# Quadrature Mismatch Shaping with a Complex, Tree Structured DAC

Stijn Reekmans, Jeroen De Maeyer, Pieter Rombouts and Ludo Weyten

Ghent University (UGent)
Electronics and Information Systems Laboratory (ELIS)
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
Email: Stijn.Reekmans@elis.ugent.be

***Abstract*— Quadrature $\Sigma\Delta$ ADCs require a feedback path for both the I and the Q part of the complex feedback signal. If two separated multibit feedback DACs are used, mismatch among the unit DAC elements leads to additional mismatch noise in the output spectrum as well as an unbalance between the I and Q DAC. This paper proposes a new quadrature bandpass mismatch shaping technique. In our approach the I and Q DACs are merged into one complex DAC, which leads to near-perfect I/Q balance. To select the unit DAC elements of the complex, multibit DAC, the well-known tree structured element selection logic is generalized toward a complex structure and necessary conditions for its correct operation are derived. Finally, a very efficient first-order quadrature shaper implementation is proposed and simulations show the effectiveness of the quadrature bandpass mismatch shaping technique.**

## I. Introduction

Nowadays, many wireless communication systems use low-IF receivers (Fig. 1) [1], [2]. Such receivers consist of an antenna, an analog part, an analog-to-digital converter (ADC) and a digital signal processor (DSP). The analog part is made up of a filter, an amplifier and two mixers that are driven by LO clocks having a 90° phase difference. The quadrature mixer demodulates the received radio frequency (RF) signal to an in-phase (I) and a quadrature (Q) signal at low intermediate frequency (IF). In order to end up with a flexible receiver, functions such as channel selection need to be shifted into the digital domain. Since the performance of the ADC will determine which functions are implemented with analog circuitry and what functionality is done in the DSP, the ADC is becoming a critical part of the receiver architecture.

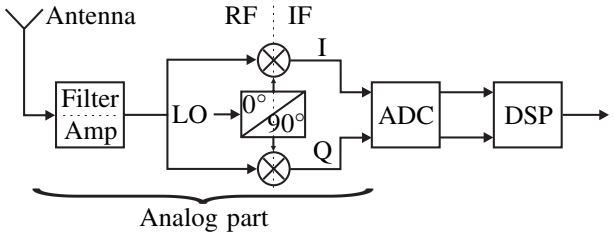


Fig. 1. Low-IF receiver architecture.

## II. Multibit quadrature bandpass $\Sigma\Delta$ ADC

A quadrature bandpass (QBP) $\Sigma\Delta$ ADC is well suited for the use in low-IF receivers. Instead of digitizing the analog I and Q signals separately with two bandpass ADCs, it performs directly the complex analog-to-digital conversion of the analog I and Q signals. Moreover, in order to achieve the same performance, the QBP $\Sigma\Delta$ ADC uses only half the integrators compared to the traditional bandpass solution [2]. This results in power- and area saving. The architecture of most QBP $\Sigma\Delta$ modulators is shown in Fig. 2. It consists of a complex loopfilter, two real quantizers and two real feedback digital-to-analog converters ($\text{DAC}_I$ and $\text{DAC}_Q$).

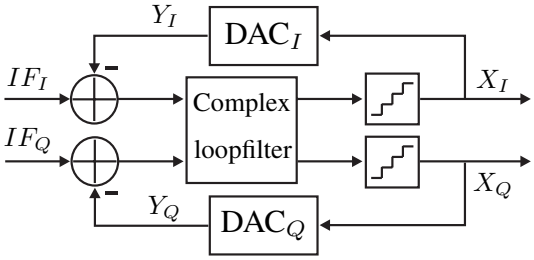


Fig. 2. A multibit quadrature bandpass $\Sigma\Delta$ ADC

In many applications, multibit $\Sigma\Delta$ ADCs are used instead of single bit converters. On the one hand, multibit ADCs can achieve much higher performance because they allow a more aggressive noise transfer function. Also, since the output of the modulator more closely resembles the desired output, it contains much less out-of-band noise. On the other hand however, multibit $\Sigma\Delta$ ADCs need a multibit DAC in their feedback path. Since any feedback DAC error is added to the modulator input, the required matching precision is of the order of the desired precision of the overall data converter and this is often beyond the practical limits of present VLSI technology [3]. To reduce the negative effects of these mismatches, dynamic element matching is used in the DAC. These mismatch-shaped DACs use digital signal processing techniques to cause most of the error's energy to reside outside the signal band.

Although the implementation of Fig. 2 is used for multibit QBP $\Sigma\Delta$ ADC [1], [2], we will show in the next section that the use of a separate DAC for the I and Q path results in a major performance degradation. Therefore, in Section IV, the concept of a complex DAC is explained which paves the way

to section V where the new QBP mismatch shaping technique is presented. In the last section a special case, the first order $f_s/4$ QBP shaped DAC, will be discussed and simulated. Simulation results confirm the effectiveness of our technique.

## III. Two real multibit DACs

A common technique for multibit DACs consists of combining unit DAC elements to form the output of the multibit DAC [3]. The architecture of such a multibit DAC is shown in Fig. 3. Each unit DAC element can generate two different analog outputs $y_i$ depending on whether its input bit $x_i$ is high or low. To represent every input sample $x[n] \in [0, 1, ..., 2^b]$, $N = 2^b$ unit DAC elements are needed. Each of these unit DAC elements can be addressed by the Element Selection Logic (ESL). One of the tasks of the ESL is to decompose each input sample $x[n]$ into $N$ output bits $x_i[n] = \{0, 1\}$ such that the sum of the $N$ bits equals $x[n]$:

$$\sum_{i=1}^{N} x_i[n] = x[n] \tag{1}$$

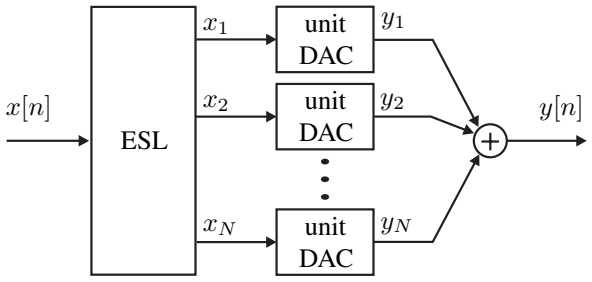


Fig. 3. A real multibit DAC with ESL.

Component mismatches, which are inevitably introduced during circuit fabrication, cause the unit DAC output levels to deviate from their ideal values. The unit DAC elements, with nominal value normalized to one, then operate according to:

$$y_i[n] = \begin{cases} 1 + \epsilon_i & \text{if selected } (x_i[n] = 1) \\ 0 & \text{if unselected } (x_i[n] = 0) \end{cases} \tag{2}$$

Here, $\epsilon_i$ is the static mismatch error of the $i^{th}$ unit DAC. As shown in [3] for a real DAC, in case of component mismatches, the overall $\text{DAC}_I$ output $y_I[n]$ and the overall $\text{DAC}_Q$ output $y_Q[n]$ has the form

$$\begin{aligned} y_I[n] &= k_I x_I[n] + e_I[n] \\ y_Q[n] &= k_Q x_Q[n] + e_Q[n] \end{aligned} \tag{3}$$

where $k_I$ and $k_Q$ indicate the constant linear gain error. The DAC mismatch noises $e_I[n]$ and $e_Q[n]$ group all nonlinear errors. The constants $k_I$ and $k_Q$ depend only on the static mismatch errors of the $\text{DAC}_I$ resp. $\text{DAC}_Q$ bank [3]. Since these two banks contain different unit DAC elements, the constants $k_I$ and $k_Q$ will in general be unequal. This inequality in the linear gain error of the I and Q feedback DAC is denoted as path mismatch. The overall effect of path mismatch becomes clear by rewriting eq. (3):

$$\begin{aligned} y[n] &= y_I[n] + jy_Q[n] \\ &= k_I x_I[n] + jk_Q x_Q[n] + e_I[n] + je_Q[n] \\ &= \tfrac{k_I + k_Q}{2} x[n] + \tfrac{k_I - k_Q}{2} x^*[n] + e[n] \end{aligned} \tag{4}$$

In the Fourier domain this is:

$$Y(f) = \frac{k_I + k_Q}{2} X(f) + \frac{k_I - k_Q}{2} X^*(-f) + E(f) \tag{5}$$

The second term of the above equation is the result of path mismatch. It causes both signals and quantization noise in the image band to alias into the desired signal band. In quadrature modulators this is highly unwanted since quantization noise is not attenuated in this image band. Next to this folding problem, also the DAC mismatch noise $E(f)$ may corrupt the correct operation of the multibit $\Sigma\Delta$ ADC.

In the next section the above folding problem will be tackled. Later on, in section V, we will ensure that meanwhile $E(f)$ exhibits low power in the signal band.

## IV. A complex DAC

In Fig. 2 two separated real DAC structures, one for the I-path and one for the Q-path, are used to give the desired feedback. As shown in Fig. 4, these two DACs could be merged into one complex DAC structure [4]. Now, there is only one DAC bank with $2N$ unit DAC elements, instead of two separated DAC banks, with each $N$ unit DAC elements.

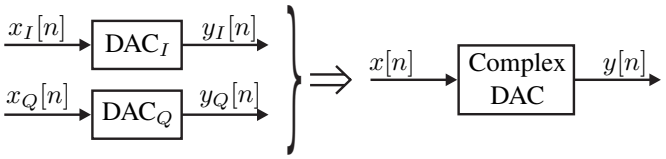


Fig. 4. Two real DACs merged into one structure

The block diagram of this complex DAC in Fig. 4 is the same as its real counterpart (Fig. 3). However, now the input $x[n]$, the outputs $y_i[n]$ and $y[n]$ and even the selection signals $x_i[n]$ are complex [1]. Moreover, each of the $2N$ unit DAC elements can be selected in three different ways. If the selection signal $x_i = 1$, the unit DAC element is selected 'I' and gives a feedback signal to the I-path. When $x_i = j$, the unit DAC element is selected 'Q' and gives a feedback signal to the Q-path and when $x_i = 0$, the unit DAC element is unselected and generates no feedback signal. So:

$$y_i[n] = \begin{cases} 1 + \epsilon_i & \text{if selected 'I' } (x_i[n] = 1) \\ j(1 + \epsilon_i) & \text{if selected 'Q' } (x_i[n] = j) \\ 0 & \text{if unselected } (x_i[n] = 0) \end{cases} \tag{6}$$

If all the unit DAC elements are of a common pool and both the I and Q signals are treated in the same way, the difference between the I- and Q-path vanishes and $k_I$ equals $k_Q$. With this technique, path mismatch and its corresponding problems are avoided. So, equation (4) becomes:

$$y[n] = kx[n] + e[n] \tag{7}$$

The remaining problem, DAC mismatch noise $e[n]$, will be tackled next.

## V. A COMPLEX, TREE STRUCTURED DAC

In order to achieve QBP mismatch noise shaping, a new element selection logic is introduced. We will discuss how the ESL maps the input sample $x[n]$ to multiple output bits $x_i[n]$. The fact that there exist different sets of $x_i$ that comply with (1), provides the necessary freedom to shape the power of the DAC mismatch noise $e[n]$ out of the signal band.

### A. Architecture

The proposed DAC architecture is shown in Fig. 5 which is a complex, tree structured DAC. For example, an architecture with four complex, unit DAC elements is shown but this can easily be generalized to any number $2^b$ of unit DAC elements. This complex, tree structured DAC is an extension of the real version [3] but exhibits some pronounced differences. Next to the fact that every signal must be interpreted as a complex value, the switching logic is different, as will be shown later.

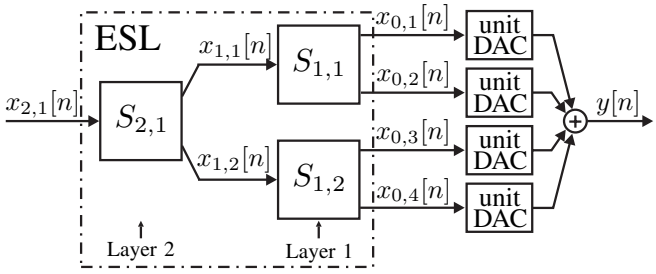


Fig. 5. Tree structure of a complex, mismatch shaping DAC

The ESL consists of $b$ (2 in Fig. 5) layers of switching blocks $S_{k,r}$, where $k$ and $r$ denote the layer number resp. the position within the layer. Each switching block $S_{k,r}$ has a single input $x_{k,r}[n]$, a top output $x_{k-1,2r-1}[n]$ and a bottom output $x_{k-1,2r}[n]$. The input/output relationship of the switching block $S_{k,r}$ is given by:

$$\begin{aligned} x_{k\text{-}1,2r\text{-}1}[n] &= \tfrac{1}{2}(x_{k,r}[n] + s_{k,r}[n]) \\ x_{k\text{-}1,2r}[n] &= \tfrac{1}{2}(x_{k,r}[n] - s_{k,r}[n]) \end{aligned} \tag{8}$$

where $s_{k,r}[n]$ is denoted as the switching sequence.

As shown in [3], the DAC mismatch noise $e[n]$ is a linear combination of the switching sequences $s_{k,r}[n]$. So, if each switching sequence is calculated as a $K^{th}$ order shaped sequence that is uncorrelated with the sequences of the other switching blocks, then the DAC noise will be a $K^{th}$ order shaped sequence.

In order to obtain QBP shaping, a QBP shaper (Fig. 6) is used to calculate the switching sequences. Essentially, it is a discrete time QBP $\Sigma\Delta$ modulator with no input signal and where the quantizers are replaced by a limiter. The limiter tries to follow its inputs $(sv^I_{k,r}[n], sv^Q_{k,r}[n])$ while forcing its outputs $(s^I_{k,r}[n], s^Q_{k,r}[n])$ to fulfill certain constraints. These constraints are required to result in a correct operation of the tree structured DAC and will be discussed next. Similar to the quantizer it replaces, the limiter can be viewed as an additive error $L(z) = L_I + jL_Q$, leading to the linear model of Fig. 6 on the right. The loopfilter $H(z)$ ensures that this additive error is shaped in order to the obtain desired shaping of $s_{k,r}[n]$.

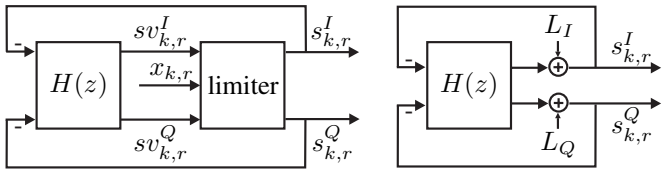


Fig. 6. QBP shaper and its linear model.

### B. Constraints

Galton presented in [3] a set of constraints for $s_{k,r}[n]$ to be satisfied in case of a real tree-structured DAC. For a complex DAC these constraints must be applied, in the first place, on both real ($\Re$) and imaginary ($\Im$) part. This is obvious, since a complex DAC should be able to process pure real or imaginary signals. Hence, a correct operation requires at least:

$$\Re(s_{k,r}[n]) = \begin{cases} \text{even} & \text{if } \Re(x_{k,r}[n]) \text{ is even} \\ \text{odd} & \text{if } \Re(x_{k,r}[n]) \text{ is odd} \end{cases} \tag{9}$$

$$\Im(s_{k,r}[n]) = \begin{cases} \text{even} & \text{if } \Im(x_{k,r}[n]) \text{ is even} \\ \text{odd} & \text{if } \Im(x_{k,r}[n]) \text{ is odd} \end{cases} \tag{10}$$

and

$$\begin{aligned} |\Re(s_{k,r}[n])| &\le \min\{\Re(x_{k,r}[n]), 2^k - \Re(x_{k,r}[n])\} \\ |\Im(s_{k,r}[n])| &\le \min\{\Im(x_{k,r}[n]), 2^k - \Im(x_{k,r}[n])\} \end{aligned} \tag{11}$$

for every $k$ and $r$. The two constraints (9) and (10) ensure a correct integer result for the division by two in eq. (8). The constraint (11) makes sure that both real and imaginary part of the two outputs of each switching block are in the range of $\{0, 1, ..., 2^{k-1}\}$. The latter is the expansion of the Number Conservation Rule formulated in [3].

However, this naive condition extension of the real tree structured DAC is inadequate and additional restrictions are to be set. For example, when $x_{1,1}[n] = 1+j$ then $s_{1,1}[n] = 1+j$ satisfies the above conditions, however with (8) this choice results in $x_{0,1}[n] = 1 + j$. In other words, the first unit DAC is selected 'I' as well as 'Q' at the same time which is impossible. In general, since $x_{0,i}$ is limited to $\{0, 1, j\}$, the sum of the real and the imaginary part of both outputs of the switching block $S_{k,r}[n]$ must be in the range of $\{0, 1, ..., 2^{k-1}\}$ and not only the real as well as the imaginary part of it. Hence, it is required that:

$$|\Re(s_{k,r}[n]) + \Im(s_{k,r}[n])| \le 2^k - \Re(x_{k,r}[n]) - \Im(x_{k,r}[n]) \tag{12}$$

The implementation of the above conditions (9), (10), (11) and (12) in a general sense is rather hardware expensive. The following proposed conditions fulfill the above constraints and are more restrictive than necessary but simplify the hardware of the switching block significantly:

$$s_{k,r}[n] = \begin{cases} 0 & \text{if } \Re(x_{k,r}[n]) \text{ \& } \Im(x_{k,r}[n]) \text{ are even} \\ \pm 1 & \text{if } \Re(x_{k,r}[n]) \text{ is odd \& } \Im(x_{k,r}[n]) \text{ is even} \\ \pm j & \text{if } \Re(x_{k,r}[n]) \text{ is even \& } \Im(x_{k,r}[n]) \text{ is odd} \\ \pm 1 \mp j & \text{if } \Re(x_{k,r}[n]) \text{ \& } \Im(x_{k,r}[n]) \text{ are odd} \end{cases} \tag{13}$$

Based upon the state variables $sv^I_{k,r}[n]$ and $sv^Q_{k,r}[n]$, the limiter will decide whether to use + or - in the above constraints.

### C. The limiter

The limiter tries to follow the state variables $sv^I_{k,r}[n]$ and $sv^Q_{k,r}[n]$ while forcing its outputs $s^I_{k,r}[n]$ and $s^Q_{k,r}[n]$ to fulfill the above constraints (13).

For the limiter, there are two usefull information sequences. First of all, we have two parity bits which are zero or one in case $x^I_{k,r}$ and $x^Q_{k,r}$ are even or odd respectively. These parity bits are important since the if-conditions of the constraints of (13) are based on the parity of $x^I_{k,r}[n]$ and $x^Q_{k,r}[n]$. Secondly, the sign of the state variables $sv^I_{k,r}[n]$ and $sv^Q_{k,r}[n]$ is important, it determines whether to use + or - in (13).

In general, the outputs $s^I_{k,r}[n]$ and $s^Q_{k,r}[n]$ will equal the sign of their state variables multiplied by the respective parity bit. However two special cases can occur:

1) In case of a zero state variable, its sign is indefinite and a choice can be made concerning the appropriate output if the parity bit is one. This choice can be fixed or a simple dithering technique, to suppress spurious tones, can be implemented in the limiter [3].

2) If both real and imaginary parity are odd, a contention can appear if both state variables have the same sign. In this case the state variable with the largest absolute value will be given preferential treatment, if both state variables are equal then again a choice can be made.

## VI. First order $f_s/4$ QBP shaper

The structure of Fig. 7, which is a special case of Fig. 6, calculates the switching sequences so that the DAC mismatch noise is a first order shaped sequence with a center frequency of $f_s/4$. The spectrum of $s_{k,r}[n]$ has one zero at $j$, resulting in first order $f_s/4$ QBP shaping:

$$S_{k,r}(z) = S^I_{k,r}(z) + jS^Q_{k,r}(z) = (1 - jz^{-1})L(z) \qquad (14)$$

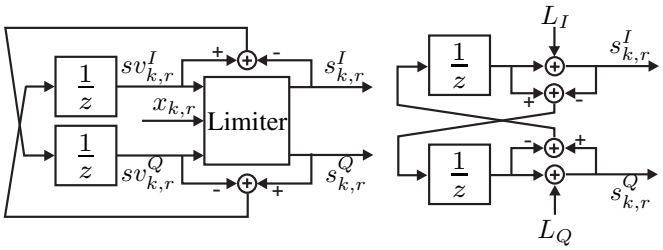


Fig. 7. First order $f_s/4$ QBP shaper and linear model.

With regards to the digital implementation, it can be proven that the state variables are bounded to $\{-2, -1, 0, 1, 2\}$ for an appropriate choice of initial conditions. As a result, the structure can be implemented with low bit-width logic.

The first order $f_s/4$ QBP tree structured DAC, with 8 unit DAC elements to which a random mismatch of $1\%$ was assigned, has been simulated. Fig. 8 shows the DAC noise spectrum. The desired first order shaping is clearly present. Fig. 9 shows which of the 8 unit DAC elements are selected 'I', which ones 'Q' and which ones are unselected: e.g. element 5 is selected 'I', 'Q' and unselected in the first three samples.

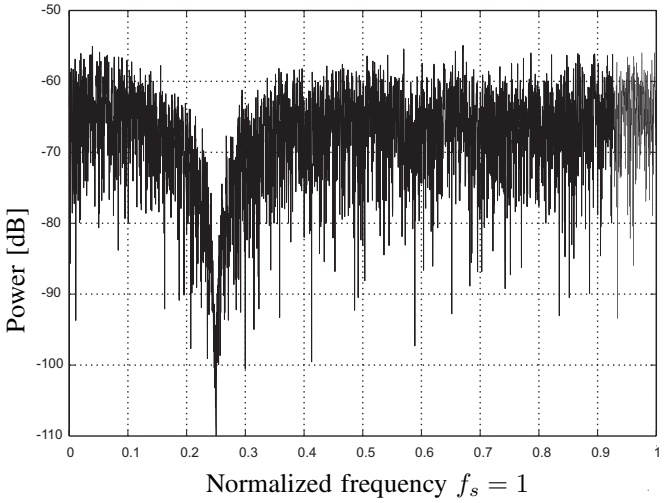


Fig. 8. DAC noise spectrum with $1\%$ mismatch (8 unit DAC elements).

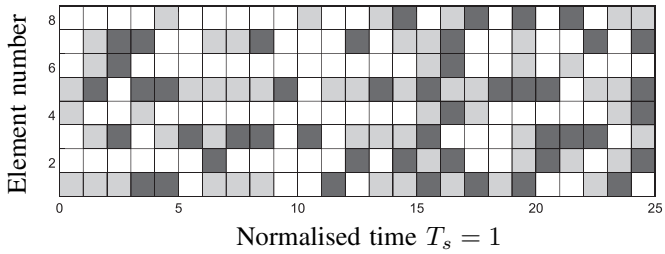


Fig. 9. Element selection pattern, light gray = selected 'I', dark gray = selected 'Q' and white = unselected.

## VII. Conclusion

The tree structured ESL has been extended to a complex DAC structure, in which the I and Q DAC are merged. As a result, near-perfect I/Q balance is achieved. Meanwhile, DAC mismatch noise is quadrature bandpass shaped. As a validation of the presented technique, the first order $f_s/4$ QBP shaper has been presented, simulated and discussed.


## Acknowledgment

S. Reekmans is supported by the Special Research Fund of Ghent University and J. De Maeyer is supported by a fellowship of the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium).


## References


[1] K. W. Martin, "Complex Signal Processing is Not Complex," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 9, pp. 1823–1836, Sept. 2004.

[2] S. Jantzi, K. Martin, and A. Sedra, "Quadrature bandpass $\Delta\Sigma$ Modulation for Digital Radio," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, Dec. 1997.

[3] I. Galton, "Spectral Shaping of Circuit Errors in Digital-to-Analog Converters," *IEEE Transactions on Circuits and Systems II*, vol. 44, no. 10, pp. 808–817, Oct. 1997.

[4] R. Schreier, "Quadrature Mismatch-Shaping," in *ISCAS '02, IEEE International Symposium on Circuits and Systems*, vol. 4, 2002, pp. 675–678.