A LOW COMPLEXITY FLEXIBLE NON–DATA–AIDED AND NON–TIME–DIRECTED FREQUENCY SYNCHRONIZER FOR QPSK MODULATED SIGNALS

Pilar M. Calvo, Juan F. Sevillano, Igone Velez, and Andoni Irizar, Member IEEE

Department of Electronics and Communications. CEIT and TECNUN (University of Navarra)
Paseo Mikeletegi, 48. 20009 San Sebastián, Spain. Email: {pcalvo, jfsevillano, ivelez, airizar}@ceit.es

ABSTRACT

This paper focuses on the estimation of large carrier frequency offsets in QPSK modulated signals. A common approach is based on searching for the maximum of a likelihood function. This search must be done operating over the same block of samples, in order to avoid a noisy likelihood curve that can lead to an erroneous estimate. We propose an efficient iterative algorithm that works with different blocks of data. This way, the need for memory is removed, and implementation costs are reduced. Additionally, the new algorithm can be adapted to the SNR in run-time.

1. INTRODUCTION

In continuous mode transmission systems, like satellite broadcasting, QPSK modulation is widely used. In a digital receiver, samples of the continuous–time signal are used to produce an estimation of the transmitted symbols. Most of the receiver structures are based on the principle of synchronized detection. That is, estimates of the synchronization parameters are used in the symbol detection [1]. In order to maximize the channel efficiency, the receiver should operate in the absence of any training sequences or pilots. That is, synchronization has to be blind, i.e., non–data–aided (NDA).

Robust NDA timing synchronization algorithms require the carrier frequency offset to be at most $10-20\%$ of the symbol rate [2]. When the frequency offset is large, a non–time–directed (NTD) frequency estimator [1,3,4] is needed prior to timing acquisition so as to reduce the frequency offset within the acquisition range of the timing synchronizer. Once timing has been acquired, powerful time–directed algorithms can be used to improve the accuracy of the frequency estimate [1].

In this paper we focus on the estimation of large carrier frequency offsets, that is, on the NTD stage of the formerly described synchronization approach. During this stage, feed–forward synchronization algorithms are preferred to avoid the need of a NTD frequency lock detection strategy.

Nowadays, it is feasible to use a single analog to digital converter in the receiver for different channels or signals. In this scenario, the matched filter (MF) can be used to separate the signal of interest. Hence, it is desirable to operate on the output of this filter. The feed–forward NTD frequency estimator presented in [3] works on the received signal before the MF, and thus adjacent signals can affect its performance. Another proposal is based on maximizing the likelihood function at the output of the MF [1]. An efficient implementation of [1] is reported in [4], but it requires memory to store the received samples during the search for the maximum.

This paper proposes a reduced complexity blind algorithm to estimate large carrier frequency offsets. As in [1,4], it uses the output of the MF to compute its statistics, which minimizes the effect of adjacent signals. The new algorithm eliminates the use of data memory, so the implementation cost is reduced. Furthermore, this algorithm can easily be adapted to the SNR operating point in run–time.

2. DESCRIPTION OF THE PROBLEM

We consider a baseband QPSK communication system. The received signal at the input of the matched filter is [1]:

$$r(kT_s)=e^{j(2\pi\nu TkT_s+\theta)}\sum_{n=0}^{NM_s-1}a_n g(kT_s-nT-\varepsilon T)+w(kT_s), \quad (1)$$

where T is the symbol period, T_s is the sampling frequency, $M_s = T/T_s$ is an integer number and N is the number of symbols. $\{a_n\}$ is a QPSK symbol sequence, $g(t)$ is the impulse response of the transmit filter and $\{w_k\}$ is complex white Gaussian noise with zero mean and spectral density N_o. The unknown parameters are the carrier frequency ν (which is normalized by the symbol frequency), the carrier phase θ, and the symbol timing ε. These parameters are independent random variables uniformly distributed in the ranges $([-1,1],[0,2\pi],[0,T])$ respectively. We assume that ν, θ and ε remain constant during the observation interval.

For optimum performance, the received signal should be frequency corrected and passed through a filter matched to the transmitter. The signal at the output of the matched filter is:

$$z(kT_s)=\sum_{n=0}^{NM_s-1}r(nT_s)e^{-j2\pi\tilde{\nu}TnT_s}g_{MF}(kT_s-nT_s)\,, \quad (2)$$

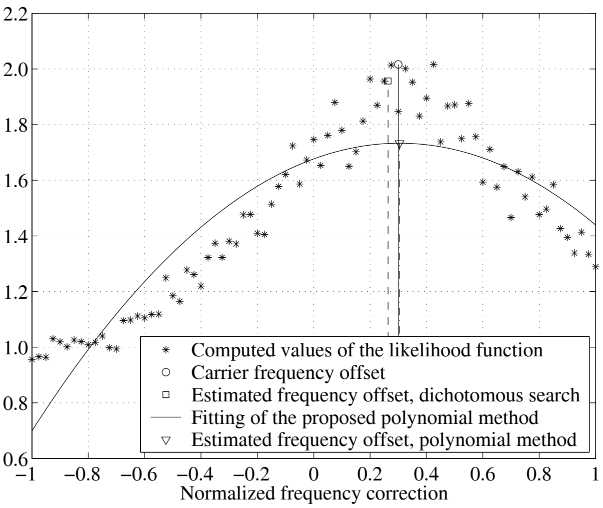

Fig. 1. Values of the likelihood function computed using different data blocks for each frequency trial. $E_s/N_o = 0\,\text{dB}$, $\nu = 0.3$, $N = 512$ symbols, $L = 81$, $M_s = 4$.

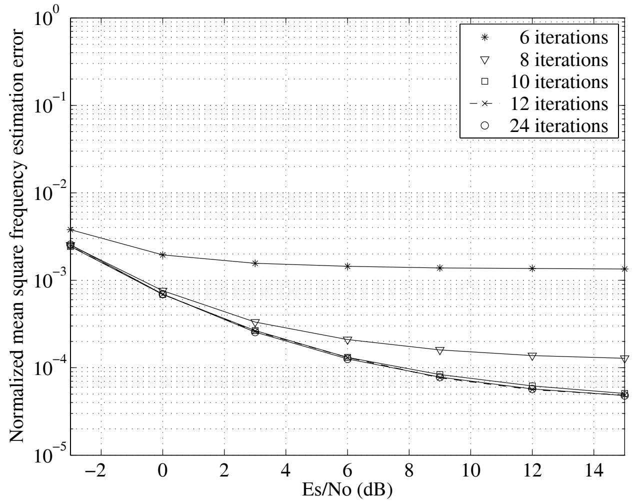

Fig. 2. Dichotomous search using different number of iterations. $N = 512$ symbols, $M_s = 4$.

where $\tilde{\nu}$ is the carrier frequency correction applied at the input of the matched filter (g_{MF}).

In [1], an NDA and NTD unbiased estimator of the carrier frequency offset is obtained by maximizing the likelihood function

$$\Lambda(z|\nu) = \sum_{n=0}^{NM_s-1} |z(nT_s)|^2 \,. \tag{3}$$

The frequency estimate is given by:

$$\hat{\nu} = \arg\max_{\nu} \Lambda(z|\nu)\,. \tag{4}$$

The classical approach for estimating (4) consists in defining a set of trial values $\{\tilde{\nu}_l\}$, $l = 1, 2, \ldots, L$, applying them to the signal at the input of the matched filter (2), and computing the values $\{\Lambda(z|\nu{=}\tilde{\nu}_l)\}{=}\{\Lambda_l\}$ (3). The frequency estimate is the value $\tilde{\nu}_m$ for which the computed Λ_m is maximum. The search for the maximum can be performed efficiently by using a dichotomous search [4]. It is important to note that during this search, all the values of $\{\Lambda_l\}$ must be computed using the same block of data $\{r_k\}$. If a different block of data is used to compute each $\{\Lambda_l\}$, then the likelihood function becomes noisy ($*$ in Fig. 1), and the search for the maximum can lead to an erroneous estimate ($\square$ in Fig. 1). Thus, in order to avoid this problem, the search for the maximum requires a memory with at least NM_s size to store the received signal $\{r_k\}$.

Fig. 2 represents the normalized mean square frequency estimation error (MSFE) for the dichotomous search. All the mean values shown in the figures throughout this paper were obtained using Monte Carlo simulations, with 10^4 runs per point. Simulations in Fig. 2 show that the MSFE has a lower bound that cannot be reduced by increasing the number of iterations in the search. This lower bound is due to the length of the data observation interval, N. Hence, in order to improve the performance, the length of the data block N, and thus, the size of the required memory must be enlarged, which increases the implementation cost. As the designer has to select the size of the memory to cope with the worst case scenario, the design is suboptimum for better operating conditions. Moreover, the selected memory fixes the minimum SNR operating point.

In the next section, we propose a new NDA and NTD frequency estimator that avoids the required memory to store the received data. Therefore, the proposed algorithm reduces the implementation cost and provides flexibility to the receiver.

3. DERIVATION OF THE PROPOSED FREQUENCY ESTIMATOR

In order to avoid the required memory in the methods described in [1] and [4], we propose to evaluate the likelihood function using different blocks of data for each trial of $\{\tilde{\nu}_l\}$. We then fit the computed values of the likelihood function to a second–order polynomial by using a least squares approximation (continuous line in Fig. 1). The frequency estimate is the maximum of this regression curve (∇ in Fig. 1), which can be computed analytically.

The generalized form of the second–order polynomial is

$$P(\nu) = a_0 + a_1 \cdot \nu + a_2 \cdot \nu^2, \tag{5}$$

where a_0, a_1 and a_2 are constants and $a_2{\neq}0$. As the likelihood function is convex (see Fig. 1), a_2 has to be negative.

The frequency estimate is the maximum of this polynomial, which is given by

$$\nu_{max} = \frac{-a_1}{2a_2}. \tag{6}$$

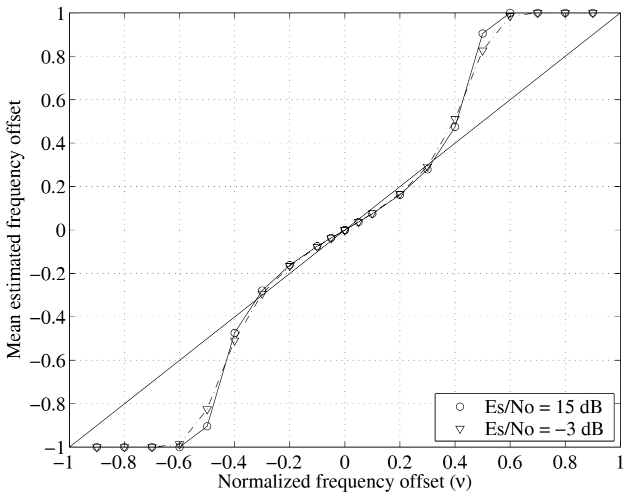

Fig. 3. Approximation to a second–order polynomial. Normalized frequency offset vs. mean estimated frequency offset. $N = 512$ symbols, $L = 21$, $M_s = 4$.

We simplify the computation of the coefficients a_1 and a_2 by setting L odd and selecting $\{\tilde{\nu}_l\}$ equally spaced around $\tilde{\nu}$=0, that is, $\tilde{\nu}_l$=0 for l=$(L + 1)/2$, and $\tilde{\nu}_l$=$-\tilde{\nu}_{L-(l-1)}$ for $l = 1, 2, \ldots, (L-1)/2$. Then, the solution of the least squares problem yields the following coefficients:

$$a_1 = \frac{\sum_{l=1}^{L} \Lambda(z|\nu{=}\tilde{\nu}_l)\, \tilde{\nu}_l}{\sum_{l=1}^{L} \tilde{\nu}_l^2} \tag{7}$$

$$a_2 = \frac{\sum_{l=1}^{L} \Lambda(z|\nu{=}\tilde{\nu}_l) \sum_{l=1}^{L} \tilde{\nu}_l^2 - L \sum_{l=1}^{L} \Lambda(z|\nu{=}\tilde{\nu}_l)\, \tilde{\nu}_l^2}{\left(\sum_{l=1}^{L} \tilde{\nu}_l^2\right)^2 - L \sum_{l=1}^{L} \tilde{\nu}_l^4} \tag{8}$$

In (7) and (8) divisions are not required, because terms depending only on $\{\tilde{\nu}_l\}$ and not on $\{\Lambda_l\}$ do not need to be computed in run–time. These terms can be precomputed and stored using very little ROM. The only division required is the one used to compute (6). However, both the division in (6) and the multiply-accumulate (MAC) operations in (7) and (8) do not have to work at the sampling rate. Hence, these operations can be performed with a very low implementation cost.

For small frequency offsets, the maximum of the polynomial approximation, ν_{max}, matches the estimation provided by the search for the maximum in the likelihood function. Nevertheless, as the frequency offset increases ($0.4{<}|\nu|{<}0.6$), the polynomial cannot always fit to the computed values of the likelihood function, and ν_{max} can be located outside the bracketing interval $[-1, 1]$. In order to solve this problem, when $|\nu_{max}|{>}1$ we select as the frequency estimate the nearest limit of the bracketing interval from ν_{max}. Moreover,

Table 1. Summary of the iterative polynomial method.

$\hat{\nu}_{i=0} = 0$
For i=1:n_iter
- *Apply a frequency correction, $\hat{\nu}_{i-1}$, to the signal at the input of the matched filter.*
- *Compute $\Lambda(z|\nu{=}\tilde{\nu}_l)$ for L trial values $\{\tilde{\nu}_l\}$ (3).*
- *Compute a_1 (7), a_2 (8), and ν_{max} (6).*

if $a_2 > 0$ *and* $\nu_{max} < 0$ *(concavity is reversed)*
$\Delta\hat{\nu} = 1$
elseif $a_2 > 0$ *and* $\nu_{max} > 0$ *(concavity is reversed)*
$\Delta\hat{\nu} = -1$
elseif $\nu_{max}{<}-1$ *(ν_{max} out of the bracketing interval)*
$\Delta\hat{\nu} = -1$
elseif $\nu_{max} > 1$ *(ν_{max} out of the bracketing interval)*
$\Delta\hat{\nu} = 1$
else
$\Delta\hat{\nu} = \nu_{max}$
end
$\hat{\nu}_i{=}\hat{\nu}_{i-1}{+}\Delta\hat{\nu}$
end

when the frequency offset approaches the limit of the bracketing interval ($|\nu|{\geq}0.6$), the regression curve becomes concave ($a_2{>}0$). When this concavity reversal occurs, we select as the frequency estimate the furthest limit of the bracketing interval from ν_{max}. These two special cases, maximum outside the bracketing interval and concavity reversal, can easily be detected by checking the overflow of ν_{max} and the sign of a_2 and ν_{max} respectively. Both checks are very simple to implement.

Fig. 3 represents the normalized frequency offset versus the mean of the estimated frequency offset for the polynomial approximation using the stated rules. Fig. 3 shows that for large frequency offsets, the polynomial approximation yields a biased estimate. In order to overcome this bias and be able to acquire ν in the whole range $|\nu|{<}1$, we propose to make the estimator iterative. Table 1 summarizes the proposed iterative polynomial method. Initially, we set $\hat{\nu}_{i=0}$=0. In each iteration the polynomial approximation together with the stated rules produce a frequency estimate, $\Delta\hat{\nu}$. Then, $\hat{\nu}_i{=}\hat{\nu}_{i-1}{+}\Delta\hat{\nu}$ is used as a frequency correction factor at the input of the matched filter before the next estimation. Iterations can be stopped by a tolerance criterion around $\Delta\hat{\nu}$=0. However, setting a fixed number of iterations simplifies the control.

4. SIMULATION RESULTS

The MSFE of the proposed iterative frequency estimator has been evaluated in order to assess its performance. The MSFE of the dichotomous search for 12 iterations is used as a reference.

The proposed iterative algorithm has three parameters: the number of points used for each least squares approximation, L, the number of iterations that are carried out, n_iter,

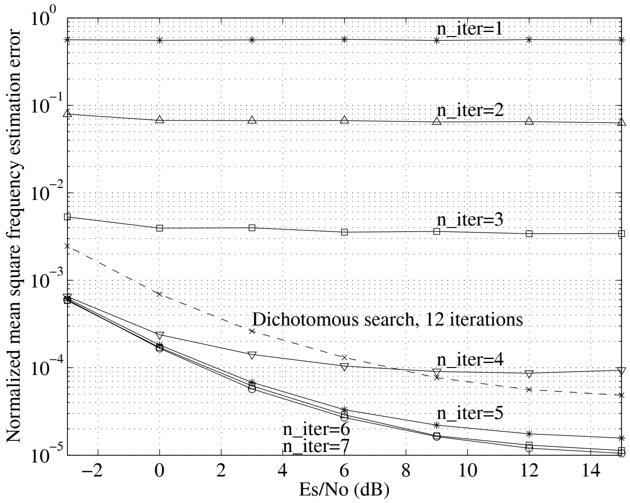

Fig. 4. Iterative polynomial method. Several number of iterations, n_iter. $L = 21$, $N = 512$ symbols, $M_s = 4$. The dashed line is the MSFE of the dichotomous search.

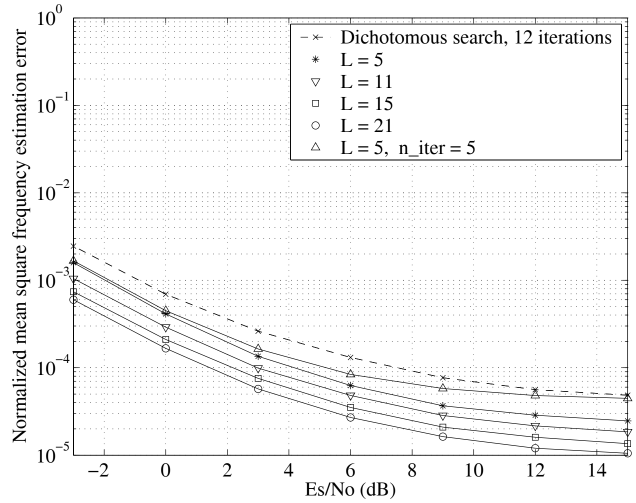

Fig. 5. Iterative polynomial method. Several number of frequency trials, L. $n\_iter = 6$, $N = 512$ symbols, $M_s = 4$. The dashed line is the MSFE of the dichotomous search.

and the length of the data observation interval, N.

Fig. 4 shows the MSFE for several numbers of iterations. It can be seen that as n_iter increases, the MSFE decreases. For N=512 and L=21, six iterations are enough to achieve the minimum MSFE (○ in Fig. 4). Fig. 5 shows the MSFE when different number of points are used to perform the polynomial fitting in each iteration. It is observed that the MSFE can be reduced by increasing L. Increasing the length of the data block N also reduces the MSFE, thought due to lack of space this result is not reported in this paper.

The time required by the proposed algorithm to perform frequency estimation is proportional to $n\_iter{\cdot}L{\cdot}N$. Therefore, Figs. 4 and 5 can be used to make a trade–off between the value of the MSFE and the estimation time. For example, when N=512, 5 iterations (n_iter=5) and 5 frequency trials per iteration (L=5) are enough for the iterative estimator to outperform the MSFE of the dichotomous search in the SNR operating range of interest for most applications (△ in Fig. 5). Even though a longer processing time is needed, this performance improvement is achieved with less complexity.

Furthermore, in the proposed algorithm, the values of N and n_iter can easily be changed during run–time in order to adapt them to the SNR operating point and to the required MSFE. This makes the receiver flexible since it can choose its parameters to adapt to the channel requirements, even to new channel conditions. This is achieved with no extra implementation costs because it only has to change the estimation time.

5. CONCLUSIONS

We have considered the problem of estimating large carrier frequency offsets without timing information. The classical approach based on the maximum–likelihood principle needs memory in order to store the received signal. This introduces a penalty in the implementation cost. Additionally, the size of this memory fixes the lowest SNR operating point of the algorithm.

This paper proposes a new algorithm that does not require such data memory, which reduces the implementation complexity. Moreover, the proposed synchronization algorithm can easily be adapted in run–time to the SNR of the received signal.

6. REFERENCES

[1] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital communication receivers: synchronization, channel estimation and signal processing*, John Wiley & Sons, 1998.

[2] A.A. D'Amico, A.N. D'Andrea, and R. Reggiannini, "Efficient non–data–aided carrier and clock recovery for satellite DVB at very low signal–to–noise ratios," *IEEE J. Select. Areas Commun.*, vol. 19, no. 12, pp. 2320–2330, Dec. 2001.

[3] F. Classen and H. Meyr, "Two frequency estimation schemes operating independently of timing information," *IEEE Global Telecommun. Conf., GLOBECOM'93*, vol. 3, pp. 1996–2000, Nov. 1993.

[4] E. Arias de Reyna and J.I. Acha-Catalina, "Non–data–aided and non–timing–directed feedforward frequency acquisition for linear digital modulations," *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp. 438–441, May 2005.