Analog frontend and filters

Carsten Wulff, carsten@wulff.no

Keywords: H(s), BiQuad, Gm-C, Active-RC, OTA

Introduction

The world is analog, and these days, we do most signal processing in the digital domain. With digital signal processing we can reuse the work of others, buy finished IPs, and probably do processing at lower cost than for analog.

Analog signals, however, might not be suitable for conversion to digital. A sensor might have a high, or low impedance, and have the signal in the voltage, current, charge or other domain.

To translate the sensor signal into something that can be converted to digital we use analog front-ends (AFE). How the AFE looks will depend on application, but it's common to have amplification, frequency selectivity or domain transfer, for example current to voltage.

An ADC will have a sample rate, and will alias (or fold) any signal above half the sample rate, as such, we also must include a anti-alias filter in AFE that reduces any signal outside the bandwidth of the ADC as close to zero as we need.

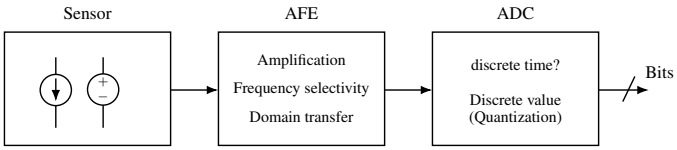

Figure 1: Analog signal chain from sensor through AFE (amplification, frequency selectivity, domain transfer) to ADC

One example of an analog frontend is the receive chain of a typical Bluetooth radio. The signal that arrives at the antenna, or the "sensor", can be weak, maybe -90 dBm.

At the same time, at another frequency, there could be a unwanted signal, or blocker, of -30 dBm

Assume for the moment we actually used an ADC at the antenna, how many bits would we need?

Bluetooth uses Gaussian Frequency Shift Keying, which is a constant envelope binary modulation, and it's usually sufficient with low number of bits, assume 8-bits for the signal is more than enough.

If we assume the maximum of the ADC must fit the blocker, and the resolution must resolve the wanted signal, we get the numbers in the table below.

What	Power [dBm]	Voltage [V]
Blocker	-30	7 m
Wanted	-90	7 u
Resolution		Wanted/255 = 28 n

Then we can calculate the number of bits as

$$\text{ADC resolution } \Rightarrow \ln \frac{7 \text{ mV}}{28 \text{ nV}} / \ln 2 \approx 18 \text{ bits}$$

If we were to sample at 5 GHz, to ensure the bandwidth is sufficient for a 2.480 GHz maximum frequency we can actually compute the power consumption.

Given the Walden figure of merit of

$$FOM = \frac{P}{2^{ENOB} fs}$$

The best FOM in literature is about 1 fJ/step, so

$$P = 1 \text{ fJ/step} \times 2^{18} \times 5\text{GHz} = 1.31 \text{ W}$$

If we look at a typical system, like the Whoop. We can have a look at teardowns, to find the battery size.

Whoop battery is 205mAh at 3.8 V

Then we can compute the lifetime running an ADC based Bluetooth Radio

$$\text{Hours} = \frac{205 \text{ mAh}}{1.31 \text{ W}/3.8 \text{ V}} = 0.6 \text{ h}$$

I know my whoop lasts for almost a week, so it can't be what Bluetooth ICs do.

I know a little bit about radio's, especially inside the Whoop, since it has

Nordic Inside

I can't tell you how the Nordic radio works, but I can tell you how others usually make their radio's. The typical radio below has multiple blocks in the AFE.

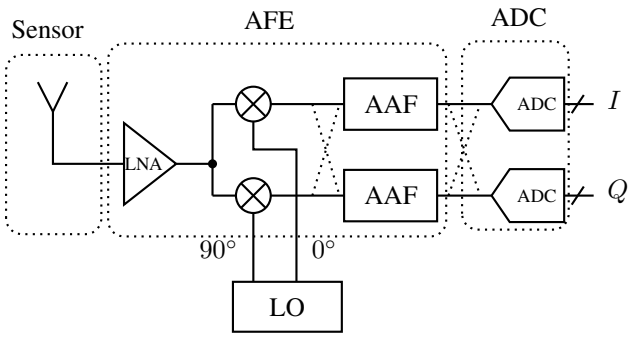

Figure 2: Typical radio receiver frontend: LNA, complex mixer driven by a local oscillator, anti-alias filters and ADCs for the I and Q paths

First is low-noise amplifier (LNA) amplifying the signal by maybe 10 times. The LNA reduces the noise impact of the following blocks. The next is the complex mixer stage, which shifts the input signal from radio frequency down to a low frequency, but higher than the bandwidth of the wanted signal. Then there is a complex anti-alias filter, also called a poly-phase filter, which rejects parts of the unwanted signals. Lastly there is a complex ADC to convert to digital.

In digital we can further filter to select exactly the wanted signal. Digital filters can have high stop band attenuation at a low power and cost. There could also be digital mixers to further reduce the frequency.

The AFE makes the system more efficient. In the 5 GHz ADC output, from the example above, there's lot's of information that we don't use.

An AFE can reduce the system power consumption by constraining digital information processing and conversion to the parts of the spectrum with information of interest.

There are instances, though, where the full 2.5 GHz bandwidth has useful information. Imagine in a cellular base station that should process multiple cell-phones at the same time. In that case, it could make sense with an ADC at the antenna.

What make sense depends on the application.

FILTERS

A filter can be fully described by the transfer function, usually denoted by $H(s) = \frac{\text{output}}{\text{input}}$.

Most people will today start design with a high-level simulation, in for example Matlab, or Python. Once they know roughly the transfer function, they will implement the actual analog circuit.

For us, as analog designers, the question becomes "given an $H(s)$, how do we make an analog circuit?" It can be shown that a combination of 1'st and 2'nd order stages can synthesize any order filter.

Once we have the first and second order stages, we can start looking into circuits.

First order filter

In the book they use signal flow graphs to show how the first order stage can be generated. By selecting the coefficients k_0 ,k_1 and ω_0 we can get any first order filter, and thus match the $H(s)$ we want.

I would encourage you to try and derive from the signal flow graph the $H(s)$ and prove to your self the equation is correct.

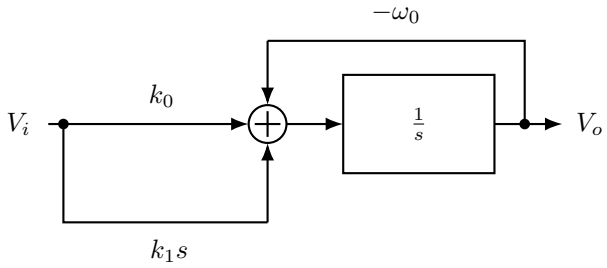

Figure 3: Signal flow graph of a general first order filter with coefficients k_0, k_1 and ω_0

Signal flow graphs are useful when dealing with linear systems.

The instructions to compute the transfer functions are

1. any line with a coefficient is a multiplier
2. any box output is a multiplication of the coefficient and the input
3. any sum, well, sum all inputs
4. be aware of gremlins (a sudden -+ swap)

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{k_1 s + k_0}{s + w_o}$$

Let's call the $1/s$ box input u

$$u = (k_0 + k_1 s)V_i - \omega_0 V_o$$

$$V_o = u/s$$

$$u = V_o s = (k_0 + k_1 s)V_i - \omega_0 V_o$$

$$(s + \omega_0)V_o = (k_0 + k_1 s)V_i$$

$$\frac{V_o}{V_i} = \frac{k_1 s + k_0}{s + \omega_0}$$

Second order filter

Bi-quadratic is a general purpose second order filter.

Bi-quadratic just means "there are two quadratic equations". Once we match the k's ω_0 and Q to our wanted $H(s)$ we can proceed with the circuit implementation.

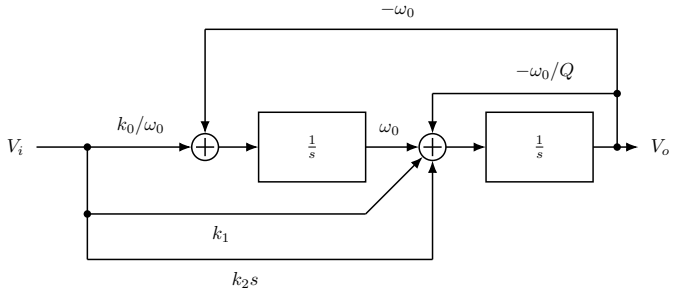

Figure 4: Signal flow graph of the bi-quadratic (second order) filter

$$H(s) = \frac{k_2 s^2 + k_1 s + k_0}{s^2 + \frac{\omega_0}{Q} s + \omega_o^2}$$

Follow exactly the same principles as for first order signal flow graph. If you fail, and you can't find the problem with your algebra, then maybe you need to use Maple or Mathcad.

I guess you could also spend hours training on examples to get better at the algebra. Personally I find such tasks mind numbingly boring, and of little value. What's important is to remember that you can always look up the equation for a bi-quad in a book.

How do we implement the filter sections?

While I'm sure you can invent new types of filters, and there probably are advanced filters, I would say there is roughly three types. Passive filters, that do not have gain. Active-RC filters, with OTAs, and usually trivial to make linear. And Gm-C filters, where we use the transconductance of a transistor and a capacitor to set the coefficients. Gm-C are usually more power efficient than Active-RC, but they are also more difficult to make linear.

In many AFEs, or indeed Sigma-Delta modulator loop filters, it's common to find a first Active-RC stage, and then Gm-C for later stages.

GM-C

In the figure below you can see a typical Gm-C filter and the equations for the transfer function. One important thing to note is that this is Gm with capital G, not the g_m that we use for small signal analysis.

In a Gm-C filter the input and output nodes can have significant swing, and thus cannot always be considered small signal.

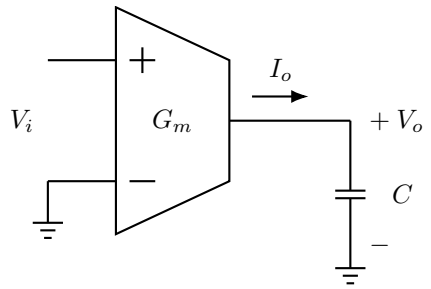

Figure 5: Single-ended Gm-C integrator: a transconductor driving a capacitor

$$V_o = \frac{I_o}{sC} = \frac{\omega_{ti}}{s} V_i$$

$$\omega_{ti} = \frac{G_m}{C}$$

Differential Gm-C

In a real IC we would almost always use differential circuit, as shown below. The transfer function is luckily the same.

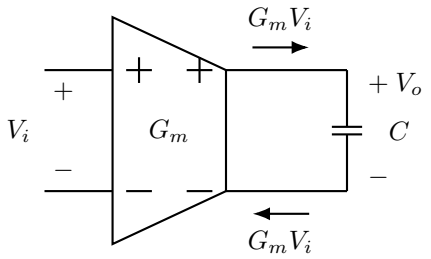

Figure 6: Fully differential Gm-C integrator

$$sCV_o = G_m Vi$$

$$H(s) = \frac{V_o}{V_i} = \frac{G_m}{sC}$$

Differential circuits are fantastic for multiple reasons, power supply rejection ratio, noise immunity, symmetric non-linearity, but the qualities I like the most is that the outputs can be flipped to implement negative, or positive gain.

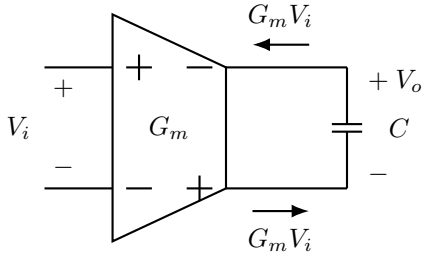

Figure 7: Differential Gm-C integrator with outputs flipped to implement negative gain

$$H(s) = \frac{V_o}{V_i} = -\frac{G_m}{sC}$$

The figure below shows a implementation of a first-order Gm-C filter that matches our signal flow graph.

I would encourage you to try and calculate the transfer function.

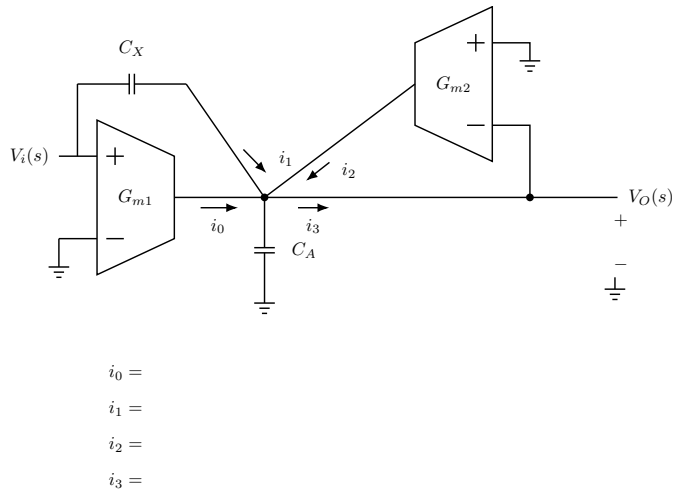

Figure 8: First-order Gm-C filter with transconductors G_{m1}, G_{m2} and capacitors C_X and C_A

Given the transfer function from the signal flow graph, we see that we can select C_x, C_a and G_m to get the desired k's and ω_0

$$H(s) = \frac{k_1 s + k_0}{s + w_o}$$

$$H(s) = \frac{s\frac{C_x}{C_a+C_x} + \frac{G_{m1}}{C_a+C_x}}{s + \frac{G_{m2}}{C_a+C_x}}$$

Below is a general purpose Gm-C bi-quadratic system.

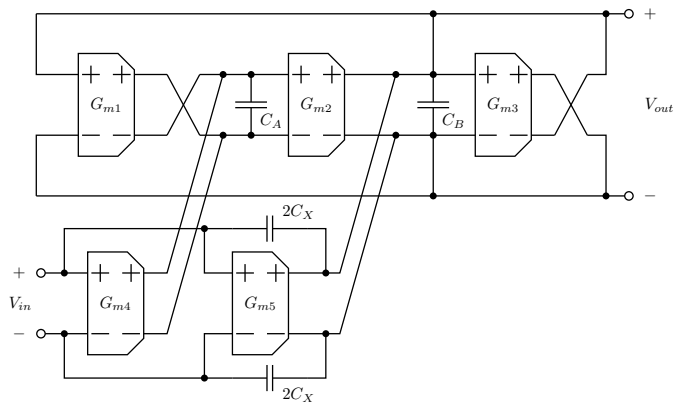

Figure 9: General purpose differential Gm-C biquad with five transconductors

$$H(s) = \frac{k_2 s^2 + k_1 s + k_0}{s^2 + \frac{\omega_0}{Q} s + \omega_o^2}$$

$$H(s) = \frac{s^2\frac{C_X}{C_X+C_B} + s\frac{G_{m5}}{C_X+C_B} + \frac{G_{m2}G_{m4}}{C_A(C_X+C_B)}}{s^2 + s\frac{G_{m3}}{C_X+C_B} + \frac{G_{m1}G_{m2}}{C_A(C_X+C_B)}}$$

Finding a transconductor

Although you can start with the Gm-C cells in the book, I would actually choose to look at a few papers first.

The main reason is that any book is many years old. Ideas turn into papers, papers turn into books, and by the time you read the book, then there might be more optimal circuits for the technology you work in.

If I were to do a design in 22 nm FDSOI I would first see if someone has already done that, and take a look at the strategy they used. If I can't find any in 22 nm FDSOI, then I'd find a technology close to the same supply voltage.

Start with IEEEXplore

I could not find a 22 nm FDSOI Gm-C based circuit on the initial search. If I was to actually make a Gm-C circuit for industry I would probably spend a bit more time to see if any have done it, maybe expanding to other journals or conferences.

I know of Pieter Harpe, and his work is usually superb, so I would take a closer look at A 77.3-dB SNDR 62.5-kHz Bandwidth Continuous-Time Noise-Shaping SAR ADC With Duty-Cycled Gm-C Integrator [1]

And from Figure 10 a) we can see it's a similar Gm-C cell as chapter 12.5.4 in [2].

One of my Ph.d's used the transconductor below on his master thesis Design Considerations for a Low-Power Control-Bounded A/D Converter.

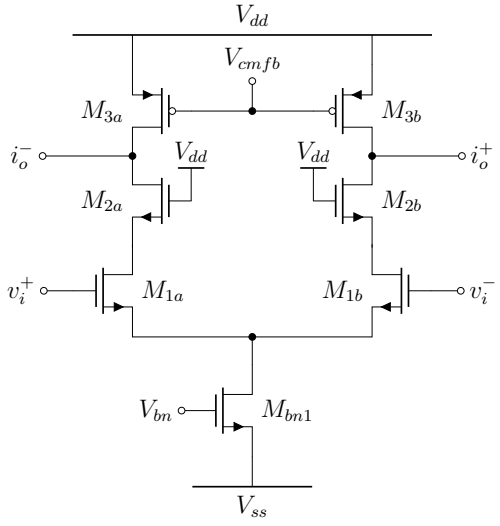

Figure 5.16: Transconductor schematic

Figure 10: Transconductor from the master thesis: a differential pair with common-gate cascodes to limit the Miller effect

ACTIVE-RC

The Active-RC filter should be well known at this point. However, what might be new is how the open loop gain A_0 and unity gain ω_{ta}

General purpose first order filter

Below is a general purpose first order filter and the transfer function. I've used the conductance $G = \frac{1}{R}$ instead of the resistance. The reason is that it sometimes makes the equations easier to work out.

If you're stuck on calculating a transfer function, then try and switch to conductance, and see if it resolves.

I often get my mind mixed up when calculating transfer functions. I don't know if it's only me, but if it's you also, then don't worry, it's not that often you have to work out transfer functions.

Once in a while, however, you will have a problem where you must calculate the transfer function. Sometimes it's because you'll need to understand where the poles/zeros are in a circuit, or you're trying to come up with a clever idea, or I decide to give this exact problem on the exam.

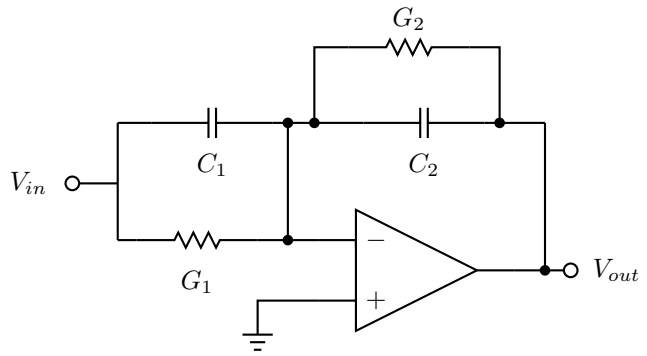

Figure 11: General purpose first-order active-RC filter with parallel RC input and feedback networks

$$H(s) = \frac{k_1 s + k_0}{s + w_o}$$

$$H(s) = \frac{-\frac{C_1}{C_2}s - \frac{G_1}{C_2}}{s + \frac{G_2}{C_2}}$$

Let's work through the calculation.
Step 1: Simplify: The conductance from V_{in} to virtual ground can be written as

$G_{in} = G_1 + sC_1$

The feedback conductance, between V_{out} and virtual ground I write as

$G_{fb} = G_2 + sC_2$
Step 2: Remember how an OTA works: An ideal OTA will force its inputs to be the same. As a result, the potential at OTA$-$ input must be 0.

The input current must then be

$I_{in} = G_{in}V_{in}$

Here it's important to remember that there is no way for the input current to enter the OTA. The OTA is high impedance. The input current must escape through the output conductance G_{fb}.

What actually happens is that the OTA will change the output voltage V_{out} until the feedback current , I_{fb}, exactly matches I_{in}. That's the only way to maintain the virtual ground at 0 V. If the currents do not match, the voltage at virtual ground cannot continue to be 0 V, the voltage must change.
Step 3: Rant a bit: The previous paragraph should trigger your spidy sense. Words like "exactly matches" don't exist in the real world. As such, how closely the currents match must affect the transfer function. The open loop gain A_0 of the OTA matters. How fast the OTA can react to a change in voltage on the virtual ground, approximated by the unity-gain frequency ω_{ta} (the frequency where the gain of the OTA equals 1, or 0 dB), matters. The input impedance of the OTA, whether the gate leakage of the input differential pair due to quantum tunneling, or the capacitance of the input differential pair, matters. How much current the OTA can deliver (set by slew rate), matters.

Active-RC filter design is "How do I design my OTA so it's good enough for the filter". That's also why, for integrated circuits, you will not have a library of OTAs that you just plug in, and they work.

I would be very suspicious of working anywhere that had an OTA library I was supposed to use for integrated filter design. I'm not saying it's impossible that some company actually has an OTA library, but I think it's a bad strategy. First of all, if an OTA is generic enough to be used "everywhere", then the OTA is likely using too much power, consumes too much area, and is too complex. And the company runs the risk that the designer has not really checked that the OTA works properly in the filter because "Someone else designed the OTA, I just used in my design".

But, for now, to make our lives simpler, we assume the OTA is ideal. That makes the equations pretty, and we know what we should get if the OTA actually was ideal.
Step 4: Do the algebra: The current flowing from V_{out} to virtual ground is

$$I_{out} = G_{out}V_{out}$$

The sum of currents into the virtual ground must be zero

$$I_{in} + I_{out} = 0$$

Insert, and do the algebra

$$G_{in}V_{in} + G_{out}V_{out} = 0$$

$$\Rightarrow -G_{in}V_{in} = G_{out}V_{out}$$

$$\frac{V_{out}}{V_{in}} = -\frac{G_{in}}{G_{out}}$$

$$= -\frac{G_1 + sC_1}{G_2 + sC_2}$$

$$= \frac{-s\frac{C_1}{C_2} - \frac{G_1}{C_2}}{s + \frac{G_2}{C_2}}$$

General purpose biquad

A general bi-quadratic active-RC filter is shown below. These kind of general purpose filter sections are quite useful.

Imagine you wanted to make a filter, any filter. You'd decompose into first and second order sections, and then you'd try and match the transfer functions to the general equations.

The interactive biquad is worth a few minutes here: the low-pass, band-pass, high-pass, notch and all-pass responses all share one denominator, and switching between them moves the zeros while the poles stay exactly where they are.

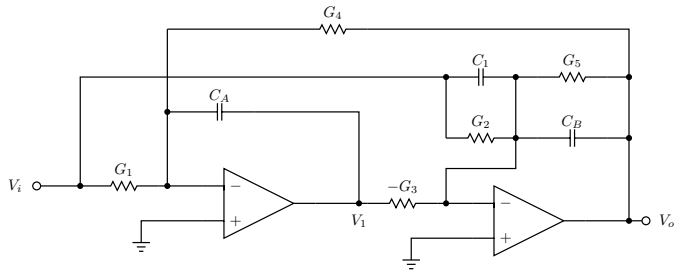

Figure 12: General purpose active-RC biquad built from two OTAs

$$H(s) = \frac{k_2 s^2 + k_1 s + k_0}{s^2 + \frac{\omega_0}{Q} s + \omega_o^2}$$

$$H(s) = \frac{\left[\frac{C_1}{C_B} s^2 + \frac{G_2}{C_B} s + (\frac{G_1 G_3}{C_A C_B})\right]}{\left[s^2 + \frac{G_5}{C_B} s + \frac{G_3 G_4}{C_A C_B}\right]}$$

THE OTA IS NOT IDEAL

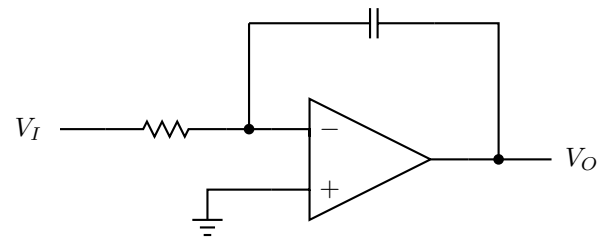

Figure 13: Active-RC integrator: OTA with resistor input and capacitor feedback

$$H(s) \approx \frac{A_0}{(1 + sA_o RC)(1 + \frac{s}{w_{ta}})}$$

where

$$A_0$$

is the gain of the amplifier, and

$$\omega_{ta}$$

is the unity-gain frequency.

At frequencies above $\frac{1}{A_0 RC}$ and below w_{ta} the circuit above is a good approximation of an ideal integrator.

See page 511 in [2] (chapter 5.8.1)

EXAMPLE CIRCUIT

One place where both active-RC and Gm-C filters find a home are continuous time sigma-delta modulators. More on SD later, for now, just know that SD is a combination of high-gain, filtering, simple ADCs and simple DACs to make high resolution analog-to-digital converters.

One such an example is

A 56 mW Continuous-Time Quadrature Cascaded Sigma-Delta Modulator With 77 dB DR in a Near Zero-IF 20 MHz Band

Below we see the actual circuit. It may look complex, and it is.

Not just "complex" as in complicated circuit, it's also "complex" as in "complex numbers".

We can see there are two paths "i" and "q", for "in-phase" and "quadrature-phase". The fantastic thing about complex ADCs is that we can have a-symmetric frequency response around 0 Hz.

It will be tricky understanding circuits like this in the beginning, but know that it is possible, and it does get easier to understand.

With a complex ADC like this, the first thing to understand is the rough structure.

There are two paths, each path contains 2 ADCs connected in series (Multi-stage Noise-Shaping or MASH). Understanding everything at once does not make sense.

Start with "Vpi" and "Vmi", make it into a single path (set Rfb1 and Rfb2 to infinite), ignore what happens after R3 and DAC2i.

Now we have a continuous time sigma delta with two stages. First stage is a integrator (R1 and C1), and second stage is a filter (Cff1, R2 and C2). The amplified and filtered signal is sampled by the ADC1i and fed back to the input DAC1i.

It's possible to show that if the gain from $V(Vpi, Vmi)$ to ADC1i input is large, then $Y1i = V(Vpi, Vmi)$ at low frequencies.

Then put R_{fb1} and R_{fb2} back. Each one takes the *first integrator output* of one path and drives it into the *summing node* of the other - the same virtual ground the input resistor R_1 and the feedback DAC already inject into. That is the whole of the cross-coupling, and it is what turns two real modulators into one complex one.

The figure draws the first stage of the cascade only. What leaves through R_3 and $DAC2$ is the quantization error of this

stage, handed to a second modulator that digitizes it; the digital outputs are then recombined so the first stage's error cancels. That is what MASH means, and it is a separate story from the complex part.

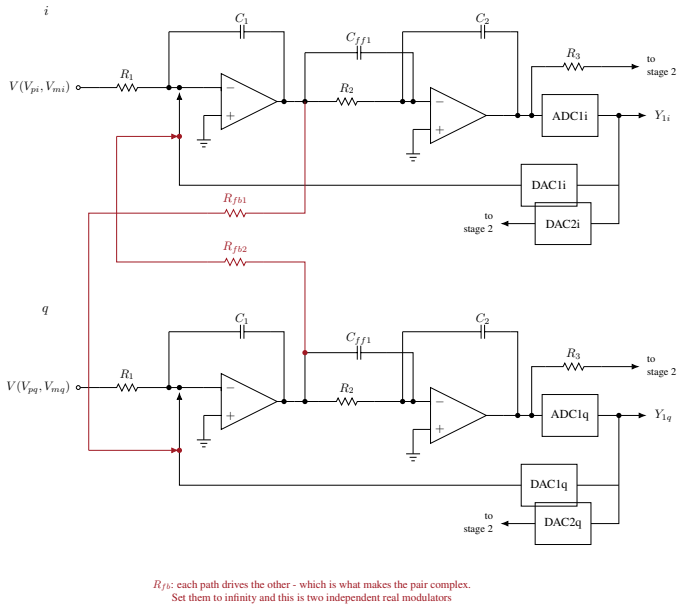

Figure 14: One stage of a continuous-time quadrature sigma-delta modulator, both paths drawn: two ordinary real modulators - R1/C1 integrating, Cff1/R2/C2 filtering, ADC1 sampling, DAC1 feeding back, R3 and DAC2 handing off to the second stage - joined by the red cross-coupling that makes the pair complex: each R_{fb} carries one path's first integrator output into the other path's summing node. Inspired by Breems et al. [3], which is fully differential, cascaded, and has a tuning resistor on everything

What does the cross-coupling actually buy? Figure 15 answers it by simulation. The same first-order loop is run twice: once real, so its coefficients are real and its noise notch must sit symmetrically about zero, and once complex, with the integrator pole rotated to $e^{j\omega_0}$ so the noise transfer function has its zero at $+f_0$ and nowhere else. The complex quantizer is just the two real quantizers, one per path - which is what the i and q ADCs are.

Read the two spectra over the *whole* sample rate, not half of it. The real loop is quiet at DC and equally noisy at $\pm f_0$: a real signal's spectrum is conjugate symmetric and cannot tell $+f$ from $-f$. The complex loop is quiet at $+f_0$ and noisy at $-f_0$, so a wanted channel just above the local oscillator gets the quiet side while its image gets the noise. That asymmetry is the entire reason a near zero-IF receiver bothers with quadrature.

Both simulations are driven by the same input, a complex tone just above f_0, and you can find it in both spectra as the tall spike. In the real loop it stands out in the middle of the noise, where the loop has done nothing for it. In the complex loop it stands in the notch. The signal did not move; the quiet part of the spectrum did.

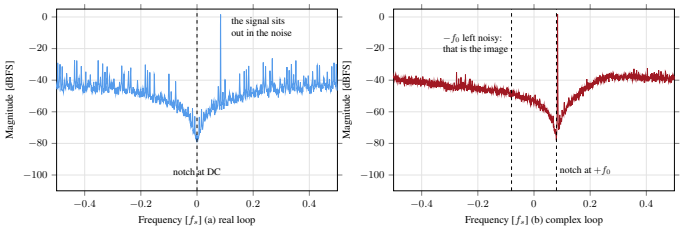

Figure 15: Simulated output spectra of a first-order sigma-delta loop over the full sample rate. (a) A real loop: the noise notch is symmetric about zero. (b) The same loop made complex: the notch moves to $+f_0$ alone, leaving $-f_0$ noisy - which is what lets a near zero-IF receiver keep the wanted side and throw the image away. The same input tone is present in both, and only in (b) does it land in the notch

My favorite OTA

Over the years I've developed a love for the current mirror OTA. A single stage, with load compensation, and an adaptable range of DC gains.

Sometimes simple current mirrors are sufficient, sometimes cascoded, or even active cascodes are necessary.

Below is the differential current mirror OTA.

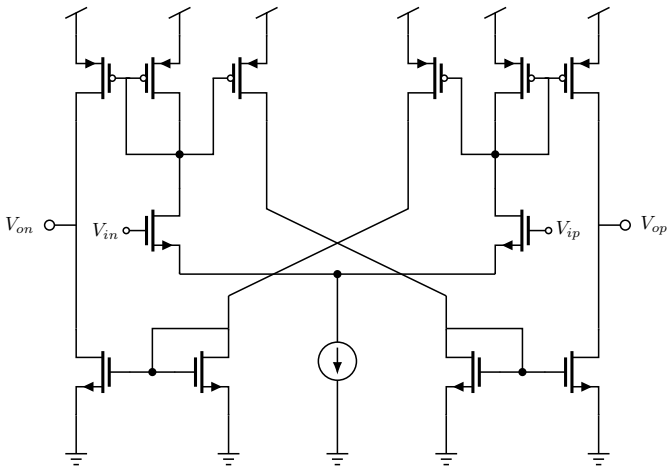

Figure 16: Differential current mirror OTA

In a differential OTA we need to control the output common mode. In order to control the common mode, we must sense the common mode.

Below is a circuit I often use to sense the common mode. Ideally the source followers would be native transistors, but those are not always available.

The reference for the common mode can be from a bandgap, or in the case below, VDD/2.

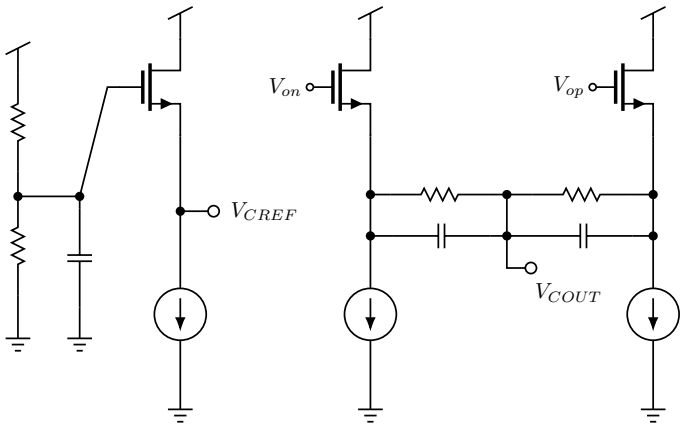

Figure 17: Common mode sense circuit with source followers and the resistor divider generating the reference V_{CREF}

Once we have both the sensed common mode, and the common mode reference, we can use another OTA to control the common mode.

The nice thing about the circuit below is that the common mode feedback loop has the same dominant pole as the differential loop.

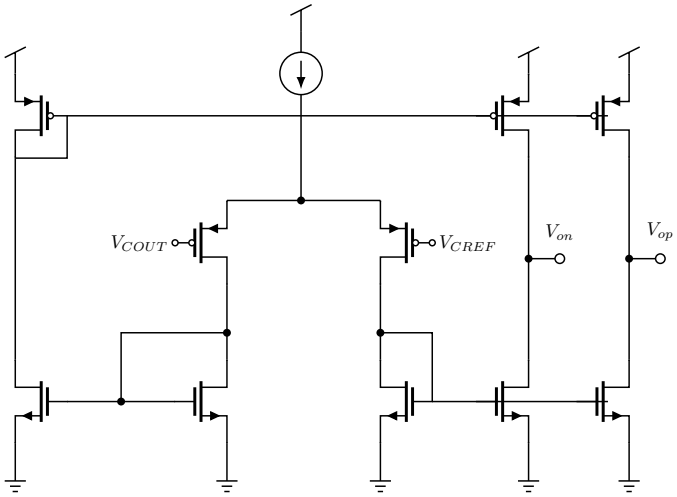

Figure 18: Common mode feedback OTA comparing the sensed V_{COUT} to the reference V_{CREF}

You can find the schematic for the OTA at

CNR_OTA_SKY130NM

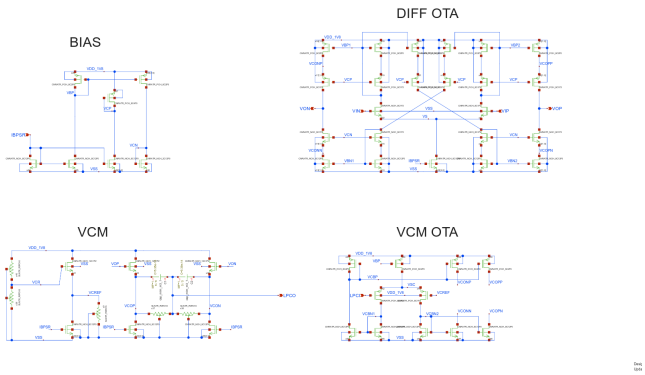

Figure 19: CNR_OTA schematic in SKY130: bias, differential OTA, common mode sense (VCM) and common mode feedback OTA

Summary

The one-page version of this chapter:

- The frontend's job is to hand the ADC a signal it can afford to convert: gain where it is cheap, filtering before sampling folds noise in
- First order sections come from one gm and one C; biquads stack them with feedback and give Q
- Gm-C is fast and open loop, active-RC is linear and closed loop - the OTA pays either way
- A real OTA's finite gain and bandwidth move the filter poles; design the OTA from the filter's error budget
- Fully differential filters double swing and cancel even harmonics, and carry the CMFB tax
- A complex (quadrature) filter is two real paths plus cross-coupling: the response stops being symmetric about zero, which a low-IF radio needs

Would you like to know more?

A 77.3-dB SNDR 62.5-kHz Bandwidth Continuous-Time Noise-Shaping SAR ADC With Duty-Cycled Gm-C Integrator [1]

Design Considerations for a Low-Power Control-Bounded A/D Converter

A 56 mW Continuous-Time Quadrature Cascaded Sigma-Delta Modulator With 77 dB DR in a Near Zero-IF 20 MHz Band [3]

Complex signal processing is not - complex [4]

[1] H. Li, Y. Shen, E. Cantatore, and P. Harpe, "A 77.3-dB SNDR 62.5-kHz bandwidth continuous-time noise-shaping SAR ADC with duty-cycled gm-c integrator," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 4, pp. 939–948, 2023, doi: 10.1109/JSSC.2022.3227678.

[2] T. C. Carusone, D. Johns, and K. Martin, *Analog integrated circuit design*. Wiley, 2011 [Online]. Available: https://books.google.no/books?id=1OIJZzLvVhcC

[3] L. J. Breems, R. Rutten, R. H. M. van Veldhoven, and G. van der Weide, "A 56 mW continuous-time quadrature cascaded ΣΔ modulator with 77 dB DR in a near zero-IF 20 MHz band," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2696–2705, 2007, doi: 10.1109/JSSC.2007.908765.

[4] K. Martin, "Complex signal processing is not - complex," in *ESSCIRC 2004 - 29th european solid-state circuits conference (IEEE cat. no.03EX705)*, 2003, pp. 3–14, doi: 10.1109/ESSCIRC.2003.1257061.

Carsten Wulff received the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electronics and Telecommunication, Norwegian University of Science and Technology (NTNU), in 2002 and 2008, respectively. During his Ph.D. work at NTNU, he worked on open-loop sigma-delta modulators and analog-to-digital converters in nanoscale CMOS technologies. In 2006-2007, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. Since 2008 he's been with Nordic Semiconductor in various roles, from analog designer, to Wireless Group Manager, to currently Principle IC Scientist. From 2014-2017 he did a part time Post.Doc focusing on compiled, ultra low power, SAR ADCs in nanoscale technologies. He's also an Adjunct Associate Professor at NTNU. His present research interests includes analog and mixed-signal CMOS design, design of high-efficiency analog-to-digital converters and low-power wireless transceivers. He is the developer of Custom IC Compiler, a general purpose integrated circuit compiler, and makes the occational video on analog integrated circuits at https://www.youtube.com/@analogicus. For full CV see https://analogicus.com/markdown-cv/.