A Strong Anti-Jamming Algorithm Based on FPGA for Estimating Loop Delay in Digital Predistortion System

Feng-Jun Li, Jing-Fu Bao, Hong-Yun Huang, and Shao-Chun Jin

Abstract—At present what are the key points focused in the research of loop-delay estimation for the digital predistorter in the radio frequency (RF) power amplifier system is reducing its complexity of engineering realization and improving anti-jamming ability and computational speed. Besides, opening up its application scope should be contained. For these targets, a novel method including integer loop delay estimation and fractional part is proposed. The integer part applies amplitude-difference summation function and the fractional one adopts the method of finite impulse response (FIR) linear interpolation. The algorithm finds wide applications. What is more, strong anti-jamming ability and low complexity are also its merits. Simulation results support the above opinion. Digital predistortion (DPD) system based on this algorithm achieves good performance.

Index Terms—Baseband, digital predistortion, linear interpolation, loop delay estimation, power amplifier.

1. Introduction

It is well known that the new generation wireless telecommunication system applies various non-constant envelope modulation technologies. For catering to the new requirements, high peak to average power ratio (PAPR), wide bandwidth, high efficiency, and power amplifier linearization have become topics. It has already been noted that digital predistortion (DPD) has been the most active area on the grounds of its low complexity, good linearization performance, and highly cost-effective[1]–[4].

Generally speaking, loop delay could be defined as the time delay between the feedback signal $x_f(n)$ and input one $x_p(n)$ indicated in Fig. 1. However, the DPD is so sensitive to the delay alignment between input and feedback waveforms that loop delay misalignment will degrade its performance. So loop delay precise estimation is one of the primary tasks for the design of DPD system[5][6]. Last but not least, we should pay close attention to the anti-jamming ability and computational speed of loop delay estimator in engineering practice.

With regard to the loop delay estimation algorithms proposed by previous works, they could be summarized as follows. First of all, the algorithm[7] adopted by Nagata is known to be lack of accuracy; the anti-jamming ability of correlation function method[3] presented by Li could be strengthened; the schemes in [3] and [8] using least mean square (LMS) and recursive least square (RLS) respectively achieve the desired accuracy at the cost of increasing the complexity of the whole system; finally, the common merits of interpolating method used in [9] and the divine dichotomy applied in [10] are easy timing control and wide applications, but specific to their computational speed and anti-jamming ability we can make some progress.

In this paper, we investigate the loop delay divided into two parts. The first one is integer loop delay estimation which adopts amplitude-difference summation function. And the second one is the fractional loop delay which applies FIR linear interpolation. They are introduced in detail in Section 2. And Section 3 reveals experimental results. The conclusion is drawn in Section 4.

2. Proposed Algorithm

2.1 Target Statement

As it was noted previously, the error of loop delay estimation is one of the significant elements which DPD designers should take into account. But its upper limit probably differs somewhat according to the sampling interval (SI). Fig. 2 shows the impact of different estimation errors on power spectral density (PSD) of PA output. 1 SI and 0.5 SI estimation errors degrade PSD performance by corresponding levels. Furthermore, their floor noises are also exacerbated compared with the ideal one (0 SI). However, the 0.03 SI curve almost goes with the ideal one. In other words, digital predistorter adopted in this paper is insensitive to the error less than 0.03 SI.
Consequently, provided that estimation accuracy requirement has been met, we will focus on simplifying the realization difficulty of algorithm and enhancing its anti-jamming ability. For these targets, a loop delay estimator is designed as the architecture illustrated by Fig. 3, including three parts: the integer and fractional loop delay estimation, coupled with integer loop delay compensation. \( x_{in}(n) \) is the signal \( x_n(n) \) adjusted by integer loop delay \( n_i \). The time delay \( n_{total} \) can be divided into two parts: integer section \( n_i \) and fractional one \( n_f \). The expression is presented as

\[
n_{total} = n_i + n_f.
\]

### 2.2 Integer Loop Delay Estimation

The anti-jamming ability of amplitude difference correlation function (ADC) proposed by Li\(^3\) for the integer loop delay could be strengthened. Because the magnitude of amplitude difference \( D[n] \) can only be one of the three numbers: \(-1\), \(0\), and \(1\). Therefore, correlation function \( R(m) \) are so small (as Fig. 4 shows) that the result will probably be wrong if the signals are distorted. Nevertheless, the nonlinear distortion, various noises, and so on are inevitable for PA system.

Considering above discussions the algorithm for integer loop delay estimation applies the amplitude difference summation function (ADS) expressed as

\[
R(n_i) = \min \sum_{n=1}^{N} D_i(n) = \min S_i(n_i)
\]

where \( N \) is the is the size of summation window and \( n_i \) is the estimated integer loop delay. \( S_i(n_i) \) is the ADS function and \( D_i(n) \) is defined as

\[
D_i(n) = \| x_{in}(n + n_i) - x_{in}(n) \|
\]

where \( x_{in}(n) \) and \( x_{in}(n) \) are the feedback and input signals respectively.

So the integer loop delay \( n_i \) can be estimated by searching the index of \( R(n_i) \) which is the minimum of \( S_i(n_i) \). This algorithm is realized in FPGA as Fig. 5. \( x_{in}(n) \), \( x_{in}(n) \) and \( x_{in}(n) \), \( x_{in}(n) \) are the I and Q signals of input and feedback signals, respectively. \( A_{in}(n) \) and \( A_{fb}(n) \) are their amplitudes.

In the condition that the size of summation window \( N \) of ADS is the same as that of ADC, \( S_i(n_i) \) and \( R(m) \) are denoted as the ordinate variables in the Fig. 4, respectively.
Firstly, the signal to noise ratio (SNR) of output signal, which is adopted by the two algorithms is 70 dB. Secondly, a variable $A_c$ used to measure anti-jamming ability of a function $x$ (e.g. ADC or ADS function) can be expressed as

$$A_c = \frac{x_{\text{min}} - x_{\text{min2}}}{x_{\text{max}} - x_{\text{min}}} \times 100\% \quad (4)$$

where $x_{\text{max}} - x_{\text{min}}$ expresses the dynamic range of the function $x$. $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum of $x(n)$, $n \in [0, N]$, and $x_{\text{min2}}$ is the value nearest to $x_{\text{min}}$. $x_{\text{min}}$ is the target which we search for. So provided that dynamic range has been given, the closer $x_{\text{min2}}$ is to $x_{\text{min}}$, the larger the probability for estimation error of the function $x$ will get. In other words, the anti-jamming ability of $x$ (i.e. $A_c$) will become poorer.

Hence, the anti-jamming abilities $A_c$ for ADC function and $A_a$ for ADS function can be calculated as

$$A_c = \frac{R(m)_{\text{min}} - R(m)_{\text{min2}}}{R(m)_{\text{max}} - R(m)_{\text{min}}} \times 100\% = 4.88\%$$

$$A_a = \frac{S(n)_{\text{min}} - S(n)_{\text{min2}}}{S(n)_{\text{max}} - S(n)_{\text{min}}} \times 100\% = 7.22\% .$$

From this point the latter algorithm has stronger anti-jamming ability than the former. This standpoint is also confirmed by the latter emulation results. What is more, the algorithm only involves add and subtract. This fact will lower difficulty of its realization based on FPGA.

2.3 Fractional Loop Delay Estimation

After the integer loop delay $n_i$ has been compensated, the residual fractional portion $n_f$ will be the next design objective. The fractional delay was estimated through comparing object function step by step in [9], the estimated result probably goes wrong if processed signals are distorted (shown in Fig. 6). The numerals ①, ②, and ③ mean comparison steps. Fig. 6 shows that the common interpolation method[9] gets false result 0.15 SI rather than the actual fractional loop delay 0.45 SI when the data difference of 0.2 SI is greater than both 0.1 SI and 0.3 SI.

For avoiding this situation, we present an FIR linear interpolation method. This algorithm can be formed simply by replacing the integer loop delay $n_i$ in (2) and (3) with the fractional loop delay $n_f$, that is

$$R(n_f) = \min_{n_f} \sum_{n=1}^{N} D_f(n) = \min_{n_f} S_f(n_f) \quad (5)$$

where

$$D_f(n) = \left\| x_{\text{in}}(n + n_f) - x_{\text{in}}(n) \right\| \quad (6)$$

The proposed method compares summations of amplitude-differences $S_f(n_f)$ at the same time instead of one by one. It takes advantage of parallel arithmetic which is an important feature of FPGA. Although distortion and interference exist in the processed signals, the exact estimation result can be acquired by this method. Taking the former data used by Fig. 6 for example, the estimation outcome of this method is illustrated in Fig. 7.

The result illustrates the proposed method has stronger anti-jamming ability than the common interpolation. Though the latter method could have got the right result, it needed about 7 steps. But the former only demands 3 steps. Consequently, the proposed method has fast computational speed which is vital for the whole DPD system.

The module of this method implemented in FPGA is manifested in Fig. 8. The input signals in Fig. 8 are the same as Fig. 5. $S_f(n_{\text{in}})$ and $S_f(n_{\text{co}})$ represent the summation of amplitude-difference for 0.1 SI and 0.05 SI step size, respectively.

Fig. 6. Invalid estimation result and comparing steps of common interpolation method used by [9].

Fig. 7. Estimation result and comparing steps of FIR linear interpolation method.

Fig. 8. Fractional algorithm module upon FPGA.
Fig. 9. Architecture of FIRx upon FPGA.

FIRx ($x \in \{0, 1, 2, 3, 4, 5\}$) is elaborated by Fig. 9. $A_{n}(n-0.x)$ and $A_{n}(n-0.5)$ are the amplitudes of input signals which are interpolated by FIR filters whose step sizes are 0.1 SI and 0.05 SI, respectively.

The polarity control operates the estimation of positive or negative residual loop delay by the enable signal $N_{en}$. If the comparator and control block in Fig. 8 detects that the fractional delay is negative, we should delay $A_{n}(n)$ by 0.5 SI to keep pace with $A_{n}(n)$. The delay compensation block is used to eliminate the latent time delay which the FIR filters occupy.

3. Experimental Results

3.1 Emulation Platform

On account of obtaining the exact simulation result of proposed algorithm, a hardware emulation platform is established by applying Xilinx SPARTAN-6 family FPGA. The input signal is a single carrier WCDMA signal.

3.2 Performance of Loop Delay Estimation

Section 2.2 has given the anti-jamming ability comparison between the two functions for integer loop delay estimation. But we need more experiments to support the previous conclusion that the amplitude-difference summation (ADS) function has stronger anti-jamming ability than the amplitude-difference correlation (ADC) function.

Table 1 summarizes the comparison performances between the two functions for integer loop delay estimation with SNR degrading gradually. Based on Table 1, Fig. 10 shows the contrast of anti-jamming ability $\Delta$ between the ADC function and ADS function. We define $\Delta = 0$ as the function of $x$ (e.g. ADC or ADS function) which has no anti-jamming ability if its estimation result goes wrong. From Fig. 10, we can see that $\Delta_S$ is always greater than the corresponding $\Delta_C$.

Fig. 11 indicates the fractional loop delay estimation results and corresponding errors. The estimated fractional delay applying the proposed algorithm always follows the actual one under the permissible error 0.03SI. That is to say, the requirement that estimation error must be lower than 0.03SI has been met.

3.3 Experimental Result of DPD System with Loop Delay Compensation Applying the Proposed Method

The DPD scheme applies lookup tables coupled with the least mean square (LMS) algorithm. Fig. 12 shows the

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>$R_{max}$</th>
<th>$R_{min}$</th>
<th>$A_{n}$ (%)</th>
<th>$n_{i}$ (SI)</th>
<th>$S_{max}$</th>
<th>$S_{min}$</th>
<th>$A_{n}$ (%)</th>
<th>$n_{i}$ (SI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>298</td>
<td>11</td>
<td>4.88</td>
<td>19</td>
<td>543333</td>
<td>12723</td>
<td>7.22</td>
<td>19</td>
</tr>
<tr>
<td>60</td>
<td>295</td>
<td>11</td>
<td>4.93</td>
<td>19</td>
<td>543416</td>
<td>13779</td>
<td>7.03</td>
<td>19</td>
</tr>
<tr>
<td>50</td>
<td>293</td>
<td>15</td>
<td>5.76</td>
<td>19</td>
<td>543577</td>
<td>13926</td>
<td>7.02</td>
<td>19</td>
</tr>
<tr>
<td>40</td>
<td>289</td>
<td>21</td>
<td>0</td>
<td>20</td>
<td>543731</td>
<td>15814</td>
<td>6.74</td>
<td>19</td>
</tr>
<tr>
<td>30</td>
<td>273</td>
<td>47</td>
<td>2.21</td>
<td>19</td>
<td>538176</td>
<td>28726</td>
<td>5.60</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>247</td>
<td>105</td>
<td>0</td>
<td>20</td>
<td>535211</td>
<td>74639</td>
<td>3.94</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>231</td>
<td>153</td>
<td>0</td>
<td>16</td>
<td>586539</td>
<td>245229</td>
<td>2.68</td>
<td>19</td>
</tr>
</tbody>
</table>
comparison of output power spectrum (PSD) without DPD (upper curve) and with DPD (lower curve) whose loop delay is estimated accurately making use of the proposed algorithm. Fig. 12 demonstrates that the improvement of adjacent channel power ratio (ACPR) for the DPD system based on this algorithm is more than 18 dB.

![Comparison of output power spectrum (PSD) without DPD (above curve) and with DPD (under curve) applying the proposed method.](image)

**Fig. 12.** Comparison of output power spectrum (PSD) without DPD (above curve) and with DPD (under curve) applying the proposed method.

## 4. Conclusions

An algorithm which has strong anti-jamming ability, low complexity, and wide application scope is presented in this paper. The formulas adopted by the integer and fractional loop delay estimation reach a unification. It means they can be realized by the same architecture. Consequently, its timing control becomes much easier and it is found wide applications. Experimental results show the computational speed has been improved and the anti-jamming ability is also strengthened. This algorithm assists the DPD system with terrible noise and distortions to improve its ACPR significantly.

## References


