# Non-Data Aided Carrier Offset Compensation for SDR Implementation

Jensen, Anders Riis; Jørgensen, Niels Terp Kjeldgaard; Laugesen, Kim; Le Moullec, Yannick

*Published in:*
IEEE NORCHIP 2008

*DOI (link to publication from Publisher):*
10.1109/NORCHP.2008.4738302

*Publication date:*
2008

*Document Version*
Accepted author manuscript, peer reviewed version

# Non-Data Aided Carrier Offset Compensation for SDR Implementation

Anders Riis Jensen[1], Niels Terp Kjeldgaard Jørgensen[1]
Kim Laugesen[1], Yannick Le Moullec[1,2]

[1]Department of Electronic Systems, [2]Center for Software Defined Radio
Aalborg University
DK-9220 Aalborg
Denmark

***Abstract* - This paper deals with the design and implementation of an algorithm for non-data aided carrier offset compensation in a QPSK communication system. The work concentrates on frequency synchronization and we propose a new modulation removal algorithm where the carrier offset is corrected in two steps, reducing the computational complexity. Moreover, a FPGA-accelerated implementation is introduced. Initial tests show that the implemented algorithm works as expected and that the Simulink simulation time is reduced by half when accelerated by the FPGA.**

## I. INTRODUCTION

Carrier synchronization of transmitters and receivers is one of challenges in modern communication systems. When transmitting data through a channel it is modulated on a predefined carrier frequency. The transmitter modulates the transmission burst using a local oscillator to generate the carrier frequency and the receiver demodulates the signal using another local oscillator. As ideal oscillators do not exist, a frequency offset between the transmitter and the receiver oscillators is inevitable. This offset increases the Bit-Error Rate (BER), which calls for compensation techniques. The offset estimation can be aided by transmitting training data, which is known as data aided estimation but has the major drawback of occupying channel bandwidth. To save bandwidth, non-data aided estimation is preferred since the offset is estimated from the actual transmitted data.

The contribution presented in this paper is the design and implementation of a new Modulation Removal (MR) algorithm for carrier offset compensation which can cope with higher sampling rates than existing algorithms and has a limited computational complexity and hardware usage.

## II. CARRIER SYNCHRONISATION SYSTEM

Fig. 1 shows the receiver model used in this work. The received signal is pass-banded to baseband in the analog domain and then converted to a digital sequence where the carrier synchronisation is performed. Finally, the symbols are estimated and decoded. If the frequency in the transmitter, $f_{Tx}$, is not equal to the frequency in the receiver, $f_{Rx}$, then the BER is increased. To accommodate this error a method for frequency offset compensation is introduced. This is done by the carrier synchronization block placed between the Analog to Digital Converter (ADC) and the symbol estimator.

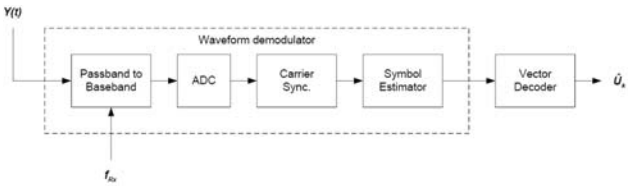


**Fig. 1.** The receiver model including carrier synchronization where *Y(t)* is the received signal, $f_{Rx}$ the receiver frequency and $\hat{U}_k$ the estimated signal.

The signal specifications in this work are as follows:

- Symbol rate: 8400 symbols/s,
- Channel bandwidth: 10.4 kHz,
- Modulation scheme: QPSK,
- Symbols encoding: Gray,
- Carrier offset frequency interval: +/- 40 kHz,
- Burst duration: 80 ms.

Several assumptions have been made:

- It is only the desired channel that is located in the interval Frequency Carrier, $f_c$,+/- 40 kHz. If more than one channel is located in the interval $f_c$ +/- 40 kHz it is impossible for a non-data aided communication system to detect the correct channel.
- In the MR algorithm, a FFT is performed on the received transmission burst. Since the MR algorithm estimates one frequency set per burst the frequency offset must be constant for the entire burst.
- The phase offset of the received transmission bursts are zero. This means that is not necessary to correct the phase before the frequency offset is estimated and the signal corrected.

This is the authors' version of a paper published in the Proceedings of the 26th IEEE NORCHIP conference, 2008.

- The transmission channel is assumed to be an AWGN channel. Distortion of the signal like echoes and Doppler effects are neglected.

## III. INITIAL MR ALGORITHM

Our initial MR algorithm is inspired by [1], which proposes a modulation removal algorithm used to estimate and correct carrier frequency offset. To remove the frequency offset of the received baseband signal, the offset has to be estimated. An intuitive approach is to make a FFT of the baseband signal and choose the frequency bin with the most energy. When noise is applied to the signal the power spectrum is added with noise. When the frequencies in the neighborhood of the offset frequency almost have the same power, it is difficult to detect the correct offset frequency. If the modulation is removed from the received signal, the bandwidth is ideally reduced to 0 Hz. The power of the received signal is then centered at a single frequency, which makes the offset easier to detect when noise is present. Such a system is illustrated in fig. 2.

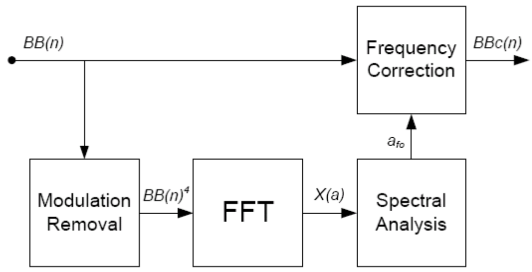


**Fig. 2.** The MR algorithm proposed by [1].

The authors of [1] assume that the baseband signal is sampled with one sample per symbol. In this work it is necessary to apply a higher sampling rate. In this section we propose a modified version of the algorithm found in [1] to support several samples per symbol and in section 4 we propose a 2-step approach to reduce the computational complexity.

In the initial algorithm proposal the baseband signal is sampled with one sample per symbol. When the modulation is removed, the offset carrier frequency is located at four times the original frequency. In our case the symbol rate of 8400 symbols/s makes that the initial algorithm is only able to correct the carrier frequency of +/- 1050 Hz when fulfilling the Nyquist criterion.

Our system specifications tell that the offset carrier frequency $f_o$ can have values in the area from -40 kHz to +40 kHz. This requires a sample rate such as samplerate > 2*4*40 = 320 kHz. The number of samples per symbol, $k$, should then be: $k$ > samplerate/symbolrate = 320/8.40 = 38.1, i.e., $k$>= 39.

The high number of samples per symbol increases the computational complexity of the algorithm components with a factor k. Unfortunately, the higher values of k decreases the accuracy of the FFT. If k is set to 40, there will be a total of k*symbolrate*burstlength = 40*8400Hz*80ms = 26880 samples per transmission burst. This supports a radix-2 FFT of length 16384.

An approach to reduce the computation complexity of the carrier synchronization is presented in section 4.

## IV. PROPOSED 2-STEP MR ALGORITHM

The key ideas behind making the carrier correction in two steps is: 1) Make a computational inexpensive low precision estimation of the carrier offset frequency and correct the frequency, 2) Lowpass filter and downsample the baseband signal and 3) Make a high precision estimation of the carrier offset frequency and correct the frequency. This approach is illustrated in fig. 3.

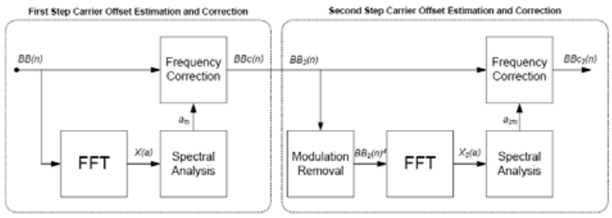


**Fig. 3.** The proposed 2-step carrier correction algorithm.

The argument for making this 2-step carrier correction is to reduce the carrier frequency error to a maximum of half the width of the SQPSK main lobe after the first frequency offset correction step. This low precision correction will make it possible to reduce the number of sample per symbol needed in the second frequency offset correction step, without loosing signal information. By doing this, the computational complexity of the high precision estimation and carrier correction is reduced.

### *A. First step of the carrier correction algorithm*

The low precision estimation is performed without the use of modulation removal. There are three reasons for this: 1) lower sample rate: the MR algorithm needs four times higher sample rate as the algorithm multiplies the carrier frequency with four. This is not the case without the MR algorithm where the sample rate only has to be one fourth compared to the MR algorithm, 2) better SNR: as only a low precision estimate is needed the length of the FFT can be reduced, the estimation without MR algorithm gives a better SNR at the offset frequency than the MR algorithm for short FFT lengths, and 3) less complex: there is no need to raise to the power of four.

Selecting the FFT length, N1, is equal to selecting the FFT precision, $\Delta f_1$. There are no high demands to $\Delta f_1$ except that it has to be smaller than half the main lobe width of 8484 Hz. With $\Delta f_1$ set to be less than 100 Hz, the length of N1 is > 840. The FFT algorithm has the best performance using a length that is a radix-2 number. Therefore N1 is set to 1024. This gives a $\Delta f_1$ of 82 Hz and a total correction error of maximum +/- 8566 Hz.

### *B. Second step of the carrier correction algorithm*

To obtain a high precision estimation the MR algorithm is used in the second correction step. From the first correction step the maximum frequency error is +/- 8566 Hz. When the signal is raised to the power of four in the

This is the authors' version of a paper published in the Proceedings of the 26th IEEE NORCHIP conference, 2008.

MR algorithm, the carrier frequency is multiplied by four which results in a maximum frequency deviation of +/- 34.3 kHz. Using this information, the number of samples per symbol in the second correction step, $k_2$, is calculated using the Nyquist theorem: $k_2$ > (2*34.3kHz)/8400Hz, ie., $k_2$ >= 9.

*C. Test of the Initial MR Algorithm versus the 2-Step Carrier Correction Algorithm*

Each algorithm has processed 1000 different transmission bursts. The Matlab profiler is used to evaluate the execution time of each algorithm. In the performance test for low SNR, each algorithm has processed 1000 different transmission bursts at each simulated SNR value. The known frequency offset for each burst is then compared to the estimated frequency offset. If the difference is larger than 10 Hz the estimation is rated as a failure. The specifications for the algorithms are listed in table 1.

TABLE 1 ALGORITHM SPECIFICATIONS

| | Initial MR algorithm | 2-step algorithm Step 1 | Step2 |
|---|---|---|---|
| FFT length | 16384 | 1024 | 4096 |
| Samples/symbol | 40 | 10 | 10 |

Each individual burst has a symbol rate of 8400 Hz and a length of 80ms which corresponds to 672 symbols. The carrier frequency offset is randomly chosen in the interval from -40 kHz to 40 kHz. The execution times for 1000 random generated bursts are listed in table 2.

TABLE 2 1000 BURST EXECUTION TIMES (S)

| | Initial MR algorithm | 2-step algorithm Step 1 | Step2 |
|---|---|---|---|
| Modulation removal | 0.844 | - | 0.197 |
| Spectral analysis | 0.537 | 0.114 | 0.179 |
| FFT | 0.471 | 0.040 | 0.121 |
| Carrier Correction | 1.524 | 0.673 | 0.639 |
| Execution time | 3.376 | 0.827 | 1.136 |
| | | 1.963 | |

Table 2 shows that the proposed algorithm is 38 % faster than the initial MR algorithm. The speed gain is obtained via the sampling rate which is lowered by 1/4. Fig. 4 illustrates the error ratio for both algorithms at different SNR values.

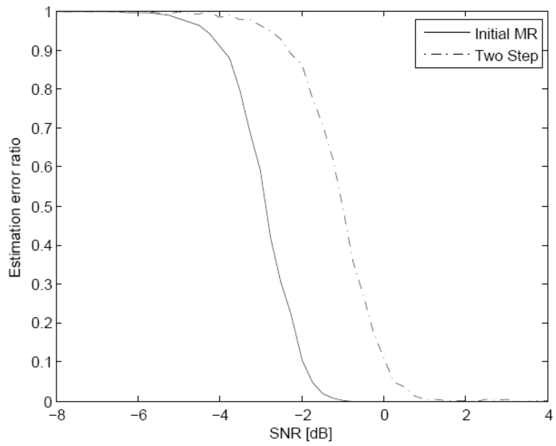


**Fig. 4.** Comparison of the SNR estimation errors.

Fig. 4 shows that the proposed 2-step algorithm has a good performance above 4 dB SNR whereas the initial MR algorithm performs well as low as 2 dB SNR. This is normal since a SNR at 4 times the offset frequency increases with the length of the FFT. The second step in the proposed algorithm is identical to the initial MR algorithm except that the FFT length is 1/4 of the one in the initial MR algorithm. So the 2-step carrier correction algorithm is faster at the cost of low-SNR performance. However, performance can be significantly improved, as discussed later on in section 6.

## V. FPGA ACCELERATED SIMULATION

The FPGA used in this work is an Altera Cyclone II [2] containing 35,000 logic elements and connected to a 50 MHz clock. The method used in this work is called Hardware In the Loop (HIL), where the FPGA executes part(s) of the algorithm and a PC sends/receives data to/from the FPGA. This can potentially accelerate the simulation if the algorithm exhibits inherent parallelism. The MR algorithm is described using Altera's DSP Builder [3] which contains Simulink blocks that can be executed on the FPGA after a synthesis process.

*A. Modulation Removal*

When the modulation removal is performed, the input signal is raised to the power of four. This is done by squaring the signal twice. The input signal to the square block is a signed 16 bit Q14 number, therefore the output of the multiplications in this block is a signed 32 bit Q28 number. This number is truncated such that the output of the block is a signed 16 bit Q14 number.

*B. FFT*

The FFT block is implemented using the Altera MegaCore FFT block. The essential settings in this work are: 1) data precision: 16 bit, 2) data flow: streaming allowing the FFT length to change dynamically, which is convenient for the two step carrier correction algorithm, 3) FFT length: 64, and 4) implementation of multipliers: logic elements and DSP blocks.

*C. Spectral Analysis*

The input to this block is a complex number from the FFT analysis; therefore the magnitude of this number is

This is the authors' version of a paper published in the Proceedings of the 26th IEEE NORCHIP conference, 2008.

computed. Next the 64 first magnitudes are stored, as they represent the FFT analysis of the input signal. This is done using a state machine, with the four states: init, write, read and wait controlling a FIFO memory block for storing the 64 magnitudes.

*D. Frequency Correction*

The frequency correction consists of an exponential function and a sine/cosine generator. The sine and cosine generator is created from a MegaCore block called NCO. The output from the NCO is a sine and a cosine that is used for the exponential function for the frequency carrier correction.

*E. Test of the FPGA implementation MR Algorithm*

The MR algorithm is tested without noise, as the hardware-implemented MR algorithm only works when the input SNR is approximately 30 dB, which practically is a noise free input. The FFT analysis with a length of 64 is covering a spectra of +/-42 kHz, meaning that the precision of the FFT is = 1.3125 kHz.

The FPGA implementation (HIL) has been tested for two scenarios, i.e., when the frequency offset is a multiple of the FFT precision or not, both on an Intel Centrino 1.7 GHz PC with 1 GB of RAM. The implementation uses 95 % of the multipliers and 51 % of the logic elements. After the data is processed by the FPGA, the results are saved in the Matlab workspace, where they are compared with the results from the non-accelerated Simulink tests.

*E.1 Frequency Offset is a Multiple of the FFT Precision*

Here the frequency offset is chosen to -13.125 kHz. The QPSK signal is then frequency-shifted according to the frequency offset.

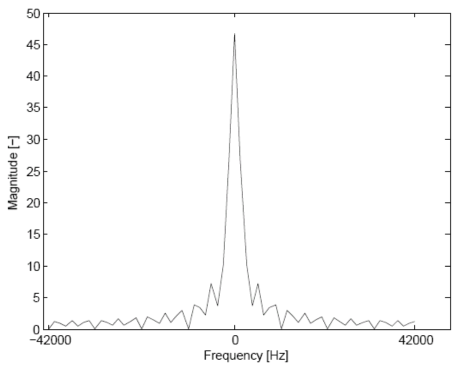


**Fig. 5.** Frequency corrected output when the offset is a multiple of the FFT Precision.

Fig. 5 shows that there is no frequency offset in the signal, and that the FFT analysis is symmetric. The output from the test of the MR algorithm is compared with the non frequency-shifted QPSK signal, and the bit error rate is 0, i.e., the algorithm works properly.

*E.2 Frequency Offset is not a Multiple of the FFT Precision*

In this second scenario the frequency offset is now +12.5 kHz. Fig. 6 illustrates the frequency corrected output: it is seen that the frequency offset is removed, but the FFT analysis is not symmetric. This results in a bit error rate of 0.5 which was expected as a consequence of the low FFT precision.

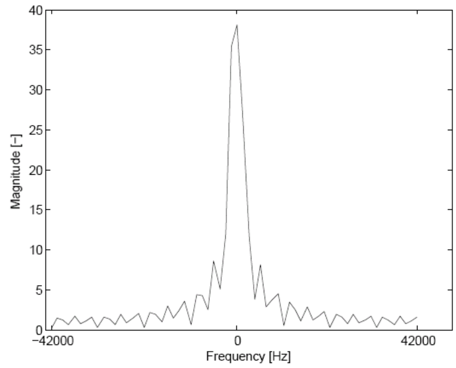


**Fig. 6.** Frequency corrected output when the offset is not a multiple of the FFT Precision.

SNR-wise the FPGA-accelerated and non-accelerated results are identical. However, the execution times are significantly different: 2.5 minutes for the non-accelerated version vs. 1.2 minute for the FPGA-accelerated one, which about twice as fast.

## VI. IMPROVED 2-STEP ALGORITHM

Inserting a Low-pass Filter (LP) between the 2 steps, cf. fig. 7, significantly improves the performance of the proposed solution. Fig. 8 shows that the LP 2-step version outperforms not only the non-LP version but also the initial one.

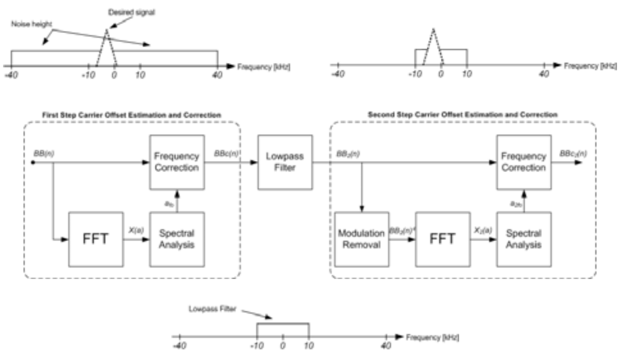


**Fig.7** LP-version of the 2-step algorithm.

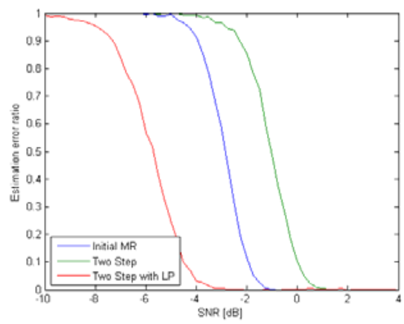


**Fig.8** SNR estimation errors. Center: initial, right: 2-step, left: 2-step with LP.

## VII. CONCLUSION

This is the authors' version of a paper published in the Proceedings of the 26th IEEE NORCHIP conference, 2008.

The design and implementation of an algorithm for non-data aided carrier offset compensation in a QPSK communication system has been presented. A new 2-step modulation removal algorithm has been proposed; it supports high sampling-rates and minimizes the computational complexity. This is achieved, in the non-LP version, at the cost of a slightly degraded performance for SNRs below 4dB. A FPGA-accelerated implementation of the non-LP version performs as expected and the simulation time is reduced by half. Finally, inserting a LP filter between the 2 steps significantly increases the performance. Future work will investigate the FPGA implementation of that version.

## REFERENCES


[1] T. Brack, U. Wasenmüller, D. Schmidt and N. When. *Design Space Exploration for Frequency Synchronization of BPSK/QPSK Bursts*. Advances in Radio Science, Volume 3, pp. 337-341, 2005.

[2] Altera Cyclone II Data Sheet. February 2008

[3] Altera DSP Builder User's Guide 8.0. May 2008



This is the authors' version of a paper published in the Proceedings of the 26th IEEE NORCHIP conference, 2008.