Laboratório 3
Filtros Ativos Passa-Tudo
OP AMPS for Everyone
Newnes, 2009
To achieve equal temporal delays for all the frequencies, we need every frequency to have a different phase shift—namely, a phase shift that results in the same delay for every frequency. More specifically, we need a phase-shift response that increases linearly with frequency.
An ideal linear-phase filter, then, exhibits phase shift that increases linearly with frequency, and it thereby provides constant temporal delay (this applies primarily to the frequencies within the passband, i.e., the frequencies of interest). Group delay \( t_{gr} \) is proportional to the derivative of the phase response with respect to frequency. The derivative of a linear function is a constant, which explains why a linear phase response is also referred to as constant group delay.

\[
t_{gr} = - \frac{d\phi}{d\omega}
\]

Now consider a situation in which a filter will see signals composed of various different frequencies that work together. Problems could arise if these different frequencies experience different delays.
All-pass filter has a constant gain across the entire frequency range, and a phase response that changes linearly with frequency.

All-pass filters are used in circuits referred to as “phase equalizers” or “delay equalizers.” As discussed in Understanding Linear-Phase Filters, it is sometimes important to ensure that all the frequency components in a signal experience equal temporal delay.

**Audio applications**: Frequencies representing different pitches must remain synchronized to ensure proper sound reproduction.

**Pitch** is an auditory sensation in which a listener assigns musical tones to relative positions on a musical scale based primarily on their perception of the frequency of vibration. Pitch is closely related to frequency, but the two are not equivalent. Frequency is an objective, scientific attribute that can be measured. Pitch is each person's subjective perception of a sound wave, which cannot be directly measured. However, this does not necessarily mean that most people won't agree on which notes are higher and lower.
Digital communications: The sinusoidal harmonic frequencies that constitute a square wave must experience constant delay to avoid distortion of the digital signal.

Similar to the low-pass filters, all-pass circuits of higher order consist of cascaded first-order and second-order all-pass stages.

\[
A(s) = \frac{\prod_i (1 - a_is + b_is^2)}{\prod_i (1 + a_is + b_is^2)}
\]

\((a_i \text{ and } b_i \text{ being the coefficients of a partial filter})\)
4  **Group Delay** $(t_{gr})$

It is the time by which the all pass filter delays each frequency within a band.

\[
t_{gr} = -\frac{d\phi}{d\omega}
\]

5  **Normalized Group Delay** $(T_{gr})$

The frequency at which the normalized group delay drops to $\frac{1}{\sqrt{2}}$ times its initial value is the corner frequency $(f_c)$ which corresponds to $\Omega=1$.

\[
T_{gr} = -\frac{1}{2\pi} \frac{d\phi}{d\Omega}
\]

\[
T_{gr} = -\frac{f_c}{2\pi} \frac{d\phi}{df}
\]

\[
T_{gr} = \frac{t_{gr}}{T_c} = t_{gr} f_c = t_{gr} \frac{\omega_c}{2\pi}
\]
At $\Omega=1$ there is a -3dB decrease in $T_{gr}$!
The $T_{gro}$ is the value of $T_{gr}$ when $\Omega < 0.1$

$$T_{gro} = \frac{1}{\pi} \sum_{i} a_i$$

**Examples:**

$n=1$

$$T_{gro} = \frac{1}{\pi} (0,6436) = 0,2049$$

$n=3$

$$T_{gro} = \frac{1}{\pi} (1,1415 + 1,5092)$$

$$T_{gro} = 0,8437$$
First Order Topology

\[ T_{gro} = 2RCf_C \]
\[ R_1 = R \]

Second Order Topology

\[ T_{gro} = 4RCf_C \]
Designing All Pass Filters (First Order Topology)
First Order Topology

1. Specify $f_c$ and $C$
2. Calculate $R$
   \[ R = \frac{a_i}{2\pi f_c \cdot C} \]
3. Delay group
   \[ T_{gro} = 2RCf_c \]

\[ A(s) = \frac{1 - RC\omega_c \cdot s}{1 + RC\omega_c \cdot s} \]
Designing All Pass Filters (Second Order Topology)
1. Specify $f_c$ and $C$  
2. Calculate $R$  
3. Calculate $R_1$, $R_2$, $R_3$  
4. Maximum delay group

**Second Order Topology**

$$A(s) = \frac{1 + (2R_1 - \alpha R_2)C\omega_c s + R_1R_2C^2\omega_c^2s^2}{1 + 2R_1C\omega_c s + R_1R_2C^2\omega_c^2s^2}$$

- $T_{gro} = 4Rf_c$
Exemple 2:
IMPLEMENTE NO LTSPICE UM FILTRO PASSA TUDO DE ORDEN 1 COM FREQUÊNCIA DE CORTE DE 1KHz. UTILIZE C=10nF.

1. Specify \( f_c \) and \( C \) and take \( a_1 \)

\( f_c = 1 \)KHz, \( C=10nF \), \( a_1 = 0,6436 \)

2. Calculate \( R \)

\[
R = \frac{a_1}{2\pi f_c \cdot C}
\]

\[
R = \frac{0,6436}{2\pi \times 10^3 \times 10 \times 10^{-9}} = 10243
\]
3 Calculate $T_{gro}$

$$T_{gro} = 2RCf_c$$

$$T_{gro} = 2 \times 10243 \times 10 \times 10^{-9} \times 1000 = 0.20486$$

![Circuit Diagram](image)
LTSPice Simulation
How to measure $\Phi$ and $T_{gr}$ in the LTSPice???

Fase ($\Phi$)

$\Phi = \arctan(\text{Im}(V(\text{out}))/\text{Re}(V(\text{out}))$)

Normalized Delay group ($T_{gr}$)

$T_{gr} = -\frac{f_c}{2\pi} \frac{d\Phi}{df}$

$T_{gr} = -(1/(2/\pi)) \cdot d(\arctan(\text{Im}(V(\text{out}))/\text{Re}(V(\text{out})))) \cdot 1000\text{Hz}$

(OBS: o * na equação ao lado deve ser o do teclado numérico). Ao exportar para o LTSPice observar todos caracteres)!
How to measure *derivada* in the LTSpice ???

There are three types of **waveform arithmetic** operations that can be performed on waveform data:

1. Plot expressions of traces.
2. Compute the average or RMS of a trace.
3. Display the Fourier Transform of a trace.
4. Plot expressions of traces.

Both the **View->Visible Traces** and **View->Add Trace** commands allow one to enter an expression of data. Another method to plot an expression of available simulation data traces is to move the mouse to the trace's label and right-click. This dialog box also allows you to set the trace's color and allows you to attach a cursor to the waveform. LTSpice will do a dimensional analysis of the expression and plot it against a vertical axis labeled with those units. For example, below you can see that LTSpice identified the dimensions (V(Vout1,2)/Abs(VVin0(0))/fetsi(2)) as Q. All waveforms in a plotting pane with the same units are plotted on the same axis.

The difference of two voltages: e.g., V(a) - V(b); can written equivalently as V(a,b). The following functions are available for real data:

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(x)</td>
<td>Absolute value of x</td>
</tr>
<tr>
<td>acos(x)</td>
<td>Arc cosine of x</td>
</tr>
<tr>
<td>acxos(x)</td>
<td>Synonym for acos()</td>
</tr>
<tr>
<td>acosh(x)</td>
<td>Arc hyperbolic cosine</td>
</tr>
<tr>
<td>asec(x)</td>
<td>Arc sec of x</td>
</tr>
<tr>
<td>asec(x)</td>
<td>Synonym for sec()</td>
</tr>
<tr>
<td>asinh(x)</td>
<td>Arc hyperbolic sine</td>
</tr>
<tr>
<td>atan(x)</td>
<td>Arc tangent of x</td>
</tr>
<tr>
<td>arctan(x)</td>
<td>Synonym for atan()</td>
</tr>
<tr>
<td>atan2(y,x)</td>
<td>Four quadrant arc tangent of y/x</td>
</tr>
<tr>
<td>atanh(x)</td>
<td>Arc hyperbolic tangent</td>
</tr>
<tr>
<td>buf(x)</td>
<td>1 if x &gt; 0.5, else 0</td>
</tr>
<tr>
<td>ceil(x)</td>
<td>Integer equal or greater than x</td>
</tr>
<tr>
<td>cos(x)</td>
<td>Cosine of x</td>
</tr>
</tbody>
</table>
How to measure *derivada* in the LTSPice ???

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(x)</td>
<td>Absolute value of x</td>
</tr>
<tr>
<td>acos(x)</td>
<td>Arc cosine of x</td>
</tr>
<tr>
<td>arccos(x)</td>
<td>Synonym for acos()</td>
</tr>
<tr>
<td>acosh(x)</td>
<td>Arc hyperbolic cosine</td>
</tr>
<tr>
<td>asin(x)</td>
<td>Arc sine of x</td>
</tr>
<tr>
<td>arcsin(x)</td>
<td>Synonym for sin()</td>
</tr>
<tr>
<td>asinh(x)</td>
<td>Arc hyperbolic sine</td>
</tr>
<tr>
<td>atan(x)</td>
<td>Arc tangent of x</td>
</tr>
<tr>
<td>arctan(x)</td>
<td>Synonym for atan()</td>
</tr>
<tr>
<td>atan2(y,x)</td>
<td>Four quadrant arc tangent of y/x</td>
</tr>
<tr>
<td>atanh(x)</td>
<td>Arc hyperbolic tangent</td>
</tr>
<tr>
<td>buf(x)</td>
<td>1 if x &gt; .5, else 0</td>
</tr>
<tr>
<td>ceil(x)</td>
<td>Integer equal or greater than x</td>
</tr>
<tr>
<td>cos(x)</td>
<td>Cosine of x</td>
</tr>
<tr>
<td>cosh(x)</td>
<td>Hyperbolic cosine of x</td>
</tr>
<tr>
<td>d()</td>
<td>Finite difference-based derivative</td>
</tr>
</tbody>
</table>
Normalized delay group ($T_{gr}$)

\[ T_{gr} = -\frac{1}{2\pi} \times \text{d}(\arctan(\text{Im}(V_{\text{out}})/\text{Re}(V_{\text{out}}))) \times 1000\text{Hz} \]

(OBS: o * na equação acima deve ser o do teclado numérico). Ao exportar para o LTSPice observar todos caracteres! 

Medida do $T_{gro}$
Normalized delay group ($T_{gr}$)

$$T_{gr} = \frac{1}{2\pi} \cdot d(\arctan(\frac{\text{Im}(V_{out})}{\text{Re}(V_{out})})) \times 1000\text{Hz}$$
Fase ($\phi$) \[\phi = \arctan(\text{Im}(V_{\text{out}})/\text{Re}(V_{\text{out}}))\]

Eixo vertical esquerdo: valor real ($r$) do arctan em função da frequência
O valor de arctan varia de 90° a -90°.

Eixo vertical direito: valor imaginário ($i$) do arctan que tem ser nulo em todas as frequências porque arctan é um número real.

Eixo vertical direito:
valor imaginário da arctag é nulo em todas as frequências.
.ac oct 100000 1 1Mega
Roteiro Experimental
Exemple 1:
A signal with the frequency spectrum, $0 < f < 1$ kHz needs to be delayed by 2 ms. To keep the phase distortions at a minimum, the corner frequency of the all-pass filter must be $f_C \geq 1$ kHz. Design a 2-ms delay all-pass filter.

The figure below shows a seventh-order all-pass is needed to accomplish the desired delay.

$T_{gro} = 2.1737$

O filtro de ordem 6 tem $T_{gro}=2$. Sendo $t_{gro}=2$ms resulta que a frequência de corte $f_c = 1$KHz. No entanto $f_C \geq 1$ kHz e, portanto, o filtro escolhido deverá ter ordem 7.

$t_{gro}=2$ms

$f_C = \frac{T_{gro}}{t_{gro}} = 1.087$ kHz
All Pass Filter – 7th Order

Equação para plotar $T_g \times f$ no LTSpice (Plot Settings – Add Trace):

$$-(1/(2\pi)) \cdot \text{d}(\arctan(\text{Im}(V(\text{Vout}))/\text{Re}(V(\text{Vout})))) \cdot 1087\text{Hz}$$
SEL393 – Laboratório de Instrumentação Eletrônica I
Escola de Engenharia de São Carlos - USP
Departamento de Engenharia Elétrica

Laboratório 3e - Filtros Ativos Passa-Tudo

Implemente em simulação um filtro passa-tudo para gerar um atraso de 2ms em um sinal com espectro de frequência $0 < f < 1$ kHz. Para minimizar a distorção de fase a frequência de corte deve ser maior que 1 kHz.

- Plote em representação Bode o atraso de grupo não normalizado $(T_{ag} \times f)$ de cada estágio do filtro. Meça a frequência de corte $(f_c)$ da $T_{ag}$ de cada estágio.

- Plote em representação Bode o atraso de grupo não normalizado $(T_{ag} \times f)$ do filtro projetado.

- Meça o $T_{ag}$ do filtro projetado e compare com o valor descrito na Tabela de parâmetros de filtro passa-tudo.