COMPUTATION OF THE ONE-DIMENSIONAL UNWRAPPED PHASE

Zahi N. Karam and Alan V. Oppenheim

Massachusetts Institute of Technology
Digital Signal Processing Group
77 Massachusetts Avenue, Cambridge, MA 02139
zahi@mit.edu, avo@mit.edu

ABSTRACT

In this paper, the computation of the unwrapped phase of the discrete-time Fourier transform (DTFT) of a one-dimensional finite-length signal is explored. The phase of the DTFT is not unique, and may contain integer multiple of 2π discontinuities. The unwrapped phase is the instance of the phase function chosen to ensure continuity. This paper compares existing algorithms for computing the unwrapped phase. Then, two composite algorithms are proposed that build upon the existing ones. The core of the proposed methods is based on recent advances in polynomial factoring. The proposed methods are implemented and compared to the existing ones.

***Index Terms*—** Phase, Phase Unwrapping, Unwrapped Phase, Cepstrum

1. INTRODUCTION

Homomorphic signal processing with the complex cepstrum [1] has been applied, with considerable success, to many areas of digital signal processing [3], most notably speech, seismic and electroencephalogram (EEG) data processing. The computation of the complex cepstrum requires the unwrapped phase, which is the continuous and periodic instance of the phase of the discrete-time Fourier transform (DTFT) of a given signal.

Existing algorithms for computing samples of the unwrapped phase of a signal are often not sufficiently reliable and in many cases fail. This paper presents two composite algorithms that build upon the existing ones. The reliability and accuracy of the proposed algorithms have been demonstrated through numerous experiments. The scope of this paper is restricted to finite length sequences, and we assume that the z-transform of the signal has no zeros on the unit circle.

2. PHASE OF THE DTFT

The phase of the DTFT ($X(e^{j\omega})$) of a discrete-time signal $x[n]$ is in general ambiguous since at any frequency an integer multiple of 2π can be added without affecting the result of the complex exponentiation. The principal value ARG$\{X(e^{j\omega})\}$ is the unique instance of the phase function where the 2π ambiguity is resolved by restricting the value of ARG$\{X(e^{j\omega})\}$ at any ω to the range $[-\pi, \pi)$. The principal value can be calculated by computing the arctangent of the DTFT.

The unwrapped phase, $\arg\{X(e^{j\omega})\}$, is the phase function for which the additive integer multiples of 2π at each frequency $\omega \in [0, 2\pi)$ are chosen to ensure that it is continuous. The unwrapped phase is directly affected by the location of the poles and zeros of the z-transform ($X(z)$) of the signal. Most importantly, the closer a pole or zero of $X(z)$ is to the unit circle in the z-plane, the sharper the change in the unwrapped phase at frequencies in its vicinity.

3. EXISTING ALGORITHMS

This section describes several existing algorithms to compute samples of the unwrapped phase $\arg\{X[k]\}$ and highlights their limitations.

The unwrapped phase can be obtained from the principal value of the phase by detecting and removing the discontinuities introduced by the arctangent routine [2]. The limitation of this method is that it assumes that the difference between two consecutive samples of the unwrapped phase is at most a set constant, usually π. This assumption fails when the phase varies rapidly. One possible way to find out if all the discontinuities have been detected is to use this method with progressively shorter frequency sampling intervals until two consecutive computed unwrapped phase functions match. However, this test is not reliable and does not guarantee that an even shorter frequency sampling interval will not yield a different unwrapped phase. Phase unwrapping by detecting and removing discontinuities using progressively shorter frequency sampling intervals will be referred to as the DD method.

Numerical integration can be used to compute samples of the unwrapped phase from samples of the derivative of the unwrapped phase ($\arg'\{X[k]\}$). The derivative can be computed by taking the imaginary part of the ratio of the derivative of the DTFT and the DTFT [1], we will refer to this as the ratio method. Unfortunately, it is not possible to determine *a priori* the integration step size needed to yield correct results. Tribolet [4] suggests an adaptive numerical integration (ANI) method that adaptively decreases the step size until the computed samples of the unwrapped phase become consistent with those of the wrapped phase. The check for consistency is as follows:

$$|\arg\{X[k]\} - \text{ARG}\{X[k]\} + 2\pi L[k]| < \epsilon, \tag{1}$$

where $L[k]$ is an integer function of k and ϵ is a set threshold. This method becomes more computationally intensive and may fail at frequencies where the phase function varies sharply due to zeros of $X(z)$ close to the unit circle.

Steiglitz and Dickinson [5] suggested computing the unwrapped phase by adding together the phase contribution of each zero. The success of this method is contingent on the success of the polynomial factoring algorithm used to find the location of the zeros of

This work was supported in part by: the Texas Instruments Leadership University Program, BAE Systems PO 112991, and MIT Lincoln Laboratory PO 3077828.

1-4244-0882-2/07/$20.00 ©2007 IEEE

$X(z)$. Sitton et al [6] propose an efficient method for factoring high degree polynomials. This algorithm focuses on searching for zeros in the area close to the unit circle, and thus works best on polynomials with zeros clustered closer to the unit circle. However, this method becomes less efficient and may fail for polynomials with zeros located farther away from the unit circle. Computing the unwrapped phase by first finding the zeros using the polynomial factoring method proposed by Sitton et al and then adding together the phase contributions of the zeros will be referred to as the PF method.

There are several other methods for computing the unwrapped phase, each with its limitations. McGowan [11] presents an algorithm that computes the unwrapped phase directly from the discrete-time sequence $x[n]$. This method is limited to short sequences due to its sensitivity to computational errors. Quatieri and Oppenheim [12] presented an algorithm that uses an iterative method to compute the corresponding minimum-phase sequence of $x[n]$. The unwrapped phase of the minimum-phase sequence is then used to compute that of the input sequence. This method has two limitations. The first is that the minimum-phase sequence is in general infinite in length even if $x[n]$ is finite length which limits the accuracy of the iterative computation of the minimum-phase sequence. The second is that it assumes that the number of zeros outside the unit circle is known *a priori* which is not usually the case. Al-Nashi [13] proposed a method that computes the unwrapped phase by only using zeros that are close to the unit circle; this method performs poorly on signals with zeros clustered close to each other near the unit circle.

4. PROPOSED COMPOSITE ALGORITHMS

This section presents two new methods for phase unwrapping that are capable of reliably computing the unwrapped phase in situations where the existing algorithms may fail. These methods are a composite of polynomial factoring and either DD or ANI.

4.1. Motivation

Phase unwrapping using the DD and ANI methods are the most frequently used in practice. These methods become inefficient and may fail as the zeros of the z-transform of the signal approach the unit circle. This limitation is of concern because, as presented in [14], sampled natural physical signals tend to have their zeros clustered in a tight annulus around the unit circle. Moreover, the area of the annulus decreases with increasing signal size. On the other hand, computation of the unwrapped phase using the PF method, performs best when all the zeros are close to the unit circle. We conclude that the DD and ANI methods tend to be complementary to the PF method: DD and ANI perform poorly when PF performs well and vice versa. This motivates two composite algorithms: the first (CA1) combines polynomial factoring and DD and the second (CA2) combines polynomial factoring with ANI.

4.2. Overview

Any signal $x[n]$ can be decomposed as

$$x[n] = x_{UC}[n] * x_{rem}[n],$$
$$X(z) = X_{UC}(z) X_{rem}(z),$$

where $X_{UC}(z)$ contains the zeros (z_{UC}) that are closer to the unit circle, and $X_{rem}(z)$ contains the remaining zeros of $X(z)$.

The proposed algorithms consist of five steps:

1. Use polynomial factoring to find the zeros z_{UC} that are clustered near the unit circle, i.e. zeros of $X_{UC}(z)$.
2. Calculate the unwrapped phase contribution ($\arg\{X_{UC}[k]\}$) of these zeros.
3. Obtain $x_{rem}[n]$ by deflating $x[n]$ using $x_{UC}[n]$. Deflation removes the zeros z_{UC} from $x[n]$ to produce the polynomial $x_{rem}[n]$ which has lower degree.
4. Use either the DD or ANI algorithm to unwrap the phase contribution of the remaining zeros to obtain $\arg\{X_{rem}[k]\}$.
5. Add the unwrapped phase calculated in Step 2 and in Step 4 to obtain the total unwrapped phase.

$$\arg\{X[k]\} = \arg\{X_{UC}[k]\} + \arg\{X_{rem}[k]\} \qquad (2)$$

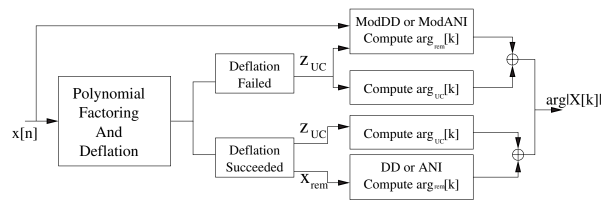

Fig. 1. Overview block diagram.

4.3. Polynomial Factoring (PFact) Subroutine

The polynomial factoring and deflation algorithms used in the proposed composite methods are taken from Sitton et al [6]. MATLAB code downloaded from the website [9] of the authors of [6] is modified and used to implement this subroutine.

PFact Subroutine

1. Create a search grid that is dense around the unit circle.
2. Find local minima, and use them as candidate locations for roots.
3. Improve the accuracy (polish) of the the candidate locations using Laguerre's algorithm [10] to obtain the correct location.
4. Remove any duplicates (zeros that polished to the same location) from the set of computed zeros.
5. Deflate the input polynomial using the computed roots.
6. i) If deflation succeeds, return the computed zeros and the deflated polynomial, $x_{rem}[n]$.
   ii) If deflation fails, and this is the first iteration through this subroutine, repeat all the steps while using a denser search grid in Step 1.
   iii) If deflation fails, and this is the second iteration, return the computed zeros.

Empirical tests suggest that a maximum of two iterations should be allowed to attempt to obtain a deflated polynomial. Additional iterations increase computation time and in most cases do not improve the result. It is important to note that the PFact subroutine finds most of the zeros that are close to the unit circle, but usually misses a few: this is mainly because the signal may have multiple zeros that are very close to each other. A block diagram outlining these steps is shown in Figure 2.

The search grid used in the subroutine is created by sampling the z-plane on concentric circles of varying radii. This sampling is performed by applying exponential weighting to the input sequence and then taking the discrete Fourier transform (DFT) of the weighted

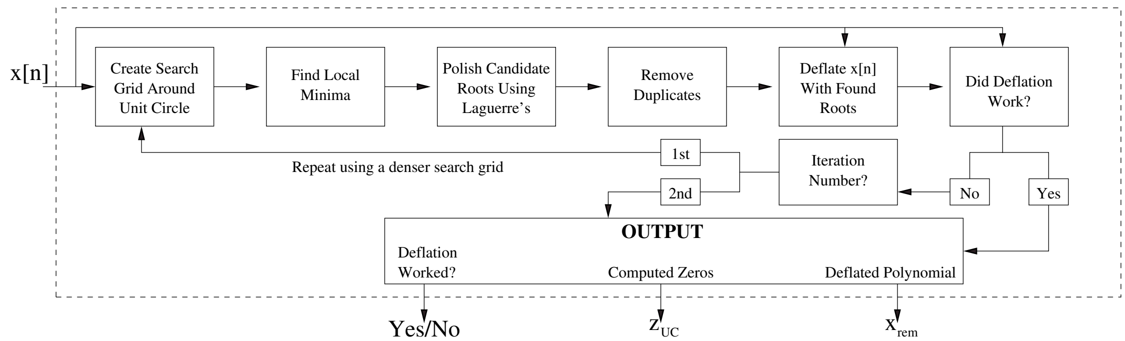

Fig. 2. Polynomial factoring and deflation block diagram.

sequence. The grid is chosen so that it is denser near the unit circle: as the unit circle is approached the concentric circles are more closely spaced and the number of samples (DFT size) on each circle increases.

The deflation step attempts to obtain a polynomial $x_{rem}[n]$ of lower degree by removing the contributions of the computed zeros from the input polynomial. The zeros that have determined are unfactored into a polynomial $x_{fnd}[n]$ which is then used to deflate $x[n]$. To check if deflation succeeded, a reconstructed polynomial $x_{rec}[n]$ is formed as

$$x_{rec}[n] = x_{rem}[n] * x_{fnd}[n],$$

and compared to the input polynomial $x[n]$.

4.4. Modified DD and ANI Subroutines (ModDD & ModANI)

When deflation fails, $x_{rem}[n]$ cannot be obtained. Thus, the DD and ANI algorithms cannot be directly applied. In this section we modify these algorithms to compute $\arg\{X_{rem}[k]\}$, without access to $x_{rem}[n]$, using only the input signal $x[n]$ and the computed roots z_{UC}.

The ModDD subroutine differs from the DD phase unwrapping algorithm in that it computes samples of $\text{ARG}\{X_{rem}[k]\}$ without using $x_{rem}[n]$. The following steps outline the subroutine:

1. Compute $\text{ARG}\{X[k]\}$, i.e. samples of the wrapped phase of $x[n]$, using the arctangent routine.
2. Compute $\arg\{X_{UC}[k]\}$ from the computed roots z_{UC} by summing the individual phase contributions of each zero.
3. Subtract modulo 2π the unwrapped phase of the computed roots from the wrapped phase of the input $x[n]$,

$$\arg\{X_{rem}[k]\} = \text{mod}\{\text{ARG}\{X[k]\} - \arg\{X_{UC}[k]\}\}_{2\pi}.$$

The ModANI subroutine differs from the ANI phase unwrapping algorithm in that it computes samples of the wrapped phase $\text{ARG}\{X_{rem}[k]\}$ and samples of the phase derivative $\arg'\{X_{rem}[k]\}$ without using $x_{rem}[n]$. ModANI uses the same steps as ModDD to compute $\text{ARG}\{X_{rem}[k]\}$, while $\arg'\{X_{rem}[k]\}$ can be computed in the following steps:

1. Compute $\arg'\{X[k]\}$ using the ratio method.
2. Compute $\arg'\{X_{UC}[k]\}$ from the computed roots z_{UC} by summing the individual phase derivative contributions of each zero.
3. Subtract the phase derivative of the computed roots from the phase derivative of the input $x[n]$,

$$\arg'\{X_{rem}[k]\} = \arg'\{X[k]\} - \arg'\{X_{UC}[k]\}.$$

4.5. Robustness To Errors In Factored Zeros

The PFact subroutine also computes an estimate of the error of each polished root. The error estimate, as proposed in the code obtained from [9], is based on the Newton correction $-f(z)/f'(z)$ evaluated at the location of the polished zero [7]. Errors incurred by the polynomial factoring section of the algorithm may cause the deflation to fail. However, in general, they will not translate into errors in the unwrapped phase. This is because an error in a zero location due to the first section of the algorithm will be absorbed into the second section and will not affect the final computed unwrapped phase [14]. It is important to note, however, that if the errors are large they will interfere with the ModDD and ModANI subroutines. Therefore, zeros that have large error estimates are removed. From empirical results, it is best to remove zeros whose estimated errors were larger than 10^{-6}.

5. ALGORITHM EVALUATION

In this section DD, ANI, PF are compared against the proposed composite algorithms CA1 and CA2. The criteria for comparison are run time and accuracy. Run times are measured from MATLAB implementations on a PC with a 1.6GHz Pentium M processor and 512 MB of RAM. In this paper we present results for real-valued input signals; however, with minor modifications the algorithms should perform similarly for complex-valued signals.

Two types of tests are used in the comparison. The first evaluates the algorithms on a large number of synthetic signals for which the zero locations are known: knowing the zero locations allows us to compute the exact unwrapped phase function for comparison. This test is used to demonstrate the reliability of the proposed algorithms. The second evaluates the algorithms on sampled speech and EEG data, which are examples of natural signals that are of practical interest.

5.1. Synthetic Signals

The algorithms are evaluated using two thousand synthetic test signals for which the zero locations are randomly chosen to span a large class of possible input signals [14]. Table 1 shows the average run time of each algorithm and the percentage of times the correct unwrapped phase was computed. These results clearly demonstrate the reliability of the composite algorithms and their superiority to the existing ones. An important observation is that CA1 is about three times faster than CA2; however, CA2 only failed once on this test set while CA1 failed five times.

Method	Percentage Correct	Average Run Time
DD	84.8%	1sec
ANI	84.8%	73sec
PF	48.5%	23sec
CA1	99.75%	8sec
CA2	99.95%	24sec

Table 1. Results from 2000 synthetic signals.

5.2. Speech and EEG Signals

This section evaluates the algorithm performance on natural signals which one might encounter in practical applications: speech, EEG, and filtered versions of these signals. The speech signal is a recording sampled at 8kHz of the utterance "unwrapped phase". The EEG data is recorded from a subject performing a multiplication task collected at Colorado State University's Computer Science Department (downloaded from [8]); the signal is EEG electrode data sampled at 250Hz for 10 seconds.

The zeros of both the speech and EEG signals are clustered close the unit circle. The locations of the zeros were calculated by the polynomial factoring algorithm presented in [6]. We therefore expect that the PF and the composite algorithms will perform well, while DD and ANI will perform poorly. Table 2 confirms these expectations, with both DD and ANI failing to provide the correct unwrapped phase.

	Speech		EEG	
Method	Correct	Run Time	Correct	Run Time
DD	No	1.87sec	No	0.49sec
ANI	No	FAIL	No	63.2
PF	Yes	26.2sec	Yes	4.7sec
CA1	Yes	17.26sec	Yes	5sec
CA2	Yes	26.23sec	Yes	5sec

Table 2. Results for speech and EEG data.

The speech and EEG signals are filtered with a 220^{th} order lowpass Parks-McClellan filter with a transition band between 0.2π and 0.3π. The filtered signals are truncated to remove leading and trailing transients such that the length of the resulting sequence is that of the original. The algorithms were evaluated using the filtered signals and the results are presented in Table 3. These two examples are of particular interest because polynomial factoring fails and, therefore, the roots of the signal are unknown. Consequently, it is not possible to check that the computed unwrapped phase is correct. In such situations we assume the unwrapped phase from CA2 to be the correct one, because as is seen in Table 1 it is the most reliable. Consequently the results presented in Table 3 are based on this assumption.

	Speech		EEG	
Method	Correct	Run Time	Correct	Run Time
DD	Yes	0.4sec	Yes	0.1sec
ANI	Yes	116.3sec	Yes	11.2sec
PF	No	FAIL	No	FAIL
CA1	Yes	31.3sec	Yes	6.5sec
CA2	Yes	121.1sec	Yes	43sec

Table 3. Results for filtered speech and EEG data.

6. CONCLUSION

Correctly and reliably computing the unwrapped phase of any one-dimensional finite-length discrete-time signal, regardless of the locations of its z-transform zeros, is an unsolved problem. Existing algorithms are unreliable and only perform well on specific classes of signals. We have proposed two composite algorithms to reliably compute the unwrapped phase of the DTFT that are robust to the locations of the z-transform zeros. The composite algorithms outperform the existing methods by combining their strength while avoiding their limitations. The reliability and efficiency of our proposed algorithms were shown by evaluating them on a large number of synthetic signals.

7. REFERENCES

[1] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*, chapter 12. Prentice Hall, Englewood Cliffs, NJ, 1989.

[2] A. V. Oppenheim, R. W. Schafer and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1998.

[3] A. V. Oppenheim and R.W. Schafer. From frequency to quefrency: A history of the cepstrum. *IEEE Signal Processing Magazine*, 2004.

[4] J. M. Tribolet. A new phase unwrapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(2), 1977.

[5] K. Steiglitz and B. Dickinson. Phase unwrapping by factorization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(6), 1982.

[6] G. A. Sitton, C. S. Burrus, J. W. Fox, and S. Treitel. Factoring very-high-degree polynomials. *IEEE Signal Processing Magazine*, 2003.

[7] Eric W. Weisstein. Newton's method. From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/NewtonsMethod.html.

[8] Colorado State University Computer Science Department. EEG data. http://www.cs.colostate.edu/eeg/?Summary#Data.

[9] Rice University Digital Signal Processing Group. Factoring very high degree polynomials Matlab code. http://dsp.rice.edu/software/fvhdp.shtml

[10] K. G. Orchard. The Laguerre method for finding the zeros of polynomials. *IEEE Transactions on Circuits and Systems*, 36(11), 1989.

[11] R. McGowan and R. Kuc. A direct relation between a signal time series and its unwrapped phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(5), 1982.

[12] T. F. Quatieri and A. V. Oppenheim. Iterative techniques for minimum phase signal reconstruction from phase or magnitude. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6), 1981.

[13] H. Al-Nashi. Phase unwrapping of Digital Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11), 1989.

[14] Z. N. Karam. Computation of the One-Dimensional Unwrapped Phase. Master's Thesis, Massachusetts Institute of Technology, 2006.