# Phase-Vocoder: About this phasiness business

*Jean Laroche and Mark Dolson*

Joint E-mu/Creative Technology Center
1600 Green Hills Road
Scotts Valley, CA95006
Email: jeanl@emu.com markd@emu.com

## ABSTRACT

The phase-vocoder is a well-known tool for the frequency domain processing of speech or audio signals, with applications such as time compression or expansion, pitch-scale modification, noise reduction, etc. In the context of time-scale or pitch-scale modification, the phase-vocoder is usually considered to yield high quality results, especially when large modification factors are used on polyphonic or non-pitched signals. However, the phase-vocoder is also known for an artifact that plagues its output, and has been described in the literature as either "phasiness", "reverberation", or "loss of presence". Recent research has been devoted to understanding and reducing this artifact, and solutions have been proposed which either significantly improve the quality of the output at the cost of a very high additional computation time, or are inexpensive but only marginally effective. This paper examines the problem of phasiness in the context of time-scale modification of signals, and presents two new phase synchronization schemes which are shown to both significantly improve the sound quality, and reduce the computational cost of such modifications.

## 1. Introduction

Time-scale and pitch-scale modification of signals has always been a subject of interest in the audio community. By contrast with time-domain techniques [1], frequency-domain techniques, and the phase-vocoder in particular, can process polyphonic signals, with large modification factors. However, the phase-vocoder exhibits an artifact that time-domain techniques do not: phasiness. Phasiness or reverberation or "loss of presence" relates to the fact that the modified signal often sounds as if it had been recorded in a small room. In particular, time-expanded speech sounds like the speaker is much further from the microphone than it was in the original sound.

The problem of phasiness has been observed by many authors and a few solutions have been proposed which either significantly improve the quality of the output at the cost of a very high additional computation time, or are inexpensive but only marginally effective. This paper proposes an explanation for the presence of phasiness in time-scaled signals, and offers new phase calculation techniques that are shown to significantly reduce the problem. In addition, the new techniques make it possible to reduce the computational cost of the phase vocoder by a factor larger than 2.

## 2. The basic phase-vocoder time scaling algorithm

Because pitch-scale modifications can be done by combining time-scaling and sample rate conversion, we will focus on time-scaling. The reader can refer to [2] or [1] for a detailed description of the standard phase-vocoder techniques, only a brief outline will be given here.

### 2.1. Phase-vocoder time-scaling

Phase-vocoder based time-scaling techniques involve an analysis stage, a modification stage and a resynthesis stage. During the analysis stage, analysis time-instants $t_a^u$ for successive values of integer $u$ are set along the original signal, possibly uniformly: $t_a^u = uR_a$ where $R_a$ is the so-called analysis hop-factor. At each of these analysis time-instants, a Fourier transform is calculated over a windowed portion of the original signal, centered around $t_a^u$, yielding what is usually called a short-time Fourier transform (STFT) representation of the signal, denoted $X(t_a^u, \Omega_k)$, where $x$ is the original signal, $\Omega_k = \frac{2\pi k}{N}$ is the center frequency of the $k$-th vocoder "channel" and $N$ is the size of the discrete Fourier transform. The resynthesis stage involves setting synthesis time-instants $t_s^u$, usually uniformly: $t_s^u = R_s u$ where $R_s$ is the synthesis hop-factor. At each of these synthesis time-instants, a short-time signals $y_u(n)$ is synthesized, based on synthesis STFT values $Y(t_s^u, \Omega_k)$, and all of these short-time signals are summed together, after applying an optional synthesis window, yielding the output signal $y(n)$.
Time-scaling involves using an analysis hop-factor $R_a$ different from the synthesis hop-factor $R_s$ and setting $|Y(t_s^u, \Omega_k)| = |X(t_a^u, \Omega_k)|$ so that the amplitude of any given sinusoid in the output signal at time $t_s^u = R_s u$ will be the same as in the input signal at time $t_a^u = R_a u$: the time-evolution of the amplitudes is modified. To calculate the phase of $Y(t_s^u, \Omega_k)$, the standard phase-vocoder technique requires phase-unwrapping, a process whereby the phase-increment between two consecutive frames is used to estimate the instantaneous frequency of a nearby sinusoid in each channel. The instantaneous frequency $\hat{\omega}_k(t_a^u)$ is estimated by first calculating the *heterodyned* phase increment

$$\Delta\Phi_k^u = \angle X(t_a^u, \Omega_k) - \angle X(t_a^{u-1}, \Omega_k) - R_a\Omega_k$$

then taking its principal determination (between $\pm\pi$) denoted $\Delta_p\Phi_k^u$ and using the following equation

$$\hat{\omega}_k(t_a^u) = \Omega_k + \frac{1}{R_a}\Delta_p\Phi_k^u \quad (1)$$

Once the instantaneous frequency at time $t_a^u$ is estimated, the phase of the time-scaled STFT at time $t_s^u$ is set according to

the following *phase-propagation* formula

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^{u-1}, \Omega_k) + R_s \hat{\omega}_k(t_a^u) \quad (2)$$

Finally the output signal is obtained by synthesizing and overlap-adding short-time signals corresponding to each STFT frame. Equation (2) makes sure that the short-time signals will overlap coherently.
Because phase propagation errors are at the heart of many of the sound quality issues in the phase vocoder, it is important to understand how sinusoidal phases are altered by vocoder-based time-scale modifications. This is the topic of the next section.

## 2.2. Phase problems in phase-vocoder time-scaling

**Phase coherence** Because the STFT frames overlap, the stream of STFT values of a given signal must satisfy strong consistency conditions, especially with regard to phase. Phase consistency conditions, or phase-coherence, exist not only from frame to frame ("horizontal phase-coherence"), but also within any given frame, between neighboring channels ("vertical phase-coherence"). If phase-coherence is not preserved in the series of synthesis STFT $Y(t_s^u, \Omega_k)$, the synthesis will yield a signal whose STFT will not be close to $Y(t_s^u, \Omega_k)$, and will most certainly sound phasy. For a constant-amplitude, constant-frequency sinusoid, the channels located around the sinusoidal frequency all have identical analysis phases (or their phases have a $\pm\pi$ alternation, if the analysis window is non-zero only for $t > 0$, as is usually the case). For a sinusoid with a slowly varying frequency, it is found that the phases in channels around the instantaneous frequency are nearly equal, although no analytical formula could be found.

**Output phase vs input phase** In this section, we seek to relate the phase of the modified short-time Fourier transform in channel $k$ to the phase of the corresponding analysis short-time Fourier transform in the same channel. Assuming a constant modification factor $\alpha = \frac{R_s}{R_a}$, and given an initial synthesis Fourier transform phase $\phi_s(0)$, we can iterate equation (2) for successive values of $u$, starting at $u = 0$, use equation (1), and after straightforward algebra, get:

$$\angle Y(t_s^u, k) = \phi_s(0, k) + \alpha\left[\angle X(t_a^u, k) - \angle X(0, k)\right] + \alpha \sum_{i=1}^{u} 2m_k^i \pi \quad (3)$$

where $m_k^i$ is the unwrapping factor at the analysis time-instant $t_a^i$ ($2m_k^i\pi = \Delta_p\Phi_k^i - \Delta\Phi_k^i$). Two important conclusions can be drawn from this equation: 1) Contrary to popular belief (!), if an analysis phase is mis-estimated at a given analysis time-instant, provided that the phase-unwrapping factor $m_k^i$ remains correct, this mis-estimation will not generate any phase drift in subsequent frames, but only have an effect local to the frame in question. 2) Potential phase-unwrapping errors manifest themselves by multiples of $2\alpha\pi$ being added to the synthesis phase. If $\alpha$ is an integer, phase-unwrapping errors are transparent since they always are multiples of $2\pi$. As a result, integer-factor time-scaling operations can be performed *without phase unwrapping* by use of equation (3) where the factor $\sum_{i=1}^{u} 2m_k^i\pi$ is dropped. Skipping the phase-unwrapping stage significantly reduces the computation cost of such modifications.

**Phase jumps due to channel crossing** Let us first assume that the modification factor $\alpha$ is a constant integer, so phase-unwrapping errors do not influence the modified signal. If a sinusoidal component in the original signal always falls in the vicinity of a given vocoder channel, then $\angle X(t_a^u) = \phi_i(t_a^u) + 2l\pi$ at all times, where $\phi_i(t)$ is the sinusoid's instantaneous phase at time $t$. As a result, Equation (3) now reads:

$$\angle Y(t_s^u, \Omega_k) = \phi_s(0, k) + \alpha\left[\phi_i(t_a^u) - \phi_i(0)\right] \quad (4)$$

where the integer number of $2\pi$ has been dropped. Equation (4) shows that depending on the choice of $\phi_s(0, k)$, the synthesized phases in neighboring channels may not be close: the vertical phase-coherence described in section may be lost, resulting in potential phasiness. For adjacent phases to be equal, *the initial synthesis phases of the channels around channel $k$ must be equal*:

$$\phi_s(0, k) = \phi_s(0, k') \quad \forall k, k' \quad (5)$$

Now imagine that the instantaneous frequency of the sinusoid slowly sweeps across several channels, say from channel $k_0$ to channel $k_0 + 10$. Assuming that the sinusoid falls in channel $k$ at time $t_a^u$, equation (3) can be rewritten as:

$$\angle Y(t_s^u, \Omega_k) = \phi_s(0, k) + \alpha\left[\phi_i(t_a^u) - \angle X(0, \Omega_k)\right] \quad (6)$$

Compared to equation (4), we now have the additional problem that $\angle X(0, \Omega_k)$ may take on different values for different channels $k$ because the channels may be quite distant from channel $k_0$ where the sinusoid initially fell and may have been influenced initially by *another sinusoid*, so that equation (6) reveals two potential problems: 1) The phases in adjacent channels around the sinusoidal channel $k$ are not necessarily equal or close unless $\theta_k \approx \theta_{k'}$ where

$$\theta_k = \phi_s(0, k) - \alpha\angle X(0, \Omega_k) \quad (7)$$

2) When the sinusoid's instantaneous frequency migrates from channel $k$ to channel $k+1$, the synthesis phase undergoes a phase jump equal to $\theta_k - \theta_{k'}$. However, when $\theta_k = C \quad \forall k$, with $C$ a channel-independent constant, the synthesis phase becomes $\angle Y(t_s^u, \Omega_k) = C + \alpha\phi_i(t_a^u)$ which is consistent with the ideal synthesis phase.

Finally, for non-integer modification factors, there is the additional problem of the phase-unwrapping terms in each channel. When the sinusoid sweeps from channel $k_0$ to channel $k_0 + 10$ between times $t_a^0$ and $t_a^u$, is it very unlikely that the unwrapping factors in channel $k_0 + 10$ will be identical to those in channel $k_0$ at each synthesis time-instant. As a result, the synthesis phase will show additional jumps $\alpha 2M\pi$ from one channel to the next, thus preventing vertical as well as horizontal phase-coherence.

Considering the results above, it seems truly amazing that the phase vocoder should work at all! When a sinusoid sweeps rapidly across channels, its phase is likely to undergo rapid shifts from channel to channel, and these shifts are responsible for a large part of the phasiness problem. For integer

modification factors however, making sure that $\theta_k = C \ \forall k$, for example by setting $\phi_s(0,k) = \alpha \angle X(0,\Omega_k)$, guarantees that all channel-crossing problems are eliminated and yields very high-quality modified signals.

Below is an example of what can go wrong when a sinusoid sweeps across channels. The signal is a sinusoid with a constant amplitude, whose frequency sweeps from the center frequency of channel 30, $\Omega_{30} = \frac{2\pi 30}{N}$ to the center frequency of channel 40 in 10240 samples. The FFT size was 1024, the input hop factor was 128 and the output hop factor was 256, resulting in a factor-2 time-stretching. The standard phase-vocoder technique described above was used, and the initial synthesis phases were set to be equal to the initial analysis phases $\phi_s(0,k) = \angle X(0,\Omega_k)$ which is a standard initialization choice. The analysis and the synthesis windows were Hanning windows with a size equal to the FFT size. Figure 1 shows the amplitude envelope of the resulting signal in the time-domain. Figure 2 shows the analysis and synthesis phases for successive short-time Fourier transform frames, measured at the peak of the Fourier transform.

The modified sinusoid is strongly amplitude-modulated,

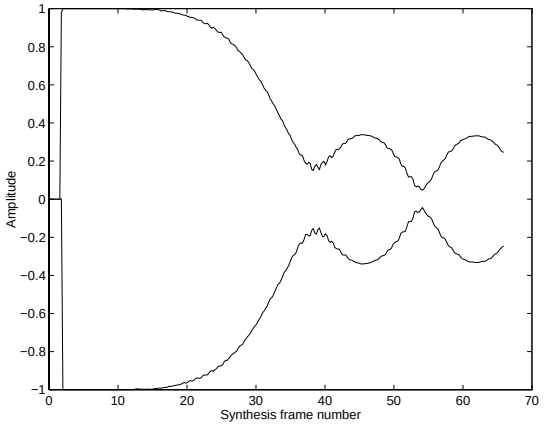


Figure 1: Factor-2 time-scaling of a constant amplitude chirp. Time-domain amplitude-envelope of the modified signal.

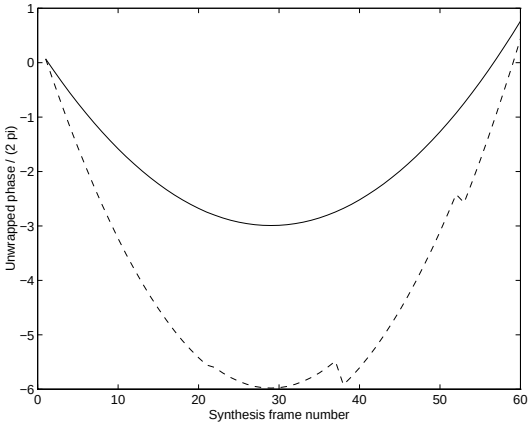


Figure 2: Analysis phase (solid line) and synthesis phase (dotted line) in numbers of $2\pi$, showing phase jumps at frames 22, 38 and 53.

a result of the lack of phase-coherence. While the analysis phase shows the characteristic parabolic shape due to a linearly varying frequency, the synthesis phase exhibits "discontinuities" at frames 38, 53 and to a lesser extent 22. These phase jumps result from $\theta_k$ not being a constant across channels, and it can be easily verified that phase jumps occur when the instantaneous frequency of the chirp jumps from one channel to the next one. Note that since the modification factor was an integer, phase-unwrapping problems cannot be blamed in this case. With a proper choice of initial conditions $\phi_s(0,k) = \alpha \angle X(0,\Omega_k)$, the time-scaling operation yields the signal shown in figure 3, where the amplitude modulation almost completely disappeared.

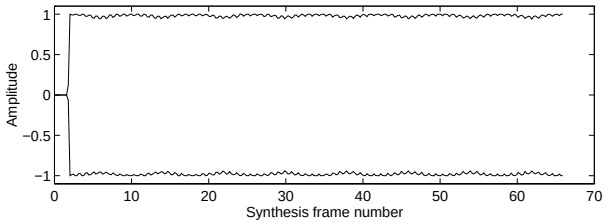


Figure 3: Factor-2 time-scaling of a constant amplitude chirp, "correct" initial phases. Time-domain amplitude-envelope of the modified signal.

The following section presents some of the solutions that have been proposed to solve the phasiness problem, along with 2 new phase-synchronization techniques.

## 3. Old and new strategies for reducing phasiness

### 3.1. Magnitude-only reconstruction

The authors in [3] propose to simply discard either the phase or the magnitude of the series of synthesis STFT, and to use iterative techniques to estimate values that would make the series a consistent one. In the context of time-scale modification, these techniques do not always significantly reduce phasiness, and are extremely costly in terms of computations, which renders them unsuitable for practical applications. Moreover, the resulting signals often sound "rough", a problem mentioned in the original article.

### 3.2. Loose phase-locking

Puckette in [4] recognized that for a constant-frequency constant-amplitude sinusoid the synthesis phases around the maximum of the Fourier transform should exhibit $\pm\pi$ alternations and proposed a very simple way to constrain them to do so. The technique used in conjunction with an alternative phase-updating procedure turns out to be extremely cheap in terms of computations, requiring only a few additional multiplications per channel. Informal listening tests showed that phasiness is generally reduced to a degree which depends on the original signal. However, the increase in quality is only limited. This idea inspired the new techniques presented below.

### 3.3. Peak phase-locking

The new phase-updating technique begins with a coarse peak-picking stage where vocoder channels are searched for local

maxima. In the simplest implementation, a channel whose amplitude is larger than its 4 nearest neighbors is said to be a peak, a criterion which is both simple and cost-effective. For each peak $\Omega_{k_l}$ and only for the peaks, the synthesis phase is calculated, according to the standard phase-propagation equation (2). The series of peaks subdivides the frequency axis into "regions of influence" located around each peak, which include channels whose phases will be "locked" in some way to that of the peak. In our experiments, the upper limit of the region around peak $\Omega_{k_l}$ was set to the middle frequency between that peak and the next one $(\Omega_{k_l}+\Omega_{k_{l+1}})/2$. Another reasonable choice would be the channel of lowest amplitude between the two peaks.

Identity phase-locking consists of constraining the synthesis phases around the peak to bear the same relations the analysis phases did. If $\Omega_{k_l}$ is the center frequency of the dominant peak, we set:

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^u, \Omega_{k_l}) + \angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_l}) \quad (8)$$

for all channel $k$ in the peak's region of influence, as defined above. This phase-locking mechanism improves significantly the consistency of the resulting series of STFT, and greatly reduces the amount of phasiness in the modified signal.
It also has two major computational advantages: Because phase-unwrapping is only performed on peak channels, one is always sure that the instantaneous frequency of the underlying sinusoid is very close to the center frequency of the channel in question, which means that input overlaps as small as 50% can be used without generating phase-unwrapping errors: compared to the standard 75% overlap, this is a factor 2 speed up! In addition, the new technique requires trigonometric calculations *only for peak channels*: once the synthesis phase of the peak channel has been determined, one can calculate the rotation $\theta$ required to rotate $X(t_a^u, \Omega_{k_l})$ into $Y(t_s^u, \Omega_{k_l})$, then calculate the phasor $Z = e^{j\theta}$ and obtain the neighboring channels by use of simple complex algebra: $Y(t_s^u, \Omega_k) = ZX(t_a^u, \Omega_k)$ which can be easily shown to satisfy the phase-locking equation (8): *neighboring channels only require one complex multiply*!

An improvement over the preceding technique consists of "following" peaks as they move across channels. Before calculating the new peak phase, the idea is to look for the corresponding peak in the preceding frame, and use its analysis and synthesis phases in the phase-unwrapping and phase-propagation equations (1) and (2). The peak in frame $u-1$ that corresponds to a peak $k_0$ in frame $u$ can be defined as the dominant peak in the frequency region to which channel $k_0$ belonged in frame $u-1$. This technique has the advantage of "locking" the peak synthesis phase to its analysis phase, thus preventing slow phase drifts that exist in the preceding technique [5]
In addition, a possible phase-locking solution consists of setting

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^u, \Omega_{k_l}) + \beta\,[\angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_l})] \quad (9)$$

after vertically unwrapping the analysis phases $\angle X(t_a^u, \Omega_k)$ around each peak. $\beta$ is a parameter between 1 and $\alpha$, which can be adjusted to further reduce phasiness. When 75% overlap is used, $\beta = \alpha$ consistently yields better results than $\beta = 1$, identity phase-locking. For 50% overlap, $\beta$ must be closer to 1 to avoid undesirable roughness. This phase-locking option is more costly than identity phase-locking since the single multiply trick can no longer be used.

## 3.4. Results and Conclusion

Informal listening tests have shown that the two phase-locking techniques above dramatically reduce the phasiness in the modified signal. Compared to the loose phase-locking of [4], both techniques perform significantly better, with the latter (equation (9)) consistently yielding better results than the former (equation (8)). On speech signals, the modified voice has much more presence, and non-voiced segments sound more natural with the latter than with the former. For integer modification factors, the standard algorithm *without phase-unwrapping* but with proper initial conditions (such that $\theta_k = C\ \forall k$) yields very high-quality results, which only the second phase-locking technique can match for non-integer modification factors. Some residual phasiness can still be heard in the modified signals, especially for larger modification factors ($\alpha > 3$).
The new phase-synchronization techniques make it possible to use 50% overlap, a cost reduction of a factor larger than 2 over the standard 75% overlap constraint.

In this paper, we have shown how input and output phases are related in the standard phase-vocoder time-scaling techniques, and brought into light potential phase problems occurring when sinusoids sweep across adjacent channels. We have shown the important role played by phase initialization for modifications by integer factors, and proposed two peak-synchronization techniques that dramatically reduce phasiness/reverberation in the modified signal, while making it possible to cut the computational cost by a factor larger than 2. The residual phasiness or reverberation present in the output signal, even when phase-synchronization is used, may have several potential causes. In any case, it seems clear that unlike what is done in standard techniques, STFT magnitudes should also be modified when time-scaling is performed: a time-scaled sweeping sinewave has a central lobe whose width varies as a function of the time-scaling factor.

## References


1. E. Moulines and J. Laroche, "Non parametric techniques for pitch-scale and time-scale modification of speech.," *Speech Communication*, vol. 16, pp. 175–205, Feb 1995.
2. M. Dolson, "The phase vocoder: A tutorial," *Computer Music J.*, vol. 10, no. 4, pp. 14–27, 1986.
3. D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 236–243, Apr 1984.
4. M. Puckette, "Phase-locked vocoder," *IEEE ASSP Workshop on app. of sig. proc. to audio and acous.*, 1995.
5. J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *to appear in trans. speech and audio proc.*