A Log Domain Pulse Model for Parametric Speech Synthesis

Gilles Degottex, Pierre Lanchantin and Mark Gales

***Abstract*—Most of the degradation in current Statistical Parametric Speech Synthesis (SPSS) results from the form of the vocoder. One of the main causes of degradation is the reconstruction of the noise. In this article, a new signal model is proposed that leads to a simple synthesizer, without the need for ad-hoc tuning of model parameters. The model is not based on the traditional additive linear source-filter model, it adopts a combination of speech components that are additive in the log domain. Also, the same representation for voiced and unvoiced segments is used, rather than relying on binary voicing decisions. This avoids voicing error discontinuities that can occur in many current vocoders. A simple binary mask is used to denote the presence of noise in the time-frequency domain, which is less sensitive to classification errors. Four experiments have been carried out to evaluate this new model. The first experiment examines the noise reconstruction issue. Three listening tests have also been carried out that demonstrate the advantages of this model: comparison with the STRAIGHT vocoder; the direct prediction of the binary noise mask by using a mixed output configuration; and partial improvements of creakiness using a mask correction mechanism.**

***Index Terms*—speech synthesis, text-to-speech, parametric speech synthesis, acoustic model, voice, pulse model**

I. Introduction

Text-to-speech is a useful technology in many industrial applications and also has application in the area of speech impairment [1]. Statistical Parametric Speech Synthesis (SPSS) systems using waveform parametrisation (vocoding) [2], [3], [4] offers a means to model and manipulate the voice where concatenative synthesis [5], [6] lacks this flexibility. This inflexibility limits the range of application area for example when adapting a voice to another one is necessary [1]. On the other hand, concatenative synthesis offers a perceived quality that is still hard to reach for SPSS [7], [8] due to the limitations of current modelling approach. Even though most SPSS statistical models are currently trained on a signal model (using a vocoder parametrisation), waveform level synthesis (without vocoder) has also been proposed [9]. This offers speech quality comparable to concatenative synthesis, but requires a large quantity of data and computation power. Thus, vocoder-based SPSS still offers a flexible and tractable solutions that could be improved in terms of quality.

The quality of current vocoder-based SPSS is sufficient for some applications (e.g. GPS devices in noisy environment). However, it is not satisfactory for many others applications (e.g. use in quiet environment, game and music industry). The vocoder is responsible for a substantial part of the quality degradation [8]. The ability of the vocoder to resynthesize all of the components of the speech signal is obviously important to retain all of the perceived characteristics that the voice can produce. This ability also needs to apply to all speaking styles, voice qualities and attributes. Otherwise, the vocoder, as well as the SPSS system using it, would be appropriate for a specific set of voices, but would systematically fail at reproducing the rest of the voice space. Within the vocoder, the flexibility of the signal model is often limited, either in its design or by using regularisation techniques. To compensate for this lack of flexibility, many ad-hoc techniques currently exists for tunning the perceived attributes of a synthesis a posteriori (e.g. variable or constant all-pass filtering, forced maximum voiced frequency). This is obviously a workaround and it eludes the modelling of the attributes these techniques target, thus limiting the range of voices that the training process can absorb and represent. The signal model should be flexible enough for representing all perceived attributes the voice can have. Using a uniform representation for voiced and unvoiced regions is a step towards this direction as it allows independent transitions from deterministic to noisy transitions and vice-vice versa at any time and any frequency. It also simplifies the learning process and relieves the architecture of the acoustic modelling. Continuous f_0 modelling and uniform features have been suggested for this purpose[10], [11], [12]. Finally, and not least, ad-hoc parametrisation of signal models often lead to intractable tuning issues that depend on very specific expertise and know-how, which can impede the overall research methodology and progress in research about vocoding.

STRAIGHT is currently the most used vocoder for SPSS [13], [14]. It uses a voicing decision in order to ensure the random excitation of unvoiced segments, similarly to other vocoders [15], [16], [17], [18]. The noise component in voiced segments is analyzed and reconstructed through an aperiodicity measure, which is expressed as a noise level below the amplitude spectral envelope. This measure computes the difference between the harmonic peaks and spectral valleys [14]. However, in noisy time-frequency regions of voiced segments, this measure systematically underestimates the noise level because the peaks-to-valleys difference is always positive and substantial in such segments whereas it should be aligned to the amplitude spectral envelope and not located below it. Therefore, the synthetic noise in the generated waveform tends to be lower than that of the original signal (as demonstrated and illustrated in Sec. IV-A). On the one hand, this underestimation favours a slight buzziness in the voiced part of the transients, while the voicing decision ensures the proper

G. Degottex, P. Lanchantin and M. Gales are with the University of Cambridge, UK. (e-mail: gad27@cam.ac.uk).

randomization of the fricatives and silences. Interestingly, it has been shown that a slight buzziness (i.e. a lack of noise) is preferred over noisiness in the transients [19]. On the other hand, by mitigating the noise component, this noise underestimation tends to produce always the same voice quality, a slightly tense and buzzy voice. This is obviously a lack of flexibility from the vocoder since it does not yield an accurate noise resynthesis that is necessary for good reconstruction of breathiness and other voice qualities that involve the presence of noise in voiced segments. As mentioned above, this is a major limitation restricting not only the coverage of the voice attributes, but also limits the overall perceived quality in general.

In this article, we want to address the issues above by suggesting a new and simple synthesizer that should reproduce the noisy time-frequency regions more accurately than the STRAIGHT vocoder, a well known candidate of the additive linear source-filter model. The synthesizer, called *Pulse Model in Log-domain* (PML), generates a time sequence of wide-band pulses, in the spectral domain, as in the STRAIGHT synthesis [13], [14], rather than the approaches adopted in HNM[20], HMPD[12] and Ahocoder[17] that synthesise sinusoidal components. In both voiced and unvoiced segments, a *pulse* is treated as a morphing between a Dirac function and a short segment of Gaussian noise, followed by convolution with a Vocal Tract Filter (VTF). Obtaining a perceptually meaningful morphing between a Dirac and a specific time segment of noise is far from straightforward. Inspired by the voice production, the traditional source-filter model suggests an additive weighting in the linear domain [21]. With this approach, the Dirac component will disappear only when the noise level masks it. This masking effect is far from obvious partly because of the noise level and Dirac amplitude are dependent on two different normalisations in the spectral domain. Indeed, in order to control the noise component, its level has to be normalised with respect to the energy of the synthesis window. On the contrary, in order to control the Dirac component, its amplitude has to be normalised with respect to the sum of the synthesis window. The masking affect in time domain is, thus, an indirect result of the mixture of noise and Dirac according to the variable window length that has to follow the fundamental period. An explicit control of this perceived element would be obviously more convenient. For this reason, and for the problem of underestimated aperiodicity mentioned above, the Dirac component tends to rise above the noise, which often leads to extra buzziness. The all-pass filter commonly used is then a convenient technique for reducing this buzziness. Another workaround is to lower the the deterministic component in noisy frequency bands (thus complicating the synthesis process) or split the signal into multiple bands of interleaved deterministic and noisy contents [22], [16], [17]. The HMPD vocoder [12] does not have this issue since it randomises the phase of the harmonics proportionally to a Phase Distortion Deviation (PDD) feature, which gradually scatters the deterministic content. Its quality is nevertheless bounded by the frequency resolution given by the fundamental frequency curve f_0 [23].

In PML, we suggest to mix the deterministic and noise components with a weighting in the log spectral domain (i.e. multiplication in the linear spectral domain and convolution in the time domain). We expect some advantages of this approach. Firstly, the convolution of the Dirac function with the noise randomises the phase spectrum and avoids any possible residual buzziness. This phase randomisation process is similar to the *structural noise* mentioned in previous works [24] and the ad-hoc all-pass filter technique. Secondly, by normalising the noise by its energy, the noise's amplitude is aligned to the deterministic content. Thus, by convolution of the two, the resulting amplitude is preserved. Finally, the convolution by the VTF spectrum will set the final amplitude of the speech pulse, independently of the nature of the source below it. This is an interesting property that splits the modelling of the amplitude from that of the nature of the phase, without having to deal with masking effects. Thirdly, and not least, this log-domain formulation leads to a very simple realisation of the synthesizer as shown in the next section. In this work, we simplify the weighting function to be a binary mask for reason explained later on. For each time-frequency bin, the Dirac function of each pulse is either left untouched or fully replaced by the corresponding bin of the spectrum of a Gaussian noise. From this perspective, the suggested vocoder is similar to the Multi-Band Excitation vocoder (MBE)[22], except that wide-band pulses are synthesized at each period instead of harmonic components, and a uniform representation for the voiced and unvoiced segments is used in PML. This binary noise mask can also be seen as a time-frequency binary voicing decision, which can take any shape and is not limited to time limits (as with voicing decisions) and/or frequency limits (as with a maximum voiced frequency [15], [16], [17]).

In Sec. II, we first describe the theory behind the synthesizer as well as the necessary technical details of the vocoder's implementation. In a first experiment in Sec. IV-A, we then demonstrate the problem of noise reduction that exists in STRAIGHT. The remaining experiments are dedicated to listening test results about SPSS comparing different training configurations. A last experiment presents some results about a correction of the noise mask for creakiness. Compared to the first presentation of PML in [23], we dropped the comparisons with HMPD. Even though this could have brought a second state-of-the-art method in the experiments, the results in [23] show clearly that PML solves all the issues of HMPD and outperform it in the listening tests. Given its broad use, the STRAIGHT vocoder [25], [26], [27], [28], [29], [30] seems to be a sufficient baseline and a solid candidate of linear source-filter model that we need for this presentation.

II. The PML synthesizer

The synthesis process of PML needs the following features that are illustrated in Fig. 1:

$f_0(t)$ A fundamental frequency curve, which does not exhibit voicing decisions. If the provided fundamental frequency does contain zeros, these segments can be filled by linear interpolation between voiced segments, and extrapolated at the beginning and end of the signal. The REAPER f_0 estimator was used in this work [31].

$V(t,\omega)$ The VTF response, which is assumed to be minimum phase. For reasons of comparison, the spectral envelope estimate provided by the STRAIGHT vocoder [13], [14] was used in this work.

$M(t,\omega)$ A binary mask in the time-frequency space. Here 0 is for deterministic regions and 1 for noisy regions. In this work, this mask is derived from the Phase Distortion Deviation (PDD) [12] PDD(t,ω) as described below. This mask can also be modified, as presented in Sec. II-B, with the aim of improving creakiness.

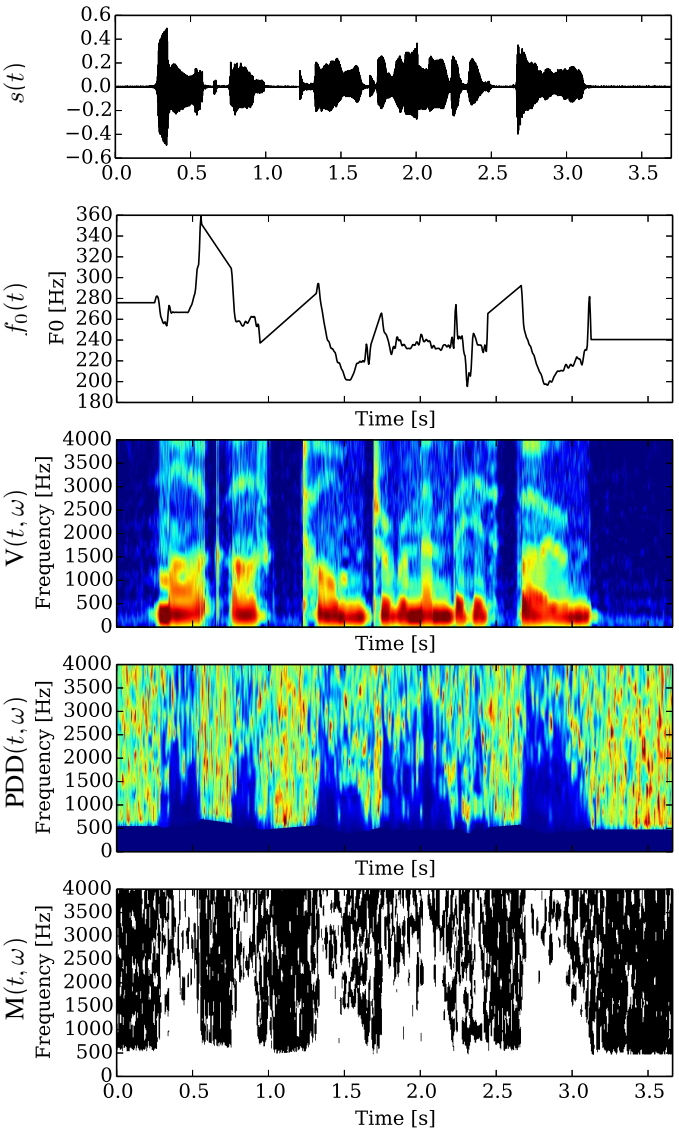


Fig. 1. From top to bottom: a recorded waveform used to extract the following elements; The continuous fundamental frequency curve $f_0(t)$; the amplitude spectral envelope $V(t,\omega)$; the Phase Distortion Deviation PDD(t,ω) (a measure of phase randomness. The warmer the colour, the bigger the PDD value and the noisier the corresponding time-frequency region); the binary mask $M(t,\omega)$ derived from PDD, which allows to switch the time-frequency content from deterministic (white) to random (black). The features that are necessary for PML synthesis are only: $f_0(t)$, $V(t,\omega)$ and $M(t,\omega)$.

Since $f_0(t)$ and $V(t,\omega)$ are extracted using state-of-the-art methods previously published (REAPER and STRAIGHT, respectively), the rest of this section describes only how to compute the noise mask and how to correct it for acoustic elements that exhibit creakiness. The synthesis process of a waveform from a given set of features then follows.

A. Estimation of the noise mask

A simple means to compute a binary mask $M(t,\omega)$ is adopted based on a measure of harmonicity or how sinusoidal is the speech signal in the time frequency plan. The Phase Distortion Deviation (PDD) [12], [32], [33] is used for this purpose and the mask is obtained by thresholding the PDD values.

In order to compute PDD, the Phase Distortion (PD) at each harmonic frequency is first computed [12]:

$$\text{PD}_{i,h} = \phi_{i,h+1} - \phi_{i,h} - \phi_{i,1} \tag{1}$$

where $\phi_{i,h}$ is the phase value at frame i and harmonic h, as measured by a sinusoidal model [34], [20], [35]. A step size of one forth of a fundamental period was used in this work to split the analysed signal into frames as in [12]. PDD is then computed as the short-term standard-deviation of PD:

$$\begin{aligned} \text{PDD}_i(\omega) &= \operatorname*{std}_i\left(\text{PD}_i(\omega)\right) \\ &= \sqrt{-2\log\left|\frac{1}{K}\sum_{n\in C} e^{j(\text{PD}_n(\omega))}\right|} \end{aligned} \tag{2}$$

where $C = \{i - \frac{K-1}{2}, \cdots, i + \frac{K-1}{2}\}$ with $K = 9$ in this work and $\text{PD}_i(\omega)$ is the continuous counterpart of $\text{PD}_{i,h}$ obtained by linear interpolation across frequency.

In [12], it is shown that this PDD measurement saturates below 1.0 and, thus, cannot estimate very high values of phase variance. Consequently, a threshold of 0.75 was used to force the variance to a fixed higher value in order to ensure the proper randomization of the noise segments. This threshold value is also supported by the first experiment of this article (Fig. 4). To summarize here briefly, this experiment shows that this threshold splits the PDD distribution computed on speech recordings into two modes, one related to the deterministic component and the other related to the noisy components. Therefore, in this work the same threshold was used for building the mask: $M(t,\omega) = 1$ if PDD$(t,\omega) > 0.75$ and zero otherwise. This bainry mask is also convenient for two complementary reasons: First, estimation noise is always present in (2) (e.g from sinusoidal estimation error, interferences from the VTF). As a result, the PDD value is always slightly over-estimated and a minimum of noise will always be generated in the resynthesis, which leads to hoarsness in voiced segments. Therefore, taking this element into account as well as the saturation issue of PDD mentionned above, the binary mask is also a convenient workaround that alleviates these two problems. Future works might focus on solving the underlying cause of these problems in order to use a continuous value in $M(t,\omega)$. Different noise measurements could also be interesting research directions (e.g. wavelet-based).

Note that the phase measurement at DC is unreliable and is forced to a zero value. Thus, the PDD computation is zero below the 2nd harmonic and, therefore, the mask $M(t,\omega)$ is zero in this frequency band. This implies that the first harmonic is never randomized. This is not a problem since, in silences

and fricatives, the corresponding amplitude is rather weak so that this sinusoid is hardly perceived. Additionally, in voiced segments, the first harmonic is always deterministic for all voice qualities.

B. Mask correction for creakiness

To model voiced and unvoiced time-frequency regions, most vocoders rely on an f_0 estimate and/or voicing detection that assume voiced segments to have sinusoidal content. However, various segments of the speech signal are voiced with very non-periodical characteristic of the pulse's position, called *creakiness* in this presentation, as in creaky voice phonatory mode [36] and sometimes in transients. Thus, the corresponding sinusoidal content in these segments is highly disturbed and often wrongly classified as unvoiced segments, leading to hoarseness and noisy transients in the synthesized voice. This problem is a recurrent issues in SPSS, which has been addressed by various means depending on the vocoder or acoustic model used [29], [37], [38].

The noise mask used in this work, based on PDD, encounters also this problem since PDD uses a harmonic model (Eq. 1). Therefore, in this section, we propose a correction mechanism of the noise mask in order to improve creakiness. An energy assignment technique is adopted for this purpose. The correction will modify the original noise mask and the synthesis stage will be kept untouched, as described in Sec. II-C.

1) Time assignment: The first and fundamental step to estimate this correction mask consists in a measurement of energy concentration in time, which is inspired by the time reassignment technique [39], [40]. The reassignment operator is defined by:

$$\hat{t}(t,\omega) = t - \frac{\partial\phi(t,\omega)}{\partial\omega} \tag{3}$$

where the second term is the group delay of $S(t,\omega)$ the Fourier transform of the speech signal at time t using an analysis window of 3 average periods in this work. This operator is commonly used to create the time-reassigned spectrogram [39], [40]:

$$\hat{S}(t,\omega) = \int_{-\infty}^{\infty} |S(s,\omega)|\delta\big(t - \hat{t}(s,\omega),\omega\big)ds \tag{4}$$

In our application, the information related to the amplitude of the signal is already modelled by the spectral envelope estimate $V(t,\omega)$. Thus, to avoid redundancy between the features, we consider that the reassigned amplitude is constant for all frequencies ($S(t,\omega) = 1 \ \forall t, \forall\omega$) in (4) and the reassignment processing becomes an assignment measure:

$$\hat{A}(t,\omega) = \int_{-\infty}^{\infty} \delta\big(t - \hat{t}(s,\omega),\omega\big)ds \tag{5}$$

In practice, the integration in (5) is obviously discrete. The analysis instant t is limited to instants t_i that are distant of a constant step size. For $\hat{A}(t,\omega)$ to be similar enough to its continuous counterpart, the time resolution has to be thin enough. Based on informal experiments, we chose a step size of 1ms, which seems enough for the targeted purpose. Once the mask correction is finished, it can be resampled to the 5ms resolution used by the other features. An example of $\hat{A}(t,\omega)$ is shown in Fig. 2 (second row from the top). Straight vertical lines appear in regions of creakiness (e.g. in the blue intervals) and plosives (e.g. in the red interval). Those straight lines correspond to concentration of energy at a time instant (e.g. glottal closure instant or plosive impulse).

Even though the next steps seem heuristic, they aim simply at obtaining a mask correction that put in emphasis time frequency regions that contain mainly creakiness.

2) Cepstral filtering: This second step aims mainly at denoising $\hat{A}(t,\omega)$ through cepstral liftering across frequencies. The used cepstral order is difficult to determine precisely as it depends solely on the nature of the analysed signal, here speech. We basically wish to denoise $\hat{A}(t,\omega)$ and avoid details that would increase the training load of the Artificial Neural Net (ANN) for no benefit. Based on observation of $\hat{A}(t,\omega)$ on a few recorded utterances, we saw that when a pulse appears, it spans a wide frequency band (e.g. in the blue intervals on 2nd plot from the top in Fig. 2). By trying different values, we found that an 8-order liftering exhibits a good balance between the denoising and the emphasis of the impulses, as shown in 3rd plot from the top in Fig. 2, leading to $\tilde{A}(t,\omega)$.

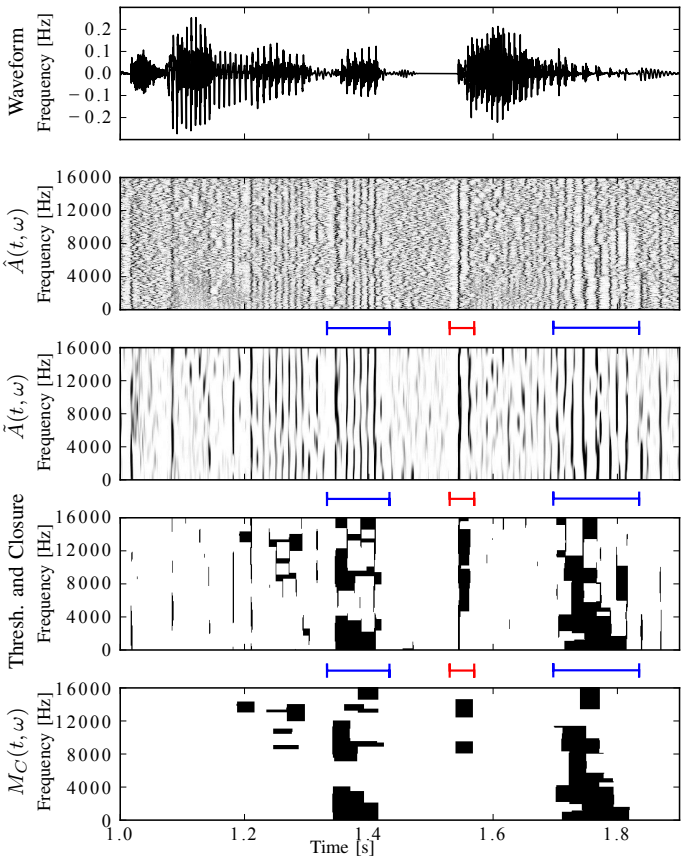


Fig. 2. Construction of the creakiness correction $M_C(t,\omega)$, the darker the colour the higher the value, from top to bottom: A utterance of recorded speech; Time assignment $\hat{A}(t,\omega)$; After cepstral liftering $\tilde{A}(t,\omega)$ in order to remove the noise and emphasize the impulses; After thresholding and morphological closure to remove the dependency on $f_0(t)$; After morphological opening to remove isolated impulses (e.g. plosives) and dilatation.

3) Thresholding and morphological operations: In noisy time-frequency regions the assignment operator distributes the energy equally in time. Thus, in these regions, values in $[0, 1]$ are the most likely to appear, whereas higher values are mainly related to time synchronous events. A first binary mask is thus obtained by thresholding $\tilde{A}(t, \omega)$ with a value of 1.

Creakiness appears then as interleaved regions of noised and voiced segments as in $\tilde{A}(t, \omega)$, because $\tilde{A}(t, \omega)$ is dependent on the original position of the glottal closure instants. The final feature correction should not be dependent on the glottal closure instants because the noise feature has to be uncorrelated from the f_0 feature. To remove this dependency, a morphological closure [41] is employed for *linking* the pulses as a spectral envelope would do the same between the harmonics in spectral content (see 4th plot from the top of Fig. 2). The morphological closure consists of a dilatation followed by an erosion of the same size. The size of the closure is chosen to be that of a period of 70Hz, which is a usual average f_0 for creaky voice [36] that composes mainly creakiness. Finally, a morphological opening (an erosion of a period of 70Hz followed by a dilatation of the same size) is also used to remove isolated impulses (e.g. plosives), followed by a dilatation (of an average period) to make the first and last impulse of a region as wide as an average pulse. These last two operations leads to the final creaky voice mask correction $M_C(t, \omega)$ (see bottom plot of Fig. 2).

To combine the original noise mask obtained by thresholding PDD with the creakiness correction $M_C(t, \omega)$, it is considered that a time-frequency sample is noisy ($M(t, \omega) = 1$) if and only if PDD$(t, \omega) > 0.75$ AND $M_C(t, \omega) = 0$.

One can note that this procedure makes use of many parameters that might need tuning. However, these tunning parameters are the same for all the voices used in the experiments in Sec. IV. This should better support the generality of the results instead of hand tuned parameters for each voice. Additionally, the parameters are used to tune the feature computation a priori, before any statistical modelling, and not a posteriori during the synthesis stage, as in ad-hoc vocoding techniques. Therefore, a training system using the noise mask has the possibility to learn all of the features' characteristics and their inter-correlations with other features. On the contrary, parameters that are tunned a posteriori, as in ad-hoc vocoding techniques, cannot be learned by the statistical model, they would have to be tunned manually a posteriori, which would limit the practicability.

C. Signal synthesis

The generation of the waveform follows a pulse-based procedure, similarly to the synthesis process of the STRAIGHT vocoder. Short segments of speech signals, called pulses (roughly the size of a glottal pulse) are generated sequentially. In both voiced and unvoiced segments, the voice source of each pulse is made of a morphing between a deterministic impulse and Gaussian noise. This source is then convolved by the Vocal Tract Filter (VTF) response and then overlapped-add with the other pulses. This section describes the details of this procedure.

A sequence of pulse positions t_i is first generated all along the speech signal according to the given $f_0(t)$ feature:

$$t_{i+1} = t_i + 1/f_0(t_i) \tag{6}$$

with $t_0 = 0$. Then, to model the speech signal around each instant t_i, the following simple formula is applied:

$$S_i(\omega) = e^{-j\omega t_i} \cdot V(t_i, \omega) \cdot N_i(\omega)^{M(t_i,\omega)} \tag{7}$$

where $N_i(\omega)$ is the Fourier transform of a segment of Gaussian noise starting at $\frac{t_{i-1}+t_i}{2}$ and finishing at $\frac{t_i+t_{i+1}}{2}$, whose central instant t_i is re-centered around 0 (to avoid doubling the delay $e^{-j\omega t_i}$ for the noise in $S_i(\omega)$). Additionally, the noise $N_i(\omega)$ is normalized by its energy to avoid altering the amplitude envelope that has to be controlled by $V(t, \omega)$ only.

To better understand the elements involved in this model, its log-domain representation should be examined:

$$\begin{aligned}\log S_i(\omega) = \quad & \overbrace{-j\omega t_i}^{\text{Position}} + \overbrace{\log|V(t_i,\omega)|}^{\text{Amplitude}} + \overbrace{j\angle V(t_i,\omega)}^{\text{Minimum phase}} \\ & + \underbrace{M(t_i,\omega)}_{\text{Noise mask}} \cdot \Big(\underbrace{\log|N_i(\omega)|}_{\text{Noise amplitude}} + \underbrace{j\angle N_i(\omega)}_{\text{Random phase}} \Big)\end{aligned} \tag{8}$$

The *Position* defines the overall position of the voice source in the speech signal. This identifies the position of the Dirac impulse of the deterministic source component. The *Amplitude* defines the amplitude spectral envelope of all of the resulting segment of speech, independently of the source properties. The *Minimum phase* is built from the *Amplitude* through the Hilbert transform using the real cepstrum, in order to delay the energy of the pulse, as natural resonators do (see [12, Eq.(5)] or more generaly [42]). The *Noise mask* provides the means to switch between deterministic or random voice source at any time-frequency point. As already above in this work, this mask is a binary value. For $M(t, \omega) = 1$, the *Noise amplitude* will mainly correct the *Amplitude* in order to account for the difference between deterministic and noise normalisation (sum and energy, respectively). This ensures that the envelope of the noise amplitude is always at the same level as that given by the *Amplitude* spectral envelope $|V(t, \omega)|$. With $M(t, \omega) = 1$, the *Random phase* will also scatter the phase of the Dirac function and replace it by that of Gaussian noise.

In terms of model control, PML drastically simplifies the handling of the noise compared to the traditional source filter model. First, the low quefrency of its amplitude is only controlled by $|V(t, \omega)|$, as with the deterministic content. Thus, the value of the noise mask does not change the perceived amplitude, it mainly changes the nature of the phase. This dissociates the control of the amplitude from that of the phase. Second, the masking effects and their mastery, as seen in the traditional additive linear source-filter model and discussed above, are alleviated. It is enough to have $M(t, \omega) = 1$ for a given t and ω, to ensure the full randomization of the corresponding spectral content. Thirdly, the value of the noise mask is binary, making it a very simple feature to model by statistical approaches, as shown in Sec. IV-C. Finally, this suggested model is still a source-filter model, but with the combination of the source and filter done in the log-domain

instead of the linear domain (thus the chosen name Pulse Model in Log domain).

In order to build the complete speech signal from the pulses generated by (7), overlap and add is applied, without any additional synthesis window, neither consideration of windows' sum nor normalisation:

$$\check{s}(t) = \sum_{i=0}^{I-1} \mathcal{F}^{-1}\Big(S_i(\omega)\Big) \tag{9}$$

where I is the number of pulses in the synthesized signal.

It is also worth mentioning the following properties that the suggested model satisfies:

1) If $M(t,\omega) = 0 \ \ \forall\omega, \forall t$, (7) reduces to:

$$S_i(\omega) = e^{-j\omega t_i} \cdot V(t_i, \omega) \tag{10}$$

The time domain signal becomes the impulse response of the filter delayed at the pulse position t_i. In this case the signal is fully deterministic.

2) If $M(t,\omega) = 1 \ \ \forall\omega, \forall t$, (7) reduces to:

$$S_i(\omega) = e^{-j\omega t_i} \cdot N_i(\omega) \cdot V(t_i, \omega) \tag{11}$$

The time domain signal is a filtered noise segment. After summing the terms $S_i(\omega)$, this corresponds to a concatenation process of coloured Gaussian noise segments into a continuous noise signal (the last noise sample of the pulse i is the sample before the first sample of the pulse $i+1$). Thus, no periodicity appears in this noise, even though the synthesis is driven by a continuous $f_0(t)$. $f_0(t)$ influences only the time resolution of the dynamic noise filtering through the size of the noise segments $(t_{i+1} - t_{i-1})/2$. For f_0 values of 70Hz, a worst case scenario, this still allows to change the noise's timbre each 14ms.

III. Implementation details

The theoretical description of PML needs a few complementary technical remarks. Note that for the sake of reproducibility, the source code of PML is available at: https://github.com/gillesdegottex/pulsemodel

- In a traditional overlap-add stage (e.g. in PSOLA-based techniques [43], [44]), a window covering the whole pulse is commonly used to avoid cutting the signal too sharply at the boundaries of the time segment. Such a window is not necessary in the PML synthesis. The source signal has no energy before $\frac{t_{i-1}+t_i}{2}$ and $V(t_i,\omega)$ is built as a minimum phase filter, a window on the left of the pulse seems thus unnecessary. However, because the Gaussian noise segment is altered in frequency by the noise mask, Gibbs phenomena [42] can appear before $\frac{t_{i-1}+t_i}{2}$ in the source signal that can lead to pre-echo effects. To avoid these artefacts, the noise mask is first smoothed across frequency (using a hanning window of 9 bins in the following experiments). Then, a half-window of 1ms is used on the left of the time segment to eliminate any residual energy. On the opposite side of the time segment, on the right, because of the delays introduced by $V(t_i,\omega)$, there is energy after $\frac{t_i+t_{i+1}}{2}$ that will overlap with the time segment of the next pulse $i+1$. This is not an issue for two reasons: i) the impulse response of $V(t_i,\omega)$ is decaying exponentially, ensuring that the signal's energy is weak enough after some time extent (50ms is a safe duration for the formants' bandwidth of natural speech). ii) Each pulse is likely to have its energy delayed in a very similar way as the next pulse (as long as VTF is very similar from one pulse to the next). As a result, the tail of each pulse roughly replaces the energy which is delayed in the next pulse, so that there is no sudden burst of energy between two pulses.
- Instead of using a DFT size that covers the whole synthetic signal, the DFT size used for each pulse can be reduced in order to cover only an interval around each instant t_i (e.g. 2 periods before t_i and 50ms after). This drastically reduces the size of the DFT used in (7) and improves the computational efficiency (A DFT size of 4096 was used for the following experiments).
- The synthesis procedure requires only 2 FFT per pulse. One FFT is needed to compute $N_i(\omega)$, which needs a specific duration for each pulse, and one inverse FFT to compute the time domain signal (Eq. 9). If it is not pre-computed and cached, the computation of the minimum phase of the VTF $\angle V(t_i,\omega)$ from a given amplitude envelope requires also 2 extra FFT per pulse. This is clearly efficient enough for allowing real-time synthesis.
- Finally, most estimators of amplitude spectral envelope overestimate the DC component (often by ignoring the lips radiation effect in the spectral envelope model [21]). To avoid any side effect of this issue, the amplitude spectral envelope is high-pass filtered at $f_0/2$. This avoids residual DC component to be cut too sharply when overlapping the pulses in the time signal.

Note that, all the parameters mentioned above (spectral smoothing to reduce Gibbs phenomena, half-window for anti-pre-echo and high-pass cutoff) are used to avoid artefacts that commonly happens in audio processing and do not alter the perceived characteristics of the voice quality and timbre. In other words, for PML-based synthesis only, there is no ad-hoc tuning parameter that control the speech characteristics.

IV. Experiments

This section presents results of various experiments using the suggested PML synthesizer. The first one address the noise reconstruction between STRAIGHT and PML-based vocoders, while the rest of the experiments aim at assessing PML in various contexts of SPSS.

A. Noise reconstruction during analysis/re-synthesis

In this first experiment, the problem that occurs with the reconstruction of the noise component in STRAIGHT vocoder is illustrated, as discussed in the introduction, and the solution offered by a PML-based vocoder is compared (similar results can be found in [23] for the HMPD vocoder [12]).

Using both STRAIGHT and PML-based vocoders, vocoded speech utterances (i.e. analysis/resynthesis without any statistical modelling) are investigated for 6 different English voices [45], [46], [7] (3 females and 3 males; 2 females (CLB

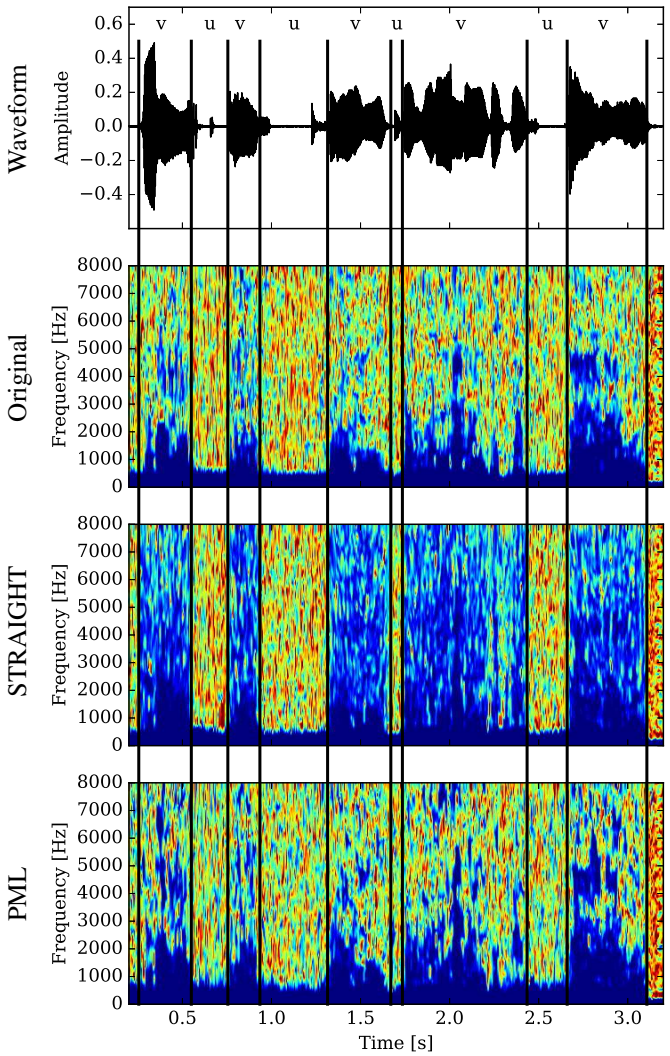


Fig. 3. An example of PDD feature computed from: an original recording and the analysis/resynthesis of STRAIGHT and PML (top to bottom). The vertical lines show the voiced/unvoiced transitions used by STRAIGHT. Voiced and Unvoiced segments are annotated by 'v' and 'u', respectively.

voice from Arctic[45] and LS from Blizzard[7]) and 2 males voices (BDL from Arctic[45] and NI from [46]) at 32kHz sampling rate and 1 female (CLB from Arctic[45]) and 1 male voice (RMS from Arctic[45]) at 16kHz; 4 American (SLT,CLB,BDL,RMS) and 2 British(LS,NI)). Then, the PDD is computed on top of the resulting vocoded signals in order to measure how well the signal randomness is reproduced by each vocoder. Fig. 3 illustrates an example of this PDD computation on top of the vocoded signal. In unvoiced segments, one can see that the randomness is reasonably well reconstructed by the two vocoders. In STRAIGHT, this is ensured by the voicing decision that forces full randomness in unvoiced segments no matter the aperiodicity model in voiced segments. Conversely, for voiced segments, the PDD feature computed from the STRAIGHT vocoded signal is often lower than that from the original signal. The PDD feature computed from PML vocoded signal shows a more accurate reconstruction of the noisy time-frequency regions.

The observation on the example from Fig. 3 is supported by the estimated probability density of PDD and aperiodicity values in the voiced segments of 100 utterances for each of the 6 voices mentioned above, shown in Fig. 4. For the PDD measures, the three distributions exhibit basically 2 modes, a small one close to zero and a larger one between 0.5 and 1.5, which roughly correspond to deterministic and noisy time-frequency regions, respectively. Firstly, one can see that the lower mode of the PML's distribution is stronger than the others. This is due to the mask that forces the PDD values below 0.75 to zero. One might actually argue that, consequently, voiced regions might lack a minimum of randomness. Using continuous values instead of binary values for $M(t,\omega)$ can be a potential solution to alleviate this issue in future works, if the pitfalls mentioned in Sec. II-A can be avoided. Secondly, and more importantly, the higher mode of the distribution corresponding to STRAIGHT's PDD is clearly lower than that of the original signal (at 0.5 instead of $\sim$1.2). This mode's maximum is below 0.75 for STRAIGHT, whereas it is above for the original signal. It was shown in [12] that values over this threshold are critically important for reconstructing the noisy characteristics of the voice, otherwise the phase is too concentrated in time and the synthesis sounds buzzy. This also demonstrates the reduction of the noise component through STRAIGHT's vocoding in the voiced segments, as discussed in the introduction. For the PML-based vocoder the higher mode of the original distribution is better reconstructed, which should lead to a better reconstruction of noisy components in voiced segments. A similar observation can be made for the aperiodicity measures in the bottom plot of Fig. 4. Even though only one mode can be observed for each distribution, STRAIGHT's mode is slightly lower than that of the orignal PML ones. On the contrary, the mode of PML is better aligned on the mode of the original signal.

B. Subjective quality of analysis/re-synthesis

In this experiments, PML is evaluated against STRAIGHT in terms of quality in analysis/resynthesis, thus, without any statistical modelling of the parameters. Two versions of PML are used in this comparison, PML with and without creakiness correction. The REAPER f_0 estimator and STRAIGHT's spectral envelope are extracted for both STRAIGHT and PML-based resynthesis. The noise features then depend one the vocoder, aperiodicity for STRAIGHT and noise mask (with or without creakiness correction) for PML.

50 sentences are analysised and resynthesized for each method compared and each of the 6 voices mentionned above. A Mean Opinion Score (MOS) listening test is then carried out to ask listeners to assess the overall quality of each resynthesis compared to the original sound, for 6 sentences [47]. Using crowdsourcing, workers from Amazon Mechanical Turk were asked to take the test [48], [49]. 83 listeners took the test and Fig. 5 shows the results (detailed results for each voice are available in the Annex A, Fig. 11 First, PML resynthesis exhibits a better quality compared to the baseline STRAIGHT, with a significance level of p-value$<$0.001. Second, even though a trend seems to favor PML+Creaky compared to PML, the difference is not significantly different. As shown by the brackets, all resynthesis are significantly different, except

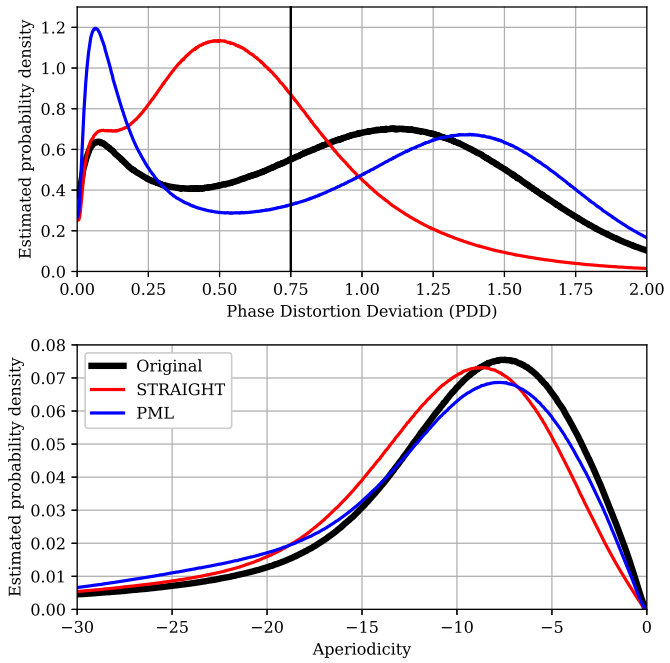
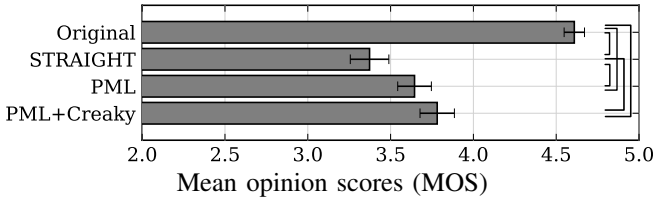


Fig. 4. Estimated probability density functions of PDD and aperiodicty values computed on top of the vocoded signals using STRAIGHT and PML-based vocoders. Computations on the original signals are also shown. The vertical line illustrates the threshold of 0.75 used for building the mask in the PML synthesizer.



(with the 95% confidence intervals and plain brackets showing p-value<0.001)

Fig. 5. MOS about the analysis/resynthesis quality over 6 voices comparing: Baseline STRAIGHT; PML resynthesis; PML resynthesis using creaky voice mask correction.

for PML+Creaky vs. PML. Experiments about resynthesis provide obviously interesting information in terms of upper bounds of quality vocoders can provide. However, they are incomplete since they do not correspond to a final application, i.e. there is no statistical modelling. The following listening tests should bring the necessary answers. A subset of the resyntheses can be found at: http://gillesdegottex.eu/Demos/DegottexG2017pml/resynth

C. Phase Distortion Deviation (PDD) vs. Noise Mask (NM) modelling

In this experiments, and the following ones, the PML synthesizer is evaluated in the context of SPSS using systems based on Long Short-Term Memory (LSTM) [3]. The pipeline used is basically the same as in [3] (called *Merlin*[4]). Comparing to the results presented in [23], 3 stacked Simplified-LSTM (SLSTM) layers [3] of 1024 units are used for the these experiments, instead of 6 $\tanh$ layers in the previous experiments [50]. Similarly to [23], we trained LSTM systems for the STRAIGHT and PML-based vocoders for the 6 voices mentioned in the previous experiment, as detailed below. For each voice, it was first necessary to align contextual labels on the recordings. These contextual labels are made of phonetic contexts (2 on the left, 2 on the right) at syllable, word and utterance level according to a question set and described in [2]. To enable this, HTS systems [2] were first trained using five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs [2]). STRAIGHT's features were used for these alignments. The features consisted of 60 Mel-cepstral coefficients [51], log f_0 values, and B-band aperiodicity coefficients (B between 21 and 24 depending on the sampling frequency), together with the first and second derivatives, extracted every 5ms. The rest of the topology of the HMM models and systems was similar to the one used for the Nitech-HTS system ([52]). Multiple iterations of training and re-alignment provided state-aligned labels used for training the following aoustic models. For the LSTM-based systems, 592 binary and 9 numerical features were derived from the questions set used in the HTS systems. For the STRAIGHT synthesizer, the output features were the same as the ones used for the HTS systems used for the alignment. Input features were normalised to [0.01, 0.99] and output features were normalised to zero mean and unit variance. For the suggested PML synthesizer, the same 60 Mel-cepstral coefficients and log f_0 values were used as for the STRAIGHT-based systems. The noise feature, however, was related to the noise mask described in Sec. II-A, using Bark-frequency bands as for the aperiodicity (same number of bands for comparison purposes). For each Bark frequency band an averaged value of noisiness is obtained by averaging the linear frequency values that fall into that band. Note that the creakiness correction was used only in the last experiment. For the following two experiments, only the mask based on thresholding PDD was used.

In this first SPSS-based experiment, two different methods for modelling the noise mask were compared. The first method models PDD values (called *PDD modelling* later on), as done for [23], which is then thresholded at synthesis stage as explained in Sec. II-A. The second method aims at modelling the values of the binary noise mask directly (called *NM modelling* later on), thus avoiding PDD in the acoustic model. In *PDD modelling*, PDD and its first and second approximate derivatives are normalized by their mean and variance. However, in *NM modelling*, the noise mask values are already bounded in $[0, 1]$. It does not seem necessary to normalise them. Moreover, using a linear output for these values is not advised as the ANN would have to model the boundaries at 0 and 1 whereas they are known a priori. For this reason, we modelled the static NM values using a sigmoid output function. For the 1st and 2nd approximate derivatives, we used hyperbolic tangent normalized in amplitude to 0.5 and 2, respectively, to match the values' intervals given by the windows used for the derivatives' approximation. The same windows are used as in [4], $w' = [-0.5, 0.0, +0.5]$ and $w'' = [1.0, -2.0, 1.0]$, for the 1st and 2nd order derivatives' approximation, respectively. Note that this leads to a mix output layer where the first 183 (3 times 80 mel-cepstral coefficients plus 3 times one f_0 value) values are linear outputs and the remaining $3 \cdot B$ (with B the number of noise bands)

are non-linear outputs.

In order to compare PDD and NM modelling and assess their impact on SPSS, a Comparative Mean Opinion Score (CMOS) listening test was carried out. The systems compared are:

STRAIGHT 60 mel cepstral coefficients, log f_0, Bark-frequency band aperiodicities.

PML-PDD 60 mel cepstral coefficients, log f_0, Bark-frequency bands PDD.

PML-NM 60 mel cepstral coefficients, log f_0, Bark-frequency bands NM.

The last 50 sentences of the test set [4] were synthesised for each of the 6 voices. As duration models are out of the scope of this study, the durations used were extracted from the original recordings. Similarly, common f_0 curves and amplitude spectral envelopes were used among the three syntheses in order to focus on the difference between PDD vs. NM modelling. The impact of these features will be presented in the next section. The systems trained for STRAIGHT were used to build the common features (for PML syntheses, $f_0(t)$ was then linearly interpolated in unvoiced segments to obtain a continuous $f_0(t)$ curve). Each listener taking the test assessed the 3 pairs of each system combinations for 8 random sentences among the 50x6=300 synthesized sentences. A 7-points scale was used in this test (3 points for the sound on the left, one neutral choice, 3 points for the sound on the right) as recommended by [47]. 47 listeners took the test properly and the results are shown in Fig. 6. Detailed and agregated results are shown. Agregated results are computed as in [47]. The "Preference test" results are deduced from the CMOS test by counting the number of assessments bigger than 1 favouring each system and those equal to zero for the no-preference choice.

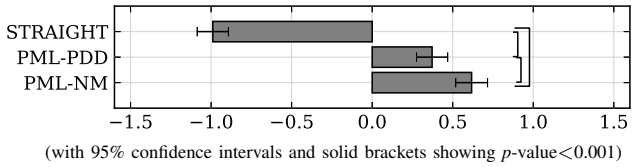
Results in Fig. 6 show that the NM modelling yielded on average better scores than both STRAIGHT and PDD-based modelling. Solid brackets on the right show significant differences for p-values$<$0.001. The improvement from PDD to NM modelling shows that the noise can be successfully modelled by a simple binary mask if the output layer configuration is setup appropriately. The clear difference between STRAIGHT and PML-based systems supports also the suggested approach. In the previous publication [23], PML had a similar quality than STRAIGHT. This difference of results is explained by the difference of acoustic models between the two experiments. 6-layers DNN were used for [23] and 3 layers of SLSTM were used for this experiment. This difference supports the idea that no matter how much the acoustic model improves in terms of training capacity, the quality provided by STRAIGHT is always limited since the noise reconstruction in the voiced segments is limited, as shown in Sec. IV-A. On the contrary, PML add noise in the voiced segments, which can be a risk, as discussed in the introduction. Consequently, if the acoustic model of the noise mask improves, the overall quality improves as well.

For the sake of the precision, detailed results for each voice are available in the Annex A, Fig. 12. A subset of the syntheses used in this listening test is also available for listening at:
http://gillesdegottex.eu/Demos/DegottexG2017pml/pdd2nm

Pairewise preferences

	STRAIGHT	PML-PDD	PML-NM
STRAIGHT		-0.86	-1.12
PML-PDD	0.86		-0.11
PML-NM	1.12	0.11	

Comparative mean opinion scores (CMOS)



(with 95% confidence intervals and solid brackets showing p-value$<$0.001)

CMOS-based preferences

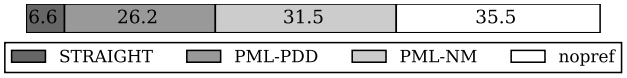


Fig. 6. Pairewise, aggregated and preference results (in %) of SPSS listening test over 6 voices comparing: Baseline STRAIGHT; PML synthesis using PDD modelling ; PML synthesis using NM modelling

D. Standalone PML vs. Stream mixtures with Baseline

In this experiment, we evaluate the impact of mixing the features' stream generated by the systems of the previous experiment. The systems compared PML-NM syntheses using:

Standalone All features from the PML-NM systems.

Baseline f_0 Noise mask, amplitude spectral envelope from PML-NM and log f_0 from STRAIGHT's systems.

Baseline f_0, V Noise mask from PML-NM and amplitude spectral envelope, log f_0 from STRAIGHT's systems.

The same setup was used for the CMOS listening test as in the previous experiment. 45 listeners took the test properly and the results are shown in Fig. 7. First, the results show that there is little differences between the syntheses. This concludes that PML-based system can be used standalone, without mixing with other systems. We can also see that the Standalone syntheses seems slightly preferred compared to the other mixed systems, with a significance level of 0.05. However, from the detailed results shown in Fig. 13, we can see that this difference mainly comes from the LS voice.

A subset of the syntheses used is also available at:
http://gillesdegottex.eu/Demos/DegottexG2017pml/expbas

E. Creakiness correction

In this section, we discuss the impact of the creakiness correction, described in Sec. II-B, on the quality.

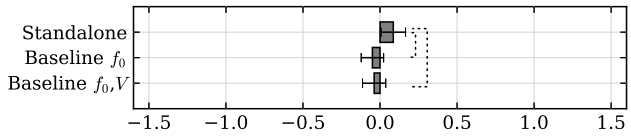
First, Fig. 8 shows an example of creaky voice synthesized using LSTM-based synthesis using the original noise mask or the corrected noise mask. One can see that between 2.6s and 2.7s, denoted as a blue interval, the synthetic signal is noisy on the middle plot, whereas it is supposed to be more creaky, as correct on the bottom plot, and as shown in the original recording.

In order to go beyond this simple illustration, we trained the following systems and carried out a listening test to compare them:

Pairewise preferences

	Standalone	Baseline f_0	Baseline f_0, V
Standalone		0.08	0.09
Baseline f_0	-0.08		-0.01
Baseline f_0, V	-0.09	0.01	

Comparative mean opinion scores (CMOS)



(with 95% confidence intervals and dotted brackets showing p-value<0.05)

CMOS-based preferences

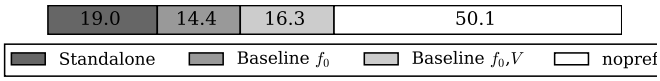


Fig. 7. Pairewise, aggregated and preference results (in %) of SPSS listening test over 6 voices comparing PML-based synthesis using 3 different stream setups: PML Standalone (f_0, amplitude spectrum and Noise Mask features generated from PML-NM-based system); Baseline f_0 (as in Standalone, except for the f_0 that is generated using the STRAIGHT-based system; Baseline f_0, V. (f_0 and amplitude spectrum generated using the STRAIGHT-based system and Noise Mask generated by the PML-NM-based system).

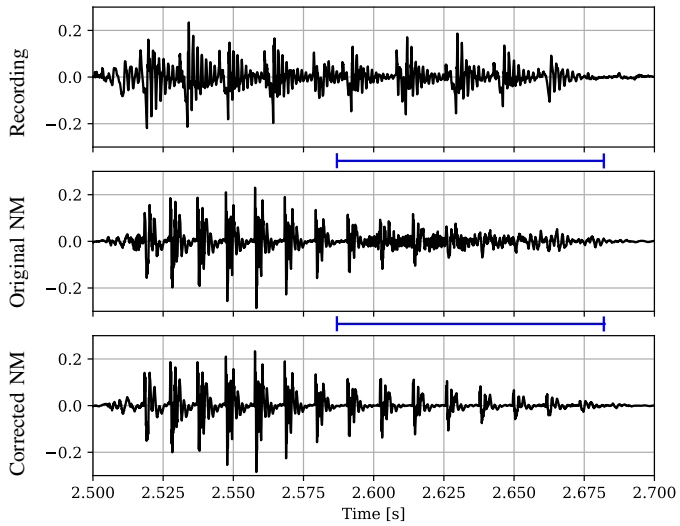


Fig. 8. Example of creakiness correction on synthesized speech compared to the original recording. Plot in the middle shows an LSTM-based synthesis using the original noise mask (computed through thresholding of PDD(t, ω)); the bottom plot shows the same time segment when the noise mask has been corrected using the suggested creakiness correction. One can see that, in the blue interval, the signal is more noisy on the middle plot than in the bottom plot. This simply illustrates that an acoustic model trained on a corrected noise mask for creaky voice can, indeed, improve the waveform reconstruction.

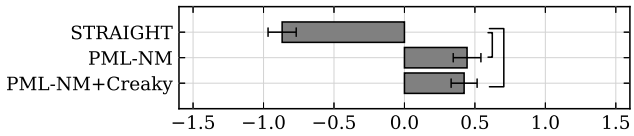
STRAIGHT The same system as in Sec. IV-C.
PML-NM The same system as in Sec. IV-C.
PML-NM+Creaky The same as **PML-NM** except that the noise mask was corrected for creakiness.

The setup of the LSTM-based systems is the same as in the first experiment in Sec. IV-C. A CMOS listening test was then carried out in order to evaluate the impact of the suggested correction on the perception of the quality. The setup of the listening test is basically the same as the two previous tests.

Pairewise preferences

	STRAIGHT	PML-NM	PML-NM Creaky
STRAIGHT		-0.90	-0.83
PML-NM	0.90		-0.01
PML-NM-Creaky	0.83	0.01	

Comparative mean opinion scores (CMOS)



(with 95% confidence intervals and solid brackets showing p-value<0.001)

CMOS-based preferences

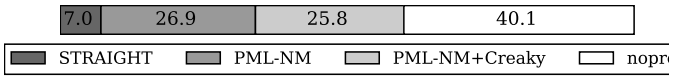
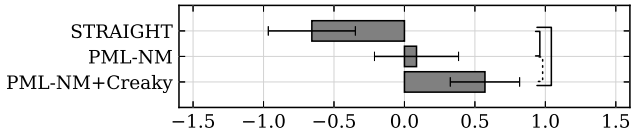


Fig. 9. Results of listening test over 6 voices comparing: STRAIGHT vocoder; the suggested synthesizer PML; PML with the creakiness correction.

Pairewise preferences

	STRAIGHT	PML-NM	PML-NM Creaky
STRAIGHT		-0.49	-0.83
PML-NM	0.49		-0.31
PML-NM-Creaky	0.83	0.31	

Comparative mean opinion scores (CMOS)



(with 95% confidence intervals, solid brackets showing p-value<0.001, dotted brackets showing p-value<0.05)

CMOS-based preferences

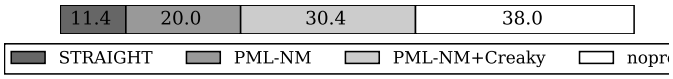


Fig. 10. Similar results as in Fig. 9, except that only the sentences with more than 6% creakiness are kept for these results.

45 listeners took the test properly and the results are shown in Fig. 9. Even though Fig. 8 suggests that the waveforms are properly corrected, on average it seems that listeners have no preferences between the original noise mask and the corrected one. The detailed results for each voice in Fig. 14 show that the BDL voice seem to take advantage of the mask correction, though without exhibiting any significant differences. Thus, comparing with the other voices, this mask correction seems as much likely to degrade the perceived quality. One possible reason of this result might be due to the fact that the mask correction also adds spurious voiced time-frequency regions as shown in mid and high frequencies of bottom plot of Fig. 2. This might increase the overall voicing of the speech signal in regions that are not supposed to be voiced, which increases then the overall buzziness of the voice.

Nevertheless, in the case the voice contains a substantial amount of creakiness, using this mask might be a solution to improve the quality. In order to investigate on this, the results of the listening test can be split among sentences with almost no creakiness and those with a minimal presence of creakiness. For this purpose we measured a rough quantity of creakiness in each sentence by computing the proportion of $M_C(t,\omega) == 1$ compared to all $M_C(t,\omega)$ values in the noise mask feature. Fig. 10 shows the listening test results using only the sentences that exhibit more than 6% of creakiness. This threshold has been chosen manually, the goal being only to show that the mask can improve the quality in some subset of the sentences. Focusing on this subset, we can see that the creakiness correction does, indeed, improve the perceived quality of these sentences compared to the original mask with a significance level of 0.05.

Even though this demonstrates the potential of the mask correction, this does not constitute a standalone system since an a priori estimation of overall creakiness is necessary to exhibit this partial result. Forthcoming works could suggest a creakiness prediction in order to decide if the corrected mask or the non-corrected mask should be predicted using the corresponding acoustic models.

Audio utterances used during the listening test can also be found at the following address:
http://gillesdegottex.eu/Demos/DegottexG2017pml/creakycorr

V. Conclusions

A new signal model, called *Pulse Model in Log-domain* (PML), was proposed and the corresponding synthesis procedure was described. We can summarize the benefits in the design of PML in the following manner.

- Compared to the state of the art, PML is mainly designed for better synthesis of the noise in voiced segments. Conversely to the traditional additive source-filter model in the linear domain, the phase randomisation approach used in PML is able to avoid any residual buzziness. Indeed, using the additive source-filter model, ad-hoc techniques must be used for forcing the randomness, otherwise the deterministic component will generate a well known buziness. On the contrary, the noise mask used in PML allows to force the full randomness of a time-frequency region.
- Because the noise used in PML is always normalised in amplitude and the noise model controls only the phase component, the amplitude and phase spectra are controlled independently. This clarifies and simplify the control of the speech elements. With the additive source-filter model, the statistical model in SPSS either assumed independence of amplitude and aperiodicity and failed in mixing them properly (as in most HMM-based synthesis using separate decision trees for each acoustic feature), or, it struggled in learning complex correlations. In PML, this problem is mostly solved since amplitude and phase are modelled independently at signal level. The statistical model can thus focus only on the qualitative correlations between phase and amplitude of the voice signals (e.g. formants bandwidth with noise presence in breathiness).
- PML makes use of a binary noise mask that represents voiced and unvoiced segments in a uniform way. Conversely to other approaches where the synthesis process has to switch between two different signal models, PML uses always the same model, no matter the nature of the speech signal. Additionally, this approach does not need hard time or frequency boundaries, but can take any shape in the time-frequency plan. This offers a flexibility that most current vocoders do not have. For example, the deterministic components can fade away in the high frequencies at the end of a voiced segment, while the low frequencies are still deterministic (e.g. when the voice relaxes).
- The design of PML leads to a very simple synthesizer, which is straightforward to understand and implement. A creakiness correction has been suggested that shows that it is also easy to build modifications on top of the current description of PML.
- Finally, the synthesis process requires very low computation time, which is encouraging for potential real-time applications.

In terms of experimental results based on listening tests, we have shown that a PML-based SPSS system better performs than a comparable system based on the well-known STRAIGHT vocoder. Another experiment has shown that the binary noise mask can be directly modelled by the acoustic model of the SPSS system by adapting the output layer accordingly. An experiment has also shown that a system trained using PML's features can be used as a standalone system. In other words, it was shown that the predicted features do not need to be crossed over with the predicted features of a STRAIGHT-based system. Because creakiness is a recurrent issues in SPSS, we also suggested a mask correction for these time frequency segments. A last listening test has been used to evaluate the impact of this correction on the perceived quality. Even though the results are not as encouraging as the previous results, we have shown that the suggested mask correction does improve the quality on sentences that exhibit creakiness.

VI. Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 655764. The research for this paper was also partly supported by EPSRC grant EP/I031022/1 (Natural Speech Technology).

The authors would like also to thanks the numerous workers on Amazon Mechanical Turk for their participation to the listening tests.

References

[1] C. Veaux, J. Yamagishi, and S. King, "Using HMM-based speech synthesis to reconstruct the voice of individuals with degenerative speech disorders.," in *Proc. Interspeech*, 2012, pp. 967–970.

[2] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[3] Zhizheng Wu and Simon King, "Investigating gated recurrent neural networks for speech synthesis," *CoRR*, vol. abs/1601.02539, 2016.

[4] Simon King Zhizheng Wu, Oliver Watts, "Merlin: An open source neural network speech synthesis system," in *Proc. 9th Speech Synthesis Workshop (SSW9)*, 2016, pp. 218–223.

[5] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996, vol. 1, pp. 373–376.

[6] C. Hamon, E. Mouline, and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modifications of speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1989, vol. 1, pp. 238–241.

[7] The Speech Synthesis Special Interest Group, "The Blizzard Challenge 2016 [Online]," http://www.synsig.org/index.php/Blizzard_Challenge_2016/, 2016.

[8] G. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, "Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech," in *Proc. Interspeech*, 2014, pp. 1504–1508.

[9] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[10] K. Yu and S. Young, "Continuous f0 modeling for HMM-based statistical parametric speech synthesis," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 19, no. 5, pp. 1071–1079, 2011.

[11] J. P. Cabral, "Uniform concatenative excitation model for synthesising speech without voiced/unvoiced classification," in *in Proc. Interspeech*, 2013, pp. 1082–1086.

[12] G. Degottex and D. Erro, "A uniform phase representation for the harmonic model in speech synthesis applications," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 38, 2014.

[13] H. Kawahara, I Masuda-Katsuse, and A. de Cheveigne, "Restructuring speech representations using a pitch-adaptative time-frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[14] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT," in *MAVEBA*, 2001.

[15] T. Drugman and T. Dutoit, "The deterministic plus stochastic model of the residual signal and its applications," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 968–981, 2012.

[16] G. Degottex, P. Lanchantin, A. Roebel, and X. Rodet, "Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis," *Speech Comm.*, vol. 55, no. 2, pp. 278–294, 2013.

[17] D. Erro, I. Sainz, E. Navas, and I. Hernaez, "Harmonics plus noise model based vocoder for statistical parametric speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, pp. 184–194, 2014.

[18] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications in speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4230–4234.

[19] J. Latorre, M. J F Gales, S. Buchholz, K. Knill, M. Tamurd, Y. Ohtani, and M. Akamine, "Continuous f0 in the source-excitation generation for HMM-based tts: Do we need voiced/unvoiced classification?," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4724–4727.

[20] Y. Stylianou, *Harmonic plus Noise Models for Speech combined with Statistical Methods, for Speech and Speaker Modification*, Ph.D. thesis, TelecomParis, France, 1996.

[21] J.D. Markel and A.H. Gray, *Linear Prediction of Speech*, Springer Verlag, 1976.

[22] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 8, pp. 1223–1235, 1988.

[23] G. Degottex, Pierre Lanchantin, and Mark Gales, "A pulse model in log-domain for a uniform synthesizer," in *Proc. 9th Speech Synthesis Workshop (SSW9)*, Sunnyvale, CA, USA, September 2016, pp. 230–236.

[24] Gaël Richard and Christophe d'Alessandro, "Analysis/synthesis and modification of the speech aperiodic component," *Speech Communication*, vol. 19, no. 3, pp. 221–244, 1996.

[25] Hideki Kawahara and Reiko Akahane-Yamada, "Perceptual effects of spectral envelope and f0 manipulations using the straight method," *The Journal of the Acoustical Society of America*, vol. 103, no. 5, pp. 2776–2776, 1998.

[26] Hideki Kawahara, "Systematic downgrading for investigating ''naturalness'' in synthesized singing using straight: A high quality vocoder," *The Journal of the Acoustical Society of America*, vol. 111, no. 5, pp. 2334–2334, 2002.

[27] Hideki Kawahara, Hideki Banno, Toshio Irino, and Jiang Jin, "Intelligibility of degraded speech from smeared STRAIGHT spectrum," in *Proc. Interspeech*. 2004, pp. 89–92, ISCA.

[28] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara, "Implementation of realtime straight speech manipulation system: Report on its first implementation," *The Acoustical Society of Japan*, 2007.

[29] Hanna Silén, Elina Helander, Jani Nurminen, and Moncef Gabbouj, "Parameterization of vocal fry in HMM-based speech synthesis," in *Interspeech*, 2009, pp. 1735–1738.

[30] A. Arakawa, Y. Uchimura, H. Banno, F. Itakura, and H. Kawahara, "High quality voice manipulation method based on the vocal tract area function obtained from sub-band lsp of straight spectrum," in *ICASSP*, 2010, pp. 4834–4837.

[31] D. Talkin, "REAPER: Robust Epoch And Pitch EstimatoR [Online]," by Google on Github: https://github.com/google/REAPER, 2015.

[32] G. Degottex and N. Obin, "Phase distortion statistics as a representation of the glottal source: Application to the classification of voice qualities," in *Proc. Interspeech*, 2014, pp. 1633–1637.

[33] M. Koutsogiannaki, O. Simantiraki, G. Degottex, and Y. Stylianou, "The importance of phase on voice quality assessment," in *Proc. Interspeech*, 2014, pp. 1653–1657.

[34] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[35] G. Degottex and Y. Stylianou, "Analysis and synthesis of speech using an adaptive full-band harmonic model," *IEEE Trans. on Audio, Speech and Lang. Proc.*, vol. 21, no. 10, pp. 2085–2095, 2013.

[36] Thomas Drugman, John Kane, and Christer Gobl, "Data-driven detection and analysis of the patterns of creaky voice," *Computer Speech & Language*, vol. 28, no. 5, pp. 1233 – 1253, 2014.

[37] T. Raitio, J. Kane, T. Drugman, and C. Gobl, "HMM-based synthesis of creaky voice," in *in Proc. Interspeech*, 2013, pp. 2316–2320.

[38] N.P. Narendra and K. Sreenivasa Rao, "Generation of creaky voice for improving the quality of HMM-based speech synthesis," *Computer Speech & Language*, vol. 42, pp. 38 – 58, 2017.

[39] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[40] Kelly R. Fitz and Sean A. Fulop, "A unified theory of time-frequency reassignment," *CoRR*, vol. abs/0903.3080, 2009.

[41] Jean Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc., Orlando, FL, USA, 1983.

[42] Alan V. Oppenheim and Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall, 2nd edition, 1978, note that this edition contains a chapter about complex cepstrum which has been removed in the 3rd edition (and seems to come back in the 4th).

[43] H. Valbret, E. Moulines, and J.P. Tubach, "Voice transformation using PSOLA technique," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1992, pp. 145–148.

[44] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, no. 5-6, pp. 453–467, 1990, Neuropeech '89.

[45] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Proc. ISCA Speech Synthesis Workshop*, 2003, pp. 223–224, http://www.festvox.org/cmu_arctic.

[46] Martin Cooke, Catherine Mayo, and Cassia Valentini-botinhao, "Intelligibility enhancing speech modifications: the Hurricane Challenge," in *in Proc. Interspeech*, 2013.

[47] The ITU Radiocommunication Assembly, "ITU-R BS.1284-1: Engeneral methods for the subjective assessment of sound quality," Tech. Rep., ITU, 2003.

[48] C. Callison-Burch and M. Dredze, "Creating speech and language data with amazons mechanical turk," in *Proc. of NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk.* 2010, pp. 1–12, ACL.

[49] Maria K. Wolters, Karl B. Isaac, and Steve Renals, "Evaluating speech synthesis intelligibility using Amazon Mechanical Turk," in *Proc. 7th Speech Synthesis Workshop (SSW7)*, 2010, pp. 136–141.

[50] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, 2015, pp. 4460–4464.

[51] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, 1992, pp. 137–140.

[52] H. Zen, T. Toda, M. Nakamura, and T. Tokuda, "Details of the nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 325–333, 2007.

APPENDIX

A. Detailed results of the listening tests

This appendix shows plots detailing the listening test results for each voice used. Horizontal intervals show the mean's 95% confidence. Solid, dashed and dotted vertical brackets show the corresponding p-value, <0.001, <0.01, <0.05, respectively.

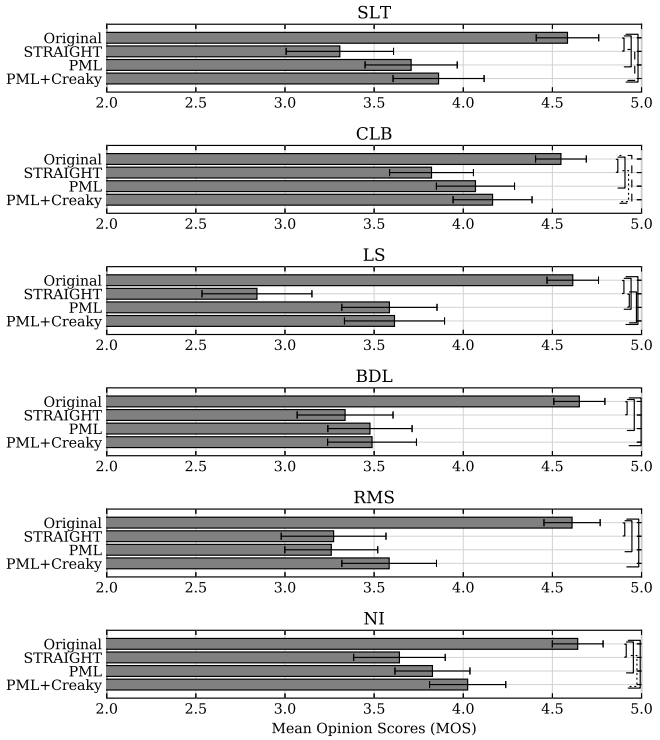


Fig. 11. **Analysis/Re-synthesis**: Mean Opinion Scores (MOS) about the analysis/resynthesis quality of 3 vocoders over 6 voices. 83 listeners participated to the test (with the 95% confidence intervals).

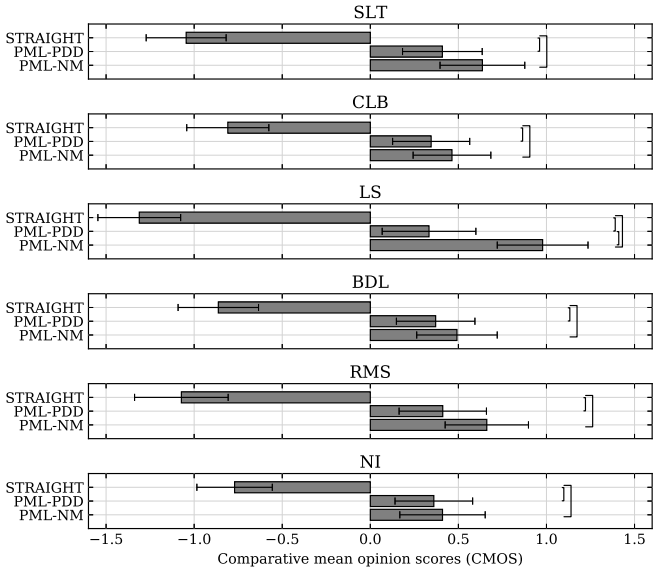


Fig. 12. **Phase Distortion Deviation (PDD) vs. Noise Mask (NM)**: Results of SPSS listening test over 6 voices comparing: Baseline STRAIGHT; PML synthesis using PDD-based training ; PML synthesis using NM-based training. 47 listeners took this test.

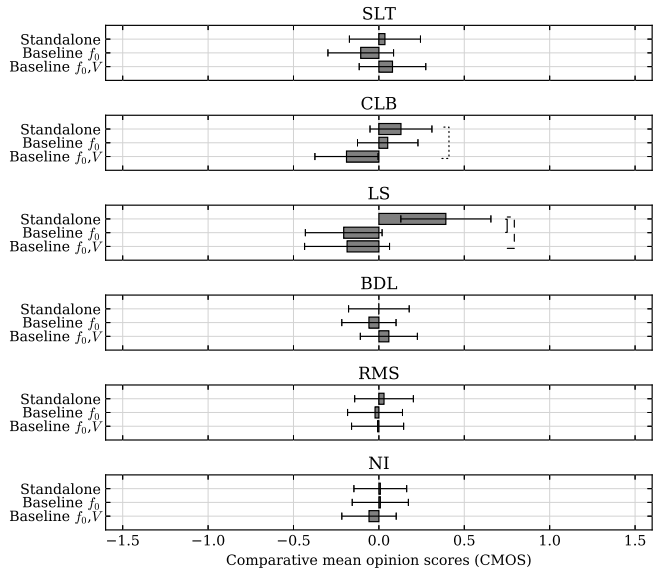


Fig. 13. **PML-Standalone vs. Streams mixture**: Results of SPSS listening test over 6 voices comparing PML-based synthesis using 3 different stream setups: PML Standalone (f_0, Spectrum and Noise Mask features generated from PML-based training); Baseline f_0 (as in Standalone, except the f_0 that is generated using the STRAIGHT-based training; Baseline f_0,Spec. (f_0 and Spectrum generated using the STRAIGHT-based training and Noise Mask generated by the PML-based training). 45 listeners took this test.

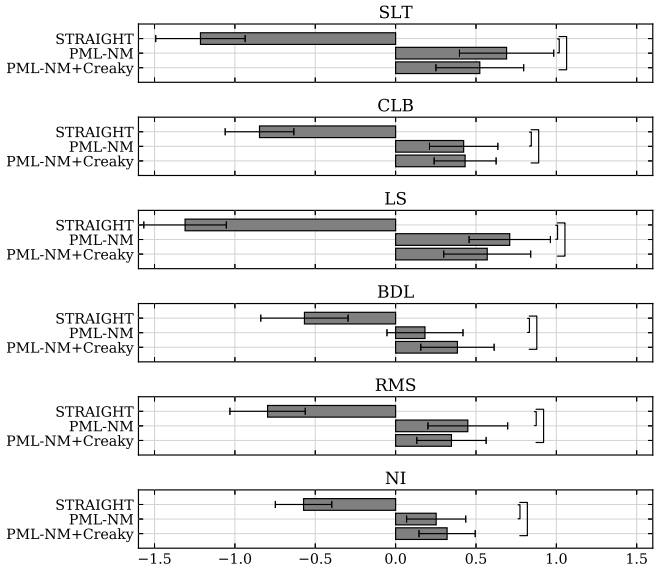


Fig. 14. **Creaky voice correction**: Results of SPSS listening test over 6 voices comparing: STRAIGHT vocoder; the suggested synthesizer PML; PML with the creaky voice correction on the noise mask. 44 listeners took this test.

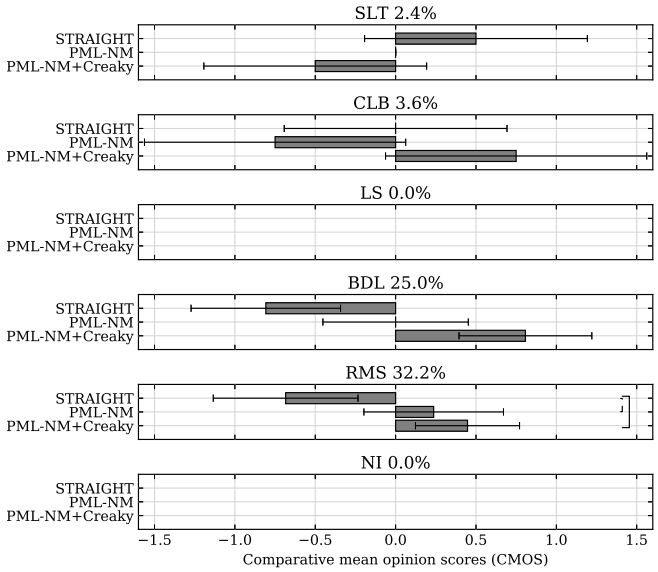
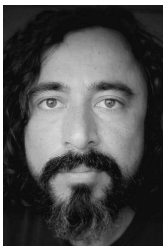


Fig. 15. **Creaky voice correction**: Similar results as in Fig. 14, except that only the sentences with more than 6% creakiness are kept for these results. Percentage of sentences kept for these results are shown in the titles of the plots.

Gilles Degottex received the Diploma degree in computer science from University of Neuchâtel (UniNE), Switzerland. After a one-year specialization at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, he obtained his Ph.D. degree in 2010 at the Institut de Recherche et Coordination Acoustique/Musique, IRCAM/UPMC, Paris, France. He held a postdoctoral position at University of Crete, Heraklion, Greece, on voice modeling, transformation and synthesis funded by the Swiss National Science Foundation (SNSF) and Foundation for Research & Technology - Hellas (FORTH), Greece. He held a 2nd postdoctoral position at Ircam, Paris, France on singing voice synthesis in the national ChaNTeR project. He is currently Research Fellow funded by the Marie Sklodowska-Curie Action (MSCA) program EU at the University of Cambridge, UK. His research interests include voice source features, sinusoidal and spectral modeling for speech and singing voice synthesis and transformation.



Pierre Lanchantin (M'11) is Research Associate in the Speech Research Group at the Cambridge University Engineering Department (CUED). He received a MSc degree in Acoustics, Signal Processing and Computer science applied to Music from Paris VI University and a Ph.D. in Statistical Signal Processing from Telecom SudParis, France. His research interests include statistical modeling of signals, speech processing and their applications to music. During his Ph.D., he studied generalizations of hidden Markov models (HMM) called Pairwise and Triplet Markov chains with applications to image segmentation. He then directed his research toward speech processing and joined the Institute for Research and Coordination in Acoustics & Music (IRCAM), working on speech recognition, speech synthesis and voice conversion. He is currently working on advanced learning and adaptation techniques for speech recognition and synthesis.

Mark Gales studied for the B.A. in Electrical and Information Sciences at the University of Cambridge from 1985-88. Following graduation he worked as a consultant at Roke Manor Research Ltd. In 1991 he took up a position as a Research Associate in the Speech Vision and Robotics group in the Engineering Department at Cambridge University. In 1995 he completed his doctoral thesis: Model-Based Techniques for Robust Speech Recognition supervised by Professor Steve Young. From 1995-1997 he was a Research Fellow at Emmanuel College Cambridge. He was then a Research Staff Member in the Speech group at the IBM T.J.Watson Research Center until 1999 when he returned to Cambridge University Engineering Department as a University Lecturer. He was appointed Reader in Information Engineering in 2004. He is currently a Professor of Information Engineering and a College Lecturer and Official Fellow of Emmanuel College. Mark Gales is a Fellow of the IEEE and a member of the Speech and Language Processing Technical Committee (2015-2017, previously a member from 2001-2004). He was an associate editor for IEEE Signal Processing Letters from 2008-2011 and IEEE Transactions on Audio Speech and Language Processing from 2009-2013. He is currently on the Editorial Board of Computer Speech and Language. He has been awarded a number of paper awards, including a 1997 IEEE Young Author Paper Award for his paper on Parallel Model Combination and a 2002 IEEE Paper Award for his paper on Semi-Tied Covariance Matrices.