Frequency and Phase Estimation for High Frequency Packet Radio Modem

Milan Oklobdžija, Nikola Nenadić, *Member, IEEE,* Miloš Kaljević and Goran Marković

***Abstract* — High frequency channel is a difficult medium for data transmission. One of the methods to avoid effects of signal degradation is to use multi-carrier transmission. In this paper, we present several algorithms for frequency and phase estimation used for coherent and non-coherent demodulation of multi-carrier signals. These algorithms perform coarse and fine frequency and phase estimation by using data-aided and non-data-aided methods.**

***Key words* — high frequency, modem, frequency offset, phase offset, estimation.**

I. Introduction

Propagation of the signals in the high frequency (HF) channel is characterized by significant signal degradation. Multipath propagation and various ionosphere effects cause time dispersion, Doppler shift and frequency selective fading. In order to increase reliability of transmission and simplify channel estimation multi-carrier signals with differential phase modulation are used. This approach is used in many commercial HF modems such as SCS Pactor IIex [1] and Codan 3012 [2].

Frequency and phase offset estimation is necessary step in coherent and non-coherent demodulation. This offset is the consequence of transmission channel effects and to a lesser extent impairment of transmitter and receiver oscillators. In this paper we present algorithms for frequency and phase estimation implemented in the software receiver of the packet HF modem.

HF radio signal of the packet modem is first demodulated in radio receiver and than fed in software modem receiver as an audio signal in 3 kHz bandwidth.

In this particular case, signals on all of the carriers start with the known preamble (header). Thus, application of data-aided methods for synchronization is provided.

Demodulation of the modem signal starts when signal presence is detected in the band of interest. Pass-band signal is translated into the base-band and then decimated and filtered with matched filter (Figure 1.). Coarse frequency estimation is then performed and demodulator frequency is adjusted accordingly. After that symbol timing is estimated and sampled symbols are fed into fine frequency and phase offset estimator. Finally, residual frequency and phase offset is estimated and corrected in phase estimator block.

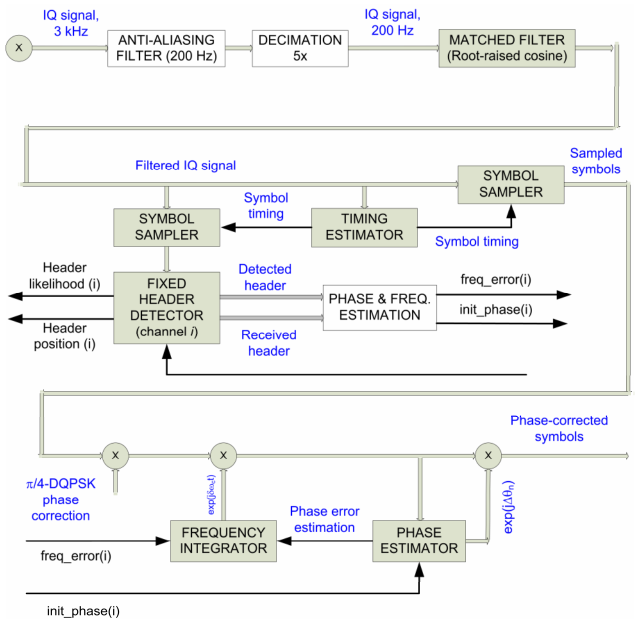

Figure 1. Schematics of demodulation process of a single carrier in multi-carrier signal

In the second section we will present algorithms for coarse frequency estimation, fine frequency and phase estimation, joint frequency estimation and residual phase offset estimation.

Third section describes practical implementation of algorithms and methods of testing, while the fourth section contains performances of algorithms.

II. Algorithm Description

A. *Coarse frequency estimation*

Coarse estimation of carrier frequencies is based on linear search for the maximum likelihood of headers [3]. The search spans over discrete frequencies in the range from -30Hz to +30Hz from the supposed carrier frequency, with 5Hz increment.

For each discrete frequency, a window of the signal on a carrier, which includes packet header, is down-converted and filtered. Then, symbol timing is estimated. The base-

This work was supported by the Ministry of Science of the Republic of Serbia, Grant TR-11040.

M. Oklobdžija, Institut Mihajlo Pupin, Volgina 15, 11060 Beograd, Serbia; (e-mail: milan.oklobdzija@institutepupin.com).

N. Nenadić, Institut Mihajlo Pupin, Volgina 15, 11060 Beograd, Serbia; (e-mail: nikola.nenadic@institutepupin.com).

M. Kaljević, Institute „Mihailo Pupin“, Volgina 15, 11060, Belgrade, Serbija; (e-mail: milos.kaljevic@institutepupin.com)

G. Marković, Elektrotehnički fakultet u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Serbia; (e-mail: gmarkovic@etf.bg.ac.rs).

band sampled symbols of the window are fed into the header detection block, which returns the likelihood of a header within the window. The closer is the discrete frequency to the true carrier frequency, the better symbol timing is obtained and the higher likelihood of a header if the header exists. Hence, the coarse frequency estimation yields the discrete frequency which is closest to the true carrier frequency with high probability.

B. *Fine frequency correction and initial phase estimation*

Phase & Frequency Estimation block employs timing-aided, data-aided approach to determine initial phase and remaining frequency offset in the signal. It takes received header (reference pulse plus N-1 symbols) and compares it with detected header (header most likely to have been transmitted). Vector of phase differences between the two headers, assuming that detected header is equal to the transmitted one can be expressed as:

$$\begin{aligned} &\vec{d} = \vec{r} - \vec{h}, \\ &\{d_k\} = \{r_k\} - \{h_k\} \\ &= \{\phi_k + 2\pi\Delta f T k + \theta_0 + n_k\} - \{\phi_k\} \\ &= \{2\pi\Delta f T k + \theta_0 + n_k\}. \end{aligned}$$

In previous equation d_k is phase difference between received and detected header at k-th symbol, φ_k is phase of k-th symbol in header, Δf is frequency offset, θ_0 is initial phase of received header, T is symbol period, and n_k is a noise sample. The phase difference between received and detected header is a linear function of frequency offset. Therefore, linear regression method can be employed to directly calculate both frequency offset and initial phase. Derivation based on linear regression method, where we minimize square distance between vector of phase differences between received header and detected header and linear regression model:

$$\left\{\hat{\theta}_0, \Delta\hat{f}\right\} = \min_{\theta_0, \Delta f}\left\{\varepsilon = \sum_{k=0}^{N-1}\left(d_k - \hat{d}_k\right)^2\right\}$$

$$\hat{d}_k = \hat{\theta}_0 + 2\pi T \Delta\hat{f}$$

Here, $\hat{\theta}_0$ is estimated initial phase offset and $\Delta\hat{f}$ is estimated frequency offset.

We obtain following equations by taking partial derivative of target function regarding to θ_0 and Δf and finding its zero:

$$\sum_{k=0}^{N-1} d_k = N \cdot \hat{\theta}_0 + \pi T \Delta\hat{f} N(N-1)$$

$$\sum_{k=0}^{N-1} d_k k = \frac{N(N-1)}{2}\hat{\theta}_0 + \frac{N(N-1)(2N-1)}{3}\pi T \Delta\hat{f}$$

By solving this set of equations, we can compute directly both estimated initial phase and estimated frequency offset.

C. *Joint frequency estimation*

Block for Joint Frequency Estimation uses frequency estimation obtained from Phase & Frequency Estimation block and likelihoods from header detection block to determine carrier frequencies more accurately. Carrier frequencies are fixed in this case. This can be used for determining accurate frequencies for carriers that are in deep fading. As sampling rate of transmitter and receiver can be different, distance between carriers change by a factor equal to ratio of transmitter and receiver sampling frequencies. Hence, block for Joint Frequency Estimation should determine central frequency offset of received signal and distance between adjacent carriers.

Weighted distance between adjacent carriers can be calculated as:

$$\Delta f_{ij} = \frac{f_j - f_i}{j - i} \cdot w_j w_i,$$

Δf_{ij}	is distance between adjacent carriers
f_i	is estimated frequency of *i*-th carrier obtained from Phase & Frequency Estimation block
w_i	is weight of *i*-th carrier frequency estimate calculated like $e^{log\_lkhd(i)}$.
log_lkhd(i)	is log-likelihood of *i*-th carrier header

Distance between adjacent carriers Δf is estimated by averaging weighted distances:

$$\overline{\Delta f} = \frac{\sum_i \sum_j \Delta f_{ij}}{\sum_i \sum_j w_{ij}}, \quad w_{ij} = w_j w_i$$

Weighted Central frequency $(f_c)_{ij}$ is calculated by using frequencies of *i*-th and *j*-th carrier. Lower carrier (index *i*) is in the range 1 to *N*/2 and higher carrier (index *j*) is in the range *N*/2+1 to *N* :

$$(f_c)_{ij} = f_i \cdot w_{ij} + (f_j - f_i) \cdot \frac{(N+1)/2 - i}{j - i} w_{ij}$$

Here, *N* is the number of carriers and we assume that there is even number of carriers.

Central frequency f_c is estimated by averaging weighted central frequencies.

$$\overline{f_c} = \frac{\sum_{i=1}^{N/2} \sum_{j=N/2+1}^{N} (f_c)_{ij}}{\sum_{i=1}^{N/2} \sum_{j=N/2+1}^{N} w_{ij}}$$

Carriers of all headers are then calculated by using estimated f_c and Δf.

D. *Phase error correction and compensation of frequency offset*

Block for Phase Error Estimation (Phase Estimator) operates on π/4-DQPSK de-rotated symbols. It estimates signal phase error and corrects it in a feed-forward loop. Phase estimation algorithm can track phase provided carrier frequency offset is small. Therefore, it is necessary to have frequency offset correction before phase estimator. Compensation for frequency offset is performed by Frequency Integrator block, which integrates phase errors estimated by Phase Estimator and corrects frequency error in a feed-back loop (phase-locked loop). Frequency Integrator is initialized at the beginning to compensate for frequency offset estimated by Phase & Frequency Correction block.

Phase Estimator calculates phase offset by using Viterbi & Viterbi non-data-aided phase estimation algorithm described in [4] and [5]. In the first step, algorithm calculates phase and amplitude of sampled symbol (z_n). Phase is then multiplied by *M,* where *M* is the modulation order, i.e. a number of symbols in a signal constellation. Multiplication by *M* eliminates data dependency of estimated phase. Amplitude of the symbols is raised to power *M*/4, because it is proven to minimize effects of noise compared to powers of *M*/2 and *M*. Resulting measure of phase error is:

$$\left|z_n\right|^{M/4} \cdot e^{j\cdot\arg(z_n)\cdot M}$$

These measures are then averaged over *N* symbols to get estimate of phase error, where *N* is chosen to minimize the delay of the algorithm, while keeping variance introduced by noise at acceptable level:

$$\theta = \frac{1}{M}\arg\left(\sum_n \left|z_n\right|^{M/4} \cdot e^{j\cdot\arg(z_n)\cdot M}\right)$$

We used averaging over eleven symbols in practical implementation. Averaging should be performed symbol-by-symbol in a loop.

Estimated phase lies in the interval [$-\pi/M$, $+\pi/M$]. The algorithm produces *M*-fold phase ambiguity, so correction of the phase estimate should be performed in order to compensate for it. When absolute difference between current and previous phase estimate is greater than π/M, then $2\pi/M$ should be added or subtracted from current phase estimate in order to "unwrap" phase. The algorithm produces accurate phase estimation for central symbol in averaging buffer even in the presence of frequency offset. However, if the frequency offset is close to or greater than $F_{sb}\cdot 2\pi/NM$ (F_{sb} – symbol rate, *N* number of symbols used for averaging), phase estimation becomes unreliable. Phase of the signal is finally corrected by employing a feed-forward phase correction on the basis of the phase estimator output (Figure 2).

Frequency correction is done by correcting phase of each input symbol in accordance with output of Frequency Integrator (FI). Frequency Integrator performs as a proportional-integral (PI) feedback regulator. Output of FI is a sum of phase error multiplied by a proportional factor (proportional part) and integral of phase error multiplied by integration factor:

$$\theta_{FI}(t+1) = \theta_0 + \theta_{FI}(t) + \frac{1}{Ki}\theta_{PHE}(t) + \frac{1}{3Kp}\left(\theta_{PHE}(t) - \theta_{PHE}(t-3)\right)$$

Here, θ_{FI} denotes phase output of FI, θ_0 is proportional to initial frequency correction, and θ_{PHE} is output of phase estimator.

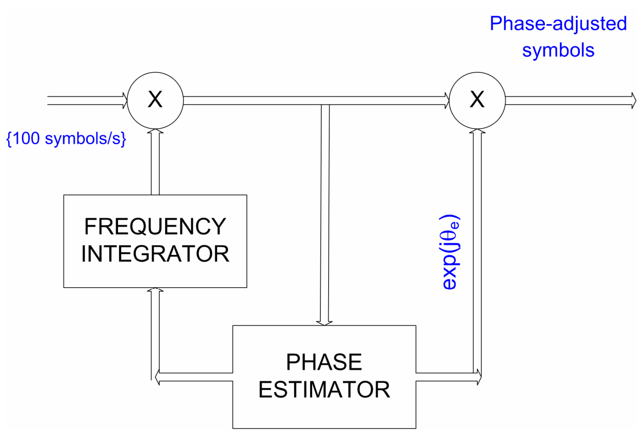

Figure 2. Schematics of phase estimator and frequency integrator block.

III. Verification of Algorithms in MATLAB and Implementation in C++ Programming Language

Described algorithms are verified and their performance is tested in MATLAB numerical computing environment. Every mentioned processing block was implemented as a function and was a part of software receiver implemented in MATLAB. Input signal for software receiver was generated by using software generator of modem signals. Testing was also performed by using recordings of modem signal in short-wave channel.

After verification in MATLAB, these algorithms were implemented in C++ programming language. Every mentioned processing block had its own C++ class. Configuration of algorithm was performed by setting values of variables which are class members, while processing is done by calling class method which as input and output arguments takes arrays of data and lengths of those arrays. Complete software receiver is implemented as Windows DLL library.

Precision of operations is 32 bit floating point. C++ classes are developed to be used in real-time software demodulator. Also, classes can be used with recorded modem signals.

Verification of C++ code is done by comparing results with results obtained in MATLAB simulation for same input data.

IV. Performance of Algorithms

Coarse Frequency Estimator provides accuracy in determining carrier frequency of less than ±5 Hz, even in the presence of strong noise (SNR = 4 dB). Error in frequency estimation of 5 Hz is not critical, because empirical measurements show that header detection algorithm works properly if error in carrier frequency is 10 Hz for SNR higher than 15dB or 5 Hz for SNR lower than 10dB. Residual frequency error can be very precisely measured by Phase & Frequency Estimation block.

Performance of the Phase & Frequency Estimation block is determined by simulation for signal with additive white Gaussian noise. Testing was performed for various frequency offsets and SNR ratios. Frequency offset had discrete values in ±45 Hz range with 5Hz steps while SNR was in range of 2 to 18dB with 2dB step. Standard deviation of frequency and phase offset estimation is determined on thousand iterations for given frequency offset and SNR.

The Table 1 shows standard deviation of frequency and phase estimation in Phase & Frequency Correction block as a function of SNR. Maximum frequency offset that can be detected 99% of cases is also given.

TABLE 1. RESULTS OF TESTING OF THE PHASE AND FREQUENCY CORRECTION BLOCK.

Signal quality		Standard deviation of frequency and phase estimation	
SNR [dB]	Maximum Δf that can be detected	std{Δf }	std{phase}
2	±2 Hz	1.18 Hz	16^o
4	±10 Hz	0.71 Hz	12^o
6	±20 Hz	0.53 Hz	9.2^o
8	±25 Hz	0.42 Hz	7.2^o
10	±35 Hz	0.33 Hz	5.7^o
12	±35 Hz	0.26 Hz	4.5^o
14	±40 Hz	0.20 Hz	3.5^o
16	±40 Hz	0.16 Hz	2.8^o
18	±40 Hz	0.13 Hz	2.2^o

Performance of the Phase Estimator block is determined by simulation. White Gaussian noise is added to the symbols of the single carrier with known phase and frequency offset. This signal is fed into Phase Estimator which provides phase offset estimation for feed-forward and feed-back loop. Residual phase offset and bit error rate in the signal with corrected phase was then calculated. Phase estimation was considered successful if errors could be corrected with forward error correction method.

Simulation was performed on signals with three different values for phase offset: zero, π/2M and π /M. Goal was to determine maximum frequency offset for given SNR that can be reliably estimated and corrected. Simulation was performed for hundred iterations for given phase offset. Frequency offset where more than 95% of packets could be successfully decoded for a given SNR was considered to be maximal frequency offset for reliable estimation. Results of testing are presented in Table 2.

TABLE 2. RESULTS OF TESTING OF THE PHASE ESTIMATOR.

Modulation	SNR (dB)	maximum frequency offset (Hz)
16DPSK	19	0.3
	17	0.2
8DPSK	13.5	0.8
	11.5	0.6
DQPSK	8	1.2
	6	1.2
DBPSK	4	2.8
	2	2.8

V. CONCLUSION

Algorithms presented in this paper are a robust solution for adverse conditions of HF channel. They are developed for usage in software receiver for specific modem but can also be used in other software receivers for similar type of signal.

Implementation of the algorithms in C++ provided fast and efficient solution with good real-time performance. Object oriented approach in algorithm implementation enables easy integration in other software receiver solutions.

REFERENCES

[1] Special Communications Systems Pactor IIex HF modem http://www.scs-ptc.com/shop/products/modems/ptc-iiex-1.

[2] Codan 3012 HF modem http://www.codan.com.au/HFRadio/Products/Modems/3012/Features/tabid/247/Default.aspx

[3] Dimić Goran, Nenadić Nikola, Nikolić Marko, *Symbol and Frame Timing Estimation for Multicarrier HF Packet Radio*, paper submitted to Telfor 2009.

[4] Heinrich Meyr & Gerd Acheid, *Synchronization in Digital Communications: Phase-, Frequency-Locked Loops and Amplitude Control,* John Wiley & Sons, Inc., 1998.

[5] Heinrich Meyr, Marc Moeneclaey, Stefan A. Fechtel, Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing, John Wiley & Sons, Inc., 1998.