EE240B Project: Checkpoint 1 & 2

See our Github repo

Harrison Liew & Vighnesh Iyer

April 7, 2019

Contents

1 Checkpoint 1: System-Level Design

1.0 Top-Level Specs

To begin, we transcribe the specs given in the project description into a more formal form to reference later.

$$\omega_{pass} = 20 \text{ Mhz} \tag{1}$$
$$A_{v,dc} = 0 \text{ dB} \tag{1}$$
$$A_v(\omega_{pass}) \geq -3 \text{ dB} \tag{1}$$
$$\omega_{stop} = 200 \text{ Mhz} \tag{2}$$
$$A_v(\omega_{stop}) \leq -55 \text{ dB} \tag{2}$$
$$\text{Group Delay } = \tau_g(\omega) = \frac{d\angle H(j\omega)}{d\omega} \tag{3}$$
$$[\max(\tau_g(\omega)) - \min(\tau_g(\omega))]\Big|_{\omega=0}^{\omega=\omega_{pass}} \leq 3 \text{ ns} \tag{3}$$
$$\text{Dynamic Range } = DR = \frac{P_{sig,max} + P_{noise}(s)}{P_{noise}(s)}\Big|_{\omega=1\cdot 2\pi}^{\omega=\omega_{pass}} \geq 50 \text{ dB} \tag{4}$$
$$C_{load} = 40 \text{ fF} \tag{5}$$
$$V_{dd} = 1.2 \text{ V} \tag{6}$$
$$\text{minimize}(P_{static}) \tag{7}$$
$$\text{Passband Ripple } = Rip_{max} \leq 1 \text{ dB} \tag{8}$$
$$\text{Stable when driven by a 200mV differential step} \tag{9}$$

1.0.1 Methodology

We approached this design problem with a generator-centric methodology. As an even higher-level than using BAG, we incorporate open-source tools based in Python. First, filter transfer functions are synthesized `scipy` and the given filter specs. Concurrently, a real model of the active filter is generated using `ahkab`, which also synthesizes symbolic transfer functions that include OTA nonidealities. We quickly fit the synthesized filter to the circuit topology using naive design equations, and then use a optimizer to refit the filter resistance and capacitance after adjusting for real non-idealities extracted from transistor-level characterization. After constraining the space of design parameters, we verify that the noise and power meet specifications, and iterate as necessary.

1.1 Filter Type Selection

Given the gain and passband spec 1, the stopband spec 2, and the group delay spec 3, we can synthesize transfer functions for various filter types that minimally meet these specs (in terms of order and extension of passband frequency). We perform this iterative synthesis using the `scipy` Python library.

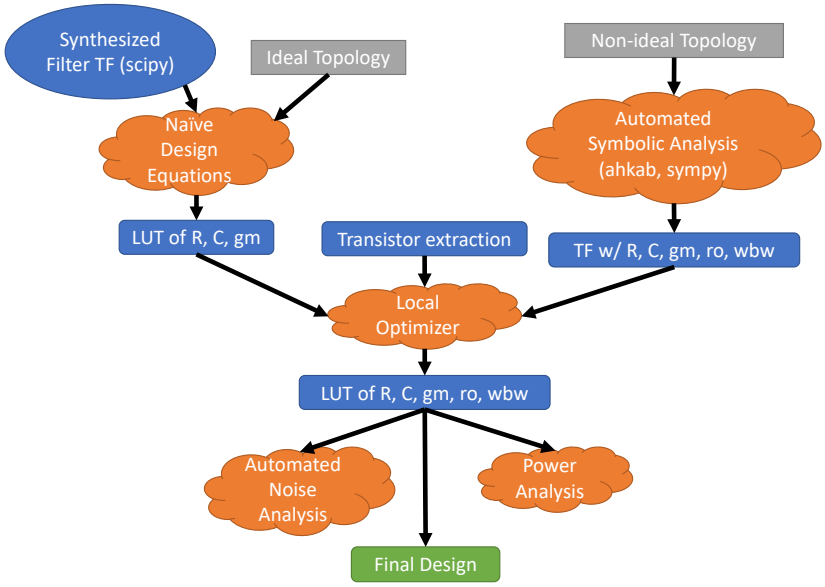


Figure 1: The flow to go from filter specs to a system-level design with parameterized blocks

Algorithm 1 Filter TF Synthesis

procedure SYNTHESIZETF($\omega_{pass}, \omega_{stop}, A_v(\omega_{stop}), Rip_{max}, \tau_{g,max}$)
  $O \leftarrow 1$
  $\omega_{pass,i} \leftarrow \omega_{pass}$
  while True do
    $BA \leftarrow$ SCIPY.IIRFILTER(N $\leftarrow O$, Wn $\leftarrow \omega_{pass,i}$, rp $\leftarrow Rip_{max}$, rs $\leftarrow A_v(\omega_{stop})$)
    if $A_v(BA, \omega_{stop}) < A_v(\omega_{stop})$ **then**
      $O \leftarrow O + 1$ ▷ Increase the filter order if stopband attenuation spec isn't met
    else if $A_v(BA, \omega_{pass}) < -3$ dB **then**
      $\omega_{pass,i} \leftarrow \omega_{pass,i} + 1$ Mhz ▷ Bump the passband corner if there's too much attenuation
    else if Passband Variation($\tau_{g,BA}$) $> \tau_{g,max}$ **then**
      $\omega_{pass,i} \leftarrow \omega_{pass,i} + 1$ Mhz ▷ Bump the passband corner if the group delay variation is excessive
    else
      return BA

This procedure returns a filter transfer function. Using this technique we can find optimal filter polynomial coefficients BA for each filter type. We plot the transfer functions' magnitude gain (Figure 2) and group delay (Figure 3) over frequency to show they meet the specs.

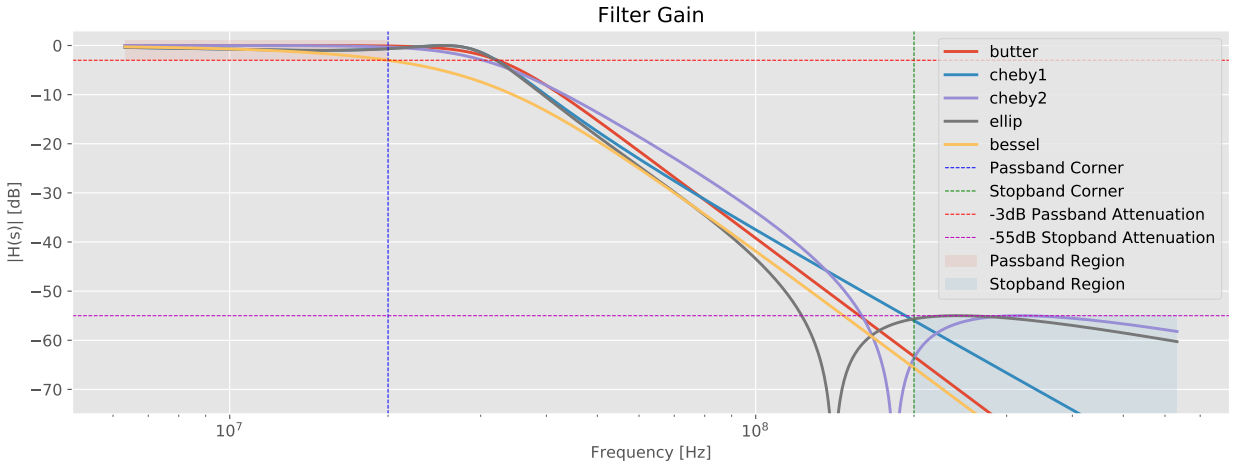


Figure 2: Magnitude gain for synthesized filter transfer functions

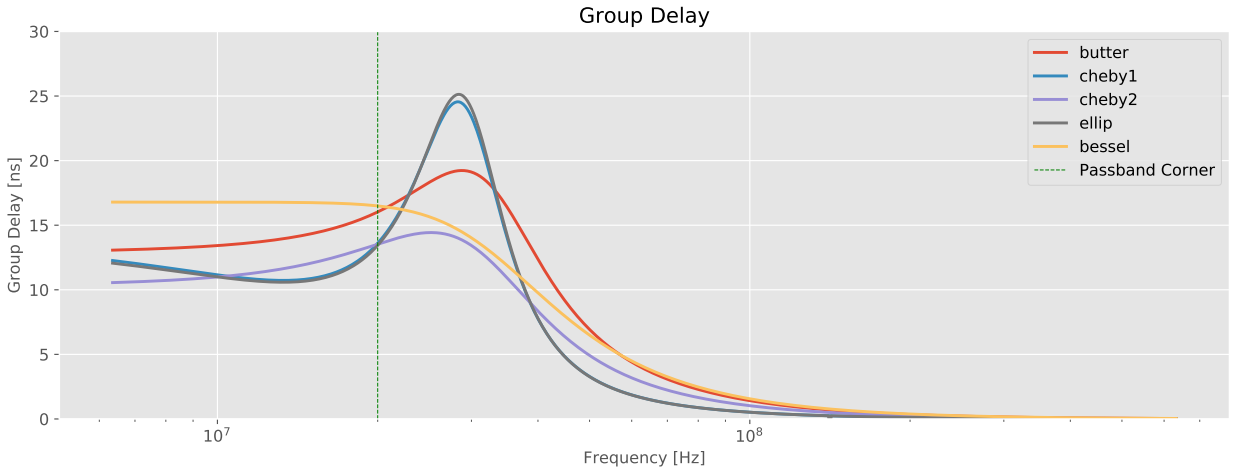


Figure 3: Group delay for synthesized filter transfer functions

We believe the *Butterworth filter* with order = 4 best fits the specifications and is the easist to design for because the passband frequency needs to be minimally pushed out to achieve the group delay spec and contains no zeros in the transfer function. The *Bessel filter* can also work, but may need to be redesigned to push forward the passband corner for the real design to not exceed -3dB attenuation at the corner frequency.

The other filter types either have zeros in the transfer function or have too much group delay variation forcing the passband corner too far up. They have the advantage of only needing an order = 3, but odd orders don't simplify the circuit topology anyways.

1.2 Filter Topology Selection

We initially considered the Sallen-Key and Multiple Feedback topologies using op-amps since they are straightforward to implement and cascading 2 stages will give us 4 poles as desired.

Later, we switched to using OTAs and decided on this simple 2 pole lowpass topology:

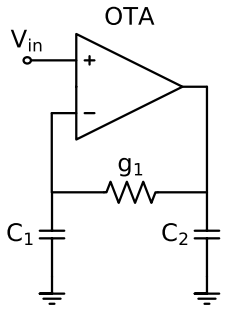


Figure 4: OTA-based lowpass filter with 2 poles and 3 passives, assuming the OTA's $R_{out} = \infty$. The output is taken from the top of C_1

The transfer function is:

$$H_{ideal}(s) = \frac{g_m g_1}{s^2 C_1 C_2 + s g_1 (C_1 + C_2) + g_m g_1}$$

The standard design equations are:

$$\omega_o = \sqrt{\frac{g_m g_1}{C_1 C_2}}$$
$$Q = \sqrt{\frac{g_m}{g_1}} \frac{\sqrt{C_1 C_2}}{C_1 + C_2}$$

Now assuming $C_1 = C_2 = C$, we can spec g_1 and g_m:

$$g2 = \frac{\omega_o C}{2Q}$$
$$g_m = 2Q\omega_o C$$

Since we are using a Butterworth filter, we set $Q = 1/\sqrt{2}$, and we extract ω_o from the transfer function provided by `scipy`.

These design equations are used to generate a table of (R, C, g_m) design points which will be refined in the next step. The rest of the analysis below considers a single-ended filter; a fully differential filter will to first-order have the same transfer function.

1.3 Op-Amp or OTA?

We initially chose op-amps as the active stage, but later realized that the output stage of op-amps is unnecessary for a filter with a capacitive load and just burns extra current. OTAs are more economical and have enough free variables to support the chosen filter family. We plan to cascade 2 of the lowpass OTA circuits (Figure 4) to form the filter.

1.4 Spec Out OTA Stages

The OTA non-idealities we consider are finite output resistance G_o, limited g_m, and finite bandwidth. We still assume that no DC current flows into the OTA.

$$g_m = \frac{g_{m0}}{1 + s/\omega_{bw}}$$
$$G_o = \frac{1}{R_o}$$

We choose to neglect output capacitance C_o and input capacitance C_i, since the parasitic caps are small compared to the filter capacitors and the parasitic caps can be lumped into the filter caps anyways once two filter stages are cascaded.

The transfer function is modified slightly after accounting for the non-idealities:

$$H_{nonideal}(s, r_o, \omega_{bw}) = \frac{g_{m0} r_o}{\left(-\frac{s}{\omega_{bw}} + 1\right)\left(C^2 R r_o s^2 + CRs + 2Cr_o s + \frac{g_{m0} r_o}{-\frac{s}{\omega_{bw}} + 1} + 1.0\right)}$$

The new ω_0 has slightly shifted and the DC gain is now slightly offset from 0 as $\frac{g_{m0} r_o}{1 + g_{m0} r_o}$. The Q factor has increased due to the bandwidth restriction of g_m, but comes at the cost of the poles being shifted towards the RHP. Also the finite output resistance greatly influences the passband loss which means either the filter's passband corner should be bumped up or more bias current is needed to boost the g_m.

We take the table of (R, C, g_m) produced by the design equations in the earlier section and resolve for their optimal values given the OTA non-idealities.

To get an idea of the space of transistor non-idealities we will encounter, we extracted transistor parameters from HW#1. Since most of the design parameters are dependent on the input differential pair, we want to bound all the relevant transistor specs for a variety of transistor bias points. By simulating a L=135nm and W=150nm NMOS over a range of bias current, we grab about 20 bias points (for $V*$ ranging from 0.1 to 0.5V) and generate a LUT containing parameters such as $g_m, r_o, w_b w, I_d, and V*$, all of which will be used for adjusting the optimizer and calculating noise and power.

With a table of realizable r_o and ω_{bw} values, we run a local optimizer which attempts to fit the non-ideal $H(s)$ to the ideal-case $H(s)$.

$$\min_{R,C,g_{m0}} \|H_{ideal}(s) - H_{nonideal}(s, r_o, \omega_{bw})\|_2^2 \Big|_{s=\omega_{pass}-\epsilon}^{s=\omega_{stop}+\epsilon}$$

The initial guess of R and C is provided by the table from section 1.2 and the local optimizer figures out how these variables should be adjusted.

As an example, for these specific non-ideality values and initial guesses for R, C:

$$A_{v0} = 20 \text{ V/V}$$
$$\omega_{bw} = 25.3 \text{ Ghz}$$
$$g_{m0} = 0.744 \text{ mS}$$
$$R = 6 \text{ k}\Omega$$
$$C = 1.32 \text{ pF}$$

We get these modifications of R, C, g_{m0} after optimization:

$$R' = 3.23 \text{ k}\Omega$$
$$C' = 2.82 \text{ pF}$$

1.5 Resistor Sizing and Noise Analysis

For a single OTA filter stage, we can calculate the output noise of R_1 and the OTA itself (thermal and flicker noise of a g_m cell). In the circuit, we model the resistor and OTA's noise as noise currents, for which we calculate a symbolic transfer function for the input-referred noise voltage. In this analysis, we can neglect the OTA's finite bandwidth, as it is too high to shape noise in the passband. We also make the assumption that cascode devices in the OTA do not contribute any noise, and current source/CMFB circuits do not contribute any differential noise.

$$H_{ni,R}(s) = \frac{R\left(sC_2 r_o + 1\right)}{g_m r_o}$$
$$H_{ni,OTA}(s) = \frac{1}{g_m}$$

Intuitively, this makes sense, because R's current noise at DC (where the caps are AC ground) circulates solely in R, and as the frequency increases past the zero created by C_2 and r_o, noise current flows into the OTA as well and gets amplified. The OTA's noise current is simply input-referred by its own transconductance as expected, and is therefore constant with bias. As the g_m of the OTA is decreased, C_2 increases, so we should expect the filter to be noisier at lower OTA bias.

If we use the following noise currents, setting $\gamma = 2$ for a relatively short channel device, and using the optimized R for a given g_m, we integrate over the passband for one stage:

$$v_{ni,T} = \int_{1Hz}^{20MHz} \left[\frac{4kT}{R}|H_{ni,R}(s)|^2 + (4kT\gamma g_m + \frac{K_f I_D}{L^2 C_{ox} f})|H_{ni,OTA}(s)|^2\right] df$$

Now if 2 stages are cascaded, we can similarly calculate the total output noise assuming isolation between the two stages. The first stage's R and OTA will have the same input-referred noise, while the second stage's input-referred noise will be attenuated by Av_0 of the first stage. Since the component values should not change much, and the DC gain is unity, the noise power should be about double that of a single stage.

Taking some values for R and g_m from the previous section, we can calculate the absolute output noise power. With the dynamic range requirement 4, we have a hard limit on this quantity:

$$P_{sig,max} = \frac{V_{sig,max}^2}{2} = \frac{0.2^2}{2} = 0.02W$$
$$P_{ni,max} = \frac{P_{sig,max}}{10^{50/10}} = 200nW$$

We run all the component values in our LUT through this noise calculation to find out which OTA bias points can satisfy the noise spec. We use the following constants in the noise calculations:

$$\gamma = 2$$
$$K_f = 6 \times 10^{-29} \text{ A*F}$$
$$L = 135 \text{ nm}$$
$$C_{ox} = 97 \text{ aF}$$

Figure 5 confirms that at low bias currents, noise is dominated by the filter resistor, and asymptotically approaches the OTA noise at high bias currents. With one stage, we have at least a 20dB margin on the dynamic range. This suggests to us that the choice of bias point is essentially input swing-limited: the V^* must be at least 100mV.

Later on, if we discover that the OTA's noise and nonidealities are much worse than we calculate in this analysis and/or we require additional stages, we can introduce an optimizer once again to adjust all component values at higher OTA bias currents to meet the dynamic range spec.

1.6 Power Estimation

We estimate the static power of the filter circuit as:

$$P = I_{ds} \cdot N_{stages} \cdot N_{branches} \cdot V_{dd} \cdot 2$$

where N_{stages} is 2 representing the number of OTAs, $N_{branches}$ is 2 estimating the number of bias current branches within each OTA, V_{dd} is from the spec, and the factor of 2 accounts for the differential OTA. I_{ds} is derived using 1-transistor simulations and it is collected along with all the OTA non-idealities we considered earlier.

We find the following plot:

All the noise numbers meet our required spec, so we are primarily concerned with getting the required output swing and minimizing distortion. To that end, we can increase V^* and by consequence I_{ds} incrementally until those specs are met without worrying too much about noise.

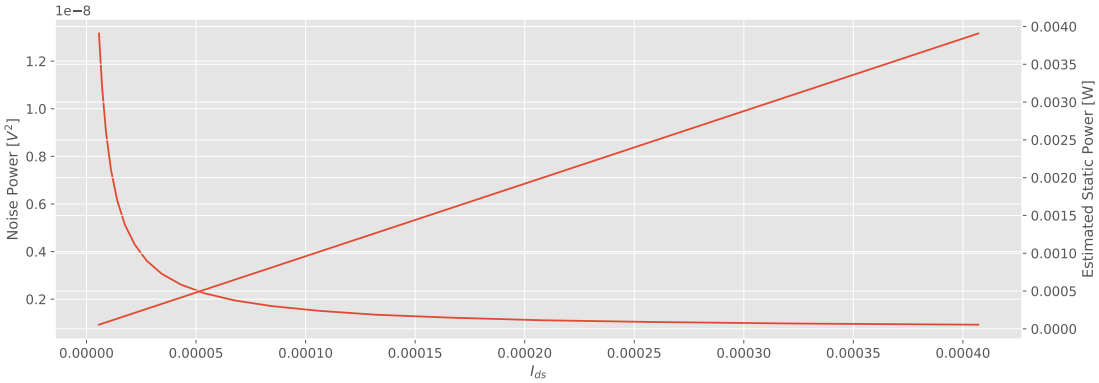


Figure 5: Considering various I_{ds} values for a fixed dimension transistor, we find the total input referred noise and estimated power consumption

1.7 Chosen Design Parameters

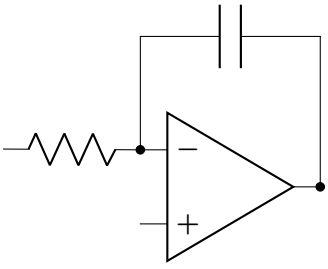
Here is one design point which meets our specs:

$$
\begin{aligned}
V^* &= 0.115 \text{ V} \\
I_{ds} &= 112\mu\text{A} \\
g_m &= 0.195 \text{ mS} \\
r_o &= 12.7\text{k}\Omega \\
w_{bw} &= 43.3 \text{ GHz} \\
R &= 12 \text{ k}\Omega \\
C &= 746.25 \text{ fF}
\end{aligned}
$$

This is the output of our optimization program:

```
Total input referred noise power: 7.424237673893265e-09 V^2
Min reqd voltage swing for DR: 0.0385337194516524 V
Passes dynamic range!
Power: 0.00010751999999999999 W
```

2 Checkpoint 2: OTA Design



3 References

We found the book, "Continuous-Time Active Filter Design" by T. Deliyannis et al. to be very useful in deriving simple design equations and choosing topologies. Ahkab, a symbolic Python circuit simulator, helped us greatly with more complex calculations such as noise analysis.