SEPTIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 28 MAI au 2 JUIN 1979

Phase Sequence Estimation and Coherent Data Transmission

Louis L. Scharf[1] and Odile Macchi[2]

[1]Electrical Engineering Dept., Colorado State University, Ft. Collins, CO 80523, U.S.A.
[2]C.N.R.S. Laboratoire des Signaux et Systèmes, Plateau du Moulon, 91190 Gif-sur-Yvette, France.

RESUME

On definit le problème d'estimation de phase et de décodage simultané de symboles de données en bande de base. On suppose que la séquence de phase est une séquence aléatoire sur le cercle et que les symboles équi-probables sont transmis sur un canal parfaitement égalisé. Un algorithme de programmation dynamique [algorithme de Viterbi] est donné pour le décodage d'une séquence de symboles et phases, maximum à postériori (MAP) sur un treillis de symboles et phases de dimension finie. Un intéressant principe d'optimalité pour simultanément éstimer la phase et décoder les symboles codés phase-amplitude, conduit à une méthode efficace à deux étapes. Les resultats de simulation pour des ensembles de symboles 8-ARY PM et 16-QASK transmis sur un canal présentant une gigue de phase sinusoidale ou aléatoire avec incrément normal et indépendant, sont présentés et comparés aux résultats que l'on peut obtenir avec des algorithmes à décision dirigée ou autres.

SUMMARY

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. An interesting principle of optimality for simultaneously estimating phase and decoding phase-amplitude coded symbols leads to an efficient two step decoding procedure. Simulation results for 8-ARY PM, and 16-QASK symbol sets transmitted over random walk and sinusoidal jitter channels are presented and compared with results one may obtain with decision-directed and other algorithms.

I. Introduction

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. The DDPLL of [5] is a first-order digital phase-locked loop (PLL) in which the phase estimate is updated on the basis of a new measured phase and symbol decision. In the jitter equalizer (JE) of [3] and [4] a complex gain is updated according to a simple decision directed stochastic approximation algorithm. The complex gain is used to scale and rotate the received signal, thereby correcting phase jitter and 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.

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 rôle in a forward dynamic programming algorithm for obtaining MAP phase symbol sequences was indicated. Because of the way phase was modelled in [1], the dynamic programming algorithm could not be solved directly. Using two approximations, Ungerboeck derived an implementable algorithm and obtained performance results that were on the order of 3dB superior in SNR to the DDPLL in a 16-QASK system at interesting values of the phase variance parameter. See [1] for details. The reader is referred also to [5] and [6] for discussions of other sub-optimum, 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 straight-forward 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 $[-\pi,\pi)$ 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. 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 near to the square of the number of admissible phase values per amplitude level. This amounts to a factor of 16 for the 16-point QASK diagram that has been recommended by CCITT for data transmission on telephone lines at 9600 b/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 k=10. This obviates the need for huge storage requirements for long sequences.

Simulation results for the proposed Viterbi algorithm (VA) are presented for several symbol sets including 8 or 16 points. Several types of phase jitter are investigated such as Gaussian and non-Gaussian random walk, and sinusoidal phase jitter. The resulting error probability is compared with that of the simple JE. As expected, performance of the VA is always superior to that of the JE. 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 rather than a simpler decision-directed algorithm (such as the JE) or approximate VA of the type discussed in [1].

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. At the receiver the output of a bandpass filter/quadrature demodulator/complex adaptive 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

$$x_k = a_k e^{j\phi_k} + n_k, \quad k \in N^+ \qquad (1)$$

The sequence $\{\phi_k\}$ represents phase fluctuations (jitter and frequency drift) in the channel. The complex noise sequence $n_k = n_k^{(1)} + jn_k^{(2)}$ is assumed to be a sequence of independent identically distributed (i.i.d.) complex Gaussian variables with $n_k : N_{n_k}(0,\sigma_n^2)$

Consider now the phase distortion $\{\phi_k\}$. We model it as

$$\phi_k = \phi_{k-1} + w_k, \quad k \in N^+ \qquad (2)$$

where $\{w_k\}$ is a sequence of i.i.d. random variables with even probability density $h(w)$.[1] When $w_k : N_w(0,\sigma_w^2)$, then $\{\phi_k\}$ is the so-called normal random walk. In detail this model falls well short of a reputable probabilistic model for phase, because at low frequencies the spectrum is unbounded. Furthermore the spectrum is not integrable, corresponding to the unbounded growth of the variance in the diffusion model of (2). However, in gross terms, i.e. 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 that it forms a convenient basis from which to derive estimator structures which may then be evaluated against more realistic phase sequences.

As the measurement model of (2) is invariant to modulo-2π translates of ϕ_k we may represent phase as if it were a random sequence on the unit circle C or equivalently on the interval $[-\pi,\pi)$. This means ϕ_k, modulo-2π, has transition density

$$f(\phi_{k+1}/\phi_k) = \sum_{\ell=-\infty}^{\infty} h(\phi_{k+1}-\phi_k-\ell 2\pi) = g_1(\phi_{k+1}-\phi_k) \qquad (3)$$

We often denote the sum in (3) by $g_1(\cdot)$ and call it the folded density of the phase increments. Usually the phase increment is small and its distribution $h(\cdot)$ is very narrow with respect to 2π. Therefore, in the sum of (3) only one term is relevant and $f(\phi_{k+1}/\phi_k) \doteq h(\phi_{k+1}-\phi_k)$. In the normal case this implies $\sigma_w \ll 2\pi$, where σ_w^2 is the variance of w_k.

In the normal case [7], [8], the density $g_1(\phi_{k+1}-\phi_k)$ may be written

$$g_1(\phi_{k+1}-\phi_k) = \sum_{\ell=-\infty}^{\infty} N_{\phi_{k+1}}(\phi_k+\ell 2\pi,\sigma_w^2) \qquad (4)$$

In the normal and Cauchy cases it may be shown that $g_1(x)$ achieves its maximum at x=0 and that it is monotone decreasing on $0 \le x \le \pi$.

The sequence $\{\phi_k\}_1^K$ is Markov. Therefore, we may write for the joint density of the K phases $\{\phi_k\}_1^K$

$$f(\{\phi_k\}_1^K) = \prod_{k=1}^{K} f(\phi_{k+1}/\phi_k) \qquad (5)$$

$$f(\{\phi_1/\phi_0) \triangleq f(\phi_1) \text{ : the marginal density of } \phi_1 .$$

Usually ϕ_1 is uniformly distributed on C, because phase acquisition starts at k=1 with no prior information about its value. By the independence of the n_k in (1) it follows that the conditional density of the measurement sequence $\{x_k\}_1^K$, given the phase and data sequences $\{\phi_k\}_1^K$, $\{a_k\}_1^K$, is

$$f(\{x_k\}_1^K/\{\phi_k\}_1^K, \{a_k\}_1^K) = \prod_{k=1}^{K} N_{x_k}(a_k e^{j\phi_k},\sigma_n^2) \quad . \qquad (6)$$

Equations (3)-(6) form the basis for the derivation of a MAP sequence estimator. The key element is

[1] That is, $h(w) = h(-w)$.

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:

$$y_k = x_k\, e^{-j\hat{\phi}_k}, \quad k \epsilon N^+ \quad . \quad (7)$$

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

$$\hat{\phi}_{k+1} = \hat{\phi}_k + \mu \mathrm{Im}[x_k \hat{a}_k^* e^{-j\hat{\phi}_k}] \quad (8)$$

$$\doteq \hat{\phi}_k + \mu_k \sin(\arg x_k - \arg\hat{a}_k - \hat{\phi}_k), \mu_k = \mu |x_k|^2$$

where * denotes complex conjugate and μ is a constant that depends on the signal-noise ratio. The estimator of (8) is called a DDPLL.

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

$$y_k = x_k G_k \quad (9)$$

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

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 $e^{-j\hat{\phi}_k}$ and $|G_k|$ as a gain correction $\hat{c}_k$. 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 rapidly fading variations.

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 $\{\phi_k, a_k\}_1^K$ that, based on the observation sequence $\{x_k\}_1^K$, appears most likely. The application of this idea to phase coherent data communication was first proposed in [1] and refined in [9]. The most likely sequence, call it $\{\hat{\phi}_k, \hat{a}_k\}$, is the sequence that maximizes the natural logarithm (or any other monotone function) of the a posteriori density of $\{\phi_k, a_k\}_1^K$, given the sequence of observations $\{x_k\}_1^K$. Thus we pose the maximization problem:

$$\max_{\{\phi_k\}_1^K, \{a_k\}_1^K} \ell n\, f(\{\phi_k\}_1^K, \{a_k\}_1^K / \{x_k\}_1^K) \quad . \quad (10)$$

Using the result of (5) and (6) we may write

$$f(\{x_k\}_1^K, \{\phi_k\}_1^K, \{a_k\}_1^K) = \prod_{k=1}^{K} N_{x_k}(a_k e^{j\phi_k}, \sigma_n^2) \cdot$$

$$f(\phi_k/\phi_{k-1}) f(\{a_k\}_1^K) \quad . \quad (11)$$

Assuming the $\{a_k\}_1^K$ to be a sequence of independent, equally likely symbols, using (3), and neglecting uninteresting constants, we may write the maximization problem as

$$\max_{\{\phi_k\}_1^K, \{a_k\}_1^K} \Gamma_K$$

$$\Gamma_K = -\frac{1}{2\sigma_n^2} \sum_{k=1}^{K} |x_k - a_k e^{j\phi_k}|^2 + \quad (12)$$

$$\sum_{k=2}^{K} \ell n\, g_1(\phi_k - \phi_{k-1}) + \ell n\, f(\phi_1) \quad .$$

Note that Γ_k satisfies the recursion

$$\Gamma_k = \Gamma_{k-1} + p_k \quad k = 2,3,\ldots \quad (13)$$

$$p_k = -\frac{1}{2\sigma_n^2} |x_k - a_k e^{j\phi_k}|^2 + \ell n\, g_1(\phi_k - \phi_{k-1}), k=2,3,\ldots$$

$$\Gamma_1 = -\frac{1}{2\sigma_n^2} |x_1 - a_1 e^{j\phi_1}|^2 + \ell n\, f(\phi_1)$$

where p_k is the so-called path-metric. For convenience, let us make explicit in Γ_K the last phase and symbol: $\Gamma_K(\phi_K, a_K)$. The other arguments $\{\phi_k\}_1^{K-1}$, $\{a_k\}_1^{K-1}$, remain implicit. Then, from (13)

$$\Gamma_K(\phi_K, a_K) = \Gamma_{K-1}(\phi_{K-1}, a_{K-1}) + p_K(x_K, a_K, \phi_K, \phi_{K-1}) \quad (14)$$

Thus, the maximizing sequence, call it $(\{\hat{\phi}_k\}_1^K, \{\hat{a}_k\}_1^K)$, passing through $(\hat{\phi}_{K-1}, \hat{a}_{K-1})$ on its way to $(\hat{\phi}_K, \hat{a}_K)$, must arrive at $(\hat{\phi}_{K-1}, \hat{a}_{K-1})$ along a route $(\{\hat{\phi}_k\}_1^{K-2}$, $\{\hat{a}_k\}_1^{K-2})$ that maximizes $\Gamma_{K-1}(\hat{\phi}_{K-1}, \hat{a}_{K-1})$. It is this observation that 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 $\Xi = \{\xi_n\}_{n=1}^m$. The function $\ell n\, g_1(\phi_k - \phi_{k-1})$ is then defined on the two-dimensional grid $\Xi \times \Xi$. However, as discussed in [8] and [9] the resulting mxm 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 of the computation lies mainly in the evaluation of mM possible values of $|x_k - a_k e^{j\phi_k}|^2$ at each step of (13). Here M is the symbolling alphabet size and m is the number of discrete phase values. For each of these calculations of $|x_k - a_k e^{j\phi_k}|^2$ there are six (6) real multiplications to be performed (plus the additions). Compared to this computational load of 6mM multiplications, the determination and addition of $2\sigma_n^2\, \ell n\, g_1(\phi_k - \phi_{k-1})$ that appears in (13) is negligible. In fact the latter function would likely be tabulated and stored in ROM. From this discussion it is clear that computation complexity is proportional to mM. When there are many symbols and short-term phase fluctuations have small amplitude (σ_w small), so that m must be large for accurate phase tracking, then the complexity is very large. For example with M=8 and m=48, c is proportional to 384. 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 variable that is the sum of ϕ_k and the symbol phase, arg a_k. And, of course, for PSK symbol sets M may be set to unity because 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. Even this figure may be reduced by using one of a variety of so-called M-algorithms in which all survivor states are saved but only a handful of candidate originator states are considered for each survivor.

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

$$a_k = e^{j\theta_k} \tag{15}$$

with $\{\theta_k\}$ drawn independently from an M-ary equiprobable alphabet $\Theta=\{(\ell=1)2\pi/M\}_{\ell=1}^{M}$. Write the meament model of (1) as

$$x_k = e^{j\psi_k} + n_k \tag{16}$$

where the total phase ψ_k is represented as follows:

$$\psi_k = \phi_k + \theta_k$$

$$\theta_k = \sum_{\ell=1}^{k} \Delta\theta_\ell,\ \Delta\theta_k = \theta_k - \theta_{k-1},\ \Delta\theta_1 = \theta_1 \quad . \tag{17}$$

It is clear that $\hat{\theta}_k = \sum_{\ell=1}^{k} \Delta\hat{\theta}_\ell$ and $\hat{\phi}_k = \hat{\psi}_k - \hat{\theta}_k$. Thus we may replace the MAP sequence estimation problem posed in (12) by the problem

$$\max_{\{\psi_k\}_1^K,\ \{\Delta\theta_k\}_1^K} f(\{x_k\}_1^K,\ \{\psi_k\}_1^K,\ \{\Delta\theta_k\}_1^K) \tag{18}$$

The joint density $f^K \triangleq f(.,.,.)$ in (18) may be written

$$f^K = \prod_{k=1}^{K} N_{x_k}(e^{j\psi_k},\sigma_n^2)\, f(\psi_k,\Delta\theta_k/\{\psi_j\}_1^{k-1},\{\Delta\theta_j\}_1^{k-1}) \tag{19}$$

where for k=1, $f(\psi_1,\Delta\theta_1/.,.)$ is simply the marginal density $f(\psi_1,\theta_1)$. The conditional density on the right hand side of (19) is easily evaluated with Bayes' rule:

$$f(\psi_k,\Delta\theta_k/\{\psi_j\}_1^{k-1},\ \{\Delta\theta_j\}_1^{k-1}) = f(\psi_k/\{\psi_j\}_1^{k-1},\{\Delta\theta_j\}_1^{k}) \cdot f(\Delta\theta_k/\{\psi_j\}_1^{k-1},\{\Delta\theta_j\}_1^{k-1}) \quad . \tag{20}$$

Now $\Delta\theta_k$ is independent of the previous data, additive noise and phase fluctuations. Thus

$$f(\Delta\theta_k/\{\psi_j\}_1^{k-1},\ \{\Delta\theta_j\}_1^{k-1}) = \frac{1}{M} \quad . \tag{21}$$

Moreover if we rewrite ψ_k as

$$\psi_k = \phi_{k-1} + w_k + \theta_{k-1} + \theta_k - \theta_{k-1}$$

$$= \psi_{k-1} + \Delta\theta_k + w_k \quad , \tag{22}$$

we see immediately that

$$f(\psi_k/\{\psi_j\}_1^{k-1},\ \{\Delta\theta_j\}_1^{k}) = g_1(\psi_k - \psi_{k-1} - \Delta\theta_k) \quad . \tag{23}$$

Recall ψ_k is defined on the circle C. Therefore, for emphasis we might think of ψ_k as $\psi_{k-1} + \Delta\theta_k + w_k$, whose density is folded in $[-\pi,\pi)$. Putting this together, we have for the joint density f^K

$$f^K = \prod_{k=1}^{K} N_{x_k}(e^{j\psi_k},\sigma_n^2)\, \frac{1}{M}\, g_1(\psi_k - \psi_{k-1} - \Delta\theta_k)$$

$$\Delta\theta_1 \triangleq \theta_1,\ \psi_0 \triangleq 0 \quad . \tag{24}$$

<u>Principle of Optimality:</u> Call $\{\psi_k\}_1^K$, $\{\Delta\theta_k\}_1^K$ the MAP sequences that maximize f^K; $\{\Delta\theta_k\}_1^K$ enters only in the $g_1(\cdot)$ term on the right hand side of (24). Now let us suppose (as is usual) that $g_1(w)$, 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 (2) has a Gaussian or Cauchy distribution h(w).

It follows that f^K is minimized by choosing

$$\Delta\hat{\theta}_k = [\hat{\psi}_k - \hat{\psi}_{k-1}] \tag{25}$$

where [x] denotes the closest value of $(\ell-1)2\pi/M$ to x. By substitution of the constraint (25) into (24) and defining the "rest" function R(x) on the circle C by

$$R(x) = x - [x] \tag{26}$$

we find that

$$\hat{f}^K = \prod_{k=1}^{K} N_{x_k}(e^{j\psi_k},\sigma_n^2)\, \frac{1}{M}\, g_1(R(\psi_k - \psi_{k-1})) \tag{26}$$

The maximization of $\hat{f}^K$ with respect of $\{\psi_k\}_1^K$ is formally equivalent to maximizing the joint density $\hat{f}(\{x_k\}_1^K,\ \{\psi_k\}_1^K)$ when the total phase ψ_k follows a Markov-model similar to (2):

$$\psi_k = \psi_{k-1} + u_k \tag{27}$$

Here the independent increments u_k have probability density, folded on the circle C,

$$f(u) = \frac{1}{M}\, g_1(R(u)) \quad . \tag{28}$$

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

$$g_1(u) = 0,\ |u| \geq \frac{\pi}{M} \tag{29}$$

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

$$\frac{1}{M} g_1(R(u)) = \frac{1}{M} \sum_{m=1}^{M} \sum_{\ell=-\infty}^{\infty} h[R(u) - \ell 2\pi - (m-1)2\pi/M] \quad . \tag{30}$$

Thus (28) 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 $\hat{f}^K$ we have the maximization problem:

$$\max_{\{\psi_k\}_1^K} \Gamma'_K$$

$$\Gamma'_k = \Gamma'_{k-1} + p'_k;\ \Gamma'_1 = -\frac{1}{2\sigma_n^2}\,|x_1 - e^{j\psi_1}|^2 + \ln g_1(R(\psi_1))$$

$$p'_k = -\frac{1}{2\sigma_n^2} |x_k - e^{j\psi_k}|^2 + \ln g_1[R(\psi_k - \psi_{k-1})] \quad (31)$$

which is solved by the dynamic programming algorithm discussed in Section IV.

Now the complexity of the computation lies mainly in the evaluation of $|x_k - e^{j\psi_k}|^2$ at each step, since the m values of the function $2\sigma_n^2 \ln g_1[R(\psi_k - \psi_{k-1})]$ will be precomputed and stored. For each evaluation of $|x_k - e^{j\psi_k}|^2$ there are two real multipliers (plus the additions). Thus the complexity is proportional to m. Compared to the brute force approach of Section IV the reduction in complexity is actually greater than M (4 and 8, respectively for 4-ary and 8-ary PSK).

Usually, the phase is differentially modulated rather than directly modulated and therefore the relevant symbol is $\Delta\hat{\theta}_k$ itself (see (17)). For the purpose of data transmission, there is no need to reconstruct the absolute data phase $\hat{\theta}_k = \sum_{\ell=1}^{k} \Delta\hat{\theta}_\ell$. This reconstruction has, however, been carried out in the simulations in order to recover the estiamtes $\hat{\phi}_k = \hat{\psi}_k - \hat{\theta}_k$ of the phase fluctuations, and to get the approximate variance of the phase estimates.

This principle of optimality is easily (but tediously) extended to phase-amplitude encoded symbols.

VI . 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 (2) with w_k:$N(0,\sigma_w^2)$ and initial phase uniformly-distributed on $[-\pi,\pi)$. See [10] for results relating to sinusoidal phase jitter. Initial phase acquisition has been achieved by transmitting a preamble according to one of the following schemes.

a) During a pre-transmission period of length N, the sequence of transmitted data is known to the receiver. Thus, in the DBVA and VA systems that are based upon MAP estimation, the Viterbi algorithm works as a pure phase estimator during that 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 that are 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 emitted. This keeps the phase away from severe sudden fluctuations, and makes the joint phase-data estimator able to adequately acquire the initial phase.

The VA reaches the same data-error probability during the emitting period for both methods; i.e. its performance does not depend upon which learning procedure is used. On the other hand, the DBVA is very sensitive to the learning procedure. For example, at a SNR of 20 dB, with phase variance $\sigma_w^2 = 4\ \sigma_n^2$, for a learning period of N=60 data, the number of errors during an emitting period of 490 data values jumps from 7 for procedure a) - known data - to 59 for procedure b) - constant but unknown data. Moreover the DBVA requires a longer learning period than does the VA, roughly twice longer. Namely N=50 is sufficient for the VA, while the DBVA needs N=100 learning iterations. 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, and these data may be unknown to the receiver, provided they are kept constant (procedure b) without degradation.

8-PSK: Shown in Figure 1 are simulation results for 8-PSK when SNR ranges from 16-19dB and $(\sigma_w^2\sigma_n^2)^{1/2}$ remains fixed at 4.4×10^{-3} rad^2. The solid circles correspond to the VA and the solid triangles correspond to the markedly simpler JE. Also shown on Figure 1 are performance bounds for fully coherent 8-PSK and 16-PSK symboling. The values of σ_w^2 under investigation range from 1.6° to 2.2° and the ratio σ_w^2/σ_n^2 is very small ranging from 0.03 to 0.12. In this case neither the VA nor the DBVA provides significant improvement over the JE or DDPLL. The latter two receivers are markedly simpler than the DBVA which, in turn, is markedly simpler than the VA. Therefore for such cases of weak phase noise, the VA is of no interest. A different conclusion is reached for high phase noise cases and phase-amplitude coded symbols, as discussed in the next example.

16-QASK: Shown in Figures 2, 3 and 4 are simulation results for 16-QASK symbols encoded according to the (4,4) CCITT rule. The decoding procedures are JE, DDPLL, DBVA and VA, for three distinct values of the ratio σ_w^2/σ_n^2. Figure 2 is concerned with a weak phase noise (σ_w^2/σ_n^2 = 0.25), Figure 3 is concerned with an average phase noise (σ_w^2/σ_n^2 = 1) and Figure 4 is concerned with a large phase noise (σ_w^2/σ_n^2 = 4). We recall [1] that the DBVA performs some kind of phase estimation along a path that satisfies

$$\hat{\psi}_n = \hat{\psi}_{n-1} \pm \sigma_w \quad , \quad (32)$$

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 6 or 8. 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].

VII. Conclusions

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 to face a phase process that is a random walk, i.e. a very severe type of phase fluctuation with rapid and large variations, and in addition the possibility of very large peaks. For such severe cases, the VA gives excellent performance. However, the number of discretized phase levels, and thus the complexity of the VA increases as the phase fluctuations decrease, i.e. when the transmission conditions improve. Thus its use must be restricted to the cases of severe phase noise, which happen usually when the data diagram has many points (8, 16 or more).

Finally, we remark that the robustness of the VA, DDPLL, and JE appears superior to that of the DBVA. This issue will be explored more fully in a forthcoming paper.

References

1. G. Ungerboeck, "New Application for the Viterbi Algorithm: Carrier Phase Tracking in Synchronous Data Transmission Systems," Nat. Telecomm. Conf., pp. 734-738 (1974).

2. H. Kobayaski, "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems," IEEE Trans. Comm. Tech., COM-19, pp. 268-289 (June 1971).

3. M. Levy and O. Macchi, "Auto-Adaptive Phase Jitter and Interference Intersymbol Suppression for Data Transmission Receivers," Nat. Telecomm. Conf., Dallas (November 1976).

4. O. Macchi, M. Levy, C. Macchi, "Improvements in Systems for Transmitting Data Between Distant Locations," U.S. patent application no. 803245.

5. D. D. Falconer, "Analysis of a Gradient Algorithm for Simultaneous Passband Equalization and Carrier Phase Recovery," BSTJ, 55, pp. 409-428 (1976).

6. F. R. Magee, "Simultaneous Phase Tracking and Detection in Data Transmission over Noisy Dispersive Channels," IEEE Trans. on Comm., pp. 712-715 (July 1977).

7. A. S. Willsky, "Fourier series and Estimation on the Circle with Applications to Synchronous Communications-Part I: Analysis," IEEE Trans. Inform. Theory IT-20, pp. 577-583 (September 1974).

8. L. L. Scharf, D. D. Cox, C. J. Masreliez, "Modulo-2π Phase Sequence Estimation," IEEE Trans. Inform. Theory (Submitted April 1978).

9. L. L. Scharf, "A Viterbi Algorithm for Modulo-2π Phase Tracking in Coherent Data Communication Systems," IEEE Trans. Comm. (Submitted December 1977).

10. O. Macchi, S. Kerbrat, L. Scharf, "A Joint Phase Estimation and Data Decoding Viterbi Procedure for Phase Jitter Channels," Inter. Conf. on Digital Signal Processing, edited by Cappellini and Constantinides, Florence, Sept. 1978.

Figures

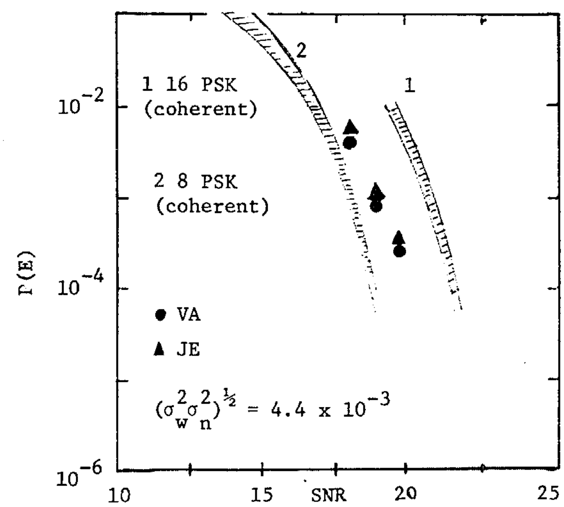

Fig. 1 P(E) vs. SNR. 8-ary PSK

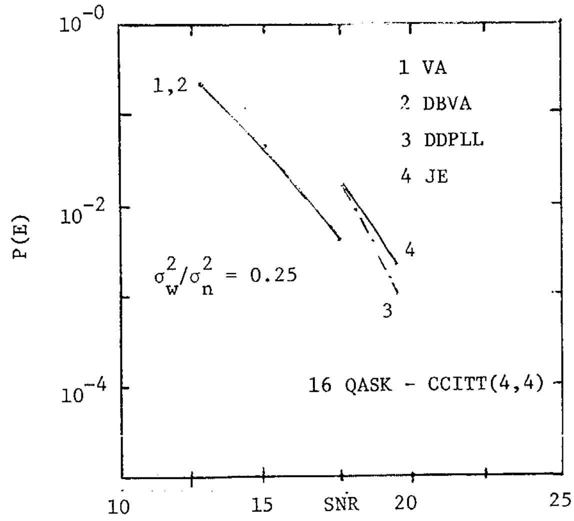

Fig. 2 P(E) vs. SNR. Small Phase Noise

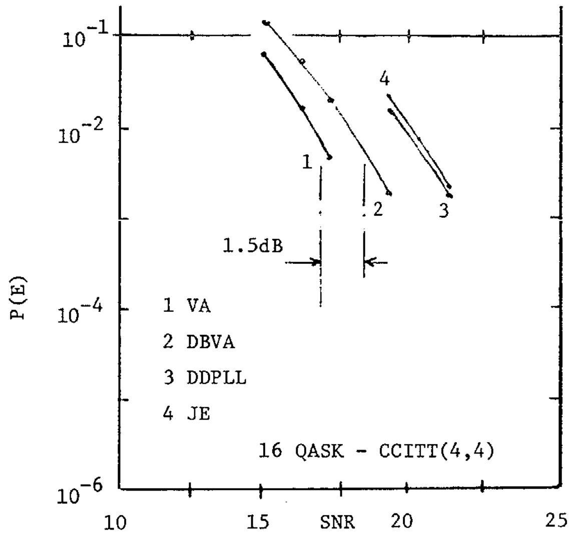

Fig. 3 P(E) vs. SNR. Moderate Phase Noise

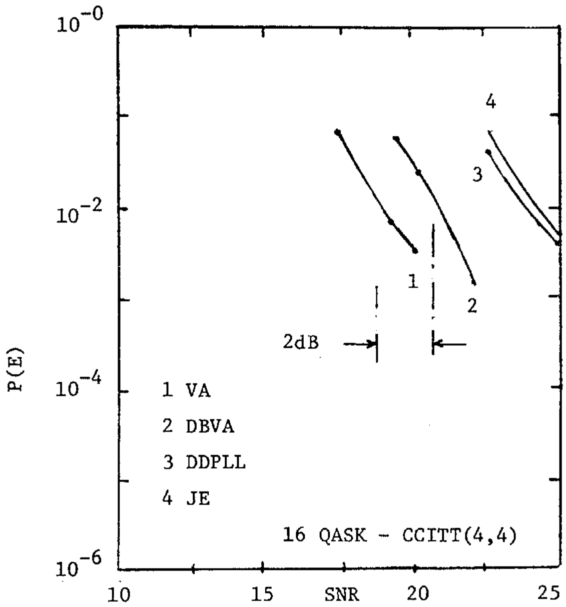

Fig. 4 P(E) vs. SNR. Strong Phase Noise