

Received July 14, 2021, accepted July 27, 2021, date of publication July 30, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101574

Compensating Processing Delay in Excess of One Clock Cycle in Noise Shaping Loops Without Altering the Filter Topology

CHRISTIAAN E. LOKIN¹, (Student Member, IEEE), DANIËL SCHINKEL², (Member, IEEE),
RONAN A. R. VAN DER ZEE¹, (Member, IEEE),
AND BRAM NAUTA¹, (Fellow, IEEE)

¹IC-Design Group, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, 7522 NB Enschede, The Netherlands

²Axign, 7521 BE Enschede, The Netherlands

Corresponding author: Christiaan E. Lokin (c.e.lokin@utwente.nl)

This work was supported by the Netherlands Organisation for Scientific Research (NWO), through the TTW-Research Program under Project 13911 ICDC.

ABSTRACT This work presents an analytical method to design a stable loop filter for a noise shaper with parasitic delays or poles in the loop. This method does not require an increase in the order of the loop or extra feedback paths and keeps the original poles and zeros of the noise-transfer function at the same location. The method has been applied to a low-pass loop filter with a parasitic unit delay and a band-pass loop filter with either three unit delays or a high-frequency pole, obtaining stable loop filters in all three cases.

INDEX TERMS Excess loop delay (ELD), loop filter, noise shaping, sigma-delta analog-to-digital converter (ADC).

I. INTRODUCTION

Excess delay in a feedback loop can cause instability when it is not taken into account. Hence, solutions have been sought to mitigate or overcome the effects of excess delay on loop stability and system performance.

In the design of a noise-shaping loop, it is desirable to maximize the in-band performance for a given loop filter order and oversampling ratio (OSR). Maximizing performance also means that the stability margins are becoming small and that any parasitic element in the loop, such as a delay, could potentially make the system unstable.

In the research field of continuous-time $\Sigma\Delta$ -modulators, techniques have been developed to compensate for excess loop delay (ELD). To restore the desired noise shaping characteristics of the delay-less system, a new set of coefficients for the loop filter has to be determined. Also, depending on the digital-to-analog converter (DAC) pulse shape and the length of the delay, a direct path around the quantizer may be required to restore the original noise transfer function (NTF).

The associate editor coordinating the review of this manuscript and approving it for publication was Guangdeng Zong¹.

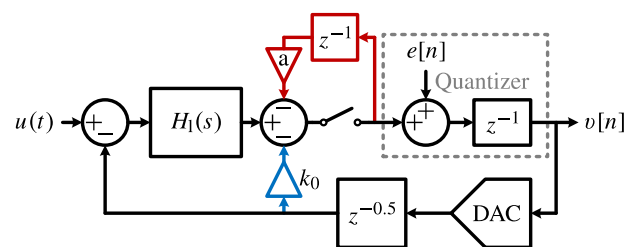


FIGURE 1. $\Sigma\Delta$ modulator with excess delay in the loop compensated using the method in [1] using a local loop (red) and a direct path (blue).

A shortcoming is, that when the delay in the digital path exceeds one clock cycle, it is not possible to apply classical ELD compensation. In that case, the NTF cannot be restored to its desired shape. The addition of local feedback as proposed in [1] as shown in Fig. 1 can further improve the stability, but at the cost of requiring analog building blocks.

We aim to make a $\Sigma\Delta$ noise shaper in an existing piece of existing digital hardware with tunable coefficients, with no option to realize a direct path around the quantizer. Given this constraint, the objective is to still realize a stable noise shaper that can cope with excess delay and/or poles, at the expense of little performance.

The novelty of the presented technique is that it does not require additional analog building blocks to mitigate excess poles and can still give a stability improvement. This does however result in lower performance than applying classical ELD compensation with a local feedback path. The application area where such a technique can have merits are, e.g., $\Sigma\Delta$ ADCs with a successive approximation register (SAR) quantizer, high-speed $\Sigma\Delta$ ADCs where the quantizers have to be pipelined, advanced dynamic element matching (DEM) algorithms [2], and mixed-signal systems with ADCs in the loop and digital processing delay.

An example of an implementation in existing hardware is the pulse-width modulation (PWM) ripple reduction technique presented in [3]. There, a noise shaping loop was designed using programmable digital filters in the hardware of the AX5689 [4]. The programmable loop filter in that system was originally designed to be used for signals in the audio band, however, for the application of PWM ripple reduction, higher frequency signals needed to be processed. The processing delay in the system caused a frequency-dependent phase shift which initially made the desired filter unstable after closing the loop.

In this paper, a technique is proposed that accommodates integer cycles of excess delay and excess high-frequency poles in a digital loop filter by appending pole-zero pairs to the desired NTF. The system is stabilized without requiring additional hardware, such as direct paths or extra filter orders.

II. PRIOR ART

A common design procedure for $\Sigma\Delta$ analog-to-digital converters (ADCs) involves choosing the desired NTF and calculating the open-loop transfer function from the NTF [5], [6]. For this design method to work, it is assumed that the system is free of excess delay. However, in an actual continuous-time $\Sigma\Delta$ ADC as shown in Fig. 1, the quantizer and the feedback DAC cannot give an instantaneous output. Consequently, the integrator(s) will integrate a delayed version of the desired feedback signal, which changes the desired closed-loop transfer and furthermore compromises the stability of the loop.

A. ELD COMPENSATION

The open-loop transfer function of a system with ELD has to be modified so that the desired, delay-less closed-loop transfer and NTF are restored. To do this, the combination of non-idealities, such as the settling, slewing and quantizer delay can all be modeled together as a single delay term to make compensation straightforward [7], [8].

In a first order modulator this can be achieved by using, for instance, return-to-zero (RZ) coding in the DAC, making the system resilient to delays up to $\frac{1}{2}T_s$ [9]. For larger delays or when non-return-to-zero (NRZ) coding is used, part of the DAC pulse will exceed the sampling window and give an incorrect result after sampling the integrated signal. To restore the NTF in that case, it is required to add a direct feedback path around the quantizer with a gain dependent on the delay in the loop. In addition to that, the loop filter

coefficients have to be tuned. Together, these measures compensate for the apparent error made by integrating the delayed DAC pulse [7], [9]–[13].

A different implementation uses a predictive comparator which estimates the delay of the output from the DAC pulse to change the comparison threshold [13] to obtain the same functional result as a direct path around the quantizer. Besides the analog solutions, it is also possible to compensate for the delay in the digital domain by adding a local digital feedback loop after the quantizer [12], [14]. However, this implementation does require a multi-bit DAC and sufficient precision in the digital registers to implement the coefficient that is conventionally used in the analog direct path.

A shortcoming of these classical ELD compensation solutions that has to be addressed is the compensation of delays in excess of one clock cycle.

B. DIFFICULTIES FOR ELD COMPENSATION

The ELD compensation technique cannot be applied to any system, e.g. when the quantizer delay exceeds one or more clock cycles it is not possible to fully restore the desired NTF. In that case, the quantizer can be bypassed by using a sample and hold circuit with the feedback coefficient of the conventional direct path to achieve a similar stabilizing effect [1], [15], [16]. Focusing on the method in [1], the result of the compensation is essentially a lead-lag filter in front of the quantizer (Fig. 1). Other examples of techniques in ADCs that complicate the implementation of ELD are DEM algorithms in the DAC or pipeline delay in high-speed noise-shaping algorithms. However, a direct path can be taken before the DEM algorithm to provide at least a less precise, fast feedback path [15]. In another more recent work, a time-interleaved quantizer based on a dual-slope ADC is utilized to have quantized data available before the conversion finishes, removing the conventional requirement of an extra delay after the quantizer and compensating for 2 cycles of unit delay [2].

The ‘delsig’ MATLAB toolbox by Schreier [17] provides a vast amount of functions to design loop filters for $\Sigma\Delta$ ADCs. It will be used here to implement the method in [1] to determine the compensated loop filter coefficients to compare the solution to the presented work.

In this paper, we want to present an alternative delay mitigation technique and highlight the differences and similarities between it and the ELD compensation technique from [1] in noise-shaping loops. The proposed method will allow the synthesis of a digital loop filter of equal order as the delay-free loop filter and can mitigate the effect of multiple unit delays and/or high-frequency poles without adding extra hardware. Furthermore, the original poles and zeros of the NTF do not change with respect to the system without parasitic poles or delay. The practical value of our approach is the usage in digital loop filters with tunable coefficients that have processing delay without changing the loop topology itself. Therefore the same hardware can be used.

The paper is organized as follows. Section III presents the filter design procedure, section IV explains the

mitigation method for a single parasitic pole, section V presents simulated results, conclusions are drawn in section VI and Appendix A provides mitigation methods for multiple parasitic poles.

III. FILTER DESIGN AND DELAY

In this section, the calculation of a loop filter from the NTF will be explained. Then, the effect of an added unit delay and a pole to a loop and the resulting NTF is shown.

A. LOOP FILTER FROM NTF

The loop filter with transfer $H(z)$ can be determined starting from the NTF of a closed loop system,

$$H(z) = \frac{1}{\text{NTF}(z)} - 1. \quad (1)$$

To obtain a strictly causal loop filter, there are two constraints to the NTF, namely, equal order of numerator and denominator, as well as a unity factorized gain (the first sample of the impulse response is 1). The reason for these constraints is further explained in [5] section 4.4.1.

B. PARASITIC ELEMENT IN A LOOP

When a loop filter is derived from the NTF without taking parasitic effects, such as delay, in the feedback path into account, the actual system does not behave like the ideal design and might become unstable.

Let us look at the system by examining the resulting NTF of the system when a unit delay is added into the loop. By cascading z^{-1} with $H(z)$ and calculating the NTF using the inverse of (1) to obtain the actually realized NTF_r,

$$\text{NTF}_r(z) = \frac{1}{H(z) \cdot z^{-1} + 1}. \quad (2)$$

Since a unit delay is a pole at the origin, this equation can be written in a more general form to also take any known parasitic pole p_{par} into account,

$$\text{NTF}_r(z) = \frac{1}{H(z) \cdot \frac{K}{z-p_{\text{par}}} + 1}, \quad (3)$$

where $K = 1 - p_{\text{par}}$ to preserve unity gain at DC. Since the poles of the open-loop transfer function are the zeros of the NTF, it is apparent that the parasitic pole p_{par} will appear as an extra zero in the NTF. The poles of the NTF will move with respect to their original locations in the desired NTF, which could push them close to, or out of the unit circle, indicating a marginally stable or unstable system. In the next section, we will address this further with a technique to mitigate this effect.

IV. MITIGATION TECHNIQUE

The goal is to design loop filters that are stable with parasitic elements in the loop without increasing the loop-filter order as this would require changing the hardware. This work proposes a method that includes the parasitic(s) in the NTF design and keeps the original poles and zeros of the NTF at

their original location. With this method, a new open-loop gain can be designed that consists of a cascade of a new loop filter and the parasitic(s) we are trying to compensate for, moreover, the filter can be stable in a closed loop. Since the parasitics are already present in the system, just realizing the new loop filter results in a stable system. The following sections explain this in more detail.

A. DESIGN GOAL

In section III we described the procedure to obtain a loop filter from an NTF and have shown the effect of a parasitic unit delay on the NTF which potentially makes the system unstable. This procedure is also shown schematically in Fig. 2. Starting from the original NTF the loop filter H_{org} is obtained, subsequently a parasitic pole, p_{par} , is added. By transforming this open-loop transfer with parasitic back to the resulting NTF, NTF_r is obtained. NTF_r will contain an extra zero and an extra pole with respect to NTF_{org}. This zero is at the same location as p_{par} because the zeros of the NTF are equal to the poles of the open-loop transfer function.

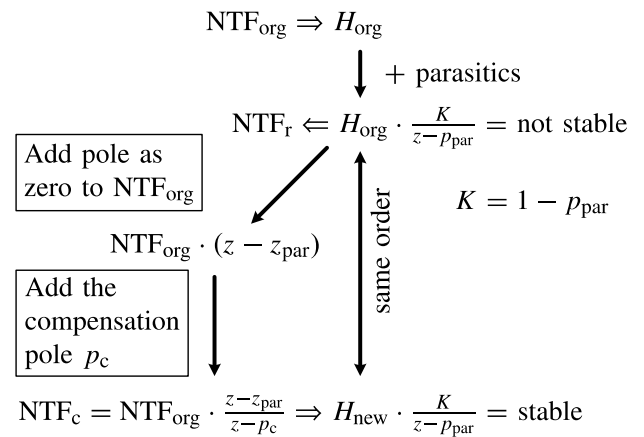


FIGURE 2. Flowchart describing the compensation method.

With this in mind we take the original NTF as a starting point and add p_{par} as a zero (z_{par}) to it as well as a pole p_c of which the value will be determined later. By doing this we aim to obtain a compensated NTF (NTF_c) that after applying (1), yields a filter H_{new} of the same order as H_{org} .

B. FEWER OPEN LOOP ZEROS

We previously described how to obtain a strictly causal loop filter by setting constraints for the NTF. For the proposed mitigation technique it is required to have even fewer open-loop zeros. This is required to be able to separate the parasitic pole p_{par} from the loop filter H_{new} in the final step of Fig. 2 after transforming the NTF. In this way, the filter H_{new} that has to be implemented is obtained.

To reduce the order of the numerator from $N - 1$ to $N - 2$, where N is the order of the filter, the second-highest numerator polynomial coefficient a_{N-1} of the loop filter should also be nullified. This coefficient is a summation of all poles and zeros, which becomes apparent when the numerator term of

the open-loop filter $H(z)$ is simplified to a polynomial,

$$H(z) = \frac{P_{NTF}(z) - Z_{NTF}(z)}{Z_{NTF}(z)}, \quad (4)$$

with,

$$Z_{NTF}(z) = (z - z_N)(z - z_{N-1}) \dots (z - z_1), \quad (5)$$

$$P_{NTF}(z) = (z - p_N)(z - p_{N-1}) \dots (z - p_1). \quad (6)$$

The a_{N-1} coefficient is then calculated as follows,

$$\begin{aligned} a_{N-1} &= -p_1 - p_2 - \dots - p_N + z_1 + z_2 + \dots + z_N, \\ &= \sum_{n=1}^N z_n - \sum_{n=1}^N p_n. \end{aligned} \quad (7)$$

However, a degree of freedom is required to be able to nullify a_{N-1} . This degree of freedom can be found in the value of the pole p_c .

C. MODIFYING THE NTF

To compensate for the parasitic pole p_{par} in the open-loop transfer, it was added as zero z_{par} to NTF_{org} . Consequently, a pole p_c was added to the NTF to satisfy the strict causality requirement. Since this pole location can be chosen freely, we would like to choose it such that we obtain a loop filter with a lower numerator order. This is accomplished by adding z_{par} and p_c to the zeros and poles in (7) respectively. Equating the result to zero and solving for p_c gives,

$$p_c = z_{par} + \sum_{n=1}^N z_n - \sum_{n=1}^N p_n. \quad (8)$$

The compensated NTF is now defined as,

$$NTF_c(z) = NTF_{org}(z) \cdot \frac{z - z_{par}}{z - p_c}. \quad (9)$$

From this equation, we also see that the compensated NTF contains all the poles and zeros of the original NTF, which is a key feature of this method.

D. CALCULATING THE REQUIRED LOOP FILTER

To calculate the open loop transfer the modified NTF (9) is substituted in (1),

$$H_c(z) = \frac{1}{NTF_c(z)} - 1. \quad (10)$$

This $H_c(z)$ contains 2 fewer zeros than poles and because the parasitic pole was introduced as a zero in the NTF it will show as a pole in H_c , so H_c can be written as the combination of a strictly causal filter $H_{new}(z)$ and the parasitic pole,

$$H_c(z) = H_{new}(z) \cdot \frac{K}{z - p_{par}}. \quad (11)$$

Since the pole was already present in the system, the filter H_{new} has to be implemented to get the noise shaping as described by NTF_c . The filter $H_{new}(z)$ is defined as,

$$H_{new}(z) = H_c(z) \cdot \frac{z - p_{par}}{K}, \quad (12)$$

and is of the same order as the original loop filter.

This method can be extended to compensate any number of poles, which is shown in Appendix A, although most practical parasitic problems are solvable with the single-pole method described in this section or possibly the 2-pole method.

V. RESULTS

In this section, the mitigation method will be applied to a filter prototype to show what effect it has on the resulting loop filter and NTF. First, a low-pass loop filter with delay in the loop is designed to illustrate the approach on a familiar filter. It is compared with the result from compensating the loop filter like in [1] by using the toolbox [17], because it has in common that the NTF is modified as a result of the ELD compensation. It is also shown that a similar result can be achieved by manually relaxing the cut-off frequency or stopband attenuation.

After that, the method is used to design a band-pass loop filter with three unit delays in the loop, which is not straightforward to do manually. Finally, the method is applied to compensate for a low-pass pole to show the generality of the method.

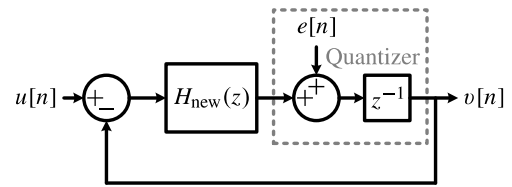


FIGURE 3. $\Sigma\Delta$ modulator with delay in the loop and a loop filter generated using this work.

A. SIMULATION SETUP

First, the systems that are used to perform the simulations are discussed. The system to show the effect of the proposed method in this work is shown in Fig. 3, and implements a loop that consists of a loop filter $H_{new}(z)$ and a quantizer modeled as an additive noise source and a unit delay element.

We examine the limiting case where exactly one unit delay is present in the quantizer, for which in a continuous-time sigma-delta ADC, compensation with a local loop around the quantizer is no longer possible. Hence, a local loop before the quantizer is used for comparison to our proposed technique.

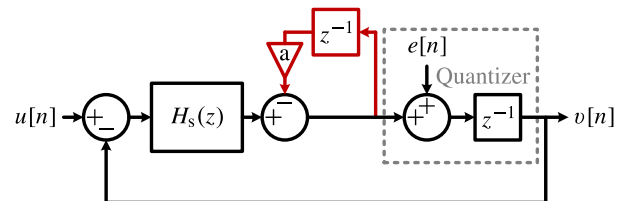


FIGURE 4. $\Sigma\Delta$ modulator with delay in the loop and a loop filter compensated using the local loop from [1].

A comparison will be done to the system in Fig. 4, which implements a similar system using the loop filter $H_s(z)$ and an extra local feedback loop as in [1]. The digital $H_s(z)$ filter

coefficients are obtained as follows using the toolbox [17] to implement the method from Singh *et al.* [1]:

- Realize a CT loop filter from the NTF with 1 cycle of delay:
`realizeNTF_ct(NTF_org, 'FF', [1 2])`
- Obtain the coefficient for the direct path from the state-space D matrix: $a = -D(3)$.
- Calculate the actual NTF for the implementation in Fig. 4:
 $NTF_s = NTF_{org} \cdot (1 - az^{-1})$.
- Obtain the open loop transfer from the NTF and remove the unit delay and the local feedback, $1/(1 + az^{-1})$, from the obtained result to obtain the tuned filter $H_s(z)$:
 $H_s = (1/NTF_s - 1) \cdot z \cdot (1 + az^{-1})$.

B. UNIT DELAY IN A LOW-PASS LOOP FILTER

To design a low-pass loop filter, we start with a high-pass NTF because we want to shape the noise away from the passband of our system. A 4th order filter is designed from a Butterworth high-pass NTF with a corner at $f_s/20$. This NTF is quite aggressive with an out-of-band gain of 3.5 dB. All responses are normalized with $f_s = 1$ Hz. The original NTF (NTF_{org}) is shown in Fig. 5 together with the resultant NTF (NTF_r) after adding a unit delay to the loop and the compensated NTFs: NTF_c and NTF_s . Adding a delay to the aggressive loop filter makes the resulting system unstable which is illustrated by the strong peaking in NTF_r . The compensated NTFs do not have this strong peaking, however, they do have an increased gain in the stopband up to the cut-off frequency and a roll-off in the passband. Comparing the proposed solution to the one provided by the method in [1], a weaker stopband attenuation is achieved, as well as a flatter response in the passband at the expense of an extra first-order filter section.

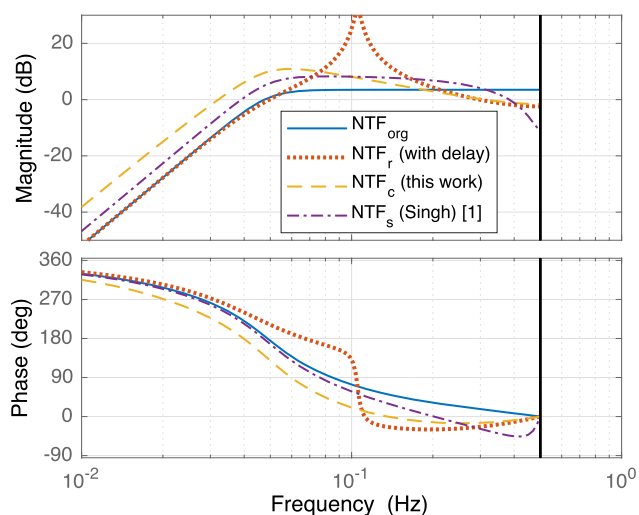


FIGURE 5. High-pass noise transfer with and without an added unit delay and with compensation.

The resultant open-loop transfers are shown in Fig. 6. Observe that the phase margin of the compensated filter has

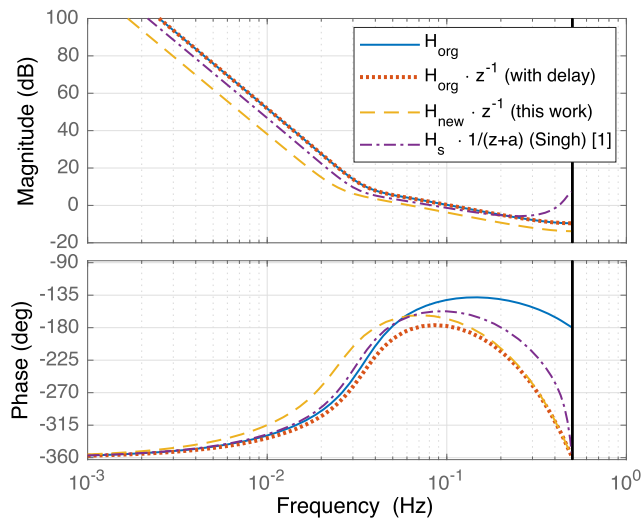


FIGURE 6. Low-pass open loop transfer with and without an added unit delay and with compensation.

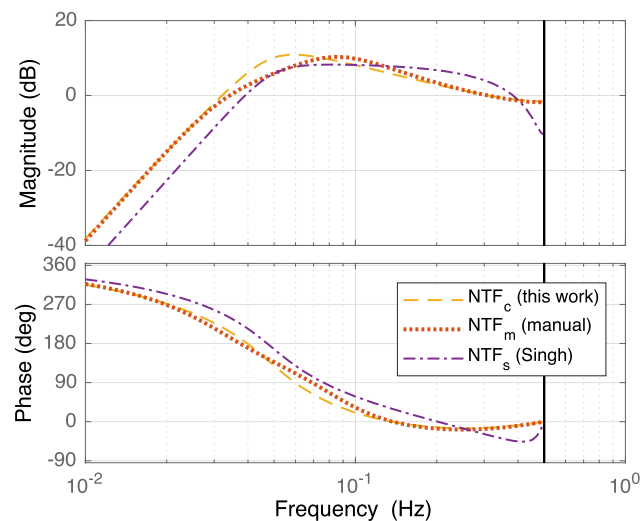


FIGURE 7. High-pass noise transfer with an added unit delay with compensation by the method versus manual tuning and the method in [1].

increased with respect to the loop filter with delay. Looking at the compensated transfer, we notice that both the loop gain and cut-off frequency have decreased.

C. LESS AGGRESSIVE FILTER

After the conclusion of the previous section, the reader might think that when the cut-off frequency and loop gain decrease by applying the proposed method, this could also be done upfront by choosing a less aggressive filter. Indeed, tuning the loop gain or cut-off frequency down by hand can also result in a more stable filter. To show this, a Butterworth filter prototype NTF_m that has an equal low-frequency stopband characteristic as the compensated NTF (NTF_c) is taken to perform a comparison. The result of this comparison is shown in Fig. 7.

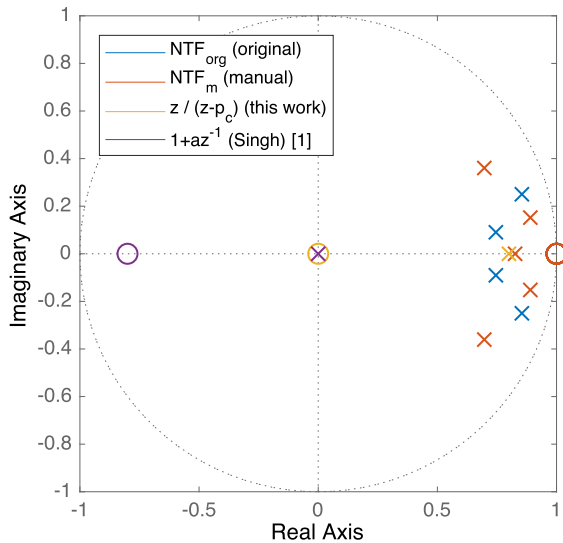


FIGURE 8. Pole zero plot of the prototype NTF, the added pole-zero pairs from the presented method and the method in [1] and the resultant NTF from the manually tuned filter.

The difference in the plot is minimal, however, there is a difference, which can be shown by comparing the poles and zeros of both systems. Fig. 8 shows the poles and zeros of the original and both compensated NTFs. The compensated NTF is a combination of the original NTF (NTF_{org}) with the pole (p_c) added by the compensation method and the zero z_{par} that originates from the unit delay pole p_{par} in the loop filter. The same holds for the pole-zero pair introduced by the method adapted from [1]. The other set of poles and zeros is the resultant NTF (NTF_m) of adding a unit delay to the loop filter that was derived from the manually tuned NTF.

D. SIMULATED RESULTS

Discrete-time simulations are performed for the proposed solution, the solution based on [1] and the manually tuned solution, all three with a unit delay in the loop and a 2-bit quantizer. The original system without delay is also simulated for reference. The results for the in-band noise (IBN), maximum stable amplitude (MSA), and maximum signal-to-noise ratio (SNR) are tabulated in Table 1. A bandwidth of 0 Hz – 0.02 Hz was used with a stimulus at 0.01 Hz.

TABLE 1. Simulated in-band performance: low-pass.

	IBN (dBFS)	MSA	Max SNR (dB)
Original (no delay)	-80.6	0.95	64.3
Singh [1]	-61.4	0.74	54.1
This work	-62.8	0.72	47.1
Manual tuning	-62.8	0.82	47.8

The proposed method, the Singh method, and the manually tuned filter perform comparably, except for the MSA, which is higher for the manually tuned filter, and the maximum SNR, which is 7 dB higher for the Singh method.

Compared to the original filter without delay, all three see a large reduction in IBN and maximum SNR while allowing a unit delay in the loop.

E. SIMILARITIES WITH PRIOR ART

An interesting observation from Fig. 8, is that the distance between the added pole and zero is the same for both the presented work and the method in [1]. The poles and zeros for both solutions adhere to nullifying (7) with the result that the open loop transfer function has 2 fewer zeros than poles. The solutions are dual in how this requirement is met: in the method in [1] the extra NTF pole gets placed at $z = 0$ and the location of the extra zero is derived from (7). In this work, the extra NTF zero is placed at $z = 0$, and the extra pole follows from (8) instead. In equation form both NTFs are related to NTF_{org} as follows,

$$NTF_c = NTF_{org} \cdot \frac{z}{z - p_c},$$

$$NTF_s = NTF_{org} \cdot (1 + az^{-1}),$$

with: $p_c = a = 0.8$ (13)

The modifications required on the open-loop transfers are quite different between the two solutions: In [1], an additional compensation filter (the local feedback before the quantizer) was added to compensate the unit delay (pole at $z = 0$) with a zero at $z = 0$ which effectively moves the open-loop pole to a new location at $z = -a$. In this work, only the coefficients of the noise-shaping filter have to be updated, as the additional NTF zero at $z = 0$ represents the unit-delay that is already in the loop and no additional compensation filter has to be added.

The poles and zeros of both the compensated NTFs still have the Butterworth alignment and are at the same location as in the original NTF apart from the extra compensation pole and zero, while the poles and zeros resulting from the manually tuned NTF do not. This is a key feature of both the proposed method and the method in [1].

F. LIMITS OF THE COMPENSATION METHOD

The presented method to obtain the NTF that results in a stable loop filter will not work for filters with a too high order or too high cut-off frequency. Recall that the compensation pole p_c was calculated using (8) by subtracting the positions of all zeros by the positions of all poles in the filter. If we increase the cut-off frequency, the original poles will move further into the unit circle. As a result of this, the compensation pole p_c moves towards the unit circle and eventually escapes it, resulting in an unstable NTF. This limit does not occur for the method in [1], because the NTF will not become unstable when the zero at $z = -a$ moves out of the unit circle, however, the system does become non-minimum phase in that case.

Similarly, with a higher-order filter, more poles will be placed on the half-circle of the Butterworth formation which increases the sum of the pole locations. This also eventually pushes the compensation pole out of the unit circle.

When compensating for multiple poles, it is less intuitive to see what happens to the compensation poles as the filter becomes more aggressive. The resulting compensation poles are calculated by solving a system of equations as is explained in Appendix A. The roots of these equations can yield complex compensation pole pairs. To obtain a stable compensation, all poles should satisfy the condition $\text{abs}(p_c) \leq 1$, stating that they reside inside the unit circle.

G. DELAY IN A BAND-PASS LOOP FILTER

A band-pass loop filter is designed by taking a 3rd order (6-pole) inverse Chebyshev band-reject NTF with a stopband attenuation of 55 dB in a bandwidth of .015 Hz centered around 1/8 Hz at a normalized sample rate of 1 Hz. Three unit delays are added into the loop, like in the system in Fig. 3 with z^{-3} substituted for z^{-1} , and their effect is mitigated using the presented method. The realization of the compensation obtained from the method in [1] is shown in Fig. 9, which now contains extra local feedback paths before the quantizer with coefficients corresponding to the direct path coefficients.

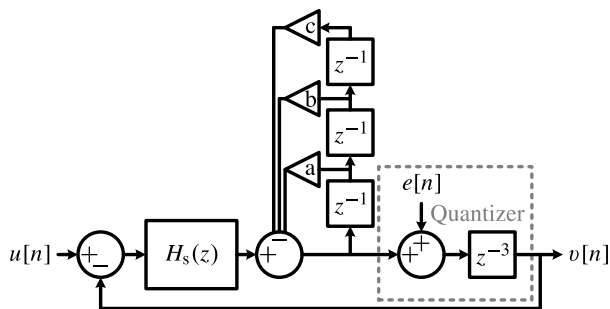


FIGURE 9. Noise shaper with delay in the loop and a loop filter compensated using the toolbox [17] and the method in [1].

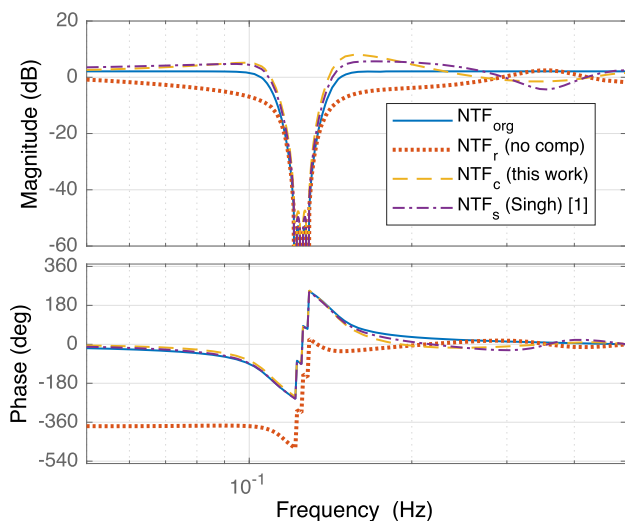


FIGURE 10. Band-reject noise transfer with and without three added unit delays and with compensation.

The desired NTF is shown in Fig. 10, together with the resulting NTFs after adding three unit delays in the loop. The corresponding open-loop transfers are shown in Fig. 11.

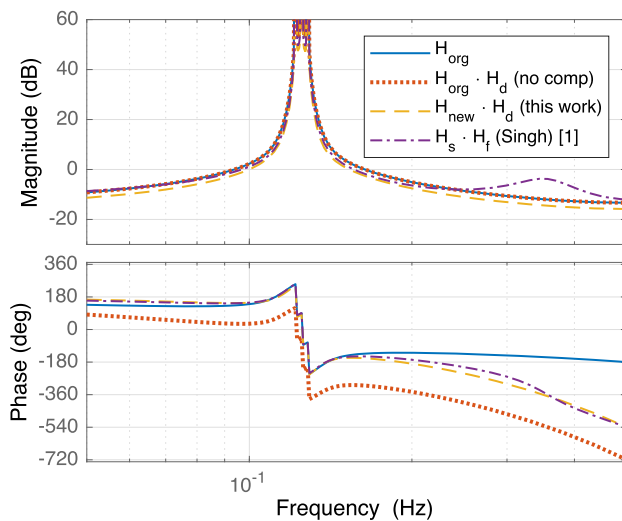


FIGURE 11. Band-pass open loop transfer with and without three added unit delays and with compensation.

H_d is used to denote the three unit delays and H_f is the transfer of the local feedback combined with the delays: $1 / (z^3 + az^2 + bz + c)$. Adding delay to the loop filter renders the closed-loop system unstable. However, both compensation methods realize a stable system by matching the phase of the compensated loop gains to the phase of the original filter.

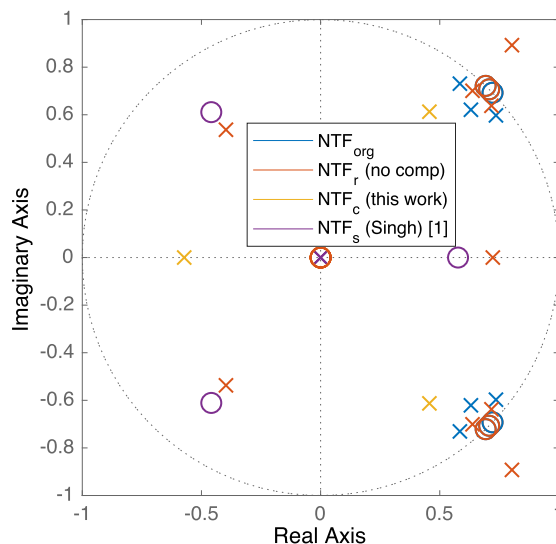


FIGURE 12. Pole zero plot of the prototype NTF, the added compensation poles and zeros and the resultant NTF from adding three unit delays.

From Fig. 12 it can be observed that the compensated NTFs have the desired poles at the exact same location as the prototype NTF, with the addition of the three compensation poles and zeros for the three unit delays. It is worth mentioning that both methods produce an NTF that satisfies the conditions that the second (7), third (17) and fourth (25) numerator coefficient of the open-loop transfer are nullified.

This shows that both solutions share a common solution space where the proposed solution has a minimal realization and the prior art gives better noise shaping performance. Intermediate solutions where both the extra poles and zeros in the NTF are not in the origin have been investigated but did not yield better results than the two presented methods.

The filters have been simulated using a discrete-time Matlab Simulink model resembling Fig. 9 with a single bit quantizer to show the effect of the added non-ideality on the performance of the modulator. Table 2 lists the IBN without any input, the MSA with a stimulus at 1/8 Hz, and the maximum SNR. Both the presented work and the method in [1] see a more than 10 dB drop in IBN compared to the ideal modulator, with the presented work performing only 2.6 dB worse than the prior art. The SNR shows a similar trend with a 5.4 dB drop compared to the prior art and the MSA is .06 lower than the prior art.

TABLE 2. Simulated in-band performance: band pass.

	IBN (dBFS)	MSA	Max SNR (dB)
Original (no delay)	-91.1	0.99	81.6
Singh method [1]	-80.7	0.91	73.8
This work	-78.1	0.85	68.4

Just like in the low-pass example from the previous section, manual tuning of the gain of the original NTF can be tried to design a stable closed-loop filter. For this case, the manually tuned NTF was designed by using a lower stopband attenuation factor which is equal to the attenuation in the compensated NTF. However, this manual tuning method does not yield stable results for band-pass filters.

The reason for this can be observed in the pole-zero plot in Fig. 12, which shows the pole and zero locations of the resulting NTFs when a delay is added to the loop. In the uncompensated filter, one complex pole pair is still outside the unit circle, showing that it is unstable. The manually tuned filter will also have a complex pole pair outside of the unit circle.

H. LOW-PASS POLE IN A BAND-PASS LOOP FILTER

A more generic case is to compensate a pole rather than a unit delay. In this section, an example is given to compensate a parasitic pole in the loop. Such a pole could originate from analog bandwidth limitations. The characterization of this pole in a system is outside of the scope of this theoretical analysis, hence this section will solely focus on the compensation of the pole.

A band-pass loop filter is designed by taking a 3rd order inverse Chebyshev band-reject NTF with a stopband attenuation of 55 dB in a bandwidth of 0.005 Hz centered around 1/16 Hz at a sample rate of 1 Hz. The added parasitic low-pass pole in the feedback path is located at 0.11 Hz. The original, uncompensated, and compensated NTFs are shown in Fig 13.

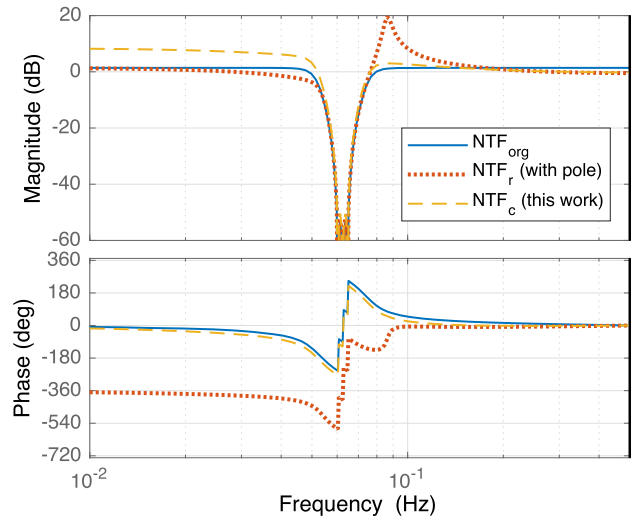


FIGURE 13. Band-reject noise transfer with and without an added pole and with compensation.

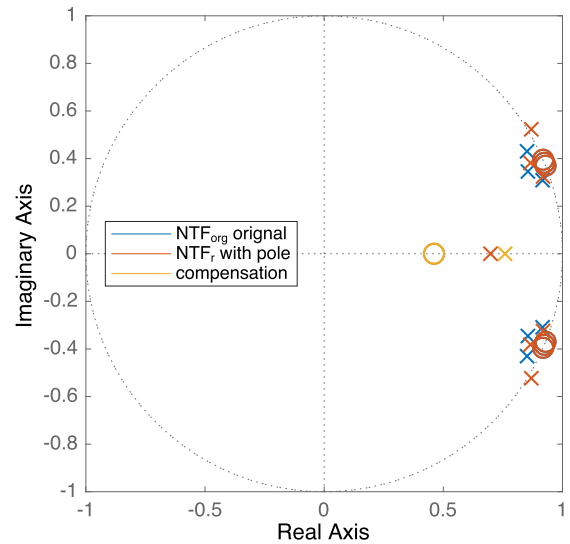


FIGURE 14. Pole zero plot of the prototype NTF, the added compensation poles and zeros and the resultant NTF with the parasitic.

The NTF resulting from the parasitic pole (NTF_r) shows peaking, while the NTF with the mitigation technique applied has a 7 dB increased passband gain at frequencies below the stopband. Looking at the pole-zero plot in Fig. 14, it is seen that NTF_r with the parasitic pole is unstable, just like the filter with the three delays from the previous section. It is also apparent that the NTF zero (z_{par}) originating from the pole (p_{par}) in the loop is not in the center of the unit circle as was the case for the delay. Hence, compensation of a pole becomes increasingly difficult as the NTF pole location p_c calculated by (8) will be offset towards 1. This amount is set by the NTF zero that originates from the parasitic pole in the loop. The lower the parasitic pole frequency, the further the compensation NTF pole is pushed towards 1, eventually making compensation impossible.

I. SPREAD OF THE LOW-PASS POLE

In an actual system, the parasitic pole can be in the analog domain. This means that the component that causes this pole can be affected by process spread. In this section, we investigate if the compensation method is robust against the resulting spread in the parasitic pole location.

The NTF prototype of the previous section is taken again and the same pole is compensated for. However, for the actual implementation we keep the designed filter H_{new} , but take a different pole in the feedback path that has a $\pm 20\%$ deviation with respect to the original pole p_{par} . From that open-loop transfer function, the resulting NTF is calculated. The resulting pole-zero plot is shown in Fig. 15. Observe that the Chebyshev poles have shifted from their original location, but that the NTFs have not become unstable. Furthermore, it can be seen that when the zero (z_{par}) at the location of the parasitic pole moves towards 1, then p_c also moves towards 1. The closer the frequency of the parasitic pole is to the desired filter, the smaller the headroom for spread becomes before the filter becomes unstable.

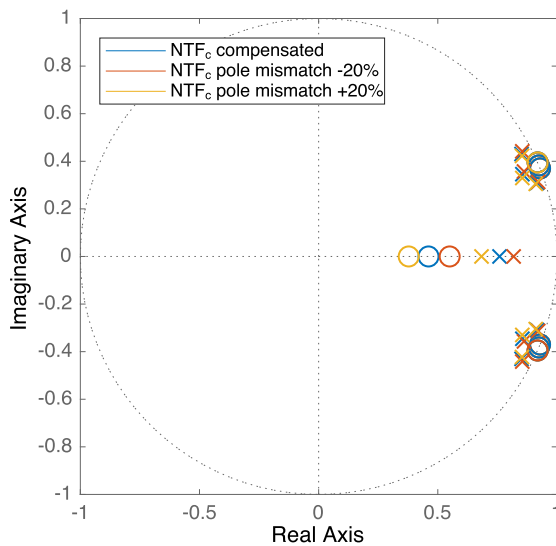


FIGURE 15. Pole zero plot of the compensated NTF and the resultant NTFs from having a deviation of the actual parasitic.

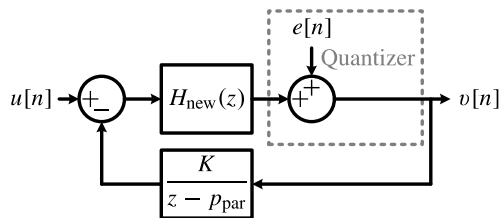


FIGURE 16. $\Sigma\Delta$ modulator with delay in the loop and a loop filter generated using this work.

The effect of the spread in the parasitic pole on modulator performance has been evaluated through simulation of

TABLE 3. Simulated in-band performance of the band-pass modulator with mismatch between the actual and compensated parasitic low-pass filter.

	IBN (dBFS)	MSA	Max SNR (dB)
Original (no parasitic)	-106.3	1.00	93.7
This work (nominal)	-91.1	0.91	87.8
This work ($f_{pole} +20\%$)	-92.2	0.86	89.0
This work ($f_{pole} -20\%$)	-95.0	0.90	86.6

the schematic in Fig. 16 with a 2-bit quantizer and K as $1 - p_{par}$ to have a unity-gain at DC for the feedback pole p_{par} . A sinusoidal stimulus at the center of the band was used like in the previous simulation to obtain the results shown in Table 3. Stable operation is obtained in the nominal case and the $\pm 20\%$ cases with very minor differences in the IBN, MSA, and maximum SNR. Compared to the original filter, the maximum SNR is reduced by at least 4 dB, the MSA drops by 9%, and the IBN increases by 15 dB.

VI. CONCLUSION

In this paper, an analytical technique to design stable loop filters with parasitic poles in their feedback paths was presented. A step-by-step method has been given to get from the desired NTF and a parasitic pole to an implementable stable loop filter. The method does not require any additional feedback paths or additional filter orders. Instead, it adapts the open-loop filter dynamics such that the closed-loop poles and zeros remain at their original location in the presence of the parasitic. Next to the original closed-loop poles and zeros, the parasitic does introduce an additional pole-zero pair in the NTF, which results in a performance penalty compared to the desired NTF. Moreover, the closed-loop filter can still become unstable if the parasitic is too dominant for the desired NTF. The method has been applied to both low-pass and band-pass loop filter design methods to show its versatility.

The presented method has been compared to a modified ELD compensation method based on the work in [1], referred to as the ‘Singh method’. It was presented which conditions the NTF has to adhere to such that the loop-filter derived from it contains $N + 1$ fewer zeros than poles, where N is the number of unit delays. For a single unit delay, the constraint is that the sum of the poles of the NTF should equal the sum of the zeros. This implies that the distance between the pole and zero that are added to the original NTF is constrained and irrespective of the chosen compensation method. And indeed this distance was found to be the same for both methods, and that it equals the coefficient for the direct path in classical ELD compensation. In summary, both solutions share the same solution space, where the presented method favors a minimal realization using the existing hardware with no required extra hardware and the prior art favors performance at the expense of extra filter orders or direct paths.

**APPENDIX A
MITIGATION OF MULTIPLE POLES**

Section III has shown the mitigation technique for a single parasitic pole. The extension to mitigate two, three, and four poles is shown in this appendix.

A. METHOD FOR TWO POLES

When two poles (p_{par1}, p_{par2}) are in the loop, two zeros (z_{par1}, z_{par2}) are added to the NTF, resulting in the requirement of two extra poles (p_{c1}, p_{c2}) as well. These two poles can be calculated as such that $H_c(z)$ contains 3 fewer zeros than poles. In order to do this the a_{N-2} coefficient in the numerator polynomial of $H_c(z)$ needs to be nullified in addition to the a_{N-1} and a_N coefficients. The coefficient a_{N-2} consists of a summation of unique products of pole pairs and zero pairs,

$$a_{N-2} = p_1 p_2 + \dots + p_{N-1} p_N - (z_1 z_2 + \dots + z_{N-1} z_N),$$

$$= \sum_{n=1}^{N-1} p_n \left(\sum_{m=n+1}^N p_m \right) - \sum_{n=1}^{N-1} z_n \left(\sum_{m=n+1}^M z_m \right), \quad (14)$$

which will be simplified to (15) for conciseness,

$$a_{N-2} = \sum P_{pairs} - \sum Z_{pairs}. \quad (15)$$

By equating the condition for one pole in (7) to 0 we can derive the first condition that has to be satisfied,

$$p_{c1} + p_{c2} + \sum_{n=1}^N p_n = z_{par1} + z_{par2} + \sum_{n=1}^N z_n. \quad (16)$$

The second condition is derived from (15),

$$p_{c1} \cdot p_{c2} + (p_{c1} + p_{c2}) \sum_{n=1}^N p_n + \sum P_{pairs} = \sum Z_{pairs}. \quad (17)$$

The constraints in (16) and (17) have to be solved first for $p_{c1} + p_{c2}$ and then for $p_{c1} \cdot p_{c2}$. By taking $p_{c1} + p_{c2} = a$ and $p_{c1} \cdot p_{c2} = b$ we can solve for p_{c1} ,

$$p_{c1} = a - p_{c2} = a - b/p_{c1}, \quad (18)$$

$$p_{c1}^2 = a p_{c1} - b, \quad (19)$$

$$p_{c1}^2 - a p_{c1} + b = 0. \quad (20)$$

Now, a and b can be derived from (16) and (17),

$$a = \sum_{n=1}^N z_n + z_{par1} + z_{par2} - \sum_{n=1}^N p_n, \quad (21)$$

$$b = \sum Z_{pairs} - a \sum_{n=1}^N p_n - \sum P_{pairs}. \quad (22)$$

Solving (20) will yield two solutions for p_{c1} , with the second solution automatically being p_{c2} . Now we can write compensated NTF as follows,

$$NTF_c(z) = NTF(z) \cdot \frac{(z - z_{par1})(z - z_{par2})}{(z - p_{c1})(z - p_{c2})}. \quad (23)$$

After transforming NTF_c to $H_c(z)$ using (1), the filter that has to be implemented can be calculated,

$$H_{new}(z) = H_c(z) \cdot \frac{K_1}{z - p_{par1}} \cdot \frac{K_2}{z - p_{par2}}, \quad (24)$$

where $K_1 = 1 - p_{par1}$ and $K_2 = 1 - p_{par2}$.

B. METHOD FOR MORE POLES

Instead of showing the full derivation, we show the equations that have to be solved. In the case of a system with three parasitic poles, the following open loop numerator coefficient should be nullified as well,

$$a_{N-3} = \sum Z_{triplets} - \sum P_{triplets}. \quad (25)$$

By adding this constraint to the prior constraints for two poles, mitigating the effect of three open loop poles can be achieved by using the solutions of (26) as extra poles in the NTF,

$$p_c^3 - a \cdot p_c^2 + b \cdot p_c - c = 0, \quad (26)$$

with,

$$a = \sum_{n=1}^N z_n + \sum_{m=1}^M z_{par,m} - \sum_{n=1}^N p_n, \quad (27)$$

$$b = \sum Z_{pairs} - a \sum_{n=1}^N p_n - \sum P_{pairs}, \quad (28)$$

$$c = \sum Z_{triplets} - b \sum_{n=1}^N p_n - a \sum P_{pairs} - \sum P_{triplets}. \quad (29)$$

where M in (27) is the number of added zeros, which is three in this case. The ‘triplets’ summation is the sum of products of unique pole/zero triplets.

Likewise for four poles the equation that has to be solved is,

$$p_c^4 - a \cdot p_c^3 + b \cdot p_c^2 - c \cdot p_c + d = 0, \quad (30)$$

with a, b and c as in (27), (28) and (29) respectively and,

$$d = \sum Z_{quads} - c \sum_{n=1}^N p_n - b \sum P_{pairs} - a \sum P_{triplets} - \sum P_{quads}. \quad (31)$$

The ‘quads’ summation is the sum of products of unique pole/zero quadruplets.

An extension to even more parasitic poles is possible by extending the equations with quintuplets, sextuplets, etc.

REFERENCES

[1] V. Singh, N. Krishnapura, and S. Pavan, “Compensating for quantizer delay in excess of one clock cycle in continuous-time $\Delta\Sigma$ modulators,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 9, pp. 676–680, Sep. 2010.

- [2] Y. Hu, H. Venkatram, N. Maghari, and U.-K. Moon, "A continuous-time $\Delta\Sigma$ ADC utilizing time information for two cycles of excess loop delay compensation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 11, pp. 1063–1067, Jul. 2015.
- [3] C. E. Lokin, R. A. R. van der Zee, D. Schinkel, and B. Nauta, "EMI reduction in class-D amplifiers by actively reducing PWM ripple," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 765–773, Mar. 2020.
- [4] (2016). *Digital Audio Converter and Amplifier Controller*. Axign B.V., Datasheet. [Online]. Available: <https://www.axign.nl>
- [5] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*. Piscataway, NJ, USA: IEEE Press, 1997.
- [6] S. Pavan, R. Schreier, and G. C. Temes, *Understanding Delta-Sigma Data Converters*, R. J. Baker, Ed., 2nd ed. Piscataway, NJ, USA: IEEE Press, 2017.
- [7] P. Benabes, M. Keramat, and R. Kielbasa, "A methodology for designing continuous-time sigma-delta modulators," in *Proc. Eur. Design Test Conf. (ED&TC)*, Mar. 1997, pp. 46–50.
- [8] S. Loeda, H. M. Reekie, and B. Mulgrew, "On the design of high-performance wide-band continuous-time sigma-delta converters using numerical optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 4, pp. 802–810, Apr. 2006.
- [9] S. Pavan, "Excess loop delay compensation in continuous-time delta-sigma modulators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 11, pp. 1119–1123, Nov. 2008.
- [10] J. A. Cherry and W. M. Snelgrove, "Excess loop delay in continuous-time delta-sigma modulators," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 4, pp. 376–389, Apr. 1999.
- [11] J. Guo and M. E. Magaña, "Compensation method of the excess loop delay in continuous-time delta-sigma analog-to-digital converters based on model matching approach," *IET Circuits, Devices Syst.*, vol. 10, no. 1, pp. 29–36, Jan. 2016.
- [12] M. Keller, A. Buhmann, J. Sauerbrey, M. Ortmanns, and Y. Manoli, "A comparative study on excess-loop-delay compensation techniques for continuous-time sigma-delta modulators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3480–3487, Dec. 2008.
- [13] K. El-Sankary, H. Alamdari, and E. El-Masry, "An adaptive ELD compensation technique using a predictive comparator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 8, pp. 619–623, Aug. 2009.
- [14] P. Fontaine, A. N. Mohieldin, and A. Bellaouar, "A low-noise low-voltage CT $\Delta\Sigma$ modulator with digital compensation of excess loop delay," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2005, pp. 498–500.
- [15] A. Yahia, P. Bénabès, and R. Kielbasa, "Bandpass delta-sigma modulators synthesis with high loop delay," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, May 2001, pp. 344–347.
- [16] F. Jiang, C. Chen, Y. Xiao, J. Xu, and J. Ren, "Beyond-one-cycle loop delay CT $\Delta\Sigma$ modulators with proper rational NTF synthesis and time-interleaved quantizers," in *Proc. Midwest Symp. Circuits Syst.*, vol. 1, Aug. 2013, pp. 558–561.
- [17] R. Schreier. (2020). *Delta Sigma Toolbox*. MATLAB. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox>



CHRISTIAAN E. LOKIN (Student Member, IEEE) received the B.Sc. and M.Sc. degrees (*cum laude*) in electrical engineering from the University of Twente, Enschede, The Netherlands, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Integrated Circuit Design Group. In 2020, he joined Axign, Enschede, where he works on class-D audio amplifiers. His current research interests include mixed-signal systems and control systems for applications in class-D power amplifiers.



DANIËL SCHINKEL (Member, IEEE) received the M.Sc. degree (*cum laude*) in electrical engineering and the Ph.D. degree from the University of Twente, Enschede, The Netherlands, in 2003 and 2011, respectively. From 2003 to 2007, he carried out his Ph.D. research at the IC-Design Group, headed by Bram Nauta. He is one of the founders of Axiom IC, an IC-design company that started in Enschede, in 2007, and which focused on the design of data-converters and mixed-signal systems. In 2013, Axiom IC was acquired by Teledyne DALSA. Since 2013, he has been active as an independent research consultant, working for various IC companies as well as for the university. He holds four patents and has authored or coauthored about 25 papers. His current research interests include analog, digital, and mixed-signal circuit design, sigma-delta data converters, class-D power amplifiers, and high-speed communication circuits.



RONAN A. R. VAN DER ZEE (Member, IEEE) received the M.Sc. degree (*cum laude*) in electrical engineering and the Ph.D. degree in high-efficiency audio amplifiers from the University of Twente, Enschede, The Netherlands, in 1994 and 1999, respectively. In 1999, he joined Philips Semiconductors, where he worked on class AB and class D audio amplifiers. In 2003, he joined the IC-Design Group, University of Twente. His research interests include linear and switching power amplifiers, RF frontends, and ultralow-power radio.



BRAM NAUTA (Fellow, IEEE) was born in Hengelo, The Netherlands, in 1964. He received the M.Sc. degree (*cum laude*) in electrical engineering and the Ph.D. degree in analog CMOS filters for very high-frequencies from the University of Twente, Enschede, The Netherlands, in 1987 and 1991, respectively. In 1991, he joined Mixed-Signal Circuits and Systems Department, Philips Research, Eindhoven, The Netherlands. In 1998, he returned to the University of Twente, where he is currently a Distinguished Professor and heading the IC Design Group. Since 2016, he has been serving as the Chair for the EE Department, University of Twente. His current research interests include high-speed analog CMOS circuits, software-defined radio, cognitive radio, and beamforming.

Prof. Nauta is a member of the Royal Netherlands Academy of Arts and Sciences (KNAW). He was in the Technical Program Committee of the Symposium on VLSI Circuits, from 2009 to 2013. He is in the Steering Committee and the Programme Committee of the European Solid State Circuit Conference (ESSCIRC). He served as the President for the IEEE Solid-State Circuits Society, from 2018 to 2019. He was a co-recipient of the ISSCC 2002 and 2009 "Van Vessel Outstanding Paper Award." In 2014, he received the 'Simon Stevin Meester' Award (500.000€), the largest Dutch national prize for achievements in technical sciences. He was the 2013 Program Chair of the International Solid State Circuits Conference (ISSCC). From 2007 to 2010, he served as the Editor-in-Chief for the IEEE JOURNAL OF SOLID-STATE CIRCUITS (JSSC). He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, from 1997 to 1999, and IEEE JOURNAL OF SOLID-STATE CIRCUITS (JSSC), from 2001 to 2006. He served as distinguished lecturer for IEEE.

• • •