Perception of Temporal Discontinuity Impairments in Coded Speech – A Proposal for Objective Estimators and Some Subjective Test Results

Stephen D. Voran

Institute for Telecommunication Sciences
325 Broadway
Boulder, Colorado 80305, USA
svoran@its.bldrdoc.gov 303-497-3839 Voice 303-497-5969 Fax

ABSTRACT

Temporal discontinuities in received speech are a reality of Internet Telephony or Voice over Internet Protocol (VoIP) systems. These relatively new impairments pose unique challenges to objective estimators of perceived speech quality. We suggest that objective estimators may benefit from the addition of a temporal discontinuity impairment processor and we provide subjective test results that may help with the design of such processors.

We added the loss, pause, and jump impairments (nine different levels of each) to random locations in active segments of G.723.1 coded speech. We then measured the resulting perceived speech quality via a formal absolute category rating subjective experiment using the mean opinion score (MOS) scale.

The results show that these three different impairments have similar influences on perceived speech quality, even though the pause and jump impairments are exact opposites (temporal dilation vs. temporal contraction). The results also demonstrate that at a fixed impairment rate, dispersion of these impairments is less detrimental to perceived speech quality than clustering of these impairments. We offer a simple mathematical model that relates impairment parameters to experimental MOS values. It is expected that these results will be of value to those who develop objective estimators of packetized speech quality as well as those who design jitter buffers and jitter buffer management (or playout) algorithms.

Keywords: speech perception, speech quality estimation, temporal discontinuity

1. INTRODUCTION

Temporal discontinuity impairments in received speech are an avoidable reality of Internet Telephony or Voice over Internet Protocol (VoIP) systems. The sources of these impairments and potential mitigating techniques are described in some detail in the next section. The appearance of these relatively new impairments has required the development of new techniques for the objective estimation of speech quality. In particular, assuming a single fixed delay value for a system that actually has a varying delay (and hence temporal discontinuities) generally yields unusable results.

Excellent progress has been made, and the most prominent example is the Perceptual Evaluation of Speech Quality (PESQ) algorithm [1]-[3]. The high-level approach used in PESQ and other algorithms is given in Figure 1. The delay estimates $\{d_i, d_{i+1}, d_{i+2} \ldots\}$ are used to allow the proper matching of like segments of the original and degraded signals. Also, when the delay estimates indicate the presence of a temporal discontinuity, the speech frames on either side of that discontinuity are examined for audible artifacts. It has been well-demonstrated that this approach is very powerful. We would suggest that the delay estimates might be used advantageously in an additional way. Our reasoning follows.

A system following Figure 1 largely removes temporal discontinuity impairments before they enter the perceptual and cognitive models. If a temporal discontinuity impairment occurs in the midst of a syllable of speech, it will very likely create some spectral distortion and thus will be *indirectly* measured by the perceptual and cognitive models. (However, the links between temporal discontinuities and the resulting spectral distortions are not well-defined.) If a temporal discontinuity impairment occurs between syllables, then it will not necessarily create any significant spectral distortion and may go completely undetected and unmeasured by the perceptual and cognitive models.

In one simple experiment, we used a recording $\mathbf{x}$, of the English language sentence "We like to see clear weather" spoken by a male. When this recording was compared with itself, the PESQ algorithm appropriately estimated the MOS at the upper limit, 4.5. In other words, PESQ($\mathbf{x}$,$\mathbf{x}$)=4.5. We then formed $\hat{\mathbf{x}}$ by removing the 430 ms segment of low-level sounds between the words "clear" and "weather (a "jump" impairment), and found PESQ($\mathbf{x}$, $\hat{\mathbf{x}}$)=4.5. That is, the quality estimate was unchanged by the addition of the 430 ms jump impairment. We consider this result to be inconsistent with the very audible and objectionable nature of the impairment.

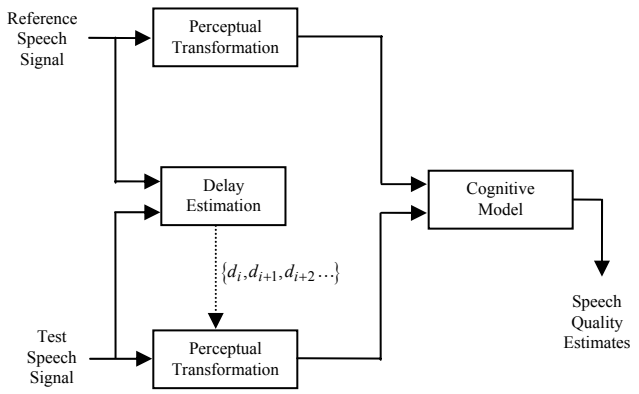

Figure 1. Block diagram for a speech quality estimator.

Since it is necessary to essentially remove temporal impairments in order to estimate quality, perhaps it would be beneficial to "add that effect back in" at or near the end of the estimation process. The idea is to use *direct* knowledge of temporal discontinuities that can be extracted from $\{d_i, d_{i+1}, d_{i+2} \ldots\}$ rather than rely on the *indirect* effects that temporal discontinuities may or may not have on speech spectra. Figure 2 describes an approach where the history of delay estimates $\{d_i, d_{i+1}, d_{i+2} \ldots\}$ is analyzed for temporal discontinuity impairments in a Temporal Discontinuity Impairment Detector and the results of this analysis are used to adjust a quality estimate in a Temporal Discontinuity Impairment Processor. Figure 3 proposes a more involved but potentially more realistic and effective approach where the output of the Temporal Discontinuity Impairment Detector feeds into the cognitive model for further processing in conjunction with the perceptual model outputs.

Such approaches clearly will require significant development efforts and will also add some implementation complexity. We would suggest however, that efforts of this sort may be required in order to reach the next level of refinement in objective speech quality estimation when temporal discontinuity impairments are present.

2. BACKGROUND: THE REAL-WORLD SOURCES OF TEMPORAL DISCONTINUITIES

The current enthusiasm for Internet Telephony or Voice over Internet Protocol (VoIP) has created renewed interest in the packetized transmission of digitally encoded speech [4]. Packetized transmission allows many speech streams to share a common network infrastructure. In most situations, this sharing of infrastructure leads to network delay variation (also called delay jitter) and makes it impossible to guarantee packet delivery on any set schedule [4]-[9]. However, during active speech segments, most speech encoders generate output at regular intervals, and most speech decoders work best when they receive input at regular intervals.

This fundamental mismatch between digital speech coding and packetized transmission is often partially bridged by a jitter buffer (also called playout buffer), typically located between the network output and the decoder input [4]-[6]. In typical use, a jitter buffer receives packets from a network as they become available (at irregular intervals), and provides them to a decoder for playout as they are needed (at regular intervals). If the variation in packet arrival times is too great or if packets are lost in the network, the jitter buffer may overflow or underflow or may simply not contain a specific packet when it is needed for playout. This will generally cause the decoder to create an impairment in the speech that it generates. When an impairment induces a discontinuity into the graph of end-to-end (microphone-to-speaker) delay versus time, we say this impairment includes a temporal discontinuity.

The selection of packet sizes, the sizing of jitter buffers, and the design of jitter buffer management algorithms (or playout algorithms) all involve fundamental trade-offs between delay and the impairments caused by buffer underflow and overflow. When larger buffers are used and larger numbers of received packets are buffered before playout, a larger delay is introduced, but the likelihood of buffer underflow or overflow and the accompanying impairments is decreased. When smaller buffers are used and smaller numbers of received packets are buffered before playout, a smaller delay is introduced, but the likelihood of buffer underflow or overflow and the accompanying impairments is increased.

End-to-end delay must be minimized to avoid inhibiting the natural flow of conversation, and to minimize the annoyance of any uncancelled echo signals. Encoding delay, packetization delay, network transmission delay, jitter buffer delay, and decoding delay all contribute to end-to-end delay. Thus there is often significant pressure to minimize jitter buffer delay, and hence to accept some impairments inherent in that trade-off.

The sizing of jitter buffers and the design of jitter buffer management algorithms are further complicated by the loss of packets in the network. (Note that network packet loss can be viewed as part of the delay variation problem. Lost packets are packets with infinite transmission delay, and there is no finite-sized buffer that can accommodate this situation unless retransmissions are allowed. This makes the buffer size vs. impairment trade-off very real indeed.) Examples of work on packet loss, packet loss mitigation, and packet loss concealment can be found in [4]-[7],[10]-[15].

One response to the challenges inherent in packetized speech transmission involves assigning speech packets to special classes of network traffic that receive preferential treatment such as expedited forwarding at network queues [16],[17]. The development and deployment of networks that support these Differentiated Services are still in process. While such networks may eventually provide some streams of speech packets with guaranteed bounds on packet delay jitter and packet loss, the need to deliver packetized speech

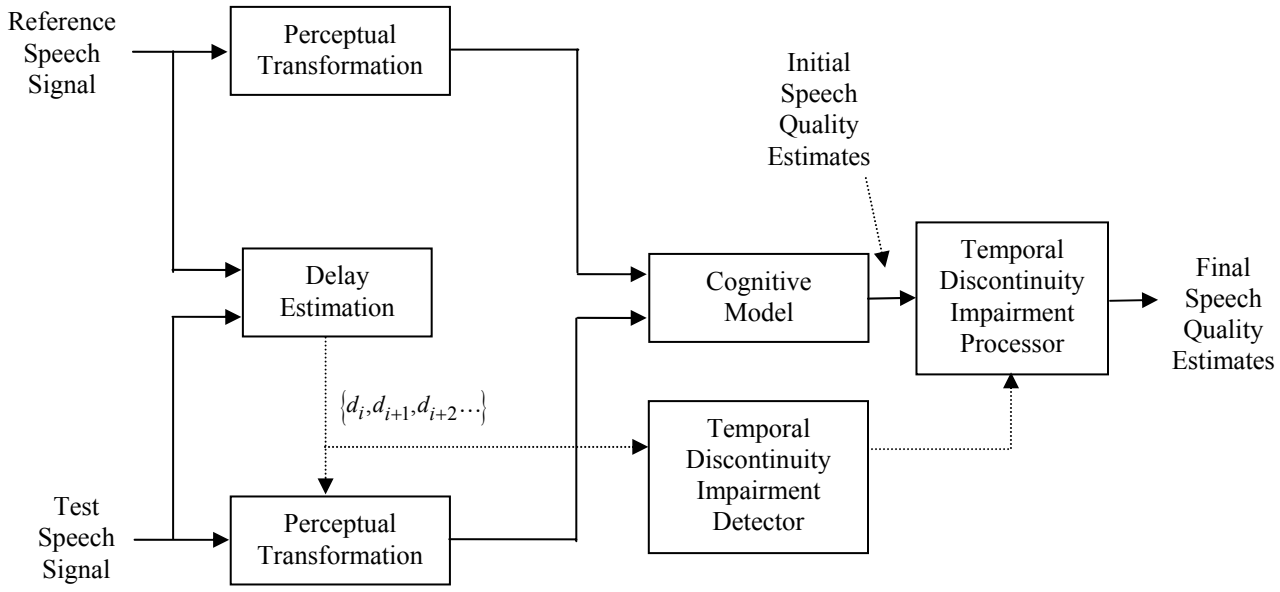

Figure 2. Proposed block diagram for a speech quality estimator with explicit accounting for perception of temporal discontinuity impairments.

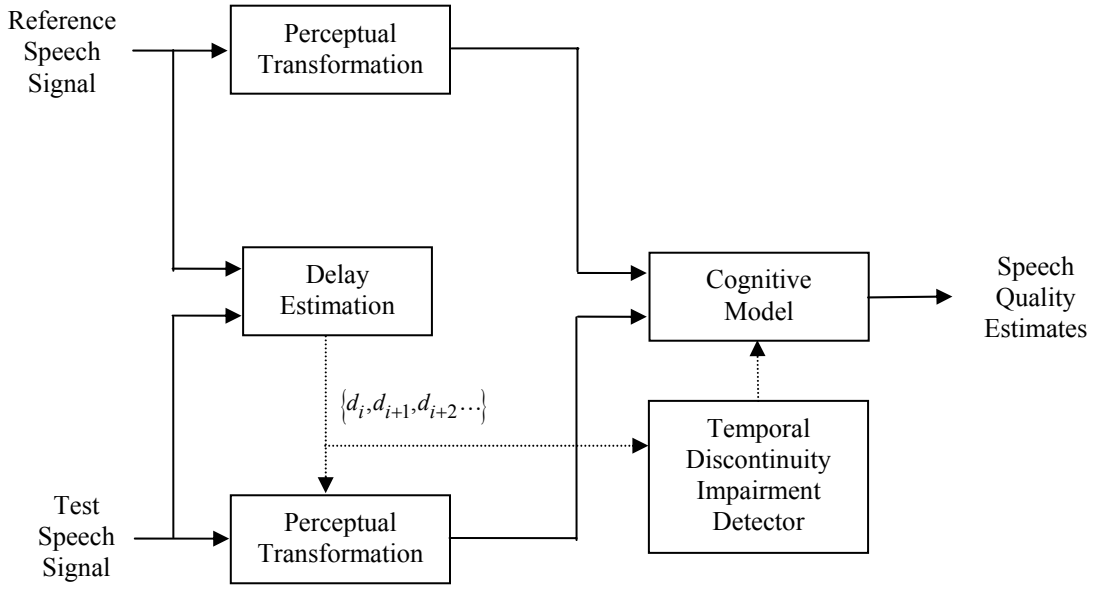

Figure 3. A second proposed block diagram for a speech quality estimator with explicit accounting for perception of temporal discontinuity impairments.

using best-effort networks is likely to continue for years into the future [16].

Another response to these challenges involves adaptively changing the playout speed (with pitch correction) in light of the current jitter buffer state. One would play slower when the buffer is too empty and play faster when the buffer is too full [18]-[20]. This exploits the listener's insensitivity to minor modulations in the speed of a speech signal. In effect, these techniques pass the delay jitter on to the listener in a relatively inconspicuous form at the cost of some added algorithmic complexity.

In spite of these efforts, it appears inevitable that some jitter buffers will occasionally overflow, underflow, or simply not hold the contents of a given packet at its scheduled playout time. Yet we are not aware of any published results on how the resulting temporal discontinuities affect perceived speech quality. In the following we describe an experiment that assesses the degradations to speech quality caused by three different impairments that can result from limitations in networks, jitter buffers, and jitter buffer management algorithms. We call these the loss, pause, and jump impairments. We note below that jitter buffer management algorithms often have the option of converting jump or pause impairments into loss impairments. One of the motivations for this work was to determine whether or not such conversions are desirable in terms of perceived speech quality.

In the following sections we describe the design and implementation of the experiment, tabulate the results, and discuss the general trends. We then provide a simple mathematical model for the experimental results and discuss the question of generalization outside the parameters of this specific experiment.

3. EXPERIMENT DESIGN

3.1 Impairments

The experiment addressed three basic impairments, two of which include temporal discontinuities. These impairments are described below and example waveforms are given in Figure 4.

Consider the situation where packet N-1 has been played and the jitter buffer is empty when it is time to play packet N. The jitter buffer management algorithm must notify the decoder that no packets are available (buffer underflow) and the decoder must take some action to fill in until a packet is available for decoding. This filling in of lost waveforms is often called packet loss concealment (PLC).

If the next packet delivered by the network after the underflow event is packet N+1, and if packet N+1 arrives in time for its scheduled playout, then the impairment is called a "loss" because packet N has effectively been lost. In the loss impairment there is a signal outage, speech is lost, but there is no temporal discontinuity. For clarity, the example waveform for the loss impairment in Figure 4 does not include any PLC.

On the other hand, if the next packet delivered by the network after the underflow event is packet N (i.e., packet N is late but not lost), then the buffer management algorithm could choose to play packet N when it arrives. The resulting impairment is called a "pause." In the pause impairment there is a signal outage, no speech is lost, and there is a temporal discontinuity. This temporal discontinuity effectively dilates the time axis, and it is analogous to placing a tape player in pause mode for an instant. For clarity, the example waveform for the pause impairment in Figure 4 does not include any PLC. (If packets N and N+1 are delivered before the originally scheduled playout time of packet N+1, then one buffer management strategy would be to declare packet N lost even though it is only late and to play packet N+1 at its originally scheduled playout time. The resulting impairment would then be a loss rather than a pause. If the long-term average network transmission delay is constant, then any pause can be converted into a loss simply by waiting to resume playout until the network packet delivery has "caught up with the original playout schedule.")

Now consider the situation where packet N-1 has been played, it is time to play packet N, packet N has not yet been delivered by the network, but packet N+1 has been delivered. One buffer management strategy would be to play packet N+1 immediately following packet N-1. This resulting impairment is called a "jump." In the jump impairment there is no signal outage, speech is lost, and there is a temporal discontinuity that effectively contracts the time axis. (An alternate buffer management strategy would be to signal the decoder to fill in for the lost packet N, and then play packet N+1 at its originally scheduled playout time. This strategy would result in a loss rather than a jump. Any jump can be converted into a loss by this technique.)

In actual packetized digital speech transmission systems, the network loss and transmission delay variation and the jitter buffer management characteristics will influence how often the loss, pause, and jump impairments occur, how severe these three impairments are (duration of the loss or pause impairments or magnitude of the temporal discontinuity in the jump impairment), and whether these three impairments appear alone or in various combinations. Time must ultimately be conserved in real-time speech transmission, so buffer management strategies must eventually compensate for time axis dilations and contractions. A simple way to compensate for a pause is to create a jump of the same magnitude, preferably between active speech segments in order to minimize the degradation to speech quality caused by this jump [21]. Similarly, jumps can be compensated with pauses. As mentioned above, more complicated approaches can be used within active speech segments. These approaches involve pitch-corrected playout speed modulation [18]-[20] and can also be viewed as converting abrupt discontinuities in the time axis to smooth dilations and contractions of the time axis.

In this experiment we introduced the loss, pause, and jump impairments in a controlled, parameterized fashion in order to evaluate the effect of each impairment on perceived speech quality

3.2 Speech Coder

We used an algebraic-code-excited linear-predictive (ACELP) speech coder that has been standardized as ITU-T Recommendation G.723.1 [22]. This coder is specified for use in packetized speech applications in ITU-T Recommendation H.323 [23]. We operated the coder at the 5.3 kbit/s rate with voice activity detection (VAD) and comfort noise generation enabled. The coder uses a frame size of 30 ms. The PLC used in G.723.1 involves extrapolation of the line-spectral pair coefficients and the excitation signal from the last received data. The first extrapolated frame is played out at the full level. The second and third extrapolated frames are attenuated by 2.5 and 5.0 dB respectively. After three frames have been extrapolated (90 ms) the output is completely muted.

3.3 Preparation of Speech Signals

We manipulated the bit stream between a G.723.1 software encoder and decoder to produce the loss, pause, and jump impairments in sentences from the Harvard phonetically-balanced sentence lists [24]. The sentences were previously recorded by two female and two male English-language talkers in a quiet environment using a wideband microphone. The recordings were filtered to conform with the intermediate reference system sending characteristic using [25] and normalized to an active speech level of 26.0 ± 0.5 dB below clipping also using [25] before software G.723.1 encoding and decoding using the software provided with [22].

Each of the three impairments (loss, pause, jump) was used to create nine conditions (see Table 1) for a total of 27 conditions. We used impairment magnitudes of 30, 60, and 120 ms (corresponding to 1, 2, and 4 G.723.1 frames respectively). For the loss impairment, the "impairment magnitude" defines the duration of the signal outage and the amount of speech lost. For the pause impairment, it defines the duration of the signal outage and the magnitude of the temporal discontinuity. For the jump impairment, it defines the amount of speech lost and the magnitude of the temporal discontinuity. (The magnitude of each example impairment shown in Figure 4 is 10 ms.) We selected impairment rates of 1, 2, and 4 impairments per sentence, and an approximately constant sentence duration of 100 frames (3 seconds). Thus the approximate averaged impairment rates are 0.01, 0.02, and 0.04 impairments per frame.

To create the loss impairment we simply set the frame erasure flag at the decoder input for the appropriate group of 1, 2, or 4 frames. To create the pause impairment, we inserted 1, 2, or 4 extra frames between the encoder and the decoder, and set the frame erasure flag for those frames. To create the jump impairment we deleted 1, 2, or 4 frames between the encoder and the decoder.

To simulate a very simple temporal discontinuity compensation technique, we created six conditions containing alternating jump and pause impairments. The impairment magnitudes were 30, 60, and 120 ms, and the average impairment rates were 0.02 and 0.04 impairments per frame. For each condition, half of the files had a pause as the first impairment and the other half of the files had a jump as the first impairment.

We also created eight reference conditions, including the unprocessed source speech, G.723.1 coding with no further impairments, G.723.1 coding with randomly distributed single frame losses at the 3% average rate (or 0.03 impairments per frame in the language of this experiment), and the modulated noise reference unit (MNRU) [26] provided in [25] with Q = 6, 12, 18, 24, and 30 dB. We also included four recordings from an actual VoIP service for a total of 45 conditions.

For each of the 41 conditions of present interest, we processed 32 sentences and combined them into 16 sentence pairs. Four sentence pairs came from each of the four talkers. The sentences were such that the G.723.1 VAD was constantly on once the sentence began. Impairment locations inside this single active speech segment were selected using multiple independent realizations of a uniformly distributed random variable, subject to the constraint that multiple impairments would not overlap each other.

3.4 Subjective Experiment

The subjective experiment was an absolute category rating experiment using the mean opinion score (MOS) scale [27]. Thirty-three subjects (fourteen females and nineteen males) were randomly recruited from an employee roster and none of them had any experience in digital speech coding or transmission. The median subject age was 47 years. The listening instrument was a high-quality headset with signal supplied only to a subject's preferred telephone ear. The listening environment was an acoustically isolated chamber with a noise floor below 30 dBA and no additional noise was injected into the chamber. Each subject was allowed to select a preferred listening level at the start of the experiment. The experiment began with a practice session containing six sentence pairs that were selected to approximately cover the entire quality range of the experiment.

Each listener heard each of the 45 conditions four times, resulting in a total of 180 sentence pairs. These sentence pairs were divided into two sessions (approximately 15 minutes each) separated by a 10-minute break. Each session contained one female talker sentence pair and one male talker sentence pair from each condition. The presentation order of sentence pairs in each session was randomized for each subject. Four different versions of the experiment were created so that all 16 sentence pairs of each condition would be heard. Three experiment versions were each heard by eight subjects and the fourth was heard by nine subjects. This design allowed for full balance and minimal repetition

from a subject's viewpoint. It also gave full balance from a talker viewpoint, and from a condition viewpoint.

After each sentence pair was presented, the subject used an electronic screen and pen to select his or her opinion of the speech quality from five choices: "excellent," "good," "fair," "poor," and "bad." These five responses were associated with the integers 5, 4, 3, 2, and 1 for analysis purposes. A total of 132 responses were collected for each condition.

4. RESULTS AND DISCUSSION

Table 1 gives the mean of the 132 responses (MOS values) and the 95% confidence intervals about those means for each condition. Figure 5 shows most of those results graphically. In Figure 5, a different line type is used for each of the four classes of conditions (loss, pause, jump, and pause & jump). The three horizontal groupings of lines correspond to the three average impairment rates (0.01, 0.02, and 0.04 impairments per frame). Within each grouping, impairment magnitude (0, 30, 60, and 120 ms) is plotted on the horizontal axis.

Figure 5 makes several results apparent. For a given impairment frequency and duration, almost all of the 95% confidence intervals overlap. This means that the four different impairments have very similar influences on speech quality. From a time-axis perspective the loss, pause and jump impairments are fundamentally different because they preserve, dilate, and contract the time axis respectively. In particular, the pause and jump impairments are temporal opposites. The near equivalence of the four impairments in terms of perceived speech quality is an unexpected result. It means that in the context of this experiment at least, the conversion of jump or pause impairments into loss impairments is unlikely to change perceived speech quality in a significant way.

In a few cases where significant differences do appear, the jump impairment leads to slightly higher speech quality than the other impairments. Since the jump impairment is the only impairment that does not include a signal outage, these results might mean that temporal discontinuities alone are slightly less detrimental to speech quality than signal outages, at least for the PLC used in this experiment.

For each condition, the total fraction of G.723.1 frames involved in impairments is the product of the impairment magnitude (in frames) and the average impairment rate. The conditions plotted above the asterisks in Figure 5 all have 4% of their frames involved in impairments, but with different levels of clustering. The leftmost point marked with an asterisk (rate=0.01) corresponds to a single four-frame impairment per sentence, the middle point marked with an asterisk (rate=0.02) corresponds to two two-frame impairments, and the rightmost point marked with an asterisk (rate=0.04) corresponds to four single-frame impairments. Thus it is clear that when 4% of the frames are impaired, dispersion of those impaired frames (right point) gives higher speech quality than clustering of those impaired frames (left point). This is true for the loss, pause, and jump impairments. Analysis of the two points that correspond to impairments in 2% of the frames, or the two points that correspond to impairments in 8% of the frames reveal this same preference for dispersion of impaired frames over clustering of impaired frames. In the context of this experiment, we draw the general conclusion that at a constant average impairment rate, the dispersion of the loss, pause, and jump impairments is preferable to the clustering of these impairments.

Figure 5 also shows that speech quality decreases approximately linearly as impairment magnitude increases. Additional analysis shows speech quality is not linearly related to average impairment rate. The relationship

$$\hat{S} = S_0 - 0.125 \cdot m \cdot r^{0.57}, \qquad (1)$$

where $S_0 = 3.98$ is the base G.723.1 speech quality in this experiment, m is the impairment magnitude in ms, and r is the average impairment rate in impairments per frame, provides good estimates of the MOS values of the 35 G.723.1 conditions in this experiment, independent of impairment type (loss, pause, jump, or pause & jump). The MOS estimates given by (1) fall within the 95% confidence intervals of the experimental results in all but nine cases. In five of these nine cases, the estimates miss the 95% confidence intervals by less than 0.05 MOS units and in the remaining four cases the estimates miss the 95% confidence intervals by less than 0.10 MOS units.

Note that these results characterize the worst-case effects of jitter buffer shortcomings: temporal discontinuities. If modifications to the G.723.1 decoder or additional buffering and operations following its output are allowed, then some of those temporal discontinuities can be converted to temporal non-linearities and this should make them less perceptible [18]-[20].

We presently do not have any direct evidence for or against the extension of these results to other speech coders or PLC techniques. However, it seems reasonable to expect that these results (with appropriate adjustment to S_0) would extend to other speech coders with a base speech quality and PLC technique similar to G.723.1. We further hypothesize that speech coders with significantly higher base quality would likely show greater sensitivity to these impairments, while speech coders with significantly lower base quality might show reduced sensitivity to these impairments.

5. CONCLUSIONS

We have designed, conducted, and analyzed an experiment to determine the effects of the loss, pause, and jump impairments on the perceived quality of G.723.1 coded speech. In most packetized speech transmission situations these impairments are currently unavoidable, but they can be

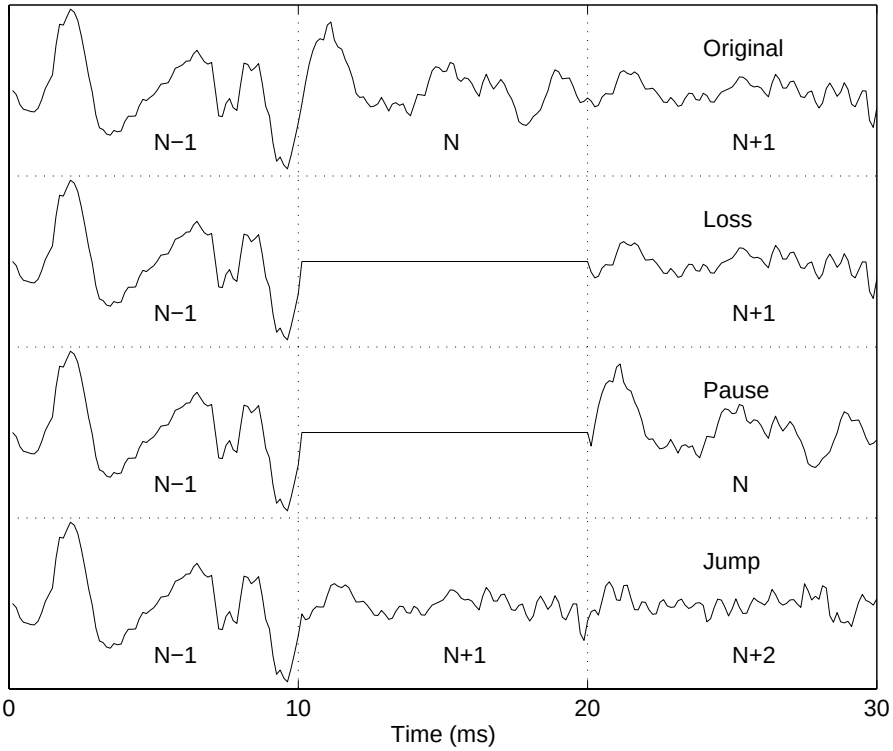

Figure 4. Example speech waveform with impairments.

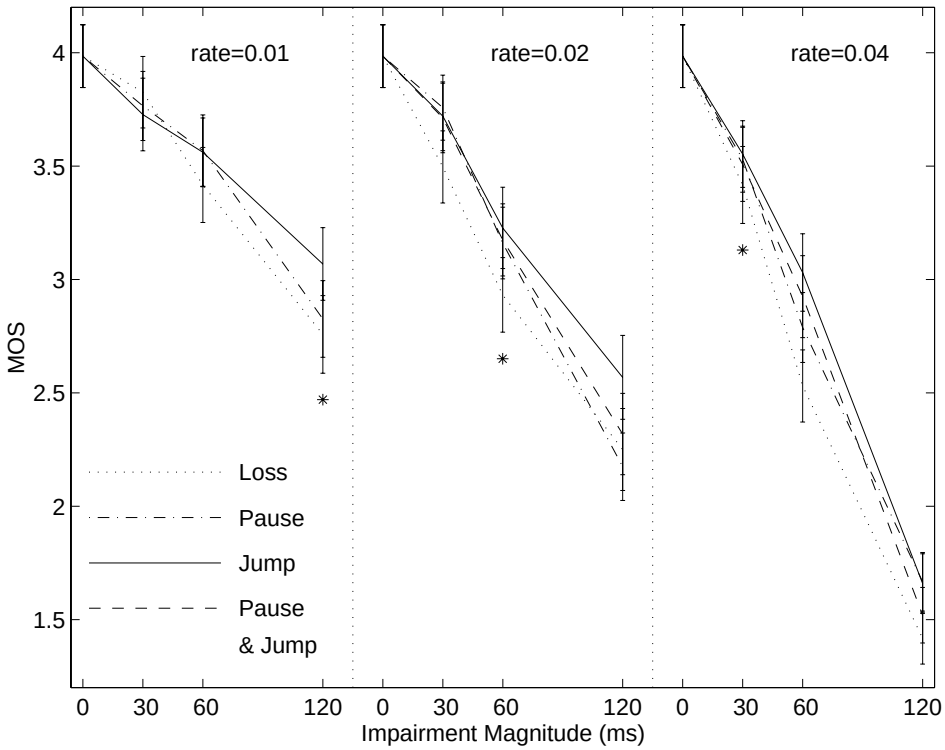

Figure 5. MOS and 95% confidence intervals for 34 conditions.

Table 1. Conditions, average impairment rates, impairment magnitudes, MOS values, and 95% confidence intervals for the experiment.

Impairment Name	Impairment Rate (impairments/frame)	Impairment Magnitude (ms)	MOS and 95% confidence interval
G.723.1 w/ Loss	0.01	30	3.83 ± 0.16
"	0.01	60	3.42 ± 0.17
"	0.01	120	2.76 ± 0.17
"	0.02	30	3.50 ± 0.16
"	0.02	60	2.93 ± 0.16
"	0.02	120	2.25 ± 0.18
"	0.04	30	3.42 ± 0.17
"	0.04	60	2.53 ± 0.16
"	0.04	120	1.42 ± 0.11
G.723.1 w/ Pause	0.01	30	3.77 ± 0.15
"	0.01	60	3.57 ± 0.16
"	0.01	120	2.83 ± 0.17
"	0.02	30	3.76 ± 0.14
"	0.02	60	3.16 ± 0.16
"	0.02	120	2.17 ± 0.15
"	0.04	30	3.53 ± 0.15
"	0.04	60	2.79 ± 0.15
"	0.04	120	1.67 ± 0.13
G.723.1 w/ Jump	0.01	30	3.73 ± 0.16
"	0.01	60	3.56 ± 0.15
"	0.01	120	3.07 ± 0.16
"	0.02	30	3.72 ± 0.15
"	0.02	60	3.23 ± 0.18
"	0.02	120	2.57 ± 0.19
"	0.04	30	3.55 ± 0.15
"	0.04	60	3.03 ± 0.17
"	0.04	120	1.66 ± 0.13
G.723.1 w/ Pause & Jump	0.02	30	3.71 ± 0.15
"	0.02	60	3.17 ± 0.16
"	0.02	120	2.32 ± 0.18
"	0.04	30	3.51 ± 0.16
"	0.04	60	2.92 ± 0.18
"	0.04	120	1.52 ± 0.12
G.723.1 alone			3.98 ± 0.14
G.723.1 w/ Loss	0.03	30	3.56 ± 0.14
Source			4.61 ± 0.09
MNRU, Q=30			4.28 ± 0.14
MNRU, Q=24			4.22 ± 0.13
MNRU, Q=18			3.45 ± 0.13
MNRU, Q=12			2.43 ± 0.13
MNRU, Q=6			1.60 ± 0.12

traded-off against each other, against end-to-end delay, and against network parameters. Using a formal absolute category rating subjective experiment on the MOS scale, we found that these impairments have very similar effects on the perceived quality of G.723.1 coded speech. We also determined that at a fixed average impairment rate, the clustering of these impairments is more detrimental to perceived speech quality than the dispersion of these impairments. In this experiment MOS decreases approximately linearly with the magnitude of these impairments, and it decreases approximately as the square root of the average rate of these impairments. We hypothesize that the results would extend to other speech coders with speech quality and PLC similar to those of G.723.1.

It is expected that the results presented here could be used to begin the work of additional temporal discontinuity detection and processing as suggested in Figures 2 and 3. Objective speech quality estimators with such processors may provide more accurate estimates of packetized speech quality when significant temporal discontinuities are present. These results may also aid those involved in jitter buffer and jitter buffer management algorithm design issues as they trade off impairments, delay, and algorithm complexity.

REFERENCES

[1] J.G. Beerends, A.W. Rix, M.P. Hollier, and A.P Hekstra, "Perceptual evaluation of speech quality (PESQ) The new ITU standard for end-to-end speech quality assessment, Part I – Time-Delay Compensation," *J. Audio Eng. Soc*., vol. 50, no. 10, pp. 755-764, Oct. 2002.

[2] J.G. Beerends, A.W. Rix, M.P. Hollier, and A.P Hekstra, "Perceptual evaluation of speech quality (PESQ) The new ITU standard for end-to-end speech quality assessment, Part II – Psychoacoustic Model," *J. Audio Eng. Soc*., vol. 50, no. 10, pp. 765-778, Oct. 2002.

[3] 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," Geneva, 2001.

[4] G. Thomsen and Y. Jani, "Internet telephony: going like crazy," *IEEE Spectrum*, vol.37, no. 5, pp. 52-58, May 2000.

[5] M. Hassan and A. Nayandoro, "Internet telephony: services, technical challenges, and products," *IEEE Communications Magazine*, vol. 38, no. 4, pp. 96-103, Apr. 2000.

[6] T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec, and J. Mahler, "Real-time voice over packet-switched networks," *IEEE Network*, vol. 12, no. 1, pp. 18-27, Jan./Feb. 1998.

[7] J. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," *J. High Speed Networks*, vol. 2, pp. 305-323, 1993.

[8] M. J. Karam and F. A. Tobagi, "Analysis of the delay and jitter of voice traffic over the Internet," in *Proc. IEEE INFOCOM 2001*, pp. 824-833.

[9] L. Zheng, L. Zhang, and D. Xu, "Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP)," in *Proc. IEEE ICC 2001*, vol. 1, pp. 123-126.

[10] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE Infocom '99*, vol. 1, pp. 345-352.

[11] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40-48, Sep./Oct. 1998.

[12] H. P. Sze, S. C. Liew, and Y. B. Lee, "A packet-loss-recovery scheme for continuous-media streaming over the Internet," *IEEE Communications Letters*, vol. 5, no. 3, pp. 116-118, Mar. 2001.

[13] C. Kuo, C. Chio, W. Hsi, and W. Chen, "Delivering voice over the Internet," in *Proc. IEEE ICCT 2000,* vol. 1, pp. 628-632.

[14] N. Erdöl, C. Castelluccia, and A. Zilouchian, "Recovery of missing speech packets using the short-time energy and zero-crossing measurements," *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 3, pp. 295-303, Jul. 1993.

[15] J. Suzuki and M. Taka, "Missing packet recovery techniques for low-bit-rate coded speech," *IEEE J. Selected Areas in Communications*, vol. 7, no. 5, pp. 707-717, Jun. 1989.

[16] B. Li, M. Hamdi, D. Jiang, X. Cao, and Y. T. Hou, "QoS-enabled voice support in the next-generation Internet: issues, existing approaches and challenges," *IEEE Communications Magazine*, vol. 38, no. 4, pp. 54-61, Apr. 2000.

[17] J. K. Muppala, T. Bancherdvanich, and A. Tyagi, "VoIP performance on differentiated services enabled network," in *Proc. IEEE ICON 2000*, pp. 419-423.

[18] Y. J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling and loss concealment for voice communication over IP networks," submitted to *IEEE Trans. Multimedia*, Apr. 2001.

[19] Y. J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," in *Proc. IEEE ICASSP '01*, vol. 3, pp. 1445-1448.

[20] F. Liu, J. Kim, and C. Kuo, "Adaptive delay concealment for Internet voice applications with packet-based time-scale modification," in *Proc. IEEE ICASSP '01*, vol. 3, pp. 1461-1464.

[21] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio

applications in wide-area networks," in *Proc. IEEE INFOCOM '94*, vol. 2, pp. 680-688.
[22] ITU-T Recommendation G.723.1, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," Geneva, 1996.
[23] ITU-T Recommendation H.323, "Packet-based multimedia communications systems," Geneva, 1999.
[24] IEEE Recommended practice for speech quality measurements, *IEEE Trans. Audio and Electroacoustics*, vol. AU-17, no. 3, pp. 225-246, Sep. 1969.
[25] ITU-T Recommendation G.191, "Software tools for speech and audio coding standardization," Geneva, 1996.
[26] ITU-T Recommendation P.810, "Modulated noise reference unit (MNRU)," Geneva, 1996.
[27] ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality," Geneva, 1996.