Packet loss concealment method based on hidden Markov model and decision tree for AMR-WB codec

Tarek Gueham
University of Sciences and Technology Houari Boumediene
Fatiha Merazka (fmerazka@usthb.dz)
University of Sciences and Technology Houari Boumediene

Research Article

Keywords: VoIP, packet loss concealment, HMM, decision tree, WB-PESQ, EMBSD, MUSHRA

Posted Date: April 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1543828/v1

multimedia systems manuscript No.
(will be inserted by the editor)

Packet loss concealment method based on hidden Markov model and decision tree for AMR-WB codec

Tarek Gueham · Fatiha Merazka

Received: date / Accepted: date

Abstract In this paper, we introduce hidden Markov model (HMM) based packet loss concealment (PLC) methods and discuss the impact of Markovian assumptions on the performance of these models. We also present a new PLC method, implemented on the G.722.2 codec, which relies on the HMM and Decision Tree (DT), namely HMDT architecture, to enhance the perceptual quality of Voice over Internet Protocol (VoIP) communications under severe packet loss conditions. The proposed method is a receiver-based model that tracks the statistical evolution of speech signals through HMM and uses the DT architecture to predict/estimate accurately the lost speech packets by exploiting the surrounding received speech packets. Objective and subjective metrics are used to evaluate the performance of the proposed method. Test results show that our proposed method enhances considerably the speech quality of the reconstructed speech signal and produces a more natural speech variation compared to conventional PLC methods.

Keywords VoIP · packet loss concealment · HMM · decision tree · WB-PESQ · EMBSD · MUSHRA

1 Introduction

Voice over Internet Protocol (VoIP) refers to the transmission of voice in data networks. This technology has attracted a lot of attention in the last decade and it is deemed the future alternative of traditional Public Switched Telephony Network (PSTN). Unfortunately, the actual architecture of the Internet cannot guarantee the minimum Quality of Service (QoS) required by this kind of real-time applications. In fact, packet loss is considered to be the main source of speech impairment

T. Gueham
LISIC Laboratory, Telecommunications Department, USTHB University, Algiers, Algeria
E-mail: gueham.tarek@gmail.com

F. Merazka
LISIC Laboratory, Telecommunications Department, USTHB University, Algiers, Algeria Tel.: +213-21-247187
Fax: +213-23-467541
E-mail: fmerazka@usthb.dz

in VoIP. This problem is mainly a result of discarding speech packets in the intermediate IP networks due to congestion or dropping packets at the receiver level due to their late arrival [1, 2].

The degradation of perceived speech quality due to packet loss can be affected by several factors such as the loss location, and the loss pattern. Authors in [1] show that the loss location has a severe impact on perceived speech quality. Their test results demonstrate that the loss at unvoiced speech segments has a minor impact on speech quality. However, the loss at the beginning of voiced segments has the most severe impact on perceived speech quality. This fact can be explained by the high inter-packet dependency in the decoding process, especially for Code Exited-Linear Prediction (CELP) based codecs like G.722.2. Moreover, the packet loss in voiced segments affects the decoding process of subsequent speech packets as the decoder needs a finite time (the convergence time) to resynchronize its state with that of the encoder [3, 4].

Additionally, the packet loss in IP networks is usually bursty, and packet losses often occur in pairs or triplets causing the loss of 40 to 60ms of speech segments, which is much longer than phoneme duration. Consequently, the degradation of the perceived speech quality becomes more severe as the size of lost segments increases. Apart from the annoying sound artifacts, bursty losses can induce misunderstanding problems between the speaker and the listener and it can lead eventually to a total desynchronization between the encoder and decoder [5].

To relieve the effect of packet loss, the approaches of PLC are proposed to enhance the perceptual speech quality at different network conditions [6]. These approaches can be divided into sender and receiver-based methods [7]. Sender-based methods rely on the extra information sent by the encoder to perform the recovery process. A common approach of sender-based methods is the use of error-correcting codes using Checksum based algorithms or via parity coding such as Low-Density Parity-Check Codes (LDPC), Reed Salomon (RS) or repetition techniques to recover the lost speech data [8–10]. Similarly, Multiple Description Coding (MDC) uses the repetition technique to counter the packet loss problem [11]. In this technique, two descriptions of the same speech packet are sent, in which the first one is a high-quality description and the second one is a redundant speech packet that has a lower bitrate coding technique. Forward Error Correction (FEC) is another sender-based approach. It is commonly used in multimedia streaming to avoid losses of clustered packets or data blocks [8–10]. In general, sender-based PLC strategies are utilized to mitigate the effect of packet loss and to improve the performance of receiver-based schemes. The main advantages of these approaches are resiliency and implementation simplicity. However, they induce extra bandwidth consumption that can increase the network overload. These solutions are also very sensitive to bursty packet errors that result in the loss of both original and redundant data, making the recovery process impossible. Another drawback of these techniques is the extra latency needed for data recovery since the receiver has to wait for the reception of redundant information before the initiation of the recovery process. Note that this redundant information will be deemed useless if it arrives after the deadline play-out of the lost packets [3].

Due to the aforementioned problems related to sender-based techniques, many researchers focused their attention on receiver-based methods instead, since they do not require the assistance of the sender and consequently they reduce the recovery time. The main goal of this approach is to minimize the perceptual distortion

of speech signals resulted from speech packet loss by replacing the lost ones with an appropriate substitute. A simple way to do that is to use the insertion method. This method replaces the lost speech segment with silence, noise, or simply with the last received speech segment. Another approach is to use the neighboring correctly received packets to regenerate a substitute packet to replace the lost one. This is carried out by using a Linear Prediction model (LP), autoregressive analysis filters to predict the lost speech packets from the last received ones or by using Time Scale Modification to stretch the surrounding speech segments of the loss to fill the speech gap induced by the loss. Waveform replication may also be used to exploit the short-stationarity of the speech signal in order to provide a suitable substitute for the lost speech portion.

Generally, receiver-based methods deploy the short stationarity feature of the speech signal to hide the lost speech segments by averaging, interpolating, or stretching its extremities. Unfortunately, the efficiency of those methods varies based on the properties of the lost segments. For instance, speech onset segments and unvoiced to voiced transitions have a fast pitch and gain variation [4]. In fact, these methods cannot follow the overall statistical trends of the speech signal and they simply ignore the dynamics of its statistical evolution [12]. Note that most of these techniques do not take into consideration the signal distortion after the loss segment (due to the desynchronization between the encoder and the decoder) and they do not provide a solution to update the decoder states as well.

n the literature, it has been shown that conventional receiver-based techniques are not effective for packet loss ratio (PLR) exceeding 7% and/or loss duration in the order of 40 ms [13]. Moreover, the perceptual speech quality guaranteed by these techniques do not provide significant improvements to the user experience due to the inevitable annoying artifacts like the noticeable sudden transition between natural and synthesized speech, long pitch repetitions, and sudden gain variations. To solve these problems, receiver-based PLC techniques have to take into consideration the statistical evolution of speech signals as well as the long-term speech dependencies to provide a more natural variation between the synthesized speech signal and the reconstructed speech signal and consequently enhance the perceptual speech quality in VoIP communications

In this context, Hidden Markov Model (HMM) is a well-known statistical tool that has been extensively used in speech signal processing. In applying HMMs to PLC, the received speech packets (or speech parameters) can be viewed as emissions produced by an HMM state with continuous probability density functions. The decoder can then track the evolution of speech signals through the HMM and provide an estimation of the lost packet using the conditional probability density functions [12]. This can be carried out using two main approaches: the first one is the prediction approach, where the HMM predicts the missing speech parameters without any knowledge of the future ones. This approach is helpful when the speech packet following the lost one is missed (considered lost due to burst errors or simply failed to arrive). In this case, the HMM has to predict the missing parameters by analyzing only the previously received speech packets. The second approach is based on estimation. In this scenario, past and future emissions are available and used to estimate the lost speech parameters. This approach provides much better results than the first one at the expense of less applicability due to real-time constraints of VoIP applications.

In spite of its promising abilities in capturing, modelling as well as and predicting the variation of speech events, the interest of HMM in the PLC context remains limited in the literature. The authors in [14] proposed a Markov Chain Prediction (MCP) method to predict the values of the missing parameters of lost speech packets. The proposed method is based on Mixture Transition Distribution (MTD), in which the probabilities of all states are calculated. The vocoder synthesizer will then use the state parameters with the highest probability to synthesize the missing speech packet. If the state probabilities are too close, the vocoder synthesizer will opt for the repetition method in order to enhance the speech quality of the reconstructed packets. Unfortunately, this method takes into consideration only two vocoder parameters: Long Term Prediction (LTP) Lag and LTP gain in both of the learning and the implementation phases. Consequently, the proposed method is unable to track sudden pitch variation in missing speech packets. Another drawback of this method is the fact that it relies only on the past speech packets/states information in the prediction process, without taking into account the future speech packets even if there are available. The complexity of the model is another problem of this method, where the receiver has to calculate all state probabilities to decide on the chosen one. Note that this processing will be even useless if state probabilities are too close, in which the receiver will have to restart the recovery process using the repetition approach.

In [12], the authors proposed an HMM-based PLC to track the evolution of speech signal parameters and to determine the most probable signal state at each time instant. The proposed HMM-based PLC analyses packet parameters such as spectral envelope, pitch, energy, and degree of voicing. In the case of speech packet loss, the model uses the available states information and probability density functions to estimate the lost speech parameters. Thus, the estimation phase is applied before signal synthesis using the MMSE approach. The used feature vector in this approach is very sensitive to pitch estimation errors like doubling or halving periods [15] which induce relatively high estimation errors in voiced/unvoiced transition segments.

The authors in [15] proposed a vocoder-independent HMM model to avoid pitch estimation sensitivity. The model analyzed the decoded speech packets to estimate a pre-defined feature vector ϕ_t that tracks the evolution of the speech signal. This vector contains different speech characteristics such as the power indicator, the spectrum description, and the voicing information. In case of speech packet loss, the HMM model injects this estimated vector in the input of the speech synthesizer to produce the concealed speech portion.Then, the lost portion of the signal will be replaced by the lost one using (Overlap and add) technique to smooth the output signal and reduce the speech discontinuities. The test results of this model were not satisfactory, since it did not exceed the results of the G711 PLC at any loss rate. This method is very useful in online speech recognition systems, where packet loss degrades considerably the speech recognition accuracy.

In [16], the authors proposed an HMM-based approach to reconstruct the spectral speech components using implicitly quantizing spectrographic data. In this approach, The feature trajectories are interpreted as transitioning through an HMM-defined trellis. The authors, in [17, 18] presented an HMM-based spectral components reconstruction of unreliable speech segments using the statistical distributions of the clean components. This approach models the temporal patterns in speech signal as a series of HMM state transitions and takes advantage of corre-

lations between speech features in both time and frequency domain simultaneously to reconstruct the damaged spectral components. Unfortunately, both of these two approaches can not be applied in PLC for VoIP communication where the speech segments are completely lost rather than damaged.

While the cited HMM-based PLC methods have many advantages and provide a significant improvement of the perceptual speech quality in VoIP communications over traditional PLC methods, they still induce a set of problems that are mainly caused by the structure and the assumptions considered in the HMMs. For instance, in an HHM, the emission of a particular symbol at a time instant t depends only on the current state, which makes the probability of generating a symbol completely independent of the surrounding states. Moreover, the observed packets in HMM have no one-to-one correspondence with each other, which means that the emissions are linked to states through the probability distribution without any direct relation between them. Even if these assumptions are very useful in many domains (like speech recognition) where they are used to decrease the complexity of the model, we argue that it cannot be accepted in PLC due to the pseudo-stationarity nature of speech signals. In this paper, we assess the impact of these two assumptions on the perceived quality obtained by HMM-based PLC methods. Then, we propose a new hybrid PLC model based on both HMM and DT architecture that take into consideration the surrounding states and emissions in the estimation process of the lost speech packets. The proposed model circumvents the cited shortcomings of HMM-based PLC and increases the accuracy of lost speech packet estimation.

The remainder of this paper is organized as follows. In Section 2, we present a brief introduction of the G.722.2 codec, where we detail the impact of speech packet loss on the performance of this codec. In Section 3, we discuss how HMM can be adapted to the PLC field. Then, we detail the different Markov assumptions that conduct this model. After that, we discuss the impact of these assumptions on the performance of HMM-based PLC methods. In Section 4, we present a brief introduction to the DT model and how it can be combined with HMM. In Section 5, we present our proposed PLC method and detail the implementation process on the G.722.2 codec. In Section 6, we show and discuss the simulation results.

2 G.722.2 Overview

Speech coding is an indispensable tool for resourceful and efficient voice communications through IP switched networks. CELP coders, in particular, provide a good compromise between bandwidth consumption and perceptual speech quality. Adaptive Multi-Rate Wideband (AMR-WB) G.722.2 codec is an ITU-T standard speech codec developed based on Adaptive Multi-Rate encoding, using the Algebraic Code Excited Linear Prediction (ACELP) approach. The AMR-WB speech codec consists of nine-bit rates. The default-framing interval for the G.722.2 codec is 20ms. Each packet includes several speech features: Voice Activity Detector (VAD), Immittance Spectral Pair (ISP), Adaptive Codebook Index (ACI), Codebook Index (CI) and Vector Quantification Gain (VQG)[5, 19]. Fig. 1 shows the bit allocation of G.722.2 speech packet at 6.6 kbit/s bit rate and its different speech parameters.

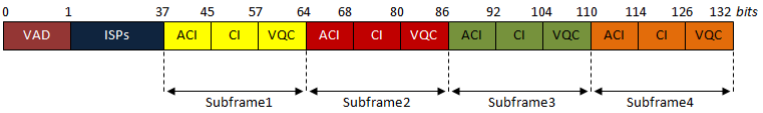

Fig. 1 Source encoder output parameters in their order of occurrence and bit allocation within the speech frame of 132 bits/20 ms, 6.60 kbit/s mode

However, this codec is known to be sensitive to bit errors and packet losses due to inter-packet dependencies during the decoding phase [19]. Lost packets not only cause the annoying artifacts at the lost segments level but also cause the decoding error propagation in the subsequent speech packets of this loss. Following a packet loss, the decoder takes a finite time (the convergence time) to resynchronize its state to that of the encoder. The depth of error propagation can reach 80ms after the loss [5]. This duration depends on the lost segment nature (silence, voiced, unvoiced) and the number of consecutive losses [5]. This problem becomes more severe in case of bursty losses and can lead to a total disconnection between the encoder and decoder. The error propagation problem is mainly caused by the decoding inter-packet dependencies of the G.722.2 codec. For example, The G.722.2 decoder uses an LP filter to convert the ISP vectors to LP coefficients that will be used to synthesize the speech signal. The order of the LP filter is $m = 16$, which means that the decoder needs the last 16 ISP vectors in order to compute the new one [20]. A similar problem can be found in the noise enhancement block where the total signal excitation is computed using the mentioned LP coefficients and the fixed codebook gain parameters of the previous sub-packet. Therefore, in order to decrease/eliminate the decoding error propagation of subsequent speech packets, not only an efficient PLC method has to enhance the perceived speech quality of lost speech segments but it has to update the decoder state as well [19, 20].

3 Hidden Markov Model (HMM) and PLC

3.1 Introduction to HMM

HMM is a powerful statistical tool that allows modeling data with sequential correlations in neighboring samples, such as in time-series data [21, 22]. It has been successfully adapted in several complex topics such as speech recognition, optical character recognition, computational biology for its robust statistical foundation, conceptual simplicity, and malleability [22, 23].

HMM is a type of Markov chain that contains hidden nodes called states (S), where $S \in (s_1, s_2, ..., s_N)$, N is the number of states in the model and observed parameters called emissions (or observations) (Φ) where $\Phi \in (\phi_1, \phi_2, \phi_3, ..., \phi_O)$, O is the number of observed emissions in the model. From an observer's perspective, only the observed emissions can be viewed and assumed to be generated after each state transition. A stochastic process is then used to identify the existence of the hidden states and their characteristics by observing and analyzing the emission data series. Clearly, HMM is a discrete-time process. In this process, the future

state s_{t+1} is conditionally independent of the past state s_{t-1} given the current one s_t [21]. HMMs are conducted by both transition and emission probabilities.

– The transition probability $p(s_t = n|s_{t-1} = m)$ is the probability of moving from the previous state s_{t-1} to the current state s_t , where $n, m = 1, 2, 3, ..., N$. The set of transition probabilities forms the transition matrix A where $A(n, m) = a_{nm}$ is the probability of moving from the previous state $s_{t-1} = m$ to the current state $s_t = n$.

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1N} \\ a_{21} & a_{22} & ... & a_{2N} \\ . & . & . & . \\ a_{N1} & a_{N2} & ... & a_{NN} \end{bmatrix} \tag{1}$$

Due to standard stochastic constrains:

$$0 \leq a_{nm} \leq 1 \quad , \quad \forall n, m \tag{2}$$

$$\sum_{m=1}^{N} a_{nm} = 1 \quad , \quad \forall n \tag{3}$$

– The emission probability is the probability of emitting a particular symbol after each state transition $p(\phi_t|s_t = n)$.

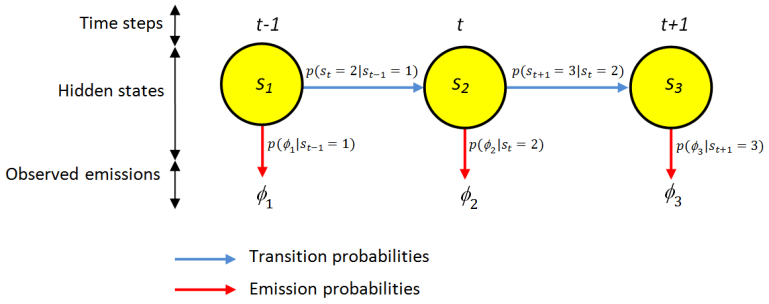

Fig. 2 An example of three HMM state transitions

Fig. 2 shows a sequence of three HMM states $s_{t-1} = 1$, $s_t = 2$ and $s_{t+1} = 3$, where we can see the hidden states and the transitions between them at each time step; Additionally, we can see how the observed emission vectors are generated after each state transition as well as the the conditional probability functions that conduct the model.

3.2 HMM based PLC

HMM has been extensively used in speech recognition and speech modeling despite the limited interest surrounding it in the literature.. In this subsection, we assess how HMM can be adapted and used in PLC to enhance the perceptual quality of

VoIP communications. In VoIP, a speech packet is sent each time step (generally each 10 to 30ms) from the encoder to the decoder. these speech packets contain the speech features of the coded speech signal segment as shown in Fig. 3.

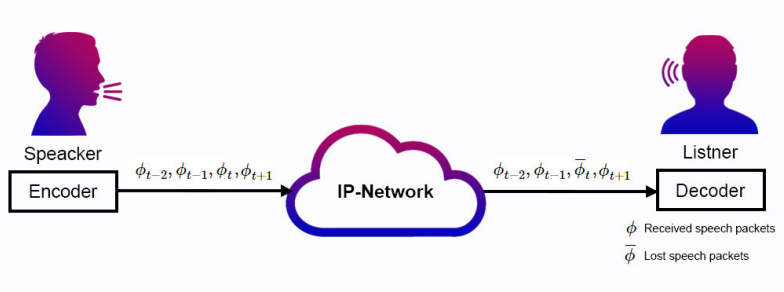

Fig. 3 General structure of VoIP communication

Thus, the receiver can consider each received speech packet as an observed emission vector ϕ_t that has been generated from a particular speech state s_t using a discrete stochastic HMM process, which progresses through a series of states with discrete probability density functions (pdfs) [21]. In such a system, a state transition is made each time step, which will generate a new speech packet. Consequently, the receiver can track the evolution of speech signals by analyzing the sequence of received speech vectors through the HMM [12]. By applying HMM to PLC when a speech packet is lost $\overline{\phi}$, the HMM-based PLC can estimate the missed speech vector employing the pdfs of state transition and emission distribution probabilities. Then, the estimated speech vector $\hat{\phi}$ will be used as a replacement for the lost one, which will enhance the perceptual speech quality of the VoIP communications.

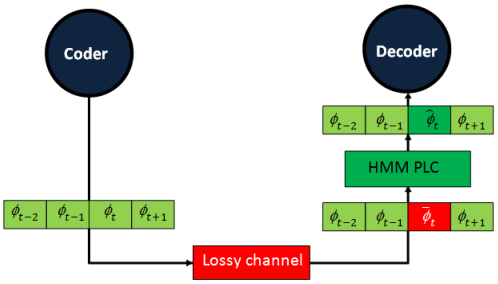

Fig. 4 HMM based PLC

Fig. 4 shows a scenario of HMM-based PLC, where 4 speech packets $\phi_{t-2}, \phi_{t-1}, \phi_t$ and ϕ_{t+1} are sent by the coder at each time step through a lossy channel. in this scenario, we assume that the speech packet ϕ_t is lost. Thus, HMM-based PLC

has to provide an estimation of the lost speech packet $\hat{\phi}_t$ that will be used as a replacement of the lost one.

3.3 HMM basic assumptions

In this subsection, we discuss the basic assumptions that conduct and control HMM. These assumptions define the basic statistical relation between the hidden states and the dependencies in the observed data. HMMs offer a computationally reasonable way to model complex time-varying processes like speech; however, they rely on a set of assumptions that could be inaccurate, when it is used in an HMM-based PLC:

- Markovian assumption: In HMM, the transition probability of moving from the current state s_t to the next state s_{t+1} depends only on the current state s_t. Consequently, we consider that:

$$p(s_{t+1}|s_1, ..., s_t) = p(s_{t+1}|s_{1:t}) = p(s_{t+1}|s_t) \tag{4}$$

  'Given that speech signal is quasi-stationary only in short analysis windows of 5 to 40ms duration [24]. The speech packet duration for most speech codecs, like G.722.2 codec, is 20ms. Therefore, the dependencies between speech packet features are respected and preserved in only two consecutive speech packets (40ms of total duration). If we consider that a speech packet is generated by an HMM state, then we can say that the speech features are preserved in only two consecutive HMM states which is the case in the Markovian assumption.
- HMM states independencies: In HMM, the individual states are conditionally independent of each other [25]. Which means that the probability of having a sequence of three HMM states $s_{t-1} = 1$, $s_t = 2$ and $s_{t+1} = 3$ as shown in Fig. 2 can be computed as:

$$p(s_{t-1} = 1, s_t = 2, s_{t+1} = 3) = p(s_{t-1} = 1, s_t = 2) \times p(s_t = 2, s_{t+1} = 3) \tag{5}$$

  Consequently, we can use a Naïve Bayes strategy to compute the joint probabilities [25].
- Emissions dependencies: In HMM, the emission of a particular symbol ϕ at time instant t depends only on the current state s_t (called the parent state) as shown in Fig. 2. Thus, the probability of generating the symbol ϕ_t is independent of the previous states $s_1, ..., s_{t-1}$ and the next states $s_{1+1}, ..., s_T$, where T is the last time step in the observed data. This means that in HMM we consider that:

$$p(\phi_t|s_1, ..., s_{t-1}, s_t, s_{t+1}, ..., s_T) = p(\phi_t|s_{1:T}) = p(\phi_t|s_t) \tag{6}$$

- Emissions independencies (independent-output assumption): In HMM, the packets/ segments in the observed data have no one-to-one correspondence with each other. The emissions in HMM are linked to states through a probability distribution .In fact, HMM is considered as a doubly stochastic process that includes a Markov chain as the basic stochastic process which describes state transitions, and other stochastic processes that describe the statistical correspondence between the states and the observed data [26, 27]. Thus, the

emission of the observed data process depends only on the parent state of the generated symbol, as given in Equation 6. Consequently, the emissions are conditionally independent of each other. In HMM, we consider that :

$$p(\phi_t|s_1, ..., s_{t-1}, s_t, s_{t+1}, ..., s_T, \phi_1, ..., \phi_{t-1}, \phi_{t+1}, ..., \phi_T) = p(\phi_t|s_t) \quad (7)$$

3.4 Discussion:

The last two cited assumptions are useful in many fields such as computational biology and speech recognition despite being considered as "Good enough" rather than strictly true [12, 25]. However, in other fields like PLC, it does not reflect the true nature of speech signal, which at any instant is more-or-less correlated with preceding and following events [28, 29]. Consequently, these assumptions cause several annoying perceptual artifacts like clicks, gain mismatching, pitch fluctuation and phase jumps due to the lack of direct dependencies between the consecutive emissions (the emissions independencies assumption) and the lack of direct dependencies between the emissions and the surrounding states (the emissions dependencies assumption), as will be shown in the simulation results section. This particular problem has not been assessed in the HMM-based PLC methods. Most studies in this field use an averaging approach of the possible emissions generated by a state s_t to estimate the lost speech packets at this time instant, without taking into consideration the dependencies between the neighboring emissions and states. In this paper we combine the HMM with the DT process in order to circumvent this particular problem and to increase the performance of HMM-based PLC methods. In the next section, we present an introduction of the DT process and how it can be integrated into the HMM.

4 DT model and HMM

DT models are regression or classification models that are based on a nested decomposition of the input space [30]. DTs structure includes a root node, branches, intermediate nodes, and leaves. This hierarchical structure allows DTs to determine the likelihood of observation by asking a series of questions about the observed data. These questions are asked at question nodes, starting at the root node, which is the main node of the tree. The answer to this question will select the path to the next child node. This process is repeated until we reach the terminal node (called the leave) that takes the final decision of this process.[31]. Fig. 5 shows an example of DT structure.

DTs have several advantageous properties, such as handling high dimensional spaces gracefully without imposing restrictions on the number or type of data [28]. Also, the decision process is extremely fast due to their hierarchical structure [32], for example, if we ask a question about the gender of the user, the answer of this question selects a path that depends only on the chosen answer and ignores all the other branches (paths) of the DT. Additionally, DTs can deal with multiple target features at the same time [31], which makes it very useful in many fields

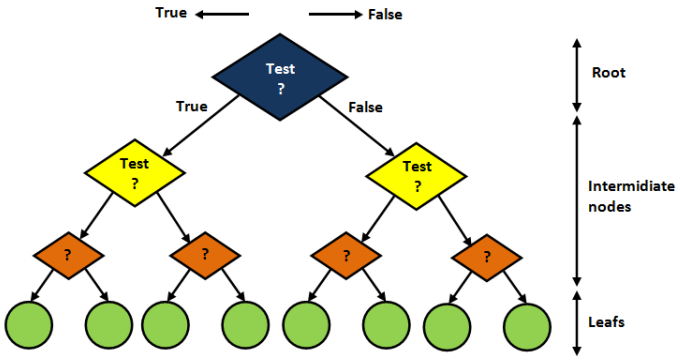

Fig. 5 DT structure

such as character recognition, associative searching and natural language modeling [28]. Another reason for DTs popularity is their prediction accuracy, which allows them to be used in several critical domains, such as robotics, medical diagnosis, and credit risk assessment [33].

DTs can be associated with HMM to determine the state likelihood by asking a series of questions about the features of the observed data [31]. The answers to these questions chose a particular path through the different nodes of DT until reaching the leave node, which provides the likelihood of the observation given the HMM state [32]. This kind of hybrid model is called HMDT. Fig. 6 shows a representation of HMM and HMDT models.

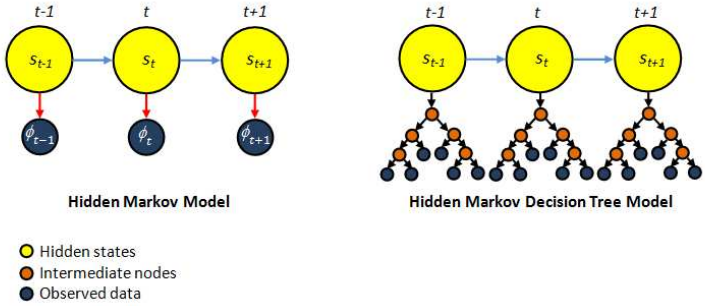

Fig. 6 Comparison between HMMs and HMDT models

This Hybrid architecture is used in speech recognition [28, 31, 32]. However, the purpose and the context of this researches are completely different from PLC. In speech recognition context, DTs trees are treated as Probability Estimation Trees (PETs) [28] that estimate the likelihood of the observed speech segment with a certain phoneme, in contrary to PLC methods that try to estimate a lost speech segment using the surrounding speech segments and/or a pre-trained speech model.

Unfortunately, HMDT models have not been used in PLC except for [34], where the authors propose a model to evaluate the impact of packet loss on the perceptual speech quality of VoIP communications. In this paper, we tried to adapt the HMDT to PLC context to circumvent the independent-output Markovian assumption and to create a content-dependent model that maximizes the dependencies between the surrounding received speech packets and the estimated ones. This method was implemented and tested on the G.722.2 codec.

5 Proposed HMDT PLC method

In this paper, we propose a new approach to the PLC method based on HMM and DT architecture. The proposed method is implemented and tested on the G.722.2 codec which is very sensitive to packet loss due to its inter-packet dependencies (as shown in Section 2). Thus, the proposed method will enhance the perceived speech quality of VoIP communications under bursty packet loss conditions by estimating the lost speech packets. These estimated speech packets are used as a replacement for the lost ones in the decoding process. They are also used to update the decoder state which will minimize the convergence time, and decrease the decoding error propagation of the following speech packets.

As explained in section 3, HMM is considered as a doubly stochastic model with two kinds of variables: Hidden states S and observed data Φ. Thus, the estimation of the lost speech packets process can be divided into two problems: the first is estimating the state s_t that generates the lost speech packet $\overline{\phi}_t$, and the second is estimating the emission speech vector (speech packet) $\hat{\phi}_t$ generated by the estimated state. In this setting, the proposed method will first estimate/predict the HMM state that generates the lost speech packet through the HMM state estimation block, using the conditional probability density functions of the missing speech packet. Once this step is done, the proposed method will estimate the lost speech packet produced by the estimated state using the DT block to circumvent the emission independencies (detailed in Section 3)) as well as to create a content-dependent model that maximizes the dependencies between the surrounding received speech packets and the estimated ones . This last step will ensure a naturally smooth transition between the received speech segments and the estimated ones.

Fig. 7 shows the block diagram of the proposed PLC method. In this figure, we can distinguish two cases:

- Speech packet is correctly received: in this case, the HMM speech tracking block uses the received speech packet ϕ_t to track the statistical evolution of speech signal through the HMM states. The resulting tracking information will be shared with the HMM state estimation block and the DT block. Also, the received speech packet is sent into the decoder input to be synthesized.
- Speech packet is lost: in this case, the HMM state estimation block estimates/predicts the HMM state s_t that generate the lost speech packet $\overline{\phi}_t$. Then, this information will be shared with the DT block. This later provides an estimation of the lost speech packet $\hat{\phi}_t$ by taking in consideration: the estimated state $\hat{s}_t$ (provided by the HMM state estimation block), the previous HMM state s_{t-1}, the previous emission ϕ_{t+1}, as well as the next state s_{t+1}, the

next emission ϕ_{t+1} if they are available. Finally, The estimated speech packet $\hat{\phi}_t$ will then be sent to the decoder as a replacement for the lost speech packet, to the HMM state estimation for updating purposes, and to the decoder buffer to update the decoder state.

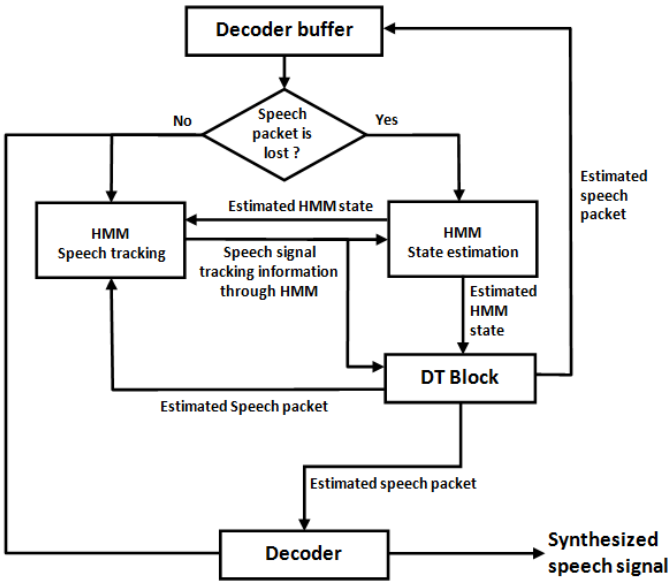

Fig. 7 The block diagram of proposed PLC method

5.1 Block diagram of the proposed method:

As explained in Fig.7,the proposed PLC process can be divided into three blocks:

- The HMM speech tracking block: to track the statistical evolution of received speech signal through a series of HMM state transitions.
- The HMM state estimation block: to estimate the HMM state of lost speech packets.
- The DT block: to estimate the lost speech packets.

The remainder of this subsection will be dedicated to detailing the role of each block.

5.1.1 HMM speech tracking block

This block tracks the evolution of received speech signals through a series of HMM state transitions, where each received speech packet is considered as a speech vector ϕ_t generated by a particular HMM state s_t at the time instant t. the proposed method is trained and implemented on the G.722.2 codec. We consider that a

speech packet is sent each 20ms over a packet lossy channel; and each speech packet contains multiple speech parameters that represent the features of speech signal during this period, like the Immittance Spectral Pair (ISP) and Adaptive Codebook Index (ACI)... as explained in Section 2. Also, we consider that each speech vector ϕ_t can be generated by only one HMM state s_t (more details are provided in Sub-section 5.4).

For example, let's consider the pre-trained HMM shown in Fig. 8. This HMM consists of three ($N = 3$) HMM states $S \in \{s_1, s_2, s_3\}$ and nine ($O = 9$) possible emission speech vectors $O \in \{\phi(1), \phi(2), \phi(3), \phi(4), \phi(5), \phi(6), \phi(7), \phi(8), \phi(9)\}$. And we consider that each emission speech vector can be generated by a particular HMM state, as shown in Fig. 8.

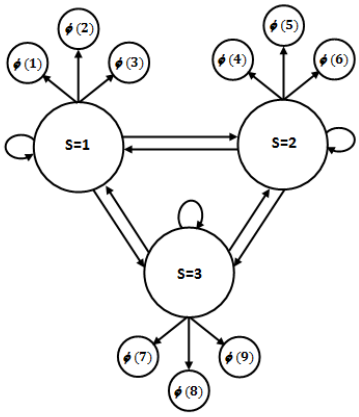

Fig. 8 Example of pre-trained HMM for PLC context

Then, we can model/supervise the evolution of speech signal through a series of state transition. For example let's consider that we received the following sequence: $\phi(9) \rightarrow \phi(2) \rightarrow \phi(5) \rightarrow \phi(3) \rightarrow \phi(7) \rightarrow \phi(2)$ at different time steps. The received speech packets are considered as emissions of particular HMM states. Thus, this speech segment can be modeled as a series of state transition using the pre-trained HMM shown in Fig. 8: $s_{t-4} = 3 \rightarrow s_{t-3} = 1 \rightarrow s_{t-2} = 2 \rightarrow s_{t-1} = 1 \rightarrow s_t = 3 \rightarrow s_{t+1} = 1$, as shown in Fig. 9.

5.2 HMM state estimation block

In case of speech packet loss, this block predicts/estimates the state $\hat{s}_t$ that generates the lost speech packet $\bar{\phi}_t$ through the HMM, using the conditional transition and emission probabilities that have been trained offline and the state of the model obtained by the HMM speech tracking block. This process can be decomposed into two approaches according to the available data.

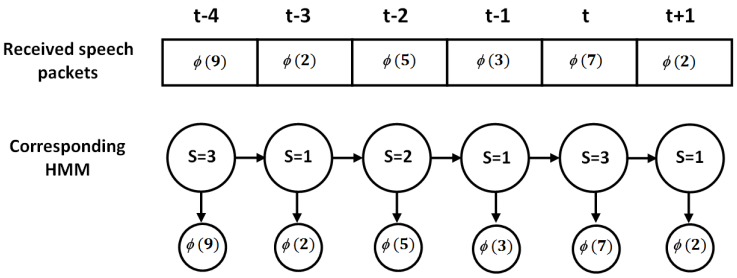

Fig. 9 Illustration of speech tracking through the HMM

5.2.1 The prediction approach:

In this approach, the proposed model has to predict the HMM state s_t that generates the lost speech vector $\overline{\phi}_t$ by analysing only the past emissions $\phi_1, ..., \phi_{t-1}$. This approach is useful in the case of bursty losses where speech packets that follow the lost one are also lost. It is also used, in the case where the decoder has to estimate the lost speech packet and cannot await the arrival of the following ones due to real-time constraints of VoIP applications. Thus, we have to determine first the marginal probability that a missing parameter vector at time t is produced by an HMM state n based only on the past observations vectors. This can be expressed as:

$$p(s_t = n|\phi_1, ..., \phi_{t-1}) = p(s_t = n|\phi_1^{t-1}) \tag{8}$$

Using the Bayes rule, equation 8 can be written as:

$$p(s_t = n|\phi_1^{t-1}) = \frac{p(s_t = n, \phi_1^{t-1})}{p(\phi_1^{t-1})} \tag{9}$$

Given that, $p(\phi_1^{t-1})$ is constant since all the previous emissions are already known and consequently can be ignored because they do not interfere on the on the marginal probability. Equation 9 can be written as

$$p(s_t = n|\phi_1^{t-1}) = \frac{1}{C} \times p(s_t = n, \phi_1^{t-1}) \approx p(s_t = n, \phi_1^{t-1}) \tag{10}$$

Then we used the low of total probability to marginalize the state $s_t = n$ by summing over all the other states. Equation 10 can be written as:

$$p(s_t = n|\phi_1^{t-1}) = \sum^{m} p(s_t = n, s_{t-1} = m, \phi_1^{t-1}) \tag{11}$$

Using the Chain rule, we can write Equation 11 as: $\mathrm{p}(\mathrm{s}_t = n|\phi_1^{t-1}) = \sum^m p(\phi_{t-1}|s_{t-1} = m, s_t = n, \phi_1^{t-1}) \times p(s_t = n|s_{t-1} = m, \phi_1^{t-1}) \times p(s_{t-1} = m, \phi_1^{t-1})$ Given that emissions in HMM depend only on the parent state at the time instant t as explained in Equation 12:

$$p(\phi_t|s_t, s_{t-1}, ..., s_1, \phi_{t-1}, ..., \phi_1) = p(\phi_t|s_t) \tag{12}$$

The term $p(\phi_{t-1}|s_{t-1} = m, s_t = n, \phi_1^{t-1})$ in equation 5.2.1 can be rewritten as:

$$p(\phi_{t-1}|s_{t-1} = m, s_t = n, \phi_1^{t-1}) = p(\phi_{t-1}|s_{t-1} = m) \tag{13}$$

Similarly, the states in HMM are independent from the previous emissions and the unrelated states as explained in 14

$$p(s_t|s_{t-1},...,s_1,\phi_{t-1},...,\phi_1) = p(s_t|s_{t-1}) \tag{14}$$

The term $p(s_t = n|s_{t-1} = m, \phi_1^{t-1})$ in equation 5.2.1 can be rewritten as:

$$p(s_t = n|s_{t-1} = m, \phi_1^{t-1}) = p(s_t = n|s_{t-1} = m) = a_{nm} \tag{15}$$

Where a_{nm} is the transition matrix from state m to state n. Now, using the Forward algorithm the term $p(s_{t-1} = m, \phi_1^{t-1})$ can be expressed as:

$$p(s_{t-1} = m, \phi_1^{t-1}) = \alpha_{t-1}(m) \tag{16}$$

By replacing the equations (13), (15) and (16) in equation 5.2.1 we find:

$$p(s_t = n|\phi_1^{t-1}) = \sum^{m} p(\phi_{t-1}|s_{t-1} = m) \times a_{nm} \times \alpha_{t-1}(m) \tag{17}$$

Now, we simply introduce the loss index k to equation 17:

$$p(s_t = n|\phi_1^{t-1}) = \sum^{m} p(\phi_{t-1}|s_{t-1} = m) \times a_{nm} \times \alpha_{t-1}(m) \tag{18}$$

Fig. 10 shows an illustration of the prediction approach in a scenario that considers two consecutive lost speech packets.

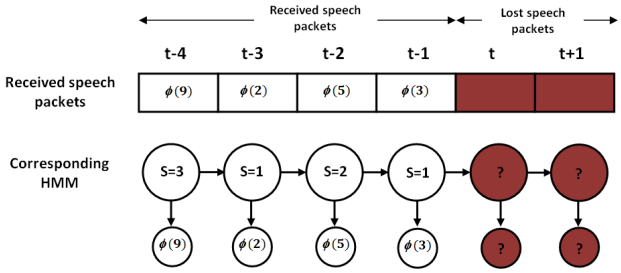

Fig. 10 Illustration of Prediction approach

5.2.2 The estimation approach:

In this approach, the proposed model has to estimate the HMM state that generates the lost speech packet $\overline{\phi}_t$ by analyzing the previous speech packets $\phi_1, ..., \phi_{t-1}$. and the following speech packet ϕ_{t+1}. Fig. 11 shows an illustration of the estimation approach in a scenario that considers an isolated speech packet loss.

Thus, we have to compute:

$$p(s_t = n|\phi_1, ..., \phi_{t-1}, \phi_{t+1}) = p(s_t = n|\phi_1^{t-1}, \phi_{t+1}) \tag{19}$$

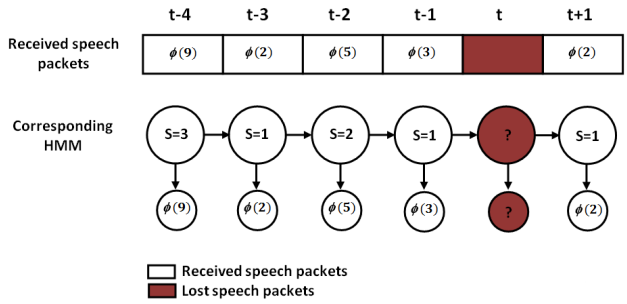

Fig. 11 Illustration of Estimation approach

Using Bayes rule, Equation 19 can be rewritten as:

$$p(s_t = n|\phi_1^{t-1}, \phi_{t+1}) = \frac{p(s_t = n, \phi_1^{t-1}, \phi_{t+1})}{p(\phi_1^{t-1}, \phi_{t+1})} = \frac{1}{C} \times p(s_t = n, \phi_1^{t-1}, \phi_{t+1}) \quad (20)$$

Giving that $p(\phi_1^{t-1}, \phi_{t+1})$ is constant because all the previous and following emissions are already known and consequently can be ignored as they do not interfere on the marginal probability. Equation 20 can be rewritten as:

$$p(s_t = n|\phi_1^{t-1}, \phi_{t+1}) \approx p(s_t = n, \phi_1^{t-1}, \phi_{t+1}) \quad (21)$$

Using Bayes rule, we can rewrite Equation 21 as:

$$p(s_t = n|\phi_1^{t-1}, \phi_{t+1}) = p(s_t = n, \phi_1^{t-1}) \times p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) \quad (22)$$

The term $p(s_t = n, \phi_1^{t-1})$ is already calculated in the previous subsection (See equation 10 , 18). Now, we can rewrite the second term in equation 22 using the Markov assumption presented in Equation 14 as:

$$p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) = p(\phi_{t+1}|s_t = n) \quad (23)$$

Using the low total probability, Equation 23 can be rewritten as:

$$p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) = \sum^{m} p(\phi_{t+1}, s_{t+1} = m|s_t = n) \quad (24)$$

Given that:

$$p(A, B|C) = \frac{p(A, B, C)}{P(C)} = p(A|B, C) \times \frac{p(B, C)}{p(C)} = p(A|B, C) \times p(B|C) \quad (25)$$

Equation 24 may be rewritten as,

$$p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) = \sum^{m} p(\phi_{t+1}|s_{t+1} = m, s_t = n) \times p(s_{t+1} = m|s_t = n) \quad (26)$$

This equation can be simplified using the Markov assumption shown in equation 12:

$$p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) = \sum^{m} p(\phi_{t+1}|s_{t+1} = m) \times p(s_{t+1} = m|s_t = n) \quad (27)$$

The left hand side of Equation 27 can be calculated using the Backward algorithm as:

$$p(\phi_{t+1}|s_{t+1} = m) = \beta_{t+1}(m) \tag{28}$$

Being the transition matrix a_{mn} between the two states $s_{t+1} = m$ and $s_t = n$, the right hand side in in equation 27 may be expressed as:

$$p(\phi_{t+1}|s_t = n, \phi_1^{t-1}) = \sum^{m} a_{mn} \times \beta_{t+1}(m) \tag{29}$$

After computing the marginal probabilities of $s_t = n$ using the estimation approach or the prediction approach, we can find out the HMM state that generates the lost speech packet by maximizing the expression $argmax(n) \quad p(s_t = n|\phi_1^{t-1}, \phi_{t+1})$ or $argmax(n) \quad p(s_t = n|\phi_1^{t-1})$ according to the used approach.

At this point, we estimated/predicted the state n at the loss instant t. However, this is not sufficient since an estimation of the parameter vector (the lost speech packet) $\hat{\phi}_t$ that will be produced by this state is also required. This can be done using only HMM [12]:

$$p(\phi_t|\phi_1^{t-1}, \phi_{t+1}) = \sum_n p(\phi_t, s_t = n|\phi_1^{t-1}, \phi_{t+1}) = \sum_n p(\phi_t|s_t = n) \times p(s_t = n|\phi_1^{t-1}, \phi_{t+1}) \tag{30}$$

However, if we analyze (30), we can see that (a) the emission at time t doesn't depend on the previous or next emissions and (b) the emissions depend only on the parent state $s_t = n$ [27]. This fact is verified by the Markov assumption shown in equation 12. In practice, we cannot accept that the emission at the instant t to be completely decorrelated from the past and future emissions due to the short-time stationarity feature of the speech signal. Therefore, we use the DT block to estimate the lost speech vector $\hat{\phi}_t$.

5.3 DT Block

After estimating/predicting the state s_t of the lost speech vector $\overline{\phi}$, the role of the DT block will be to figure out the speech vector $\hat{\phi}_t$ produced by the estimated state. The estimation process adopts the DT architecture to circumvent the HMM independent-output assumption (detailed in Section 3) as well as to add context-dependent information to the model. In this setting, a DT(n) will be assigned to each state of the proposed HMM. In the case of speech packet loss, the DT(n) will estimate the lost speech vector that should be produced by the estimated HMM state $s_t = n$ according to the previous speech packet ϕ_{t-1} and the following speech packet ϕ_{t+1} (if it is available). The decision process is performed by asking a series of questions to select the appropriate speech vector $\hat{\phi}_t$ that will be generated by the estimated HMM state $s_t = n$ among all the possible speech vectors that can be generated by this state. The DTs are pre-trained offline which means that the estimation process is very fast and doesn't need any further computing.

Fig. 12 shows the general diagram of the proposed DT. The series of questions in the DT block are:

– The first question in $DT(n)$ is: Amongst all the speech vectors that can be generated by the HMM state $s_t = n$, which are the generated speech vectors when

the previous HMM state $s_{t-1} = m$? Then, $DT(n)$ selects the intermediate node belonging to the previous HMM state s_{t-1}.
- The second question: Amongst all selected speech vectors in the previous node, which are the generated speech vectors when the previously observed speech vector is ϕ_{t-1}? Then, $DT(n)$ selects the intermediate node belonging to the previous speech vector ϕ_{t-1}.
- The third question: Is the next HMM state information available? if not (which means that we are in the prediction approach), the $DT(n)$ will generate directly the pre-computed speech vector that is considered as a leave node. Otherwise, the estimation approach is adopted (the next speech packet ϕ_{t+1} is correctly received).
- The fourth question: If the answer to the third question is yes, amongst all selected speech vectors in the previous node, which are the generated speech vectors when the next HMM state $s_{t+1} = l$? Then, $DT(n)$ selects the intermediate node belonging to the next HMM state s_{t+1}.
- The fifth question: Amongst all selected speech vectors in the previous node, which are the generated speech vectors when the next observed speech vector is ϕ_{t+1}? Then, $DT(n)$ selects the intermediate node belonging to the next speech vector ϕ_{t+1}.

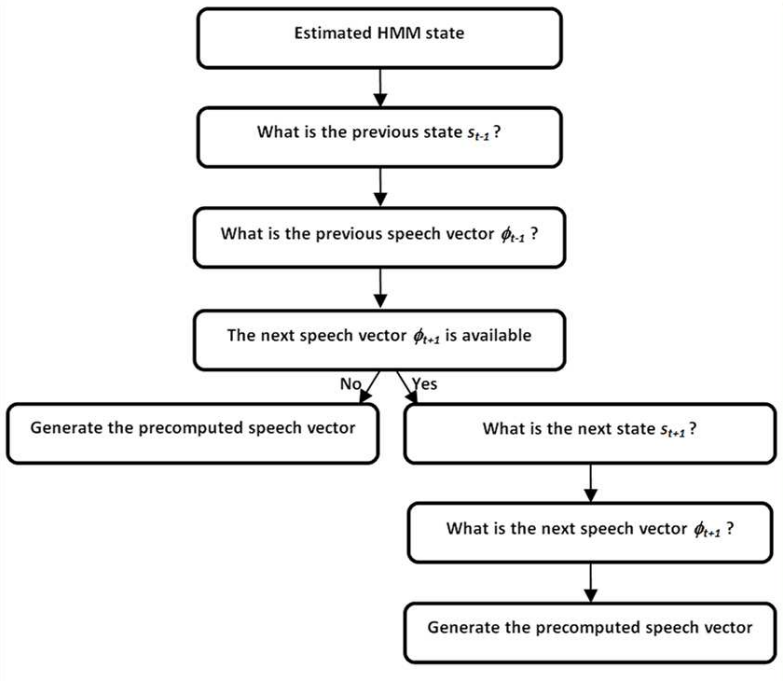

Fig. 12 Proposed DT block diagram

The different DTs of the Proposed HMM model are constructed and trained off-line. Each state of the model s_n has its $DT(n)$. Each $DT(n)$ is trained to

distinguish and analyze only speech packets that correspond to s_n (positive speech vectors) and ignore the others (negative speech vectors). All the speech vectors that belong to a particular HMM state s_n are considered as the leaves of a $DT(n)$. These leaves are classified according to the neighboring HMM states and their emissions. Thus, the construction of the initial DTs is performed in a top-down manner using a greedy strategy. Then, the obtained initial DTs are refined using the pruning approach to reduce their complexity. Finally, HMM parameters are re-estimated after the pruning stage using the backword-forward algorithm. The details of the HMM training and DTs construction are presented in Section 5.4.

The DT architecture will allow the HMM to:

–
– Circumventing the HMM independent-output assumption by adding context-dependent information to the model, will allow for the estimation of the parameter vector $\hat{\phi}_t$ that should be produced by the estimated state $s_t = n$ according to neighboring emissions ϕ_{t-1} and ϕ_{t+1}.
– Since the proposed model contains several states that each one of them can generate multiple speech vectors, The DTs allow for the handling of these high-dimensional spaces gracefully, essentially by ignoring all the following branches, which do not meet the criteria required at the intermediate nodes of the DT.
– Due to the hierarchical structure of DTs, finding a tree-based output parameter vector ϕ_t probability given the input state vectors is extremely fast, which is required in real-time applications such as VoIP. DTs provides an easy and dynamic structure updating which allows the model to be adapted to the speaker as well as to decrease the dependence of the training set.

5.4 Training phase details

5.4.1 HMM Training

The training phase is the key to HMM success. in this stage, we use a large database to estimate the HMM parameters like the state priors $p(s_t = n)$, the transition matrix A, and the conditional emission probabilities $p(\phi_t|s_t = n)$. To do that, we regrouped all the speech files available in the TIMIT database in one speech file of 4 hours duration. The resulting speech file contains 6312 sentences produced by 662 male and female speakers of different ages and accents. This speech file was coded by the G.722.2 codec at a 6.60 kbit/s bit rate resulting in a binary file that contains more than 720.000 speech packets. These speech packets are regrouped into 270 initial states using the k-means clustering algorithm [35]. Following this procedure, we have made a crucial assumption.: "the states are treated as being observable, and during training, we know the exact state from which each speech packet originates [27]. This is a common approximation in the speech processing field since a voiced state cannot produce an unvoiced parameter vector, and a silence state cannot produce a parameter vector that represents a speech event. We then used the memory-optimized Forward-Backward algorithm [33] to train the model and estimate the initial HMM parameters.

5.4.2 DTs construction

As detailed in the previous subsection, The DTs are constructed in a top-down manner by classifying the emissions ϕ_t of a particular state $s_t = n$ according to the neighboring HMM states (s_{t-1} and s_{t+1}) and their emissions (ϕ_{t-1} and ϕ_{t+1}). Therefore, each $DT(n)$ is trained to analyse only speech packets that corresponds to its HMM state s_n. The steps of the DTs construction are:

- Initialisation phase: DTs are constructed using a greedy strategy that turns the initial leaves of the model into intermediate nodes and produces new leaves that can be split too, which means that each DT in the model $DT(n)$ has to identify all the speech vectors produced by the HMM state s_n in the training file. Then, It has to classify these speech vectors according to neighboring HMM states (s_{t-1} and s_{t+1}) and their emissions (ϕ_{t-1} and ϕ_{t+1}) using the series of questions explained in Fig. 12. Fig. 13 shows a 2D graphic illustration of how the initial DTs are obtained. In this figure, we assume that we are constructing the $DT(3)$ that belongs to the HMM state s_3. In (A) we can see how the first node of this DT is obtained by collecting all the speech vectors that belong to the HMM state s_n in the training file. In (B) we can see how these speech vectors are divided into two categories: speech vectors that are generated by $s_t = 3$ when the previous state is $s_{t-1} = 1$, and those that are generated by $s_t = 3$ when the previous state is $s_{t-1} = 2$. Then, in (C) each obtained leave (category) in (B) is divided into two categories according to the speech vector that has been produced by the previous HMM state (s_{t-1}). In (D), each obtained leave-in (C) is divided into two categories according to the value of the next HMM state s_{t+1}. In (E), each obtained leave-in (D) is divided according to the value of the speech vector ϕ_{t+1} generated by the next HMM state s_{t+1}. At this point, we can see how we classify all speech vectors generated by the HMM state $s_t = 3$ into several sub-categories according to the neighboring HMM information, and how the initial $DT(3)$ is constructed accordingly. At the end of this stage, the total number of leaves nl in the model is approximately $nl = 20000$.
- Pruning stage: in this stage, we reduced the size of the DTs by turning the intermediate nodes into leaves on a bottom-up approach. First, we calculate the centroid of each leave obtained in the initialization phase. This means that all the selected vectors in the initial DTs leaves will be represented by one speech vector. Fig. 14 shows how the centroids are obtained from the selected speech vectors in each leave. Then, the obtained centroids are compared to each other. If they are too close and represent almost the same features, these centroids will be regrouped and we will change the structure of the DT accordingly by turning the associated intermediate nodes into leaves. Thus, we used the Perceptual Evaluation of Speech Quality (PESQ) metric [36] as a pruning threshold. The proposed pruning process will keep turning nodes into leaves (in the bottom-up approach) until the PESQ value between the original leaves and the pruned nodes becomes inferior to PESQ $<$ 4,4. Thus, the pruned speech vectors are very similar to each other, and replacing a centroid with the other one will not affect the perceptual quality of the perceived speech signal. For example, let's consider that the centroids $C(9), C(10)$ and $C(11)$, see Fig. 15(A), represent almost the same speech features. This means that if we compare these centroids

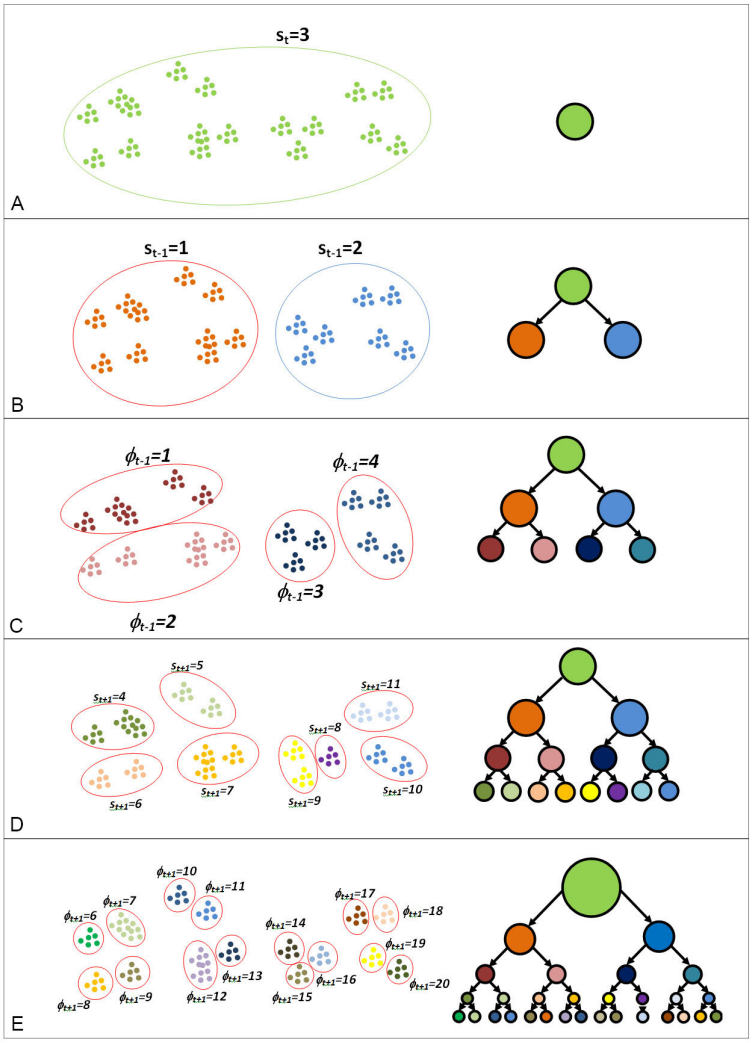

Fig. 13 Proposed implementation of DT architecture to the HMM

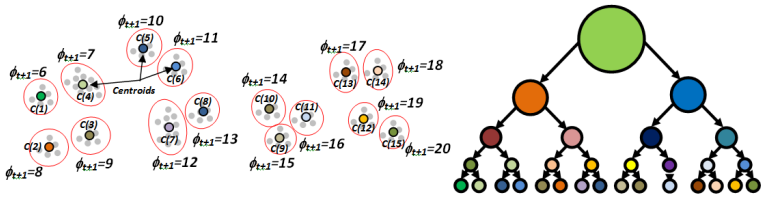

Fig. 14 Calculating the centroids of the selected speech vectors in each leave

to each other using the PESQ metric, the obtained value is more than 4,4. For that, these three centroids can be regrouped by calculating the mean of these centroids which will change the structure of the DT accordingly, As shown in Fig. 15(B).

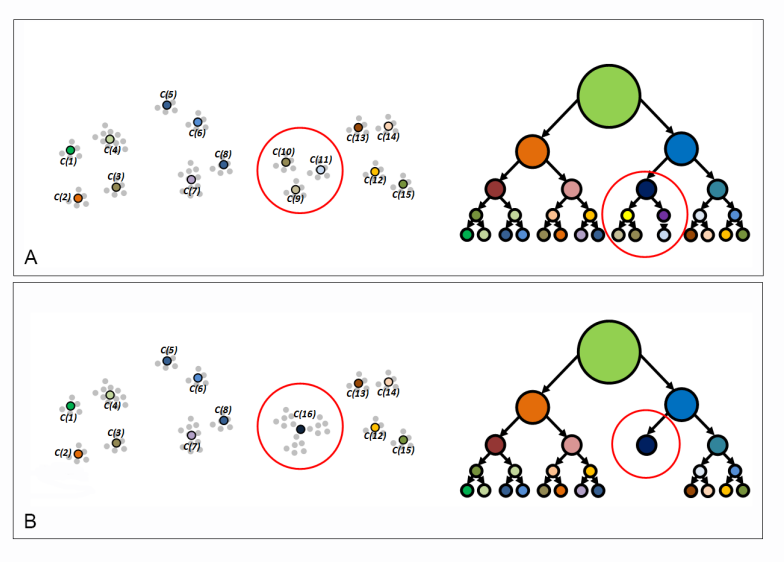

Fig. 15 Calculating the centroids of the selected speech vectors in each leave

AT this point, the proposed DTs can be seen as DT Vector Quantization (DTVQ) because it will reduce significantly the leaves number (the number of observed speech vectors in the training file) from $nl = 20000$ to $nl = 4723$ speech vectors, without reducing the perceptual speech quality of the model since the pruning process is controlled using the PESQ metric. At the end of this process, we obtain a new codebook that specifies for each speech vector in the model its corresponding state and its representing centroid.

– HMM parameter's updating stage: After the pruning stage, the HMM model was retrained using the new parameter vector codebook obtained in the pruning stage utilizing the forward-backward algorithm.

6 Simulation results

6.1 Experimental environment

To evaluate the performance of our PLC model, simulation tests were produced using a speech file of male and female speakers with different accents and ages, that has been extracted from the TIMIT test dataset. The duration of this speech file is 1 hour sampled at 16 kHz where each segment of this file comprises two sentences by the same speaker. This file was encoded using the G.722.2 codec to

obtain a binary file that contains 180.000 speech packets encoded at a 6.6 kbit/s bit rate. Table 1 shows the different learning and testing files parameters.

Table 1 Learning and testing files parameter.

Purpose	File length	Number of speech packet	Sample rate	Coding bit rate
Learning phase	04 hours	720.000	16 kHz	6.6 kbit/s
Simulation and test	01 hours	180.000	16 kHz	6.6 kbit/s

To simulate the packet loss, we erase a single speech packet of 20 ms duration and size of 135 bytes from the test binary file. The PLRs considered were 0%, 5%, 10%, 15% and 20%. For each PLR, 100 runs of the experiment were performed to simulate different packet loss patterns in the speech files. The packet loss patterns were obtained using the Extended Gilbert Model (EGM) [37]. This model allows the simulation of end-to-end burst packet losses, where each state in this model represents the number of consecutive loss packets. The maximum number of consecutive packet losses considered in our simulations is 4 which results in the loss of a speech segment of 80ms duration. Fig. 16 shows samples of loss patterns generated by the EGM for different PLRs.

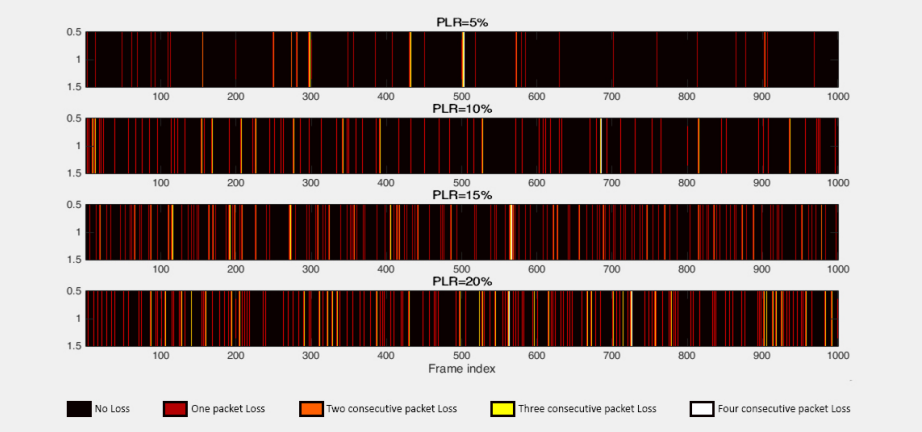

Fig. 16 Loss pattern samples for different PLRs (5%, 10%, 15%, and 20%).

To evaluate the speech quality obtained in each experiment as well as the performance of our proposed method, we used the following speech metrics:

- Perceptual Evaluation of Speech Quality (PESQ) [36]: as defined in the ITU-T P.862 standard, it is an objective metric used to evaluate the speech quality at different transmission channel conditions. This metric assesses the perceptual voice quality by comparing the original speech file with the deteriorated one to result in a value ranging from 1 (for bad speech quality) to 5 (for excellent speech quality).
- Enhanced Modified BSD (EMBSD) [38]: it is an Objective Speech Quality Measure Based on Audible Distortion and Cognitive Model. It estimates speech

distortion in the loudness domain to include only audible distortions. This metric gives a value of 0 for two identical speech files. Differently, this value increases with the level of speech segment distortion.
- Log Spectral distortion (LSD) metric [39]: it is an objective metric that measures the similarity in the magnitude spectrum between original and reconstructed signals. LSD is commonly used in the quantization of Immittance Spectral Pair (ISP), which is a form of linear prediction coefficients (LPCs).
- Short-Time Objective Intelligibility (STOI) [40, 41]: it is an objective intelligibility measure that shows the impact of speech quality degradation of the speech intelligibility.
- MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [42]: it is a subjective speech quality metric based on a listening test to evaluate the perceived speech quality. It is defined by ITU-R recommendation BS.1534-3. In our simulations, the listening tests were performed using ten listeners, In all experiments, the sample speech files were randomly presented to the listeners that evaluate the perceived speech quality by giving a score ranging from 0 (for very bad speech quality) and 100 (for excellent speech quality).
- Signal to Noise Ratio (SNR) [43]: it is an objective metric that indicates the amount of background noise present in a speech. It is defined as the ratio of signal intensity to noise intensity, expressed in decibels(dB)
- Energy Entropy [44]: it is the entropy of sub-frames normalized energies. It can be interpreted as a measure of abrupt changes. In our simulations the considered subframe length is 10ms;
- Spectral Entropy [44]: it is the entropy of the normalized spectral energies for a set of sub-frames of 10ms length.
- Spectral Rolloff [44]: it is the frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
- Zero-Crossing Ratio (ZCR) [44]: it is the rate of sign-changes of the signal during the duration of a particular frame.
- Pitch Estimation [45]: it shows a continuous fundamental frequency estimation of speech signal using a Deep Learning algorithm [45]; The pitch estimation is made each 10ms subframes.
- Chroma STD [44]: it is the spectral energy where the bins represent the 12 equal-tempered pitch classes of English semitone spacing.

Our proposed method is compared to the PLC method embedded in the original G.722.2, the repetition PLC method, and the HMM-based PLC. Fig.17 shows the simulation results at different PLRs (5%, 10%, 15%, and 20%).

From the test results, we notice that our proposed method outperforms the other methods at different PLRs, where it achieved a significant improvement of perceived speech quality compared to the standard G.722.2 PLC method in all scenarios.

These results can be explained by the fact that the G.722.2 PLC method extrapolates the Linear Prediction (LP) parameters from previously received packets to conceal the lost speech packets [34]. In the case of multiple consecutive speech packet losses (bursty losses), the gain of concealed speech segments is attenuated towards comfort noise level, which will result in a noticeable annoying speech artifact. Fig. 18 shows a comparison between the original speech signal (No loss), and the concealed speech signals using the G.722.2 PLC method and our proposed

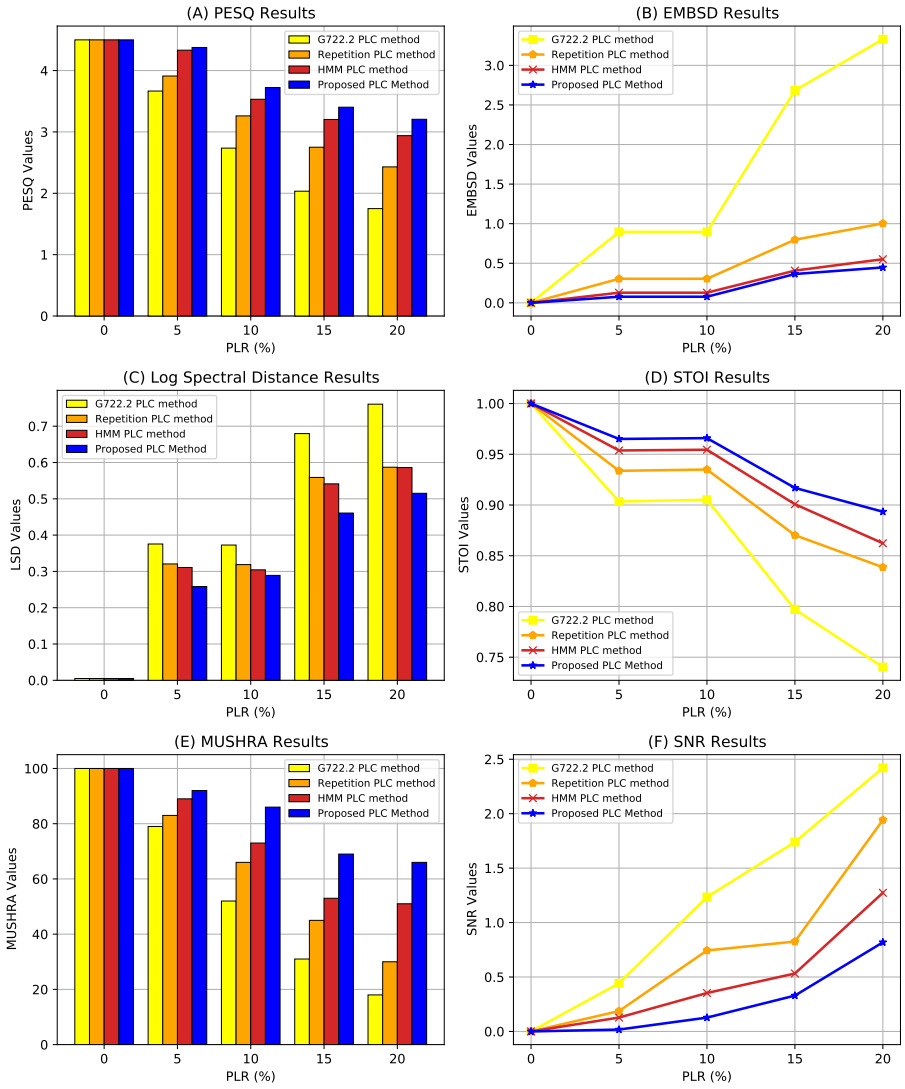

Fig. 17 Simulation results in terms of A) PESQ, B) EMBSD, C) Log Spectral Distance, D) STOI, E) MUSHRA and F) SNR.

PLC method, where two consecutive voiced speech packets are lost. In Fig.18[C] we can see the gain attenuation problem of G.722.2 PLC method in consecutive packet loss patterns and how our proposed PLC method handles gracefully the lost speech segment (Fig. 18[E]); Fig. 18[G] confirms this observation where we we notice the gain desynchronization between G.722.2 synthesized speech signal and original speech signal.

Additionally, G.722.2 PLC method uses pitch period repetitions of the previous correctly received packets to conceal the lost ones. Unfortunately, this simplistic

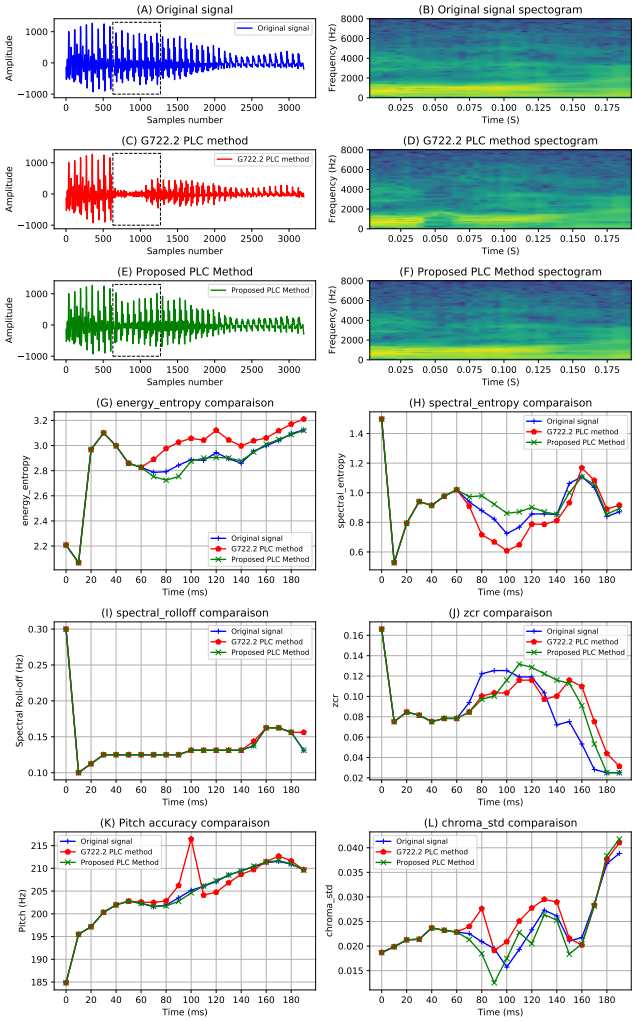

Fig. 18 Comparison between the synthesised speech signal of G.722.2 PLC method and our proposed method (a voiced segment) in terms of: A) Original speech signal, B) Spectogram of original speech signal, C) synthesised speech signal recovered by G.722.2 PLC method, D) Spectogram of synthesised speech signal recovered by G.722.2 PLC method, E) synthesised speech signal recovered by our proposed PLC method, F) spectogram of synthesised speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off cmparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison

approach is not efficient in all speech events, such as glides, onsets and unvoiced to voiced segment transitions, where the speech features change rapidly. Fig. 19 presents a comparison between the synthesized speech signals of the original signal (No loss), G.722.2 PLC method, and our proposed PLC method (three consecutive packet loss of an unvoiced to voiced speech transition). In Fig. 19[C] we can see how the G.722.2 PLC method fails to follow the dynamic variation of the lost segment

shown in Fig. 19[A], unlike our proposed method that kept the signal features of the original signal using HMM state transitions, Fig. 19[E]. Fig. 19[D] shows clearly the pitch repetition issue of G.722.2 PLC method. Fig. 19[K] confirms this observation where we can see the pitch fluctuation caused by the G.722.2 PLC method and how our proposed method succeeds in estimating the pitch lost segment accurately.

The packet repetition PLC method involves the use of the last received speech packet instead of the lost one. This PLC method gives acceptable results when the speech features are more or less constant (voiced speech segments). Otherwise, this method causes noticeable sound artifacts in case of high speech feature variations (glides, onsets, and unvoiced/voiced transitions) or in case of bursty packet loss where the last received speech segment is repeated several times as replacement of the last one. Fig. 20 shows a comparison between the synthesized speech signals of the original signal (No loss), repetition PLC method, and the proposed PLC method (three consecutive packet losses of unvoiced speech segments). Fig. 20[C] shows the annoying speech artifact caused by repeating the last received speech packet three times and how our proposed method succeeds in estimating the speech features even after three speech packet losses as shown in Fig. 20[E].

Test results of our proposed method were slightly higher than HMM method at low PLRs (5% and 10%). However, this difference becomes more important in bursty loss scenarios and high PLRs (15% and 20%). These results can be explained by the fact that conventional These results can be explained by the fact that conventional HMM-based PLC methods use Minimum Mean Square Error (MMSE) to estimate the lost speech packet based on its pdf; the pdf of a missing speech packet can be represented as [12]:

$$p(\phi_t|\phi_1^{t-1},\phi_{t+1}) = \sum_n p(\phi_t, s_t = n|\phi_1^{t-1},\phi_{t+1}) = \sum_n p(\phi_t|s_t = n)\times p(s_t = n|\phi_1^{t-1},\phi_{t+1}) \tag{31}$$

We can also write,

$$\hat{\phi}_t = w_n \times u_n \tag{32}$$

where: u_n is the mean emission vector in state n and w_n is the mixture weights of being the conditional state probabilities in a Gaussian Mixture Model (GMM) graphical presentation [12]; Equation 32 may be rewritten as,

$$\hat{\phi}_t = p(s_t = n|\phi_1^{t-1},\phi_{t+1}) \times u_n \tag{33}$$

At this point we can see in Equation.32 that the estimated speech vector depends only on two factors: the mean emissions of a state $s_t = n$ and its PDF; which will induce the lack of direct dependencies between the consecutive emissions (the emissions independencies assumption) and the lack of direct dependencies between the emissions and the surrounding states (the emissions dependencies assumption) as explained and detailed in Section 3. Consequently, HMM PLC method estimates/predicts the missing parameter vector $\hat{\phi}_t$ at time t according only to the parent state s_t without taking into consideration the neighboring emissions ϕ_{t-1} and ϕ_{t+1} due to the emissions independencies rule. In a speech PLC context, we cannot accept that emission at the instant t to be completely decorrelated from the past and future emissions due to the short-time stationarity feature of speech signal [27]. This assumption causes annoying perceptual artifacts like pitch

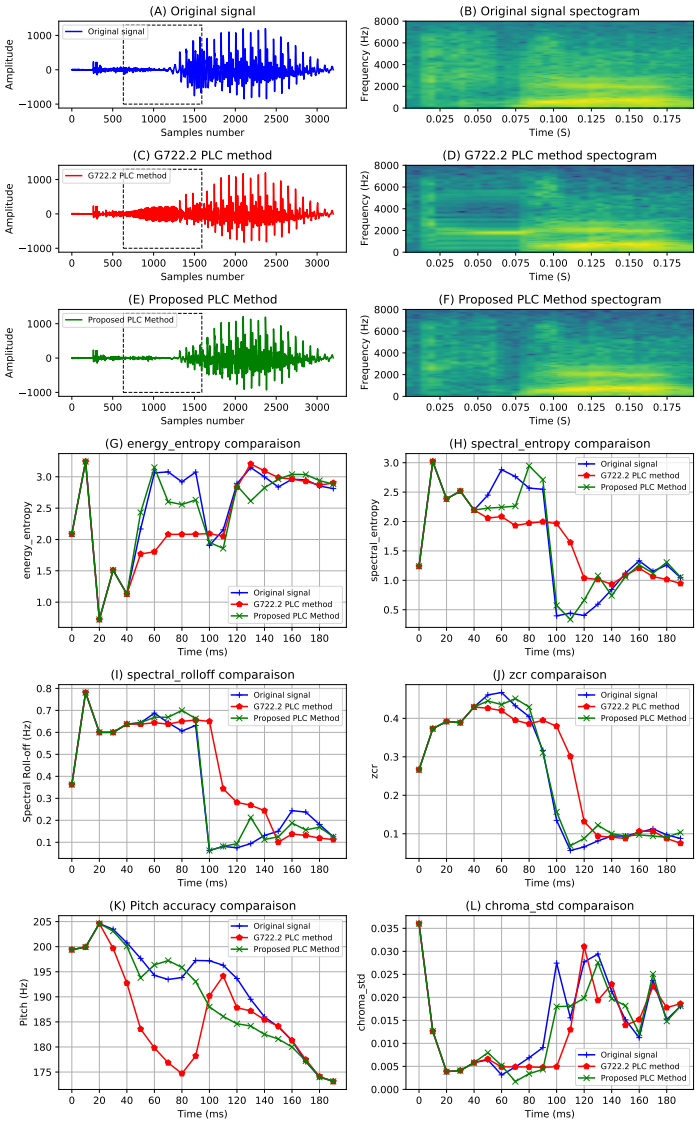

Fig. 19 Comparison between the synthesised speech signal of G.722.2 PLC method and our proposed method (unvoiced to voiced speech transition) in terms of: A) Original speech signal, B) Spectogram of original speech signal, C) synthesised speech signal recovered by G.722.2 PLC method, D) Spectogram of synthesised speech signal recovered by G.722.2 PLC method, E) synthesised speech signal recovered by our proposed PLC method, F) spectogram of synthesised speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off cmparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison

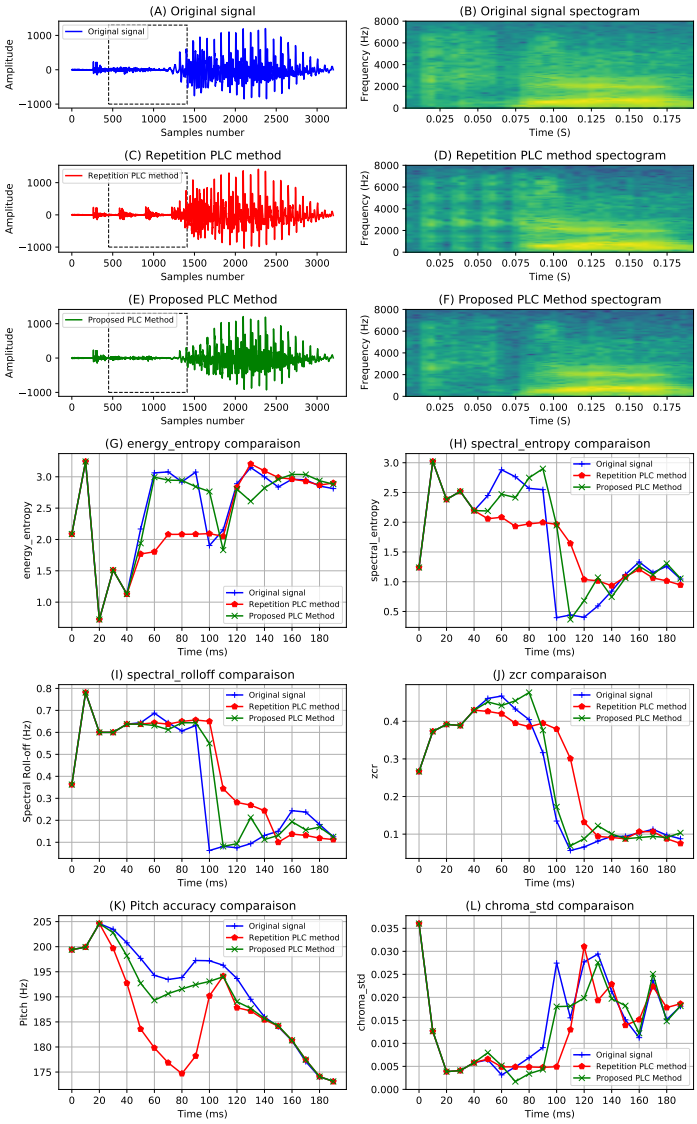

Fig. 20 Comparison between the synthesised speech signal of repetition enhanced with repetition PLC method and our proposed method (unvoiced segment) in terms of: A) Original speech signal, B) Spectrogram of the original speech signal, C) synthesized speech signal recovered by repetition PLC method, D) Spectrogram of synthesized speech signal recovered by repetition PLC method, E) Synthesized speech signal recovered by our proposed PLC method, F) Spectrogram of synthesized speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off comparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison.

fluctuation, gain mismatching, and phase discontinuities. Our proposed method circumvents the two mentioned HMM assumptions by adding context-dependent information to the HMM using the DT approach. The DT allows the HMM model to estimate which parameter vector $\hat{\phi}_t$ should be produced by the state $s_t = n$ according to neighboring emissions ϕ_{t-1} and ϕ_{t+1}, which allows for a smooth transition between recovered and received speech signals.

Fig. 21 shows a comparison between the synthesized speech signals of the original signal (No loss), HMM PLC method, and our proposed PLC method (one packet loss of onset speech segment). We can see the smooth transition achieved by our proposed method by taking into consideration the previous and next emissions using the DT approach, instead of the MMSE estimator that induces a sudden gain fluctuation as shown in Fig. 21[G]. Fig. 21[H, I, L, K] shows that our proposed method succeeds in capturing the pitch variation on the onset accurately unlike HMM PLC method which shows a sudden unstable pitch variation. This observation can also be seen when we compare the spectrograms of the three signals shown in Fig. 21[B, D, F]. Even the ZCR of our proposed was much better than the HMM as shown in Fig. 21[J].

HMM PLC method implicates annoying gain jumps and clicks by averaging all the emissions (observed parameter vectors) of state $s_t = n$ to produce the estimated one. Our proposed method allows a smooth gain transition between received and estimated packets because it estimates the missing parameter vector by analyzing the neighboring emissions in the DT process and not by averaging the observations of the current state.

Fig. 22 shows a comparison between the synthesized speech signals of the original signal (No loss), HMM PLC method, and our proposed PLC method (three lost packets of voiced to unvoiced transition), here we can see the gain jump induced by the HMM PLC method due to the emissions independencies assumption and how our proposed method accurately captured the gain transition and generates accordingly a speech packet that ensures a smooth speech loss concealment as shown in Fig. 21[G].

The HMM, approach induces annoying pitch replication in the case of a high stationary speech segment because the HMM-MMSE model will generate the same parameter vector if the neighboring states are the same (see equations 32 and 30). Due to the DT architecture, our proposed method can generate different parameter vectors even if the neighboring states are the same which avoids the pitch replication artifacts.

Fig. 23 shows a comparison between the synthesized speech signals of the original signal (No loss), HMM PLC method, and our proposed PLC method (one packet loss of a voiced speech segment). We can see how HMM PLC method generates the same speech vector if the HMM state does not change, which induces a pitch replication and gain fluctuation artifacts, as shown in Fig. 23[G, H, K], as opposed to our proposed method that generates a different speech packet even if the HMM state is the same using the DT approach.

7 Conclusion

In this paper, we studied the implementation of HMM in the PLC context, in which we demonstrated the limitation of this model due to the emissions indepen-

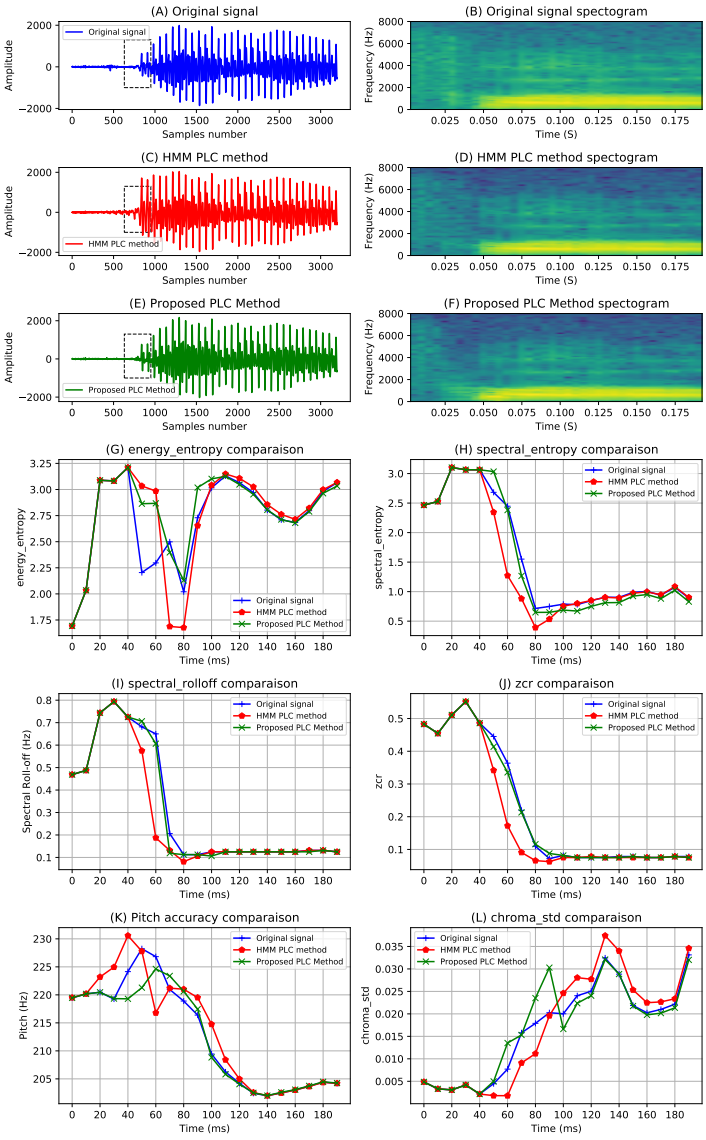

Fig. 21 Comparison between the synthesized speech signal of G.722.2 enhanced with HMM PLC method and our proposed method (Onset segment) in terms of: A) Original speech signal, B) Spectrogram of the original speech signal, C) synthesized speech signal recovered by HMM PLC method, D) Spectrogram of synthesized speech signal recovered by HMM PLC method, E) Synthesized speech signal recovered by our proposed PLC method, F) Spectrogram of synthesized speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off comparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison.

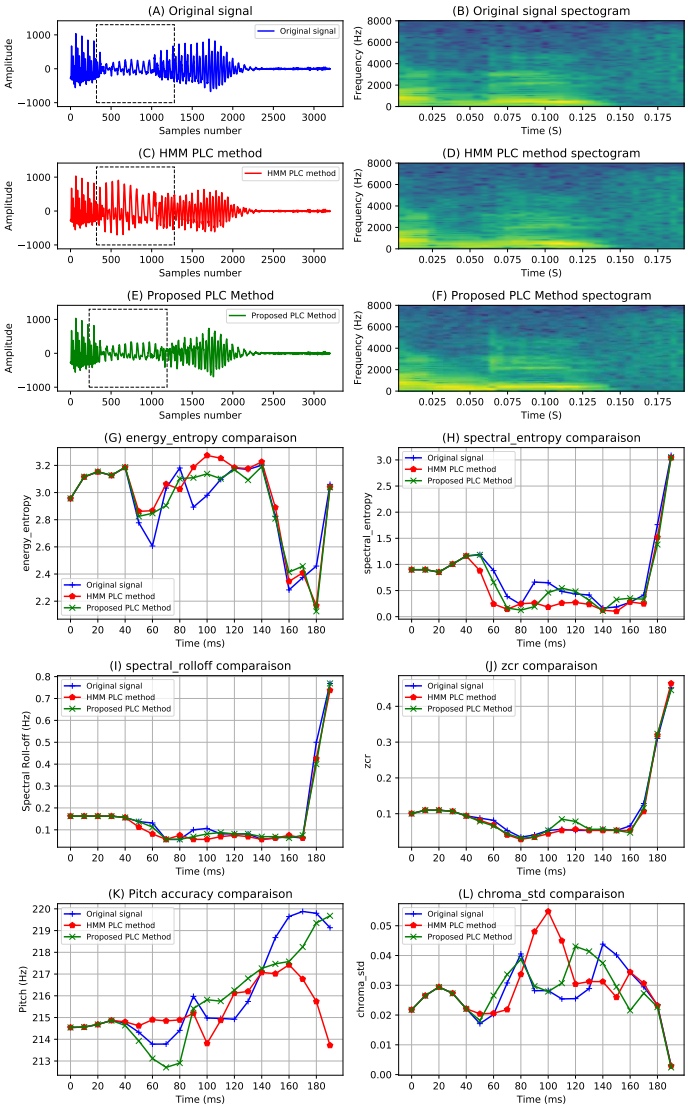

Fig. 22 Comparison between the synthesized speech signal of G.722.2 enhanced with HMM PLC method and our proposed method (transition between unvoiced and voiced segments:3 consecutive losses) in terms of: A) Original speech signal, B) Spectrogram of the original speech signal, C) synthesized speech signal recovered by HMM PLC method, D) Spectrogram of synthesized speech signal recovered by HMM PLC method, E) Synthesized speech signal recovered by our proposed PLC method, F) Spectrogram of synthesized speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off comparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison.

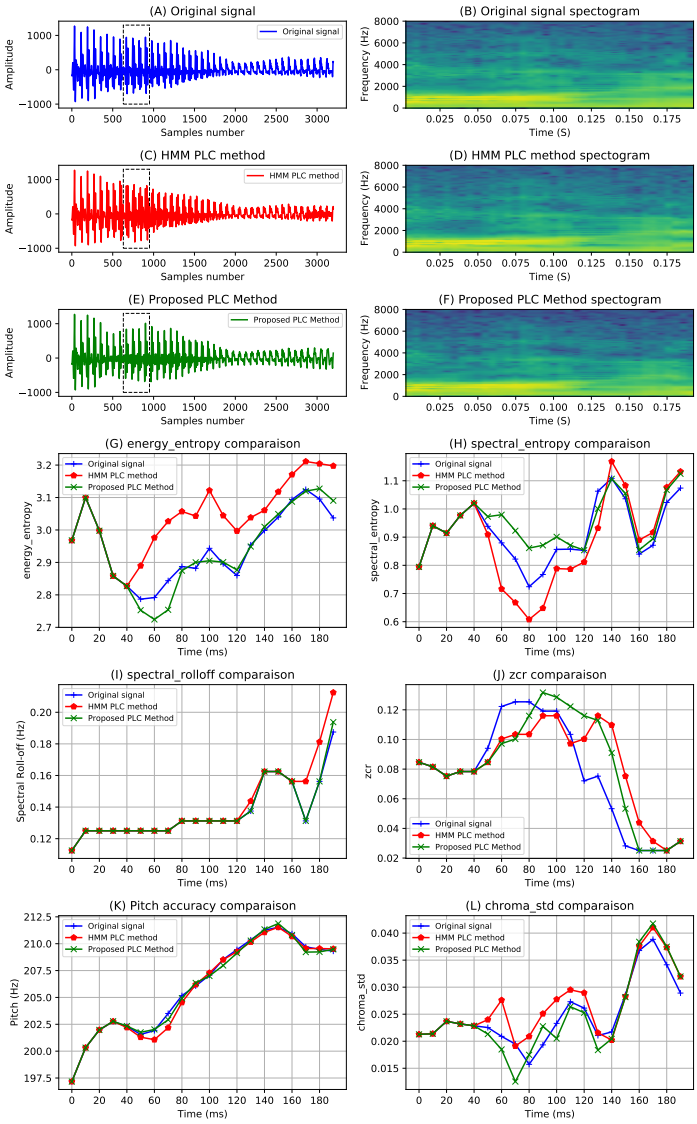

Fig. 23 Comparison between the synthesised speech signal of G.722.2 enhanced with HMM PLC method and our proposed method (voiced segment: 1 frame loss) in terms of: A) Original speech signal, B) Spectrogram of the original speech signal, C) synthesized speech signal recovered by HMM PLC method, D) Spectrogram of synthesized speech signal recovered by HMM PLC method, E) Synthesized speech signal recovered by our proposed PLC method, F) Spectrogram of synthesized speech signal recovered by our proposed PLC method, G) Energy entropy comparison, H) Spectral entropy comparison, I) Spectral roll-off comparison, J) ZCR comparison, K) Pitch accuracy comparison and L) Chroma std comparison.

dencies assumption, where the generated emission at time t does not depend on the previous or next emissions but rather on the parent state only. Mathematical and practical proofs were presented in this paper to support our point of view as well as to highlight the annoying artifacts resulting from this Markovian assumption such as pitch fluctuation and gain mismatching ... etc. This was followed by an integration of the DT's architecture into this model to create a new HMDT PLC concept. This concept, can track the statistical evolution of speech signals and predict the lost speech packets according to both, the estimated HMM states and the received speech packets. The proposed model allows our PLC method to not only circumvent the HMM limitations but also to add a piece of context-dependent information to the model and to exploit the short-stationarity feature of the speech signal to produce a natural and smooth transaction between the concealed speech segments and the correctly received ones. Moreover, the proposed PLC can handle the high-dimensional speech features gracefully with a fast generation of the estimated speech packets due to its hierarchical structure, which makes it a very prominent solution for real-time applications like VoIP, where the low complexity and latency constraints have to be respected. Note that while our proposed method is implemented on the G.722.2 codec, it can still be adapted to other codecs like Enhanced Voice System (EVS) as well as used in real-time speech recognition applications context, which will be the topic of our future projects.

References

1. Circus, Drake, Sun, Lingfen, Wade, Ifeachor, G., Emmanuel: Impact of packet loss location on perceived speech quality. Computer Communications **28**, 582–588 (2012)
2. Verma, P., Mezza, A., Chafe, C., Rottondi, C.: A deep learning approach for low-latency packet loss concealment of audio signals in networked music performance applications. In: Conference of Open Innovations Association, pp. 268–275 (2020)
3. Santiago Pascual Joan Serrà, J.P.: Adversarial auto-encoding for packet loss concealment. Computer Science **10**(12), 78–89 (2021)
4. Lin, J., Wang, Y., Kalgaonkar, K., Keren, G., Zhang, D., Fuegen, C.: A time-domain convolutional recurrent network for packet loss concealment. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7148–7152 (2021)
5. Gournay, P., Rousseau, F., Lefebvre, R.: Improved packet loss recovery using late frames for prediction-based speech coders. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. I–108 (2003)
6. Yeh, J., Lin, P., Kuo, M., Hsu, Z.: Bilateral waveform similarity overlap-and-add based packet loss concealment for voice over ip. Journal of applied research and technology **11**, 559–567 (2013)
7. Colin, P., Orion, H., Vicky, H.: A survey of packet-loss recovery techniques for streaming audio. Network, IEEE **12**, 40–48 (1998)
8. Emin, M., C.E.W, S.: Burst erasure correction codes with low decoding delay. Information Theory **50**, 2494–2502 (2004)
9. Parijat, D., Eitan, A.: Utility analysis of simple fec schemes for voip. Information Theory **2345**, 226–239 (2002)

10. Ahmed, B., Ashish, K., Wai-tian, T., Xiaoqing, Z., John, A.: Fec for voip using dual-delay streaming codes. Information Theory pp. 1–9 (2017)
11. Teck-Kuen, C., D.C, P.: Effects of loss characteristics on loss-recovery techniques for voip. Computer Communications pp. 204–204 (2006)
12. C.A., R., Manohar, M., Søren, A., Søren, J.: Hidden markov model-based packet loss concealment for voice over ip. Audio, Speech, and Language Processing, IEEE Transactions on **14**, 1609–1623 (2006)
13. Jari, T., Pekka, L., Tarmo, L.: Assessment of objective voice quality over best-effort networks. Computer Communications **28**, 582–588 (2005)
14. M.A, K., R.K, Y.: Markov chain prediction for missing speech frame compensation. Speech Coding pp. 75–77 (2000)
15. Koenig, L., André-Obrecht, R., Mailhes, C., Fabre, S.: A new feature vector for hmm-based packet loss concealment. In: European Signal Processing Conference, pp. 2519–2523 (2009)
16. Borgström, B.J., Alwan, A.: Hmm-based reconstruction of unreliable spectrographic data for noise robust speech recognition. IEEE Transactions on Audio, Speech, and Language Processing **18**(6), 1612–1623 (2010)
17. Goodarzi, M.M., Almasganj, F.: A gmm/hmm model for reconstruction of missing speech spectral components for continuous speech recognition. International Journal of Speech Technology **19** (2016)
18. Goodarzi, M.M., Almasganj, F., Ahadi, M.: Reconstructing missing speech spectral components using both temporal and statistical correlations. In: 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA), pp. 125–128 (2010). DOI 10.1109/ISSPA.2010.5605492
19. Agnello, G., Dansereau, R.M.: Parametric mixing for centralized voip conferencing using itu-t recommendation g.722.2. In: Canadian Conference on Electrical and Computer Engineering, pp. 2045–2048 (2006)
20. Wideband coding of speech at around 16 kbps using Adaptive Multi-Rate Wideband (AMR-WB). ITU-T Standard G.722.2 (2003)
21. Lai, K., Twine, N., O'Brien, A., Guo, Y., Bauer, D.: Artificial intelligence and machine learning in bioinformatics. In: S. Ranganathan, M. Gribskov, K. Nakai, C. Schönbach (eds.) Encyclopedia of Bioinformatics and Computational Biology, pp. 272–286. Academic Press, Oxford (2019)
22. Hofmann, P., Tashman, Z.: Hidden markov models and their application for predicting failure events. In: V.V. Krzhizhanovskaya, G. Závodszky, M.H. Lees, J.J. Dongarra, P.M.A. Sloot, é. Brissos, J. Teixeira (eds.) Computational Science – ICCS, pp. 464–477. Springer International Publishing, Cham (2020)
23. Franzese, M., Iuliano, A.: Correlation analysis. In: S. Ranganathan, M. Gribskov, K. Nakai, C. Schönbach (eds.) Encyclopedia of Bioinformatics and Computational Biology, pp. 706–721. Academic Press, Oxford (2019)
24. Zhou, W., Zhu, Z.: A new online bayesian nmf based quasi-clean speech reconstruction for non-intrusive voice quality evaluation. Neurocomputing **349**, 261–270 (2019)
25. Nadkarni, P.: Core Technologies: Machine Learning and Natural Language Processing, chap. 4, pp. 85–114. Academic Press (2016)
26. Yang, W.: Chapter 3 - development of early warning models. In: Early Warning for Infectious Disease Outbreak, pp. 35–74. Academic Press (2017)
27. Theodoridis, S., Koutroumbas, K.: Chapter 9 - context-dependent classification. In: S. Theodoridis, K. Koutroumbas (eds.) Pattern Recognition, fourth

edition edn., pp. 521–565. Academic Press, Boston (2009)
28. Foote, J.T.: Decision-tree probability modeling for hmm speech recognition. Ph.D. thesis, USA (1994)
29. Wellekens, C.: Explicit time correlation in hidden markov models for speech recognition. ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing **12**, 384–386 (1987)
30. Rodriguez, D.Z., Rosa, R.L., Bressan, G.: Intelligent learning techniques applied to quality level in voice over ip communications. International Journal on Advances in Internet Technology **6**, 261–270 (2013)
31. Teunen, R., Akamine, M.: Hmm-based speech recognition using decision trees instead of gmms. In: Proc. Interspeech 2007, pp. 2097–2100 (2007)
32. Akamine, M., Ajmera, J.: Decision tree-based acoustic models for speech recognition. EURASIP Journal on Audio, Speech, and Music Processing **2012**, 1–8 (2012)
33. Khreich, W., Granger, E., Miri, A., Sabourin, R.: On the memory complexity of the forward–backward algorithm. Pattern Recognition Letters **31**(2), 91–99 (2010)
34. Mittag, G., Möller, S.: Detecting packet-loss concealment using formant features and decision tree learning. In: Proc. Interspeech 2018, pp. 1883–1887 (2018)
35. Srivastava, D.K., Yadav, R., Agrwal, G.: Map reduce programming model for parallel k-mediod algorithm on hadoop cluster. In: 7th International Conference on Communication Systems and Network Technologies (CSNT), pp. 74–78 (2017)
36. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. IEEE International Conference on Acoustics, Speech, and Signal Processing **2**, 749–752 (2001)
37. Sanneck, H.A., Carle, G.: Framework model for packet loss metrics based on loss runlengths. Multimedia Computing and Networking pp. 1–23 (1999)
38. Yang, W.: Enhanced modified bark spectral distortion (embsd): An objective speech quality measure based on audible distortion and cognition model. PhD, thesis, Temple University Graduate Board (1999)
39. Ilk, H.G., Tugaç, S.: Channel and source considerations of a bit-rate reduction technique for a possible wireless communications system's performance enhancement. IEEE transactions on wireless communication **4**(1), 93–99 (2005)
40. Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J.: A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4214–4217 (2010)
41. Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J.: An algorithm for intelligibility prediction of time–frequency weighted noisy speech. IEEE Transactions on Audio, Speech, and Language Processing **19**(7), 2125–2136 (2011)
42. Sanneck, H.A., Carle, G.: Framework model for packet loss metrics based on loss runlengths. Multimedia Computing and Networking **12**, 245–256 (1999)
43. Tu, Y.H., Du, J., Gao, T., Lee, C.H.: A multi-target snr-progressive learning approach to regression based speech enhancement. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 1608–1619 (2020)

44. Giannakopoulos, T.: pyaudioanalysis: An open-source python library for audio signal analysis. PloS one **10**(12), 78–89 (2015)
45. Salamon, J., Bittner, R., Bonada, J., Bosch, J.J., Gómez, E., Bello, J.P.: An analysis/synthesis framework for automatic f0 annotation of multitrack datasets. In: 18th International Society for Music Information Retrieval Conference, pp. 56–87 (2017)