Analog Integrated Circuits and Signal Processing (2018) 96:495–507
https://doi.org/10.1007/s10470-018-1131-7

Analysis of switched-capacitor circuits using driving-point signal-flow graphs

Hanspeter Schmid[1] · Alex Huber[1]

Received: 12 January 2018 / Accepted: 13 February 2018 / Published online: 1 March 2018

Abstract
This paper extends the driving-point signal-flow graphs to switched-capacitor (SC) circuits by introducing a new theoretical element: an auxiliary voltage source that transfers no charge. In contrast to existing SFG methods, our method has no restrictions as to what types of SC circuits can be analysed, it requires no equivalent circuits or tables, and it works with two-phase as well as multi-phase SC circuits of any complexity. Compared to charge-equation matrix methods, it requires more effort, but is better suited for hand analysis because it makes causal relationships visible. Three illustrative examples are given to show the efficiency of the method and present a few application hints: a voltage doubler, the standard SC integrator, and a four-phase circuit simulating an inductor.

Keywords Switched-capacitor circuits · Signal-flow graph · Driving-point impedance · Multi-phase SC networks · Circuit analysis

1 Introduction

Switched-capacitor (SC) circuits are analog discrete-time circuits that consist of switches, capacitors, and amplifiers. They are as diverse and powerful as they are hard to analyse. This is mainly so because SC circuits *reconfigure* between phases, so they will not have the same topology, nor even the same number of nodes, in the different phases.

Signal-flow graphs (SFGs) are often used for hand analysis of circuits (and for teaching), because they make causal relationships visible and provide insight into a circuit even before transfer functions are calculated [1–4]. While the SFG analysis of arbitrary continuous-time circuits is solved, there is no SFG method yet that can be applied to all SC circuits without restrictions on their structure or number of phases. In this paper we propose such an SFG method, compare it to previous SFG method to highlight the improvement, and compare it to the established charge-equation method to show that it describes the same equation systems in a different way.

The DPSFG was introduced in 1998 [5] when Ochoa combined driving-point impedance techniques with Mason signal-flow graphs (SFGs) [6, 7]. Ochoa used auxiliary voltage generators (which we call auxiliary voltage sources or *aux sources*) and explained the SFG derivation by first splitting schematics into sub-circuits and then coupling the sub-circuits with voltage-controlled current sources.

We cast Ochoa's method into a different cognitive framework in [8], making it unnecessary to resort to intermediate circuit representations and simplifying the application of the method to the point where no written material or tables at all are required to use it. This way of explaining the method followed closely Mason's original idea to use SFGs because they "offer a visual structure, a universal graph language, a common ground upon which causal relationships among a number of variables may be laid out and compared [7]". The one-to-one correspondence between circuit and graph also made it straightforward to derive transposed circuits, as shown in [8].

Concerning SC networks, two groups devised and published general analysis methods in parallel. The first publication used the concepts of nodal analysis on SC circuits

✉ Hanspeter Schmid
hanspeter.schmid@fhnw.ch

Alex Huber
alex.huber@fhnw.ch

1 Institute of Microelectronics, University of Applied Sciences and Arts NW Switzerland, Steinackerstrasse 5, 5210 Windisch, Switzerland

and also extended classical two-port theory to SC circuits [9]. In the second publication, the derivation of the matrices is simpler, because it uses a switch matrix that is straightforward to derive, but since the author does not use the z transform, but only discrete-time equations with time indices, the derivations do not result in z transfer functions [10].

The nodal analysis was then further simplified by using the indefinite admittance matrix (IAM) in [11] and was extended to the analysis of multiphase SC networks [12]. Since there is a very close correspondence between signal-flow graphs and IAMs, SFGs started to be used at that point in time because, by making causal relationships and signal flow visible, authors gained more insight into the circuits.

The first comprehensive paper about SFG analysis of SC circuits was published in 1984 [13], it bases on the theory introduced in [11] and makes it possible to easily assemble signal-flow graphs of SC filters by first tabulating basic building blocks and the corresponding graph parts. This method was extended to multiphase SC networks [14].

Both [13] and [14] succeed in making the signal flow visible, but the graphs are not one-to-one maps of the circuits. Furthermore, the methods only work for source-sink-node (SSN) networks [14], which means that all nodes are either at a voltage source or at an amplifier input. Many circuits, e.g., those doing correlated double sampling and voltage multiplication, are therefore not covered.

These methods assume from the start that there are no parasitic capacitors and that the gain of the op-amps used in the SC circuits is infinite. They cannot possibly analyse the effect of finite opamp gain on the transfer functions, which is quite important, as was, e.g., discussed in [15] for SC integrators.

This was remedied in Chichocki et al. [16] who made it possible to analyse *any* SC circuit, with non-ideal op-amps and also parasitic capacitors. They used a small number of equivalent z-domain circuits and the Coates flow-graph technique. In their own words, they “require a smaller number of elements than the equivalent circuits previously proposed”. But they still require equivalent circuits containing some additional elements.

In a previous paper we applied DPSFGs to the noise analysis in SC circuits, [17]. In this paper, we introduce a DPSFG analysis method for SC circuits that requires no equivalent circuits whatsoever and imposes no restrictions on amplifier gains, parasitic capacitors, or the number of clock phases. What results is a general method for visual-based (i.e., intuitive) analysis of switched-capacitor networks of any complexity, with any number of phases.

As in all cited papers, the theory is simple (here it is the introduction of a new type of auxiliary voltage source). We want to show that this one intuitive understanding—the understanding of that new aux source—makes it possible to analyse continuous-time and switched-capacitor circuits with the same SFG technique. The purpose of this paper is to extend and unify SFG analysis and enable SFG users to tackle all SC circuits as easily as they can treat continuous-time circuits.

Therefore, we proceed as follow: In Sect. 2, we give a short introduction to SFGs and DPSFGs of continuous-time circuits. Our explanation complements [4, 5, 8]; it is structured such that the introduction of the new theoretical element is as simple as possible in Sect. 3, where we also give a minimal example. Section 4 contains the main example in which we show how to efficiently apply our method. Section 5 extends this to multi-phase networks and shows how to calculate a discrete-time input impedance. Finally, in Sect. 6 we show what to do when graphical evaluation becomes too complicated.

2 Driving-point signal-flow graphs (DPSFG) of continuous-time circuits

This section presents the DPSFG method introduced in [5], brought to its present form in [8], and provided as a sequence of video tutorials in [4].

2.1 Signal-flow graphs and Mason's gain rule

Mason graphs are just graphical representations of linear equation systems. The so-called *branches* are multipliers, and the so-called *nodes* are variables. All branches going into a node are added. Therefore, each node that has branches going into it describes a linear equation, and the example shown in Fig. 1 represents the following linear equation system:

$$
\begin{aligned}
v &= au + fw + gx,\\
w &= bv,\\
x &= cw,\\
y &= ku + dx + hz,\\
z &= ey.
\end{aligned}
\tag{1}
$$

This can also be expressed in matrix notation:

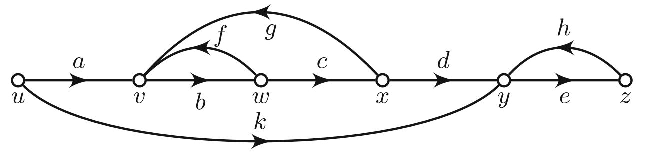

Fig. 1 Example of a signal-flow graph (SFG) with nodes u, v, w, x, y, z and path weights a, b, c, d, e, f, g, h, k. This graph represents a linear equation system with five dependent variables, v, w, x, y, z, five equations, and one independent variable u

$$\begin{pmatrix} v \\ w \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cdot & f & g & \cdot & \cdot \\ b & \cdot & \cdot & \cdot & \cdot \\ \cdot & c & \cdot & \cdot & \cdot \\ \cdot & \cdot & d & \cdot & h \\ \cdot & \cdot & \cdot & e & \cdot \end{pmatrix} \begin{pmatrix} v \\ w \\ x \\ y \\ z \end{pmatrix} + \begin{pmatrix} a \\ \cdot \\ \cdot \\ k \\ \cdot \end{pmatrix} u, \qquad (2)$$

where we have simply replaced all numbers 0 by $\cdot$ such that it becomes immediately apparent that this graph corresponds to a sparse-matrix equation system.

It would be equally simple to take any matrix equation, e.g., an IAM representation of a circuit, and draw it as a Mason graph, but that would not necessarily be helpful as it would then not show the causal relationships.

Equation (2) could now be solved by matrix manipulations, but Mason showed a better way in [7]: A transfer function from a node u to a node z can be calculated as

$$T = \frac{z}{u} = \frac{\sum_i P_i \Delta_i}{\Delta}. \qquad (3)$$

The graph determinant Δ can be calculated as

$$\Delta = 1 - \sigma_1 + \sigma_2 - \sigma_3 + \cdots = 1 - \sum_1^{\infty} (-1)^i \sigma_i, \qquad (4)$$

where σ_1 is the sum of all loops, σ_2 is the sum of products of 2 loops that do not have nodes in common, and σ_3 is the sum of products of 3 loops neither of which have nodes in common, and so on. The sum (4) can, in principle, go to σ_N in a system with N loops, but most feedback systems derive their special properties from interacting loops, so the sum often ends soon in practice. In more technical terms, whenever a matrix representation of the equation system would be sparse, then Mason's formula is efficient.

In Fig. 1, we have three loops:

$$L_1 = bf, \quad L_2 = bcg, \quad L_3 = eh. \qquad (5)$$

L_1 and L_2 have nodes in common, but L_3 touches neither of the other loops, so (4) becomes

$$\Delta = 1 - L_1 - L_2 - L_3 + L_1 L_3 + L_2 L_3. \qquad (6)$$

The P_i in the numerator of (3) are the forward paths, the possible ways to get from u to z. The Δ_i are their respective sub-determinants, which are calculated just like Δ, but only using loops that have no nodes in common with the forward path in question.

In Fig. 1, we have two forward paths:

$$P_1 = abcde, \quad P_2 = ke. \qquad (7)$$

Path P_1 touches all loops, so when calculating the sub-determinant, all terms containing loops in (6) disappear and $\Delta_1 = 1$. Path P_2 only touches loop L_3, so all terms containing L_3 in (6) have to be omitted to calculate $\Delta_2 = 1 - L_1 - L_2$.

Inserting everything into (3) results in

$$T = \frac{abcde + ke(1 - bf - bcg)}{1 - bf - bcg - eh + befh + bcegh}. \qquad (8)$$

Note that if a different transfer function has to be calculated (e.g., $T_y = y/u$), then only the numerator can change, because Δ does not depend on the forward paths.

Mathematically, an SFG is just a graphical representation of an equation system, which can even be manipulated and simplified graphically [6]. But as Mason stated there, it can also be used to visualise causal relationships—if it is drawn in the right way.

2.2 The driving-point signal-flow graph (DPSFG)

The most important aspect of a signal-flow graph that is to be used for hand analysis is that it maintains and represents causal relationships. The common tool used in literature is source superposition, as it is also done when circuit equations are derived from circuits.

Source superposition will give individual contributions from all sources to all circuit nodes for the condition that all other sources are set to zero. So unless there is a voltage source in every node of a circuit, it is necessary to tabulate all possible networks that could reasonably appear between sources and sinks, which invariably means restricting methods to certain classes of circuits, or to use another analysis method to derive the individual contributions.

As explained in [8], the DPSFG method solves this problem by making sure there is a voltage source at *every* circuit node. In order to achieve this, the first step in circuit analysis is to introduce an auxiliary source (aux source) at every node that does not yet have a voltage source. This aux source is defined as follows: *It produces the voltage that is already present at a node, and therefore conducts no current.*

To give an example, Fig. 2(a) shows the standard continuous-time integrator with a resistor having conductance G and a capacitor having capacitance C, and an opamp having gain A. Node 1 is connected to the input source, node 3 to the opamp output, which is also a voltage source, but there is no voltage source on node 2. Therefore an aux source is connected to node 2, resulting in Fig. 2(b).

To draw the DPSFG, we can first draw all nodes. Observe that we also draw the current flowing into the aux source, I_2, in Fig. 3(a).

Then we simply do source superposition, voltage by voltage. V_{in} controls V_1, and we have Fig. 3(b). V_1 lets a current $I_2 = G\,V_1$ flow, which gives Fig. 3(c). V_2 controls the opamp output as $V_3 = -A\,V_2$, giving Fig. 3(d). V_3 lets a current $I_2 = sC\,V_3$ flow, and it also is the same as V_{out}, and we have Fig. 3(e).

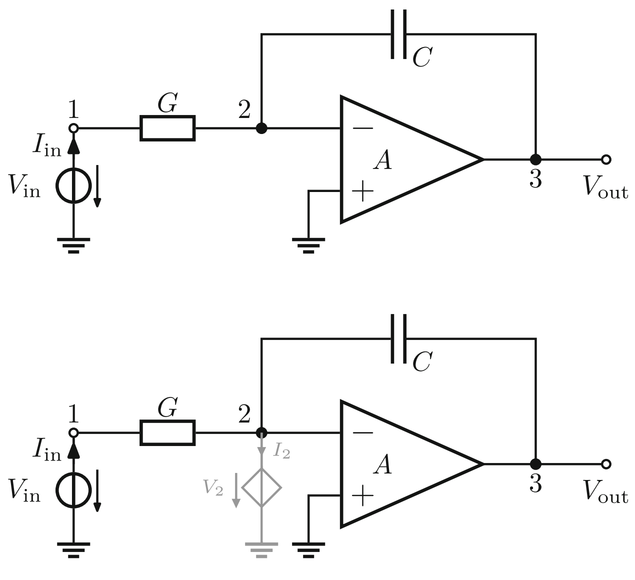

Fig. 2 Continuous-time integrator without and with aux source

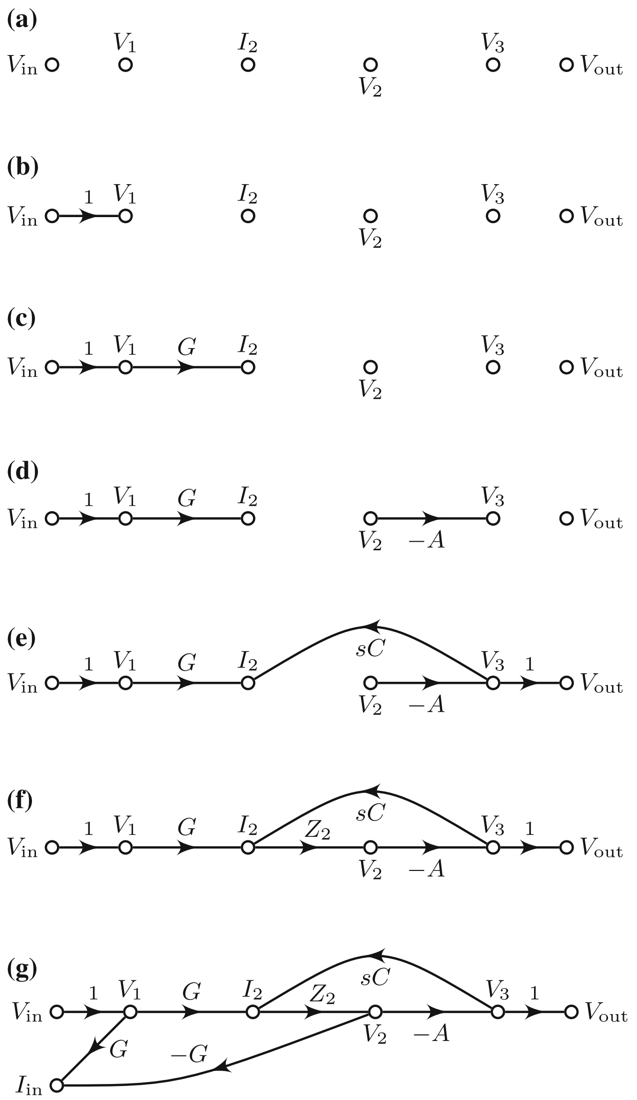

Fig. 3 The step-wise development of the DPSFG for the circuit in Fig. 2(b)

This is almost all, but now we have a problem: I_2 is not zero! The aux source was defined to conduct no current. However, I_2 is not the true current into the source, but the current flowing *into* the aux source for $V_2 = 0$: the short-circuit current of the aux source. So one part of the source superposition is missing. In order to have zero current under superposition, we now need to calculate the necessary voltage $V_2 = Z_2 I_2$ to have I_2 flow *out of* the aux source when all other sources are set to zero. This is Fig. 3(f).

This Z_2 is the driving-point impedance (DPI) which gives the DPSFG part of its name. Since we have made sure that *every* node now has a voltage source attached, all of which are set to zero during superposition, this DPI is always the inverse of the sum of all conductances connected to the node. Here,

$$Z_2 = \frac{1}{G + sC}. \tag{9}$$

This is all, and it becomes apparent that the result is an SFG that shows the causality as well as describing the equation system of the circuit. There is only one loop, and only one path, which even touches the loop. Therefore:

$$L_1 = -A\,Z_2\,sC, \tag{10}$$

$$\Delta = 1 - L_1 = 1 + A\,Z_2\,sC, \tag{11}$$

$$P_1 = -G\,Z_2\,A, \tag{12}$$

$$\Delta_1 = 1, \tag{13}$$

$$T = \frac{V_{\rm out}}{V_{\rm in}} = \frac{P_1\,\Delta_1}{\Delta} = -\frac{G\,Z_2\,A}{1 + A\,Z_2\,sC} \approx -\frac{G}{sC}, \tag{14}$$

where the last approximation is for $A \to \infty$.

Additional things can now be calculated with comparatively small effort. For example, what happens if a parasitic capacitance C_p is connected to node 2? Then only Z_2 changes to

$$Z_2 = \frac{1}{G + s(C + C_p)}. \tag{15}$$

What if we need to know the input admittance? The input current is $I_{\rm in} = (V_1 - V_2)\,G$. Drawn as SFG branches, this gives Fig. 3(g). The only thing that changes are the forward paths, and we get:

$$P_1 = G, \tag{16}$$

$$\Delta_1 = \Delta, \tag{17}$$

$$P_2 = -G^2 Z_2, \tag{18}$$

$$\Delta_2 = 1, \tag{19}$$

$$Y_{\text{in}} = \frac{I_{\text{in}}}{V_{\text{in}}} = \frac{P_1\,\Delta_1 + P_2\,\Delta_2}{\Delta} = G - \frac{G^2\,Z_2}{1 + A\,Z_2\,sC}. \tag{20}$$

For $A \to \infty$, this gives $Y_{\text{in}} = G$, as expected, because then node 2 is virtual ground.

And this method always works, for every linear circuit. The only thing that happens for larger circuits is that there are more aux sources, and of course Mason's rule becomes more tedious to evaluate, but the derivation of the branches is never different from, or more difficult than, what we just showed. Note, however, that from a didactic point of view, it can be better to fill in the DPSFG in a different order (c.f. [4, Chap. 7]).

This method works because the aux sources make Kirchhoff current-law (KCL) equations appear in the SFG. We illustrate this by looking at the equations for the aux-source-related nodes I_2 and V_2:

$$I_2 = V_1\,G + V_3\,sC, \tag{21}$$

$$V_2 = Z_2\,I_2. \tag{22}$$

We insert (22) into (21), take all sum terms to the same side, and collect expressions with G and sC:

$$(V_1 - V_2)\,G + (V_3 - V_2)\,sC = 0. \tag{23}$$

And we see that this is the KCL for node 2.

In summary, the connection of an aux source causes the KCL equation for that node to appear in the graph. This is basically why the DPSFG method must work for any linear circuit.

3 The charge-mode auxiliary source

At this point we are ready to extend the method to SC circuits. After the lengthy introduction it is clear that (a) SC circuits reconfigure when the phases change, and (b) if the proper auxiliary source is chosen, then source superposition becomes simple and drawing an SFG becomes straightforward.

Theorem 1 *The DPSFG method as introduced in* [5], *and as presented without the need to split a circuit into subcircuits in Sect.* 2, *works for all SC circuits with an arbitrary number of phases if (a) one circuit diagram per phase is drawn, with switches shorted or opened, and (b) in each of those circuit diagrams, auxiliary sources are attached to all nodes that are not connected to a voltage source. Such an aux source is defined as follows: It produces the voltage that is already present at a node, and therefore* ***conducts no charge****.*

The validity of this theorem will now be demonstrated by giving examples. Figure 4 shows an SC voltage doubler, a simple SC circuit, which nevertheless is *not* a source-sink network. The figure shows the circuit with the switches, and the two circuit configurations in the odd phase (phase 1) and the even phase (phase 2). We have given all nodes individual numbers.

Node 1 in the odd phase is at a voltage source terminal, but nodes 2 and 3 in the even phase are not, so there we attach aux sources. This time the sources have a voltage $V^{\text{e}}_{2,3}$ and a charge $Q^{\text{e}}_{2,3}$ that is the charge flowing into the source while the even-phase circuit settles to its final state.

Now we can construct the DPSFG just as in Sect. 2.2 but by using charges and voltages instead of currents and voltages. First we write down all nodes of the SFG, in two rows for the two phases, giving Fig. 5(a). Now source by source: V^{o}_{in} is V^{o}_1: Fig. 5(b). If all sources other than V^{o}_1 are set to zero, then C_1 and C_2 are discharged into the aux sources V^{e}_2 and V^{e}_3.

Observing the respective plates of the capacitors discharged into the nodes, and since this is another phase, $Q^{\text{e}}_2 = (C_2\,z^{-1} - C_1\,z^{-1})\,V^{\text{o}}_1$ and $Q^{\text{e}}_3 = -C_2\,z^{-1}$: Fig. 5(c). If V^{e}_2 is switched on and all others are zero, a charge $Q^{\text{e}}_3 = C_2\,V^{\text{e}}_2$ flows: Fig. 5(d). Similarly, if V^{e}_3 is switched on and

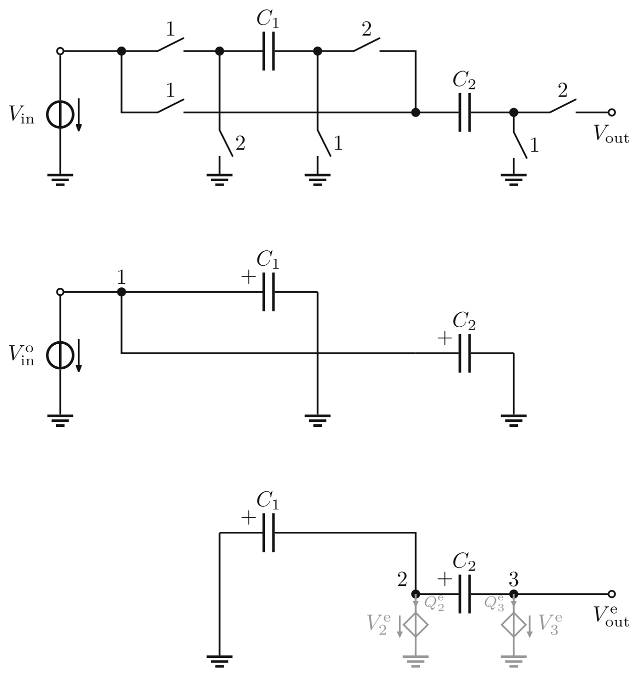

Fig. 4 Inverting capacitive voltage doubler. Top: circuit with all switches. Centre: phase 1 (odd). Bottom: phase 2 (even), with aux sources. (Note: Normally non-inverting doublers are used, but the inverting-doubler schematic is easier to read and thus better suited to discuss an analysis method.)

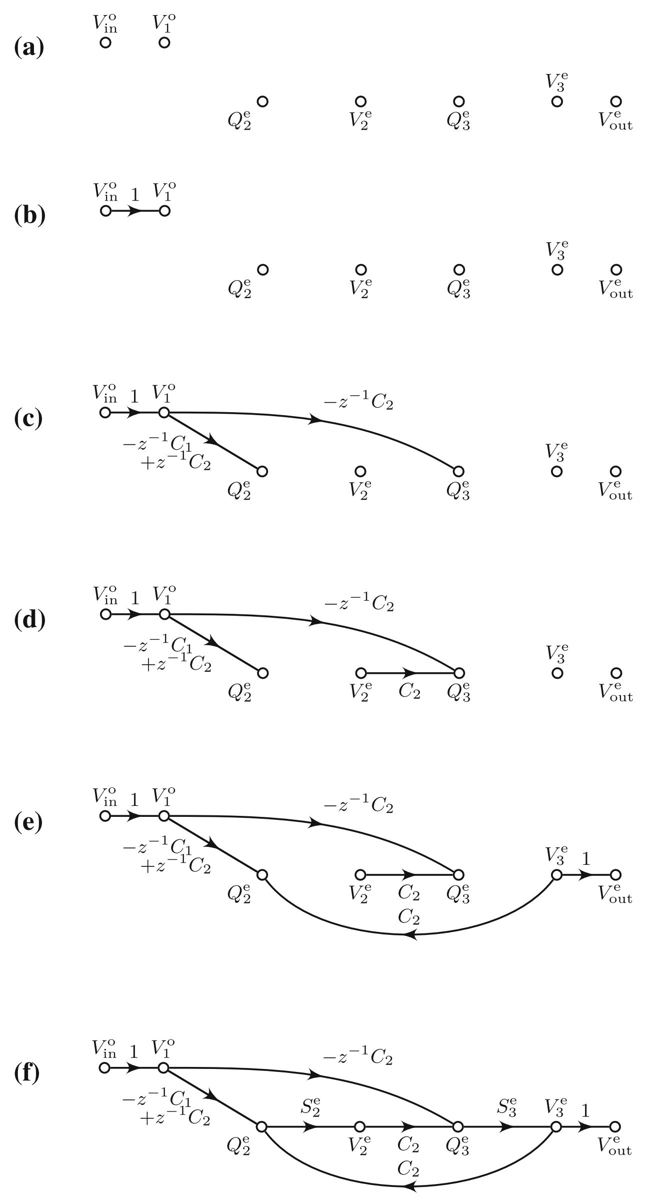

Fig. 5 The step-wise development of the DPSFG for the circuit in Fig. 4(b)

all others are zero, a charge $Q_2^e = C_2\,V_3^e$ flows, and this also is V_{out}^e: Fig. 5(e).

As in Sect. 2.2, we now have calculated the short-circuit charges of the aux sources and need a last step answering the question: what should the voltage of the aux sources be such that, under superposition, the aux source conducts no charge? This gives Fig. 5(f). Before, we got driving-point impedances in this step, and now we get $S_{2,3}^e$, which are two *driving-point elastances*, an elastance being an inverse capacitance (unit F^{-1}), and obviously

$$S_2^e = \frac{1}{C_1 + C_2}, \quad S_3^e = \frac{1}{C_2}, \tag{24}$$

for the exact same reason why driving-point impedances have this structure.

Mason's rule is simple since there is only one loop and two forward paths that touch the loop:

$$L_1 = S_2^e C_2 S_3^e\,C_2, \tag{25}$$

$$\Delta = 1 - L_1, \tag{26}$$

$$P_1 = z^{-1}(C_2 - C_1)S_2^e C_2 S_3^e, \tag{27}$$

$$P_2 = -z^{-1}C_2 S_3^e, \tag{28}$$

$$\Delta_{1,2} = 1, \tag{29}$$

$$T = \frac{V_{out}^e}{V_{in}^o} = \frac{P_1\Delta_1 + P_2\Delta_2}{\Delta} = -2\,z^{-1}. \tag{30}$$

This is very straightforward to extend, particularly with an analysis of parasitic capacitances, which eluded most of the hitherto presented SFG methods. For example, if a C_p is attached to node 2, the only thing that changes is S_2^e in (24):

$$S_2^e = \frac{1}{C_1 + C_2 + C_p}. \tag{31}$$

Then (30) can be calculated again:

$$T = -\frac{2C_1 + C_p}{C_1 + C_p}\cdot z^{-1}. \tag{32}$$

As with the continuous-time DPSFG in Sect. 2, every problem could be solved by step-wise source superposition, but changing the order of how the DPSFG is drawn may make an analysis quicker and less error-prone.

4 Main example: the SC integrator

This is already all of the theory, but the strength of the method lies in the systematic application, so it is best exemplified with a simple (but sufficiently complex) SC circuit: the stray-sensitive SC integrator in Fig. 6.

In the previous section we just numbered all nodes, but now we introduce a technique that makes the node numbers in all phases unique: the nodes are numbered with powers of 2, i.e., $1, 2, 4, 8, 16, \ldots$, in the schematic with switches, Fig. 6 (top). During the derivation of the odd and even schematics in Fig. 6, nodes of the original circuit are connected by closed switches, and the number of such a new node is the sum of the numbers of the nodes connected.

Since the original numbers were powers of 2, the new nodes are guaranteed to have unique numbers and contain local topological information, with the effect that all elastances S_i^{ϕ} with the same i are the same, irrespective of their phase ϕ.

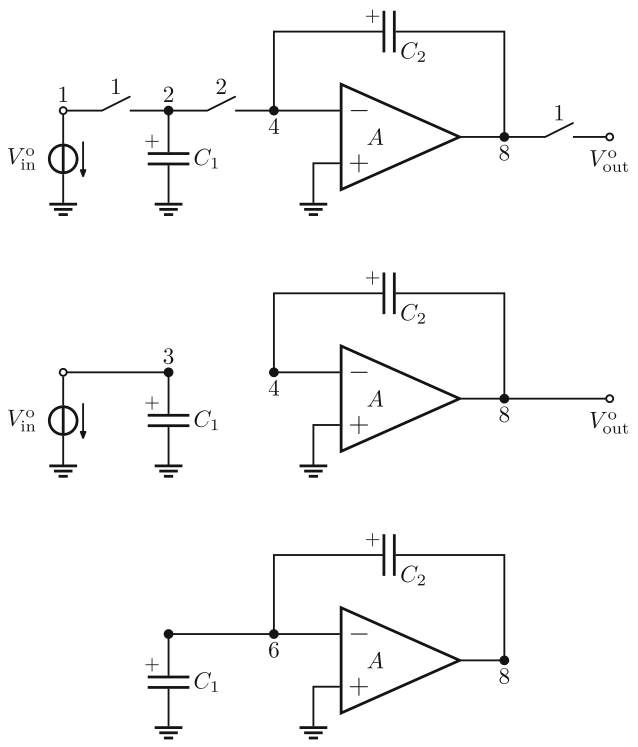

Fig. 6 Schematics of the simple, stray-sensitive SC integrator. The top schematic has all the switches; the centre schematic shows the odd phase 1, and the bottom schematic shows the even phase 2

4.1 Calculation with charge equations

The circuit in Fig. 6 can now be analysed using charge equations. The capacitor charges are

$$Q_1^{\text{o}} = V_3^{\text{o}} C_1, \quad Q_2^{\text{o}} = \left(V_4^{\text{o}} - V_8^{\text{o}}\right) C_2 \tag{33}$$

in the odd phase and

$$Q_1^{\text{e}} = V_6^{\text{e}} C_1, \quad Q_2^{\text{e}} = \left(V_6^{\text{e}} - V_8^{\text{e}}\right) C_2 \tag{34}$$

in the even phase, where $Q_{1,2}$ is the charge on $C_{1,2}$ with the positive plate of the capacitors indicated in Fig. 6.

The amplifier with gain A works in both phases, giving two equations:

$$V_8^{\text{o}} = A\, V_4^{\text{o}}, \quad V_8^{\text{e}} = A\, V_6^{\text{e}}. \tag{35}$$

Now it is straightforward to write down the charge differences on the capacitors during the phase transitions:

$$\Delta Q_1^{\text{e}} = Q_1^{\text{e}} - z^{-1} Q_1^{\text{o}}, \quad \Delta Q_2^{\text{e}} = Q_2^{\text{e}} - z^{-1} Q_2^{\text{o}}, \tag{36}$$

$$\Delta Q_1^{\text{o}} = Q_1^{\text{o}} - z^{-1} Q_1^{\text{e}}, \quad \Delta Q_2^{\text{o}} = Q_2^{\text{o}} - z^{-1} Q_2^{\text{e}}. \tag{37}$$

Then there is charge conservation. In each phase, there are nodes without voltage sources on them where no charge can appear or disappear. In the odd phase, this is only node 4, giving the equation

$$\Delta Q_2^{\text{o}} = 0. \tag{38}$$

In the even phase, only node 6 has no voltage source attached, giving one more equation:

$$\Delta Q_1^{\text{e}} + \Delta Q_2^{\text{e}} = 0. \tag{39}$$

Observe how both ΔQ_1^{e} and ΔQ_2^{e} are on the same side of the equation; this is because both capacitor plates connected to node 6 in the even phase have the same sign on them.

The final step is to specify the input and the output of the circuit:

$$V_{\text{in}}^{\text{o}} = V_3^{\text{o}}, \quad V_{\text{out}}^{\text{o}} = V_8^{\text{o}}. \tag{40}$$

The equation system (33)–(40) can now be solved for $V_{\text{out}}^{\text{o}}$:

$$\begin{aligned}\frac{V_{\text{out}}^{\text{o}}}{V_{\text{in}}^{\text{o}}} &= -\frac{A\, C_1 z^{-2}}{(A+1)C_2 + C_1 - z^{-2}(A+1)C_2} \\ &\approx -\frac{C_1}{C_2}\frac{z^{-2}}{1-z^{-2}} \quad \text{for A} \to \infty.\end{aligned} \tag{41}$$

Considering that our z^{-2} means we have *one* delay from an odd phase to the next odd phase (see “Appendix”), this is the well-known transfer function of the SC integrator (e.g., [15]).

4.2 Calculation with a driving-point signal-flow graph

The same calculation can now be done with signal-flow graphs. In this section we show how to do it systematically. **First,** auxiliary sources have to be attached to all circuit nodes that do not already have voltage sources on them. These are the nodes 4 and 6 in Fig. 7.

Second, a signal-flow graph can be prepared, as shown in Fig. 8(a). This graph contains the input and output paths, the amplifying paths, and, in every node where there is an

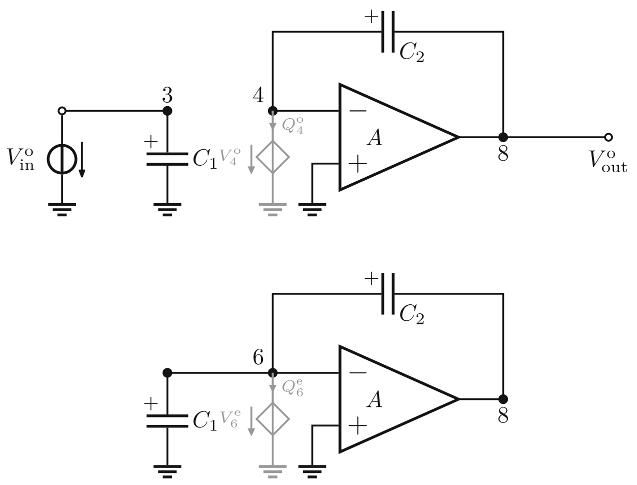

Fig. 7 Schematics from Fig. 6 with aux sources

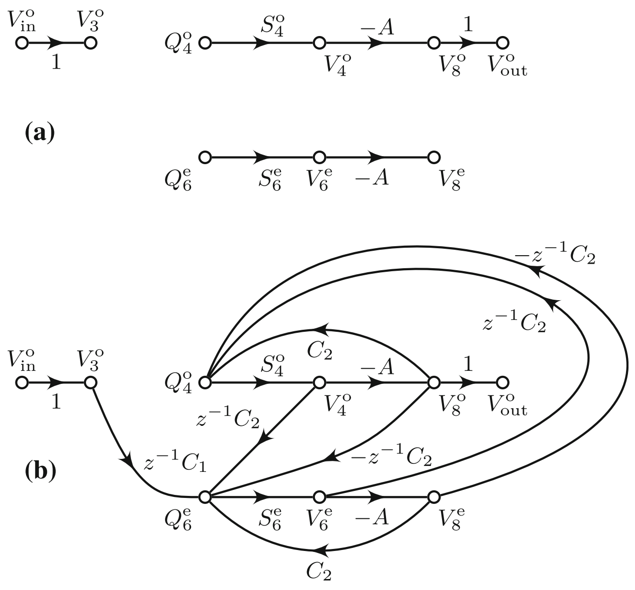

Fig. 8 DPSFG derived from Fig. 7; **a** shows the first step, **b** the complete graph

aux source, also an aux-source charge and an elastance path. The elastances are:

$$S^o_4 = \frac{1}{C_2} \quad S^e_6 = \frac{1}{C_1 + C_2}. \tag{42}$$

Third, all paths due to all voltage sources can be drawn, leading from Fig. 8(a, b). We will just explain two of these paths. First, let all voltage sources be zero, and then let V^o_8 be switched to its value. Then the charge $C_2 V^o_8$ will flow into the aux source at node 4, giving a contribution to Q^o_4. This is the path with weight C_2 from V^o_8 to Q^o_4.

However, this is not the only effect V^o_8 will have, because in this step of calculating source superposition, all other voltages are zero, particularly all voltages in the even phase. The capacitor C_2 has now been charged to $-C_2 V^o_8$ in the odd phase (observe the + sign on the left plate of C_2 to understand why the charge is negative), and in the even phase it is completely discharged because the voltage is zero on both sides. Therefore the charge $-C_2 V^o_8$ flows into the aux source at node 6. Since this is after a phase change, we get a path $-z^{-1}C_2$ from V^o_8 to Q^e_6.

All other missing paths can then be found by applying the same method to V^o_3, V^o_4, V^e_6, and V^e_8, and then the DPSFG is complete.

Finally, it would already be possible to apply Mason's gain rule, and if our only goal were to get a transfer function as quickly as possible, then we would stop here. However, using one of the graphical simplification methods mentioned in Sect. 2, Fig. 8(b) can readily be re-drawn as Fig. 9. Then a term $C_2(A+1)$ appears, and it becomes visible in the graph already that C_2 is subject to the Miller effect.

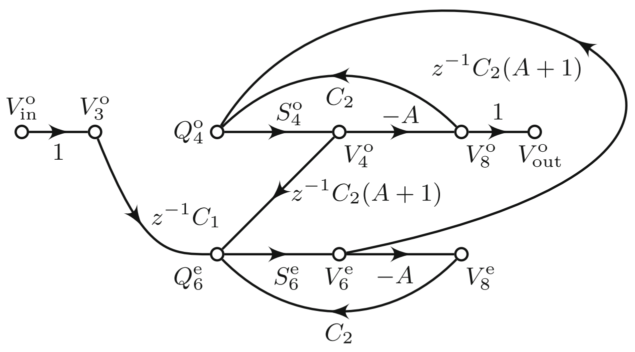

Fig. 9 DPSFG derived from Fig. 8(b) by graphical simplification. In Fig. 8(b), there were two ways to go from V^o_4 to Q^e_6, one had the weight $z^{-1}C_2$, the other the weight $Az^{-1}C_2$. Their sum is $z^{-1}C_2(A+1)$, which is the path from V^o_4 to Q^e_6 in this SFG. The same was done for the path from V^e_6 to Q^o_4

The graph in Fig. 9 has only three loops:

$$L_1 = -A\, S^o_4 C_2, \quad L_2 = -A\, S^e_6 C_2, \tag{43}$$

$$L_3 = S^o_4 z^{-1} C_2(A+1) S^e_6 z^{-1} C_2(A+1). \tag{44}$$

According to Mason's gain rule, the graph determinant becomes

$$\Delta = 1 - L_1 - L_2 - L_3 + L_1 L_2. \tag{45}$$

Note that the term $L_1 L_3$ does not appear, because Loop 1 and Loop 3 have nodes in common, and the same is true for $L_2 L_3$.

There is one path from V^o_{in} to V^o_{out},

$$P_1 = z^{-1} C_1 S^e_6 z^{-1} C_2 (A+1) S^o_4 (-A), \tag{46}$$

and since this path touches all three loops, its sub-determinant $\Delta_1 = 1$.

With this information, the transfer function of the circuit can be calculated as

$$\frac{V^o_{out}}{V^o_{in}} = \frac{P_1 \Delta_1}{\Delta}, \tag{47}$$

which, of course, gives the same result obtained in (41).

As before, calculating the effects of parasitic capacitors would be simple with this method as they would only change the elastance equations (42).

4.3 Comparison of the two methods

This DPSFG method gives a direct, graphical way to draw a graph from a circuit and solve it using Mason's gain rule. What remains to be shown now is how the two methods relate.

In the classical method, we wrote down charge conservation equations at all nodes where there were no voltage sources. In the DPSFG method, we introduced aux sources to all nodes where there were no voltage sources. So it stands to reason that the SFG branches pertaining to the aux-source nodes should be charge conservation equations, just as they were KCL equations in the continuous-time case.

Proof for (39): First, substitute (36) into (39):

$$Q_1^e - z^{-1}Q_1^o + Q_2^e - z^{-1}Q_2^o = 0 \tag{48}$$

Now substitute (33) and (34) into (48):

$$\begin{aligned} V_6^e C_1 - z^{-1}V_3^o C_1 + \left(V_6^e - V_8^e\right)C_2 \\ - z^{-1}\left(V_4^o - V_8^o\right)C_2 = 0. \end{aligned} \tag{49}$$

This can be re-ordered as follows:

$$\begin{aligned} V_6^e(C_1 + C_2) = z^{-1}C_1V_3^o + C_2V_8^e \\ + z^{-1}C_2V_4^o - z^{-1}C_2V_8^o. \end{aligned} \tag{50}$$

Now let us look at Fig. 8(b), at the DPSFG branch from Q_6^e to V_6^e. It has the weight S_6^e, which corresponds to the equation

$$V_6^e = S_6^e Q_6^e. \tag{51}$$

Substituting (42) into this and solving for Q_6^e gives:

$$Q_6^e = V_6^e(C_1 + C_2). \tag{52}$$

This can now be used together with (50) to calculate Q_6^e:

$$\begin{aligned} Q_6^e = z^{-1}C_1V_3^o + C_2V_8^e \\ + z^{-1}C_2V_4^o - z^{-1}C_2V_8^o, \end{aligned} \tag{53}$$

and it is immediately apparent that this is the equation obtained from adding all branches going into the node Q_6^e in the DPSFG.

In summary, the connection of an aux source causes the charge conservation equation for this node to appear in the graph. This is why the method will work for any SC circuit operating with voltage settling and charge transfer.

5 Second example: multi-phase SC networks and discrete-time input impedance

The real power of this method lies in the fact that it only uses very basic principles, and therefore can be applied to *any* switched-capacitor circuit that works with settling and charge transfer. We will now demonstrate this using the simulated inductor in Fig. 10. This circuit does not only have four phases, but it requires the calculation of the input charge difference ΔQ_{in} that the input source V_{in} has to provide.

The first step is to draw the circuit in its four phases; we call them a, b, c and d instead of e and o. This is shown in Fig. 11.

Drawing the DPSFG can be done as before: first, we draw the graph in Fig. 12(a) by observing the source superposition theorem in all four phases individually, and then we obtain Fig. 12(b) by inserting the paths showing charge transfer between the phases.

The elastances on the branches of Fig. 12 are:

$$S_4^a = S_4^c = S_4^d = (C_2)^{-1}, \tag{54}$$

$$S_6^b = (C_0 + C_2)^{-1}, \tag{55}$$

$$S_{16}^b = S_{16}^c = (C_1)^{-1}, \tag{56}$$

$$S_{18}^d = (C_0 + C_1)^{-1}, \tag{57}$$

and as explained in Sect. 4, all elastances with the same index number are the same, even if they are in different phases.

Two more things should be noticed when going from Fig. 12(a, b): first, we have applied the simplification demonstrated in Fig. 9, which is why the integrating capacitor C_2 gives rise to four paths with the weight $(1 + A)z^{-1}C_2$. Second, the input charge ΔQ_{in} has two contributions. At the beginning of phase a, the charge of C_0 is $C_0V_{18}^d$. At the end it is $C_0V_2^a$. The charge ΔQ_{in} that has to be supplied by the voltage source is therefore

$$\Delta Q_{in} = C_0V_2^a - z^{-1}C_0V_{18}^d. \tag{58}$$

This gives the two branches from V_{18}^d and from V_2^a to ΔQ_{in}, and the signal-flow graph is complete.

At this point we can evaluate Mason's gain rule: There is one loop each in each of the four phases, and one big loop going through everything:

$$L_1 = S_4^a\,(-A)\,C_2, \tag{59}$$

$$L_2 = S_6^b\,(-A)\,C_2, \tag{60}$$

$$L_3 = S_4^c\,(-A)\,C_2, \tag{61}$$

$$L_4 = S_4^d\,(-A)\,C_2, \tag{62}$$

$$L_5 = S_4^a\,S_6^b\,S_4^c\,S_4^d\left((1 + A)z^{-1}C_2\right)^4. \tag{63}$$

The graph determinant is:

$$\begin{aligned} \Delta = 1 &- L_1 - L_2 - L_3 - L_4 - L_5 \\ &+ L_1L_2 + L_1L_3 + L_1L_4 + L_2L_3 + L_2L_4 + L_3L_4 \\ &- L_1L_2L_3 - L_1L_2L_4 - L_2L_3L_3 \\ &+ L_1L_2L_3L_4. \end{aligned} \tag{64}$$

There are three forward paths,

$$P_1 = C_0, \tag{65}$$

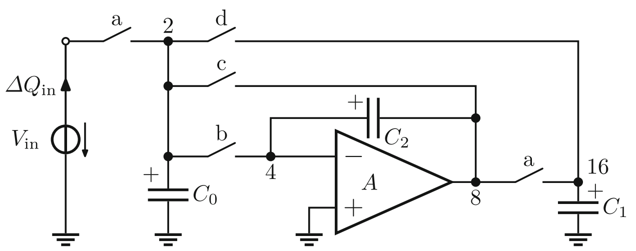

Fig. 10 A four-phase SC circuit: the simulated inductor from [16, Fig. 4]

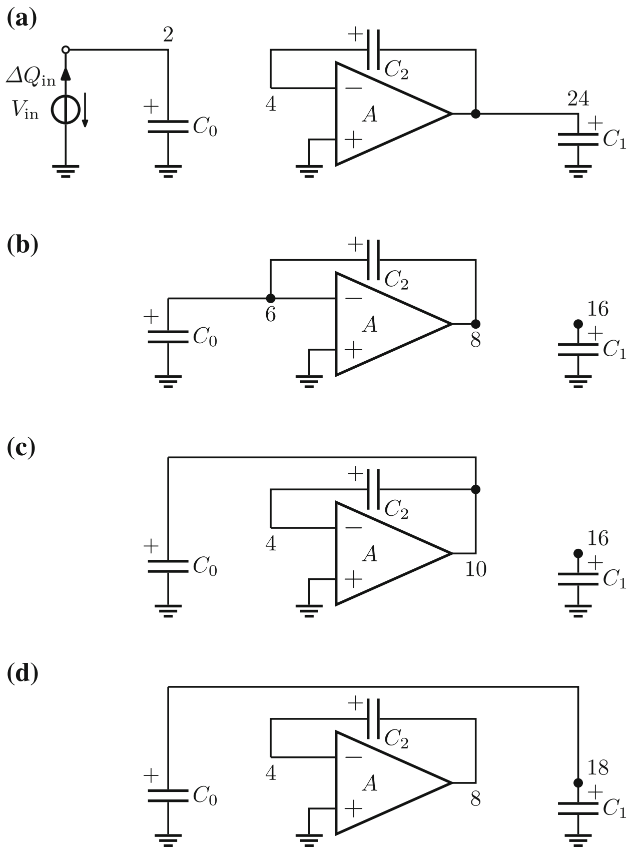

Fig. 11 The circuit from Fig. 10 shown in its four phases (called a, b, c, and d, from top to bottom)

$$P_2 = z^{-1}C_0\,S_6^{\mathrm{b}}\left((1+A)z^{-1}C_2\right)S_4^{\mathrm{c}}\,(-A) \cdot z^{-1}C_0\,S_{18}^{\mathrm{d}}\left(-z^{-1}C_0\right), \tag{66}$$

$$P_3 = z^{-1}C_0\,S_6^{\mathrm{b}}S_4^{\mathrm{c}}S_4^{\mathrm{d}}S_4^{\mathrm{a}}\left((1+A)z^{-1}C_2\right)^3(-A) \cdot z^{-1}C_1\,S_{16}^{\mathrm{b}}S_{16}^{\mathrm{c}}S_{18}^{\mathrm{d}}\left(z^{-1}C_1\right)^2\left(-z^{-1}C_0\right), \tag{67}$$

with the respective sub-determinants

$$\Delta_1 = \Delta, \tag{68}$$

$$\Delta_2 = 1 - L_1 - L_4 + L_1L_4, \tag{69}$$

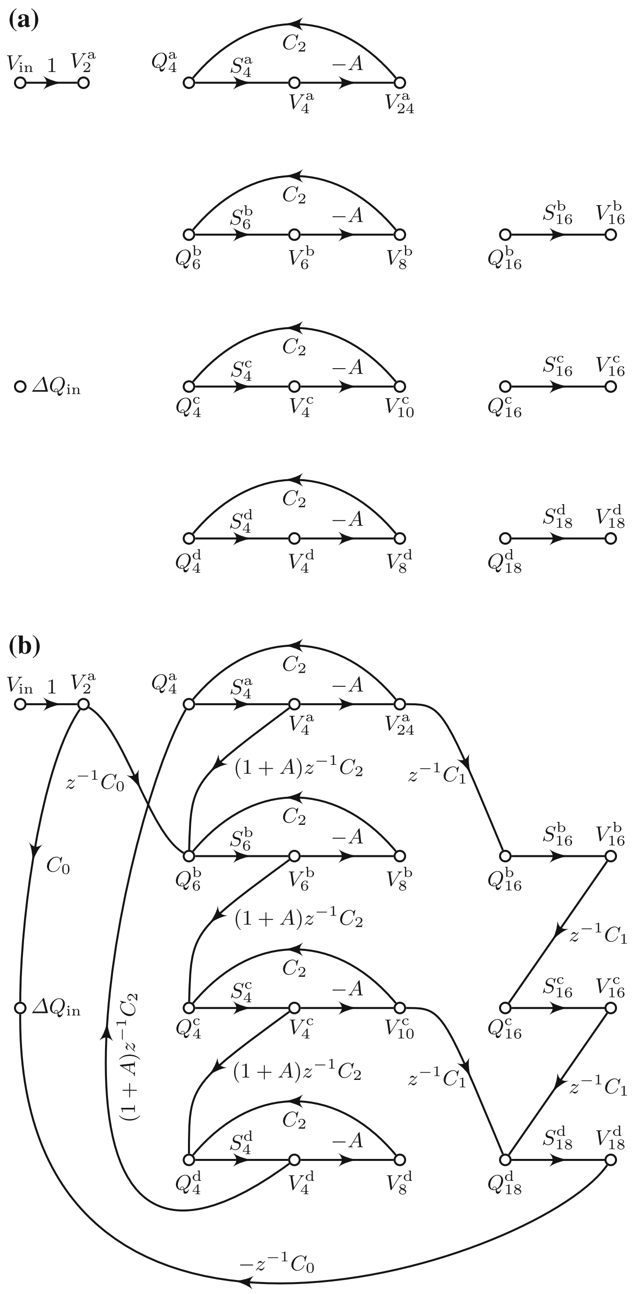

Fig. 12 The DPSFGs derived from Fig. 11. Graph **a** shows the portions of each graph derived from the individual phases, **b** adds the paths for charge transfer between phases and for calculating the input charge according to (58)

$$\Delta_3 = 1. \tag{70}$$

Now it is possible to compute the discrete-time input admittance, which has the unit F, as follows:

$$Y_{\mathrm{in}} = \frac{\Delta Q_{\mathrm{in}}}{V_{\mathrm{in}}} = \frac{P_1\Delta_1 + P_2\Delta_2 + P_3\Delta_3}{\Delta} \tag{71}$$

$$= P_1 + \frac{P_2\Delta_2 + P_3\Delta_3}{\Delta}, \tag{72}$$

the latter line comes from the fact that $\Delta_1 = \Delta$.

Substituting (54) to (70) into (72) gives a rather complicated expression, which we will not reproduce here. In [16], the whole calculation is made for $A \to \infty$; if we do this as well, we get

$$Y_{\text{in}} = C_0 \left[1 + \frac{C_0 z^{-4}}{C_2(C_0 + C_1)} \frac{C_0 + C_1 z^{-4}}{1 - z^{-4}} \right], \tag{73}$$

which is the same as Eq. (15) in [16]. This acts like a discrete-time simulated inductor for a specific choice of capacitor ratios: Inserting $C_1 = C_0/3$ and $C_2 = C_0/4$ into (73) gives

$$Y_{\text{in}} = C_0 \frac{(1 + z^{-4})^2}{1 - z^{-4}}. \tag{74}$$

Since this is a four-phase system which only samples the input at phase 1, this corresponds to

$$Y_{\text{in}} = C_0 \frac{(1 + z^{-1})^2}{1 - z^{-1}}. \tag{75}$$

relative to phase one (see "Appendix"). As expected for an inductance, the admittance has one pole at $z = 1$ (which corresponds to $f = 0$).

6 When Mason is too difficult...

The example in the previous section resulted in a very complicated and tedious evaluation of Mason's gain rule. Clearly, this example is at the border of what is reasonably calculatable by hand.

Note, however, that this only concerns Mason's rule. We can still use the DPSFG method to obtain a correct graph that corresponds to the circuit diagram (which makes it easily reviewable), but then use a computer algebra tool (Mathematica, Python/NumPy/SciPy,...) to actually evaluate the equation system.

Such a computation entails listing the dependent variables, listing all equations (one per node), and then solving the equations and substituting the elastances.

This can easily be done for the graph in Fig. 12, but we will show the script for the graph in Fig. 9 such that it is more instructive to the reader. In Mathematica[1] notation, this would be:

```
depvar = {v3o, q4o, v4o, v8o, voo, q6e, v6e, v8e}

eqs = {v3o == vio,
       q4o == C2 v8o + z1 C2 (A+1) v6e,
       v4o == S4o q4o,
       v8o == -A v4o,
       voo == v8o,
       q6e == z1 C1 v3o + z1 C2 (A+1) v4o + C2 v8e,
       v6e == S6e q6e,
       v8e == -A v6e}

elast = {S4o -> 1/C2,
         S6e -> 1/(C1 + C2)}

sols = Solve[ eqs, depvar ]

T = voo/vio /. sols /. elast

Simplify[Limit[ T , A->\[Infinity]]]
```

A brief explanation: In Mathematica, `=` is an assignment, `==` an equality, `->` a substitution rule, and the operator `/.` performs substitutions. The solution `sols` consists of one substitution rule for every dependent variable. Therefore, the last line calculates the transfer function `voo/vio`, substitutes the solution for `voo` and then the elastances `elast`, takes the $\lim_{A\to\infty}$, and finally simplifies. The result of this code is:

```
(C1 z1^2)/(C2 (-1 + z1^2))
```

This is evidently the same as (41).

7 Conclusions

In this paper, we have adapted the driving-point signal-flow graphs (SFGs) to switched-capacitor circuits by one main modification: we have exchanged the zero-current auxiliary source by a zero-charge aux source. This let the driving-point impedances become driving-point elastances.

We have demonstrated the application of the method by examples, in which we also showed a few tricks helping simplify the analysis even further, and have shown by argument that this method is applicable to any SC circuit relying on charge transfer just as the original method is applicable to any linear continuous-time circuit.

Therefore, we have unified the SFG analysis methodology for continuous-time and SC circuits. Our method only requires that the users remember Mason's gain rule and understand source superposition and the aux source. Written material like tables of partial transfer functions, equivalent circuits, etc., are unnecessary. In this respect our method is unique among graph-based SC analysis methods.

Of course, the same analysis can be done by deriving and solving charge equations, which would give the same

[1] Note that at the time we write this paper, Mathematica is available on the cheap Raspberry Pi computers for free, and calculations such as the ones described here do not take much computation time even on such a comparatively slow computer.

symbolic results with fewer steps and more quickly using a computer algebra tool. The graph method, however, maintains causal relationships and makes them visible, and when studying and comparing different SC structures, this insight is often worth the additional effort.

While the last example gave a very complicated graph determinant and is at the limit of what can be done by hand (in fact, we, the authors, evaluated Mason's rule using Mathematica to be sure we do not make stupid mistakes), the one-to-one correspondence makes it straightforward to review a drawn SFG versus its circuit. It is uncommon with this method that an equation is omitted, and near impossible to introduce redundant equations, which is something that everyone who has analyzed circuits by hand using equations knows all too well.

Using Mason's gain rule to derive transfer functions can give more insight into a circuit, but if evaluating Mason's rule becomes too complicated, then calculating transfer functions from a signal-flow graph can also be done directly by entering the euqations into a computer algebra tool.

Appendix

In most papers cited in this paper, the z transform is used to describe discrete-time transfer functions, but there are different ways in literature to do this. A discrete-time integrator with delay has the transfer function

$$T(z) = \frac{z^{-1}}{1 - z^{-1}}. \qquad (76)$$

If this is achieved with a two-phase SC circuit like the one in Fig. 6 that measures the input and provides the output in phase 1 (the odd phase) and integrates in phase 2 (the even phase), then there are three different ways to use the z transform: (a) assign no delay between odd and even phase, and z^{-1} between even and next odd phase; (b) assign $z^{-\frac{1}{2}}$ to both phase transitions; (c) assign z^{-1} to both phase transitions.

(a) and (b) are often used in teaching, because (c) has the disadvantage that the integrator in Fig. 6 then looks like

$$T(z) = \frac{z^{-2}}{1 - z^{-2}}. \qquad (77)$$

In general, to compare a z transfer function of an n-phase SC circuit to a standard z transfer function, all terms z^{-n} have to be replaced by z^{-1} in the end.

Method (c) is the mathematically soundest of the three and the easiest to apply to multiphase circuits, and is therefore used in most theoretical papers, including this one.

References

1. Wanhammar, L. (2009). *Analog filters using MATLAB*. Berlin: Springer.
2. Chan Carusone, T., Johns, D., & Martin, K. (2011). *Analog integrated circuit design* (2nd ed.). Hoboken: Wiley.
3. Ndjountche, T. (2016). *CMOS analog integrated circuits: high-speed and power-efficient design*. Boca Raton: CRC Press.
4. Schmid, H. (2016). Signal-flow graphs in 12 short lessons. HT FHNW EIT: Analog and mixed-signal circuits and signal processing (video channel). FHNW. Retrieved February 28, 2018 from https://tube.switch.ch/channels/d206c96c.
5. Ochoa, A. (1998). A systematic approach to the analysis of general and feedback circuits and systems using signal flow graphs and driving-point impedance. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *45*(2), 187–195.
6. Mason, S. J. (1953). Feedback theory: Some properties of signal flow graphs. *Proceedings of the IRE*, *41*(9), 1144–1156.
7. Mason, S. J. (1956). Feedback theory: Further properties of signal flow graphs. *Proceedings of the IRE*, *44*(7), 920–926.
8. Schmid, H. (2002). Circuit transposition using signal-flow graphs. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, *2*, 25–28.
9. Kurth, C., & Moschytz, G. (1979). Nodal analysis of switched-capacitor networks. *IEEE Transactions on Circuits and Systems*, *26*(2), 93–105.
10. Tsividis, Y. (1979). Analysis of switched capacitive networks. *IEEE Transactions on Circuits and Systems*, *26*(11), 935–947.
11. Hökenek, E., & Moschytz, G. S. (1980). Analysis of general switched-capacitor networks using indefinite admittance matrix. *IEE Proceedings G - Electronic Circuits and System*, *127*(1), 21–33.
12. Hökenek, E., & Moschytz, G. S. (1980). Analysis of multiphase switched-capacitor (msc) networks using the indefinite admittance matrix (iqm). *IEE Proceedings G - Electronic Circuits and System*, *127*(5), 226–241.
13. Moschytz, G. S., & Brugger, U. W. (1984). Signal-flow graph analysis of SC networks. *IEE Proceedings G - Electronic Circuits and System*, *131*(2), 72–85.
14. Dabrowski, A., & Moschytz, G. S. (1990). Direct analysis of multiphase switched-capacitor networks using signal-flow graphs. *IEEE Transactions on Circuits and Systems*, *37*(5), 594–607.
15. Ki, W. H., & Temes, G. C. (1990). Offset-compensated switched-capacitor integrators. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, *4*, 2829–2832.
16. Cichocki, A., & Unbehauen, R. (1987). Simplified analysis of arbitrary switched-capacitor networks. *IEE Proceedings G - Electronic Circuits and System*, *134*(1), 45–53.

17. Schmid, H., Eichelberger, L., & Huber, A. (2016). A tutorial to switched-capacitor noise analysis by hand. *Analog Integrated Circuits and Signal Processing*, *89*(1), 249.

Hanspeter Schmid received the diploma in Electrical Engineering in 1994, the post-graduate degree in Information Technologies in 1999, and the degree Doctor of Technical Sciences in 2000, all from the Swiss Federal Institute of Technology (ETH Zürich), Switzerland. He joined the Signal and Information Processing Laboratory of the ETH Zürich as a teaching assistant in 1994 and later became a research assistant and junior lecturer in the field of analog integrated filters. From 2000 to 2005, he was an analog-IC designer with Bernafon AG, Switzerland, where he was part of a design team who developed a new IC platform for hearing aids. In that team he worked on audio low-noise amplifiers and voltage regulators in particular and on full-system signal integrity in general. He joined the Institute of Microelectronics of the University of Applied Sciences Northwestern Switzerland (IME/FHNW) as a research fellow in 2005 and became a full professor in February 2012. He is also a part-time senior lecturer at ETH Zürich (Analog Signal Processing and Filtering). His main research interests are fast low-power circuits (mainly for sensor electronics), signal integrity in analog signal processing, and sigma-delta conversion. Hanspeter Schmid was IEEE CAS Analog Signal Processing Technical Committee Co-Chair from 2008 to 2010; he was an Associate Editor of TCAS-I from 2007 to 2011 and an Associate Editor of TCAS-II from 2016 to 2019; he is a member of the standing ESSCIRC technical committee (sensors, imagers and biomedical devices), and in 2011–2012 he was a Distinguished Lecturer of the IEEE CAS Society.

Alex Huber received the Dipl. Ing. and the Ph.D. degrees in Electrical Engineering from the Swiss Federal Institute of Technology (ETH), Zürich, in 1993 and 2000, respectively. From 1993 to 2000, he was with the Electronics Laboratory, ETH Zürich, as a research assistant, where he worked on RF circuit design and modelling of InP/InGaAs HBT devices. Since October 1999, he has been with the Institute of Microelectronics of the University of Applied Sciences Northwestern Switzerland, (FHNW/IME). His main research interests include low-power and high-speed integrated circuits in CMOS technologies for sensor and communication applications.