In this final report we formulate the problems of phase and frequency estimation as fixed interval smoothing problems. Forward dynamic programming algorithms are derived to find maximum posterior sequence estimates that pass likely sequences through the data. The algorithms extend the threshold usually associated with phase lock loops. The algorithms are applied to simultaneous phase estimation and data decoding in phase jitter channels. The resulting error probabilities are the lowest currently achievable.
The results indicate that there are many problems in the domain of filtering and signal processing that can be profitably reformulated as sequence estimation problems.

Our interest in likelihood leads us from frequency tracking to exact likelihood for autoregressive moving average (ARMA) data. Fast Kalman filtering algorithms are derived for constructing exact likelihood in multivariable ARMA models. Several interesting connections are established between Wold, Kolmogorov, Wiener, and Kalman representations of stationary time series. Ideas are proposed for associating spectra with linear transformations.
Viterbi Tracking of Randomly Phase-Modulated Data

Final Report

Louis L. Scharf
University of Rhode Island
Kingston, RI 02881

August 5, 1982

Submitted to

U. S. ARMY RESEARCH OFFICE
P.O. Box 12211
Research Triangle Park, NC 27709

Contract/Grant Number
DAAG 29 79 C 0176

Colorado State University
Ft. Collins, Colorado 80523

Approved for Public Release;
Distribution unlimited.

The view, opinions, and/or findings contained in this report are
those of the author(s) and should not be construed as an official
Department of the Army position, policy, or decision, unless so
designated by other documentation.
Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the realm of signal processing. These problems have held the interest of control and communication theorists at least since 1953/54 when Lehan and Parks, and Youla published their work on maximum likelihood and optimum demodulation on an interval. Over the years Cox, Viterbi, Cahn, Forney, and a host of others have advocated dynamic programming for the solution of nonlinear filtering problems. This research follows that tradition.

Dynamic Programming is advocated as a technique for finding the maximum a posteriori (MAP) phase or frequency modulated sequence to pass through a data set. The key idea is to pose a Markov chain model on the circle ($0, 2\pi$) for phase or frequency, and then generate candidate MAP sequences that are consistent with the data and the a priori probability structure.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Statement of Problem Studied</td>
<td>2</td>
</tr>
<tr>
<td>Summary of Most Important Results</td>
<td>3</td>
</tr>
<tr>
<td>Publications Supported by This Grant</td>
<td>12</td>
</tr>
<tr>
<td>Scientific Personnel Earning Degrees While Employed on Project</td>
<td>14</td>
</tr>
<tr>
<td>Bibliography</td>
<td>15</td>
</tr>
<tr>
<td>Appendixes</td>
<td>16</td>
</tr>
</tbody>
</table>
LIST OF APPENDIXES

APPENDIX A: Selected Reprints of Major Publications

APPENDIX B: Progress Reports and Miscellaneous Documents

APPENDIX C: Army Sponsored Meetings Attended by Principal Investigator
Phase and Frequency tracking are the classic nonlinear filtering problems. They arise in narrowband analog communication, data transmission, and spread spectrum communication. As usually stated, the problem is to obtain a causal estimate of the phase or frequency based on noisy phase modulated observations. The best known solutions are phase-locked loops (PLL's).

In any truly nonlinear filtering approach to optimum phase tracking, the basic problem is to propagate an a posteriori density, conditioned on an increasing measurement record, much as is done in Kalman Filtering. Unfortunately, there exist no finite-dimensional schemes for propagating the exact conditional density or for propagating a finite-dimensional sufficient statistic. One must approximate.

Under this contract we have developed an approach to phase and frequency sequence estimation (emphasis on the word sequence) that has its logical antecedents in the filtering philosophy of Youla and the data decoding philosophy of Viterbi. We have posed a maximum a posteriori probability (MAP) sequence estimation problem that leads to nonlinear MAP equations not unlike the continuous-time MAP interval equations. We have derived dynamic programming algorithms to efficiently solve for survivor phase and frequency sequences that approximate the desired MAP sequence. The algorithms also provide a handy mechanism for generating fixed-lag phase estimates, although this is not the problem for which the algorithm is derived.

In a loosely related set of problems we have studied exact likelihood for autoregressive moving average (ARMA) processes. We have derived fast algorithms for constructing likelihood, and established interesting connections between the work of Wold, Kolmogorov, Wiener, and Kalman. A fast Kalman filter has been realized in 16-bit arithmetic on an 8086 microprocessor.

In the sections to follow, we outline the problems studied and summarize important results.
STATEMENT OF PROBLEMS STUDIED

We summarize here the main problems studied under this contract.

**Phase Modelling.** For phase and frequency tracking the first problem to be studied is one of deriving suitable models for random phase and/or random frequency modulation.

**Dynamic Programming Algorithm Development.** Once a phase model is derived, the next problem is to derive a likelihood function and find a dynamic programming algorithm to find the maximum of likelihood.

**Performance Evaluation for Phase and Frequency Sequence Estimation.** The next problem is to simulate the dynamic programming algorithms on stochastic data and calculate Monte-Carlo performance results.

**Simultaneous Phase Tracking and Data Decoding.** When complex data are transmitted over phase jitter channels, there arises the problem of simultaneously tracking phase and decoding data symbols. The problem is to derive a joint likelihood function for phase and symbol sequences, maximize it with a dynamic program algorithm, and compute Monte-Carlo performance results.

**Maximum Likelihood Identification of ARMA Systems.** The problem here is to derive a fast algorithm to compute likelihood for autoregressive moving average (ARMA) sequences.

**Fixed Point Implementation of Kalman Filters.** The fast Kalman gain algorithm is a fixed point algorithm ideally suited for computation on a fixed point machine. But the problems of scaling and rounding remain. The question here is one of deriving scaling rules and calculating rounding error variances in time varying Kalman filters.
SUMMARY OF MOST IMPORTANT RESULTS

The most important results of this study are summarized below.

Phase Modelling. We have derived phase models for random phase, random FM, and random chirp modulation. Each model is a Markov chain defined on a cyclic group. Corresponding correlation and spectral results have been derived. The results generalize existing results on the spectral theory of chains, and leave one with the problem of selecting states, transition probabilities, and run lengths to achieve model matching with more conventional models. The results apply for coherent and noncoherent FM. The figure on the following page gives a geometric picture of the kinds of phase and frequency models we have used in most of our work on algorithm development, phase and frequency tracking, and simultaneous phase tracking and data decoding. See references 1, 2, and 3 for additional details.

Dynamic Programming Algorithm Development. The basic measurement model in all of our work has been the following:

\[ z_t = a_t e^{j\phi_t} + n_t \]

- \( a_t \): symbol drawn from a finite alphabet
- \( \phi_t \): either a directly modulated phase sequence for which we know the transition probability density \( p(\phi_{t+1}/\phi_t) \) or a function \( \phi(\omega_t) \) of a frequency sequence \( \omega_t \) for which we know the transition probability density \( p(\omega_{t+1}/\omega_t) \)
- \( n_t \): a sequence of independent and identically distributed normal random variables

With this model we have derived expressions for likelihood and found dynamic programming algorithms for exactly maximizing or approximately maximizing likelihood. Generally the algorithms take the form:

\[
\max_{\phi_K, \phi_{K-1}, \ldots, \phi_2} \max_{\phi_1} \left[ L_{K-1} + \ln p(\phi_K/\phi_{K-1}) + g(\phi_K) \right]
\]

The function \( g(.) \) depends on the details of the problem. See references 1, 2, and 3 for details about selecting \( g(.) \) and implementing the algorithm on a finite trellis. The function \( L \) is likelihood.

It is our opinion that a variety of filtering problems in signal and image processing can be reformulated as sequence or interval estimation problems for which likelihood can be derived and for which algorithms can be found for approximating the maximum.
\[ z_t = e^{j\phi_t} + n_t \]
\[ Z_t \rightarrow (\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_K) \]

Constant Phase

Random Walk Phase

Constant Frequency

Random Walk Frequency

PHASE AND FREQUENCY MODELS
Performance Evaluation for Phase and Frequency Sequence Estimation. Our results are nicely summarized on the graphs of the following pages. The first compares estimation error variance for the dynamic programming (or Viterbi) solution with a host of other algorithms ranging from the phase lock loop to the point mass filter and the Fourier coefficient filter. The results apply to the problem of random walk phase tracking.

The next graph shows output SNR versus input CNR for sinusoidal modulation of a carrier. We have adapted our random walk FM frequency tracker to this problem and compared its performance with linear prediction trackers, and the trackers of Tufts and of Toomey and Short.

Our performance results indicate that sequence estimation by the method of dynamic programming to maximize likelihood on a finite trellis provides a way of improving on the performance of more classical causal estimators. This improvement can be significant at low SNR.

Simultaneous Phase Tracking and Data Decoding. The performance results for this problem are contained in Reference 2, where a variety of binary, phase shift keying, and quadrature shift keying communication problems are considered. The third figure in the sequence of three figures that appears on the next three pages shows just one of the many examples contained in Reference 2. The graph shows how two simultaneous phase trackers and data decoders, namely the Viterbi tracker and the jitter equalizer, achieve performance very close to that achievable under coherent phase conditions. The results apply to the decoding of 8-ary phase shift keyed symbols.
COMPARATIVE PERFORMANCE RESULTS FOR RANDOM PHASE TRACKING
COMPARATIVE PERFORMANCE RESULTS FOR SINUSOIDAL FM TRACKING

\[ SNR = 10\log_{10} \frac{A^2}{\sum_{i} A_i^2 - A_m^2} \]
PERFORMANCE RESULTS FOR SIMULTANEOUS PHASE TRACKING AND DATA DECODING
Maximum Likelihood Identification of ARMA systems. We have followed the lead of Akaike and Anderson and Moore to write down the innovations representation that reproduces the second order statistics of a stationary ARMA sequence. We have then associated the gain of a Kalman filter with the triangular square root of a Toeplitz matrix to rederive Morf's fast Kalman filter algorithm. The result is a fast algorithm for implementing likelihood. The results are summarized in References 4 and 5.

Fixed Point Implementation of Kalman Filters. Beginning with the innovations representation of a stationary ARMA sequence, we have derived scaling rules to prevent overflow in time varying Kalman filters and derived formulas for rounding error variance. The scaling rule is

\[
\frac{q(k,k)^{1/2}}{s(k)} = \frac{\epsilon 2^{m-1}}{\delta}
\]

\(s(k): \) inverse of time varying scale constant
\(q(k,k)^{1/2}: (k,k)\)th element of the state variance matrix
\(\epsilon 2^{m-1}: \) dynamic range of the fixed point representation
\(\delta : \) design parameter that allows designer to control the probability of overflow

This formula generalizes the results of Mullis and Roberts to time varying cases.

The figure on the following page illustrates our experimental setup for implementing the Kalman filter on an 8086 microprocessor. The figure on the next page shows a typical simulation showing performance on the fixed point machine with that achievable on a floating point machine. The results apply to one-step prediction. The circles highlight places where the fixed point and floating point results differ by more than 1 bit in 6.
EXPERIMENTAL SET UP
PREDICTIONS USING FLOATING POINT AND FIXED POINT ARITHMETIC
PUBLICATIONS SUPPORTED BY THIS GRANT

Papers and Conference Presentations


Technical Reports


SCIENTIFIC PERSONNEL EARNING ADVANCED DEGREES WHILE EMPLOYED ON PROJECT

Of the graduate students supported in full or in part under this grant, one has earned a Ph.D. and four have earned M.S. Degrees.

Ph.D.


M.S.


BIBLIOGRAPHY


This appendix contains selected reprints of papers published with ARO sponsorship under DAAG 29 79 C 0176.
Modulo-2\(w\) Phase Sequence Estimation
LOUIS L. SCHARF, SENIOR MEMBER, IEEE, DENNIS D. COX, AND C. JOHAN MASRELIEZ, MEMBER, IEEE

Abstract—The probabilistic evolution of random walk on the circle is studied, and the results are used to derive a maximum a posteriori probability (MAP) sequence estimator for phase. The sequence estimator is a Viterbi tracker for tracking phase on a finite-dimensional grid in \([-\pi, \pi]\). The algorithm is shown to provide a convenient method for obtaining fixed-lag phase estimates. Performance characteristics are presented and compared with several published nonlinear filtering algorithms.

I. INTRODUCTION

Phase tracking is the classic nonlinear filtering problem. It arises in narrowband analog communication, data transmission, and spread spectrum communication. As usually stated, the problem is to obtain a causal estimate of the phase based on noisy phase-modulated observations. The best known solutions are phase-locked loops (PLL's).

In any truly nonlinear filtering approach to optimum phase tracking, the basic problem is to propagate the a posteriori density of the phase, conditioned on an increasing measurement record, much as is done in Kalman filtering. Unfortunately, there exist no finite-dimensional schemes for propagating the exact conditional density or for propagating a finite-dimensional sufficient statistic. One must approximate. The interested reader may consult [18] for a review of the best known techniques or, better yet, go directly to the appropriate source [1]-[13].

In this correspondence we propose an approach to phase sequence estimation (emphasis on the word sequence) that has its logical antecedents in the filtering philosophy of Youla [2] and the data decoding philosophy of Viterbi [14]. We pose a maximum a posteriori probability (MAP) sequence estimation problem that leads to nonlinear MAP equations not unlike the continuous-time MAP interval equations. Fortunately there exists a dynamic programming algorithm to efficiently solve for survivor phase sequences that approximate the desired MAP sequence. The algorithm also provides a handy mechanism for generating fixed-lag phase estimates, although this is not the problem for which the algorithm is derived. As is common in most of the current communications literature we call our dynamic programming algorithm a Viterbi algorithm.

Cahn [15] has suggested that phase may be tracked with delay in order to extend the so-called threshold. He proposes a Viterbi-like algorithm for tracking carrier phase sequences whose realizations satisfy dynamics constraints. There is certainly a philosophical link between Cahn's work and ours. In fact it was Cahn's paper that first aroused our interest in phase sequence estimation. However the approaches are really quite different. Ungerboeck [16] has proposed an algorithm for phase tracking that makes use of a delta-modulation approximation to the phase sequence and an approximate version of the Viterbi algorithm. Tufis and Francis [17] have also recently proposed an algorithm for obtaining smoothed phase estimates.

II. THE BASIC PROBLEM

Let \(\{Z_k\}\) denote the complex observation sequence

\[
Z_k = e^{\theta_k} + N_k, \quad k = 1, 2, \ldots
\]

where

\[
N_k = U_k + jV_k, \quad N_k \perp \perp N_l \text{ for } k \neq l,
\]

\[
U_k : \mathbb{R} \left(0, \sigma_u^2\right), \quad V_k : \mathbb{R} \left(0, \sigma_v^2\right),
\]

\[
U_k \perp \perp V_l \text{ for all } k, l.
\]

(1)

\((\Phi_k)\) is a discrete-time phase sequence to be discussed shortly and \((N_k)\) is an additive noise sequence of independent identically distributed (i.i.d.) normal random variables. Our notation is that \(N_1 \perp \perp N_l\) means \(N_1\) and \(N_l\) are independent, and \(U_k : \mathbb{R}\left(0, \sigma_u^2\right)\) indicates that \(U_k\) is a normal random variable with mean 0 and variance \(\sigma_u^2\). The sequence \((Z_k)\) may be thought of as a complex representation for the sample values appearing at the output of a quadrature demodulator. The problem is to estimate a realization of the entire sequence \((\Phi_k)\), say \(\{\hat{\Phi}_k\}\), from the measurement record \((Z_k)\). It turns out that this formulation also provides a convenient way to generate a sequence of fixed-lag estimates. However, we emphasize that the basic problem under investigation is one of estimating an entire sequence, not one of generating a sequence of fixed-lag, fixed-interval, or fixed-point smoothing solutions. We make the obvious but important observation that the signal model (1) is invariant under a modulo-2\(w\) transformation on the phase.

III. RANDOM WALK ON THE CIRCLE AS A MODEL FOR PHASE NOISE

The first, seemingly natural, choice for a random phase model is the Wiener process \(W(t)\) with incremental variance \(\sigma^2\). This is the most commonly used model for random phase acquisition. Most of the results in this correspondence may be obtained in a formal way using this phase model, but certain technical difficulties arise. First, there is no stationary distribution and, second, there is no rigorous way of defining a unique conditional probability for transitions from a modulo-2\(w\) value of \(W(t)\) to another modulo-2\(w\) value at a later time \(t + \tau\). The latter difficulty is particularly troublesome as one of the crucial parts of our modulo-2\(w\) phase sequence estimator is a transition probability matrix that characterizes phase transitions between modulo-2\(w\) values. By modeling phase as a random walk on the circle we avoid these technical difficulties. Other authors (see for example [7]) have also noted that the circle is the appropriate domain on which to study modulo-2\(w\) type sequences.

Let \(\Phi(t)\) be a random walk on the circle, taking values in \([-\pi, \pi]\). Denote by the function \(P(\phi|\phi_k)\) the conditional density of a transition from the value \(\Phi(t) = \phi\) at time \(s\) to the value \(\Phi(t) = \phi\) at time \(t > s\). This conditional density satisfies the
partial differential equation [20]
\[
\frac{\partial^2}{\partial t^2} \rho(\phi_1/\phi_0) - \frac{1}{2} \sigma_0^2 \frac{\partial}{\partial \phi_0} \rho(\phi_1/\phi_0) = 0
\]
where $\sigma_0^2$ is the infinitesimal variance. This equation holds in the strip $-\pi < \phi_1 < \phi_2$, for any fixed $-\pi < \phi_2 < \pi$. The boundary conditions are
\[
\lim_{t \to \pm \infty} \rho(\phi_1/\phi_0) = \delta(\phi_1 - \phi_0),
\]
where $\delta$ is the Dirac delta function. It is commonly the case that
\[
\rho(\phi_1 - \pi/2) = \rho(\phi_1 + \pi/2) = 0,
\]
and the notation $\rho(\phi_1 - \pi/2)$ is used to denote the function $\rho(\phi_1 + \pi/2)$. The notation $\rho(\phi_1 - \pi/2)$ is used to denote the function $\rho(\phi_1 + \pi/2)$.

The solution for $\rho(\phi_1/\phi_0)$ is
\[
\rho(\phi_1/\phi_0) = \frac{1}{2\pi \sigma_0^2 (t - s)} \exp \left\{ -\frac{1}{4\sigma_0^2 (t - s)} \left[ (\phi_1 - \phi_0 - n\pi)^2 \right] \right\}. \tag{4}
\]
It is easily seen that the process $\Phi(t)$ is conditionally approximately Gaussian $\mathcal{N}(\phi_0, \sigma_0^2(t - s))$, given $\Phi(t) = \phi_0$, for small $t - s$. An eigen-function expansion of the following form is also useful:
\[
\rho(\phi_1/\phi_0) = \frac{1}{2\pi \sigma_0^2 (t - s)} \exp \left\{ -\frac{1}{4\sigma_0^2 (t - s)} \left[ (\phi_1 - \phi_0 - n\pi)^2 \right] \right\}. \tag{5}
\]
This is simply Poisson's summation formula for (4). From this expression it is clear that $\Phi(t)$ becomes uniformly distributed as $t \to \pm \infty$. Equation (5) has also been noted in [7].

Consider the discrete-time sequence $\{\Phi_k\}$ obtained by sampling $\Phi(t)$ at the periodic sampling instants $t = kT$, $k = 0, 1, \cdots$. Call $\Phi_k$ a realization of $\Phi$. The transition density from $\Phi_{k-1}$ to $\Phi_k$ is found from (4) with $\sigma_0^2 = \sigma_T^2 T$ to be
\[
\rho(\Phi_k/\Phi_{k-1}) = \frac{1}{2\pi \sigma_0^2} \exp \left\{ -\frac{1}{2\sigma_0^2} \left[ (\Phi_k - \Phi_{k-1} - n\pi)^2 \right] \right\}. \tag{6}
\]
By the Markov property of $\Phi(t)$ it follows that $\{\Phi_k\}$ is a Markov sequence for which the joint distribution of $\{\Phi_k\}^M_1$ may be written
\[
\rho((\Phi_k)_k^{M_1}) = \prod_{k=1}^{M_1} \rho(\Phi_k/\Phi_{k-1}). \tag{7}
\]
where
\[
\rho(\phi_1/\phi_0): U[-\pi, \pi],
\]
and the notation $\rho(\phi_1/\phi_0): U[-\pi, \pi]$ indicates that $\Phi(t)$ is uniformly distributed on $[-\pi, \pi]$. Other choices are also admissible: for example, $\rho(\phi_1/\phi_0) = \delta(\phi_1 - \phi_0)$ with $\phi_0$ known corresponds to a given initial phase.

One may obtain the same discrete-time model for $\{\Phi_k\}$ by considering $\Phi_k$ to be a modulo-$2\pi$ version of the following discrete-time random walk:
\[
\Phi_k = \Phi_{k-1} + W_k, \quad W_k \sim \omega, \quad k = 1, \ldots, M_1.
\]

The modulo-$2\pi$ version of $\Theta_1$, call it $\Theta_1$, may be written
\[
\Theta_1 = \Theta_{11} + W_1, \quad \Theta_1 = \Theta_{11} + W_1.
\]
Given $\Theta_{11} = \Theta_{11}^0$, the random variable $\Theta_1$ is $\mathcal{N}(\Theta_{11}^0, \sigma_{11}^2)$. As $\Theta_1$ is a modulo-$2\pi$ version of $\Theta_1$, it follows that the conditional density of $\Theta_1$, given $\Theta_{11} = \Theta_{11}^0$, is the folded normal density of (6). For this reason we will often call the discrete-time process on the circle $\{\Phi_k\}$ a modulo-$2\pi$ version of the discrete-time random walk $\{\Theta_k\}$.

In Section VII we discretize the phase space $[-\pi, \pi]$ to phase values $\xi_m, m = 0, 1, \cdots, M - 1$ with $M$ odd. It is then necessary to characterize the transition probability from $\xi_m$ to $\xi_{m+1}$ for all $M^2$ pairs $(\xi_m, \xi_{m+1})$. We choose for our definition of this transition probability
\[
\rho(\phi_1 = \xi_m/\phi_{k-1} = \xi_{m+1}) = b, \quad b = \rho(\phi_1 = \xi_{m+1}/\phi_{k-1} = \xi_m) \tag{10}
\]
with $b_2$ selected so that
\[
\sum_{m=0}^{M-1} \rho(\phi_1 = \xi_m/\phi_{k-1} = \xi_{m+1}) = 1, \quad l = 0, 1, \cdots, M - 1. \tag{11}
\]
The sum on $n$ in (6) must, of course, be truncated. This truncation may be selected to give the desired accuracy before the algorithm of Section VII is run. There is no series truncation whatsoever in the algorithm itself.

There is an important symmetry property of (10). If the $\xi_i$ are equally spaced points on $[-\pi, \pi]$, for example $\xi_i = 2\pi i / (M - 1)$, then the function $\rho$ depends only on $\xi_{m+1} - \xi_m$. Thus if the values of (10) are organized into an $M \times M$ matrix of transition probabilities, the matrix is Toeplitz. We may compute the $M$-dimensional vector $\hat{Q} = (\hat{Q}_0, \hat{Q}_1, \cdots, \hat{Q}_{M-1})$ with $\hat{Q}_{m} = \rho(\phi_1 = \xi_{m+1}/\phi_{k-1} = \xi_m)$ as $\hat{Q}_m$ with $m = [M/2]$. In this way only an $M$-vector of transition probabilities need be stored for cyclic reading.

IV. THE MAP SEQUENCE ESTIMATION PROBLEM FOR MODULO-$2\pi$ PHASE

Consider the following maximization with respect to the modulo-$2\pi$ phase sequence $\{\Phi_k\}$:
\[
\max_{\{\Phi_k\}} \rho((\xi_k), \{\Phi_k\}). \tag{12}
\]
where $\rho(\cdot, \cdot)$ is the joint density function for the $K$ measurements $\{\xi_k\}$ and the $K$ modulo-$2\pi$ phase values $\{\Phi_k\}$. Maximization of this joint density function is equivalent to maximization of the posterior density $\rho((\xi_k), \{\Phi_k\})$. The joint density in (12) may be written
\[
\rho((\xi_k), \{\Phi_k\}) = \rho((\xi_k), \{\Phi_k\}) \prod_{k=1}^{K} \Theta_{nk}(\xi_m, \sigma_{m}^2). \tag{13}
\]
where the last line follows since the $\Theta_{nk}$ in (1) are i.i.d. normal random variables and because, conditionally,
\[
\rho(\xi_k/\Phi_k): \mathcal{N}(\xi_k, \sigma_{m}^2). \tag{14}
\]
Here $\Theta_{nk}(\xi_m, \sigma_{m}^2)$ indicates that the conditional density of the complex random variable $\Theta_k$ (conditioned on $\Phi_k$) is normal with mean $e^{i\xi_k}$ and variance $\sigma_{m}^2$:
\[
N(\xi_k, \sigma_{m}^2) = \frac{1}{2\pi \sigma_{m}^2} \left\{ 1 - \frac{1}{2\sigma_{m}^2} |z_k - e^{i\xi_k}|^2 \right\}. \tag{15}
\]
Dropping phase independent terms we may write the MAP sequence estimation problem as
\[
\max_{\{\Phi_k\}} \rho((\xi_k), \{\Phi_k\}). \tag{16}
\]

Here $\mathcal{N}(\xi_k, \sigma_{m}^2)$ indicates that the conditional density of the complex random variable $\Theta_k$ (conditioned on $\Phi_k$) is normal with mean $e^{i\xi_k}$ and variance $\sigma_{m}^2$:
\[
\max_{\{\Phi_k\}} \rho((\xi_k), \{\Phi_k\}). \tag{17}
\]
where

\[ l_k = \frac{1}{\alpha_k^2} \text{Re} \sum_{k=1}^{K} \Delta_k + \sum_{k=1}^{K} \ln p(\phi_k/\psi_k). \]  

(16)

\( \Lambda_k \) is the phase-corrected vector

\[ \Lambda_k = \sum_{k=1}^{K} c_k e^{i\phi_k - \alpha_k} = \sum_{k=1}^{K} z_k e^{-j\alpha_k}, \]  

(17)

and \( c_k \) and \( \psi_k \) are, respectively, envelope and phase variables: \( z_k = c_k e^{j\phi_k}, \psi_k \in [-\pi, \pi], c_k \in [0, \infty) \). It is clear from this form that the MAP phase sequence will be one that stays reasonably close to the noisy phase variables \( \phi_k \) (to make \( \cos(\psi_k - \phi_k) \) large) while also maintaining a trajectory that is \( a \) priori reasonable likely. Thus the MAP sequence strikes a balance between what the noisy data \( \psi_k \) says the phase is doing and what the transition probabilities \( p(\alpha_k/\alpha_{k-1}) \) say the phase can do. When the envelope \( c_k \) is large there is more of a tendency to believe the measured \( \psi_k \). This curious effect may be explained by noting that the phase statistic \( \psi_k \) is a modulo-2\( \pi \) unbiased estimate of \( \phi_k \) with a variance that decreases approximately inversely with increasing \( c_k \) [18].

V. CHARACTERISTICS OF THE MAP SEQUENCE

Given the envelope and phase variables \( (c_k) \) and \( (\psi_k) \), the MAP phase sequence \( (\hat{\phi}_k) \) may be obtained by equating the derivatives of \( \Gamma_k \) to zero:

\[ \frac{\partial}{\partial \phi_k} \ln p(\hat{\phi}_k/\phi_k) + \frac{\partial}{\partial \phi_k} \ln p(\hat{\phi}_{k+1}/\phi_k) + \frac{1}{\alpha_k^2} f_m(\Lambda_k - \Lambda_{k-1}) = 0, \quad k = 1, 2, \ldots, K. \]  

(18)

The boundary conditions are \( \Gamma_0 = 0 \) and

1) \[ \frac{\partial}{\partial \phi_k} \ln p(\hat{\phi}_k/\phi_k) = 0, \]

2) \[ \frac{\partial}{\partial \phi_k} \ln p(\hat{\phi}_{k+1}/\phi_k) = 0. \]

(19)

Condition 1) simply reflects the fact that \( p(\phi_k/\phi_k) \) is uniform on \([-\pi, \pi]\). Condition 2) is a mathematical convenience that allows us to put all the equations of (18) in the same form. Of course \( \phi_k \) and \( \hat{\phi}_{k+1} \) are not computed from the data \( (c_k) \) and \( (\psi_k) \).

Equations (18) are nonlinear equations with two-point boundary conditions. They are analogous to the time-continuous MAP equations obtained for phase tracking on an interval. While we cannot solve the equations of (18) explicitly we can make some very interesting observations regarding the properties of the MAP phase sequence.

It is easily verified from the conditional density of (6) that

\[ \frac{\partial}{\partial \phi_k} \ln p(\phi_{k+1}/\phi_k) = - \frac{\partial}{\partial \phi_{k+1}} \ln p(\phi_{k+1}/\phi_k). \]  

(20)

Therefore when the \( K \) equations of (18) are summed and the boundary conditions applied, all terms involving \( \ln p(\phi_{k+1}/\phi_k) \) cancel. The sum on the terms involving \( \operatorname{Im}(\Lambda_k - \Lambda_{k-1}) \) telescopes, and we are left with the result

\[ \operatorname{Im} \dot{\Lambda}_k = 0. \]  

(21)

Here \( \Lambda_k \) is \( \Lambda_k \) with \( \phi_k \) set to the MAP estimate \( \hat{\phi}_k \) for \( k = 1, 2, \ldots, K \). This allows us to make the following observation: while maximizing the objective function \( \Gamma_k \), the MAP sequence \( (\hat{\phi}_k) \) yields a maximum value for \( \Gamma_k \) of

\[ \Gamma_k = \frac{1}{\alpha_k^2} \text{Re} \sum_{k=1}^{K} \ln p(\hat{\phi}_k/\phi_k). \]  

(22)

with the property that \( \operatorname{Im} \dot{\Lambda}_k = 0 \). This property is illustrated in Fig. 1. We note that there are many other sequences that satisfy the condition \( \operatorname{Im} \dot{\Lambda}_k = 0 \) (e.g., the sequence of maximum likelihood estimates \( \hat{\phi}_k = \psi_k \), but these sequences do not also maximize \( \Gamma_k \).

VI. THE MAP SEQUENCE FOR FIXED PHASE ACQUISITION

Suppose the underlying phase sequence \( (\phi_k) \) is known to be a constant sequence with the value of the constant uniformly distributed on \([-\pi, \pi]\). In this case the MAP sequence estimate is identical with the maximum likelihood (ML) estimate of an unknown phase parameter \( \phi \) in a complex normal model. For this reason, and for the insight it gives into phase estimation, we include in the following paragraphs a short discussion of constant phase and envelope models. The inclusion of an unknown envelope \( c \) generalizes the discussion without changing the nature of the phase estimate. This follows from the fact that the phase estimate is uncoupled from the envelope estimate. The converse is not true.

Consider the joint density function for the data \( (z_k) \), parameterized by the envelope \( c \) and the phase \( \phi \):

\[ g((z_k)) = \frac{1}{(2\pi\sigma_c^2)^K} \exp \left\{ -\frac{1}{2\sigma_c^2} \sum_{k=1}^{K} |z_k - ce^{j\phi}|^2 \right\}. \]  

(23)

where

\[ d(\phi, c) h((z_k)) \exp \left\{ -\frac{1}{2\sigma_c^2} \sum_{k=1}^{K} |z_k|^2 \right\}. \]  

(24)

It follows from the factorization theorem [21, p. 115] that the complex statistic \( K^{-1} \sum_{k=1}^{K} z_k = c e^{j\phi} \) is

\[ d = K^{-1} \sum_{k=1}^{K} z_k. \]  

(25)

This ML estimator is consistent, unbiased, efficient, and minimum variance unbiased. The corresponding ML estimates for \( c \) and \( \phi \) are

\[ c = \arg \sum_{k=1}^{K} z_k. \]  

(26)

\[ \phi = \arg \sum_{k=1}^{K} z_k. \]  

(27)
Let $\hat{c}$ and $\hat{\phi}$ be the estimators corresponding to the estimates $\hat{c}$ and $\hat{\phi}$. The estimator $\hat{c}$ is consistent, unbiased, efficient, and minimum variance unbiased. The phase estimator $\hat{\phi}$ is not efficient and no efficient estimator exists. It is consistent but biased. However it is modulo-$2\pi$ unbiased, which is the property we want. The phase estimate $\hat{\phi}$, also obtained in [8] and [17], in different ways, is illustrated in Fig. 2.

Define the modulo-$2\pi$ estimator error $\Delta \hat{\phi} = (\hat{\phi} - \phi) \mod 2\pi$. We may write

$$
K^{-1} \sum_{k=1}^{K} Z_k e^{-i\hat{\phi}_k} = C_k \hat{\phi} - \hat{c} = C_k \hat{\phi} + \hat{\phi} - \hat{c}.
$$

(28)

The statistic $K^{-1} \sum_{k=1}^{K} Z_k e^{-i\phi}$ is $\mathcal{N}(c, \sigma^2/K)$. The Jacobian of the transformation between $(c, \Delta \hat{\phi})$ and $K^{-1} \sum_{k=1}^{K} Z_k e^{-i\phi}$ is $C$. Therefore the joint density of $C$ and $\Delta \hat{\phi}$ is

$$
g(c, \Delta \hat{\phi}) = \frac{c}{2\pi \sigma^2} \exp \left( - \frac{1}{2\pi \sigma^2} \left( |c - e^{i\Delta \hat{\phi}}| \right)^2 \right).
$$

(29)

This result is equivalent to [22, eq. (9.46), p. 413] with appropriate change of notation. In (29) it is assumed that $-\pi < \Delta \hat{\phi} < \pi$. On this interval $g(c, \Delta \hat{\phi})$ is symmetrical about zero and therefore unbiased. We emphasize that $\phi$ is only modulo-$2\pi$ unbiased.

VII. The Viterbi Algorithm

The MAP sequence estimation problem is stated in (18). Note that $\Gamma_k$ satisfies the recursion

$$
\Gamma_k = \Gamma_{k-1} + \frac{1}{s_k} c_k \cos(\phi_k - \hat{\phi}_k) + \ln p(\phi_k | \phi_{k-1}).
$$

(30)

The so-called path metric is

$$
\frac{1}{s_k} c_k \cos(\psi_k - \hat{\phi}_k) + \ln p(\phi_k | \phi_{k-1}).
$$

(31)

The maximization problem for obtaining the MAP phase sequence may now be written

$$
\max_{(\hat{\phi}_k)} \left[ \max_{(\hat{\phi}_{k-1})} \Gamma_{k-1} + \ln p(\phi_k | \phi_{k-1}) + \frac{1}{s_k} c_k \cos(\psi_k - \phi_k) \right].
$$

(32)

This form leads to the following observation: the maximizing trajectory (call it $(\hat{\phi}_k)$), passing through $\hat{\phi}_k$ on its way to $\hat{\phi}_{k-1}$, must arrive at $\hat{\phi}_k$ along a route $(\phi_k)$ which maximizes $\Gamma_k$.

For if it did not we could retain $\hat{\phi}_{k-1}$ and $\hat{\phi}_k$ and replace $(\phi_k)$ with a different sequence to get a larger value for $\Gamma_k$. It is this observation which forms the basis of forward dynamic problem.

The trellis of Fig. 3 illustrates how the maximization of (32) proceeds. Tabulated values of $p(\phi_k | \phi_{k-1})$ are stored in a square array (or vector which is read cyclically) whose dimensions depend upon how finely the interval $[-\pi, \pi]$ is discretized. Let $\Omega = \{(\xi_k)\}_{k=0}^\infty$, be the finite-dimensional grid for which $p(\phi_k | \phi_{k-1})$ is defined. That is, $\phi_k$ is assumed to take on only the values $\phi_k = \xi_k$, $k = 1, 2, \ldots, M$, for each $k$. Let $\Gamma_k(\xi_k, \psi_k)$ be the value of the metric $\Gamma_k$ corresponding to a phase trajectory $(\phi_k)$ which terminates at phase-state $\psi_k$ at stage $k$, after passing through stage $\xi_k$ at stage $k-1$.

The algorithm begins with a computation of $\Gamma(\xi_0, \psi_0)$, $l = 1, 2, \ldots, M$ based on measured values of $c_1$ and $\phi_1$. If all phase values are equally likely a priori, then $p(\phi_1 | \phi_0)$ is constant on all values $\xi_0$. Otherwise there is some a priori weighting in favor of some of the $\xi_0$. A new measurement pair $(c_2, \phi_2)$ is obtained at $k=2$ and $\Gamma_2(\xi_2, \psi_2)$ is computed for $m=1, 2, \ldots, M$ using a table look-up (for example in a read-only memory) for the $(\phi_1 = \xi_2, \psi_2 = \phi_2)$. The maximum value of $\Gamma_2(\xi_2, \psi_2)$ is determined (over all originating series $\xi_2$) and the corresponding sequence $(\xi_2, \psi_2)$ is saved as a survivor sequence terminating at $\psi_2$ at stage 2; $\xi_1$ denotes the originating state. The survivor sequence is labeled with its corresponding length $l_2 = l_1 + 1$. This calculation is repeated for each possible value of phase until all pairs $(\xi_l, \xi_{l+1})$ and corresponding lengths $l_2 = l_1 + 1, 2, \ldots, M$, have been computed and stored (for example in a random access memory). There is a unique survivor sequence corresponding to each state $\xi_l$, $l = 1, 2, \ldots, M$. Caution: In the pair $(\xi_l, \xi_{l+1})$ the originating state $\xi_l$ depends on $\xi_l$; i.e., $\xi_l = \xi_{l+1}$. The measurements $c_l$ and $\psi_l$ may now be discarded along with all extinct sequences: A new measurement pair $(c_{l+1}, \phi_{l+1})$ is now obtained and the procedure continues.

Let $(\phi_k(k), \psi_k(k), \phi_{k+1}(k), \psi_{k+1}(k))$ be the MAP sequence based on $k$ measurements; this sequence has the maximum value of $\Gamma_k$. The parenthetical notation $(k)$ denotes dependence on measurement-estimation interval. In general the MAP sequence estimate $(\phi_k(k + 1), \psi_k(k + 1))$ based on measurements up to stage $k+1$ may differ from the previous sequence estimate at every stage from 1 to $k$. However, as a practical matter, one can choose a sufficiently large depth parameter $N_k$ so that the sequence of fixed-lag estimates

$$
\hat{\phi}_k N_k(k) = k_0 + 1, k_0 + 2, \ldots
$$

(33)

gives an approximate MAP sequence estimate. Here $k_0(k)$ is simply the phase value $k_0$ at stage $k$ in the MAP sequence.
estimate based on $k$ measurements. In this way one obtains a phase track with delay $k_p$.

Following Forney [23] we may summarize the storage and computational requirements for the phase tracking algorithm as follows:

Storage

$d_k, c_{i,k-1}, \ldots, c_1, c_0, b_1, b_0, b_{-1}, \ldots, b_{-M}, a_1, a_0, a_{-1}, \ldots, a_{-M+1}$

(time index).

$\Gamma_k(\xi_k, \xi_{k-1})$ (survivor phase sequence terminating in $\xi_k$ at stage $k$).

$\rho(\xi_k, \xi_{k-1})$, $1 \leq m \leq M$ (survivor metric).

$\Delta_k(\xi_k, \xi_{k-1})$, $1 \leq m \leq M$ (transition probability matrix).

Initialization

$k = 1, \xi_1 = \xi_1, \xi_2 = \xi_2, \ldots, \xi_k = \xi_k, \ldots, \xi_M = \xi_M$.

$\Gamma_1(\xi_1, \xi_1) = \rho(\xi_1, \xi_1) + \min \rho(\xi_{1,k-2} = \xi_{1,k-2}, \ldots, \xi_{1,k-2} = \xi_{1,k-2})$.

Recursion

$\Gamma_{k+1}(\xi_{k+1}, \xi_{k+1}) = \Gamma_k(\xi_k, \xi_{k+1}) + \min \rho(\xi_{k+1}, \xi_{k+1})$,

$\quad + 2 \frac{1}{2a_2} \cos(\theta_{k+1} - \xi_k), 1 \leq k \leq M$.

Measurement/Computation

$c_k$ (envelope).

$b_{k-1} \cos(\theta - \xi_k) + \min \rho(b_{k-1} \cos(\theta - \xi_k), \ldots, b_{k-2} \cos(\theta - \xi_k))$ (phase).

VIII. PERFORMANCE RESULTS

The phase space $[-\pi, \pi]$ has been divided into $M = 11$ equally spaced points and the Viterbi algorithm for phase tracking implemented as outlined in Section VII. The crucial conditional probabilities $\rho(\xi_k = \xi_k|\xi_{k-1} = \xi_{k-1})$ have been computed as outlined in Section III and stored in an $M$ vector for cyclic reading. Random phase trajectories and measurement variables have been generated according to (8) and (9). The results of several Monte-Carlo simulations are presented in Fig. 4. Each Monte-Carlo result has been obtained by running the Viterbi phase tracker (and the PLL) over 40 different trajectories, each trajectory beginning with a uniformly distributed phase variable at $k = 1$ and continuing for 500 points. Various values of depth parameter $k_p$ have been used, as indicated in the figure. (See [8] for a discussion of corresponding statistical sampling errors and [18] for additional Monte-Carlo results.)

IX. CONCLUSIONS

We have derived a Viterbi algorithm for obtaining approximate MAP phase sequence estimates on $[-\pi, \pi]$. The algorithm is simple and fast by nonlinear filtering standards and ideally organized for hardware implementation. More dramatic performance gains than those illustrated in Fig. 4 may be achieved when phase fluctuations are severe, i.e., when $a^2 < a^2 > 0.01$. The reader is referred to [19] for applications of these results to phase coherent communication.

ACKNOWLEDGMENT

The authors acknowledge the support of J. Lord, R. McGough, and B. Picinbobo. They thank J. Farley and Vaclav Macchi for helpful discussions and for her critique of the manuscript. C. Pariente conducted the Monte-Carlo simulations at the University of Paris-Sud using software originally developed C. J. Masreliez.

Fig. 4. Performance results for $a^2/a^2 = 0.01$. In Fig. 4 Monte-Carlo simulation results for the Viterbi tracker are presented for the parametrizations commonly considered in the literature. The results are compared with the point mass filter (PMF) [8], the Fourier coefficient filter (FCF) [7], the linear quadrature filter (LQF) [9], and the Gaussian sum filter (GSF) [13]. Also shown are our simulation results for the PLL. These results are presented to legitimize the simulation. The Viterbi tracker makes up more than 1.0 of the 2.0 dB performance gap between the PLL and an idealized linear tracker. In terms of rms phase error (in radians), the comparison between the PLL and the Viterbi tracker goes as follows. The PLL has an rms phase error of 1.26 rad at $r = 1.0$. The maximum achievable percentage improvement is 21 percent, corresponding to an ideal filter with rms phase error of 1.0 rad. The rms error for the Viterbi tracker operating at $r = 1.0$ with $k_p = 10$ is 1.12. This represents an improvement of 11 percent over the PLL. The results for $k_p = 0$ show that (as expected) the Viterbi tracker is not as good as a PLL as a zero-lag filter. In Fig. 4 the heavy squares denoted by VT(MAP) (see the symbol key) correspond to the smoothing variance achieved when the MAP sequence for a 500 sample run is used as the phase estimate. The results are averaged over 40 such runs. The tabulated results in Fig. 4 summarize the performance characteristics of many different nonlinear phase trackers. In the figure, performance results for the Viterbi tracker are plotted just to the left of their true positions to avoid cluttering the presentation.

We hasten to emphasize in the interest of fair play that all results presented here for nonzero $k_p$ are in reality smoothing solutions. Such solutions are expected to deliver the usual smoothing gains over filtering solutions. This does not detract from the Viterbi tracker as an attractive alternative in those applications where a short delay may be accepted in exchange for 1–2 dB performance gains.
REFERENCES

DYNAMIC PROGRAMMING FOR PHASE AND FREQUENCY TRACKING

Louis L. Scharf

Electrical Engineering Department
Colorado State University
Fort Collins, CO 80523 USA

Abstract

The techniques of dynamic programming have found a variety of successful applications in signal and system theory. In this paper we show how two knotty nonlinear filtering problems—phase and frequency tracking—may be formulated and solved as forward dynamic programming problems. The resulting solutions are fixed interval smooths in which a most likely sequence is passed through a data record.

1. INTRODUCTION

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the realm of signal processing. These problems have held the interest of control and communication theorists at least since 1953/54 when Leham and Parks [1] and Youla [2] published their work on maximum likelihood and optimum demodulation on an interval. Over the years Cox [3], Viterbi [4], Cahn [5], Forney [6], and a host of others have advocated dynamic programming for the solution of nonlinear filtering problems. This is a paper in the same tradition.

In this paper we discuss forward dynamic programming as a
technique for finding the maximum a posteriori (MAP) phase or frequency modulated sequence to pass through a data set. The key idea is to pose a Markov chain model on the circle $[0, 2\pi)$ for phase or frequency, and then generate candidate MAP sequences that are consistent with the data and the given probability structure. More details may be found in [7] & [8].

2. PHASE SEQUENCE ESTIMATION

Figures 1 & 2 depict two classical phase estimation problems: constant phase estimation and random walk phase estimation. In these figures and throughout the paper $Z_1, Z_2, \ldots, Z_k$ denotes the data set and the $\mathbf{X}_{k+1}$ are complex i.i.d. $\mathcal{N}(0, I)$ r.v.s. A Markov transition density (or probability mass function) is denoted $p(\cdot|\cdot)$; $f(\cdot, \ldots, \cdot)$ denotes a joint density function; $\lfloor \cdot \rfloor$ denotes integer part.

**Figure 1: Constant Phase Estimation**

**Figure 2: Random Walk Phase Estimation**
This problem involves mapping the data set $Z_k$ into a phase sequence estimate $\phi_1, \ldots, \phi_t \in [0,2\pi)$ when measurements are generated according to

$$z_k = \exp(i\phi) + n_k, \quad k = 1, \ldots, t$$

It is a straightforward exercise in maximum likelihood (ML) theory to show

$$\phi = \arg \max_{\phi} z_k$$

As shown in Figure 1a, this estimator maximizes the log-likelihood of $[z_1, \ldots, z_t]$:

$$\phi = \arg \max_{\phi} \ln f_c(z_1, \ldots, z_t | \phi)$$

Geometrically, the estimator is obtained by piecing measurements together, feather-to-tip, and measuring the angle to the resulting vector. This is illustrated in Figure 1b. The diagram in Figure 1c illustrates that if each measurement is rotated through an angle $\phi$, and each rotated measurement added to the previous, the result is purely real.

**Problem 0: Constant Phase (Figure 1)**

Here the problem is to map the data set $Z_k$ into a phase sequence estimate $[\phi_1, \ldots, \phi_t] \in [0,2\pi)$ when measurements are generated according to

$$z_k = \exp(i\phi_k) + n_k, \quad k = 1, 2, \ldots, t$$

$p(\phi_k | \phi_{k-1})$ given

As shown in Figure 2a, the MAP phase sequence maximizes the log-likelihood of $[z_1, \ldots, z_t]$:

$$(\phi_1, \ldots, \phi_t) = \arg \max_{\phi} \ln f_c(z_1, \ldots, z_t, \phi_1, \ldots, \phi_t)$$

Here $f$ is the joint density of the measurements and the phase sequence. The likelihood $v_t = \ln f_t$ may be written

$$v_t = v_{t-1} - \frac{1}{2} z_t^2 + \log p(\phi_t | \phi_{t-1})$$

**Problem 1: Random Walk Phase (Figure 2)**

Here the problem is to map the data set $Z_k$ into a phase sequence estimate $[\phi_1, \ldots, \phi_t] \in [0,2\pi)$ when measurements are generated according to

$$z_k = \exp(i\phi_k) + n_k, \quad k = 1, 2, \ldots, t$$

$p(\phi_k | \phi_{k-1})$ given

As shown in Figure 2a, the MAP phase sequence maximizes the log-likelihood of $[z_1, \ldots, z_t]$:

$$(\phi_1, \ldots, \phi_t) = \arg \max_{\phi} \ln f_c(z_1, \ldots, z_t, \phi_1, \ldots, \phi_t)$$

Here $f$ is the joint density of the measurements and the phase sequence. The likelihood $v_t = \ln f_t$ may be written

$$v_t = v_{t-1} - \frac{1}{2} z_t^2 + \log p(\phi_t | \phi_{t-1})$$
1. FREQUENCY SEQUENCE ESTIMATION

Figures 3 and 4 depict two classical frequency estimation problems: constant frequency estimation and random walk frequency estimation.

**Problem 1:**

\[ \lambda \in \{ \lambda_0, \lambda_1, \ldots, \lambda_N \} \]

\[ \hat{\lambda} = \arg \max_i |X(\lambda_i)/X(\lambda)| \]

\[ \hat{\phi} = \arg \max_i \hat{X}(\phi) \]

\[ X(e^{j\lambda\phi}) = 1 \]

\[ X(e^{j\lambda_i\phi}) = \hat{X}(\phi) \]

\[ X(e^{j\lambda_{i+1}\phi}) = \hat{X}^{-1}(\phi) \]

**Problem 2:**

**Figure 3:** Constant Frequency Estimation

**Figure 4:** Random Walk Frequency Estimation

Problem 3: Constant Frequency Estimation (Figure 3)

This problem involves mapping the data set \( Z_k \) into a frequency estimate \( \hat{\lambda}_k \) when measurements are generated according to...
Problem 1: Random Walk Frequency Estimation (Figure 4)

Here the problem is to map the data set $Z_t$ into a frequency sequence estimate $\{\hat{f}(0), \ldots, \hat{f}(t/N)\}$ when measurements are generated according to

$$z_k = \exp(j\omega[k/N]) + n_k, \quad k=0,1,\ldots,t$$

$$p(\omega[k/N] | \omega[k/N]-1)$$

As shown in Figure 4a the MAP phase sequence maximizes the log-likelihood of $\{z_0, \ldots, z_t\}$.

$$\hat{f}(0), \ldots, \hat{f}(t/N) = \arg \max_{f(0), \ldots, f(t/N)} \log f_t \left( z_0, \ldots, z_t | 0, \ldots, t/N \right)$$

The log-likelihood $L = \log f_t$ may be written

$$L = -\sum_{k=1}^{t} \log |X(k)|^2 + n_p \left( |t|^2 / |t|-1 \right)$$

$X(k)$ is the Fourier transform of $x(t)$.
Here $X$ is DFT over the $i$th data block of $N$ samples. So if $\omega(t)$ takes values in a discrete set (say $q_{2}/q, q=0, 1, \ldots, Q-1$), one can implement a dynamic programming algorithm on the lattice of Figure 4b to decode the MAP sequence. See [8] for details.

4. CONCLUSIONS

The problems discussed here generalize. The basic idea is to select states and transition probabilities to characterize an underlying probabilistic structure, and then to assign characters $\mathbf{c}_k$ (such as $e^{-i 2\pi k}$ or $e^{i 2\pi k/N}$) to the states. The resulting sequence estimation algorithms are attractive because storage goes like $Q$ (number of states) and computations are naturally parallel.

REFERENCES


A DYNAMIC PROGRAMMING ALGORITHM FOR
PHASE ESTIMATION AND DATA DECODING
ON RANDOM PHASE CHANNELS

Odile Macchi and Louis L. Scharf

0018-9448/81/0900-0581$00.75 © 1981 IEEE
A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels

ODILE MACCHI, MEMBER, IEEE, AND LOUIS L. SCHARF, SENIOR MEMBER, IEEE

Abstract—The problem of simultaneously estimating phase and decoding data symbols from baseband data is posed. The phase sequence is assumed to be a random sequence on the circle, and the symbols are assumed to be equally likely symbols transmitted over a perfectly equalized channel. A dynamic programming algorithm (Viterbi algorithm) is derived for decoding a maximum a posteriori (MAP) phase-symbol sequence on a finite dimensional phase-symbol trellis. A new and interesting principle of optimality for simultaneously estimating phase and decoding phase-amplitude coded symbols leads to an efficient two-step decoding procedure for decoding phase-symbol sequences. Simulation results for binary, 8-ary phase shift keyed (PSK), and 16-quadrature amplitude shift keyed (QASK) symbol sets transmitted over random walks and sinusoidal jitter channels are presented and compared with results one may obtain with a decision-directed algorithm or with the binary Viterbi algorithm introduced by Ungerboeck. When phase fluctuations are severe and when occasional large phase fluctuations exist, MAP phase-symbol sequence decoding on circles is superior to Ungerboeck's technique, which in turn is superior to decision-directed techniques.

I. INTRODUCTION

Phase fluctuations can significantly increase the error probability for symbols transmitted over a channel that may or may not have been equalized. This is especially true for phase shift keyed (PSK) and quadrature amplitude shift keyed (QASK) symboling, in which case accurate phase discrimination is essential for symbol decoding. Even when the receiver contains a decision-directed phase-locked loop (DDPLL), performance loss in signal-to-noise ratio (SNR) with respect to a coherent decoding system can be in the range 5–10 dB. This fact is established in [1] for practical symbol sets and typical values of the phase variance parameter and symbol error probability.

On telephone lines, linear distortion and phase jitter dictate the use of a channel equalizer and some kind of phase estimator to achieve high rate, low error probability data transmission. A common approach to phase estimation and data decoding is to use a decision-directed algorithm in which a phase estimate is updated on the basis of old phase estimates and old symbol decisions. In the jitter equalizer (JE) of [3] and [4] a complex gain is updated according to a simple decision-directed stochastic ap-
proximation algorithm. The complex gain is used to scale and rotate the received signal, thereby correcting phase jitter and normalizing rapid fading variations. Although there is no explicit interest in phase estimation itself in the JE, it is possible to interpret the structure as an adaptive gain-phase correcting equalizer.

Both the DDPLL and the JE are very simple to implement, but apparently neither achieves optimality with respect to any statistical criterion for symbol (or data) decoding. Furthermore, neither the DDPLL nor the JE is optimum for estimating and/or correcting phase. Both are zero-lag phase estimators that cannot benefit from future signal samples. Therefore, an important question to be answered is whether or not symbol decoding can be improved using a better phase estimator. The answer, based on the results of [1] and this paper, is that significant improvements can be realized when the phase fluctuations are severe if one is willing to pay the price of an increased computational burden. In practice, cases of severe phase fluctuation can occur in high data rate PSK and QASK systems in which the angular distance between symbols is small.

In [1] Ungerboeck recognized the potential of maximum a posteriori (MAP) sequence estimation for jointly estimating phase and decoding data symbols. A path metric was derived and its role in a forward dynamic programming algorithm for obtaining MAP phase-symbol sequences was indicated. Because of the way phase was modeled in [1], the dynamic programming algorithm could not be solved directly. Ungerboeck approximated the phase sequence as a process that could make discrete binary jumps and then derived a dynamic programming algorithm for decoding likely paths around a developing most likely path. The result is a tree-search algorithm which may branch left or right but never go straight. He obtained performance results that were on the order of 3 dB superior in SNR to the DDPLL in a 16-QASK system, at interesting values of the phase variance parameter. We call the algorithm of [1] a discrete binary Viterbi algorithm (DBVA). The reader is referred also to [5] and [6] for discussions of other suboptimal, but computationally tractable, algorithms for simultaneously estimating phase and decoding data symbols.

In this paper we observe that baseband data is invariant to modulo-2π transformations on the phase sequence. This motivates us to wrap the phase around the circle, so to speak, and obtain folded probability models for transition probabilities on the circle. When the phase process is normal random walk on the circle, then the transition probabilities are described by a folded normal model. This model has also been used in [7] and [8]. It is then straightforward to pose a MAP sequence estimation problem for simultaneous phase and symbol sequence decoding as described in [8] and [9]. The basic idea is to discretize the phase space [−π, π) to a finite dimensional grid and to use a dynamic programming algorithm (Viterbi algorithm) to keep track of surviving phase-symbol sequences that can ultimately approximate the desired MAP phase-symbol sequence. The MAP phase-symbol sequence itself is the entire sequence of past phases and symbols that is most likely, given an entire sequence of recorded observations. It is this use of "future" and "past" received signal samples that provides performance improvement over zero-lag estimators such as the DDPLL. Details of the algorithm are given in [8] and [9]. For PSK and QASK symbol sets an interesting principle of optimality leads to an efficient two-step decoding procedure. With this procedure, computational complexity is reduced by a factor greater than the number of admissible phase values per amplitude level. This amounts to a factor of four for the 16-point QASK diagram that has been recommended by CCITT for data transmission on telephone lines at 9600 bits/s. Finally, in order to make the computation and storage requirements tractable in the Viterbi algorithm, we use it in a fixed delay mode, as do other authors. By appealing to known results for fixed-lag smoothing of linearly observed data, we are able to intelligently choose the fixed delay. Without significant performance loss we decode phase-symbol pairs at a depth constant of k0 = 10. This obviates the need for huge storage requirements for long sequences. With these modifications the Viterbi algorithm becomes a feasible, albeit sophisticated, decoding procedure.

Simulation results for the proposed Viterbi algorithm (VA) are presented for several symbol sets consisting of two, eight, or 16 symbols. Several types of phase jitter are investigated such as Gaussian and non-Gaussian random walk and sinusoidal phase jitter. The resulting error probabilities are compared with those of the simpler decision-directed algorithms (JE and DDPLL) and with those of the DBVA. As expected, performance of the VA is always superior to that of the other systems. On the other hand, the increase in computational burden is substantial, and the improvement in performance is not always great enough to warrant the use of the VA. In our concluding remarks we discuss situations in which one might reasonably use the VA or the DBVA rather than a simpler decision-directed algorithm such as the JE or the DDPLL.

Remarks on Notation:

Throughout this paper I denotes statistical independence. The notation \( \{ \phi_k \}^K \) will mean the set \( \{ \phi_k, k = 1, 2, \ldots, K \} \). When the indexes 1 and K are missing (e.g., \( \{ \phi_k \} \)), it is understood that K is infinite. The symbol \( N^+ \) denotes the positive integers. The notation \( x : N \) means the random variable x is normally distributed with mean \( \mu \) and variance \( \sigma^2 \); \( N \) will also be used to denote the function \( \exp(2\pi x) \exp(-x) \). When x is complex, \( x : N \) means x is complex with density \( N \), \( \sigma^2 \) is the conditional probability density of the random variable x, given the random variable y. Thus f(x/y) is generally a different function than f'(x/2), even though we use no explicit subscripting such as \( f(x/\cdot) \) to indicate so. We make no notational distinction between a random variable and its realizations, relying instead on context to make the meaning clear. A density function for a random variable, evaluated at a particular realization of the random variable is termed a likelihood function.
“Hatted” variables such as $\hat{\phi}_k$ refer always to MAP estimates that maximize an a posteriori density. Finally, it is convenient to define the function

$$g_M(x) = M^{-1} \sum_{m=1}^{M} \sum_{l=-\infty}^{\infty} h[x - l2\pi - (m-1)2\pi/M]$$

where $h(\cdot)$ is a probability density. The function $g_M(\cdot)$ plays an important role in our discussion of phase-symbol decoding on QASK symbol sets.

II. SIGNAL AND PHASE MODELS

Assume complex data symbols $(a_k)$ are phase or phase-amplitude modulated onto a carrier and transmitted over a channel with linear distortion and phase jitter. The received signal—call it $y(t)$—is typically processed as illustrated in Fig. 1. The signal $y(t)$ is passed through a bandpass noise filter and demodulated with two quadrature waveforms. The resulting complex baseband signal $x_1(t) + jx_2(t)$ is equalized with a complex adaptive equalizer in order to reduce the intersymbol interference due to linear distortion in the channel. The equalized signal is a sequence of samples at symbol rate $1/\Delta$ ($\Delta$ is the interval between successive data symbols). The output of the equalizer is a complex sequence $x_k = x_k^{(1)} + jx_k^{(2)}$ which is a noisy, phase-distorted, version of the original transmitted sequence. Thus we write

$$x_k = a_k e^{j\phi_k} + n_k, \quad k \in N^+$$

Here, $(a_k)$ is the complex symbol sequence, typically encoded according to one of the diagrams illustrated in Fig. 2. The sequence $(\phi_k)$ represents phase fluctuations (jitter and frequency drift) in the channel. The two real components $n_k^{(1)}$ and $n_k^{(2)}$ of the complex noise sequence $n_k = n_k^{(1)} + jn_k^{(2)}$ are the noise variables in the respective baseband quadrature equalized channels. The variables $n_k^{(1)}$ and $n_k^{(2)}$ can be shown to be independent when the carrier frequency is in the middle of the input noise filter bandwidth and the additive channel noise is white. If the equalizer is perfect, then $n_k$ is the usual Gaussian, additive noise with zero-mean. If the equalizer is not perfect, then $n_k$ contains a residual of the intersymbol interferences, and is not Gaussian; nor are successive variables $n_k^{(1)}$, $n_k^{(2)}$ independent. However, for a reasonably good equalizer, we may assume that $(n_k)$ is a sequence of independent identically distributed (i.i.d.) complex Gaussian variables. Strictly speaking, this assumption is valid only at the input to the equalizer when the baseband equivalent of the input noise filter and low-pass demodulator is the so-called sampled whitened matched filter of [10]. In practice, the assumption of Gaussianity is more realistic than the assumption of independence for the sequence $(n_k)$. Assuming that the equalizer of Fig. 1 is perfect, we model the noise sequence $(n_k)$ as follows:

$$n_k = n_k^{(1)} + jn_k^{(2)}, \quad k \in N^+$$

$$n_k^{(1)} \perp n_k^{(2)}, \quad \forall (k, l)$$

$$n_k^{(1)} \perp n_l^{(1)}, \quad k \neq l, \quad n_k^{(1)} \perp n_l^{(2)}, \quad k \neq l$$

Here $2\sigma^2$ is the variance of the complex noise variable $n_k$, and $\sigma^2_k$ is the variance of each real component.

Consider now the phase distortion $(\phi_k)$. The term generally reflects two effects, one long-term and the other short-term. In modern high speed data modems no carrier or pilot tone is transmitted for locking the local oscillator at the receiver. Thus long-term large-range linear phase variations result from frequency drift in the channel which cannot be eliminated. In addition, nonlinear intermodulations with local power supplies gives rise to short-term
small-range phase variations. The variations exhibit energet-  
getic harmonic content at the harmonics of the fundamental  
power supply frequency. Hence a realistic model for  
\( \Phi_k \) is

\[
\Phi_k = (\Phi_0 + 2\pi Bk) + \sum_{i=1}^{\rho} A_i \sin(2\pi v_i k\Delta + \rho_i), \quad k \in N^+
\]

(4)

where \( v_i = 1 \cdot 50 \, \text{Hz} \) or \( v_i = 1 \cdot 60 \, \text{Hz} \), depending on the  
place of use. A typical phase process is depicted in Fig. 3.  
The first term in parentheses in (4) is the so-called frequency  
drift term and the summation term is the phase jitter. In  
practice, the constants \( \Phi_0, B, \{A_i, v_i, \rho_i\}\) vary with time  
\( k\Delta \) but at an extremely slow rate.

The spectrum of the phase jitter, i.e., the behavior of \( A_i \)  
versus \( v_i \), has been investigated experimentally in [14]. The  
spectrum is roughly fitted by a \( 1/f^2 \) curve. A phenomeno-  
logical model for phase having a \( 1/f^2 \) spectrum (like that of  
phase jitter at high frequencies) is the Wiener-Levy  
continuous time process,

\[
\frac{d\Phi(t)}{dt} = w(t), \quad t \geq 0.
\]

(5)

where \( \{w(t)\} \) is a white noise process. The discrete time  
analog is the independent increments sequence

\[
\Phi_k = \Phi_{k-1} + w_k, \quad k \in N^+
\]

(6)

where \( \{w_k\} \) is a sequence of i.i.d. random variables with  
even probability density \( h(w) \).\(^1\) When \( w_k : N_0(0, \sigma_w^2) \), then  
\( \{\Phi_k\} \) is the so-called normal random walk.

For short-term fluctuations, the model captures, with  
appropriate selection of \( h(w) \), the correlated evolution of  
phase. The main virtue of the independent increments model is  
it forms a convenient basis from which to derive optimum estimator structures which may then be evaluated against more realistic phase sequences.

Since the measurement model of (2) is invariant to  
modulo-2\( \pi \) translates of \( \Phi_k \), we may represent phase as if it  
were a random sequence on the unit circle \( C \) or equivalently  
on the interval \([-\pi, \pi)\). Call \( \phi_k \) this representation of \( \Phi_k \). Note \( \Phi_{k+1} \) may be written

\[
\Phi_{k+1} = \Phi_k + \bar{w}_k
\]

(7)

where the plus sign denotes modulo-2\( \pi \) addition of real  
variables or equivalently rotation with positive (counterclockwise) sense on \( C \). The variable \( \bar{w}_k \) is a modulo-2\( \pi \)  
version of \( w_k \).

The conditional density of \( \Phi_{k+1} \) is \( h(\Phi_{k+1} - \Phi_k) \). Since \( \Phi_{k+1} \) is a modulo-2\( \pi \) version of \( \Phi_{k+1} \),

\[
\text{we may reflect all of the conditional probability mass into } C \text{ to obtain the transition (or conditional) probability density}
\]

\[
f(\Phi_{k+1} | \Phi_k) = \sum_{i=0}^{\infty} h(\Phi_{k+1} - \Phi_k - 2\pi i)
\]

\[
= g_i(\Phi_{k+1} - \Phi_k)
\]

(8)

where \( g_i \) is the function defined in (1). Hereafter, \( g_i(\cdot) \)  
is called the folded density of the phase increments. Usually,  
the phase increment is small and its distribution \( h(\cdot) \) is very narrow with respect to \( 2\pi \). Therefore, in the sum of  
(8) only one term is relevant and \( f(\Phi_{k+1} | \Phi_k) \approx h(\Phi_{k+1} - \Phi_k) \). In the normal case, this implies \( \sigma_w^2 \ll 2\pi \), where \( \sigma_w^2 \)  
is the variance of \( w_k \). As it is cumbersome to carry around the  
overbar notation \( \Phi_{k+1} - \Phi_k \), we drop it with the caution  
that from here on \( \Phi_k \) is defined on \( C \) unless otherwise  
noted.

In the normal case [7], [8], the density \( g_i(\Phi_{k+1} - \Phi_k) \)  
may be written

\[
g_i(\Phi_{k+1} - \Phi_k) = \sum_{i=-\infty}^{\infty} N_{\sigma_w^2}(\Phi_{k+1} + 2\pi i, \sigma_w^2).
\]

(9)

This case and the Cauchy case (in which the distribution  
tails are much heavier than the normal tails) are studied in  
the Appendix. It is shown that \( g_i(x) \) achieves its maximum  
at \( x = 0 \) and that it is monotonically decreasing on \( 0 \leq x \leq \pi \).

The sequence \( \{\Phi_k\} \) is Markov. Therefore, we may write  
for the joint density of the \( K \) phases \( \{\Phi_k\} \)

\[
f(\{\Phi_k\}) = \prod_{k=0}^{K-1} f(\Phi_{k+1} | \Phi_k)
\]

\[
f(\{\Phi_k\}) \propto f(\Phi_0): \text{the marginal density of } \Phi_0.
\]

(10)

Usually, \( \Phi_0 \) is uniformly distributed on \( C \) because phase  
acquisition starts at \( k = 1 \) with no prior information about  
its value. By the independence of the \( \Phi_k \) in (2), it follows  
that the conditional density of the measurement sequence  
\( \{x_k\}_k \), given the phase and data sequences \( \{\Phi_k\}_k, \{a_k\}_k \), is

\[
f(\{x_k\}_K | \{\Phi_k\}_K, \{a_k\}_K) = \prod_{k=1}^{K} N_{\sigma_e^2}(a_k e^{j\phi_k}, \sigma_e^2).
\]

(11)

Equations (8)–(11) form the basis for the derivation of a  
MAP sequence estimator. The key element is that \( \{\Phi_k\} \)  
is a Markov sequence with a bounded range space \((-\pi, \pi)\).  
Discretization of this bounded interval leads to a finite-state  
model from which a finite dimensional dynamic programming  
algorithm can be derived.

III. DECISION-DIRECTED ALGORITHMS

The usual way of dealing with phase fluctuations is to  
design a phase estimator and use the estimated phase, call it  
\( \hat{\phi}_k \), to rotate the received signal as follows:

\[
y_k = x_k e^{-j\hat{\phi}_k}, \quad k \in N^+.
\]

(12)

The phase corrected signal \( y_k \) is then fed to a decision  
device which, in turn, delivers the symbol estimate \( \hat{d}_k \).
Typically, the phase estimate \( \hat{\phi}_k \) is functionally dependent on the old measurements \( \{ x_{k-2}, \ldots, x_k \} \) and the past symbol estimates \( \{ \hat{a}_{k-2}, \ldots, \hat{a}_{k-1} \} \). If a carrier or pilot tone is transmitted as in typical single sideband (SSB) systems, then \( \phi_k \) is obtained from a simple phase-locked loop (PLL). In suppressed carrier systems such as PSK or QASK systems, the PLL is decision-directed. That is, \( \phi_k \) is updated on the basis of \( \hat{a}_{k-1} \). For instance in [5]

\[
\hat{\phi}_{k+1} = \hat{\phi}_k + \mu \text{Im}[x_k \hat{a}_k^* e^{-\hat{\phi}_k}]
\]

\[
= \hat{\phi}_k + \mu_k \sin(\arg x_k - \arg \hat{a}_k - \phi_k),
\]

where the asterisk denotes complex conjugate and \( \mu \) is a constant that depends on the SNR. The estimator of (13) is called a DDPLL.

In the jitter equalizer (JE) of [3] and [4], \( x_k \) is rotated and scaled as follows:

\[
y_k = x_k G_k
\]

\[
G_k = G_{k-1} + \mu (\hat{a}_{k-1} - y_{k-1}) x_{k-1}^*.
\]

The complex gain \( G_k \) is the single complex coefficient of a one-coefficient rapidly adaptive equalizer. We may think of \( G_k / |G_k| \) as the phase correction \( e^{j\Delta_k} \), and \( |G_k| \) as a gain correction \( G_k \). Thus, although there is no explicit formulation of a phase-gain estimation problem in [3] and [4], the net effect of the JE is to correct phase and normalize rapid fading variations. As explained in [4], when phase fluctuations are large, the JE performance may be improved by setting a constraint on \( G_k \) that keeps its value inside a given domain including the complex point \( \pm 1, 0 \).

**Geometrical Comments**

The combined effects of random phase fluctuations and additive noise may be illustrated as in Fig. 4(a). The transmitted symbol \( a_t = a^{(0)} \) (say) is rotated by the random phase angle \( \phi_k \) to give \( a_t e^{j\phi_k} \). To this is added the complex noise sample \( n_k \) to give the measurement \( x_k \) defined in (2). For the case illustrated, the resultant measurement is closer to symbol \( a^{(0)} \) than to \( a^{(0)} \) and consequently, with no phase or phase-gain correction, a decoding error would be made. To emphasize the combined effects of phase fluctuation and additive noise, we have illustrated a case for which either phase jitter or additive noise alone would cause no error. See [11] for a probabilistic discussion of this issue. Fig. 4(b) is an illustration of how a DDPLL works. The angle \( \psi_k \) is the noisy measured phase \( \arg x_k \) minus the sum of the phase of the decoded symbol and the previously estimated phase \( \arg \hat{a}_k + \phi_k \).

A given amount \( \mu_k \) of this angle is added to \( \phi_k \) as a correction to get the new phase estimate \( \hat{\phi}_{k+1} = \phi_k + \mu_k \psi_k \). Note that only phase is corrected. In the JE both phase and gain are corrected, offering potential for improved performance. This potential is particularly important in QASK symbol sets where amplitude errors in \( x_k \) can result in decoding errors.

![Fig. 4. Geometry of phase jitter and additive noise with and without phase correction of DDPLL. (a) Without (b) With.](image)

**IV. MAP PHASE AND SYMBOL SEQUENCE DECODING WITH THE VITERBI ALGORITHM**

The basic idea behind MAP sequence decoding is to find a sequence of phase-symbol pairs \( \{ \phi_k, a_k \}^K \) that, based on the observation sequence \( \{ x_k \}^K \), appears most likely. The application of this idea to data communication was first proposed in [1] and refined in [9]. The most likely sequence, call it \( \{ \phi_k, \hat{a}_k \} \), is the sequence that maximizes the natural logarithm (or any other monotone function of) the a posteriori density of \( \{ \phi_k, a_k \}^K \), given the sequence of observations \( \{ x_k \}^K \). Thus we pose the maximization problem:

\[
\max_{\{ \phi_k, a_k \}^K} \ln f(\{ \phi_k \}^K, (a_k)^K / \{ x_k \}^K).
\]

This is equivalent to maximizing the natural logarithm of the likelihood function \( f((x_k)^K, \{ \phi_k \}^K, (a_k)^K) \), obtained by evaluating the joint density function for \( \{ x_k \}^K, \{ \phi_k \}^K, \) and \( (a_k)^K \), at the observed values of \( \{ x_k \}^K \). Using the results of (10) and (11) we may write

\[
f((x_k)^K, \{ \phi_k \}^K, (a_k)^K)
\]

\[
= \left[ \prod_{k=1}^K N_s(a_k e^{j\phi_k}, \sigma_k^2) / f(\phi_k / \phi_{k-1}) \right] f((a_k)^K).
\]

Assuming the \( (a_k)^K \) to be a sequence of independent, equally likely symbols, using (8), and neglecting irrelevant constants, we may write the maximization problem as

\[
\max_{\{ \phi_k, a_k \}^K} \Gamma_K
\]

\[
\Gamma_k = -\frac{1}{2\sigma_k^2} \sum_{k=1}^K |x_k - a_k e^{j\phi_k}|^2
\]

\[+ \sum_{k=2}^K \ln g_i(\phi_k - \phi_{k-1}) + \ln f(\phi_k).
\]

Note that \( \Gamma_k \) satisfies the recursion

\[\Gamma_k = \Gamma_{k-1} + p_k, \quad k = 2, 3, \ldots
\]

\[p_k = -\frac{1}{2\sigma_k^2} |x_k - a_k e^{j\phi_k}|^2 + \ln g_i(\phi_k - \phi_{k-1}), \quad k = 2, 3, \ldots
\]

where \( p_k \) is the so-called path-metric. For convenience, let us make explicit in \( \Gamma_K \) the last phase and symbol:

\[\Gamma_K(\phi_K, a_K). \]

The other arguments \( (\phi_k)^K-1, (a_k)^K-1, \) re-
main implicit. Then, from (18)
\[ \Gamma_K(\Phi_K, a_K) = \Gamma_{K-1}(\Phi_{K-1}, a_{K-1}) + p_K(x_K, a_K, \Phi_K, \Phi_{K-1}). \]  

Thus the maximizing sequence—call it \((\phi_{K})^*, (\theta_{K})^*, (\phi_{K-1})^*, (\theta_{K-1})^*\)—passing through \((\phi_{K-1}, a_{K-1})\) on its way to \((\phi_{K}, a_{K})\), must arrive at \((\phi_{K-1}, a_{K-1})\) along a route \((\phi_{K-1}^*, (\theta_{K-1})^*), (\phi_{K})^*, (\theta_{K})^*)\) that maximizes \(\Gamma_{K-1}(\phi_{K-1}, a_{K-1})\). It is this observation which forms the basis of forward dynamic programming.

In the actual implementation of a dynamic programming algorithm, one must discretize the phase space \(C\) to a finite dimensional grid of phase values \(\Xi = \{\xi_m\}_{m=1}^M\). The function \(\ln g_i(\phi_m - \phi_{m-1})\) is then defined on the two-dimensional grid \(\Xi \times \Xi\). However, as discussed in [8] and [9], the resulting \(M \times M\) matrix of conditional probabilities has Toeplitz symmetry which means only an \(M\) vector of conditional probabilities must be computed and stored.

The Viterbi algorithm for simultaneous phase and symbol decoding consists simply of an algorithm which determines survivor phase-symbol sequences terminating at each possible phase-symbol pair. One of these surviving sequences is ultimately decoded as the approximate MAP phase-symbol sequence. The complexity \(c\) of the algorithm lies mainly in the evaluation of the \(M\) possible approximate values of \(|x_k - a_k e^{i\theta_k}|^2\), for each new measurement \(x_k\). Here \(M\) is the symbolizing alphabet size, and \(m\) is the number of discrete phase values. For each calculation of \(|x_k - a_k e^{i\theta_k}|^2\) there are six real multiplies. Compared to this multiplication load of \(6mM\) per sample, the determination and addition of the \(M\) possible values of \(\ln g_i(\phi_m - \phi_{m-1})\) that appear in (18) is negligible. The determination of \(|x_k - a_k e^{i\theta_k}|^2\) would likely be computed in a pipelined parallel architecture, while the terms in \(g_i(\cdot)\) would be read by appropriately addressing read only memory (ROM). When short-term phase fluctuations have small amplitude \(a_k\) small) so that \(m\) must be large for accurate phase tracking, the complexity increases. For example, with \(M = 8\) and \(m = 48, c \approx 384\), indicating on the order of \(2 \times 10^3\) computations at each \(k\)-step.

As we show in the next section, the complexity of the Viterbi algorithm can be dramatically reduced by making a change of variable and tracking a total phase amplitude that is the sum of \(\phi_k\) and the symbol phase, \(\arg a_{k}\). Also, of course, for PSK symbol sets only one symbol amplitude is admissible, and admissible symbol phases may be chosen to fall on one of the discrete phase values. Thus for PSK symbol sets the complexity is simply \(m\), and the number of path metric computations is on the order of 300 for \(m = 48\). Even this figure may be reduced by using one of a variety of so-called \(M\) algorithms in which all surviving phase-symbol pairs are saved, but only a handful of candidate originator pairs are considered for each survivor [16]-[18].

V. A PRINCIPLE OF OPTIMALITY FOR PHASE-AMPLITUDE CODED SYMBOLS AND AN EFFICIENT TWO-STEP DECODING PROCEDURE

In order to simplify matters and to illustrate the key ideas, let us consider PSK symbols of the form
\[ a_k = e^{i\theta_k}. \]

with \((\theta_k)\) drawn independently from an \(M\)-ary equiprobable alphabet \(\Theta = \{(l-1)2\pi/M\}_{l=1}^M\). Write the measurement model of (2) as
\[ x_k = e^{i\theta_k} + n_k \]

where the total phase \(\psi_k\) is represented as
\[ \psi_k = \phi_k + \theta_k \]
\[ \theta_k = \sum_{i=1}^k \Delta \theta_i. \quad \Delta \theta_k = \theta_k - \theta_{k-1} \quad \Delta \theta_1 = \theta_1. \]

It is clear that \(\hat{\theta}_k = \sum_{i=1}^k \Delta \theta_i\) and \(\hat{\psi}_k = \hat{\psi}_k - \hat{\theta}_k\). Thus we may replace the MAP sequence estimation problem posed in (15) by the problem
\[ \max_{(\psi_i)_{kM}=(\Delta \theta_i)^{k-1}} f((x_k)^k, (\psi_k)^k, (\Delta \theta_k)^k). \]

The joint density \(f \triangleq f(\cdot, \cdot, \cdot)\) in (23) may be written
\[ f_K = \prod_{k=1}^K p_{\xi_k}(e^{i\psi_k}, a_k^2) f(\psi_k, \Delta \theta_k / \psi_j^{k-1}, (\Delta \theta_j)^{k-1}) \]

where for \(k = 1\), \(f(\psi_1, \Delta \theta_1 / \psi_j^1, (\Delta \theta_j)^1)\) is simply the marginal density \(f(\psi_1, \Delta \theta_1)\). The conditional density on the right-hand side of (24) is easily evaluated with Bayes' rule:
\[ f(\psi_k, \Delta \theta_k / \psi_j^1, (\Delta \theta_j)^1) = f(\psi_k / \psi_j^{k-1}, (\Delta \theta_j)^{k-1}) \cdot f(\Delta \theta_k / \psi_j^{k-1}, (\Delta \theta_j)^{k-1}) \]

Now \(\Delta \theta_k\) is independent of the previous data, additive noise and phase fluctuations. Thus
\[ f(\Delta \theta_k / \psi_j^{k-1}, (\Delta \theta_j)^{k-1}) = \frac{1}{M}. \]

Moreover, if we rewrite \(\psi_k\) as
\[ \psi_k = \phi_{k-1} + w_k + \theta_{k-1} + \theta_k - \theta_{k-1} \]
\[ = \phi_{k-1} + \Delta \theta_k + w_k, \]

we see immediately that
\[ f(\psi_k / \psi_j^{k-1}, (\Delta \theta_j)^{k-1}) = g_1(\psi_k - \psi_{k-1} - \Delta \theta_k). \]

Recall \(\psi_k\) is defined on the circle \(C\). Therefore, we might think of \(\psi_k\) as a random variable \(\psi_k \sim \Delta \theta_k + w_k\), whose density is folded in \((-\pi, \pi)\).

Putting (24)-(28) together, we have for the joint density \(f_K\)

\[ f_K = \prod_{k=1}^K p_{\xi_k}(e^{i\psi_k}, a_k^2) \frac{1}{M} \mathbf{g}_1(\psi_k - \psi_{k-1} - \Delta \theta_k) \]
\[ \Delta \theta_1 \triangleq \theta_1, \quad \psi_0 \triangleq 0. \]

Principle of Optimality

Call \((\psi_k)^k, (\Delta \theta_k)^k\) the MAP sequences that maximize \(f_K; (\Delta \theta_k)^k\) enters only in the \(g_1(\cdot)\) term on the right-hand side of (29). Now let us suppose (as is usual) that \(g_1(\cdot)\), which is even, is also unimodal with a peak at \(w = 0\). This single-mode assumption for \(g_1(\cdot)\) is valid in particular when the phase increment \(w_k\) in the Markov process (6) has a Gaussian or Cauchy distribution \(h(w)\) (see the Appen-
\[ \mathbf{h}(x) \text{ on } [-\infty, \infty] \]

(a)

\[ \mathbf{g}_1(x) \text{ on } [-\pi, \pi] \]

(b)

\[ \mathbf{g}_1(R(x)) \text{ on } [-\pi, \pi] \]

(c)

\[ \mathbf{g}_1(R(x)) \text{ Wrapped on the Circle } \mathbf{C. M+4} \]

(d)

Fig. 5. Density functions of phase increment before and after folding.

dix). It follows that \( f^K \) is maximized by choosing

\[ \Delta \delta_k = [\psi_k - \psi_{k-1}] \]  \hspace{1cm} (30)

where \([x]\) denotes the closest value of \((l-1)2\pi/M\) to \(x\).

By substitution of the constraint (30) into (29) and defining the "rest" function \( R(x) \) on the circle \( \mathbf{C} \) by

\[ R(x) = x - [x], \]  \hspace{1cm} (31)

we find that one must maximize

\[ f^K = \prod_{k=1}^{K} N_{s_k}(e^{i\psi_k}, \sigma^2_s) \frac{1}{M} g_s(R(\psi_k - \psi_{k-1})). \]  \hspace{1cm} (32)

The maximization of \( f^K \) with respect of \((\psi_k)^K\) is formally equivalent to maximizing the joint density \( f((x_k)^K, (\psi_k)^K) \) when the total phase \( \psi_k \) follows a Markov-model similar to \( (6)\):

\[ \psi_k = \psi_{k-1} + u_k. \]  \hspace{1cm} (33)

Here the independent increments \( u_k \) have "probability density," folded on the circle \( \mathbf{C} \),

\[ f(u) = \frac{1}{M} g_s(R(u)). \]  \hspace{1cm} (34)

This interpretation is purely formal since \( f(u) \) is not generally a probability density. However, when

\[ g_1(u) = 0, \quad |u| \geq \frac{\pi}{M} \]  \hspace{1cm} (35)

then \( f(u) \) is a probability density because in that case

\[ \frac{1}{M} g_s(R(u)) = g_m(u). \]  \hspace{1cm} (36)

Thus (34) can be interpreted as an approximate density when the peak of \( g(u) \) is narrower than the minimum phase distance between the symbols. This condition is always satisfied in communications applications; otherwise, phase distortion is so large that data transmission is not possible. Thus we have a pure phase-tracking problem as in [8] and [9], and we may proceed accordingly. Taking the natural logarithm of \( f^K \), we have the maximization problem

\[ \max \Gamma_K, \quad (\psi_1)^1 \]

\[ \Gamma_k = \Gamma_{k-1} + p_k; \]

\[ \Gamma_i = -\frac{1}{2\sigma^2_s} |x_i - e^{i\psi_i}|^2 + \ln g_1(R(\psi_i)) \]

\[ p_k' = -\frac{1}{2\sigma^2_s} |x_k - e^{i\psi_k}|^2 + \ln g_1(R(\psi_k - \psi_{k-1})) \]  \hspace{1cm} (37)

which is solved by the dynamic programming algorithm discussed in Section IV. The complexity \( c' \) of this algorithm lies essentially in the evaluation of the \( m^2 \) possible values of \( |x_k - e^{i\psi_k}|^2 \) for each new data value \( x_k \). The \( m \) different values of \( \ln g_1(R(\cdot)) \) will be precomputed and stored in ROM. For each computation of \( |x_k - e^{i\psi_k}|^2 \) there are two multiplies, so complexity is simply proportional to \( m \). This represents a reduction in complexity greater than \( M \) for \( M \)-ary PSK.

Usually, the phase is differentially modulated rather than directly modulated, and therefore the relevant symbol is \( \Delta \delta_k \) itself (see (30)). For the purpose of data transmission there is no need to reconstruct the absolute data phase \( \delta_k = \sum_{i=1}^{k} \Delta \delta_i \). This reconstruction has, however, been carried out in the simulations in order to recover the estimates \( \hat{\psi}_k = \hat{\psi}_k - \hat{\delta}_k \) of the phase fluctuations and to get the approximate variance of the phase estimates

\[ \sigma^2_k = \frac{1}{K} \sum_{k=1}^{K} |\hat{\psi}_k - \hat{\delta}_k|^2. \]  \hspace{1cm} (39)

### Density Functions and Geometrical Comments

The entire development of this section has a nice geometric interpretation which we illustrate in Fig. 5. In Fig. 5(a) the basic phase noise density \( h(x) \) is illustrated on
(−∞, ∞). Fig. 5(b) is the folded version g₁(x) of h(x) to account for the wrapping on the unit circle C. Fig. 5(c) is the function g₁[R(x)] that arises in our discussion of the principle of optimality, sketched in the case of 4-ary phase modulation. Fig. 5(d) shows g₁[R(x)] wrapped around the circle C. Since g₁(x) is very narrow, g₁[R(x)] is approximately the repeated copy of g₁(x) at all possible values of data phase. With k = ψₖ − ψₖ₋₁, Fig. 5(d) illustrates the choice of ∆θₖ nearest ψₖ − ψₖ₋₁ (∆θₖ = π/2 is the best choice here), and the resulting value of g₁[R(ψₖ − ψₖ₋₁)] is shown by the heavy segment on the axis ψₖ, terminated by the heavy dot.

We now extend this principle of optimality to phase-amplitude encoded symbols. Assume the independent, equally probable data symbols are complex symbols of the form

\[ a_k = A_k e^{j\theta_k} \]  

(40)

with the \( A_k \) positive real numbers drawn independently from the alphabet \( A = (a_1, a_2, \ldots, a_M) \). Denote by \( p(A_k) \) the probability mass function for the random variable \( A_k \). Assume the \( \theta_k \) are drawn from the alphabet \( B = (\beta_1, \beta_2, \ldots, \beta_M) \). Denote the conditional probability mass function of \( \theta_k \), given \( A_k \), by \( p(\theta_k/A_k) \). For the (4, 4) diagram of Fig. 2(d),

\[ A = (\sqrt{2} a_1, 3a_1, 3\sqrt{2} a_1, 5a_1) \]

\[ B = (b_1)^{-1}, b_i = (i - 1)^2/4. \]

The probabilistic description of the source is

\[ p(A_k) = 1/4, \quad \text{for all } A_k \]

\[ p(\theta_k/A_k = a_1) = \begin{cases} 1/4, & \theta_k = \beta_2, \beta_4, \beta_6, \beta_8 \\ 0, & \text{otherwise} \end{cases} \]

\[ p(\theta_k/A_k = a_2) = p(\theta_k/A_k = a_3) \]

\[ p(\theta_k/A_k = a_4) = p(\theta_k/A_k = a_5). \]

In place of the maximization problem posed in (23), we write

\[ \max_{(\psi_k, \Delta \theta_k, A_k)} f(\{x_k\}^*_K, \{\psi_k\}^*_K, (\Delta \theta_k)^*_K, (A_k)^*_K) \]

(41)

with \( \psi_k \) and \( \Delta \theta_k \) defined as in (22). The density \( f^{K*(\ldots, \ldots)} \) appearing in (42) may be written

\[ f^K = \prod_{k=1}^K N_x(A_k e^{j\psi_k}, \sigma_k^2) \]

\[ f(\{\psi_k, \Delta \theta_k, A_k\}/\{\psi_k\}^{K-1}, (\Delta \theta_k)^{K-1}, (A_k)^{K-1}). \]  

(43)

The conditional density on the right-hand side of (43) is simply

\[ f(\psi_k, \Delta \theta_k, A_k) = g(\psi_k, \psi_k - \Delta \theta_k)p(\Delta \theta_k/A_k) \]

(44)

where \( p(\Delta \theta_k/A_k, A_{k-1}) \) is the conditional probability mass function for \( \Delta \theta_k \), given \( A_k \) and \( A_{k-1} \). Putting (43) and (44) together, we have as the joint density function to be maximized

\[ f^K = \prod_{k=1}^K N_x(A_k e^{j\psi_k}, \sigma_k^2) g(\psi_k, \psi_k - \Delta \theta_k) \]

\[ \cdot p(\Delta \theta_k/A_k, A_{k-1}) p(A_k). \]  

(45)

It is important to note in this expression that the \( N(\cdot, \cdot) \) term is dependent only on the measurement model; \( g(\cdot) \) is dependent only on the random phase model, and \( p(\Delta \theta_k/A_k, A_{k-1}) \) is dependent only upon the symboling constellation (or encoding scheme). Thus (45) is a useful canonical decomposition that is generally applicable to communications problems involving additive independent noise and independent increments phase processes.

For the (4, 4) diagram of Fig. 2(d) we may compute \( p(\Delta \theta_k/A_k, A_{k-1}) \) as follows:

\[ p(\Delta \theta_k/A_k = a_i, A_{k-1} = a_j) = \begin{cases} 1/4, & \Delta \theta_k = \beta_i, \beta_j, \beta_3, \beta_7. \\ i, j \text{ even-even or odd-odd} \end{cases} \]

(46)

It is straightforward to substitute these results into (45) and derive a path metric as in (37).

VI. LINEAR PERFORMANCE RESULTS AND THE SELECTION OF A FIXED LAG

There is one more simplification to be made: namely, the selection of a depth constant \( k_0 \) such that phase-symbol pairs may be decoded at a fixed-lag \( k_0 \), thereby obviating the need to store long survivor sequences. Call \( \{\hat{\psi}_{k/K}\} \) the MAP phase sequence based on measurements \( \{x_k\}^K \). The subscript \( k/K \) indicates that \( \hat{\psi}_{k/K} \) depends on all measurements up to time \( K \). In general the MAP sequence \( \{\hat{\psi}_{k/K}\}^K \) based on measurements to time \( (K + 1) \) may differ from \( \{\hat{\psi}_{k/K}\}^K \) at all values of \( 1 \leq k \leq K \). However, one expects that for large \( K \) and for \( k \leq K - k_0 \), the sequences \( \{\hat{\psi}_{k/K}\}^K \) and \( \{\hat{\psi}_{k/K}\}^K \) will not be very different for a well-chosen depth \( k_0 \). In other words, long survivor sequences tend to have one common trunk up to \( K - k_0 \), at which point they may diverge as illustrated in Fig. 6. Thus we may use \( \hat{\psi}_{K-k_0/k} \) as a final estimate of \( \hat{\psi}_{K-k_0} \), since \( \hat{\psi}_{K-k_0/k} \) is the same phase value, \( k_0 \) samples back, in the MAP sequence based on measurements up to time \( K \). In this way, phase values are estimated with delay \( k_0 \) and only survivor sequences of length \( k_0 \) must be stored.

How should \( k_0 \) be chosen? This is a difficult question to answer precisely, because no analytical results exist for the performance of nonlinear phase trackers of the Viterbi-type. We can, however, study the filtering behavior of a related linear problem and find how performance varies with
fixed-lag \( k_0 \). To this end, we consider the problem of tracking phase when there is no data symboling. Assume \( \{ \psi_k \} \) is a normal random walk of the form (6) with \( \psi_k : N_0(0, \sigma^2) \). Let \( x_k = e^{j\psi_k} + n_k \), \( \{ n_k \} \) be a sequence of complex random variables whose real and imaginary parts are i.i.d. \( N_0(0, \sigma^2) \) random variables. A PLL with gain \( K_1 \) for estimating \( \{ \psi_k \} \) is the following:

\[
\hat{\psi}_k = \hat{\psi}_{k-1} + K_1 |x_k| \sin(\arg x_k - \hat{\psi}_{k-1}).
\]  

(47)

Note that this is similar to (13) when there is no data.

For \( \sigma^2 \equiv 1 \) we approximate (47) with

\[
\hat{\psi}_k = \hat{\psi}_{k-1} + K_1 (\arg x_k - \hat{\psi}_{k-1}).
\]  

(48)

When \( K_1 \) is selected to be

\[
K_1 = ( \sigma^2 / \sigma^2 ) \left[ -0.5 + 0.5 (1 + 4 \sigma^2 / \sigma^2) \right]^{1/2}.
\]

(49)

then (48) is the Kalman filter for the "linear observation model"

\[
\arg x_k = \psi_k + n_k - x_k = \exp \left[ j(\psi_k + n_k) \right].
\]  

(50)

The steady-state filtering error \( P_0 \) for this linear problem is related to \( K_1 \) as follows:

\[
K_1 = \frac{\sigma^2}{\sigma^2} \cdot \frac{P_0}{\sigma^2}.
\]  

(51)

A general result due to Hedelin [12] for fixed-lag smoothing may be adapted to random walk smoothing from observations of the form (50). The steady-state fixed-lag smoothing variance \( P_\infty \) at delay \( k_0 \) is

\[
P_{k_0} / \sigma^2 = P_0 / \sigma^2 - \sum_{l=1}^{k_0} G^{2l}
\]

\[
= P_0 / \sigma^2 - G^2(1 - G^{2k_0}) / (1 - G^2)
\]

\[
G = 1 - K_1.
\]  

(52)

The infinite-lag smoothing variance is

\[
P_\infty / \sigma^2 = P_0 / \sigma^2 - G^2 / (1 - G^2).
\]  

(53)

In Fig. 7 several error expressions and asymptotic forms are plotted versus \( \sigma^2 / \sigma^2 \), which is a kind of SNR. For large \( \sigma^2 / \sigma^2 \), the error variances \( P_0 / \sigma^2, P_{10} / \sigma^2, \) and \( P_\infty / \sigma^2 \) go as \( (\sigma^2 / \sigma^2)^{-1} \). For small \( \sigma^2 / \sigma^2 \), they go as \( (\sigma^2 / \sigma^2)^{-1/2} \) although infinite-lag smoothing offers 6 dB improvement in \( \sigma^2 / \sigma^2 \) over zero-lag smoothing for a fixed smoothing variance. Over the range of values 0.01 \( \leq \sigma^2 / \sigma^2 \leq 10 \), a delay of \( k_0 = 10 \) offers all but 1–2 dB of the theoretically achievable gain from infinite delay. In communication problems for which random phase is a significant effect, the ratio \( \sigma^2 / \sigma^2 \) is typically in this range. Only at very small values of \( \sigma^2 / \sigma^2 \) can very large delays \( k_0 \) provide large performance gains, but in this case there is no real phase fluctuation problem for the purpose of data decoding, and the gain is not worth the large delay. Shown also in Fig. 7 is the Kalman gain \( K_1 \) versus \( \sigma^2 / \sigma^2 \).

The problem considered in Section IV is admittedly different from the linear problem considered here. However, the numerical results given in Fig. 8 for the Viterbi phase tracker illustrate that the performance gain to be achieved with a fixed-lag of \( k_0 = 10 \) is much as predicted by the linear theory. For the results of Fig. 8, the phase space was discretized to \( m = 48 \) values, data transmission was 8-ary PSK, and the decoding algorithm was the VA. The circles, dots, and squares represent experimental phase estimation error variances, and the heavy solid lines represent theoretical results. Over the range of values 0.1 \( \leq \sigma^2 / \sigma^2 \leq 2 \), the phase estimator variance for the Viterbi phase tracker operating with delay \( k_0 = 10 \) is essentially equivalent to the filtering variance of a Kalman filter that has access to linear observations and provides estimates without delay. Performance is not measurably degraded by the presence of data which are concurrently decoded.

VII. SIMULATION RESULTS: GAUSSIAN INCREMENTS

For all simulation results discussed in this section the phase space \((-\pi, \pi)\) has been discretized to 48 equally spaced phase values and a Viterbi algorithm has been programmed to solve the MAP sequence estimation problem. The principle of optimality established in Section V has been used to derive the appropriate path metric and thereby reduce computational complexity. The choice of a fixed-lag decoding (or depth) constant is \( k_0 = 10 \). Source symbols have been generated independently. The random phase sequence has been governed by the independent increments model of (6) with \( \psi_k : N(0, \sigma^2) \) and initial phase uniformly distributed on \([-\pi, \pi)\). Initial phase acquisition has been achieved by transmitting a preamble according to one of the following schemes.

a) During a pretransmission period of length \( N \), the sequence of transmitted data is known to the receiver. Thus in the DBVA and VA systems, based upon MAP estimation, the Viterbi algorithm works as a pure phase estimator during this period. At the end of the preamble, the Viterbi algorithm is turned into a joint phase-data MAP estimator. In the DDPLL and JE systems, based upon decision-directed algorithms, the algorithm is directed by the true data during the preamble period.

b) During the preamble period, identical (but unknown) data are transmitted. This keeps the phase from making phase jumps associated with symbol changes and makes the joint phase-data estimator able adequately to acquire the initial phase.

In our simulations the VA has achieved the same data-error probability for both methods; i.e., its performance
has not depended upon which learning procedure was used. On the other hand, Ungerboeck's DBVA have proved to be sensitive to the learning procedure. For example, at SNR = 20 dB with phase variance $\sigma_n^2 = 4 \sigma_a^2$ for a learning period of $N = 60$ data, the number of errors during a transmission period of 490 data values has jumped from seven for procedure a)—known data—to 59 for procedure b)—constant but unknown data. Moreover, the DBVA typically requires a longer learning period than does the VA (roughly twice longer). A value of $N = 50$ is sufficient for the VA, while the DBVA needs $N = 100$ learning iterations in our simulations. The decision-directed systems (DDPLL and JE) work as the VA in these respects. That is, a preamble period of 50 data values is sufficient. These data may be unknown to the receiver, provided they are kept constant (procedure b)). No degradation with respect to procedure a) results.

**Binary Symboling**

Shown in Fig. 9 are binary symboling results for the VA when $\sigma_n^2 = 0.01 \text{ rad}^2 (\sigma_a = 5.7^\circ)$ and SNR ranges from 4 to 10 dB. (Recall SNR = 10 log$_{10} \frac{1}{2\sigma_n^2}$.) The results indicate that performance with the VA is essentially equivalent to that of a fully coherent receiver, even for a relatively large value of $\sigma_n$. For comparison, the curves for coherent binary orthogonal and coherent binary antipodal systems are also shown. The simulation results for binary orthogonal symboling are interesting because they serve to validate the simulation. Indeed, as expected, the performance of the VA is seen in Fig. 9 to lie between that of an incoherent receiver and that of a fully coherent receiver. Of course, the margin between coherent and incoherent performance is small at SNR's of practical interest. The simulation results for binary antipodal symboling are interesting on their own.
because incoherent reception is not possible with antipodal symboling.

**Eight-PSK**

Shown in Fig. 10 are simulation results for eight-PSK when SNR ranges from 16–19 dB and \((\sigma_0^2/\sigma_s^2)^{1/2}\) remains fixed at \(4.4 \times 10^{-3}\) rad\(^2\). This choice of parameters corresponds to a loop SNR of 23 dB where the PLL is well into its linear region of operation and little can be gained from improvements to the phase tracking. The values of \(\sigma_0^2\) under investigation range from 1.6\(^\circ\) to 2.2\(^\circ\) and the ratio \(\sigma_0^2/\sigma_s^2\) is very small, ranging from 0.03 to 0.12. The solid circles of Fig. 10 correspond to the VA, and the solid triangles correspond to the markedly simpler JE. Also shown in Fig. 10 are performance bounds for fully coherent eight-PSK and 16-PSK symboling. In this case neither the VA nor the DBVA provides significant improvement over the JE or DDPLL. The latter two receivers are simpler than the DBVA which, in turn, is simpler than the VA. Therefore, for such cases of weak phase noise, neither the VA nor the DBVA would be favored over the JE or the DDPLL.

**16-QASK**

Shown in Fig. 11–13 are simulation results for 16-QASK symbols encoded according to the \((4, 4)\) CCITT rule. The decoding procedure are JE, DDPLL, DBVA, and VA, for three distinct values of the ratio \(\sigma_0^2/\sigma_s^2\): Fig. 11 is concerned with a weak phase noise \((\sigma_0^2/\sigma_s^2 = 0.25)\). Fig. 12 is concerned with an average phase noise \((\sigma_0^2/\sigma_s^2 = 1)\), and Fig. 13 is concerned with a large phase noise \((\sigma_0^2/\sigma_s^2 = 4)\). We recall [1] that the DBVA performs some kind of phase estimation along a path that satisfies

\[
\hat{\psi}_n = \hat{\psi}_{n-1} \pm \sigma_n, \tag{54}
\]

using a Viterbi algorithm. The DBVA that we have simulated is somewhat different from Ungerboeck’s DBVA, in which the number of possible phase states at each iteration is limited to six or eight. In our simulation the number of phase states is not limited, thus avoiding one possible cause of errors and improving the error rate, but also increasing the computational complexity with respect to [1].
Behavior of DDPLL and JE on CCITT (4, 4) Constellation

The decision-directed algorithms (DDPLL and JE) have essentially the same performance, as shown in Figs. 11–13. The DDPLL is superior to the JE by only 0.5 dB. The slight inferiority of the JE is largely compensated by the fact that the complex gain of the JE can also correct rapid gain fluctuations in the channel. We emphasize that the curves of the DDPLL and JE are biased and cannot be trusted just as they are because of the occurrences of very large bursts of errors at relatively high error probabilities. When such bursts have occurred in the simulation runs, they have been withdrawn from the error rate computation. For instance, with $a_4^2 = 0.25a_2^2$ and SNR = 17 dB, at an error probability on the order of $10^{-2}$, between one fourth and one third of the simulation runs, there is no gain when $\theta_2$ is monotone increasing function of $a_2^2$. When errors are grouped in twos or threes, and no error multiplication occurs since the phase estimator is not decision-directed. Thus such MAP sequence estimators can be used even at high error probabilities on the order of $10^{-2}$ or $10^{-1}$.

<table>
<thead>
<tr>
<th>Percent bursts</th>
<th>0</th>
<th>7</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_\text{E}$ (Isolated Error)</td>
<td>$&lt; 10^{-4}$</td>
<td>$2.0 \times 10^{-3}$</td>
<td>$1.5 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Comparison Between MAP and Decision-Directed Phase Estimators

The improvement that can be gained by using any type of MAP estimator for phase rather than a simple decision-directed algorithm is again an increasing function of $a_4^2/a_2^2$. Fig. 11 shows that only 1 dB is gained by the DBVA and the VA over the DDPLL if $a_4^2 = 0.25a_2^2$. This gain is realized at a high computational price. For the phase fluctuations and additive noise of the same importance ($a_4^2/a_2^2 = 1$), the VA outperforms the DDPLL by 3 dB (see Fig. 12), but the gain is reduced to 2 dB for the simpler DBVA. For large phase fluctuations, the gain is important. For instance, Fig. 13 shows that the VA outperforms the DDPLL by 5 dB when $a_4^2/a_2^2 = 4$. In addition, the VA brings the insurance that no burst of errors can occur, even for very poor SNR and large phase fluctuations. In fact, the true power gain of the VA over a decision-directed algorithm is even higher than just claimed if one takes account of the additional power required in the decision-directed schemes to ensure against burst as well as random errors.

Sensitivity to Imperfect Knowledge of $a_4^2/a_2^2$

It is easily seen in (18) or (37) that the only parameter required in order to proceed with the VA algorithm is the ratio of phase variance to additive noise power. The same holds for the DDPLL whose optimal gain $K_1$ depends on this ratio (see (49)), and for the JE whose step-size $\mu$ (see (14)) is to be kept close to $K_1$, but smaller, provided the data diagram has unit power. As for the DBVA, it requires only the knowledge of $a_4^2$ in order to determine the number $m$ of discretized phase levels. Thus an important feature of each system is its sensitivity to an imperfect knowledge of $a_4^2/a_2^2$ (or $a_4^2$) because, first, $a_4^2$ can vary with time and, second, the actual phase can fluctuate according to a statistical model that is different from the one expected. The less sensitive the system is to the knowledge of $a_4^2/a_2^2$ (or $a_4^2$), the more robust it is.

a) Sensitivity of the Decision-Directed Systems: Let us denote $a_4^2/a_2^2$ by $\alpha$. The function $K_1(\alpha)$ that gives the optimum loop-gain of the DDPLL is sketched in Fig. 14. It is quite flat except for $\alpha$ very close to zero (e.g., $\alpha < 0.2$).

Now the case $\alpha < 1$ is of no real interest for the purpose of this paper. Indeed, it has been seen previously that, in

Behavior of DBVA and VA on CCITT (4, 4) Constellation

The performance of the VA is superior to that of the DBVA. The gain achieved by the VA over the simpler DBVA is monotone increasing in the ratio of phase fluctuation variance $a_4^2$ to additive noise variance $a_2^2$. While there is no gain when $a_4^2/a_2^2 = 0.25$, the gain is 1 dB for $a_4^2/a_2^2 = 1$ and 2 dB for $a_4^2/a_2^2 = 4$. Both systems perform better than the DDPLL or JE, the improvement again being a monotone increasing function of $a_4^2/a_2^2$.

A very important point is that the use of either of the two MAP phase estimators precludes the occurrence of error bursts. The errors seem to be grouped in twos or
this case, no MAP phase estimator is worth being worked out. Moreover, any reasonable phase estimator will perform satisfactorily. When $\alpha$ is not negligible, $K_i(\alpha)$ is slowly varying. For example, $K_i(1)/K_i(0.25) = 1.59$, and $K_i(4)/K_i(1) = 1.34$. Thus the value $K_i(1) = 0.62$ for the DDPLL gain is correct for a large range of values of $\alpha$. This fact is largely confirmed by the simulations. Hence, due to the risk of error multiplication that increases very rapidly with $K_i$, it should rather be set to the lower bound $K_i(\alpha_{\text{min}})$ corresponding to the smallest $\alpha$ that can be expected, rather than to an average value $K_i(\alpha_{\text{avg}})$, which will sometimes be too large and bring error bursts. Thanks to this precaution, the DDPLL is insensitive to $\alpha$. It is a robust system.

The robustness of the JE is also excellent. This fact was checked on numerous computer simulations: as a function of the step size $\mu$, the error probability $P(E; \mu)$ exhibits a minimum which is very flat, as sketched in Fig. 15. The range where the minimum is reached does not depend critically upon $\alpha$. A value such as $\mu = 0.4$ corresponds to the minimum of error probability for $\alpha$ in the range $[0.25-1]$ and for a unity energy data diagram.

b) Sensitivity of the MAP Phase Estimators: The VA sensitivity to imperfect knowledge of $\alpha$ has been tested in our computer simulations. It appears that the VA performance is not appreciably degraded by an error of $\pm6\text{dB}$ for $\alpha$. Hence the VA robustness is at least as good as that of the decision-directed algorithms.

On the other hand, the DBVA robustness has turned out to be poor. For instance, with SNR = $21\text{dB}$ and $\alpha = 4$, the DBVA is supposed to work with $m = 2\pi/\sigma_\text{w} = 50$ phase levels. If only 45 levels are used, corresponding to a $0.9\text{dB}$ error for $\alpha$, then the error probability is increased by a factor of two. In fact, as a function of $m$, $P(E; m)$ exhibits a minimum, but it is a sharp minimum. This poor robustness can be understood by noting that in the DBVA, the path metric is not a function of $\alpha = \sigma_\alpha^2/\sigma_\text{w}^2$; but only of $\sigma_\alpha^2$. This may be one of the main drawbacks of the DBVA.

VIII. SIMULATION RESULTS: BOUNDED-INCREMENTS PHASE JITTER

For all simulation results of this section the phase space $[-\pi, \pi]$ has been discretized to 32 equally spaced phase values, and a VA has been programmed to solve (17). The assumed increment density $h(w)$ is the uniform density

$$h(w) = \frac{1}{2a}, \quad -a \leq w < a; \quad 2a = 2\pi/16 \quad (55)$$

The corresponding discrete transition density for use in the path metric is

$$f(\phi_k/\phi_{k-1}) = \begin{cases} 1/3, & \phi_k - \phi_{k-1} = -\pi/16, 0, \pi/16 \\ 0, & \text{otherwise}. \end{cases} \quad (56)$$

The resulting VA is related to the class of so-called $M$ algorithms [16]--[18] in which all survivors are saved, but only $M$ (in this case 3) candidate originator states are allowed. This significantly reduces calculations and results in an algorithm similar in spirit to the DBVA of [1]. Still, however, phase is tracked only on $[-\pi, \pi]$ rather than on $(-\infty, \infty)$.

Source symbols have been generated independently from a four-PSK alphabet and used to differentially encode phase according to a Gray code. The random phase sequence has been generated in ways to be discussed below.

Markov Phase with Non-Gaussian Increments

Here the phase is generated according to (6) with $h(w)$ given by (55). Thus the algorithm is matched to the actual phase sequence. Shown in Fig. 16 are performance results for the VA and for the JE. The VA outperforms the JE by $1.5\text{dB}$ over the range $10\text{dB} \leq \text{SNR} < 15\text{dB}$. The probability of error is "probability of bit error."

Sinusoidal Phase Jitter

Here the phase jitter is sinusoidal (see (4)) with uniformly distributed initial phase and frequency $\nu$. The frequency is chosen such that $\nu \Delta = 1/24$, corresponding to a transmission rate of 4800 bits/s with baud rate $1/\Delta = 2400\text{Hz}$ and jitter frequency $\nu = 100\text{Hz}$. The runs are 2000--10,000 steps long, corresponding to 4000--20,000 transmitted bits. The peak-to-peak phase deviation is 20° or 60°. For these experiments the VA outperforms the JE by $1.5$--$1.7\text{dB}$. This gain is, of course, achieved at a high price in complexity.
Comparison of the JE and VA

In the simulations reported above, the ratio $\alpha = \sigma_2^2/\sigma_2^2$ ranges from 0.02 to 0.81, that is from small to average values. No burst of errors has ever been observed for the JE. This is due to the fact that the phase increment is always bounded as appears in (4) and also (55). The bound is much smaller than the angular distance between adjacent data. Thus there is no risk of a $\pm 90^\circ$ slip (corresponding to the four-PSK diagram) in the JE phase estimation. Hence the errors will be scattered rather than grouped, and no error multiplication phenomenon can happen.

Owing to this consideration, to the fact that the VA outperforms the JE by only 1.5 dB, and to the complexity of the VA, a practical system will implement the JE (or DDPLL) rather than the VA (or DBVA), in the case of bounded increment phase jitter.

IX. CONCLUSION

We have derived a principle of optimality for phase-amplitude encoded symboling that allows one to simultaneously track random phase and decode data symbols using the VA derived in [8] and [9]. The VA is designed for a random walk phase process, a very severe type of phase process. In such a process there exists the possibility of large phase jumps. The VA gives excellent performance because it benefits from the use of a lag to observe future data samples which make large phase jumps look unlikely.

In order to reach conclusions about the type of phase estimation that should be used for given types of phase fluctuations, performance comparison of the VA with two simple decision-directed (zero-lag) phase estimators, namely, the JE of [3] and the DDPLL of [5], and with the DBVA of [1], have been thoroughly investigated by computer simulations, with various data diagrams. They indicate that the choice among the four systems is to be made according to four parameters:

1) the error probability $P(E)$ at which the system is to be used;
2) the relative importance $\alpha = \sigma_2^2/\sigma_2^2$ of phase fluctuations with respect to additive noise;
3) the complexity $c$ that is technologically feasible and acceptable;
4) the maximum phase increment $\Delta \Phi_{\text{max}}$ that is to be expected, as compared to the angular distance between points of the data diagram.

Suggestions for this choice are sketched in Tables II and III where Table III is concerned with cases 2 and 3 of Table II.

The choice between the two decision-directed phase estimators, JE or DDPLL, is irrelevant for the matters discussed in this paper. It appears in Tables II and III that the VA and DBVA are preferred when $\alpha$, $P(E)$, and $\Delta \Phi_{\text{max}}$ are large. The comparison between these two MAP phase estimators shows that the VA is more robust, has a smaller learning period, and outperforms the DBVA by 2 dB or more when $\alpha$ is at least equal to four.

Only Viterbi, or Viterbi-like, algorithms can survive and correct error bursts by effectively using the weight of future evidence to render such bursts too unlikely to occur. Thus it seems likely that the VA (really dynamic programming) will grow in importance in such applications as spread spectrum communication where the phase of a wide-band carrier can be tracked for symbol decoding.

ACKNOWLEDGMENT

The authors wish to thank S. Kerbrat and C. Pariente for their assistance with software development and simulations.
APPENDIX

MONOTONICITY OF FOLDED NORMAL AND CAUCHY DENSITIES

There are many choices for the phase increment density \( h(x) \) that are physically interesting and mathematically tractable. Two of particular interest are the normal density and the Cauchy, the latter being useful in the modelling of "heavy-tailed" behavior. When folded around the unit circle according to (8) these densities yield transition densities which achieve their maximum at \( \phi_k - \phi_{k-1} = 0 \) and decrease monotonically on the interval \( 0 \leq \phi_k - \phi_{k-1} \leq \pi \).

Consider first the Cauchy case

\[
g_1(x) = \sum_{k=-\infty}^{\infty} \frac{e^{x}}{\pi a^2 + (x + k\pi)^2}.
\]  

(57)

According to Poisson's summation formula [13], this may be written

\[
g_1(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} e^{-x/2} e^{ikx}
\]

\[
= \frac{1}{2\pi} (1 - e^{-x})(1 - 2e^{-x} \cos x + e^{-2x})^{-1}.
\]

(58)

This function achieves its maximum value at zero and decreases monotonically.

In the normal case

\[
g_1(x) = \sum_{k=-\infty}^{\infty} (2\pi a^2)^{-1/2} \exp \left\{- (x + 2k\pi)^2 / 2a^2 \right\}.
\]

(59)

Again, by Poisson's summation formula,

\[
g_1(x) = \sum_{k=-\infty}^{\infty} (2\pi)^{-1} \exp \left\{ jkx - k^2 a^2 / 2 \right\}.
\]

(60)

This infinite sum goes by the name \( J_q(x, q) = e^{-x^{1/2}} \) in the theory of Jacobi elliptic functions and theta functions [15]. The theta function \( J_q(x, q) \) is known to be monotonically decreasing on the interval \( 0 \leq x \leq \pi \).

REFERENCES

Aspects of Dynamic Programming in Signal and Image Processing

LOUIS L. SCHARF, SENIOR MEMBER, IEEE, AND HOWARD ELLIOTT, MEMBER, IEEE

Abstract—The techniques peculiar to dynamic programming have found a variety of successful applications in the theory and practice of modern control. Successes in the theory and practice of signal and image processing are less numerous and prominent, but they do exist. In this paper, we sound a call for renewed attention to the potential of dynamic programming for solving knotty nonlinear filtering problems in signal and image processing, and outline successes we have recently enjoyed in nonlinear frequency tracking and random boundary estimation in noisy black and white images. Two classical results, the fast Fourier transform and Levinson's recursion for determining autoregressive parameters, are treated in the context of dynamic programming simply to reinforce the point that many of the algorithms we take for granted, and which were derived without recourse to dynamic programming, can be nicely interpreted as dynamic programming algorithms.

I. INTRODUCTION

IN THIS PAPER it is our aim to show that dynamic programming, a fundamental technique in control theory since Bellman's introduction and advocacy of it in the mid-1950's, can be of considerably more value in signal and image processing than has generally been recognized. This is not to say others have failed to recognize the potential of dynamic programming for solving interesting signal processing problems. We mention in particular Cox's early work [1], [2] on Kalman filtering and dynamic programming for the estimation of state variables and the identification of system parameters; Viterbi's dynamic programming algorithm for decoding convolutional code sequences [3]; Cahn's dynamic programming algorithm for FM demodulation [4]; and Forney's discussion of inference problems on finite-state Markov sequences that can be solved with the techniques of dynamic programming [5].

In the sections to follow we rederive classical algorithms in discrete Fourier analysis and linear prediction using the principle of dynamic programming. We then present two new dynamic programming algorithms. One is for nonlinear frequency tracking and the other is for edge detection in noisy black and white images.

The organization is as follows. In Section II, we present an elementary dynamic programming formalism. In Section III we use dynamic programming arguments to rederive the Goertzel and decimation-in-frequency fast Fourier transform (FFT) algorithms for efficiently computing the discrete Fourier transform (DFT). In Section IV, we discuss the connections between control, detection, estimation, and prediction of autoregressive sequences observed in additive noise. We highlight the central role played by the so-called normal equations and rederive the Levinson algorithm for recursively solving them in the order of $p^2$ operations. The derivation is a dynamic programming one.

The new results follow in Sections V and VI. In Section V, a dynamic programming algorithm for tracking the frequency of a frequency modulated sequence in additive noise is derived. Several simulations illustrate the performance of the algorithm. This provides a solution to a classical nonlinear filtering problem. The results of Section VI show how dynamic programming may be used to derive a new algorithm for estimating local segments of object boundaries in noisy black-and-white images. Some examples are given to illustrate the use of the algorithm in estimating complete object boundaries as well.

II. A DYNAMIC PROGRAMMING FORMALISM

Traditionally, dynamic programming has been used to find "optimum" solutions to multistage decision problems [6], [7]. An "optimum" solution has generally been one that maximizes or minimizes a performance or cost functional. When the multi-stage decision problem is cast in a probabilistic framework and the criterion of optimality is maximum a posteriori (MAP) probability, then the cost functional is typically a multivariable likelihood function or some monotone function of it.

The following is a formalism that is rich enough to embrace most of the "signal-in-noise" problems encountered in signal and image processing. Let \( \{x_k\}_{k=0}^{\infty} \) denote a process with state variable representation

\[
x_{k+1} = f_k(x_k, u_k)
\]

\[
y_k = g_k(x_k).
\]

Here \( f_k \) and \( g_k \) may be random functions; the sequence \( \{u_k\} \) is a parameter, decision, or control sequence that may be functionally dependent on the measurement sequence \( \{y_k\} \). The range spaces for the state \( x_k \), the parameter \( u_k \), and the measurement \( y_k \) are \( X, U, Y \), respectively. These
spaces may be finite, countable, or noncountable. When the spaces $X$ and $U$ are countable then their respective elements may be placed in one-to-one correspondence with the integers and the formalism of Markov chain theory may be used. Even though the states of $X$ may be chosen to appear uninteresting, the mapping $g_X$ may be chosen so that the signal component of $g_X(\cdot)$ generates characters or observations $C_X$ that are of great interest. The idea is simply to let a Markov chain, say on the integers spectrum (so to speak) of signal processing applications, control the dynamical state of the problem and reserve the role of character or observation generation for the observation mechanism $g(\cdot)$. This point is illustrated in Fig. 1 where the generated characters can be almost anything: contours, sequences, images, etc.

Consider a finite version of the process $(x_k)_{k=0}^\infty$:

$$X_N = (x_0, x_1, \ldots, x_N) = F_N(X_{N-1}, U_{N-1})$$

$$U_N = (u_0, u_1, \ldots, u_N)$$

$$Y_N = (y_0, y_1, \ldots, y_N).$$

Typically, one wants to maximize a performance criterion

$$I_N(X_N, U_N, Y_N)$$

with respect to $U_N$, subject to constraints $C_N(X_N, U_N) = 0$. Call $I_N^*(X_N, U_N, Y_N)$ the maximum. When $I_N^*$ obeys a recursion of the form

$$I_N^*(X_N, U_N, Y_N) = I_{N-1}^*(X_{N-1}, U_{N-1}^*, Y_N) + P_N(x_N, U_{N-1}^*, U_N, Y_N)$$

then dynamic programming comes to the fore and the solution $U_N^*$ may be generated recursively as the limit of the following sequence of solutions:

$$U_n^* = S_n(U_{n-1}^*, Y_n), \quad n = 1, 2, \ldots, N.$$  \hspace{1cm} (5)

The functional $S_n$ describes the recursion for computing $U_n^*$. Thus the central theme is to imbed the solution to an $N$ stage problem in a sequence of simpler $n$ stage problems. When the underlying state and parameter spaces are finite, the solution algorithm is finite-dimensional and implementable on a digital computer. When they are uncountable, but the function $l$ is quadratic, then it is still often possible to find a closed-form recursive solution that may be programmed.

A very large class of problems may be formulated as above. Two particularly noteworthy examples are the linear discrete-time quadratic regulator problem in deterministic and stochastic control, and Markov chain sequence estimation in additive noise. On the other hand, there are a great number of problems that admit dynamic programming solutions, but which are not naturally formulated in the style above.

One of the points we wish to make is the following: recognizing that a solution is a limit of a sequence of approximants which may be recursively computed is perhaps more fundamental than the search for a corresponding optimization problem. The chief value of an optimization formulation is that it often simplifies the search for the recursive solution algorithm.

### III. Dynamic Programming, the DFT, and the FFT

The DFT certainly constitutes one of the cornerstones of modern Fourier analysis. Its uses range over the entire spectrum (so to speak) of signal processing applications. The DFT is a mapping, DFT: $(x_n)_{n=0}^{N-1} \rightarrow (X_m)_{m=0}^{N-1}$, that takes the sequence $(x_n)_{n=0}^{N-1}$ into the sequence $(X_m)_{m=0}^{N-1}$ according to the rule

$$X_m = \sum_{k=0}^{N-1} x_k W_n^m, \quad m = 0, 1, \ldots, N-1,$$

$$W_N = \exp(-j2\pi/N).$$

Noting that $W_N^{-m} = 1$, $\forall m$, we may write $X_m$ as follows:

$$X_m = \sum_{n=0}^{N-1} x_n W_n^{-m(n-m)}. \quad (7)$$

This calculation may be viewed as the limit of the following sequence of imbedded approximations:

$$X_m^{(k)} = \sum_{n=0}^{k-1} x_n W_n^{-m(k-n)}, \quad k = 1, 2, \ldots, N. \quad \ldots \hspace{1cm} (8)$$

Note $X_m^{(k)}$ obeys the following recursion:

$$X_m^{(k+1)} = W_n^{-m} X_m^{(k)} + W_n^{-m} x_k$$

$$X_m^{(N)} = X_m$$

$$X_m^{(1)} = x_0 W_n^{-m}. \quad \ldots \hspace{1cm} (9)$$

So $X_m$ is obtained as the limit of a sequence of approximations that begins at $X_m^{(1)} = x_0 W_n^{-m}$ and terminates at $X_m^{(N)} = X_m$. This is the so-called Goertzel algorithm [8] for obtaining the $m$th DFT variable $X_m$, as the output of a digital filter excited by the sequence $(x_n)_{n=0}^{N-1}$. The output of the filter is read at time $k = N$ (see Fig. 2).
Dynamic Programming and the Decimation-in-Frequency FFT

The Goertzel algorithm is a nice dynamic programming-like solution for the DFT. However, it is not efficient. Computational complexity is of order $N^2$. Let us see if we can improve upon it. Consider $X_m^{(k)}$ for even frequency indices $m=2r$:

$$X_m^{(k)} = \sum_{n=0}^{k-1} x_n W_N^{-2(n-k)}, \quad k = 1, 2, \ldots, N$$

$$= \sum_{n=0}^{k-1} x_n W_N^{-n(2-k)}, \quad k = 1, 2, \ldots, N.$$  

(10)

For $k$ even (say $k=2s$),

$$X_{2s}^{(1+)} = \sum_{n=0}^{2s-1} x_n W_N^{-n(2s-n)}$$

$$= \sum_{n=0}^{2s-1} x_n W_N^{-n(2s-n)} + \sum_{n=0}^{2s-1} W_N^{-n(2s-n)}$$

$$= W_N^{-s} \sum_{n=0}^{s-1} x_n W_N^{-n(s-n)} + \sum_{j=0}^{s-1} x_{s-j} W_N^{-n(s-j)}$$

$$= W_N^{-s} X_s^{(1)} + X_s^{(1)+}.$$  

(11)

This shows that the two $s$-point DFT approximant $X_s^{(1)}$ may be obtained from two $s$-point approximants. By choosing $s=N/2$ and continuing backwards in this way (for odd subindices, as well) one arrives at a backward dynamic programming derivation of the decimation-in-frequency FFT. See Fig. 3 for an elementary representation of a four-point decimation in frequency FFT. The decimation-in-frequency algorithm improves on the Goertzel algorithm by requiring complexity on order $N \log N$.

IV. DETECTION, ESTIMATION, AND CONTROL IN THE AR($N$) CASE: KALMAN FILTERS, LEVINSON RECURSIONS, AND DYNAMIC PROGRAMMING

Autoregressive (AR) models for signals, states, and data play a starring role in many areas of signal processing and control. By appropriately selecting model parameters (and order) one can model the covariance structure and spectral characteristics of more general models. The so-called normal equations for identifying AR parameters are elegant and easily solved with recursions of the Levinson type.

In this section we tie up control, prediction, detection, and estimation in the special case where we are dealing with a zero-mean wide-sense stationary, scalar autoregres-

sive time series. The usual state-variable and matrix block diagrams give way to scalar variables and digital filter blocks of moving average filters. The normal equations are highlighted and dynamic programming is used to derive the famous Levinson recursions.

A. Models

Let \( \{x_k\} \) denote a scalar zero-mean wide-sense stationary AR sequence of order $p$ (denoted AR($p$)) that obeys the recursion

$$x_k = \sum_{n=1}^{p} a_n x_{k-n} + w_k, \quad \forall k.$$  

(12)

\(w_k\): sequence of i.i.d. $N(0, \sigma_w^2)$ random variables (r.v.s).

It is easy to see that the covariance sequence \(\{r_m\}_{m=-\infty}^{\infty}\), \(r_m = r_{-m}\), associated with the sequence \(\{x_k\}\) obeys the recursion

$$r_m = \sum_{n=1}^{p} a_n r_{m-n} + \sigma_w^2 \delta_m, \quad m=0, 1, \ldots.$$  

(13)

From here one may write out the so-called normal equations:

$$R_p a_p = r_p$$

$$= \left[ \begin{array}{c} r_0 \\ r_1 \\ \vdots \\ r_{p-2} \\ r_{p-1} \\ \cdots \\ r_0 \end{array} \right]$$

$$a_p = (a_1, \ldots, a_p)$$

$$r_p' = (r_1, \ldots, r_p).$$  

(14)

We note at this juncture that turning \(r_p\) upside down turns the solution to the normal equations upside down. To see this, let

$$J = \left[ \begin{array}{cccccc} 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{array} \right], \quad JJ = I$$  

(15)

denote the exchange matrix and note by the Toeplitz symmetry of \(R_p\) we have \(J R_p J = R_p\). Thus

\(^1\)Here and elsewhere i.i.d stands for independent, identically distributed and r.v. stands for random variable.
Use (20) in (22) to get a different recursion for \( Q_p(r_p) \): \[
Q_p(r_p) = \min_{a_p} \left[ a_p^2 r_0 - 2a_p r_0 + \min_{a_p} Q_p \left( r_p-1 - a_p J_{r_p-1} \right) \right]
\]
\[
= \min_{a_p} \left[ a_p^2 r_0 - 2a_p r_0 + Q_p \left( r_p-1 - a_p J_{r_p-1} \right) \right].
\]

As \( J \) turns vectors upside down, this proves the claim.

B. The Normal Equations are Fundamental

The AR coefficients \( a_p = (a_1, \cdots, a_p) \) that characterize the sequence \( \{x_k\} \) are fundamental to the implementation of control, prediction, and detection algorithms on noisily observed AR sequences. Unfortunately, sequences rarely come tagged with their corresponding AR parameters. More typically finite records of them come to use and we estimate a covariance function (or power spectrum), often by FFT-ing, squaring and windowing, and inverse FFT-ing. These estimates may then be used to solve for the coefficients \( a_p \) from the normal equations. This makes the normal equations fundamental and arouses our interest in efficient ways of solving them. The derivation that follows is an adaptation of Bellman's discussion of quadratic forms and dynamic programming in [9].

C. Dynamic Programming and Levinson's Algorithm

Consider the quadratic form
\[
Q_p(r) = r_0 - 2a_p r_p + a_p^2 R_p a_p
\]
This quadratic form is minimized for some choice of \( a_p \) that we denote \( \alpha_p^* \). It is easy to see that
\[
\alpha_p^* = a_p
\]
where \( a_p \) comes from the normal equation:
\[
a_p = R_p^{-1} r_p.
\]
The corresponding minimum of \( Q_p(r_p) \) we denote \( Q_p^*(r_p) \):
\[
Q_p^*(r_p) = r_0 - \alpha_p^* r_p
\]
\[
= r_0 - r_p R_p^{-1} r_p.
\]
The quadratic form \( Q_p(\cdot) \) may be written recursively as
\[
Q_p(r) = a_p^2 r_0 - 2a_p r_p + Q_p \left( r_p - 1 - a_p J_{r_p - 1} \right).
\]

So minimization of \( Q_p(r_p) \) with respect to \( a_p \) may be written
\[
Q_p^*(r_p) = \min_{a_p} Q_p(r_p)
\]
\[
= \min_{a_p} \left[ a_p^2 r_0 - 2a_p r_p + \min_{a_p} Q_p \left( r_p - 1 - a_p J_{r_p - 1} \right) \right]
\]
\[
= \min_{a_p} \left[ a_p^2 r_0 - 2a_p r_p + Q_p \left( r_p - 1 - a_p J_{r_p - 1} \right) \right].
\]

This equation contains the essence of dynamic programming and the principle of optimality: once the solution \( \alpha_p^* \), and corresponding minimum \( Q_p^* \), have been found for the order \( (p-1) \) problem, \( \alpha_p^* \) may be found as a function of \( r_0 \), \( r_p \), and \( \alpha_p^* \). At each step of the way the minimization on \( a_p \) is quadratic.

D. Noisy Prediction and the Kalman Predictor

Assume the sequence \( \{x_k\} \) is observed in zero-mean additive white Gaussian noise (WGN):
The companion form state model is
\[ X_{k+1} = AX_k + bw_{k+1} \]
\[ Z_k = c^T X_k \]
\[ X_k = \begin{bmatrix} X_{k-p+1} \\ \vdots \\ X_{k-1} \\ X_k \end{bmatrix} \]
\[ A = \begin{bmatrix} 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \\ 1 \\ -a_p & a_p-1 & \cdots & a_0 \end{bmatrix} \]
\[ b = c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \]

The stationary Kalman one-step predictor for the noiselily observed AR(\(p\)) sequence is
\[ \hat{X}_{k+1} = A\hat{X}_k + K(z_k - \hat{X}_k) \]
\[ \hat{X}_k = c^T \hat{X}_k \]

where \( K \) is a \((p \times 1)\) Kalman gain: 
\[ K = APC(c^TPc + \sigma^2)^{-1} \]
\[ = \begin{bmatrix} k_1, k_2, \cdots, k_N \end{bmatrix}^T \]
\[ P = (A - Kc')P(A - Kc')' + \sigma^2 KK' + \sigma^2 bb' \]

The Kalman prediction sequence \( \{\hat{X}_{k+1}\} \) for \( \{z_k\} \) can be interpreted as the output of an AR moving average (ARMA) filter with \(p\) poles and \(p-1\) zeros (denoted ARMA(\(p, p-1\)), driven by the prediction error sequence \( \{\epsilon_k = z_k - \hat{X}_k\} \) or an ARMA(\(p, p-1\)) filter driven by the observation sequence \( \{z_k\} \). The resulting filter equations are
\[ \hat{X}_{k+1} = \sum_{i=1}^{p} a_i \hat{X}_{k+1-i} + \sum_{i=1}^{p} g_i z_{k+1-i} \]

or
\[ \hat{X}_{k+1} = \sum_{i=1}^{p} (a_i - g_i) \hat{X}_{k+1-i} + \sum_{i=1}^{p} g_i z_{k+1-i} \]

where the coefficients \( g_i \) can be defined by the characteristic polynomial \( \Delta(\lambda) \) of \((A - Kc'):\n\[ \Delta(\lambda) = \lambda^p + \sum_{i=1}^{p} (g_i - a_i)\lambda^{p-i}. \]

See Fig. 4 for a block diagram of this predictor. Note that the noise-free moving average (MA) predictor filter, \( P(z) = \sum_{i=1}^{p} a_i z^{-i} \), is preserved in the feedback loop, but that the residual sequence \( r_k = z_k - \hat{X}_k \) is now weighted with a feedforward MA filter, \( Q(z) = \sum_{i=1}^{p} b_i z^{-i} \).

Why is the noisy Kalman predictor ARMA and not MA? The answer is that \( \{z_k\} \), a noisy version of an AR signal process, obeys an ARMA(\(p, p\)) difference equation. As an ARMA(\(p\)) model has an MA(\(p-1\)) predictor, it is at least logical (if not intuitive) that an ARMA(\(p, p\)) process has an ARMA(\(p, p-1\)) predictor.

### E. The Noise-Free Predictor

The prediction vector \( \hat{X}_k \) consists of the terms
\[ E[x_{k-p+1}/z_{k-1}, z_{k-2}, \cdots] \]
\[ \vdots \]
\[ E[x_{k-1}/z_{k-1}, z_{k-2}, \cdots] \]
\[ E[x_k/z_{k-1}, \hat{X}_{k-1}, \cdots] \]

When \( \sigma^2 = 0 \), then \( z_k = \hat{X}_k \), \( \forall k \) and
\[ E[x_{k-n}/z_{k-n}, z_{k-n-1}, \cdots] = E[x_{k-n}/x_{k-n}, \cdots] = x_{k-n}, \]
\[ n = 1, 2, \cdots. \]

So in this case the prediction vector is
\[
X_k = \begin{bmatrix}
X_{k-p+1} \\
X_{k-p+2} \\
\vdots \\
x_{k-1} \\
\hat{x}_{k-1}
\end{bmatrix}
\]  
\tag{39}

It follows that \( P \), the covariance \( E[X_k - \hat{X}_k | X_{k-1}] \) is
\[
P = \begin{bmatrix}
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{bmatrix}
\]  
\tag{40}

Calculating \( K \) by substituting (40) into (33), we find that \( \Delta(\lambda) = \lambda P \) and hence \( g_i = a_i \). This implies, as one would expect, that the prediction filter of (35) reduces to the purely MA relation:
\[
\hat{x}_{k+1} = \sum_{i=1}^{p} a_i x_{k+1-i}
\]  
\tag{41}

### F. Minimum Variance Control

One of the simplest control strategies is minimum variance regulation where one desires to minimize the variance of the AR(\( p \)) output sequence \( \{x_k\} \), and force \( E(x_k) = 0 \). The well known separation principle allows one to generate a feedback control strategy assuming noise free measurements, i.e., \( n_k = 0 \), and then use the same strategy in the noisy case but with the Kalman filter estimates \( \{\hat{x}_k\} \) replacing the actual filter outputs \( \{x_k\} \).

Assume then we have the system
\[
x_k = \sum_{i=1}^{p} a_i x_{k-i} + w_k + v_k
\]  
\tag{42}

where \( \{v_k\} \) is our feedback control sequence. We would like to minimize
\[
E(x_k^2) = E\left( \sum_{i=1}^{p} a_i x_{k-i} + w_k + v_k \right)^2
\]
\[
= E(w_k^2) + 2E(w_k v_k) + E\left( \sum_{i=1}^{p} a_i x_{k-i} \right)
\]
\[
+ E(v_k^2) + \left( \sum_{i=1}^{p} a_i x_{k-i} \right)^2
\]
\[
= E(w_k^2) + 2E(E(x_k^2 | x_k) v_k)
\]
\[
+ 2E(w_k \left( \sum_{i=1}^{p} a_i x_{k-i} \right))
\]
\[
+ E\left( \left( v_k + \sum_{i=1}^{p} a_i x_{k-i} \right)^2 \right)
\]  
\tag{43}

Since \( \{w_k\} \) is uncorrelated with \( \{x_{k-i}\}, i > 1 \), and since \( E(w_k v_k | v_k) = 0 \), it is clear that \( E(x_k^2) \) is minimized by choosing
\[
v_k = -\sum_{i=1}^{p} a_i x_{k-i}
\]  
\tag{44}

This control is illustrated in Fig. 4 as a feedback loop running up the left side of the figure. The feedback loop to the top "compute" box shows how \( \hat{x}_k \) would be used for minimum variance control in the noisy case.

### G. Detection and the Likelihood Ratio

Consider the hypothesis test \( H_0 \) versus \( H_1 \) with
\[
H_0: z_k = n_k, \quad k = 0, 1, \ldots, K
\]
\[
H_1: z_k = x_k + n_k, \quad k = 0, 1, \ldots, K
\]  
\tag{45}

and the data assumed stationary over the interval. This test is equivalent to the test \( H_0 \) versus \( H_1 \), where
\[
\hat{H}_0: P(0) = z_k \sim N(0, \sigma_k^2)
\]
\[
\hat{H}_1: P(1) = z_k \sim N(0, \sigma_k^2 + \sigma_n^2)
\]  
\tag{46}

Thus the statistics \( \Sigma y_k^2 \) and \( \Sigma z_k^2 \) are sufficient and the log-likelihood ratio may be computed as in Fig. 4.

### V. Frequency Tracking and Dynamic Programming

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the entire realm of signal processing. Nonlinear filtering and MAP solutions have been reported recently by Bucy and Mallinckrodt [11], Ungerboeck [12], Tufts [13], Scharf et al. [14], [15], and Wolcin [16]. A typical problem is the following: observe the signal-plus-noise sequence \( \{z_k\} \) with
\[
z_k = s_k + n_k, \quad n_k: \text{sequence of i.i.d. } N(0, \sigma_n^2)
\]
and estimate the phase sequence \( \{\phi_k\} \) or some underlying function of it. Here the character assigned to state \( \phi_k \) is
\[
s_k = e^{j\phi_k}
\]  
\tag{48}

and estimate the phase sequence \( \{\phi_k\} \) or some underlying function of it. Here the character assigned to state \( \phi_k \) is
\[
s_k = \exp(j\phi_k)
\]

In all that follows it will be convenient to organize the observed data into contiguous data blocks:
\[
Z = \begin{bmatrix}
z_0 \\
z_1 \\
\vdots \\
z_{K-1}
\end{bmatrix}
\]

Since \( \{w_k\} \) is uncorrelated with \( \{x_{k-i}\}, i > 1 \), and since \( E(w_k v_k | v_k) = 0 \), it is clear that \( E(x_k^2) \) is minimized by choosing
\[
v_k = -\sum_{i=1}^{p} a_i x_{k-i}
\]  
\tag{44}

This control is illustrated in Fig. 4 as a feedback loop running up the left side of the figure. The feedback loop to the top "compute" box shows how \( \hat{x}_k \) would be used for minimum variance control in the noisy case.

### G. Detection and the Likelihood Ratio

Consider the hypothesis test \( H_0 \) versus \( H_1 \) with
\[
H_0: z_k = n_k, \quad k = 0, 1, \ldots, K
\]
\[
H_1: z_k = x_k + n_k, \quad k = 0, 1, \ldots, K
\]  
\tag{45}

and the data assumed stationary over the interval. This test is equivalent to the test \( H_0 \) versus \( H_1 \), where
\[
\hat{H}_0: P(0) = z_k \sim N(0, \sigma_k^2)
\]
\[
\hat{H}_1: P(1) = z_k \sim N(0, \sigma_k^2 + \sigma_n^2)
\]  
\tag{46}

Thus the statistics \( \Sigma y_k^2 \) and \( \Sigma z_k^2 \) are sufficient and the log-likelihood ratio may be computed as in Fig. 4.

### V. Frequency Tracking and Dynamic Programming

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the entire realm of signal processing. Nonlinear filtering and MAP solutions have been reported recently by Bucy and Mallinckrodt [11], Ungerboeck [12], Tufts [13], Scharf et al. [14], [15], and Wolcin [16]. A typical problem is the following: observe the signal-plus-noise sequence \( \{z_k\} \) with
\[
z_k = s_k + n_k, \quad n_k: \text{sequence of i.i.d. } N(0, \sigma_n^2)
\]
and estimate the phase sequence \( \{\phi_k\} \) or some underlying function of it. Here the character assigned to state \( \phi_k \) is
\[
s_k = \exp(j\phi_k)
\]  
\tag{48}

and estimate the phase sequence \( \{\phi_k\} \) or some underlying function of it. Here the character assigned to state \( \phi_k \) is
\[
s_k = \exp(j\phi_k)
\]

In all that follows it will be convenient to organize the observed data into contiguous data blocks:
\[
Z = \begin{bmatrix}
z_0 \\
z_1 \\
\vdots \\
z_{K-1}
\end{bmatrix}
\]
Therefore, we may write \( s_k \) as follows:

\[
\phi_k = \phi_{k-1} + w_k
\]

where \( \phi_k \) is the total accumulated phase after \( tN \) steps. It depends on the entire history of frequency terms \( \nu_0, \nu_1, \ldots, \nu_{r-1} \) and obeys the recursion

\[
\phi_k = \phi_{k-1} + \frac{2\pi}{Q} \nu_{r-1}
\]

Between \( tN \) and \((t+1)N - 1\) the phase grows linearly as

\[
\phi_k = \phi_{tN} + (i - tN) \frac{2\pi}{Q} \nu_i
\]

This additional phase increase is accounted for in the vector \( d(\nu_i) \). See Fig. 5 for an illustration.

To complete the model we assume \( \nu_i \) is a sequence of \( \nu_i \) discrete random variables that take values in the set \( \{0, 1, \ldots, Q-1\} \) and evolve according to the rule

\[
\nu_i = \nu_{i-1} \mod Q
\]

where \( \nu_i \in \{0, 1, \ldots, Q-1\} \) and addition is modulo-\( Q \). The distribution of the sequence of i.i.d. random variables is selected in such a way that the transition probability

\[
p(\nu_i/\nu_{i-1})
\]

This corresponds to our notion of physical reality. We may think of the resulting frequency sequence \( \nu_i \) as a finite-state random walk on the circle with an unusual transition probability structure. Typical trajectories for \( \nu_i \) and \( s_k \) are illustrated in Fig. 5.

The joint likelihood function for \( Z \) and \( \nu_i \) is proportional to

\[
L = \sum_{i=0}^{K-1} \frac{1}{2\sigma_i^2} |z_i - s_i|^2 + \sum_{r=0}^{K-1} \ln p \left( \frac{\nu_i}{\nu_{i-1}} \right).
\]

Using our representation for \( s_k \) and dropping terms independent of \( \nu_i \) we obtain

\[
L \approx \frac{1}{2\sigma_i^2} \sum_{i=0}^{K-1} \text{Re} \left( \exp(-j\phi_{tN})z_i d^*(\nu_i) \right) + \sum_{r=0}^{K-1} \ln p \left( \frac{\nu_i}{\nu_{i-1}} \right).
\]

The term \( z_id^*(\nu_i) \) is nothing more than the DFT of \( z_i \) evaluated at the DFT frequency \((2\pi/Q)\nu_i\). The best way to compute it is to zero-pad \( z_i \) to obtain a \( Q \) point sequence that may be FFT-ed. See Fig. 6.

Our notion of the most likely sequence \( \nu_i^* \) is the
sequence that maximizes \( l \). This is the MAP sequence. Write the maximization problem as

\[
\max_{\{r_i\}^K_{i=1}} \Gamma_{K-1} \quad (58)
\]

with

\[
\Gamma_i = \Gamma_{i-1} + \frac{1}{\sigma_n^2} \Re \{\exp(-j\phi_i)z_i' d^*(r_i)\} + \ln p \left( \frac{r_i}{r_{i-1}} \right).
\]

So our maximization problem becomes

\[
\max_{\{r_i\}^K_{i=1}} \left[ \max_{\{r_i\}^K_{i=1}} \Gamma_{K-2} + \ln p \left( \frac{r_{K-1}}{r_{K-2}} \right) \right. \\
+ \frac{1}{2\sigma_n^2} \Re \{\exp(-j\phi_{K-1})z_{K-1}' d^*(r_{K-1})\} \right].
\]

Thus for each node on Fig. 6 we evaluate the FFT \( z_i' d^*(r_i) \), phase it by \( \exp(-j\phi_i) \), and find the best route through the trellis with the dynamic programming algorithm of (61). This completes our algorithm for moderating the sequential estimator which is also used for this purpose can be found in [19]. In this section we outline a new dynamic programming algorithm for sequentially estimating short boundary segments. We then briefly discuss an algorithm which pieces together the short segments and present some examples of its use on complete images.

**VI. LOCAL BOUNDARY ESTIMATION IN NOISY BLACK AND WHITE IMAGES**

In digital image processing one is interested in developing computer algorithms which can either automatically extract information from pictures or at least simplify the process of manually interpreting them. In either case, a basic step involves segmenting a picture into regions with similar features such as gray level or texture. This involves the estimation of region boundaries. Boundary estimation algorithms make use of operators which estimate short segments of boundaries using picture data in small picture sections. Examples are simple gradient operators and the well-known Heuckel operator [18]. An example of a local sequential estimator which is also used for this purpose can be found in [19]. In this section we outline a new dynamic programming algorithm for sequentially estimating short boundary segments. We briefly discuss an algorithm which pieces together the short segments and present some examples of its use on complete images.

**A. Image and Boundary Models**

Let a digitized black and white image be represented by a matrix with components \( g_{ij} \) corresponding to the gray level value of a picture element (pixel) centered at position \((i, j)\). The value \( g_{ij} \) will have two components—a true picture component \( b_{ij} \) and a noise component \( n_{ij} \) so that \( g_{ij} = b_{ij} + n_{ij} \). A picture is assumed to consist of a single region of gray level \( r_{in} \) lying in a background of gray level \( r_{out} \), so that \( b_{ij} \) can take on either of the two values \( r_{in} \) or \( r_{out} \). The noise components \( n_{ij} \) are assumed to be independent identically distributed Gaussian random variables with mean zero and variance \( \sigma_n^2 \), denoted \( n_{ij} \sim N(0, \sigma_n^2) \).

An edge element is defined as the line segment separat-
CONTINUOUS PHASE RANDOM WALK FM MAP FREQUENCY ESTIMATION

LAG = 10 BLOCKS
R. WALK VAR. = 0.01000
CNR = -3.0

--- DECODED FREQ.  --- ACTUAL FREQ.
NORMAL DENSITY
(N.O.) = (8, 32)

Fig. 7. Frequency tracking at CNR = -3.0 dB. Random walk variance = 0.01 rad².

CONTINUOUS PHASE RANDOM WALK FM MAP FREQUENCY ESTIMATION

LAG = 10 BLOCKS
R. WALK VAR. = 0.10000
CNR = -3.0

--- DECODED FREQ.  --- ACTUAL FREQ.
NORMAL DENSITY
(N.O.) = (8, 32)

Fig. 8. Frequency tracking at CNR = -3.0 dB. Random walk variance = 0.1 rad².

ing two adjacent pixels, and as shown in Fig. 9 a boundary segment consists of a directed sequence of edge elements \( t_i \). As illustrated in Fig. 10, we assume short boundary segments to be generated by constructing a sequence of edge elements that terminate at the boundary of a rectangular box. Longer sequences of edge elements defining longer more complicated boundaries are obtained by exiting successive rectangles \( A_k \), \( 1 \leq k \leq N \) containing \( \rho_k = kx(2k-2) \) pixels. The key constraint built into this generating scheme is that sequences departing one rectangle cannot re-enter it. Fig. 11(a) gives an example of a boundary which is consistent with this model while Fig. 11(b) shows
Fig. 9. A boundary segment in small picture segment.

Fig. 10. Example of boundary segment generation.

Fig. 11. Example of a boundary. (a) Consistent with model. (b) Inconsistent with model.

Fig. 12. CAT scan of abdominal section of human body.

a similar but inconsistent boundary. In the latter case the edge sequence reenters $R_4$. Although this scheme restricts somewhat the types of boundary segments that can be generated, it is still very reasonable for region boundaries with low and slowly varying curvatures such as those in the body computerized axial tomography (CAT) scan shown in Fig. 12. The maximum rectangle size $\rho_5$ is assumed fixed $a$ priori and is a function of the boundary curvature properties for the region of interest.

Boundary segments generated by such a model are naturally represented by a sequence of states in a Markov chain where the index parameter $k$ for the rectangle of size $\rho_k$ is also the index parameter for the Markov process. A process state $x_k$ at "time" $k$, will correspond geometrically to the end point of a boundary sequence passing out of $R_k$. Fig. 13 shows all possible locations for $x_k$, denoted as $x_k$, when $k = 1, 2, \cdots, 5$. The number of possible states at time $k$ is 1 for $k = 1, 3$ for $k = 2$, and $9 + 4(k - 3)$ for $k \geq 3$.

Note that there is only one edge sequence between any two states $x_{k-1}$ and $x_k$ which is consistent with the generation model and which does not pass through another state $x'_k$. As a result, a boundary segment $\{x_i\}_{i=1}^m$ is uniquely characterized by a state sequence $\{x_i\}_{i=1}^m$.

Fig. 14 contains an abstract representation of a typical realization of the Markov process, together with a description of the picture or character $C_k$ associated with each state. The observed image will be a noise corrupted version of each such picture.

If the regions of interest have smooth, low curvature boundaries then a reasonable rule for assigning transition probabilities $p(x_k|x_{k-1})$ is to choose $p(x_k|x_{k-1})$ to be inversely related to the distance (measured in edge elements) between states $x_{k-1}$ and $x_k$. We must also impose the total probability constraint that

$$ \sum_{j=1}^{9 + 4(k - 3)} p(x'_i|x_{k-1}) = 1. \quad (62) $$
B. A Dynamic Programming Algorithm for Estimating Boundary Segments

Using the pixel data in an \( N \times 2(N-1) \) block, \( R_N \), we next formulate a dynamic programming algorithm for estimating the most likely state sequence consistent with the generation model. The algorithm is optimal in the sense that it finds the state sequence that maximizes the joint likelihood of the data in \( R_N \) and the corresponding edge sequence through \( R_N \).

To begin we first define the pixel data sets
\[
D_k = \{ g_{ij} : \text{pixel } (i, j) \in R_k \}
\]
\[
d_k = \{ g_{ij} : \text{pixel } (i, j) \in R_k, \text{ pixel } (i, j) \notin R_{k-1} \}.
\]

This implies that \( D_k = D_{k-1} \cup d_k, D_1 = \text{empty set} \). This recursion is essential. Next let \( l(\cdot) \) denote a log-likelihood function, and \( S_N = \{ x_j \}_N^{i=1} \) denote a boundary state sequence of length \( N \). Then \( l(D_N, S_N) \), the joint log-likelihood of a boundary state sequence and the picture data, must satisfy
\[
l(D_N, S_N) = l(D_N | S_N) + l(S_N)
\]
where \( l(S_N) \) is the log-likelihood of the state sequence \( S_N \) and \( l(D_N | S_N) \) is the pixel data log-likelihood conditioned on the boundary \( \{ x_j \}_N^{i=1} \) described by \( S_N \). Since the state sequence \( S_N \) is a Markov chain we can use
\[
l(S_N) = l(S_{N-1}) + \ln p(x_1 | x_{N-1})
\]
\[
l(S_1) = l(x_1) = \ln P_i(x_1)
\]
where \( P_i(x_1) \) is the probability of a particular starting state \( x_1 \). Since boundary edge sequences are prohibited from reentering rectangles they have already passed out of, we can express
\[
l(D_N | S_N) = l(D_{N-1} | S_{N-1}) + l(d_n | x_N)
\]
where \( l(d_n | x_N) \) is the log-likelihood of the data added in extending the state sequence \( S_{N-1} \) to \( S_N \) conditioned on the specific new state \( x_N \). Substitution of (65) and (64) into (63) leads to the following recursive expression for \( l(D_N, S_N) \):
\[
l(D_N, S_N) = l(D_{N-1}, S_{N-1}) + \ln p(x_N | x_{N-1}) + l(d_n | x_N).
\]

The transition probabilities \( p(x_N | x_{N-1}) \) can be calculated using a distance rule such as the one discussed above, while incremental data log-likelihoods, \( l(d_n | x_N) \) can be calculated by observing that the pixel gray level values \( g_{ij} \) are \( N(\mu_0, \sigma^2) \) if \( g_{ij} \) lies inside the region and \( N(\mu_1, \sigma^2) \) when \( g_{ij} \) lies outside the region. Furthermore, once \( x_N \) has been specified, all pixel values \( g_{ij} \) in \( d_k \) can be associated with pixels either inside of or outside of the region. Hence if we define
\[
f_0(x) \triangleq \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left( -\frac{x^2}{2\sigma^2} \right)
\]
we can use
\[
l(d_k | x_N) = \sum_{(i,j) \in I_k} \ln f_0(g_{ij} - r_{in})
\]
\[
+ \sum_{(i,j) \in I_k} \ln f_0(g_{ij} - r_{out})
\]
\[
= C - \sum_{(i,j) \in I_k} \frac{(g_{ij} - r_{in})^2}{2\sigma^2} - \sum_{(i,j) \in I_k} \frac{(g_{ij} - r_{out})^2}{2\sigma^2}
\]
where \( C \) is a constant which is independent of the choice of \( x_k \), and
\[
I_k = \{(i,j) : \text{pixel } (i,j) \text{ is in region and } g_{ij} \in d_k \}
\]
\[
I_{k+} = \{(i,j) : \text{pixel } (i,j) \text{ is in background and } g_{ij} \in d_k \}.
\]

Finally, a dynamic programming algorithm for estimating a state sequence \( S_N \) and hence a boundary edge sequence \( \{ x_j \}_N^{i=1} \) which maximizes \( l(D_N, S_N) \) can be derived by...
observing that
\[
\max i(D_n,S_n) = \max_{x_n} \left[ \max_{S_{n-1}} i(D_{n-1},S_{n-1}) + \ln p(x_n|x_{n-1}) + i(d_n|x_n) \right].
\]

This boundary segment estimator has been incorporated into a complete boundary estimation scheme. At the last stage of the forward dynamic programming algorithm the three most likely states \(x_n\) are used to generate three complete paths out of a rectangle. These are stored as nodes on a tree, and an \(A^k\) types tree search algorithm [20] is used to piece together complete boundaries. Fig. 15 shows some examples of the overall algorithm performance. Fig. 15(a) shows the algorithm performance on an ellipse imbedded in Gaussian noise such that the signal-to-noise ratio \((r_n - \sigma_{out})/\sigma = 1\). Both the actual and estimated boundaries are plotted. Fig. 15(b) shows the boundary obtained for a section of the CAT scan given in Fig. 12, and Fig. 15(c) shows the result of estimating the boundary of a satellite image of a cloud.

VII. CONCLUDING REMARKS

By constructing finite-state Markov chains, and assigning characters or observations to these states, one can model such things as continuous-phase FM signals and random boundaries in black and white images. The model may then be used to construct a likelihood function that may be written recursively and maximized with the techniques of dynamic programming. The resulting algorithms are tractable by nonlinear filtering and image processing standards, and the results often superior to what can be achieved with other approaches.

A great variety of signal and image processing problems may be phrased along the lines of this paper, and solved using the techniques of dynamic programming. This chain illustrates once again the great power of dynamic programming as a recursive optimization device.

ACKNOWLEDGMENT

The authors would like to thank M. Orfali and L. Srinivasan, graduate students at Colorado State University, for writing the software used to generate the examples given in Sections V and VI, respectively.

REFERENCES


Louis L. Scharf (M'69-\(\Phi\)76) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1964, 1966, and 1969, respectively. Since 1971 he has been a member of the Electrical Engineering Department, Colorado State University, Fort Collins, where he is currently Professor of Electrical Engineering and Statistics. He enjoys drawing inferences from random data in real-time at high speeds. During the academic year 1974 he was a Visiting Associate Professor at Duke University, Durham, NC. He spent the academic year 1977 in France as a Professeur Associé at the University of Paris, South Orsay, and as a member of the Technical Staff in the CNRS Laboratoire des Signaux et Systèmes Gif-sur-Yvette. He has served as a Consultant to Honeywell, Inc., Seattle, The Applied Physics Laboratory, Seattle, and the Research Triangle Institute. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING.

Dr. Scharf is a member of Eta Kappa Nu, Sigma Xi, and the Acoustical Society of America. He was Technical Program Chairman for the 1980 International Conference on Acoustics, Speech and Signal Processing.

Howard Elliott (S'75-M'78) was born in Boston, MA, on October 11, 1952. He received a combined Sc.B. and A.B. degree in electrical engineering and economics, the M.S. degree in electrical engineering in 1975, and the Ph.D. degree in electrical sciences in 1979, all from Brown University, Providence, RI.

He was appointed Post Doctoral Teaching and Research Fellow for one semester. He is presently an Assistant Professor of Electrical Engineering at Colorado State University, Fort Collins. His research interests are in multivariable and adaptive control, and digital image processing.

Dr. Elliott is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.
APPENDIX B: Progress Reports and Miscellaneous Documents
PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRXRO-PR P-16437-EL

2. PERIOD COVERED BY REPORT: 1 September 1979 thru 31 December 1979

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG29-79-C-0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis L. Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:
   a. Louis L. Scharf (not actually supported during this period)
   b. Helen Anderson, M.S. student
   c. Kazam Kazampur, M.S. student
   d. Freddie Hanson, Work-study
   e. David C. Farden, Ph.D. - consultant

LOUIS L. SCHARF
COLORADO STATE UNIVERSITY
ELECTRICAL ENGINEERING DEPARTMENT
FT. COLLINS, CO 80523
BRIEF OUTLINE OF RESEARCH FINDINGS

We are pursuing research on three distinct but related problems: (1) phase model extension to include random phase modulation, random FM modulation, and random chirp modulation; (2) frequency estimation in signal-plus-noise and autoregressive models; (3) dynamic programming algorithm development for FM tracking; and (4) simultaneous phase tracking and data decoding on random phase channels.

(1) Phase Model Extension: Here we have derived phase models for random phase, random FM, and random chirp modulation. Each model is a Markov chain defined on cyclic group. Covariance and spectral results have been derived. The results - not yet published - generalize existing results on the spectral theory of chains, and leave us with the problem of selecting states, transition probabilities, and "run lengths" to achieve model matching with more conventional models.

(2) Frequency Estimation: We have derived maximum likelihood frequency estimators and Cramer-Rao bounds for estimating frequency in complex normal signal-plus-noise and autoregressive models. The estimators have been simulated and modulo-2π errors studied. The results explode a currently popular myth regarding frequency tracking at low signal-to-noise ratios. Work will probably be published shortly.

(3) Dynamic Programming Algorithm Development: In reports (a) and (b) under item 7 of this document we have derived a dynamic programming algorithm for picking the optimum frequency track through a sequence of contiguous FFT maps to decode the MAP frequency sequence. Algorithm properties are under study and software development will begin soon.

(4) Simultaneous Phase Tracking and Data Decoding: A principle of optimality for phase tracking/data decoding has been derived and implemented in software to decode data symbols transmitted over random phase channels. Algorithm performance is treated in report (c) under item 7 of this report. The algorithm - though complex - outperforms all competitors.
PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRXRO-PR P-16437-EL

2. PERIOD COVERED BY REPORT: 1 September 1979 thru 30 June 1980

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG29-79-C-0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis L. Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:
   a. Louis L. Scharf
   b. Claude Gueguen, Visiting Professor
   c. David C. Farden, Ph.D. - consultant
   d. Helen Anderson, M.S. awarded May 1980
   e. Freddie Hanson, Work-study

Dr. Louis L. Scharf 16437-EL
Colorado State University
Electrical Engineering Department
Fort Collins, CO 80523
7. (con't).


PROGRESS REPORT
(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRVO-PR-P-16437-EL

2. PERIOD COVERED BY REPORT: Through 31 December 1982

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG 29 - 79 - C - 0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:
   a. LL Scharf
   b. JP Dugre, Ph.D., July 1981

N.B. Copies of abstracts follow.

Dr. Louis L. Scharf 16437-EL
Colorado State University
Electrical Engineering Department
Fort Collins, CO 80523
The last progress report contained a complete list of accomplishments and ongoing work. That outline remains in force, with the addition of the following:

a. **Phase Model Extension.** We are in the process of writing up our work on phase models on the circle. This work could lead the way to filtering on finite groups, a topic I raised to ARO in a letter to Suttle a year ago.

b. **ARMA Systems.** We have reformulated the autoregressive moving average (ARMA) modelling problem in terms of linear transformations, rather than linear filters. It's too early to give a prognosis, but new insights are developing. An invited paper for IEEE Trans on ASSP is in progress.
Dr. Jimmie R. Suttle, Director  
Electronics Division  
U.S. Army Research Office  
P.O. Box 12211  
Research Triangle Park, NC 27709

Dear Dr. Suttle:

Here is my brief report on scientific accomplishments.

PROJECT: Viterbi Tracking of Randomly Phase Modulated Data  
DAAG29-79-C-0176

OUTLINE: At Colorado State University the principal investigator and his associates are working on a nonlinear smoothing theory for randomly phase- and frequency-modulated information. The investigator's phase tracker has been generalized to a frequency tracker. Simulation and theoretical performance evaluations are in progress. Analytical investigation of Markov chains as approximants to FM signals is proceeding.

Application of these results arise in 1) detection and estimation of feeble sinusoidal signals (such as oscillation modes), 2) phase synchronization of data transmission systems, and 3) decoding of frequency-hopped FM signals.


Louis L. Scharf  
Professor  
Electrical Engineering

LS:fr
The problem of FM demodulation has a long history of research and development in electrical engineering. In its modern form the problem is to estimate phase or frequency sequences from noisy data and to use these estimates in conventional and spread spectrum communication systems.

The principal investigator and his associates have developed models for random phase and frequency sequences and derived likelihood expressions for noisy observations of them. The investigators have applied dynamic programming to find an algorithm for computing the maximum of the likelihood. The algorithm has been applied to the decoding of binary, phase-shift-keyed, and quadrature-shift-keyed data sequences.

The results of this research suggest that there are numerous nonlinear filtering problems in signal and image processing that can be formulated and solved as nonlinear sequence estimation problems. Among the possibilities are boundary estimation in noisy black and white images, tomographic image reconstructive in noisy CAT Scans, and vehicle tracking from incomplete and noisy measurements.
APPENDIX C: Army Sponsored Meetings Attended by PI

The Principal Investigator attended the following ARD-sponsored meetings at Fort Monmouth:

- Spread Spectrum, 29 May 1980
  Fort Monmouth, New Jersey

- Spread Spectrum Seminar, 22 May 1981
  Fort Monmouth, New Jersey

At the 22 May 1981 meeting he presented a paper titled,

"Viterbi Tracking of Randomly Phase-Modulated Date"
TENTATIVE AGENDA
SPREAD SPECTRUM SEMINAR
Fort Monmouth, NJ
May 22, 1981

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0830</td>
<td>Army Presentations</td>
</tr>
<tr>
<td>1030</td>
<td>Break</td>
</tr>
<tr>
<td>1045</td>
<td>Spread Spectrum Receiver Using SAW Devices</td>
</tr>
<tr>
<td></td>
<td>Prof. Pankaj Das, Rensselaer Polytechnic Institute</td>
</tr>
<tr>
<td>1130</td>
<td>Viterbi Tracking of Randomly Phase Modulated Data</td>
</tr>
<tr>
<td></td>
<td>Prof. Louis Scharf, Colorado State University</td>
</tr>
<tr>
<td>1215</td>
<td>Lunch</td>
</tr>
<tr>
<td>1315</td>
<td>Research in Digital Communications</td>
</tr>
<tr>
<td></td>
<td>Prof. Robert Scholtz, University of Southern California</td>
</tr>
<tr>
<td></td>
<td>Prof. William Lindsey, University of Southern California</td>
</tr>
<tr>
<td>1445</td>
<td>Break</td>
</tr>
<tr>
<td>1500</td>
<td>Spread Spectrum Communications</td>
</tr>
<tr>
<td></td>
<td>Prof. Michael Pursley, University of Illinois</td>
</tr>
<tr>
<td></td>
<td>Prof. Robert McEliece, University of Illinois</td>
</tr>
<tr>
<td>1630</td>
<td>Closing</td>
</tr>
</tbody>
</table>