basic noise. The features were calculated by using the HTK toolkit [8].

We again calculated the relative error reduction scores. We did 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 calculated both ratios. The (original) error value is the difference between the accuracies 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 accuracies of the actual and the baseline recordings, and divided it by the error value.

V. RESULTS

First we had to set the parameter of both algorithms to the optimal value: for this we determined an interval of well-performing values 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 $\frac{1}{32}$ and $\frac{10}{32}$ with a step size of $\frac{1}{32}$. We found that the best values were $step = 3$ and $w = \frac{5}{32}$.

A. Results Using the Energy Level

First the total energy ratio was calculated for each recording (see Table II). As can be seen, the distance clearly affects the energy ratios when we do not use a gain control: the

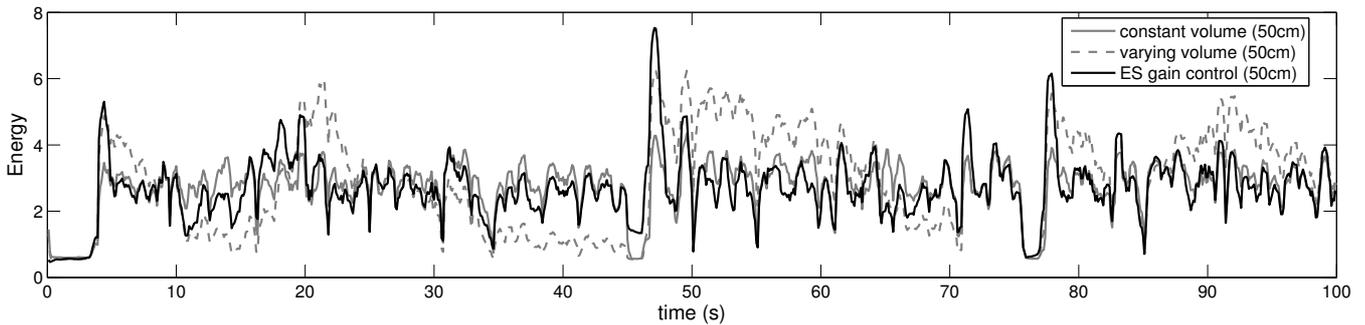


Fig. 1. Energy levels of the reference recording (grey continuous line), the baseline recording (grey rugged line) and the ES algorithm with $step = 3$.

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 III

ENERGY DIFFERENCES OF THE TWO BASIC RECORDING TYPES AND OF THE TWO GAIN CONTROL ALGORITHMS (ES AND WA) FROM THE REFERENCE RECORDING.

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%

TABLE IV

ENERGY DIFFERENCE RELATIVE ERROR REDUCTIONS SCORES OF THE TWO BASIC RECORDING TYPES AND OF THE TWO GAIN CONTROL ALGORITHMS (ES AND WA).

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.

The energy level difference $diff$ values, calculated according to Eq. (10), can be seen in Table III, while the relative error reduction scores are shown in Table IV. 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 that 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 that for the basic one at 200cm, leading to the negative RER score for the latter. 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.

Visually inspecting the energy levels at 50cm using the ES algorithm with $step = 3$ (see Figure 1), we may say that the algorithm seems effective. (The WA method produced a very

similar graph with $w = \frac{5}{32}$.) 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 of the sensor and the sound source strongly affects the energy levels when there is no gain control: the four corresponding curves are quite far 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 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 based on the total energy ratios.

B. Results in Phoneme Classification

The best phoneme classification results of the methods can be seen in Table V. It may seem surprising that even the recording made in the very same circumstances (i.e. the basic recording at 50cm) only attained a score of 83.19%. It is, however, due to the indeterministic nature of the recording process: two recordings made in the same way will not be exactly the same, just very similar. This effect, combined with the fact that the frames are very small-sized (1/40th of a second) and were independently classified, led to this 83.19% score, and this value then served as the glass ceiling.

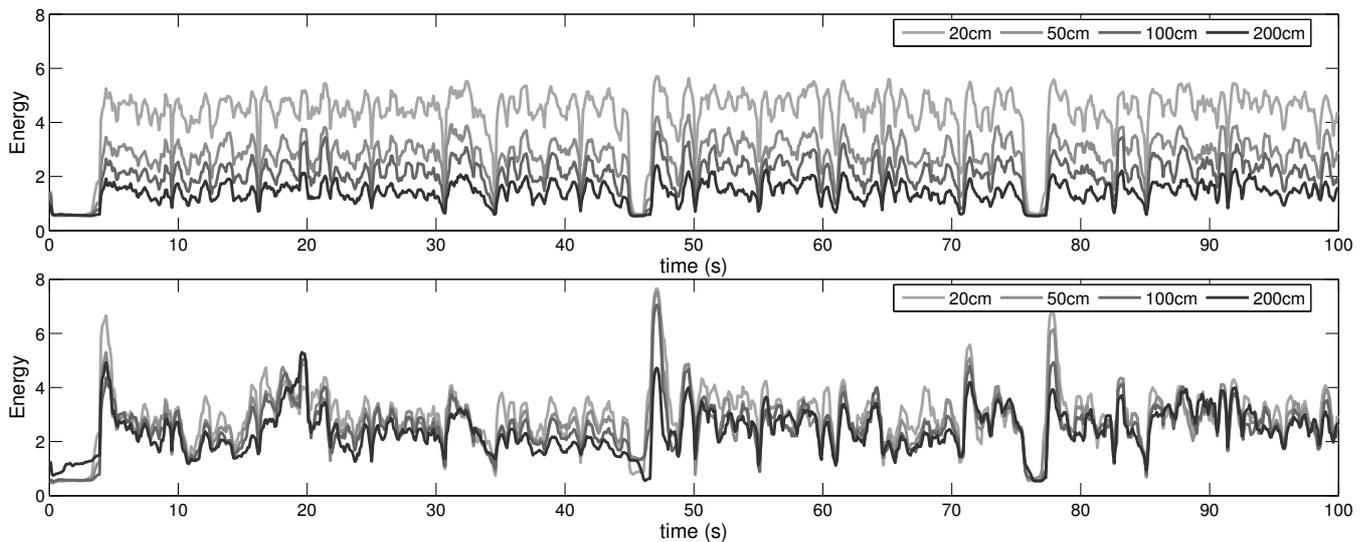


Fig. 2. Energy levels of the basic recording (up), and the signal with varying volume using the ES algorithm with $step = 3$ (down) at different distances.

Recording	Distance			
	20cm	50cm	100cm	200cm
Basic	68.88%	83.19%	64.71%	53.75%
Baseline	62.32%	72.02%	61.84%	53.03%
ES AGC, $step = 3$	69.40%	76.65%	64.39%	58.41%
RER (same distance)	107.93%	41.45%	88.85%	747.22%
RER (glass ceiling)	33.97%	41.45%	11.94%	17.84%
WA AGC, $w = 5/32$	68.23%	74.54%	64.73%	57.77%
RER (same distance)	90.09%	22.56%	100.70%	658.33%
RER (glass ceiling)	28.32%	22.56%	13.54%	15.72%

TABLE V

BEST PHONEME CLASSIFICATION RESULTS OF THE TWO GAIN CONTROL ALGORITHMS (ES AND WA).

We may also conclude that when the conditions become less ideal by going farther or closer from the optimal 50cm distance, the difference between the corresponding basic and baseline recordings decreases: from the 11.17% at 50cm it fell back to the hardly noticeable 0.72% at 200cm, where probably neither of them could be heard adequately. In a normal situation (i.e. at 50cm) both gain control algorithms were able to improve the recognition accuracy to a fair extent: the ES algorithm achieved a 76.65% accuracy, meaning a 41.45% relative error reduction, whereas the WA algorithm produced scores of 74.45% and 22.56%, respectively. In the less ideal cases, however, these performed much better: although they produced scores that fell much farther than the glass ceiling value, they matched or even outperformed the basic recording score. This is probably because at 20cm the general volume was just too high, requiring a lower gain value, which was only possible by using a gain control algorithm. In the other two cases the greater distance called for higher sensitivity, which could also be done easily via gain control. This led to a performance that was practically identical to the basic recording at 20 and 100 cm (meaning 107.93% and 88.85% same-distance RER scores for the ES algorithm, and 90.09%

and 100.70% for the WA method), and a much better value at 200cm. In addition, the RER scores compared to the glass ceiling value are also good.

VI. CONCLUSIONS

Due to the unknown positioning of wireless sensors, their distance from the speaker or speakers represents a serious problem during sound recording as the optimal recording volume level cannot be known beforehand. One good solution is to apply an Automatic Gain Control (AGC) algorithm. In this study we applied two simple solutions, keeping the limited capacity of sensors in mind, and tested their behaviour at different distances. Evaluating both via energy level comparisons and phoneme classification from the field of speech recognition, we showed that their application is indeed worthwhile: we could achieve significant improvements in the quality of the recordings, which could be improved still further when the recording distance changed from the ideal one.

REFERENCES

- [1] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," *Mobile Networks and Applications*, vol. 1, no. 2, 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," in *Proceedings of the 2006 SICE-ICASE International Joint Conference*, Busan, South Korea, 2006, pp. 1085–1090.
- [3] C. S. Bae, "An automatic pacemaker sensing algorithm using automatic gain control," in *Proceedings of the TENCON 99 IEEE Region 10 Conference*, Cheju Island, South Korea, 1999, pp. 375–378.
- [4] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [5] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*. Prentice Hall, 2001.
- [6] C. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.
- [7] K. Vicsi, L. Tóth, A. Kocsor, and J. Csirik, "MTBA – a Hungarian telephone speech database (in Hungarian)," *Híradástechnika*, vol. LVII, no. 8, 2002.
- [8] S. Young, *The HMM Toolkit (HTK) (software and manual)*, <http://htk.eng.cam.ac.uk/>, 1995.