Continuous-Time Signals and Systems

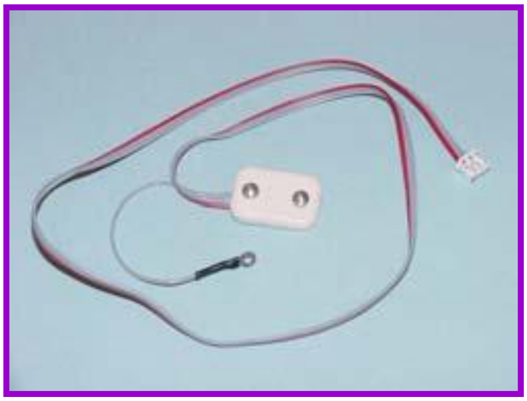
Second-Order Sallen-Key Circuit Analysis

Edward (Ted) A. Clancy

Department of Electrical and Computer Engineering
Department of Biomedical Engineering
Worcester Polytechnic Institute
Worcester, MA U.S.A.

EMG Amplitude (EMGamp) Estimation

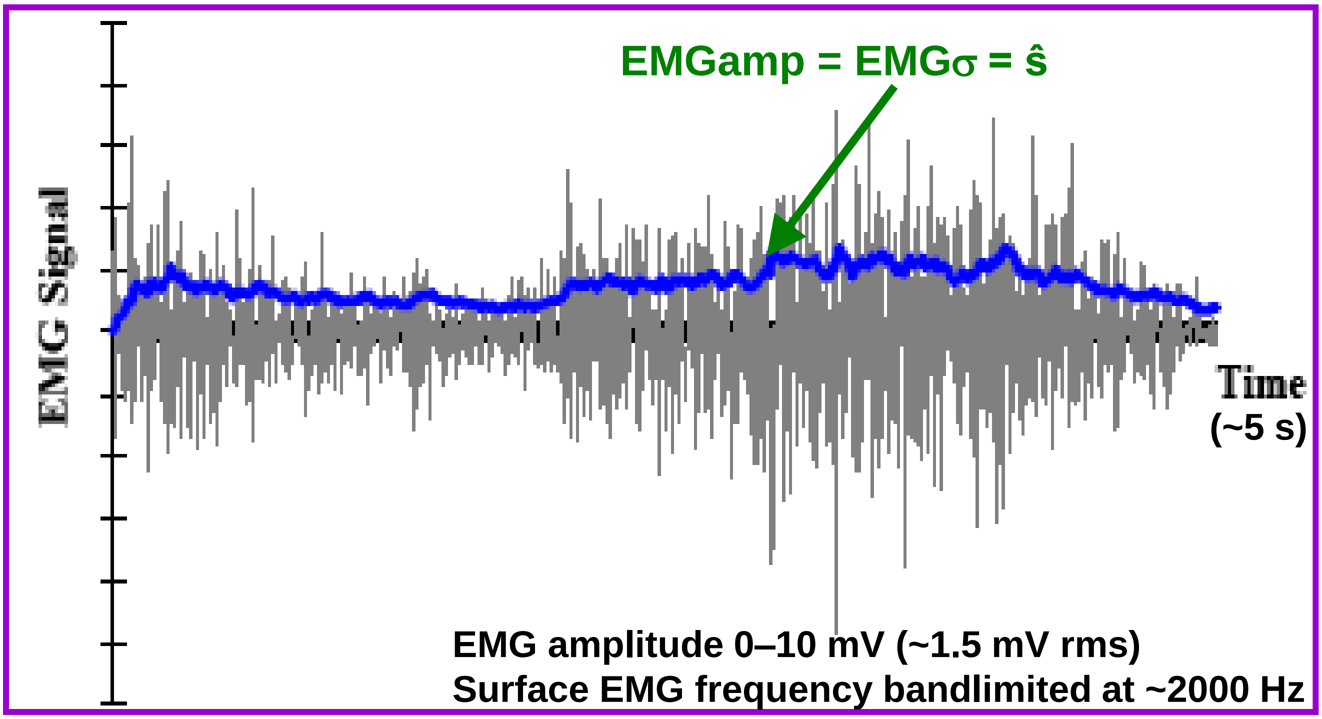
- EMG Amplitude (EMGamp or EMGσ): "Intensity" of recorded EMG
  - Time-varying standard deviation of EMG signal
  - EMG amp varies with muscle tension, localized muscle fatigue
- Original estimator: Inman *et al.* [*EEG Clin Neurophysiol* 4: 187–194, 1952]
  - Analog full-wave rectify and RC low pass filter



Electrode-Amplifier

[Liberating Technologies]

EMG signal well modeled as amplitude-modulated, random process. EMGamp is the modulation.



Surface EMG Recordings (2)

Clancy, Ph.D., 1991, Fig. 1.1

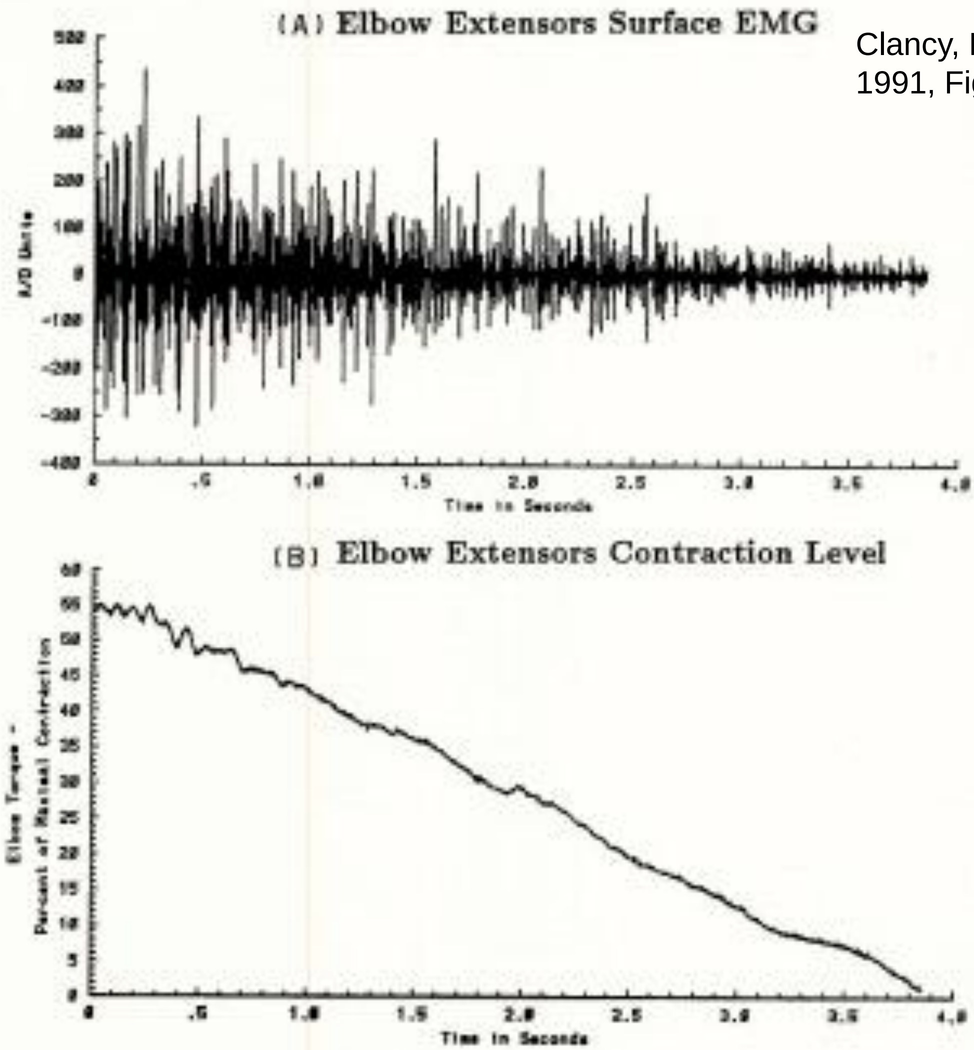


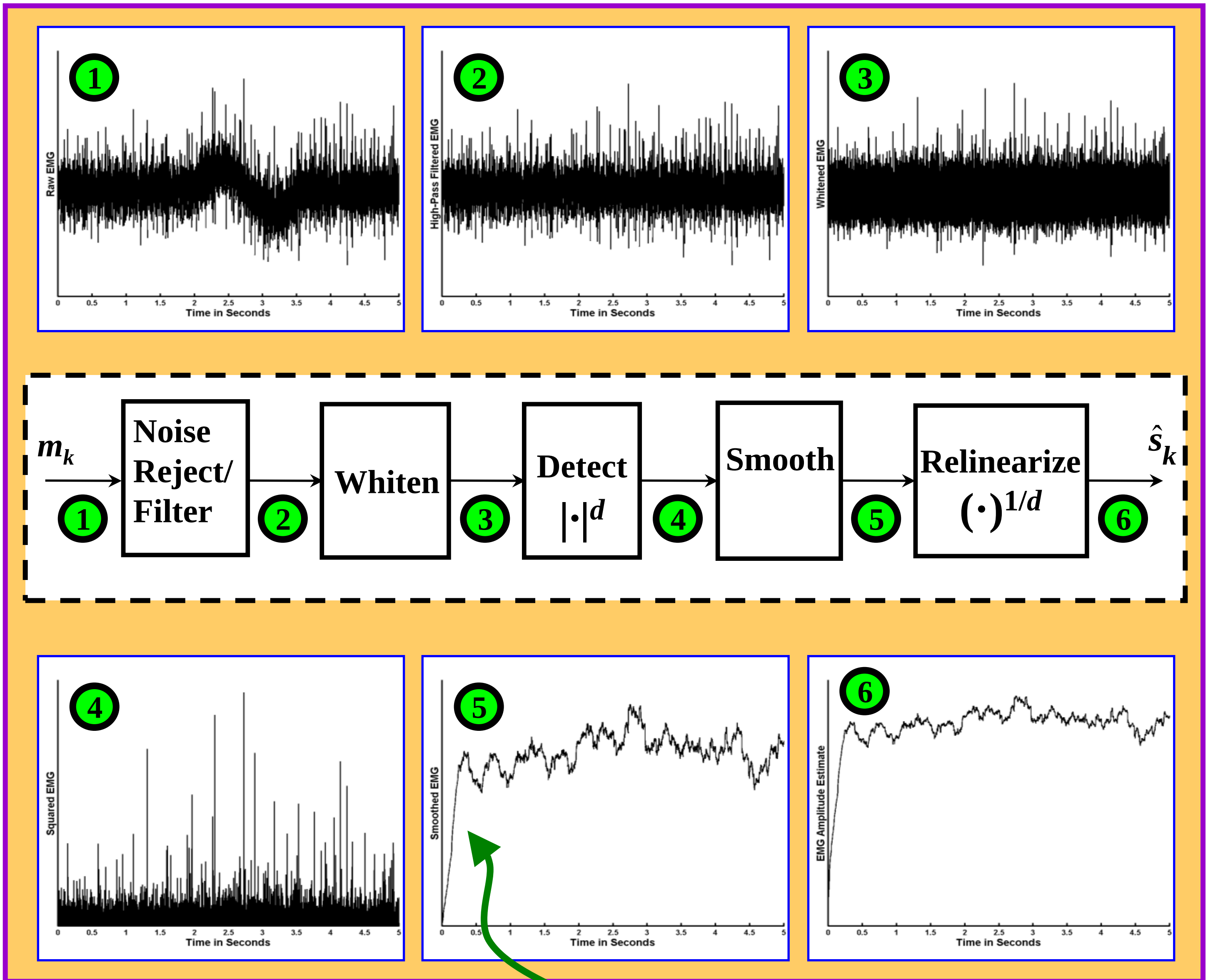
Figure 1.1: Surface EMG Waveform and Corresponding Joint Torque

A) Surface EMG waveform recorded from the triceps muscle with a bipolar electrode during an isometric non-fatiguing contraction. EMG is normalized to its maximum value in this trial.
B) Torque generated about the elbow during elbow extension for the same trial shown in (A) above. Torque is normalized to its maximum value in this trial.

Effort level ←→ Time-varying standard deviation

Optimal Single-Channel EMG Amplitude Estimation

• More mature/detailed algorithms exist for EMG amplitude estimation



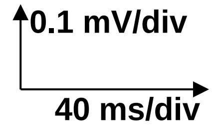
Discard “start-up” transient

Clancy, Morin, Merletti, 2004.

Can also combine information from multiple electrodes placed on same muscle, …

General Modeling Information

Normal Rate, Sinus Tachycardia, Sinus Bradycardia



- **Normal rate: 60–100 bpm**

Rate ≈ 75 bpm

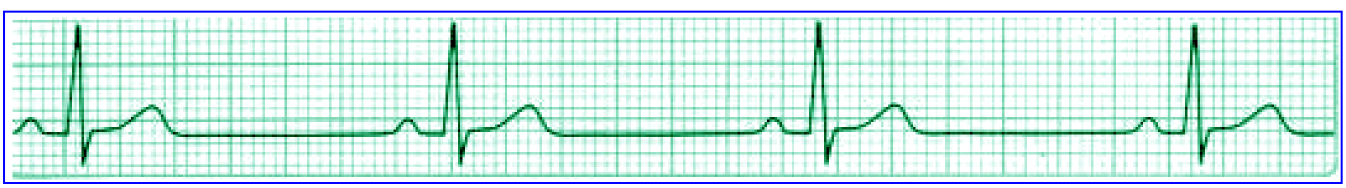
http://www.technion.ac.il/~eilamp/arrythmiasintro.html

- **Sinus Tachycardia: >100 bpm**

Rate ≈ 136 bpm

http://www.technion.ac.il/~eilamp/sinustachycardia.html

- **Sinus Bradycardia: <60 bpm**



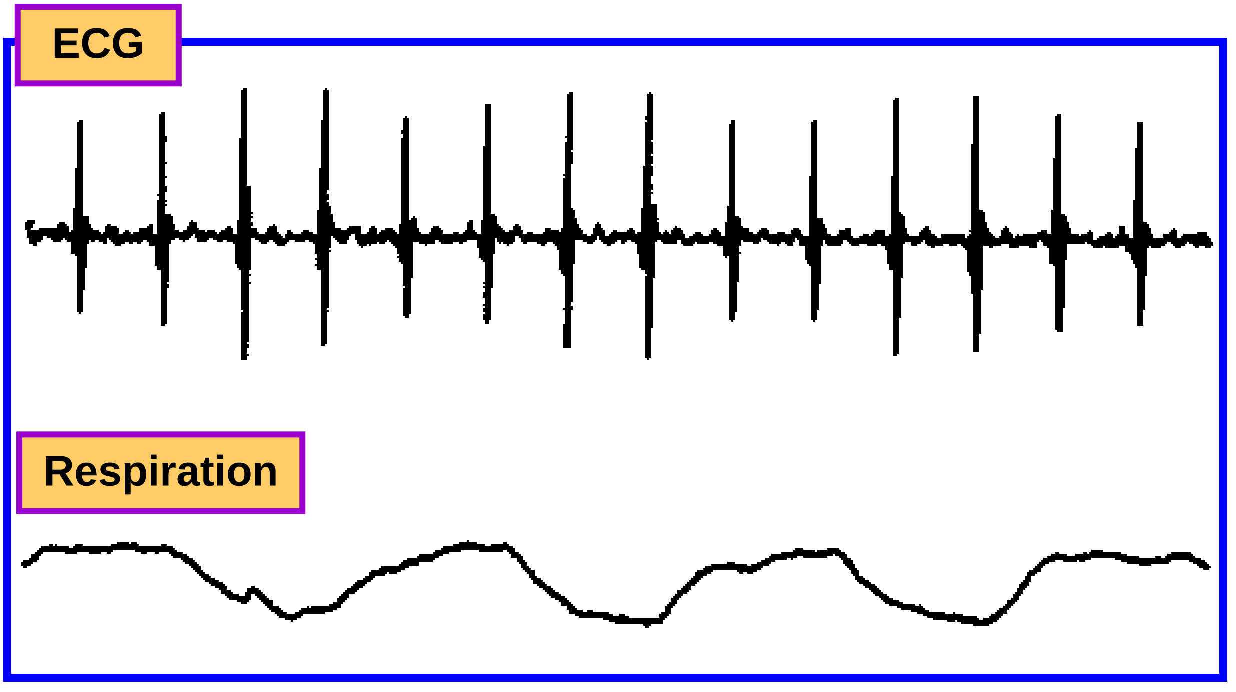
Rate ≈ 45 bpm

http://www.technion.ac.il/~eilamp/sinusbrady.html

Many, many other defined rhythms.

Respiratory Modulation

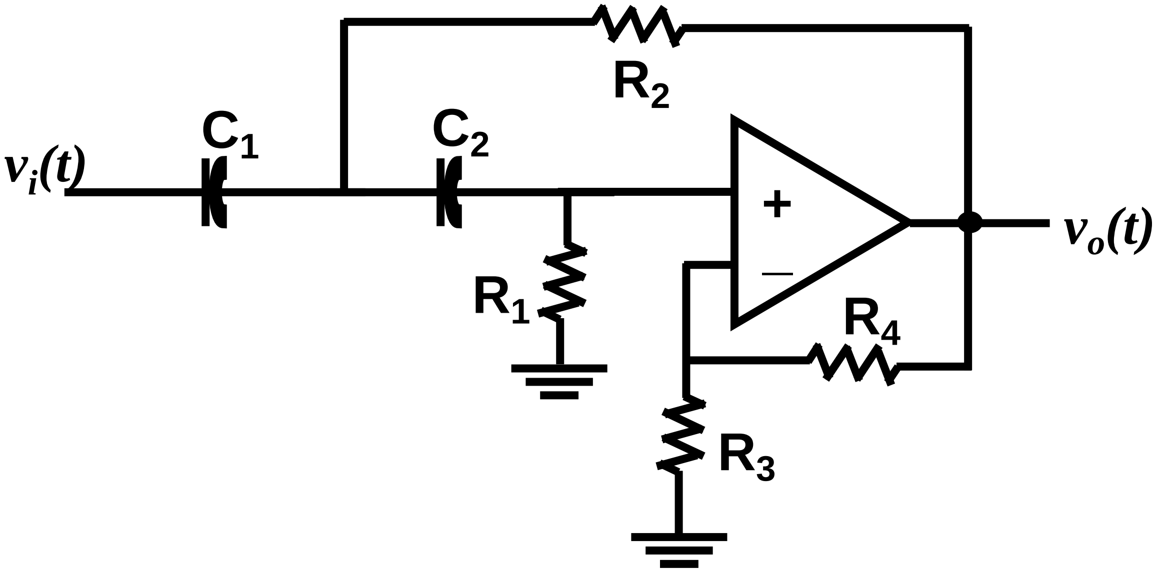
- Respiration modulates ECG rate, amplitude
- Rate increases during inspiration, decreases during expiration



Moody et al., *Computers in Cardiology* 12:113–116, 1985.

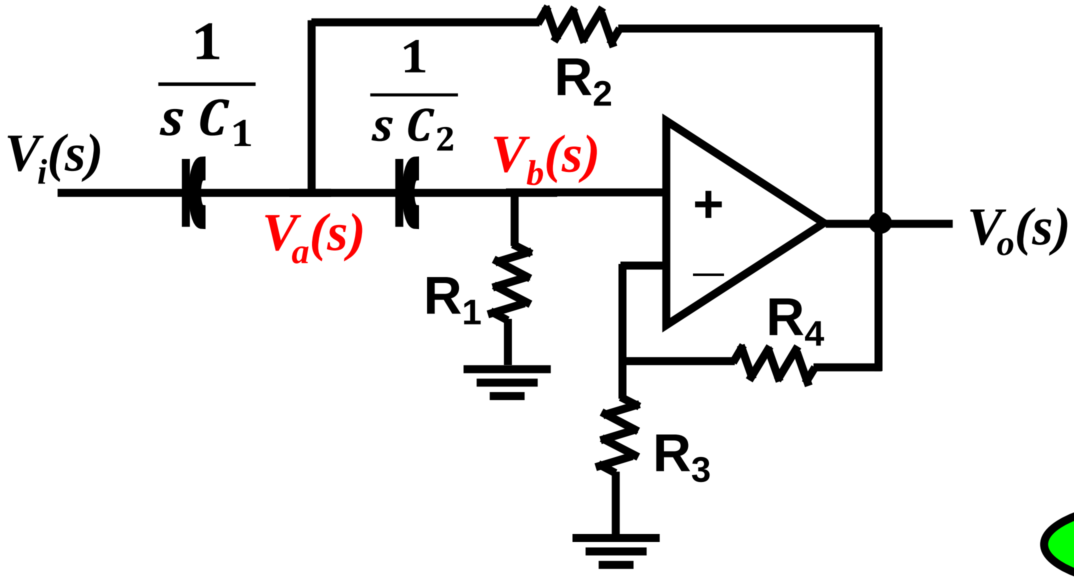
Frequency Response of Sallen-Key Highpass Filter (1)

- **Find the frequency response of the Sallen-Key highpass filter, shown below:**



- **In doing so: (1) Define** $A_\infty = 1 + \frac{R_4}{R_3} = \frac{R_3+R_4}{R_3}$

  (2) Use "*s*" as the frequency variable



Continued

Frequency Response of Sallen-Key Highpass Filter (2)

(1) By voltage division: $V_b(s) = V_o(s) \cdot \frac{R_3}{R_3+R_4} = \frac{V_o(s)}{A_\infty}$

(2) KCL at node V_a, then substituting (1) for $V_b(s)$:

$$[V_a(s) - V_i(s)] \cdot s\, C_1 + \frac{V_a(s) - V_o(s)}{R_2} + [V_a(s) - V_b(s)] \cdot s\, C_2 = 0$$

$$V_a(s) \cdot sR_2C_1 - V_i(s) \cdot sR_2C_1 + V_a(s) - V_o(s) + V_a(s) \cdot sR_2C_2$$
$$-V_o(s) \cdot \frac{sR_2C_2}{A_\infty} = 0$$

$$V_a(s) \cdot A_\infty\, [sR_2\,(C_1 + C_2) + 1]$$
$$= V_i(s) \cdot A_\infty\, [sR_2C_1] + V_o(s) \cdot A_\infty \left[1 + \frac{sR_2C_2}{A_\infty}\right]$$

(3) KCL at node V_b, then substituting (1) for $V_b(s)$:

$$[V_b(s) - V_a(s)] \cdot s\, C_2 + \frac{V_b(s)}{R_1} = 0$$

$$V_o(s) \cdot [sR_1C_2] - V_a(s) \cdot A_\infty\, [sR_1C_2] + V_o(s) = 0$$

$$V_a(s) = \frac{1}{A_\infty} \cdot V_o(s) \left[\frac{1 + sR_1C_2}{sR_1C_2}\right]$$

Continued

Frequency Response of Sallen-Key Highpass Filter (3)

• Substituting (3) into (2) :

$$V_o(s) \cdot \frac{[1 + sR_1C_2][sR_2\,(C_1 + C_2) + 1]}{sR_1C_2} = V_i(s) \cdot A_\infty\,[sR_2C_1] + V_o(s)\,[A_\infty + sR_2C_2]$$

Thus,

$$V_o(s) \cdot \left\{ \frac{[1 + sR_1C_2][sR_2C_1 + sR_2C_2 + 1] - [sR_1C_2][A_\infty + sR_2C_2]}{sR_1C_2} \right\} = V_i(s) \cdot A_\infty \cdot sR_2C_1$$

Giving,

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{A_\infty \cdot s^2 R_1R_2C_1C_2}{s^2[R_1R_2C_1C_2] + s[R_2(C_1 + C_2) + R_1C_2(1 - A_\infty)] + 1}$$

or

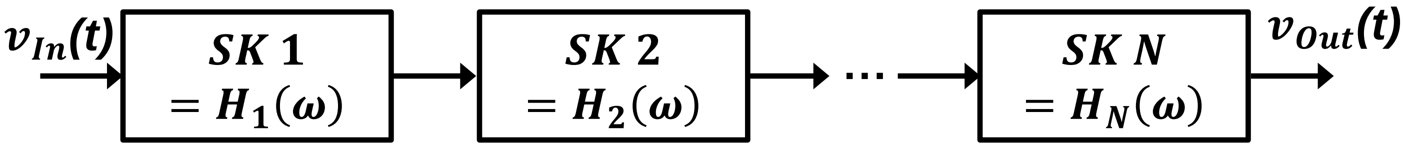
$$H(\omega) = \frac{A_\infty \cdot (j\omega)^2\, R_1R_2C_1C_2}{(j\omega)^2[R_1R_2C_1C_2] + (j\omega)[R_2(C_1 + C_2) + R_1C_2(1 - A_\infty)] + 1}$$

At $\omega = 0,\ H(\omega) = 0.$ At $\omega = \infty,\ H(\omega) = A_\infty.$

Note: Second-order filter

Higher-Order Filters (1)

- Common approach: Cascade filters

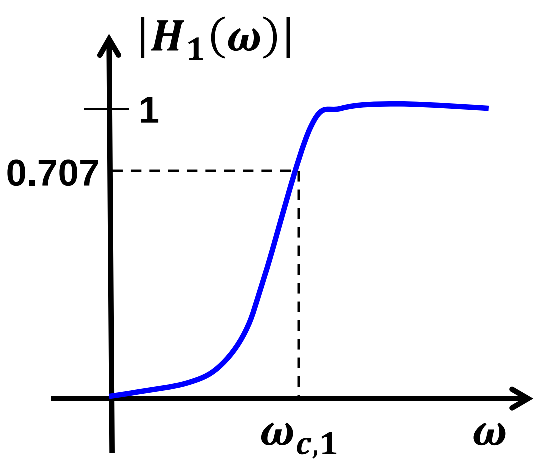


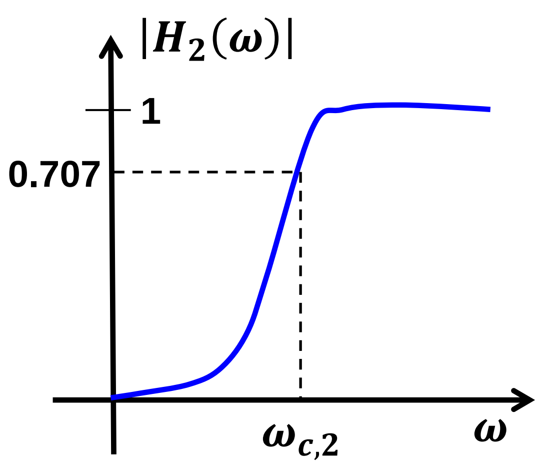
  – Overall: $H_T(\omega) = H_1(\omega) \cdot H_2(\omega) \cdots H_N(\omega)$

- Issue: Consider cascade of two filters:

$$|H_T(\omega)| = |H_1(\omega)| \cdot |H_2(\omega)|$$

  – For simplicity, let $H_1(\omega) = H_2(\omega) \quad \Longrightarrow \quad \omega_{c,1} = \omega_{c,2}$





Question: What is ω_c for $H_T(\omega)$?

1. $\omega_{c,T} = (\omega_{c,1} = \omega_{c,2})$
2. $\omega_{c,T} > (\omega_{c,1} = \omega_{c,2})$
3. $\omega_{c,T} < (\omega_{c,1} = \omega_{c,2})$

?

Continued

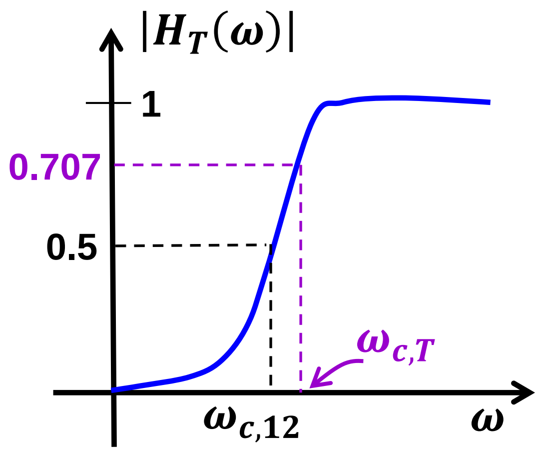
Higher-Order Filters (2)

- Denote: $\omega_{c,12} \equiv \omega_{c,1} = \omega_{c,2}$

  – Know: $\left|H_{1\text{ or }2}\left(\omega_{c,12}\right)\right| = \frac{\sqrt{2}}{2} = 0.707$

  – Thus,

$$\left|H_T\left(\omega_{c,12}\right)\right| = \left|H_1\left(\omega_{c,1}\right)\right| \cdot \left|H_2\left(\omega_{c,2}\right)\right| = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2}}{2} = \frac{1}{2}$$



- Cascaded $\omega_{c,T}$ is <u>greater</u> than $\omega_{c,12}$ (for highpass cascade)

- Cascade design must account for $\omega_{c,T}$ shift
  – Complicated analysis

  or

  – ***Normalized design tables !!!***

Highpass Design: Selecting *R*'s, *C*'s (1)

- Re-write $H(\omega)$ [Divide numerator, denom. by $(j\omega)^2 R_1 R_2 C_1 C_2$]:

$$H(\omega) = \frac{A_\infty}{1 + \frac{1}{j\omega} \cdot \left[\frac{R_2(C_1 + C_2) + R_1 C_2 (1 - A_\infty)}{R_1 R_2 C_1 C_2}\right] + \frac{1}{(j\omega)^2} \cdot \left[\frac{1}{R_1 R_2 C_1 C_2}\right]}$$

- To normalize cut-off freq., replace: $\omega = \frac{\omega}{\omega_c} \cdot \omega_c \equiv \omega_n \cdot \omega_c \equiv \omega \cdot \omega_c$:

$$H_{Norm}(\omega) = \frac{A_\infty}{1 + \frac{1}{j\omega} \cdot \left[\frac{R_2(C_1 + C_2) + R_1 C_2 (1 - A_\infty)}{\omega_c \cdot R_1 R_2 C_1 C_2}\right] + \frac{1}{(j\omega)^2} \cdot \left[\frac{1}{\omega_c^2 \cdot R_1 R_2 C_1 C_2}\right]}$$

- So,

$$H_{Norm}(\omega) = \frac{A_\infty}{1 + \frac{a}{j\omega} + \frac{b}{(j\omega)^2}},$$

$\boxed{\omega = 1 \Rightarrow \text{cut-off}}$

where

$$a = \frac{R_2(C_1 + C_2) + R_1 C_2 (1 - A_\infty)}{\omega_c \cdot R_1 R_2 C_1 C_2},$$

$$b = \frac{1}{\omega_c^2 \cdot R_1 R_2 C_1 C_2}$$

and

$\omega_c = 2\,\pi\, f_c \;\Longrightarrow$ Desired cut-off frequency

Continued

Highpass Design: Selecting *R*'s, *C*'s (2)

- Tables of normalized a_i, b_i values exist ("*i*" refers to stage)
  - Various filter orders
  - Various filter shapes (e.g., Butterworth, Chebyshev)

  - E.g.: Fourth-order Butterworth highpass filter:
    $a_1 = 1.8478$, $b_1 = 1$ (Stage 1)
    $a_2 = 0.7654$, $b_2 = 1$ (Stage 2)

- Any R_1, R_2, C_1, C_2 satisfying a_i, b_i is OK
  - Underdetermined problem. Solution not unique.

  - Common solution (each stage) is:
    1. Select $C_1 = C_2$ = Some Easy Value
       → Pick *C*'s first, because fewer manufactured values
    2. R_1:
       - If $A_i = 1$, then $R_1 = \frac{C_1 + C_2}{a_i\, \omega_c\, C_1\, C_2}$
       - If $A_i > 1$, then

$$R_1 = \frac{a_i\, \omega_c\, C_1\, C_2 - \sqrt{a_i^2\, \omega_c^2\, C_1^2\, C_2^2 - 4\left[\omega_c^2 b_i\, C_1\, C_2^2\, (1 - A_i)\right](C_1 + C_2)}}{2\, \omega_c^2 b_i\, C_1\, C_2^2\, (1 - A_i)}$$

    3. $R_2 = \frac{1}{\omega_c^2 \cdot b_i \cdot R_1 C_1 C_2}$
    4. Replace ideal R_1, R_2 with manufactured values
    5. Repeat for each stage: $A_T = A_1 \cdot A_2 \cdot A_3 \cdots A_N$

Example Highpass Design (1)

- **Desire** 4th-order, Butterworth, f_c = 15 Hz, A_T = 10

- From Kugelstadt (Chapter 16 of "Op Amps for Everyone," literature # SLOD006A, www.ti.com)

  $a_1 = 1.8478,\ b_1 = 1$ (Stage 1)

  $a_2 = 0.7654,\ b_2 = 1$ (Stage 2)

  – **Stage 1:** Choose $C_1 = C_2 = 68\ n$F

  $\rightarrow R_1 = 59.3$ kΩ, $R_2 = 410.8$ kΩ

  Choose R_4 = 90 kΩ, R_3 = 10 kΩ $\Rightarrow A_1 = 10$

  – **Stage 2:** Choose $C_1 = C_2 = 68\ n$F

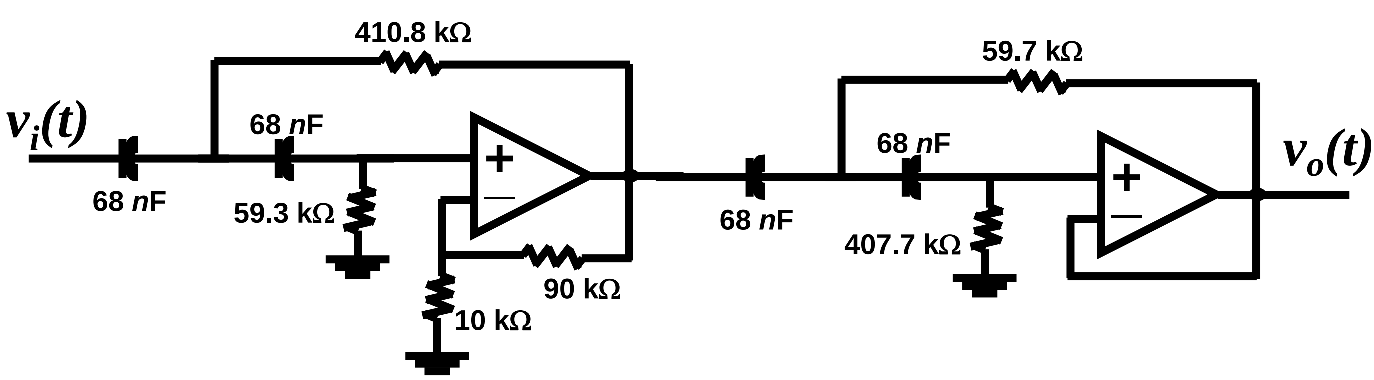
  $\rightarrow R_1 = 407.7$ kΩ, $R_2 = 59.7$ kΩ

  Short R_4, omit R_3 $\Rightarrow A_2 = 1$

*****

√ Reasonable range of R, C values.

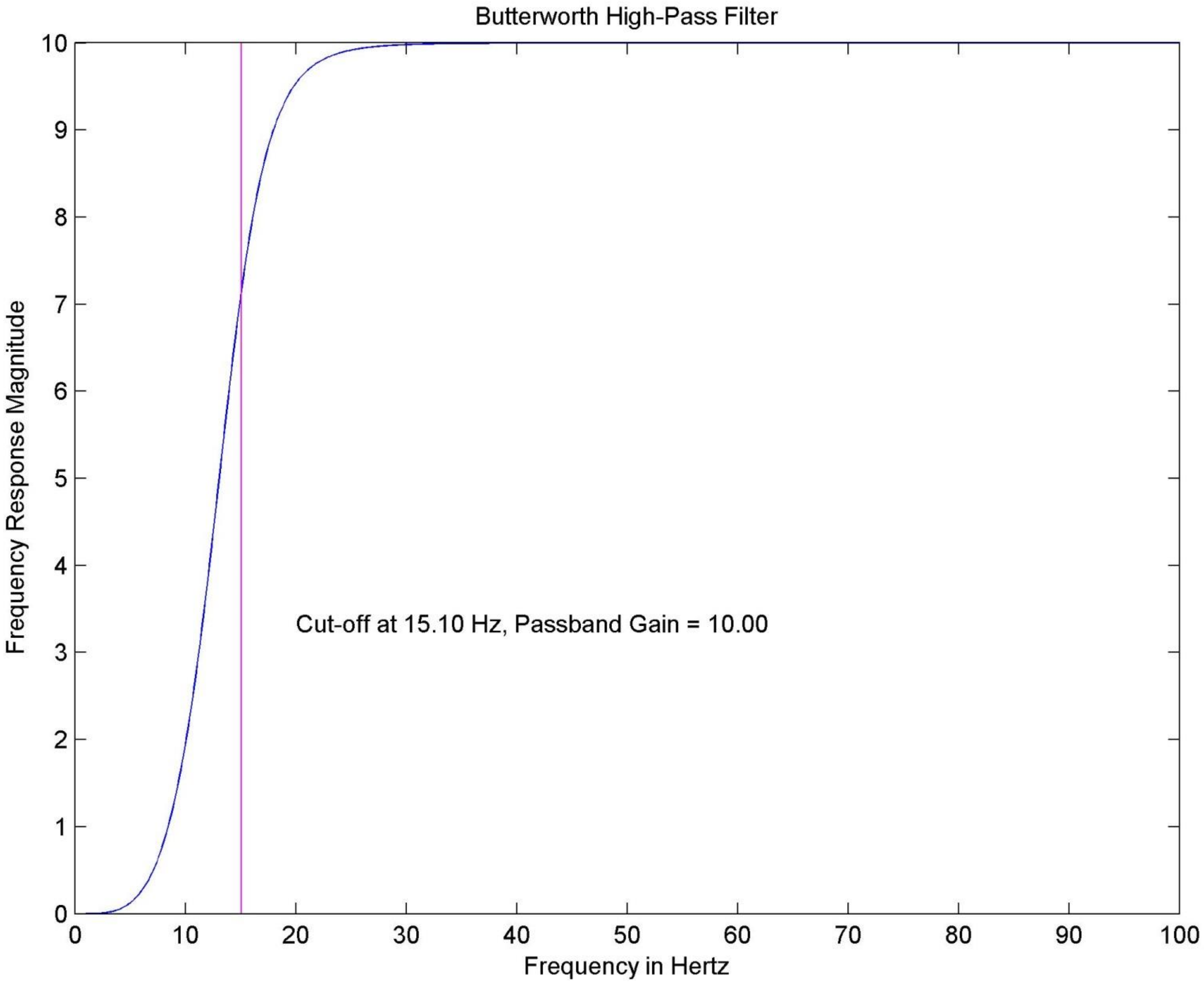
$\Rightarrow$ Circuit:



Continued

Example Highpass Design (2)

Magnitude response of 4th-order, Butterworth, f_c = 15 Hz, A_T = 10



MATLAB Design Software (1)

9/18/16 8:52 PM C:\Users\Ted\Desktop...\butter hi design.m 1 of 3

```
function H = butter_hi_design(f, Fc, A, Stage1, Stage2, Stage3, Stage4, Stage5)
%
% H = b_hi_des(f, Fc, A, Stage1[, Stage2[, Stage3[, Stage4[, Stage5]]]])
%
%   Helps design an electronic circuit to build even-order, high-pass
% Butterworth filters of orders 2, 4, 6, 8, 10.  Plots the magnitude of
% the resulting frequency response.
%   See [Thomas Kugelstadt, "Chapter 16: Active
% Filter Design Techniques," Literature Number SLOA088, Texas Instruments
% Incorporated, Post Office Box 655303, Dallas, Texas  75265, 2001.
% Excerpted from "Op Amps for Everyone," Literature Number SLOD006A,
% Texas Instruments.  Available on the Internet at: http://www.ti.com].
%
% f:      Frequency axis (Hertz) for all calculations and plotting (vector).
% A:      Overall circuit gain (>=1).  Applied in 1st stage.
% Fc:     Desired cutoff frequency in Hertz (scalor).
% StageX: Up to 5 StageX arguments (one per stage) are permitted.  For each stage,
%         StageX is a vector of 1-4 elements, corresponding to C1, C2,
%         R1 and R2, respectively.  C1 must be supplied (in Farads) and
%         is the first vector element.  If a second argument is supplied,
%         then it is C2 (in Farads).  If C2 is not supplied, it is set
%         equal to C1.  If
%         a third argument is supplied, it is R1 (Ohms).  If not supplied,
%         it is set as required to form a Butterworth filter.  If a fourth
%         argument is supplied, it is R2 (Ohms).  If not supplied, it is
%         set as required to form a Butterworth filter.
%
% H: Resulting (complex-valued) frequency response corresponding to the
%     frequency axis f.
%
% USAGE RECOMENDATIONS:  To build a filter, initially call script with only
%   C1 and C2 specified for each stage.  Use the recommended
%   R1 and R2 values to find R1's and R2's that are manufactured.  Call the
%   script a second time with all values to see the resultant nominal frequency
%   response.

% Table of ai values.  For even-order Butterworth, all bi values equal 1.
aiTable = [1.4142    NaN    NaN    NaN    NaN;  % For 1-stage filter.
           1.8478 0.7654    NaN    NaN    NaN;  % For 2-stage filter.
           1.9319 1.4142 0.5176    NaN    NaN;  % For 3-stage filter.
           1.9616 1.6629 1.1111 0.3902    NaN;  % For 4-stage filter.
           1.9754 1.7820 1.4142 0.9080 0.3129]; % For 5-stage filter.

% Determine the number of stages.
Nstage = nargin - 3;
if Nstage<1 | Nstage>5, error('Must have 1-5 stages.'); end
if A<1, error('"A" must be >= 1.'); end

% Extract/develop parameters for each stage and build the frequency response.

H = ones( 1, length(f) );  % Initialize frequency response to unity.
```

Continued

MATLAB Design Software (2)


```
w = 2*pi*f;                 % Convert to radian frequency.
MagBig = A;                 % Cascade passband gain.

for S = 1:Nstage

  if S>1, A = 1; end  % Set gain to one after first stage.

  % Prepare to parse parameter vector.
  eval(['Param = Stage' int2str(S) ';']);  % Copy parameters to a scratch vector.
  if length(Param)<1, error(['Stage ' int2str(S) ' parameter list < 1.']); end;
  if length(Param)>4, error(['Stage ' int2str(S) ' parameter list > 4.']); end;
  ai = aiTable(Nstage,S);

  % Parse parameter vector.
  % C1.
  C1 = Param(1);
  % C2.
  if length(Param)>1,  C2 = Param(2);  else,  C2 = C1; end
  % R1.
  if length(Param)>2,  R1 = Param(3);
  elseif A==1
    R1 = ( C1+C2 ) / ( 2*pi*Fc*ai*C1*C2 );
  else
    a = ((2*pi*Fc)^2) * C1*C2*C2*(1-A);
    b = -ai * 2*pi*Fc * C1*C2;
    c = C1 + C2;
    R1 = ( -b - sqrt(b*b - 4*a*c) ) / (2*a);
  end
  % R2.
  if length(Param)>3,  R2 = Param(4);
  else,  R2 = 1 / ( ((2*pi*Fc)^2)*R1*C1*C2 ); end

  % Print component values.
  fprintf('Stage %d: C1=%e F, C2=%e F, R1=%e Ohms, R2=%e Ohms\n', S, C1, C2, R1, R2);

  % Update frequency response.
  a = ( R2*(C1+C2) + R1*C2*(1-A) ) ./ (R1*R2*C1*C2);
  b = 1 ./ (R1*R2*C1*C2);
  Hstage = A ./ (  1 + ( a./(j*w) ) + ( b./(-w.*w) )  );
  H = H .* Hstage;
end

% Now, plot the frequency response magnitude.
plot( f, abs(H) )
xlabel('Frequency in Hertz')
ylabel('Frequency Response Magnitude')

L1 = find( abs(H) >= (MagBig*sqrt(2)/2) );
hold on, plot([f(L1(1)) f(L1(1))], [0 MagBig], 'm'), hold off
Thing = sprintf('Cut-off at %0.2f Hz, Passband Gain = %0.2f', f(L1(1)), MagBig);
text( max(f)/5, max(abs(H))/3, Thing);
```

Continued

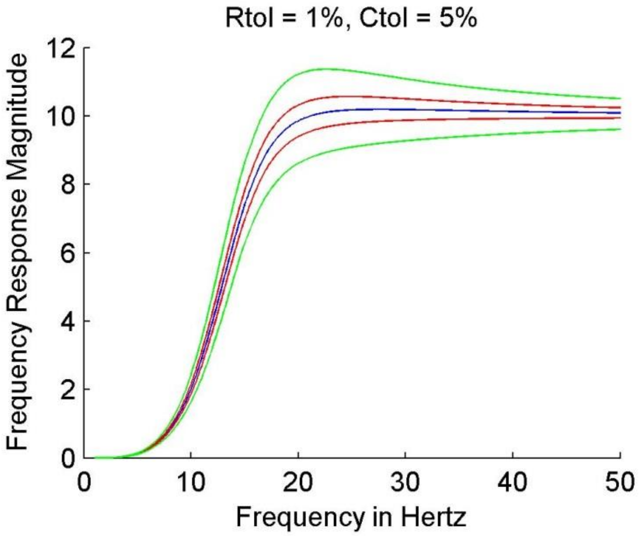
MATLAB Design Software (3)

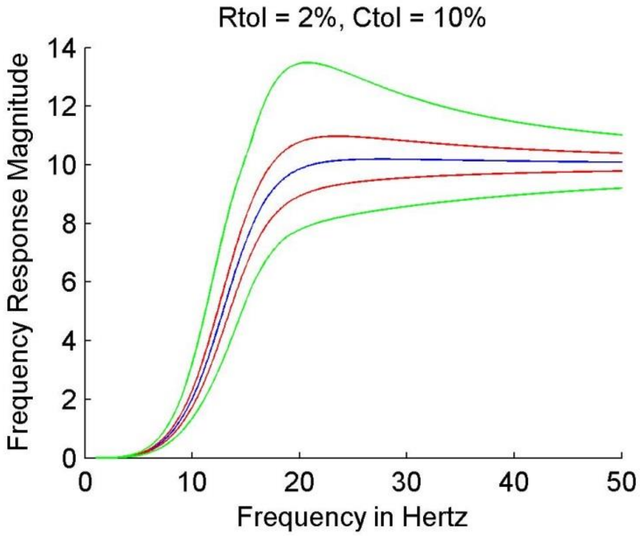
9/18/16 8:52 PM C:\Users\Ted\Desktop...\butter_hi_design.m 3 of 3

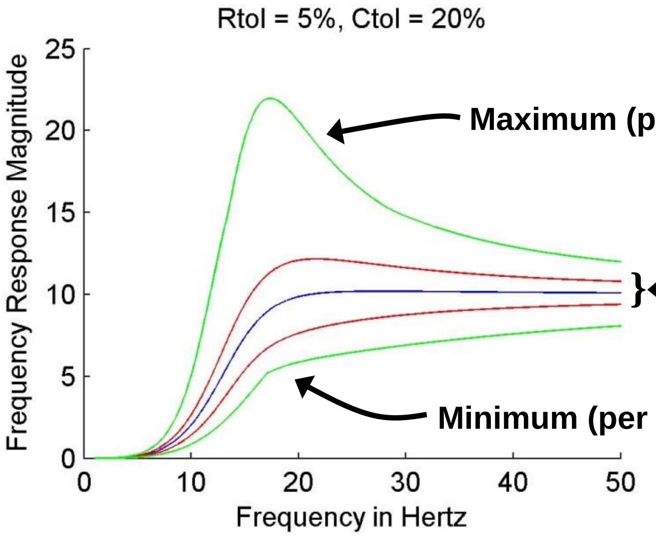
```
title('Butterworth High-Pass Filter');

figure(gcf);

return
```

Influence of Component Tolerances

- **Fourth-order, Butterworth, f_c = 15 Hz, A_T = 10**
  - **All gain in first stage**
  - **Random *R*'s, *C*'s using tolerance**
  - ***N* = 1000 iterations per condition**







Maximum (per frequency)

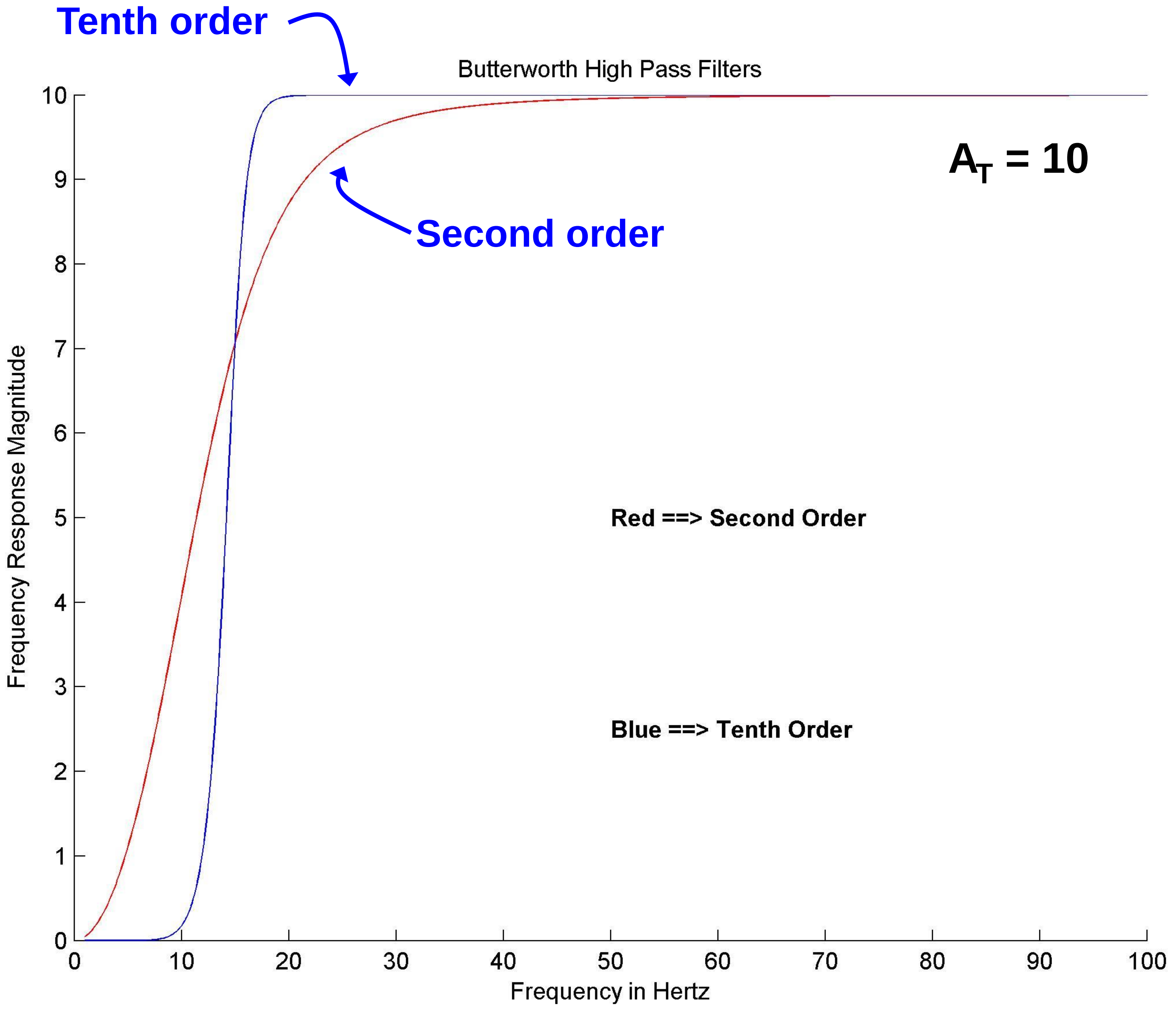
Ideal ± 1 std. dev.

Minimum (per frequency)

i.e., composite result
⇒ Not any one $|H_i(\omega)|$

Comparison of Filter Orders

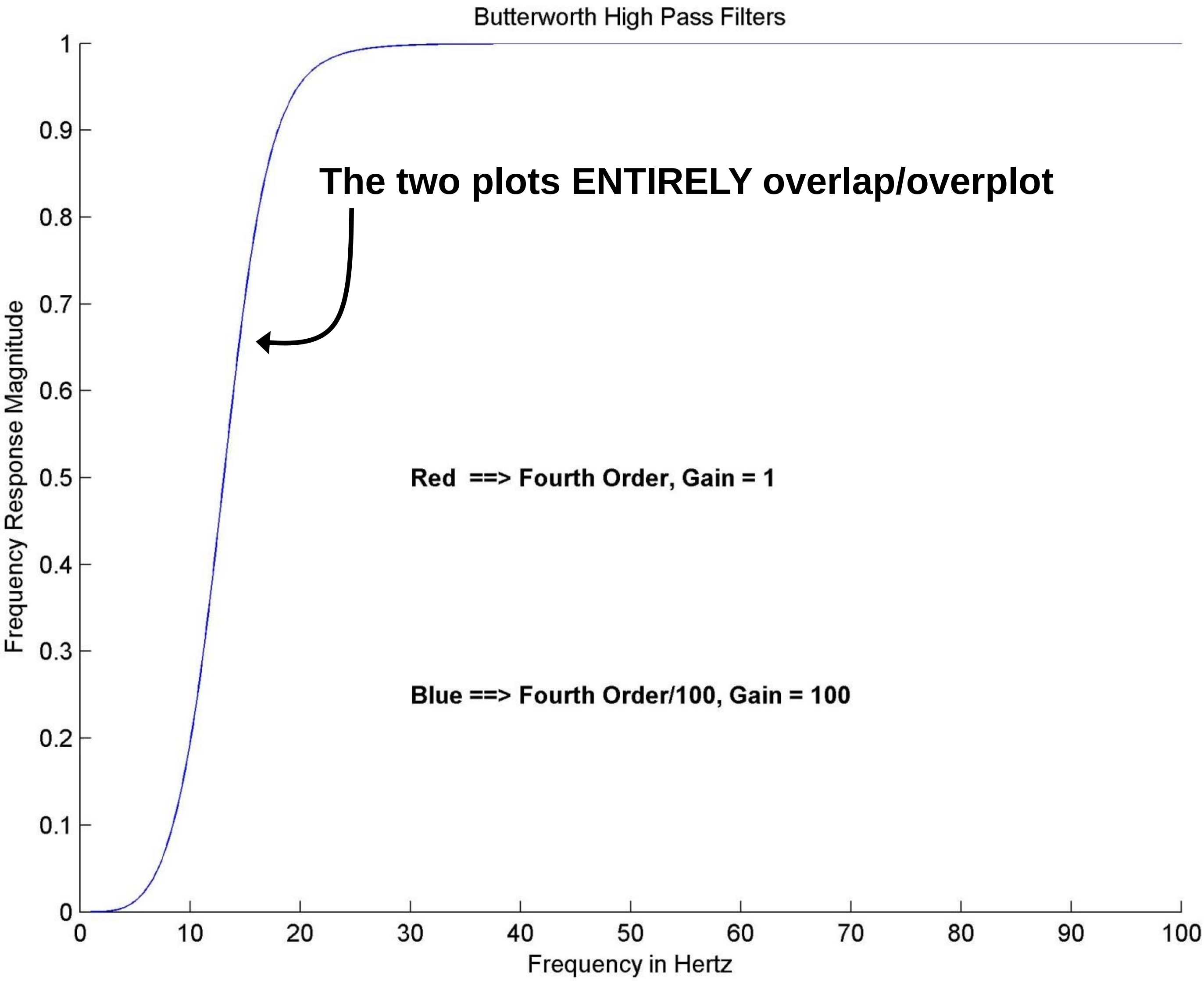
Magnitude Response of Butterworth Filter



Comparison of Shape vs. Gain

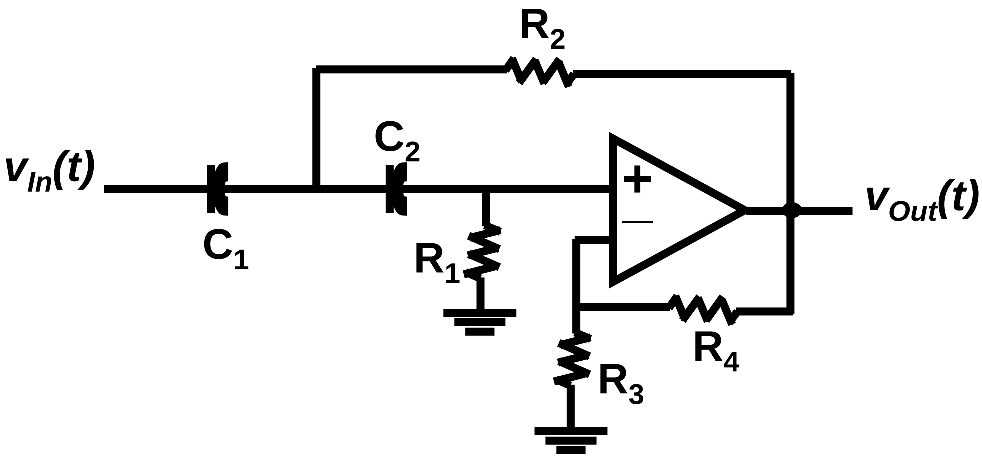
- **Red (covered): Fourth-order, Butterworth, $A_T = 1$, f_c = 15 Hz**

- **Blue: Fourth-order, Butterworth, fc = 15 Hz**
  - **$A_T = 100$, then (to compare shape) divide $\frac{|H_{Blue}(\omega)|}{100}$**

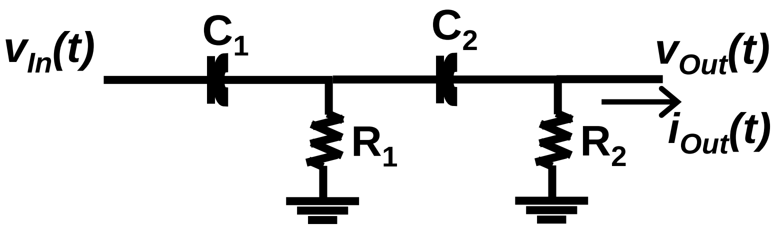


Active vs. Passive Filters

Active:



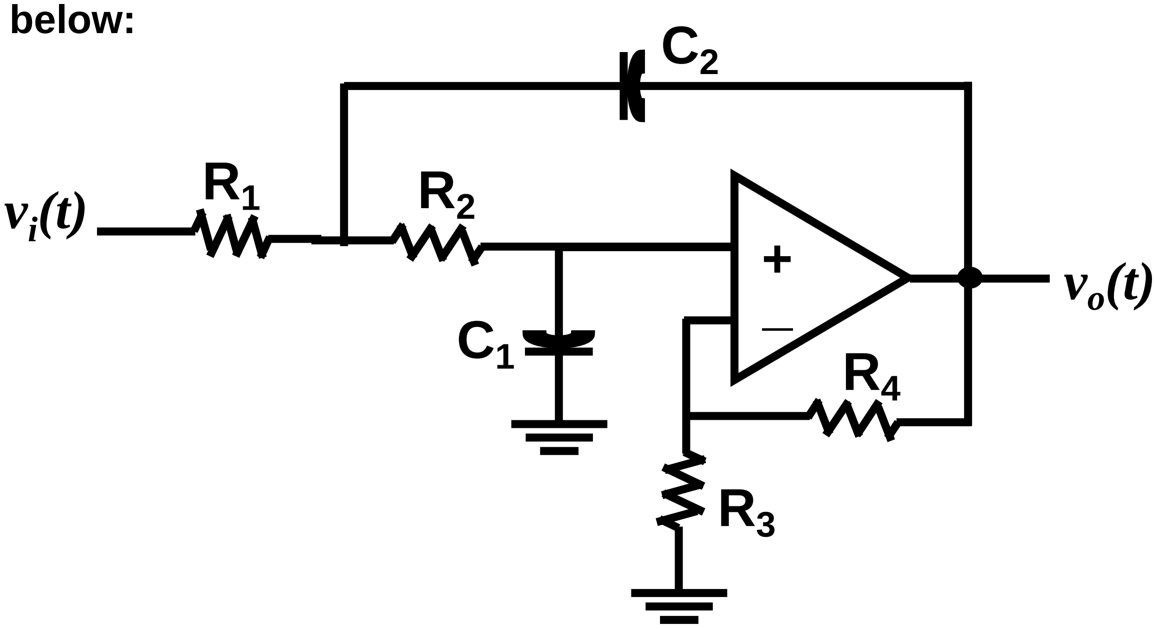
Passive:



	PROs	CONs
Active	• Low output impedance $\Rightarrow$ easy to cascade • Can apply gain	• More components • Feedback can be unstable • Need a power supply
Passive	• Fewer components • Always stable (No feedback)	• No gain • High output impedance $\Rightarrow i_{Out}$ influences $H(\omega)$ • Complicated to cascade

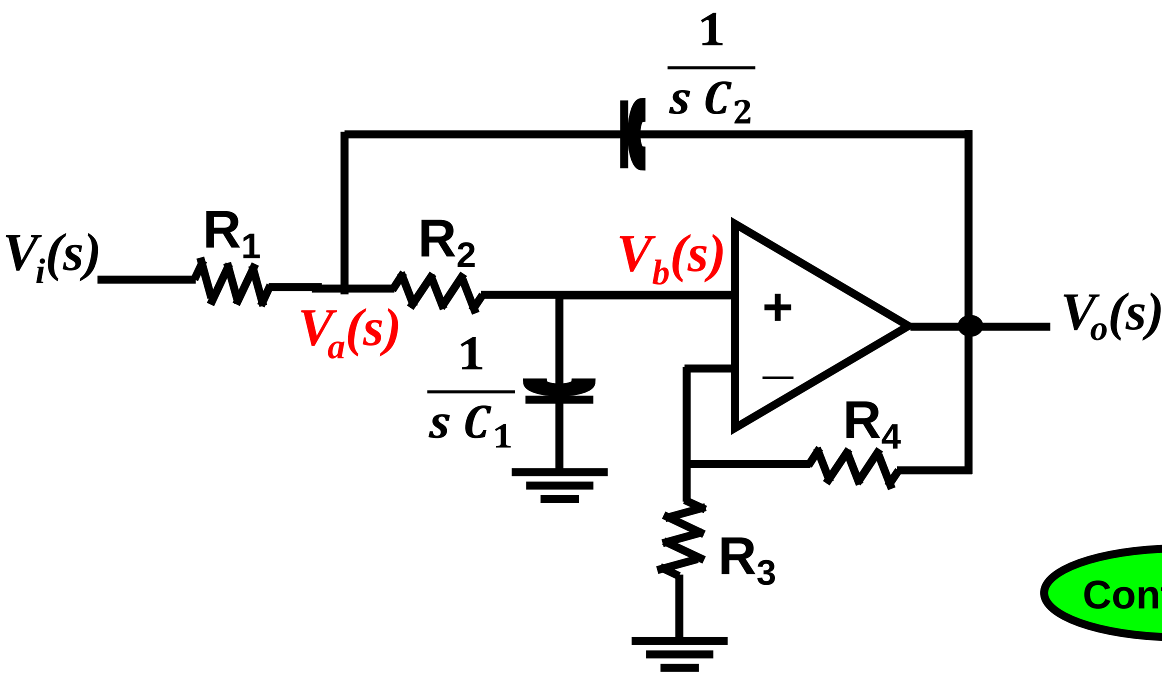
Frequency Response of Sallen-Key Lowpass Filter (1)

- **Find the frequency response of the Sallen-Key lowpass filter, shown below:**



- **In doing so: (1) Define** $A_0 = 1 + \frac{R_4}{R_3} = \frac{R_3+R_4}{R_3}$

(2) Use "*s*" as the frequency variable



Continued

Frequency Response of Sallen-Key Lowpass Filter (2)

(1) **By voltage division:** $V_b(s) = V_o(s) \cdot \frac{R_3}{R_3+R_4} = \frac{V_o(s)}{A_0}$

(2) **KCL at node V_a, then multiply by R_1R_2 and substitute (1) for $V_b(s)$:**

$$\frac{V_a(s) - V_i(s)}{R_1} + [V_a(s) - V_o(s)] \cdot s\, C_2 + \frac{V_a(s) - V_b(s)}{R_2} = 0$$

$$V_a(s) \cdot R_2 - V_i(s) \cdot R_2 + V_a(s) \cdot s\, R_1\, R_2\, C_2 - V_o(s) \cdot s\, R_1\, R_2\, C_2 + V_a(s) \cdot R_1 - V_o(s) \cdot \frac{R_1}{A_0} = 0$$

$$V_a(s) \cdot A_0\, [R_1 + R_2 + s\, R_1\, R_2\, C_2] = V_i(s)\, [A_0 R_2] + V_o(s) \left[s\, R_1\, R_2\, C_2 + \frac{R_1}{A_0}\right] \cdot A_0$$

(3) **KCL at node V_b, then substituting (1) for $V_b(s)$:**

$$\frac{V_b(s) - V_a(s)}{R_2} + V_b(s)\, [s\, C_1] = 0$$

$$V_o(s) - V_a(s)\, [A_0] + V_o(s) [s R_2 C_1] = 0$$

$$V_a(s) = \frac{1}{A_0} \cdot V_o(s)\, [1 + s R_2 C_1]$$

Continued

Frequency Response of Sallen-Key Lowpass Filter (3)

• **Substituting (3) into (2) :**

$$V_o(s)[1 + sR_2C_1][R_1 + R_2 + s\,R_1\,R_2\,C_2] = V_i(s)[A_0R_2] + V_o(s)\,[s\,A_0\,R_1\,R_2\,C_2 + R_1]$$

$\Rightarrow$

$$V_o(s)\left[s^2\,R_1R_2^2\,C_1\,C_2 + s\left(R_1\,R_2\,C_1 + R_2^2\,C_1 + R_1\,R_2\,C_2\right) + (R_1 + R_2)\right] - V_o(s)\,[s\,A_0\,R_1\,R_2\,C_2 + R_1] = V_i(s)[A_0R_2]$$

$\Rightarrow$

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{A_0R_2}{s^2\left(R_1R_2^2\,C_1\,C_2\right) + s\left(R_1\,R_2\,C_1 + R_2^2\,C_1 + R_1\,R_2\,C_2 - A_0\,R_1\,R_2\,C_2\right) + R_2}$$

$$H(s) = \frac{A_0}{s^2(R_1R_2C_1C_2) + s[(R_1 + R_2)\,C_1 + (1 - A_0)\,R_1\,C_2] + 1}$$

or

$$H(\omega) = \frac{A_0}{(j\omega)^2[R_1R_2C_1C_2] + (j\omega)[(R_1 + R_2)\,C_1 + (1 - A_0)\,R_1\,C_2] + 1}$$

At $\omega = 0,\; H(\omega) = A_0.$

At $\omega = \infty,\; H(\omega) = 0.$

Lowpass Design: Selecting *R*'s, *C*'s

$$H_{Low,Norm}(\omega) = \frac{A_0}{1 + a(j\omega) + b(j\omega)^2},$$

where, $a = \omega_c \cdot C_1(R_1 + R_2) + \omega_c \cdot (1 - A_0)R_1C_2$

$$b = \omega_c^2 \cdot R_1R_2C_1C_2$$

$$\omega_c \Rightarrow \text{Desired cut} - \text{off frequency}$$

- **Again: R_1, R_2, C_1, C_2 underdetermined (each stage)**

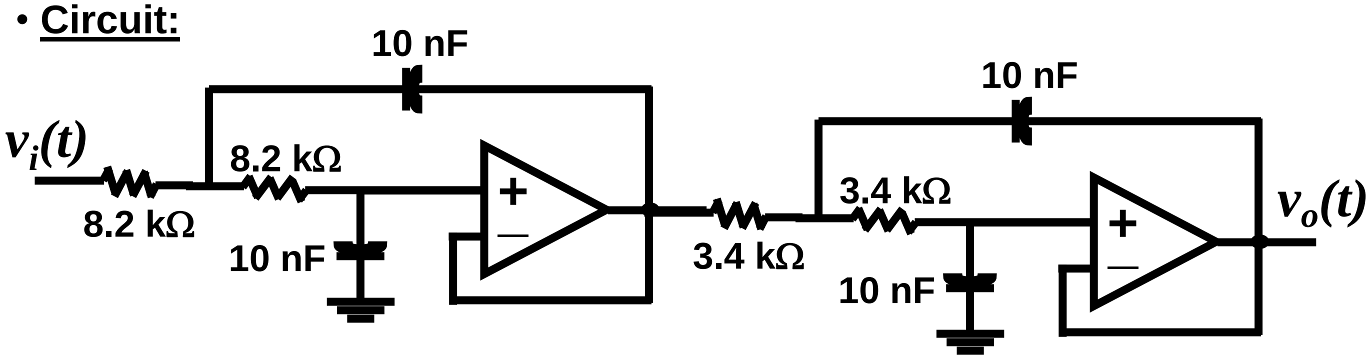
1. **Select $C_1 = C_2$ = Some Easy Value**
   → **Pick *C*'s first, because fewer manufactured values**
2. **R_1:**
   – **If $A_i \geq 1$, then**
   $$R_1 = \frac{a_i\,\omega_c^2\,C_2 - \sqrt{a_i^2\,\omega_c^4\,C_2^2 - 4\,\omega_c^4\,b_i\,C_2[C_1 + (1 - A_0)\,C_2]}}{2\,\omega_c^3 C_2[C_1 + (1 - A_0)\,C_2]}$$
3. $R_2 = \frac{b_i}{\omega_c^2 \cdot R_1C_1C_2}$
4. **Replace ideal R_1, R_2 with manufactured values**
5. **Repeat for each stage:** $A_T = A_1 \cdot A_2 \cdot A_3 \cdots A_N$

Example Lowpass Design

- **Design:** Fourth-order, Butterworth, f_c = 1800 Hz, A_T = 1
  - Choose all C's = 10 nF, $\underbrace{R_3\text{'s} = \infty\ \Omega,\ R_4\text{'s} = 0\ \Omega}_{\text{Unity Gain}}$
  - **Stage 1:** $R_1 = R_2$ = 8.2 kΩ
  - **Stage 2:** $R_1 = R_2$ = 3.4 kΩ

From Kugelstadt:
a_1 = 1.8478, b_1 = 1 (Stage 1)
a_2 = 0.7654, b_1 = 1 (Stage 2)

- **Circuit:**



- **Magnitude Response:**