AbraconCrystalError

July 9, 2018

In [5]:
```
import matplotlib.pyplot as plt
import numpy as np

plt.rcParams['figure.dpi'] = 600
```

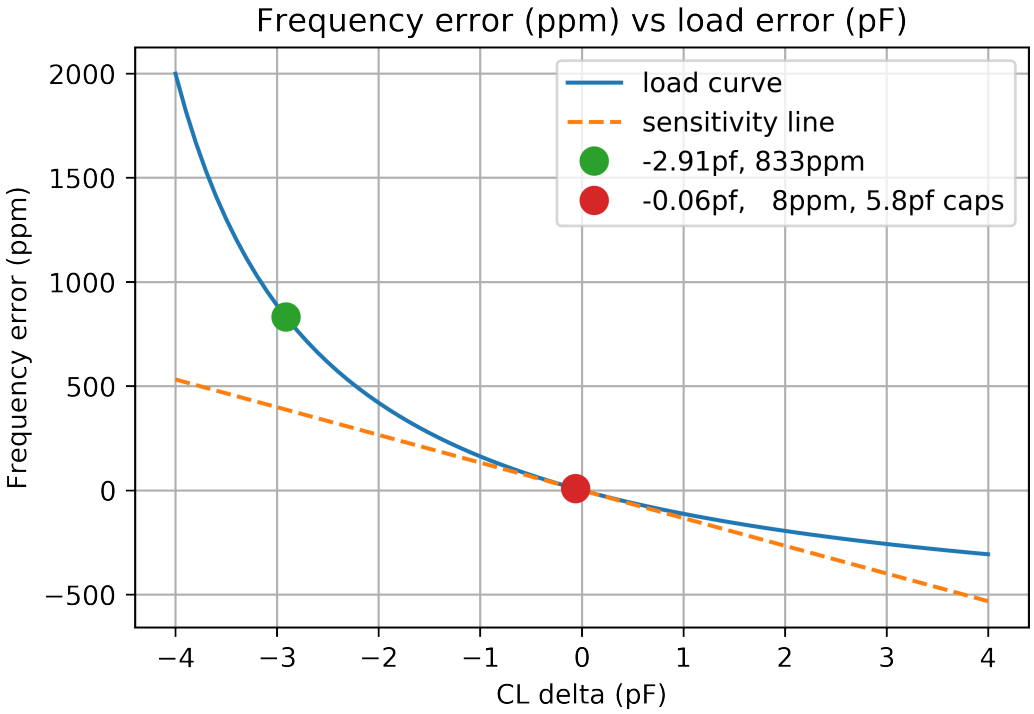
In [6]:
```
# C1, C0 are crystal parameters (from crystal datasheet) CLn is the
# specified load.
#
# deltaCL is the load capacitance offset from CLn in ppm
# output is frequency error in ppm
# equation 6 from IDT "the crystal load curve" AN-831
#

def Fn(deltaCL, C1, C0, CLn):
    deltaCL = deltaCL * 1.0E-12
    T1 = (C1/(C0 + C1 + CLn))
    T2 = (deltaCL/(C0+CLn))
    return -0.5*1.0E6*T1*T2/(1+T2)

#
# the traditional sensitivity function -- a linearization
# of the preceding.
#

def FnSens(deltaCL, C1, C0, CLn):
    deltaCL = deltaCL * 1.0E-12
    T1 = (C1/(C0 + C1 + CLn))
    T2 = (deltaCL/(C0+CLn))
    return -0.5*1.0E6*T1*T2

#
#  calculate the deltaCL needed for a given frequency error
#  equation 8
#
#  F = desired frequency (hz)
#  Fn = frequency error (hz)
#  Ferr = Fn/F
```

```
        def FnDeltaCL(Ferr,C1, C0, CLn):
            T1 = (C0 + CLn)
            return 2*(T1*T1/C1)*Ferr/(1 + 2*Ferr*T1/C1)
```

In [7]:
```
        # Abracon ABS05W 32khz, 4pf load crystal parameters from data sheet

        C1 = 7.91E-15
        C0 = 1.45E-12

        # The expected load

        CLn = 4.0E-12

        # Test range

        tr = np.arange(-4.0,4.1,0.1)
```

In [8]:
```
        vecFn = np.vectorize(Fn)
        vecFnSens = np.vectorize(FnSens)
        t = tr
        s = vecFn(t,C1,C0,CLn)
        s2 = vecFnSens(t,C1,C0,CLn)
        plt.plot(t, s, label='load curve')
        plt.plot(t, s2, label='sensitivity line',linestyle='--')
        plt.xlabel('CL delta (pF)')
        plt.ylabel('Frequency error (ppm)')
        plt.title('Frequency error (ppm) vs load error (pF)')
        plt.grid(True)

        # the following points were chosen to match a current
        # configuration
        # Ferr

        Ferr = (32795.3/32768)-1
        t1 = -FnDeltaCL(Ferr,C1,C0,4E-12)*1.E12
        s1 = (Ferr)*1E6

        Ferr2 = (32768.27/32768)-1
        t2 = -FnDeltaCL(Ferr2,C1,C0,4E-12)*1.E12
        s2 = (Ferr2)*1E6

        plt.plot(t1,s1, 'o', label='{:3.2f}pf, {:3.0f}ppm'.format(t1,s1), markersize=10)

        plt.plot(t2,s2, 'o', label='{:3.2f}pf, {:3.0f}ppm, 5.8pf caps'.format(t2,s2), markersi

        plt.legend()
        plt.savefig("loaderror.png")
```

```
plt.show()
```


Frequency error (ppm) vs load error (pF)
load curve
sensitivity line
-2.91pf, 833ppm
-0.06pf, 8ppm, 5.8pf caps
Frequency error (ppm)
CL delta (pF)
2000
1500
1000
500
0
−500
−4
−3
−2
−1
0
1
2
3
4