Addressing static and dynamic errors in unit element multibit DACs

J. De Maeyer, P. Rombouts and L. Weyten

Nonlinearity in multibit current-steering digital-to-analogue converters (DACs) originates from static mismatch and switching imperfections. Several techniques are presented to avoid this nonlinearity. When applied, the DACs are suitable for use in the feedback of continuous-time ΣΔ-converters.

Introduction: For high-speed applications continuous-time ΣΔ analogue-to-digital converters (ADCs) are more attractive than their discrete-time counterparts. They usually have a current-steering DAC in the feedback path. When a multibit variant is used, static mismatch and dynamic switching imperfections (e.g. turn-on and turn-off delay, rise time, ...) will limit the linearity of the converter [1]. In the past dynamic element matching techniques were presented to tackle static mismatch [2–4]. However, problems with dynamic errors were paid little attention. This Letter provides several implementations to tackle the problems with dynamic errors while avoiding performance degradation due to static error sources.

Dynamic error model: Consider a current-steering DAC consisting of N nominally matched current sources. A typical waveform for the ith element is shown in Fig. 1. Three deviations from the ideal output can be recognised. First, the sources exhibit static mismatch (ε_i). Secondly, switching is not ideal (e.g. nonzero rise time, ...). Thirdly, the switching imperfections differ from element to element. This is called dynamic mismatch. As in [1] our analysis will be performed in discrete-time and only the low frequency information will be considered. The grey-coloured surfaces of Fig. 1 correspond to the dynamic error of the current source. This error can be defined as to consist of two parts. One part δ is the mean value of the dynamic error over the N elements. The other part δ_i is caused by the variance in the dynamic error from element to element, i.e. dynamic mismatch. By definition the mean value of δ_i is zero.

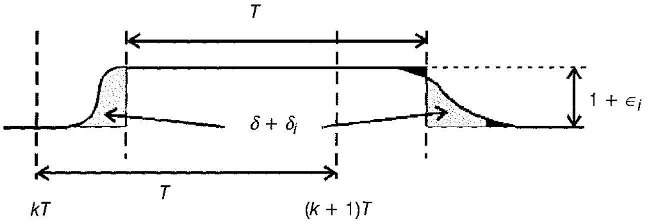

Fig. 1 *Typical output waveform of a current source*

T is clock period. Both grey-coloured regions represent the dynamic error and have an equal surface. The two black surfaces add in an opposite way to the total dynamic error.

Let k indicate time and $D(k)$ be the input value of the DAC. For each element i a selection signal $S_i(k)$ is generated, in such a way that $D(k) = \sum_{i=1}^{N} S_i(k)$. Thus, $S_i(k)$ equals 1 if element i is used at point k in time, and 0 otherwise. Then, it can be derived that the total DAC error, referred to as $e(k)$, is given by:

$$e(k) = \underbrace{\sum_{i=1}^{N} \varepsilon_i S_i(k)}_{e_1(k)} + \underbrace{\delta \sum_{i=1}^{N} |S_i(k) - S_i(k-1)|}_{e_2(k)} + \underbrace{\sum_{i=1}^{N} \delta_i |S_i(k) - S_i(k-1)|}_{e_3(k)} \tag{1}$$

A similar model was used in [1], but there $e_3(k)$ was not taken into account.

The first error contribution $e_1(k)$ in (1) originates in the static mismatch ε_i between the current sources. By properly selecting $S_i(k)$, $e_1(k)$ can be spectrally shaped. The switching imperfections cause the second term $e_2(k)$ and dynamic mismatch the third term $e_3(k)$. They will only contribute to the total DAC error when elements are switched. Surprisingly this is independent of whether these elements are turned on or off. In general, $e_2(k)$ and $e_3(k)$ depend in a complex and nonlinear way on the input value $D(k)$. This causes distortion.

Tackling $e_2(k)$: As can be seen from (1), the error $e_2(k)$ is proportional to the number of elements that switch. Based on this observation modified mismatch shaping (MMS) [1] was proposed. Here the number of elements that switch was targeted to be constant and the static error $e_1(k)$ was first-order shaped. However, a hardware-expensive vector quantiser structure was used and problems with $e_3(k)$ were not tackled.

Basic idea. Suppose the following restriction is set on the selection signal $S_i(k)$: '$S_i(k)$ and $S_i(k-1)$ are never both 1'. Then it is possible to simplify $e_2(k)$ and $e_3(k)$ to:

$$\begin{aligned} e_2(k) &= \delta \cdot (D(k) + D(k-1)) \\ e_3(k) &= \sum_{i=1}^{N} \delta_i S_i(k) + \sum_{i=1}^{N} \delta_i S_i(k-1) \end{aligned} \tag{2}$$

Now $e_2(k)$ depends linearly on the input value $D(k)$. So, under this restriction switching imperfections no longer cause nonlinearity. Furthermore, (1) and (2) show that, when $S_i(k)$ is generated by any kind of static mismatch shaping technique, $e_1(k)$ and $e_3(k)$ will be shaped. Summarising, implementations that fulfil the restriction will not suffer from distortion and if these implementations shape static mismatch they will automatically shape dynamic mismatch as well. In the following Section such implementations will be proposed.

Implementations: In hardware terms the restriction becomes: 'Every current source may only be turned on if it was previously turned off and after usage it needs to be turned off'. The technique of return-to-zero (RZ) achieves this by dividing the clock period into two phases. An element that is turned on during the first phase, is turned off during the second. However, this constant return to zero greatly increases the jitter sensitivity and the slew rate requirements on the analogue output stage [2]. In dual-return-to-zero [2] these problems were avoided. However, the system now operates in both phases of the clock, actually requiring a double-frequency system-clock.

The restriction above can be stated differently as: 'A current source can only be used during one clock cycle.' Based on this, we propose double data weighted averaging (double DWA). In this technique elements are selected sequentially, starting from the next available unused element. Only unused elements at time k can be used at $k+1$. To ensure that there are always sufficient unused elements available to realise $D(k)$, the number of unit elements should be twice the maximum input value. An eight-element selection diagram is shown in Fig. 2*a*. As in data weighted averaging (DWA) [3] the resulting $S_i(k)$ will shape $e_1(k)$. Moreover, as the restriction is satisfied, $e_2(k)$ will not cause distortion and $e_3(k)$ will also be shaped by $S_i(k)$. Another implementation is the tree structure without dither [4]. Again, the amount of unit elements is doubled. Then the restriction is automatically satisfied, which is illustrated in Fig. 2*b* for a simple example.

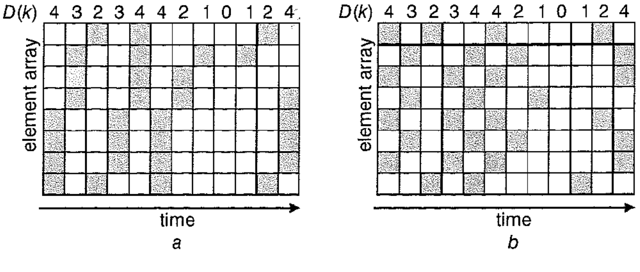

Fig. 2 *Selection diagram for double DWA and a tree without dithering*

a For double DWA
b For a tree without dithering
For both eight-element DACs maximum input is 4. As can be seen the restriction is satisfied.

In general, the restriction can only be satisfied if the unit elements are divided into two groups, one could be described as 'previously used → need to be turned off', another as 'previously not used → could be turned on'. Both techniques mentioned above realise this division implicitly and shape the static and dynamic mismatch to the first-order. Both can be implemented in a hardware efficient way. Unfortunately their performance will slightly be degraded by spurious tones [3, 4]

If the spurious tones cannot be accepted and/or higher-order shaping is desired, the proposed division of the unit elements can also be performed explicitly. One possible way is to use a vector quantiser structure [1] with a selection logic that for each element decides in which group it belongs. Hence, the restriction can be satisfied. Thus, again $e_2(k)$ will not cause distortion and $S_i(k)$ can be generated as to shape $e_1(k)$ and $e_3(k)$ now to the nth order.

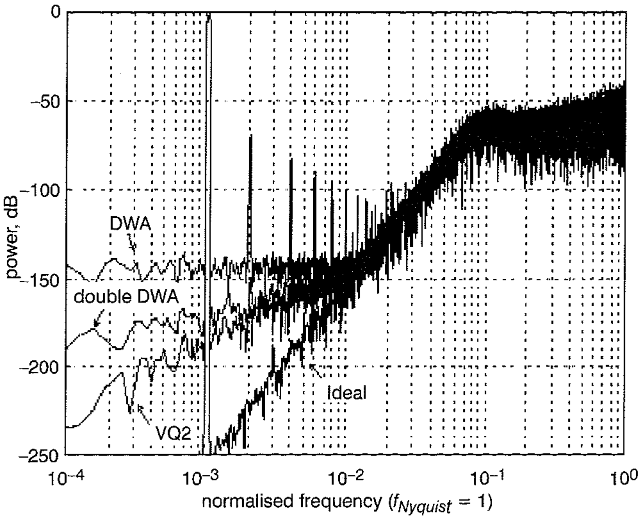

Fig. 3 *Simulated output spectrum for an ideal DAC, a DAC running conventional DWA, a DAC running proposed technique of double DWA and a DAC running a vector quantiser structure with second-order filter (VQ2)*

Simulations: First, several SPICE simulations in continuous-time were compared with MATLAB simulations based on (1). A close resemblance was noticed, which proves the accuracy of the model.

To illustrate the strength of the proposed techniques, a 3-bit datastream generated with a fifth-order $\Sigma\Delta$ encoded sinusoid was provided to a current-steering DAC. The elements of the DAC were afflicted with a 0.1% static mismatch, 0.1% of the clock period difference between turn-on and turn-off delay and 0.1% dynamic mismatch. Typical results of MATLAB simulations are shown in Fig. 3 for the case of conventional DWA, double DWA and a second-order modified vector quantiser structure. It is clear that the proposed double DWA technique achieves a first-order spectral shaping of both dynamic and static errors, while conventional DWA suffers from pronounced distortion. Similar results (not shown) are obtained with the proposed tree of Fig. 2*b*. From Fig. 3 it can also be seen that the proposed vector quantiser structure can indeed realise a second-order spectral shaping of both the static and dynamic errors.

Conclusion: We have analysed dynamic errors in current-steering DACs and shown that they contribute to nonlinearity in a different way as static (mismatch) errors. Several techniques are presented to avoid nonlinearity caused by both these error sources.

Acknowledgment: J. De Maeyer is supported by a fellowship of the Fund for Scientific Research—Flanders (F.W.O.-V., Belgium).

23 May 2003
Electronics Letters Online No: 20030698
DOI: 10.1049/el:20030698

J. De Maeyer, P. Rombouts and L. Weyten (*CAS-Group, Department of Electronics and Information Systems Laboratory (ELIS), Ghent University, Belgium*)

References

1 SHUI, T., SHREIER, R., and HUDSON, F.: 'Mismatch shaping for a current-mode multibit delta-sigma DAC', *IEEE J. Solid-State Circuits*, 1999, **34**, pp. 331–338

2 ADAMS, R., NGUYEN, K.Q., and SWEETLAND, K.: 'A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling', *IEEE J. Solid-State Circuits*, 1998, **33**, pp. 1871–1878

3 BAIRD, R.T., and FIEZ, T.S.: 'Linearity enhancement of multibit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging', *IEEE Trans. Circuits Syst. II*, 1995, **42**, pp. 753–762

4 GALTON, I.: 'Spectral shaping of circuit errors in digital-to-analog converters', *IEEE Trans. Circuits Syst. II*, 1997, **44**, pp. 808–817