

Envelope and phase delays correction in an EER radio architecture

Jean-François Bercher, Corinne Berland

▶ **To cite this version:**

Jean-François Bercher, Corinne Berland. Envelope and phase delays correction in an EER radio architecture. Analog Integrated Circuits and Signal Processing, 2008, 55 (1), pp.21-35. hal-00621953

HAL Id: hal-00621953
https://hal.science/hal-00621953

Submitted on 17 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analog Integrated Circuits and Signal Processing manuscript No.
(will be inserted by the editor)

J.-F. Bercher · C. Berland

Envelope and Phase delays correction in an EER radio architecture

Received: March 30, 2007 / Revised: date

Abstract This article deals with synchronization in the Envelope Elimination and Restoration (EER) type of transmitter architecture. To illustrate the performances of such solution, we choose to apply this architecture to a 64 carriers 16QAM modulated OFDM. We first introduce the problematic of the realisation of a highly linear transmitter. We then present the Envelope Elimination and Restoration solution and draw attention to its major weakness: a high sensitivity to desynchronization between the phase and envelope signal paths. To address this issue, we propose an adaptive synchronization algorithm relying on a feedback loop, a Least Mean Square formulation and involving an interpolation step. It enables the correction of delay mismatches and tracking of possible variations. We demonstrate that the quality of the interpolator has a direct impact on Error Vector Magnitude (EVM) value and output spectrum. Implementation details are provided along with an analysis of the behaviour and performances of the method. We present HPADS and Matlab simulation results and then focus on the enhancement of the transmitter performances using the proposed algorithm.

1 Introduction

Recent radiocommunication systems aiming at high data rate are based on efficient modulation schemes in which Quadrature Amplitude Modulations (QAM) are obviously preferred to frequency or phase modulation. While the 3GPP standard employs QPSK and 16QAM, higher data rate are achieved (in WLAN system for example) using Orthogonal Frequency Data Multiplexing (OFDM) modulation. Although these modulations are suitable for signal processing, the realization of the RF front end, particularly the transmitter, becomes more and more complex.
In the conception of a transmitter, it is essential to achieve both efficiency and linearity. In radiocommunication standards, the quality of the transmitted signal is well defined, usually in terms of output spectrum, Adjacent Channel Power Ratio (ACPR) and Error Vector Magnitude (EVM). When the modulated signal presents an envelope variation, similar to an amplitude modulated signal, the compression effect of the Power Amplifier (PA) generates intermodulation products which directly impact the three mentioned figures of merit. This implies that with a classic class A PA, the input signal would have to present a mean power (depending on the type of modulation) lower than the input compression point: the difference is quantified in terms of back off. In order to gain in efficiency, a linearization system is often preferred to a linear amplification.
In this paper, we deal with the *Envelope Elimination and Restoration* (EER) principle, and illustrate it in the case of a 16QAM 64 carriers OFDM modulation. The EER principle was proposed by Kahn in 1952 [14] and is based on the splitting of a modulated signal into two signals. The first one is a constant envelope phase modulated signal, while the second one is the envelope of the original signal. In the original EER transmitter, the splitting is realized in an analog way using a limiter and an envelope detector. The phase modulated signal is the input of an efficient PA whereas the envelope signal is sent to a switching power supply which feeds the last stage of the PA. The transmitter now evoluates toward a fully digital transmitter [23] as shown in Fig. 1.

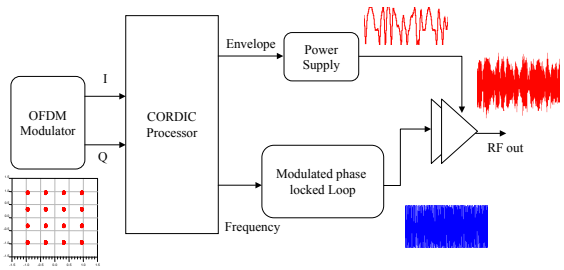


Fig. 1 Polar transmitter using a CORDIC processor

J.-F. Bercher and C. Berland
ESYCOM-ESIEE, *Université Paris-Est*
Cité Descartes, BP99, 93162 Noisy-le-Grand Cedex, France
Tel.: +33-1-45926642
Fax: +33-1-45926699
E-mail: {jf.bercher,c.berland}@esiee.fr

The digital creation of the two signals is achieved using a CORDIC processor [22]. The principle of this processor relies on the rotation of a vector (X_i, Y_i) to a new vector. In our case, the final vector is (Z_i, θ), where Z_i is the magnitude of the vector and the rotation angle is our phase. VLSI implementation can profit from of a suitable iterative formulation. The phase signal is sent to the PA amplifier using a modulated PLL. This solution is preferred to a I/Q RF modulator in order to reduce transmitted spurious emissions. To limit distortion, the envelope signal is usually amplified with a class S modulator using a $\Sigma\Delta$ modulator rather than a classic PWM. The final amplification is realized by a high efficiency amplifier. As the input signal is a constant envelope signal, switched class of power amplifiers are preferred due to their high efficiency. As demonstrated in [20], the class E power amplifier is very suitable for this application since the expression of the output voltage is directly proportionnal to its power supply. This property is mandatory as we intend to reinject the envelope variation through the modulation of its power supply.

However the main drawback of this architecture is its sensitivity to delay mismatch between the two signals.

The delays introduced by the two paths can be mismatched due to pipeline differences in the paths and the delay in the anti-alias filter (amplitude path), as well as small contributions from other analog delays [21]. Furthermore, in a production environment, delays should be matched to variations in process including supply voltage, frequency, output power and temperature [18]. This usually requires careful factory calibration procedures. The mismatch deteriorates both the EVM and the output spectrum of the transmitted signal [4; 24] and has to be corrected. A linear interpolation was suggested in [11] to compensate for the mismatch, while a group delay equalizer was proposed in [17]. In these two cases, it still remains to identify the delay mismatch and track its possible variations. In this article, we propose a scheme and an associated algorithm that covers the whole calibration problem: identification, correction and tracking.

In the first part, we demonstrate the sensitivity of EER applied to a 16QAM 64 carriers OFDM modulation and bring forward the maximum tolerable delay mismatch for this modulation. The second part presents an efficient algorithm which corrects this default. We then focus on the implementation of the algorithm and on the importance of the interpolation filter used to resynchronize the signals. In the final part, an analysis of the behaviour and performances of the algorithm is provided. Simulation results performed on HPADS are presented and show the performances achieved with this solution in terms of output spectrum, EVM and ACPR.

2 Impact of delay mismatches on an OFDM modulation

Using a 16QAM 64 carriers OFDM modulation is an interesting case study for the validation of this kind of architecture because of its high Peak to Average Power Ratio (PAPR). In fact, for an OFDM modulation, the PAPR is equal to the number of carriers, which corresponds to $10\log(64) = 18$dB in our simulation case. A few results have already been presented on the study of the whole transmitter with this modulation [2] and this paper indicates that the critical specification is the synchronization of signals. Fig. 2 and Fig. 3 present the impact of delay mismatch in the range of $\pm\frac{T}{3}$, with T the time symbol, on the output spectrum and on the EVM. Table 1 gives the values of ACPR for these delays. The symbol rate is 20MHz.

Table 1 ACPR obtained for different delay mismatches. It is evaluated for the full 64 carriers 20MHz OFDM modulation, with a channel spacing of 25MHz.

Delay	$T/27$	$T/13$	$T/5$	$T/3$
ACPR (dB)	-37	-29.7	-20.7	-15.7

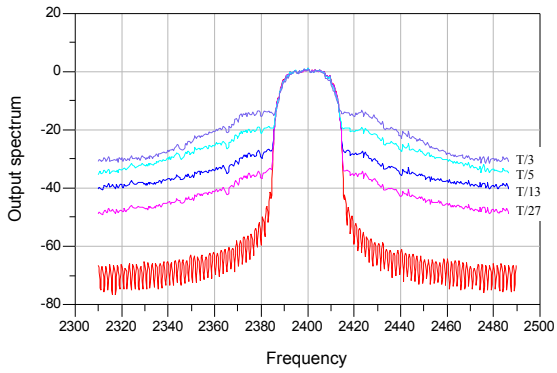


Fig. 2 Impact of the delay mismatch on output spectrum (taken in 200kHz bandwidth)

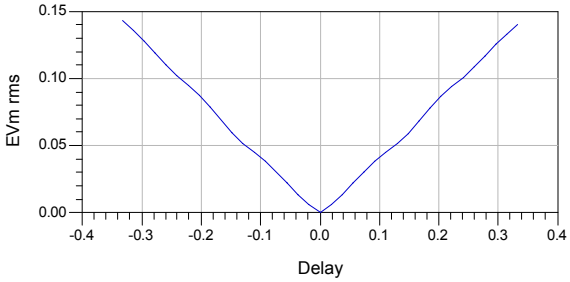


Fig. 3 Impact of the delay mismatch on the EVM

It is important to notice that the EVM calculation for OFDM modulations must be realized on each carrier separately and then averaged. Fig. 4 shows that the effect of the desynchronization on the first subcarrier for delays $T/27$ and $T/5$ acts as additional noise.

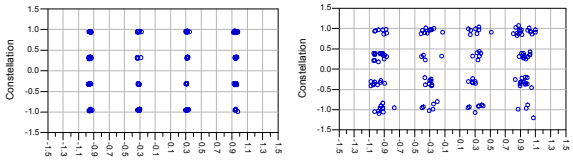


Fig. 4 Impact of the delay mismatch on the constellation for T/27 and T/5

As a guideline, we take specifications from 802.11.a standard: the rms EVM is specified at a maximum of 11.22% and the output spectrum has to remain under -40dBc above a 30MHz frequency offset from the carrier measured in a 1MHz resolution bandwidth. When analysing the EVM results, we can observe that a desynchronization of $T/3$ produces an EVM value of 14% and when considering the output spectrum, in 1MHz bandwidth, the value at 30MHz from the carrier is about -7.5dBc. In fact, the value of -40dBc is only obtained for a delay of $T/27$ which gives an EVM value of about 0.6% This demonstrates that the synchronization of signals impacts so strongly on the output spectrum that it is mandatory to implement a correction algorithm.

3 The correction algorithm

This paper is an extended version of [3] where we presented a preliminary version of the algorithm. In this version, the algorithm is presented in more details and implementation issues are discussed. More specifically, we discuss the role, performances and implementation of the interpolation step and provide a detailed analysis of the algorithm performances.

3.1 Envelope and phase alignment

According to the central limit theorem, for complex modulation schemes such as the OFDM, when the number of subcarriers is large, the emitted signal can be approximated as a gaussian distributed complex random variable. For a narrowband stationary signal written as

$$\xi(t) = x(t)\cos(\omega_0 t) + y(t)\sin(\omega_0 t) \tag{1}$$

it is well known that $R_{xx}(\tau) = R_{xy}(\tau)$ and that $R_{xy}(\tau) = -R_{yx}(\tau)$, where R_{xx} and R_{xy} are respectively the autocorrelation and intercorrelation functions. The processes $x(t)$ and $y(t)$ are always uncorrelated at the same instant, that is $R_{xy}(0) = 0$. If $\xi(t)$ is normal, then $x(t)$ and $y(t)$ are independent at the same instant. The complex gaussian process $\xi(t)$ is completely characterized by its mean and autocorrelation function

$$R_{\xi\xi}(\tau) = R_{xx}(\tau)\cos(\omega_0\tau) + R_{xy}(\tau)\sin(\omega_0\tau). \tag{2}$$

Let us denote $S_{\xi\xi}(f)$ and $S_{xx}(f)$ the spectra of $\xi(t)$ and $x(t)$, that is the Fourier transforms of the autocorrelation functions $R_{\xi\xi}(\tau)$ and $R_{xx}(\tau)$. When $S_{\xi\xi}(f)$ is symmetric with central frequency f_0, the in-phase and quadrature components $x(t)$ and $y(t)$ are uncorrelated, that is $R_{xy}(\tau) = 0$. The baseband spectrum $S_{xx}(f)$ is generally proportional to the square of the transfer function of the emission filter, which shall be shaped as a square-root Nyquist filter. Consequently, the autocorrelation function $R_{xx}(\tau) = R_{yy}(\tau)$, which is the inverse Fourier transform of the baseband spectrum, is the impulse response h of a (full) Nyquist filter. The process $\xi(t)$ can also be written as

$$\xi(t) = \rho(t)\cos(\omega_0 t - \phi(t)) \tag{3}$$

where $\rho(t)$ is the envelope and $\phi(t)$ is the phase process, with

$$\begin{cases} x(t) = \rho(t)\cos(\phi(t)), \\ y(t) = \rho(t)\sin(\phi(t)). \end{cases}$$

In the case of a complex gaussian process, it is well known that the envelope and phase are independent *at the same instant* and respectively distributed according to the Rayleigh and uniform distributions. The case of delayed envelope and phase is less known. In fact, it appears that for gaussian processes, the envelope $\rho(t)$ and phase $\phi(t-\Delta)$ are also Rayleigh and uniform distributed, and are *always* independent (see [4]) with no reference to the correlation coefficient, whatever the delay between envelope and phase components. In consequence, the output do not convey any information on the time alignment or mismatch between the envelope and phase components. As a result, it is *not* possible to correct the relative delay between the envelope and phase components from the sole observation of the system output. Calibration then needs to rely on a feedback loop and involve a direct comparison between the initial "aligned" signal and the observed signal.
In a calibration step without data transmission, we can also modify the modulation scheme and design non-gaussian sequences with some dependence between envelope and phase. In such a situation, a 'contrast' based on the output properties may be devised in order to align the components. However, we will focus here on the feedback solution that preserves the data and modulation technique.

3.2 The compensation algorithm

Let us denote $z(t)$ the output of the system. Due to the delays Δ_1 and Δ_2 that affect the envelope and phase components, we have

$$\begin{aligned} z(t) &= \rho(t-\Delta_1)\cos(\omega_0 t - \phi(t-\Delta_2)) \\ &= \rho(t-\Delta_1)\cos(\phi(t-\Delta_2))\cos(\omega_0 t) \\ &\quad + \rho(t-\Delta_1)\sin(\phi(t-\Delta_2))\sin(\omega_0 t). \end{aligned}$$

We propose here to correct the delays using an adaptive precompensation. The synchronization algorithm relies on the

idea of introducing two advances μ_1 and μ_2 in order to precompensate the delays, as illustrated in Fig. 5. In such a case, the output becomes

$$z(t) = \rho(t+\mu_1-\Delta_1)\cos\left(\omega_0 t - \phi(t+\mu_2-\Delta_2)\right) \qquad (4)$$

and we will adjust the advances μ_1 and μ_2 in order to minimize a statistical distance between $z(t)$ and $\xi(t)$. A natural criterion is the minimization of the quadratic distance

$$J(\mu_1,\mu_2;t) = \mathrm{E}\left[|\xi(t)-z(t)|^2\right], \qquad (5)$$

where $\mathrm{E}[\bullet]$ is the statistical expectation operator. This expression can also be rewritten as

$$\begin{aligned} J(\mu_1,\mu_2;t) = {} & \mathrm{E}\left[e_x(t)^2\right]\cos(\omega_0 t)^2 + \mathrm{E}\left[e_y(t)^2\right]\sin(\omega_0 t)^2 \\ & + \mathrm{E}[e_x(t)e_y(t)]\sin(2\omega_0 t) \end{aligned}$$

where $e_x(t)$ and $e_y(t)$ are the errors for the in-phase and quadrature components respectively,

$$e_x(t) = (\rho(t)\cos\phi(t) - \rho(t_1)\cos\phi(t_2)) \qquad (6a)$$
$$e_y(t) = (\rho(t)\sin\phi(t) - \rho(t_1)\sin\phi(t_2)) \qquad (6b)$$

and where, in order to simplify the expressions, we noted

$$t_1 = t+\mu_1-\Delta_1 \text{ and } t_2 = t+\mu_2-\Delta_2. \qquad (7)$$

After time averaging, this simply reduces to

$$J(\mu_1,\mu_2) = \frac{1}{2}\left\{J_x(\mu_1,\mu_2) + J_y(\mu_1,\mu_2)\right\} \qquad (8)$$

with

$$J_x(\mu_1,\mu_2) = \mathrm{E}\left[e_x(t)^2\right] \text{ and } J_y(\mu_1,\mu_2) = \mathrm{E}\left[e_y(t)^2\right]. \qquad (9)$$

The global criterion equals the sum of two elementary criteria on the in-phase and quadrature components. We can readily obtain the same criterion (up to a factor) using the demodulated, baseband, version of the signal.

In practice, we indeed work in baseband with digital signals. After sampling, we compare the digital input signal to the sampled baseband output. In the following, we will keep T for the symbol period and note T_s the sampling period. Consequently, we will note the different discrete time indexes as follow

$$\begin{cases} t(n) = nT_s \\ t_1(n) = nT_s + \mu_1 - \Delta_1 \\ t_2(n) = nT_s + \mu_2 - \Delta_2. \end{cases} \qquad (10)$$

The output sampling clock does not need to be synchronous to the input: it may be a divided version of the input clock, and any propagation delay will be absorbed in the correction procedure.

Since it is simpler to generate delayed signals than advanced signals, we introduce a small processing delay D in Fig. 5.

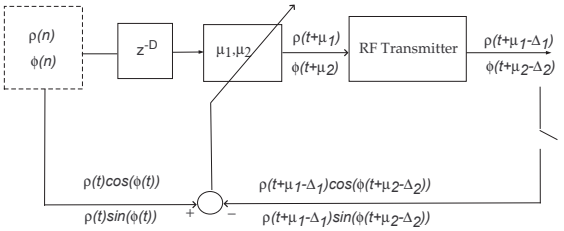


Fig. 5 Principle of delays correction.

Let us consider the criterion $J_x(\mu_1,\mu_2)$ in (9) on the in-phase component. Developing and taking into account the independence between $\rho(t_1)$ and $\phi(t_2)$ lead to

$$J_x(\mu_1,\mu_2) = 2C_x(0,0) - 2C_x(\mu_1-\Delta_1,\mu_2-\Delta_2) \qquad (11)$$

where

$$C_x(\tau_1,\tau_2) = \mathrm{E}\left[\rho(t)\cos(\phi(t))\rho(t-\tau_1)\cos(\phi(t-\tau_2))\right] \qquad (12)$$

with $\tau = 1 = \mu_1-\Delta_1$ and $\tau_2 = \mu_2-\Delta_2$, is a kind of 'correlation function'. The term $C_x(0,0)$ simply reduces to $C_x(0,0) = \mathrm{E}\left[\rho(t)^2\right]\mathrm{E}\left[\cos(\phi(t))^2\right] = R_{xx}(0)$, the variance of the in-phase $x(t)$ component. It is important to note that $C_x(\tau,\tau)$, obtained with $\tau_1 = \tau_2 = \tau$ is nothing else but the correlation function $R_{xx}(\tau)$. Since we know that the correlation function is proportional to the shaping filter h, it appears that the behaviour of the criterion $J_x(\mu_1,\mu_2)$ is closely related to the shaping filter. Regarding the quadrature component and criterion $J_y(\mu_1,\mu_2)$, the same conclusions and formulas found in (11,12) are easily obtained by substituting x by y and cos by sin.

In the case of an OFDM modulation, the criterion $J(\tau_1,\tau_2)$ was evaluated numerically by Monte-Carlo simulations with a square root Nyquist filter (square root raised cosine with 0.5 roll-off). This is illustrated in Fig. 6 for delays (advances) between $-4T$ and $4T$. We can recognize here the general shape of the (inverted) impulse response of a raised cosine, somewhat distorded and modulated. The criterion does not only present a global minimum at $\tau_1 = 0$, $\tau_2 = 0$, but also several other minima. Derivation of a closed-form formula for criteria (9) involving (12) is a challenging if not impossible task. However, in Fig. 7, where the criterion for delays less than $1.5T$ is shown, we clearly observe that any descent algorithm will avoid local minima for delays $\Delta_1,\Delta_2 \leq T$, with initial conditions set to zero.

3.3 Gradient algorithm

Since we do not have a closed-form for the criterion nor a direct explicit solution for its global minimizer, we need to exhibit the solution using a descent algorithm. We simply

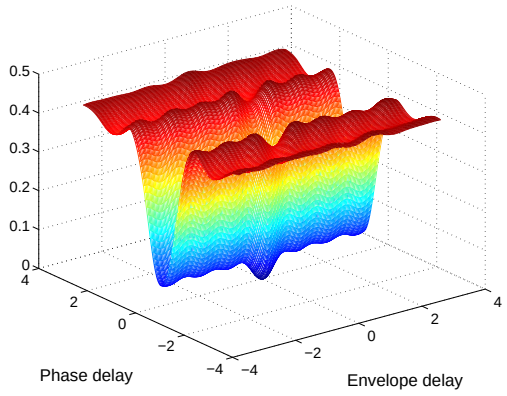


Fig. 6 Shape of the criterion for the envelope and phase delays $\in [-4T, 4T]$, with T the symbol period. The shape of the criterion is related to the impulse response of the shaping filter and presents local minima.

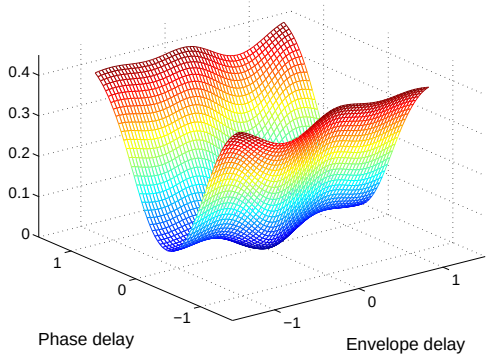


Fig. 7 Shape of the criterion for the envelope and phase delays $\in [-3T/2, 3T/2]$, with T the symbol period.

use a gradient algorithm that consists in iterating the following formulas:

$$\begin{cases} \mu_1(n+1) = \mu_1(n) - \gamma_1(n) \left.\frac{\partial J(\mu_1,\mu_2)}{\partial \mu_1}\right|_{\mu_1=\mu_1(n)} \\ \mu_2(n+1) = \mu_2(n) - \gamma_2(n) \left.\frac{\partial J(\mu_1,\mu_2)}{\partial \mu_2}\right|_{\mu_2=\mu_2(n)} \end{cases} \tag{13}$$

where $\gamma_1(n)$ and $\gamma_2(n)$ are two adaptation steps that possibly depend on the iteration index n. The gradients are given by

$$\frac{\partial J(\mu_1,\mu_2)}{\partial \bullet} = \frac{1}{2}\left[\frac{\partial J_x(\mu_1,\mu_2)}{\partial \bullet} + \frac{\partial J_y(\mu_1,\mu_2)}{\partial \bullet}\right] \tag{14}$$

and we readily obtain

$$\frac{\partial J_x(\mu_1,\mu_2)}{\partial \mu_1} = -E\left[\left.\frac{d\rho(u)}{du}\right|_{u=t_1(n)} \cos\phi(t_2(n))e_x(t(n))\right] \tag{15a}$$

$$\frac{\partial J_y(\mu_1,\mu_2)}{\partial \mu_1} = -E\left[\left.\frac{d\rho(u)}{du}\right|_{u=t_1(n)} \sin\phi(t_2(n))e_y(t(n))\right] \tag{15b}$$

and

$$\frac{\partial J_x(\mu_1,\mu_2)}{\partial \mu_2} = -E\left[\left.\frac{d\cos\phi(u)}{du}\right|_{u=t_2(n)} \rho(t_1(n))e_x(t(n))\right] \tag{16a}$$

$$\frac{\partial J_y(\mu_1,\mu_2)}{\partial \mu_2} = -E\left[\left.\frac{d\sin\phi(u)}{du}\right|_{u=t_2(n)} \rho(t_1(n))e_y(t(n))\right] \tag{16b}$$

with e_x and e_y defined by (6a,6b).
The update equations are obtained using these gradients in (13). However, we do not have the analytical expressions of the statistical expectations involved in these formulas. Therefore, we have to resort to using a stochastic approximation of these theoretical recursions. A popular solution in adaptive filtering is the *Least Mean Squares* (LMS) algorithm that simply consists in omitting the statistical expectation. The LMS then involves the *instantaneous gradient* rather than the (correct) statistical average. Furthermore, the equations are updated at each new sample. This gives

$$\mu_1(n+1) = \mu_1(n) + \gamma_1(n)\left.\frac{d\rho(u)}{du}\right|_{t_1(n)} \times \left(\cos(\phi(t_2(n)))e_x(t) + \sin\phi(t_2(n))e_y(t(n))\right) \tag{17}$$

$$\mu_2(n+1) = \mu_2(n) + \gamma_2(n)\rho(t_1(n)) \times \left(\left.\frac{d\cos\phi(u)}{du}\right|_{t_2(n)} e_x(t) + \left.\frac{d\sin\phi(u)}{du}\right|_{t_2(n)} e_y(t(n))\right). \tag{18}$$

Practical implementation of formulas (17) and (18) requires

- computation of the errors $e_x(t(n))$ and $e_y(t(n))$ defined in (6a) and (6b) which results from the comparison of the system input and output,
- computation of the derivatives that can be simply approximated by finite differences of $\rho(t_1(n))$ and $\cos(\phi(t_2(n)))$.

Of course, the algorithm can be simplified by considering a sole error component rather than two. For instance, one can simply put $e_y = 0$ in the previous equations. In practice, simulations show that the gain associated with the second component is extremely small.

For formulas (17) and (18), the computational load is about 8 real multiplications per iteration. However, since the derivatives must be computed at time $t_1(n)$ and $t_2(n)$, that is at the output of the corrected system, $\rho(t)$ and $\cos(\phi(t))$ must be separately accessible. This implies a quadrature demodulation before the feedback loop. Furthermore, with this approach, we need to adjust two advances μ_1, μ_2 and apply them to the input signal. Adopting very high sampling frequencies in order to get the required precision may not be an efficient solution. Digital interpolation is a more effective solution as it keeps reasonable sampling frequencies and save consumption.

4 Impact of the interpolator

Contrary to what we indicated in [3], the interpolation procedure has a significant impact on the performances of the algorithm in terms of the EVM and output spectrum.
According to the Shannon-Nyquist sampling theorem, we know [29] that any band-limited signal $x(t)$ can be recovered exactly from its samples $x(m) = x(mT_s)$ taken at the sampling frequency $1/T_s$ by the formula

$$x(t) = \sum_m x(m)\mathrm{sinc}(\pi(t - mT_s)/T_s), \tag{19}$$

where sinc is the cardinal sine. This indicates that, in principle, the samples convey enough information to reconstruct the original signal at any desired time. In particular, it is possible to reconstruct $x(t-\tau)$, for any τ, and therefore new shifted samples $x(kT_s - \tau)$ from the original samples $x(kT_s)$, according to

$$x(kT_s - \tau) = \sum_m x(mT_s)\mathrm{sinc}(\pi(kT_s - \tau - mT_s)/T_s). \tag{20}$$

The above expression is in the form of a digital convolution and can be implemented as a filtering operation. However, because the underlying filter has an infinite (sinc) impulse response, and is non causal, practical implementation introduces truncation and delay. Another possibility is to use a convenient approximation of the ideal interpolator mentioned above. The MMSE FIR interpolator [19] is the minimum mean square error approximation of the ideal filter with finite impulse response. Although optimum implementations of these interpolators exist [6], the coefficients shall be pre-computed and tabulated for each possible fractional delay and these types of structures should be reserved for the interpolation with fixed delays.

Another class of interpolators relying on polynomial approximation can be used instead. Indeed, the Weierstrass approximation theorem states that every continuous function defined on an interval can be uniformly approximated as closely as desired by a polynomial function. We can then use a polynomial to approximate the value of the function, given a series of samples, at the desired delay. Such interpolators are especially interesting for our application since they can be described by FIR filters. They can be implemented very efficiently in hardware, and their coefficients can be computed in real time rather than taken from a table. An efficient structure was devised by Farrow [8; 15] and improvements to the structure can be found in [25; 7; 5]. In this structure, the delay is directly adjustable without modification so that it is suitable for our adaptive synchronization problem.
A related problem is Sample Rate Conversion (SRC) which is often considered in digital front ends [10; 9]. In SRC, a digital signal has to be converted into another digital signal but with a different sampling frequency. Caution must be exercised to avoid aliasing in the operation. In this situation, polynomial interpolators are usually disqualified because they do not provide enough anti-aliasing. In the last decade, following [27], many solutions have been developed for the synthesis of adjustable fractional delay filters with larger bands and better anti-aliasing capabilities [26; 13; 12; 30; 32].
It is worth mentioning that adjustable fractional delay filters can also be obtained using programmable allpass Infinite Impulse Response (IIR) filters [16; 31]. However, such filters are more sensitive to quantization, transients may occur when changing coefficients, and synthesis is complicated by the stability issues.
In our application, interpolation operates directly on the digital input signal, without any rate change. Aliasing can still occur due to the sampling operation of the output, which is needed for our feedback loop. However, the system specifications, in particular the power limitation in adjacent channels, severely constrain the design and limit the images of the original band-limited spectrum. Furthermore, since the over-sampling ratio T/T_s is typically greater than 5, an anti-aliasing filter can be easily designed.
The over-sampling ratio being high is an important factor since it allows the use of very low-order interpolators. Indeed, the frequency response of the corresponding filters is almost flat in magnitude and linear in phase in the band of interest. This is illustrated in section 4.1. Then, in section 4.2, we examine and compare the performances of the different interpolators in terms of interpolation error and EVM at the output of the transmitter.

4.1 The interpolators and their frequency responses

In this section, we choose to compare four interpolators [1, Chapter 25]: Linear, Bessel, third and fifth order Lagrange interpolators. We first give the expressions of these interpolators and then compare their frequency responses.
The first interpolator, the forward linear interpolator, is the simplest, and is given by:

$$x(m,\tau) = x(m) + \tau\,(x(m+1) - x(m)) \tag{21}$$

where $x(m,\tau)$ represents the interpolated value of the input value at the time $(m+\tau)T_s$.
The second interpolator studied is the third order Lagrange

interpolator:

$$x(m,\tau) = x(m) + \tau\,(x(m) - x(m-1)) + \frac{\tau(\tau+1)}{2}\,(x(m+1) - 2x(m) + x(m-1)). \quad (22)$$

This interpolation uses three successive points. The first part of the expression is similar to the linear interpolator and the second part is a correction term calculated using the point before and after the central one.
The third interpolator we looked at is the Bessel central difference interpolator, fourth order, described as follow:

$$x(m,\tau) = x(m) + \tau\,(x(m+1) - x(m)) + \frac{\tau(\tau-1)}{4}\,(x(m+2) - x(m+1) - x(m) + x(m-1)).$$

This interpolation uses four successive points and is also similar to the linear interpolation with an additional correction term. Compared to the third order Lagrange formulation, the correction term is not symmetrical.
The last interpolator is the fifth order Lagrange interpolator:

$$x(m,\tau) = \frac{(\tau^2-1)(\tau-2)\tau}{24}x(m-2) - \frac{(\tau^2-4)(\tau-1)\tau}{6}x(m-1) + \frac{(\tau^2-1)(\tau^2-4)}{4}x(m) - \frac{(\tau^2-4)(\tau+1)\tau}{6}x(m+1) + \frac{(\tau^2-1)(\tau+2)\tau}{24}x(m+2). \quad (23)$$

These different interpolators can be clearly viewed as FIR filters, where impulse responses can be deduced from the above equations. Therefore, it is certainly interesting to compare their frequency responses to the frequency response of a pure delay. This comparison is shown in Fig. 8 for the magnitude and in Fig. 9 for the phase. With the exception of the linear interpolator, it appears that the interpolators have interesting performances for normalized frequencies below 0.2 (over-sampling ratio greater than 5). While the two Lagrange interpolators show the flattest magnitude, the Bessel interpolator exhibits a better phase linearity. This information will be completed by other measures of performances, namely interpolation error and EVM at the output of the transmitter.

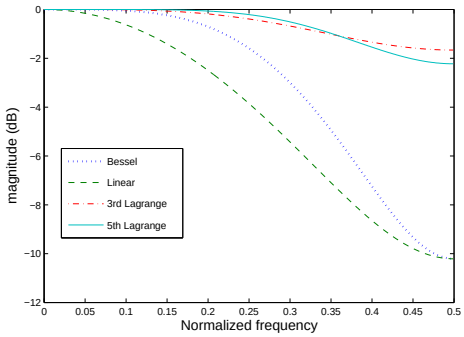


Fig. 8 Magnitudes of the frequency responses of the four interpolators.

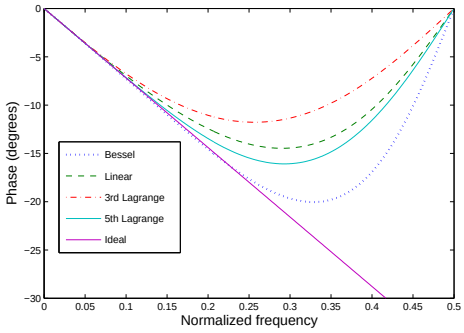


Fig. 9 Phases of the frequency responses of the four interpolators.

4.2 Comparison of performances

In order to evaluate interpolation errors, we compare an original signal to a reconstructed one. This comparison is realized on the envelope of the OFDM modulated signal previously introduced. Starting with an original signal with 300 samples per time symbol, we subsample it by a factor of 100. This gives an over-sampling ratio of 3. The signal is then reconstructed by interpolating the missing 200 values between $x(m-1)$ and $x(m+1)$, and the interpolated values at a delay Δ from the sample, are compared to the value of the original signal. Performances of the different interpolators are presented in Fig. 10 in terms of the rms quadratic error, given in percent. These experimental results are confirmed by the theoretical analysis in section 4.4.

Let us note T_s the sampling period and call E_o the original signal, E_i the interpolated signal. The quadratic error (rms) is expressed as:

$$Err(\Delta) = 100\sqrt{\frac{\sum_k (E_o(kT_s - \Delta) - E_i(kT_s - \Delta))^2}{\sum_k (E_o^2(kT_s - \Delta)}} \quad (24)$$

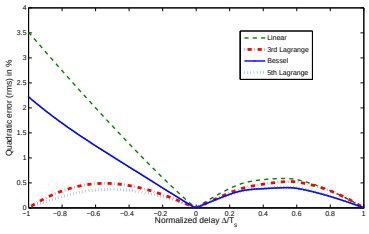


Fig. 10 Comparison of the normalized quadratic error (rms) between the four interpolators.

When the interpolators are used between $x(m)$ and $x(m+1)$, their rms quadratic errors are similar with a maximum error in the middle of the two samples. The Bessel and 5th order Lagrange interpolators achieve similar results with a maximum error of about 0.4%, while the two other interpolators present less than 0.6%. However, a problem arises when the interpolator is used in the range of $x(m-1)$ and $x(m)$. The linear interpolator is obviously not up to par with a 3.5% of error followed by the Bessel with 2.5%. The asymmetry can degrade the quadratic error performances. During iterations, positive as well as negative values of τ may indeed appear. The best solution remains certainly the two Lagrange formulas which are quasi symmetrical and more appropriate to our problem. The order of the interpolator can also make a difference in terms of the complexity of the implementation. Using HPADS, the interpolators can be validated with the study of the EVM (computed for the OFDM after FFT demodulation) and the output spectrum. The simulation is realized differently than with the previous quadratic error evaluation. Here the signal sampled at $T/3$ is delayed or advanced by τ taken between ± 1 and resynchronized using the interpolator. This is a complementary analysis to the previous analysis and is better suited to standard transmitter analysis. Fig. 11 shows the EVM at the output of the transmitter and presents the same profile as the error quadratic curves in Fig. 10.

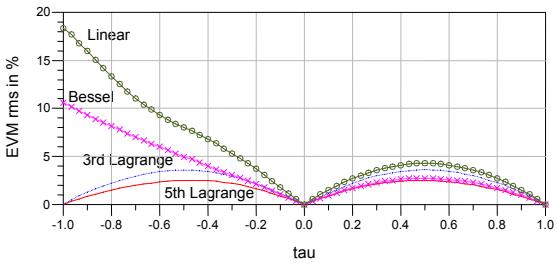


Fig. 11 Comparison of EVM values for the different interpolators.

The main problem is the relatively high values of the EVM. Values reach a maximum of 2% for the 5th order Lagrange interpolator and 3.5% for the 3rd order Lagrange interpolator. The latter is not acceptable for our application. We then need to sample the signal to a higher rate.
Fig. 12 presents the EVM values when changing the sampling period of the signal from $T/3$ to $T/9$ (odd values are here preferred to facilitate the demodulation of the OFDM signal in simulation). The x axis of the curve is normalized to symbol duration. The difference between samplings at $T/3$ and $T/9$ is dramatic, going from 2% to 0.1%.
The sampling rate not only has an impact on the EVM value, it also affects the output spectrum.

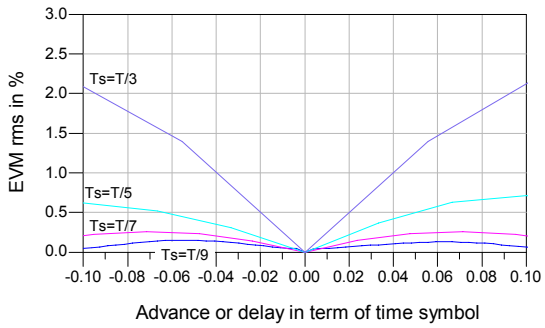


Fig. 12 EVM values for the 5th order Lagrange interpolator with different sampling periods.

4.3 Implementation using the Farrow structure

A polynomial interpolator can efficiently be implemented using a Farrow structure [8] or its extensions [27; 13; 5]. We only describe here the original Farrow structure but other possibilities will deserve further investigation. The Farrow structure relies on the parallelization of N FIR filters which can be expressed as

$$y_i(m) = \sum_{k=-M}^{M-1} h_i(k) x(m-k). \tag{25}$$

The output of these filters are combined so that the output signal can be expressed as

$$y(m,\tau) = \sum_{i=0}^{N-1} \tau^i y_i(m) = \sum_{i=0}^{N-1} \tau^i \sum_{k=-M}^{M-1} h_i(k)\, x(m-k) \tag{26}$$

For instance, rewriting the equations of the 5th order Lagrange interpolator (23) gives

$$\begin{aligned}
y_0(m) &= \tfrac{5}{4}x(m), \\
y_1(m) &= -\tfrac{2}{24}(m+2) + \tfrac{4}{6}x(m+1) \\
&\quad -\tfrac{4}{6}x(m-1) - \tfrac{2}{24}(m-2) \\
y_2(m) &= -\tfrac{1}{24}(m+2) + \tfrac{4}{6}x(m+1) - \tfrac{5}{4}x(m) \\
&\quad +\tfrac{4}{6}x(m-1) - \tfrac{1}{24}(m-2) \\
y_3(m) &= \tfrac{2}{24}(m+2) - \tfrac{1}{6}x(m+1) + \tfrac{1}{6}x(m-1) - \tfrac{2}{24}(m-2) \\
y_4(m) &= \tfrac{1}{24}(m+2) - \tfrac{1}{6}x(m+1) + \tfrac{1}{4}x(m) - \tfrac{1}{6}x(m-1) \\
&\quad +\tfrac{1}{24}(m-2)
\end{aligned}$$

and can be implemented on HPADS as presented in Fig. 13.

For the 5th order Lagrange filter, this implementation uses 5 multiplications, 20 additions and 15 arithmetic divisions. As for the 3rd order Lagrange interpolator, the implementation leads to only 2 multiplications, 5 additions and simpler divisions that only requires to divide by 2. This can certainly be an argument for the choice of the interpolator, not only in terms of performances but also in terms of simplicity of implementation.

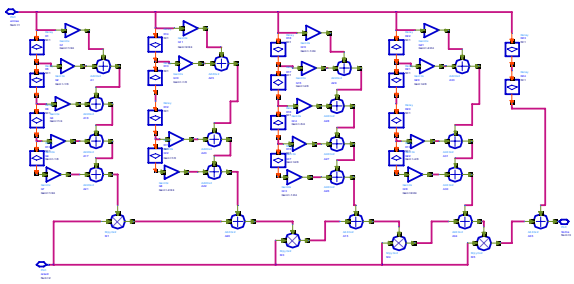


Fig. 13 Farrow structure for the 5th order Lagrange interpolator.

4.4 On the interpolation noise

If the interpolator were perfect, the algorithm, in the absence of observation noise, would find the *exact* solution. However, as illustrated in Figs. 10, 11 and 12, the system is characterized by an internal "self-noise" associated with the interpolator. Indeed, for equally spaced points $t_0, t_1, \ldots t_n$, the remainder of the Lagrange interpolation polynomial of degree n for estimating $x(t_0+\tau T)$ is [28]

$$n_i(t_0+\tau T) = T^{n+1}\omega(\tau)\frac{x^{(n+1)}(\xi)}{(n+1)!} \qquad (27)$$

with $\omega(\tau)=\tau(\tau-1)\ldots(\tau-n)$, and where ξ is an unknown point in the interval $[t_0,t_n]$. Thus, we can evaluate the variance $\sigma_i^2(\tau)=\mathrm{E}\left[n_i(t_0+\tau T)^2\right]$ of this interpolation noise $n_i(t)$ and obtain

$$\mathrm{E}\left[n_i(t_0+\tau T)^2\right] = (-1)^{n+1}\frac{T^{2(n+1)}}{(n+1)!}|\omega(\tau)|^2 R_{xx}^{(2(n+1))}(0) \qquad (28)$$

using the relation $\mathrm{E}\left[|x^{(n+1)}(t)|^2\right] = (-1)^{n+1}R_{xx}^{(2(n+1))}(0)$.
As a result, the variance of the interpolation noise is non-stationary since it depends on the interpolation point $t_0+\tau T$. With this formula, it also appears that the higher the sampling rate and the higher the interpolation order, the lower the variance. To illustrate this, Fig. 14 compares the theoretical formula (28) to simulation results (see also the results in Fig. 10).

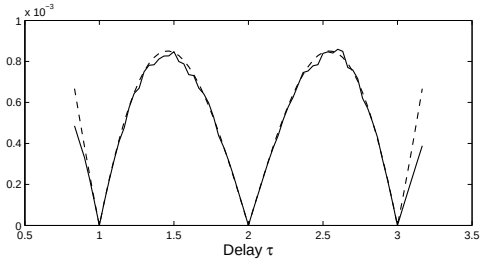


Fig. 14 Standard deviation of the interpolation noise for a Lagrange 5^{th} order interpolator. Theoretical formula (28) (plain line) is compared to simulation results (dashed line).

5 Analysis of the algorithm and results

Typical results for delays $0.18T$ and $0.47T$ for envelope and phase respectively, are shown in Fig. 15. The measured EVM after convergence is only 0.7% compared to 14% without correction. The resulting spectrum is reported in Fig. 16. It shows very interesting performances: the spectrum is improved by 30dB compared to the uncorrected case. The remaining noise floor at -50dBc in 200kHz bandwidth corresponds to the interpolation errors. Surprisingly, it appears that the spectrum obtained with the values at the output of the correction algorithm is slightly better than the spectum obtained with the true values of delays. This is commented in §5.2.

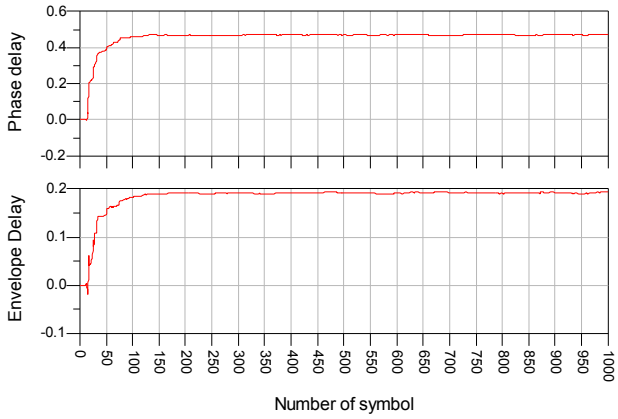


Fig. 15 Convergence of μ_1 and μ_2 to the true delays $\Delta_1 = 0.47T$ and $\Delta_2 = 0.18T$.

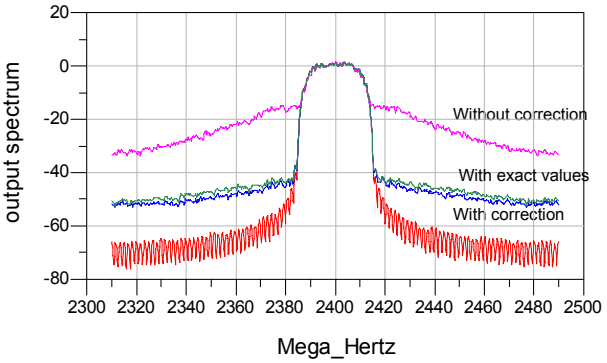


Fig. 16 Comparison of ideal, uncorrected and corrected spectra, in the case of delays $\Delta_1 = 0.47T$ and $\Delta_2 = 0.18T$.

However interesting these results are, it is important to examine the role of the interpolator, the impact of the values

of the adaptation step and to study figures of merit such as the settling time (convergence speed), the bias and variance of results as well as the overall EVM. We evaluated these different points by Monte-Carlo experiments for a varying adaptation step γ and for all delays less than T. In the study, we took the same adaptation $\gamma = \gamma_1 = \gamma_2$ step for both adaptations. Furthermore, the signal power was normalized to one in order to be independent of signal scales.
To analyse these different characteristics, we use a simplified 'toy' model that is simpler to analyse than our original problem. Let $z(t) = x(t-\Delta)$ be an observed, delayed version of an original signal $x(t)$. In order to identify and correct the delay, we adopt the following recursion

$$\mu(n+1) = \mu(n) + \gamma \left.\frac{dx(u)}{du}\right|_{u=t-\Delta+\mu(n)} [x(t) - \tilde{x}(t-\Delta+\mu(n))] \tag{29}$$

with $\tilde{x}(t-\Delta+\mu(n))$ the interpolated value at time $(t-\Delta+\mu(n))$.

5.1 Settling time

The convergence speed is an important issue for the practical use of such algorithm. With $\Delta\mu(n) = (\mu(n) - \Delta)$ small, we have

$$x(t) - \tilde{x}(t+\Delta\mu(n)) \simeq -\Delta\mu(n)\dot{x}(t) + n_i(t), \tag{30}$$

where $\dot{x}(t)$ is the derivative of process $x(t)$ and where $n_i(t)$ represents the interpolation noise. Then,

$$\Delta\mu(n+1) = \Delta\mu(n) - \gamma\Delta\mu(n)\dot{x}(t+\Delta\mu(n))\dot{x}(t) + \gamma\dot{x}(t+\Delta\mu(n))n_i(t) \tag{31}$$

Therefore, the mean trajectory is

$$\mathrm{E}\left[\Delta\mu(n+1)\right] = \mathrm{E}\left[\Delta\mu(n)\right]\Big(1 - \gamma R_{\dot{x}\dot{x}}(\Delta\mu(n))\Big), \tag{32}$$

with $\mathrm{E}\left[\dot{x}n_i\right] = 0$. Considering that $R_{\dot{x}\dot{x}} = -\ddot{R}_{xx}$, and since we supposed $\Delta\mu(n)$ small, so that $\ddot{R}_{xx}(\Delta\mu(n)) \simeq \ddot{R}_{xx}(0)$, the previous equation can be solved recursively and we get

$$\mathrm{E}\left[\Delta\mu(n+1)\right] \simeq \Big(1 + \gamma\ddot{R}_{xx}(0)\Big)^n \mathrm{E}\left[\Delta\mu(0)\right], \tag{33}$$

with $\ddot{R}_{xx}(0) < 0$. This means that $\mu(n)$ converges exponentially to Δ, with a time constant $t_c = -1/\log\left(1 + \gamma\ddot{R}_{xx}(0)\right) \simeq 1/\left(\gamma\ddot{R}_{xx}(0)\right)$. This is illustrated in Fig. 17, where are presented the settling times for the algorithm measured as the rise from 0 to 95% of the final value, $t_s = 3t_c$. This figure shows that the convergence speed is clearly independent of the interpolator and the simulation results are in line with the above development.

We also examined the variation of the convergence speed with respect to the values Δ_1, Δ_2. The results reported in Fig. 18 clearly indicates that the convergence speed is independent of the final values.

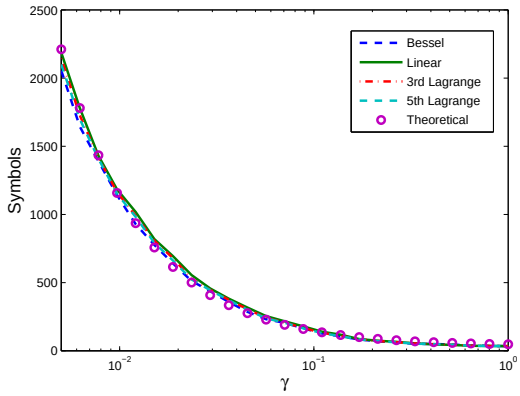


Fig. 17 Settling times for the different interpolators, as a function of the adaptation step, for $\Delta_1 = 0.36T$ and $\Delta_2 = 0.12T$.

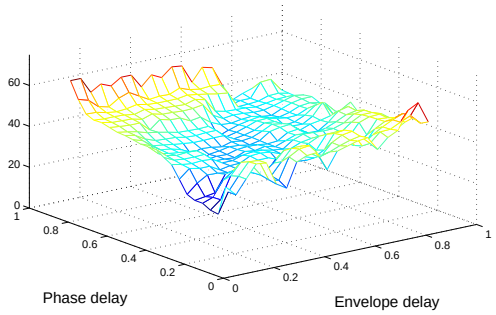


Fig. 18 Settling time for all possible delays (less that T), with $\gamma = 0.2$.

5.2 Bias

A careful examination of the results shows that both envelope and phase are often affected by a small, but undeniable, bias. In addition, we have observed in practice that the quadratic error associated with the biased estimates is lower than the error obtained with the true values. Similarly, it appears that the spectrum with identified values is slightly better than the spectrum associated with the true values. This can be explained as follow. The error $e(t)$ is

$$e(t) = x(t) - \tilde{x}(t-\Delta+\mu(n)) = \Big(x(t) - x(t-\Delta+\mu(n))\Big) + n_i(t), \tag{34}$$

so that its variance is

$$\sigma_e^2 = \mathrm{E}\left[e(t)^2\right] = \mathrm{E}\left[\Big(x(t) - x(t-\Delta+\mu(n))\Big)\right] + \mathrm{E}\left[n_i(t)^2\right], \tag{35}$$

that is

$$\sigma_e^2 = \mathrm{E}\left[e(t)^2\right] = 2\left(R_{xx}(0) - R_{xx}(\mu(n)-\Delta)\right) + \sigma_i(\mu(n)), \tag{36}$$

where $\sigma_i(\mu(n)) = \mathrm{E}\left[n_i(t)^2\right]$ is the variance of the interpolation noise that depends on the value of the advance $\mu(n)$, see the discussion in §4.4. The purpose of the algorithm is

to find a value μ minimizing σ_e^2. The first term clearly decreases when $\mu(n) \to \Delta$ while the second term may increase. Therefore, the procedure will find the best values that minimize the sum of the two terms, realizing a trade-off between bias and variance of the interpolation noise. The solution is theoretically given by

$$\dot{R}_{xx}(\mu(n)-\Delta) = -\frac{d\sigma_i^2(\mu(n))}{d\mu(n)}. \tag{37}$$

Fig. 19 presents the bias measured for the envelope and phase component with respect to the adaptation parameter. We find that the bias still exist and do not depend on the adaptation parameter γ as it was indicated by the analysis. The bias is a function of the interpolator type and order.

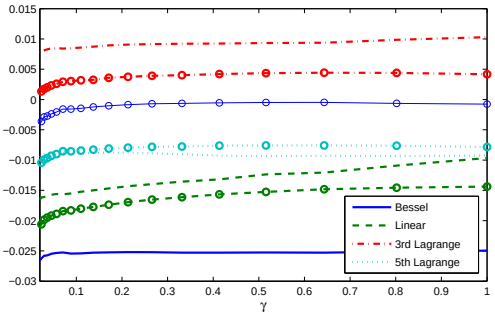


Fig. 19 Bias $\mu - \Delta$ for the different interpolators, as a function of the adaptation parameter γ. Phase (indicated by $-\circ-$) and envelope bias are almost constant with respect to γ.

5.3 Variance

Variations after convergence are also an important issue. These variations can be modeled as an additive noise on the estimates. With our toy model, we obtained expression (31) for the trajectory of the algorithm. This expression is only valid, on average, if $\Delta\mu(n+1)$ has a zero mean. Thus, taking now $\Delta\mu(n+1) = \mu(n+1) - \bar{\mu}$, where $\bar{\mu}$ is the value at convergence and rearranging the terms, we get

$$\begin{aligned}\Delta\mu(n+1) &= \Delta\mu(n)\Big(1-\gamma\dot{x}(t+\Delta\mu(n))\dot{x}(t)\Big) \\ &\quad + \gamma\dot{x}(t+\Delta\mu(n))n_i(t). = A_n\Delta\mu(n) + B_n n_i(t)\end{aligned} \tag{38}$$

Taking the square and the expectation and making the, clearly false, hypothesis that A_n and $\Delta\mu(n)$, B_n and $n_i(t)$ are uncorrelated, we obtain

$$\sigma_\mu^2(n+1) = \mathrm{E}\left[A_n^2\right]\sigma_\mu^2(n) + \mathrm{E}\left[B_n^2\right]\sigma_i^2 \tag{39}$$

with $\sigma_\mu^2(n)$ the variance of μ at step n. We also find $\mathrm{E}\left[A_n^2\right] \simeq 1 + 2\gamma\ddot{R}_{xx}(0) + \gamma^2\mathrm{E}\left[\dot{x}(t)^4\right]$ (with $\mu - \Delta \simeq 0$), and $\mathrm{E}\left[B_n^2\right] = -\gamma^2\ddot{R}_{xx}(0)$. At convergence, we finally obtain

$$\sigma_\mu^2 = \frac{\mathrm{E}\left[B_n^2\right]\sigma_i^2}{1-\mathrm{E}\left[A_n^2\right]} = \frac{\gamma\sigma_i^2}{2+\gamma\mathrm{E}\left[\dot{x}(t)^4\right]/\ddot{R}_{xx}(0)}. \tag{40}$$

This expression shows that the variance of the estimates increases linearly with γ and that it is only associated to the 'self-noise' in the system. Despite using approximations in the derivation, the simulation results are once more in line with this formula. Fig. 20 gives the variance of the estimates of the envelope and phase delays as a function of γ. The linear dependence can be clearly seen.

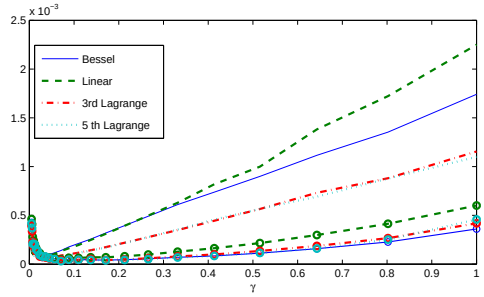


Fig. 20 Variance σ_μ^2 of delays estimates, $\Delta_1 = 0.36T, \Delta_2 = 0.12T$, for the different interpolators, as a function of the adaptation parameter γ. Phase (indicated by $-\circ-$) and envelope variances increase linearly with γ.

5.4 EVM - Quadratic error

The EVM is a crucial parameter in digital communications. As we will observe, it sums up some of the other properties. In the case of a single carrier modulation, the EVM is defined by $\mathrm{E}\left[|Z_e - Z_o|^2\right]/\mathrm{E}\left[|Z_o|^2\right]$ where $Z_e = \rho_e e^{j\phi_e}$ and $Z_o = \rho_o e^{j\phi_o}$ stand for the complex envelope of the emitted signal and the original signal respectively. Let us insist on the fact that the situation is more complicated for multi-carrier modulations. In our setting, the expression becomes

$$\mathrm{EVM} = \frac{\mathrm{E}\left[\rho_o^2 + \rho_e^2 - 2\rho_o\rho_e\cos(\phi_o - \phi_e)\right]}{\mathrm{E}\left[\rho_o^2\right]}. \tag{41}$$

Therefore, if $\Delta\phi = \phi_o - \phi_e$ is small, $\cos\Delta\phi \simeq 1 - \Delta\phi^2/2$, the expression becomes

$$\mathrm{EVM} = \frac{\mathrm{E}\left[(\rho_o-\rho_e)^2 + \rho_o\rho_e\Delta\phi^2\right]}{\mathrm{E}\left[\rho_o^2\right]} \simeq \frac{\mathrm{E}\left[\Delta\rho^2 + \rho_o^2\Delta\phi^2\right]}{\mathrm{E}\left[\rho_o^2\right]} \tag{42}$$

or

$$\mathrm{EVM} \simeq \frac{\mathrm{E}\left[\Delta\rho^2\right]}{\mathrm{E}\left[\rho_o^2\right]} + \mathrm{E}\left[\Delta\phi^2\right] \tag{43}$$

assuming that ρ_o and $\Delta\phi$ are independent. As a result, the EVM is a function of the quantities we previously studied.

We already established that the estimates are biased and that their variances, which are proportional to interpolation noise, increase linearly with γ. Therefore, the EVM behaves as

$$\mathrm{EVM} \simeq \frac{\gamma}{2}\left[\frac{\sigma_\rho^2}{\mathrm{E}\left[\rho_o^2\right]}+\sigma_\phi^2\right]+\left[\frac{B_\rho^2}{\mathrm{E}\left[\rho_o^2\right]}+B_\phi^2\right], \qquad (44)$$

where σ_ρ^2 and σ_ϕ^2 are the variances of interpolation noise for the envelope and phase, and B_ρ and B_ϕ the bias terms. Since $\sqrt{1+\alpha x} \simeq 1+\alpha x/2$, the EVM rms may also show the same behaviour. This is the case in Fig. 21 where we clearly see a linear slope and an offset value. It also shows that the Lagrange and Bessel interpolators have the best performances.

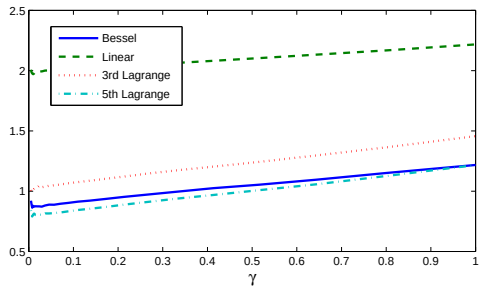


Fig. 21 EVM rms (in %) at the output of the algorithm, for delays $\Delta_1 = 0.36T, \Delta_2 = 0.12T$ as a function of the adaptation parameter γ.

It is also interesting to look at the EVM with respect to all possible delays and with a fixed γ. Fig. 22 presents these results for $\gamma = 0.05$. We can observe a 'modulation' which is related to the variable variance of the interpolation noise with respect to the delay (see Fig. 14). Here, as in previous examples, we took $T/Ts = 5$ and indeed notice error minima at interpolation nodes.

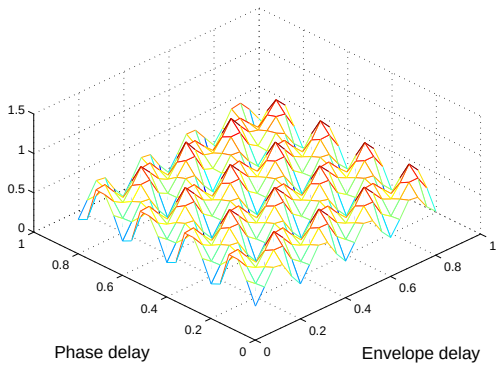


Fig. 22 EVM at output of the algorithm with $\gamma = 0.05$, for all possible delays. The 'modulation' is related to the non-stationary interpolation noise that vanishes at interpolation nodes.

Table 3 Settling times of the algorithm and EVM rms (in %) for the output, for different interpolators and several values of the adaptation step γ.

	Settling time		EVM rms %	
γ		3rd Lag	Bessel	5th Lag
0.05	228	1.045	0.89	0.815
0.1	127	1.07	0.9	0.84
0.5	45	1.24	1.05	1.009

5.5 Summary of the results

In the previous developments we have examined the behaviour of our algorithm according to the choice of interpolators and adaptation steps. Using specific examples, we have demonstrated a vast improvement can be achieved in terms of EVM and output spectrum, as shown in figures 15 and 16. Furthermore, using an approximate 'toy' model of the algorithm, we have shown theoretically and checked numerically, that

- the settling time decreases exponentially with the adaptation step (Fig. 17),
- a small estimation bias exists that does not depend on the adaptation parameter but rather of interpolation order (Fig. 19). The presence of this bias is explained as a way to reduce the whole quadratic error,
- the variance (40) behaves approximately as a linear function of the adaptation parameter (Fig. 20),
- the EVM (44) increases linearly with the adaptation step γ (Fig. 21).

As a consequence, a trade-off has to be made between the two important parameters that are the settling time and the EVM. The choice of the order of the interpolation is also important with respect to performances and implementation costs. The main values for the bias and variance can be found in Table 2 for different interpolators and several values of the adaptation step γ. The results for the settling time and the EVM are reported in Table 3 for the same conditions. In order to ensure independence in signal scales, the signal power was normalized to one.

6 Conclusion

In this paper, we proposed and analysed a new synchronization algorithm tailored for an EER architecture. Performances are indeed severely degraded in case of delay mismatch between envelope and phase paths. The proposed algorithm is based on an adaptive LMS structure. We studied the influence of the adaptation on the quadratic error, EVM and output spectrum. The algorithm uses an interpolation procedure that is characterized in terms of performances and in terms of implementation. We also examined the influence of the sampling rate. These studies demonstrated the strong interest of such a mandatory correction algorithm and the enhancement of transmitter performances.

Table 2 Bias $(\mu - \Delta)$ and variance σ_μ^2 of the estimates of delays, for different interpolators and several values of the adaptation step γ.

			Bias $\times 10^3$			Variance $\times 10^5$	
γ		3rd Lag	Bessel	5th Lag	3rd Lag	Bessel	5th Lag
0.05	ρ	-15.5	8	-9	6.2	10	5.8
0.05	ϕ	-19	2	-9	5.8	6.59	5.75
0.1	ρ	-15	8	-8	11.5	21	11.5
0.1	ϕ	-28	4	-8	4.5	4.6	4.5
0.5	ρ	-14	8	-7	56.5	89	56
0.5	ϕ	-17	4	-9	13	11.5	11.5

Regarding the algorithm, a few points can be further investigated: adopting different adaptation steps for the two recursions, decrasing the number of adaptation steps and alternating the two recursions. Higher system view point has to be taken into account in order to define the best use of the procedure: full tracking of delays or training periods. A for the implementation, modified and optimized Farrow structures should be considered.

This solution has drawbacks. For instance, it requires I/Q demodulation and uses a Cordic processor to split the envelope and phase of the PA output signal (derivative computation). We are currently working on these issues to improve our system.

Other concerns are the behaviour of the system with further mismatches such as a complex gain in the feedback loop, non-linear distortions and quantization noise. Preliminary results show that the algorithm is robust against the gain and noise mismatch, in the sense that the optimum solution remains unchanged. Evidently further mismatches imply a degradation of performances so future work will include attempts to also correct these mismatches with the same feedback loop.

Acknowledgements The authors are indebted to the anonymous referees for their careful reading of the manuscript as well as their valuable comments and suggestions that helped improve the presentation of this paper. Thanks are extended to S. Trudel, P. J. Duriez and L. Garbell for their invaluable and friendly proofreading of the manuscript.

References

1. Abramowitz M, Stegun IA (1965) Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables. Dover Publications
2. Baudoin G, Berland C, Villegas M, Diet A (2003) Influence of time and processing mismatches between phase and envelope signals in linearization systems using envelope elimination and restoration, application to hiperlan2. In: Microwave Symposium Digest, 2003 IEEE MTT-S International, vol 3, pp 2149–2152
3. Bercher JF, Berland C (2006) Envelope/phase delays correction in an EER radio architecture. In: Proceedings of the 13th IEEE International Conference on Electronics, Circuits, and Systems, ICECS2006, pp 443–446
4. Bercher JF, Diet A, Berland C, Baudoin G, Villegas M (2004) Monte-carlo estimation of time mismatch effect in an OFDM EER architecture. In: Proceedings of the 2004 IEEE Radio and Wireless Conference, pp 283–286
5. Candan C (2007) An efficient filtering structure for Lagrange interpolation. IEEE Signal Processing Letters 14:17–19
6. Crochiere R, Rabiner L (1975) Optimum fir digital filter implementations for decimation, interpolation, and narrow-band filtering. IEEE Transactions on Acoustics, Speech, and Signal Processing 23(5):444–456
7. Dempster A, Murphy N (2000) Efficient interpolators and filter banks using multiplier blocks. IEEE Transactions on Signal Processing 48:257–261
8. Farrow CW (1988) A continuous variable digital delay element. In: Proc. IEEE Int. Symp. Circuits Systems, pp 2641–2645
9. Hentschel T (2002) Sample Rate Conversion in Software Configurable Radios. Artech House Publishers
10. Hentschel T, Fettweis G (2002) The Digital Front-End – Bridge Between RF-and Baseband-Processing. In: Software Defined Radio Enabling Technologies, John Wiley & Sons Ltd, pp 151–198
11. Jau JK, Horng TS (2001) Linear interpolation scheme for compensation of path delay difference in an envelope elimination and restoration transmitter. In: Proceedings of Asia-Pacific Microwave Conference, 2001. APMC 2001, vol 3, pp 1072–1075 vol.3
12. Johansson H, Gustafsson O (2005) Linear-phase FIR interpolation, decimation, and *m*th-band filters utilizing the Farrow structure. IEEE Transactions on Circuits and Systems I 52:2197–2207
13. Johansson H, Lowenborg P (2003) On the design of adjustable fractional delay FIR filters. IEEE Transactions on Circuits and Systems II 50:164–169
14. Kahn L (1952) Single-sideband transmission by envelope elimination and restoration. Proceedings of the IRE 40(7):803–806
15. Laakso TI, Valimaki V, Karjalainen M, Laine UK (1996) Splitting the unit delay. IEEE Signal Processing Magazine 13:30–60
16. Makundi M, Laakso T, Valimaki V (2001) Efficient tunable IIR and allpass filter structures. Electronics Letters 37:344–345
17. Mártires J, Borg C, Larsen T (2006) Differential delay equalization in a Kahn EER transmitter. In: ST Mobile & Wireless Communications Summit, nr. 15th, Myconos, Grækenland
18. Morgan P (2006) Highly integrated transceiver enables high-volume production of GSM/EDGE handsets. RFdesign pp 36–42
19. Oetken G (1979) A new approach for the design of digital interpolating filters. IEEE Transactions on Acoustics, Speech, and Signal Processing 27(6):637–643
20. Raab F (1977) Idealized operation of the class e tuned power amplifier. Circuits and Systems, IEEE Transactions on 24:725–735
21. Sander WB, Schell SV, Sander BL (2003) Polar modulator for multi-mode cell phones. Proceedings of the IEEE 2003 Custom Integrated Circuits Conference pp 439–445
22. Sarigeorgidis K, Rabaey J (2004) Ultra low power cordic processor for wireless communication algorithms. Journal of VLSI Signal processing 38:115–130
23. Staszewski R, Wallberg J, Rezeq S, Hung CM, Eliezer O, Vemulapalli S, Fernando C, Maggio K, Staszewski R, Barton N, Lee MC, Cruise P, Entezari M, Muhammad K, Leipold D (2005) All-digital PLL and transmitter for mobile phones. IEEE Journal of Solid-State Circuits 40(12):2469–2482
24. Strasser G, Lindner B, Maurer L, Hueber G, Springer A (2006) On the spectral regrowth in polar transmitters. In: Microwave Sympo-

sium Digest, 2006. IEEE MTT-S International, pp 781–784
25. Valimaki V (1995) A new filter implementation strategy for Lagrange interpolation. In: Proceedings of the 1995 IEEE International Symposium on Circuits and Systems, ISCAS '95, vol 1, pp 361–364 vol.1
26. Valimaki V, Laakso TI (2000) Principles of fractional delay filters. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'00 6
27. Vesma J, Saramaki T (1997) Optimization and efficient implementation of FIR filters with adjustable fractional delay. In: Proceedings of 1997 IEEE International Symposium on Circuits and Systems, ISCAS '97, vol 4, pp 2256–2259 vol.4
28. Volkov EA (1986) Numerical Methods. Mir Publisher Moscow
29. Whittaker JM (1935) Interpolatory Function Theory. Cambridge Univ. Press, Cambridge, England
30. Yli-Kaakinen, Saramaki (2006) Multiplication-free polynomial-based FIR filters with an adjustable fractional delay. Circuits, Systems, and Signal Processing 25:265–294
31. Yli-Kaakinen J, Saramaki T (2004) An algorithm for the optimization of adjustable fractional-delay all-pass filters. In: Proceedings of the 2004 International Symposium on Circuits and Systems ISCAS'04, vol 3, pp III–153–6 Vol.3
32. Yli-Kaakinen J, Saramaki T (2007) A simplified structure for FIR filters with an adjustable fractional delay. IEEE International Symposium on Circuits and Systems, 2007 ISCAS 2007 pp 3439–3442