EE247
Lecture 3

• Last lecture’s summary
• Active Filters
  – Active biquads
    • Sallen-Key & Tow-Thomas
    • Integrator-based filters
      – Signal flowgraph concept
      – First order integrator-based filter
      – Second order integrator-based filter & biquads
  – High order & high Q filters
    • Cascaded biquads
      – Cascaded biquad sensitivity to component variations
    • Ladder type filters

Summary of Last Lecture

– Nomenclature
– Filter specifications
  • Quality factor
  • Frequency characteristics
  • Group delay
– Filter types
  • Butterworth
  • Chebyshev I
  • Chebyshev II
  • Elliptic
  • Bessel
– Group delay comparison example
– RLC filters
Integrated Filters

- Implementation of RLC filters in CMOS technologies requires on-chip inductors
  - Integrated L<10nH with Q<10
  - Combined with max. cap. 10pF
  \( \Rightarrow \) LC filters in the monolithic form feasible: freq>500MHz

- Analog/Digital interface circuitry require fully integrated filters with critical frequencies << 500MHz
- Hence:
  \( \Rightarrow \) Need to build active filters built without inductors

Filters

2\textsuperscript{nd} Order Transfer Functions (Biquads)

- Biquadratic (2\textsuperscript{nd} order) transfer function:

\[
H(s) = \frac{1}{1+\frac{s}{\omega_p Q_p} + \frac{s^2}{\omega_p^2}}
\]

\[
|H(j\omega)|_{\omega=0} = 1\quad |H(j\omega)|_{\omega=\omega_p} = 0\quad |H(j\omega)|_{\omega=\omega_p} = Q_p
\]

Biquad poles @:

\[
\omega = \frac{\omega_p}{2Q_p} \left( 1 \pm \sqrt{1-4Q_p^2} \right)
\]

for \( Q_p \leq \frac{1}{2} \) poles are real, complex otherwise
Biquad Complex Poles

\[ Q_P > \frac{1}{2} \quad \rightarrow \quad \text{Complex conjugate poles:} \]

\[
s = -\frac{\omega_p}{2Q_P} \left(1 \pm j\sqrt{4Q_P^2 - 1}\right)
\]

Distance from origin in s-plane:

\[
d^2 = \left(\frac{\omega_p}{2Q_P}\right)^2 (1 + 4Q_P^2 - 1) = \omega_p^2
\]
Implementation of Biquads

- Passive RC: only real poles - can’t implement complex conjugate poles

- Terminated LC
  - Low power, since it is passive
  - Only fundamental noise sources → load and source resistance
  - As previously analyzed, not feasible in the monolithic form for $f < 500\text{MHz}$

- Active Biquads
  - Many topologies can be found in filter textbooks!
  - Widely used topologies:
    - Single-opamp biquad: *Sallen-Key*
    - Multi-opamp biquad: *Tow-Thomas*
    - Integrator based biquads

Active Biquad
Sallen-Key Low-Pass Filter

- Single gain element
- Can be implemented both in discrete & monolithic form
- "Parasitic sensitive"
- Versions for LPF, HPF, BP, ...

→ Advantage: Only one opamp used
→ Disadvantage: Sensitive to parasitic – all pole no zeros

\[ H(s) = \frac{G}{1 + \frac{s}{\omega_p Q_p} + \frac{s^2}{\omega^2 p}} \]

\[ \omega_p = \frac{1}{\sqrt{R_1 C_1 R_2 C_2}} \]

\[ Q_p = \frac{\omega_p}{\frac{1}{R_1 C_1} + \frac{1}{R_2 C_1} + \frac{1}{R_2 C_2}} \]
Addition of Imaginary Axis Zeros

- Sharpen transition band
- Can "notch out" interference
- High-pass filter (HPF)
- Band-reject filter

$$H(s) = K \frac{1 + \left(\frac{s}{\omega_Z}\right)^2}{1 + \frac{s}{\omega_p Q_p} + \left(\frac{s}{\omega_p}\right)^2}$$

$$|H(j\omega)|_{\omega \to \infty} = K\left(\frac{\omega_p}{\omega_Z}\right)^2$$

**Note:** Always represent transfer functions as a product of a gain term, poles, and zeros (pairs if complex). Then all coefficients have a physical meaning, and readily identifiable units.

---

Imaginary Zeros

- Zeros substantially sharpen transition band
- At the expense of reduced stop-band attenuation at high frequency

$$f_p = 100kHz$$
$$Q_p = 2$$
$$f_z = 3f_p$$

[Diagram showing pole-zero map with frequency and magnitude plots]
Moving the Zeros

\[ f_p = 100kHz \]
\[ Q_p = 2 \]
\[ f_z = f_p \]

![Pole-Zero Map](image)

Tow-Thomas Active Biquad

- Parasitic insensitive
- Multiple outputs

Frequency Response

\[ \frac{V_{o1}}{V_{in}} = -k_2 \frac{(b_2a_1 - b_1)s + (b_2a_0 - b_0)}{s^2 + a_1s + a_0} \]

\[ \frac{V_{o2}}{V_{in}} = \frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \]

\[ \frac{V_{o3}}{V_{in}} = -\frac{1}{k_1\sqrt{a_0}} \frac{(b_0 - b_2a_0)s + (a_1b_0 - a_0b_1)}{s^2 + a_1s + a_0} \]

- \(V_{o2}\) implements a general biquad section with arbitrary poles and zeros
- \(V_{o1}\) and \(V_{o3}\) realize the same poles but are limited to at most one finite zero

---

Component Values

\[ b_i = \frac{R_i}{R_i R_i R_i C_i C_i} \]
\[ b_1 = \frac{1}{R_i C_i} \left( \frac{R_i}{R_i} \right) \frac{R_i}{R_i R_i} \]
\[ b_2 = \frac{R_i}{R_i} \]
\[ a_i = \frac{R_i}{R_i R_i C_i C_i} \]
\[ a_1 = \frac{1}{R_i C_i} \]
\[ a_2 = \frac{1}{R_i R_i C_i} \frac{R_i}{R_i R_i} \]
\[ k_i = \frac{R_i}{R_i R_i C_i} \]
\[ k_1 = \frac{R_i}{R_i} \]
\[ k_2 = \frac{R_i}{R_i} \]

Given \(a_i, b_i, k_i, C_i, C_2\) and \(R_i\), it follows that

\[ \omega_o = \frac{R_i}{\sqrt{R_i R_i R_i C_i C_i C_i}} \]
\[ Q_i = \frac{a_i R_i C_i}{k_i R_i} \]
\[ Q_i = \frac{a_i R_i C_i}{k_i R_i} \]
\[ Q_i = \frac{a_i R_i C_i}{k_i R_i} \]
\[ Q_i = \frac{a_i R_i C_i}{k_i R_i} \]
Higher-Order Filters

- Higher-order filters (N>2) can be built with cascade of 2nd order biquads, e.g. Sallen-Key, or Tow-Thomas

\[ \text{2nd order Filter} \rightarrow \text{2nd order Filter} \rightarrow \cdots \rightarrow \text{2nd order Filter} \]

\[ N \text{ 2nd order sections} \rightarrow \text{Filter with 2N order} \]

As will be shown later:
- High-Q high-order filters built with cascade of 2nd order sections
- Highly sensitive to component variations
- Good alternative: Integrator-based ladder type filters

Integrator Based Filters

- Main building block for this category of filters \( \rightarrow \) integrator
- By using signal flowgraph techniques \( \rightarrow \) conventional filter topologies can be converted to integrator based type filters
- Next few pages:
  - Signal flowgraph techniques
  - 1st order integrator based filter
  - 2nd order integrator based filter
  - High order and high Q filters
What is a Signal Flowgraph (SFG)?

- SFG → Topological network representation consisting of nodes & branches - used to convert one form of network to a more suitable form (e.g. passive RLC filters to integrator based filters)
- Any network described by a set of linear differential equations can be expressed in SFG form
- For a given network, many different SFGs exists
- Choice of a particular SFG is based on practical considerations such as type of available components

Signal Flowgraph (SFG) Examples

<table>
<thead>
<tr>
<th>Circuit</th>
<th>State-space description</th>
<th>SFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{in}$</td>
<td>$I_{in} \times R = V_o$</td>
<td>$I_{in} \rightarrow V_o$</td>
</tr>
<tr>
<td>$V_{in}$</td>
<td>$V_{in} \times \frac{I}{SL} = I_o$</td>
<td>$V_{in} \rightarrow I_o$</td>
</tr>
<tr>
<td>$I_{in}$</td>
<td>$I_{in} \times \frac{I}{SC} = V_o$</td>
<td>$I_{in} \rightarrow V_o$</td>
</tr>
</tbody>
</table>

Signal Flowgraph (SFG) Rules

- Two parallel branches can be replaced by a single branch with overall BMF equal to sum of two BMFs
  
  $V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} b \end{array} \begin{array}{c} a \end{array} V_2 \quad \Rightarrow \quad V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} a+b \end{array} V_2$

- A node with only one incoming branch & one outgoing branch can be replaced by a single branch with BMF equal to the product of the two BMFs
  
  $V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} a \end{array} \begin{array}{c} b \end{array} V_2 \quad \Rightarrow \quad V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} a.b \end{array} V_2$

- An intermediate node can be multiplied by a factor ($k$). BMFs for incoming branches have to be multiplied by $k$ and outgoing branches divided by $k$
  
  $V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} a \end{array} \begin{array}{c} b \end{array} V_2 \quad \Rightarrow \quad V_1 \begin{array}{c} \circ \end{array} \begin{array}{c} \frac{k.a}{k} \end{array} \begin{array}{c} \frac{b}{k} \end{array} V_2$
Signal Flowgraph (SFG) Rules

- Simplifications can often be achieved by shifting or eliminating nodes

\[
\begin{align*}
V_0 & \quad l \quad V_2 \quad a \quad V_3 \quad V_4 \\
& \quad -1 \quad 1 \quad 1 \quad V_o \\
\end{align*}
\]

- A self-loop branch with BMF \( y \) can be eliminated by multiplying the BMF of incoming branches by \( 1/(1-y) \)

\[
\begin{align*}
V_0 & \quad l \quad V_2 \quad a \quad V_3 \quad V_4 \\
& \quad l/b \quad -1/b \quad 1 \quad V_o \\
\end{align*}
\]

Integrator Based Filters

1st Order LPF

- Conversion of simple lowpass RC filter to integrator-based type by using signal flowgraph technique

\[
\frac{V_o}{V_{in}} = \frac{1}{1+sRC}
\]
What is an Integrator?
Example: Single-Ended Opamp-RC Integrator

\[ V_o = -V_{in} \frac{1}{sRC} \quad V_o = -\frac{1}{RC} \int V_{in} \, dt \]

\[ \tau = RC \]

Note: Practical integrator in CMOS technology has input & output both in the form of voltage and not current \( \rightarrow \) Consideration for SFG derivation

Integrator Based Filters
1st Order LPF

1. Start from circuit prototype-
   - Name voltages & currents for all components

2. Use KCL & KVL to derive state space description- to have BMFs in the integrator form
   - Capacitor voltage expressed as function of its current \( V_{Cap} = f(I_{Cap}) \)
   - Inductor current as a function of its voltage \( I_{ind} = f(V_{ind}) \)

3. Use state space description to draw signal flowgraph (SFG) (see next page)
Integrator Based Filters
First Order LPF

\[ V_I = V_{in} - V_C \]
\[ V_C = I_2 \cdot \frac{1}{sC} \]
\[ V_o = V_C \]
\[ I_1 = V_I \cdot \frac{1}{Rs} \]
\[ I_2 = I_1 \]

- All voltages & currents \( \rightarrow \) nodes of SFG
- Voltage nodes on top, corresponding current nodes below each voltage node

Normalize

- Since integrators - the main building blocks - require in & out signals in the voltage form (not current)
  \( \rightarrow \) Convert all currents to voltages by multiplying current nodes by a scaling resistance \( R' \)
  \( \rightarrow \) Corresponding BMFs should then be scaled accordingly

\[ V_I = V_{in} - V_o \]
\[ I_1 = \frac{V_I}{Rs} \]
\[ V_o = \frac{I_2}{sC} \]
\[ I_2 = I_1 \]
\[ I_1\cdot R' = \frac{R'}{Rs} \cdot V_I \]
\[ I_2\cdot R' = \frac{I_2\cdot R'}{sC} \]
\[ V_x' = I_3\cdot R' \]
\[ V_I' = \frac{R'}{Rs} \cdot V_I \]
\[ V_o' = \frac{V_o'}{sC \cdot R'} \]
\[ V_2' = V_{I_1}' \]
Normalize

\[
\frac{1}{R_s} + \frac{1}{sC} \quad \rightarrow \quad \frac{1}{R_s} \quad \rightarrow \quad \frac{1}{sC R^2}
\]

Synthesis

Choosing

\[ R = R_s \]

Consolidate two branches

\[ R^* = R_s, \quad \tau = R^* \times C \]
First Order Integrator Based Filter

\[ V_{in} - I_{I} - l \rightarrow V_{o} \]

\[ V_{2} \]

\[ \frac{1}{\tau \times s} \int + \]

\[ V_{in} \]

\[ H(s) = \frac{1}{\tau \times s} \]

1st Order Filter
Built with Opamp-RC Integrator

- Single-ended Opamp-RC integrator has a sign inversion from input to output
  - Convert SFG accordingly by modifying BMF

\[ V_{in} \rightarrow V_{o} \]

\[ V_{in} \rightarrow V_{o} \]

\[ V_{in} = -V_{in} \]
1\textsuperscript{st} Order Filter
Built with Opamp-RC Integrator

- To avoid requiring an additional opamp to perform summation at the input node:

\[
V_{in}' = -V_{in}
\]

1\textsuperscript{st} Order Filter
Built with Opamp-RC Integrator (continued)

\[
\frac{V_o}{V_{in}} = \frac{I}{I + sRC}
\]
Opamp-RC 1st Order Filter Noise

Identify noise sources (here it is resistors & opamp)
Find transfer function from each noise source to the output (opamp noise next page)

\[
\overline{v_n^2} = \sum_{m=1}^{\infty} \int \frac{H_m(f)}{R C} S_m(f) \, df
\]

\( \text{Noise spectral density of m\textsuperscript{th} noise source} \)

\( \text{Typically, } \alpha \text{ increases as filter order increases} \)

Opamp-RC Filter Noise

Opamp Contribution

- So far only the fundamental noise sources are considered.

- In reality, noise associated with the opamp increases the overall noise.

- For a well-designed filter opamp is designed such that noise contribution of opamp << contribution of other noise sources

- The bandwidth of the opamp affects the opamp noise contribution to the total noise
Integrator Based Filter
2nd Order RLC Filter

• State space description:

\[
\begin{align*}
V_R & = V_L = V_C = V_o \\
V_C & = \frac{I_C}{sC} \\
I_R & = \frac{V_R}{R} \\
I_L & = \frac{V_L}{sL} \\
I_C & = I_{in} - I_R - I_L
\end{align*}
\]

Integrator form

• Draw signal flowgraph (SFG)

Normalize

• Convert currents to voltages by multiplying all current nodes by the scaling resistance \( R^* \)
Synthesis

\[ V_1 \rightarrow \frac{1}{R'} \frac{1}{sCR'} \rightarrow \frac{1}{sL} \rightarrow V_o \]

\[ \tau_1 = R' \times C \quad \tau_2 = \frac{L}{R'} \]

Second Order Integrator Based Filter

Filter Magnitude Response

Normalized Frequency [Hz]

Magnitude [dB]
Second Order Integrator Based Filter

\[
\begin{align*}
\frac{V_{BP}}{V_{in}} &= \frac{\tau_2 s + \beta}{\tau_1 s^2 + \beta s + 1} \\
\frac{V_{LP}}{V_{in}} &= \frac{1}{\tau_1 s^2 + \beta s + 1} \\
\frac{V_{HP}}{V_{in}} &= \frac{\tau_2 s + \beta}{\tau_1 s^2 + \beta s + 1}
\end{align*}
\]

\[
\tau_1 = R \times C \\
\tau_2 = L / R
\]

From matching point of view desirable:
\[
\tau_1 = \tau_2 \rightarrow Q = R / R
\]

Second Order Bandpass Filter Noise

\[
\bar{v}_n^2 = \sum_{m=1}^{\infty} W_m(f) \delta^2 S_n(f) df
\]

- Find transfer function of each noise source to the output
- Integrate contribution of all noise sources
- Here it is assumed that opamps are noise free (not usually the case!)

\[
v_{n1}^2 = v_{n2}^2 = 4KTRdf
\]

\[
\sqrt{\bar{v}_n} = \sqrt{\frac{2QkT}{C}}
\]

Typically, \(\alpha\) increases as filter order increases

Note the noise power is directly proportion to \(Q\)
Second Order Integrator Based Filter

Biquad

By combining outputs can generate general biquad function:

\[
\frac{V_0}{V_{in}} = \frac{a_1 \tau_1 \tau_2 s^2 + a_2 \tau_2 s + a_3}{\tau_1 \tau_2 s^2 + \beta \tau_2 s + 1}
\]

\[ j \omega \]

\[ s \text{-plane} \]

Summary

Integrator Based Monolithic Filters

Signal flowgraph techniques utilized to convert RLC networks to integrator based active filters

Each reactive element (L & C) replaced by an integrator

Fundamental noise limitation determined by integrating capacitor value:

- For lowpass filter:
  \[ \sqrt{\frac{\tau_2}{\tau}} V_0 = \sqrt{\frac{\alpha k T}{C}} \]

- Bandpass filter:
  \[ \sqrt{\frac{\tau_2}{\tau}} V_0 = \sqrt{\alpha Q \frac{k T}{C}} \]

where \( \alpha \) is a function of filter order and topology
Higher Order Filters

• How do we build higher order filters?
  – Cascade of biquads and 1st order sections
    • Each complex conjugate pole built with a biquad and real pole
      with 1st order section
    • Easy to implement
    • In the case of high order high Q filters → highly sensitive to
      component variations
  – Direct conversion of high order ladder type RLC filters
    • SFG techniques used to perform exact conversion of ladder
      type filters to integrator based filters
    • More complicated conversion process
    • Much less sensitive to component variations compared to
      cascade of biquads

• Lecture 3 ended here
Higher Order Filters
Cascade of Biquads

Example: LPF filter for CDMA baseband receiver
- LPF with
  - fpass = 650 kHz, Rpass = 0.2 dB
  - fstop = 750 kHz, Rstop = 45 dB
  - Assumption: Can compensate for phase distortion in the digital domain
- 7th order Elliptic Filter
- Implementation with cascaded Biquads
  Goal: Maximize dynamic range
  - Pair poles and zeros
  - Highest Q poles with closest zeros is a good starting point, but not necessarily optimum
  - Ordering:
    Lowest Q poles first is a good start

Filter Overall Frequency Response

Bode Diagram
Pole-Zero Map

<table>
<thead>
<tr>
<th>Qpole</th>
<th>f_{pole} [kHz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.7902</td>
<td>659.496</td>
</tr>
<tr>
<td>3.6590</td>
<td>611.744</td>
</tr>
<tr>
<td>1.1026</td>
<td>473.643</td>
</tr>
<tr>
<td>319.568</td>
<td>319.568</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f_{zero} [kHz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1297.5</td>
</tr>
<tr>
<td>836.6</td>
</tr>
<tr>
<td>744.0</td>
</tr>
</tbody>
</table>

CDMA Filter
Built with Cascade of 1\textsuperscript{st} and 2\textsuperscript{nd} Order Sections

• 1\textsuperscript{st} order filter implements the single real pole
• Each biquad implements a pair of complex conjugate pole and a pair of imaginary zero
Intermediate Outputs

<table>
<thead>
<tr>
<th>Frequency [Hz]</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10kHz</td>
<td>0</td>
</tr>
<tr>
<td>100kHz</td>
<td>-20</td>
</tr>
<tr>
<td>1MHz</td>
<td>-40</td>
</tr>
<tr>
<td>10MHz</td>
<td>-60</td>
</tr>
</tbody>
</table>

Magnitude (dB) for LPF1 + Biquad 2

<table>
<thead>
<tr>
<th>Frequency [Hz]</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10kHz</td>
<td>0</td>
</tr>
<tr>
<td>100kHz</td>
<td>-20</td>
</tr>
<tr>
<td>1MHz</td>
<td>-40</td>
</tr>
<tr>
<td>10MHz</td>
<td>-60</td>
</tr>
</tbody>
</table>

Magnitude (dB) for LPF1 + Biquads 1, 2, 3, & 4

<table>
<thead>
<tr>
<th>Frequency [Hz]</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10kHz</td>
<td>0</td>
</tr>
<tr>
<td>100kHz</td>
<td>-20</td>
</tr>
<tr>
<td>1MHz</td>
<td>-40</td>
</tr>
<tr>
<td>10MHz</td>
<td>-60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency [Hz]</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10kHz</td>
<td>0</td>
</tr>
<tr>
<td>100kHz</td>
<td>-20</td>
</tr>
<tr>
<td>1MHz</td>
<td>-40</td>
</tr>
<tr>
<td>10MHz</td>
<td>-60</td>
</tr>
</tbody>
</table>

Magnitude (dB) for LPF1 + Biquads 2, 3, 4

<table>
<thead>
<tr>
<th>Frequency [Hz]</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10kHz</td>
<td>0</td>
</tr>
<tr>
<td>100kHz</td>
<td>-20</td>
</tr>
<tr>
<td>1MHz</td>
<td>-40</td>
</tr>
<tr>
<td>10MHz</td>
<td>-60</td>
</tr>
</tbody>
</table>

Sensitivity

Component variation in Biquad 4 (highest Q pole):
- Increase $\omega_p$ by 1%
- Decrease $\omega_z$ by 1%

High Q poles $\rightarrow$ High sensitivity in Biquad realizations