Decoder State-Copying for Bluetooth CVSD Packet Loss Concealment

Xuejing Sun and Kuan-Chieh Yen

Cambridge Silicon Radio
Auburn Hills, MI USA
{xuejing.sun,kuan.yen}@csr.com

Abstract— **This paper describes a novel packet loss concealment (PLC) algorithm for Bluetooth CVSD codec by updating decoder states efficiently through state-copying. In the proposed method, the decoder states are stored in buffers and then selected based on the estimated pitch period of the decoded signal. The selected states are used to reset the decoder states when the packet is lost or corrupted. Objective evaluations show the proposed algorithm consistently outperforms the decoded domain PLC algorithm.**

Keywords-state copy; packet loss concealment; CVSD; Bluetoot;, speech enhancement

I. INTRODUCTION

Packet loss occurs frequently in VoIP or wireless voice communications under adverse connection conditions [1]. Lost packets result in clicks and pops or other artifacts that greatly degrade the perceived speech quality at the receiver side. Packet loss also poses significant challenges to automatic speech recognition engines [2]. In the increasingly ubiquitous Bluetooth wireless communication, voice is transmitted over synchronous connection oriented (SCO) or extended synchronous connection oriented (eSCO) logical links, using primarily Continuous Variable Slope Delta Modulation (CVSD) codec [3]. Like many other wireless communication protocols, Bluetooth, operating within the public 2.4 GHz ISM (Instrumentation, Scientific, and Medical) band, subjects to various interferences. Besides limited retransmission mechanism in eSCO links, SCO/eSCO does not offer sophisticated sender-based packet loss recovery techniques due to the stringent requirements of low power consumption and low delay. Furthermore, unlike many modern speech codecs, there is no built-in error concealment mechanism in the Bluetooth CVSD codec. Rather, as pointed out by the specification [3], it is left to the developer to "'fill' in the lost speech segment".

To combat the adverse impact of packet loss, many packet loss concealment algorithms, also known as frame erasure concealment algorithms, have been proposed in the past [1]. Such algorithms normally operate at the receiver side by generating a synthetic speech signal to cover missing data (erasures) in a received bit stream. Among various PLC methods, pitch based waveform substitution, such as G.711 Appendix I [4], is highly popular due to its relative simplicity and good performance. Using the quasi-periodic property of speech signals, the basic idea of such methods is to fill the gap using waveform segment that is one or multiple pitch periods

apart from the gap. Such an approach can be very effective for simple, memoryless coding scheme like PCM. However it is unable to fully address packet loss problems for decoders that rely on internal states in addition to input bit stream to decode data. In such codecs, when a packet is lost or corrupted, the internal states of the decoder typically remain at the pre-loss condition assuming that the corrupted packet is not decoded. If the corrupted packet is decoded, not only erroneous output is generated, the decoder is also left with incorrect internal states. In either case, the disturbance of the state values will render the decoded outputs incorrect for subsequent packets even though correct bit sequences are received. This is because the incoming coded bits were generated by the encoder based on the true states that are different from the ones left in the decoder after a packet loss. Depending on the codec, erroneous internal states can result in an unstable decoder and serious audio artifacts. This is the case for the Bluetooth CVSD codec, where incorrect internal states such as step-size lead to erroneous decoded waveform offsets [2]. Therefore, errors in the decoder states should be corrected as much as possible in order to achieve satisfactory audio quality.

Some methods have been proposed in the literature to address the state mismatch issue. In [2], CVSD decoder states are implicitly updated by using silence substitution or packet repetition on the decoder input bits. Although being efficient, the effectiveness of such methods is limited as the updated state values could still be very different from the true values. Pitch based bit stream substitution is proposed in [5], where the corrupted/lost coded data are replaced with coded data in a history buffer based on pitch. The states are implicitly updated by decoding the replacement bits. In [6], lost frames are replaced with decoded data and the states are obtained by re-encoding the replacement data. Finally a re-phasing process is employed to update the decoder states in [7], where the phase difference is the time lag between the extrapolated signal and the decoded audio signal. With the extra encoding, decoding, or extrapolation operations, the above methods incur additional computational cost which is undesirable for highly resource limited platforms such as Bluetooth.

In this paper, we propose an efficient alternative approach to update CVSD decoder states via state-copying. Specifically, proper state values are selected from a buffer which stores all the pre-loss states up to a pre-defined duration. The selection criterion is based on the pitch of the decoded signal. The remainder of this paper is organized as follows. In section II a brief overview on the Bluetooth CVSD codec is given. The

state-copying algorithm is described in section III. Finally the simulation results are presented in section IV, and a conclusion is made in section V.

II. BLUETOOTH CVSD CODEC

The CVSD codec in Bluetooth protocol is an adaptive delta modulation algorithm, where the encoded bits indicate whether the prediction value is smaller or larger than the input waveform. The input to the CVSD encoder is linear PCM with 64 kHz sampling rate. Sampling rate conversion is needed before CVSD encoder and after CVSD decoder for narrowband speech sampled at 8 kHz. Unlike many other codecs where some bits are more crucial, bits have equal weight in CVSD. Although this avoids a devastating decoder breakdown on receiving corrupted data, it also means that all the bits need to be equally protected in contrast with only needing to protect or recover the important bits. Random bit errors in CVSD bit streams are manifested by a higher level background noise in the decoded PCM speech, but the signal can remain intelligible except in the extremely high Bit Error Rate (BER) condition. This can be attributed to the Bluetooth CVSD's 1-bit delta modulation mechanism and the low pass filtering effect of the down-sampling conversion from 64 kHz to 8 kHz.

The main components of a CVSD decoder are the step control module and the accumulator [3]. Let $b(k)$ denote the CVSD decoder input bit, $y(k)$ the accumulator contents, and $\delta(k)$ the step size. The output of the decoder $x(k)$ is given by

$$x(k) = hy(k) \tag{1}$$

where h is the decay factor for the accumulator. $y(k)$ is derived from the previous decoder output, the input bit, and the step size, as expressed by

$$y(k) = x(k-1) + b(k)\delta(k) \tag{2}$$

Note that $b(k)$ has values -1 and 1, mapped from 1 and 0, respectively.

To reduce slope overload effects, Bluetooth CVSD codec uses 4-bit syllabic companding, where the adaptive step-size value is adjusted based on whether the four most recent encoded bits are equal or not.

$$\delta(k) = \begin{cases} \min\{\delta(k-1) + \delta_{\min}, \delta_{\max}\}, & \text{if 4 bits equal,} \\ \max\{\beta\delta(k-1), \delta_{\min}\}, & \text{otherwise,} \end{cases} \tag{3}$$

where β is the decay factor, $\delta_{\min}$ and $\delta_{\max}$ are the minimum and maximum step sizes.

It can be seen that in the CVSD codec, in order to generate the output $x(k)$, both previous decoder output and the current step size are needed. In the next section we will describe the state-copying method for updating these two state variables.

III. DECODER STATE UPDATE VIA STATE-COPYING

In stead of performing decoding or encoding to get the proper state values, we select the states from a buffer which stores all the intermediate states ($x(k)$, $\delta(k)$) before the corrupted packet. The selection criterion is based on the pitch

of the decoded signal. Fig. 1 is the block diagram of the proposed algorithm.

When the received packet is corrupted or simply lost, a pitch-based waveform substitution PLC is performed, which estimates the pitch of the corrupted packet using a buffer of past decoded signal. As shown in Fig. 1, the pitch is not only used for selecting a slice of buffered decoded signal to replace the corrupted signal, but also used to select a state from the state buffer to be the ending state of the lost packet or the initial state of the first good packet after erasure. Specifically, the selected state has a time lag with the ending state of the lost packet corresponding to the estimated pitch period or its integer multiples. This is based on the same principle used by the decoded domain pitch-based PLC algorithms, which takes advantage of the quasi-periodic property of speech signals. Note that since the pitch is estimated using the down-sampled decoded signal, the actual pitch period for state selection is eight times of the estimated value.
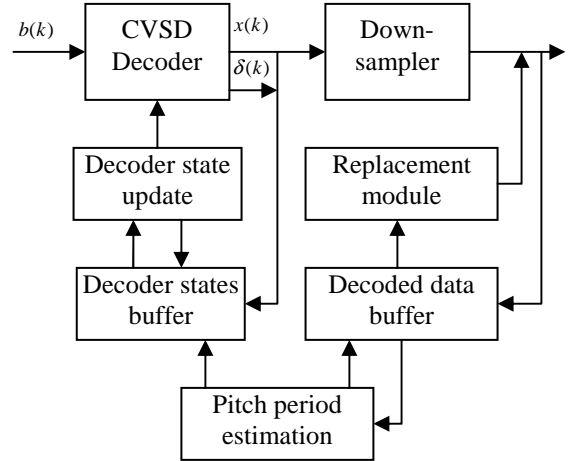

Figure 1.   Block diagram of the proposed PLC system.

Consider the case where a packet length (L) is 240 bits (@ 64 kbps) and the pitch period is estimated to be: $P_{\text{decoded}} = 60$ samples (@ 8 kHz). This corresponds to a pitch period of: $P_0 = 60 \times 8 = 480$ bits. Since the ending state of the decoder is required, the decoder state update module selects the decoder state that is located $P_0 - L$ bits from the end of the decoder state buffer. In this case: *Selected state* $= P_0 - L = 480 - 240 = 240$ bits. If $P_0 - L < 0$, then the decoder state update module selects the decoder state that is located $nP_0 - L$ bits from the end of the decoder state buffer, where n is the smallest integer that satisfies $nP_0 - L >= 0$. The minimum length of the decoder state buffer is $P_{\max} - L$, where $P_{\max}$ is the maximum pitch period under consideration. For example, $P_{\max}$ might be chosen to be 107×8 bits which corresponds to a minimum pitch frequency of 75 Hz.

Note that if the selected state is too far apart from the ending state of the lost packet, it may become unsuitable for replacement due to potential amplitude mismatch. This issue can be alleviated by performing some simple adjustment on the selected state as described below.

For a Bluetooth CVSD codec, the two state variables can be represented as a state vector, S: $S = [\delta, x]$. Fig. 2 illustrates the decoder states in the decoder state buffer. S_i is the state vector selected to update the decoder, S_j is the ending state vector before the degraded packet, S_k is the state vector that is one pitch period previous to S_j, L is the length of the degraded packet and P_0 is the estimated pitch period. Adjustments may be made to the selected state, S_i, based on the amplitude ratio between the state just before the degraded packet, S_j, and the state with the time lag of one pitch period from it, S_k:

$$S_i^{'} = \frac{S_j}{S_k} S_i \qquad (4)$$

This is based on the assumption that when the packet length is not too long, the amplitude relationship between the selected state S_i and the ending state of the lost packet can be approximated by that of S_j and S_k.
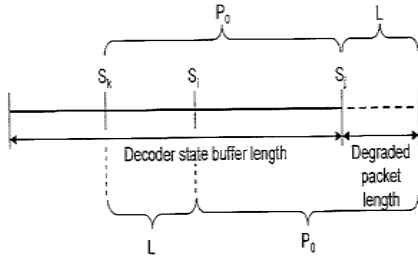

Figure 2. Schematic representation of the decoder state buffer.

When the received packet is good, it is passed to the decoder directly and all the intermediate states emitted by the decoder and the decoded signal are stored in the corresponding buffers. Note that updating all the buffers with corresponding substitutes is also required after a lost packet.

As shown in Fig. 1, the second part of the proposed system employs a typical decoded domain PLC algorithm. Such an algorithm consists of a pitch estimation module, a data buffer, and a replacement module. The data buffer contains history signal samples before the lost packet, which is used for both pitch estimation and data replacement. The length of the buffer is at least two times of the expected maximum pitch period. For example, if the minimum pitch is 75 Hz, the buffer length is at least 213 samples at 8 kHz sampling rate. The replacement module is responsible for selecting samples from the buffer to fill the gap. Overlap-add is performed at the concatenation boundary to smooth out discontinuities.

In the proposed system, the pitch period is estimated using a highly efficient three-stage pitch determination algorithm [8]. In stage one, pitch is calculated using the standard normalized cross-correlation (NCC) method with time lag covering the pitch range of [75, 150] Hz. The input signal is decimated by a factor of two for reduced complexity. Stage two selects the best pitch candidate based on the pitch determined in stage one and its integer multiples. In stage three the estimated pitch is further refined by selecting the pitch that minimizes distortions at the concatenation boundary where an overlap-add is performed. Using this pitch estimation method, the decoded domain only PLC algorithm has been shown to outperform the commonly

used G.711 Appendix I PLC algorithm [4] yet being substantially more efficient.

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed state-copying method, we compared the PLC systems with and without state-copying. The ITU standard - PESQ (Perceptual Evaluation of Speech Quality) [9] was used as the objective measure. In this paper we reported PESQ MOS-LQO score [10], which is converted from the raw PESQ score to allow a linear comparison with subjective MOS. The clean speech files were from NOIZEUS [11], which contains 30 IEEE sentences (produced by three male and three female speakers).). To simulate packet losses in Bluetooth voice communication, input signals are first up-sampled to 64 kHz, and then coded using CVSD encoder. The bit stream is corrupted by dropping signal packets at random locations, with the loss rate at 5%, 10%, 20%, 30%, 40%, and 50%. The corrupted bit stream is then decoded using CVSD decoder and the output is down-sampled to 8 kHz. In the place of a packet drop, a packet of zeros is inserted into the decoded signal to generate the unprocessed corrupted signal. The corrupted speech signal is then processed by the proposed PLC with and without decoder state update, respectively. By comparing the two outputs, the effectiveness of the proposed state-copying method can be validated. The length of the packet obviously has impacts on the PLC performance. In Bluetooth, there are 11 types of synchronous (SCO and eSCO) packets, with payload length ranging from 1 to 540 bytes [3]. To be representative, we chose packet sizes of 30 and 60 samples at 8 kHz sampling rate in the simulation test, which correspond to the maximum length of the commonly used HV3/EV3 and 2-EV3 packets.
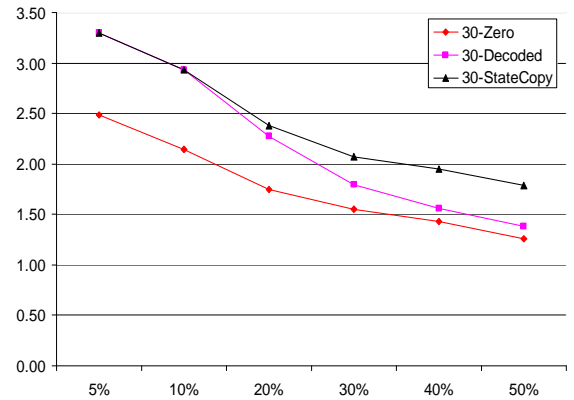

Figure 3. PLC PESQ MOS-LQO scores for packet size of 30 samples at various loss rates

Fig. 3 illustrates the PESQ MOS-LQO scores of the proposed PLC algorithms for packet size of 30 samples at various loss rates. The curve '30-Zero' represents the scores of zero-inserted signals, whereas '30-StateCopy' and '30-Decoded' are the results of the proposed algorithm with and

without codec state update. It can be seen that both PLC systems outperform the baseline system at all the loss rates. Notably the advantage of state-copying becomes more substantial at higher loss rates. This shows that at higher loss rates, the CVSD decoder state drifts further away from the true values, which causes prolonged degradation on subsequent packets. Without proper states, the PLC algorithm relying solely on the decoded signal fails to yield significant improvements over the unprocessed input.
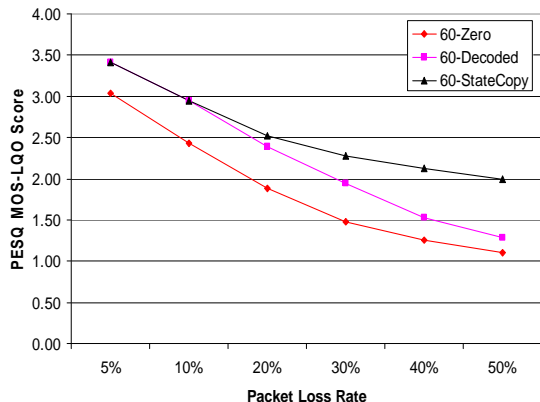

Figure 4. PLC PESQ MOS-LQO scores for packet size of 60 samples at various loss rates

Fig. 4 shows the results for 60-sample packet size, which reveals similar patterns. One noticeable difference between the two packet sizes is that the improvement of '60-StateCopy' over '60-Zero' and '60-Decoded' appears to be more substantial than that of 30-sample packet size. This implies that for longer packets maintaining proper state values may be more critical.
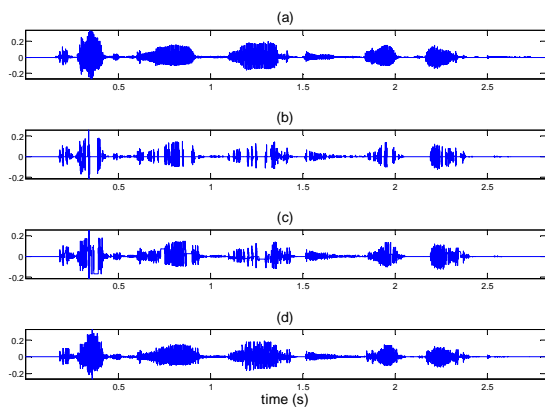

Figure 5. Waveform examples of PLC output for packet size of 60 samples: (a) orginal clean speech; (b) zero-inserted corrupted signal; (c) output waveform of the proposed PLC system without decoder state update; (d) output waveform of the proposed PLC system with decoder state update via state-copying

Fig. 5 shows waveform examples at 50% loss rate for packet size 60. It is evident from Fig. 5(c) and 5(d) that the output of state-copying method has fewer discontinuities and is much closer to the original speech signal. Perceptually it

sounds smoother and less robotic.

The state-copying method is highly efficient as it requires neither encoder nor decoder operation during packet loss, thus saves the precious computing resource to the decoded domain PLC module. The data buffering process can be efficiently implemented on DSP via circular buffering. The state-copying approach, however, does have its drawback of potentially high memory consumption. For CVSD, it is not a serious issue since there are only two state variables. For codecs containing nontrivial numbers of state variables, a large amount of memory may be needed. For example, ITU-T ADPCM speech codec G.726 [12] has over 20 internal state variables, which makes state-copying a less attractive solution.

V. CONCLUSIONS

We have described a novel packet loss concealment algorithm for Bluetooth CVSD codec. The unique contribution is the efficient decoder state update method using state-copying. In this approach, the internal states of CVSD decoder after packet loss are set to the states selected from the state buffers, which store all the pre-loss state values. The selection criterion is based on the estimated pitch period of the decoded signal. Objective tests using PESQ demonstrate that the proposed algorithm consistently outperforms both the zero-inserted approach and the decoded domain PLC without decoder state update.

REFERENCES

[1] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, Sept./Oct. 1998.

[2] A. H. Nour-Eldin, H. Tolba and D. O'Shaughnessy, "Automatic recognition of Bluetooth speech in 802.11 interference and the effectiveness of insertion-based compensation techniques", *Proc. ICASSP*, 2004.

[3] Bluetooth Core Specification v2.1 + EDR (http://www.bluetooth.org)

[4] ITU-T Recommendation G.711 Appendix I, "*A high quality low-complexity algorithm for packet loss concealment with G.711,*" 1999.

[5] J. Stemerdink, A. Meijerink, Method for replacing corrupted audio data, Patent number: 7206986, 2007

[6] J. Thyssen, R. W. Zopf, J-H Chen, "Updating of Decoder States After Packet Loss Concealment" US Patent Application No. 11/838,895, Aug 15, 2007

[7] R. W. Zopf, J. Thyssen, J-H Chen, "Re-phasing of Decoder States After Packet Loss", US Patent Application No. 11/838,905, Aug 15, 2007

[8] X. Sun and S. Gadre, "Efficient three-stage pitch estimation for packet loss concealment," *Proc. Interspeech*, 2010 (to appear).

[9] ITU-T P.862, *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2001.

[10] ITU-T P.862.1, *Mapping function for transforming P.862 raw result scores to MOS-LQO*, 2003

[11] Y. Hu and P. Loizou, "Subjective evaluation and comparison of speech enhancement algorithms," *Speech Communication*, 49, 588-601, 2007.

[12] ITU-T Recommendation G.726, "*40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM),*" 1990