FAST AND RECURSIVE ALGORITHMS FOR MAGNITUDE RETRIEVAL FROM DTFT PHASE AT IRREGULAR FREQUENCIES

Andrew E. Yagle

Dept. of EECS, The University of Michigan, Ann Arbor MI 48109-2122 aey@eecs.umich.edu

ABSTRACT

We derive two new algorithms for reconstructing a discrete-time 1-D signal from the phase of its discrete-time Fourier transform (DTFT) at irregular frequencies. Previous algorithms for this problem have either required the computation of a matrix nullspace, requiring $O(N^3)$ computations, or have been iterative in nature; for the latter, the irregularity of the frequency samples precludes use of the fast Fourier transform. Our first algorithm requires only $O(N^2)$ computations ($O(N \log^3 N)$ asymptotically). In the special case of equally-spaced frequency samples, it is related to a previous algorithm. The second algorithm is recursive–at each recursion a meaningful magnitude retrieval problem is solved. This is useful for updating a solution; it also allows checking of the result at each recursion, avoiding any errors due to computational roundoff error and ill-conditioning of the problem.

1. INTRODUCTION

The problem of reconstrucing a discrete-time signal from its DTFT phase has been studied extensively over the last fifteen years. Ref. [1] gives the basic uniqueness results for reconstrucing a 1-D discrete-time finite-support signal from its DTFT phase, and two approaches were proposed. The so-called *closed-form solution* required solution of a large and ill-conditioned linear system of equations (see (2) below). The other was an alternating projections (AP) algorithm, in which the support and given DTFT phase value constraints were alternately imposed in the time and frequency domains. When the frequencies at which the phases are known are uniformly distributed on the unit circle, the linear system of equations has a Toeplitz-plus-Hankel structure [2]. This is related to a special case of the first of our two algorithms. We will not attempt to review all recent approaches to this problem; for a more complete reference list see [3].

The main application of reconstruction of Fourier magnitude from phase is blind deconvolution of an unknown symmetric or Hermitian blurring function to determine an unknown signal, when only the supports of both unknown signals are known (see below). For other applications see [3]. The results of this paper are applicable if phase information at some frequencies is difficult to measure or corrupted by noise. Actually, this work was motivated by some recent work in phase retrieval, in which the central part of the approach requires the solution of a magnitude retrieval problem at nonuniformly-spaced frequencies.

This paper derives two new algorithms for the magnitude retrieval from DTFT phase problem. The first algorithm reduces the computation required for an exact, noniterative solution from $O(N^3)$ to $O(N^2)$ computations (and asymptotically $O(N \log^3 N)$ for sufficiently large N). The main requirement is solution of a block-Toeplitz linear system of equations with 2×2 blocks, in lieu of finding the nullspace of an unstructured linear system of equations. The second algorithm is recursive, and solves magnitude retrieval problems of increasing size. This has two major advantages:

1. At each recursion, it can be confirmed that the solution at that stage does indeed solve its problem. Thus the algorithm is self-correcting, since errors due to roundoff or ill-conditioning can be detected and corrected (using AP) before they increase;
2. The solution to a magnitude retrieval problem can be updated if additional phase information becomes known. If the phase values come in a data stream, this can be utilized to incorporate each phase value as it arrives.

This conference paper is a condensed version of the longer paper [3].

2. BASICS OF MAGNITUDE RETRIEVAL

2.1. PROBLEM FORMULATION

The z-transform and DTFT of the discrete-time signal $x(n)$ are defined as

$$X(z) = \sum x(n) z^{-n} \qquad (1)$$

$$X(e^{j\omega}) = \sum x(n)e^{-j\omega n} = |X(e^{j\omega})|e^{jARG[X(e^{j\omega})]}$$

where $ARG[X(e^{j\omega})]$ is the principal value (wrapped phase) of the phase of $X(e^{j\omega})$. In the sequel, "phase" refers to the former.

We are given that $x(n)$ is real, with support on the interval $[-N/2, N/2]$ for some even N; thus $x(n)$ is nonzero only for $N+1$ (an odd number of) points, centered on the origin $n = 0$. This avoids ambiguities due to linear phase. The goal is to reconstruct $X(e^{j\omega})$ (or equivalently $x(n)$) from its phase $\phi(\omega) = ARG[X(e^{j\omega})]$ at any N frequencies $0 < \omega_1 < \omega_2 < \ldots < \pi$. This restriction is necessary since $ARG[e^{-j\omega_n}] = -ARG[e^{j\omega_n}]$; there is no additional information at the conjugate frequencies. Otherwise, there is no restriction on the values of ω_k; they need not be evenly spaced. We refer to irregularly-spaced frequencies as irregular frequencies.

From [1], this problem has a unique solution to an overall scale factor, provided that $X(z)$ has no zeros in reciprocal pairs. We assume this throughout in the sequel, and we consider the problem solved when $x(n)$ is determined to within an overall scale factor (which may be negative).

2.2. Previous Solutions

The obvious way to solve this problem is as follows [1]. Insert the data into (1), multiply by $e^{-j\phi(\omega_k)}$, and take the imaginary part. This produces the following linear system of N equations in $N+1$ unknowns (the underdetermination produces the arbitrary scale factor):

$$\sum_{n=-N/2}^{N/2} x(n)sin(\omega_k n + \phi(\omega_k)) = 0, 1 \le k \le N \qquad (2)$$

Determination of the nullspace $x(n)$ of this unstructured linear system requires $O(N^3)$ computations. Furthermore, this system tends to be ill-conditioned [1].

The other way is to use an alternating projections (AP) algorithm. At each iteration, the present iterate is projected on the finite support $[-N, N]$ in the time domain, and on the given phases $\phi(\omega_k)$ in the frequency domain. Since both sets are convex, the algorithm is guaranteed to converge. However, there are two major problems with this approach:

1. Since the phase is given at irregular frequencies ω_k, the FFT cannot be used to go back and forth between the time and frequency domains. This greatly increases the amount of computation;

2. We have observed that having only *partial* phase information (knowledge of $\phi(\omega)$ only at the irregular ω_k) slows down convergence of the AP algorithm (see [3] for details).

2.3. Application to Blind Deconvolution

Suppose we observe $y(n) = h(n) * x(n)$, where $h(n)$ and $x(n)$ are both known to have support on finite intervals, $h(n)$ is known to be symmetric ($h(-n) = h(n)$), and $x(n)$ can be assumed to have no zeros in reciprocal locations. Otherwise both the blur function $h(n)$ and the desired signal $x(n)$ are unknown. The blind deconvolution problem is to compute both $h(n)$ and $x(n)$ from $y(n)$. The assumption that the unknown blurring, defocusing, or point-spread function $h(n)$ is even is often reasonable in optics, for example.

Taking the DTFT of $y(n) = h(n) * x(n)$ and taking ARG of the result gives $ARG[Y(e^{j\omega})] = ARG[X(e^{j\omega})]$. The reconstruction of $x(n)$ from $y(n)$ (to a scale factor) is thus a magnitude retrieval problem.

3. FAST ALGORITHM FOR MAGNITUDE RETRIEVAL

3.1. Interpolation of $y(n)$ from $\phi(\omega_k)$

We define the signal $y(n)$ such that:

1. $y(n)$ has support $[-N-1, N+1]$ (twice the size of the support of $x(n)$);

2. $y(n)$ has DTFT phase $\phi(\omega_k)$;

3. $y(n)$ has even part $\delta(n)$, so that $Y(z) - 1$ is an odd function.

$Y(z)$ can be computed recursively using

$$Y(z) = 1 + (z - \frac{1}{z})[Y_0 + Y_1(z - 2\cos(\omega_1) + \frac{1}{z})$$

$$+Y_2(z - 2\cos(\omega_1) + \frac{1}{z})(z - 2\cos(\omega_2) + \frac{1}{z})$$

$$+ \ldots]|_{z=e^{j\omega_k}} = tan(\phi(\omega_k)) \qquad (3)$$

by setting $z = e^{j\omega_k}$ for $k = 1, 2 \ldots$ in succession. After all N phase values have been incorporated, the resulting $Y(z)$ clearly satisfies the above conditions and is almost surely unique.

$y(n)$ is an almost-odd function which satisfies the magnitude retrieval problem, but its support is too big by a factor of two. This shows why this recursive interpolation formula cannot be used to solve the magnitude retrieval problem directly–we would need to interpolate at both $e^{\pm j\omega_k}$, resulting in a solution with too large a support.

3.2. Reduction from $y(n)$ to $x(n)$

Let $x_e(n) = (x(n)+x(-n))/2$ be the even part of $x(n)$. Since $x_e(n)$ has zero phase (or phase π, but this only appears in the overall scale factor), $x_e(n)*y(n)$ has the correct phases $\phi(\omega_k)$. The even part of $x_e(n)*y(n)$ is $x_e(n)$ (as it should be), but its odd part $x_e(n)*(y(n)-\delta(n))$ has support on $[-\frac{3N}{2}-1, \frac{3N}{2}+1]$, which is now too large by a factor of three!

However, we know that the even part of $x_e(n)*y(n)$ is indeed $x_e(n)$. Since the even part is correct, the odd part must equal the odd part $x_o(n) = (x(n)-x(-n))/2$ of $x(n)$ at the frequencies ω_k:

$$X_o(z)|_{z=e^{j\omega_k}} = X_e(z)Y(z)|_{z=e^{j\omega_k}}. \tag{4}$$

This can be rewritten as

$$X_0(z) \equiv X_e(z)Y(z) \, mod \, P(z), \tag{5}$$

$$P(z) = \prod_{k=1}^{N}(z - 2\cos(\omega_k) + \frac{1}{z})$$

which in turn can be rewritten as

$$Y(z)X_e(z) + P(z)Q(z) = X_0(z) \tag{6}$$

where

1. $Y(z)$ is known and has degree $2(N+1)$;
2. $P(z)$ is known and has degree $2N$;
3. $X_e(z)$ and $X_o(z)$ are unknown with degrees N;
4. $Q(z)$ is unknown and has degree $N+2$.

We cannot apply the Euclidian algorithm to (6) to compute $X_e(z)$ and $Q(z)$ from the known $Y(z)$ and $P(z)$, since the right side $X_o(z)$ is unknown. However, we can equate coefficients of z^i for $|i| > \frac{N}{2}$. This results in the linear system of equations

$$\begin{bmatrix} y(N+1) & 0 & | & 0 & 0 \\ y(N) & \ddots & | & p(N) & 0 \\ y(N-1) & \ddots & | & p(N-1) & \ddots \\ \vdots & \ddots & | & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x_e(\frac{N}{2}) \\ \vdots \\ q(\frac{N}{2}+1) \\ \vdots \end{bmatrix} = [0, 0 \ldots 0]^T \tag{7}$$

This is clearly a matrix with Toeplitz blocks. Hence the linear system of equations can be reorganized into a block-Toeplitz linear system with 2×2 blocks. This linear system can be solved using the multichannel Levinson algorithm in $O(N^2)$ computations. Due to the even symmetry of all of the signals, the size of the linear system can be reduced by a factor of two, at the price of making it Toeplitz-plus-Hankel. For sufficiently large N, the so-called "superfast" algorithms can be used to solve the linear system in $O(N \log^2 N)$ computations.

3.3. Summary of Procedure: Fast Algorithm

The overall procedure can be summarized as follows:

1. Compute $Y(z)$ from the phases $\phi(\omega_k)$ using (3);
2. Compute $P(z)$ from the frequencies ω_k using (5);
3. Set up and solve the linear system of equations (7), yielding $X_e(z)$ and $Q(z)$;
4. Compute $X_o(z)$ with (6). $X(z) = X_e(z)+X_o(z)$.

Note that for large N, each of these steps can be performed in $O(N \log^3 N)$ or fewer computations (see [4] for details). Hence the overall algorithm is $O(N \log^3 N)$ for sufficiently large N. This is a considerable improvement over the $O(N^3)$ computations required by solution to the linear system (2).

3.4. Relation to Previous Algorithm for Equally-Spaced Frequencies

Consider now the special case $\omega_k = 2\pi\frac{k}{2N+1}$, which corresponds to equally-spaced frequencies on the unit circle for the zero-padded discrete Fourier transform (DFT) of $x(n)$. In this case,

1. The interpolation (3) becomes an inverse DFT;
2. $P(z) = \frac{z^{2N+1}-1}{z-1}$;
3. (7) can be derived more easily by taking the inverse DFT of $jX_o(k) = X_e(k)\tan\phi(k)$, where indices now pertain to the DFT.

This results in a Toeplitz-plus-Hankel linear system of equations, which can be solved in $O(N^2)$ operations. There is no real advantage over irregular frequency samples, but it is interesting to note that this approach is closely related to [2], in which the inverse DFT of $X(k) = X(-k)e^{j2\phi(k)}$ was taken, resulting in a different Toeplitz-plus-Hankel linear system of equations.

In this regard, the approach we have taken can be viewed as a generalization of [2] to irregular frequency samples, with no significant attendant increase in computatation. Our approach here reduces to [2] in the special case of "complete" phase information at equally-spaced frequencies.

4. RECURSIVE ALGORITHM FOR MAGNITUDE RETRIEVAL

We now present another magnitude retrieval algorithm that is *recursive*: At each step an actual magnitude retrieval problem, which is a subproblem of the given problem, is solved. At each recursion, the size of the problem solved increases by one, until the actual problem is reached. Note that this is similar in concept to the Levinson algorithm, which recursively solves Toeplitz systems of equations of increasing size.

Although this algorithm requires $O(N^3)$ computations, and requires the storage of all the solutions to the smaller problems (unlike the Levinson algorithm, which only requires storage of the most recent solution), the computation and storage requirements are still only a fraction of those for direct solution of (2). And the recursive nature of this algorithm makes it very useful if computational roundoff error may be a problem–at each recursion, we can confirm that we have solved the problem we should be solving, and if there is a slight error an AP algorithm can be used to correct the error. Thus the algorithm is self-checking and self-correcting.

4.1. Problem Formulation

We now alter the formulation of the magnitude retrieval problem. We are still given the phase $\phi(\omega_k) = ARG[X(e^{j\omega_k})]$ of $x(n)$, but now $x(n)$ is defined to have support for $n \geq 0$.

Define the *nested set* of magnitude retrievals:

$$ARG[X_K(z)|_{z=e^{j\omega_k}}] = \phi(\omega_k), 1 \leq k \leq K \leq N \quad (8)$$

Thus $X_K(z)$, which has degree K, has the specified phase $\phi(\omega_k)$ at the K frequencies $\omega = \omega_1 \ldots \omega_K$. $x_K(n)$ has support on $[0, K]$ and has $K+1$ nonzero values. At each recursion, we update from $\{X_1(z) \ldots X_K(z)\}$ (the results from all previous recursions are available) to $\{X_1(z) \ldots X_{K+1}(z)\}$ (we compute the solution $X_{K+1}(z)$ to the next larger problem).

4.2. Problem Solution

Let $W_{K+1}(z)$ be *any* polynomial of degree $K+1$ which has phase $\phi(\omega_1)$ at $z = e^{j\omega_1}$. For example, we may use $W_{K+1} = c_0 + z^{K+1}$ or $c_0 z + z^{K+1}$. Choose the coefficient c_0 so that $W_{K+1}(z)$ has the proper phase.

Next, note that $W_{K+1}(z) + \sum_{k=1}^{K} c_k X_k(z)$ has the proper phase at ω_1, since each term in the sum has the proper phase by construction, and phase is closed (to addition by π) under linear combination. Note also that $\sum_{k=2}^{K} c_k X_k(z)$ has the proper phase at ω_2. Hence we can choose the coefficient c_1 so that $c_1 X_1(k) + W_{K+1}(z)$ has the proper phase at ω_2, and then the overall sum will have the proper phase at ω_2. Note the choice of c_1 is unique.

Next, note that $\sum_{k=3}^{K} c_k X_k(z)$ has the proper phase at ω_3. Hence we can choose the coefficient c_2 so that $c_1 X_1(z) + c_2 X_2(z) + W_{K+1}(z)$ has the proper phase at ω_3. Then the overall sum will have the proper phase at ω_3. Note the choice of c_2 is unique.

Continuing in this way, we successively compute c_k in increasing k, until $k = K$ is reached. The resulting sum has the proper phase at ω_{K+1}, so this sum is the solution to the magnitude retrieval problem for $k = K+1$. This can be continued until the ultimate problem at $K = N$ is reached.

4.3. Example

We present a small numerical example to illustrate this algorithm. We wish to compute the signal that has the following phases at the following frequencies:

$$\phi(\pi/2) = -1.107; \phi(\pi/3) = -0.606$$

These frequencies are *not* evenly spaced on $|z| = 1$.

We proceed as follows, replacing z with $\frac{1}{z}$.

1. Let $W_2(z) = c_0 z - z^2$. Then:
2. $ARG[W_2(e^{-j\frac{\pi}{2}})] = ARG[1 - jc_0] = -1.107 \rightarrow c_0 = 2$
3. $ARG[X_1(e^{-j\frac{\pi}{2}})] = ARG[1 - jx_1] = -1.107 \rightarrow x_1 = 2$
4. $ARG[(W_2 + c_1 X_1)] = ARG[(2z - z^2) + c_1(1+2z)]$
5. $= ARG[(2c_1 + 1.5) - j\sqrt{3}(c_1 + 0.5)] = -0.606 \rightarrow c_1 = 0.5$
6. $X_2(z) = (2z - z^2) + 0.5(1+2z) = 0.5 + 3z - z^2$

and indeed any multiple of the signal $[0.5, 3, -1]$ has the desired phases.

5. REFERENCES

[1] M.H. Hayes, J.S. Lim, and A.V. Oppenheim, "Signal Reconstruction from Phase or Magnitude," IEEE Trans. ASSP 28(6), 672-680, Dec. 1980.

[2] G.A. Merchant and T.W. Parks, "Reconstruction of Signals from Phase: Efficients Algorithms, Segmentation, and Generalizations," IEEE Trans. ASSP 31(5), 1135-1147, Oct. 1983.

[3] A.E. Yagle, "Fast and Recursive Algorithms for Magnitude Retrieval from Partial Phase Information at Irregular Frequencies," to be submitted to IEEE Trans. SP.

[4] E. Horowitz, *Fundamentals of Computer Algorithms,"* Computer Science Press, 1978.