Lab_6_Passive_Filters

Paige Brady and Austin Jones

01-29-2020

[30]:
```
%pylab inline
```

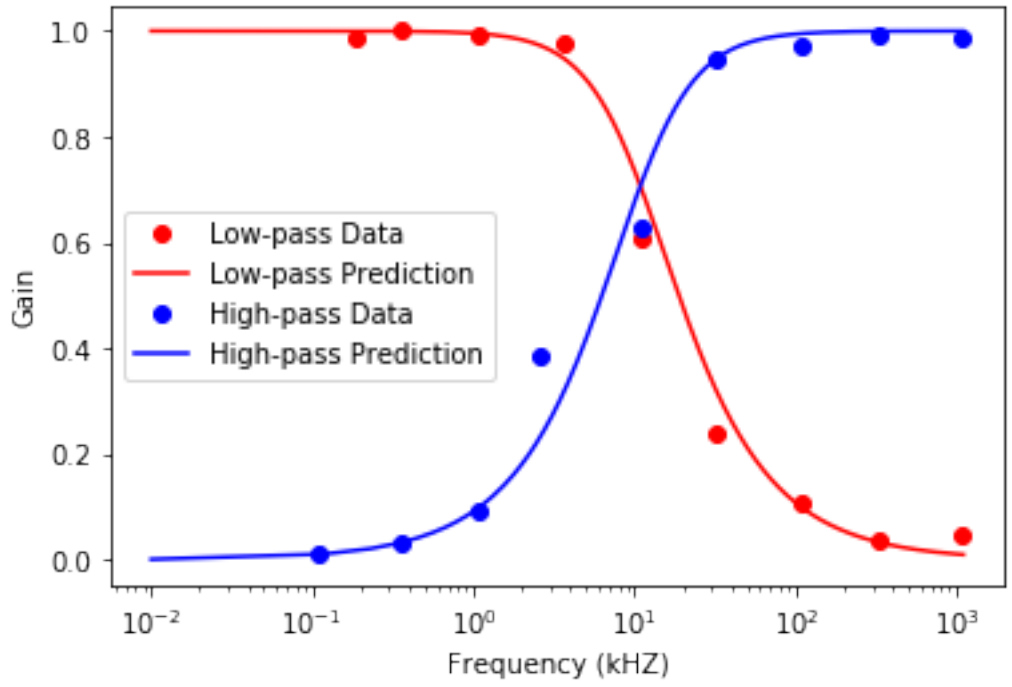
```
Populating the interactive namespace from numpy and matplotlib
```

[31]:
```
# Remark: All low-pass plots and theoretical lines are blue, and all high pass␣
 ↪plots and theoretical lines are red
```

[32]:
```
# Low-pass RMS Channel 1 V(in) (V)
x_low = np.array([1.49, 1.48, 1.47, 1.45, 1.16, 1.48, 1.38, 1.42, 1.42])

# Low-pass RMS Channel 2 V(out) (V)
y_low = np.array([1.47, 1.48, 1.46, 1.42, 0.707, 0.357, 0.148, 0.0559, 0.0632])

# Low-pass frequency (kHz)
f_low = np.array ([0.183, 0.361, 1.083, 3.61, 10.83, 32.50, 108, 325, 1083])

# Low-pass measured gain
g_low = y_low/x_low

# Corner frequency (kHz)
corner = 10.87
freq = np.linspace(0.01,1100,10000)


# Plotting measured gain as a function of the frequency for low-pass filter
plt.plot(f_low,g_low,"ro",label="Low-pass Data")
plt.xlabel("Frequency (kHZ)")
plt.ylabel("Gain")

# Low-pass Theoretical gain
G = 1/(sqrt(1+(freq/corner)**2))
plt.plot(freq,G,"r-",label="Low-pass Prediction")




# High-pass RMS Channel 1 V(in) (V)
x_high = np.array([1.44, 1.54, 1.52, 1.21, 1.51, 1.49, 1.47, 1.33, 1.38])
```

```
# High-Pass RMS Channel 2 V(out) (V)
y_high = np.array([0.0151, 0.0474, 0.140, 0.465, 0.951, 1.41, 1.43, 1.32, 1.36])

# High-pass frequency (kHz)
f_high = np.array([0.1087, 0.3623, 1.087, 2.623, 10.87, 32.61, 108.7, 326, 1087])

# High-pass measured gain
g_high = y_high/x_high

# Plotting measured gain as a function of the frequency for high-pass filter
plt.plot(f_high,g_high,"bo",label="High-pass Data")
plt.xscale("log")

# High-pass Theoretical gain
G2 = 1/(sqrt(1+(corner/freq)**2))
plt.plot(freq, G2,"b-",label="High-pass Prediction")
plt.legend()

print("Jupyter Notebook 6.1")
plt.show()
print("""Measured gain plotted against the frequency of the low-pass filter and␣
 ↪the high-pass filter frequencies
and compared to the theoretical gain calculated using the mesured values of the␣
 ↪resistor and capacitor""")
```

```
Jupyter Notebook 6.1
```

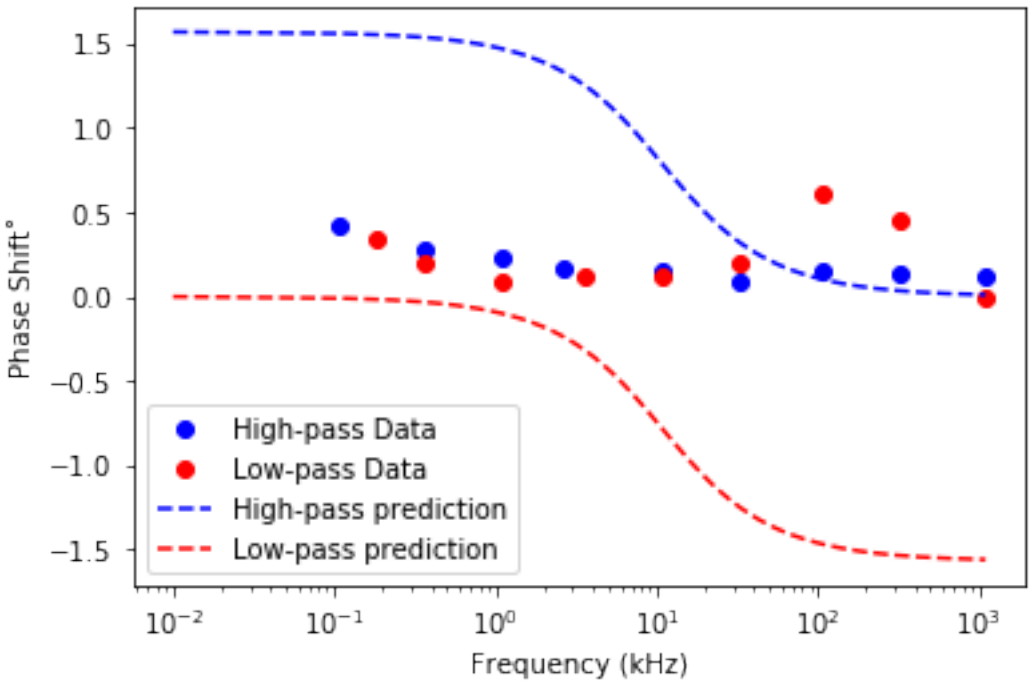


Measured gain plotted against the frequency of the low-pass filter and the high-pass filter frequencies
and compared to the theoretical gain calculated using the mesured values of the resistor and capacitor

[33]:
```
# High-pass measured phase shift (degrees)
p_high = np.array([148.7,99.1,82.9,82.9,57.1,34.0,54.0,51.6,45.4])

# Time values for high-pass (milliseconds)
t_high = np.array([3.8, 0.760,0.212,0.06360,0.01460,0.0029,0.00138,0.000440,0.
 ↪000116])

# Experimental phase shift for high-pass (degrees)
exp_phi_high = (t_high*f_high)



# Plotting measured phase shift as a function of frequency for high-pass filter
plt.plot(f_high,exp_phi_high,"bo",label="High-pass Data")
plt.xlabel("Frequency (kHz)")
plt.ylabel("Phase Shift")
plt.xscale("log")

# Low-pass measured phase shift (degrees)
p_low = np.array([71.7,72.8,32.7,44.18,44.4,69.3,220.8,163.8,0])

# Time values for low-pass (milliseconds)
t_low = np.array([1.84,0.560,0.084,0.034,0.0114,0.00592,0.00568,0.00140,0])

# Experimental phase shift for low-pass (degrees)
exp_phi_low = (t_low*f_low)


# Plotting measured phase shift as a function of frequency for low-pass
plt.plot(f_low,exp_phi_low,"ro",label="Low-pass Data")


# Theoretical phase shift for high-pass filter
theory_high = np.arctan(corner/freq)
plt.plot(freq,theory_high,"b--",label="High-pass prediction")

# Theoretical phase shift for low-pass filter
theory_low = -np.arctan(freq/corner)
plt.plot(freq,theory_low,"r--",label="Low-pass prediction")
plt.legend()
```

```
print("Jupyter Notebook 6.2")
plt.show()
print("""Measured phase shift as a function of frequency for high pass and low␣
 ↪pass filter and compare to
expected response.""")
print("""Remark: Unable to resolve issue with the plot fitting. Consulted with␣
 ↪TA Andrew Diggs and sought out of class
help from computer sciene graduate student, and it was confirmed the issue is␣
 ↪not in the coding of the plot, but
perhaps mismeasurement of the data.""")
```

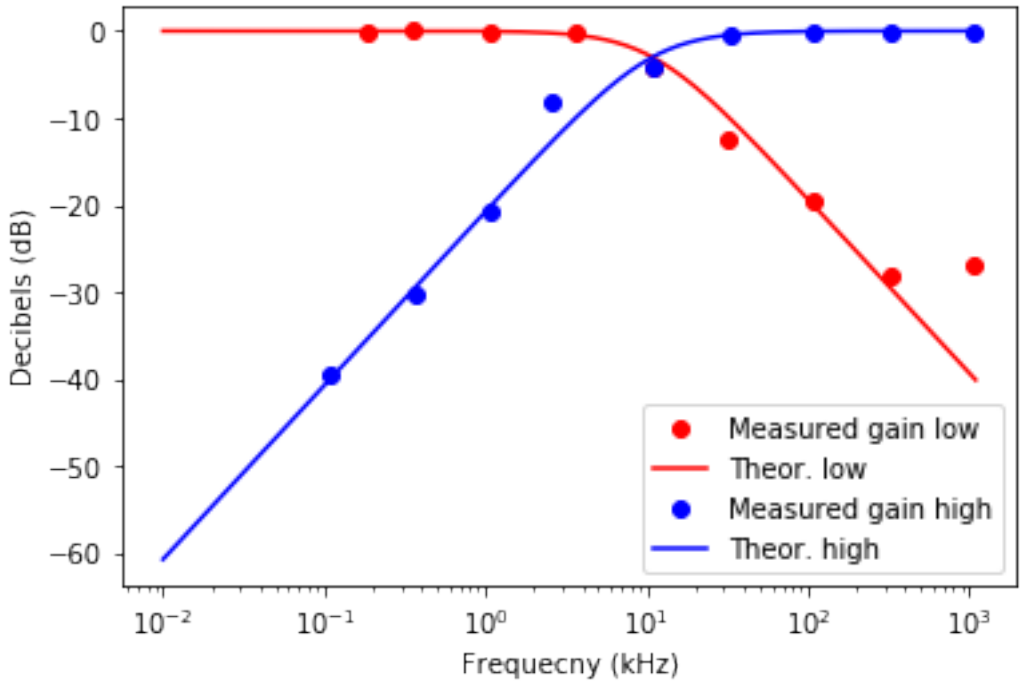
```
Jupyter Notebook 6.2
```




```
Measured phase shift as a function of frequency for high pass and low pass
filter and compare to
expected response.
Remark: Unable to resolve issue with the plot fitting. Consulted with TA Andrew
Diggs and sought out of class
help from computer sciene graduate student, and it was confirmed the issue is
not in the coding of the plot, but
perhaps mismeasurement of the data.
```

[34]:
```
# Gain in decibels as a function of frequency for high-pass and low pass filters
g_low_dB = 20*np.log10(g_low)
g_high_dB = 20*np.log10(g_high)

# Low pass theoretical gain in decibels
G_dB = 20*np.log10(G)

# High pass theoretical gain in decibels
G2_dB = 20*np.log10(G2)

# Plotting measured gain values in decibels versus frequency for low-pass filter
plt.plot(f_low,g_low_dB,"ro",label="Measured gain low")

# Plotting theoretical gain values for freqcuency for low-pass filter
plt.plot(freq, G_dB,"r-", label="Theor. low")

# Plotting measured gain values in decibels versus frequency for high-pass␣
 ↪filter
plt.plot(f_high,g_high_dB,"bo",label="Measured gain high")

# Plotting theoretical gain values for frequency for high-pass filter
plt.plot(freq,G2_dB,"b-",label="Theor. high")

plt.xlabel("Frequecny (kHz)")
plt.ylabel("Decibels (dB)")
plt.xscale("log")
plt.legend()

print("Jupyter Notebook 6.3")
plt.show()
print("""Plot measured gain in decibels as a function of frequency for high-pass␣
 ↪and low-pass filters and compare
to expected response using measured values of R and C""")
```

```
Jupyter Notebook 6.3
```




```
Plot measured gain in decibels as a function of frequency for high-pass and low-
pass filters and compare
to expected response using measured values of R and C
```

[ ]: