WAVE PULSE PHASE MODULATION: HYBRIDISING PHASE MODULATION AND PHASE DISTORTION

Matthew Smart

School of Arts and Humanities
University of Huddersfield
Huddersfield, UK
matthew.smart2001@gmail.com

ABSTRACT

This paper introduces Wave Pulse Phase Modulation (WPPM), a novel synthesis technique based on phase shaping. It combines two classic digital synthesis techniques: Phase Modulation (PM) and Phase Distortion (PD), aiming to overcome their respective limitations while enabling the creation of new, interesting timbres. It works by segmenting a phase signal into two regions, each independently driving the phase of a modulator waveform. This results in two distinct pulses per period that together form the signal used as the phase input to a carrier waveform, similar to PM, hence the name Wave Pulse Phase Modulation. This method provides a minimal set of parameters that enable the creation of complex, evolving waveforms, and rich dynamic textures. By modulating these parameters, WPPM can produce a wide range of interesting spectra, including those with formant-like resonant peaks. The paper examines PM and PD in detail, exploring the modifications needed to integrate them with WPPM, before presenting the full WPPM algorithm alongside its parameters and creative possibilities. Finally, it discusses scope for further research and developments into new similar phase shaping algorithms.

1. INTRODUCTION

There are two well-established phase shaping techniques [1] that can be commonly found in digital synthesisers: Phase Modulation (PM) [2] and Phase Distortion (PD) [3]. Both techniques typically use a sinusoidal wave as a carrier wave, but traverse its phase with a non-linear phase shaped signal, thus creating more complex waveforms and timbral variation. In PM, two oscillators are used, with the phase of the carrier being driven by the additive combination of a secondary waveform, called the modulator, and a linear phase signal that determines the carrier frequency. PD involves adding non-linear transformations to a phase signal used to drive a carrier waveform.

Both techniques come with limitations and offer scope for further developments. PM can produce relatively predictable tones unless complex configurations involving multiple modulator and carrier waveforms are used [2]. However, such configurations can result in an overwhelming number of parameters, leading to a less intuitive user experience. PD, by contrast, offers limited parametric control within a single algorithm, often providing only a single morphing parameter that is available for modulation. For example, on Casio CZ series synthesisers [4], users might morph between sinusoidal and sawtooth waveforms, but creating a new timbre requires the user to select an entirely different algorithm, with no possibility for continuous control between the timbres produced by different algorithms.

This paper introduces a novel synthesis method based on phase shaping that combines elements of PM and PD [5, 6], seeking to address their respective limitations. This approach involves using a PD-style morphing parameter to segment a linear phase signal into two regions, each of which is independently used to drive the phase of a modulator waveform. This creates two distinct PM pulses within a single period of the fundamental frequency, which when additively combined with the phase distorted signal are used to traverse the phase of the carrier waveform. This combination of pulses and phase modulation techniques leads to the name: Wave Pulse Phase Modulation (WPPM).

WPPM offers control over both PM modulator pulses, along with a single highly expressive parameter for morphing between the two by controlling their respective durations in relation to the carrier frequency. Thus, a minimal selection of continuous parameters are provided, making the algorithm easy to control whilst producing a spectrally rich sound palette including resonant formant-like spectra.

2. WAVE PULSE PHASE MODULATION COMPONENTS

Throughout this paper, a number of non-linear phase signal functions are outlined, which are designated $F_n(t)$. Each of these phase shaped signals are used to drive the phase of an inverted cosine wave shown in Equation (1) to generate the final output waveform. The inverted cosine is chosen to match the outputs illustrated in [4]; however, there is potential to use other sinusoidal waveforms or to experiment with more complex waveforms.

$$h(t) = -\cos\left(2\pi F_n(t)\right) \tag{1}$$

2.1. Phase Modulation Synthesis

The classic Phase Modulation (PM) synthesis technique, often mistaken for Frequency Modulation (FM) due to their similar timbral characteristics, was first implemented by Yamaha in their DX series synthesisers [2, 7, 8]. In FM, an audio-rate waveform modulates the frequency of a carrier signal, with the modulation depth, called the index, defined relative to the fundamental frequency. In contrast, PM operates by additively combining a modulation waveform with a linear phase signal, using the resultant signal to drive the phase of a carrier waveform. This approach yields a similar output, while offering a simpler implementation, as the modulator

index does not need to be scaled relative to the fundamental frequency.
The algorithm for classic PM synthesis is described in [5] as:

$$e = sin(\alpha t + I sin(\beta t)) \tag{2}$$

where

e = the instantaneous amplitude of the output

α = the carrier frequency in rad/s

β = the modulator frequency in rad/s

I = the modulation index, which defines the depth of modulation applied by the modulator waveform.

In this implementation, higher modulation frequencies produce a brighter sound, while increasing the index enhances both brightness and spectral density. When $I = 0$, no modulation is applied and the output remains a pure sinusoidal wave. When the modulation frequency is not an integer multiple or division of the fundamental frequency, the resulting spectrum becomes inharmonic, making it suitable for synthesising bell-like tones. This behaviour is undesirable within the context of the WPPM algorithm, necessitating a modification for seamless integration.

2.2. Modified Phase Modulation

To enhance the suitability of the PM algorithm for WPPM, Equation 2 can be modified and written in terms of phase to produce Equation (3):

$$F_1(\phi[n]) = \phi[n] + I \sin(2\pi \cdot R\phi[n]) \tag{3}$$

where

$F_1(\phi[n])$ = the modulated phase signal

n = the time sample index

I = the modulation index

R = the ratio of the modulator frequency to the fundamental frequency.

The phase signal $\phi[n]$ is a linear ramp wave $0 \leq \phi[n] \leq 1$ with fundamental frequency f_0 and sampling frequency f_s defined by:

$$\phi[n] = \left(\frac{f_0}{f_s} + \phi[n-1]\right) \bmod 1 \tag{4}$$

Like Equation (2), the timbre of Equation (3) is determined by R and I, producing a pure sinusoidal wave when substituted into Equation (1) if $I = 0$.
The modified PM equation differs from Equation (2) by using the same linear phase signal to drive both the modulator and carrier wave, rather than having an independently driven modulator waveform. This ensures that the modulator wave always starts at a consistent phase, regardless of its frequency ratio R. In the case of a sine wave with no phase offset, the modulator has an amplitude of zero at the start of each carrier period.
This modification prevents the algorithm from correctly producing modulation frequencies lower than the fundamental frequency, as, similar to a hard-sync effect, the phase is reset at the start of each carrier period. Consequently, this prevents the inharmonic bell-like tones typically produced by conventional PM when $R \notin \mathbb{Z}$. Instead, it introduces an abrupt discontinuity in the phase signal, resulting in a spectrum resembling that of a sawtooth wave, and is more susceptible to aliasing [1]. For $R \in \mathbb{Z}$ and $R \geq 1$, the modulation waveform completes exactly R full cycles within each carrier period, avoiding these discontinuities and thereby reducing aliasing artifacts.
To avoid non-integer R values producing an output susceptible to aliasing, the modulator wave can be multiplied by a window function [9] driven by the linear phase signal. This ensures that the modulator always begins and ends at an amplitude of zero regardless of its shape, frequency or phase offset.
Throughout this paper, a sinusoidal window function defined by:

$$w(t) = \sin(\pi t) \tag{5}$$

is used each example, though it can be easily replaced with a more complex window such as those listed in [9].
Applying this window function to the PM algorithm defined by Equation (3) produces:

$$F_2(\phi[n]) = \phi[n] + w(\phi[n]) \cdot I \sin(2\pi \cdot R\phi[n]) \tag{6}$$

Using a sine wave as the modulator ensures that the amplitude is always zero when $\phi[n] = 0$, provided no phase offset is applied. This offers greater flexibility when selecting a window function, as the initial value can be arbitrary as long as the window function smoothly transitions to zero at the end. For example, an inverted ramp function defined by

$$w(\phi[n]) = 1 - \phi[n]$$

could be used as an alternative.
Applying a window function enhances the tonal flexibility of PM by enabling non-integer R values, including modulator frequencies lower than the fundamental ($R < 1$), without introducing inharmonic content or phase discontinuities. This is especially beneficial in the context of WPPM, as it eliminates the dependence on specific R values for harmonic outputs, thus allowing for continuous modulation of R while maintaining musically useful sonic results.
Different window functions shape the spectral characteristics of the output, making their selection an important factor in tonal design. Adjustable window functions, such as Gaussian windows, offer dynamic control over their shape, and can expand the expressive control set of the modified PM synthesis algorithm.
Fig. 1 compares the effect of an integer and a non-integer R value without windowing, whereas Fig. 2 illustrates the same R values with a sinusoidal window applied.

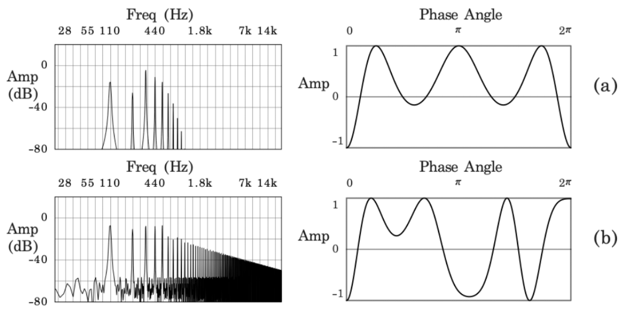

Figure 1: *Phase Modulated cosine wave ($f_s = 44100$, $f_0 = 110$ (A2)) with an integer R value [(a):$R = 1$, $I = 0.5$] and with a non-integer R value [(b): $R = 1.3$, $I = 0.5$].*

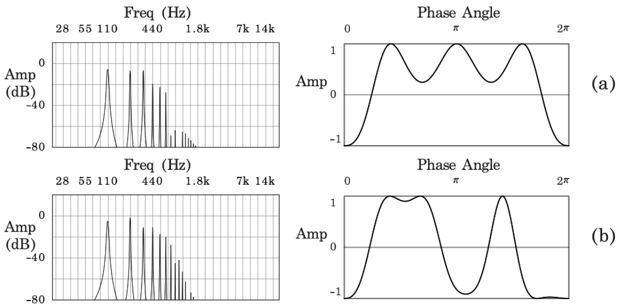

Figure 2: *Sine windowed Phase Modulated cosine wave (f_s = 44100, f_0 = 110 (A2)) with an integer R value [(a): $R = 1$, $I = 0.5$] and with a non-integer R value [(b): $R = 1.3$, $I = 0.5$].*

2.3. Phase Distortion Synthesis

Phase Distortion (PD) is a synthesis technique based on reshaping and adding non-linearities to a phase signal used to traverse a carrier waveform [3]. First introduced in Casio's CZ series synthesisers [4], PD is commonly used to recreate analogue waveforms [10] and emulate subtractive synthesis by dynamically adjusting one or more breakpoints in a phase signal driving a sinusoidal waveform. This process transforms a simple sine wave into a spectrally richer waveform, mimicking the effect of opening a lowpass filter, without the need for complex digital filter maths, or expensive analogue filter components.

The Casio CZ series includes eight distinct waveforms, each generated by a unique PD algorithm. However, the manual [4] primarily explains the phase signal algorithm needed to generate a sawtooth wave when indexing a cosine lookup table, for which an equation is proposed in [6]:

$$F_3(\phi[n]) = \begin{cases} \frac{\phi[n]}{2d}, & \phi(n) < d \\ \frac{1}{2} + \frac{(\phi[n]-d)}{2(1-d)}, & d \leq \phi(n) \end{cases} \quad (7)$$

where d is the position of the breakpoint on the horizontal axis constrained to $0 \leq d \leq 1$.

At $d = 0.5$, the phase signal remains linear, producing a pure sine wave when substituted into Equation (1). As d decreases toward 0, the waveform becomes brighter, resembling the effect of opening a lowpass filter on a sawtooth wave. At $d = 0$, the resulting waveform is almost spectrally identical to a sawtooth wave. Due to the symmetry of the cosine lookup table at its midpoint (π radians), increasing d toward 1 produces a mirrored version of the same transformation. Fig. 3 illustrates this effect, although extreme values of d have been omitted to suppress aliasing.

3. WAVE PULSE PHASE MODULATION

3.1. Algorithm and Implementation

The algorithm in Equation (7) uses a piecewise equation to divide $\phi[n]$ into two related phase signals, with their duration ratio controlled by d. These signals are then scaled to form a continuous PD signal when played in succession. However, by maintaining both phase signals within the amplitude range [0,1], they can be used independently in the PM algorithm in Equation (6) to generate two wave pulses, the combined duration of which equals one carrier period.

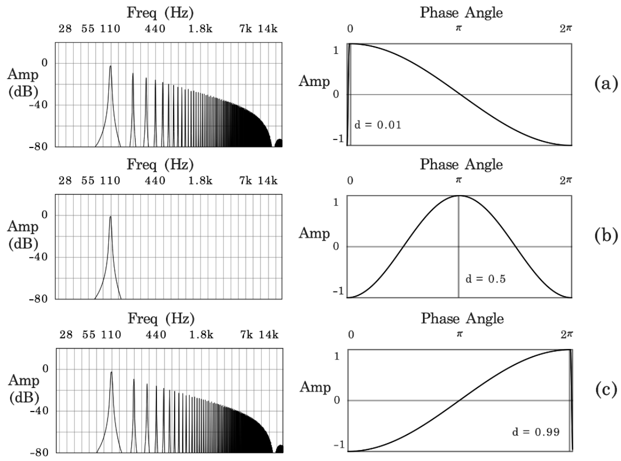

Figure 3: *Phase Distorted cosine wave using Equation (7) (f_s = 44100, $f_0 = 110$ (A2)).*
(a): $d = 0.01$, (b): $d = 0.5$, (c): $d = 0.99$.

This operates similarly to Pulsar Synthesis [11], as both methods divide a phase signal into two segments before applying a window function [9] to a waveform with a frequency independent of the fundamental. However, unlike traditional Pulsar Synthesis, where the second segment is typically silence, WPPM enables independent control over a waveform in both segments. By additively combining these wave pulses with the PD algorithm from Equation (7), the resulting phase signal can be used in the cosine lookup table from Equation (1) to achieve unique phase shaping. Fig. 4 illustrates each step in the creation of the WPPM phase signal. Equations (8) and (9) define the technique:

$$g_n(t) = w(t) \cdot I_n \sin\left(2\pi[R_n t + P_n]\right) \quad (8)$$

$$F_4(\phi[n]) = \begin{cases} \frac{\phi[n]}{2d} + g_1\left(\frac{\phi[n]}{d}\right), & \phi(n) < d \\ \frac{1}{2} + \frac{\phi[n]-d}{2(1-d)} + g_2\left(\frac{\phi[n]-d}{1-d}\right), & d \leq \phi(n) \end{cases} \quad (9)$$

where

I_n = the modulation index

R_n = the modulation ratio

P_n = the normalised phase offset, where $0 \leq P_n \leq 1$ corresponds to a phase angle between 0 and 2π radians

d = the duration ratio between the two segments

By incorporating n into Equation (8), unique values can be assigned to I, R and P each time $g_n(t)$ is applied in Equation (9), enabling independent control over each wave pulse.

3.2. Musical and Expressive Potential

In the WPPM algorithm, R_n is no longer a multiple of the fundamental frequency as R is in Equation (3). Instead, because d controls the slope, or frequency, of each wave segment, R_n becomes a multiple of the phase segment duration, making it directly proportional to d. For example:

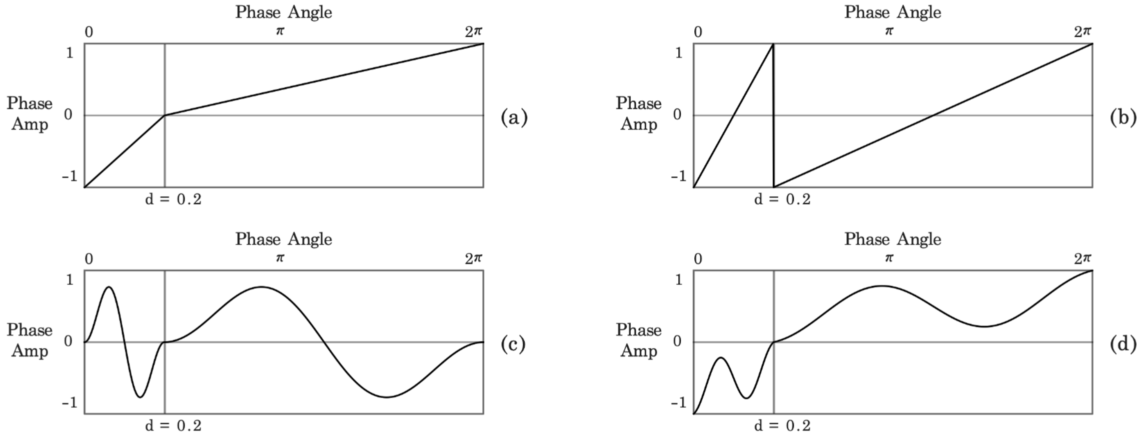

Figure 4: *Breakdown of the WPPM phase signal algorithm for* $d = 0.2$. *(a): PD phase signal from Equation (7). (b): PD phase signal with no scaling, meaning that each segment stays in the amplitude range [0, 1]. (c): Two related windowed sine waves created using the phase signal from (b) substituted into Equation (8) with* $R_n = 1$, $I_n = 1$ *and* $P_n = 0$. *(d): WPPM phase signal created by adding together (a) and (c) (Equation (9)) using* $I_n = 0.3$.

- If $d = 0.5$ and $R_1 = R_2 = 1$, both wave pulses complete one full cycle each within half the period of the phase signal, making their frequencies double the fundamental frequency.
- If $d = 0.25$ and $R_1 = R_2 = 1$, the first wave pulse has a frequency four times the fundamental frequency, while the second wave pulse has a frequency $4/3$ times the fundamental frequency.

This effect can be observed in Fig. 4.

Because of this, d becomes a highly expressive musical parameter when modulated. Just as modulating d in PD can mimic a lowpass filter opening on a sawtooth wave, WPPM can produce unique resonant filter sweeps when $I \neq 0$. Assigning distinct wave pulses for each segment allows d to morph between them, with $d = 1$ sounding only the first pulse, and $d = 0$ sounding only the second, effectively morphing between two presets.

Modulation becomes especially powerful when LFOs and envelopes are mapped to any of the WPPM parameters. Mapping an envelope to d can create the classic filter modulation effects of the Casio CZ series sythesisers [3, 4], while modulating WPPM's index and ratio parameters unlocks evolving resonant characteristics. Due to its substantial effect on the sonic output of WPPM, d functions as a highly performative parameter when assigned to MIDI controls such as velocity. Mapping d to the modulation wheel, for example, allows users to perform tonal gestures dynamically.

In terms of control complexity, WPPM occupies a middle ground between PD and PM. Traditional PD, like in the Casio CZ series sythesisers [3, 4], typically involves a minimal set of parameters, often only a single control over the shape of the phase signal, often mapped to an envelope. This simplicity, whilst being highly accessible, is somewhat limited in tonal complexity. PM synthesis, by contrast, offers much greater sonic complexity. Synthesisers such as Yamaha's DX-7 [2, 8] utilise multiple operators, each with their own envelopes, frequency ratios, modulation indexes, and more arranged in complex algorithms. While this architecture enables the creation of complex and evolving timbres, it also introduces a high degree of complexity that often leads users to rely more heavily on factory presets rather than custom sound design. WPPM sits between these two approaches. It expands on PD by offering independent ratio, index and phase controls for two wave pulses, as well as the duration ratio control d, totalling seven parameters. Though more complex than PD, WPPM is significantly more accessible than PM, offering a balance between control complexity and usability whilst affording a large range of tonal possibilities.

The number of parameters in WPPM can be reduced by linking the ratio, index and phase controls of one wave pulse to those of the other, effectively reducing the controllable parameters from seven to four. This configuration introduces a symmetry around $d = 0.5$, causing WPPM to behave more similarly to traditional PD synthesis. In the context of interface design, a toggle control could be implemented to link these parameters by default, streamlining the user experience, whilst still allowing them to be unlinked for finer control.

Fig. 5 illustrates three variations of the algorithm defined in Equations (8) and (9) used as a phase signal to drive Equation (1).

- In variation (a), the first segment of the waveform consists of a complex Phase Modulated wave, while the second segment is the second half of a cosine wave, resembling the PD saw wave in Fig. 3.
- Variation (b) shows a similar waveform, but with a longer duration for the first segment, achieved by setting $d > 0.5$.
- In variation (c), the first segment remains the same as in (b), while the second segment has its own separate Phase Modulation, introducing resonant formant frequencies [12, 13] above 2 kHz, as can be observed in the spectrogram.

When implemented in a musical context, WPPM responds particularly well to effects. Due to its resonant characteristics, it pairs effectively with distortion and can produce aggressive, acid-like basslines in a vein similar to the distorted resonant tones of the Roland TB-303 [14]. Although a filter is not essential for WPPM, it can be advantageous when higher ratio values are set, especially when modulating d, as the resulting spectral content can become increasingly dense and difficult to control, thus benefitting from additional taming. WPPM also responds effectively to time-based effects such as chorus and reverb, which can enhance its

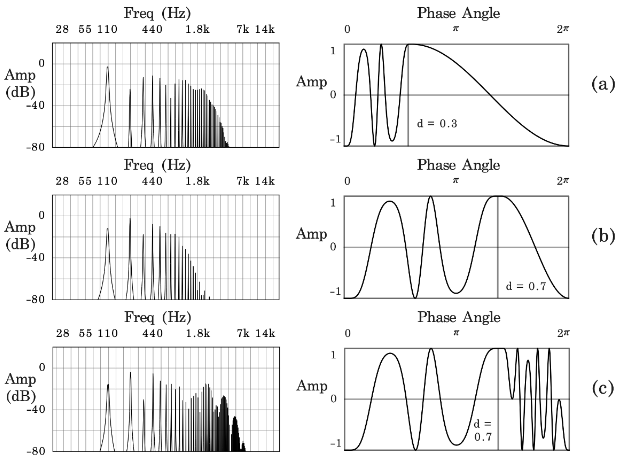

Figure 5: *WPPM waveform examples ($f_s = 44100$, $f_0 = 110$ (A2)).*
(a): $d = 0.3$, $R_1 = 1$, $I_1 = 0.75$, $P_1 = 0.5$, $R_2 = 1$, $I_2 = 0$, $P_2 = 0$.
(b): $d = 0.7$, $R_1 = 1$, $I_1 = 0.75$, $P_1 = 0.5$, $R_2 = 1$, $I_2 = 0$, $P_2 = 0$.
(c): $d = 0.7$, $R_1 = 1$, $I_1 = 0.75$, $P_1 = 0.5$, $R_2 = 2$, $I_2 = -0.75$, $P_2 = 0$.

already complex output into richly textured, evolving atmospheres that work especially well as pads.

4. ANTIALIASING

Due to the modulation frequencies of each segment being proportional to the duration ratio d, aliasing issues arise at extreme d values. Assuming $R_n \neq 0$ and $I_n \neq 0$, as d approaches 0, the frequency of the first wave pulse f_1 approaches infinity, whereas the frequency of the second wave pulse f_2 approaches the fundamental frequency f_0 multiplied by R_2:

$$\lim_{d \to 0} f_1 = \infty, \qquad \lim_{d \to 0} f_2 = f_0 R_2$$

Conversely, as d approaches 1, the frequency of the first wave pulse, f_1 approaches the fundamental frequency f_0 multiplied by R_1, and the frequency of the second wave pulse f_2 approaches infinity:

$$\lim_{d \to 1} f_1 = f_0 R_1, \qquad \lim_{d \to 1} f_2 = \infty$$

When d is close to 0 or 1, ultra-high frequency wave pulses are generated, leading to significant aliasing. To mitigate this, a window function driven by d can be applied to $g_n(t)$. For this purpose, the window function should be symmetric and have peak amplitude half way through its duration, making simple ramp windows unsuitable. Only the first half of the window is applied to $g_1(t)$, ensuring that when $d = 0$, the amplitude of $g_1(t)$ is 0, and when $d = 1$, the user-defined I_1 value is reached without attenuation. Conversely, only the second half of the window is applied to $g_2(t)$, ensuring that when $d = 1$, the amplitude of $g_2(t)$ is 0, and when $d = 0$, it reaches the user-defined I_2 value. By implementing this windowing method, ultra-high frequency wave pulses are attenuated as their frequencies increase, resulting in significantly less aliasing.

Equation (10) demonstrates how a sinusoidal window can be incorporated into the WPPM algorithm to attenuate the amplitudes of $g_1(t)$ and $g_2(t)$ as their respective frequencies approach infinity. Applying a full sine window to both wave pulses would be problematic as it would cause both $g_n(t)$ components to have zero amplitude at either extreme d value, and only reach the specified index values at $d = 0.5$.

$$F_5(\phi[n]) = \begin{cases} \frac{\phi[n]}{2d} + w\left(\frac{d}{2}\right) g_1\left(\frac{\phi[n]}{d}\right), & \phi(n) < d \\ \frac{1}{2} + \frac{\phi[n]-d}{2(1-d)} + w\left(1 - \frac{d}{2}\right) g_2\left(\frac{\phi[n]-d}{1-d}\right), & \\ & d \leq \phi(n) \end{cases} \quad (10)$$

A limitation of the sinusoidal window is that the user-defined indexes are only reached without attenuation when $d = 1$ for $g_1(t)$ and when $d = 0$ for $g_2(t)$. To address this, an alternative symmetrical window function, such as a Tukey window [9], is more effective as it quickly reaches the maximum gain and plateaus, allowing the target index values to be achieved more efficiently.

Further aliasing can arise from abrupt phase discontinuities present at the extreme d values of 0 and 1. This can be mitigated by smoothing the transition between the end of one phase cycle and the beginning of the next. One simple approach is to limit d to a smaller range, such as $0.01 \leq d \leq 0.99$, ensuring that the phase signal has time to smoothly return to an optimal looping point (as shown in Fig. 3).
Another approach is to apply a polynomial bandlimited step function (polyBLEP) at extreme d values [1]. PolyBLEP smooths the phase transition by modifying the two samples before and after the 1-to-0 transition using a second-order correction polynomial. Although a complete implementation of polyBLEP is beyond the scope of this paper, it provides a sufficient solution for suppressing aliasing. More details on polyBLEP can be found in [15].

5. FURTHER DEVELOPMENTS

The WPPM algorithm offers a wide range of possibilities for further research and modifications to expand its sonic capabilities. The following are several potential areas for development.

- A simple variation could involve applying phase modulation to only one of the wave segments. This approach would more closely resemble Curtis Roads' equation for Pulsar Synthesis, where $p = d + s$, with p as the pulsar period, d as the duty cycle and s as the silent segment [16]. This can be easily applied by simply setting one of the index values to 0.
- WPPM is based on the first algorithm of the Casio CZ series synthesizers [4]; however, there is potential to explore how the remaining algorithms could be integrated into the PD portion of the WPPM algorithm.
- Alternative waveforms could be used for the modulator or carrier [17] through buffer-based lookups. Implementing this would require adjustments to the phase signal algorithm, incorporating a signal wrapping algorithm to ensure its amplitude does not exceed the $[0, 1]$ range for proper lookup table indexing.
- Continuously sequencing different waveforms and window functions [16] has the potential to produce more dynamic

evolving soundscapes. Rather than limiting the modulation waves to windowed sinusoidal waveforms, short grains from an audio sample could be used as modulation pulses, similar to granular synthesis.

- Like in [11], astronomical data from celestial pulsars could be used as wave pulses, potentially leading to unique and unpredictable sonic textures.
- Additional control over the PD breakpoint could be introduced by modifying its position on the Y-axis by integrating Vector Phase Synthesis [18] into the WPPM algorithm.
- Increasing the number of segments within the phase signal [18] would allow for more wave pulses per carrier period, but may also suffer from becoming increasingly complex to control and interface due to an increased number of parameters.

6. CONCLUSIONS

This paper introduced Wave Pulse Phase Modulation (WPPM), a novel synthesis technique that merges elements of Phase Distortion (PD) and Phase Modulation (PM) to address their respective limitations while enabling the creation of unique timbres and offering an expressive set of controls. It explored both components in detail, outlining their individual algorithms and the modifications required to integrate them into WPPM. The proposed algorithm was presented with a breakdown of its creative possibilities, as well as methods for aliasing suppression, and suggestions for possible further research and modifications.
The WPPM synthesis prototype described in this paper has been implemented in Max/MSP using GenExpr. The code is open source and available at
https://github.com/MattS6464/WPPM-Oscillator.git.
A selection of audio examples showcasing a range of musical and timbral possibilities, as well as examples corresponding to each relevant figure, are available at:
https://soundcloud.com/wppm/tracks.

7. ACKNOWLEDGMENTS

This research was supported by the University of Huddersfield. The author also thanks Alexander Harker for his valuable insights and support throughout this project.

8. REFERENCES

[1] J. Kleimola, V. Lazzarini, J. Timoney, and V. Välimäki, “Phaseshaping oscillator algorithms for musical sound synthesis,” in *7th Sound and Music Computing Conference*, July 2010.

[2] J. Chowning, “Method of synthesising a sound,” United States Patent 4018121, 1977.

[3] M. Ishibashi, “Electronic musical intrument,” United States Patent 4658691, 1985.

[4] Casio, *Casio CZ-1000 Operation Manual*, 1984.

[5] J. M. Chowning, “The synthesis of complex audio spectra by means of frequency modulation,” *Computer Music Journal*, vol. 1, no. 2, pp. 46–54, 1977.

[6] V. Lazzarini, J. Timoney, J. Pekonen, and V. Välimäki, “Adaptive phase distortion synthesis,” in *DAFx 09 proceedings of the 12th International Conference on Digital Audio Effects, Politecnico di Milano, Como Campus, Sept. 1-4, Como, Italy*. Dept. of Electronic Engineering, Queen Mary Univ. of London, 2009, pp. 1–8.

[7] D. Bristow J. Chowning, *FM theory and applications: By musicians for musicians*, Hal Leonard Corp, 1987.

[8] Yamaha, *DX7 Digital Programmable Algorithm Synthesizer Owner’s Manual*, 1983.

[9] K. M. M. Prabhu, *Window Functions and Their Applications in Signal Processing*, chapter Discrete-Time Windows and Their Figures of Merit, pp. 145–188, Taylor & Francis, 2014.

[10] J. Timoney, V. Lazzarini, B. Carty, and J. Pekonen, “Phase and amplitude distortion methods for digital synthesis of classic analog waveforms,” in *Audio Engineering Society Convention 126*. Audio Engineering Society, 2009.

[11] C. Roads, *Microsound*, chapter Varieties of Particle Synthesis, pp. 178–199, The MIT Press, 2004.

[12] C. Roads, *The computer music tutorial*, chapter Physical Modelling and Formant Synthesis, pp. 263–316, MIT press, 1996.

[13] M. Puckette, “Formant-based audio synthesis using nonlinear distortion,” *Journal of the Audio Engineering Society*, vol. 43, no. 1/2, pp. 40–47, 1995.

[14] Roland, *Roland TB-303 Owner’s Manual*, 1982.

[15] V. Välimäki and A. Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 116–125, 2007.

[16] H. Mara, “Letting pulsars sing: Sonification with granular synthesis,” *Journal of the Audio Engineering Society*, vol. 72, pp. 352–359, July 2024.

[17] M. Le Brun, “Digital Waveshaping Synthesis,” *Journal of the Audio Engineering Society*, vol. 27, pp. 250–266, April 1979.

[18] J. Kleimola, V. Lazzarini, J. Timoney, and V. Välimäki, “Vector phase shaping synthesis,” in *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11)*, September 2011.