Feedforward Frequency Estimation for PSK: a Tutorial Review

MICHELE MORELLI, UMBERTO MENGALI
Dipartimento di Ingegneria della Informazione, Università di Pisa
Via Diotisalvi 2, 56100 Pisa - Italy
morelli, mengali@iet.unipi.it

Abstract. This paper considers feedforward carrier frequency estimation methods in burst-mode digital transmission, with tutorial objectives foremost. Assuming PSK modulation, two scenarios are envisaged in which frequency estimates are derived either from a preamble appended to the data block, or directly from the modulated signal. Several estimation algorithms are considered with different and somewhat contrasting characteristics. The characteristics we focus on are: estimation accuracy, estimation range, minimum operating signal-to-noise ratio (threshold), and implementation complexity. They provide a framework within which current estimation methods can be evaluated. The paper reviews and compares some prominent algorithms proposed in literature, trying to single out the best ones for a given application.

1. INTRODUCTION

Burst-mode transmission of digital data is employed in many applications such as satellite time-division multiple-access (TDMA) techniques and terrestrial mobile cellular radio. In conventional systems a preamble of known symbols is placed somewhere in each burst for carrier and clock recovery purposes. The use of the preamble allows data-aided (DA) operation and results in superior performance in comparison with non-data-aided (NDA) methods. Even so, synchronization may prove difficult, especially with coded modulations. For example, with QPSK signaling and rate 1/2, constraint length 7, convolutional coding, the signal-to-noise ratio E_s/N_0 needed to achieve a bit error rate of 10^{-6} is only 5 dB. Since it is desirable that the transmission system continues to operate even at higher error rates, a minimum of 2-3 dB is often set as a design target. Clearly, very efficient synchronization algorithms are needed in these conditions.

Ideally, preambles should be as short as possible for they reduce the transmission rate. For example, current trends in satellite transmission for LAN interconnection indicate that an overhead of 10% is quite feasible [1]. Clearly, the next step is to get rid of preambles altogether and estimate the synchronization parameters in an NDA fashion [2]. This route raises the following question. Lack of data information is expected to degrade synchronization accuracy *for a fixed estimation time*. On the other hand, the estimation time does not need to be as short as when a preamble is used. In fact the whole data burst is available, not just a segment of it. Then, one wonders whether the extended estimation time will compensate for the NDA operation. The answer is not obvious and depends on the specific algorithms being used. For example, in [3] it is shown that carrier phase can be efficiently estimated without any preamble.

In this paper we concentrate on carrier frequency estimation with PSK signaling. Our aim is to compare various feedforward estimation algorithms, either DA or NDA, that have been proposed in literature. In doing so we describe their features in terms of four performance indexes:

i) *estimation accuracy*;
ii) *estimation range*;
iii) *threshold (the critical signal-to-noise ratio below which large estimation errors begin to occur)*;
iv) *implementation complexity*.

It is worth noting that these indexes may be in contrast with each other. For example, achieving a low threshold implies a high complexity. Likewise, good estimation accuracy is often met at the price of a narrow estimation range. So, a trade off is called for between conflicting requirements and a judicious choice between various options can only be made by carefully specifying the actual operating conditions.

The remainder of the paper is organized as follows. The next section concentrates on the signal model and describes the statistics of the sequence $[z(k)]$ as obtained

by eliminating the modulation from the samples at the matched filter output. Section 3 describes an algorithm derived from maximum likelihood criteria. Simpler methods are indicated in section 4. Other algorithms, based on correlation calculations, are reported in section 5. Computational complexity considerations are addressed in section 6. Finally, some conclusions are offered in section 7.

2. Signal Model

In this study we make the following major assumptions. The modulation is PSK and the channel noise is additive, white and Gaussian, with two-sided power spectral density $N_0/2$. The channel filtering is equally apportioned between transmitter and receiver and the overall channel response is Nyquist *in the absence of frequency errors*. Clock recovery is ideal. Indeed, excellent timing information can normally be derived even with frequency errors on the order of 10-20% of the symbol rate. In these conditions it is readily shown that the samples from the matched filter are given by

$$x(k) = c_k\, e^{j(2\pi f_d kT+\theta)} + n(k) \tag{1}$$

where $\{c_k\}$ are data symbols from the PSK alphabet $\{e^{j2\pi m/M};\ m = 0, 1, \ldots, M-1\}$, f_d is the carrier frequency offset we want to estimate, θ represents the carrier phase, T is the symbol period, $[n(k)]$ are zero-mean Gaussian random variables with independent real and imaginary components, each of variance $(E_s/N_0)^{-1}/2$, and E_s is the signal energy per symbol. The phase θ is uniformly distributed over $[0, 2\pi)$.

It should be stressed that eq. (1) holds true only with rather limited values of f_d (on the order of few percents of the symbol rate). In fact the right hand side does not reflect the mismatching between the incoming signal and the receive filter due to frequency errors. Actually, an exact expression for $x(t)$ is

$$x(k) = h(0)c_k\, e^{j(2\pi f_d kT+\theta)} + \sum_{i\neq k} h\left[(k-i)T\right]c_i\, e^{j(2\pi f_d iT+\theta)} + n(k) \tag{2}$$

where $h(t)$ is the convolution

$$h(t) \triangleq \left[g(t)e^{j2\pi f_d t}\right] \otimes g(-t) \tag{3}$$

and $g(t)$ is the shape of the modulation pulses. Bear in mind that we have assumed a Nyquist $h(t)$ for $f_d = 0$, i.e.,

$$g(t) \otimes g(-t)\big|_{t=kT} = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{4}$$

Comparing (1) and (2) we see that they coincide only for very small frequency offsets. Otherwise, they are different because of the presence of intersymbol interference in (2) and a reduction in the useful signal amplitude ($|h(0)| < 1$). For heuristic reasons in the sequel we adopt the model (1) but the effects of intersymbol interference will be pointed out in due time.

It is clear from (1) that the samples $x(t)$ depend on the modulation. As most frequency estimation algorithms are tailored for unmodulated carriers, the data symbols must be wiped out in some way. Two different approaches can be followed to do so, depending on whether the trasmitted symbols $\{c_k\}$ are known or not. The first instance corresponds to DA algorithms and is handled by multiplying both sides of (1) by c_k^* (the superscript "star" means complex conjugate) to yield

$$z(k) = x(k)\, c_k^* \qquad \text{DA operation} \tag{5}$$

from which, bearing in mind that c_k has unit amplitude, we get

$$z(k) = e^{j(2\pi f_d kT+\theta)} + n'(k) \tag{6}$$

where $n'(k) \triangleq n(k)\, c_k^*$ is a noise sequence statistically equivalent to $[n(k)]$. Eq. (6) indicates that $z(k)$ is a sine wave embedded in noise. In the next sections we discuss how to estimate its frequency from the observation of $[z(k)]$.

In the absence of a preamble the modulation can be removed by feeding $x(k)$ into some *ad hoc* non linearity [3]. The simplest and only type of non linearity we consider in this study consists of raising $x(k)$ to the M-power (M is the number of modulation levels) and scaling the result to unit amplitude. Formally, the output of the non linearity is related to the input $x(k)$ by

$$z(k) = e^{jM\arg[x(k)]} \qquad \text{NDA operation} \tag{7}$$

It is readily checked that (7) may also be written as

$$z(k) = e^{jM(2\pi k f_d T+\theta+\eta_k)} \tag{8}$$

where

$$\eta_k \triangleq \arg\left[1 + n''(k)\right] \tag{9}$$

$$n''(k) \triangleq n'(k)\, e^{-j(2\pi k f_d T+\theta)} \tag{10}$$

Thus, we have again a sine wave but, in comparison to (6), its frequency is M times larger and the channel noise affects $z(k)$ only through a non-Gaussian phase disturbance η_k.

It is interesting to compare DA and NDA operations at high SNR. To this purpose let us rewrite (6) in the form

$$z(k) = \left[1 + n''(k)\right] e^{j(2\pi f_d kT+\theta)} \tag{11}$$

or, equivalently,

$$z(k) = \left|1 + n''(k)\right|\, e^{j(2\pi f_d kT+\theta+\eta_k)} \tag{12}$$

Clearly, the amplitude of $n''(k)$ becomes smaller and smaller relative to unity as E_s/N_0 increases. Then, (12) tends to (8) with $M = 1$. In other words, at high *SNR* the model (8) is valid for both DA and NDA operations provided that we set $M = 1$ when dealing with DA.

3. Rife and Boorstyn algorithm

Having established a statistical model for the observables $[z(k)]$, we now concentrate on frequency estimation algorithms. A powerful method has been indicated by Rife and Boorstyn (R&B) in [4]. Let us see how it works when the data are available (DA operation). Taking (6) as an exact model (1), Rife and Boorstyn have shown that the maximum likelihood (ML) estimate of f_d is the location where the amplitude of

$$Z(f) \triangleq \frac{1}{L_0} \sum_{k=0}^{L_0-1} z(k) e^{-j2\pi kfT} \tag{13}$$

achieves a maximum, L_0 being the observation length in symbol intervals. Formally, the R&B estimator reads

$$\hat{f}_d = \arg\left\{ \max_f \left[|Z(f)| \right] \right\} \tag{14}$$

From (6) it is readily checked that $\hat{f}_d$ is independent of the carrier phase θ. It is also evident that $Z(f)$ is a periodic function of period $1/T$. This implies that the estimates provided by (14) are ambiguous by multiples of the symbol rate or, in other words, that the estimation range is $\pm 1/(2T)$.

One difficulty with locating the maximum of $Z(f)$ is apparent from Fig. 1 a) which illustrates a typical realization of $|Z(f)|$ as obtained by simulation with QPSK signaling and root-raised-cosine-rolloff (RRCR) pulses with rolloff $\alpha = 0.5$. The observation length is $L_0 = 64$ and E_s/N_0 is 10 dB. Also, the normalized frequency offset f_dT is chosen equal to 0.1. As there are many local maxima, the largest maximum must be sought in two steps. The first one (*coarse search*) calculates $Z(f)$ over a discrete set of f-values covering the uncertainty range of f_d and determines that f which maximizes $|Z(f)|$. The second step (*fine search*) interpolates between samples of $|Z(f)|$ and computes the local maximum nearest to the f-value picked up earlier. Note that, occasionally, $|Z(f)|$ will be so distorted by noise that its highest peak will be far from f_d. When this happens the R&B algorithm makes large errors (*outliers*). The *SNR* below which the outliers start to occur is referred to as the *threshold* of the estimator.

Fig. 1 b) illustrates a realization of $|Z(f)|$ as obtained with the same parameters of Fig. 1 a), except that E_s/N_0 is now −10 dB. An outlier is clearly evident. Therefore,

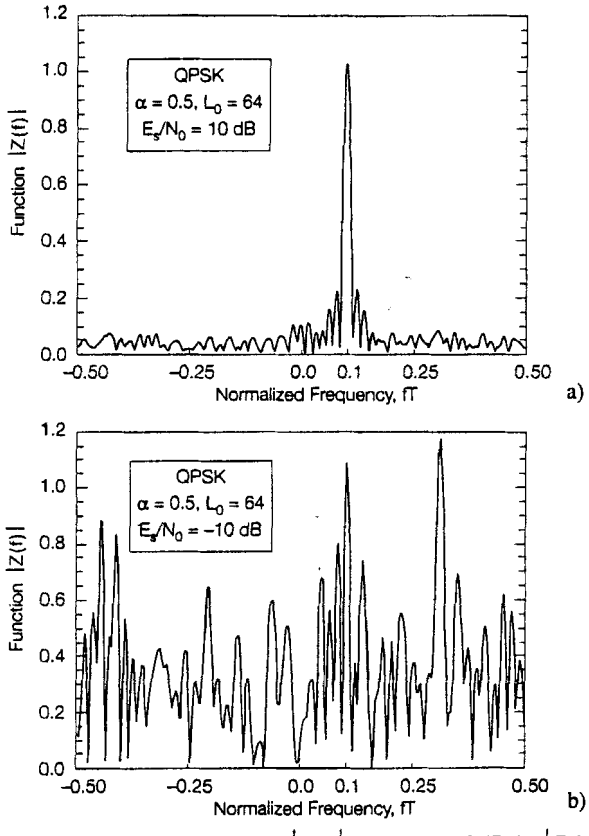

Fig. 1 a) A typical realization of $|Z(f)|$ for $E_s/N_0 = 10$ dB; b) $|Z(f)|$ for $E_s/N_0 = -10$ dB

the coarse search will provide an estimate of f_d of about $0.3/T$ instead of $f_d = 0.1/T$. Since large errors have disabling effects on the receiver performance, frequency estimators must operate *above* threshold.

In practice the coarse search can be efficiently performed using Fast Fourier Transform (FFT) techniques [4], as is now explained. First, $z(k)$ is zero-padded up to some length L_0K, yielding the sequence

$$z'(k) \triangleq \begin{cases} z(k) & 0 \le k \le L_0 - 1 \\ 0 & L_0 \le k \le L_0 K - 1 \end{cases} \tag{15}$$

where K is a parameter called *pruning factor*. Second, the FFT of $[z'(k)]$ is computed at the points

$$f_n = \frac{n}{KL_0T} \qquad -\frac{KL_0}{2} \le n < \frac{KL_0}{2} \tag{16}$$

to produce the set $[Z(f_n)]$. Finally, the largest $|Z(f_n)|$ is sought and this gives the coarse frequency estimate.

The choice of the pruning factor considerably affects the estimation performance. Rife and Boorstyn recommend K values in the range from 2 to 8 to keep the threshold low. In these conditions their estimator turns out to be unbiased in the range $\pm 1/(2T)$.

Fig. 2 illustrates the accuracy of the R&B algorithm with QPSK modulation and root-raised-cosine-rolloff pulses with 50% rolloff, say RRCR(50%). The ordinates give the estimation error variance normalized to the squared symbol rate. The pruning factor is $K = 4$

(1) As mentioned in section 2, eq. (6) is only valid with small frequency offsets. When f_d increases, the receive filter is no longer matched to the incoming pulses and the more complex model (2) must be taken into account.

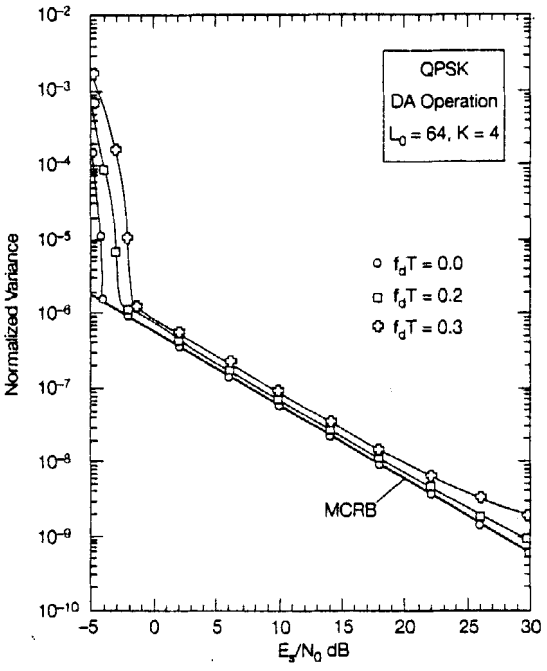

Fig. 2 - Accuracy of R&B algorithm with DA operation.

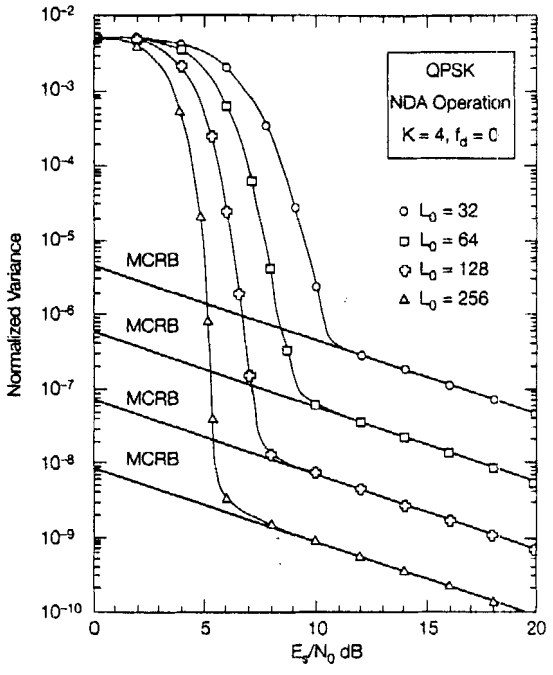

Fig. 3 - Accuracy of R&B algorithm with NDA operation.

and the observation interval is of 64 symbols. The lowest line represents the *modified* Cramer-Rao bound (MCRB) [5] which is given by

$$T^2 \times \mathrm{MCRB}(f_d) = \frac{3}{2\pi^2 L_0^3} \frac{1}{E_s/N_0} \tag{17}$$

We see that the estimation accuracy keeps quite close to the bound up to E_s/N_0 values below 0 dB. If E_s/N_0 is decreased further, however, a rapid increase in the error variance is observed. The abscissa at which the slope of the curve starts to change indicates the estimator threshold and is a manifestation of the occurrence of outliers. The mismatching between the incoming signal and the receiving filter deteriorates the performance as f_d increases.

Next we turn our attention to NDA operation. Recalling the comments at the end of the previous section it is clear that the R&B estimator can be used even for NDA operation provided that the sequence $[z(k)]$ is computed from (7) rather than (5). Also, bearing in mind that eq. (8) holds for either DA and NDA at high *SNR* (with M = 1 for DA), it can be shown that NDA operation leads to an estimation ambiguity by multiples of $1/MT$, not $1/T$ as happens with DA operation. For example, with 8PSK modulation the estimation range reduces from $\pm 0.5/T$ to $\pm 0.0625/T$ in passing from DA to NDA.

Fig. 3 shows simulation results for NDA operation. The modulation is still QPSK, the pruning factor is K = 4 and the frequency offset is chosen equal to zero. The modified Cramer-Rao bounds corresponding to the various observation lengths are indicated. We see that the threshold is a decreasing function of L_0. Approximately, doubling L_0 results in a threshold decrease by 2 dB. This feature of the R&B estimator is of great importance with coded modulation for it makes this estimator suitable for operation at very low *SNR* (by adequately increasing the observation length). As we shall see, other estimators have a threshold almost independent of L_0 and, in consequence, they can only be employed at intermediate/high *SNR*.

Fig. 4 illustrates further results with NDA operation. Everything is as in Fig. 3, except that the modulation is 8PSK. The curves have still the same shape but, as expected, the threshold is now considerably higher for a given observation length.

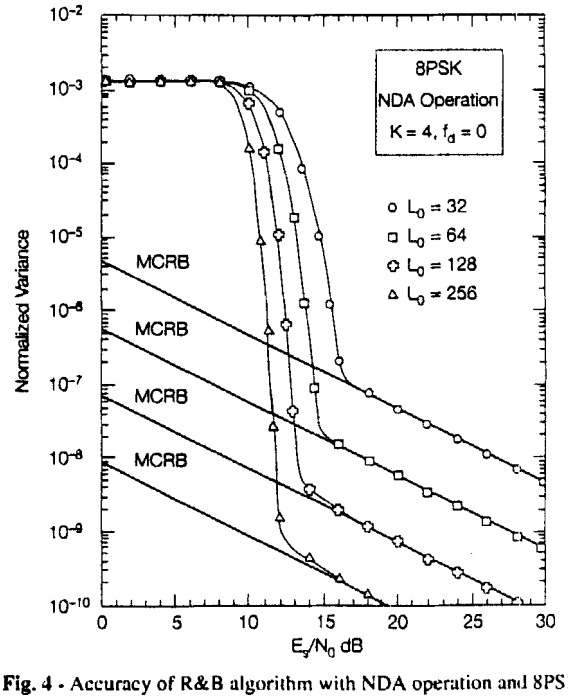

Fig. 4 - Accuracy of R&B algorithm with NDA operation and 8PSK.

4. LEAST-SQUARES-BASED ESTIMATORS

4.1. *Tretter Estimator*

The R&B estimator has very good performance but, as we shall see later, has a rather high computational complexity. Simpler methods are desirable and, in fact, a number of alternatives have been proposed in literature. In this section we describe two schemes derived from least squares estimation criteria. Our goal is to indicate their logical basis and give an idea of their performance. In pursuing this task we shall employ the model (8) which is valid at *high SNR* for either DA and NDA operation. It should be stressed that adopting this model has only heuristic purposes for, in effect, the algorithms we shall obtain are useful at *any SNR* (not necessarily *high*).

The first algorithm has been proposed by Tretter [6] and is based on the following considerations. From (8) we see that

$$\arg\,[z(k)] = [M\,(2\pi k f_d T + \theta + \eta_k)]_{-\pi}^{\pi} \tag{18}$$

where the modulo-2π operation $[x]_{-\pi}^{\pi}$ means that x is reduced to the interval $(-\pi, \pi]$. Suppose for the moment that the variable $M\,(2\pi k f_d T + \theta + \eta_k)$ is within the interval $(-\pi, \pi]$ (which may not be true since the amplitude of $M\,(2\pi k f_d T + \theta + \eta_k)$ increases unboundedly with k). Then the modulo-2π operation in (18) is immaterial and the right hand side may be viewed as noisy samples of a straight line with slope $M2\pi f_d T$. Clearly, estimating the slope of this line amounts to estimating f_d, for the two quantities are proportional. Tretter [6] approaches this problem by least squares methods and comes up with the following solution

$$\hat{f}_d = \frac{1}{2\pi MT}\sum_{k=0}^{L_0-1} w_k^{(T)} \arg\left[z(k)\right] \tag{19}$$

where $[w_k^{(T)}]$ are weighting coefficients given by

$$w_k^{(T)} = \frac{6\left(2k - L_0 + 1\right)}{L_0\left(L_0^2 - 1\right)} \qquad 0 \le k \le L_0 - 1 \tag{20}$$

To reiterate, eq. (19) is valid for either DA and NDA operation. In the former, the parameter M equals unity and $z(k)$ is computed from (5); in the latter, M equals the number of points in the signal constellation and $z(k)$ is computed from (7).

A drawback with (19) is that the modulo-2π operation cannot be ignored as it produces jumps by 2π in the trajectory of arg $[z(k)\,]$ when $M\,(2\pi k f_d T + \theta + \eta_k)$ crosses odd multiples of π. Luckily the jumps can be eliminated by unwrapping the sequence $\{\arg\,[z(k)]\}$. The unwrapping algorithm produces a new sequence $[\phi^{(\mathrm{un})}(k)]$ which is related to $\{\arg\,[z(k)]\}$ by

$$\phi^{(\mathrm{un})}(k) = \phi^{(\mathrm{un})}(k-1) + \{\arg\,[z(k)] - \arg\,[z(k-1)]\}_{-\pi}^{\pi} \tag{21}$$

With this adjustment the Tretter estimator takes its final form

$$\hat{f}_d = \frac{1}{2\pi MT}\sum_{k=0}^{L_0-1} w_k^{(T)} \phi^{(\mathrm{un})}(k) \tag{22}$$

4.2. *Kay estimator*

Kay [7] has shown that the unwrapping process can be obviated if the sequence $\{\arg\,[z(k)\,z^*\,(k-1)]\}$ is used in place of $\{\arg\,[z(k)]\}$. To see how this comes about observe that

$$\arg\,[z(k)\,z^*\,(k-1)] = \{\arg\,[z(k)] - \arg\,[z\,(k-1)]\}_{-\pi}^{\pi} \tag{23}$$

and, in general,

$$[(x)_{-\pi}^{\pi} - (y)_{-\pi}^{\pi}]_{-\pi}^{\pi} = (x-y)_{-\pi}^{\pi} \tag{24}$$

Then, from (18) we have

$$\arg\,[z(k)\,z^*\,(k-1)] = [2\pi M f_d T + M\,(\eta_k - \eta_{k-1})]_{-\pi}^{\pi} \tag{25}$$

Comparing with (18) it appears that the useful term in the right hand side in (18) has been turned into a constant, $2\pi M f_d T$. In consequence, the chance of phase jumps has been greatly reduced, especially if $2\pi M\left|f_d\right|T$ is well internal to $\pm\pi$ and the *SNR* is high. Then, paralleling the arguments leading to (19) produces the Kay estimator

$$\hat{f}_d = \frac{1}{2\pi MT}\sum_{k=1}^{L_0-1} w_k^{(K)} \arg\left[z(k)\,z^*\,(k-1)\right] \tag{26}$$

where the weights $w_k^{(K)}$ are given by

$$w_k^{(K)} = \frac{6k\left(L_0 - k\right)}{L_0\left(L_0^2 - 1\right)} \qquad 1 \le k \le L_0 - 1 \tag{27}$$

An essentially identical algorithm has been proposed by Bellini, Molinari and Tartara in [8].

It is a simple matter to show that Tretter and Kay estimators are equivalent. To see why, let us compare (21) with (23). We have

$$\arg\,[z(k)\,z^*\,(k-1)] = \phi^{(\mathrm{un})}(k) - \phi^{(\mathrm{un})}(k-1) \tag{28}$$

Next, substituting into (26) yields

$$\hat{f}_d = \frac{1}{2\pi MT}\left[w_{L_0-1}^{(K)}\,\phi_{L_0-1}^{(\mathrm{un})} - w_1^{(K)}\phi_0^{(\mathrm{un})} + \sum_{k=1}^{L_0-2}\left(w_k^{(K)} - w_{k+1}^{(K)}\right)\phi_k^{(\mathrm{un})}\right] \tag{29}$$

On the other hand it is readily checked from (20) and (27) that $w_{L_0-1}^{(K)} = w_{L_0-1}^{(T)}$, $w_1^{(K)} = -w_0^{(T)}$ and

$$w_k^{(K)} - w_{k+1}^{(K)} = w_k^{(T)} \qquad 1 \le k \le L_0 - 2 \tag{30}$$

Thus, (29) is identical to (22) and our claim is proved.

Having established the equivalence between Tretter and Kay algorithms, in the sequel we concentrate on the latter. For convenience we start with DA operation, which implies that the $z(k)$ are computed from (5). Fig. 5 shows simulations for the expectation of $\hat{f}_d T$ as a function of the actual frequency offset. We see that the estimates are unbiased over a range that gets wider as the *SNR* increases. With an infinite *SNR* the range becomes ([2]) $\pm 0.5/T$.

A physical explanation of this fact is as follows. Assume $SNR = \infty$ and $|f_d| \le 0.5/T$. From (6) we have

$$\arg [z(k)\, z^*(k-1)] = 2\pi f_d T \tag{31}$$

Then, substituting into (26) (with $M = 1$) and bearing in mind that the weights $w_k^{(K)}$ add to unity, we see that f_d equals f_d, in agreement with the simulations. With a finite *SNR* the problem is more complex because the modulo-2π operation in (25) comes into play ([3]). To get some insight into the problem suppose that $2\pi f_d T$ is a little smaller than π and the *SNR* is large. Then, depending on the noise realizations, $2\pi f_d T + \eta_k - \eta_{k-1}$ may either be confined to $\pm\pi$ or it falls *to the right* of π. In the first instance $\arg [z(k)\, z^*(k-1)]$ equals $2\pi f_d T + \eta_k - \eta_{k-1}$; in the latter it equals $2\pi f_d T - 2\pi + \eta_k - \eta_{k-1}$ (as a consequence of the modulo-2π operation). On average $\arg [z(k)\, z^*(k-1)]$ is less than $2\pi f_d T$ and this agrees with the simulations.

The above considerations are readily extended to NDA operation with the conclusion that the estimation range is $\pm 0.5/MT$ for $SNR = \infty$ and gets narrower as *SNR* decreases.

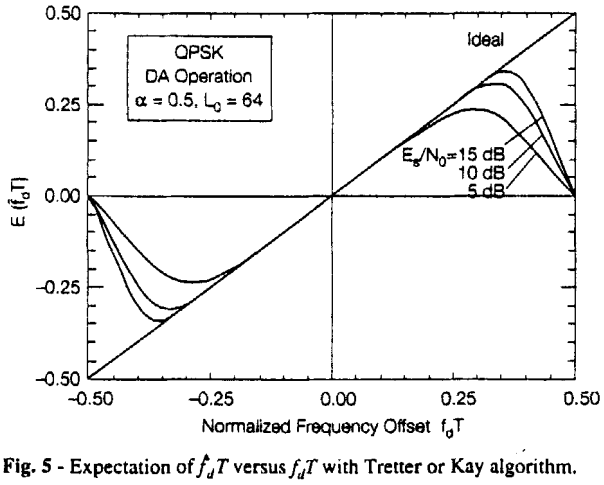

Fig. 5 - Expectation of $\hat{f}_d T$ versus $f_d T$ with Tretter or Kay algorithm.

([2]) Actually this is only true if the receive filter mismatching due to the frequency offset is ignored. In practice this approximation is not valid and the estimation range is narrower than $\pm 0.5/T$.

([3]) It is understood that $M = 1$ in (25) for we are considering DA operation.

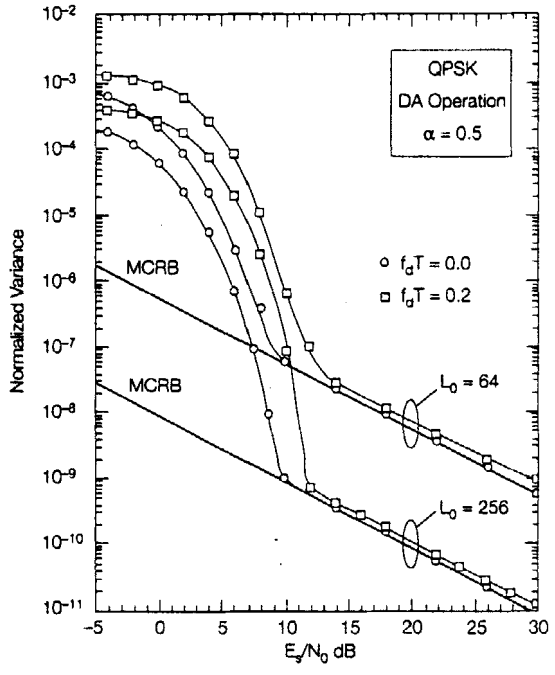

Fig. 6 - Accuracy of Kay algorithm with DA operation.

Fig. 6 illustrates the estimation accuracy of Kay algorithm with DA operation for L_0 equal to 64 and 256. The simulation model is still as in Fig. 5. We see that the threshold is the same in both cases, which means that it cannot be lowered by increasing the observation length, as happens with the R&B algorithm.

Fig. 7 compares DA versus NDA estimation variance with Kay algorithm for QPSK modulation. The parameter f_d is set to zero. We see that NDA operation makes the threshold increase by about 6 dB.

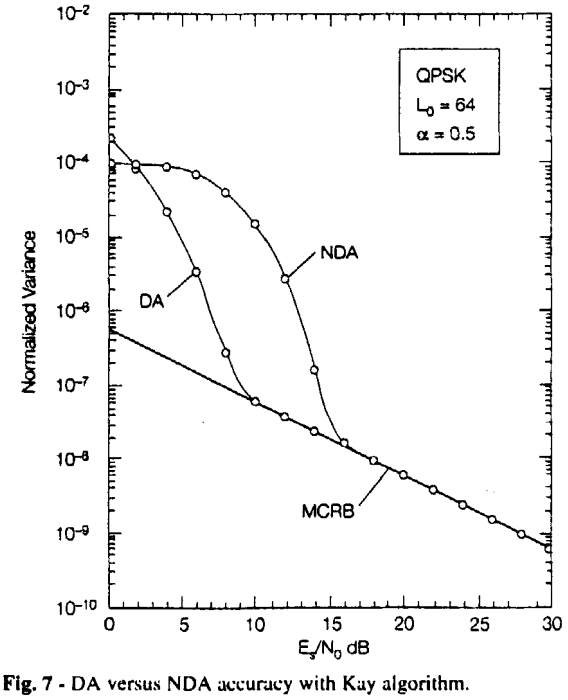

Fig. 7 - DA versus NDA accuracy with Kay algorithm.

5. AUTOCORRELATION-BASED ESTIMATORS

5.1. *Luise and Reggiannini estimator*

We have seen that Kay and Tretter algorithms are not a valid alternative to R&B method for they exhibit a rather high threshold that makes them ill-suited for most applications in digital transmission. In this section we report on alternative methods, all based on the sample autocorrelation of the sequence $[z(k)]$

$$R(m) \triangleq \frac{1}{L_0 - m} \sum_{k=m}^{L_0-1} z(k) z^*(k-m) \tag{32}$$

Again, we point out that $z(k)$ must be computed from (5) or (7) depending on whether DA or NDA operation is intended. Let us start with the estimation method proposed by Luise and Reggiannini (L&R) in [9]:

$$\hat{f}_d = \frac{1}{\pi M T (N+1)} \arg\left[\sum_{m=1}^{N} R(m) \right] \tag{33}$$

where N is a design parameter and M is either unity (with DA operation) or equal to the number of signal constellation points (with NDA operation).

An intuitive explanation of (33) is as follows. For $SNR >> 1$ the autocorrelation $R(m)$ takes the form [4]

$$R(m) = e^{j2\pi M m f_d T} [1 + j\gamma(m)] \tag{34}$$

where $\gamma(m)$ is a zero-mean random variable which, statistically, is much less than unity. Thus, $R(m)$ is approximately equal to $e^{j2\pi M m f_d T}$ and the sum in the right hand side of (33) becomes

$$\sum_{m=1}^{N} R(m) \approx \rho(f_d T) e^{j\pi M (N+1) f_d T} \tag{35}$$

with

$$\rho(f_d T) \triangleq \frac{\sin(\pi M N f_d T)}{\sin(\pi M f_d T)} \tag{36}$$

Note that $\rho(f_d T)$ is a bell shaped function of f_d which takes positive values in the range $|f_d| \leq 1/(MNT)$. Therefore, in this range, the argument of (35) equals $\pi M (N+1) f_d T$ and eq. (33) yields the correct estimate, f_d.

Important issues about L&R algorithm are the estimation range and the role of the parameter N. At high SNR the estimation range is clearly $\pm 1/(MNT)$. Beyond these limits, in fact, $\rho(f_d T)$ takes negative values and this makes the estimates incorrect. Also, in [9] it is shown that the estimation variance considerably depends on N and achieves a minimum for $N = L_0/2$. This minimum coincides with the MCRB at high SNR.

Unfortunately, choosing $N = L_0/2$ may result in a too narrow estimation range for practical purposes. For example, suppose we want an estimation variance of $10^{-8}/T$ at $E_s/N_0 = 10$ dB with NDA operation and QPSK modulation (say, with RRCR(50%) pulses). From Fig. 3 we see that this corresponds to the MCRB for $L_0 = 128$. The L&R estimator achieves this bound for $N = L_0/2 = 64$. However, the corresponding estimation range is $\pm 1/(MNT) \approx \pm 4 \cdot 10^{-3}/T$, which is insufficient in many applications. In summary, estimation accuracy and estimation range are contrasting characteristics in the L&R estimator. A trade off must be sought on the basis of the actual design requirements.

Fig. 8 illustrates the accuracy of L&R algorithm for some values of L_0. The modulation is QPSK as in Fig. 7 and the estimation operation is NDA. The true frequency error is zero. The parameter N is chosen equal to 16 for any L_0 so that the estimation range is constant and equal $\pm 0.015/T$. We see that the MCRB is achieved only for $L_0 = 32$, corresponding to the condition $N = L_0/2$. With longer observation intervals the variance is always above the MCRB. It is difficult to gather from the figure the exact position of the threshold because the curves have smoothly varying slopes. Precise threshold measurements can only be made looking for the first appearance of outliers as the SNR decreases. For example, it is found that the threshold for $L_0 = 32$ occurs at 9 dB while for $L_0 = 256$ it decreases to 5.5 dB. This says that the L&R algorithm can operate at low SNR provided that sufficiently long observation intervals are used.

5.2. *Fitz estimator*

The following alternative estimator has been proposed by Fitz in [10]

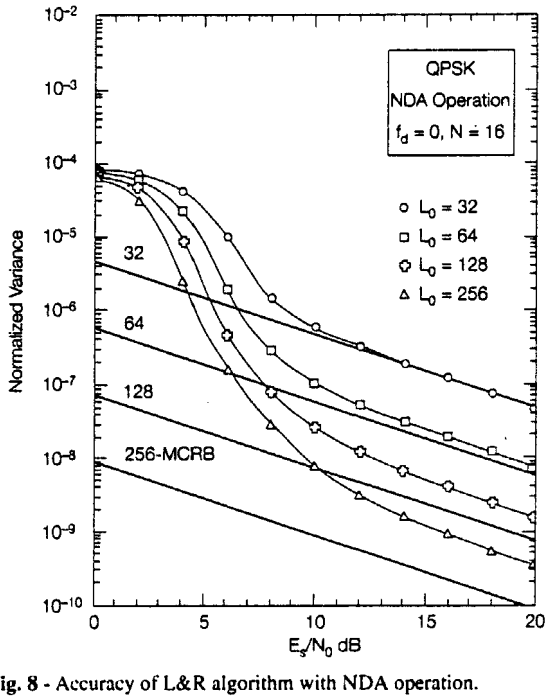

Fig. 8 - Accuracy of L&R algorithm with NDA operation.

(4) This formula is readily derived by inserting (8) into (32) and bearing in mind that the noise-induced disturbance η_k is small for $SNR >> 1$.

$$\hat{f}_d = \frac{1}{2\pi MT} \sum_{m=1}^{N} w_m^{(F)} \arg\left[R(m)\right] \tag{37}$$

where N is a design parameter and $[w_m^{(F)}]$ are weighting coefficients given by

$$w_m^{(F)} = \frac{6m}{N(N+1)(2N+1)} \qquad 1 \le m \le N \tag{38}$$

An intuitive explanation of Fitz algorithm is as follows. Assuming $SNR >> 1$, from (34) we have

$$\arg\,[R(m)] = [2\pi M m f_d T + \mu(m)]_{-\pi}^{\pi} \tag{39}$$

with

$$\mu(m) \triangleq \arg\,[1 + j\gamma(m)] \tag{40}$$

Now, suppose that $2\pi M m f_d T$ is well internal to the interval $\pm\pi$ so that the modulo-2π operation is ineffective in (39). Then, arg $[R(m)]$ may be viewed as a noisy measurement of $2\pi M m f_d T$ and the problem of estimating f_d can be approached by smoothing out the statistics {arg $[R(m)]$} so as to improve the estimation process. This leads to eq. (37).

As with the L&R algorithm, the estimation range and the role of the parameter N are of interest. We have seen earlier that a critical condition to make arg $[R(m)]$ an unbiased estimate is that $2\pi M m f_d T$ be internal to the interval $\pm\pi$. As this must be true for any index m comprised between 1 and N, it follows that f_d must be limited within $\pm 1/(2MNT)$. In conclusion, Fitz estimator has an estimation range half of L&R's (for the same parameters M and N).

Fig. 9 illustrates the performance of the Fitz estimator with L_0 as a parameter. Everything is as in Fig. 8 except

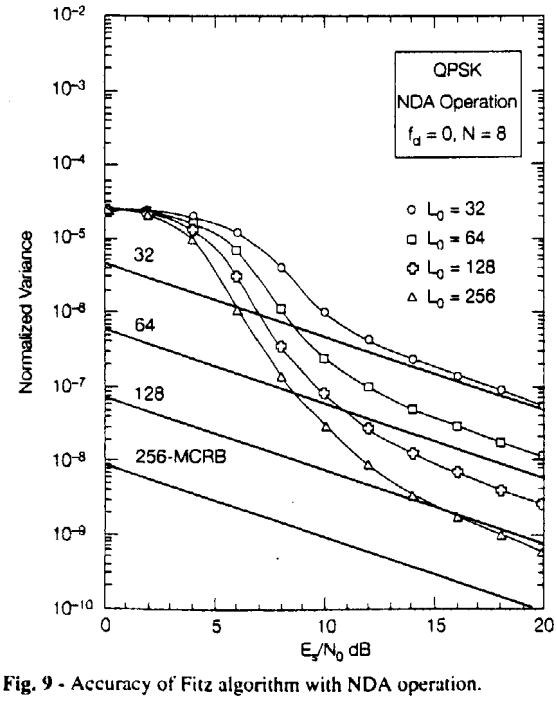

Fig. 9 - Accuracy of Fitz algorithm with NDA operation.

that N is chosen equal to 8 (to obtain the same estimation range of $\pm 0.015/T$ as in Fig. 8). We see that the estimation accuracy is slightly inferior to L&R's. As with L&R estimator, it is difficult to understand from the figure the exact position of the threshold. Searching for the SNR value corresponding to the first appearance of the outliers it turns out that the threshold decreases steadily as L_0 increases. For example, the threshold is 10 dB at $L_0 = 64$ and 7 dB at $L_0 = 256$.

5.3. *Lank, Reed and Pollon estimator*

A considerable simplification to Fitz estimator is obtained by keeping only the last term in the sum (37). As a result we get

$$\hat{f}_d = \frac{1}{2\pi MNT} \arg\left[R(N)\right] \tag{41}$$

This is the algorithm proposed by Lank, Reed and Pollon (LRP) in [11]. It turns out that its estimation range coincides with that of Fitz's (for the same N) but its performance is inferior.

5.4. *Mengali and Morelli estimator*

One problem with Fitz algorithm is that the modulo-2π operation in (39) cannot be ignored at intermediate and low SNR and, in fact, it leads to significant degradations in estimation accuracy. A possible solution would be to first unwrap the sequence {arg $[R(m)]$} and then insert the result into Fitz eq. (37), perhaps with different weighting coefficients. Mengali and Morelli (M&M) [12] have shown that the unwrapping process can be avoided by using the quantities {arg $[R(m)\,R^*\,(m-1)]$} in place of {arg $[R(m)]$}. Their reasoning essentially follows the steps outlined with Kay's algorithm. Skipping the details, they come up with the formula

$$\hat{f}_d = \frac{1}{2\pi MT} \sum_{m=1}^{N} w_m^{(M\&M)} \arg\left[R(m)\,R^*(m-1)\right] \tag{42}$$

where

$$w_m^{(M\&M)} = 3 \cdot \frac{(L_0 - m)(L_0 - m + 1) - N(L_0 - N)}{4N^2 - 6NL_0 + 3L_0^2 - 1} \tag{43}$$

and N is a design parameter. For $N = L_0/2$ the algorithm achieves the MCRB at high SNR [12].

The curves in Fig. 10 illustrate the estimation accuracy of the M&M algorithm for some values of L_0 and QPSK modulation. NDA operation is assumed. We see that the estimator achieves the MCRB for any L_0 at high SNR and the threshold decreases steadily as L_0 increases.

5.5. *Crozier and Moreland estimator*

The following alternative method has been proposed by Crozier and Moreland (C&M) in [13]. Write (39) in the form

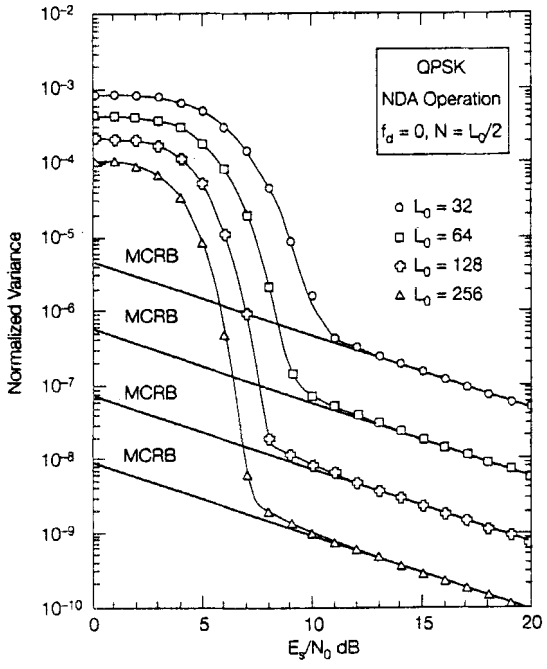

Fig. 10 - Accuracy of M&M algorithm with NDA operation.

$$\arg[R(m)] = 2\pi M m f_d T + \mu(m) + 2\pi k(m) \tag{44}$$

where $k(m)$ is an integer that serves to reduce the sum $2\pi M m f_d T + \mu(m) + 2\pi k(m)$ to the interval $\pm\pi$. Denoting $\omega \triangleq 2\pi f_d T$ and $\nu(m) \triangleq \mu(m)/(Mm)$, the above equation becomes

$$\frac{\arg[R(m)]}{Mm} - k(m)\frac{2\pi}{Mm} = \omega + \nu(m) \tag{45}$$

which indicates that the left hand side may be viewed as a noisy measurement of ω. In other words, the quantity $\arg[R(m)]/(Mm)$ is an *ambiguous* measurement of ω, with an ambiguity *quantum* of $2\pi/(Mm)$. The larger the index m, the smaller the quantum and the more potential phase ambiguities. If the ambiguity could be resolved, an estimate of ω (and eventually of f_d) could be derived from $\arg[R(m)]/(Mm)$.

Crozier and Moreland discuss the following procedure to overcome the ambiguity. Consider a sequence of estimates $[\hat{\omega}(m_i)]$ associated with the following particular values of the autocorrelation lag

$$m_i = 2^{i-1} \qquad i = 1, 2, \ldots, B \tag{46}$$

Eq. (45) suggests computing $\hat{\omega}(m_i)$ as follows

$$\hat{\omega}(m_i) = \frac{\arg[R(m_i)]}{Mm_i} - k(m_i)\frac{2\pi}{Mm_i} \tag{47}$$

where $k(m_i)$ is a suitable integer. How can we determine $k(m_i)$ and, ultimately, compute $\hat{\omega}(m_i)$? Suppose that a previous unambiguous estimate $\hat{\omega}(m_{i-1})$ is available. Then, a reasonable value for $k(m_i)$ is the one which makes the right hand side of (47) closest to $\hat{\omega}(m_{i-1})$. Bearing in mind that $\hat{\omega}(m_i)$ has an ambiguity quantum of $2\pi/(Mm_i)$ this leads to (for $i = 2, 3, \ldots, B$)

$$\hat{\omega}(m_i) = \hat{\omega}(m_{i-1}) + \left\{\frac{\arg[R(m_i)]}{Mm_i} - \hat{\omega}(m_{i-1})\right\}_{-\pi/(Mm_i)}^{\pi/(Mm_i)} \tag{48}$$

or, equivalently,

$$\hat{\omega}(m_i) = \hat{\omega}(m_{i-1}) + \frac{1}{Mm_i}\left\{\arg[R(m_i)] - Mm_i\,\hat{\omega}(m_{i-1})\right\}_{-\pi}^{\pi} \tag{49}$$

For $i = 1$ we have $m_1 = 1$. The corresponding estimate can be computed from (47) setting $k(1) = 0$, provided that ω is within $\pm\pi/M$ or, which is the same, that the frequency error f_d is within $\pm 1/(2MT)$. Assuming that this is true, one starts with the initial value

$$\hat{\omega}(1) = \frac{\arg[R(1)]}{M} \tag{50}$$

and computes the successive estimates by application of (49) up to the final estimate corresponding to $i = B$. In [13] it is shown that the minimum estimation variance is achieved for $B = 1 + \log_2(2L_0/3)$. The degradation incurred with slightly different values is limited, however. In fact in the simulations reported later we have set $B = \log_2(L_0)$ for convenience.

Fig. 11 gives an idea of the estimation ranges for some algorithms discussed so far. Note that the LRP estimator has essentially the same characteristics as Fitz's and is not shown. The ordinates yield the expectation of $\hat{f}_d T$ as obtained by simulation versus the true frequency error $f_d T$. The modulation is QPSK with RRCR(50%) pulses. DA operation is assumed and the *SNR* is 5 dB. The observation length L_0 equals 64 and the parameter N is chosen equal to 5. We see that Fitz algorithm gives (on average) accurate estimates over a range slightly smaller than ±10% of $1/T$. The L&R algorithm has an estimation range of about ±15% of $1/T$ and, finally, the M&M and C&M algorithms give correct results over ±45% of $1/T$.

Fig. 12 shows the estimation accuracy of the C&M method for various observation lengths and the same

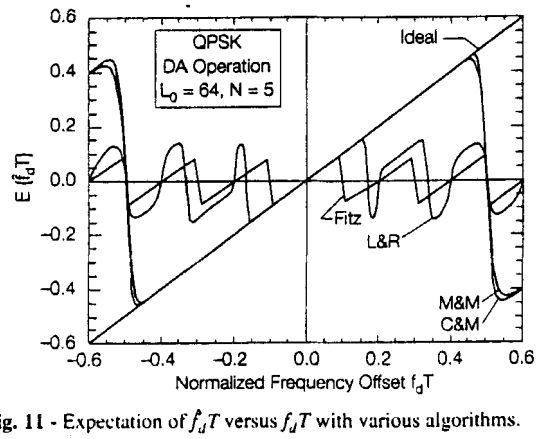

Fig. 11 - Expectation of $\hat{f}_d T$ versus $f_d T$ with various algorithms.

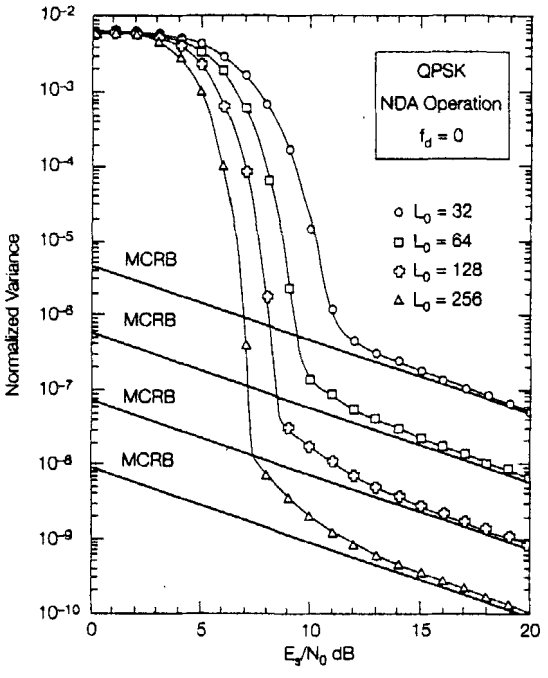

Fig. 12 - Accuracy of C&M algorithm with NDA operation.

modulation format as with the M&M algorithm in Fig. 10. Although the results with the latter are slightly superior, we shall soon see that the former requires a smaller computational load.

6. Computational complexity

One important issue that influences the choice of an estimation algorithm is its computational complexity. In this section we address this problem and, in particular, we assess the computational load associated with the algorithms described so far. This load is expressed in terms of number of operations (real multiplications and additions) and accesses to a Read-Only-Memory (ROM). For convenience we make a distinction between DA and NDA operations. In the first instance the raw data are represented by (5), in the second they are given by (7). Note that, depending on the estimator being used, it is useful to think of $z(k)$ as expressed in polar rather than rectangular coordinates. This is the case with estimators using arg $[z(k)]$ for it obviates the need of a ROM to map $z(k)$ into arg $[z(k)]$.

6.1. *R&B estimator*

6.1.1. DA operation

We limit ourselves to the consideration of the coarse search since the fine search involves only minor extra operations. The problem is to compute KL_0 samples of $|Z(f)|$ (see (13)) via an FFT. As is known [14], the FFT involves about $KL_0 \log_2 (KL_0)/2$ complex multiplications, $KL_0 \log_2 (KL_0)$ complex additions and KL_0 modulus extractions. This leads to the R&B line in Table 1 with $\rho = 1$. In writing this line we have taken into account the fact that a complex multiplication corresponds to four (real) multiplications and two (real) additions, each modulus extraction requires two multiplications and one addition and searching for the maximum in the set $[|Z(f_n)|]$ requires KL_0 comparisons. All these operations have been put together since in a DSP implementation they have approximately the same weight.

Table 1 - *Computational load with DA operation*

DA Operation		
Algorithm	Real Products & Additions	ROM Access
R&B	$KL_0 [4 + 7 \rho \log_2 (KL_0)]$	
Tretter	$4L_0 - 3$	
Kay	$3L_0 - 4$	
L&R	$4L_0 (N + 1) - 2$	1
L&R (FFT-based)	$3L_0 + 7N - 2 + 10 (L_0 + N) \log_2 (L_0 + N)$	1
Fitz	$4N (2L_0 - N - 1)$	N
Fitz (FFT-based)	$3L_0 + 5N - 1 + 10 (L_0 + N) \log_2 (L_0 - N)$	N
LRP	$8 (L_0 - N) - 1$	1
M&M	$N (8L_0 - 4N - 3) - 2$	N
M&M (FFT-based)	$3L_0 + 6N - 2 + 10 (L_0 + N) \log_2 (L_0 + N)$	N
C&M	$8 (L_0 B - 2^B) + 2B + 5$	B

Some saving in the FFT calculation is obtained by eliminating operations on zeros, which is commonly referred to as *pruning* [15]. It can be shown that pruning reduces the number of multiplications and additions by a factor

$$\rho = 1 - \frac{\log_2(K) + 2(1/K - 1)}{\log_2(KL_0)} \tag{51}$$

Correspondingly the R&B line becomes as indicated in Table 1 for a generic ρ. Note that for $K = 2$ we have $\rho = 1$ and the pruning does not allow any computational saving.

6.1.2. NDA OPERATION

Recalling that $|z(k)| = 1$, the generic term in (13) may be written as

$$z(k)\, e^{-j2\pi kfT} = e^{j\{\arg[z(k)] - 2\pi kfT\}} \tag{52}$$

So, starting from $\{\arg[z(k)];\ k = 0, 1, \ldots, L_0 - 1\}$, we need one addition and two ROM accesses to compute $z(k)\, e^{-j2\pi kfT}$ (for $\sin(x)$ and $\cos(x)$). Therefore the pruned FFT requires $5KL_0\, \rho \log_2(KL_0)/2$ additions and $KL_0\, \rho \log_2(KL_0)$ ROM accesses. Adding the operations for modulus extraction and maximization over the set $[|Z(f_n)|]$ produces the line R&B in Table 2.

6.2. *Tretter estimator*

Using the data $\arg[z(k)]$, there are $2(L_0 - 1)$ additions involved in (21). Also, $L_0 - 1$ additions and L_0 multiplications are required in (19). This yields the Tretter lines in Tables 1 and 2.

6.3. *Kay estimator*

Again, we use the data $\arg[z(k)]$. As is seen from (23), the calculation of the set $\arg[z(k)\, z^*(k-1)]$, $1 \le k \le L_0$, requires $L_0 - 1$ additions while the right hand side in (26) involves $L_0 - 1$ products and $L_0 - 2$ additions. This leads to the Kay line in Tables 1 and 2.

6.4. *L&R estimator*

6.4.1. DA OPERATION

Luise and Reggiannini [9] have proposed a clever method to compute $\hat{f}_d$ that significantly reduces the computational load. The interested reader is referred to their paper for details. The resulting complexity is indicated in the L&R line in Table 1. Alternatively, the computation of the set $[R(m)]$, $m = 1, 2, \ldots N$, can be carried out in the frequency domain through the following steps [14, p. 556]:

i) form an $(L_0 + N)$-point sequence by adding N zeroes to $[z(k)]$
ii) compute the $(L_0 + N)$-point Discrete Fourier Transform (DFT)

$$Z(n) = \sum_{k=0}^{L_0+N-1} z(k) e^{-j2\pi kn/(L_0+N)} \qquad 0 \le n \le L_0 + N - 1 \tag{53}$$

Table 2 - *Computational load with NDA operation*

NDA Operation		
Algorithm	Real Products & Additions	ROM Access
R&B	$KL_0[4 + 2.5\, \rho \log_2(KL_0)]$	$KL_0\, \rho \log_2(KL_0)$
Tretter	$4L_0 - 3$	
Kay	$3L_0 - 4$	
L&R	$0.5N(6L_0 - 3N + 1) - 2$	$N(2L_0 - N - 1) + 1$
L&R (FFT-based)	$3L_0 + 7N - 2 + 7.5(L_0 + N)\log_2(L_0 + N)$	$1 + (L_0 + N)\log_2(L_0 + N)$
Fitz	$0.5N(6L_0 - 3N - 3) - 1$	$N(2L_0 - N)$
Fitz (FFT-based)	$3L_0 + 5N - 1 + 7.5(L_0 + N)\log_2(L_0 + N)$	$N + (L_0 + N)\log_2(L_0 + N)$
LRP	$3(L_0 - N) - 1$	$2(L_0 - N) + 1$
M&M	$0.5N(6L_0 - 3N - 1) - 2$	$N(2L_0 - N)$
M&M (FFT-based)	$3L_0 + 6N - 2 + 7.5(L_0 + N)\log_2(L_0 + N)$	$N + (L_0 + N)\log_2(L_0 + N)$
C&M	$3(L_0B - 2^B) + 2B$	$2(L_0B - 2^B) + B + 2$

iii) compute the $(L_0 + N)$-point inverse DFT of $|Z(n)|^2$

$$y(m) = \frac{1}{L_0 + N} \cdot \sum_{n=0}^{L_0+N-1} |Z(n)|^2 e^{j2\pi nm/(L_0+N)} \qquad 0 \le m \le L_0 + N - 1 \tag{54}$$

iv) finally, compute $R(m)$ as

$$R(m) = \frac{1}{L_0 - m} y(m) \qquad 1 \le m \le N \tag{55}$$

The FFT is used to efficiently perform operations ii) and iii). The resulting complexity is indicated in the L&R (FFT-based) line in Table 1.

6.4.2. NDA OPERATION

It turns out that the procedure described in [9] is not useful with NDA operation. Instead, the following approach can be used. From the formula

$$z(k)\, z^*(k-m) = e^{j[\arg[z(k)] - \arg[z(k-m)]]} \tag{56}$$

it is clear that the computation of $z(k)\, z^*(k-m)$ requires one addition and two ROM accesses (for $\sin(x)$ and $\cos(x)$). Thus, the calculation of the set $[R(m)]$, $m = 1, 2, \ldots N$, in the time domain involves $2N$ multiplications, $N(6L_0 - 3N - 7)/2$ additions and $N(2L_0 - N - 1)$ ROM accesses. Viceversa, in the frequency domain (FFT-based method), we need $2(L_0 + 2N) + 2(L_0 + N)\log_2(L_0 + N)$ products, $L_0 + N + 5.5(L_0 + N)\log_2(L_0 + N)$ additions and $(L_0 + N)\log_2(L_0 + N)$ ROM accesses. The final computation of $\hat{f}_d$ from (33) requires $2(N - 1)$ further additions and one ROM access.

6.5. *Fitz estimator*

6.5.1. DA OPERATION

Bearing in mind the definition (32) we can rewrite the Fitz estimator in the form

$$\hat{f}_d = \frac{1}{2\pi MT} \sum_{m=1}^{N} w_m^{(F)} \theta(m) \tag{57}$$

with

$$\theta(m) \triangleq \arg\left[\sum_{k=m}^{L_0-1} z(k) z^*(k-m)\right] \tag{58}$$

Then, the computation of the set $[\theta(m)]$ in the time domain requires $2N(2L_0 - N - 1)$ multiplications, $2N(2L_0 - N - 2)$ additions and N ROM accesses. In the frequency domain, viceversa, it needs $2(L_0 + N)[1 + 2\log_2(L_0 + N)]$ products, $(L_0 + N)[1 + 6\log_2(L_0 + N)]$ additions and N ROM accesses. In both cases the final calculation of $\hat{f}_d$ involves N further multiplications and $N - 1$ additions.

6.5.2. NDA OPERATION

Writing $z(k)\, z^*(k-m)$ as in (56) allows us to compute the set $[\theta(m)]$ through $N(6L_0 - 3N - 7)/2$ additions and $N(2L_0 - N)$ ROM accesses in the time domain. Alternatively, in the frequency domain we need $2(L_0 + N)[1 + \log_2(L_0 + N)]$ products, $(L_0 + N)[1 + 5.5\log_2(L_0 + N)]$ additions and $N + (L_0 + N)\log_2(L_0 + N)$ ROM accesses. The steps to compute $\hat{f}_d$ remain the same as with DA operation.

6.6. *LRP estimator*

The LRP estimator can be treated as a special case of Fitz method. Here, however, the FFT-based method is not useful since only one autocorrelation is needed.

6.7. *M&M estimator*

From (32) and (58) we have

$$\arg[R(m)\, R^*(m-1)] = [\theta(m) - \theta(m-1)]_{-\pi}^{\pi} \tag{59}$$

So, once the sequence $[\theta(m)]$ is computed (see Fitz estimator), the right hand side of (42) requires N multiplications and $2(N - 1)$ additions.

6.8. *C&M estimator*

6.8.1. DA OPERATION

The computation of the set $\{\arg[R(m_i)]\}$ ($m_i = 2^{i-1}$ and $i = 1, 2, \ldots, B$) requires $L_0 B + 1 + 2^B$ complex products, $(L_0 - 1)B + 1 - 2^B$ complex additions and B ROM accesses. Also, the recursions (49) involve $2(B - 1)$ multiplications and additions.

6.8.2. NDA OPERATION

The computation of the set $\{\arg[R(m_i)]\}$ requires $3(L_0 B + 1 - 2^B) - 2B$ additions and $2(L_0 B + 1 - 2^B) + B$ ROM accesses while the steps involved in (49) remain the same. The FFT-based method has not been considered since only few autocorrelations are needed.

6.9. COMPARISONS

Fig. 13 shows complexity comparisons between some of the above estimation schemes with DA operation. Each curve gives the total number of multiplications and additions as a function of the observation length L_0. A pruning factor of $K = 4$ has been chosen with the R&B algorithm and three different values for N have been used with the M&M method. Finally, the parame-

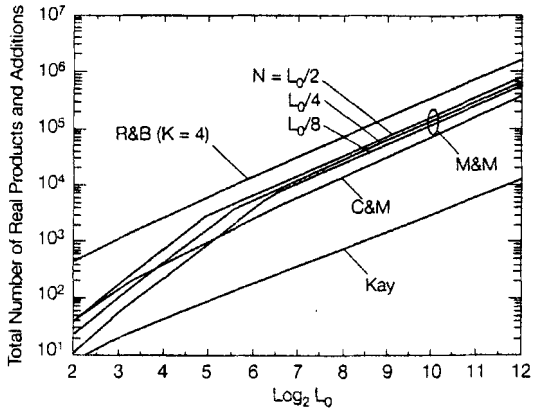

Fig. 13 - Complexity comparisons.

ter B in the C&M estimator has been set equal to $\log_2 (L_0)$. The complexity of the M&M estimator depends on whether the set $R(m)$, $m = 1, 2, ..., N$, is computed in the time or frequency domain. For each L_0 value in Fig. 13 we have chosen the method requiring the minimum complexity. The sudden variation in the curve slopes indicates the passage from time to the frequency domain calculations. Curves for the L&R estimator are not indicated as they are very close to those for the M&M algorithm. We see that the Kay algorithm is by far the simplest. The M&M estimator is always simpler than R&B, but the difference is not overwhelming. Finally, the C&M method is simpler than M&M except with very short observation lengths.

It should be pointed out that the above conclusions apply to high *SNR* operating conditions. In these circumstances, in fact, the same estimation accuracy is essentially obtained for a given L_0 with any of the above estimators. At intermediate or low *SNR*, however, the situation may be quite different. To be specific suppose we want a threshold of 3 dB, which might not be that outlandish with coded modulation. It turns out that the R&B method attains this threshold with $L_0 = 1024$ while the M&M and C&M methods need $L_0 = 8196$ and $L_0 = 16384$ respectively. Assuming $B = \log_2 (L_0)$ and $N = L_0/2$, we see from Table 2 that this corresponds to a total of about 181 kilo-operations (including multiplications, additions and ROM accesses) with R&B, 1472 kilo-operations with M&M and 1065 kilo-operations with C&M.

7. Conclusions

We have illustrated some feedforward methods for estimating the carrier frequency in data transmission. A distinction has been made between DA and NDA operation. In the former a preamble of known symbols is exploited for estimation purposes whereas, in the latter, the carrier frequency is directly derived from the modulated signal making use of some non linearity. Generally speaking, the effect of the non linearity is to increase the estimation threshold while leaving the estimation accuracy unaffected at intermediate/large *SNR* values.

When the estimation range is limited (to the order of few percents of the symbol rate) either the LRP or Fitz or L&R algorithms can be used. Greater estimation ranges call for other techniques such as R&B, M&M and C&M. All these methods exhibit an estimation accuracy that is close to the modified CRB when operating well above threshold. The threshold value is strongly related to L_0 and decreases as L_0 increases.

Comparisons have been made between estimation algorithms in terms of estimation accuracy, estimation range, threshold and complexity. Complexity seems a tricky subject in so far as it gives different answers depending on the operating conditions. At *SNR* well over the threshold the C&M method is the simplest. On the contrary, at low *SNR* the R&B method seems preferable.

Acknowledgements

The authors would like to thank Professor Marco Luise for his helpful comments and suggestions during the preparation of the final version of the manuscript.

Manuscript received on November 3, 1997.

References

[1] R. A. Harris: *Transmission considerations for LAN interconnection by satellite*. COST 226 Workshop on "Integrated Space/Terrestrial Networks", Graz, Austria, 25-26 February 1993.

[2] F. Ananasso, G. Pennoni: *Clock synchronous multicarrier demodulator for multifrequency TDMA communications satellites*. "Proc. ICC'90", Atlanta, GA, p. 1059-1063.

[3] A. J. Viterbi, A. M. Viterbi: *Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission*. "IEEE Trans. Inform. Theory", Vol. IT-29, July 1983, p. 543-550.

[4] D. C. Rife, R. R. Boorstyn: *Single-tone parameter estimation from discrete-time observations*. "IEEE Trans. Inform. Theory", Vol. IT-20, Sept. 1974, p. 591-598.

[5] A. N. D'Andrea, U. Mengali, R. Reggiannini: *The modified Cramer-Rao bound and Its applications to synchronization problems*. "IEEE Trans. Commun.", Vol. COM-42, Feb./March/April 1994, p. 1391-1399.

[6] S. A. Tretter: *Estimating the frequency of a noisy sinusoid by linear regression*. "IEEE Trans. Inf. Theory", Vol. IT-31, Nov. 1985, p. 832-835.

[7] S. M. Kay: *A fast and accurate single frequency estimator*. "IEEE Trans. Acoustic, Speech and Signal Processing", Vol. ASSP-37, Dec. 1989, p. 1987-1990.

[8] S. Bellini, C. Molinari, G. Tartara: *Digital frequency estimation in burst mode QPSK transmission*. "IEEE Trans. Commun.", Vol. COM-38, July 1990, p. 959-961.

[9] M. Luise, R. Reggiannini: *Carrier frequency recovery in all-digital modems for burst-mode transmissions*. "IEEE Trans. Commun.", Vol. COM-43, March 1995, p. 1169-1178.

[10] M. P. Fitz: *Further results in the fast estimation of a single frequency*. "IEEE Trans. Commun.", Vol. COM-42, March 1994, p. 862-864.

[11] G. W. Lank, I. S. Reed, G. E. Pollon: *A semicoherent detection and Doppler estimation statistic*. "IEEE Trans. Aerosp. Electron. Syst.", Vol. AES-9, March 1973, p. 151-165.

[12] U. Mengali, M. Morelli: *Data-aided frequency estimation for burst digital transmission*. "IEEE Trans. Commun.", Vol. COM-45, Jan. 1997, p. 23-25.

[13] S. Crozier, K. Moreland: *Performance of a simple delay-multiply-average technique for frequency estimation*. Canadian Conf. on Electrical and Computer Eng., Toronto, Canada, paper WM-10.3, Sept. 13-16, 1992.

[14] A. V. Oppenheim, R. W. Schafer: *Digital signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[15] J. D. Markel: *FFT pruning*. "IEEE Trans. Audio Electroacoustic", Vol. AU-19, Dec. 1971, p. 305-311.