TUNABLE MISMATCH SHAPING FOR QUADRATURE BANDPASS DELTA-SIGMA DATA CONVERTERS

Waqas Akram and Earl E. Swartzlander, Jr.

Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas 78712–0240

ABSTRACT

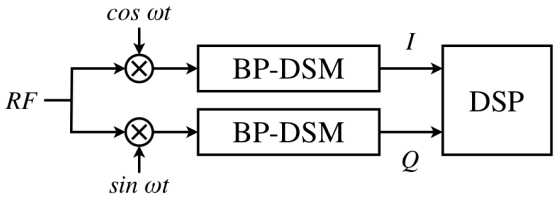
This paper presents an architecture for quadrature bandpass mismatch shaping that allows the center frequency of the mismatch suppression band to be tunable over the entire Nyquist range. The approach is based on the previously reported complex-valued tree-based mismatch shaper, and extends this to allow tunable operation. The proposed design has been implemented using VHDL and synthesized to logic gates. The hardware complexity and mismatch shaping performance of the proposed architecture are compared to that of a reference architecture, which uses separate tunable mismatch shapers for each complex component path. Simulation results show consistent mismatch shaping performance across the entire tuning range.

***Index Terms*—** Bandpass Delta-Sigma; Quadrature Delta-Sigma; Mismatch Shaping; Tunable Delta-Sigma; DAC Element Matching; Unit Element DAC

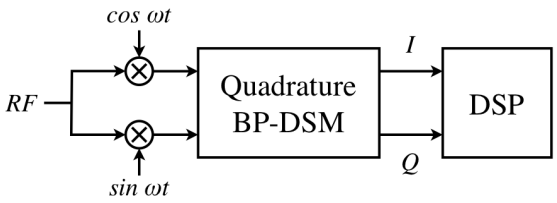
1. INTRODUCTION

[1] Modern wireless systems are facing a proliferation of wireless standards, making it necessary to maximize hardware reconfigurability in order to minimize costs. In addition, the continued scaling of CMOS technology has led to greater degrees of chip integration between the analog radio frequency (RF) front-ends and the digital signal processing (DSP) back-ends. As a result, it is far more attractive to process the intermediate frequency (IF) signal directly within the digital domain. Due to their high linearity over narrow bandwidths, bandpass delta-sigma ($\Delta\Sigma$) modulators are rapidly becoming the data converter of choice for these applications [2][3][4][5][6].

In a typical low-IF receiver architecture, the RF signal is first demodulated into a complex IF signal, which consists of the in-phase (I) and quadrature (Q) component signals. Subsequently, these component signals are individually digitized using a pair of bandpass $\Delta\Sigma$ modulators, as shown in Figure 1(a). One complication arising from the use of this method is the potential for path mismatch between the I and the Q channels, which leads to performance degradation [7][8]. This architecture can be improved by using a single complex-valued or quadrature bandpass (QBP) $\Delta\Sigma$ modulator, as shown in Figure 1(b) [7].



(a) Separate BP-$\Delta\Sigma$ ADCs



(b) Quadrature BP-$\Delta\Sigma$ ADC

Fig. 1. Low-IF bandpass $\Delta\Sigma$ receiver architecture

Higher-order $\Delta\Sigma$ analog-to-digital converters (ADC) can provide a much higher signal-to-noise ratio (SNR), for a given oversampling ratio (OSR), by using a multibit internal quantizer [9]. However, performance is limited due to distortion caused by device mismatch errors when the quantized signals emerge from the DAC embedded within the $\Delta\Sigma$ ADC feedback loop. In order to reduce these nonlinearities in the DAC transfer function, the DAC element mismatch errors can be randomized or spectrally shaped away from the signal band using a mismatch shaper [10][9][11].

Although mismatch shaping reduces DAC nonlinearity within the signal band, it fails to eliminate the overall linear gain error through the DAC. In a quadrature bandpass $\Delta\Sigma$ modulator, using separate DACs for the I and Q compo-

[1]This is an expanded version of a paper presented at the 2010 SiPS conference [1].

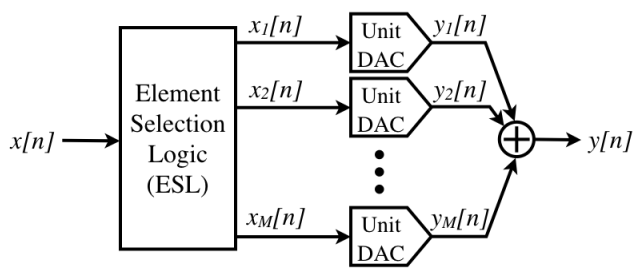


Fig. 2. Unit-element DAC with element selection logic

nents of the quantized complex signal can lead to severe path mismatch due to the differing gain errors through each path. However, the two real-valued DACs can be combined into a single mismatch shaping complex DAC. This leads to identical gain errors in the I and Q paths, and thereby eliminates I/Q path mismatch through the DAC itself [12][13].

An important feature of data converters used in multi-standard wireless transceivers is the ability to place the center of the signal-band at different frequencies within the Nyquist band, according to the requirements placed by the system design, as well as the particular wireless standard in use. $\Delta\Sigma$ data converters can easily be designed with programmable loop filter coefficients in order to achieve this. However, $\Delta\Sigma$ modulators employing multibit quantization still require the mismatch noise to be removed or noise-shaped away from wherever the signal-band has been placed. This can be achieved with the use of a tunable mismatch shaper [4][14][15].

In the case of a quadrature bandpass $\Delta\Sigma$ modulator employing a multibit quantizer, the center-frequency of the mismatch suppression band through a mismatch shaping complex DAC needs to be tunable. This paper extends previously known mismatch shaping techniques to enable such a feature.

Section 2 provides an overview of quadrature bandpass mismatch shaping, along with previously reported techniques. Section 3 describes the proposed tunable mismatch shaping technique, followed by the hardware implementation in Section 4. Hardware complexity and mismatch shaping performance of the proposed architecture are compared to those of a reference architecture in Section 5. The paper is concluded in Section 6.

2. MISMATCH SHAPING

A unit-element DAC is commonly used in multibit $\Delta\Sigma$ systems. It consists of M unit-sized elements that can be combined to generate $M+1$ different output levels, as shown in Figure 2. The element-selection logic (ESL) converts the binary DAC input to a vector of single-bit controls to the unit-element DACs, $x[n] = \sum_{k=1}^{M} x_k[n]$, where $x_k[n] \in \{0,1\}$ The DAC output is similarly formed by summing the outputs of all the DAC elements, $y[n] = \sum_{k=1}^{M} y_k[n]$, where $y_k[n] = \Delta x_k[n]$ and Δ denotes the nominal step-size of each DAC element.

Unfortunately, component mismatch due to non-idealities in the manufacturing process leads to non-ideal values for each of the DAC elements. As a result, each DAC element exhibits errors in the output levels:

$$y_k[n] = \begin{cases} \Delta + \epsilon_{h_k} & \text{if } x_k[n] = 1 \\ \epsilon_{l_k} & \text{if } x_k[n] = 0 \end{cases} \quad (1)$$

where ϵ_{h_k} and ϵ_{l_k} denote the static mismatch error when the DAC element is enabled and disabled, respectively. The overall DAC output is given by:

$$y[n] = \alpha x[n] + \beta + \epsilon[n], \quad (2)$$

where α, β and ϵ are the gain error, offset error and aggregate mismatch error, respectively, and all three quantities depend exclusively on the element mismatch errors [16].

Generally during element selection, there are multiple ways in which the input $x[n]$ can be used to select DAC elements to form the output, $y[n]$. These additional degrees of freedom can be exploited to vary the pattern of unit element selection in a way that spectrally shapes the mismatch error away from the signal-band [10][16]. This is known as mismatch noise shaping, and is achieved without changing the actual number of selected DAC elements from the number that need to be activated in order to reproduce a given DAC input.

2.1. Quadrature Path Mismatch

In a quadrature bandpass $\Delta\Sigma$ modulator, both the I and Q components of the quantized complex signal require a DAC. In addition to the mismatch noise component $\epsilon[n]$, differences between mismatch errors in the two DACs results in path mismatch:

$$y_I[n] = \alpha_I x_I[n] + \beta_I + \epsilon_I[n] \quad (3)$$
$$y_Q[n] = \alpha_Q x_Q[n] + \beta_Q + \epsilon_Q[n] \quad (4)$$

The composite DAC output is thus given by:

$$y[n] = \frac{\alpha_I + \alpha_Q}{2} x[n] + \frac{\alpha_I - \alpha_Q}{2} x^*[n] + \beta + \epsilon[n] \quad (5)$$

where:

$$y[n] = y_I[n] + j y_Q[n] \quad (6)$$
$$\beta = \beta_I + j\beta_Q \quad (7)$$
$$\epsilon[n] = \epsilon_I[n] + j\epsilon_Q[n] \quad (8)$$

As a result, any gain mismatch ($\alpha_I \neq \alpha_Q$) will lead to folding from the negative frequency image band to the positive frequency signal band. Since the $\Delta\Sigma$ modulator loop filter

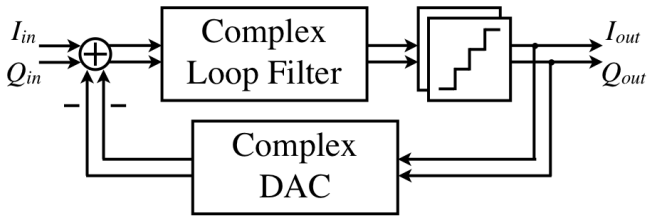


Fig. 3. QBP-$\Delta\Sigma$ ADC with complex DAC

is asymmetric, i.e., there is little or no noise shaping in the image band, gain mismatch results in a higher quantization noise within the signal band.

The two real-valued DACs can be combined into a single mismatch shaping complex DAC, as shown conceptually in Figure 3. This allows both paths to use all available DAC elements in the process of shaping the mismatch noise. As a result, the combined DAC eliminates path mismatch by producing equal gain errors in the I and Q paths ($\alpha_I \equiv \alpha_Q$), as described in [13] and [17]. This is the key advantage to using a quadrature mismatch shaping DAC, but comes at the expense of increased hardware complexity.

2.2. Quadrature Mismatch Shaping

A mismatch shaping scheme for complex DACs has been proposed in [12]. This method generalizes the basic vector-based mismatch shaper from [10] to complex-valued signals. This technique is realized by implementing the mismatch shaper loop filter in complex arithmetic and allowing the vector quantizer to take values from $\{0, 1, j\}$. The technique requires overly complicated hardware that results in reduced mismatch suppression. A complex butterfly shuffler mismatch shaping DAC is considered in [17]. This is an extension of previously reported methods used to whiten or shape the mismatch noise [18]. These methods require a higher level of hardware complexity than the tree-structured approach [17].

2.3. Tunable Mismatch Shaping

A novel technique for performing tunable mismatch shaping on real signals has been proposed in [14]. This technique relies upon the generalized N-path filter principle in conjunction with a prototype mismatch shaper to replicate the mismatch transfer function N times around the unit circle. By using the well-known data-weighted averaging (DWA) technique, a hardware-efficient first-order tunable mismatch shaper can be realized [15]. However, when applied to quadrature signals, DWA mismatch shaping exhibits lower levels of performance [12]. The technique proposed in this paper extends the tree-based complex mismatch shaper from [17] so as to allow control over the center frequency of the mismatch transfer function.

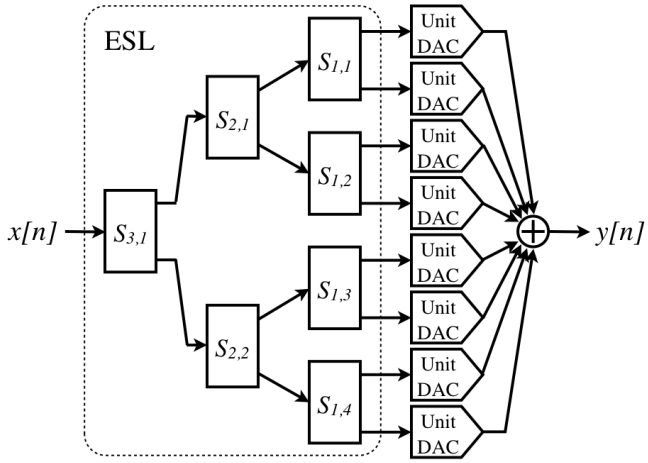


Fig. 4. Tree-structured element selection logic

3. PROPOSED TUNABLE TECHNIQUE

The mismatch shaper used in this work is largely based on the tree-based approach described in [16] and adapted to handle complex signals as described in [17]. Figure 4 shows an example of a $M + 1$ level DAC, with $M = 8$, using the tree structure to perform element selection. Each of the blocks labeled S_k, r is a switching block that routes the input data in two possible directions. There are $log_2 M$ layers of switching blocks. Each switching block operates according to

$$x_{k-1,2r-1}[n] = \frac{1}{2}(x_{k,r}[n] + s_{k,r}[n]) \tag{9}$$

$$x_{k-1,2r}[n] = \frac{1}{2}(x_{k,r}[n] - s_{k,r}[n]) \tag{10}$$

where $s_{k,r}[n]$ is a switching sequence generated within each switching block. The value of the switching sequence $s_{k,r}[n]$ at each sample interval n dictates what portion of the input data $x_{k,r}[n]$ is routed through each of the outputs of the switching block [16].

As the input travels through the tree, portions of the input data word are spread across the branches of the tree until they arrive at the unit-DACs, where only a single bit determines whether or not the DAC element is selected for activation. In the case of quadrature mismatch shaping, the input to the DAC is complex-valued. Therefore, each unit-DAC output $y_{k,r}[n]$ can produce one of three possible output values: $y_{k,r}[n] \in \{0, 1, j\}$ [12][13]. The combined DAC has twice the number of unit elements as each individual I and Q DAC, so the total number of unit-DACs remains the same as when using a pair of real-valued DACs.

Figure 5 shows the structure of a switching block that implements the operations described by Equations 9−10 for complex-valued inputs. Since the actual data must remain unchanged from the input of the ESL block to the output, each switching block must ensure that it generates a switching se-

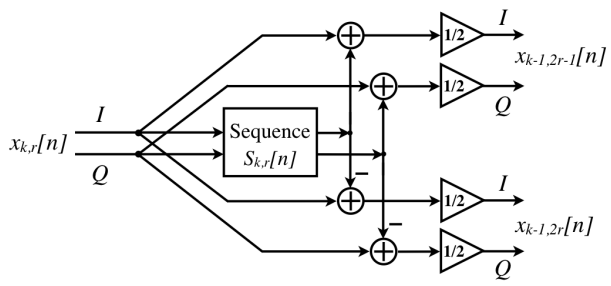


Fig. 5. Switching block for complex data

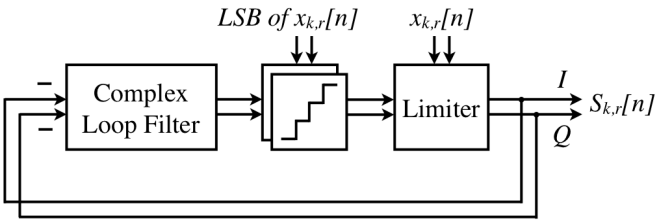


Fig. 6. Complex switching sequence generator

quence that forces the output data to satisfy this condition. This restriction is known as the *number conservation rule* [16]. These restrictions need to be modified for the case of quadrature signals in order to ensure that the complex data also satisfies this number conservation rule [17].

The DAC mismatch error sequence $\epsilon[n]$ is expressed as

$$\epsilon[n] = \sum_{k} \sum_{r} \Delta_{k,r} s_{k,r}[n] \tag{11}$$

where $\Delta_{k,r}$ is the nominal value of the unit-DAC step size. If the switching sequence $s_{k,r}[n]$ is generated as an *L*th-order noise-shaped sequence, uncorrelated with the switching sequences in the other switching blocks, this will result in an *L*th-order noise-shaped DAC mismatch error sequence [16]. It follows that in order to gain control over the center frequency of the mismatch transfer function, there needs to be a way to control the center frequency of the noise shaping function within each switching sequence generator.

Figure 6 shows the structure of a complex-valued sequence generator. The noise shaping is achieved by employing a zero-input $\Delta\Sigma$ modulator, with a complex loop filter. The complex-valued number conservation rule is enforced by the quantizer and limiter blocks [17]. The order and frequency location of the noise shaping function are determined by the complex loop filter. Therefore, a tunable mismatch shaping function can be achieved by simultaneously controlling the complex loop filters in all the switching blocks.

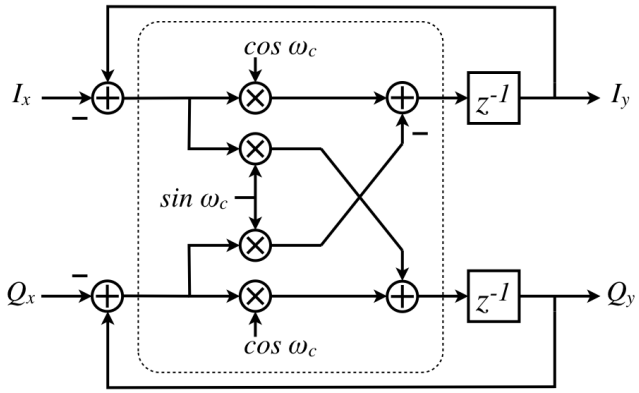


Fig. 7. Tunable 1^{st}-order complex loop filter

4. HARDWARE IMPLEMENTATION

In this section, comparisons are made between the hardware complexity of the proposed technique and a reference architecture. The latter is a conventional implementation using a pair of real-valued tunable bandpass mismatch shaping DACs. These techniques extend the complex-valued tree-based mismatch shaper from [17] and the real-valued tree-based mismatch shaper from [16] in such a way that allows control over the center frequency of the mismatch transfer function.

As described in the previous section, each switching block in the mismatch shaping tree contains a sequence generator. The sequence generator shown in Figure 6 is essentially a zero-input $\Delta\Sigma$ modulator, and forms the heart of the proposed implementation. There are three parts to the sequence generator: (i) a tunable complex loop filter, (ii) a quantizer for each of the *I* and *Q* paths, and (iii) a complex limiter.

4.1. Tunable Complex Loop Filter

The loop filter within the switching sequence modulator is implemented as a tunable complex filter, with the center frequency tuned to the center frequency of the desired signal band. The structure of a tunable first-order complex filter is shown in Figure 7, where ω_c is the center frequency of the signal band, $0 \leq \omega_c \leq \pi$, and $[I_x, Q_x]$ and $[I_y, Q_y]$ are the complex-valued input and output vectors of the loop filter. The core of this unit is a rotation unit whose operation is described by:

$$\begin{bmatrix} I_y \\ Q_y \end{bmatrix} = z^{-1} \begin{bmatrix} \cos\omega_c & -\sin\omega_c \\ \sin\omega_c & \cos\omega_c \end{bmatrix} \begin{bmatrix} I_y - I_x \\ Q_y - Q_x \end{bmatrix} \tag{12}$$

In order to simultaneously tune the loop filters of all the switching sequence generators, the same complex coefficients must be provided to all $log_2 M$ switching blocks and their loop filters. The filter coefficients associated with each tuning setting can be stored in a lookup-table or read-only memory (ROM), and indexed by the tuning frequency.

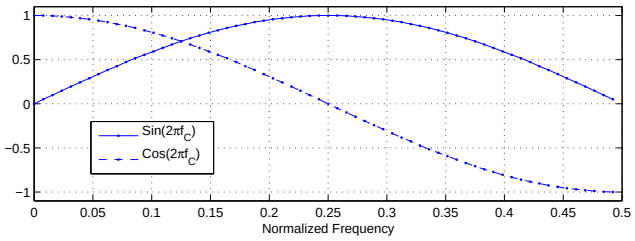


Fig. 8. Coefficient values over the entire tuning range

Figure 8 shows the coefficients plotted over the entire tuning range. Since the coefficients are essentially the sine function computed at the tuning frequency, the storage requirements can be reduced by taking advantage of the natural quadrant-symmetry exhibited by the sine wave. The second quadrant of the sine function can be reproduced from the first quadrant of the sine by inverting the phase. The first quadrant of the cosine function is identical to the second quadrant of the sine function, and the second quadrant of the cosine function can be reproduced from the first quadrant of the sine function by inverting the polarity.

The size of the ROM can be further reduced by using a direct digital frequency synthesizer (DDFS), such as one using polynomial interpolation [19]. Alternatively, the coefficients can be computed using existing hardware from elsewhere in the wireless system. Most modern transceiver systems include a DDFS, which can be borrowed for a few cycles in order to compute the pair of coefficients. The coefficients would only need to be re-computed when the location of the signal band center frequency is changed.

4.2. Path Quantizer

As described in Section 3, each switching block in the tree must satisfy a restriction known as the *number conservation rule* [16]. The restriction is imposed in order to ensure that data remains unchanged between the input and the output of the overall tree, and hence the ESL block (shown in Figure 2). This restriction can be transferred from the overall tree to each switching block by ensuring that the two outputs of each switching block always add up to equal the value of the switching block input.

Within each switching block, the sequence generator produces a number that is used to create the two switching block outputs, as shown in Figure 5. The sequence output is first simultaneously added to and subtracted from the switching block input. These two results are then divided by two (or truncated by a single LSB) before supplying them to subsequent switching blocks. Due to this division by two, and since the switching block outputs cannot be allowed to take on fractional values, the results of the addition and subtraction must always be maintained as even numbers.

This condition can be satisfied by restricting the switching sequence to always be even-valued when the switching block input data is even-valued, and vice-versa. As described in [16], this can be achieved by selecting a mid-tread quantizer when the switching block input is even, and a mid-rise quantizer when it is odd. Since the addition and subtraction within the switching block occurs separately for each of the I and Q paths, it is sufficient to maintain separate quantizers for each path. Each of the quantizers must independently choose between mid-rise and mid-tread quantization according to the respective I or Q components of the switching block input.

4.3. Complex Limiter

The purpose of the limiter is to prevent the switching block from ever producing a negative number. In order to achieve this, the magnitude of the sequence generator output is forcibly limited according the switching block input data. In the case of the combined complex sequence generator, additional limiter constraints are needed to satisfy the number conservation rule [17].

The operation of the limiter in the lowest layers of the switching-block tree is shown in Table 1. The first column shows the possible switching block inputs. The middle columns show the real and complex components of the quantizer output. The last column shows the corresponding sequence output resulting from the combination of switching block input and quantizer output. These sequence generator outputs guarantee that each of the two switching block outputs take on valid control values for the complex DAC unit-element $s_k[n] \in \{0, 1, j\}$.

There are a few input combinations where two different output sequence values can produce the same valid complex switching block outputs. These cases are shown in Table 1, indicated with a pair of possible output sequence options. In the proposed technique, whenever such a condition arises, the choice between the two options is made according to the current output of a pseudo-random sequence, which is created by a linear-feedback shift register (LFSR).

As shown in Figures 4−6, the combinatorial logic path connecting the DAC input to the DAC unit-element output contains every single quantizer and limiter block in the mismatch shaping tree. This path can be pipelined in order to shorten the critical path for high speed operation. However, pipelining creates additional clock cycle delays through the DAC, and any increase in this delay can pose modulator stability problems when these DACs are employed within the modulator feedback loop of a $\Delta\Sigma$ ADC [9]. It is therefore beneficial to minimize the complexity of all limiter blocks.

The limiter at the lowest switching levels need to be as complex as described by Table 1, in order to ensure that the combined complex output is a valid control for the individual complex DAC elements. However, the limiters in the upper levels need only satisfy the number conservation rule. It is possible to reduce the complexity of the limiters used in

Table 1. Limiter table for the lowest layer of the tree

Input	Re{Quan}	Im{Quan}	Sequence Output
0	X	X	0
2	X	X	0
$2j$	X	X	0
1	0	X	$1, -1$
1	$+$	X	1
1	$-$	X	-1
j	X	0	$j, -j$
j	X	$+$	j
j	X	$-$	$-j$
$1+j$	0	0	$1-j, -1+j$
$1+j$	0	$+$	$-1+j$
$1+j$	0	$-$	$1-j$
$1+j$	$+$	0	$1-j$
$1+j$	$-$	0	$-1+j$
$1+j$	$+$	$-$	$1-j$
$1+j$	$-$	$+$	$-1+j$
$1+j$	$+ > +$		$1-j$
$1+j$	$- > -$		$1-j$
$1+j$	$+ < +$		$-1+j$
$1+j$	$- < -$		$-1+j$
$1+j$	$+ = +$		$1-j, -1+j$
$1+j$	$- = -$		$1-j, -1+j$

the upper layer switching blocks by drastically simplifying the limiter rules. In the proposed technique, the switching sequences generated by all switching blocks are limited to producing I and Q values from the ranges $\{-1, 0, 1\}$ and $\{-j, 0, j\}$, respectively. This effectively places tighter constraints on values allowed in the switching sequence output.

Table 2 shows the operation of the simplified limiter used in all switching blocks in the upper layers of the tree. The table applies to both I and Q paths. The first column shows the switching block inputs as either even or odd. The second column shows the possible quantizer outputs. The third column shows the corresponding sequence output resulting from the combination of switching block input and quantizer output.

Table 2. Limiter table for all upper layers of the tree

Input Data	Quantizer Output	Sequence Output
even	0	0
even	$+$	0
even	$-$	0
odd	1	1
odd	-1	-1
odd	> 1	1
odd	< -1	-1

Simulation results have shown that using this highly constrained limiter for upper layer switching blocks does not significantly reduce mismatch noise suppression in the signal band. The advantages of using such a scheme extend beyond the hardware savings within the limiter blocks: the range of possible sequence values fed back into the loop filter now take on essentially trivial values, thereby reducing the hardware complexity of the initial stages of the loop filter.

4.4. Simulation of Tunable Operation

A $\Delta\Sigma$ modulator can be used either as an analog-to-digital converter (ADC) or a digital-to-analog converter (DAC). In a $\Delta\Sigma$ ADC, the quantizer is itself an ADC, but with a far lower precision than the overall ADC. A correspondingly coarse DAC is required within the modulator loop in order to provide feedback, as shown in Figure 3. In a $\Delta\Sigma$ DAC, the entire modulator loop remains in the digital domain, including the coarse quantizer. However, the quantized signal still needs to be converted to an analog signal and thus requires a coarse DAC at the output. In either case, the use of a mismatch-shaping complex DAC as the coarse DAC provides the same benefit of suppressing non-linearities within the band of interest [9].

In this section, the tunable quadrature mismatch shaper is simulated using data generated by a programmable quadrature bandpass $\Delta\Sigma$ modulator, whose specifications are given in Table 3. The tunable modulator noise-transfer function places three poles in the signal band, and one in the corresponding image band, resulting in a 4^{th}-order modulator. An oversampling ratio (OSR) of 128 allows coverage of the entire Nyquist band using just 64 tuning settings, resulting in a normalized signal bandwidth as specified in Table 3.

Table 3. Quadrature BP $\Delta\Sigma$ modulator specifications

Modulator order	4
OSR	128
DAC elements	16
Tuning settings	64
Normalized signal bandwidth	0.0078π

The 16-element DAC employs complex unit elements, as described in [12]. The sequence generators in the mismatch shaper tree all use the tunable 1^{st}-order complex filter shown in Figure 7, tuned to the same frequency setting. The DAC mismatch errors are modeled as uniformly distributed random mismatch, with a standard deviation of 3% of a single unit element value.

To illustrate the operation of the tunable quadrature mismatch shaper, Figure 9 shows the frequency spectra at three different signal band tuning frequencies, centered at (a) 0.1875π, (b) 0.2812π, and (c) 0.3750π. Each figure shows three distinct frequency responses. The ideal response is labeled *no mismatch*, and corresponds to that of a complex

DAC without any mismatch errors, where performance is limited only by the tunable bandpass $\Delta\Sigma$ modulator itself. The *unshaped* response is when random mismatch is added to the DAC, but no mismatch shaping is performed. In this case, the element selection for each path is performed using a simple thermometer code. The *shaped* response is when mismatch is added and mismatch noise shaping is performed using the proposed technique.

Figure 10 shows the zoomed-in views of each corresponding signal-band. These simulations are with a specific configuration of randomly-generated DAC errors. The in-band SNR shows a variation across frequency due to a combination two distinct effects: (1) the variation of ideal modulator performance at the specific tuning frequency, and (2) the in-band spurious tones generated by those particular randomly-generated DAC mismatch errors. The next section presents simulation results for a wider set of randomly-generated DAC mismatch errors.

5. COMPARISON

In this section, the hardware complexity and mismatch shaping performance of the proposed architecture is compared to a reference architecture. The latter is a conventional implementation using a pair of real-valued tunable bandpass mismatch shaping DACs, as described in [16], and extended to provide tunable operation over the Nyquist range.

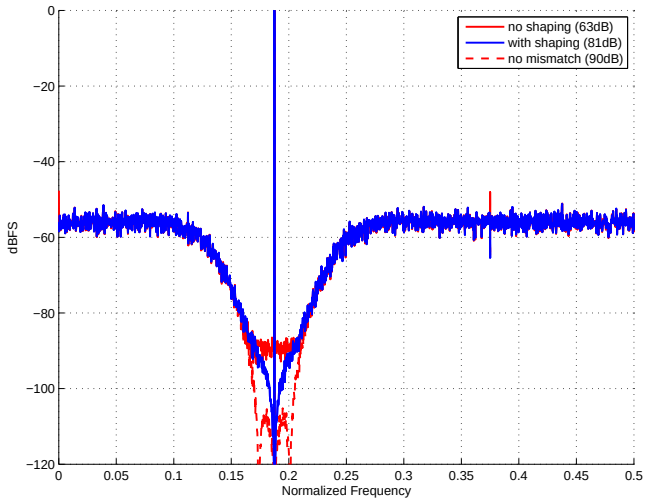
Table 4 compares the structures of the mismatch shaping trees used in the two mismatch shaper architectures for 16-element complex DACs. It is evident that the proposed architecture will necessarily occupy a larger chip area because of the extra switching node, as well as the larger complex-valued switching nodes.

Table 4. Specifications of reference and proposed designs

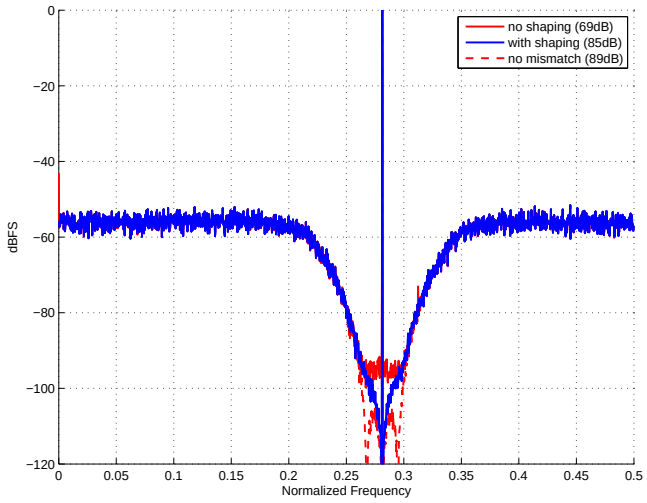
Architecture	REF.	PROP.
Total DAC unit elements	16	16
Max. elements for each I/Q path	8	8
Number of tree structures	2	1
Number of switching nodes/tree	7	15
Total number of switching nodes	14	15
Type of switching node data	Real	Complex

Despite the increased hardware complexity, the primary motivation for using the mismatch shaping complex DAC remains intact: the elimination of gain mismatch between the I/Q paths, which is achieved by definition with the decision to employ a combined mismatch shaping complex DAC in the first place [12][13][17].

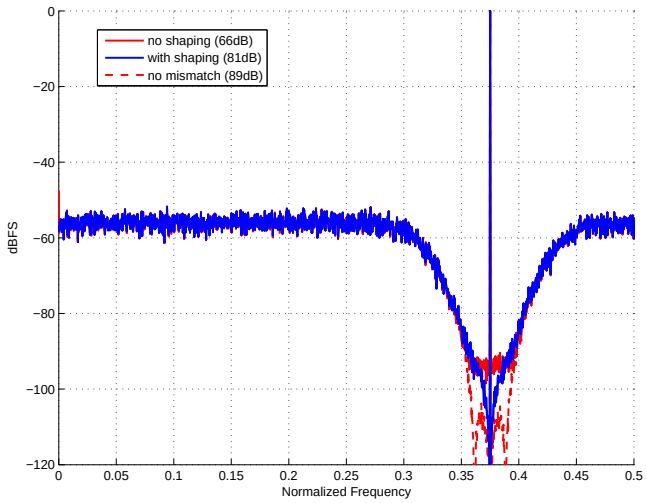
The reference and proposed architectures for tunable bandpass mismatch shaping have been implemented using VHDL for coarse DAC sizes ranging from $2 - 128$ unit DAC



(a) Center frequency tuned to 0.1875π

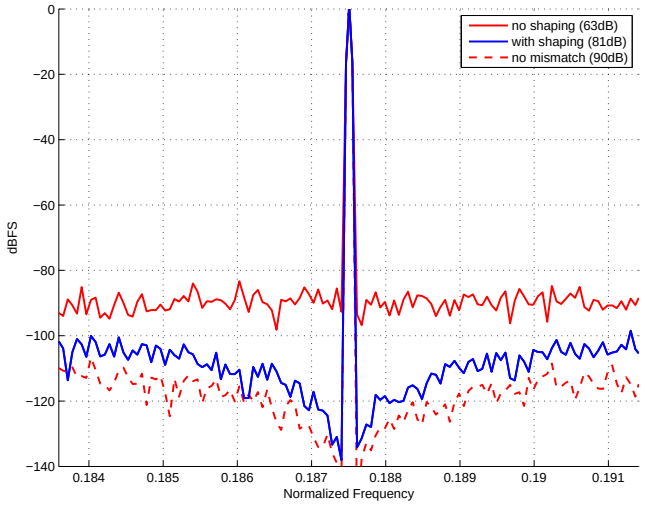


(b) Center frequency tuned to 0.2812π

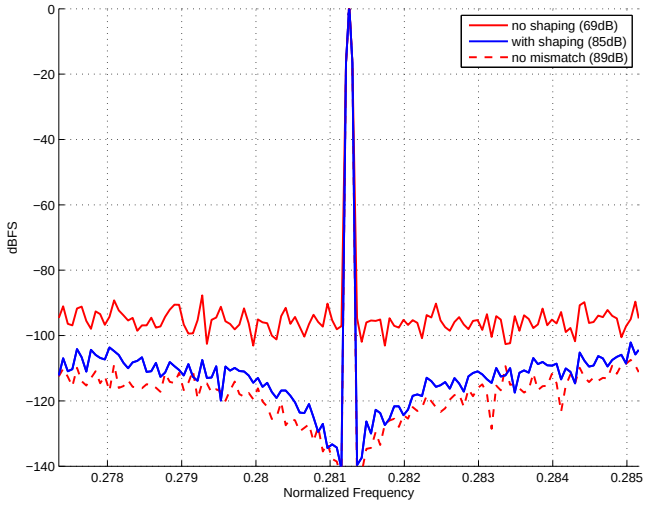


(c) Center frequency tuned to 0.3750π

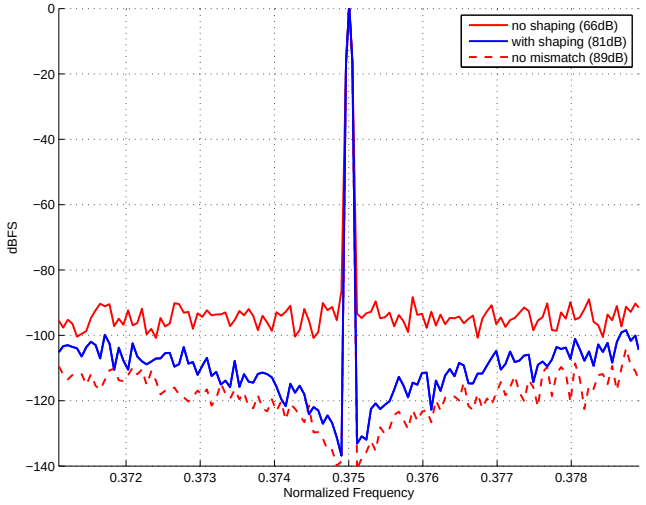
Fig. 9. Full spectrum with 3% mismatch error



(a) Center frequency tuned to 0.1875π



(b) Center frequency tuned to 0.2812π



(c) Center frequency tuned to 0.3750π

Fig. 10. Signal band detail with 3% mismatch error

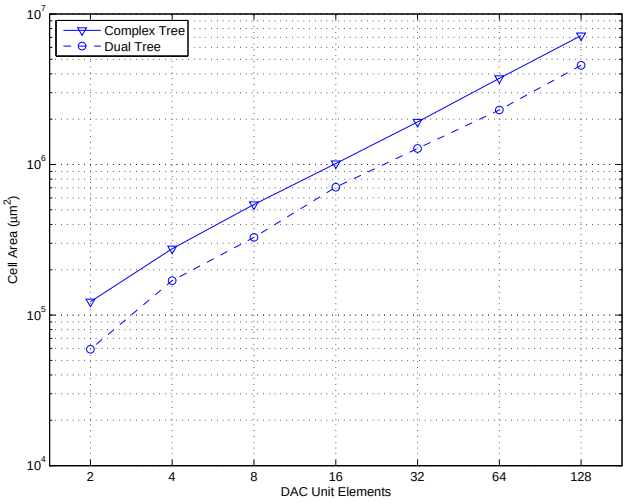


Fig. 11. Hardware complexity for different DAC widths

elements These are typical values as found in the literature. The data-path and coefficient widths in all architectures have been selected such that the in-band SNR performance of the two architectures is roughly equivalent at each tuning setting, across the entire tuning range. The designs for each DAC precision have been synthesized using Synopsys Design Compiler [20] under the conditions specified in Table 5.

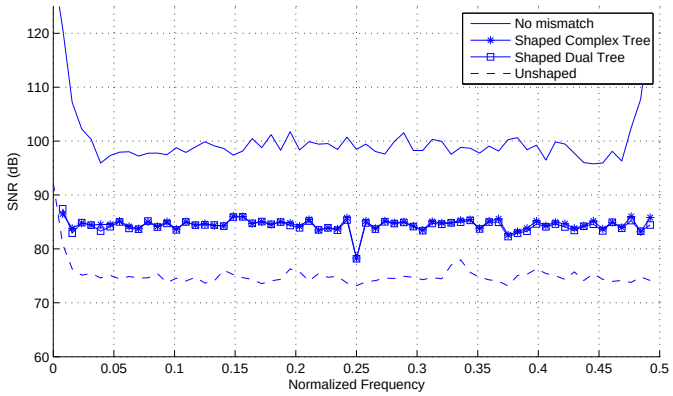
Table 5. Hardware synthesis conditions

Process	0.18 μm CMOS
Device corners	slow-$NMOS$ / slow-$PMOS$
Voltage	1.62 V
Temperature	125^oC
Clock speed	100 MHz

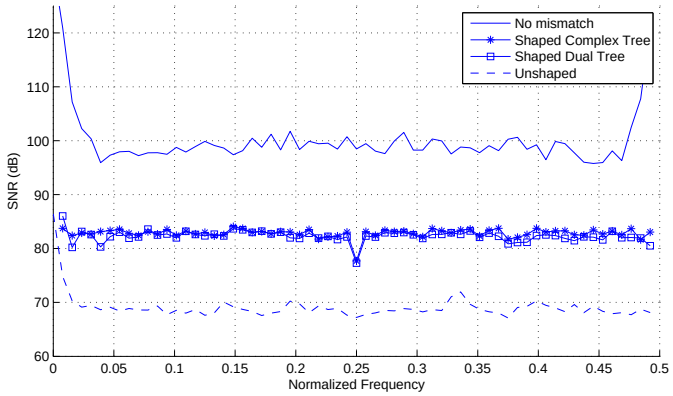
Logic synthesis results for several DAC sizes are shown in Figure 11. The reference architecture is labeled *dual tree*, and the proposed architecture is denoted *complex tree*. As is expected, the proposed quadrature structure requires a greater area than the conventional twin path structure, for all DAC sizes. However, the complex tree structure completely eliminates the gain mismatch between the two paths.

Each architecture has been simulated using data generated by a tunable 4^{th}-order quadrature bandpass $\Delta\Sigma$ modulator whose specifications are shown in Table 3. Figure 12 plots the in-band SNR when applying a full-scale complex-valued sinusoid with a frequency randomly selected from within the signal band. This is repeated for each tuning setting across the entire tuning range, and results are shown for mismatch errors with (a) 1%, (b) 2%, and (c) 3% standard deviation.

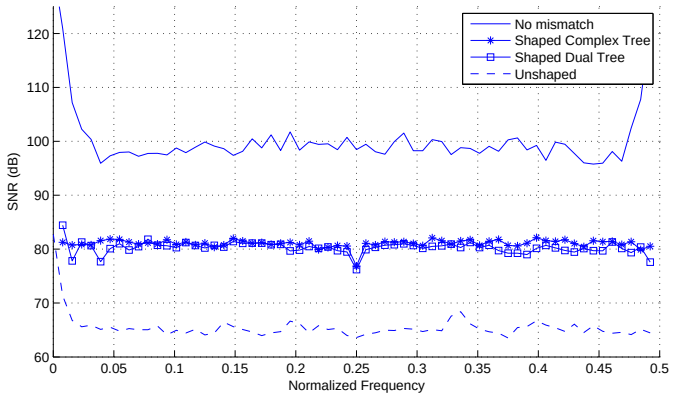
In each figure, the SNR with *no mismatch* is that of a DAC without any mismatch errors, where the performance is limited by the tunable bandpass $\Delta\Sigma$ modulator specified



(a) SNR with 1% mismatch error



(b) SNR with 2% mismatch error



(c) SNR with 3% mismatch error

Fig. 12. SNR across tuning range with single-tone input

in Table 3. The *unshaped* SNR is the response when mismatch is added, but no shaping is performed. In this case, the element selection is performed using a simple thermometer code. The *shaped* SNR is the response when mismatch is added and mismatch shaping is performed using the corresponding shaper. In all figures, the proposed tunable shaper is denoted by *shaped complex tree* while the conventional tunable shaper is denoted by *shaped dual tree*. Each data-point corresponds to the SNR average computed for a thousand different randomly-generated DAC mismatch configurations.

The mismatch-shaped performance shows consistent improvement over the unshaped performance, for the entire tuning range. For the sake of comparison, the performance of the reference and proposed architectures have been designed to be roughly equivalent in order to compare the hardware complexity overhead of eliminating the gain mismatch between the I and Q paths. This performance equivalence is achieved by appropriately reducing data-path and coefficient widths.

6. SUMMARY AND CONCLUSION

Wireless transceivers are increasingly employing quadrature bandpass $\Delta\Sigma$ data converters in applications where high linearity is required over a narrow bandwidth. The dynamic range of a $\Delta\Sigma$ data converter can be improved by using a multibit quantizer in the modulator loop. However, device mismatch errors in the multibit DAC cause distortion that has a direct impact on the effective modulator performance. Mismatch shaping is an established technique for alleviating the effects of mismatch errors, but previously reported techniques for mismatch shaping DACs require the signal band be located at a fixed frequency. Multi-standard wireless systems can benefit from the ability to place the center of the signal band at arbitrary frequency locations within the Nyquist range of the oversampled data converter. This requires that the quadrature mismatch shaper also retain the ability to arbitrarily select the frequency location of the mismatch suppression band. In a quadrature bandpass $\Delta\Sigma$ modulator, using separate DACs for the I/Q components of the quantized complex signal can lead to path mismatch due to the differences in the gain through each DAC path. Using a single complex-valued mismatch shaping DAC instead of a pair of real-valued DACs effectively eliminates the I/Q path mismatch through the coarse DAC itself. This paper extends previously known quadrature mismatch shaping techniques to enable the center frequency of the mismatch noise-transfer function to be tunable over the entire Nyquist range.

The proposed design has been implemented for different DAC widths, using VHDL to model the hardware. The limiter in upper layer switching blocks has been simplified in order to reduce the path delay from input to output, as well as to lower the overall hardware complexity. In order to evaluate the hardware complexity overhead of the proposed approach, a reference architecture utilizing a pair of real-valued tunable

mismatch shapers has also been implemented with the same specifications. All architectures have been synthesized into logic gates and results show a consistent increase in the hardware complexity as compared to the conventional approach, for the same level of mismatch shaping performance. By definition, the quadrature mismatch shaper eliminates path mismatch through the course DAC, and simulation results show consistent in-band SNR performance over the entire tuning range.

7. REFERENCES

[1] W. Akram and E. E. Swartzlander, Jr., "Tunable mismatch shaping for quadrature bandpass delta-sigma data converters," *IEEE Workshop on Signal Processing Systems*, pp. 198–203, Oct. 2010.

[2] A. Jerng and C. G. Sodini, "A wideband $\Delta\Sigma$ digital-RF modulator for high data rate transmitters," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 1710–1722, 2007.

[3] S. M. Talaeie, T. Copani, B. Bakkaloglu, and S. Kiaei, "A bandpass $\Delta\Sigma$ RF-DAC with embedded FIR reconstruction filter," *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 578–579, 2006.

[4] T. S. Kaplan, J. F. Jensen, C. H. Fields, and M.-C. F. Chang, "A 2-GS/s 3-bit $\Delta\Sigma$-modulated DAC with tunable bandpass mismatch shaping," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 603–610, 2005.

[5] S. Luschas, R. Schreier, and H.-S. Lee, "Radio frequency digital-to-analog converter," *IEEE Journal of Solid-Sate Circuits*, vol. 39, pp. 1462–1467, 2004.

[6] M. Miller and C. S. Petrie, "A multibit sigma-delta ADC for multimode receivers," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 475–482, 2003.

[7] S. A. Jantzi, K. W. Martin, and A. S. Sedra, "Quadrature bandpass delta-sigma modulation for digital radio," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 1935–1950, 1997.

[8] L. J. Breems, E. C. Dijkmans, and J. H. Huijsing, "A quadrature data-dependent DEM algorithm to improve image rejection of a complex sigma-delta modulator," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1879–1886, 2001.

[9] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters*, IEEE Press, New York, NY, 1997.

[10] R. Schreier and B. Zhang, "Noise-shaped multibit D/A converter employing unit elements," *Electronics Letters*, vol. 31, pp. 1712–1713, 1995.

[11] I. Galton, "Why dynamic-element-matching DACs work," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, pp. 69–74, 2010.

[12] R. Schreier, "Quadrature mismatch-shaping," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 675–678, 2002.

[13] S. Reekmans, J. De Maeyer, P. Rombouts, and L. Weyten, "Quadrature mismatch shaping with a complex, tree structured DAC," *IEEE International Symposium on Circuits and Systems*, pp. 2969–2972, 2006.

[14] W. Akram and E. E. Swartzlander, Jr., "Tunable N-path mismatch shaping for multibit bandpass delta-sigma modulators," *43rd Asilomar Conference on Signals, Systems and Computers*, pp. 1288–1291, Nov. 2009.

[15] W. Akram and E. E. Swartzlander, Jr., "A novel technique for tunable mismatch shaping in oversampled digital-to-analog converters," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1534–1537, Mar. 2010.

[16] I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Transactions on Circuits and Systems II*, vol. 44, pp. 808–817, 1997.

[17] S. Reekmans, J. De Maeyer, P. Rombouts, and L. Weyten, "Quadrature mismatch shaping for digital-to-analog converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, pp. 2529–2538, 2006.

[18] M. Vadipour, "A bandpass mismatch noise-shaping technique for sigma-delta modulators," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, pp. 130–135, 2004.

[19] W. Akram and E. E. Swartzlander, Jr., "Digital frequency synthesis using piece-wise polynomial interpolation," *37th Asilomar Conference on Signals, Systems and Computers*, pp. 2237–2241, Nov. 2003.

[20] Synopsys, Inc, "Design-Compiler Ultra 2010," `http://www.synopsys.com`.