Computation of the One-Dimensional Unwrapped Phase

by

Zahi Nadim Karam

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2006
© Massachusetts Institute of Technology 2006. All rights reserved.

Author ............................................................

Department of Electrical Engineering and Computer Science

January 24, 2006

Certified by ..............................................

Alan V. Oppenheim
Ford Professor of Engineering

Department of Electrical Engineering and Computer Science

Thesis Supervisor

Accepted by ..............................................

Arthur C. Smith

Chairman, Department Committee on Graduate Students

Department of Electrical Engineering and Computer Science
Computation of the One-Dimensional Unwrapped Phase

by

Zahi Nadim Karam

Submitted to the Department of Electrical Engineering and Computer Science on January 24, 2006, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Abstract

In this thesis, 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 $2\pi$ discontinuities. The unwrapped phase is the instance of the phase function chosen to ensure continuity. This thesis presents existing algorithms for computing the unwrapped phase, discussing their weaknesses and strengths. Then two composite algorithms are proposed that use the existing ones, combining their strengths while avoiding their weaknesses. 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.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering
Department of Electrical Engineering and Computer Science
Acknowledgments

I would like to thank my family for all their love, commitment and support. Pop, you’ve always been my role model. Mom, your tenderness and unconditional love always made me feel warm and safe. Amme, for your wisdom, moral guidance and youthfulness. Hanan, your compassion and selflessness have always inspired me. Samer, for your determination, sense of adventure and passion for life. Amme Najwa and Amme Salwa for always being there and more. Teta and Jiddo, for your love and confidence.

To my advisor Al Oppenheim. You’ve been there from day one, making sure that my transition into MIT was smooth, comfortable and successful. Thank you for your wisdom, care and guidance. Your passion for teaching, your out of the box thinking and your sense of adventure have inspired me.

Alaa Kharbouch (3alaa2). We’ve watched each other’s back since freshman year. We are complete opposites, and yet best of friends. Thank you for sharing the pain and the experience.

Amer Abufadel, you’re like a big brother to me. You got me started, and have always watched out for me.

I would like to thank the DSPG: Joonsung for tolerating me and for being a great office mate. Petros for always having the answer. Sourav for the playful atmosphere and the long discussions. Tom for your musical ear. Maya for your help. Eric for your willingness to always help. Ross and Matt, Alecia, Charlie, Melanie, Dennis, Archana, Joe and Andrew, for making the office a pleasant place to hang out.

Prof. Vivek Goyal, thank you for a wonderful and stimulating experience working on the class with you. Prof. George Verghese thank you for looking out for me.

To all my friends, thank you for adding flavors and purpose to my life. Alieh, thanks for your love, care and advice. The Atrashes, thanks for being there for me. Victor, we’ve shared a lot of experiences together, thank you for being a great friend. Akram, thank you for the encouragement and support. Yousef, Woody, Pierre, Loucy, and George thanks for all the crazy times we’ve had, my breaks in Lebanon would not
have been the same without you. Wafaa2, Cathy, Pranav, Demba, Tamer, Viktorija, Lololna, Chrisann, Sandra, Lara, Zaid, Lolita and Mauri thanks for your friendship and for making my life in the US not just about study and work.
Contents

1 Introduction and Background ........................................ 15
   1.1 Problem Statement ............................................. 16
   1.2 Instances of the Phase and the Derivative of the Unwrapped Phase . 17
      1.2.1 Principal Value of the Phase ............................... 18
      1.2.2 Derivative of the Unwrapped Phase ....................... 19
      1.2.3 Unwrapped Phase ........................................... 19
   1.3 Example Phase Functions ......................................... 21

2 Existing Algorithms to Compute the Unwrapped Phase ............. 23
   2.1 Phase Unwrapping by Detecting Discontinuities ................... 23
   2.2 Phase Unwrapping Using Adaptive Numerical Integration .......... 26
   2.3 Computation of the Unwrapped Phase Using Polynomial Factoring . 28
   2.4 Phase Unwrapping from the Time Series ........................... 29
   2.5 Iterative Method to Compute the Unwrapped Phase ................ 30
   2.6 Phase Unwrapping by Isolating Sharp Zeros ....................... 31
   2.7 Summary .................................................................. 32

3 Method for Factoring High Degree Polynomials .................. 33
   3.1 Outline of the Algorithm ......................................... 33
   3.2 Deflation Failure .................................................. 34
   3.3 Discussion .......................................................... 35

4 Proposed Composite Methods ..................................... 37
4.1 Motivation ........................................... 37
4.2 Overview ........................................ 38
4.3 Polynomial Factoring and Deflation ................. 39
   4.3.1 Steps to Find Zeros Close to the Unit Circle .... 39
   4.3.2 Search Grid ................................ 40
   4.3.3 Deflation .................................. 42
4.4 Modified DD and ANI Algorithms .................. 43
   4.4.1 Modified DD Algorithm ..................... 43
   4.4.2 Modified ANI Algorithm .................... 44
4.5 Robustness To Errors In Factored Zeros .............. 44
4.6 Summary ........................................ 46

5 Algorithm Evaluation ........................................ 47
   5.1 Algorithm Evaluation Using Synthetic Signals .......... 48
      5.1.1 Synthetic Signal with Zeros Close to the Unit Circle .... 49
      5.1.2 Synthetic Signal with Zeros Far from the Unit Circle ... 50
      5.1.3 Synthetic Signal with Zeros Close to and Far from the Unit Circle 51
      5.1.4 Large Number of Synthetic Signals with Randomly Chosen Zeros 52
      5.1.5 Discussion .................................. 53
   5.2 Algorithm Evaluation Using Speech Data ............... 54
      5.2.1 Speech Data .................................. 54
      5.2.2 Filtered Speech Data ........................ 55
   5.3 Algorithm Evaluation Using EEG Data ................. 57
      5.3.1 EEG Data .................................. 57
      5.3.2 Filtered EEG Data .......................... 58
   5.4 Summary ........................................ 59

A Zeros of Random Polynomials ......................... 61
   A.1 Zeros of I.I.D. Random Polynomials .................. 61
   A.2 Zeros of Bounded Coefficient Random Polynomials .... 62
B Cepstrum Bibliography

B.1 Signal ......................................................... 67
B.2 Computation ................................................. 69
B.3 Speech/Audio ............................................... 70
B.4 Filter Design ............................................... 88
B.5 2D/Image .................................................... 88
B.6 Deconvolution .............................................. 90
B.7 Geophysics .................................................. 92
B.8 Applications ............................................... 92
B.9 Other ......................................................... 97
List of Figures

1-1 Example Wrapped Phase Function ........................................ 18
1-2 Example unwrapped phase .............................................. 20
1-3 Unwrapped phase and phase derivative ............................... 21
1-4 Unwrapped phase, principal value of the phase, and phase derivative. 22
2-1 Unwrapped phase using detection of discontinuities ............... 25
2-2 Unwrapped phase using adaptive numerical integration .......... 28
2-3 Sample Search Grid .................................................. 29
3-1 Z-plane plots demonstrating deflation failure ....................... 35
4-1 Block diagram of proposed algorithms ................................. 39
4-2 Block diagram for the polynomial factoring part of the algorithm .. 40
4-3 Sample search grid inside the unit circle ............................ 41
5-1 Results for the synthetic signal with zeros close to the unit circle. 49
5-2 Results for the synthetic signal with zeros far from the unit circle. 51
5-3 Results for the synthetic signal with zeros close to and far from the unit circle. ................................. 52
5-4 Speech segment of the utterance “Unwrapped Phase” ............... 54
5-5 Results for speech data ................................................ 55
5-6 Results for filtered speech data ....................................... 56
5-7 EEG data while performing a multiplication task .................... 57
5-8 Results for EEG data .................................................. 58
5-9 Results for filtered EEG data ......................................... 59
A-1 Observed clustering near the unit circle of zeros of polynomials with
i.i.d. uniformly distributed random coefficients. .......... 63

A-2 Observed clustering near the unit circle of zeros of the z-transform of
speech signals. ................................................. 65
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Results for the synthetic signal with zeros close to the unit circle.</td>
<td>50</td>
</tr>
<tr>
<td>5.2</td>
<td>Results for the synthetic signal with zeros far from the unit circle.</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>Results for the synthetic signal with zeros close to and far from the</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>unit circle.</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>Results from 2000 synthetic signals.</td>
<td>53</td>
</tr>
<tr>
<td>5.5</td>
<td>Results for speech data.</td>
<td>54</td>
</tr>
<tr>
<td>5.6</td>
<td>Results for filtered speech data.</td>
<td>56</td>
</tr>
<tr>
<td>5.7</td>
<td>Results for EEG data.</td>
<td>57</td>
</tr>
<tr>
<td>5.8</td>
<td>Results for filtered EEG data.</td>
<td>58</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction and Background

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 EEG data processing. The computation of the complex cepstrum requires obtaining the unwrapped phase, which is the continuous and periodic instance of the phase of the discrete-time Fourier transform (DTFT) of the input signal. Refer to Appendix B for a bibliography we have compiled on the complex cepstrum.

The phase of the DTFT, however, is in general ambiguous since at any frequency $2\pi k$ ($k$ is an integer) can be added without affecting the result of the complex exponentiation [1]. This ambiguity, therefore, allows the phase of the DTFT to contain discontinuities in the form of $2\pi k$ jumps. The unwrapped phase is the instance of the phase function where the additive integer multiples of $2\pi$ at each frequency are chosen to ensure that it is continuous; therefore, it is suitable for the computation of the complex cepstrum.

Reliably computing samples of the unwrapped phase of a given mixed-phase signal is an open problem: existing algorithms are not reliable and in many cases fail. The difficulty of computing the unwrapped phase carries over directly to the computation of the complex cepstrum. This thesis presents two composite algorithms that use the existing ones. The reliability and accuracy of the proposed algorithms have been demonstrated through numerous experiments, the goal being to render homomorphic
signal processing using the complex cepstrum more computationally reliable.

1.1 Problem Statement

The complex cepstrum is a sequence \( \hat{x}[n] \) that is related to a discrete-time signal \( x[n] \) by the following invertible transformation,

\[
X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \\
\hat{X}(z) = \log(X(z)) \\
\hat{x}[n] = \frac{1}{2\pi j} \oint \hat{X}(z)z^{n-1}dz.
\]

For \( \hat{x}[n] \) to be a stable sequence we require that the region of convergence of \( \hat{X}(z) \) contain the unit circle, which leads to the equivalent relationships

\[
\hat{X}(e^{j\omega}) = \log |X(e^{j\omega})|e^{j\chi X(e^{j\omega})} \\
\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{X}(e^{j\omega})e^{j\omega n}d\omega.
\]

The angle \( \chi X(e^{j\omega}) \) in Equation (1.2) is only unique to an additive multiple of \( 2\pi \) at each frequency \( \omega \); therefore, the imaginary part of \( \hat{X}(e^{j\omega}) \) is ambiguous. To remove this ambiguity and for \( \hat{X}(e^{j\omega}) \) to exist we require that \( \chi X(e^{j\omega}) \) be a continuous function of \( \omega \). The unwrapped phase \( \arg |X(e^{j\omega})| \) is defined as the unique instance of \( \chi X(e^{j\omega}) \),

\[
\arg |X(e^{j\omega})| = \chi X(e^{j\omega}) + 2\pi k(\omega) \quad k \in \mathbb{Z},
\]

where the additive multiple of \( 2\pi \) at each frequency is chosen to satisfy the continuity condition.

For \( \hat{X}(e^{j\omega}) \) to be a valid DTFT of the complex cepstrum \( \hat{x}[n] \) it must be periodic with period \( 2\pi \); therefore, the unwrapped phase function must be periodic. When
the input signal is real, the complex cepstrum is also real and \( \hat{X}(e^{j\omega}) \) is a conjugate symmetric function. This implies that when the input signal is real, the unwrapped phase function is odd symmetric. Refer to [1] for more details on the complex cepstrum and its properties.

Reliable computation of the unwrapped phase is an open problem to which existing solutions cannot guarantee accuracy for any given input sequence. The difficulty of the problem becomes apparent in the next chapter where existing algorithms are presented. This thesis focuses on computing the unwrapped phase of the DTFT of finite-length discrete-time signals.

1.2 Instances of the Phase and the Derivative of the Unwrapped Phase

In this section we present two instances of the phase function: the unwrapped phase and the principal value of the phase. We also present the derivative of the unwrapped phase.

A finite-length discrete-time signal \( x[n] \) has a z-transform of the form

\[
X(z) = Az^r \prod_{k=1}^{M_i}(1 - a_k z^{-1}) \prod_{k=1}^{M_o}(1 - b_k z) \quad |a_k|, |b_k| < 1.
\]  

(1.5)

\( M_i \) corresponds to the number of zeros inside the unit circle, and \( M_o \) the number outside. The term \( z^r \) is a time shift that contributes a linear phase term with slope \( r \). When the signal is causal \( r = -M_o \). \( A \) is real when \( x[n] \) is real. The phase of \( X(e^{j\omega}) \) is, therefore, equal to

\[
\varphi X(e^{j\omega}) = \varphi A + \varphi e^{j\omega r} + \sum_{k=1}^{M_i} \varphi(1 - a_k e^{-j\omega}) + \sum_{k=1}^{M_o} \varphi(1 - b_k e^{j\omega}).
\]

(1.6)
1.2.1 Principal Value of the Phase

Section 1.1 presented the unwrapped phase \( \arg |X(e^{j\omega})| \) as a unique instance of the phase function. Another unique instance is the principal value \( \text{ARG}|X(e^{j\omega})| \) for which the \( 2\pi \) ambiguity is removed by restricting the value of \( \text{ARG}|X(e^{j\omega})| \) at any \( \omega \) to the range \([-\pi, \pi]\).

The principal value can be calculated using the arctangent routine

\[
\text{ARG}|X(e^{j\omega})| = \arctan \left( \frac{X_I(e^{j\omega})}{X_R(e^{j\omega})} \right),
\]

where the subscripts \( R \) and \( I \) denote the real and imaginary parts. This calculation clearly forces

\[
-\pi < \text{ARG}|X(e^{j\omega})| \leq \pi,
\]

and thus values of the phase outside this range will be “wrapped” so that they fall within it; for this reason the principal value is sometimes referred to as the “wrapped phase”. The wrapping causes the principal value to have discontinuities in the form of jumps of \( 2\pi \) as seen in Figure 1-1. It is also important to note that

\[
\text{ARG}|X(e^{j\omega})| = \text{mod} \left\{ \text{ARG}|A| + \text{ARG}|e^{j\omega}| + \sum_{k=1}^{M_1} \text{ARG}|(1-a_k e^{-j\omega})| + \sum_{k=1}^{M_0} \text{ARG}|(1-b_k e^{j\omega})| \right\} 2\pi.
\]

![Figure 1-1: Example Wrapped Phase Function](image-url)
1.2.2 Derivative of the Unwrapped Phase

The phase derivative \( \arg' |X(e^{j\omega})| \) is the derivative of the unwrapped phase with respect to \( \omega \). Exact values of the derivative may be computed using

\[
\arg' |X(e^{j\omega})| = \Im\{\frac{X'(e^{j\omega})}{X(e^{j\omega})}\}
\]

\[
= \frac{X_R(e^{j\omega})X_I'(e^{j\omega}) - X_I(e^{j\omega})X_R'(e^{j\omega})}{X_R^2(e^{j\omega})X_I^2(e^{j\omega})},
\]

where \( \Im\{W\} \) is the imaginary part of \( W \), the superscript ' denotes differentiation with respect to \( \omega \), and \( X'(e^{j\omega}) \) can be obtained by:

\[
X'(e^{j\omega}) = \sum_{n=-\infty}^{\infty} (-jnx[n])e^{-jwn}.
\]

The derivative is a linear operator and therefore

\[
\arg' |X(e^{j\omega})| = \arg' |e^{j\omega}| + \sum_{k=1}^{M_t} \arg' |(1 - a_k e^{-j\omega})| + \sum_{k=1}^{M_o} \arg' |(1 - b_k e^{j\omega})|. \tag{1.13}
\]

1.2.3 Unwrapped Phase

Section 1.1 placed three restrictions on the unwrapped phase function for it to be used in the computation of the complex cepstrum:

The first is that the unwrapped phase needs to be a continuous function of \( \omega \). This condition therefore forces \( a_k, b_k \neq 1 \) in (1.5), which is equivalent to \( X(z) \) not having any zeros on the unit circle. A zero on the unit circle at frequency \( \omega_0 \) contributes a discontinuity of \( \pi \) to the phase function at that frequency, thus contradicting the continuity condition.

The linear phase term \( \omega \) in (1.5) is undesirable because it leads to discontinuities in the unwrapped phase at \( \omega = 2\pi \); hence, it should not appear in the unwrapped phase. This can be done either by computing the unwrapped phase of \( x[n] \) and subtracting out the linear term, or by appropriately shifting the input.

Finally, when \( x[n] \) is real-valued the unwrapped phase must be odd symmetric.
However, if $A < 0$ then the unwrapped phase will contain a constant term equal to $\pi$ added at each frequency of the phase. These effects can be fixed by subtracting out the constant $\pi$ or can be avoided by using $-x[n]$ as the input signal. An example of the unwrapped phase is plotted in Figure 1-2.

The unwrapped phase is related to the principal value by the addition of a multiple of $2\pi$ at each frequency

$$\arg |X(e^{j\omega})| = \arg (e^{j\omega}) + 2\pi l(\omega) \quad l \in \mathbb{Z}. \quad (1.14)$$

Integrating the phase derivative also yields the unwrapped phase

$$\arg |X(e^{j\omega})| = \int_0^\omega \arg' |X(e^{j\omega})|d\omega. \quad (1.15)$$

It is also possible to obtain the unwrapped phase of a signal $x[n]$ by summing up the phase contribution of the individual zeros, as follows:

$$\arg |X(e^{j\omega})| = \sum_{k=1}^{M_t} \arg |(1 - a_k e^{-j\omega})| + \sum_{k=1}^{M_o} \arg |(1 - b_k e^{j\omega})|. \quad (1.16)$$

Figure 1-2: a) Example unwrapped phase function. b) Unwrapped phase function without the linear phase and the contribution of the additive $\pi$ constant.
1.3 Example Phase Functions

This section presents several examples of the wrapped and unwrapped phase functions. The goal is to study how these functions vary with the locations in the $z$-plane of the zeros of the $z$-transform of a signal.

Consider a real finite-length signal with one pair of conjugate zeros at angles $\pi/4$ and $-\pi/4$ and radius $r$. The $z$-transform of this signal is

$$X(z) = (1 - re^{j\pi/4}z^{-1})(1 - re^{-j\pi/4}z^{-1}).$$  \hspace{1cm} (1.17)

Figure 1-3 shows the unwrapped phase and phase derivative of $X(e^{j\omega})$ for $r = .8, .9, &.99$. Note that in this case the unwrapped phase is equal to the principal value.

![Unwrapped Phase and Phase Derivative](image)

Figure 1-3: a) Unwrapped phase function. b) Phase Derivative.

Figure 1-4 shows the unwrapped phase with the contribution of the linear phase component, the principal value of the phase, and the phase derivative of $X(e^{j\omega})$ for $r = \frac{1}{8}, \frac{1}{9}, & \frac{1}{99}$. 

![Unwrapped Phase with Linear Component](image)
Figures 1-3 and 1-4 demonstrate that the closer the zeros of the z-transform are to the unit circle the sharper the phase change at frequencies in their vicinity. These sharp variations are the principle cause of the difficulty of phase unwrapping.
Chapter 2

Existing Algorithms to Compute the Unwrapped Phase

When the complex cepstrum is computed, the discrete Fourier transform (DFT) is used instead of the DTFT. The DFT is a sampled version of the DTFT, usually at the roots of unity $\omega = 2\pi k/N$ where $k \in \mathbb{Z}$ and $N \in \mathbb{N}$.

$$\hat{X}[k] = \hat{X}(e^{j2\pi k/N}) = \log |X[k]| + j \arg |X[k]|$$

Hence, only the corresponding samples of the unwrapped phase are needed. This section will describe several existing algorithms to compute these samples.

2.1 Phase Unwrapping by Detecting Discontinuities

The unwrapped phase can be obtained from the principal value of the phase, as in equation 1.14, by detecting and removing the discontinuities introduced by the arctangent routine. One way to implement this is by finding and removing differences between adjacent samples of the principal value that are greater than $\pi$ in magnitude
The algorithm starts by calculating $\Delta[k]$, the first difference of the samples of the principal value, using

$$\Delta[k] = \text{ARG}|X[k + 1]| - \text{ARG}|X[k]|.$$  \hspace{1cm} (2.1)

Next, any values of $\Delta[k]$ that lie outside the range $[-\pi, \pi]$ get wrapped to obtain $W\Delta[k]$ whose values lie within that range ($L[k]$ is an integer function of $k$):

$$W\Delta[k] = \Delta[k] + 2\pi L[k]$$ \hspace{1cm} (2.2)

such that $-\pi < W\Delta[k] \leq \pi$.

A correction sequence will be defined as:

$$C\Delta[k] = W\Delta[k] - \Delta[k] = 2\pi L[k].$$ \hspace{1cm} (2.3)

The unwrapped phase is then obtained by

$$\text{arg}|X[k + 1]| = \text{ARG}|X[k + 1]| + \sum_{m=0}^{k-1} C\Delta[m].$$ \hspace{1cm} (2.4)

The limitation of this method is that it only uses samples of the principal value of the phase to perform the unwrapping. It also assumes that there is at most a change of $\pi$ in the unwrapped phase from one sample in frequency to the next. This assumption fails when the phase varies rapidly; the closer the zeros of the z-transform of a signal are to the unit circle the more rapidly its DTFT phase varies. The ability of this method to detect the $2\pi$ discontinuities is determined by the distance $(2\pi/N)$ between adjacent samples of the phase. This distance is controlled by $N$, the size of the DFT; therefore, if the DFT size is too small relative to the variation of the phase this method will fail to compute the correct unwrapped phase.

One possible way to check that the correct DFT size is chosen, is to compute the unwrapped phase using progressively longer DFTs until two consecutive unwrapped phase functions match, and assume that is the correct DFT size. An efficient imple-
mentation uses the FFT and increases the size by consecutive powers of two.

This method mainly fails in the following two situations: the first, is when a match is not found before the maximum allowed (by the hardware) DFT size is reached. The second, is when a match is found, but it is not the correct unwrapped phase (i.e. an even larger DFT size would yield a different answer). An example of the latter is shown in Figure 2-1, the unwrapped phases with DFT sizes equal to $2^{14}$ & $2^{15}$ matched; however, the correct unwrapped phase function requires a $2^{16}$ DFT size. Note that the linear phase components are different between the correct and the computed phase functions, thus when they are removed the error between the two functions is no longer limited to be a multiple of $2\pi$ as in Figure 2-1 (b).

![Figure 2-1: a) Correct and computed unwrapped phase functions after removal of linear phase component. b) Error in the correct and computed unwrapped phase (before removal of linear phase component).](image)
2.2 Phase Unwrapping Using Adaptive Numerical Integration

Equation (1.15) suggests that samples of the unwrapped phase can be obtained by numerical integration of samples of the phase derivative. The accuracy of the numerical integration will be limited by the integration step size, which is the distance between adjacent samples. Unfortunately, it does not seem possible to determine \textit{a priori} the DFT size needed to obtain a correct unwrapped phase function.

Tribolet [4] suggested an adaptive numerical integration method. He assumed that the value of the unwrapped phase at the frequency sample \( \omega_k \) is known, and used the trapezoidal rule to perform the numerical integration and obtain an estimate of the unwrapped phase at \( \omega_{k+1} \):

\[
\text{arg} |X[k+1]| = \arg |X[k]| + \frac{\omega_{k+1} - \omega_k}{2} \{ \arg' |X[k+1]| + \arg' |X[k]| \}. \tag{2.5}
\]

However, depending on the step size \((\omega_{k+1} - \omega_k)\), the obtained value \(\text{arg} |X[k+1]|\) may be inconsistent relative to \(\text{ARG}|X[k+1]|\). The calculated value is assumed inconsistent when:

\[
|\text{arg} |X[k+1]| - \text{ARG}|X[k+1]| + 2\pi l[k+1]| \geq \epsilon, \tag{2.6}
\]

where \(l[k+1]\) is an integer and \(\epsilon\) is a set threshold. If the calculated value of \(\text{arg} |X[k+1]|\) is considered inconsistent then the step size \(\omega_{k+1} - \omega_k\) is halved. Equations (2.5) and (2.6) are then repeated until a consistent estimate of the unwrapped phase is obtained. The unwrapped phase at sample \(\omega_{k+1}\) is then computed by:

\[
\arg |X[k+1]| = \text{ARG}|X[k+1]| + 2\pi l[k+1] \tag{2.7}
\]

This method becomes less appealing the closer the zeros of the z-transform of the signal are to the unit circle. As was observed in Section 1.3, the closer a zero is to the unit circle the faster the unwrapped phase will vary in the vicinity of that zero, and
in consequence the smaller the integration step size needs to be. The smaller step size is obtained by repeated iterations of the step size adaptation. Hence, this algorithm will become more computationally intensive as more of the zeros of the z-transform of the signal approach the unit circle.

An issue of greater concern with this method is that a false yet consistent (according to (2.6)) estimate of a sample of the unwrapped phase could be obtained, causing the calculated unwrapped phase to be incorrect. Using a smaller threshold \( \varepsilon \) will most likely yield a more accurate unwrapped phase. However, this would increase the number of step size adaptations. It is important to note that there is no guarantee that an even smaller step size would not yield a different unwrapped phase function.

Figure 2-2 shows an example that clearly highlights these issues. Consider calculating samples of the unwrapped phase of a signal of length 1024 using adaptive numerical integration. Three different thresholds are used: \( \varepsilon = .1, .05, .01 \). Of these only \( \varepsilon = .01 \) yields the correct unwrapped phase. As can be seen in the figure the higher thresholds yielded erroneous unwrapped phase functions with the errors equal to multiples of \( 2\pi \). It is also important to note that a threshold of \( .1 \) required 2673 iterations of the step size adaptation, while \( \varepsilon = .05 \) required 3207 and \( \varepsilon = .01 \) required 5245. These numbers clearly show the inverse relationship between the value of the threshold and computational intensity; in our implementation of this algorithm we used \( \varepsilon = .05 \).

A modification to this method presented by Scott [5] uses the phase derivative at the two endslopes (arg' \( |X(\omega_{k+1})| \) & arg' \( |X(\omega_k)| \)) individually in the adaptive integration along with their average to improve the confidence of the algorithm. Another modification [6] to the adaptive integration method used the second derivatives of the phase (arg'' \( |X(e^{j\omega})| \)) to perform piecewise polynomial interpolation using cubic splines, as a substitute to the trapezoidal integration used by Tribolet.
2.3 Computation of the Unwrapped Phase Using Polynomial Factoring

Steiglitz and Dickinson [7] suggested calculating the roots of the polynomial (z-transform) of a finite length signal. The unwrapped phase is then obtained by adding together the phase contributions of each root, as in Equation (1.16). The success of this method depends on the success of the polynomial factoring algorithm used to calculate the roots.

A new efficient method for factoring high degree polynomials was proposed by Sitton and Burrus [8]. The method uses the Fast Fourier Transform (FFT) to create a search grid around the unit circle by evaluating the z-transform of the signal at evenly spaced samples on concentric circles with radii close to 1 in the z-plane (see Figure 2-3).

Local minima within this grid are used as a rough estimate of the location of zeros,
and then Newton’s method [9] is used to obtain the exact locations of the zeros. This polynomial factoring algorithm focuses on searching for zeros in the area close to the unit circle, and thus works better on polynomials with zeros clustered closer to the unit circle. However, this factoring method becomes less efficient and is more likely to fail on signals for which the zeros of their z-transforms are farther from the unit circle, and in such cases the correct unwrapped phase cannot be obtained from the factored roots. A detailed explanation of this polynomial factoring algorithm is provided in the next chapter.

2.4 Phase Unwrapping from the Time Series

McGowan [13] presented a method that uses the discrete-time sequence $x[n]$ to determine the additive factor required at each sample in frequency to unwrap the phase. The algorithm is based on finding the integer-valued function $l(\omega_{k+1})$ in the following equation:

$$\arg[X(e^{j\omega_{k+1}})] - \arg[X(e^{j\omega_k})] = -\{\arctan\left[\frac{X_I(e^{j\omega_{k+1}})}{X_R(e^{j\omega_{k+1}})}\right] + \pi l(\omega_{k+1})\}. \quad (2.8)$$

In this case, the $\arctan$ routine used returns a value for the phase between $-\pi/2$ and $\pi/2$. The term $l(\omega_{k+1})$ is found by tracking how many times the sign of $\frac{X_I(e^{j\omega})}{X_R(e^{j\omega})}$ changes as $\omega$ goes through a singularity of the ratio or similarly the change in sign of $X_I(e^{j\omega}) \cdot X_R(e^{j\omega})$ as $\omega$ goes through the zeros of $X_R(e^{j\omega})$. For example, as $\omega$
goes through a zero of $X_R(e^{j\omega})$ and the sign of $X_I(e^{j\omega}) \cdot X_R(e^{j\omega})$ changes from $+ve$ to $-ve$, $l(\omega_{k+1})$ is incremented by one and it is decremented by one as the sign of $X_I(e^{j\omega}) \cdot X_R(e^{j\omega})$ goes from $-ve$ to $+ve$. To calculate the number of sign changes $l(\omega_{k+1})$ at each $\omega_{k+1}$ the discrete-time sequence and Sturm Polynomials [14] are used.

This method unwraps the phase at any frequency directly from the time series without the need to obtain the unwrapped phase of any intermediate values. Also the unwrapped phase at a certain frequency can be obtained from a fixed number of calculations. However, a major limitation of this method is that it is restricted to extremely short time sequences. An improvement on this method, that allows for longer time sequences, was presented by Long [15]. This improvement, however, becomes less accurate as the signal length increases due to computational errors.

### 2.5 Iterative Method to Compute the Unwrapped Phase

This method proposed by Quatieri and Oppenheim in [17] computes the unwrapped phase of the DTFT of an input $x[n]$ in the following steps:

1. Iterative techniques are used to compute a minimum-phase sequence $x_{mp}[n]$ from the principal value of the phase of the mixed-phase input sequence $x[n]$ with the linear-phase component removed.

2. The log of the magnitude of the DTFT of the minimum-phase sequence $\log |X(e^{j\omega})|$ is then used to compute the unwrapped phase $\arg |X_{mp}(e^{j\omega})|$ of $x_{mp}[n]$ using the Hilbert transform relationships [2] of minimum-phase sequences.

3. The unwrapped phase $\arg |X(e^{j\omega})|$ is then obtained by adding the linear-phase component removed in step 1 to $\arg |X_{mp}(e^{j\omega})|$.

This was successfully tested in [17] on simple mixed-phase sequences. However, no results are presented in [17] for long sequences and it suffers from several complications. Specifically, $x_{mp}[n]$ is infinite in length even if $x[n]$ is finite length; therefore,
the DFT size used in the iteration needs to be large enough to ensure aliasing will not occur. Another issue with this algorithm is that it requires the linear phase component of the phase of $x[n]$ to be known \textit{a priori} which is in general not the case.

2.6 Phase Unwrapping by Isolating Sharp Zeros

The method presented in [18] uses the locations of the zeros of the signal that are close to the unit circle to compute the unwrapped phase. The method attempts to determine the radial positions of sharp zeros within segments of the z-plane. A segment is the frequency step between two consecutive samples of the DFT ($2\pi/N$). Sharp zeros as defined in [18] are ones that are close to the unit circle, particularly whose radii are between .99 and 1.01. A zero is in a segment if it lies between the two radial lines that pass through the DFT sample points and extend to infinity. The algorithm develops and uses two measures to obtain the radial positions:

1. Phase derivative with respect to radial distance, which is the derivative with respect radial distance of the phase change across a segment.

2. Phase derivative at any frequency sampling point $k$ with respect to radial distance.

Once the radial positions of the sharp zeros within a segment are computed, they are used to determine the theoretical phase change across that segment. This is then compared to the change in the principal value over that segment to obtain the appropriate multiple of $2\pi$ required to unwrap the phase.

This method performs well provided the signal does not have many sharp zeros clustered close to each other. When such clustering occurs a sharp zero might be missed resulting in a false unwrapped phase.
2.7 Summary

Of the methods presented in this section only the first three are of particular interest to us. Phase unwrapping by detecting discontinuities (DD) and by adaptive numerical integration (ANI) are the most widely used methods, and recent advances in polynomial factoring [8] have made computation of the unwrapped phase by polynomial factoring (PF) a more attractive method.

Both DD and ANI assume that there is a limit to how fast the unwrapped phase may vary in frequency. However, Section 1.3 demonstrated that the unwrapped phase has sharp variations near zeros that are close to the unit circle. Therefore, these methods will perform poorly on signals that have most of their z-transform zeros clustered near the unit circle. The polynomial factoring algorithm presented in [8] is capable of factoring high degree polynomials whose zeros are located close to the unit circle. Thus, using this factoring algorithm in PF leads to correct computation of the unwrapped phase for signals that cause DD and ANI to perform poorly. The fact that PF is complementary to both DD and ANI motivates the proposed composite methods presented in Chapter 4.
Chapter 3

Method for Factoring High Degree Polynomials

In [7] a new method for factoring high degree polynomials was presented that can factor polynomials with random coefficients with degree as high as a million. This algorithm uses the property that zeros of polynomials with random coefficients cluster in an annulus around the unit circle, and the area of the annulus gets smaller as the degree increases, as presented in Appendix A. This property allows the algorithm to focus the search for zeros in the vicinity of the unit circle.

3.1 Outline of the Algorithm

1. A search grid that is dense near the unit circle is created by sampling the z-plane at concentric circles. This is done by expanding and contracting the z-plane (by applying an exponential weighting to the input) and then computing its FFT.

2. The grid is searched for local minima. These present candidate locations for the zeros of the signal.

3. The candidate locations from step 2 are then polished using Newton’s [9] or Laguerre’s [12] method, and duplicates are removed. The convergence of Newton’s
method is quadratic, while that of Laguerre is cubic.

4. If the search missed some roots then the computed roots are used to deflate the original polynomial to obtain one of lower degree.

5. The deflated polynomial is then factored, and the newly found roots polished against the original polynomial.

6. Ideally steps 4 and 5 are repeated until all roots are found. However, the deflation process is highly prone to errors. When deflation fails, the algorithm returns to step 1 and searches over a denser grid.

3.2 Deflation Failure

One of the major sources of error in this polynomial factoring algorithm, is the deflation step. Deflation errors usually happen when either the input polynomial is ill-conditioned or the resulting deflated polynomial is ill-conditioned, i.e. one for which a slight change in the coefficients results in a large change in the zero locations. Below is an example of deflation failure. This example is taken from the authors of this factoring algorithm.

Consider a well conditioned polynomial of degree 1000 with coefficients drawn from a zero mean independent identically distributed uniform distribution. Assume the grid searching and polishing found 986 of the 1000 zeros of the z-transform of the signal. Moreover, assume the remaining 14 zeros are located near $z = 1$ in the $z$-plane, as shown in Figure 3-1(a).

The zeros found are then used to deflate the original polynomial. The resulting fourteenth degree polynomial has zeros that are plotted in Figure 3-1(b). Comparing these zero locations to the ones in Figure 3-1(a) shows that the deflation failed.
Figure 3-1: a) Z-plane plot showing the correct location of the remaining 14 zeros. b) Z-plane plot showing the zero locations of the deflated polynomial.

### 3.3 Discussion

This factoring algorithm focuses its search for zeros in the vicinity of the unit circle. Hence, this algorithm performs well on polynomials with zeros that cluster in a tight annulus around the unit circle. Appendix A shows that polynomials with coefficients that are i.i.d. random variables exhibit this property. Moreover, this property also holds for polynomials whose coefficients are samples of naturally occurring physical signals such as speech and Electroencephalography (EEG). However, when the assumption of the zeros clustering around the unit circle does not hold, this method will perform poorly and may even fail to factor the polynomial. Another case where this method could fail is when the input polynomial is ill-conditioned.
Chapter 4

Proposed Composite Methods

This chapter will present two new methods for phase unwrapping that are capable of reliably computing the unwrapped phase in situations where the existing algorithms may fail.

4.1 Motivation

Phase unwrapping by detecting discontinuities (DD) presented in Section 2.1 and phase unwrapping by adaptive numerical integration (ANI) presented in Section 2.2 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 was presented in Appendix A, 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 by polynomial factoring (PF) using the algorithm presented in Chapter 3 performs best when all the zeros are close to the unit circle. Consequently, 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 an algorithm that uses polynomial factoring with either DD or ANI, and combines their strengths while avoiding their weaknesses.
4.2 Overview

The proposed composite methods consider any signal $x[n]$ as a convolution of two signals $x_{UC}[n]$ and $x_{rem}[n]$.

\[ x[n] = x_{rem}[n] * x_{UC}[n] \quad \text{and} \quad X(z) = X_{rem}(z)X_{UC}(z) \quad (4.1) \]

$X_{UC}(z)$ contains the zeros of $X(z)$ that are problematic for DD and ANI, i.e. zeros that are closer to the unit circle ($z_{UC}$), and $X_{rem}(z)$ contains the remaining zeros of $X(z)$.

The proposed algorithm is a five step process:

1- Use polynomial factoring to find the zeros that are clustered near the unit circle, i.e. zeros of $X_{UC}(z)$.

2- Calculate the unwrapped phase contribution $(\arg |X_{UC}(e^{j\omega})|)$ of these zeros.

3- Obtain $x_{rem}[n]$ by deconvolving $x_{UC}[n]$ from $x[n]$.

4- Use either the DD or ANI algorithm to unwrap the phase contribution of the remaining zeros to get $\arg |X_{rem}(e^{j\omega})|$.

5- Add the unwrapped phase calculated in step 2 and in step 4 to obtain the total unwrapped phase.

\[ \arg |X(e^{j\omega})| = \arg |X_{UC}(e^{j\omega})| + \arg |X_{rem}(e^{j\omega})| \quad (4.2) \]

This five step process is a simplification of the actual proposed algorithm, and assumes that $x_{rem}[n]$ can correctly be calculated by deflation. However, this is in many cases not true; therefore, step 4 cannot be directly applied. Modifications are made to the DD and ANI methods to accommodate these cases. These modifications require the original signal $x[n]$ and $z_{UC}$ as inputs. Figure 4-1 is a block diagram representation of the proposed algorithms. The next sections of this chapter will describe the details of the polynomial factoring part, and those of the modified DD and ANI algorithms.
4.3 Polynomial Factoring and Deflation

The concepts of polynomial factoring and deflation in the algorithm are taken from Sitton and Burrus [8]. MATLAB code downloaded from the website [11] of the authors of [8] was modified and used to implement this part of the algorithm.

4.3.1 Steps to Find Zeros Close to the Unit Circle

This section briefly outlines the process used to search for zeros of the z-transform of a signal that are close to the unit circle.

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. Polish the candidate locations using Laguerre’s algorithm 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. (a) If deflation worked, output the computed zeros and the deflated polynomial, $x_{rem}[n]$.

(b) If deflation failed, and this is the first iteration through these steps, repeat all the steps while using a denser search grid in step 1.
(c) If deflation failed, and this is the second iteration, output the computed zeros.

Our experience showed that a max of two iterations should be allowed to attempt to obtain a deflated polynomial because a third iteration increases computation time and in most cases does not help. It is important to note that this section of the algorithm finds most of the zeros that are close to the unit circle, but will usually miss 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 4-2. The following sections will describe in detail how the search grid was created, and how the polynomial deflation was performed.

![Polynomial factoring and deflation block diagram.](image)

Figure 4-2: Polynomial factoring and deflation block diagram.

### 4.3.2 Search Grid

The search grid 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 DFT of the weighted sequence. Applying exponential weighting to a signal refers to a pointwise multiplication of the signal with a positive sequence $\alpha^n$

$$x_{\text{exp}}[n] = \alpha^n x[n].$$
Taking the DFT of \( x_{\exp}[n] \) is equivalent to sampling the z-transform of \( x[n] \) at a radius \( \alpha^{-1} \):

\[
X_{\exp}(e^{\frac{j2\pi k}{N}}) = \sum_{n=-\infty}^{\infty} \alpha^n x[n] e^{-\frac{j2\pi kn}{N}}
\]

\[
= \sum_{n=-\infty}^{\infty} x[n] (\alpha^{-1} e^{\frac{j2\pi k}{N}})^{-n}
\]

\[
X_{\exp}(e^{\frac{j2\pi k}{N}}) = X(\alpha^{-1} e^{\frac{j2\pi k}{N}}) \quad (4.3)
\]

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 (size of DFT) on each of the circles increases. A sample search grid is shown in Figure 4-3.

As is shown in the figure, the search grid only spans the inside of the unit circle. To

![Figure 4-3: Sample search grid, only upper right hand quadrant shown.](image-url)
obtain samples on concentric circles outside the unit circle the time-reversed signal 
\[x[-n]\] is used. A zero \(z_o\) of \(x[-n]\) corresponds to a zero \(z_o^{-1}\) of \(x[n]\) [1] because

\[
z\text{-transform}\{x[-n]\} = X(1/z).
\]

The goal of the grid search is to find candidate locations of zeros that are close
to the unit circle. However, as is observed in Figure 4-3 the choice of grid covers the
whole of the inside of the unit circle. The reason for this is that restricting the search
to an area around the unit circle does not considerably speed up the algorithm (since
the grid becomes sparse away from the unit circle). Also our experience showed that
not restricting the grid found more zeros which led the deflation step to be more
likely to succeed. In general, the majority of the candidate locations found by the
grid correspond to zeros that are close to the unit circle.

### 4.3.3 Deflation

The deflation step attempts to obtain a polynomial \(x_{\text{rem}}[n]\) of lower degree by removing the contributions of the computed zeros from the input polynomial. The deflation separates the computed zeros into two different groups: those that are close to the unit circle, and those that are not. The zeros far away from the unit circle are unfactored into a polynomial \(x_{\text{far}}[n]\) which is then used to deflate \(x[n]\) and obtain an intermediate polynomial \(\tilde{x}[n]\); this deflation is carried out in the frequency domain using point-wise division

\[
\tilde{X}[k] = X[k]/X_{\text{far}}[k].
\]

The zeros close to the unit circle are unfactored to form a polynomial \(x_{\text{close}}[n]\) which is then used to deflate \(\tilde{x}[n]\) and obtain \(x_{\text{rem}}[n]\); this deflation is done as a time domain deconvolution to avoid division by very small numbers that occur in \(X_{\text{close}}[k]\).
To check if deflation has worked, a polynomial $x_{\text{rec}}[n]$ is formed as

$$x_{\text{rec}}[n] = x_{\text{rem}}[n] * x_{\text{far}}[n] * x_{\text{close}}[n],$$

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

### 4.4 Modified DD and ANI Algorithms

When deflation fails, $x_{\text{rem}}[n]$ cannot be obtained. Thus, the DD and ANI algorithms cannot be directly applied. This section shows how these algorithms can be modified to compute $\arg |X_{\text{rem}}[k]|$, without access to $x_{\text{rem}}[n]$, using only the input signal $x[n]$ and the computed roots $z_{UC}$.

#### 4.4.1 Modified DD Algorithm

To apply the DD phase unwrapping algorithm all that is needed are samples of the wrapped phase $\text{ARG}|X_{\text{rem}}[k]|$ of $x_{\text{rem}}[n]$ (refer to Section 2.1). $\text{ARG}|X_{\text{rem}}[k]|$ can be computed by the following steps:

1. Compute $\text{ARG}|X[k]|$, 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 $2\pi$ the unwrapped phase of the computed roots from the wrapped phase of the input $x[n]$,

$$\arg |X_{\text{rem}}[k]| = \text{mod}\{\text{ARG}|X[k]| - \arg |X_{UC}[k]|\}_2\pi. \quad (4.4)$$

43
4.4.2 Modified ANI Algorithm

In ANI (Section 2.2) samples of the wrapped phase $\text{ARG}|X_{\text{rem}}[k]|$ and samples of the phase derivative $\text{arg}'|X_{\text{rem}}[k]|$ of $x_{\text{rem}}[n]$ are needed. The previous section showed how $\text{ARG}|X_{\text{rem}}[k]|$ can be computed. Similarly, $\text{arg}'|X_{\text{rem}}[k]|$ can be computed in the following steps:

1. Compute $\text{arg}'|X[k]|$, samples of the phase derivative of $x[n]$, using

$$\text{arg}'|X[k]| = \Re\{\frac{X'[k]}{X[k]}\}.$$

2. Compute $\text{arg}'|X_{UC}[k]|$ 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 $x[n]$,

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

This algorithm also requires that the step size be reduced when the numerical integration fails to produce consistent results (Section 2.2). With each step size reduction a new sample of $\text{ARG}|X_{\text{rem}}(e^{j\omega})|$ and $\text{arg}'|X_{\text{rem}}(e^{j\omega})|$ is required. This can be done using the same steps described above for one DFT point.

4.5 Robustness To Errors In Factored Zeros

The polynomial factoring part of the algorithm also computes an estimate of the error of each polished root. The error estimate, as proposed in the code obtained from [11], is based on the Newton correction $-f(z)/f'(z)$ evaluated at the location of the polished zero. 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.
To see this, consider a signal $x[n]$ with the following z-transform and unwrapped phase

$$X(z) = X_{rem}(z)(1 - z_o z^{-1}) \quad (4.5)$$

$$\arg |X(e^{j\omega})| = \arg |X_{rem}(e^{j\omega})| + \arg |(1 - z_o e^{-j\omega})|. \quad (4.6)$$

We assume that the polynomial factoring algorithm erroneously computed a zero at $\tilde{z}_o$ instead of $z_o$. Therefore, the wrapped phase and the phase derivative of the input deflated by the computed root are

$$\text{ARG}|\tilde{X}_{rem}(e^{j\omega})| = \text{ARG}|X_{rem}(e^{j\omega})| + \text{ARG}|(1 - z_o e^{-j\omega})| - \text{ARG}|(1 - \tilde{z}_o e^{-j\omega})|$$

and

$$\arg' |\tilde{X}_{rem}(e^{j\omega})| = \arg' |X_{rem}(e^{j\omega})| + \arg' |(1 - z_o e^{-j\omega})| - \arg' |(1 - \tilde{z}_o e^{-j\omega})|.$$  

Performing the unwrapping of $\text{ARG}|\tilde{X}_{rem}(e^{j\omega})|$ using either the modified DD or the modified ANI algorithms yields:

$$\arg |\tilde{X}_{rem}(e^{j\omega})| = \arg |X_{rem}(e^{j\omega})| + \arg |(1 - z_o e^{-j\omega})| - \arg |(1 - \tilde{z}_o e^{-j\omega})|.$$

Finally, combining the unwrapped phase of the found roots with that of $\tilde{X}_{rem}(e^{j\omega})$ gives the unwrapped phase of $X(e^{j\omega})$:

$$\arg |X(e^{j\omega})| = \arg |\tilde{X}_{rem}(e^{j\omega})| + \arg |(1 - \tilde{z}_o e^{-j\omega})|$$

$$= \arg |X_{rem}(e^{j\omega})| + \arg |(1 - z_o e^{-j\omega})| - \arg |(1 - \tilde{z}_o e^{-j\omega})| + \arg |(1 - \tilde{z}_o e^{-j\omega})|$$

$$= \arg |X_{rem}(e^{j\omega})| + \arg |(1 - z_o e^{-j\omega})| \quad (4.7)$$

Thus, we have shown that an error in a zero location due to the first section of the algorithm will be absorbed into the second section as the phase contribution of a pole, and will not affect the final computed unwrapped phase. It is important
to note, however, that if the errors are large they will interfere with the DD and ANI algorithms; therefore, zeros whose errors estimates are large are removed. Our experience has led us to remove zeros whose estimated errors were larger than $10^{-6}$.

### 4.6 Summary

The two proposed composite algorithms for computing the unwrapped phase exploit the fact that PF is complementary to DD and ANI. PF favors signals whose $z$-transform zeros are tightly clustered near the unit circle. DD and ANI favor signals whose $z$-transform zeros are located far from the unit circle. By combining these algorithms it is possible to unwrap the phase of most mixed-phase input signals, regardless of the distribution of their zeros.

The first proposed composite algorithm (CA1) combines polynomial factoring (Section 4.3) with phase unwrapping by detecting discontinuities (Section 2.1). The second proposed composite algorithm (CA2) combines polynomial factoring with phase unwrapping by adaptive numerical integration. CA2 is more accurate than CA1 due to the consistency test in ANI (Section 2.2); in our implementation of CA2 the threshold in the consistency test is chosen to be $\epsilon = .05$.

The next chapter presents experimental results that demonstrate the superiority of the proposed algorithms over the existing.
Chapter 5

Algorithm Evaluation

This chapter presents experimental results that demonstrate the superiority of the proposed composite algorithms over the existing methods. We compare phase unwrapping by detecting discontinuities (Section 2.1), phase unwrapping by adaptive numerical integration (Section 2.2), phase unwrapping by polynomial factoring (Section 2.3), and both proposed composite algorithms CA1 and CA2. The criterion for comparison is both speed and accuracy. The signals that are used to evaluate these algorithms are:

1. A synthetic signal with z-transform zeros that are clustered near the unit circle.

2. A synthetic signal with z-transform zeros that are away from the unit circle.

3. A synthetic signal with z-transform zeros that are both close to and far away from the unit circle.

4. A large number of synthetic signals with z-transform zeros whose locations are randomly chosen in the z-plane.

5. Sampled physical signals, specifically speech and EEG signals.

6. Filtered speech and EEG signals.

For the first three synthetic signals and the sampled physical signals a table will be used to summarize the performance of the different phase unwrapping algorithms.
The table consists of the following:

1. Whether the algorithm correctly computes the unwrapped phase.

2. Run time of a MATLAB implementation of the algorithm on a PC with a 1.6GHz Pentium M processor and 512 MB of RAM.

3. Additional relevant details:
   (a) Number of step size adaptation iterations required to unwrap the phase of $x[n]$ and $x_{rem}[n]$ in ANI and CA2 respectively.
   (b) Size of DFT used to unwrap the phase of $x[n]$ and $x_{rem}[n]$ in DD and CA1 respectively.

Our evaluation using the large number of synthetic signals is based on the average run time of the algorithms and percentage of times the algorithms correctly computed the unwrapped phase.

In this thesis we only consider real-valued signals; however, with minor modifications the algorithms perform in a similar fashion when applied to complex-valued input signals.

5.1 Algorithm Evaluation Using Synthetic Signals

This section uses synthetic signals to compare the different phase unwrapping algorithms. Since we generate the data, the exact locations of the roots are known and therefore from Section 2.3 the correct unwrapped phase can be computed and used to check the different algorithms. The locations of the zeros are chosen to evaluate algorithm performance for three main classes of signals:

1. signals with z-transform zeros that cluster near the unit circle.

2. signals with z-transform zeros that lie far away from the unit circle.

3. signals with z-transform zeros that are located both close to and away from the unit circle.
We also evaluate the performance using a large number of synthetic signals whose zero locations are randomly chosen to span these classes.

The synthetic signals are generated as a convolution of several exponentially weighted random real coefficient (either uniformly or normally distributed) signals. As is presented in Appendix A zeros of polynomials with random coefficients are clustered in a tight annulus around the unit circle. Applying exponential weighting to a sequence (4.3.2) changes the center radius of the annulus. Therefore, the zeros of the z-transform of the generated synthetic signals are in concentric annuli of varying center radii.

### 5.1.1 Synthetic Signal with Zeros Close to the Unit Circle

We generate a signal, of length=4093, with z-transform zeros concentrated in annuli with center radii of: 1, .9999 & 1/.9999. Figure 5-1 (a) shows the histogram of the distribution of the radii the zeros.

![Histogram of radii of zeros](image)

![Correct unwrapped phase](image)

Figure 5-1: Results for the synthetic signal with zeros close to the unit circle.
As we see from Table 5.1 PF works well, while DD and ANI perform poorly. Specifically, DD fails, and even though ANI correctly unwraps the phase it is highly inefficient, requiring 11572 step size adaptations and 87 seconds to run. On the other hand, in CA2 the polynomial factorization finds and removes most of the zeros close to the unit circle. Therefore, the phase contribution of the remaining zeros is unwrapped using only 105 step size adaptations.

### 5.1.2 Synthetic Signal with Zeros Far from the Unit Circle

We generate a signal, of length=4093, with z-transform zeros concentrated in annuli with center radii of: .9,.8 & 1/.9. Figure 5-2 (a) shows the histogram of the distribution of the radii of the zeros.

As we see from Table 5.2 DD and ANI correctly and efficiently unwrap the phase. Specifically, ANI requires no step-size adaptation and calculates the correct phase in only .2 seconds. This signal also shows an example where PF fails to factor the signal.
5.1.3 Synthetic Signal with Zeros Close to and Far from the Unit Circle

We generate a signal of length=4093, with z-transform zeros concentrated in annuli
with center radii of: 1, .9, .999, & 1/.9. Figure 5-3 (a) shows the histogram of the
distribution of the radii of the zeros.

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct</th>
<th>Run Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>No</td>
<td>0.57sec</td>
<td>DFT size used $N=2^{32}$</td>
</tr>
<tr>
<td>ANI</td>
<td>Yes</td>
<td>43.27sec</td>
<td>Number of step size adaptations=8398</td>
</tr>
<tr>
<td>PF</td>
<td>No</td>
<td>N/A</td>
<td>Factoring failed</td>
</tr>
<tr>
<td>CA1</td>
<td>Yes</td>
<td>8.02sec</td>
<td>DFT size used $N=2^{13}$</td>
</tr>
<tr>
<td>CA2</td>
<td>Yes</td>
<td>30.5sec</td>
<td>Number of step size adaptations=56</td>
</tr>
</tbody>
</table>

Table 5.3: Results for the synthetic signal with zeros close to and far from the unit
circle.

This synthetic signal illustrates a case where the existing algorithms perform
poorly. Specifically, Table 5.3 shows that DD and PF fail, while ANI requires 8398
step size adaptations to correctly unwrap the phase. In this example the advantage
of the composite algorithms is clear: the polynomial factorization found and removed most of the zeros close to the unit circle leaving a phase function that is unwrapped in only 56 step size adaptations; i.e. the unwrapped phase of $x_{\text{rem}}[n]$ varies significantly slower than that of $x[n]$. 

5.1.4 Large Number of Synthetic Signals with Randomly Chosen Zeros

We evaluate the algorithms for speed and accuracy using two thousand signals whose $z$-transform zero locations are randomly chosen to span most possible input signals. The synthetic signals are generated as a convolution of three exponentially weighted random signals, whose coefficients are zero mean uniformly distributed i.i.d. random variables. The signals used are of length 4096 with 510 of their $z$-transform zeros located in two concentric circles whose center radii are randomly chosen from the range $[0.2, 2.2]$. The rest of the signal’s $z$-transform zeros are located in a tight annulus around the unit circle. Table 5.4 shows the average run time of each algorithm and the percentage of times the correct unwrapped phase was computed. These results
clearly demonstrate the advantage of the composite algorithms over the existing ones.

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage Correct</th>
<th>Average Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>84.8%</td>
<td>1sec</td>
</tr>
<tr>
<td>ANI</td>
<td>84.8%</td>
<td>73sec</td>
</tr>
<tr>
<td>PF</td>
<td>48.5%</td>
<td>23sec</td>
</tr>
<tr>
<td>CA1</td>
<td>99.75%</td>
<td>8sec</td>
</tr>
<tr>
<td>CA2</td>
<td>99.95%</td>
<td>24sec</td>
</tr>
</tbody>
</table>

Table 5.4: Results from 2000 synthetic signals.

5.1.5 Discussion

This section presents examples of the three main classes of signals whose unwrapped phase we would like to correctly compute. The first two examples clearly demonstrate that PF is complementary to DD and ANI. The third example presents a situation where all three existing perform poorly. In all three cases the composite algorithms correctly computed the unwrapped phase, and the signal with z-transform zeros away from the unit circle is the only one where DD and ANI outperform CA1 and CA2. Also, it is clear from evaluating the algorithms on 2000 synthetic signals, that our proposed composite algorithms almost always compute the correct unwrapped phase. An important observation is that CA1 is about three times faster than CA2; however, CA2 only fails once while CA1 fails 5 times.
5.2 Algorithm Evaluation Using Speech Data

This section evaluates algorithm performance using a segment of recorded speech of the utterance “unwrapped phase” sampled at 8kHz (Figure 5-4).

![Speech segment of the utterance “Unwrapped Phase”](image)

Figure 5-4: Speech segment of the utterance “Unwrapped Phase”.

5.2.1 Speech Data

Figure 5-5 illustrates that the zeros of the z-transform of the speech signal are clustered close to the unit circle. Hence, this speech signal falls into the same class as the synthetic signal in Section 5.1.1. We therefore expect that PF and the composite algorithms will perform well, while DD and ANI will perform poorly. Table 5.5 confirms these expectations, with both DD and ANI failing to provide the correct unwrapped phase.

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct</th>
<th>Run Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>No</td>
<td>1.87sec</td>
<td>DFT size used N=2^64</td>
</tr>
<tr>
<td>ANI</td>
<td>No</td>
<td>N/A</td>
<td>Failed to unwrap</td>
</tr>
<tr>
<td>PF</td>
<td>Yes</td>
<td>26.2sec</td>
<td>Factoring worked</td>
</tr>
<tr>
<td>CA1</td>
<td>Yes</td>
<td>17.26sec</td>
<td>DFT size used N=2^13</td>
</tr>
<tr>
<td>CA2</td>
<td>Yes</td>
<td>26.23sec</td>
<td>Number of step size adaptations=59</td>
</tr>
</tbody>
</table>

Table 5.5: Results for speech data.
5.2.2 Filtered Speech Data

In this section the algorithms are evaluated on the speech utterance filtered with a 23rd order Butterworth filter whose cutoff frequency is $\pi/2$. This case is of particular interest because polynomial factoring fails and the roots of the signal are, therefore, unknown. Consequently, it is not possible to check that the computed unwrapped phase is correct. In this situation we consider the unwrapped phase from CA2 to be the correct one, because as was seen in Table 5.4 it is the most accurate. In addition, only 22 step-size adaptations were required in CA2 to unwrap the phase of $x_{rem}$, this indicates that the polynomial factoring section of the algorithm removed most of the zeros that are problematic for ANI. Thus, the confidence in the computed unwrapped phase is increased.
<table>
<thead>
<tr>
<th>Method</th>
<th>Correct</th>
<th>Run Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>Assumed Yes</td>
<td>1.91sec</td>
<td>DFT size used N=2^64</td>
</tr>
<tr>
<td>ANI</td>
<td>Assumed Yes</td>
<td>337.12sec</td>
<td>Number of step size adaptations= 20191</td>
</tr>
<tr>
<td>PF</td>
<td>No</td>
<td>N/A</td>
<td>Factoring failed</td>
</tr>
<tr>
<td>CA1</td>
<td>Assumed Yes</td>
<td>111.802sec</td>
<td>DFT size used N=2^16</td>
</tr>
<tr>
<td>CA2</td>
<td>Assumed Yes</td>
<td>97.34sec</td>
<td>Number of step size adaptations=22</td>
</tr>
</tbody>
</table>

Table 5.6: Results for filtered speech data.

Figure 5-6: Results for filtered speech data.
5.3 Algorithm Evaluation Using EEG Data

In this section we evaluate the algorithms using EEG data, downloaded from [10], of a subject performing a multiplication task collected at Colorado State University’s Computer Science Department. The signal is EEG electrode data sampled at 250Hz for 10 seconds (Figure 5-7).

![EEG data while performing a multiplication task](image)

Figure 5-7: EEG data while performing a multiplication task.

5.3.1 EEG Data

The zeros of the z-transform of the EEG signal, like those of the speech signal, are clustered around the unit circle (Figure 5-8). Therefore, we expect similar performance. In this case, both DD and ANI failed to compute the correct unwrapped phase, while PF and the proposed composite algorithms correctly unwrapped the phase.

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct</th>
<th>Run Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>No</td>
<td>0.49sec</td>
<td>DFT size used N=2^{32}</td>
</tr>
<tr>
<td>ANI</td>
<td>No</td>
<td>63.2sec</td>
<td>Number of step size adaptations=13348</td>
</tr>
<tr>
<td>PF</td>
<td>Yes</td>
<td>4.7sec</td>
<td>Factoring worked</td>
</tr>
<tr>
<td>CA1</td>
<td>Yes</td>
<td>5sec</td>
<td>Found all roots in pol factoring part</td>
</tr>
<tr>
<td>CA2</td>
<td>Yes</td>
<td>5sec</td>
<td>Found all roots in pol factoring part</td>
</tr>
</tbody>
</table>

Table 5.7: Results for EEG data.
5.3.2 Filtered EEG Data

In this section the EEG data was filtered with a 23rd order Butterworth filter with a cutoff frequency of $\pi/2$ and used to evaluate the algorithms. Similar to the filtered speech data polynomial factoring failed. Thus, we cannot confirm that the unwrapped phase we compute is correct. We handle this case, as with the filtered speech, by assuming that CA2 computed the correct unwrapped phase. Note in Table 5.8 that the adaptive integration part of the algorithm required zero step size adaptations, which indicates that all of the zeros close to the unit circle were removed by the polynomial factoring section.

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct</th>
<th>Run Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>No</td>
<td>0.27sec</td>
<td>DFT size used N=2^{16}</td>
</tr>
<tr>
<td>ANI</td>
<td>No</td>
<td>42.74</td>
<td>Number of step size adaptations=7828</td>
</tr>
<tr>
<td>PF</td>
<td>No</td>
<td>N/A</td>
<td>Factoring failed</td>
</tr>
<tr>
<td>CA1</td>
<td>Assumed Yes</td>
<td>12.5sec</td>
<td>DFT size used N=2^{13}</td>
</tr>
<tr>
<td>CA2</td>
<td>Assumed Yes</td>
<td>30.96sec</td>
<td>Number of step size adaptations=0</td>
</tr>
</tbody>
</table>

Table 5.8: Results for filtered EEG data.
Figure 5-9: Results for filtered EEG data.

5.4 Summary

Based on the evaluation of the algorithms presented in this chapter and those of numerous other experiments, we suggest the following conclusions:

1. Phase unwrapping by detecting discontinuities is the fastest and most unreliable algorithm.

2. Phase unwrapping by adaptive numerical integration becomes highly inefficient and less reliable as the zeros of the signal approach the unit circle. This is a major issue since, as was shown in Appendix A, the zeros of the z-transform of sampled physical data cluster closely around the unit circle.

3. Computing the unwrapped phase by polynomial factoring works best on signals whose zeros are clustered around the unit circle, and is likely to fail if that is not the case.

4. The proposed composite algorithms have proven to be reliable for signals whose z-transform has many zeros close to the unit circle. They also run faster than ANI and PF. Another advantage of the proposed algorithms is that they perform
well regardless of the locations of the zeros of their z-transform with respect to the unit circle.

5. When CA2 is used, the confidence in the unwrapped phase increases with the decrease in total step size adaptations required to unwrap the phase of $x_{rem}[n]$. This is because a small total number of adaptations indicates that most of the zeros close to the unit circle have been found and removed by the polynomial factorization and deflation steps. Therefore, the adaptive integration step will most likely succeed.

6. Of the two proposed algorithms CA2 has proven to be more reliable than CA1, while taking about three times longer to compute the unwrapped phase.

It is only possible to guarantee that the correct unwrapped phase is computed when the z-transform of the signal can be correctly factored. However, if the number of zeros of the z-transform of the signal outside the unit circle are known, a test can be formulated to correctly tell when the unwrapping fails. The test simply compares the number of zeros outside the unit circle to the linear-phase term in the unwrapped phase [1]. Unfortunately, methods for computing the number of zeros outside the unit circle suffer from the same difficulties that arise in computing the unwrapped phase. Therefore, we cannot guarantee that the unwrapped phase computed using the proposed composite algorithms is correct. For example, CA1 and CA2 may fail on a signal whose z-transform contains zeros of high multiplicity that lie close to the unit circle. However, as was observed by many experiments, the proposed algorithms are reliable and outperform any of the existing algorithms. This allows us to conclude that in general the unwrapped phase computed using the proposed composite algorithms can be considered correct.

This thesis is a proof of concept, and our software implementation of the algorithms is suboptimal. Undoubtedly, refinements in the implementation will yield improved run time.
Appendix A

Zeros of Random Polynomials

This appendix summarizes why the roots of polynomials with independent and identically distributed (i.i.d.) random coefficients cluster near the unit circle. It also explains why the zeros of polynomials with random coefficients that are bounded in amplitude cluster near the unit circle. This result applies to samples of naturally occurring signals since they can be modeled as such. In this appendix we present the main results of [16], and adopt the notation used in the paper.

A.1 Zeros of I.I.D. Random Polynomials

Consider a polynomial $p_N(z)$ with complex coefficients $a_{N,k}$ and zeros $z_n$

$$p_N(z) = \sum_{k=0}^{N-1} a_{N,k} z^k.$$  

We denote the number of zeros of the polynomial that lie within an annulus bounded by $1 - \rho$ and $\frac{1}{1-\rho}$ using

$$v_N(\rho) = \# \{ z_n : 1 - \rho \leq |z_n| \leq \frac{1}{1-\rho} \}$$
where $0 < \rho < 1$. We also denote the number of zeros with angles between $\theta$ and $\phi$ using

$$v_N(\theta, \phi) = \#\{z_n : \theta \leq \arg z_n \leq \phi\}$$

where $0 \leq \theta < \phi \leq 2\pi$.

In [16] it is proven that when the $a_{N,k}$ are i.i.d. random variables with very general conditions on their distribution, e.g. a zero mean uniform distribution, the following holds in the $p^{th}$ mean:

$$\lim_{N \to \infty} \frac{v(\alpha(N)/N)}{N} = 1$$

(A.1)

and

$$\lim_{N \to \infty} \frac{v(\theta, \phi)}{N} = \frac{\phi - \theta}{2\pi}$$

(A.2)

where $\alpha(N) = o(N)$.

Equation (A.1) states that the zeros are located in an annulus around the unit circle that becomes smaller as $N$ increases. Equation (A.2) states that the zeros are uniformly distributed in angle. Figure A-1 shows the distribution of zeros for two i.i.d. uniformly distributed random variables of length $N = 512$ and $N = 2048$.

### A.2 Zeros of Bounded Coefficient Random Polynomials

In this section we show that a discrete-time signal obtained from sampling a naturally occurring signal in general has a z-transform with zeros that cluster near the unit circle. In practice, samples of naturally occurring signals can be considered to be bounded in amplitude random variables due to the randomness of the generating process, the measurement noise and the dynamic range of the analog to digital (A/D) converter. Note that we do not assume any additional structure, e.g. i.i.d. or wide sense stationarity.
We consider $p_{N,k}$, $v_N(\rho)$ and $v_N(\theta, \phi)$ to be defined as in the previous section. In [16] it was shown that the following theorem holds:

**Theorem A.1.** For $N \geq 1$, let $(a_{N,k})_{0 \leq k \leq N}$ be an array of random complex numbers such that $\mathbb{P}[a_{N,0} = 0] = 0$ and $\mathbb{P}[a_{N,N} = 0] = 0$ for all $N$, where $\mathbb{P}$ is the probability. Define

$$F_N = \log(\sum_{k=0}^{N} |a_{N,k}|) - \frac{1}{2} \log |a_{N,0}| - \frac{1}{2} \log |a_{N,N}|.$$  \hspace{1cm} (A.3)

If

$$\mathbb{E}[F_N] = o(N) \quad \text{as} \quad N \to \infty,$$ \hspace{1cm} (A.4)

where $\mathbb{E}$ is the expected value, then there exists a positive function $\alpha(N)$ satisfying $\alpha(N) = o(N)$ such that

$$\lim_{N \to \infty} \mathbb{E}\left[ \frac{1}{N} v_N\left(\frac{\alpha(N)}{N}\right) \right] = 1$$ \hspace{1cm} (A.5)

and

$$\lim_{N \to \infty} \mathbb{E}\left[ \frac{1}{N} v_N(\theta, \phi) \right] = \frac{\phi - \theta}{2\pi}$$ \hspace{1cm} (A.6)
This theorem states that if the distributions of the coefficients $a_{N,k}$ are chosen such that (A.4) is satisfied then the zeros of the polynomial cluster near the unit circle. Specifically, as $N$ increases, the expected fraction of zeros within an annulus with area that decreases with $N$ tends to 1. Moreover, the angles of the zeros are uniformly distributed.

**Corollary A.1.** Assume that $E[\log|a_{N,0}|] = o(N)$, and $E[\log|a_{N,N}|] = o(N)$, and that there exists a fixed $s > 0$ and a sequence $\varepsilon_N$ tending to zero such that

$$
\sup_{0 \leq k \leq N} E[|a_{N,k}|^s] \leq \varepsilon_N N. \quad (A.7)
$$

Then, $E[F_N] = o(N)$.

In practice, naturally occurring signals are real-valued and the A/D conversion has a full scale level $X_m$. Therefore, the output of the A/D is a sequence of random variables $a_{N,k}$ where $|a_{N,k}| \leq X_m$.

**Corollary A.2.** Let $a_{N,k}$ be a sequence of random variables such that $P[a_{N,0} = 0] = 0$, $P[a_{N,N} = 0] = 0$ and $|a_{N,k}| \leq X_m$ for some fixed $X_m$. Then, Corollary A.1 is satisfied for such a sequence.

**Proof.**

We know $\log(|a_{N,k}|) \leq \log(X_m)$ because $|a_{N,k}| \leq X_m$. Thus, $E[\log(|a_{N,0}|)] = o(N)$ and $E[\log(|a_{N,N}|)] = o(N)$.

We now show that for a fixed $s = 2$, (A.7) holds:

$E[|a_{N,k}|^2]$ is the second moment of the random variable $a_{N,k}$, thus,

$$
E[|a_{N,k}|^2] \leq E[X_m^2] = X_m^2. \quad (A.8)
$$

Define

$$
\varepsilon_N = \frac{\ln(X_m^2)}{N},
$$

64
which tends to zero as \( N \) tends to infinity. Therefore,

\[
\varepsilon_N^N = e^{\ln(X_m^2)} = X_m^2.
\] (A.9)

Equations (A.8) and (A.9) show that there exists a fixed \( s = 2 \) and a sequence \( \varepsilon_n = \frac{\ln(X_m^2)}{N} \) tending to zero such that equation A.7 in Corrolary A.1 is satisfied. Therefore, Theorem A.1 holds.

These analytical results were observed in sampled EEG and speech signals. Figure A-2 shows the distribution of zeros of the z-transform of sample speech signals of length \( N = 512 \) and \( N = 2048 \).

Figure A-2: Observed clustering near the unit circle of zeros of the z-transform of speech signals.
Appendix B

Cepstrum Bibliography

This appendix contains a bibliography, which we have compiled, on the complex cepstrum. The references are organized into nine categories: Signal, Computation, Speech/Audio, Filter Design, 2D/Image, Deconvolution, Geophysics, Applications and Other.

B.1 Signal

- Oppenheim, A. V.; “Nonlinear filtering of convolved signals”. Quarterly Progress Report, No: 80, Research Lab of Electronics, MIT.


B.2 Computation


### B.3 Speech/Audio


• Jingping Xu; Jingzhi Cheng; Yanjun Wu; “A cepstral method for analysis of acoustic transmission characteristics of respiratory system”. 

• Zilovic, M.S.; Ramachandran, R.P.; Mammone, R.J.; “Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions”. 

• Hwang, T.-H.; Lee, L.-M.; Wang, H.-C.; “Cepstral behaviour due to additive noise and a compensation scheme for noisy speech recognition”. 

• JeongJin Kim; MyungJin Bae; “On a modified cepstral pitch control technique for the high quality text-to-speech type system”. 


• In-Chul Hwang; Sung-Nam Kim; Young-Woo Kim; Soo-Won Kim; “An LPC cepstrum processor for speech recognition”. 

• Kobayashi, H.; Shimamura, T.; “A modified cepstrum method for pitch extraction”. 

• Tolba, H.; O'Shaughnessy, D.; “Automatic speech recognition based on cepstral coefficients and a mel-based discrete energy operator”. 


• Ephraim, Y.; Rahim, M.; “On second order statistics and linear estimation of cepstral coefficients”. 

• McCourt, P.; Vaseght, S.; Harte, N.; “Multi-resolution cepstral features for phoneme recognition across speech sub-bands”. 

• Digalakis, V.; Neumeyer, L.; Perakis, M.; “Quantization of cepstral parameters for speech recognition over the World Wide Web”. 


• Hong Kook Kim; Seung Ho Choi; Hwang Soo Lee; “On approximating line spectral frequencies to LPC cepstral coefficients”. *Speech and Audio Processing, IEEE Transactions on*, Volume: 8, Issue: 2, March 2000.


• Fjell, P. O.; "Use Of Cepstrum Method For Arrival Times Extraction Of Overlapping Signals Due To Multipath Conditions In Shallow-Water". *Journal Of The Acoustical Society Of America*, 59, 1, 1976.

B.4 Filter Design


B.5 2D/Image


**B.6 Deconvolution**


B.7 Geophysics


B.8 Applications


• Joshi, S. S.; Arora, K. K.; “Noise-Analysis Applications Made At Rajasthan-
Atomic-Power-Station, Including Primary Coolant Flow-Rate Measurements
Using N-16 Signal And The Cepstrum Technique Of Analysis”. Progress In
Nuclear Energy, 9, 1982.

• Khalili, N.; Hammond, J. K.; “Application Of Cepstral Techniques For The
Determination Of Reflection Coefficients For Dispersive Systems.2. Comparison
Between Theory And Experiment”. Mechanical Systems And Signal Processing,
7, 5, Sep, 1993.

• Khalili, N.; Hammond, J. K.; “Application Of Cepstral Techniques For The
Determination Of Reflection Coefficients For Dispersive Systems.1. Theory And

• Kim, J. T.; Lyon, R. H.; “Cepstral Analysis As A Tool For Robust Processing,
Deverberation And Detection Of Transients”. Mechanical Systems And Signal
Processing, 6, 1, Jan, 1992.

• Kirmse, D. W.; “Cepstral Analysis Of Hyperfine-Structure In Electron-Paramagnetic

• Klimov, A. V.; “Determination Of The State Of Gear Transmissions By The

• Lizzi, F.; King, D.; Feleppa, E.; Jaremko, N.; Wai, P.; “Liver-Tissue Character-
ization By Digital Spectrum And Cepstrum Analysis”. Ieece Transactions On
Sonics And Ultrasonics, 29, 3, 1982.

• Meadows, K. R.; Hardin, J. C.; “Removal Of Spurious Reflections From Comput-
tational Fluid Dynamic Solutions With The Complex Cepstrum”. Aiaa Journal,
30, 1, Jan, 1992.

• Minamiyama, M.; Yagi, S.; “Ultrasonic Measurement Of Small Vessel Dimen-
sions By Mem Cepstrum”. Journal of the Physiological Society Of Japan, 50,

Depths Using Cepstrum”. Journal Of The Acoustical Society Of America, 58,
1975.

• Molinari, L.; Dumermuth, G.; “Cepstral Analysis Of Background Eeg”. Electro-

• Nyitrai, L.; Nagy, Z.; “Quantitative-Determination Of Left-To-Right Shunts By
Transfer-Function Tf And Cepstral Transformation Cet”. European Journal Of
Nuclear Medicine, 9, 7, 1984.

• Nyitrai, L.; Nagy, Z.; Dicso, F.; Rusznak, M.; “Quantitative Determination Of
Left-To-Right Shunts By Transfer Function Tf And Cepstral Transformation
Cet In Clinical Practice”. Kekes, E., L. Matos And L. Mihoczy (Ed.). Non-
Invasive Cardiology, 1985; 4th European Conference On Mechanocardiography,
Budapest, Hungary, September 11-14, 1985. Xxii+658p. Elsevier Science Pub-

waves on gels”. Japanese Journal Of Applied Physics Part I-Regular Papers

• Pattichis, C. S.; Elia, A. G.; “Autoregressive and cepstral analyses of motor
unit action potentials”. Medical Engineering & Physics, 21, 6-7, Jul-Sep, 1999.


B.9 Other


Bibliography


