Lab session: acoustic device spectral characterization

S. Benchabane, J.-M Friedt, October 25, 2022

Objective

The objective of this preliminary analysis is to understand the meaning of quantities measured by a network analyzer during the spectral characterization of an unknown device: impedance, admittance, scattering coefficient, real/imaginary part or angle/magnitude. The initial steps aim at becoming familiar with these general concepts, while the **last three questions** will use this prior knowledge to analyze the electric characteristics of surface acoustic wave devices. These last three questions **are the most important part of the lab session** and must be answered.

Background

We have discussed how any resonant circuit can be modelled as a RLC electrical circuit with R the motional resistance, C the motional capacitance and L the motional inductance. When characterizing an unknown circuit, we wish to estimate the values of these parameters. Their dependence with frequency will allow us to separate each contribution, with the inductor acting as a short circuit at low frequency and an open circuit at high frequency, as opposed to the capacitor acting as a short at high frequency and an open at low frequency. At resonance, the inductor and capacitor phase shift compensate and only the resistor remains.

A network analyzer measures the (vectorial) ratio of the output to the input voltage. This ratio defines the scattering coefficient S. This scattering coefficient can be characterized in reflection (S_{11}) or in transmission (S_{21} when powering port 1 and reading port 2).

The scattering coefficient is an electrical matching coefficient that does not relate to the physical parameters of the RLC circuit. The electrical behavior of the circuit is determined by its **admittance** (or its inverse, the impedance), while the scattering parameters define the matching condition between input and output (i.e. the power transfer efficiency). Because the S-parameters are a ratio of voltages, conversion to dB requires computing $20\log_{10}(S)$, the factor 20 taking care of squaring the voltage to compute a power (normalized to the reference impedance 50 Ω).

Numerical simulation

Using a numerical simulation software (GNU/Octave, Matlab, Scilab, numpy ...), we wish to compute the impedance, admittance and scattering coefficients of a RLC circuit.
Consider a circuit with $L = 1.5$ mH, $C = 0.5$ pF and $R = 50$ Ω.

1. What is the theoretical resonance frequency of this circuit? [1]
2. What is the theoretical quality factor of this circuit? [1]
3. Plot the impedance (real and imaginary part) of this circuit in a range from 0.9 to 1.1 times the resonance frequency.
4. Plot the admittance (real and imaginary part) of this circuit in a range from 0.9 to 1.1 times the resonance frequency.
5. What is the physical meaning of the imaginary part of the admittance in the low frequency region?
6. What is the meaning of the magnitude of the impedance at resonance frequency?
7. What is the imaginary part of the impedance at resonance frequency?
8. Plot the magnitude of the scattering coefficient of this circuit. The admittance and scattering coefficient of a one-port device are related by

$$S = \frac{1 - Z_0 \times Y}{1 + Z_0 \times Y}$$

   with $Z_0 = 50$ Ω the reference impedance.
9. Plot the magnitude of the scattering coefficient of this circuit, replacing $R = 50$ Ω with $R = 150$ Ω. Compare with the previous chart and interpret.
10. A common mistake is to compute the quality factor as the width at half-height of the magnitude of the scattering coefficient. The quality factor can be retrieved by measuring the full width at half maximum of the real part of the admittance (in other terms, as the 3-dB bandwidth in log scale). Compare these two quantities – scattering coefficient magnitude width and real part of the admittance width – with the theoretical quality factor computed in the second question.

11. A practical surface acoustic wave transducer is fabricated by patterning electrodes on a piezoelectric substrate. This structure creates a capacitance in parallel to the RLC circuit: the resulting circuit is called the Buttworth-Van Dyke model, with a static capacitance C_0 in parallel to RLC. Repeat the numerical simulations (3, 4) with $C_0 = 200$ pF.

12. How does the angle of the admittance compare between RLC and Butterworth-Van Dyke?

13. How does the piezoelectric electromechanical coupling coefficient relate to C_0 and C? [1]

14. Two measurements of SEAS10 resonators sold by SENSeOR as temperature sensors have been collected on a network analyzer and are available at `http://jmfriedt.org/seas10w2.s1p` and `http://jmfriedt.org/seas10z2.s1p`. The standard format for such datasets is the Touchstone format with `snp` extension (`s1p` for a measurement in reflection, `s2p` for a measurement in transmission).

    - Find the format of this file and by reading the header analyze its content.
    - Plot the magnitude of the reflection coefficient from each file. Reading Touchstone files in GNU/Octave is easily achieved using `dlmread()` and skipping the five first lines, or with `gnuplot` whose imaginary part notation is `{0,1}` so that a complex number is written as `re*{1,0}+im*{0,1}`.
    - What is the difference between the W and Z files? Which one is most appropriate to characterize the sensor properties?

15. From the data collected, what is the quality factor of the resonance? The sensor is made of two resonances in parallel, so that two quality factors must be provided for each sensor.

16. The operator has mistakenly forgotten to calibrate the instrument before characterizing the instrument, and provides the datasets at `http://jmfriedt.org/seas10w2uncal.s1p` and `http://jmfriedt.org/seas10z2uncal.s1p`. Repeat the previous analysis and comment.

References

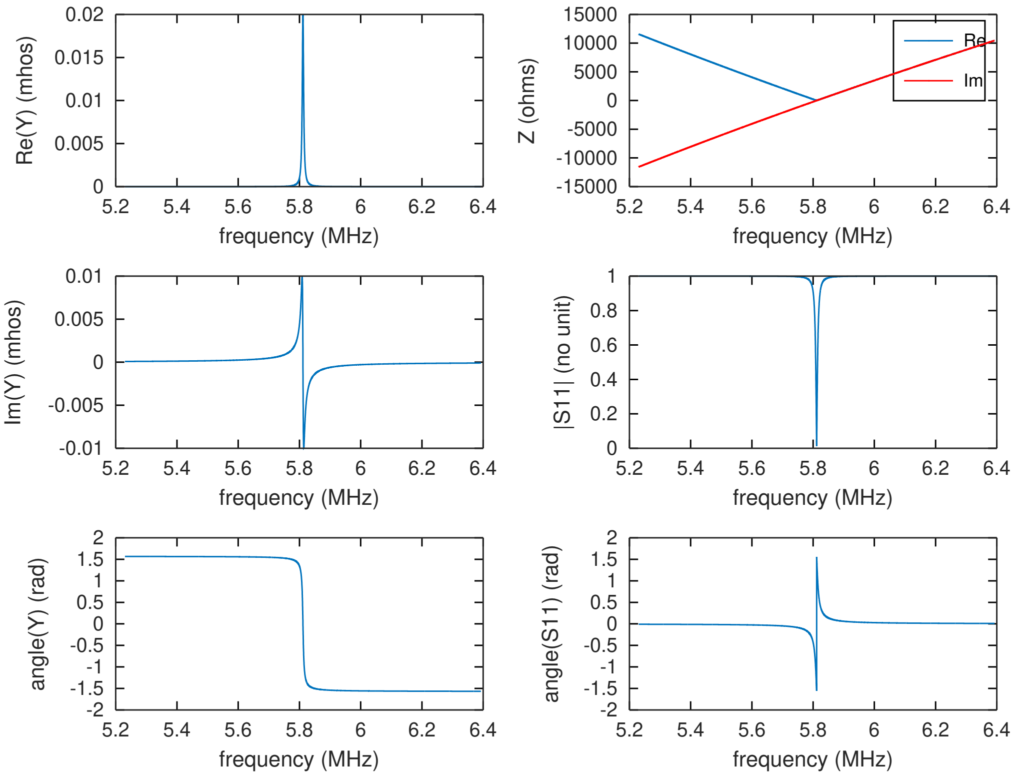
[1] J. Vig, *Quartz crystal resonators and oscillators – A tutorial*, 2000 at `https://www.am1.us/wp-content/uploads/Documents/U11625_VIG-TUTORIAL.pdf`

Answers:

1. $f_R = \frac{1}{2\pi\sqrt{LC}}$: $f \simeq 5.811$ MHz

2. $Q = \frac{1}{R}\sqrt{\frac{L}{C}}$: $Q \simeq 1095$

3.



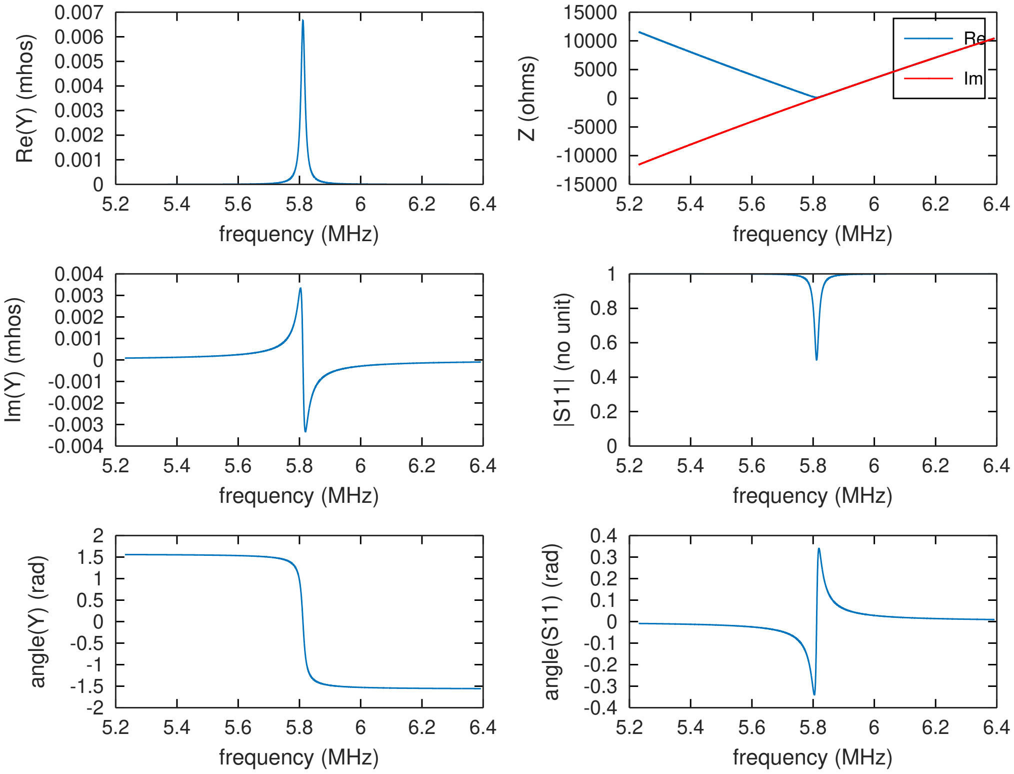
4.

5. at low frequency when the inductor is negligible and behaves as a wire, the imaginary part is $C \times w$ so allows identifying the motional capacitance

6. at resoance, the imaginary parts of the inductor and capacitor cancel and only the motional resistance R is left

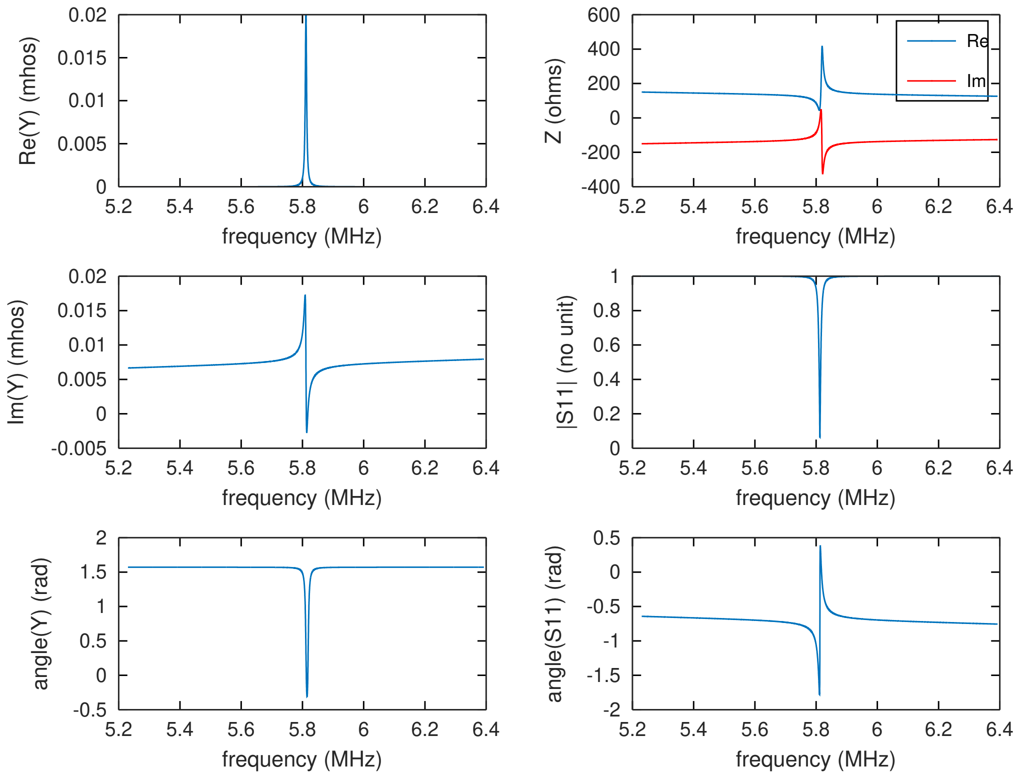
7. the imaginary part of the impedance is null at resonance

8. see above



9. The scattering coefficient no longer drops to 0 (lack of reflected power) since the device under test is mismatched with the 50 Ω load at resonance

10. $Q = 1095$ theoretically, the width at half height of the real part of the admittance is 1096 while the erroneous measurement on the scattering coefficient yields $Q = 949$. Care must be taken to select a high enough resolution on the frequency range when computing numerically Q



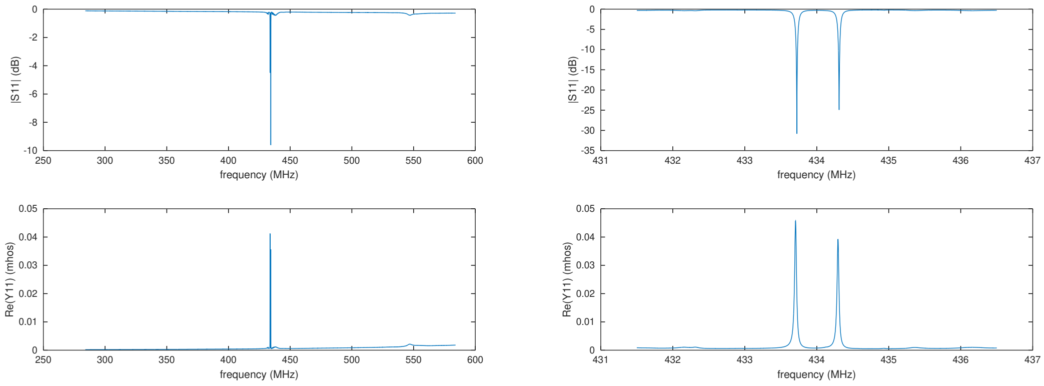
11.

12. The angle of the admittance rotated from $\pi/2$ to $-\pi/2$ at resonance in the RLC circuit. The new anti-resonance component brings the angle back to π after anti-resonance

13. the phase rotation is mandatory to achieve oscillation: ere from $\pi/2$ to $-\pi/2$

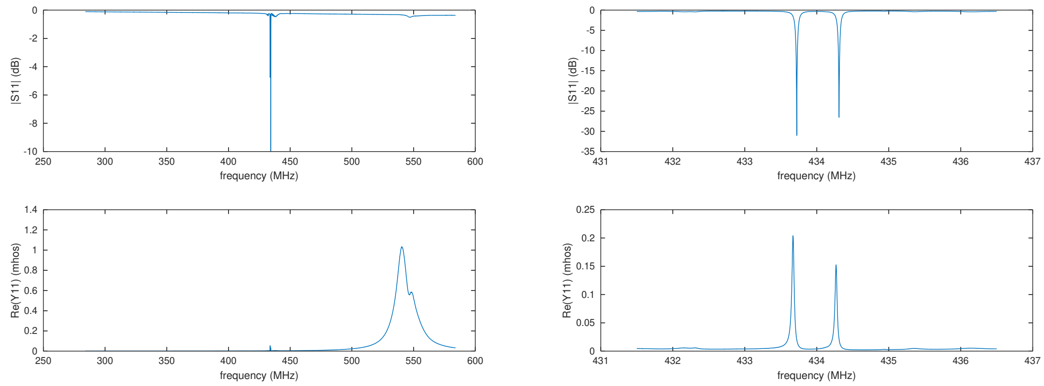
14. $K^2 \propto C1/C0$

15. W=wide, Z=zoom. The wide measurements are not suitable due to the few samples lying within the bandpass of each resonator.



16. from the zoomed measurement $Q_1 \simeq 14000$ and $Q_2 \simeq 14500$. The quality factor cannot be measured from the wideband measurement.

17. if the calibration is not applied, the reflection coefficients are qualitatively correct but the conversion to admittance is completely erroneous. The impact is most visible on the wideband measurement with the artifact around 550 MHz.



$Q_1 \simeq 12000$ and $Q_2 \simeq 12500$ are wrong, and so is the admittance at resonance

Numerical simulation software listing (GNU/Octave):

```
L=15e-4; C=5e-13;   % simulation parameters
C0=0; % C0=2e-10    % BvD parameters for simulation parameters

R=50;   % match
%R=150; % mismatch

fR=1/2/pi/sqrt(L*C)
Q=1/R*sqrt(L/C)
f=linspace(0.9*fR,1.1*fR,65536);
w=2*pi*f;

ZL=j*L*w;
ZC=1./(j*C*w);
ZR=R;

Z=ZL+ZC+ZR; % impedance of motional branch
Y=1./Z;      % admittance of motional branch

if (C0>0)     % RLC -> BvD
   df=C/2/(C0)*fR % resonance to antiresonance
   YC0=(j*C0*w);
   Y=Y+YC0;
   Z=1./Y;
end
[val,pos]=max(real(Y));df=find(real(Y)>=(val/2));df=f(df(end))-f(df(1));Qy=f(pos)/df

S11=(1-Y*50)./(1+Y*50);
[v,pos]=min(abs(S11));df=find(abs(S11)<=((1+v)/2));df=f(df(end))-f(df(1));Qs=f(pos)/df

subplot(321);plot(f/1e6,real(Y))   % Q
xlabel('frequency (MHz)');ylabel('Re(Y) (mhos)');

subplot(323);plot(f/1e6,imag(Y))   % capacite' hors resonance
xlabel('frequency (MHz)');ylabel('Im(Y) (mhos)');

subplot(322);plot(f/1e6,abs(Z))    % resistance a la resonance
```

```
hold on      ;plot(f/1e6,imag(Z),'r') % impedance matching when imag=0
hold off
xlabel('frequency (MHz)');ylabel('Z (ohms)'); legend('Re','Im')

subplot(324);plot(f/1e6,abs(S11)) % couplage electrique
ylim([0 1]);xlabel('frequency (MHz)');ylabel('|S11| (no unit)')

subplot(325);plot(f/1e6,angle(Y))  % antires
xlabel('frequency (MHz)');ylabel('angle(Y) (rad)')
%
subplot(326);plot(f/1e6,angle(S11))  % Barkhausen
xlabel('frequency (MHz)');ylabel('angle(S11) (rad)')
```

Numerical simulation software listing (Python):

```
#!/usr/bin/python3
import numpy as np
import math
import matplotlib.pyplot as plt

L=15e-3; C=5e-15;  # real parameters
C0=0; C0=2e-12     # BvD parameters for real parameters
L=15e-4; C=5e-13;  # simulation parameters
C0=0; C0=2e-10     # BvD parameters for simulation parameters

R=50;    # match
#R=150; # mismatch

fR=1/2/np.pi/math.sqrt(L*C)
Q=1/R*math.sqrt(L/C)
f=np.linspace(0.9*fR,1.1*fR,65536);
w=2*np.pi*f;

ZL=1j*L*w;
ZC=1./(1j*C*w);
ZR=R;

Z=ZL+ZC+ZR; # impedance of motional branch
Y=1./Z;     # admittance of motional branch

if (C0>0):   # % RLC -> BvD
  df=C/2/(C0)*fR # resonance to antiresonance
  YC0=(1j*C0*w);
  Y=Y+YC0;
  Z=1./Y;
# [val,pos]=max(real(Y));df=find(real(Y)>=(val/2));df=f(df(end))-f(df(1));Qy=f(pos)/df

S11=(1-Y*50)/(1+Y*50);
# [val,pos]=min(abs(S11));df=find(abs(S11)<=((1+val)/2));df=f(df(end))-f(df(1));Qs=f(pos)/df
plt.subplot(321)
plt.plot(f/1e6, Y.real)
plt.xlabel("frequency (MHz)");plt.ylabel("Re(Y) (mhos)");

plt.subplot(323);plt.plot(f/1e6,Y.imag)  # capacite' hors resonance
plt.xlabel("frequency (MHz)"); plt.ylabel("Im(Y) (mhos)");

plt.subplot(322);plt.plot(f/1e6,np.abs(Z))    # resistance a la resonance
plt.plot(f/1e6,Z.imag,'r')                    # impedance matching when imag=0
plt.xlabel('frequency (MHz)');plt.ylabel('Z (ohms)'); plt.legend(['Re','Im'])

plt.subplot(324);plt.plot(f/1e6,np.abs(S11))  # couplage electrique
plt.ylim([0 , 1]);
plt.xlabel('frequency (MHz)');plt.ylabel('|S11| (no unit)')

plt.subplot(325);plt.plot(f/1e6,np.angle(Y))  # antires
plt.xlabel('frequency (MHz)');plt.ylabel('angle(Y) (rad)')

plt.subplot(326);plt.plot(f/1e6,np.angle(S11))# Barkhausen
plt.xlabel('frequency (MHz)');plt.ylabel('angle(S11) (rad)')
plt.show()
```

Touchstone file data processing (GNU/Octave):

```
l=dir('*2*.s1p');
for k=1:length(l)
 x=dlmread(l(k).name,'',5,0);
 figure
 subplot(211)
 f=x(:,1)/1e6;df=f(2)-f(1);N=floor(.15/df) % Q>=3000
 plot(f,x(:,2))
 xlabel('frequency (MHz)');ylabel('|S11| (dB)')
 s11=10.^(x(:,2)/20).*exp(j*x(:,3)*pi/180);
 y11=1/50*(1-s11)./(1+s11);
 subplot(212)
 plot(f,real(y11),'-')
 xlabel('frequency (MHz)');ylabel('Re(Y11) (mhos)')
 if (N>2)
  [a,b]=max(real(y11));
  k=find(real(y11(b-N:b+N))>a/2);
  Q=f(b)/(f(k(end))-f(k(1)))
  y11(b-N:b+N)=0;
  [a,b]=max(real(y11));
  k=find(real(y11(b-N:b+N))>a/2);
  Q=f(b)/(f(k(end))-f(k(1)))
 end
end
```

Touchstone file data processing (Python):

```python
#!/usr/bin/python3
import numpy as np
import matplotlib.pyplot as plt

x=np.loadtxt('seas10z2uncal.s1p',skiprows=5)
s=10**(x[:,1]/20)*np.exp(1j*x[:,2]*np.pi/180);
plt.subplot(211)
plt.plot(x[:,0]/1e6,20*np.log10(abs(s)));
yu=1/50*(1-s)/(1+s)

x=np.loadtxt('seas10z2.s1p',skiprows=5)
s=10**(x[:,1]/20)*np.exp(1j*x[:,2]*np.pi/180);
plt.plot(x[:,0]/1e6,20*np.log10(abs(s)))
plt.ylabel('|S11| (no unit)');plt.xlabel('frequency (MHz)')
yc=1/50*(1-s)/(1+s)
plt.subplot(212)
plt.plot(x[:,0]/1e6,yu.real);
plt.plot(x[:,0]/1e6,yc.real)
plt.ylabel('Re(Y) (mhos)');plt.xlabel('frequency (MHz)')
plt.show()
```