Single–sided Real–time PESQ Score Estimation*[†]

Sebastián Basterrech, Gerardo Rubino
INRIA/Rennes – Bretagne Atlantique,
Rennes, France

Martín Varela
Converging Networks Laboratory
VTT Technical Research Centre of Finland
Oulu, Finland

Abstract

For several years now, the ITU-T's Perceptual Evaluation of Speech Quality (PESQ [1]) has been the reference for objective speech quality assessment. It is widely deployed in commercial QoE measurement products, and it has been well studied in the literature. While PESQ does provide reasonably good correlation with subjective scores for VoIP applications, the algorithm itself is not usable in a real–time context, since it requires a reference signal, which is usually not available in normal conditions. In this paper we provide an alternative technique for estimating PESQ scores in a single–sided fashion, based on the PSQA technique [2].

1 Introduction

For several years now, the ITU-T's Perceptual Evaluation of Speech Quality (PESQ [1]) has been the reference for objective speech quality assessment. It is widely deployed in commercial QoE measurement products, and it has been well studied in the literature.

In previous work [3], we have studied the performance of PESQ for VoIP over a wide range of network conditions, and found that

1. the correlation with subjective scores was good, even for cases in which losses were relatively abundant and bursty (but still within reasonable limits, see [4, 5] for some limitations of PESQ with respect to network impairments), and
2. PESQ scores were fairly consistent for all combinations of speech samples and loss patterns.

These results lead to thinking that a good approximation of PESQ can be achieved at the receiving terminal as long as network performance can be measured and some application–level knowledge (such as the codec in use, the presence or absence of loss concealment, etc.) is available. In the past we have advocated the use of Pseudo–Subjective Quality Assessment (PSQA [6]) for VoIP QoE estimations, which allows for very accurate estimations of MOS values based on network

*Partially supported by the EuroNF "QUAXIME" project.

[†]This work was partially supported by Celtic Easy Wireless 2 project, funded in Finland by Tekes (Finnish Funding Agency for Technology and Innovation) and VTT

and application parameters. In this paper we analyze the applicability of the PSQA approach to the estimation of PESQ scores.

In principle, given that both PESQ and PSQA correlate very well with subjective perception, it is expected that the approach presented herein will lead to a hybrid approach offering the best of both worlds. On the one hand, using PESQ as a target function eliminates the costly part of PSQA, namely the need to perform a non–trivial subjective LQ assessment campaign. On the other hand, it allows to have these results in real–time, without the need for a reference signal. This enables the use of these PESQ-like results in situations in which some quick reaction is desirable, for instance in order to improve the perceived quality by means of real–time controlling actions on the communication system whose delivered QoE is automatically assessed, one of the main goals behind our research efforts.

The paper is organized as follows. In Section 2 we describe the experiments realized and their motivation. Section 3 presents the results obtained. We conclude the paper in Section 4.

2 Methodology

2.1 Motivation

Our previous work on PSQA is based on a rather simple concept, to wit: the quality of a media stream (be it voice or video), as perceived by an average user, and assuming no extraneous, non-measurable degradations at the source (such as faulty equipment) is usually determined by a number of factors that can be divided in two categories. These categories are

Application–related factors, such as the encoding used, the type of error correction and concealment chosen, play-out buffer sizes, etc.

Network–related factors, such as the loss rate in the network, delay, jitter, etc.

These premises, coupled with the fact that PSQA provides very good correlation with subjective scores, imply a certain independence of the perceived quality from the actual media streamed (there are of course some limitations to this claim, especially concerning video, mostly related to scene types and amount of motion, but those can be measured and hence considered as an application–level factor).

In turn, the previous observation leads to the prediction that for VoIP, the scores given by reference–based tools such as PESQ should be quite consistent for a given configuration of application and network factors or parameters. This was the subject of our work in [3]. The results from that study show that PESQ scores taken for a single encoding and over consistent network conditions are remarkably stable. So much so, that for a given configuration of parameters (in the case of the previous study, a given codec, whether PLC was in use and the loss rate and loss distribution in the network) a fairly good prediction of the PESQ-LQ values could be given by taking the median of a series of PESQ-LQ assessments taken over similar configurations. For *reasonable* network conditions (i.e. conditions that do not degrade the VoIP stream's quality badly enough to break PESQ's assessment), the median–based estimations are very close to actual PESQ scores.

Using this approach in practice, however, has some limitations. Firstly, it requires a rather large number of assessments to be performed in order to acquire enough information to reliably cover the parameter space. This, in itself is not a serious issue if the parameters considered are not too numerous, but it is an area that could be improved. The second issue is more important, since it may actually limit the applicability of the approach. This issue is the lack of generalization and hence the inability to extrapolate for parameter values not present in the original measurements. While this could be palliated by a brute–force approach (i.e. cover a larger parameter space, in a more fine–grained fashion if needed), this is not an elegant solution, and it basically doesn't solve the issue, but only masks it.

PSQA, on the other hand, relies upon the ability of the Neural Network (NN) it uses as a learning tool in order to reduce the number of samples required to reliably cover the whole parameter space. This is important since PSQA is usually trained with subjective scores, which are expensive and time–consuming to obtain. The NN's ability to generalize, coupled with PESQ's regularity over similar application and network configurations hint at the feasibility of obtaining a flexible, cheap and accurate way of single–sidedly estimating PESQ scores by using PSQA.

2.2 Experimental Setup

The experimental setup used for this study is very similar to the one used for [3]. We used G.711 encoding, with and without loss concealment, and considered the loss rate and distribution in the network as our network parameters. While jitter is a relevant parameter for LQ, it can be folded into the loss rate if no particular attention is being payed to the dejittering buffer sizes and algorithms. Hence, it is not considered explicitly in this study.

The network loss model used is a simplified Gilbert model [7] in which the lossy–state loss probability is 1 (i.e. all packets are lost in the lossy state). This model has the advantage of eliminating one free variable, and it provides a reasonably good model of losses on the Internet.

The network impairments are thus represented not only by the packet loss rate (LR), but also by the dispersion of losses in the stream, captured by the mean loss burst size (MLBS) [2]. The MLBS is the expected number of consecutive losses in a loss episode, that is, the mean length of loss bursts in the flow, a real number ≥ 1. We considered loss rates between 1% and 30%, more specifically values 1%, 2%, 3%, ..., 30%, and mean loss burst sizes of up to 6 consecutive packets (values 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6). Given that standard–length (approximately 10s) samples were used, it was not possible to have all possible combinations of LR and MLBS, since some of them are not really feasible within the $\sim$ 400 packets that each speech sample uses when transmitted. Thus, only valid combinations were considered, and for those, each loss trace created was verified to ensure that it had the desired characteristics.

It should also be noted that PESQ is not expected to behave correctly with respect to subjective scores when the network impairments are too high. In any case, since the goal of the study was to mimic PESQ's performance, we anyway considered very impaired networks.

For each combination of values of the two loss-related parameters LR and MLBS, 10 different traces (all statistically similar) and 20 standard speech samples (50% male and 50% female) were used[1]. The number of samples generated and then evaluated with PESQ was slightly above 128500. For each combination of LR, MLBS and packet loss concealment (PLC, either active, coded PLC = 1, or not, coded PLC = 0) several sequences were analyzed (around 200 of them, except in some cases with high loss rates, where more samples were generated and used). In other words, we sent each one of the error-free voice sequences through a simulated/emulated network varying the three considered variables, and we used PESQ to evaluate the resulting quality. Since with every considered triple of values for LR, MLBS, PLC (we call a *configuration* such a triple [6]) we had many different associated PESQ values, we generated a second smaller table having around 600 entries, each corresponding to a different configuration of our platform. Again, in this table, each considered configuration (a loss rate, a value for the mean loss burst size, and the indicator of packet concealment active or inactive), there is one row in this new table.

For each of the entries (configurations) of the compact table, we evaluated statistical descriptors of the set of PESQ values associated with, such as the empirical mean, median, variance, etc. As in [3], the median was a good approximation of PESQ scores. We therefore used it to train a Neural Network using the AMORE package for the R statistical analysis language. That is, we built a function f mapping each possible configuration into a quality value in the interval $[1, 5]$ (actually, given that the target function is PESQ, the interval will be $[1, 4.5]$), that approximates the median of the values obtained using PESQ. Function f is defined in the space $[1, 30] \times [1, 6] \times \{0, 1\}$, corresponding to LR in %, MLBS and PLC. This function f is our approximation tool for PESQ, whose behavior is analyzed in next Section.

3 Results

The learning phase consisted of using a standard Neural Network (NN) for learning the mapping from configurations to (median) PESQ values. This was also partly done in the context of a larger study comparing the performance of the AMORE–based NNs against the Random Neural Networks (RNN) we have used previously. This comparison work is still ongoing at the time of writing. Some preliminary results were published in [8]; for the tool itself and its use in the PSQA approach, see [9]. Any of the numerous good references on Neural Network methodology can provide background material to the reader if this is necessary; for a classic one, see [10].

For training the NN, we randomly (and uniformly) separated the data in the small compacted table into two parts, corresponding to a 80%–20% decomposition for training and validation respectively. Since we have a binary variable PLC in the configurations, we actually built 2 NN, that is, two functions,

[1] For some configurations in the higher–end of the impairment values we actually used more samples, in order to mitigate the variability of PESQ's results.

f_0 corresponding to the case PLC= 0, and f_1 for the case of PLC= 1. This proved to be a better solution in this case than having a single NN with the PLC considered as a third input.

We used the usual 3-layer feed-forward perceptron structure with two inputs (LR and MLBS) and one output (estimated, or predicted PESQ value). For the hidden layer, we varied the number of neurons starting from 1, in order to select the best architecture for our neural networks. We finally chose an architecture with 30 hidden neurons for both f_0 and f_1. As stated before, the whole data set for learning (coming from the small table) has around 600 entries, half corresponding to the case PLC= 0 and half for the case PLC= 1.

Let us denote by $\mathcal{TS}, i$ the set of configurations corresponding to the 80% used for training the f_i network, the *Training Set* for the case PLC $= i$, with cardinality $K_{\mathcal{TS},i}$, and by $\mathcal{VS}, i$ the similar set of configurations corresponding to the 20% used for validation (the *Validation Set* when PLC $= i$), with cardinality $K_{\mathcal{VS},i}$. The *Training Error* when PLC $= i$, $i = 0$ or 1, is then

$$(K_{\mathcal{TS},i})^{-1} \sum_{\text{all config.}\gamma\in\mathcal{TS},i} \big[f_i(\gamma) - \text{MedianOfPESQ}(\gamma)\big]^2,$$

and the *Validation Error* is

$$(K_{\mathcal{TV},i})^{-1} \sum_{\text{all config.}\gamma\in\mathcal{VS},i} \big[f_i(\gamma) - \text{MedianOfPESQ}(\gamma)\big]^2.$$

In both expressions, we call configuration (denoted by γ) just the pair (LR,MLBS), since we separated the data into two sets thus eliminating the need for a third variable PLC. For each such γ, MedianOfPESQ(γ) is the value obtained from the analysis of the original table having fixed PLC to 0 or to 1, according to the case we are analyzing, for instance, the number defined by

$$\text{argmin}_x K^{-1} \sum_{\text{all config. } \gamma} \big|\text{PESQ}(\gamma) - x\big|$$

if K is the size of the small table (around 600 in our experiments). Table 1 provides some data for this step of the analysis. Given the fact that we are using PESQ values in the range [1,5], the reached error levels are indeed extremely small.

Table 1: Performances of the learning phase, for the two selected Neural Networks f_0 and f_1

neural network	training error	validation error
f_0	0.064	0.069
f_1	0.040	0.042

Figure 1 shows, on the left, PESQ values and on the right, the predictions provided by the Neural Network, everything for PLC = 0 (no Packet Loss Concealment). In the x-axis we put LR values. Each point in the graphs corresponds to a configuration (LR,MLBS,0). Different points on the same vertical line, that is, with same LR, correspond to configuration with same LR but varying MLBS. It is interesting to see that the PESQ plot shows a significant amount of dispersion compared to the estimation when the loss rate goes over 10 to 15%. This is due to the NN being trained with median values, which significantly suppress the impact of outlier values in the data set. It is also known that PESQ tends to behave in a more variable way when the network impairments become large, and this behavior is exacerbated in this case by the lack of PLC on the decoder

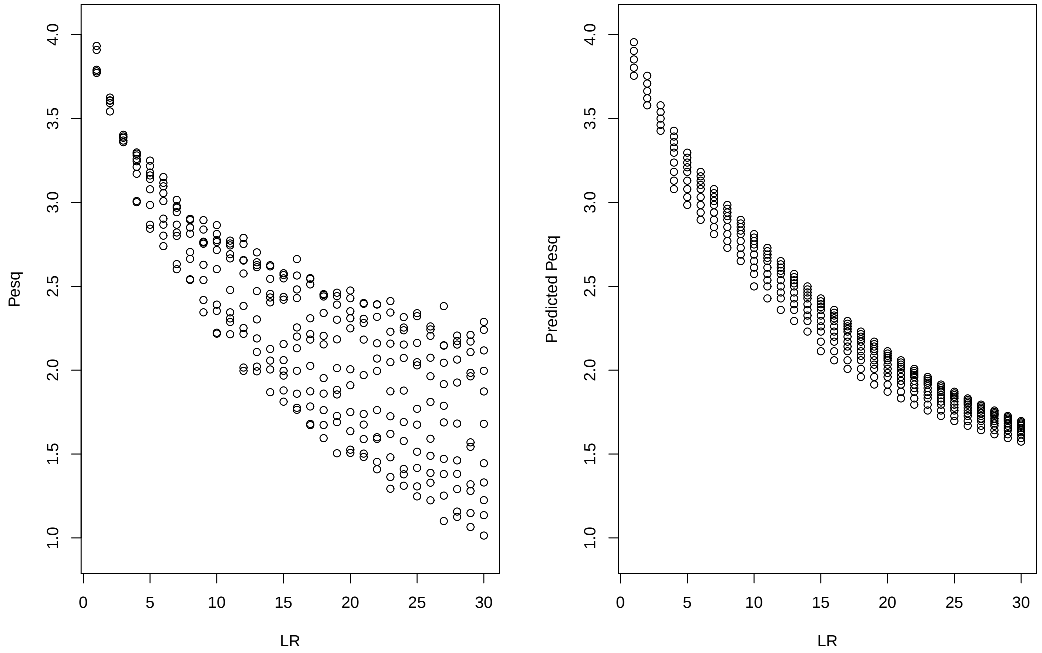

Figure 1: Case of PLC= 0. PESQ and its predictor f_0, as an explicit function of LR. Each spot corresponds to a specific configuration in the small table. Different spots for a same LR correspond to different values for MLBS.

end.

Figure 2 provides an analogous view, plotting PESQ and its estimation as a function of MLBS. It can be noticed in this plot that the estimated values are not as expected for burst losses higher than two or three packets, in which case the estimations are overly optimistic with respect to actual PESQ values. We do not, at the time, have a definitive explanation for this phenomenon. However, given the good correlation for PSQA and subjective scores obtained in previous study, we suspect that the variability of PESQ results with respect to MLBS might have precluded the NN from capturing the correct behavior.

Figures 3 and 4 illustrate the case of PLC = 1. As expected, the overall values in this case are higher (by about 0.5 MOS points) than in the non-PLC scenario. Otherwise, the overall behavior of PESQ and the NN–based estimations are comparable to the non–PLC case, but with smaller errors.

Consider again the original set of data (the large table), over 10^5 voice samples, with the corresponding values of loss rate, mean loss burst size and PLC, together with the quality assessment made by PESQ. If we use our functions f_i for approximating the PESQ scores for all of the data points, what would be the mean error? Observe that this is quite close of a field application of our approach, even if this table of values is the original data with which the training data sets were built. Denote

- by s a generic entry in the original table (a sample); there are more than 10^5 such samples;
- by $\text{PLC}(s)$ the value of PLC in sample s;
- by $f_{\text{PLC}(s)}(s)$ the value predicted by the right NN when the configuration is the one in sample s;
- finally, let $\text{PESQ}(s)$ be the PESQ assessment of sample s.

Table 2 shows the Mean Square Error (MSE_i), its square root and the Mean Absolute Error (MAE_i), corresponding to function f_i, defined as follows:

$$\text{MSE}_i = \frac{1}{N_i} \sum_{s:\text{PLC}(s)=i} \left[f_{\text{PLC}(s)}(s) - \text{PESQ}(s) \right]^2,$$

$$\text{MAE}_i = \frac{1}{N_i} \sum_{s:\text{PLC}(s)=i} \left| f_{\text{PLC}(s)}(s) - \text{PESQ}(s) \right|.$$

Table 2: Performances of the two selected Neural Networks f_0 and f_1

neural network	MSE	$\sqrt{\text{MSE}}$	MAE
f_0	0.236	0.486	0.412
f_1	0.076	0.276	0.221

This implies that the NN–based estimations are on average, at about 0.41 points from actual PESQ scores for samples in which PLC was not used, and at about 0.22 points for samples in which it was enabled. Given the average listener's appreciation in terms of the MOS scale, it seems that the estimations are indeed very close to the actual values. This closeness can be seen in Figure 5, which shows, for a loss rate of 12% all the PESQ scores in the complete

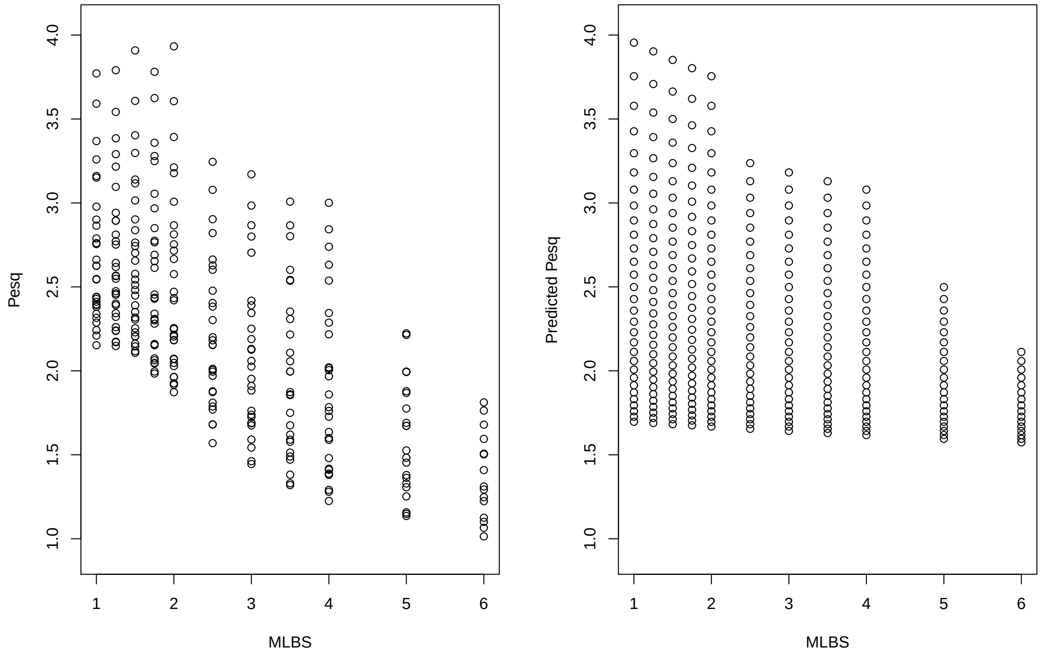

Figure 2: Case of PLC= 0. PESQ and its predictor f_0, as an explicit function of MLBS. Each spot corresponds to a specific configuration in the small table. Different spots for a same MLBS correspond to different values for LR.

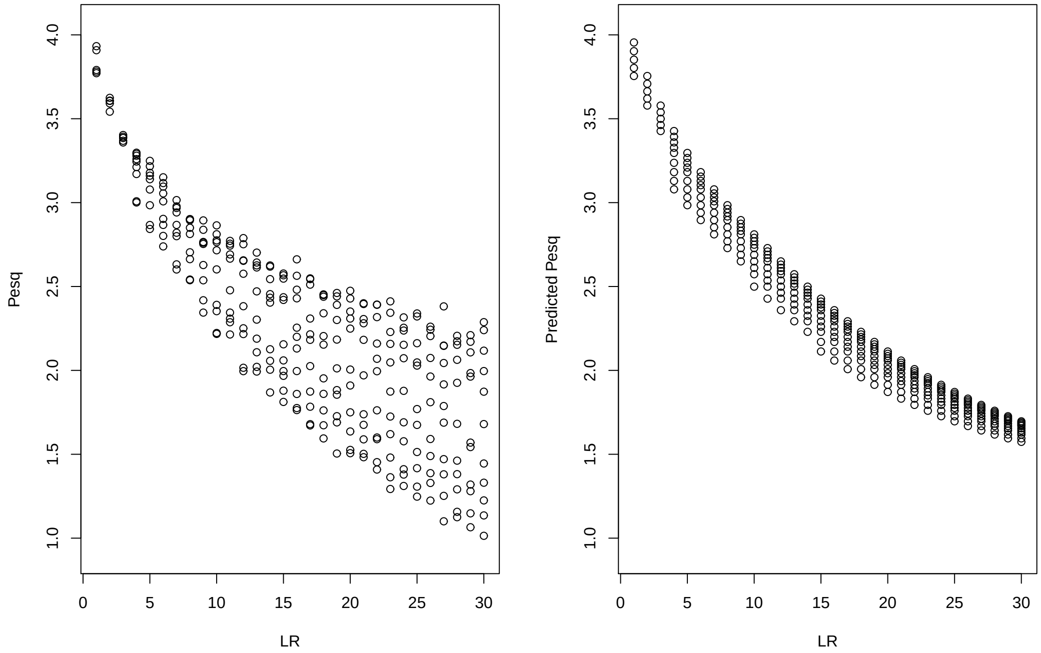

Figure 3: Case of PLC= 1. PESQ and its predictor f_0, as an explicit function of LR. Each spot corresponds to a specific configuration in the small table. Different spots for a same LR correspond to different values for MLBS.

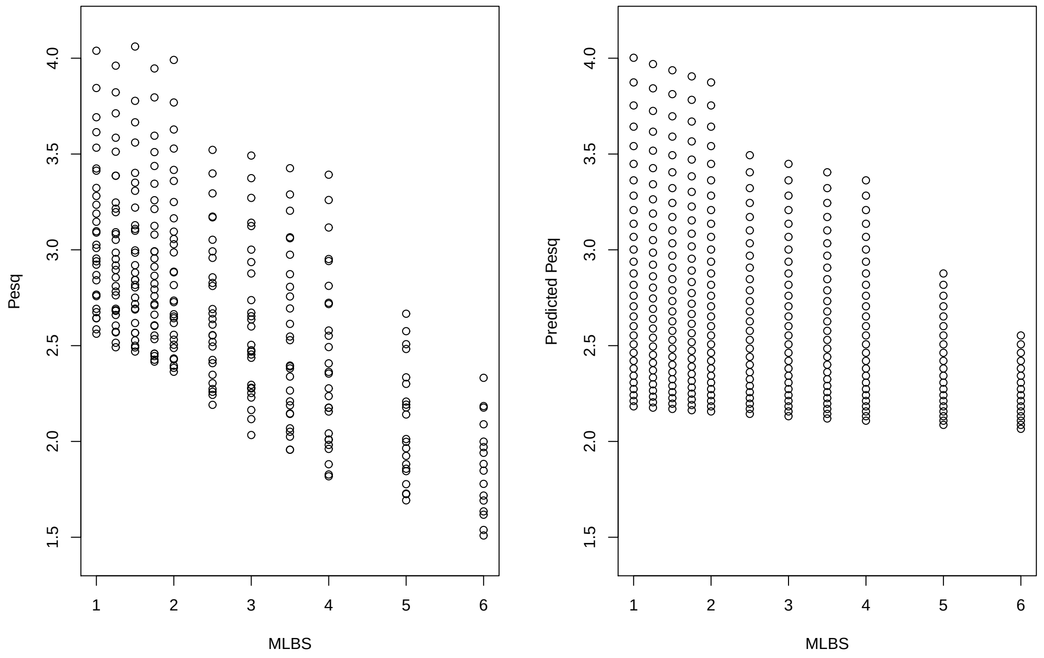

Figure 4: Case of PLC= 1. PESQ and its predictor f_0, as an explicit function of MLBS. Each spot corresponds to a specific configuration in the small table. Different spots for a same MLBS correspond to different values for LR.

data set, separated by MLBS value, and the NN's estimation of them.

4 Conclusions

In this paper we have presented a simple, efficient way of providing single–sided, reference–free estimations of PESQ scores for VoIP samples or ongoing streams. The method used was PSQA (Pseudo–Subjective Quality Assessment), but using PESQ as a target function instead of subjective scores, as was done previously.

While this will evidently not increase the correlation of PSQA with respect to subjective scores, it provides a very cheap and efficient way of having a single–sided quality assessment tool. Moreover, the evaluation by NNs is very computationally efficient, which allows this mechanism to be used in real–time, for control purposes, for example, even in resource–constrained devices such as mobile phones or Internet tablets (unlike, say, the ITU's P.563 [11] single sided metric, which is very resource–intensive).

The reliability of the results obtained is slightly variable with network conditions, as depicted in Figures 1 through 4. However, it should be noted that firstly, for *normal* operating conditions, in which network impairments are not too high, the estimations are remarkably close to actual PESQ scores. Secondly, since PESQ itself shows reliability issues in cases where the network is severely impaired, a different approach should be tried in these scenarios, as needed.

In future work on this subject, we plan on determining the impairment bopundaries in which using this sort of approach works well in practice, and using it to implement some sort of QoE control mechanism (either application or network–based). It would be also interesting to use different kinds of neural networks (for example in a recurrent architecture, instead of feed–forward) and also to re–use the data obtained in this work to train a Random Neural Network (RNN, cf [12]), which we have previously used with success for PSQA applications.

References

[1] ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (Pesq), an Objective Method for End-To-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, 2001.

[2] G. Rubino, M. Varela, and S. Mohamed. Performance evaluation of real-time speech through a packet network: a random neural networks-based approach. *Performance Evaluation*, 57(2):141–162, May 2004.

[3] Martín Varela, Ian Marsh, and Björn Grönvall. A systematic study of PESQ's performance (from a networking perspective). In *Proceedings of the Measurement of Speech and Audio Quality in Networks workshop (MESAQIN'06)*, Prague, Czech Republic, June 2006.

[4] S. Pennock. Accuracy of the perceptual evaluation of speech quality (PESQ) algorithm. In *Measurement of Speech and Audio Quality in Networks Line Workshop, MESAQIN '02*, January 2002.

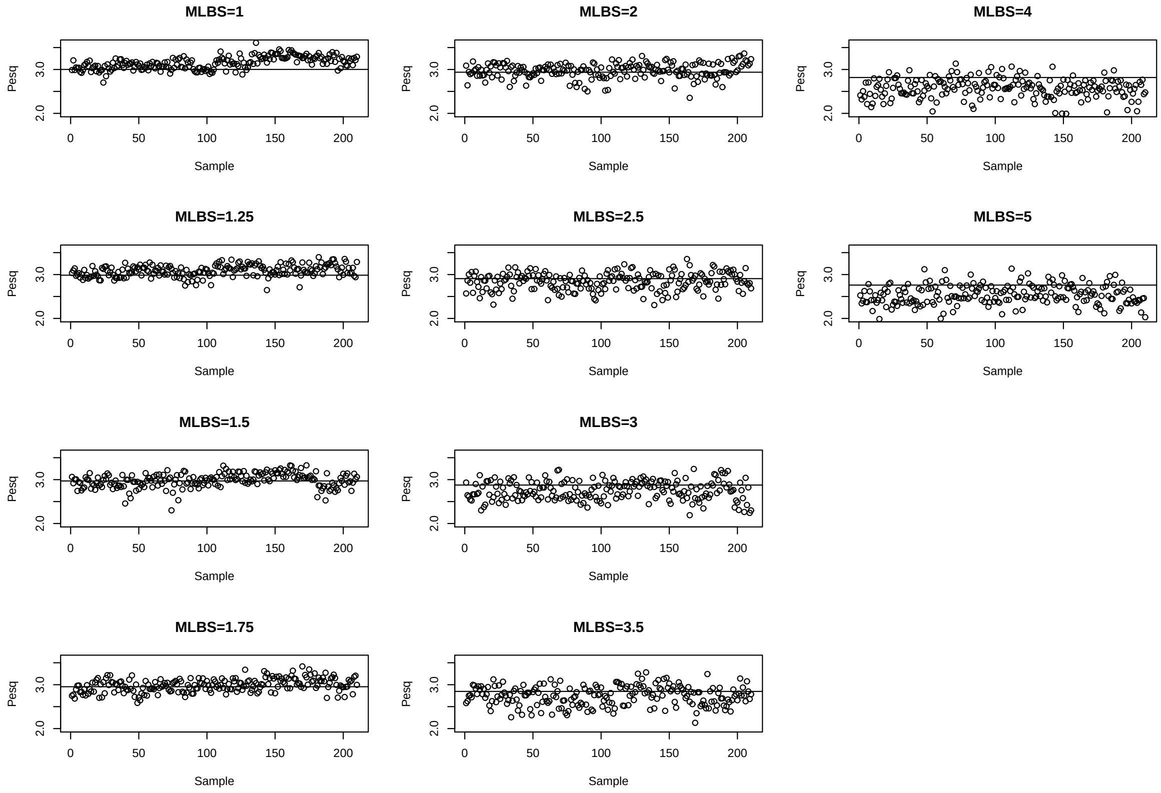

Figure 5: PESQ values for all points in the data set, where LR=12. Each area represents a separate MLBS value, and the horizontal lines represent the NN's estimation of the PESQ score. PLC=1.

[5] Psytechnics Ltd. PESQ: an Introduction. `http://www.psytechnics.com`, September 2001.

[6] Martín Varela. *Pseudo–subjective Quality Assessment of Multimedia Streams and its Applications in Control*. PhD thesis, INRIA/IRISA, univ. Rennes I, Rennes, France, November 2005.

[7] E. Gilbert. Capacity of a burst–loss channel. *Bell Systems Technical Journal*, 5(39), September 1960.

[8] G. Rubino, P. Tirilly, and M. Varela. Evaluating users' satisfaction in packet networks using Random Neural Networks. In *16th International Conference on Artificial Neural Networks (ICANN'06)*, Athens, Greece, September 2006.

[9] G. Rubino. Quantifying the Quality of Audio and Video Transmissions over the Internet: the PSQA Approach. In *Design and Operations of Communication Networks: A Review of Wired and Wireless Modelling and Management Challenges*, Edited by J. Barria. Imperial College Press, 2005.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] ITU-T Recommendation P.563. Single ended method for objective speech quality assessment in narrow-band telephony applications, 2004.

[12] E. Gelenbe. Random neural networks with negative and positive signals and product form solution. *Neural Computation*, 1(4):502–511, 1989.